IMAGE REGISTRATION UNDER CONFORMAL DIFFEOMORPHISMS

A thesis presented in partial fulfilment of the requirements for the degree of
Doctor of Philosophy
in
Mathematics
at Massey University, Palmerston North,
New Zealand.

Muhammad Yousuf Tufail

2017

Contents

Declaration **xviii**

Acknowledgements **xix**

1 Introduction **1**

1.1 Image registration . . . 1

1.1.1 Brief Overview of the Literature . . . 2

1.2 Definitions . . . 5

1.2.1 Continuous and discrete images . . . 5

1.2.2 Diffeomorphism . . . 5

1.2.3 Conformal diffeomorphism . . . 5

1.2.4 Action of diffeomorphisms on images . . . 6

1.3 Applications of image registration . . . 8

1.3.1 Morphometrics . . . 8

1.3.2 Medical imaging . . . 9

1.3.3 Computer vision . . . 10

1.3.4 Remote sensing . . . 10

1.3.5 Visual Cortex . . . 11

1.4 Motivation . . . 11

1.5 Brief overview of the thesis . . . 18

2 Finite Dimensional Image Registration **20**

2.1 Image Registration in Practice . . . 20

2.1.1 Missing values in image registration . . . 22

2.1.2 Image interpolation . . . 23

2.2 Image registration using the rigid group . . . 25

2.2.1 Using coarse search . . . 29

2.2.2 Using gradient descent . . . 33

2.3 Möbius registration . . . 35

2.3.1 Image Registration with the Möbius Group . . . 37

3 Method of Control Points 45

3.1 Introduction 45

3.1.1 Smoothing 46

3.1.2 Image Registration Using a Conformal Diffeomorphism 47

3.2 Control points method 48

3.3 The Cauchy–Riemann equations 53

3.4 First discrete form of the Cauchy–Riemann equations 54

3.4.1 First form with smooth images 56

3.4.2 First form with non-smooth images 60

3.5 Second and third discrete forms of the Cauchy–Riemann equations . . . 68

3.5.1 Second/Third form with smooth images 72

3.5.2 Second/third form with non-smooth images 74

3.6 Fourth form of the penalty term 80

3.6.1 Fourth form with smooth images 81

3.6.2 Fourth form with non-smooth images 84

4 Gradient Flow 94

4.1 A Gradient Flow Algorithm for Conformal Diffeomorphisms 95

4.1.1 Implementation of the gradient flow method 98

4.2 Image registration with smooth images 100

4.3 Image registration with non-smooth images 108

5 Experiments and Comparison 120

5.1 Comparing the Algorithms 147

6 Conclusions and Future Work 149

6.1 Conclusion 149

6.2 Future work 150

List of Tables

1.1 The transformation groups relevant to this thesis. 2
1.2 This table suggests the connection between the transformations group and Thompson's images in [101]. This table is is taken from [66]. 13
1.3 *Top:* Four positions are marked on middle finger. *Bottom:* Numerical results of the computation of the cross-ratio of points on the middle finger in man at different ages, taken from [81]. 15
1.4 *Top:* Numeric results of cross-ratios for postnatal stages are given. *Bottom:* Numeric results of cross-ratios for antenatal stages are provided. These numerical results are taken from [81] 16
1.5 Computation of cross-ratio at (marked) position of the middile finger (see the figure of a human hand in Table 1.3). 18

3.1 Registration errors and penalty terms for Example 3.7. Here the same registration error as in all other tables is used: $RE=||(I_1 \circ \varphi^{-1})(x_{ij}) - I_2(x_{ij})||$. However, we refer to it by name to save space. P_2 is the second (linear) penalty term given in Equation (3.15), and P_3 is the third (nonlinear) penalty term given in Equation (3.19). 73
3.2 Registration errors and penalty terms for Example 3.8. 74
3.3 Registration errors and penalty terms for Example 3.9. (The images are smoothed in the first 5 rows, and unsmoothed in the last row.) 76
3.4 Registration errors and penalty terms for Example 3.10. (The images are smoothed in the first 5 rows, and unsmoothed in the last row.) . . . 77
3.5 Registration errors and penalty terms for Example 3.11. (The images are smoothed in the first 5 rows, and unsmoothed in the last row.) . . . 78

5.1 The Taylor series (for Example 5.4) found by four different runs. Each run used an initial guess given by the results of the $n = 4$ registration plus a small random perturbation (independent normal random variates with mean 0 and standard deviation 0.04). While the algorithm has located a local minimum in each case, the minima are all different, and the coefficients a_n for $n \geq 4$ carry essentially no information about the image registrations. 131

List of Figures

1.1 A set of discrete images. *Left column:* 2-D and 3-D colour images. *Right:* greyscale images. 6

1.2 *Left:* A uniform grid (100×100) is given. *Right:* The image of the uniform grid under the conformal diffeomorphism $\varphi(z) = \frac{az+b}{cz+d}$, $z \in [-0.5, 0.5] \times [-0.5, 0.5]$, where $a = 0.4i, b = 0.2 + 0.4i, c = 0.2i, d = 1$). . 7

1.3 Position $y \in \Omega$ in the pixel grid is mapped under φ to position $x = \varphi(y)$ in the transformed grid. In order to obtain the intensity of x it is necessary to invert φ and recover $y = \varphi^{-1}(x)$. 7

1.4 Examples of Thompson's transformations between related species. These images are taken from [7] (p. no. 404). 12

1.5 These images of skulls are taken from Thompson's book [101] 12

1.6 *Top:* Images of foot bones of an ox, a sheep and a giraffe. Four positions are marked in each image. These images are taken from [101]. *Bottom:* Distances and cross-ratios between the foot bones of an ox, a sheep and a giraffe. 14

1.7 Scanned images of [81] (a) An adult, (b) a five year old and (c) a newborn. 15

1.8 *Left:* Antenatal stages (in lunar months). *Right:* Postnatal stages (in years). These images are taken from [81]. 16

1.9 *Top:* Four positions are marked on each image showing the appearance of a human at different ages. *Below:* Numerical results for the cross-ratios are given (taken from [73]). 17

2.1 Set of images indicating a perfect registration in which a transformation φ is applied to the source and the transformed source is apparently identical to the destination image or target. This transformed source is subtracted from the target (using Equation (2.1)) and a mid-grey screen is obtained which is labelled as 'Difference'. 21

2.2 Set of images indicating a poor registration because the transformed source is not aligned with the target. In fact, the transformed source is rotated in the wrong direction. The poor registration is demonstrated by the fact that the difference image is not blank. 21

2.3 Missing values are shown in a pair of images that are obtained after two different rotations of the source image of Figure 2.1. These missing values have been set to 0 (black). 22

2.4 A uniform grid represents sixteen grid points in which four known values of a function f are marked with red circles corresponding to a set of grid points. A point 'P' at which f needs to be determined is marked with a blue circle. Two green circles represent two points P_1 and P_2 on a vertical line passing through a point P at which the function value is calculated (with the help of linear interpolation) before its computation at a point P. 24

2.5 The LOF caption 25

2.6 Square grid of 10,000 points in the range $[-0.5, 0.5] \times [-0.5, 0.5]$ with $N = 100$, i.e., 100 points per side. 28

2.7 Images for Example 2.2. In the first row, the source image, the Gaussian $\exp(-5x^2 - 3y^2)$, is shown on the left; the target image, a rigid transformation of the source, is shown on the right. The second row contains corresponding contour plots of these images. 30

2.8 The LOF caption 31

2.9 Coarse search results for Example 2.2. All of the parameter combinations tested are listed along the x-axis, with the registration errors plotted on the y-axis. The red circle indicates the smallest minimum that the coarse search optimisation obtained. 31

2.10 The non-smooth images used in Example 2.3. The target is a rigid transformation of the source. 32

2.11 *Top:* Results of rigid registration using coarse search for Example 2.3. A perfect registration is obtained. *Bottom:* All values of the objective function for coarse search optimisation for Example 2.3. 32

2.12 The non-smooth images used in Example 2.4. The target is a rigid transformation of the source. 33

2.13 *Top:* Results of rigid registration for Example 2.4. A satisfactory, but imperfect, registration is obtained from the coarse search. *Bottom:* All values of the objective function for coarse search optimisation for Example 2.4. Note that the y-axis only goes down to 20, not to 0 as in the previous Examples. 33

2.14 Results of least-squares rigid registration for Example 2.4 using *lsqnonlin* with initial guess the identity. 34
2.15 Results of least-squares rigid registration for Example 2.4 using *lsqnonlin* with initial guess the best result of coarse search. A perfect registration is obtained. 35
2.16 Data for Example 2.5. The source is a Gaussian and the target is a Möbius transformation of the source. 37
2.17 Results of Möbius registration for Example 2.5, with *lsqnonlin* and initial guess the identity. A perfect registration is obtained. 38
2.18 Data for Example 2.6, Möbius registration with synthetic non-smooth images. The mapping that generates the source from the target is also shown. 38
2.19 Results of first attempt at Möbius registration for Example 2.6. The desired mapping and the mapping computed by *lsqnonlin* are also shown; the results are poor. 39
2.20 Selection of corresponding landmarks on both the fish images. 40
2.21 Results of the second attempt at Möbius registration for Example 2.6, using an initial guess calculated from landmark matching. A perfect registration is obtained. 40
2.22 Data for Example 2.7 of Möbius registration, a cartoon version of Thompson's fish. 41
2.23 Selection of the first set of landmarks on both the images. 41
2.24 Results of Möbius registration for Example 2.7: first set of landmarks. . 42
2.25 Selection of a second set of landmarks on both the images. 42
2.26 Results of Möbius registration for Example 2.7: second set of landmarks. 42
2.27 Second dataset for Example 2.7: swapped source and target. 43
2.28 Selection of corresponding landmarks: first set. 43
2.29 Results of Möbius registration for Example 2.7: second dataset, first set of landmarks. 43
2.30 Selection of corresponding landmarks: second set. 44
2.31 Results of Möbius registration for Example 2.7: second dataset, second set of landmarks. 44

3.1 *Left:* Example smooth images. *Right:* Corresponding contour plots of the smooth images. 46
3.2 Examples of non-smooth images. 47
3.3 Smoothed images from Figure 3.2 using a 15 × 15 Gaussian filter with standard deviation of 5 pixels. 47
3.4 Grid of 10,000 points with $N = 100$. 48

3.5 Blue circles represent selected control points from the discrete domain S. 49

3.6 Red circles represent the transformed control points under the action of some φ^{-1}. 50

3.7 Bilinear interpolation over the transformed control points, marked with red circles, was used to generate this grid. 50

3.8 In the first grid, the blue circles represent selected control points from the discrete domain. In the second grid, the images of the control points under φ^{-1} are shown with red circles. Bilinear interpolation is used to generate the rest of the grid points between these red circles. 51

3.9 The source and target images for Example 3.1 are shown as greyscale images in the first row and as contour plots in the second row. The source is $I_1(x, y) = \exp(-7x^2 - 2y^2)$, and the target is defined by $I_2 = I_1 \circ \varphi^{-1}$, where $\varphi^{-1}(z) = 0.1 + z + 0.3z^2$. 57

3.10 The results of conformal image registration using Algorithm 2 applied to Example 3.1. The source and target images are shown, together with the final transformed source, which nearly matches the target on the part of the domain on which it is defined. The difference $I_2 - I_1 \circ \varphi^{-1}$ is shown in greyscale, where uniform mid-grey indicates a perfect match. The grid of the transformed control points, i.e. $\varphi^{-1}(\hat{x}_{ij})$ (we call it the deformation grid) and the contour plots of target and transformed source are shown on the right. In the table, each row gives the registration error and the value of the penalty term for a step of Algorithm 2. 57

3.11 Four grids are shown that are corresponding to $\lambda = 20^4, 20^3, 20^2, 20$ respectively for sixteen control points for Example 3.1. The two grids in the first row are rigid because squares are mapped to squares under φ^{-1}; the third and fourth grids, with smaller values of λ, are nonrigid. The deformation grid corresponding to $\lambda = 20^2$ shows asymmetry, which we are unable to explain. 58

3.12 The source and target images for Example 3.2. The source is $\exp(-5x^2 - 7y^2)$, the target is $\exp(-5x^2 - 2y^2)$. The corresponding contour plots can be seen in the second row. 59

3.13 The results of conformal registration with Algorithm 2 applied to Example 3.2, showing a near-perfect registration because some of the red contours are slightly mismatched with the corresponding green contours. 59

3.14 The images for Example 3.3. The source I_1 is a standard reference image; the target is $I_2(z) = I_1(0.1 + 1.2z + 0.3z^2)$ and is therefore known to be conformally related to the source. 60

3.15 The results of the first attempt at registration for Example 3.3. An extremely poor registration is obtained. 61
3.16 Left: the original set of control points; right: the same points with a small perturbation added to each point independently in x and y. . . . 61
3.17 The LOF caption 62
3.18 Two non-smooth independent images of ellipses are used as the source and target respectively. 62
3.19 The results of the first attempt at registration for Example 3.4. The optimiser remains stuck at the initial condition, the identity. 63
3.20 The results of the second attempt at registration for Example 3.4. The registration is still not successful even after the control points are perturbed. 64
3.21 Left: the image matching results for Example 3.4 initialised with smoothed versions of the images. The top row shows the source and the target, while the transformed source and the difference between the transformed source and the target can be seen in second row. Right: the deformation grid $\varphi^{-1}(\hat{x}_{ij}), \hat{x}_{ij} \in S$ with $\lambda = 20$. In the table below, each row gives the registration error and value of penalty term P_1 for each step of Algorithm 2. In the first 5 rows the source and target images are smoothed by a Gaussian filter with standard deviation 5 pixels; in the last row they are not smoothed. 64
3.22 Modified versions of the images of Thompson's fish are shown. The first image serves as the source and the second image as the target for image registration. 65
3.23 The LOF caption 66
3.24 The LOF caption 66
3.25 The LOF caption 67
3.26 In both the figures, the three red circles represent three complex numbers. The blue circles represent the fourth complex number for the linear (left) and nonlinear (right) forms. 71
3.27 Results of conformal image registration for Example 3.7 using the linear discrete Cauchy–Riemann equations as a penalty term. 72
3.28 Results of conformal image registration for Example 3.7 using the nonlinear discrete Cauchy–Riemann equations as a penalty term. 73
3.29 The grids in Example 3.7 corresponding to $\lambda = 20^4, 20^3, 20^2, 20$ for (left) the linear form of the penalty term and (right) for the nonlinear form. . 74
3.30 Results of conformal image registration for Example 3.8 using the linear discrete Cauchy–Riemann equations as a penalty term. 75

3.31 Results of conformal image registration for Example 3.8 using the nonlinear discrete Cauchy–Riemann equations as a penalty term. 75
3.32 Results of conformal image registration for Example 3.9 using the linear discrete Cauchy–Riemann equations as a penalty term. 75
3.33 Results of conformal image registration for Example 3.9 using the nonlinear discrete Cauchy–Riemann equations as a penalty term. 76
3.34 Results of conformal image registration for Example 3.10 using the linear discrete Cauchy–Riemann equations as a penalty term (top) and the nonlinear form (bottom). 77
3.35 Results of conformal image registration for Example 3.11 using the linear discrete Cauchy–Riemann equations as a penalty term (top) and the nonlinear form (bottom). Note that the size of the top grid is $[-8,-1]\times[1,4]$ which is far away from the domain $[-0.5,0.5]\times[-0.5,0.5]$. However, the majority portion of the bottom grid is within the domain. 78
3.36 *Left:* A uniform square grid of size sixteen is presented. *Right:* Its image under the conformal map $\varphi^{-1}(z) = 1.5 + 0.2z - 0.4z^2 + 0.1z^3 - 0.1z^4$. . . 80
3.37 The results of conformal image registration with the fourth penalty term applied to the images in Example 3.12. 82
3.38 Four grids are shown that were calculated for Example 3.12 and correspond to $\lambda = 20^4, 20^3, 20^2, 20$ respectively for sixteen control points. The first three grids in the first row represent perfect rigid grids because squares are mapped to squares in each grid without showing any deformation. The fourth grid is a non-rigid conformal grid. 82
3.39 Results of conformal image registration applied to Example 3.37 with the fourth penalty term and $\lambda = 150$. 83
3.40 The LOF caption 83
3.41 Results of conformal image registration for Example 3.14 with the fourth penalty term and $\lambda = 150$. 84
3.42 Results of conformal image registration for Example 3.15 with the fourth penalty term with $\lambda = 150$ are given. 85
3.43 *Top:* A series of transformed sources as λ decreases geometrically. *Bottom:* The corresponding grids. 86
3.44 *Top:* Series of transformed sources are given when λ is increasing geometrically. *Bottom:* Series of grids are presented corresponding to transformed sources. A striking global change in the grid is observed between $\lambda = 200$ and $\lambda = 300$. 87
3.45 L-curve for Example 3.15. 88

3.46 *Top:* Deformation grid of Example 3.4 (which used the first form of the penalty term); red circles indicate the non-smooth or non-conformal parts. *Bottom:* Deformation grid of Example 3.15 using the 4th form of the penalty term. 89
3.47 Results of conformal image registration for Example 3.15 with the fourth penalty term with $\lambda = 150$. 90
3.48 *Top:* A series of transformed sources as λ decreases geometrically for Example 3.16. *Bottom:* The corresponding grids. A marked deterioration in the grid is observed at the smallest values of λ. 91
3.49 *Top:* A series of transformed sources as λ increases geometrically for Example 3.16. *Bottom:* The corresponding grids. 92
3.50 Red lines shows the path of error while λ is decreasing. On the other hand, path of error function while λ is increasing is given with the green line. $\lambda = 150$ is a threshold point where both paths coincide. 93

4.1 *Top:* The source, the ellipse $I(x, y) = \exp(-7x^2 - 2y^2)$ is shown on the left, while the right shows the target, which is generated from the source using $\varphi^{-1} = \Sigma_{k=0}^{2} a_k z^k$ using coefficients $a_0 = 0.3, a_1 = 0.8, a_2 = 0.4$. *Bottom:* The corresponding contour plots. 101
4.2 Registration with target generated from smooth source. *Left:* The top row shows the source and target, while the transformed source and the difference between the transformed source and the target can be seen in the second row. *Right:* A deformation grid $\varphi^{-1}(z_{ij}), \forall z_{ij} \in S$ is shown on top and the corresponding contour plots of the transformed source and the target are on the bottom. 102
4.3 The error graph shows the successful convergence of Algorithm 3 (with $K = 3$) for Example 4.1. 102
4.4 *Top:* Two ellipses for Example 4.2, the source $I_1 = \exp(-7x^2 - 2y^2)$, and the target $I_2 = \exp(-6.5x^2 - 1.5y^2)$. *Bottom:* The corresponding contour plots of the source and the target images. 103
4.5 Registration with independent smooth images for Example 4.2. *Left:* The top row shows the source and target images, while the bottom row shows the transformed source and the difference image. *Right:* The deformation grid $\varphi^{-1}(z_{ij}), \forall z_{ij} \in S$ on top and the corresponding contour plots on the bottom. 103
4.6 The algorithm successfully found a local minimum in Example 4.2. Note that the y-axis on this plot does not go down to 0, but stops at 1. Axis extrema are selected to make the plot as clear as possible for all of these error plots. 104

4.7 *Top:* Two ellipses, the source $I_1 = \exp(-7x^2 - 2y^2)$, and the target $I_2 = \exp(-4x^2 - y^2)$ respectively for Example 4.3. *Bottom:* The corresponding contour plots of the source and the target images for the same example. 104

4.8 Registration results for Example 4.3 with four terms. Successful convergence with good registration is obtained. 105

4.9 Registration results for Example 4.3 with eight terms. The final value of the objective function is rather lower than with four terms. 106

4.10 Registration results for Example 4.3 with twelve terms. The final value of the objective function is not much better than with eight terms. . . . 107

4.11 Set of images for Example 4.4. *Left:* The source image. *Right:* The target image, which is generated from the source using $\varphi^{-1} = \Sigma_{k=0}^{2} a_k z^k$ with coefficients $a_0 = 0.1\mathrm{i}, a_1 = 1.5, a_2 = 0.2$. 108

4.12 Registration with dependent, non-smooth images for Example 4.4 with $K = 3$. *Left:* The top row shows the source and target images, while the bottom row shows the transformed source and the difference image. *Right:* The deformation grid $\varphi^{-1}(z_{ij}), \forall z_{ij} \in S$. 109

4.13 Plots of the error function for Example 4.4 with three terms of the Taylor series. *Left:* Smoothed images (first step). *Right:* Original images (second step). 109

4.14 Four sets of image registrations (along with their corresponding error graphs) for Example 4.4 are given. Algorithm 3 is run with four different stopping criteria (for $K = 3$ in each case) based on a common ratio $(r = 20\sqrt{3})$. Results (c) and (d) display the successful convergence of Algorithm 3 along with good registration, whereas set (a) has clearly stopped too early. Note that the runs lower down take many more steps, this is particularly clear for the registration of the smoothed image in (d).110

4.15 *Top:* An ellipse with axes 0.8 units and 0.4 units (on the left) serves as the source, while another ellipse of axes 0.6 units, 0.4 units (on the right) serves as the target. *Bottom:* Corresponding contour plots of the source and the target respectively. Note that the background is not perfectly white (which has caused the artefacts in the contour plot) because of the resizing and resampling of the images. 111

4.16 *Top:* Registration with four terms using conformal gradient flow on independent images related by a conformal transformation (Example 4.5). *Bottom:* Error plots for smoothed (left) and original (right) images. . . 112

4.17 *Top:* Registration with eight terms using conformal gradient flow on independent images related by a conformal transformation (Example 4.5). *Bottom:* Error plots for smoothed (left) and original (right) images. The final error is little different to the run with four terms. 113

4.18 *Top:* Registration with twelve terms using conformal gradient flow on independent images related by a conformal transformation (Example 4.5). *Bottom:* Error plots for smoothed (left) and original (right) images. The final error is better than those with four or eight terms. 114

4.19 *Top:* A circle of radius 0.5 units serves as the source (on the left). The ellipse on the right has axes 0.5 units and 0.3 units and serves as the target. *Bottom:* Corresponding contour plots of the source and the target respectively. 115

4.20 *Top:* Registration using conformal gradient flow with four terms in the Taylor series of images of a circle and ellipse. *Bottom:* Error graphs for smoothed (left) and non-smoothed (right) images. 116

4.21 *Top:* Registration using conformal gradient flow with eight terms in the Taylor series of images of a circle and ellipse. *Bottom:* Error graphs for smoothed (left) and non-smoothed (right) images. 117

4.22 *Top:* Registration using conformal gradient flow with twelve terms in the Taylor series of images of a circle and ellipse. *Bottom:* Error graphs for smoothed (left) and non-smoothed (right) images. 118

5.1 *Top:* The source and target for Example 5.1. *Bottom:* Corresponding contour plots. 121

5.2 *Left:* Results of image registration for Example 5.1 using the control points method with the 4th penalty term and $\lambda = 150$. Images are smoothed for first two rows and non-smooth for the last row. Note that the algorithm has found a spurious rotation of the circle as well as the correct scaling. There is nothing to penalise this rotation in the registration. 121

5.3 Results of image registration with the gradient flow method for Example 5.1. *Top left:* The source and the target images, the transformed source, and the difference image. *Top right:* The final mapping $\varphi^{-1}(x_{ij})$ and the contour plots of the transformed source and the target. *Bottom:* The progress of the gradient descent algorithm for the smoothed (left) and original (right) images. 122

5.4 *Left:* The source and target images for Example 5.2 are shown in first row. The source is $I_1(x,y) = \exp(-7x^2 - 2y^2)$, and the target is defined by $I_2 = I_1 \circ \varphi^{-1}$, where φ^{-1}, where $\varphi^{-1}(z) = 0.1 + z + 0.2z^2 + 0.5\mathrm{i}\bar{z}^2$. Corresponding contour plots are given in the second row. *Right:* The non-conformal mapping that generates the target. 123
5.5 *Top:* Results of image matching for Example 5.2 with the control points method. *Bottom:* Numerical results of the registration for $\lambda = 150$. . . . 123
5.6 Results of conformal registration using the gradient flow method (with four terms) for Example 5.2. 123
5.7 Progress of the gradient descent algorithm towards finding a minimum for Example 5.2. 124
5.8 *Left:* The source and target images for Example 5.3 are given. The target is defined by $I_2 = I_1 \circ \varphi^{-1}$, where $\varphi^{-1}(z) = 0.1 + z + 0.2z^2 + 0.1\bar{z}^3$. *Right:* The non-conformal map that generates the target. 124
5.9 Results of conformal registration using the control points method for Example 5.3. 125
5.10 *Top:* Results of conformal registration using the gradient flow method (with four terms) for Example 5.3. *Bottom:* Progress of the gradient descent algorithm towards finding a minimum for Example 5.3. 125
5.11 The source and target images for Example 5.4, the cartoon versions of Thompson's fish that he believed were 'isogonally' related. 126
5.12 Results of conformal registration using the control points method for Example 5.4. Although the shapes are very different, an extremely good registration is obtained, with an invertible mapping, at $\lambda = 150$. 126
5.13 Results of the continuation method for Example 5.4 for $25 \leq \lambda \leq 150$. . 127
5.14 Results of the continuation method for Example 5.4 for $150 \leq \lambda \leq 800$. The basic shape of the mapping remains unchanged over a wide range of λ values. 127
5.15 The L-curve for Example 5.4. 127
5.16 Registration results for Example 5.4 with the gradient flow method with four terms. 128
5.17 Registration results for Example 5.4 with the gradient flow method with eight terms. 129
5.18 Registration results for Example 5.4 with the gradient flow method with twelve terms. 129

5.19 The three grids found by the gradient flow method for Example 5.4 are plotted on the same graph. It can be seen that all three grids (with 4 terms in green, 8 terms in blue and 12 terms in black) display slightly different deformations. 130
5.20 Convergence of the gradient flow for Example 5.4; note that the y axis is different for the smooth case. *Top left*: Smoothed images, 4 terms; *top right*: original images, 4 terms; *bottom left*: original images, 8 terms; *bottom right*: original images, 12 terms. In each case the gradient flow ran successfully to a local minimum. 130
5.21 Registration results for Example 5.4 with the gradient flow method with six terms. 132
5.22 Registration results for Example 5.4 with the gradient flow method with nine terms. 132
5.23 Registration results for Example 5.4 with the gradient flow method with twelve terms. 133
5.24 The three grids found with six, nine and twelve terms for Example 5.4; compare to the somewhat different registrations found with 4, 8, and 12 terms shown in Figure 5.19. 133
5.25 Convergence of the gradient flow for Example 5.4. *Top left*: Smoothed images, 6 terms; *top right*: original images, 6 terms; *bottom left*: original images, 9 terms; *bottom right*: original images, 12 terms. The two graphs on the left do not show convergence to a local minimum because the gradient flow was stopped due to non-invertibility of the mapping. . . . 134
5.26 A pair of black and white images. *Left:* Image of a cat, which serves as the source. *Right:* Image of a chicken, which is used as the target. . . . 135
5.27 Results of conformal registration using the control points method for Example 5.5. A good registration is obtained at the cost of an irregular grid. Note that the first two values in the table are with the smoothed images, while the last row is not. 135
5.28 Results of the continuation method for Example 5.5 for $25 \leq \lambda \leq 150$. . 136
5.29 Results of the continuation method for Example 5.5 for $150 le \lambda \leq 800$. When λ is sufficiently large (greater than about 240) the mapping is invertible. 136
5.30 The L-curve for Example 5.5. 137
5.31 Registration results for Example 5.5 with the gradient flow method with four terms. The deformation is conformal, but the match is poor. 137

5.32 Registration results for Example 5.5 with the gradient flow method with eight terms. The registration is not markedly different to that with four terms. 138
5.33 Registration results for Example 5.5 with the gradient flow method with twelve terms. The registration is not markedly different to that with four or eight terms. 138
5.34 Three grids for Example 5.5 (with four, eight and twelve terms) are shown. Each grid shows a slightly different deformation. 139
5.35 Convergence of the gradient flow for Example 5.5. *Top left*: Smoothed images, 4 terms; *top right*: original images, 4 terms; *bottom left*: original images, 8 terms; *bottom right*: original images, 12 terms. For 12 terms the gradient flow stopped at $t = 0.0115$ (before locating a local minimum), due to the emergence of a non-invertible grid. 139
5.36 Five images of immature human skulls are presented at different ontogenetic times. These images of skulls are taken from [63]. 140
5.37 Results of conformal registration using the control points for Example 5.6 at $\lambda = 150$, 'Skull1' (source) registered to 'Skull4' (target). 141
5.38 Results of the continuation method for Example 5.6 for $25 \leq \lambda \leq 150$. . 141
5.39 Results of the continuation method for Example 5.6 for $150 \leq \lambda \leq 800$. . 142
5.40 L-curve for Example 5.6, 'Skull1' registered to 'Skull4'. 142
5.41 Results of conformal registration using the gradient flow method, 'Skull1' registered to 'Skull4'. 143
5.42 Results of registration (using gradient flow) when 'Skull1' is registered with 'Skull2'. 144
5.43 Results of registration (using gradient flow) when 'Skull1' is registered with 'Skull3'. 145
5.44 Results of registration (using gradient flow) when 'Skull1' is registered with 'Skull5'. 146
5.45 This graph indicates a smooth conformal pattern of skull growth in Example 5.6, with shape data relating the skull shapes (from the set of conformal mapping) projected to two dimensions. The data points are the absolute values of two Taylor coefficients in the conformal registration of 'Skull1' to each of the 5 skulls. 147

6.1 One of Thompson's isogonally related fish examples. These scanned images are taken from [101] (p 1064). 150

Declaration

It is hereby declare that this dissertation is my own work. It is being submitted for the degree of Doctor of Philosophy in Mathematics at the Massey University, Palmerston North. It has not been submitted before for any degree or examination at this or any other institution.

- - - - - - - - - - - - - - - - - -
Muhammad Yousuf Tufail
(Candidate's Signature)

Acknowledgements

Glory and Praise to Allah, the most gracious and merciful. *Indeed, the worst of living creatures in the sight of Allah are the deaf and dumb who do not use reason.*

- Al-Quran (8: 22)

Thousands and hundreds thousands salam to Prophet Muhammad Sall Allāhu ʿalayhi wa-sallam. *No two things have been combined better than knowledge and patience.*

- Sahih Bukhari (Book 11: Hadees 644)

I would like to express my deepest gratitude to my supervisors, Prof. Dr. Stephen Richard Marsland and Dist. Prof. Dr. Robert Ian McLachlan. This journey was so long that I thought it would never end, but they have been supportive and encouraging throughout my research. Their reassuring comments and strong belief in me were sometimes all that kept me going. I am truly indebted to them for their selfless time and unforgettable help during the writing process of this thesis. Their rational ideas, honest criticism, valuable suggestions and patience made this thesis possible. I am also very grateful to my supervisors for organizing many useful workshops where I got the chance to know amazing people. These workshops were a great source of learning through sharing and exchanging ideas with well-known researchers.

I gratefully acknowledge the Marsden Fund scholarship that I received towards my PhD at Massey University. This PhD study would not have been possible without their financial support. I would like to thank all technical and non-technical staff at SEAT for helping me through various stages.

I also appreciate the support and trust of my employer NED University of Engineering and Technology.

My heartfelt thanks to Abu, Ami and my siblings for always believing in me and encouraging me to follow my dreams. I am also thankful to my close friends who kept my sense of humour alive.

And finally to my sweetheart Hamza who wonders what I do all day and without whose never ending love, this thesis would have been finished in due time. And to my lovely wife who has been a source of encouragement and forbearance through thick and thin. She also provided motivation, often by asking: when are you going to submit this thesis?

Abstract

Image registration is the process of finding an alignment between two or more images so that their appearance matches. It has been widely studied and applied to several fields, including medical imaging and biology (where it is related to morphometrics). In biology, one motivation for image registration comes from the work of Sir D'Arcy Thompson. In his book *On Growth and Form* he presented several examples where a grid superimposed onto a two-dimensional image of one species was smoothly deformed to suggest a transformation to an image of another species. His examples include relationships between species of fish and comparison of human skulls with higher apes.

One of Thompson's points was that these deformations should be as 'simple' as possible. In several of his examples, he uses what he calls an *isogonal* transformation, which would now be called conformal, i.e., angle-preserving. His claims of conformally-related change between species were investigated further by Petukhov, who used Thompson's grid method as well as computing the cross-ratio (which is an invariant of the Möbius group, a finite-dimensional subgroup of the group of conformal diffeomorphisms) to check whether sets of points in the images could be related by a Möbius transformation. His results suggest that there are examples of growth and evolution where a Möbius transformation cannot be ruled out. In this thesis, we investigate whether or not this is true by using image registration, rather than a point-based invariant: we develop algorithms to construct conformal transformations between images, and use them to register images by minimising the sum-of-squares distance between the pixel intensities. In this way we can see how close to conformal the image relationships are.

We develop and present two algorithms for constructing the conformal transformation, one based on constrained optimisation of a set of control points, and one based on gradient flow. For the first method we consider a set of different penalty terms that aim to enforce conformality, based either on discretisations of the Cauchy-Riemann equations, or geometric principles, while in the second the conformal transformation is represented as a discrete Taylor series. The algorithms are tested on a variety of datasets, including synthetic data (i.e., the target is generated from the source using a known conformal transformation; the easiest possible case), and real images, including some that are not actually conformally related. The two methods are compared on a set of images that include Thompson's fish example, and a small dataset demonstrating the growth of a human skull. The conformal growth model does appear to be validated for the skulls, but interestingly, not for Thompson's fish.

Chapter 1

Introduction

It is not knowledge, but the act of learning, not possession but the act of getting there, which grants the greatest enjoyment.

- Carl Friedrich Gauss (1777-1855)

Image registration is the process of finding an alignment between two or more images so that their appearance matches. It has been widely studied and applied to several fields, including medical imaging and biology (where it is related to morphometrics). One of the key choices of image registration is the set of allowable transformations that can be used to deform the images. In this thesis methods of image registration using the group of conformal diffeomorphisms are derived and demonstrated. The motivation for the use of this group is the pioneering work of Thompson [101], as shall be discussed in Section 1.4. However, prior to that, we set the scene by introducing the overall idea of image registration, together with some practical implementation details, give a brief definition of conformal diffeomorphisms and the relevant mathematical setting, and also summarise some application areas of image registration.

1.1 Image registration

Given a pair of images I_1 and I_2, often called the source and target, defined on some domain Ω (typically, $\Omega \subset \mathbb{R}^2$), the process of image registration can be described as the minimisation of an objective function that measures the mismatch between the images by varying the deformation φ^{-1} that is applied to one of the images. (The reason for using φ^{-1} will be explained in Section 1.2.4.) One such objective function is

$$E_{I_1,I_2}(\varphi) = \int_\Omega \|I_1 \circ \varphi^{-1}(x_1, x_2) - I_2(x_1, x_2)\|^2 dx_1 dx_2, \quad \forall (x_1, x_2) \in \Omega. \tag{1.1}$$

The principal choices to be made here are the set of allowable transformations from which φ can be chosen (see Table 1.1), the norm $\|\cdot\|$ that measures the mismatch between the target image and the transformed source after φ has been applied to it, and the optimisation method that is applied. Other terms may be added to this matching functional, for example to make the path between the images as short as possible, or to minimise the amount of bending of the space that is performed.

Other aspects that arise are discretization of the space of images, see Def. 1.1, Figure 2.6, and Example 2.5, interpolation of the transformed image (see Section 2.1.2), and smoothing of the images (see Section 3.1.1, Chapter 3).

Equation (1.1) is written for the registration of two-dimensional greyscale images, since that is what we consider in this thesis. However, it can be easily modified to match 3D images (such as many medical images) or colour images, see Definition 1.1.

In standard image registration there are two commonly-used transformation groups: the rigid group (generated by translations, rotations, and scalings, hence four parameters in 2D), which gives a global transformation of the space, and the diffeomorphism group, which is infinite-dimensional, and enables the local deformation of all points in the image. Together, these groups allow for smooth registration of pairs of related images, as shall be summarised later. However, they are certainly not the only possible choices, and in this thesis we consider two intermediate groups that are related to each other, one finite dimensional, and one infinite dimensional. Table 1.1 lists the transformation groups of interest in this thesis, their dimension in 2D, and their brief features.

Transformations groups	**Dimensions**	**Features**
Rigid group	4	Subgroup of the Möbius group, shapes remain invariant (see Chapter 2).
Möbius group	6	Subgroup of the conformal group, cross ratios preserved (see Chapter 2).
Conformal group	∞	Subgroup of the group of diffeomorphisms, local angles preserved (focus of this thesis, see Chapter 3, 4, 5).
Group of diffeomorphisms	∞	The biggest group in image registration, [114] [12], [54].

Table 1.1: The transformation groups relevant to this thesis.

1.1.1 Brief Overview of the Literature

Brown [21] gave a comprehensive review of image registration as it existed in 1992. The basic steps have not changed since: Brown suggests that the requirements are (i) a feature space (contains information about the images), (ii) a search space (consists of the set of transformations), (iii) a search strategy (selection of transformations to get an optimal solution), and (iv) a similarity metric (to measure the discrepancy between the images). However, the choice of transformations certainly has changed, since only three sets of transformations are listed by Brown, all finite dimensional: (a) affine transformation (rigid, shearing and aspect ratio, 2D), (b) perspective transformation (3D to 2D, the special case is projective transformation where the scene is a flat plane such as an aerial photograph) and (c) polynomial transformation that can be used when less information is available about the camera geometry.

While there have been many developments in image registration since 1992 (see reviews such as [118], [42], [10], [44], [90] for more details), the most significant change is the adoption of diffeomorphic image registration, which is now a huge area of research. Diffeomorphisms are smooth functions with smooth inverses; they will be defined in Section 1.2.2. The area where diffeomorphisms are most used in image registration is medical imaging, where images of parts of the human body from a variety of scanning methods, such as CT and MRI, are aligned either to assist in diagnosis of a disease, or assist in surgical planning. This area is now known as *Computational Anatomy* [45]. Trouvé proposed using the action of an infinite dimensional group to deform images in [102]. As the composition of infinitesimal diffeomorphisms is diffeomorphic, Dupuis et al. [34] solved the transport equation (introduced in [25]) to construct a velocity field that deformed the source image. This method was developed in [45] and [54], and is known as the *Large Deformation Diffeomorphic Metric Mapping* (LDDMM) method. Trouvé constructed a diffeomorphism (in 1996) by solving the ordinary differential equation formulation numerically in [103] using gradient descent.

Given that the diffeomorphism group is infinite dimensional, there will be many possible solutions to any matching between two images. However, the diffeomorphism group is a manifold as well as a group (although as it is infinite dimensional, it is not a Lie group). This enables differential geometry methods to be applied in order to choose particular elements of the group, such as the geodesic between the representations of the two images (which is the source of the word *metric* in the LDDMM method). The metric on the diffeomorphism group can be used to identify images that are close together and, it is ultimately hoped, to perform statistical analysis of groups of images [37], [80], [97].

To date, diffeomorphic image registration has used the full group of diffeomorphisms. However, as shall be discussed in Section 1.4, there are natural processes such as biological growth and evolution that seem to produce conformal deformations. In this thesis we describe two alternative methods for the construction of conformal diffeomorphisms.

Image registration can be based either on a set of points (often called landmarks), or upon the intensity values of the pixels. The former is sometimes also known as point mapping, and requires that corresponding points be defined in the images to be registered [118], [90]. However, in either case the matching functional is based on the difference between the two images. There have been a large variety of methods to measure this proposed in the literature. The most common is the L_2 norm between the pixels of the images (also known as the sum-of-squares difference), which is what we will use in this thesis. Other options include cross-correlation [21], Fourier-based comparison [21], mutual information (based on Shannon's entropy [92]) [29], [59], [86], and normalised gradient field [44]. While some of these methods aim to produce fast algorithms in particular cases, most of them are used in multi-modal image registration, where the pixel intensities in the images differ. For example, medical images taken with different imaging modalities will show the same objects with different colours, and normal camera images of the same scene under different lighting will have a non-linear transformation of the colour range applied. We do not consider these cases further in this thesis.

Equation (1.1) is continuous. However, images in computers are discrete objects, based upon a set of pixels. Therefore, for the sake of numerical computation, we discretise the domain, which we will call S; the details are given in Chapter 2. This discretisation results in the following discrepancy function that will be used throughout this thesis (where x_{ij} is the entry at (i, j) in a matrix). For convenience we consider square images that have N pixels per side (N^2 in total in 2D), but there is no difficulty working with images of size $M \times N$ in practice. However, the source and the target images must have same number of pixels for our methods of image registration:

$$E(\varphi) = \sum_{i=1}^{N} \sum_{j=1}^{N} \{(I_1 \circ \varphi^{-1})(x_{ij}) - I_2(x_{ij})\}^2, \quad x_{ij} \in S \subset \Omega. \tag{1.2}$$

On the basis of this discrete function we define image registration as: *A process in which we find a φ that minimises the function in Equation* (1.2) *is called an image registration.* After the selection of discrepancy function, the next step is to minimise it. Optimisation is itself an independent field of research and plenty of methods have already been developed; possible references for further details include [56], [13], [15]. There is no single method that can solve every minimisation problem, instead a variety of standard methods of optimisation are available, primarily based on *gradient descent* [11], [43], [103], [114]. For the purposes of this thesis, least-squares optimisation will be sufficient, as will be discussed in Chapter 2. However, we note in passing that image registration is subject to many local minima in the search space, and it is generally necessary to employ a coarse search to find a reasonable initial starting point for the optimisation.

In the next section we present a few basic definitions and terminologies that will be used throughout this thesis.

1.2 Definitions

1.2.1 Continuous and discrete images

Definition 1.1. *Continuous images are functions defined as follows:*

$$\begin{aligned}
&I(\mathbf{x}) : \Omega \subset \mathbb{R}^2 \to [0,1]^3 \subset \mathbb{R}^3, && \text{2-D colour image.}\\
&I(\mathbf{x}) : \Omega \subset \mathbb{R}^2 \to [0,1] \subset \mathbb{R}, && \text{2-D greyscale image.}\\
&I(\mathbf{x}) : \Omega \subset \mathbb{R}^3 \to [0,1]^3 \subset \mathbb{R}^3, && \text{3-D colour image.}\\
&I(\mathbf{x}) : \Omega \subset \mathbb{R}^3 \to [0,1] \subset \mathbb{R}, && \text{3-D greyscale image.}
\end{aligned}$$

Discrete images are defined analogously with the domain Ω replaced by a finite subset $S \subset \Omega$ (the pixels), and the range $[0,1]^n$ replaced by a discrete subset (the allowable colours).

Typically, for greyscale images each pixel value is represented using 8 bits, so the discrete range contains 256 values between 0 (black) and 1 (white). For colour images, three channels corresponding to red, green, and blue (the primary colours of light) are used. We shall generally take $\Omega = [-0.5, 0.5]^2$ and S a uniform $N \times N$ grid in Ω. We will later call continuous images *smooth* if φ is a smooth function.

Example 1.1. Figure 1.1 displays a set of discrete images.

1.2.2 Diffeomorphism

Definition 1.2. [114] (p.no. 161) *A homeomorphism of $\Omega \subset \mathbb{R}^d$ is a continuous map $\varphi : \Omega \to \mathbb{R}^d$ such that its inverse, $\varphi^{-1} : \varphi(\Omega) \to \Omega$ is continuous.*
A diffeomorphism of Ω is a continuously differentiable homeomorphism $\varphi : \Omega \to \mathbb{R}^d$ such that φ^{-1} is continuously differentiable.

The work in this thesis is based on 2D discrete images therefore, $d = 2$ throughout.

1.2.3 Conformal diffeomorphism

Definition 1.3. *A differentiable function $\varphi\colon \Omega \to \mathbb{R}^d$ is said to be conformal, if it preserves local angles, that is, if*

$$\angle(u, v) = \angle(D\varphi(x).u, D\varphi(x).v)$$

(a) 2-D colour image

(b) 2-D greyscale image

(c) 3-D colour image

(d) 3-D greyscale image

Figure 1.1: A set of discrete images. *Left column:* 2-D and 3-D colour images. *Right:* greyscale images.

for all tangent vectors u, v *based at* $x \in \Omega$, where $\angle$ denotes an angle and D represents the Jacobian.
A differentiable function $\varphi\colon \Omega \to \mathbb{R}^d$ *is said to be a conformal diffeomorphism if it is conformal and a diffeomorphism. The set of conformal diffeomorphisms will be denoted* $con(\Omega, \mathbb{R}^d)$.

When $d = 2$, taking standard coordinates x_1, x_2 on $\mathbb{R}^2$ and writing $z = x_1 + \mathrm{i}x_2$, it is well known that $\varphi(z)$ is conformal if and only if it is complex differentiable with respect to the complex variable z and has non-zero derivative. That is, any complex analytic function is conformal. φ is a conformal diffeomorphism if it is complex analytic and $\varphi'(z) \neq 0$ for all $z \in \Omega$. Equivalently, φ is a conformal diffeomorphism if it is biholomorphic.

Example 1.2. Figure 1.2 indicates the transformation of a uniform square grid by a conformal diffeomorphism—in this case, a Möbius transformation of the form $\varphi(z) = (az + b)/(cz + d)$; these will be discussed more in the next chapter.

1.2.4 Action of diffeomorphisms on images

Let $I\colon \Omega \to \mathbb{R}$ be a continuous greyscale image and let $\varphi\colon \Omega \to \mathbb{R}^d$ be a diffeomorphism. Then the action of this diffeomorphism on the image I is defined as:

$$\varphi.I(x) = I \circ \varphi^{-1}(x) = I(\varphi^{-1}(x)), \forall x \in \varphi(\Omega) \subset \mathbb{R}^2, \tag{1.3}$$

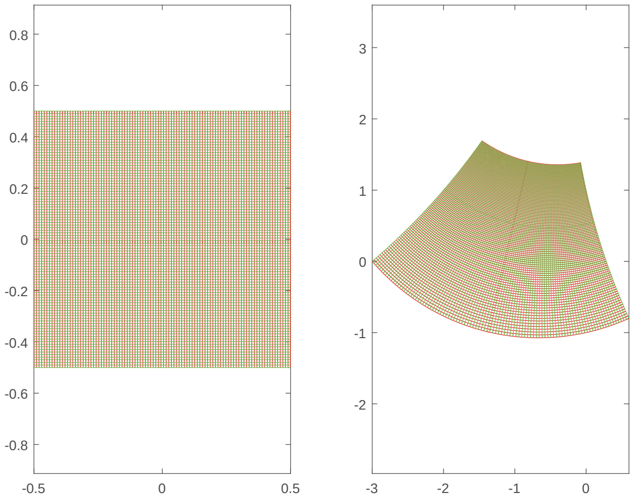


Figure 1.2: *Left:* A uniform grid (100×100) is given. *Right:* The image of the uniform grid under the conformal diffeomorphism $\varphi(z) = \frac{az+b}{cz+d}$, $z \in [-0.5, 0.5] \times [-0.5, 0.5]$, where $a = 0.4i, b = 0.2 + 0.4i, c = 0.2i, d = 1$).

where '.' denotes the action of φ on I and $\circ$ denotes composition.

To explain the appearance here of φ^{-1} instead of φ, consider a pixel at $y \in \Omega$. The transformation φ carries this pixel to $x = \varphi(y)$. Therefore it is natural to consider that the *values* of the image are unchanged by this transformation of the domain, and thus to define $(\varphi.I)(x) = I(y)$. See Figure 1.3. Inserting $y = \varphi^{-1}(x)$ gives Equation (1.3). Note that the domain of the transformed image is $\varphi(\Omega)$, not Ω.

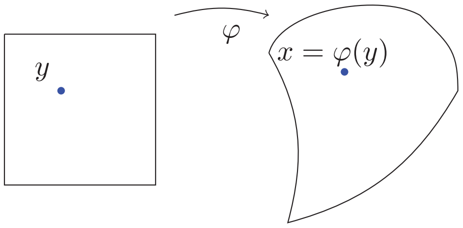


Figure 1.3: Position $y \in \Omega$ in the pixel grid is mapped under φ to position $x = \varphi(y)$ in the transformed grid. In order to obtain the intensity of x it is necessary to invert φ and recover $y = \varphi^{-1}(x)$.

Proposition 1.1. *If the composition of conformal diffeomorphisms is defined in the domain $\Omega \subset \mathbb{C}$ or $\Omega \subset \mathbb{R}^2$ then the following equation holds:*

$$(\varphi_1 \circ \varphi_2) \cdot (I(z)) = \varphi_1 \cdot (\varphi_2 \cdot I(z)), \forall \varphi_1, \varphi_2 \in con(\Omega, \mathbb{R}^2)$$

where $\circ$ denotes the composition, '.' indicates the action of φ over the image I.

Proof.

Suppose φ_1 and φ_2 are conformal diffeomorphisms in Ω such that their composition $\varphi_1 \circ \varphi_2$ is also a conformal diffeomorphism in Ω, then

$$\begin{aligned}
(\varphi_1 \circ \varphi_2) \cdot (I(z)) &= I((\varphi_1 \circ \varphi_2)^{-1}(z)), \qquad \text{From equation 1.3}\\
&= I((\varphi_2^{-1} \circ \varphi_1^{-1})(z)),\\
&= I(\varphi_2^{-1}(z) \circ \varphi_1^{-1}(z)),\\
&= I \circ \varphi_2^{-1}(z) \circ \varphi_1^{-1}(z),\\
&= (I \circ \varphi_2^{-1}(z)) \circ \varphi_1^{-1}(z),\\
&= (\varphi_2 \cdot I(z)) \circ \varphi_1^{-1}(z),\\
&= \varphi_1 \cdot (\varphi_2 \cdot I(z)).
\end{aligned}$$

□

This explains the terminology of calling $\varphi.I$ an *action* of φ on I. The proposition shows (roughly) that it is analogous to a left action of a group, where the group operation is composition of diffeomorphisms. Note, however, that composition is only defined where the respective domains and ranges overlap, so $con(\Omega, \mathbb{R}^2)$ is not strictly speaking a group. However, this aspect is not important to the thesis.

1.3 Applications of image registration

Image registration is an interdisciplinary subject and it is widely used in several branches of science such as morphometrics, medical imaging, computer vision, and remote sensing. This section is dedicated to a brief discussion of these fields.

1.3.1 Morphometrics

Morphometrics is a field of biological sciences in which the differences between the *forms* of organisms are studied. The word comes from the combination of two words 'morpho' or morphology (the study of structures or shapes of organisms) and 'metric' (distances). The analysis of shapes or forms plays an important role in biological studies and many biological processes involve shape changes such as ontogenetic development, growth of animals, and the development of disease [116]. As we will see in Section 1.4, some of these processes can lead to angle-preserving shape changes, which will be the primary focus of this thesis.

Morphometrics can be described as pre-image image registration, in that it developed methods to study changes in form before image registration was commonly used.

The study of shapes and shape change has been of interest to humans since as far back as the Middle Kingdom (*c.* 1986–1633 BC) [93]. However most of the mathematical activity in this field has taken place in the last 20 years of the 20th century (e.g., [16], [17], [64], [84], [65], [94], [33]), at least 10 to 15 years before the development of the field of image registration. As with other fields related to image registration, the original credit goes to Sir D'Arcy Thompson, who is considered to be the father of morphometrics even though this word had not been coined at that time. According to [63] (p. no. 3) *"Thompson's work is still regarded as advanced for his time, and his transformational grids are most modern morphometrician's envy. The information communicated by Thompson's transformational grids remain the goal of most morphometric studies, e.g., [16], [18], [83]"*.

Methods of morphometric measurement are based on landmarks [2] (Chapter 6, p. no. 197-233). These landmarks can be physical locations on objects, or features that can be easily and repeatably defined on sets of objects (such as corners), and the measurements can include distances, ratios, and angles between the selected landmarks. Bookstein divides landmarks into three types in [18] (Chapter 3, p. no. 63-66): Type I (*discrete juxtapositions of tissues*) which includes three type of structures such as bony structures, branching points and centre or centroids, Type II (*maxima of curvature*) that includes the local properties of shapes such as curvature and torsion, and Type III (*extremal points*), which includes the farthest points in shapes such as the end points of the diameter. The set of landmarks is deformed to match another set of landmarks based on either (i) the principle of superimposition (which is based on least squares), or (ii) thin-plate splines, an interpolating function [36]. By contrast, image registration provides an alternative way to study related *forms* (the appearance of the whole or part of an organism) within and between organisms. The object of interest is the entire image, not landmarks, and the distance between objects is used to provide information rather than derived measurements from the landmarks.

1.3.2 Medical imaging

The most common application of image registration is in medical image analysis, where images based on Magnetic Resonance Imaging (MRI) or Computed Tomography (CT) of anatomical parts such as the brain, heart, or indeed any part of the body are analysed and studied. There are a variety of reasons why these can be of interest, but any disease that causes anatomical regions to change shape or size (such as Alzheimer's disease, dementia, schizophrenia in the brain, or some heart problems and kidney disease) is of relevance.

Rigid image registration is used pre-operatively to align real-time images of the patient with various scans taken prior to surgery [52], [96], [51]. However, for the kinds of deformations that are seen in disease, non-rigid transformations are required. While there are a variety of methods that can be used, such as the spline-based approaches in [47], [75], [87], [48] to perform non-rigid image registration, most of the research in the past 15 years has been focussed on diffeomorphisms. The reasons for these are not just that the warps[†] are smooth and invertible, but they define an infinite-dimensional Lie group, and give access to a Riemannian metric that can be used to describe how much difference there is between pairs of images. There is a huge body of work on this now, from both the mathematical and the medical imaging analysis viewpoint, from analysis of the metrics [72], [115], through fast algorithms [8], statistical analysis on Riemannian manifolds [53] to applications. To see the wide range of applications of diffeomorphic registration in medical image analysis, consider that it has been applied to Alzheimer's disease in [24] and [117], rheumatoid arthritis in [20], a disease related to the retina in [1], a model for Krabbe's disease (a white matter demyelinating disease) in [46] and chronic kidney-related diseases in [88]. Image registration provides a platform to researchers (in general) and anatomists (in particular) to analyse and compare the geometry of a patient's organ who is carrying (shape related) diseases with corresponding organs of a healthy person.

1.3.3 Computer vision

We live in a three dimensional world, but most imaging methods are two dimensional, and objects can look surprisingly different from fairly similar angles. Transforming one image to more closely match the appearance of another (i.e., image registration) can help in object recognition, which is an important part of computer vision [77]. It can also be useful for detecting and tracking moving objects [113], [27].

1.3.4 Remote sensing

Remote sensing (sometimes known as 'earth observation' [3]) is a field of computer science that obtains information of objects on earth from space using satellites or from the air with the help of aircraft such as drones [26], [98]. In remote sensing, the procedure of registration involves five steps that are (i) preprocessing, (ii) feature selection, (iii) feature correspondence, (iv) determination of a transformation function and (v) resampling. The feature selection phase is divided into two categories, (a) orientation of structure (that include lines, curves, corners, etc.), and (b) intensity based [32] (p. no. 103-104), [30]. Methods of registration include cross correlation, Fourier transformations, etc., (for further details: see table 1, p. no. 107 in [32]) that are based on finite dimensional transformations, mostly affine. In remote sensing, shapes are invariant, therefore finite dimensional transformations are preferred. Remote sensing is

---

[†] the function φ that deforms the image is known as a warp

widely used in agriculture [9], forestry [62], geology [105] and geography [35] amongst other fields. Looking at the earth from above produces deformations from a variety of groups, including affine and projective [76], and these need to be removed for most analysis techniques to be usefully applied.

1.3.5 Visual Cortex

The functionality of the human brain has always been a subject of curiosity. The basic understanding of central visual system anatomy developed between 1600 and 1800 AD [91]. The visual cortex is a part of the brain that processes visual information. In [39], the authors state that the visual field (in the brain) can be represented in the form of a conformal mapping. They say, *in the case of visual cortex, there is considerable experimental evidence that the mapping of the retina to the surface of primary visual cortex is approximately isotropic. Thus, to model the representation of a visual image on the surface of the cortex, we need to construct a conformal approximation to this map* (p. no. 59). At that time, diffeomorphic image registration was not being explored. But, now we have a framework of diffeomorphic mapping. Thus, the study of the visual cortex provides a motivation for the development of conformal image registration.

1.4 Motivation

100 years ago, D'Arcy Wentworth Thompson produced the first edition of his seminal book *On Growth and Form* [101]. In a chapter of that book entitled 'On the Theory of Transformations' he suggested that in many cases animals that were closely related in evolutionary space would have relatively small deformations between their appearances, and that these deformations would be mathematically 'simple'. Over the past 15 years the book has been used as justification for the study of diffeomorphic deformations of images of different objects, a field that is now extremely well studied, particularly for the registration of medical images. While this recent research work is interesting mathematically and appears to have many useful applications, it is not the focus of this thesis.

Thompson's examples are hand-drawn, and consist of a two dimensional outline of an animal (or part of an animal) and a grid superimposed on it. The deformation to another animal form is demonstrated by the picture of the new form and what is intended to be the same grid pushed forward so that the smoothness of the deformations can be clearly seen. Examples are given in Figure 1.4, of several species of fish, and in Figure 1.5, which shows the skulls of a human, chimpanzee, and baboon.

As previously mentioned, Thompson's principal point was that the deformations between closely related species should be simple. It has been suggested in [66] that

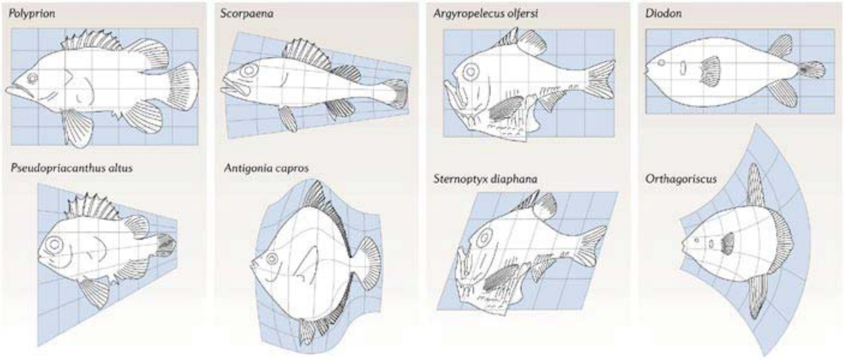


Figure 1.4: Examples of Thompson's transformations between related species. These images are taken from [7] (p. no. 404).

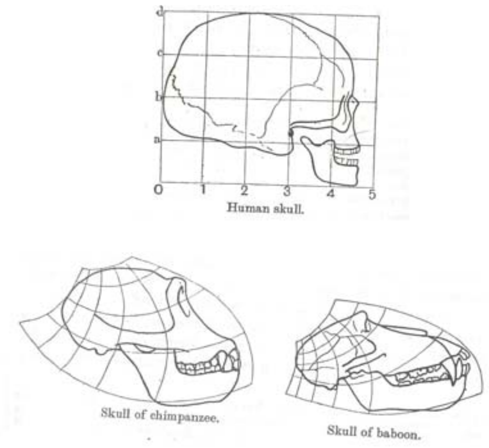


Figure 1.5: These images of skulls are taken from Thompson's book [101]

this idea of simplicity can be interpreted in modern parlance as being examples of low dimensional groups. Some evidence of this, from [66], is given in Table 1.2, where some possible transformation groups for several of Thompson's figures, are given. Clearly, the infinite dimensional diffeomorphism group would not be simple, but the four dimensional rigid group or the six dimensional Möbius group would be.

The transformations shown in Figure 1.4 are certainly simple; two of them are linear. Considering the pair of images shown on the right of (Figure 1.4), the *Diodon* and the *Orthaoriscus* (which are Figs. 525 and 526 in [101]), Thompson says: *"This is a particularly instructive case of deformation or transformation. It is true that, in a mathematical sense, it is not a perfectly satisfactory or perfectly regular deformation, for the system is no longer isogonal; but nevertheless, it is symmetrical to the eye, and obviously approaches to an isogonal system under certain conditions of fraction or*

Figure no. (in [101])	**Transformation group** (as suggested in [66])
515	$x \to ax$, $y \to y$ (scaling in horizontal direction)
513.2	$x \to ax$, $y \to by$ (scaling in both directions)
509,510,518	$x \to ax$, $y \to cx + dy$ (shears)
521-522,513.5	$x \to ax + by$, $y \to cx + dy$ (affine)
506,508	$x \to ax$, $y \to g(y)$
511	$x \to f(x)$, $y \to g(y)$
517-520,523,513.1,513.3,513.4,513.6,514,525	Conformal
524	Peculiar

Table 1.2: This table suggests the connection between the transformations group and Thompson's images in [101]. This table is is taken from [66].

constraint". (p. no. 1064).

We would nowadays term an isogonal transformation to be a *conformal* one, and these conformal transformations are mentioned several times in his book; regarding the deformation of the grid of the human skull that he has sketched to match the chimpanzee and the baboon, he comments that *"the empirical coordinates which I have sketched in for chimpanzee as a conformal transformation of the Cartesian coordinates of the human skull look as if they might find their place in an equipotential elliptic field* (p. no. 1084) and *"I have shewn the similar deformation in the case of baboon, and it is obvious that the transformation is of precisely the same order"* (p. no. 1083).

It might seem odd that conformal deformations would be mentioned at all in Thompson's work. However, they appear frequently (we shall give another example next), and two other works [81, 73] have also investigated related questions, showing that two dimensional representations of biological growth appear to be conformal, based on a few examples. These works have used the cross-ratio between hand-selected points on sketches of biological growth as their evidence. In this thesis we provide an alternative method of verification, by studying the conformal registration of images, and developing algorithms that will identify the best match between pairs of images under the requirement that the transformation be conformal. We will use these algorithms to test whether or not there is, in fact, a conformal transformation between these images.

In fact, it is somewhat surprising, especially in view of Thompson's pioneering work, that so few sets of transformations have been used to relate shapes; rigid, affine, projective, and diffeomorphic registration dominate the literature. However, we have found two discussions in the literature that explore this question in more detail, at least for finite-dimensional groups: a seminar presentation of John Milnor [73] and a rarely-cited 1989 paper by Petukhov [81]. These are key motivations for the thesis and we now discuss their approach.

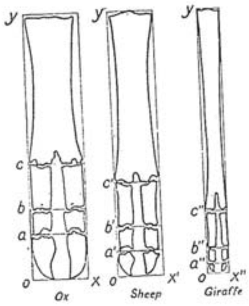
The cross-ratio of four points $A, B, C, D \in \mathbb{C}$ is defined by

$$[A, B, C, D] = \frac{(C - A)(D - B)}{(C - B)(D - A)}. \tag{1.4}$$

It is an invariant of Möbius transformations. That is, if $\varphi(z) = (az + b)/(cz + d)$, then $[\varphi(A), \varphi(B), \varphi(C), \varphi(D)] = [A, B, C, D]$ for all $A, B, C, D \in \mathbb{C}$. However, if A, B, C, D are collinear, as in the examples of Milnor described below, then it is also an invariant of some other transformation groups, such as the projective group and the 1D linear fractional group. Thus, some care must be exercised in interpreting cross-ratios.

Comparison of foot bones

In [101], Thompson computed the vertical distances between four marked position of the foot bones of an ox, a sheep and a giraffe (see Figure 1.6). We use these numerical values and compute the cross-ratio between these positions in order to test whether or not they are related by a Möbius transformation. The measurements are given in the following table.



Age	o	a	b	c	y	cy/oa	$[o,a,b,c]$	$[a,b,c,y]$
Ox	0	18	27	42	100	3.2	1.71	1.4
Sheep	0	10	19	36	100	6.4	1.52	1.4
Giraffe	0	5	10	24	100	15.2	1.58	1.3

Figure 1.6: *Top:* Images of foot bones of an ox, a sheep and a giraffe. Four positions are marked in each image. These images are taken from [101]. *Bottom:* Distances and cross-ratios between the foot bones of an ox, a sheep and a giraffe.

It can be seen in the sixth column of table in Figure 1.6 that the distance ratio (cy/oa) of a giraffe is almost five time the corresponding distance ratio for an ox and for sheep, and almost twice that of the ox. However, apparently the cross-ratios (which are given in the last two columns) are nearly invariant. Milnor regarded this as supporting the suggestion that the relationship between them is a Möbius transformation. However, note that it could also lend evidence to other transformation groups. Either way, it looks extremely significant.

Human skulls

In [81], Petukhov used Thompson's grid method to study a set of images of a human skull as it grew, using a set of skull images from [79]; see Figure 1.7. He used a Möbius transformation as the set of transformations, and argued that his study supported the idea that the growth of human skulls follows a path in the Möbius group.

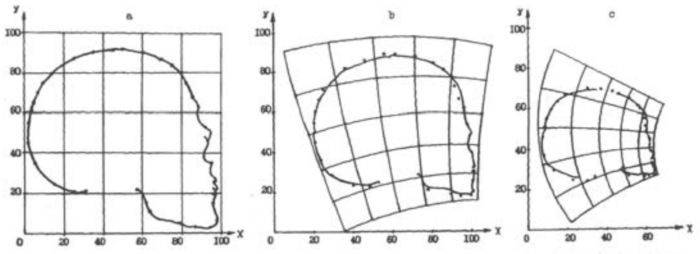


Figure 1.7: Scanned images of [81] (a) An adult, (b) a five year old and (c) a newborn.

Growth of the middle (human) finger

In [81], Petukhov used the data from [85] of the size of the middle finger of a man at different ages (shown at the top of Table 1.3 with the four points A, B, C, D marked). He computed the cross-ratio (see Equation (1.4)), which he called the 'Wurf' and denoted by W. The values of W he computed are given in Table 1.3.

Age	AB	BC	CD	W
4	2.42	1.43	0.86	1.31
6	2.64	1.65	1.02	1.31
8	3.00	1.88	1.19	1.31
10	3.10	1.96	1.25	1.31
12	3.34	2.13	1.37	1.31
14	3.56	2.27	1.46	1.31
16	4.08	2.57	1.64	1.31
18	4.19	2.65	1.69	1.31
21	4.41	2.78	1.76	1.31

Table 1.3: *Top:* Four positions are marked on middle finger. *Bottom:* Numerical results of the computation of the cross-ratio of points on the middle finger in man at different ages, taken from [81].

In this example the cross-ratio appears to be invariant during the human growth, again suggesting that Möbius or a related group is acting.

Human growth

In [81], Petukhov borrowed data (length in cm) from [22] from measurements of human growth and computed the cross-ratio W between the physical locations (Euclidean distances measured by Bunak in [22] for arm growth and for human growth in the vertical direction at different ages, see Figure 1.8). The numerical results (of the cross-ratio) are given (for postnatal and antenatal stages respectively) in Table 1.4.

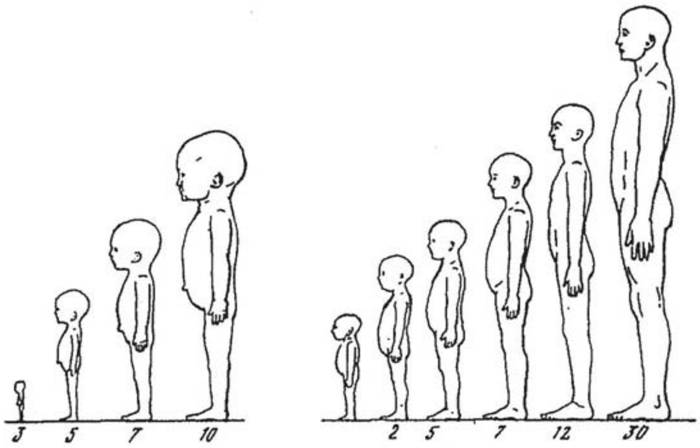


Figure 1.8: *Left:* Antenatal stages (in lunar months). *Right:* Postnatal stages (in years). These images are taken from [81].

Age	Upper part	Torso	Lower part	W	Shoulder	Forearm	Wrist	W
New born	10.6	18.3	28.6	1.29	8.80	7.40	6.30	1.33
1	15.6	25.0	41.7	1.32	12.5	10.4	8.70	1.33
4	18.9	31.4	58.4	1.32	16.7	13.7	11.2	1.33
7	21.0	35.3	71.7	1.33	20.7	16.5	13.1	1.33
10	22.5	38.0	80.6	1.34	23.2	18.4	14.4	1.32
13	24.5	42.2	93.5	1.34	26.3	20.7	16.0	1.32
17	26.9	47.8	108.6	1.33	30.5	23.9	18.3	1.32
20	25.3	51.8	109.9	1.29	32.3	24.5	18.8	1.33

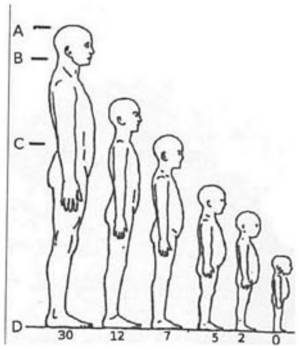
Age	Hip	Shin	Foot	W	Shoulder	Forearm	Wrist	W
4	2.76	2.66	1.84	1.26	2.37	1.90	1.55	1.33
6	5.45	5.45	4.30	1.27	4.40	3.65	3.23	1.34
8	7.40	7.60	5.80	1.27	6.17	5.05	4.53	1.35
9	8.40	8.60	6.61	1.27	6.95	5.70	5.14	1.35

Table 1.4: *Top:* Numeric results of cross-ratios for postnatal stages are given. *Bottom:* Numeric results of cross-ratios for antenatal stages are provided. These numerical results are taken from [81]

Petukhov says *“This is especially interesting because the growth of human body is essentially nonlinear; for instance, the upper part grows 2.4-fold, the torso 2.8-fold and the lower part 3.8-fold. The values of these cross-ratios in all blocks, at least during the entire individual post-natal development, group around the benchmark of* $P \approx 1.31$*”* (p. no. 522). Further he added: *“A non-Euclidean analysis of the structure and growth reveals that all three-membered blocks of the human kinematics in the straightened posture are practically Möbius equivalent and Möbius invariable during the lifetime”* (p. no. 522)

The same data was reinterpreted by the mathematician John Milnor in a lecture that he gave to the Institute for Advanced Studies in Princeton in 2010 [73]. Milnor’s talk was inspired by Thompson’s book, and also used Petukhov’s data (from [81]) and the cross-ratio with the different orientation. His conclusion was that the data did not support conformal growth in general, but that straight lines were often conserved.

Milnor says in his lecture [73]: *“A human child has a relatively large head and small torso in comparison to an adult. The ratio of head size to torso size is not at all invariant under growth. But if we look at the transformation which carries each point of a child’s body to the corresponding point of an adult’s body, then it does seem to be very conformal.”* Figure 1.9 displays the four positions marked on the human body (the points from which he computes the cross-ratio); the results are shown in the table in that figure.



Age	AB(head)	BC(torso)	CD(legs)	$[A\ B\ C\ D]=\frac{(AC)(BD)}{(AB)(CD)}$	$\frac{CD}{AB}$
0	10.6	18.3	28.6	4.47	2.70
4	18.9	31.4	58.4	4.09	3.09
7	21	35.3	71.7	4.00	3.41
10	22.5	38	80.6	3.96	3.58
20	25.3	51.8	109.9	4.48	4.34

Figure 1.9: *Top:* Four positions are marked on each image showing the appearance of a human at different ages. *Below:* Numerical results for the cross-ratios are given (taken from [73]).

Milnor says about this computation *“The legs grow almost twice as much as the head; but the cross-ratio remain relatively stable. In particular, cross ratios change far* *less than simple ratios of distances.".*

Growth of middle (human) finger

Milnor says in [73] that *"If we consider only a very small region of the body, then conformal growth would approximately be growth of similarity transformation"*. We know that the similarity transformations (which include translation and rotation) lie inside the conformal group. As a transformation he took Petukhov's [81] data (see Table 1.3).

Age	AB	BC	CD	$[A\ B\ C\ D]=\frac{(AC)(BD)}{(AB)(CD)}$	Ratios
4	2.42	1.43	0.86*	4.24	0.51:0.30:0.18
8	3.00	1.88	1.19	4.19	0.49:0.31:0.20
14	3.56	2.27	1.46	4.18	0.49:0.31:0.20
21	4.41	2.78	1.76	4.21	0.49:0.31:0.20

Table 1.5: Computation of cross-ratio at (marked) position of the middile finger (see the figure of a human hand in Table 1.3).

Milnor says *"In this case ratios, and hence cross-ratios, are quite stable"*. As the distance ratios (which Milnor refers to as the similarity transformation) and cross-ratios are invariant (which is in fact a well-known invariant of the Möbius group) and both similarity and Möbius transformation are inside the conformal group that motivated us to study this bigger (conformal) group.

Milnor argues that some biological functions may require essentially the same shape (i.e., rigid transformations) of small parts in order to function. The transformations that are rigid on small parts (i.e., whose derivative equals that of a rigid mapping everywhere) are conformal. The 3D conformal group is finite dimensional. For Milnor this motivated the consideration of the 2D Möbius group. For us, the same argument motivates the consideration of more general 2D conformal mappings.

1.5 Brief overview of the thesis

This thesis consists of six chapters. The first chapter is a brief review of the field of image registration. A few important definitions and symbols that will be used throughout this thesis are included. Applications of image registration such as medical imaging, computer vision, morphometrics and remote sensing are briefly discussed. In this chapter, we also explained the actual motivation of our work which is a construction of conformal diffeomorphism for 2-D images.

In the second chapter, we present image registration in finite dimensional groups that are widely studied and still very popular in the field of image registration. These

---

*Data corrected from Milnor's talk

groups include the *rigid group* (four dimensional) and the *Möbius group* (six dimensional). First we will present mathematical derivations for the representation of the elements in each group and then image registration of our 2D discrete images will be provided. Several practical issues that arise, including interpolation and smoothing of images, and dealing with domain mismatch, will be discussed; these are also needed for later chapters.

Chapters 3, 4, and 5 contain the original research contribution of the thesis, namely the development, testing, and application of two methods of conformal image registration. As far as we have been able to determine, this has never been considered or attempted before.

The third chapter describes our first method of conformal image registration, the *control points* method. Here the values of the desired conformal mapping φ are calculated from a set of control points, which form a coarse grid. The values of the mapping φ at the control points must be constrained in some way so that the mapping is approximately conformal. In Chapter 3 we develop and test four different forms of a penalty term that is added to the objective function to ensure that the mapping is approximately conformal. This chapter contains all the gradual development and difficulties that were faced during this development. Many examples of the development of the method are presented.

Our second method for the construction of conformal diffeomorphism will be presented in the fourth chapter. In this method the desired conformal mapping φ is represented by a finite Taylor series, that is, a polynomial. This method will be called the *gradient flow* method as we compute the minimum of the objective function by a method based on the gradient descent algorithm. As in the method of control points (that is presented in Chapter 3), we present in Chapter 4 the mathematical derivation and demonstrate the efficacy of the gradient flow method on several examples.

After development of two alternative ways of conformal image registration, we then provide a comparison between these methods with several examples in the fifth chapter. Neither method is clearly better on every single example; we discuss the advantages and disadvantages of the two main methods.

The sixth chapter is dedicated to the conclusion and overall summary of the thesis. Further research is also discussed in this chapter.

Chapter 2

Finite Dimensional Image Registration

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

- Albert Einstein (1879-1955)

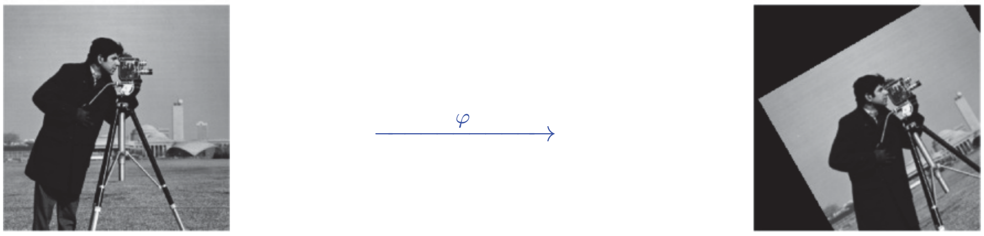
We introduce and describe simple examples of the registration of two-dimensional images in this chapter. The registrations are based on two particular finite dimensional groups, the rigid group and the Möbius group. In the next section, we present the essential steps of image registration based on discrete greyscale images.

2.1 Image Registration in Practice

In this section we start to develop the methods, both mathematical and computational, that are required in order to perform image registration. One very useful tool to visualise the alignment between two images is the difference image (we call it difference or subtraction as a title of this image), DI. This computes the difference between the transformed source image $I_1 \circ \varphi^{-1}$ and the target image I_2 and then scales it to lie in the range [0,1]:

$$DI(x, y) = \frac{1}{2}((I_1 \circ \varphi^{-1}(x, y) - I_2(x, y)) + 1). \tag{2.1}$$

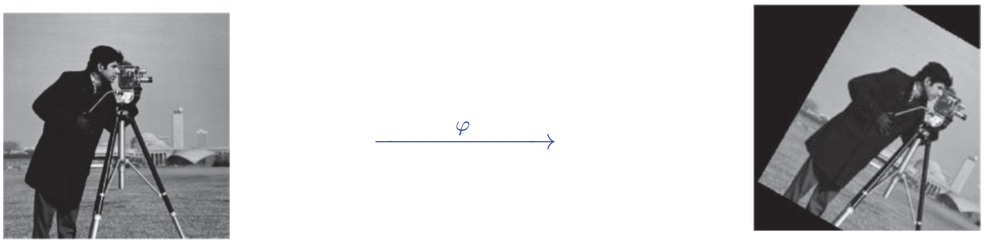
As pixel intensities lie in the range 0 to 1, a pixel that matches perfectly (has the same intensity) in the two images will have intensity 0.5 in the difference image, which corresponds to mid-grey. Thus, a uniform mid-grey block shows a perfect registration. Simplified examples (after applying a rotation and translation) of perfect image registration and poor image registration are illustrated in figures (2.1) and (2.2) respectively[†].



(a) Source (b) Transformed source

(c) Target (d) Difference

Figure 2.1: Set of images indicating a perfect registration in which a transformation φ is applied to the source and the transformed source is apparently identical to the destination image or target. This transformed source is subtracted from the target (using Equation (2.1)) and a mid-grey screen is obtained which is labelled as 'Difference'.



(a) Source (b) Transformed source

(c) Target (d) Difference

Figure 2.2: Set of images indicating a poor registration because the transformed source is not aligned with the target. In fact, the transformed source is rotated in the wrong direction. The poor registration is demonstrated by the fact that the difference image is not blank.

[†] This image of the cameraman is one of the standard Matlab images.

2.1.1 Missing values in image registration

As the images are defined on a finite subset of the plane, there will be transformations that move regions of the image out of the domain, and that move regions that are currently undefined from the background into the domain. The first is not a problem, since those portions of the image are no longer seen, but the second one is a difficulty and these regions of the image mark those points where the background is pulled in to the image, and so no colour information is available for this region. This is known as the missing value problem [74]. Figure 2.3 illustrates the problem, and also one possible solution to it.

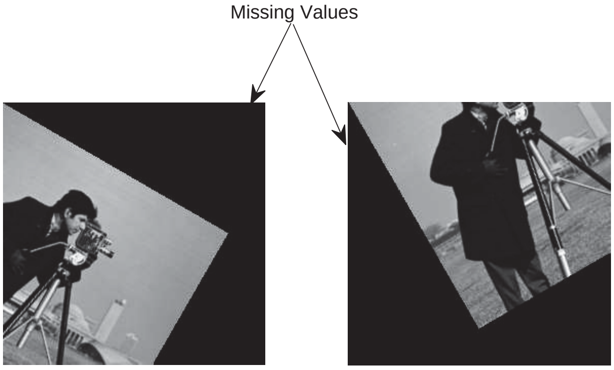


Figure 2.3: Missing values are shown in a pair of images that are obtained after two different rotations of the source image of Figure 2.1. These missing values have been set to 0 (black).

Our numerical computation is based on Matlab and this software assigns 'NaN' (Not-a-Number) to every missing value. Therefore, no numerical computation is possible for these points [67]. In image registration these missing values can be dealt with by the following two options:

- Set an arbitrary number c corresponding to all missing values. For example, in our greyscale images that number c can be assigned in the interval [0 1] or it can be set as a background colour of images such as: pixel intensity 1 in case of white background and 0 for black background. We will use this approach in Chapter 2.
- We can also ignore the missing values (as we ignore those values that are mapped out of the domain) and minimize the norm of the difference of those pixels that do overlap. This means that the sum is only over the overlap pixels rather than all of them. However, it was found that by itself this method does not work, as the objective function can decrease when difficult, non-matching parts of the source image are moved out of the domain. Therefore, it is necessary to rescale

the objective function by the number of overlapping pixels, and use:

$$\widetilde{E}(\varphi) := \frac{N^2}{O}\sum_{i=1}^{N}\sum_{j=1}^{N}\{(I_1 \circ \varphi^{-1})(x_{ij}) - I_2(x_{ij})\}^2, \quad x_{ij} \in S \subset \Omega. \tag{2.2}$$

where S denotes the discrete domain (see Section 2.2), N^2 denotes the number of pixels in each image (recall from Section 1.1.1 that we consider square images, and that the source and target images must have the same number of pixels), and $O \leq N^2$ indicates the number of overlapping pixels. Thus, we are including the information about how much of the images overlap as a multiplier of the objective function. This approach will be used in Chapters 3, 4 and 5. When images are plotted in this thesis, pixels not in the overlap (i.e. NaN pixels) are shown as black pixels.

2.1.2 Image interpolation

Once the deformed grid has been obtained (by applying a transformation to the uniform grid, see Example 2.1), the next step is to compute the transformed source $\varphi.I_1$. We assume that the values of $y_{ij} := \varphi^{-1}(x_{ij})$ are known, where $x_{ij} \in S$ are the pixel locations. It is desired to calculate $I_1(y_{ij})$; however, typically $y_{ij} \notin S$ is not a pixel location. Only the pixel values $I_1(x_{ij})$ are known. The required computation is known as re-sampling or image interpolation. There are several methods for image interpolation such as: *bilinear* interpolation, *spline* interpolation, and *multiscale* interpolation [74]. Though spline interpolation based on *cubic splines* provides greater smoothness and accuracy for smooth images, we prefer *bilinear* interpolation, because it is relatively computationally cheap.

Bilinear interpolation works as follows. Suppose that an image $f(\mathbf{x}) = I(\mathrm{x}, \mathrm{y}), \mathbf{x} \in \mathbb{R}^2$ is defined on a uniform grid that consists of sixteen grid points, as shown in Figure 2.4, i.e., we know the value of this image only on the 4×4 grid points. If we need to interpolate the value of this image at a point $P = (x, y)$ (marked with the blue circle in Figure 2.4). For bilinear interpolation we need to compute the values of image I corresponding to the four neighbouring grids points (marked with the red circles. We label these four values of f as $f_{11} = f(\mathrm{x}_1,\mathrm{y}_1)$, $f_{21} = f(\mathrm{x}_2,\mathrm{y}_1)$,$f_{12} = f(\mathrm{x}_1,\mathrm{y}_2)$ and $f_{22} = f(\mathrm{x}_2,\mathrm{y}_2)$ on a grid. We now identify P_1 and P_2, which have the same x coordinate as P, but have y coordinates y_1 and y_2 respectively. They are labelled marked in green in the figure.

To compute the interpolated value of the function f at point P, we first perform the two linear interpolations in the x-direction for the points P_1 and P_2. The Lagrangian polynomial for interpolation [112] (p. no. 28), [70] yields the values of f at P_1 and P_2

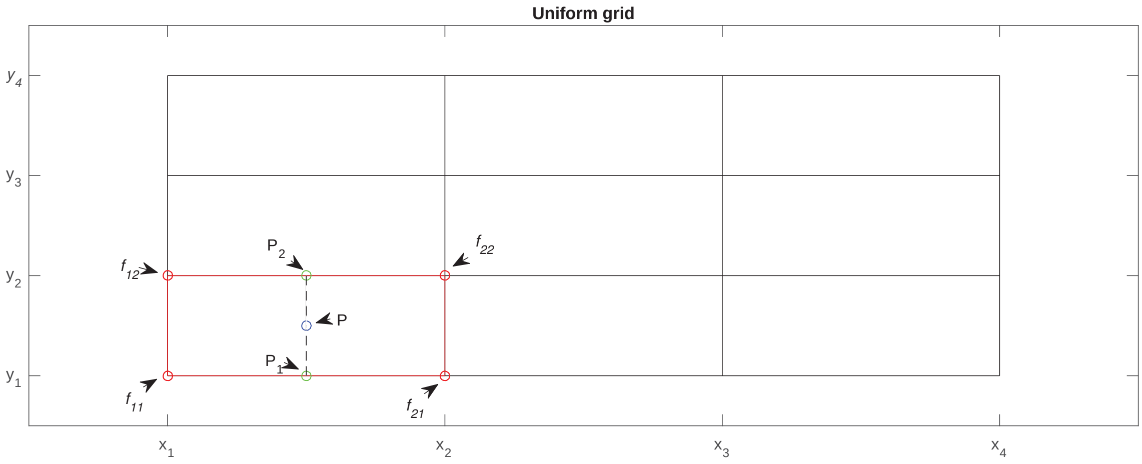


Figure 2.4: A uniform grid represents sixteen grid points in which four known values of a function f are marked with red circles corresponding to a set of grid points. A point 'P' at which f needs to be determined is marked with a blue circle. Two green circles represent two points P_1 and P_2 on a vertical line passing through a point P at which the function value is calculated (with the help of linear interpolation) before its computation at a point P.

as:

$$\begin{aligned} f(x,y_1) &= \frac{(x_2-x)}{(x_2-x_1)}f(x_1,y_1)+\frac{(x-x_1)}{(x_2-x_1)}f(x_2,y_1), \qquad [\text{at} \quad P_1] \\ &= \frac{(x_2-x)}{(x_2-x_1)}f_{11}+\frac{(x-x_1)}{(x_2-x_1)}f_{21}. \end{aligned} \tag{2.3}$$

Similarly,

$$f(x,y_2) = \frac{(x_2-x)}{(x_2-x_1)}f_{12}+\frac{(x-x_1)}{(x_2-x_1)}f_{22}. \qquad [\text{at} \quad P_2] \tag{2.4}$$

Eqs. 2.3 and 2.4 give the values of f at P_1 and P_2. We now use these values and compute the interpolation in the y-direction as:

$$f(x,y) = \frac{(y_2-y)}{(y_2-y_1)}f(x,y_1)+\frac{(y-y_1)}{(y_2-y_1)}f(x,y_2).$$

From eqs. 2.3 and 2.4, we get:

$$\begin{aligned} f(x,y) = \frac{1}{(x_2-x_1)(y_2-y_1)}\{&(x_2-x)(y_2-y)f_{11}+(x-x_1)(y_2-y)f_{21} \\ &+(x_2-x)(y-y_1)f_{12}+(x-x_1)(y-y_1)f_{22}\}. \end{aligned} \tag{2.5}$$

Remark. We will get the same result if the interpolation is done first along the y-direction and then along the x-direction. Equation (2.5) represents 2 dimensional bilinear interpolation for f corresponding to a point P=(x,y) in the presence of four known neighbouring points. All the examples that are presented in this thesis are based on Matlab code. The `interp2` function provides bilinear interpolation based on the formula in Equation (2.5). When we use this command we need to define the domain on which an image `I` is defined and the set of points at which the image `I` needs to be determined. Our discrete images are defined in 2D therefore the domain can be defined with two matrices, say $\mathtt{x}_{\mathtt{input}}$ and $\mathtt{y}_{\mathtt{input}}$ (which in fact represent horizontal and vertical position of the pixels in images `I`). Similarly, the output matrices are $\mathtt{x}_{\mathtt{output}}$ and $\mathtt{y}_{\mathtt{output}}$. Now the value of image `I` over the output matrices can be determined with the following command:

$\mathtt{I}_{\mathtt{output}} = \mathtt{interp2}(\mathtt{x}_{\mathtt{input}}, \mathtt{y}_{\mathtt{input}}, \mathtt{I}, \mathtt{x}_{\mathtt{output}}, \mathtt{y}_{\mathtt{output}})$

Example 2.1. A rigid transformation (see Section 2.2) is applied to the grid on which the cameraman image is defined and then the image is interpolated onto the transformed grid using Matlab's `interp2` command. The results are given in Figure 2.5.

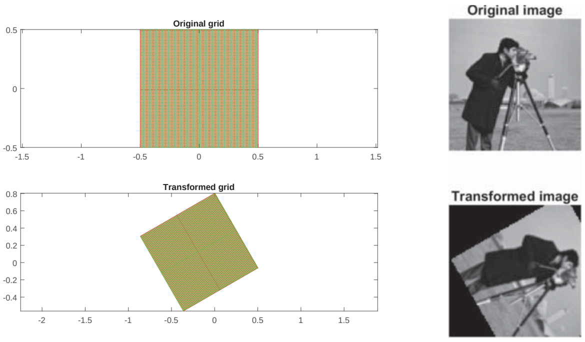


Figure 2.5: *Top:* Left grid represents the domain at which transformation is applied. On the right, the original image is given which is defined over this grid. *Bottom:* Left grid represents the transformed form of the original grid (after applying a rigid transformation with parameters rotation $\theta = -\pi/3^*$, scale $\lambda = 1$, translation $t_x = 0.2, t_y = 0.1$; rigid transformations will be discussed in the next section). The right image is the interpolation of the original image over this deformed grid.

2.2 Image registration using the rigid group

The rigid group is one of the most widely used in image registration. This is the group of uniform scalings, rotations, and arbitrary translations [21] (p. no. 334). We first present each of these separately.

---

[*] Recall from Section 1.2.4 that the action of diffeomorphisms over an image requires φ^{-1}. Therefore, $-\pi/3$ is the angle of rotation for φ and $\pi/3$ is the angle of rotation for φ^{-1} .

Step 1 (Rotation)

Consider an arbitrary point $\begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^2$. Let $\theta \in [0, 2\pi)$ represent an angle of rotation in the anticlockwise direction; then the rotation φ_1 of $\begin{pmatrix} x \\ y \end{pmatrix}$ in $\mathbb{R}^2$ is represented as:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \varphi_1\left(\begin{pmatrix} x \\ y \end{pmatrix}\right) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Step 2 (Scaling)

Let $\lambda \in \mathbb{R}^+$ be a non-zero scalar; then the uniform scaling φ_2 of $\begin{pmatrix} x \\ y \end{pmatrix}$ in $\mathbb{R}^2$ is represented as:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \varphi_2\left(\begin{pmatrix} x \\ y \end{pmatrix}\right) = \lambda \begin{pmatrix} x \\ y \end{pmatrix}$$

Step 3 (Translation)

Let t_x and t_y denote the coefficients of translations along the x and y axes, respectively; then the translation φ_3 of $\begin{pmatrix} x \\ y \end{pmatrix}$ in $\mathbb{R}^2$ is represented as:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \varphi_3\left(\begin{pmatrix} x \\ y \end{pmatrix}\right) = \begin{pmatrix} x + t_x \\ y + t_y \end{pmatrix}$$

The rigid group is the transformation group generated by all of the above transformations, and is therefore easily seen to be given by:

$$\begin{aligned} \begin{pmatrix} x' \\ y' \end{pmatrix} &= \varphi\left(\begin{pmatrix} x \\ y \end{pmatrix}\right) \\ &= (\varphi_3 \circ \varphi_2 \circ \varphi_1) \begin{pmatrix} x \\ y \end{pmatrix} \\ &= \underbrace{\lambda \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}}_{\text{Scaling and Rotation}} + \underbrace{\begin{pmatrix} t_x \\ t_y \end{pmatrix}}_{\text{Translation}}, \quad \theta \in [0, 2\pi), 0 \neq \lambda \in \mathbb{R}^+. \end{aligned} \tag{2.6}$$

The set of all such φ form a group under composition and this group is called the rigid group. The inverse of a rigid transformation is given by (from Equation (2.6)):

$$\begin{aligned}\begin{pmatrix} x \\ y \end{pmatrix} &= \frac{1}{\lambda}\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}^{-1}\left\{\begin{pmatrix} x' \\ y' \end{pmatrix} - \begin{pmatrix} t_x \\ t_y \end{pmatrix}\right\} \\ &= \frac{1}{\lambda}\begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}\left\{\begin{pmatrix} x' \\ y' \end{pmatrix} - \begin{pmatrix} t_x \\ t_y \end{pmatrix}\right\} \qquad (2.7)\end{aligned}$$

As both φ and φ^{-1} are diffeomorphisms, the rigid group is a subgroup of the group of all diffeomorphisms of $\mathbb{R}^2$.

Definition 2.1. Suppose I_1, I_2 are two greyscale images. As in Chapter 1, we define a matching functional that computes the discrepancy between the images as:

$$E(\varphi) = \int_\Omega \left((I_1 \circ \varphi^{-1})(\mathbf{x}) - I_2(\mathbf{x})\right)^2 dx\, dy, \quad \mathbf{x} = (x, y)^T \in \Omega \subset \mathbb{R}^2. \qquad (2.8)$$

If φ represents an element of the rigid group that we discussed above, then the process in which we find a φ that minimises Equation (2.8) is called a rigid registration.

Equation (2.8) is the continuous form of the objective function. For the purposes of numerical computation, we need to discretise this continuous form. Therefore, we define the discrete domain that we call S. This is the domain of the discrete images. Let N be a positive integer and let $S = \{([0, N-1]/(N-1) - 0.5) \times ([0, N-1]/(N-1) - 0.5)\}$ be a uniform square grid that consists of N^2 grid points, an example with $N = 100$ is illustrated in Figure 2.6. Throughout this thesis we will use $N = 100$ unless otherwise stated.

The discrete objective function is (where x_{ij} denotes the image element at position (i, j) of the image matrix):

$$E(\varphi) = \sum_{i=1}^{N}\sum_{j=1}^{N} \left((I_1 \circ \varphi^{-1})(x_{ij}) - I_2(x_{ij})\right)^2. \qquad (2.9)$$

Note that the values of I_1 are known only at the discrete grid points x_{ij}, not at $\varphi^{-1}(x_{ij})$. In addition, even at the continuous level, the domain of $I_1 \circ \varphi^{-1}$ is not Ω, the domain of I_1, but $\varphi(\Omega)$. Because Ω is a restricted subset of $\mathbb{R}^2$ it is possible for points of $I_1 \circ \varphi^{-1}$ to move outside Ω. Therefore for computational purposes, in this chapter we use the

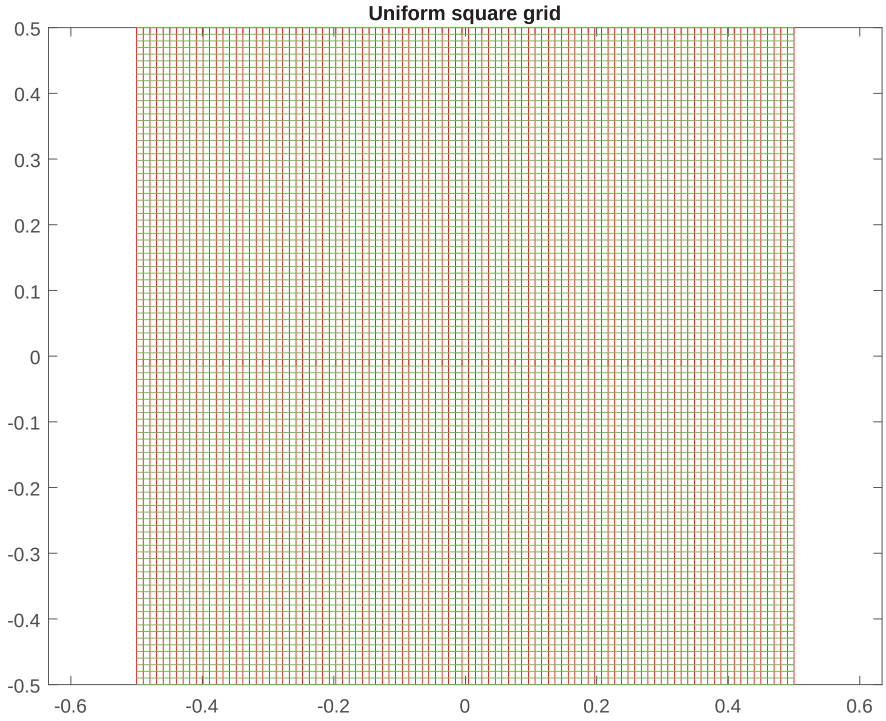


Figure 2.6: Square grid of 10,000 points in the range $[-0.5, 0.5] \times [-0.5, 0.5]$ with $N = 100$, i.e., 100 points per side.

following values for $I_1 \circ \varphi^{-1}(x_{ij})$:

1. If $\varphi^{-1}(x_{ij}) \in \Omega$, we use bilinear interpolation from the values of I_1 at the 4 vertices of the square containing $\varphi^{-1}(x_{ij})$;
2. If $\varphi^{-1}(x_{ij}) \notin \Omega$, we use a constant background value.

In the examples of registration in this chapter, the images are all assumed to have constant backgrounds, either 0 (black) or 1 (white), although in some cases the image does not completely reach its background value before the edges of the image are reached. In the next chapter we will see that this simple treatment of the 'missing' values of the transformed image is not always sufficient and we will use an alternative approach (see Equation (2.2)).

Our goal is to minimize $E(\varphi)$ for $\varphi \in G$. Equation (2.9) denotes the general form of the least-squares optimisation function for any choice of planar transformation group (That is, G is a Lie group that acts on the plane; in this chapter, G will be either the rigid group or the Möbius group). This is a numerical optimisation problem. Optimisation is a huge field, with many known algorithms, whose applicability depends on the nature of the objective function. Our purpose here is not to survey this field but to demonstrate some simple applications. We will consider two methods in this chapter: (i) coarse search, and (ii) the *trust-region-reflective* method for nonlinear least squares problems, as implemented in Matlab's optimisation toolbox.

2.2.1 Using coarse search

In the coarse search method the objective function is evaluated at many points $\varphi \in G$; Algorithm 1 presents an overview of the method, which consists of four nested loops. A decision needs to be made as to how many points φ to check, and how they should be distributed in G. The rigid group is 4-dimensional for registration of two dimensional images, and the coarse search can take φ to lie in a grid of values of θ, λ, t_x, and t_y, as sketched in Algorithm 1. The number of values computed in the algorithm are $M_\theta = 21, M_\lambda = 13, M_x = 11, M_y = 11$, hence the search is over the product of these sizes, requiring a total of 33,033 evaluations of the objective function. Thus, although the coarse search method is extremely simple, it is necessary to search over relatively small numbers of values for each parameter or else the computational cost becomes prohibitive.

Algorithm 1: Coarse search approach to minimising Equation (2.9)

```
input  : $I_1$ and $I_2$: source and target images
output: Warp $\varphi^{-1}$ and deformed image $I_1 \circ \varphi^{-1}$
for $\theta = 0 : \pi/10 : 2\pi$ do
    for $\lambda = 0.1 : 0.1 : 1.3$ do
        for $t_x = -0.5 : 0.1 : 0.5$ do
            for $t_y = -0.5 : 0.1 : 0.5$ do
                use bi-linear interpolation to compute transformed version of the
                  source, $I_1 \circ \varphi^{-1}(x_{ij}) \forall x_{ij} \in S$
                compute $d = \left\| I_1 \circ \tilde{\varphi}^{-1}(x_{ij}) - I_2(x_{ij}) \right\|^2 \forall x_{ij} \in S$
for the minimum value of $d$, compute $\varphi^{-1} = \tilde{\varphi}^{-1}$
compute $I_1 \circ \varphi^{-1}$
```

In the light of modern research on optimisation it might seem to be not worth bothering with. In fact, it is still worth considering. First, it is unnecessary to calculate the derivative, i.e., the function φ does not necessarily have to be differentiable. Second and more important, the function φ may have numerous local minima scattered all over the domain and even nested in a fractal-like way. This can make it very difficult for local, derivative-based methods like gradient descent to find the global minimum. Coarse search does at least survey the whole domain, or a large part of it, and can provide a good starting guess or guesses for more sophisticated methods.

Rigid Registration using Algorithm (1)

Example 2.2. In this first example, we use a 2D Gaussian function as an example of a smooth image. We call this Gaussian the source. We now apply a rigid transformation to this source in order to generate the target. The parameters values that are used for the rigid transformation are $\theta = \pi/2, \lambda = 0.8, t_x = 0.2$ and $t_y = 0.4$. Both images, the source and the target, are shown in Figure 2.7.

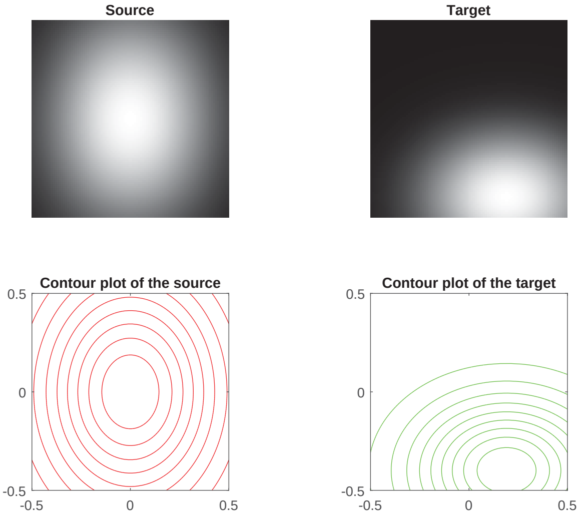


Figure 2.7: Images for Example 2.2. In the first row, the source image, the Gaussian $\exp(-5x^2 - 3y^2)$, is shown on the left; the target image, a rigid transformation of the source, is shown on the right. The second row contains corresponding contour plots of these images.

We now apply Algorithm 1 to this pair of images for image registration. At the end of the optimisation we have a set of parameters corresponding to the minimum value (over the set that the algorithm searched over) of the error in registration. These parameters values are $\theta_{opt} = \pi/2, \lambda_{opt} = 0.8, t_{x_{opt}} = 0.2$ and $t_{y_{opt}} = 0.4$. We use these optimised parameters and transform the source by the rigid transformation they specify, and then subtract it from the target in order to investigate whether the transformed source is aligned with the target or not. This means that large discrepancies between the images appear as black or white, while a perfect match is indicated by mid-grey pixels. The image registration results are given in Figure 2.8. As expected, a perfect registration is obtained, as shown by the image plot of $I_1 \circ \varphi^{-1} - I_2$ (a constant grey, indicating 0.5), and the identical contour plots of $I_1 \circ \varphi^{-1}$ (the transformed source) and I_2 (the target). In Figure 2.9 the values of the objective function for all parameter values are plotted, and the minimum highlighted.

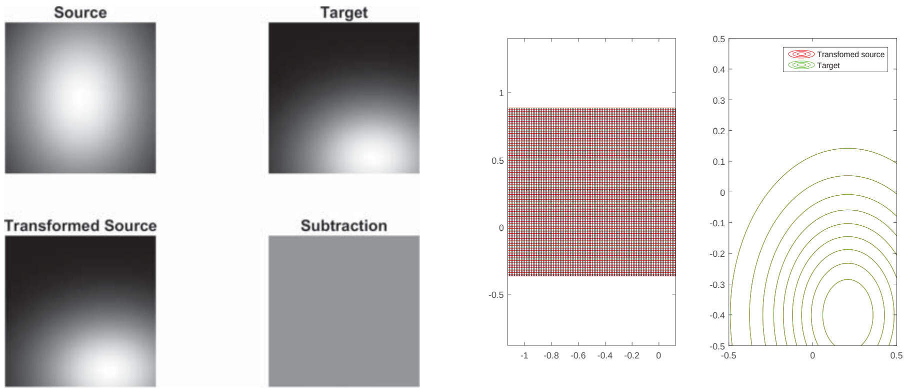


Figure 2.8: Results of rigid registration using coarse search in Example 2.2. The image labelled ‘Subtraction’ on the left is $(I_1 \circ \varphi^{-1} - I_2 + 1)/2$, so that perfect registration shows a uniform grey. The (inverse) transformed grid $\varphi^{-1}(x_{ij})$, $x_{ij} \in S$ is shown on the right, along with the contour plots[‡]of the transformed source and the target.

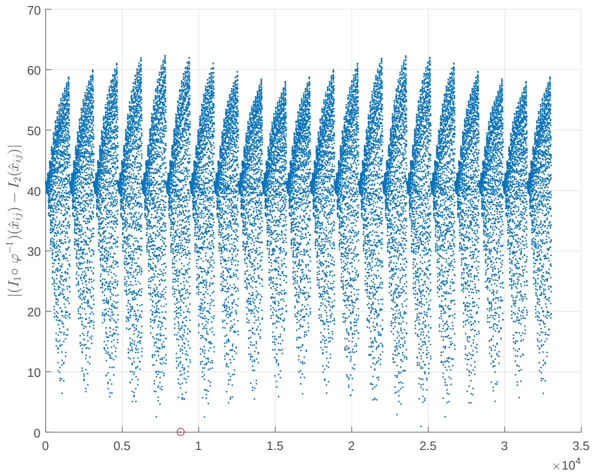


Figure 2.9: Coarse search results for Example 2.2. All of the parameter combinations tested are listed along the x-axis, with the registration errors plotted on the y-axis. The red circle indicates the smallest minimum that the coarse search optimisation obtained.

Example 2.3. In this example, we consider the pair of non-smooth images that are shown in Figure 2.10. As in the previous example, a rigid transformation is applied to the source in order to generate the target. The parameters that are used for rigid transformation are $\theta = \pi/2, \lambda = 1.1, t_x = -0.2$ and $t_y = 0.1$. We now use Algorithm 1 on this example. At the end of optimisation, the optimised parameters that are obtained are $\theta_{opt} = \pi/2, \lambda_{opt} = 1.1, t_{x_{opt}} = -0.2$ and $t_{y_{opt}} = 0.1$. We then use these optimised parameters and transform the source. The image registration results are given in Figure 2.11. As in the previous example, a perfect registration is obtained.

[‡] The contour plots here and elsewhere have been made by plotting red contours first and then the green ones. In cases of perfect registration, no red line can be seen.

Figure 2.10: The non-smooth images used in Example 2.3. The target is a rigid transformation of the source.

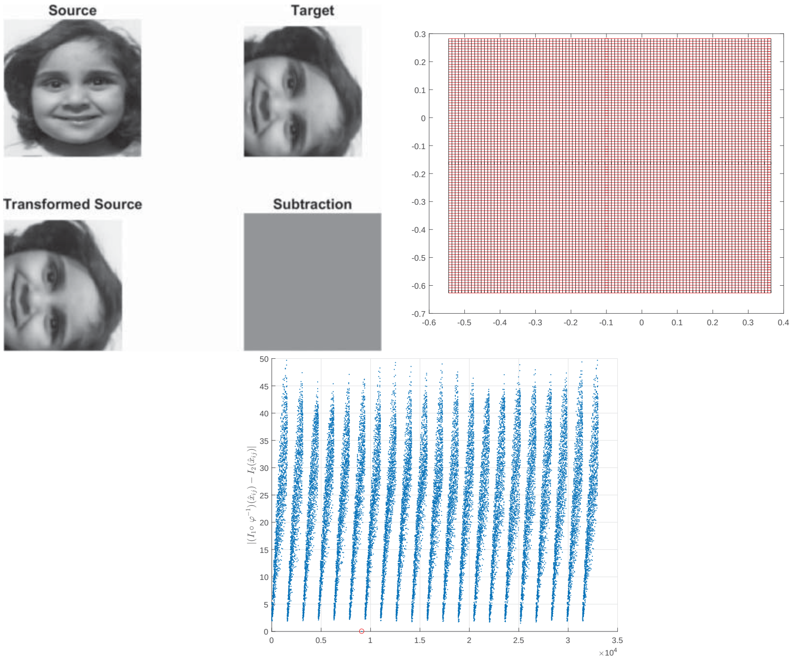


Figure 2.11: *Top:* Results of rigid registration using coarse search for Example 2.3. A perfect registration is obtained. *Bottom:* All values of the objective function for coarse search optimisation for Example 2.3.

Example 2.4. Figure 2.12 shows another pair of non-smooth images in which the target is generated from the source using a rigid transformation. The values of the parameters for this rigid transformation are $\theta = 1.6, \lambda = 1.25, t_x = -0.27$ and $t_y = 0.13$. Algorithm (1) returns the optimised parameters $\theta_{opt} = \pi/2, \lambda_{opt} = 1.2, t_{x_{opt}} = -0.3$ and $t_{y_{opt}} = 0.1$, which are different from those used to construct the target. The problem is that the set of allowable values for the coarse search do not include the actual values used to generate the target. The results of image registration, showing adequate but not perfect registration, are given in Figure 2.13.

Figure 2.12: The non-smooth images used in Example 2.4. The target is a rigid transformation of the source.

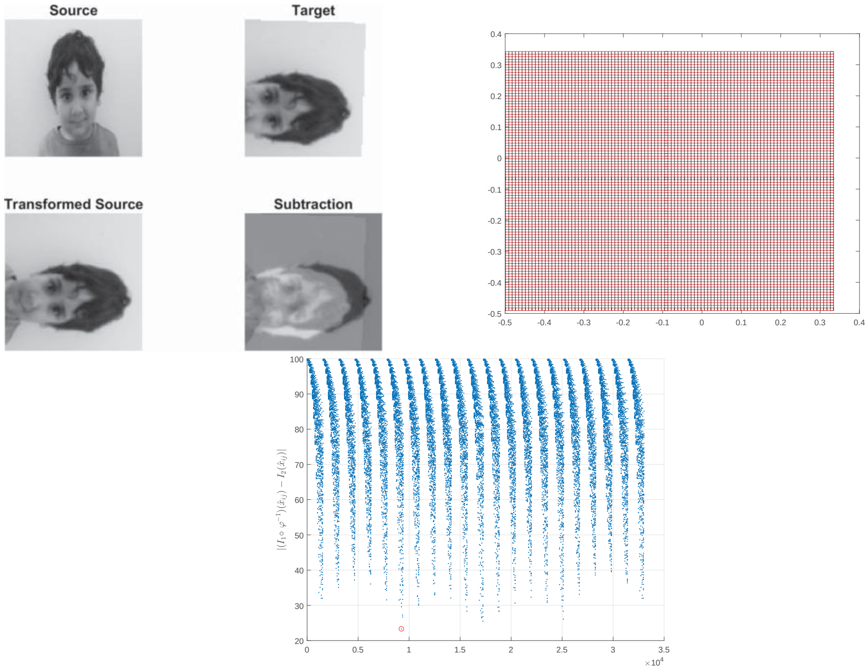


Figure 2.13: *Top:* Results of rigid registration for Example 2.4. A satisfactory, but imperfect, registration is obtained from the coarse search. *Bottom:* All values of the objective function for coarse search optimisation for Example 2.4. Note that the y-axis only goes down to 20, not to 0 as in the previous Examples.

2.2.2 Using gradient descent

We therefore repeat Example 2.4 using the Matlab-based optimiser *lsqnonlin* (throughout the thesis, this optimiser will be used). This optimiser is a nonlinear optimiser for least-squares functions. It is based on the gradient descent method and it is typically able to find a local minimum of the function $f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_n(\mathbf{x}))$ by computing

$$\min \|f(\mathbf{x})\|_2^2 = \min_{\mathbf{x}}(f_1(\mathbf{x})^2 + f_2(\mathbf{x})^2 + \cdots + f_n(\mathbf{x})^2).$$

In fact, *lsqnonlin* can use a variety of built-in algorithms such as '*trust-region-reflective*' and '*levenberg-marquardt*'. The first algorithm cannot solve under-determined systems and works only when the number of equations and number of variables are equal [68]. For under-determined systems *levenberg-marquardt* often works well. In our case, the problem is well-posed because we know that there is a solution, as the target is generated from the source, therefore we use *trust-region-reflective* for image registration.

In order to utilise *lsqnonlin* for image registration, we need an initial guess for optimisation. We start off by setting the identity as an initial guess, i.e., we set $\theta_{ini} = 0, \lambda_{ini} = 1, t_{x_{ini}} = 0, t_{y_{ini}} = 0$ as an initial guess for image registration. The results of the optimisation using *lsqnonlin* are the parameters $\theta_{opt} = 0.2765, \lambda_{opt} = 0.6933, t_{x_{opt}} = -0.0661$ and $t_{y_{opt}} = 0.0805$. The results of this image registration are given in Figure 2.14.

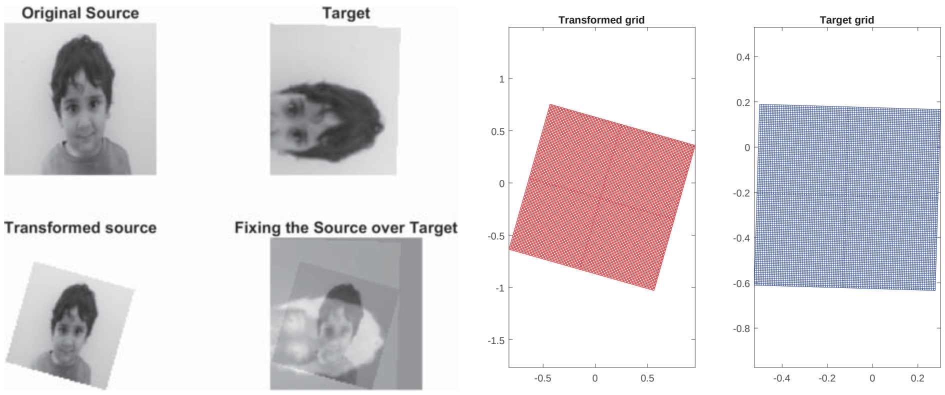


Figure 2.14: Results of least-squares rigid registration for Example 2.4 using *lsqnonlin* with initial guess the identity.

Clearly, the results are very poor. The reason is that the optimiser *lsqnonlin* has found a local minimum near the initial guess. In addition, the optimiser may have had trouble with the non-smoothness of the objective function due to the non-smooth images and the necessary step of interpolation to find the transformed source; we will return to this point later. For now we change the initial guess from the identity to a point nearer to the global minimum, namely, to the results of the optimisation using the *coarse search* method, i.e., $\theta_{ini} = \pi/2, \lambda_{ini} = 1.2, t_{x_{ini}} = -0.3$ and $t_{y_{ini}} = 0.1$. At the end of the optimisation, the optimiser managed to find the parameters corresponding to the global minimum and whose values are $\theta_{opt} = 1.6, \lambda_{opt} = 1.25, t_{x_{opt}} = -0.27$ and $t_{y_{opt}} = 0.13$. Figure 2.15 shows the results of this image registration; a perfect registration is obtained.

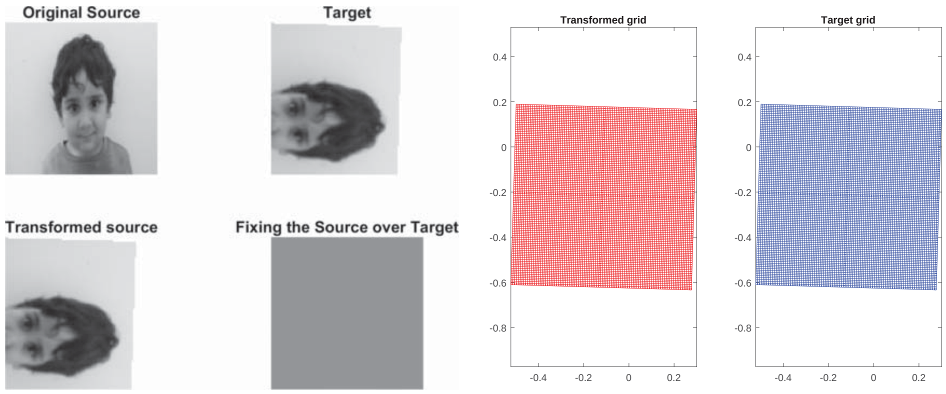


Figure 2.15: Results of least-squares rigid registration for Example 2.4 using *lsqnonlin* with initial guess the best result of coarse search. A perfect registration is obtained.

2.3 Möbius registration

Recall from the introduction that the research goal of this thesis is to develop and apply conformal image registration. The set of conformal maps $\varphi\colon \Omega \to \mathbb{C}$ is infinite dimensional. In this section, as a warm-up problem, we consider image registration by the Möbius group, which is a finite-dimensional group of conformal transformations. We identify the plane $\mathbb{R}^2$ with the complex plane $\mathbb{C}$, and likewise the image domain Ω with a domain of $\mathbb{C}$. In fact, Möbius transformations are best regarded as maps of the extended complex plane $\hat{\mathbb{C}} = \mathbb{C} \cup \tilde{\infty}$ to itself, where $\tilde{\infty}$ denotes complex infinity. This extended complex plane is also know as the Riemann sphere [5], [60] (section 6.3.2, p. no 83), [58].

A Möbius transformation has the form

$$\mathfrak{M}(z) : \hat{\mathbb{C}} \to \hat{\mathbb{C}} = \frac{az+b}{cz+d}; \qquad a, b, c, d \in \hat{\mathbb{C}} \quad \& \quad ad - bc \neq 0. \tag{2.10}$$

Note that if $c = 0$ in Equation (2.10) then this transformation will become a rigid transformation in which the first term $\left(\frac{a}{d}z\right)$ represents the scaling and rotation and the second term $\left(\frac{b}{d}\right)$ represents the translation. That is, the rigid group is a subgroup of the Möbius group. A Möbius transformation can be obtained with the composition of the four simple transformations of scaling, rotation, translation and inversion [41]. In the next proposition we derive Equation (2.10) with the composition of these individual transformations.

Proposition 2.1. *A Möbius transformation is the composition of four simple transformations that are scaling, rotation, translation and inversion.*

Proof. For any complex number $z, \forall a, b, c, d \in \hat{\mathbb{C}}$:

$$f_1(z) = z + \frac{d}{c} \qquad \text{(translation).} \tag{2.11}$$

$$f_2(z) = \frac{1}{z} \qquad \text{(inversion).} \tag{2.12}$$

$$f_3(z) = \frac{-ad+bc}{c^2} z \quad \text{(scaling and rotation).} \tag{2.13}$$

$$f_4(z) = z + \frac{a}{c} \qquad \text{(translation).} \tag{2.14}$$

Now, composition yields:

$$\begin{aligned} f_4 \circ f_3 \circ f_2 \circ f_1(z) &= f_4 \circ f_3 \circ f_2 \left(z + \frac{d}{c} \right), && \text{From 2.11} \\ &= f_4 \circ f_3 \left(\frac{1}{z + d/c} \right), && \text{From 2.12} \\ &= f_4 \left(\frac{-ad+bc}{c(cz+d)} \right), && \text{From 2.13} \\ &= \frac{az+b}{cz+d} \\ &= \mathfrak{M}(z). \end{aligned}$$

□

The inverse of a Möbius transformation is:

$$\mathfrak{M}^{-1}(z) = \frac{dz-b}{a-cz}.$$

which can be seen to be another Möbius transformation. Similarly, it is easy to check that the composition of two Möbius transformations is another Möbius transformation. Since the identity transformation is clearly in the set, thus the set of Möbius transformations forms a group. Although it appears to have 8 real parameters in the 4 complex numbers (a, b, c, d), as the transformation with parameters $(\lambda a, \lambda b, \lambda c, \lambda d)$ is the same as that with parameters (a, b, c, d), for any $\lambda \in \mathbb{C}$, as a *transformation* group it is 6 dimensional. A popular choice of normalization for (a, b, c, d) is to take $ad - bc = 1$. In our work, since Ω is centred on the origin and the singularity of $\mathfrak{M}(z)$ is at $z = -d/c$, which we want to be outside Ω, we take $d = 1$ and use the parameterization

$$\mathfrak{M}(z) = \frac{az+b}{cz+1},$$

where, to avoid singularities, $-1/c \notin \Omega$.

We now consider how to perform image registration using the Möbius group as the set of allowable transformations.

2.3.1 Image Registration with the Möbius Group

In theory there are no significant differences between using the rigid group and the Möbius group for image registration. However, in practice the strategy used for rigid registration—a combination of coarse search and gradient descent—needs to be adapted because of the higher dimensionality of the Möbius group. A grid of M points in each parameter would need M^6 function evaluations. One option would be to decrease the value of M. However, instead we develop a new method to get the initial guess.

As with rigid registration, we will first consider smooth synthetic data, in which the target is generated from the source by a Möbius transformation; then non-smooth synthetic data; then non-smooth data in which the target and the source are not known to be related by a Möbius transformation.

Example 2.5. Figure 2.16 shows a pair of smooth images. In this example, the target is generated from the source by implementing a six dimensional Möbius transformation with coefficients $a = \sqrt{3/2} + \mathrm{i}/2, b = 0.2 - 0.3\mathrm{i}, c = 0.1 - 0.2\mathrm{i}$.

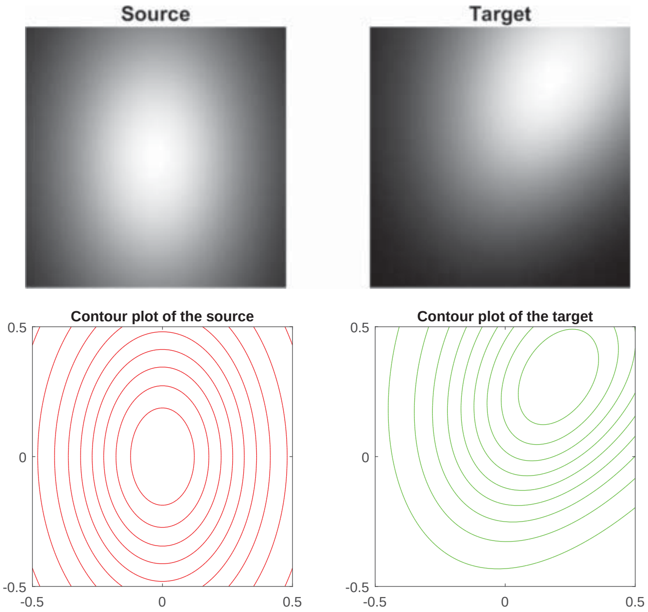


Figure 2.16: Data for Example 2.5. The source is a Gaussian and the target is a Möbius transformation of the source.

We choose the identity as initial guess, i.e, $a_{ini} = 1, b_{ini} = 0, c_{ini} = 0$. The Matlab optimiser *lsqnonlin* returns values corresponding numerically to $a_{opt} = (\sqrt{3/2} + \mathrm{i}/2), b_{opt} = 0.2 - 0.3\mathrm{i}, c_{opt} = 0.1 - 0.2\mathrm{i}$. The results of image registration are given in Figure 2.17; they show a perfect registration. Thus, the optimiser can in this case locate the global minimum starting from the identity.

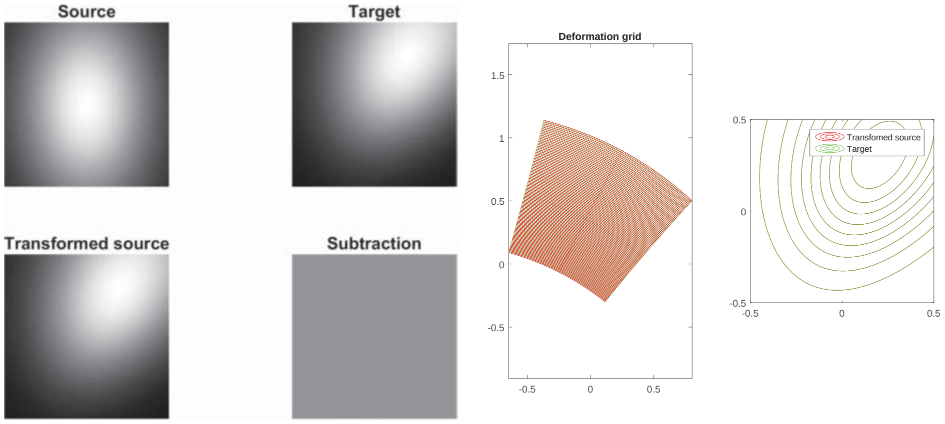


Figure 2.17: Results of Möbius registration for Example 2.5, with *lsqnonlin* and initial guess the identity. A perfect registration is obtained.

Example 2.6. In this example, we consider a pair of non-smooth images in which the target is generated with the source by using a Möbius transformation with coefficients $a = \sqrt{3/2} + \mathrm{i}/2, b = 0.2 - 0.3\mathrm{i}, c = 0.7 - 0.6\mathrm{i}$. These images are given in Figure 2.18.

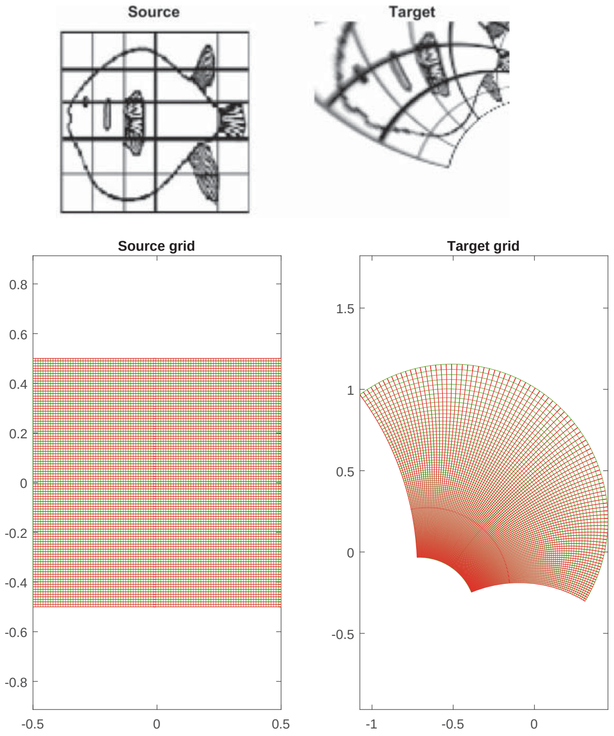


Figure 2.18: Data for Example 2.6, Möbius registration with synthetic non-smooth images. The mapping that generates the source from the target is also shown.

We set the initial guess to be the identity and supply these values to the optimiser. After optimisation the set of optimised values, $a_{opt} = 1.01 - 0.00\mathrm{i}, b_{opt} = 0.002 - 0.002\mathrm{i}, c_{opt} = 0.0038 + 0.0072\mathrm{i}$, are obtained. These values are very near to the identity and we suspect that we will not have a perfect registration. The results of the image registration are given in Figure 2.19; they are very poor, presumably because the optimiser became stuck in a local minimum.

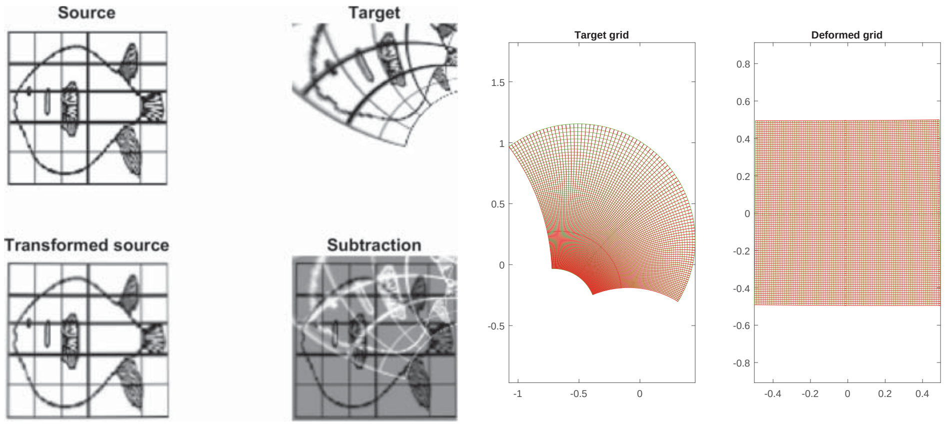


Figure 2.19: Results of first attempt at Möbius registration for Example 2.6. The desired mapping and the mapping computed by *lsqnonlin* are also shown; the results are poor.

We therefore develop an alternative method of obtaining an initial guess called *landmark* or *labelled point matching*. This method is inspired by landmark-based image registration (see [54], [43], [18]) in which corresponding sets of points $z_i \in \Omega$ in the source and z'_i in the target are chosen, and the objective function

$$\sum_i \|\varphi(z_i) - z'_i\|^2$$

is minimized over $\varphi \in G$. As the objective function does not refer to the images at all, the optimisation is significantly easier. On the other hand, the selection of the landmarks is a huge extra task. Automatic selection of landmarks is a major subject of research in the computer science image processing community [38], [89]. Despite extensive research it remains extremely difficult and in practice landmarks are still often selected by hand.

Here we use this approach in a slightly different manner. It is known that there is a Möbius transformation that maps any set of 3 distinct points to any other set of 3 distinct points. We can apply this by limiting ourselves to 3 landmarks. Furthermore, the Möbius transformation in this case can be determined by solving a system of 3 *linear* equations.

Suppose z_i and z'_i, $i = 1, 2, 3$, represent three corresponding points on the source and the target grids respectively. The Möbius transformation mapping z_i to z'_i, $i = 1, 2, 3$, satisfies

$$\begin{aligned} z_i' =& \frac{az+b}{cz+1}, i = 1,2,3 \\ =& az_i + b - cz_i z_i'. \end{aligned} \tag{2.15}$$

The solution of these three linear equations yields parameters a, b, and c that register the three chosen landmarks (but not the images). We use these parameters as the initial guess for the optimisation. In the present example, the chosen landmarks are shown in Figure 2.20 and the results of the optimisation are $a_{opt} = \sqrt{3/2} + \mathrm{i}/2, b_{opt} = 0.2 - 0.3\mathrm{i}, c_{opt} = 0.7 - 0.6\mathrm{i}$. The results of image registration are given in Figure 2.21; a perfect registration is obtained.

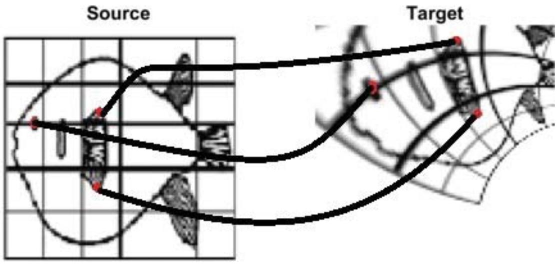


Figure 2.20: Selection of corresponding landmarks on both the fish images.

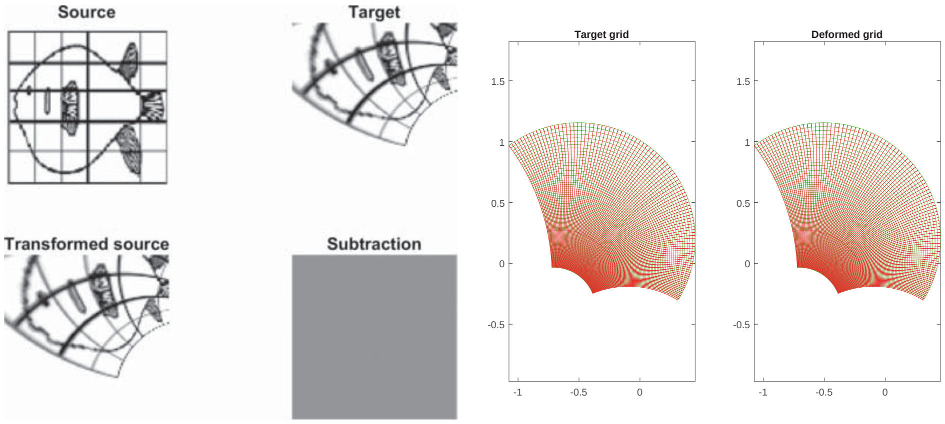


Figure 2.21: Results of the second attempt at Möbius registration for Example 2.6, using an initial guess calculated from landmark matching. A perfect registration is obtained.

Example 2.7. We now apply the landmark method of obtaining an initial guess for optimisation to a more challenging example. In this example the source and target are non-smooth and there is not known to be a Möbius transformation between them. The source is the same fish as in the previous example, but the target is now the second of Thompson's fish. The original images of Thompson's fish have already been shown in

Chapter 1. For the present experiment in image registration the images were scanned and the outline of the boundary of the fish drawn with black marker. Unwanted grid lines were removed digitally. The hand drawn images were scanned and the insides filled with grey using Microsoft Paint. The images were then resized (using Matlab's `imresize` function, which performs bicubic interpolation) to 100 × 100. This pair of fish images are given in Figure 2.22. It can be seen that we have shaded the inside of the fish in grey in order to help the optimisation.

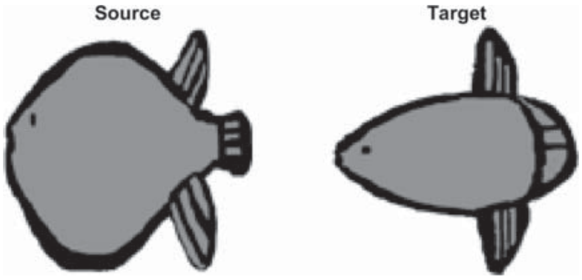


Figure 2.22: Data for Example 2.7 of Möbius registration, a cartoon version of Thompson's fish.

Three landmarks are marked at corresponding locations in the images; these are shown in Figure 2.23.

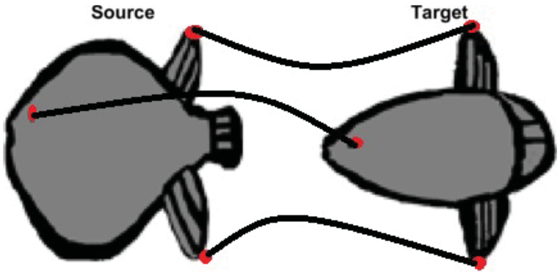


Figure 2.23: Selection of the first set of landmarks on both the images.

Following the method of the previous example, the results of image registration are given in Figure 2.24. The fins and tail are well matched, but the body shape is not. The final step of optimisation has only made a small change to the landmark registration; this can be seen in the small mismatch of the eyes.

We repeat this example using a different set of landmarks, shown in Figure 2.25.

The results are shown in Figure 2.26. Once again the optimiser has only made a small correction to the landmark matching, and the body shapes are not well matched.

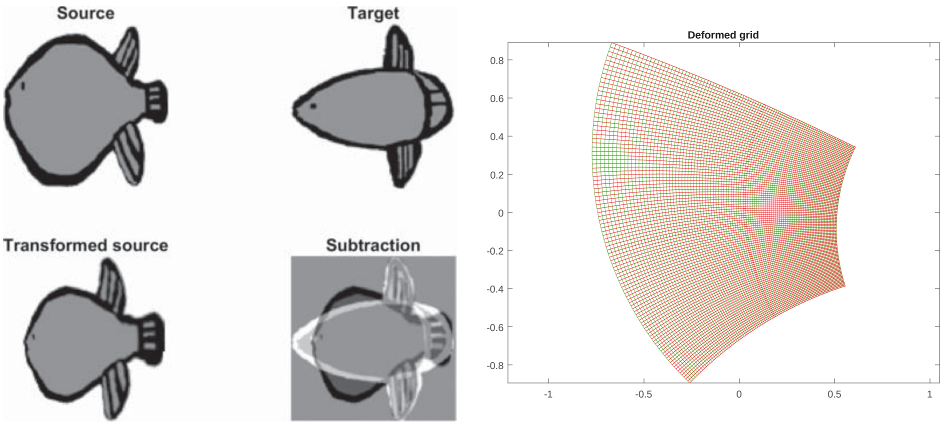


Figure 2.24: Results of Möbius registration for Example 2.7: first set of landmarks.

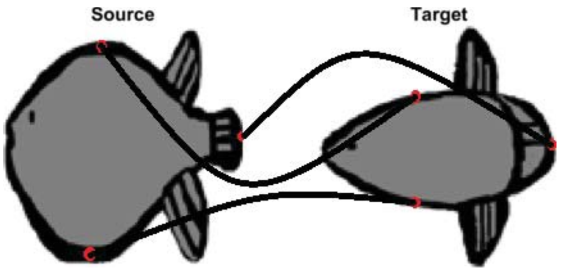


Figure 2.25: Selection of a second set of landmarks on both the images.

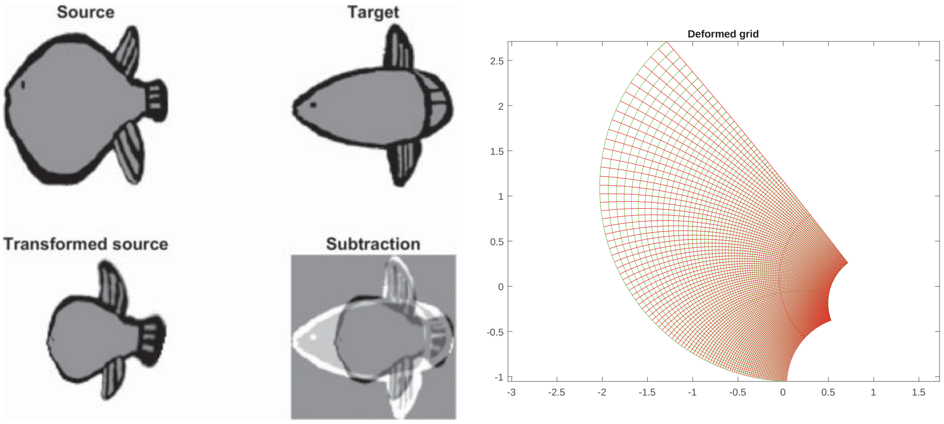


Figure 2.26: Results of Möbius registration for Example 2.7: second set of landmarks.

We now swap the source and the target, see Figure 2.27.

The landmarks are shown in Figure 2.28 and the results of image registration in Figure 2.29.

Again, the fins and tail are well registered, but not the body shape. However, this time the optimiser has made a large change to the landmark matching (see the position of the mouths of the fish). It is striking that the conformal map obtained looks very

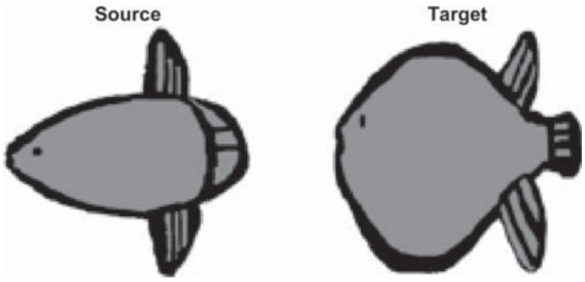


Figure 2.27: Second dataset for Example 2.7: swapped source and target.

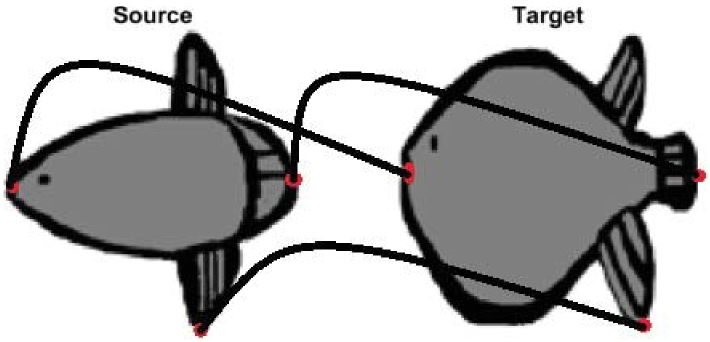


Figure 2.28: Selection of corresponding landmarks: first set.

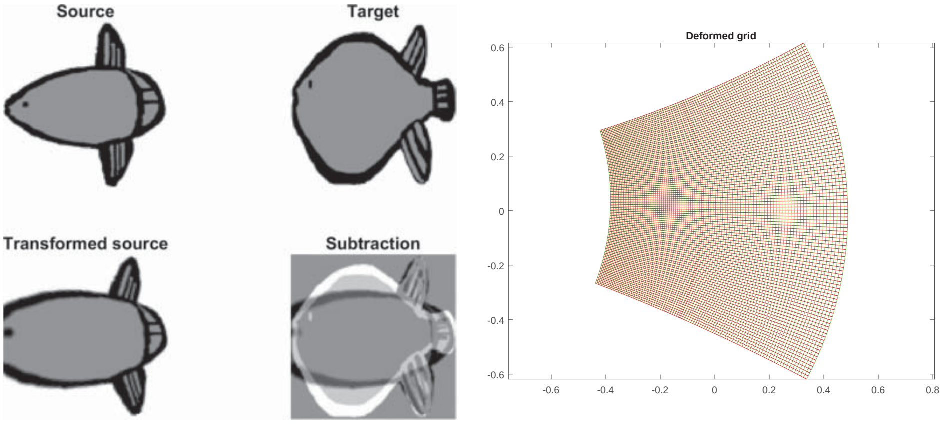


Figure 2.29: Results of Möbius registration for Example 2.7: second dataset, first set of landmarks.

similar to the grid obtained by Thompson and shown in Chapter 1.

As a check, we continue with the swapped source and target and try another set of landmarks, shown in Figure 2.30. The results are given in Figure 2.31. This time the registration is virtually identical to Figure 2.29, indicating that in this case the results are not that sensitive to the choice of initial guess and that a global minimum may have been obtained.

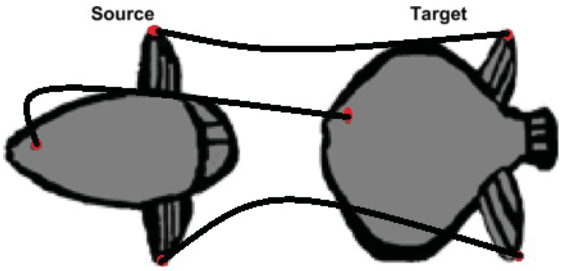


Figure 2.30: Selection of corresponding landmarks: second set.

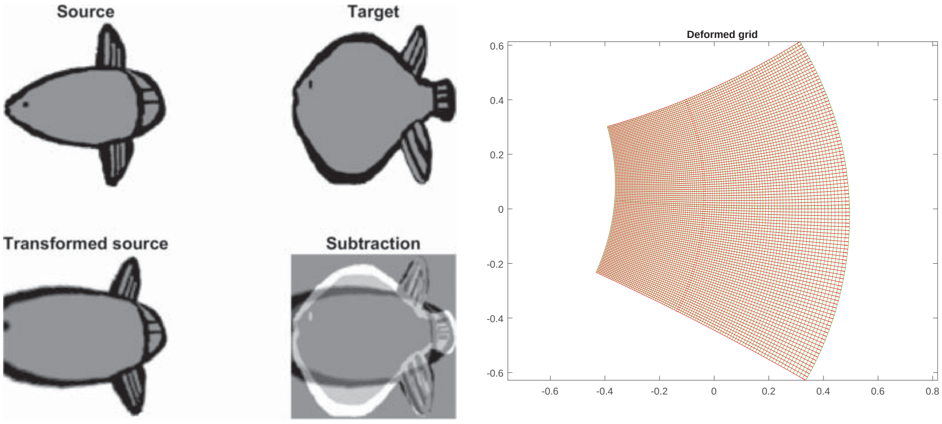


Figure 2.31: Results of Möbius registration for Example 2.7: second dataset, second set of landmarks.

To summarize the results of the chapter, we have found that we are able to register images in two finite-dimensional groups: the similarity group and the Möbius group. In both cases, the choice of initial condition is important to the quality of the registration obtained, and whether or not the algorithm reaches the neighbourhood of the global minimum. Although we have no guarantee, it does appear from the numerical examples that we have reached either the global minimum or something very close to it. The Thompson fish examples are not definitive but it does appear that we have registered the images in the Möbius group and that the poor registrations that we have obtained are evidence that the images are not, in fact, related by a Möbius transformation.

In the following chapters we will consider image registration by a much larger, in fact infinite-dimensional set of conformal maps, that contains the Möbius transformations as a subset, and return to the Thompson fish images.

Chapter 3

Method of Control Points

I do not know what I may appear to the world, but to myself I seem to have been only like a boy playing on the seashore, and diverting myself in now and then finding a smoother pebble or a prettier shell than ordinary, whilst the great ocean of truth lay all undiscovered before me.

-Sir Isaac Newton (1642-1727)

3.1 Introduction

In the previous chapter we discussed image registration using finite dimensional transformation groups. We now proceed further and discuss here and in the next chapter image registration using an infinite dimensional group, namely the group of conformal diffeomorphisms. Recall from Chapter 1 that D'Arcy Thompson, in his book *On Growth and Form* [101], tended to choose transformations from suitable groups. In general, he preferred to keep the transformation, and the class from which it was chosen, as simple as possible.

The basic aim of our research is to construct a conformal diffeomorphism φ that registers two 2-D images. In our research, we have developed, tested, compared, and applied two alternative methods of conformal registration. These are called the *control points method* (discussed in this chapter) and the *gradient flow method* (discussed in the next chapter).

The gradient flow method is explicitly conformal, since we choose the flow field in a conformal way. For example, we can represent φ^{-1} in the form of a Taylor series as we explain in the next chapter. For the method of control points, we choose to use constraints in the optimisation by adding a penalty term to the objective function. This penalty terms forces the diffeomorphism to be approximately conformal. Four different forms of the penalty term are introduced in this chapter and we will discuss them one by one. In the previous chapter, we defined the essential steps of image registration (Section 2.1). Another important part of successful image registration is to use smoothing, which we define below. From now onward, the missing values in all the examples will be treated with the second approach of Section 2.1.1, i.e., computing the objective function only in the overlapping region of the images, but multipling the objective function by the fraction of overlap.

3.1.1 Smoothing

Although images are based on pixels, and thus discrete representations of objects, some of them represent smooth scenes, while others do not. Most images have edges, which are discontinuities in intensity across the image. This can be a challenge for image registration. Examples of smooth images include a Gaussian blob, images of circular functions (trigonometry functions), hyperbolic functions, etc. Figure 3.1 shows a few examples of smooth images, while Figure 3.2 displays a few examples of non-smooth images.

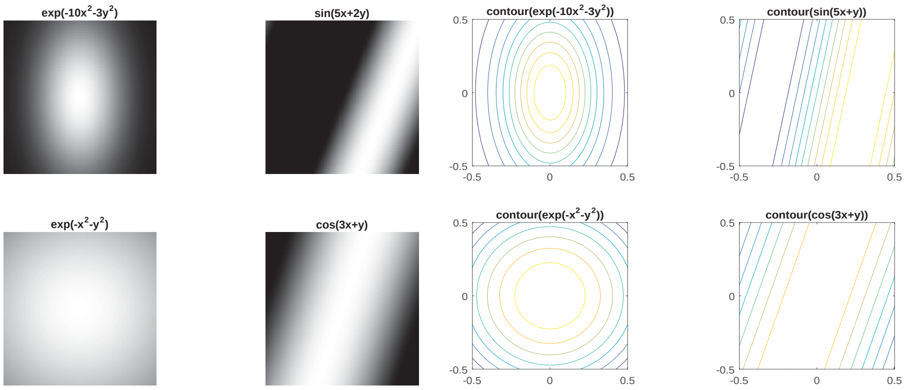


Figure 3.1: *Left:* Example smooth images. *Right:* Corresponding contour plots of the smooth images.

One way to deal with the discontinuities is to remove them by *smoothing* or *filtering*, using methods that are related to image blurring [109]. There are several ways of smoothing such as taking a *moving average*, *exponential smoothing*, Holt-Winters smoothing [99] and kernel based smoothing [108]. Kernel-based smoothing can be used for images whereas, other (mentioned) methods of smoothing are used for statistical data. Examples of kernels that can be used for smoothing include the *Gaussian lowpass filter*, the *averaging filter*, and the *Laplacian filter* [69]. We use a *Gaussian filter* of width (standard deviation) 5 pixels, and a filter of size 15×15 for an image of size 100×100 (these numbers were chosen experimentally). Figure 3.3 shows Gaussian-smoothed versions of the images in Figure 3.2. It can be seen that one effect of smoothing is to blur the images, spreading the change in image intensity corresponding to an edge over several pixels. This has two important effects: it makes the images smoother, and it makes

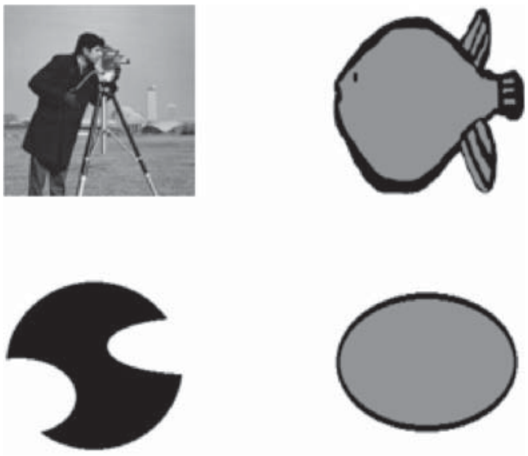


Figure 3.2: Examples of non-smooth images.

the edges overlap more, helping the optimiser.

Figure 3.3: Smoothed images from Figure 3.2 using a 15 × 15 Gaussian filter with standard deviation of 5 pixels.

We will use smoothed versions of images as a pre-processing step to obtain good initial guesses when registering non-smooth images here and in the following chapters. We now define image registration under a conformal diffeomorphism.

3.1.2 Image Registration Using a Conformal Diffeomorphism

Definition 3.1. Suppose I_1, I_2 are two greyscale images. We define the usual objective function:

$$E(\hat{\varphi}) = \int_\Omega \left((I_1 \circ \hat{\varphi}^{-1})(\mathbf{x}) - I_2(\mathbf{x})\right)^2 dx_1\, dx_2, \quad \mathbf{x} = (x_1, x_2)^T \in \Omega \subset \mathbb{R}^2. \tag{3.1}$$

If we constrain $\hat{\varphi} \in \mathrm{con}(\Omega, \mathbb{R}^2)$, that is, the conformal diffeomorphisms of Ω, then a process in which we find a $\hat{\varphi}$ that minimises the function in Equation (3.1) is called a conformal image registration. The question is how to represent the conformal warps.

The task of minimising $E(\hat{\varphi})$ may be realised as a constrained optimisation problem in which $E(\hat{\varphi})$ in Equation (3.1) is defined for all invertible $\hat{\varphi}$, but we wish to find its minimum subject to the constraint that $\hat{\varphi}$ is conformal. We thus choose a penalty term and perform constrained optimisation. This penalty term forces the diffeomorphism to be conformal. From now on, instead of using the objective function given in Equation (3.1), we will use the following objective function:

$$E(\varphi)_{con} = \int_{\Omega} \big((I_1 \circ \varphi^{-1})(\mathbf{x}) - I_2(\mathbf{x})\big)^2 dx_1\, dx_2 + \lambda \int_{\Omega} P(\mathbf{y})^2\, dy_1\, dy_2. \tag{3.2}$$

Here $P(\mathbf{y})$ is a penalty term that vanishes iff φ is a conformal diffeomorphism, i.e., iff φ satisfies the Cauchy–Riemann equations at some points $\mathbf{y}$ and has non-zero derivative. Equation (3.2) is continuous and hence needs to be discretised to enable numerical computation, as is discussed next.

3.2 Control points method

Recall from Chapter 2 that the discrete domain S consists of 10,000 grids point that we are (again) showing in Figure 3.4.

Figure 3.4: Grid of 10,000 points with $N = 100$.

In the control points method, the conformal map φ is represented by its values of

φ^{-1} on a *much coarser* grid than that of the discrete image domain S. Thus, we first choose n^2 grid points $\hat{x}_{ij}$ $(i, j = 1, \ldots, n)$ from the discrete domain S, where $n \ll N$, that are configured in an $n \times n$ grid. We then transform these chosen points under φ^{-1} to generate another set of points of same size. We define two sets of points, the set of chosen points $\hat{x}_{ij} = \{\hat{x}_{11}, \hat{x}_{12}, ..., \hat{x}_{nn}\}$, and the set of transformed points $\hat{y}_{ij} = \{\hat{y}_{11}, \hat{y}_{12}, ..., \hat{y}_{nn}\}$ such that $\hat{y}_{ij} = \varphi^{-1}(\hat{x}_{ij})$. Here $\hat{x}_{ij} \in S$ and $\hat{y}_{ij} \in \mathbb{R}^2$. The variables $\hat{y}_{ij}$, which determine the map φ, are the dependent variables whose value will be determined during the optimisation process. The remaining values of φ^{-1} that are needed to transform the source image, namely $\varphi^{-1}(x_{ij})$, are determined by bilinear interpolation from the known values $\varphi^{-1}(\hat{x}_{ij})$; bilinear interpolation was explained in Section 2.1.2. We illustrate this approach with $n = 4$, i.e. 16 control points.

Step 1 (Selection of control points):

First we select sixteen control points positioned in a 4×4 grid, $\hat{x}_{ij} = \{\hat{x}_{11}, \hat{x}_{12}, ..., \hat{x}_{44}\}$, from the discrete domain S. We choose the points to cover the full extent of S, so that some of the control points lie on the edges of S. Figure 3.5 shows the locations of the 16 control points as blue circles.

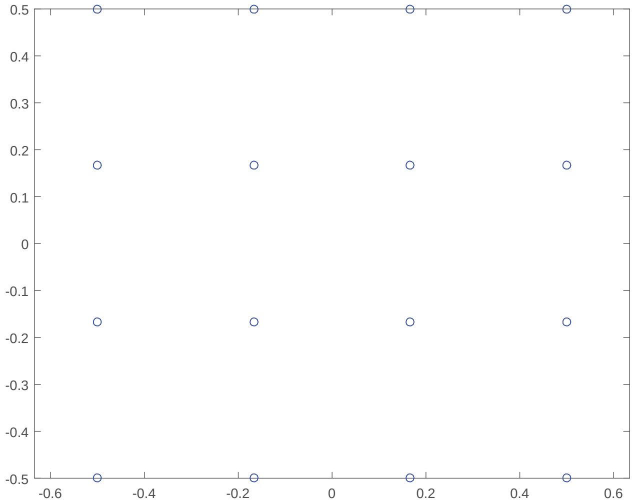


Figure 3.5: Blue circles represent selected control points from the discrete domain S.

Step 2 (Transformation of control points):

We now transform these sixteen points by applying φ^{-1} to generate another set of sixteen points, which we represent by $\hat{y}_{ij} = \{\hat{y}_{11}, \hat{y}_{12}, ..., \hat{y}_{44}\}$. These transformed points are marked with red circles and an example is shown in Figure 3.6.

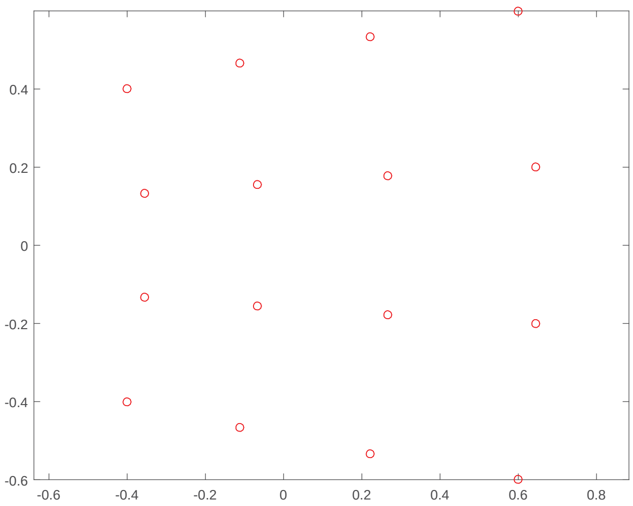


Figure 3.6: Red circles represent the transformed control points under the action of some φ^{-1}.

Step 3 (Bilinear interpolation):

We next apply bilinear interpolation to generate the values of $\varphi^{-1}(x_{ij})$ for all N^2 source points x_{ij}. This transformed grid is shown in Figure 3.7.

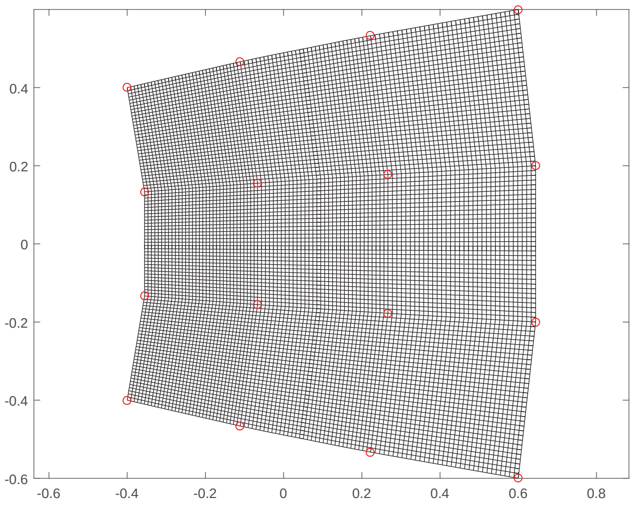


Figure 3.7: Bilinear interpolation over the transformed control points, marked with red circles, was used to generate this grid.

We represent this transformed grid as $y_{ij} = \varphi^{-1}(x_{ij}), \forall x_{ij} \in S$. By combining the above individual steps, we express the whole process of mapping in Figure 3.8.

Step 4 (Resampling of the source):

After step 3, we know the values of $y_{ij} = \varphi^{-1}(x_{ij})$ for $i, j = 1, \ldots, N$. But we

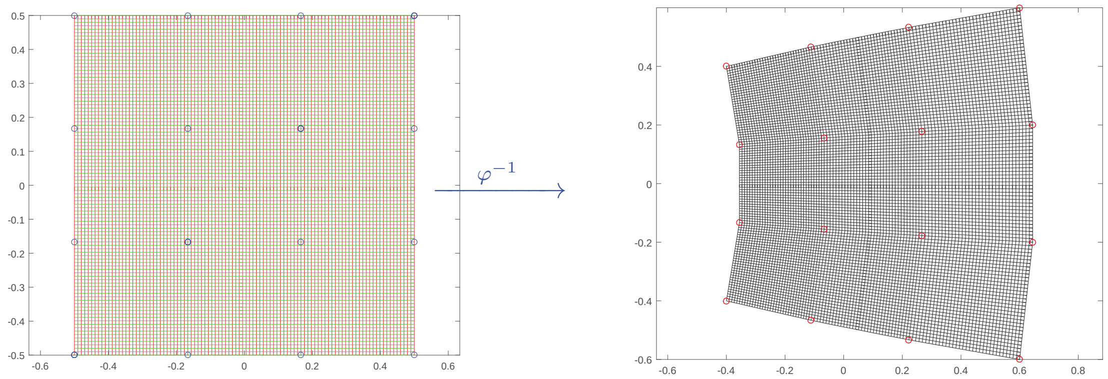


Figure 3.8: In the first grid, the blue circles represent selected control points from the discrete domain. In the second grid, the images of the control points under φ^{-1} are shown with red circles. Bilinear interpolation is used to generate the rest of the grid points between these red circles.

need to know the values of $I_1(y_{ij})$ and the y_{ij} do not usually lie in the discrete domain. Therefore in this step we locate the pixel that contains each point y_{ij} and use bilinear interpolation from the values of I_1 at the four corners of this square to obtain an interpolated value for $I_1(y_{ij})$. This yields an approximate value for $(I_1 \circ \varphi^{-1})(x_{ij})$ for $i, j = 1, \ldots, N$ and hence allows the first term in the discrete objective function to be calculated.

The discrete form of Equation (3.2) is similar to the form used in the previous chapter, except for the addition of the penalty term:

$$E(\varphi) = \sum_{i=1}^{N} \sum_{j=1}^{N} \left((I_1 \circ \varphi^{-1})(x_{ij}) - I_2(x_{ij}) \right)^2 + \lambda \sum_{i=1}^{n} \sum_{j=1}^{n} P(\hat{y}_{ij})^2. \tag{3.3}$$

Equation (3.3) represents the general form of optimisation function that will be used throughout this chapter. It has two parts (both of which are functions of $\hat{y}_{ij}$): the first part measures how well registered the images are, while the second part contains a penalty term P related to the conformality of φ. The two terms are balanced by a parameter λ. We shall see that for different choices of P the second term may be either a constraint (in which case we will be interested in large values of λ so that the constraint is satisfied, or nearly satisfied) or a regularisation term, in which case λ controls the tradeoff between the quality of the registration and the conformality of φ. We minimise Equation (3.3) with respect to $\hat{y}_{ij}$ using the Matlab optimisation function *lsqnonlin*.

The reasons for the introduction of the coarse grid $\hat{y}_{ij}$ are:

1. The cost and difficulty of numerical optimisation increase dramatically with the dimension of the optimisation problem; full optimisation with $2N^2$ degrees of

freedom would be extremely expensive; it would also likely be difficult to locate the global minimum in such a high dimensional space.

2. Allowing $\varphi^{-1}(x_{ij})$ to vary independently for all i, j would be likely to make the problem ill-posed, which would require further regularisation terms to control.

It is to be noted that $n \ll N$. Where n^2 denotes the control points and N^2 represents number of pixels or size of an image. Our general approach is based on the continuation method, pioneered by H B Keller [55] and used here in a very simple form. The continuation parameters are n and λ. For λ there are two possible strategies to consider: (i) starting with a small or zero value of λ and increasing it; and (ii) starting with a large or infinite value of λ and decreasing it. In approach (i), if $\lambda = 0$, then the optimisation problem reduces to the known problem of image registration, that is, the conformality condition has been dropped. In theory, this approach would first find the best registration with a diffeomorphism (which would necessarily be at least as good as the best registration with a conformal diffeomorphism), and then, by increasing λ, relax it towards the best conformal registration. In approach (ii), if λ is very large, we are initially enforcing the constraint or regularisation term, which acts to better control the allowable transformations.

Depending on the choice of penalty term, it seems that both approaches could have some advantages and disadvantages. However, after conducting many numerical experiments, we settled on a uniform strategy of starting with large values of λ (specifically, $\lambda = 20^4$) and decreasing it in steps of factors of 20. The problem is that if starting at $\lambda = 0$ the algorithm may fail to locate a smooth diffeomorphism registering the images even if we know (e.g. because we are using synthetic data) that one exists.

In all the numerical experiments we have started with $n = 4$, i.e. 16 control points, and minimised the objective function in Equation (3.3) for $\lambda = 20^4$, 20^3, 20^2, 20, using the output of each optimisation as the input for the next. We then set $n = 8$, i.e. 64 control points, initialising the values for the new points by using bilinear interpolation from the $n = 4$ solution. The approach is summarised in Algorithm 2.

In the remainder of this chapter we will develop four alternative forms of the penalty term, that is four alternative conceptions of a ‘discrete conformal map’, and test them using this algorithm. The performance of the algorithm may depend on the choice of the penalty term, whether we are using synthetic or real data (in synthetic data, the target is generated from the source using a known transformation, which may or may not be conformal), the smoothness of the images, the quality of the initial guess for the transformation, and how closely related the images are by a conformal map. Thus each form of the penalty term will be tested on a variety of images.

Algorithm 2: Image Registration using Control Points and Equation (3.3)

```
input  : $I_1$ and $I_2$: source and target images
         $N^2$: number of grid points
         $n^2$: number of control points
         $\hat{x}_{ij}$: selected control points from the discrete domain $S$
output: Warp $\varphi^{-1}$ and deformed image $I_1 \circ \varphi^{-1}$
for $n = 4$ do
    initialise $\hat{y}_{ij}$ to $\hat{x}_{ij}+$ small random perturbation
    for $\lambda = 20^4, 20^3, 20^2, 20$: do
        optimize Equation (3.3) to determine $\hat{y}_{ij} = \varphi^{-1}(\hat{x}_{ij})$
for $n = 8$ do
    for $\lambda = 20$ do
        $\hat{x}_{ij}$: initialise the values for the new points by using bilinear interpolation
          from the $n = 4$ solution.
        optimise Equation (3.3) to determine $\hat{y}_{ij} = \varphi^{-1}(\hat{x}_{ij})$
        use bilinear interpolation to produce transformed grid,
          $y_{ij} = \varphi^{-1}(x_{ij}), \forall x_{ij} \in S$.
        use bilinear interpolation to obtain transformed version of the source,
          $I_1 \circ y_{ij}$.
```

3.3 The Cauchy–Riemann equations

Our first three penalty terms are based on discrete forms of the Cauchy–Riemann equations, whereas the fourth penalty term involves a geometrical interpretation of conformality. As a precursor to the first three methods, we first recall the Cauchy-Riemann equations.

If

$$\varphi(x, y) = u(x, y) + \mathrm{i}v(x, y)$$

is a conformal map, then

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} \quad \text{and} \quad \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}. \tag{3.4}$$

Equation (3.4) represents the standard form of the Cauchy–Riemann equations for continuous variables. We can also express them in complex notation. Partial derivatives of φ are defined as:

$$\frac{\partial \varphi}{\partial x} = \frac{\partial u}{\partial x} + \mathrm{i}\frac{\partial v}{\partial x} \tag{3.5}$$

and

$$\frac{\partial \varphi}{\partial y} = \frac{\partial u}{\partial y} + \mathrm{i}\frac{\partial v}{\partial y}. \tag{3.6}$$

Now, equations (3.5) and (3.6) yield:

$$\begin{aligned}
\frac{\partial \varphi}{\partial x} + \mathrm{i}\frac{\partial \varphi}{\partial y} &= \frac{\partial u}{\partial x} + \mathrm{i}\frac{\partial v}{\partial x} + \mathrm{i}\left(\frac{\partial u}{\partial y} + \mathrm{i}\frac{\partial v}{\partial y}\right) \\
&= \frac{\partial u}{\partial x} + \mathrm{i}\frac{\partial v}{\partial x} + \mathrm{i}\frac{\partial u}{\partial y} - \frac{\partial v}{\partial y} \\
&= \frac{\partial v}{\partial y} + \mathrm{i}\frac{\partial v}{\partial x} + \mathrm{i}\left(-\frac{\partial v}{\partial x}\right) - \frac{\partial v}{\partial y}, \quad \text{from Equation (3.4)} \\
&= 0 \\
\implies \frac{\partial \varphi}{\partial x} &= -\mathrm{i}\frac{\partial \varphi}{\partial y} \qquad (3.7)
\end{aligned}$$

Equation (3.7) serves as a complex form of the Cauchy–Riemann equations for continuous variables. For numerical computation we discretise our domain, thus we need a discrete form of the Cauchy–Riemann equations. There are several ways to discretise them, and they give different results, as we shall see. We now present our first discrete form of the Cauchy–Riemann equations

3.4 First discrete form of the Cauchy–Riemann equations

In our first form, the Cauchy–Riemann equations are defined by finite differences at each of the n^2 control points. First we compute the numerical gradient and then the first form of the Cauchy–Riemann equations is determined. Note that we identify $\mathbb{R}^2$ with the complex plane $\mathbb{C}$ as needed.

We calculate the gradient of $\hat{y}_{ij}$ using central differences at the interior points:

$$(\Delta_x \hat{y})_{i,j} = (\hat{y}_{i+1,j} - \hat{y}_{i-1,j})/2, \quad i = 2, 3, ..., n-1. \tag{3.8}$$

$$(\Delta_y \hat{y})_{i,j} = (\hat{y}_{i,j+1} - \hat{y}_{i,j-1})/2, \quad j = 2, 3, ..., n-1. \tag{3.9}$$

and single-sided differences at edge points:

$$(\Delta_x \hat{y})_{1,j} = \hat{y}_{2,j} - \hat{y}_{1,j}, \quad (\Delta_x \hat{y})_{n,j} = \hat{y}_{n,j} - \hat{y}_{n-1,j}$$

$$(\Delta_y \hat{y})_{i,1} = \hat{y}_{i,2} - \hat{y}_{i,1}, \quad (\Delta_y \hat{y})_{i,n} = \hat{y}_{i,n-1} - \hat{y}_{i,n}$$

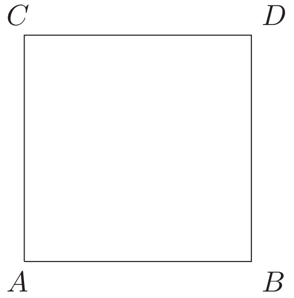
Identifying $\hat{y}$ with φ, the first discrete form of the Cauchy–Riemann equations that we use is:

$$\Delta_x \hat{y}_{ij} = -\mathrm{i}\Delta_y \hat{y}_{ij}, \quad i, j = 1, \dots, n \tag{3.10}$$

Note that we are using here the fact that the reference grid of control points, $\hat{x}_{ij}$, is a uniform square grid, as we use such a grid in all our experiments. The mesh size of this grid has been effectively absorbed into the parameter λ. A non-uniform or non-rectangular mesh would require more complicated finite differences. This discrete form of the Cauchy–Riemann equation is used to define the penalty term as:

$$P_1(\hat{y}_{ij}) = |(\Delta_x \hat{y}_{ij} + \mathrm{i}\Delta_y \hat{y}_{ij})|. \tag{3.11}$$

We found in our numerical experiments that when $\lambda \to \infty$, $\|P(\hat{y})\| \to 0$ and that the discrete maps produced in this way were always rigid transformations, i.e. Euclidean similarities. Before proceeding to the experiments we show this here for the case $n = 2$, i.e., mapping a single square. Label the 4 vertices of this square as follows:



Let A, B, C, D map to $A', B'C', D'$. These are the vertices of a quadrilateral that is obtained after optimisation and which represents the image of the square $ABCD$ under φ^{-1}. Collect the complex coordinates into a matrix as:

$$R = \begin{pmatrix} C' & D' \\ A' & B' \end{pmatrix};$$

then the discrete gradient (denoted by *grad*) of R is:

$$grad\ R = \left\{ \begin{pmatrix} D' - C' & D' - C' \\ B' - A' & B' - A' \end{pmatrix}, \begin{pmatrix} C' - A' & D' - B' \\ C' - A' & D' - B' \end{pmatrix} \right\}.$$

If the penalty term is zero then it must satisfy the first discrete form of the Cauchy–Riemann equations $\Delta_x R + \mathrm{i}\Delta_y R = 0$, i.e., by equating real and imaginary parts, we get:

$$0 = \Delta_x R = \begin{pmatrix} D' - C' & D' - C' \\ B' - A' & B' - A' \end{pmatrix}, 0 = \Delta_y R = \begin{pmatrix} C' - A' & D' - B' \\ C' - A' & D' - B' \end{pmatrix}.$$

These equations state that the two sides at each vertex are of equal length and form a right angle. That is, they describe a square.

3.4.1 First form with smooth images

We now test Algorithm 2, which we described in Section 3.2, using the first penalty term, Equation (3.11), by applying it to a sequence of progressively more difficult examples:

- Smooth images with synthetic data, i.e. the target is generated from the source by a known conformal map;
- Smooth images with real data, i.e. the source and target are independent and are not known *a priori* to be conformally related;
- non-smooth images with synthetic data; and
- non-smooth images with real data.

We present two examples of smooth images in this section. In the first example, the target is generated from the source. In the second example the source and the target are two independent images.

Example 3.1. We consider a Gaussian as an example of a smooth image that serves as the source. We now transform this Gaussian by applying a conformal transformation, $\varphi^{-1} = \Sigma_{n=0}^{2} a_n z^n$ with $a_0 = 0.1, a_1 = 1, a_2 = 0.3, z \neq 0$, to generate the target. The source and the target images can be seen in Figure 3.9.

The image registration results at the final step of optimisation, where $n^2 = 64$ and $\lambda = 20$, can be seen in Figure 3.10.

The second row of Figure 3.10 confirms that the transformed source is aligned with the target because the difference image is uniformly mid-grey. The deformation grid is also reasonably conformal, and the corresponding contour plots between the transformed source and the target confirms that the registration is very good. This is confirmed by the values of the registration error (in the Frobinius norm, denoted by $||.||$, of the difference image) and the values of the penalty term, Equation (3.11), which can be seen in the table below Figure 3.10.

This experiment was started with a very large value of λ that makes the penalty term dominant in Equation (3.3). Hence the optimiser enforces $||P_1|| = 0$, which results in a

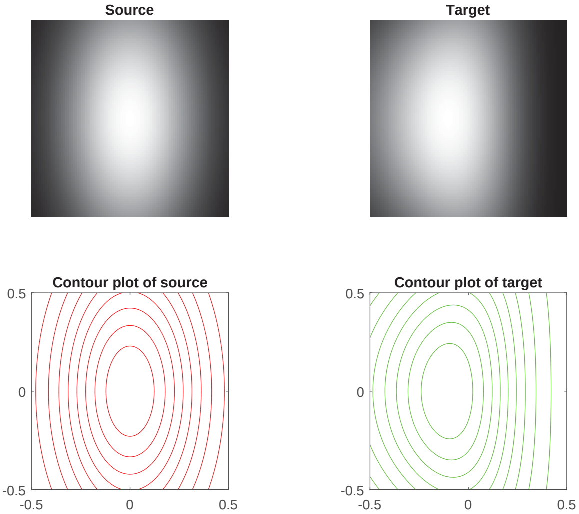
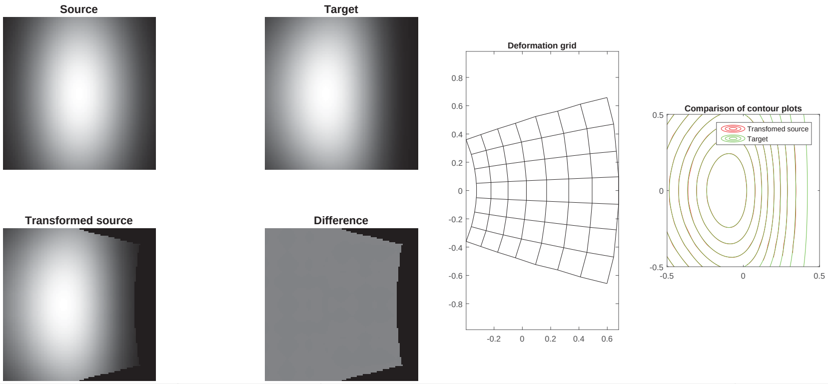


Figure 3.9: The source and target images for Example 3.1 are shown as greyscale images in the first row and as contour plots in the second row. The source is $I_1(x, y) = \exp(-7x^2 - 2y^2)$, and the target is defined by $I_2 = I_1 \circ \varphi^{-1}$, where $\varphi^{-1}(z) = 0.1 + z + 0.3z^2$.



	λ	$\|\|(I_1 \circ \varphi^{-1})(\hat{x}_{ij}) - I_2(\hat{x}_{ij})\|\|$	$\|\|P_1\|\|$
$n^2 = 16 \left\{ \right.$	20^4	17.9328	0.0000
	20^3	17.9328	0.0000
	20^2	11.8447	0.0209
	20	0.5073	0.0269
$n^2 = 64 \ \{$	20	0.1690	0.0086

Figure 3.10: The results of conformal image registration using Algorithm 2 applied to Example 3.1. The source and target images are shown, together with the final transformed source, which nearly matches the target on the part of the domain on which it is defined. The difference $I_2 - I_1 \circ \varphi^{-1}$ is shown in greyscale, where uniform mid-grey indicates a perfect match. The grid of the transformed control points, i.e. $\varphi^{-1}(\hat{x}_{ij})$ (we call it the deformation grid) and the contour plots of target and transformed source are shown on the right. In the table, each row gives the registration error and the value of the penalty term for a step of Algorithm 2.

rigid diffeomorphism, as was discussed earlier. These rigid diffeomorphisms can be seen in the grids of Figure 3.11. Due to this, the mismatch error between the transformed source and the target is significantly higher than the errors that are obtained when using comparatively smaller values of λ, as are presented in the table of Figure 3.10. This numerical evidence also demonstrates that the transformed source is getting closer to the target as λ decreases. In this example, we have found that $\lambda = 20$ with 64 control points is good enough (based on visual inspection of the two images and the fact that the deformation grid indicates that the map is invertible and conformal) to register this pair of images. We continue our investigation with these values of parameters in further examples. In next example, two independent images are chosen.

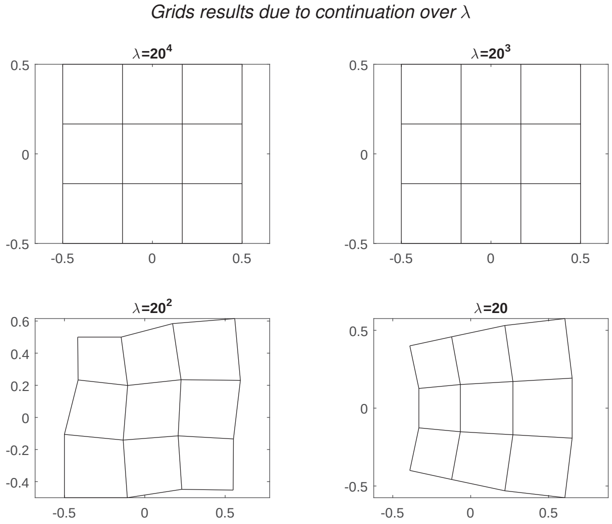


Figure 3.11: Four grids are shown that are corresponding to $\lambda = 20^4, 20^3, 20^2, 20$ respectively for sixteen control points for Example 3.1. The two grids in the first row are rigid because squares are mapped to squares under φ^{-1}; the third and fourth grids, with smaller values of λ, are nonrigid. The deformation grid corresponding to $\lambda = 20^2$ shows asymmetry, which we are unable to explain.

Example 3.2. In the previous example, the target was generated from the source. We now consider two independent smooth images, both Gaussians, that are not known *a priori* to be conformally related. These images of Gaussians and their corresponding contour plots are shown in Figure 3.12.

Note that from the Riemann mapping theorem; [39] (p. no. 55), [4] (p. no. 221-224), [104] (p. no. 3), there is a conformal map that takes any *one* contour of the source to the same contour of the target. This map is unique up to an element of the (3-dimensional) disk-preserving Möbius group. Thus, in general there is not enough freedom left to match any of the other contours. The conformal map that maps a circle to an ellipse maps concentric circles to confocal ellipses. As the ellipses in the present example are not confocal, we expect that that the source and target are not *exactly* conformally related. However, the object of conformal image registration is to find the

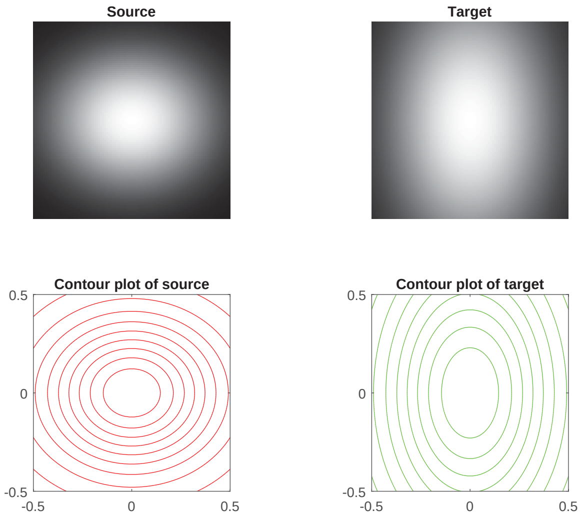
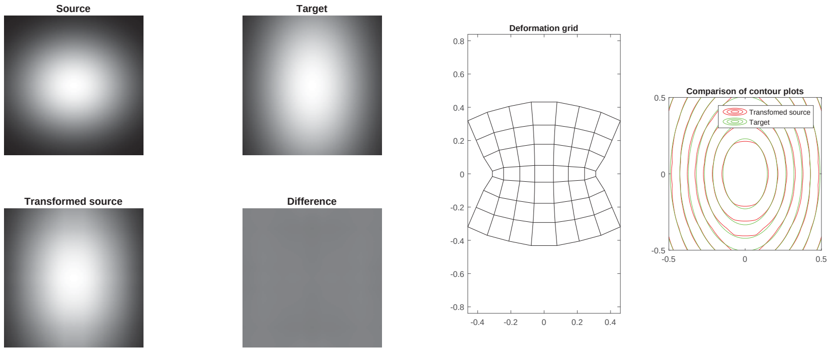


Figure 3.12: The source and target images for Example 3.2. The source is $\exp(-5x^2 - 7y^2)$, the target is $\exp(-5x^2 - 2y^2)$. The corresponding contour plots can be seen in the second row.

best possible conformal map, and to examine how good a job it does of registering the images.

We now apply Algorithm 2 to this example. The results are shown in Figure 3.13 and indicate that a successful conformal registration has been achieved.



	λ	$\|\|(I_1 \circ \varphi^{-1})(\hat{x}_{ij}) - I_2(\hat{x}_{ij})\|\|$	$\|\|P_1\|\|$
$n^2 = 16$ $\{$	20^4	10.5332	0.0000
	20^3	4.9578	0.0000
	20^2	4.7113	0.0027
	20	1.4932	0.0796
$n^2 = 64$ $\{$	20	0.9199	0.0562

Figure 3.13: The results of conformal registration with Algorithm 2 applied to Example 3.2, showing a near-perfect registration because some of the red contours are slightly mismatched with the corresponding green contours.

3.4.2 First form with non-smooth images

We now present three experiments in conformal image registration with non-smooth images. As before, we proceed from easy to difficult examples:

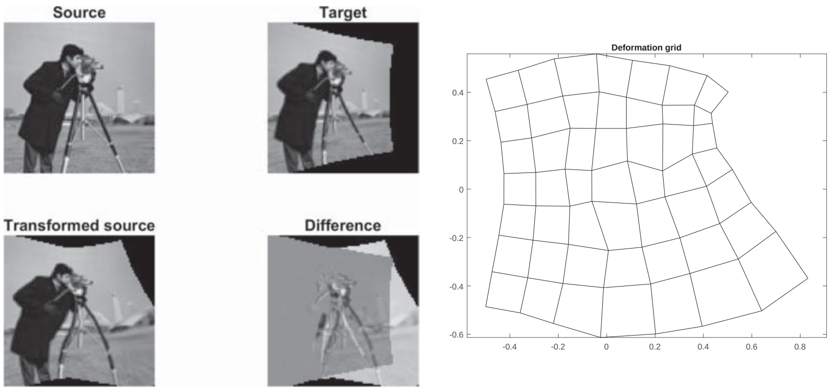
1. A non-smooth source, with the target being generated from the source by a known conformal map;
2. A non-smooth source and independent, but nearly conformally related, target; and
3. A non-smooth source and target that are not known to be conformally related.

Example 3.3. In Figure 3.14, two examples of the standard image of a cameraman are presented. In this example, the target is generated from the source using the conformal transformation $\varphi^{-1}(z) = 0.1 + 1.2z + 0.3z^2$ and we construct the target as $I_2(z) := I_1(\varphi^{-1}(z))$. Here bilinear interpolation is used to obtain the needed values of I_1.

Figure 3.14: The images for Example 3.3. The source I_1 is a standard reference image; the target is $I_2(z) = I_1(0.1 + 1.2z + 0.3z^2)$ and is therefore known to be conformally related to the source.

Our first results are shown in Figure 3.15. Clearly, they are extremely poor.

One feature of Matlab's optimisation algorithm *lsqnonlin* is that it may return a local minimum instead of a global minimum; this minimum may be very far from the global minimum. Indeed, this problem pervades the entire field of optimisation (see [56], [13], [78], [111]) and many attempts have been made to address it. Here we try a very simple strategy of adding a small random perturbation to the initial guess for φ which, until now, has been the identity. As we want to avoid giving the optimiser too much of a clue as to where the global minimum lies, we add a uniform pseudorandom number in the interval $[0, 0.05]$ to the x- and y-components of each $\hat{y}_{ij}$. This perturbed grid can be seen in Figure 3.16.



	λ	$\|\|(I_1 \circ \varphi^{-1})(\hat{x}_{ij}) - I_2(\hat{x}_{ij})\|\|$	$\|\|P_1\|\|$
$n^2 = 16$	20^4	22.3044	0.0000
	20^3	22.3044	0.0000
	20^2	19.1620	0.0079
	20	15.3616	0.2175
$n^2 = 64$	20	12.2815	0.1341

Figure 3.15: The results of the first attempt at registration for Example 3.3. An extremely poor registration is obtained.

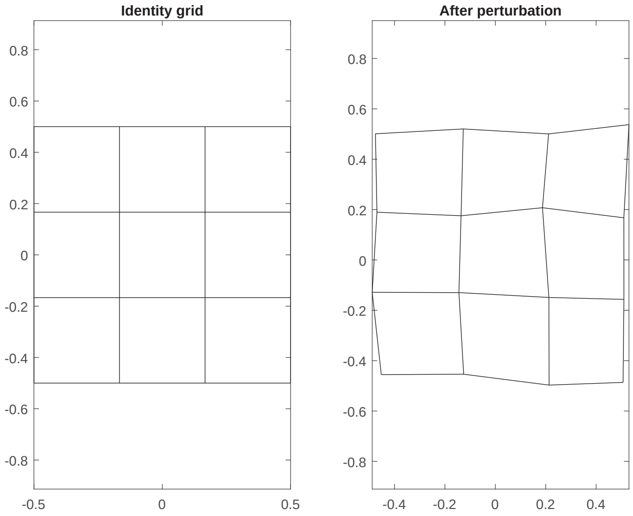
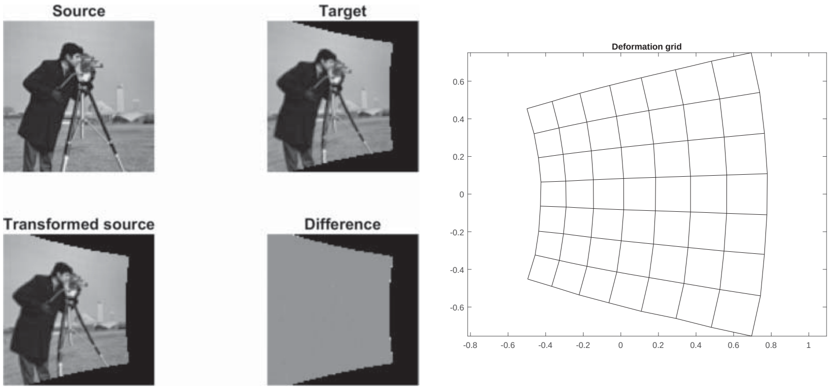


Figure 3.16: Left: the original set of control points; right: the same points with a small perturbation added to each point independently in x and y.

We now set these perturbed points as an initial guess and then use Algorithm 2 for the pair of images that are shown in Figure 3.14; the results can be seen in Figure 3.17. Clearly they are much improved! The algorithm has found an excellent conformal registration, with a smooth grid. The only remaining errors are due to the presentation of the quadratic $\varphi^{-1}(z)$ on the grid, and the form of the discrete Cauchy–Riemann equations used. The root-mean-square error is 0.005225 per pixel (relative to the image values, which lie in [0,1]); to achieve this all the main features of the images have been



	λ	$\vert\vert(I_1 \circ \varphi^{-1})(\hat{x}_{ij}) - I_2(\hat{x}_{ij})\vert\vert$	$\vert\vert P_1\vert\vert$
$n^2 = 16$ {	20^4	11.4395	0.0000
	20^3	11.3751	0.0001
	20^2	6.4348	0.0149
	20	2.5425	0.0589
$n^2 = 64$ {	20	0.5225	0.0161

Figure 3.17: The results of the second attempt at registration for Example 3.3, using a random initial guess close to the identity. A good registration is obtained.

aligned.

Example 3.4. We now proceed as for smooth images; that is, having successfully registered synthetic data we now attempt to register data in which the source and target are generated independently, but are known to be (close to) conformally related. These images are shown in Figure 3.18.

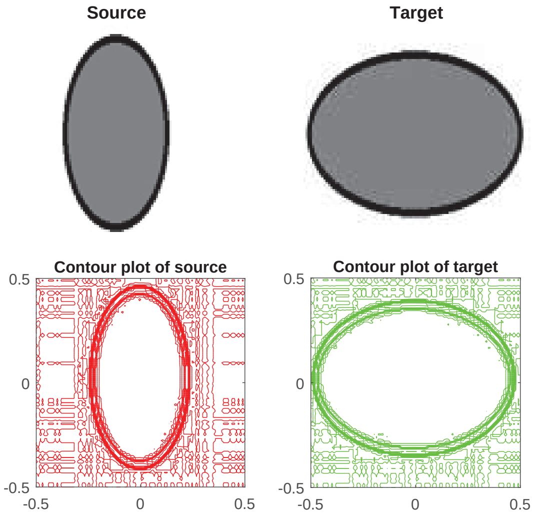
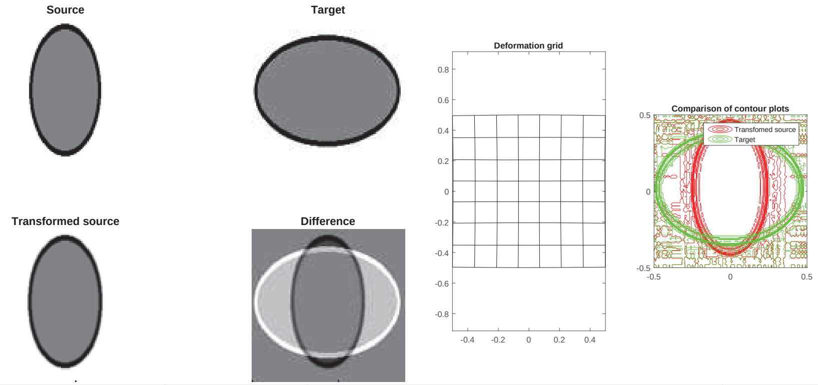


Figure 3.18: Two non-smooth independent images of ellipses are used as the source and target respectively.

The source and target are filled drawings of ellipses of different aspect ratios. Some small textural differences between the two images can be seen in the background of the images. This is an interpolation artefact caused by the interpolation of the images onto the 100×100 pixel grid. From the Riemann mapping theorem, there is a conformal map from the interior of an ellipse to the interior of any other ellipse. However, this map has a singularity outside the ellipse, so the example could be challenging. The black borders encourage the algorithm to match the boundaries closely. However, we expect that there exists a conformal map which nearly registers the images.

The results of the first attempt, with the identity as the initial guess, are shown in Figure 3.19. The optimiser remains stuck near to the identity. The results of the second attempt, with a small random perturbation added to the initial guess, are shown in Figure 3.20.

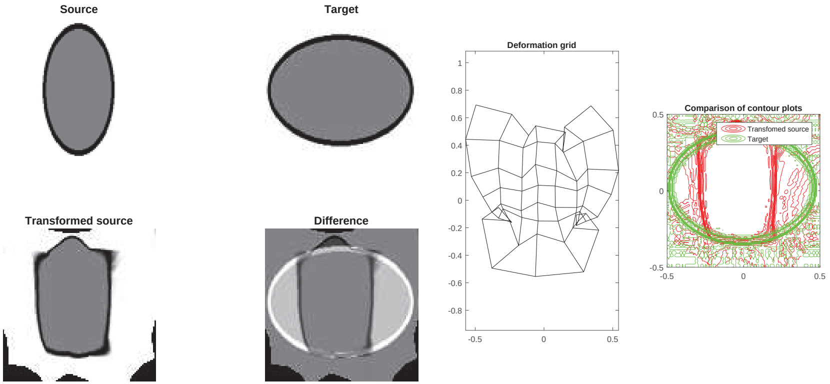


	λ	$\|\|(I_1 \circ \varphi^{-1})(\hat{x}_{ij}) - I_2(\hat{x}_{ij})\|\|$	$\|\|P_1\|\|$
$n^2 = 16$	20^4	35.7936	0.0000
	20^3	35.7936	0.0000
	20^2	35.1007	0.0047
	20	34.8754	0.0312
$n^2 = 64$	20	34.9018	0.0339

Figure 3.19: The results of the first attempt at registration for Example 3.4. The optimiser remains stuck at the initial condition, the identity.

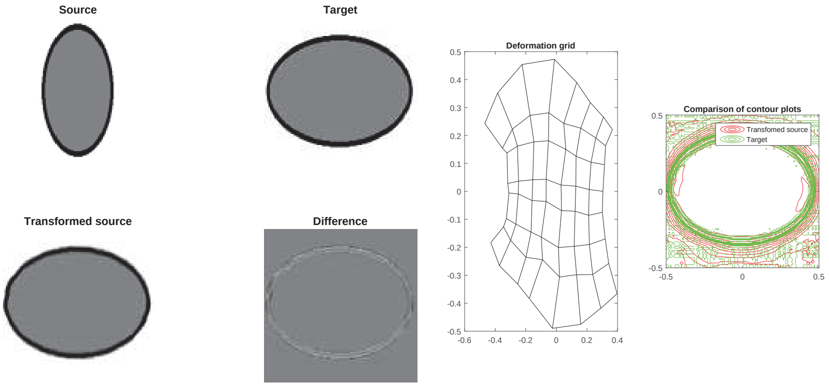
Although the optimiser is no longer stuck at the initial guess, it is apparent that it has not moved in the desired direction either. In addition, the transformation is not invertible. As the new feature of the images in this section is their non-smoothness, we transform the source and target to become smooth images by smoothing them (as explained in Section 3.1.1).

Algorithm 2 is applied first to the smoothed images, and then a final step is taken with the unsmoothed images. In this way the preliminary steps with the smoothed images can be seen as a way of finding a good initial guess for the registration. The results are shown in Figure 3.21. They are much better than the first two attempts and a satisfactory registration is obtained; the two elliptical shapes overlap precisely. The root-mean-square error in the registration is 0.05 per pixel. The discrete mapping is invertible, although it is still not as smooth as we would like.



	λ	$\|\|(I_1 \circ \varphi^{-1})(\hat{x}_{ij}) - I_2(\hat{x}_{ij})\|\|$	$\|\|P_1\|\|$
$n^2 = 16$	20^4	36.4727	0.0000
	20^3	36.4726	0.0000
	20^2	36.4161	0.0040
	20	35.7643	0.1048
$n^2 = 64$	20	31.4145	0.2473

Figure 3.20: The results of the second attempt at registration for Example 3.4. The registration is still not successful even after the control points are perturbed.



	λ	$\|\|(I_1 \circ \varphi^{-1})(\hat{x}_{ij}) - I_2(\hat{x}_{ij})\|\|$	$\|\|P_1\|\|$
$n^2 = 16$	20^4	19.5767	0.0000
	20^3	19.5709	0.0000
	20^2	17.0609	0.0168
	20	3.3041	0.1581
$n^2 = 64$	20	2.9840	0.1192
	20	4.6213	0.1177

Figure 3.21: Left: the image matching results for Example 3.4 initialised with smoothed versions of the images. The top row shows the source and the target, while the transformed source and the difference between the transformed source and the target can be seen in second row. Right: the deformation grid $\varphi^{-1}(\hat{x}_{ij}), \hat{x}_{ij} \in S$ with $\lambda = 20$. In the table below, each row gives the registration error and value of penalty term P_1 for each step of Algorithm 2. In the first 5 rows the source and target images are smoothed by a Gaussian filter with standard deviation 5 pixels; in the last row they are not smoothed.

We next attempt to register two non-smooth images that are hypothetically conformally related.

Example 3.5. Recall from Chapter 1 that D'Arcy Wentworth Thompson claimed that the forms of different species can be smoothly transformed into one another; that is, they can be registered. Examples include the comparison of human, chimpanzee, and baboon skulls and species of fish. Furthermore, he used the simplest possible transformations, suggesting that they should be 'perfectly regular' or 'isogonal', i.e., conformal.

We will investigate the validity of Thompson's claim regarding one specific pair of his fish images (see Figure 3.22), that they are related by an isogonal (conformal) transformation.

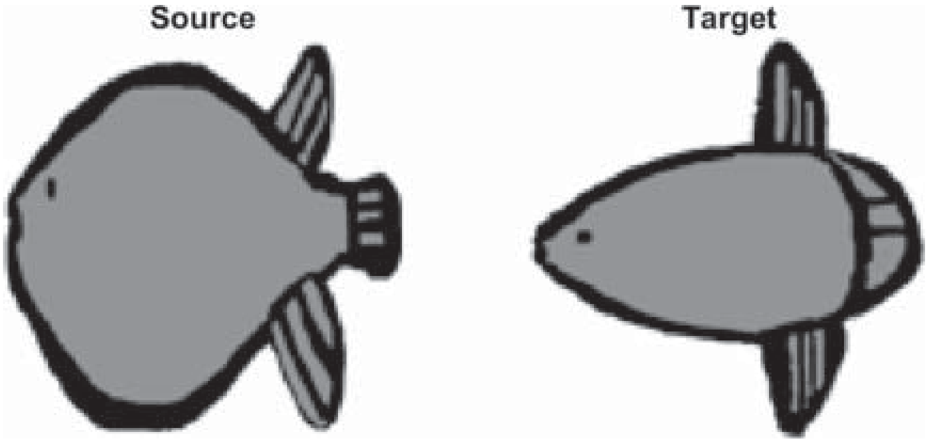
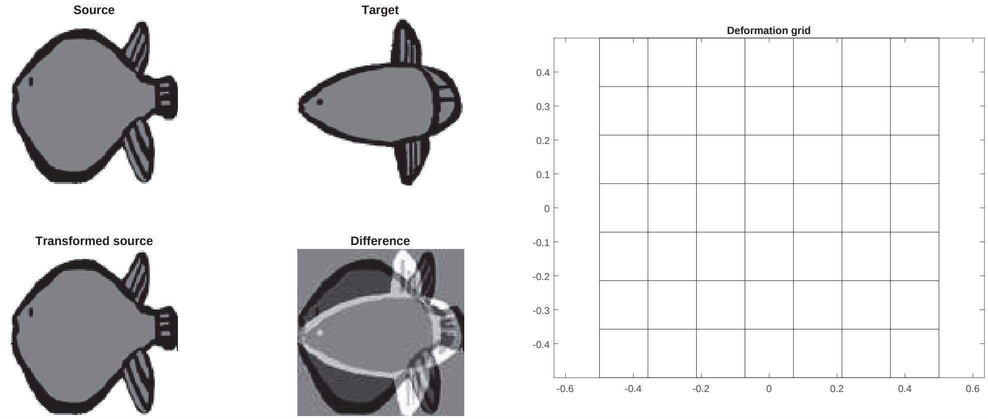


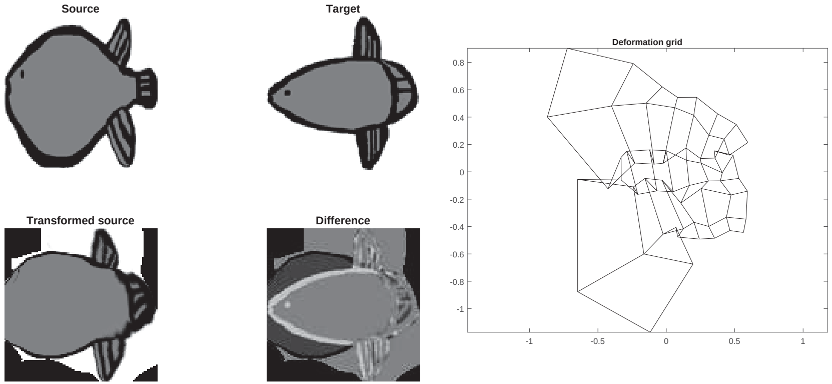
Figure 3.22: Modified versions of the images of Thompson's fish are shown. The first image serves as the source and the second image as the target for image registration.

The first set of results are shown in Figure 3.23 for initial guess the identity; in Figure 3.24 for initial guess a small perturbation of the identity; and in Figure 3.25 together with initial runs on smoothed versions of the images.



	λ	$\|\|(I_1 \circ \varphi^{-1})(\hat{x}_{ij}) - I_2(\hat{x}_{ij})\|\|$	$\|\|P_1\|\|$
$n^2 = 16$ {	20^4	46.0611	10^{-5}
	20^3	46.0611	10^{-5}
	20^2	46.0611	10^{-5}
	20	46.0185	0.2×10^{-5}
$n^2 = 64$ {	20	45.9637	$0.1 \times 10^{-}5$

Figure 3.23: The results of the first attempt at conformal image registration for Thompson's fish, with initial guess the identity. The optimiser remains stuck at the identity.



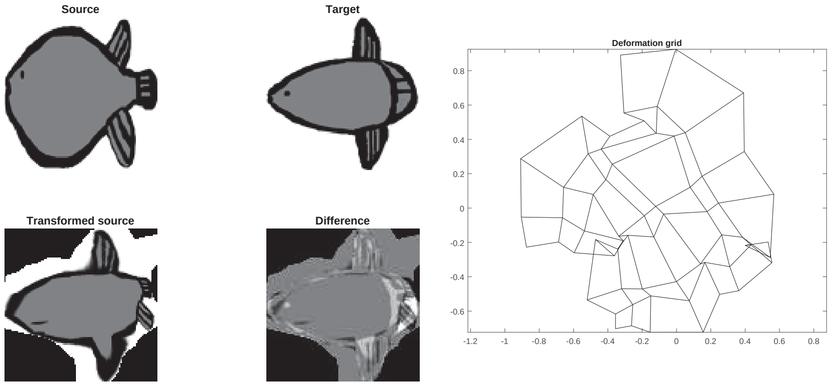
	λ	$\|\|(I_1 \circ \varphi^{-1})(\hat{x}_{ij}) - I_2(\hat{x}_{ij})\|\|$	$\|\|P_1\|\|$
$n^2 = 16$ {	20^4	45.6515	0.0000
	20^3	45.1794	0.0001
	20^2	45.1621	0.0030
	20	34.0983	0.4542
$n^2 = 64$ {	20	28.9476	0.4437

Figure 3.24: Results[†]of the second attempt at conformal registration of Thompson's fish, with initial condition a small perturbation of the identity. The optimiser has partially registered the images, but the grid is poor.

As in the previous example, the first attempt remains stuck at the identity (mean error 0.46 per pixel); the second does a bit better, aligning the top fin and the back

---

[†] Recall that the black region in the transformed source indicates the set of points that have no colour information (missing values). These missing values are dealt with using the scaled optimisation function that was presented in Equation (2.2).



	λ	$\|\|(I_1 \circ \varphi^{-1})(\hat{x}_{ij}) - I_2(\hat{x}_{ij})\|\|$	$\|\|P_1\|\|$
$n^2 = 16$	20^4	21.7128	0.0000
	20^3	21.7127	0.0000
	20^2	21.6140	0.0033
	20	12.2772	0.4093
$n^2 = 64$	20	9.2000	0.3455
	20	19.0665	0.4806

Figure 3.25: Results of the third attempt at registration of Thompson's fish, using a small perturbation of the identity as an initial guess and (in the first 5 rows of the table) smoothing applied to the images; the results are better than the first two attempts, but still not satisfactory.

but not the bottom fin nor the front of the fish, and producing a noninvertible transformation (mean error 0.29 per pixel); and the third manages to align the overall body shapes of the fish, including the fins. However, despite this apparent success, the mean error is still 0.19 per pixel, the transformation has some (small) regions in which it is not invertible (see triangular regions at the right bottom and the central far left of the transformed grid in Figure 3.25), and the grid is far from regular (and it is not conformal, i.e., it does not satisfy the Cauchy-Riemann equations). However, note that for piecewise-constant images like this, the grid is not constrained at all in the regions far from the edge of the fish, either inside or outside; small changes in the grid in these regions do not affect the objective function. This makes it difficult for the optimiser to control the grid in these regions and indicates that some further regularisation may be needed. However, it also means that a poor grid in these regions does not necessarily indicate a poor registration.

Overall, we are as yet unable to ratify Thompson's claim regarding the pair of fish in Figure 3.22. In other words, it is not clear if it is the fish or the algorithm that is at fault. We therefore turn to exploring alternative forms of the penalty function that control the conformality of the grid. In the next section, two other discrete forms of the Cauchy–Riemann equations are discussed.

3.5 Second and third discrete forms of the Cauchy–Riemann equations

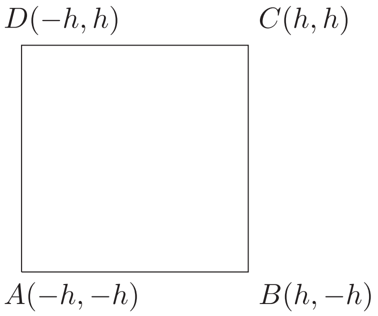
We have seen that the first discrete form of the Cauchy–Riemann equations is able to produce the desired conformal map in some of the cases that we presented, but it is unable to produce the desired conformal map in Examples 3.4 and 3.5. Although the transformed source is aligned with the target in Example 3.4, the corresponding grid is quite irregular. In Example 3.5 both the registration and the map are poor. Thus, we consider two alternative discrete forms of the Cauchy–Riemann equations. These methods were described by Bobenko et al. in [14]. They termed the two discretisations as the linear and nonlinear forms. The linear form is based on the cross-ratio, whereas the nonlinear form is based on the notion of a circle pattern.

1. **Linear form**

   The *linear discrete Cauchy-Riemann equations* are [14]:

   $$\varphi_{m,n+1} - \varphi_{m+1,n} = \mathrm{i}(\varphi_{m+1,n+1} - \varphi_{m,n}). \tag{3.12}$$

   We now show that these equations are a second order discretisation of the Cauchy–Riemann equations. Consider a square with four vertices that we label as A, B, C, D.



   Equation (3.12) for this square is:

   $$(\varphi(-h,h) - \varphi(h,-h)) = \mathrm{i}(\varphi(h,h) - \varphi(-h,-h)). \tag{3.13}$$

Expanding in Taylor series at $(0,0)$ gives

$$\left.\begin{aligned}\varphi(-h,h) &= \varphi - h\varphi_x + h\varphi_y + \tfrac{h^2}{2}\varphi_{xx} + \tfrac{h^2}{2}\varphi_{yy} - h^2\varphi_{xy} + O(h^3)\\ \varphi(h,-h) &= \varphi + h\varphi_x - h\varphi_y + \tfrac{h^2}{2}\varphi_{xx} + \tfrac{h^2}{2}\varphi_{yy} - h^2\varphi_{xy} + O(h^3)\\ \varphi(-h,-h) &= \varphi - h\varphi_x - h\varphi_y + \tfrac{h^2}{2}\varphi_{xx} + \tfrac{h^2}{2}\varphi_{yy} + h^2\varphi_{xy} + O(h^3)\\ \varphi(h,h) &= \varphi + h\varphi_x + h\varphi_y + \tfrac{h^2}{2}\varphi_{xx} + \tfrac{h^2}{2}\varphi_{yy} + h^2\varphi_{xy} + O(h^3)\end{aligned}\right\} \tag{3.14}$$

By plugging these equations into Equation (3.13), we get:

$$(-2h\varphi_x + 2h\varphi_y) - \mathrm{i}(2h\varphi_x + 2h\varphi_y) = O(h^3)$$

or

$$(\varphi_x + \mathrm{i}\varphi_y)(-1-\mathrm{i}) = O(h^2).$$

That is, if $\varphi(z)$ satisfies the Cauchy–Riemann equations then it satisfies the linear discrete Cauchy–Riemann equations with a discretisation error of $O(h^2)$.

We therefore define our second penalty term, based on the linear discrete Cauchy–Riemann equations, as:

$$P_2(\hat{y}_{ij}) = (|\hat{y}_{i,j+1} - \hat{y}_{i+1,j} - \mathrm{i}(\hat{y}_{i+1,j+1} - \hat{y}_{i,j})|, \quad i,j = 1,\ldots,n-1 \tag{3.15}$$

Note that the first form has one complex equation per grid point, while the second form has one complex equation per cell, that is, $(n-1)^2$ complex equations in total.

2. **Nonlinear form:** The *nonlinear discrete Cauchy-Riemann equations* are [14]:

$$\frac{(\varphi_{m+1,n} - \varphi_{m,n})(\varphi_{m+1,n+1} - \varphi_{m,n+1})}{(\varphi_{m,n+1} - \varphi_{m,n})(\varphi_{m+1,n+1} - \varphi_{m+1,n})} = -1. \tag{3.16}$$

We now show that this form is also a second order discretisation of the Cauchy–Riemann equations. Equation (3.16) for the square $ABCD$ is:

$$\frac{(\varphi(h,-h) - \varphi(-h,-h))(\varphi(h,h) - \varphi(-h,h))}{(\varphi(-h,h) - \varphi(-h,-h))(\varphi(h,h) - \varphi(h,-h))} = -1.$$

Rearranging,

$$\begin{aligned}&(\varphi(h,-h) - \varphi(-h,-h))(\varphi(h,h) - \varphi(-h,h)) +\\ &\qquad(\varphi(-h,h) - \varphi(-h,-h))(\varphi(h,h) - \varphi(h,-h)) = 0.\end{aligned} \tag{3.17}$$

Expanding in Taylor series about (0,0) gives:

$$\begin{aligned}&(2h\varphi_x - 2h^2\varphi_{xy} + O(h^3))(2h\varphi_x + 2h^2\varphi_{xy} + O(h^3))+\\&\quad(2h\varphi_y - 2h^2\varphi_{xy} + O(h^3))(2h\varphi_y + 2h^2\varphi_{xy} + O(h^3)) = 0\\\Rightarrow&(\varphi_x - h\varphi_{xy} + O(h^2))(\varphi_x + h\varphi_{xy} + O(h^2))+\\&\quad(\varphi_y - h\varphi_{xy} + O(h^2))(\varphi_y + h\varphi_{xy} + O(h^2)) = 0\\\Rightarrow&(\varphi_x)^2 - h^2\varphi_{xy} + (\varphi_y)^2 - h^2\varphi_{xy} = O(h^2)\\\Rightarrow&(\varphi_x)^2 + (\varphi_y)^2 - 2h^2\varphi_{xy} = O(h^2)\\\Rightarrow&(\varphi_x + \mathrm{i}\varphi_y)(\varphi_x - \mathrm{i}\varphi_y) = O(h^2)\end{aligned} \tag{3.18}$$

Therefore, if $\varphi(z)$ is either conformal or anticonformal (a conformal map of $\bar{z}$, i.e., a composition of a reflection and a conformal map) then it will satisfy the nonlinear discrete Cauchy–Riemann equations with a discretization error of $O(h^2)$. Thus, for the third penalty term we take

$$\begin{aligned}P_3(\hat{y}_{ij}) = |&(\hat{y}_{i+1,j} - \hat{y}_{i,j})(\hat{y}_{i+1,j+1} - \hat{y}_{i,j+1})\\&+ (\hat{y}_{i,j+1} - \hat{y}_{i,j})(\hat{y}_{i+1,j+1} - \hat{y}_{i+1,j})|, \quad i,j = 1,\dots,n-1\end{aligned} \tag{3.19}$$

It is possible that, when using the third form, the penalty term could be zero and the grid would approximate an anticonformal map instead of a conformal map. However, this has not happened in any of our experiments, but it is a difference between Bobenko's linear and nonlinear forms.

Both of the forms P_2 and P_3 are less restrictive than P_1. This is because they contain fewer equations (one per cell instead of one per grid point) and they can be satisfied by many non-rigid mappings. A discrete mapping is determined by $2n^2$ real parameters, while the equations $P_2(\hat{y}_{ij}) = 0$ for $i,j = 1,\dots,n-1$ (resp. P_3) requires $2(n-1)^2$ equations. Thus we expect there to be $2n^2 - 2(n-1)^2 = 4n-2$ free parameters in their solution. We study this freedom first for the case $n = 1$, i.e. a single square. In this section, first we compute the equations for the fourth vertex for both the linear and nonlinear forms and then we will provide an example to represent a non-square quadrilateral by computing the fourth vertex for both the forms when three vertices are known. Now for a square $ABCD$ $(A,B,C,D \in \mathbb{C})$ we have for the linear form given in Equation (3.12):

$$D - B = \mathrm{i}(C - A) \implies D = B + \mathrm{i}(C - A) \tag{3.20}$$

while for the nonlinear form (Equation (3.16)):

$$\begin{aligned} \frac{(B-A)(C-D)}{(D-A)(C-B)} &= -1 \\ \implies D &= \frac{A(C-B)+C(A-B)}{(C-2B+A)} \\ &= \frac{2AC-AB-BC}{(C-2B+A)}. \end{aligned} \tag{3.21}$$

Example 3.6. Let $A = 0.4 + 0.5i$, $B = -0.1 + 0.3i$ and $C = -0.3 - 0.4i$ be 3 points in the complex plane. We now use Eqs. 3.20 and 3.21 for the computation of the fourth vertex for both of the forms; the corresponding values of the fourth vertex are $D = 0.8 - 0.4i$ for the linear form and $D = 1.12 - 0.27\text{i}$ for the nonlinear form. We draw a quadrilateral, with the four vertices A, B, C, D. Three vertices A, B, C are marked with the red circles. The computed fourth vertex is marked with the blue circle. These vertices can be seen in Figure 3.26.

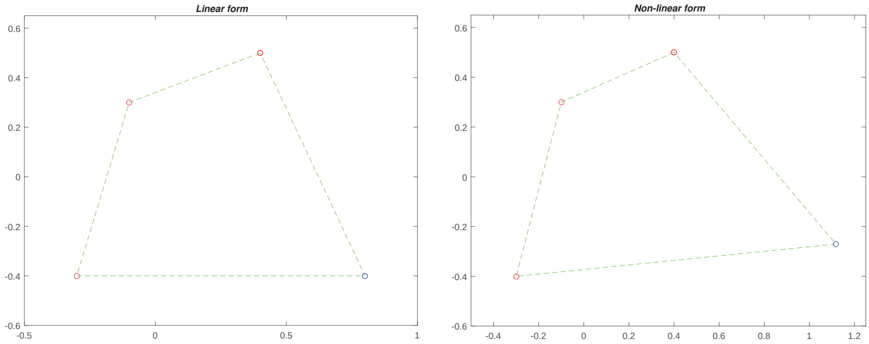


Figure 3.26: In both the figures, the three red circles represent three complex numbers. The blue circles represent the fourth complex number for the linear (left) and nonlinear (right) forms.

For $n > 1$, suppose that the values of $\hat{y}_{ij}$ are specified on the left and bottom edges of the grid. This is equivalent to adding $4n - 2$ real conditions to the grid, which we expect to use up all of its freedom. And indeed, the values of all $\hat{y}_{ij}$ can now be filled in sequentially, starting with the bottom left cell as above, then the whole bottom row, then the 2nd to bottom row, and so on. Thus, we can say that there are many solutions other than the rigid one that satisfy the linear form (Equation (3.12)) and nonlinear form (Equation (3.16)). This is not the case for P_1.

Note that the counting of these discrete forms is different from that of the continuous Cauchy-Riemann equations. For the continuous form, the real part of φ can be specified on the boundary and its values in the interior determined by solving Laplace's equation $\nabla^2(\Re(\varphi)) = 0$ which has a unique solution. The imaginary part of φ is then determined up to a constant. But in the discrete case, once the real part of $\hat{y}_{ij}$ is specified on the boundary (i.e. $4n-4$ conditions imposed), there remain 2 (not 1) degrees of freedom for the imaginary part.

Also note that there are discrete maps satisfying $P_2 = 0$ or $P_3 = 0$ which contain 'butterflies' in which the edges of the image quadrilateral intersect. The resulting continuous map obtained by bilinear interpolation would not then be invertible. We will see examples of this when we test these forms.

3.5.1 Second/Third form with smooth images

We now repeat the same sequence of examples with the second and third forms of the penalty term, corresponding to the linear and nonlinear discrete Cauchy–Riemann equations of Bobenko, respectively. By treating the same examples in parallel, changing only one feature (the penalty term), we can assess its impact.

Example 3.7. In this example, the images of Example 3.1—a smooth Gaussian blob and its image under a conformal map—are reconsidered for image registration. The results for the linear form (linear discrete Cauchy–Riemann equations) are shown in Figure 3.27, and for the nonlinear form in Figure 3.28.

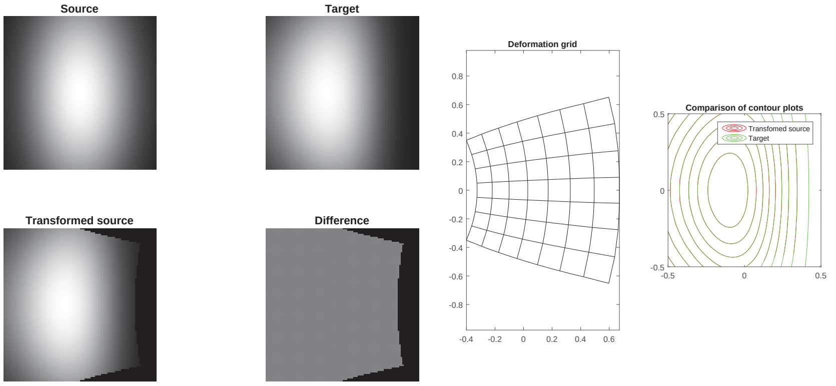


Figure 3.27: Results of conformal image registration for Example 3.7 using the linear discrete Cauchy–Riemann equations as a penalty term.

In both cases, the transformed source is perfectly aligned with the target, as in Example 3.1. Further, the corresponding deformation grids are identical to the deformation grid of Figure 3.10. The registration errors and the penalty terms, Eqs. 3.15 and 3.19, are provided in Table 3.1.

The registration errors at $\lambda = 20$ and $n^2 = 64$ are nearly identical in both the linear and nonlinear forms. These errors are less than the error we found previously with the first form in Example 3.1. It is interesting to compare the numerical results of Table 3.1 with Figure 3.10 of Example 3.1. Large values of λ do not significantly effect the registration error in the case of the linear form for this pair of images: the third column

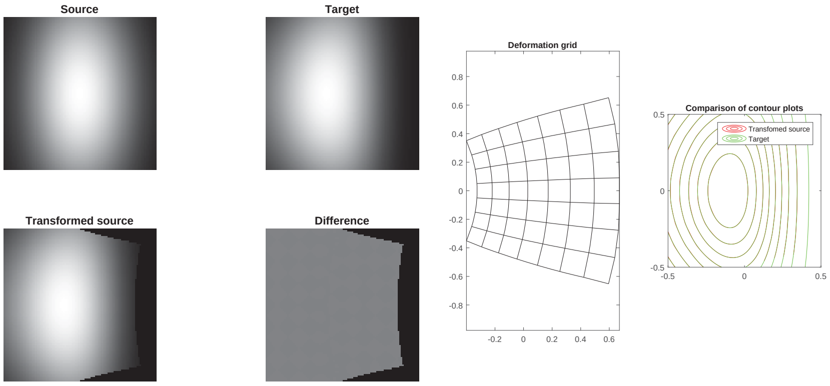


Figure 3.28: Results of conformal image registration for Example 3.7 using the nonlinear discrete Cauchy–Riemann equations as a penalty term.

		Linear form		Nonlinear form	
	λ	RE_{lin}	$\|\|P_2\|\|$	RE_{nonlin}	$\|\|P_3\|\|$
$n^2 = 16$	20^4	0.5383	0.0000	0.8930	0.0000
	20^3	0.5383	0.0000	0.5599	0.0000
	20^2	0.5383	0.0000	0.5596	0.0000
	20	0.5233	0.0043	0.5025	0.0075
$n^2 = 64$	20	0.1023	0.0002	0.1016	0.0005

Table 3.1: Registration errors and penalty terms for Example 3.7. Here the same registration error as in all other tables is used: $RE{=}||(I_1 \circ \varphi^{-1})(x_{ij}) - I_2(x_{ij})||$. However, we refer to it by name to save space. P_2 is the second (linear) penalty term given in Equation (3.15), and P_3 is the third (nonlinear) penalty term given in Equation (3.19).

in Table 3.1 indicates that even at $\lambda = 20^4$, the registration error (0.5383) is close to its minimum at smaller values of λ (0.5233).

Recall from Section 3.4 that large values of λ enforce $||P_1|| = 0$, and thus result in rigid diffeomorphisms; and from Section 3.5 that large values of λ along with $P_2 = 0$ or $P_3 = 0$ may or may not result in rigid diffeomorphisms. In this example, at $\lambda = 20^4$ both forms produce non-rigid diffeomorphisms. In Figure 3.29, we show all the grids corresponding to each value of λ for both the linear and nonlinear forms. These images of grids confirm our claim that the linear and nonlinear forms may have solutions other than rigid, as we explained in Section 3.5 with the help of Example 3.6.

These grids also illustrate that there is value in retaining the parameter λ, even though we could apparently send $\lambda \to \infty$ and enforce the constraint $P_2 = 0$ or $P_3 = 0$. In Figure 3.29, the grids on the right appear to be less regular at $\lambda = 20^4$ than at smaller λ. There may, in fact, be no need to enforce the discrete Cauchy–Riemann equations *exactly*, as due to discretization errors the grids correspond to conformal maps only in an approximate sense anyway; by lowering λ, which allows a better registration, we may be able to get a better registration and a smoother grid.

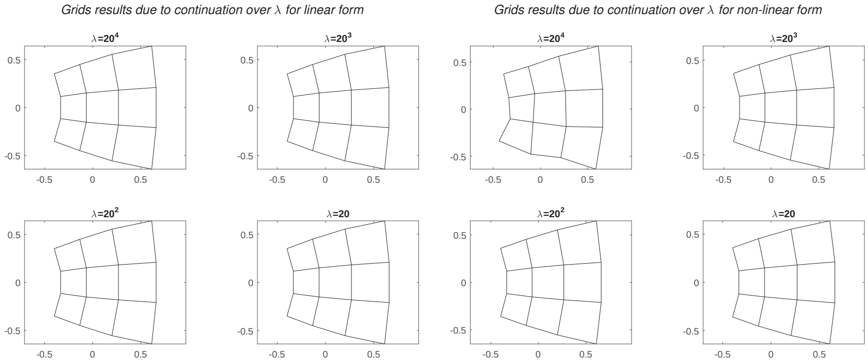


Figure 3.29: The grids in Example 3.7 corresponding to $\lambda = 20^4, 20^3, 20^2, 20$ for (left) the linear form of the penalty term and (right) for the nonlinear form.

The registration errors for the first, second (linear) and third (nonlinear) penalty terms are 0.1690, 0.1023, and 0.1016, respectively. Thus the second and third penalty terms have allowed the optimiser to find a better image registration.

Example 3.8. In this example we consider the pair of smooth images that were presented in Example 3.2. The image registration results can be seen in Figure 3.30 for the second form Equation (3.15) and in Figure 3.31 for the third form (Equation (3.19)). The numerical errors are given in Table 3.2.

		Linear form		**Nonlinear form**	
	λ	RE_{lin}	$\|\|P_2\|\|$	RE_{nonlin}	$\|\|P_3\|\|$
$n^2 = 16$	20^4	3.6002	0.0000	3.2304	0.0000
	20^3	2.8332	0.0000	3.2269	0.0000
	20^2	2.8308	0.0002	1.5946	0.0003
	20	2.2355	0.0550	0.7778	0.0360
$n^2 = 64$	20	1.0888	0.0340	0.1587	0.0177

Table 3.2: Registration errors and penalty terms for Example 3.8.

In this example, the linear and nonlinear form generate a grid close to that found with the first form. The registration errors for the first, second (linear) and third (nonlinear) penalty terms are 0.9199, 1.0888, and 0.1587, respectively. Thus the nonlinear form has found a markedly superior solution while still retaining a smooth grid.

3.5.2 Second/third form with non-smooth images

For non-smooth images, we adopt the strategies that proved effective with the first form of the penalty term, namely (i) choosing an initial guess that is a small random perturbation of the identity, and (ii) improving the initial guess by registering smoothed versions of the images first.

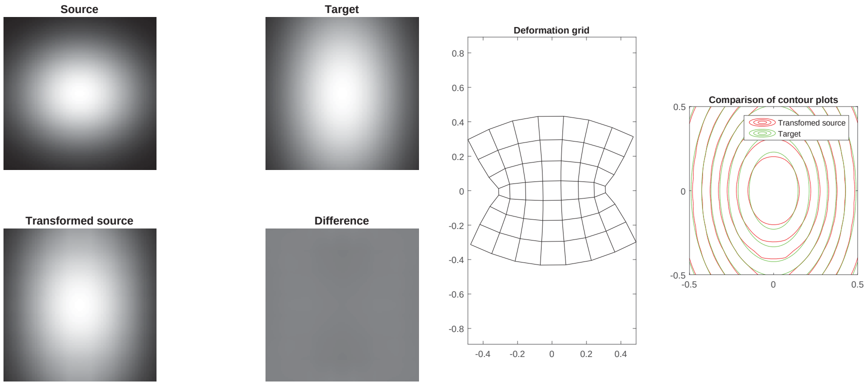


Figure 3.30: Results of conformal image registration for Example 3.8 using the linear discrete Cauchy–Riemann equations as a penalty term.

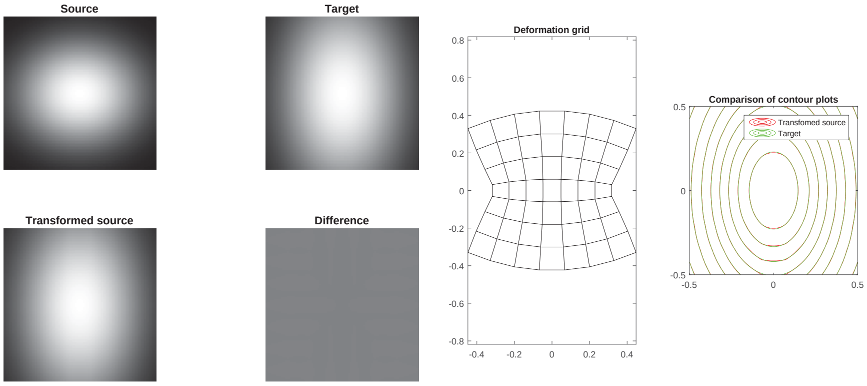


Figure 3.31: Results of conformal image registration for Example 3.8 using the nonlinear discrete Cauchy–Riemann equations as a penalty term.

Example 3.9. In this example, the cameraman, the source is non-smooth and the target is generated from the source by a known conformal map. The results are shown in Figure 3.32 for the linear form and in Figure 3.33 for the nonlinear form.

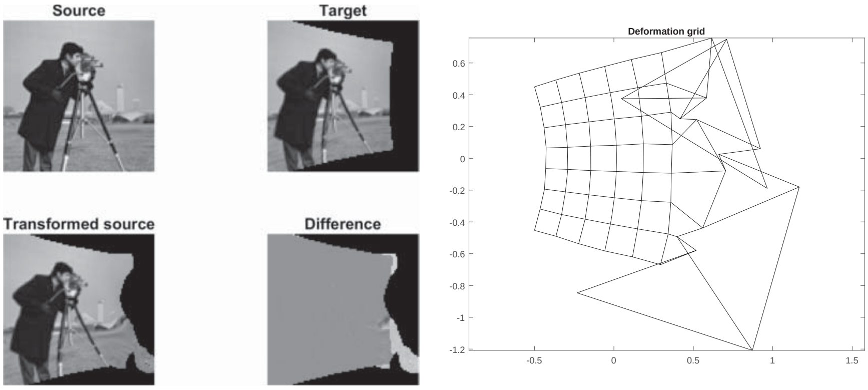


Figure 3.32: Results of conformal image registration for Example 3.9 using the linear discrete Cauchy–Riemann equations as a penalty term.

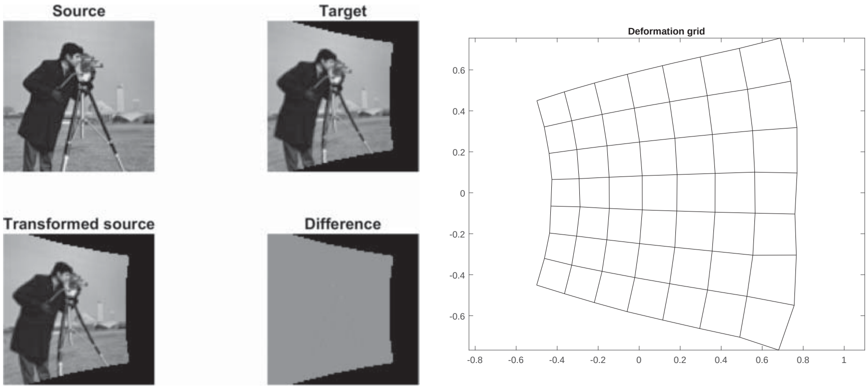


Figure 3.33: Results of conformal image registration for Example 3.9 using the nonlinear discrete Cauchy–Riemann equations as a penalty term.

It can be seen that the linear form has registered the images, but that the grid has become very irregular at the right hand edge, corresponding to those parts that map out of the domain under φ^{-1} and are thus relatively less constrained. The nonlinear form has not suffered from this phenomenon. Overall, the nonlinear form has given a better registration for $n^2 = 16$. This is confirmed by the registration errors given in Table 3.3. However, the linear form has produced a significantly larger error than the nonlinear form in the final registration (when $n^2 = 64$, bottom line of Table 3.3). It is also worse than the first form (see the table in Figure 3.17; the final registration errors for the first, second (linear), and third (nonlinear) penalty terms are 0.5225, 2.5611, and 0.4778, respectively (note that smoothing was not used at all for the first method).

		Linear form		**Nonlinear form**	
	λ	RE_{lin}	$\|\|P_2\|\|$	RE_{nonlin}	$\|\|P_3\|\|$
$n^2 = 16$	20^4	0.9876	0.0000	1.5826	0.0000
	20^3	0.9876	0.0000	1.0171	0.0000
	20^2	0.9876	0.0000	0.9897	0.0000
	20	0.9774	0.0044	0.9694	0.0053
$n^2 = 64$	20	0.9462	0.0086	0.7942	0.0153
	20	2.5611	0.0138	0.4778	0.0030

Table 3.3: Registration errors and penalty terms for Example 3.9. (The images are smoothed in the first 5 rows, and unsmoothed in the last row.)

Example 3.10. We now present the second example of non-smooth images. In this example we reconsider the images of Example 3.4. The results are shown in Figure 3.34 and in Table 3.4.

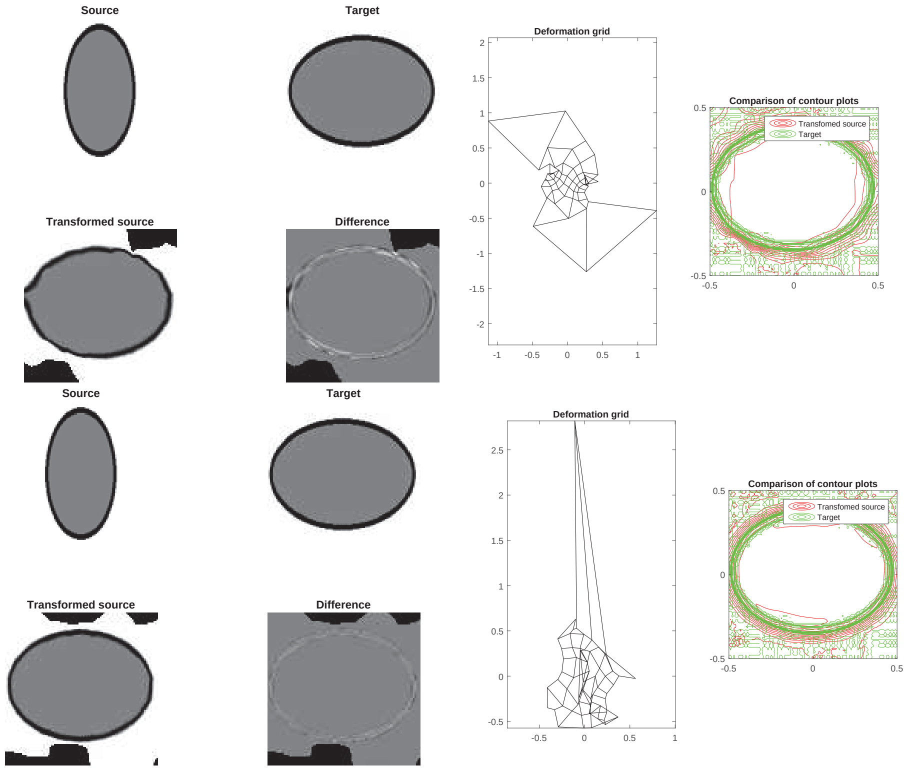


Figure 3.34: Results of conformal image registration for Example 3.10 using the linear discrete Cauchy–Riemann equations as a penalty term (top) and the nonlinear form (bottom).

		Linear form		**Nonlinear form**	
	λ	RE_{lin}	$\|\|P_2\|\|$	RE_{nonlin}	$\|\|P_3\|\|$
$n^2 = 16$	20^4	6.9561	0.0000	20.4114	0.0000
	20^3	6.6985	0.0000	7.6690	0.0004
	20^2	6.6922	0.0005	4.9011	0.0006
	20	5.2208	0.1310	2.7072	0.1034
$n^2 = 64$	20	4.0329	0.1106	1.0654	0.0413
	20	8.5167	0.2287	3.9411	0.0678

Table 3.4: Registration errors and penalty terms for Example 3.10. (The images are smoothed in the first 5 rows, and unsmoothed in the last row.)

The final registration errors for the first, second (linear), and third (nonlinear) penalty terms are 4.6213, 8.5167, and 3.9411, respectively. The error in the registration for the non-linear form changes more rapidly than the error for the linear form (see 3rd and 5th column in Table 3.4). Although the image registration is good, both the linear and nonlinear forms have produced poor grids, with noninvertible regions and unnecessarily extreme local distortions, such as nearly triangular quadrilaterals. We were unable to produce a regular discrete conformal grid for this example using the Bobenko discrete forms.

Example 3.11. We now move to the last example of this section, in which we reconsider the images of Thompson's pair of fishes, as in Example 3.5. The results for both the linear and nonlinear forms can be seen in Figure 3.35 and Table 3.5.

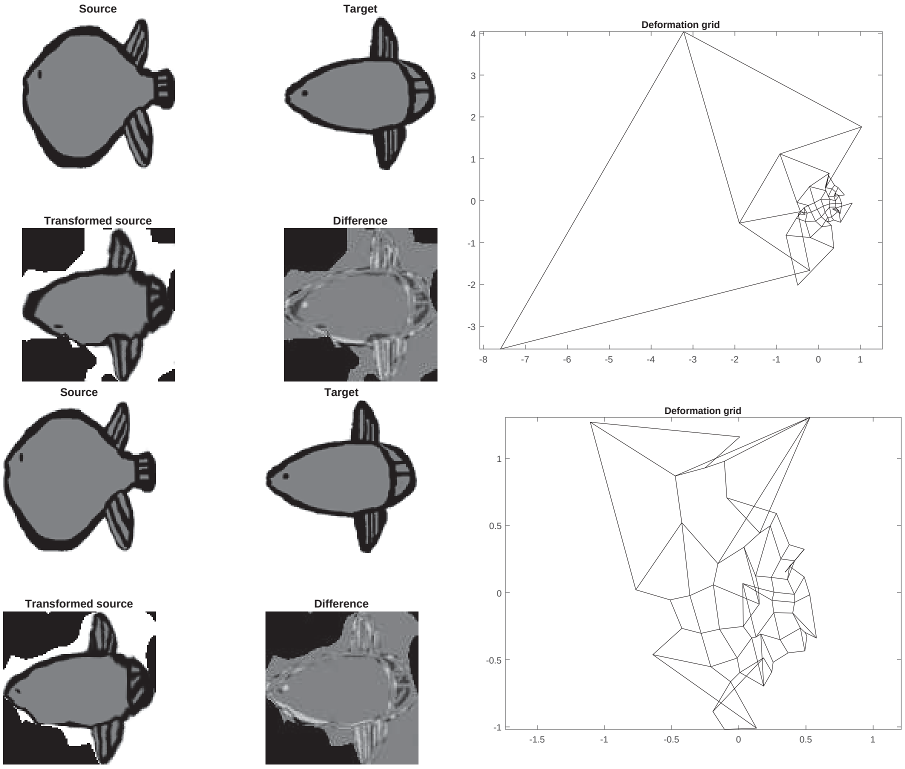


Figure 3.35: Results of conformal image registration for Example 3.11 using the linear discrete Cauchy–Riemann equations as a penalty term (top) and the nonlinear form (bottom). Note that the size of the top grid is $[-8,-1]\times[1,4]$ which is far away from the domain $[-0.5,0.5]\times[-0.5,0.5]$. However, the majority portion of the bottom grid is within the domain.

		Linear form		**Nonlinear form**	
	λ	RE_{lin}	$\|\|P_2\|\|$	RE_{nonlin}	$\|\|P_3\|\|$
$n^2 = 16$	20^4	12.5076	0.0000	17.3039	0.0000
	20^3	12.5076	0.0000	12.1365	0.0001
	20^2	12.4979	0.0008	10.2304	0.0016
	20	8.9796	0.2175	7.1921	0.1591
$n^2 = 64$	20	6.2780	0.1517	5.3147	0.1848
	20	15.3656	0.3739	13.7724	0.1712

Table 3.5: Registration errors and penalty terms for Example 3.11. (The images are smoothed in the first 5 rows, and unsmoothed in the last row.)

The registrations in both cases are good, certainly better than with the first form of the penalty term. The final registration errors for the first, second (linear), and third (nonlinear) penalty terms are 19.0665, 15.3656, and 13.7724, respectively. However, the grids are much less regular than before, even in the central part of the images, and both grids indicate non-invertible mappings.

Our overall conclusion among penalty terms P_1, P_2, P_3 is that P_2 and P_3 produce better results for image registration than P_1, but between P_2 and P_3 we are unable to say which form is better: for some pair of images P_2 gives better results and sometimes P_3 produces better results. Therefore, the question of the choice of the best penalty term for the optimisation function (which was defined in Equation (3.3)) is still open. While it is possible that further development of P_2 and P_3, including perhaps a deeper study of the effect of λ and additional constraining terms to nudge the mapping towards a diffeomorphism, could be helpful, we have chosen instead to proceed towards the development of a final, and completely different, form of the penalty term. This is discussed in the next section.

3.6 Fourth form of the penalty term

Let $\varphi\colon \Omega \to \mathbb{C}$ be a conformal map. Expanding about any point $z_0 \in \Omega$ in Taylor series, we have

$$\varphi(z) = \varphi(z_0) + (z - z_0)\varphi'(z_0) + \mathcal{O}(|z - z_0|^2).$$

Thus, conformal maps are local similarities in the neighbourhood of any point. Therefore, small squares map to squares, with an error than vanishes as the square becomes smaller. This can be seen in the example in Figure 3.36. Therefore, we use this geometric property of the conformal transformation to compute our fourth penalty term.

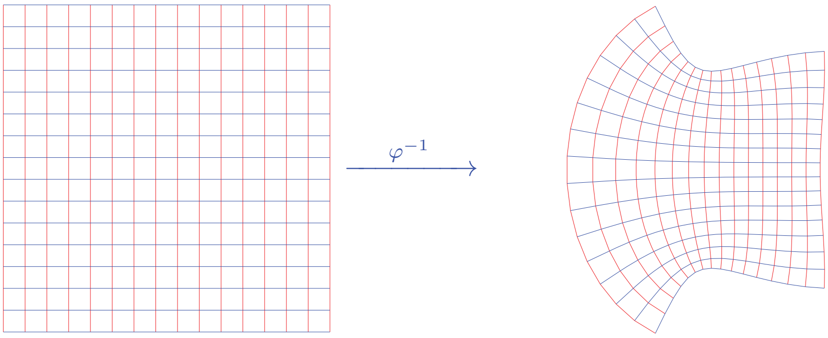
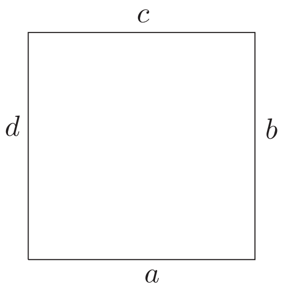


Figure 3.36: *Left:* A uniform square grid of size sixteen is presented. *Right:* Its image under the conformal map $\varphi^{-1}(z) = 1.5 + 0.2z - 0.4z^2 + 0.1z^3 - 0.1z^4$.

Consider a square whose sides are a, b, c, d, respectively, considered as displacement vectors in the complex plane (equivalently, as complex numbers) taken in an anticlockwise sense.



The following equations hold for a square:

$$\left.\begin{aligned} a + \mathrm{i}b &= 0\\ b + \mathrm{i}c &= 0\\ c + \mathrm{i}d &= 0\\ d + \mathrm{i}a &= 0 \end{aligned}\right\} \tag{3.22}$$

This leads us to define our fourth penalty term as:

$$P_4(\hat{y}_{ij}) = \frac{(a + \mathrm{i}b, b + \mathrm{i}c, c + \mathrm{i}d, d + \mathrm{i}a)}{||(a, b, c, d)||}$$

for $i, j = 1, \ldots, n-1$. Here a, b, c, and d are the vectors of the sides of the quadrilateral with vertices $\hat{y}_{i,j}$, $\hat{y}_{i+1,j}$, $\hat{y}_{i+1,j+1}$, and $\hat{y}_{i,j+1}$. Thus the fourth penalty term in Equation (3.22) contains 4 complex or 8 real terms per cell. There are n^2 cells, so the penalty contains $8n^2$ real terms altogether.

Each term $P_4(\hat{y}_{ij})$ corresponding to a single cell vanishes if, and only if, the quadrilateral is a square. In addition, P_4 is invariant under translations and rotations of the vertices of the quadrilateral, as well as under cyclic permutation of the vertices. It therefore has a claim to be a reasonable concept of 'discrete conformality'. Like P_2 and P_3 it is associated with the cells, rather than the control points like P_1, but compared to P_1, P_2, and P_3 it controls far more terms. Note that for any quadrilateral, a, b and c together determine d, so only 6 of the 8 real terms are independent.

If $P_4(\hat{y}_{ij}) = 0$ for all $i, j = 1, \ldots, n-1$, then the transformation determined by the $\hat{y}_{ij}$ is rigid. So P_4 is like P_1 in this regard. Therefore $\lambda \to \infty$ will yield only rigid transformations, and we again regard λ as a regularisation parameter, offering the opportunity of parameter continuation between $\lambda = 0$ (unconstrained image registration), moderate λ (squares map to near squares), and large λ (rigid registration).

As we will be minimising the sum of the registration error plus λ times $\|P_4\|^2$, the solutions will map squares to near squares only in an average sense; some squares may suffer extreme distortion. In view of the difficulties faced by our first three penalty terms, we argue that the extra flexibility this allows will be needed. A possible topic for future research would be to replace P_4 by related functions (such as P_4^α for $\alpha > 1$) that penalise non-squareness more severely.

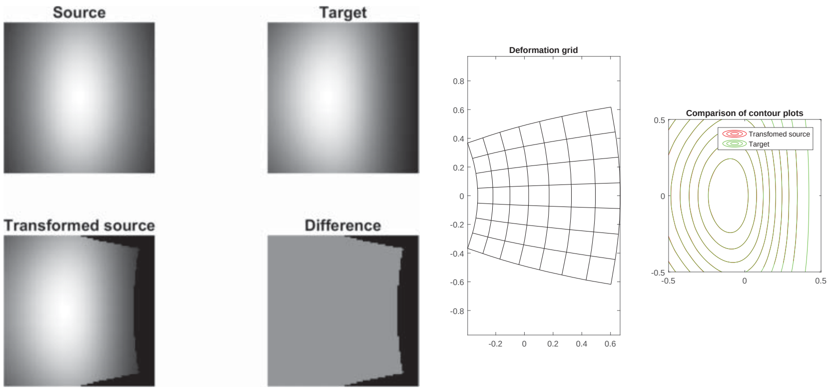
3.6.1 Fourth form with smooth images

In this section we present image registration using penalty term P_4 of those smooth images that were first discussed in Section 3.4.1 and then in Section 3.5.1.

Example 3.12. In this example, we reconsider the pair of images of Example 3.1 for image registration. The results can be seen in Figure 3.37.

This verifies that the fourth penalty term reproduces essentially the same results as the first three on this (easy) example. The grids that are calculated for the various values of λ are shown in Figure 3.38.

We have also considered an alternative way of selecting λ. When $N = 100$ and $n = 4$, there are 10,000 terms in the registration error (each of which lies in the range $[0, 1]$ in magnitude), and $8(n-1)^2 = 72$ terms in the penalty term (each of which also



	λ	$\|\|(I_1 \circ \varphi^{-1})(\hat{x}_{ij}) - I_2(\hat{x}_{ij})\|\|$	$\|\|P_4\|\|$
$n^2 = 16$	20^4	2.7638	0.0000
	20^3	2.7635	0.0000
	20^2	2.6720	0.0012
	20	0.4708	0.0343
$n^2 = 64$	20	0.1001	0.0082

Figure 3.37: The results of conformal image registration with the fourth penalty term applied to the images in Example 3.12.

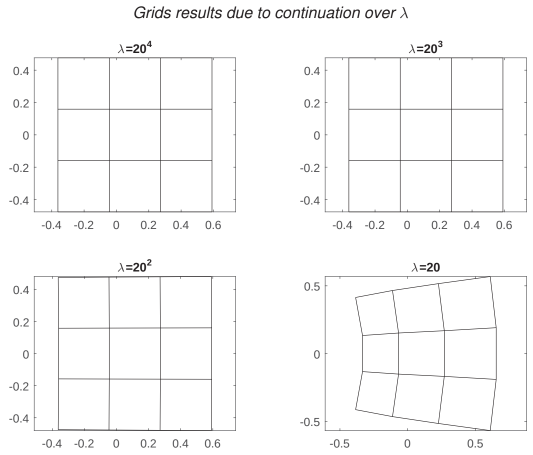
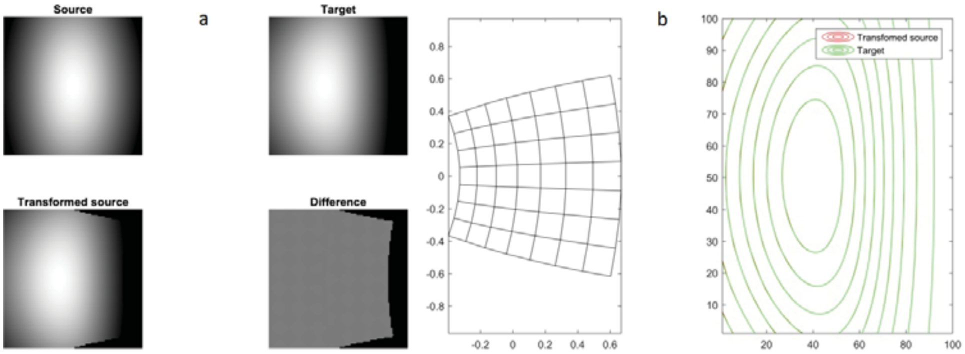


Figure 3.38: Four grids are shown that were calculated for Example 3.12 and correspond to $\lambda = 20^4, 20^3, 20^2, 20$ respectively for sixteen control points. The first three grids in the first row represent perfect rigid grids because squares are mapped to squares in each grid without showing any deformation. The fourth grid is a non-rigid conformal grid.

lies in the range $[0, 1]$). Therefore if we choose $\lambda = 150$, the two terms are roughly of equal magnitude. Choosing the regularisation parameter so that the objective and the regularisation term are roughly of equal magnitude is a standard rule of thumb in the regularisation method, for example, in inverse problems.

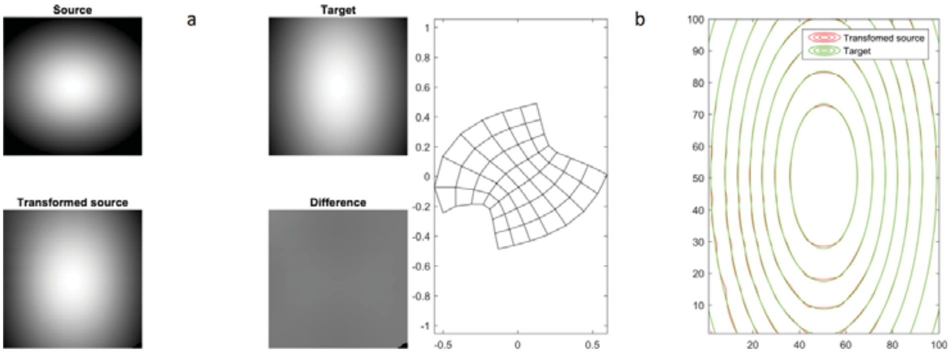
We repeat Example 3.12 with λ fixed at 150. The results are shown in Figure 3.39 and are essentially the same as in Figure 3.37. The registration errors with the 4 penalty terms are 0.1690, 0.1023, 0.1016, and 0.1001, respectively. The fourth form produced lower registration error than the previous penalty terms for this example.



n^2	λ	$\|\|(I_1 \circ \varphi^{-1})(\hat{x}_{ij}) - I_2(\hat{x}_{ij})\|\|$	$\|\|P_4\|\|$
16	150	2.2166	0.0072
64	150	0.1001	0.0082

Figure 3.39: Results of conformal image registration applied to Example 3.37 with the fourth penalty term and $\lambda = 150$.

Example 3.13. The second example involves the smooth, but independent, images from Example 3.2. The results with the fourth penalty term and $\lambda = 150$ are shown in Figure 3.40.



n^2	λ	$\|\|(I_1 \circ \varphi^{-1})(\hat{x}_{ij}) - I_2(\hat{x}_{ij})\|\|$	$\|\|P_4\|\|$
16	150	13.4432	0.0304
64	150	2.2772	0.0183

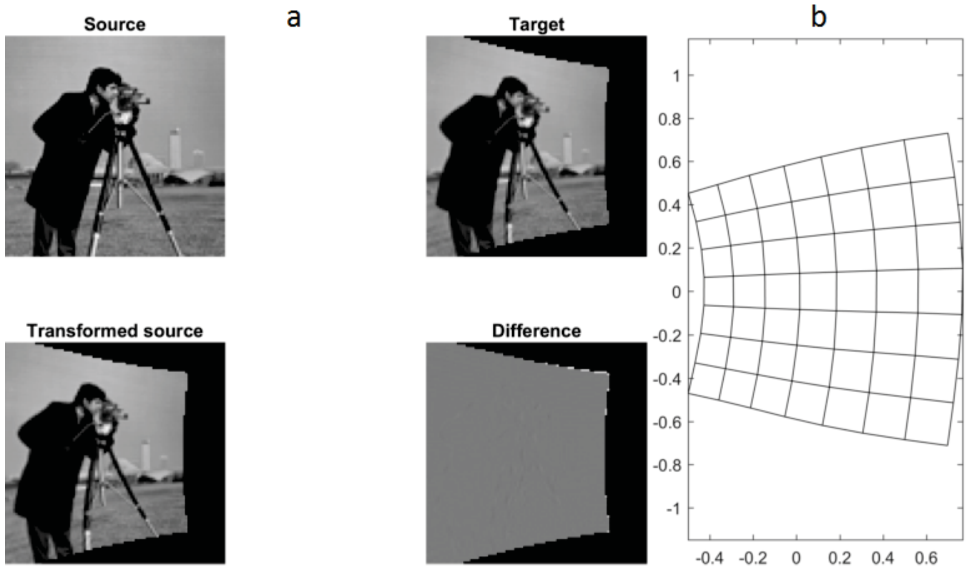
Figure 3.40: Results of conformal image registration for Example 3.13 with the fourth penalty term and $\lambda = 150$. The deformation grid in this example is rather twisted compared to the grids in Examples 3.2 and 3.8. We have used Matlab's *lsqnonlin* optimiser in all three examples, but for some reason in this case the optimiser first rotated the grid before making small deformations. This kind of rotation can also be seen in Ch 5, Example 5.1.

The registration errors with the 4 penalty terms are 0.9199, 1.0888, 0.1587, and 2.2772, respectively. The map found by the 4th penalty term does visibly possess the 'squares map to squares' property of discrete conformality, although it is not as regular as those found with the earlier penalty terms. The registration is satisfactory (mean error 0.023 per pixel) but not as good as those found earlier. We will see in Example 3.15 that the registration found by this method may depend sensitively on λ and on the precise continuation method used to solve the optimisation problem.

3.6.2 Fourth form with non-smooth images

After successful conformal image registration with smooth images, we now present three examples of non-smooth images. In all the examples, we first smooth the images to obtain a better initial guess, as we have done previously. This initial guess is obtained without the continuation method over the values of λ, instead $\lambda = 150$ is fixed throughout the experiment; see the next example for further discussion of this point. Except for the continuation method over the values of λ, the whole procedure will remain the same as in Sections 3.4 and 3.5.

Example 3.14. This is the cameraman image, which is not smooth, with the target generated from the source. The results with the fourth penalty term are shown in Figure 3.41.

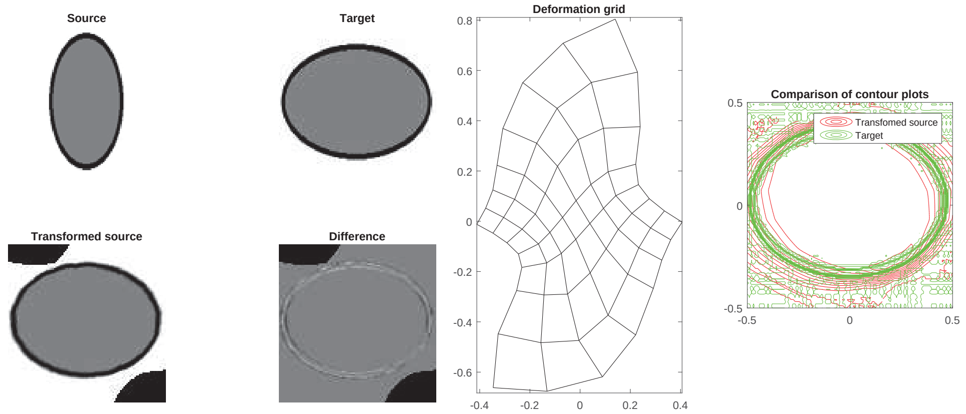


n^2	λ	$\|\|(I_1 \circ \varphi^{-1})(\hat{x}_{ij}) - I_2(\hat{x}_{ij})\|\|$	$\|\|P_4\|\|$
16	150	2.0377	0.0084
64	150	1.0215	0.0053
64	150	0.5278	0.0069

Figure 3.41: Results of conformal image registration for Example 3.14 with the fourth penalty term and $\lambda = 150$.

The images are registered perfectly and the deformation grid is identical to the grids of the first form and third form (nonlinear) for this example. The registration errors for the first, second (linear), third (nonlinear) and fourth penalty terms are 0.5225 (see Figure 3.17), 2.5611 (see bottom row in the third column of Table 3.3), 0.4778 (see bottom row in the fifth column of Table 3.3) and 0.5278 respectively. We now move to more challenging examples that contains the pairs of images of Examples 3.4 and 3.5.

Example 3.15. The fourth example contains non-smooth independent images, presumed to be conformally related, and that have already been attempted with the first, second (linear) and third (nonlinear) penalty terms. These are the ellipses of Example 3.4. Instead of setting $\lambda = 150$, we explore the dependence of the registration on λ in detail.* It can be extremely difficult to get conclusive information about the existence of local minima. After some experimentation, we have settled on a continuation strategy of first solving the optimisation problem directly at $\lambda = 150$ (first smooth and then non-smooth), and then performing parameter continuation in λ by increasing or decreasing λ in a constant ratio for non-smooth images. The results of these experiments are given in Figures 3.43 (smaller λ, i.e. less conformality enforced) and 3.44 (larger λ, i.e. more conformality enforced) respectively. The results for $\lambda = 150$ itself are given in Figure 3.42.



n^2	λ	$\|\|(I_1 \circ \varphi^{-1})(\hat{x}_{ij}) - I_2(\hat{x}_{ij})\|\|$	$\|\|P_4\|\|$
16	150	16.0863	0.0514
64	150	4.1880	0.0440
64	150	5.7348	0.0542

Figure 3.42: Results of conformal image registration for Example 3.15 with the fourth penalty term with $\lambda = 150$ are given.

---

[*] The results presented here are the the endpoint of an extensive series of numerical experiments and explorations, not presented in detail. Early experiments yielded very puzzling results. The example presented here shows our resolution of this puzzle. Although not universal, is an attempt to portray a not uncommon feature of the method as we observed it.

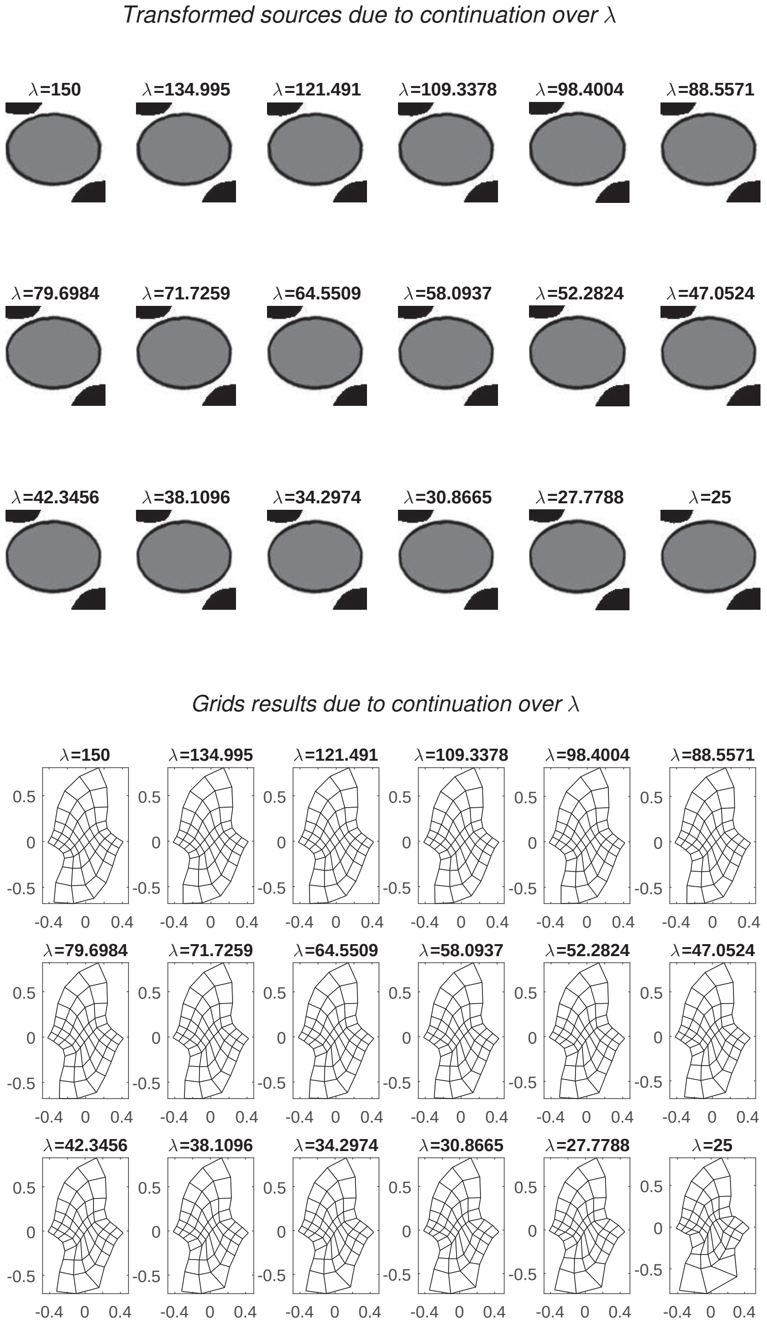


Figure 3.43: *Top:* A series of transformed sources as λ decreases geometrically. *Bottom:* The corresponding grids.

Recall that the first, second, and third forms were unable to combine good registration with a smooth conformal grid in this example. The first set of results here, for $\lambda \in [25, 150]$, are as expected. All the registrations are visually good[‡]. The registration errors are in [3.9,5.73]. In the earlier experiments the registration errors were 4.6213,

---

[‡] Good visual registration means that there are at most small mismatched portion between the transformed source and the target, and the grid looks smooth and conformal.

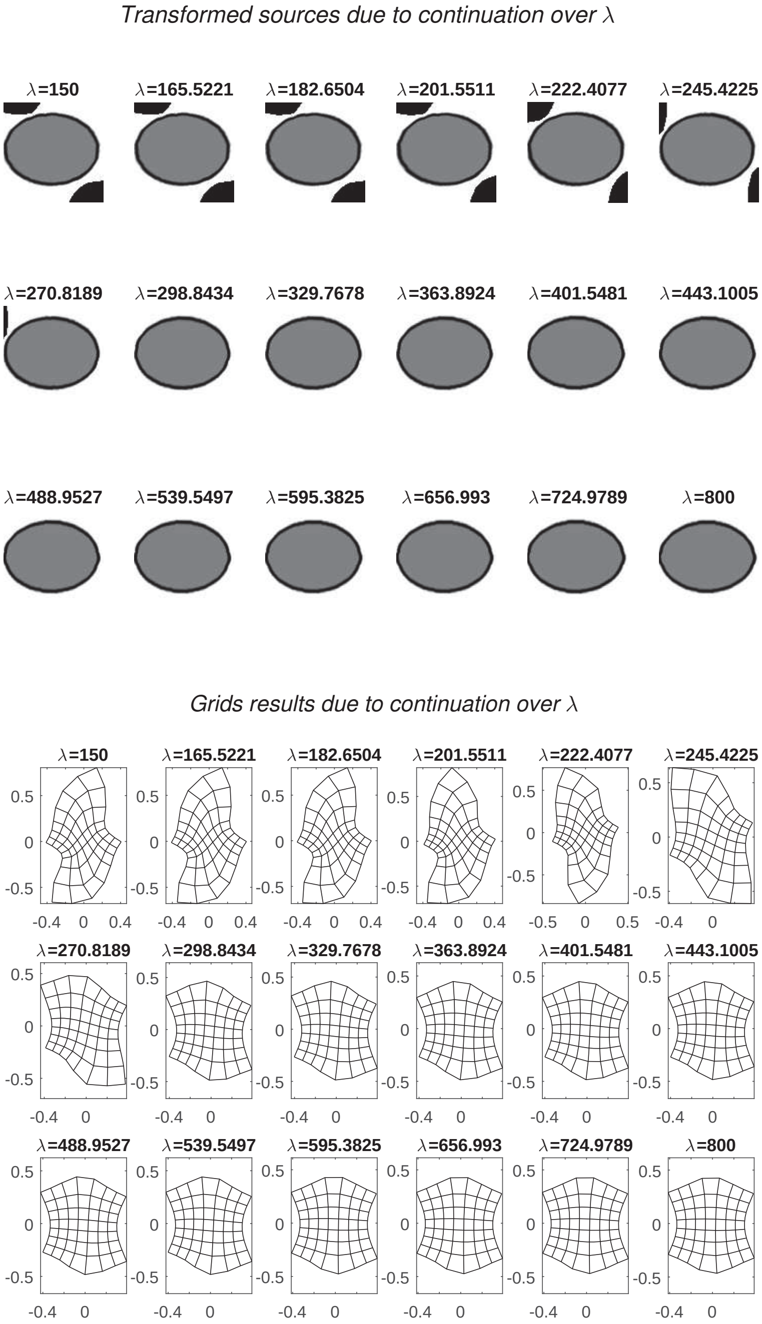


Figure 3.44: *Top:* Series of transformed sources are given when λ is increasing geometrically. *Bottom:* Series of grids are presented corresponding to transformed sources. A striking global change in the grid is observed between $\lambda = 200$ and $\lambda = 300$.

8.5167, and 3.9411 for the three penalty terms, see Figure 3.21 and Table 3.4, but the grids were non-invertible for the second and third forms. In this example, deformation grids are smooth and possess the 'squares map to squares' property, degrading only slightly at the smaller values of λ (see Figure 3.43).

One way to determine a regularisation parameter is via the L-curve, shown in Figure 3.45, named for its typical shape (a letter L).

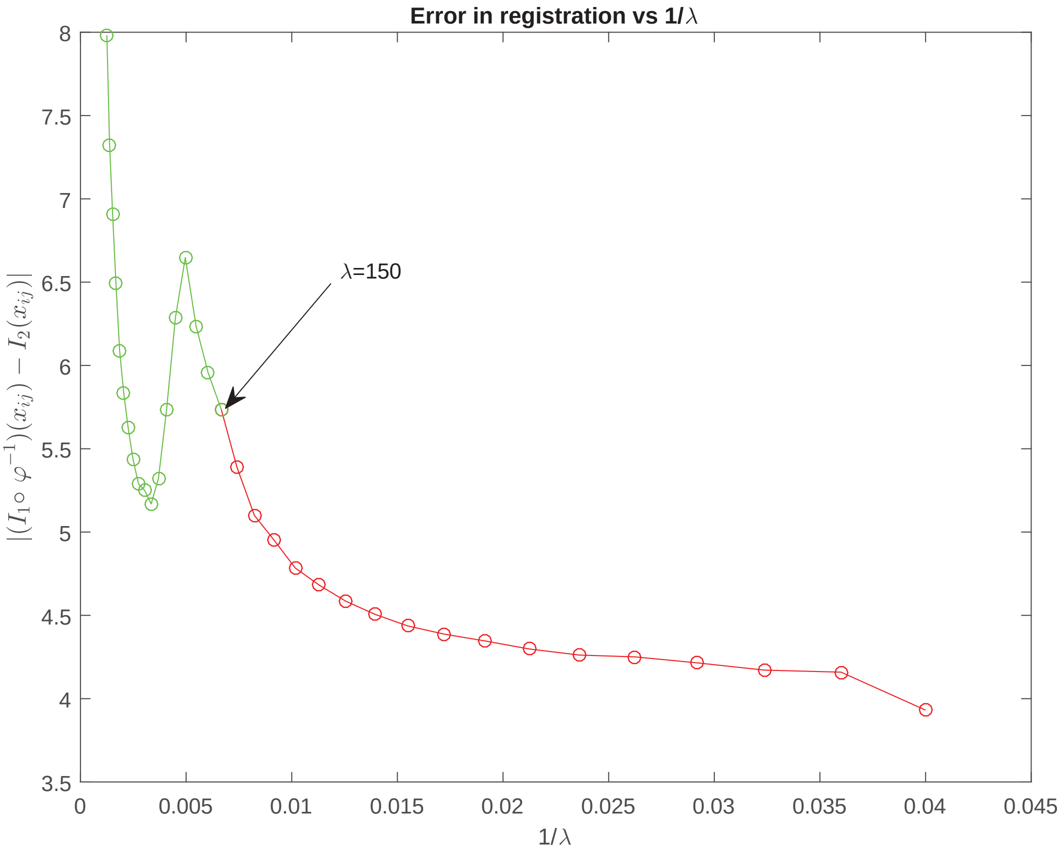


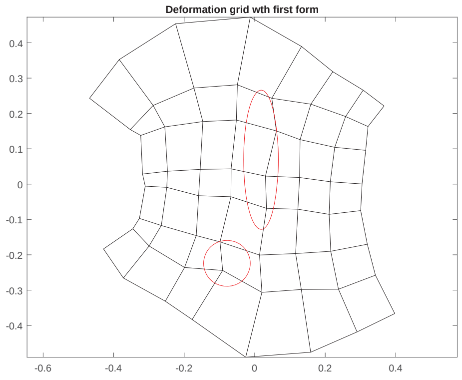
Figure 3.45: L-curve for Example 3.15.

This is the graph of the objective function (here the registration error) vs the regularisation parameter $1/\lambda$. A very standard rule of thumb is to choose λ at the 'corner' (point of greatest Euclidean curvature) of the L-curve. (Note, however, that this point is not invariant under rescalings of the axes.) In this case there is a corner around $\lambda = 100$ to $\lambda = 150$. Thus our earlier, naive, strategy of simply choosing λ to balance the two terms in the objective function does actually work well in this example.

However, the L-curve is not at all L-shaped in this example! The reason why is revealed in Figure (3.44), which shows $\lambda \in [150, 800]$. As λ increases between 200 and 300 there is a dramatic change in the mapping and the registration error actually starts to decrease again, reaching a local minimum of 5.2 (better than before!) at $\lambda \approx 300$ before again rapidly increasing.

What we suspect is going on here is that the objective function has two strong local minima. Each shows a good registration. The continuation method has followed one local minimum—the one that was more or less accidentally landed on at $\lambda = 150$—until it ceased to be a minimum; it has then rapidly dropped to the other one. A possible topic for future research is to explore the landscape of the objective function in more detail for this and related examples.

However, we stress the overall conclusion that for this example, the 2nd and 3rd forms produced disastrously poor grids, while the first form was better and is compared to the 4th form in Figure 3.46. Even though the first form is sitting on what we now know is the other local minimum, it has still not produced a very regular grid. The fourth form, in contrast, can produce a *range* of very smooth discrete conformal mappings combined with excellent registration. In this example the 4th form is clearly the best choice.



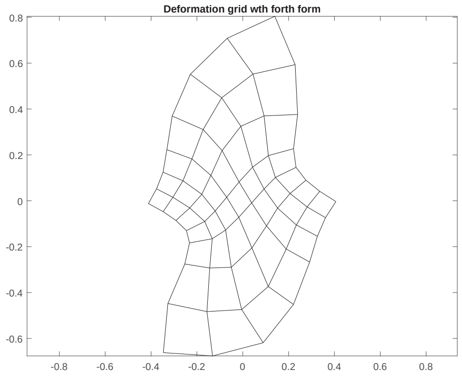
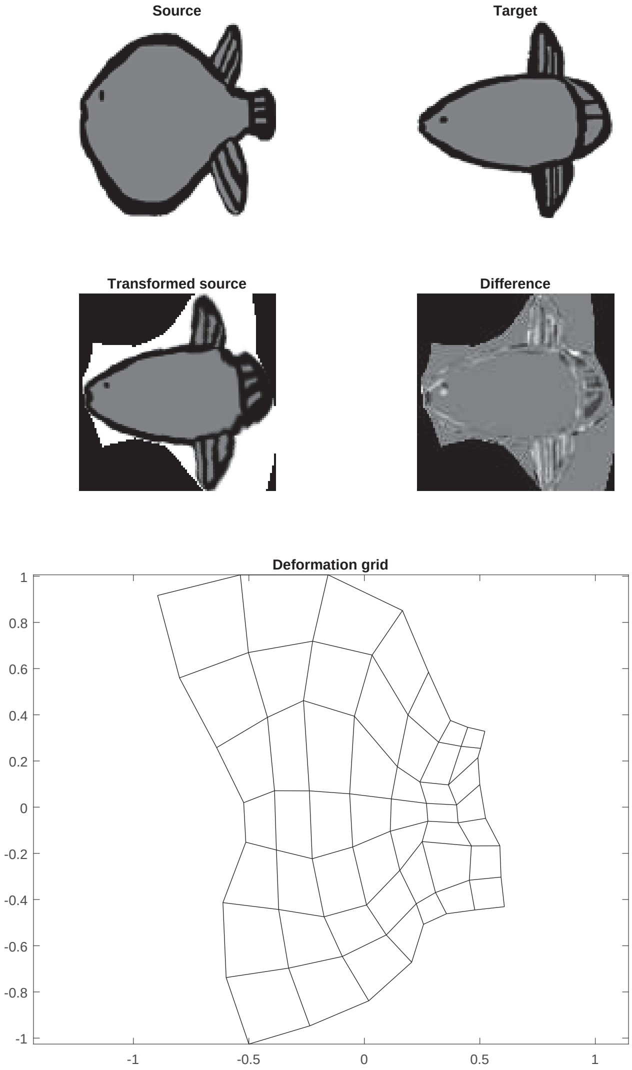


Figure 3.46: *Top:* Deformation grid of Example 3.4 (which used the first form of the penalty term); red circles indicate the non-smooth or non-conformal parts. *Bottom:* Deformation grid of Example 3.15 using the 4th form of the penalty term.

Example 3.16. We finally present the example of Thompson's pair of fishes, for which none of the previous 3 penalty terms produced acceptable grids, even though they succeeded in finding a reasonable image match. In this example, we use the same strategy that was used in the previous example. The series of transformed sources and corresponding grids are given in Figures 3.47 (for $\lambda = 150$), 3.48 and 3.49 respectively. The errors in registration for both the experiments are given in Figure 3.50.



n^2	λ	$\|\|(I_1 \circ \varphi^{-1})(\hat{x}_{ij}) - I_2(\hat{x}_{ij})\|\|$	$\|\|P_4\|\|$
16	150	15.6537	0.0825
64	150	6.7064	0.0502
64	150	13.1865	0.0688

Figure 3.47: Results of conformal image registration for Example 3.15 with the fourth penalty term with $\lambda = 150$.

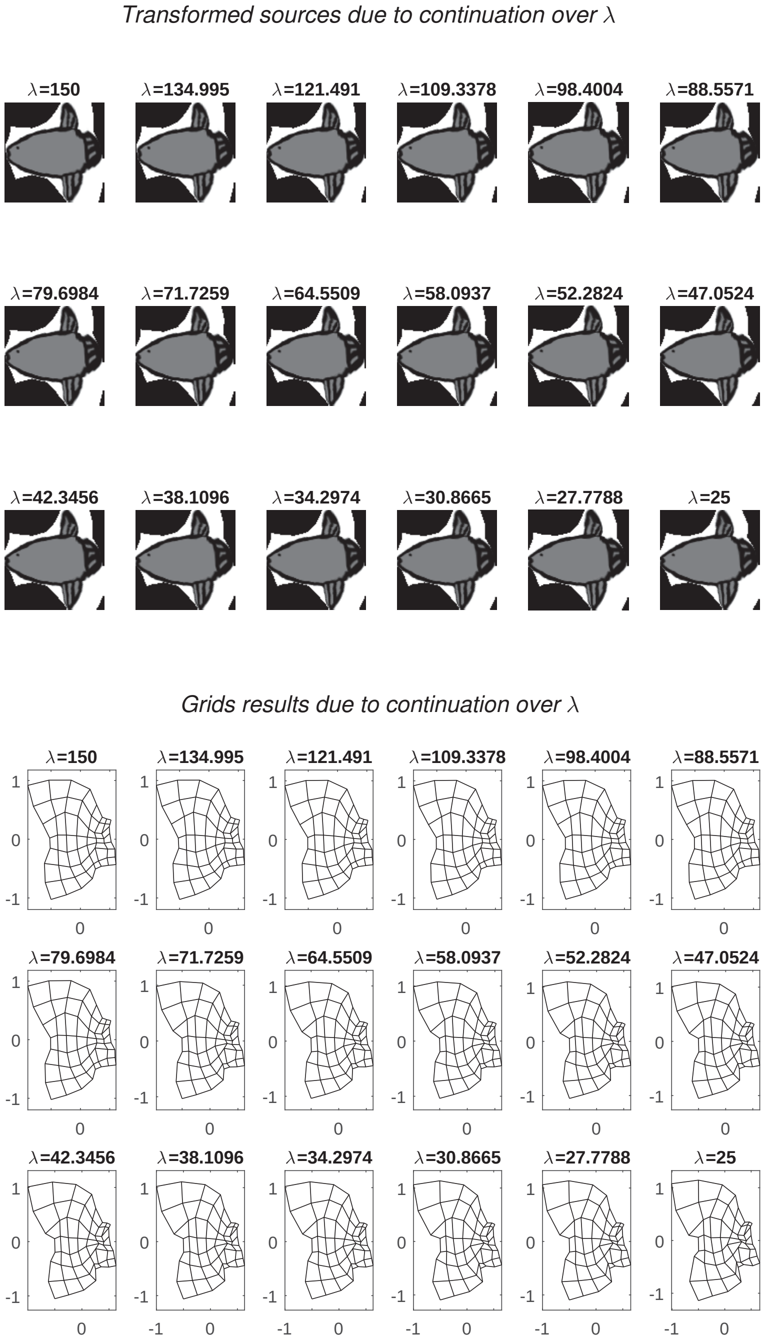


Figure 3.48: *Top:* A series of transformed sources as λ decreases geometrically for Example 3.16. *Bottom:* The corresponding grids. A marked deterioration in the grid is observed at the smallest values of λ.

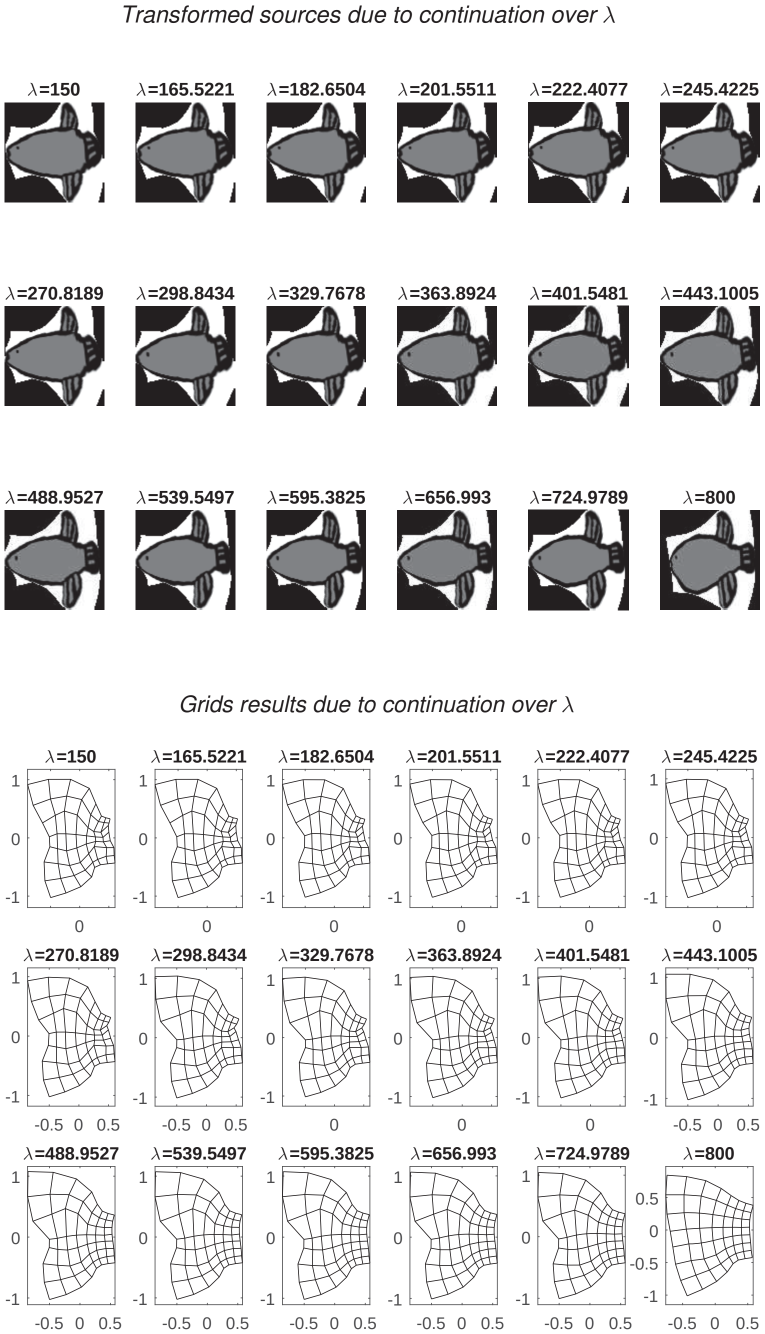


Figure 3.49: *Top:* A series of transformed sources as λ increases geometrically for Example 3.16. *Bottom:* The corresponding grids.

This time, the variation of the grids with λ is not as marked. Instead, what is striking is that the registration attempts to 'hold on' to a specific choice of grid over a very wide range of λ. (See, for example, the non-square cell near the top of the tail fin, and the dimple at the mouth.) This indicates that the grid is very highly forced by the image matching part in these regions. The final results, both for the registration and the grid, are very good; there remains only a partly subjective question of whether the grid shown in Figure 3.47 qualifies as 'discrete conformal'. (Although the registration errors were smaller with the first 3 penalty terms, their grids were so poor, e.g. noninvertible, that we discount them.) The actual registration of the fishes here, including fins, tail, body shape, and head, is excellent.

As λ increases, it has to reach as high as 700 before the registration becomes visibly worse. Based solely on the L-curve in Figure 3.50, a value of $\lambda \approx 200$ would be chosen. The L-curve does not display the turning point found in the previous example, although there are a couple of puzzling plateaus.

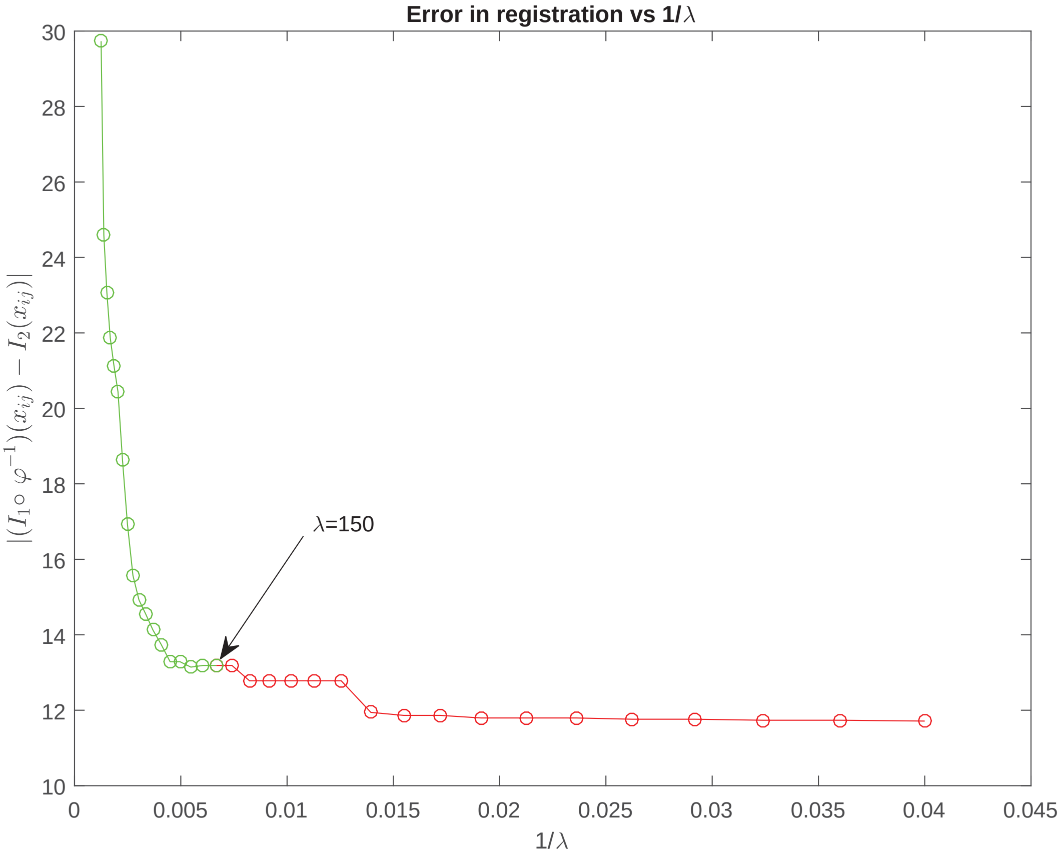


Figure 3.50: Red lines shows the path of error while λ is decreasing. On the other hand, path of error function while λ is increasing is given with the green line. $\lambda = 150$ is a threshold point where both paths coincide.

In this chapter we have presented the construction and development of an algorithm for image registration using conformal diffeomorphisms using a set of control points in a uniform grid. In the next chapter we present a method of conformal image registration for 2D discrete images using gradient descent.

Chapter 4

Gradient Flow

Mathematicians have tried in vain to this day to discover some order in the sequence of prime numbers, and we have reason to believe that it is a mystery into which the human mind will never penetrate.

– Leonhard Euler (1707-1783)

In the previous chapter, we presented a method for the construction of conformal diffeomorphisms by using a set of control points. That method was a constrained optimisation approach, and we considered four different penalty terms that aimed to enforce conformality in the warps that were used to make the images match. In this chapter we consider an alternative approach to conformal image registration that is based on computing the gradient flow between a pair of images (the source and the target). One benefit of this approach is that rather than requiring a penalty term to enforce conformality, the gradient can be computed such that the deformation is conformal by construction.

The idea of constructing diffeomorphisms by solving ordinary differential equations numerically using gradient descent seems to have been introduced by Trouvé in [102], where he first proposed using the action of an infinite dimensional group to deform images. The idea is to find the directional derivative of the usual sum-of-squares error E between deformed image $I_1 \circ \varphi^{-1}(x)$ and image I_2 with respect to the deformation φ, and then to compute the Eulerian gradient of E. This gives a simple gradient descent algorithm, which is demonstrated in [103] on black-and-white images, including the 'C-experiment' (where a small section of the letter C is deformed into the complete letter) that he borrowed from [25].

A variant on the method is introduced in Section 10.4 of Younes' book [114], where the silhouettes of a hen and a cat are registered. Although these images are black-and-white, they are relatively complicated shapes and the method appears to work well. Two versions of the update PDE for the algorithm are described by Younes, one providing an update of the diffeomorphism φ, and the other describing how the image

$I_1 \circ \varphi^{-1}$ evolves. In the first approach the source and the target images are fixed and the estimate of the diffeomorphism is updated at each timestep, while in the second approach the source image evolves. The two methods are equivalent in theory, but there can be numerical differences between them in practice. In either case, a stopping criterion is needed to decide when to halt the integration, since there is no guarantee that there is a diffeomorphism that perfectly registers any given pair of images. We will discuss stopping criteria more as we develop our algorithm.

Methods of building diffeomorphisms for image registration, including Trouvé's method, rely principally on the fact that composition of infinitesimal diffeomorphisms produces another diffeomorphism. This is true for conformal diffeomorphisms as well: composition of infinitesimal conformal diffeomorphisms produces another conformal diffeomorphism (see Proposition 1.1 in Chapter 1). Therefore, a similar approach can be used to construct a registration between images that are (or at least, may be) related by a conformal diffeomorphism, and we derive the algorithm and demonstrate it in this chapter. The conformal diffeomorphism φ^{-1} is represented in the form of a Taylor series whose coefficients are determined by solving a system of ordinary differential equations. In our implementation this is performed numerically by using gradient descent.

We now introduce our gradient flow method for constructing conformal diffeomorphisms, and test it on a series of test images. The algorithms from the previous chapter and this chapter are compared in the next chapter.

4.1 A Gradient Flow Algorithm for Conformal Diffeomorphisms

Definition 4.1. A function $f(z)$ $(z \in \mathbb{C})$ is said to be analytic in an open set Ω of the complex plane $\mathbb{C}$ if, for any $z \in \Omega$,

$$f(z) = a_0 + a_1 z + a_2 z^2 + \ldots = \sum_{k=0}^{\infty} a_k z^k.$$

In the next theorem, we present a construction of a conformal diffeomorphism that is represented by a Taylor series and whose coefficients are determined by solving a system of ordinary differential equations. The numerical implementation is based on gradient descent method [110] (Chapter 7), [23], [19] as will be described in the following section. We define an integration time t, such that the source image is found at $t = 0$, and successive deformations are found as time increases.

Theorem 4.1. *Suppose that φ is a conformal diffeomorphism such that φ and φ^{-1} are analytic inside a disc $\Omega \subset \mathbb{C}$ and I_1 and I_2 are differentiable images defined on some* *subset of $\mathbb{C}$. If the discrepancy between the images is measured with the sum-of-squares error functional term*

$$M = \int_\Omega \{I_1 \circ \varphi^{-1}(z) - I_2(z)\}^2 dxdy \tag{4.1}$$

(where $z = (x, y)$ represents a complex number) then the registration between the images is subject to the solution of the following system (where $\bar{\cdot}$ denotes the complex conjugate and ∇I_1 is the matrix of spatial gradients of the image):

$$\dot{a}_k = -2 \int_\Omega \{I_1 \circ \varphi^{-1}(z) - I_2(z)\} \nabla I_1 \circ \varphi^{-1}(z) \overline{z^k} dxdy; \quad k = 0, 1, 2, \ldots \tag{4.2}$$

Proof. Since φ and φ^{-1} are analytic inside a disc they can be represented in the form of a Taylor series. In particular, $\varphi^{-1} = \sum_{k=0}^{\infty} a_k z^k$, where $a_k = A_k + \mathrm{i}B_k, A_k, B_k \in \mathbb{R}, k = 0, 1, 2, \ldots, \infty$; we need to find values for the coefficients, $a_k = A_k + \mathrm{i}B_k$ in order to construct a φ^{-1} so that the matching functional (4.1) is as small as possible. We use gradient descent in each a_k independently, starting from $k = 0$, since each subsequent term is based on the previous ones.

The path of each a_k is computed using:

$$\dot{a}_k = -\nabla_{a_k} M, \quad k = 0, 1, 2, \ldots . \tag{4.3}$$

Here, $\dot{a}$ denotes the derivative of a with respect to (integration) time (from source to target) and $\nabla_{a_k} = \frac{\partial}{\partial A_k} + \mathrm{i}\frac{\partial}{\partial B_k}$ represents the complex gradient with respect to each coefficient a_k of the Taylor series. Now, equations (4.1) and (4.3) yield (where the second line follows because $I_2(z)$ is not a function of the a_k):

$$\begin{aligned} \dot{a}_k &= -2 \int_\Omega (I_1 \circ \varphi^{-1}(z) - I_2(z)) \nabla_{a_k} (I_1 \circ \varphi^{-1}(z) - I_2(z)) dxdy, \\ &= -2 \int_\Omega (I_1 \circ \varphi^{-1}(z) - I_2(z)) \nabla_{a_k} (I_1 \circ \varphi^{-1}(z)) dxdy. \end{aligned} \tag{4.4}$$

The first term is the pixel-wise difference between the transformed source and target and is easy to compute. The second term, which is the gradient of the mapping, can be computed as:

$$\begin{aligned}
\nabla_{a_k}\left(I_1\circ\varphi^{-1}(z)\right) &= \left(\frac{\partial}{\partial A_k}+\mathrm{i}\frac{\partial}{\partial B_k}\right)\left(I_1\circ\sum_{k=0}^{\infty}a_kz^k\right),\\
&= \left(\frac{\partial}{\partial A_k}+\mathrm{i}\frac{\partial}{\partial B_k}\right)\left(I_1\left(\sum_{k=0}^{\infty}a_kz^k\right)\right).
\end{aligned}$$

Separating the real and imaginary parts of $\sum_{k=0}^{\infty}a_kz^k$ (represented by the variables X and Y respectively), we get:

$$\begin{aligned}
\nabla_{a_k}\left(I_1\circ\varphi^{-1}(z)\right) &= \left(\frac{\partial}{\partial A_k}+\mathrm{i}\frac{\partial}{\partial B_k}\right)\left(I_1\left(\underbrace{Re\left(\sum_{k=0}^{\infty}a_kz^k\right)}_{X},\underbrace{Im\left(\sum_{k=0}^{\infty}a_kz^k\right)}_{Y}\right)\right),\\
&= \left(I_{1X}\frac{\partial X}{\partial A_k}+I_{1Y}\frac{\partial Y}{\partial A_k}\right)+\mathrm{i}\left(I_{1X}\frac{\partial X}{\partial B_k}+I_{1Y}\frac{\partial Y}{\partial B_k}\right), \qquad (4.5)
\end{aligned}$$

where I_{1X} and I_{1Y} denote the partial derivatives of image I_1 with respect to X and Y respectively. Writing $z^k=u+\mathrm{i}v$ and recalling that $a_k=A_k+\mathrm{i}B_k$ we obtain (where ∇ denotes the complex gradient):

$$\begin{aligned}
\nabla_{a_k}\left(I_1\circ\varphi^{-1}(z)\right) &= \left(I_{1X}\frac{\partial}{\partial A_k}(A_ku-B_kv)+I_{1Y}\frac{\partial}{\partial A_k}(B_ku+A_kv)\right)+\\
&\qquad \mathrm{i}\left(I_{1X}\frac{\partial}{\partial B_k}(A_ku-B_kv)+I_{1Y}\frac{\partial}{\partial B_k}(B_ku+A_kv)\right)\\
&= \left(I_{1X}(u)+I_{1Y}(v)\right)+\mathrm{i}\left(I_{1X}(-v)+I_{1Y}(u)\right),\\
&= \left(I_{1X}+\mathrm{i}I_{1Y}\right)(u-\mathrm{i}v),\\
&= \nabla I_1(X,Y)\overline{z^k},\\
&= \nabla I_1\circ\varphi^{-1}(z)\overline{z^k}.
\end{aligned}$$

Plugging this into Equation (4.4), we get:

$$\dot{a}_k=-2\int_{\Omega}\{(I_1\circ\varphi^{-1})(z)-I_2(z)\}\nabla I_1\circ\varphi^{-1}(z)\overline{z^k}dxdy,\quad k=0,1,2,\ldots \qquad (4.6)$$

□

Equation (4.6) denotes the system of differential equations for the coefficients of the Taylor series. The first term (in the curly brackets) computes the pixel-wise difference between the transformed image I_1 and I_2. For the computation of the complex gradient, we calculate the x and y gradients and then combine them. $\nabla I_1\circ\varphi^{-1}(z)$ denotes the image derivatives ∇I_1 evaluated at points of the image after the action of φ^{-1}.

Proposition 4.2. *If φ^{-1} is represented by the truncated Taylor series $k = 0, 1, \ldots, K$ then Equation (4.6) still holds for each $a_k, k = 0, 1, \ldots, K$.*

Proof. The only difference between this proposition and the previous theorem is the representation of φ^{-1}, which is now an approximation to the full Taylor expansion. Representing φ^{-1} by the first K terms makes the finite Taylor series:

$$\varphi^{-1} \approx \sum_{k=0}^{K} a_k z^k. \tag{4.7}$$

Following the steps of theorem (4.1) we see that:

$$\dot{a}_k = -2 \int_{\Omega} \{(I_1 \circ \varphi^{-1})(z) - I_2(z)\} \nabla I_1 \circ \varphi^{-1}(z) \overline{z^k} dxdy. \quad k = 0, 1, 2, \ldots, K \tag{4.8}$$

□

4.1.1 Implementation of the gradient flow method

In this section we describe the implementation of the gradient flow method that we have just derived.

In order to produce a numerical implementation we need to compute the values of the coefficients a_k. This requires discrete approximations in space and time of the continuous functions described in equations (4.6) and (4.8), which can then be used to numerically solve the differential equations. Equation (4.8) represents a system of ordinary differential equations, independently for each a_k, with some (currently unspecified) initial conditions. For the numerical integration we have chosen to use Euler's method, since it provides a simple first-order scheme that is easy to code and computationally cheap. Given that the algorithm uses finite differences in the image, a higher order integration method is unlikely to improve the results.

We will use the same discrete domain $S \in \mathbb{C}$ as was used in the previous chapter (based on N^2 pixels in a square grid), writing $z_{ij} = x_{ij} + \mathrm{i}y_{ij}$ to represent the value at pixel (i, j). The discrete form of Equation (4.8) for the K terms of the finite Taylor series $\varphi^{-1}(z_{ij}) = \sum_{k=0}^{K} a_k z_{ij}^k$ is:

$$\frac{\Delta a_k}{\Delta t} = \sum_{i=1}^{N} \sum_{j=1}^{N} \{(I_1 \circ \varphi^{-1})(z_{ij}) - I_2(z_{ij})\} (\nabla I_1 \circ \varphi^{-1}(z_{ij})) \overline{z^k} \quad k = 0, 1, 2, \ldots, K. \tag{4.9}$$

The discrete approximations to the gradient operator ∇I_1 are based on the same finite difference approximations in x and y as we described in Section 3.4, combined using:

$$\nabla I_1 = \Delta_x I_{1ij} + \mathrm{i}\Delta_y I_{1ij}, \quad i, j = 1, 2, \ldots, N.$$

It is necessary to choose an appropriate number of terms in the Taylor series expansion of φ^{-1}. While a particular number that worked for all examples would be nice, in practice it clearly depends upon the images that are being registered. For example, if two images of circles are being registered then there must be a rigid registration between them, and so two terms would be sufficient. For non-rigid conformal image registration at least three terms are required. As the optimisation works sequentially on increasing values of the coefficients k, it is possible to add further terms to the Taylor series if the registration is not sufficient with the initial choice of K. We present experiments showing the effect of increasing the number of coefficients later in this chapter.

At each iteration the values of each of the a_k are updated using Euler's method to give an approximate solution to Equation (4.9). Thus a new φ^{-1}_{new} is constructed, and we then use bi-linear interpolation to obtain a transformed source $I_1 \circ \varphi^{-1}_{new}(z_{ij}) : \forall z_{ij} \in S$. This transformed source is used in the computation of the error function (which is the sum-of-squares value of the difference between the transformed source and the target).

We also need to define the starting point of the registration as part of the initialisation. In the previous chapter we saw that for some pairs of images the identity appears to be a local minimum, and so it is not always a good choice of starting point. In that chapter we then made a small perturbation to the identity and used that as the new initial deformation. However, this was not always sufficient, and we thus applied a *Gaussian filter* with standard deviation five pixels to smooth the image. An initial registration using the smoothed images, and starting from close to the identity, significantly improved the performance of the registration algorithm in that chapter. We therefore use the same method here: a Gaussian smoothing kernel with standard deviation five is applied to both images, and a small perturbation of the identity warp used as the initial starting point. A second registration using the original (non-smoothed) images starts from the stopping point of the first registration.

The process of generating new coefficients a_k is performed until a local minimum is found. As was mentioned previously, for a given pair of images there is no guarantee that there is a (conformal) diffeomorphism that matches the two images, and thus there is no guarantee that the algorithm will ever converge. We have chosen to stop the algorithm when the decrease in the error function drops below a preset tolerance, although we also include a maximum integration time as well. The final version of the transformed source is the one represented by the last timestep of the optimisation.

The computation of the gradient flow is shown in Algorithm 3. For most of our examples, as in the previous chapter, the algorithm is actually used twice, initially based on smoothed images in order to find a reasonable initial guess, and then on the original images.

```
Algorithm 3: Optimisation by gradient flow
  input : $I_1$ and $I_2$: source and target images
          $K$: number of terms in Taylor series
          $\Delta t$: time step
          $T$: final integration time or $tol$ error tolerance
          Initial guess $\varphi_0(x)$
  output: Warp $\varphi^{-1}$ and deformed image $I_1 \circ \varphi^{-1}$
  $t = 0$
  while $E(t - m\Delta t) - E(t) > -m(\Delta t)tol$ and $t < T$ do
      $t = t + \Delta t$
      Use Taylor series to obtain transformed grid $\varphi^{-1}(x_{ij}), \forall x_{ij} \in S$
      Use bi-linear interpolation to obtain transformed version of the source,
       $I_1 \circ y_{ij}$
      for each coefficient $a_k$ do
          Use Euler's step to update each $a_k$ using Equation (4.9)
```

Throughout this thesis, $T = 20$, $\Delta t = 10^{-3}$ for smoothed images (step 1 of the registration of non-smooth images) and $T = 10$, $\Delta t = 10^{-5}$ for non-smooth images unless otherwise specified, and $tol = 0.3$. In practice it was found that the algorithm would sometime stop in the first few steps, since the error would not decrease very much near the start, and also the algorithm would sometimes increase the objective function for one or two steps; and thus we actually used $E(t-m\Delta t)-E(t) > -m(\Delta t)tol$ as the stopping criteria in order to avoid the problem for the first few iterations. Here m is a parameter; we found that $m = 9$ gave a reliable stopping criterion.

Note. When we apply Algorithm 3 to smooth and non-smooth images, we make sure that map is invertible throughout the integration (by computing the area of each cell on the discrete domain S) because invertibility is an esssential requirement for diffeomorphic mapping. If the area of any cell is negative, our algorithm will stop before the convergence. This non-invertibility did not happen to any example in this chapter, but in Examples 5.4 and 5.5 of Chapter 5, it did sometimes happen.

4.2 Image registration with smooth images

As in previous chapters, we first test the algorithm using smooth images, since these present the fewest problems. We start with the simplest case, where a smooth target image is generated from the source using a known conformal transformation, and then consider the case of two independent smooth images, although still related by a conformal transformation.

Example 4.1. We use the smooth image of an ellipse as the first example of this section. The particular ellipse used is generated with intensity values $I(x,y) = \exp(-7x^2 - 2y^2)$ (for $x, y \in S$ where, as in the previous chapter, S is a square grid of 100 points running between -0.5 and 0.5 in both x and y). This smooth image serves as the source for image registration. The target was generated from the source by applying the conformal transformation $\varphi^{-1} = \Sigma_{k=0}^{2} a_k z^k$ with $a_0 = 0.3, a_1 = 0.8, a_2 = 0.4$ to the pixels, where these values were chosen to be reasonably small values that generate a transformation that is distinctly non-linear. The target was reinterpolated onto a pixel grid using bilinear interpolation. The two ellipses can be seen in Figure 4.1.

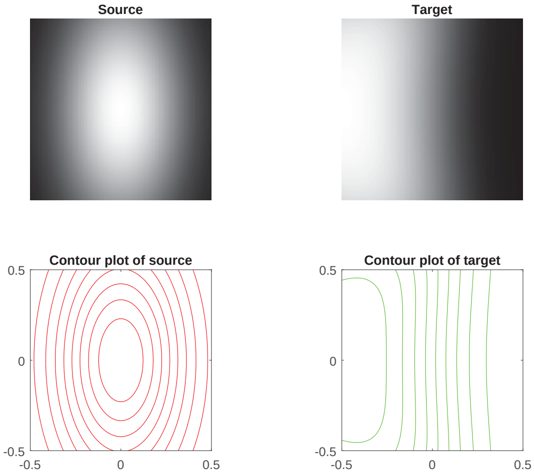


Figure 4.1: *Top:* The source, the ellipse $I(x,y) = \exp(-7x^2 - 2y^2)$ is shown on the left, while the right shows the target, which is generated from the source using $\varphi^{-1} = \Sigma_{k=0}^{2} a_k z^k$ using coefficients $a_0 = 0.3, a_1 = 0.8, a_2 = 0.4$. *Bottom:* The corresponding contour plots.

The results of using Algorithm 3 (without applying any smoothing to the images first, and with $K = 3$) to register these images is shown in Figure 4.2.

These images are perfectly registered by Algorithm 3 as can be seen from the mid-grey colour of the difference image in Figure 4.2 (bottom row, second image from the left). In this plot, the black regions of the image mark those points where the background is pulled in to the image, and so there is no colour information available. The contour plots on the right of the figure show that the contours of the transformed source are perfectly aligned with those of the target. The coefficients of the Taylor series that are obtained by the registration were $[0.3000 + 0.0000\mathrm{i}, 0.8001 - 0.0000\mathrm{i}, 0.4001 + 0.0000\mathrm{i}]$, which differ by only 10^{-4} in the second and third coefficients from those that were used to generate the target.

The Frobenius norm (denoted by $\|.\|$) between the transformed source and the target for each step of integration is plotted in Figure 4.3. This shows that the algorithm converges well on this simple example.

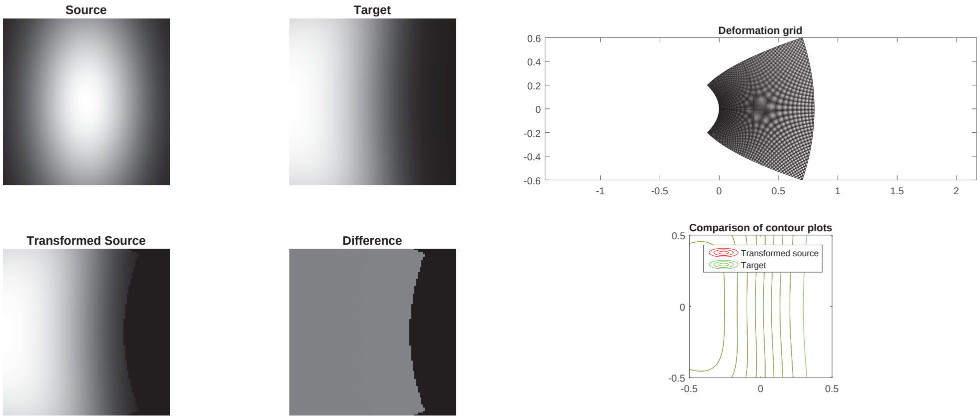


Figure 4.2: Registration with target generated from smooth source. *Left:* The top row shows the source and target, while the transformed source and the difference between the transformed source and the target can be seen in the second row. *Right:* A deformation grid $\varphi^{-1}(z_{ij}), \forall z_{ij} \in S$ is shown on top and the corresponding contour plots of the transformed source and the target are on the bottom.

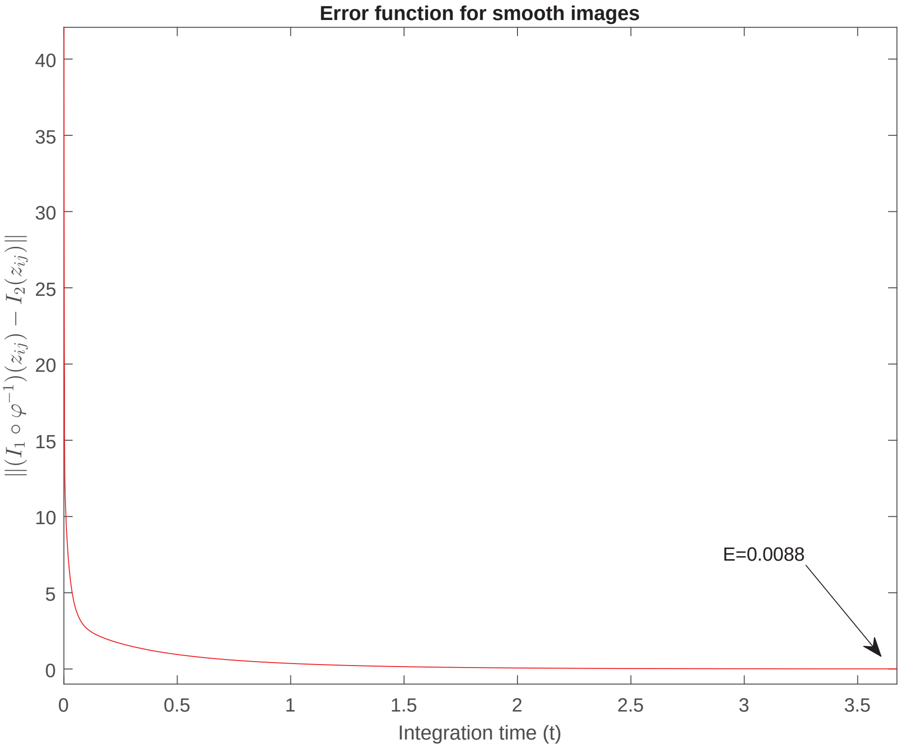


Figure 4.3: The error graph shows the successful convergence of Algorithm 3 (with $K = 3$) for Example 4.1.

Example 4.2. The previous example demonstrated that the algorithm is able to successfully register images that are related. As a slightly harder test, we next generated two independent images of ellipses. The images of these ellipses and their corresponding contour plots are shown in Figure 4.4, while Figure 4.5 shows the results of image registration when $K = 4$ is used in Algorithm 3.

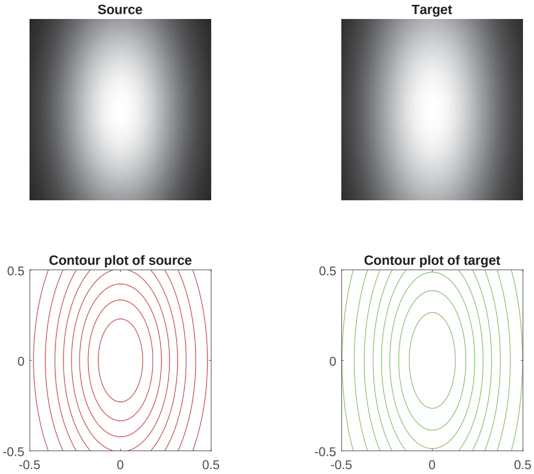


Figure 4.4: *Top:* Two ellipses for Example 4.2, the source $I_1 = \exp(-7x^2 - 2y^2)$, and the target $I_2 = \exp(-6.5x^2 - 1.5y^2)$. *Bottom:* The corresponding contour plots of the source and the target images.

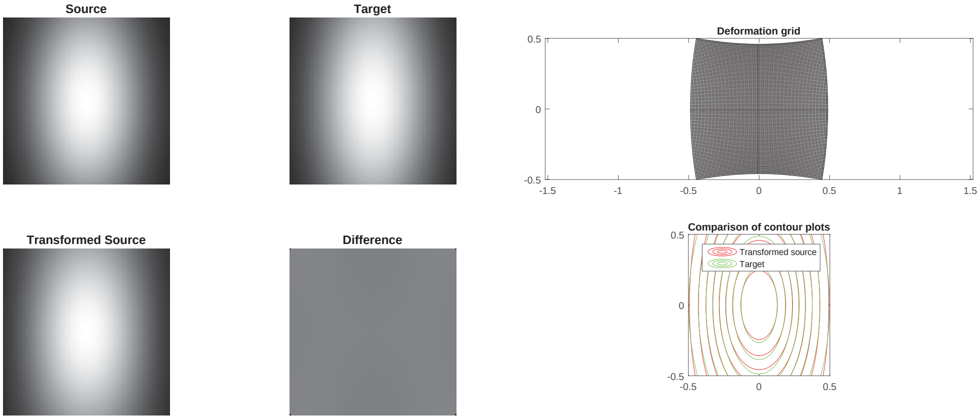


Figure 4.5: Registration with independent smooth images for Example 4.2. *Left:* The top row shows the source and target images, while the bottom row shows the transformed source and the difference image. *Right:* The deformation grid $\varphi^{-1}(z_{ij}), \forall z_{ij} \in S$ on top and the corresponding contour plots on the bottom.

It can be seen in Figure 4.5 that in this registration Algorithm 3 has found a good local minimum. Further, the contour plots between the transformed source and the target are nearly perfectly aligned. It can be seen in Figure 4.6 that a local minimum has been found successfully, and in rather less time than for Example 4.1.

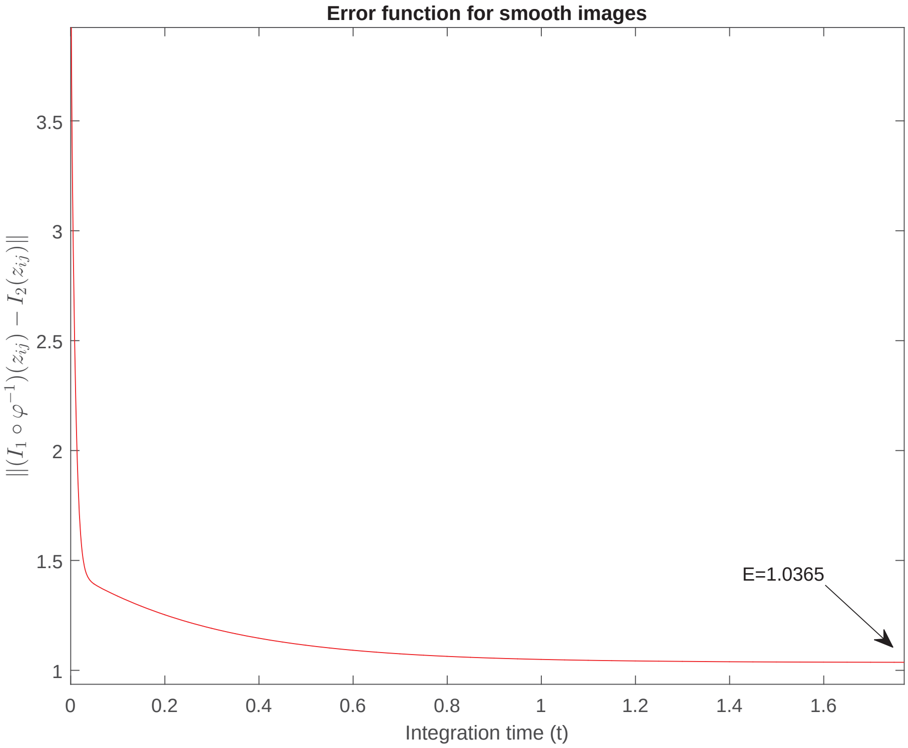


Figure 4.6: The algorithm successfully found a local minimum in Example 4.2. Note that the y-axis on this plot does not go down to 0, but stops at 1. Axis extrema are selected to make the plot as clear as possible for all of these error plots.

Example 4.3. We now take another pair of independent smooth images. In this example, the source image is the same as in Examples 4.1 and 4.2, but the target image is different. Both images are given in Figure 4.7.

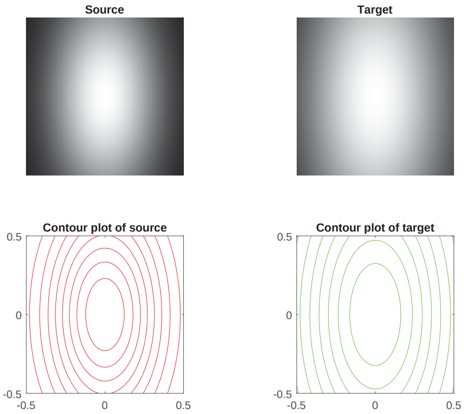


Figure 4.7: *Top:* Two ellipses, the source $I_1 = \exp(-7x^2 - 2y^2)$, and the target $I_2 = \exp(-4x^2 - y^2)$ respectively for Example 4.3. *Bottom:* The corresponding contour plots of the source and the target images for the same example.

This example is slightly harder than the previous example because the target is very far from the source. We use this example to investigate the convergence of Algorithm 3 with varying numbers of terms in the Taylor series. The results of registration with $K = 4, 8, 12$ are given in Figures 4.8, 4.9 and 4.10 respectively.

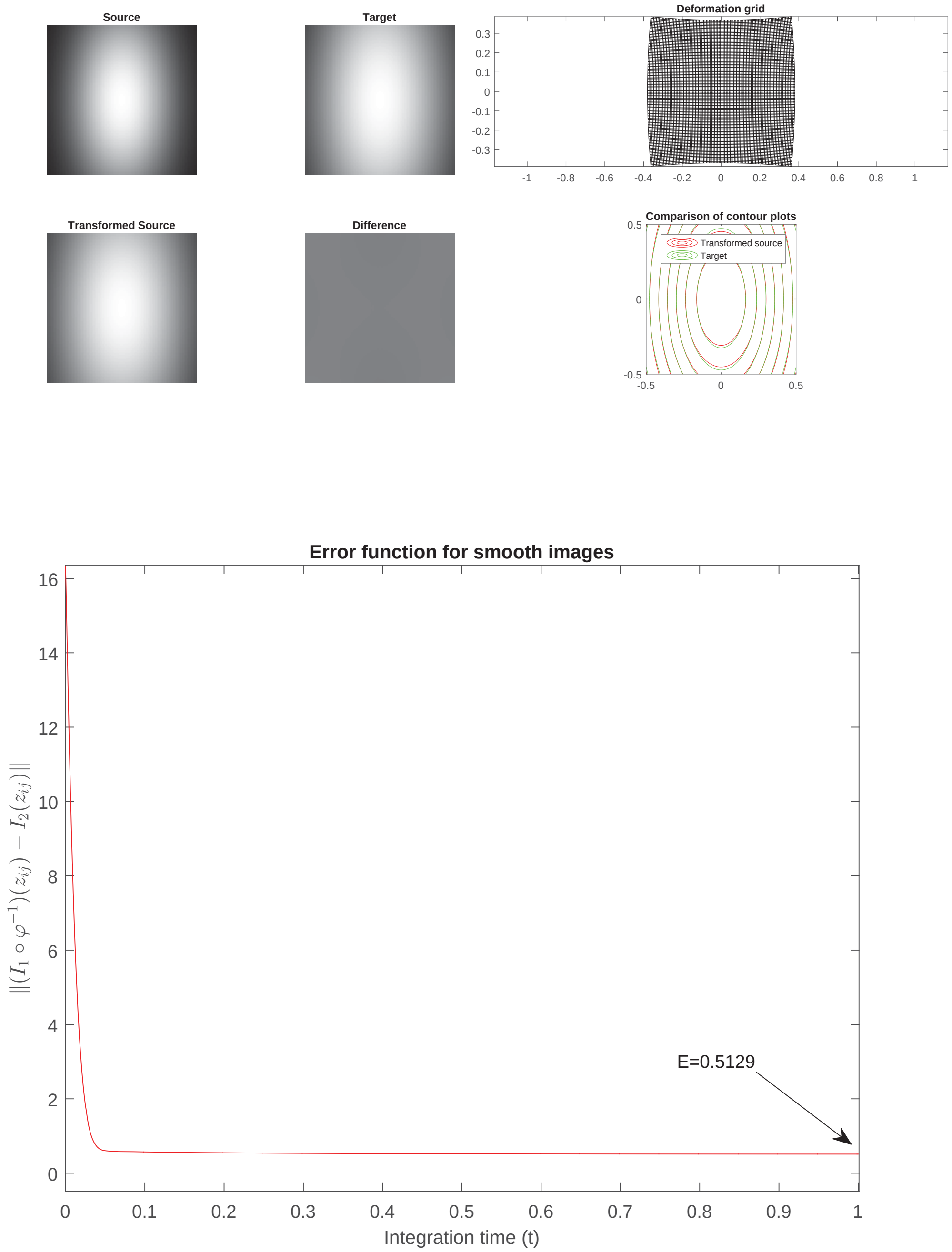


Figure 4.8: Registration results for Example 4.3 with four terms. Successful convergence with good registration is obtained.

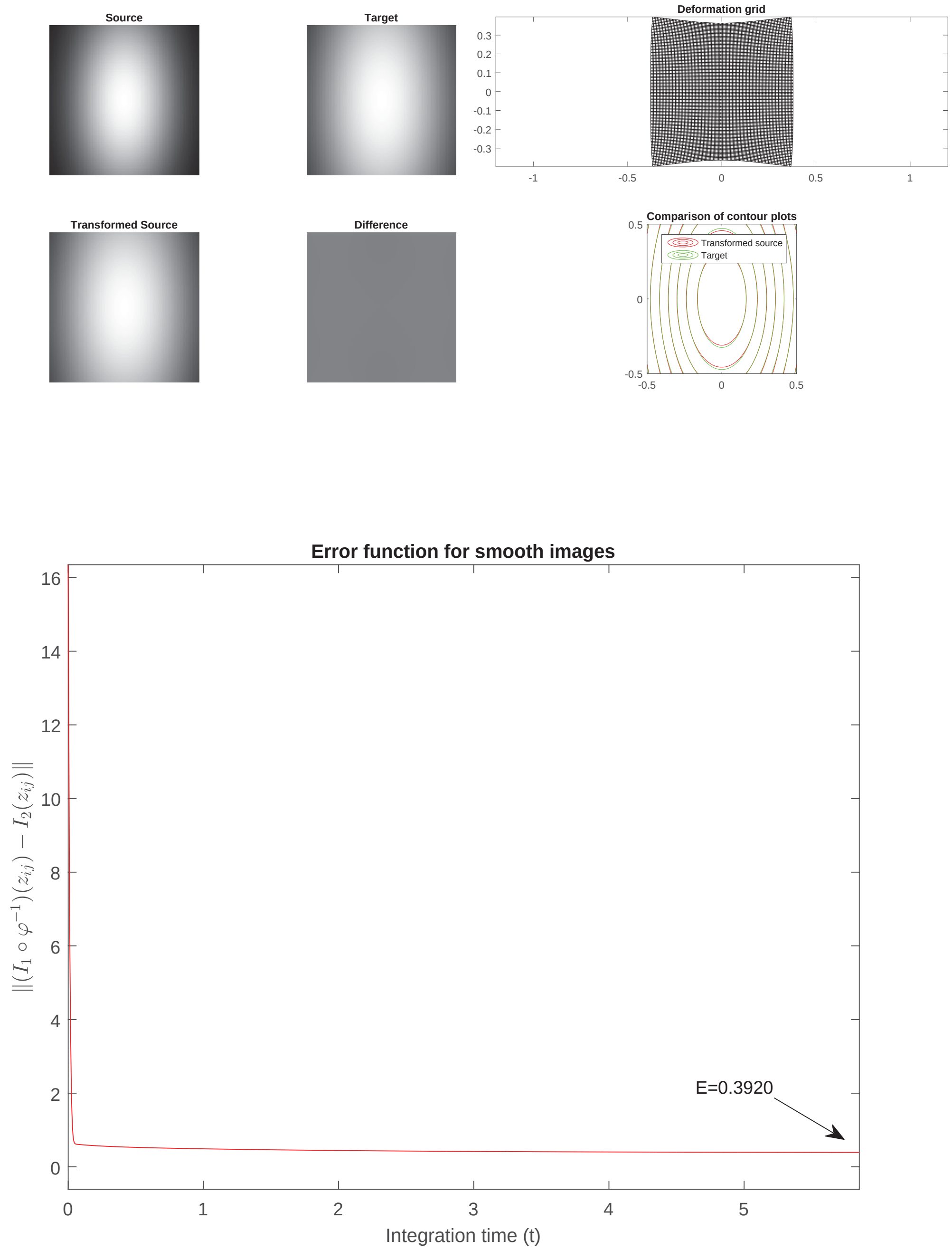


Figure 4.9: Registration results for Example 4.3 with eight terms. The final value of the objective function is rather lower than with four terms.

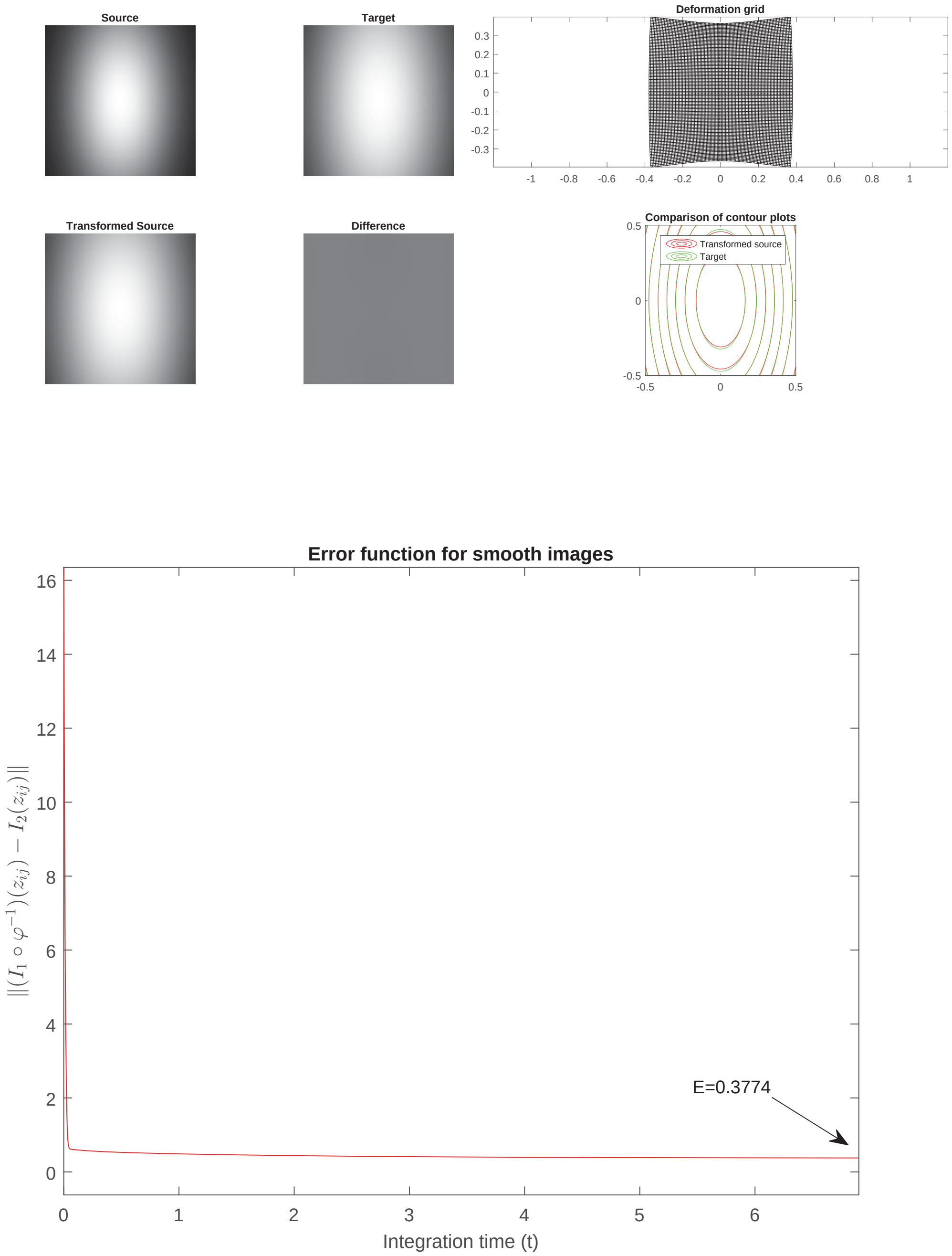


Figure 4.10: Registration results for Example 4.3 with twelve terms. The final value of the objective function is not much better than with eight terms.

In all the attempts the algorithm converged successfully and produced excellent registration. Errors in registration for four, eight and twelve terms are 0.5129, 0.3920, 0.3774 respectively.

4.3 Image registration with non-smooth images

The previous examples have demonstrated that our algorithm can successfully register smooth images that are related by a conformal map. We now test the gradient flow method of conformal image registration on non-smooth images, presenting three examples of such registrations. In the first example, the target is generated from the source in a similar manner to Example 4.1, i.e., by applying a known conformal transformation made with a Taylor series with a small number of terms. The second and third examples involve independent source and target that are conformally related. As was described earlier, we use the same approach as in Chapter 3, first smoothing the images with a Gaussian filter in order to find a good initial estimate of the deformation, and then using this initial estimate to initialise the deformation for the original (non-smoothed) images. The stopping criteria does not change between the steps.

Example 4.4. In this Example, we again use the photo of a cameraman as the source and generate the target by applying the conformal transformation $\varphi^{-1} = \Sigma_{k=0}^{2} a_k z^k$ with $a_0 = 0.1\mathrm{i}, a_1 = 1.5, a_2 = 0.2$. The images are presented in Figure 4.11.

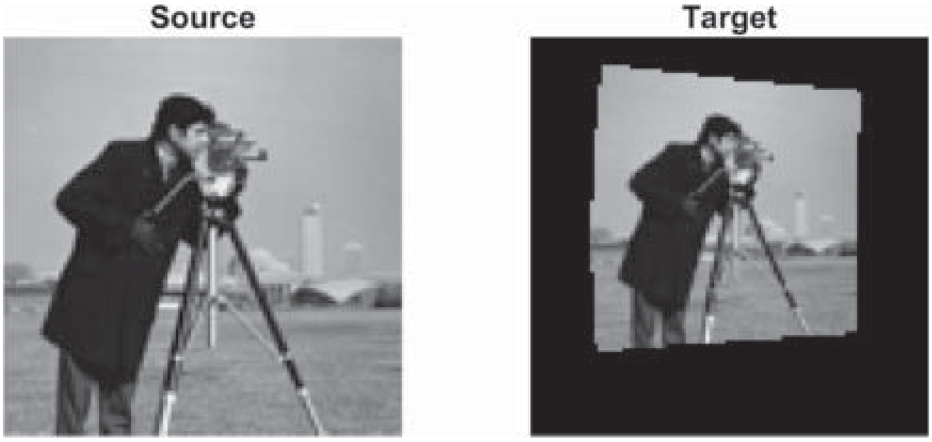


Figure 4.11: Set of images for Example 4.4. *Left:* The source image. *Right:* The target image, which is generated from the source using $\varphi^{-1} = \Sigma_{k=0}^{2} a_k z^k$ with coefficients $a_0 = 0.1\mathrm{i}, a_1 = 1.5, a_2 = 0.2$.

The results of registration using the two step (smoothed and then non-smoothed for $K = 3$) version of Algorithm 3 are shown in Figure 4.12.

The coefficients of the Taylor series that are obtained due to this registration are $[-0.0+0.1\mathrm{i}, 1.50+0.0\mathrm{i}, 0.20+0.0\mathrm{i}]$, which are exactly the same as were used to generate the target. Further, when the transformed source is subtracted from the target, the mid-grey difference image indicates excellent registration. The value of the error function (see Figure 4.13) at the global minimum is 3.2463×10^{-4}.

We used this Example to investigate the effect of the tolerance parameter in the

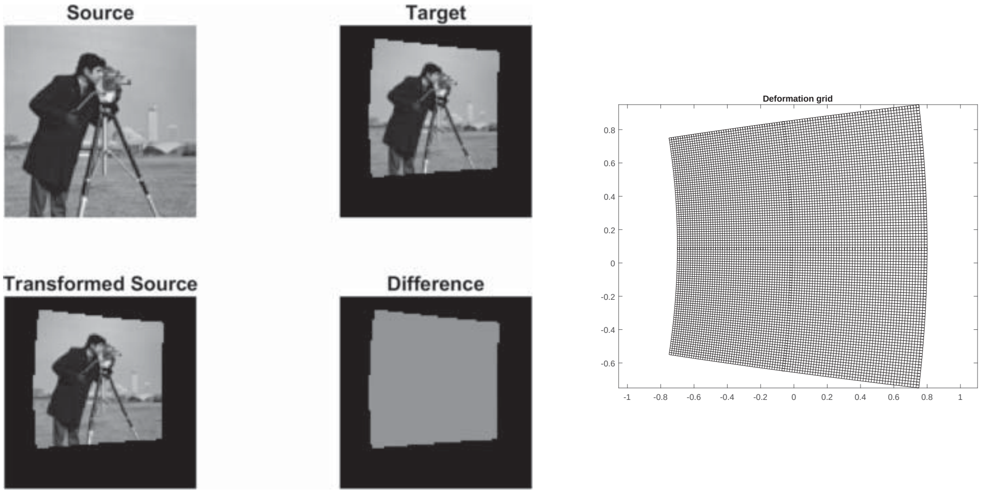


Figure 4.12: Registration with dependent, non-smooth images for Example 4.4 with $K = 3$. *Left:* The top row shows the source and target images, while the bottom row shows the transformed source and the difference image. *Right:* The deformation grid $\varphi^{-1}(z_{ij}), \forall z_{ij} \in S$.

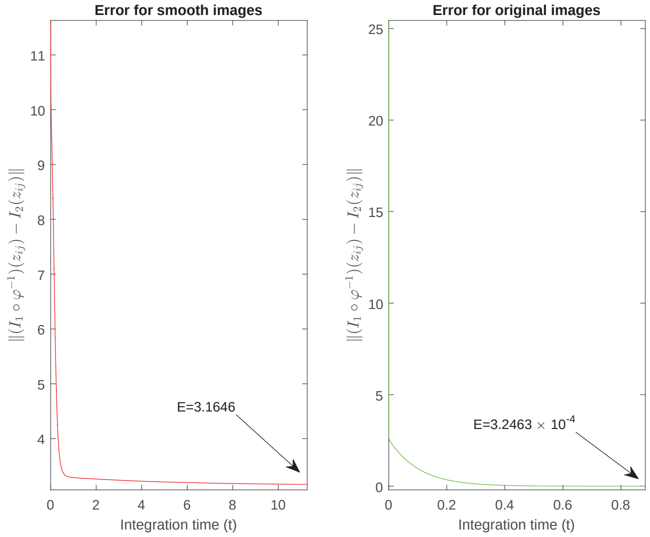
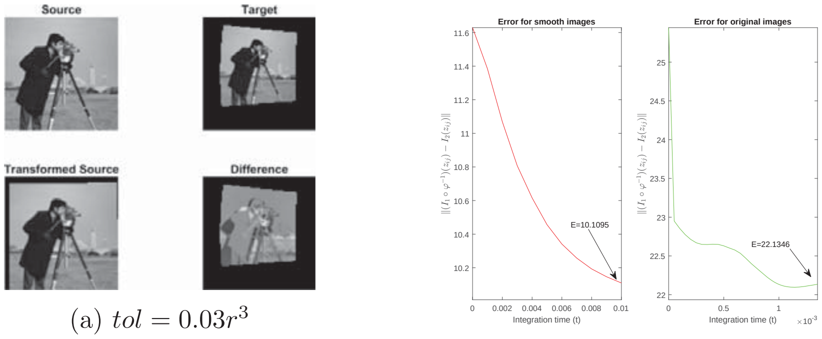
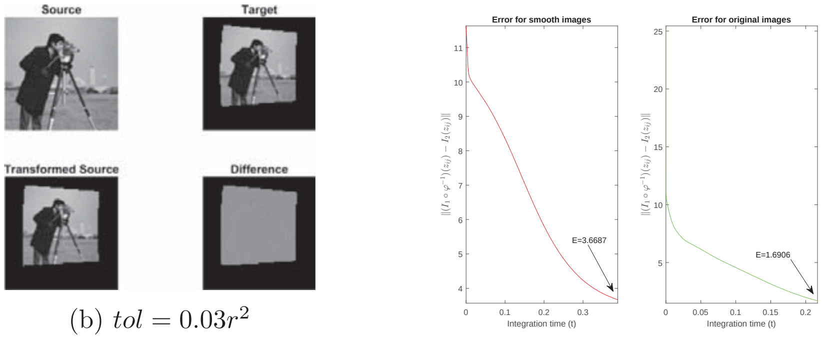


Figure 4.13: Plots of the error function for Example 4.4 with three terms of the Taylor series. *Left:* Smoothed images (first step). *Right:* Original images (second step).

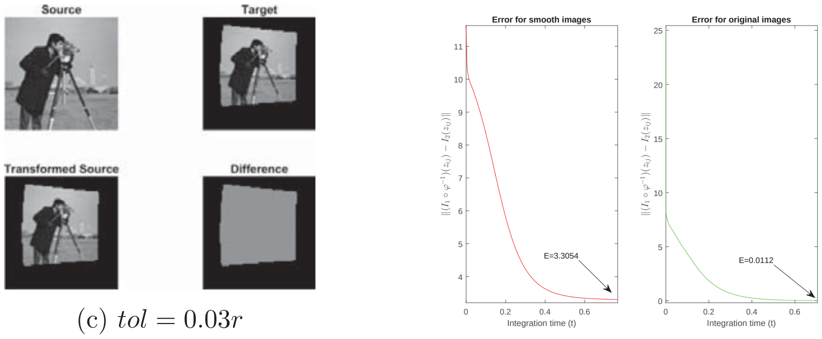
stopping criterion. In Figure 4.14 we compare the results of the registration when four different values for the tolerance in the stopping criterion are used, namely $tol = 0.03r^3, 0.03r^2, 0.03r$, and 0.03, where $r = 20\sqrt{3}$. It can be seen that with values of $tol = 0.03r$ and below the registration produces a difference image where no residual can be seen. This experiment is the reason that we have chosen to use to use $tol = 0.03$ in this thesis.



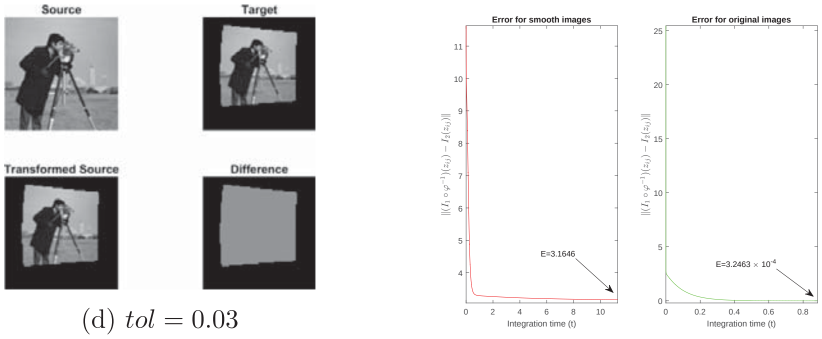
(a) $tol = 0.03r^3$



(b) $tol = 0.03r^2$



(c) $tol = 0.03r$



(d) $tol = 0.03$

Figure 4.14: Four sets of image registrations (along with their corresponding error graphs) for Example 4.4 are given. Algorithm 3 is run with four different stopping criteria (for $K = 3$ in each case) based on a common ratio ($r = 20\sqrt{3}$). Results (c) and (d) display the successful convergence of Algorithm 3 along with good registration, whereas set (a) has clearly stopped too early. Note that the runs lower down take many more steps, this is particularly clear for the registration of the smoothed image in (d).

Example 4.5. We now move on to some examples in which the source and target are independently generated, but conformally related, images. We consider two ellipses (of major axes 0.8 units, 0.6 units and minor axes 0.4 units, 0.4 units in the discrete domain S) as the source and target respectively. These images are shown in Figure 4.15.

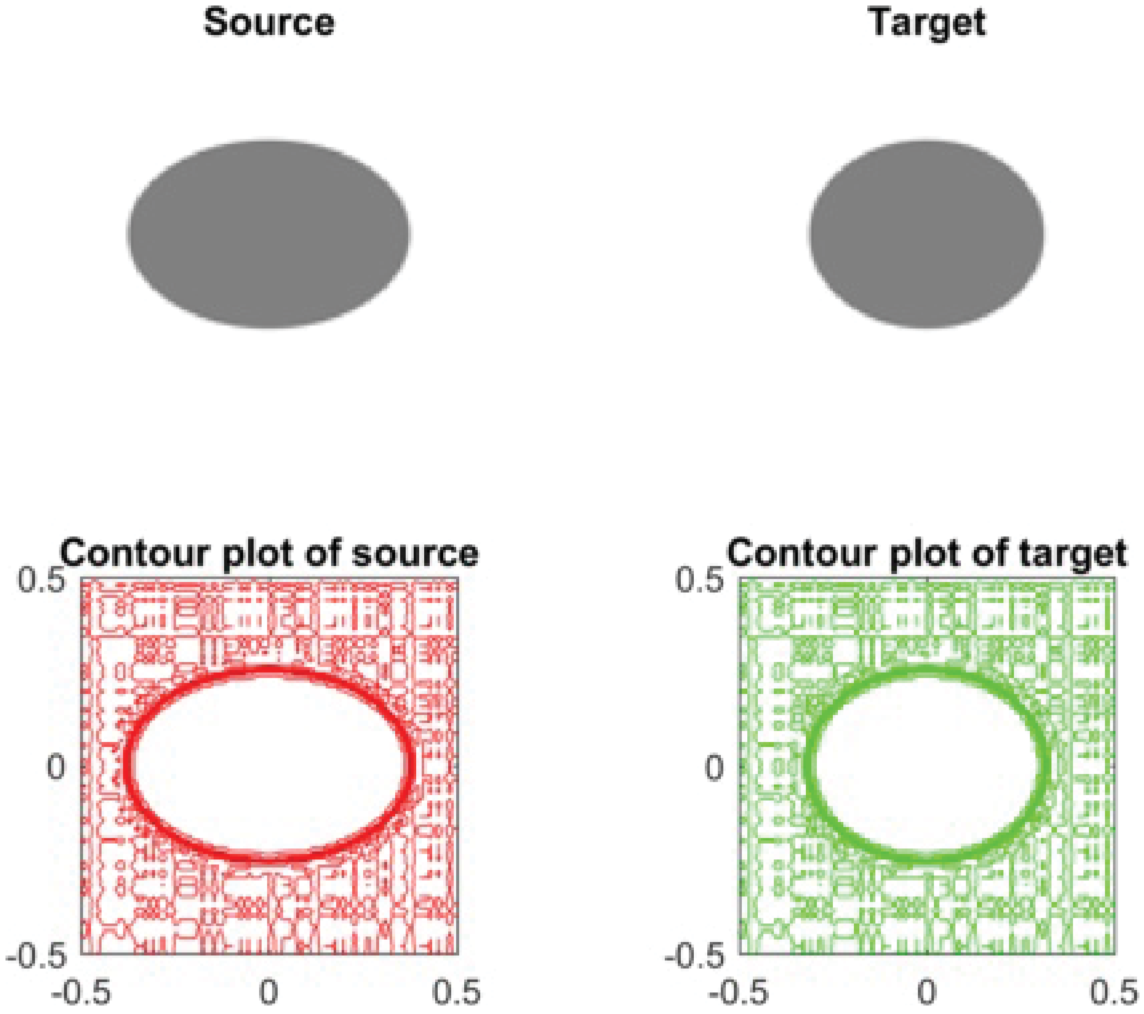


Figure 4.15: *Top:* An ellipse with axes 0.8 units and 0.4 units (on the left) serves as the source, while another ellipse of axes 0.6 units, 0.4 units (on the right) serves as the target. *Bottom:* Corresponding contour plots of the source and the target respectively. Note that the background is not perfectly white (which has caused the artefacts in the contour plot) because of the resizing and resampling of the images.

According to the Riemann mapping theorem the interior of two ellipses are related by a conformal transformation. Therefore this Example makes a good test for our algorithm. We demonstrate the use of Algorithm 3 on this pair of images with four, eight and twelve terms in the Taylor series; the results are shown in Figures 4.16, 4.17

and 4.18 respectively.

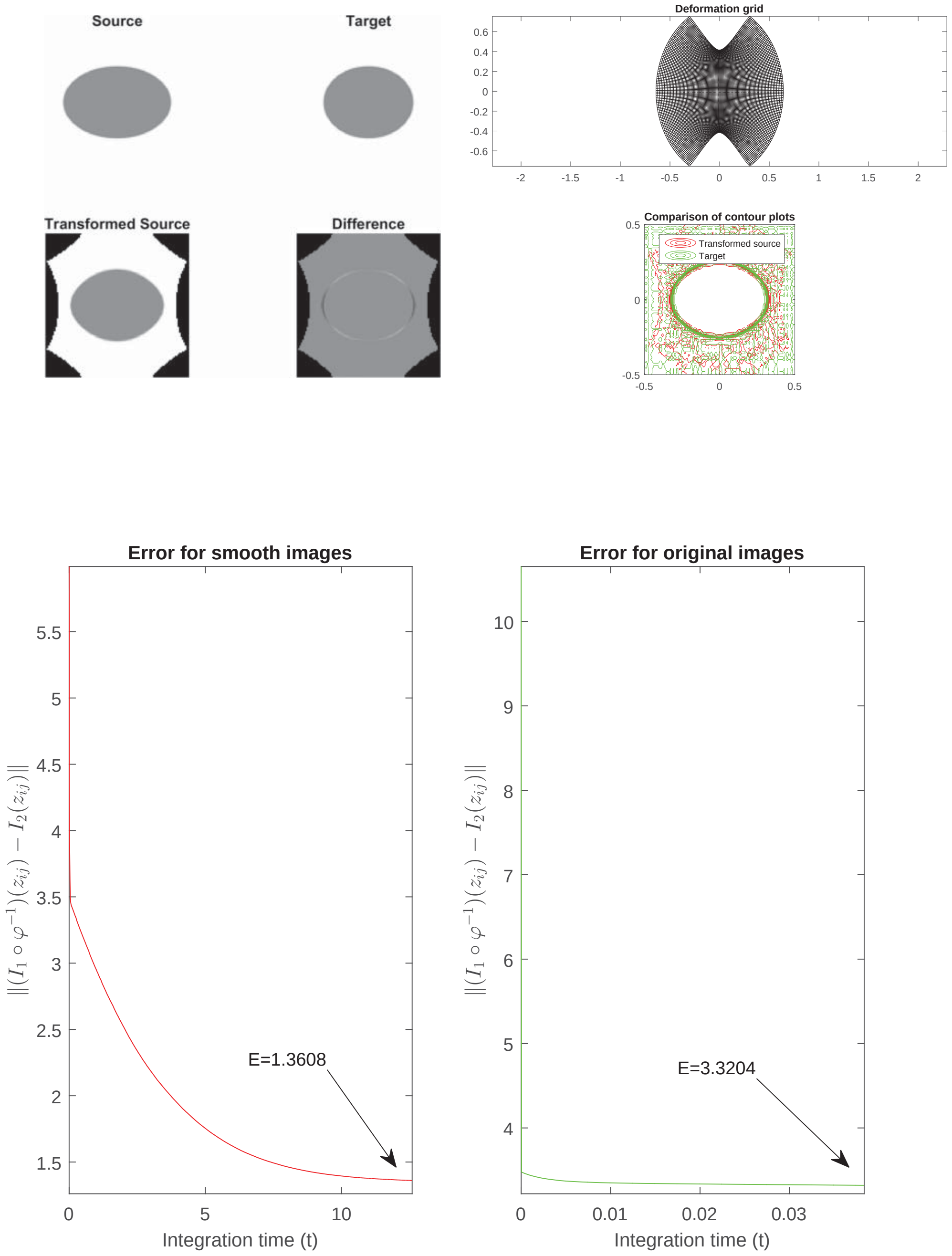


Figure 4.16: *Top:* Registration with four terms using conformal gradient flow on independent images related by a conformal transformation (Example 4.5). *Bottom:* Error plots for smoothed (left) and original (right) images.

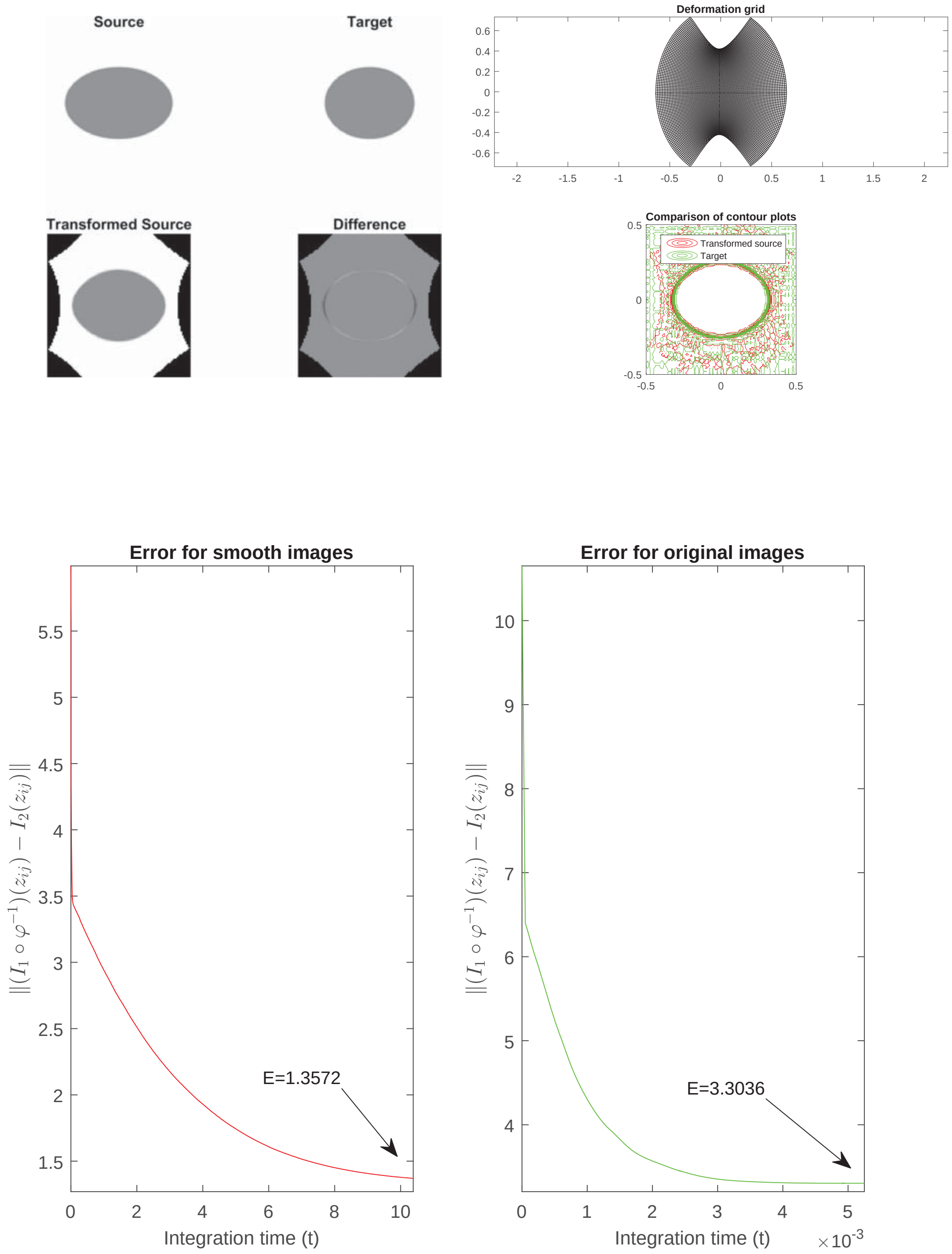


Figure 4.17: *Top:* Registration with eight terms using conformal gradient flow on independent images related by a conformal transformation (Example 4.5). *Bottom:* Error plots for smoothed (left) and original (right) images. The final error is little different to the run with four terms.

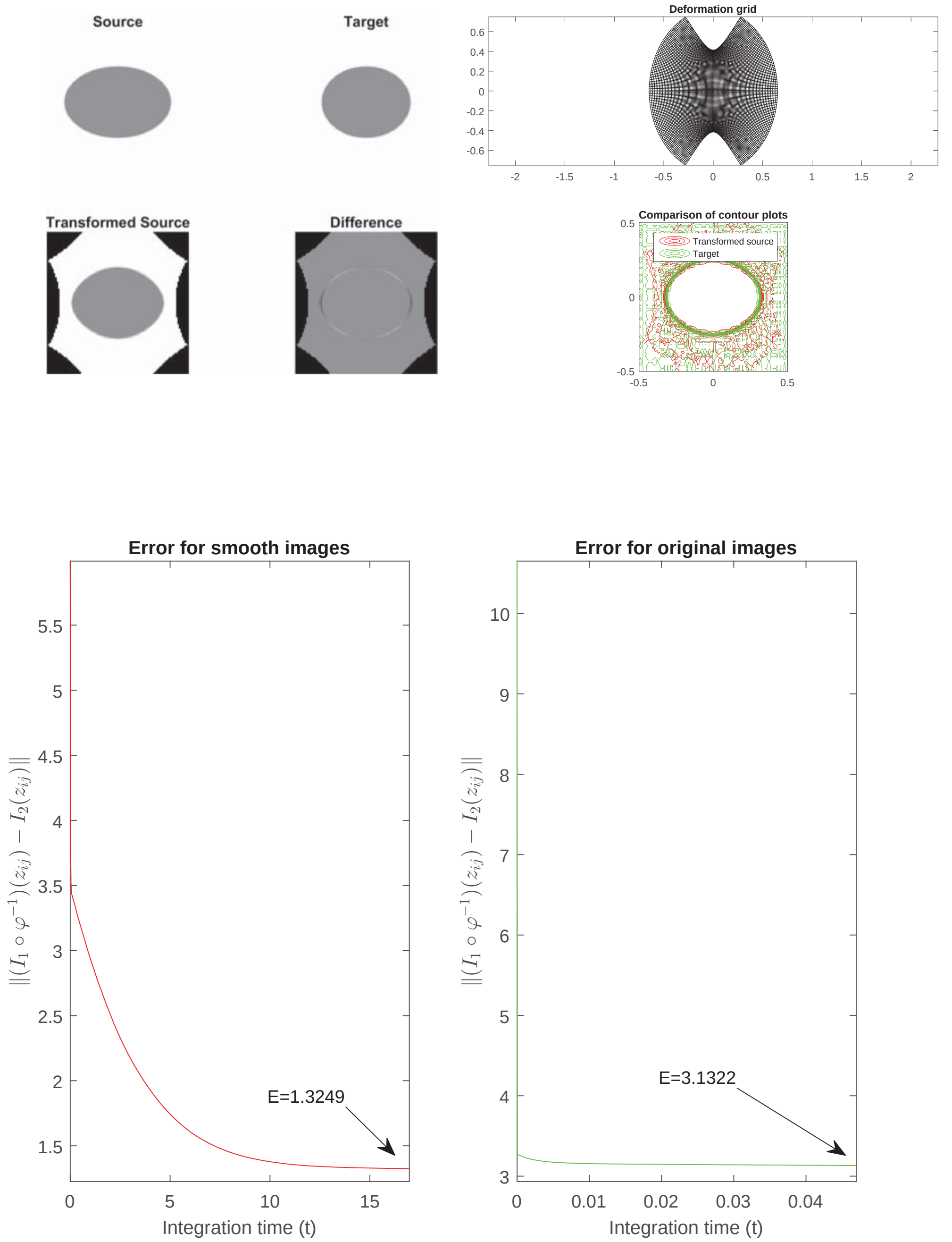


Figure 4.18: *Top:* Registration with twelve terms using conformal gradient flow on independent images related by a conformal transformation (Example 4.5). *Bottom:* Error plots for smoothed (left) and original (right) images. The final error is better than those with four or eight terms.

In all three attempts, Algorithm 3 converged successfully and good registration was obtained in each case. In this example, error in registration slightly improved at each iteration. The final registration errors with four, eight and twelve terms are 3.3204, 3.3036 and 3.1322 respectively.

Example 4.6. We now present the last example of this section, which is another simple variant of the two ellipses: we use the image of a circle as the source and an ellipse as the target. These images are given in Figure 4.19.

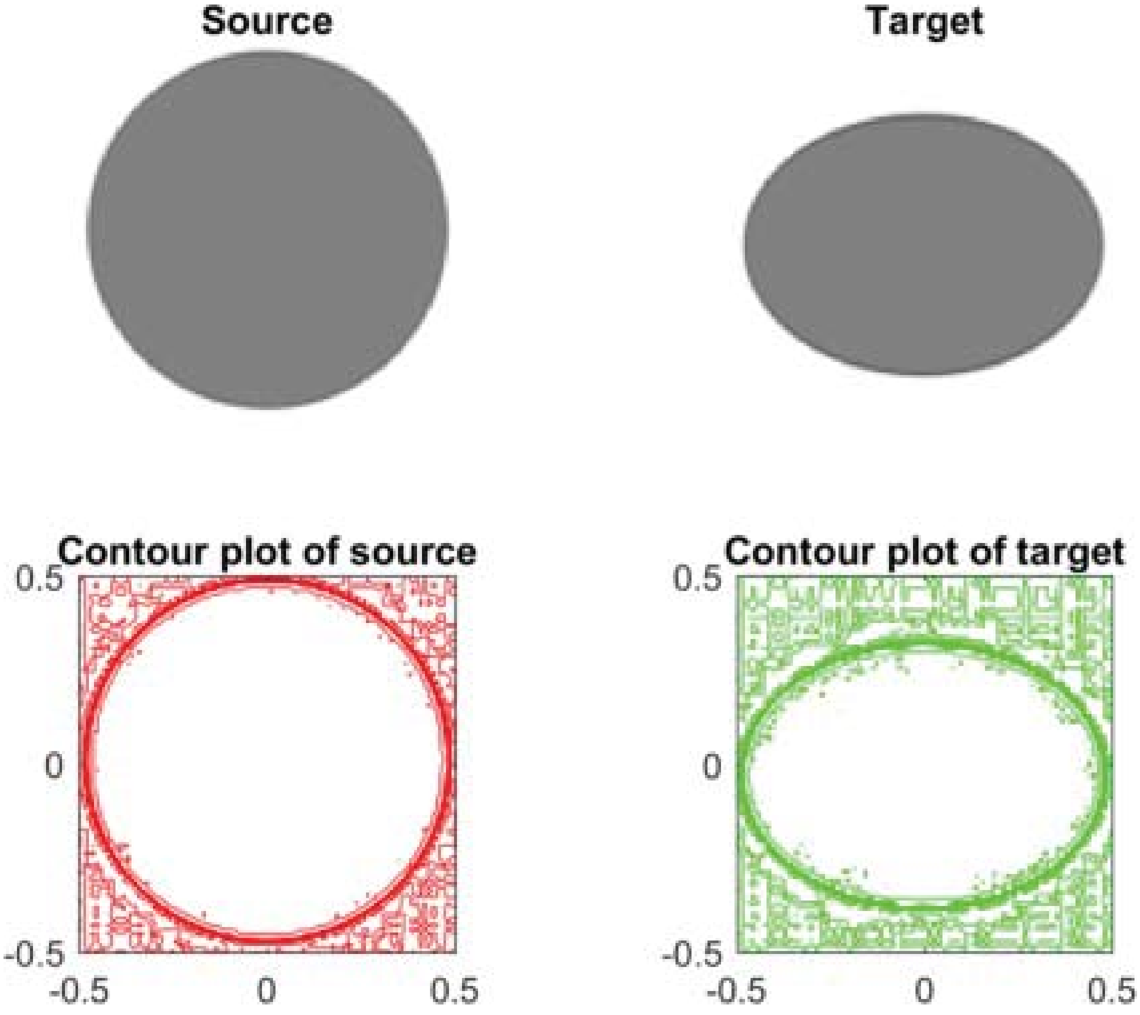


Figure 4.19: *Top:* A circle of radius 0.5 units serves as the source (on the left). The ellipse on the right has axes 0.5 units and 0.3 units and serves as the target. *Bottom:* Corresponding contour plots of the source and the target respectively.

As in the previous example, Algorithm 3 was run for four, eight and twelve terms in the Taylor expansion. The results of these registrations are given in Figures 4.20, 4.21 and 4.22 respectively.

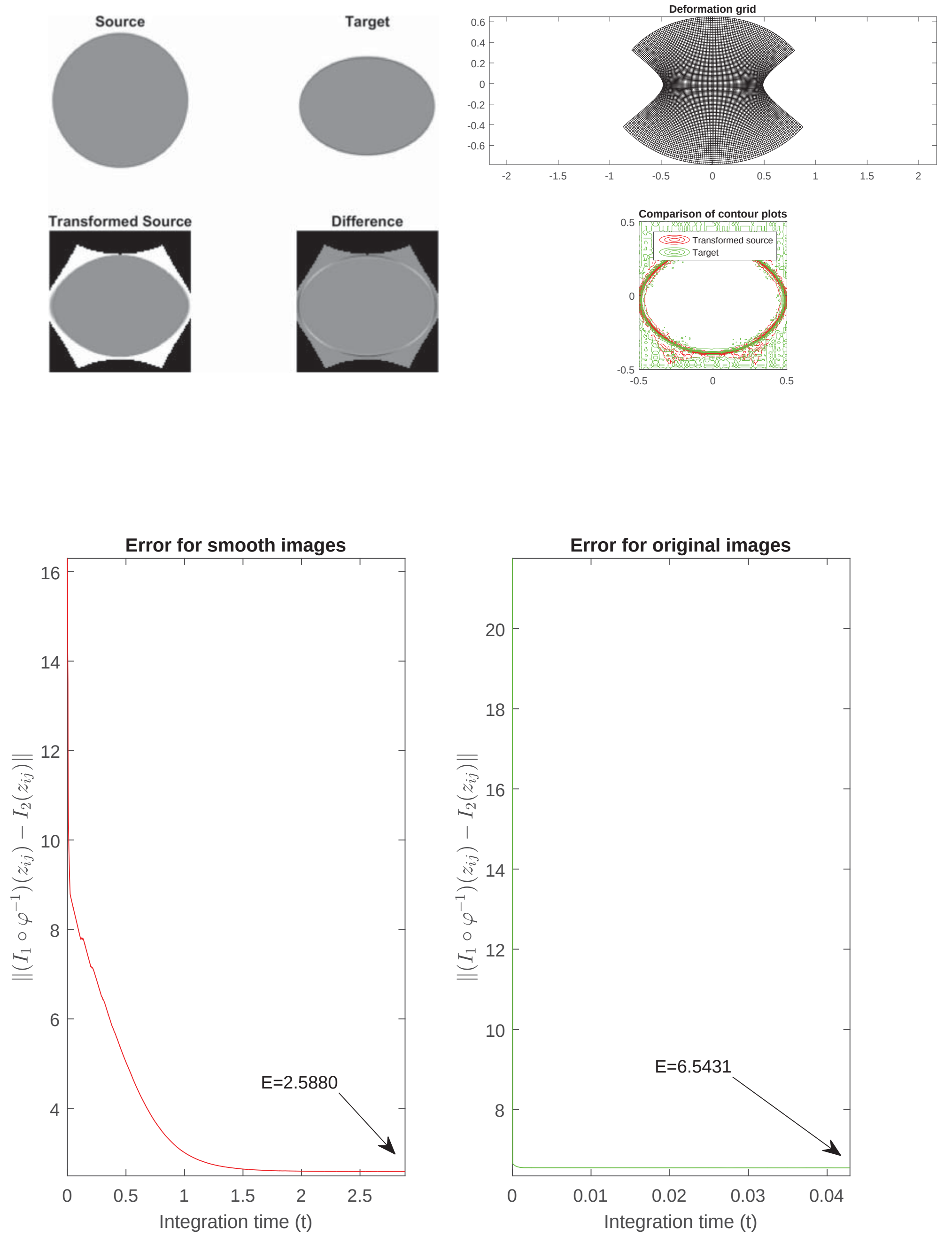


Figure 4.20: *Top:* Registration using conformal gradient flow with four terms in the Taylor series of images of a circle and ellipse. *Bottom:* Error graphs for smoothed (left) and non-smoothed (right) images.

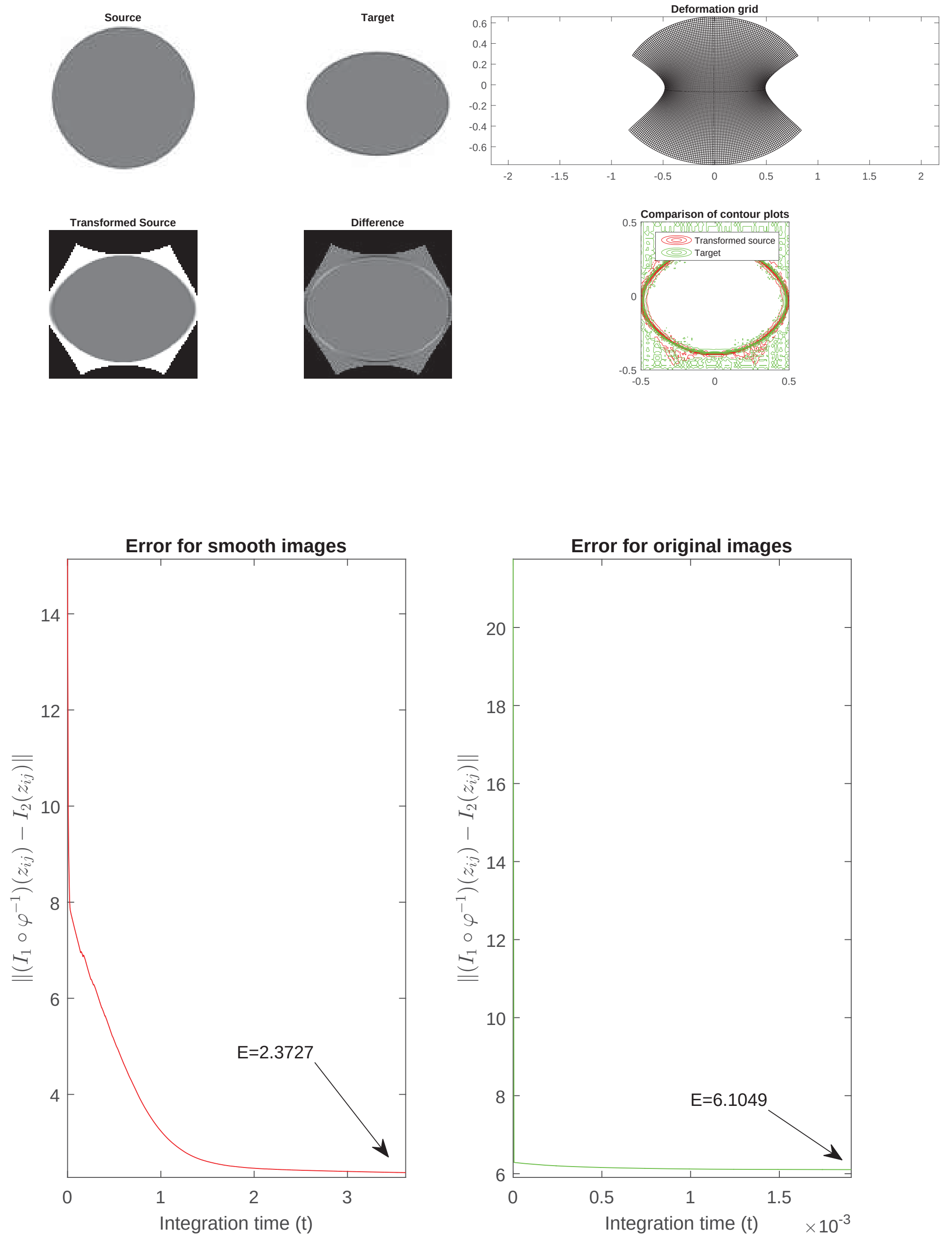


Figure 4.21: *Top:* Registration using conformal gradient flow with eight terms in the Taylor series of images of a circle and ellipse. *Bottom:* Error graphs for smoothed (left) and non-smoothed (right) images.

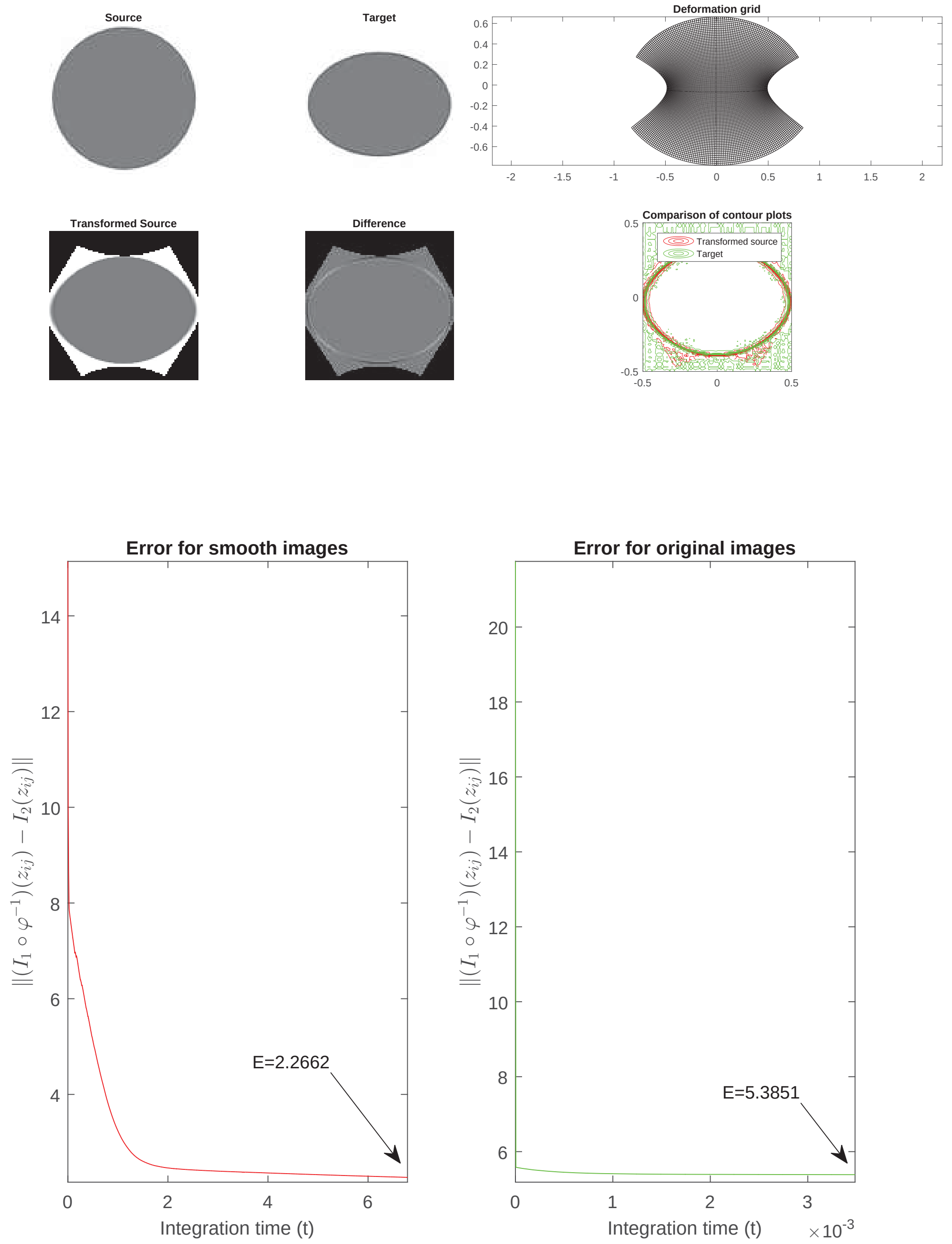


Figure 4.22: *Top:* Registration using conformal gradient flow with twelve terms in the Taylor series of images of a circle and ellipse. *Bottom:* Error graphs for smoothed (left) and non-smoothed (right) images.

Again, while all attempts produced successful convergence and reasonably good registrations, increasing the number of terms in the Taylor series improved the final error: with four, eight and twelve terms respectively, the final errors were 6.5431, 6.1049, 5.3851.

In this chapter, we have described an algorithm for image registration using conformal diffeomorphisms that are generated by a gradient flow algorithm. In order to test the algorithm, we have presented six examples. Three examples demonstrated the registration of smooth images, whereas three demonstrated the registration of non-smooth images. In Sections 4.2 and 4.3 we have presented two examples of registrations of pairs of independent images and one example using a pair of dependent images. The algorithm that we have described (Algorithm 3) works very well in either case. We have tested two parameters of our algorithm: the tolerance used in the stopping criterion for the gradient descent, and the number of terms used in the Taylor series expansion of the conformal diffeomorphism. With regard to this second parameter, we have seen that (admittedly, for fairly small values of the number of terms) the registration error is lower with more terms. It is not clear how long this trend would continue for, and we did not explore it further. Note, however, that if a small number of terms was judged to be insufficient for any particular registration, then more terms could be added and the registration could continue by setting the initial values of these new terms to 0, but retaining the current values of the original terms. This enables an iterative algorithm to identify a suitable number of terms in the Taylor series.

We now have two alternative algorithms for conformal image registration. The first is the control point method that was presented in the previous chapter and the second is the gradient flow in this chapter. In the next chapter, we provide the comparison of these algorithms on smooth and non-smooth images, including examples of registrations where there is not actually a conformal transformation between the images.

Chapter 5

Experiments and Comparison

The profound study of nature is the most fertile source of mathematical discoveries.

- Joseph Fourier (1768-1830)

In the previous two chapters, we have presented two alternative ways of constructing conformal diffeomorphisms, both of which we developed. The first is based on the method of *control points* and the second on *gradient flow*. In this chapter we compare these two methods experimentally on some more example pairs of images. We first present four examples of pairs of images that are hypothetically conformally related, and then we present four examples of pair of images that are unlikely to be conformally related. At the end of each example, we provide a comparison between the results found by the two methods. We have seen in Chapters 3 and 4 that the registration of non-smooth images is much more challenging than that of smooth images. Therefore, the majority of the examples consist of pairs of non-smooth images. We have used numerical smoothing (as in the previous two chapters) to obtain better initial guesses for the registration of non-smooth images.

Example 5.1 (Two circles). We start off with a relatively simple example; we consider two circles of different radii. The circles are related by a scaling, and scalings are part of the rigid group of transformations, which is a subgroup of the group of conformal transformations. The radii of these circles are 0.4 units and 0.2 units respectively in our discrete domain 'S'. These images are shown in Figure 5.1. The original images were very large and were resized to 100×100 with the Matlab command `imresize`; as has been mentioned previously, this resizing causes some texture to appear in the background of the images, as can be seen in the contour plots.

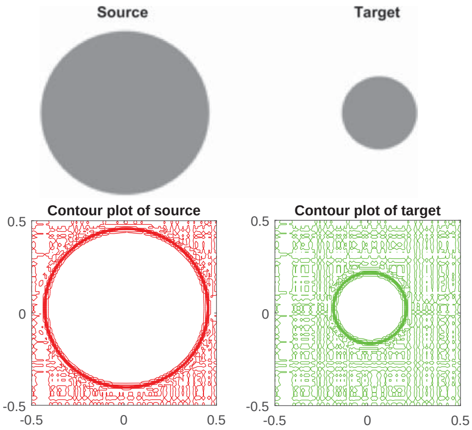
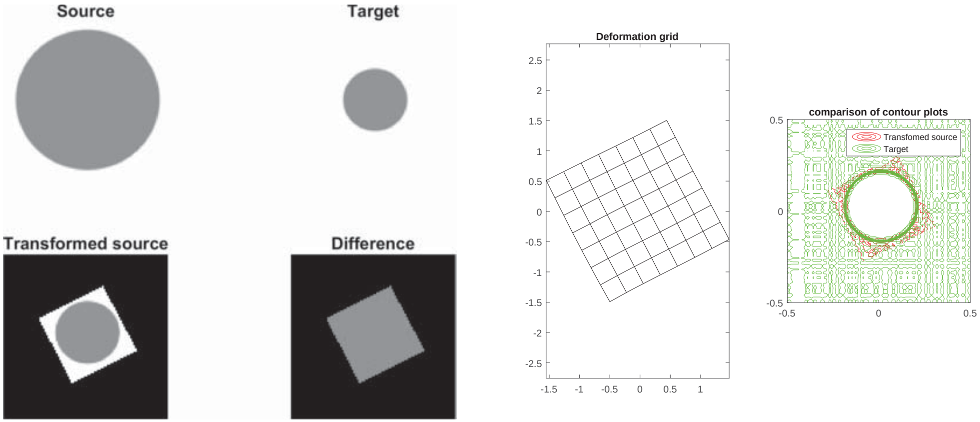


Figure 5.1: *Top:* The source and target for Example 5.1. *Bottom:* Corresponding contour plots.

First we apply Algorithm 2 of Chapter 3, which is based on the optimisation function given in Equation (3.3). We use the fourth penalty term with $\lambda = 150$. The results are shown in Figure 5.2.



n^2	λ	$\|\|(I_1 \circ \varphi^{-1})(\hat{x}_{ij}) - I_2(\hat{x}_{ij})\|\|$	$\|\|P_4\|\|$
16	150	5.8471	0.0050
64	150	4.6034	0.0162
64	150	1.0060	0.0020

Figure 5.2: *Left:* Results of image registration for Example 5.1 using the control points method with the 4th penalty term and $\lambda = 150$. Images are smoothed for first two rows and non-smooth for the last row. Note that the algorithm has found a spurious rotation of the circle as well as the correct scaling. There is nothing to penalise this rotation in the registration.

We now apply the gradient flow algorithm (Algorithm 3 of Chapter 4) for the registration of these two circles. As we know that there is a scaling transformation between these pair of images, we truncate the Taylor series at two terms. The image registration results are given in Figure 5.3.

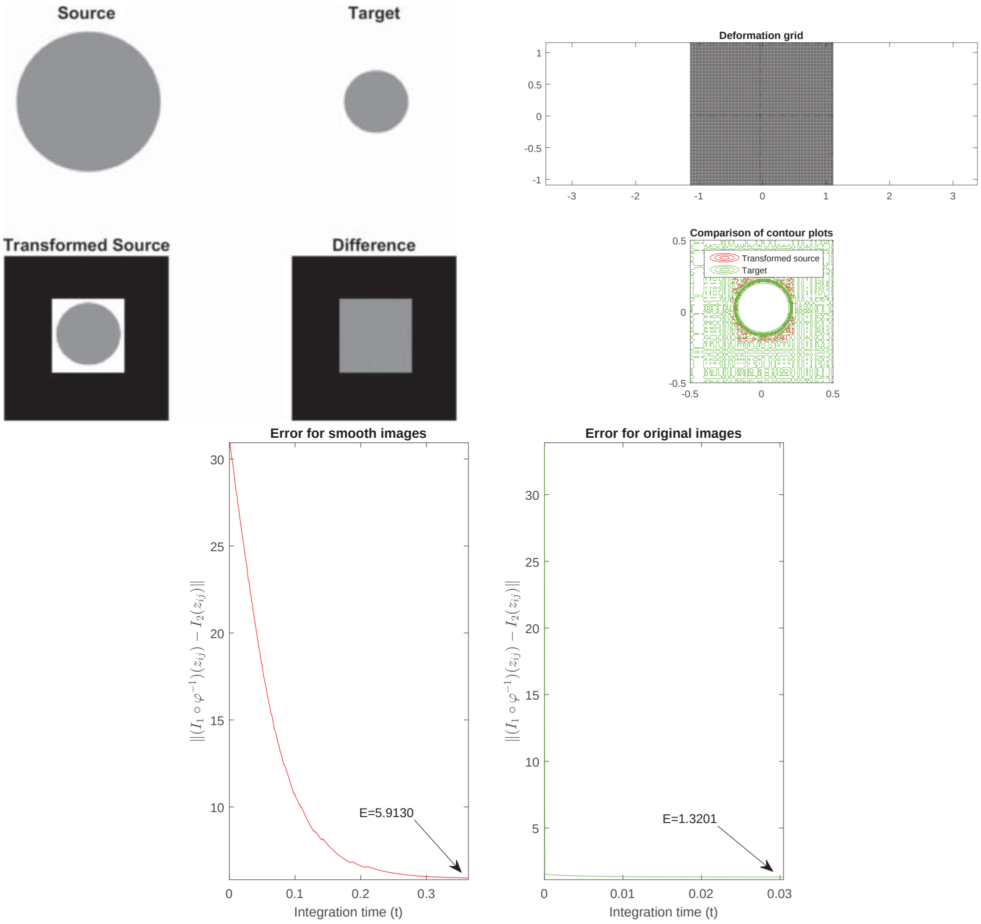


Figure 5.3: Results of image registration with the gradient flow method for Example 5.1. *Top left:* The source and the target images, the transformed source, and the difference image. *Top right:* The final mapping $\varphi^{-1}(x_{ij})$ and the contour plots of the transformed source and the target. *Bottom:* The progress of the gradient descent algorithm for the smoothed (left) and original (right) images.

Both methods find very good solutions: the final registration errors are 1.0060 for the control points and 1.3201 for the gradient flow. The most notable difference is that different mappings are obtained from each method. This is because of the rotational symmetry of the problem; apart from discretisation errors, any rotation of the grid yields an equally good solution. Conversely, tiny errors can lead to large rotations.

Example 5.2 (Smooth, non-conformally related images). In this example we consider two smooth images that are diffeomorphically, but not conformally, related. We take as the source the Gaussian ellipse $I_1(x, y) = \exp(-7x^2 - 2y^2)$, whose level sets are ellipses, and generate the target using $I_2 = I_1 \circ \varphi^{-1}$, where $\varphi^{-1}(z) = 0.1 + z + 0.2z^2 + 0.5\mathrm{i}\bar{z}^2$. While this mapping is a diffeomorphism, due to the $\bar{z}$ term it is not conformal. The images are shown in Figure 5.4.

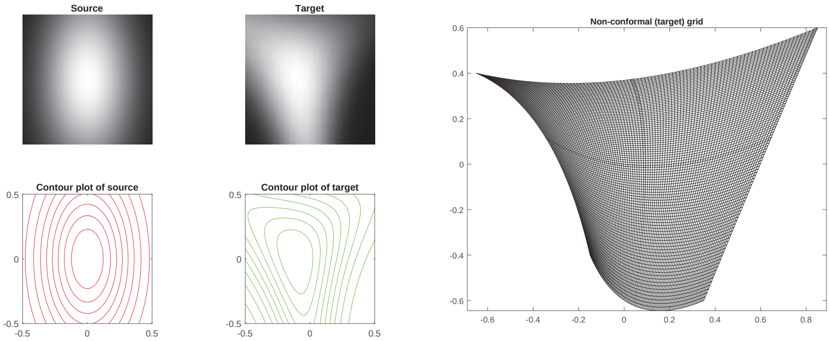
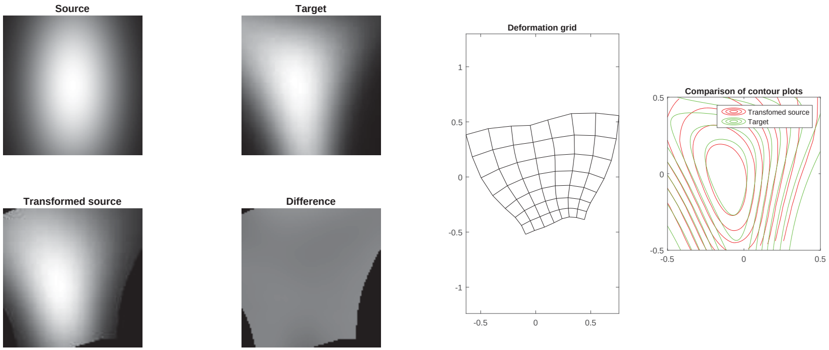


Figure 5.4: *Left:* The source and target images for Example 5.2 are shown in first row. The source is $I_1(x, y) = \exp(-7x^2 - 2y^2)$, and the target is defined by $I_2 = I_1 \circ \varphi^{-1}$, where φ^{-1}, where $\varphi^{-1}(z) = 0.1 + z + 0.2z^2 + 0.5i\bar{z}^2$. Corresponding contour plots are given in the second row. *Right:* The non-conformal mapping that generates the target.

The results of conformal registration using the control point method are given in Figure 5.5 and using the gradient flow method (with four terms) in Figures 5.6 and 5.7.



n^2	λ	$\|\|(I_1 \circ \varphi^{-1})(\hat{x}_{ij}) - I_2(\hat{x}_{ij})\|\|$	$\|\|P_4\|\|$
16	150	7.0706	0.0272
64	150	3.0862	0.0208

Figure 5.5: *Top:* Results of image matching for Example 5.2 with the control points method. *Bottom:* Numerical results of the registration for $\lambda = 150$.

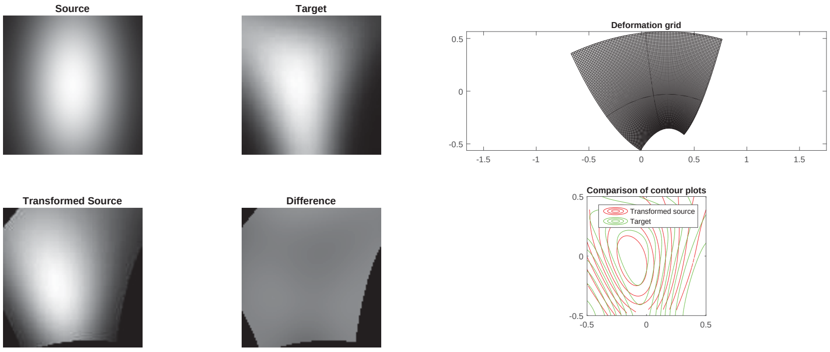


Figure 5.6: Results of conformal registration using the gradient flow method (with four terms) for Example 5.2.

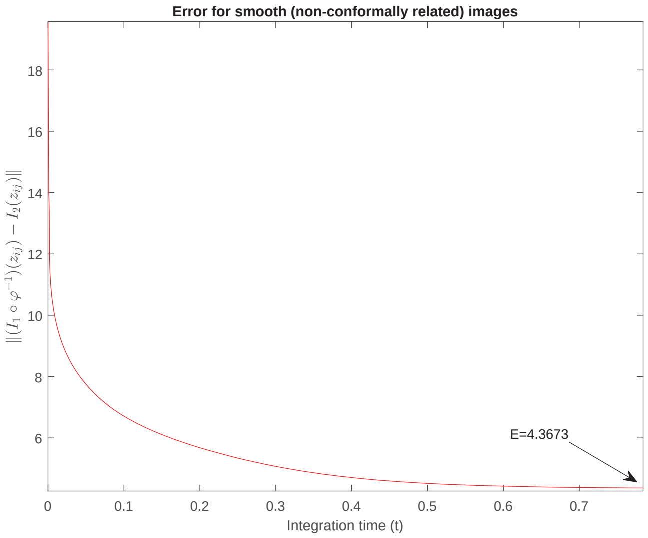


Figure 5.7: Progress of the gradient descent algorithm towards finding a minimum for Example 5.2.

Both methods found nearly identical transformed sources along with nearly aligned contours with the target. The errors in registration for the control points method and the gradient flow method are 3.0862 and 4.3673 respectively.

Example 5.3 (Non-smooth, non-conformally-related images). We consider the standard cameraman photo as the source. We now apply a non-conformal transformation $\varphi^{-1}(z) = 0.1 + z + 0.2z^2 + 0.1\bar{z}^3$ to the source and generate a target. Both images are given in Figure 5.8.

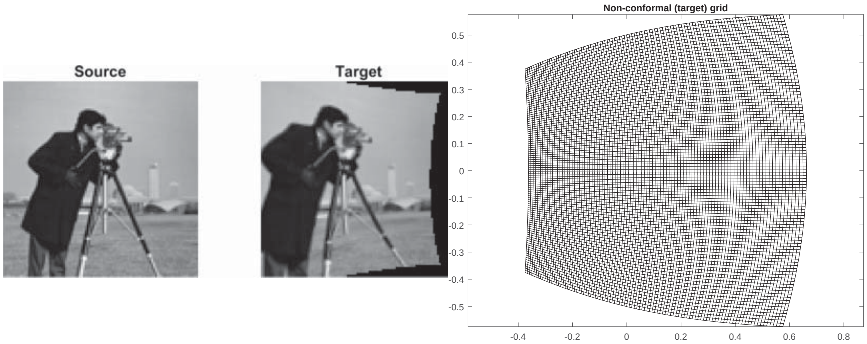
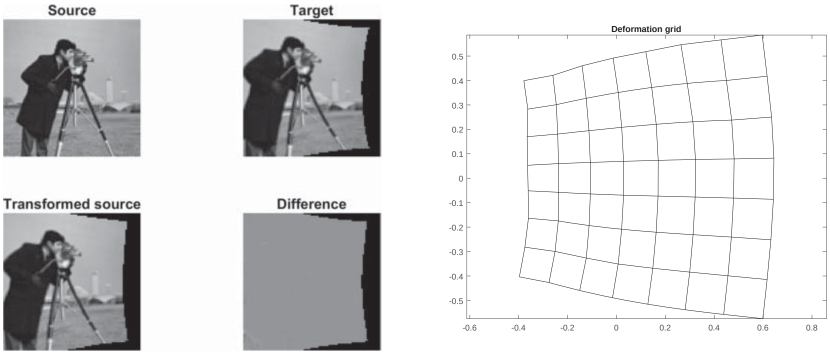


Figure 5.8: *Left:* The source and target images for Example 5.3 are given. The target is defined by $I_2 = I_1 \circ \varphi^{-1}$, where $\varphi^{-1}(z) = 0.1 + z + 0.2z^2 + 0.1\bar{z}^3$. *Right:* The non-conformal map that generates the target.

The results for the control point method (for $\lambda = 150$) are given in Figure 5.9. The results of the gradient flow method (with four terms) are given in Figure 5.10.



n^2	λ	$\|\|(I_1 \circ \varphi^{-1})(\hat{x}_{ij}) - I_2(\hat{x}_{ij})\|\|$	$\|\|P_4\|\|$
16	150	2.3633	0.0092
64	150	1.2291	0.0066
64	150	1.0019	0.0098

Figure 5.9: Results of conformal registration using the control points method for Example 5.3.

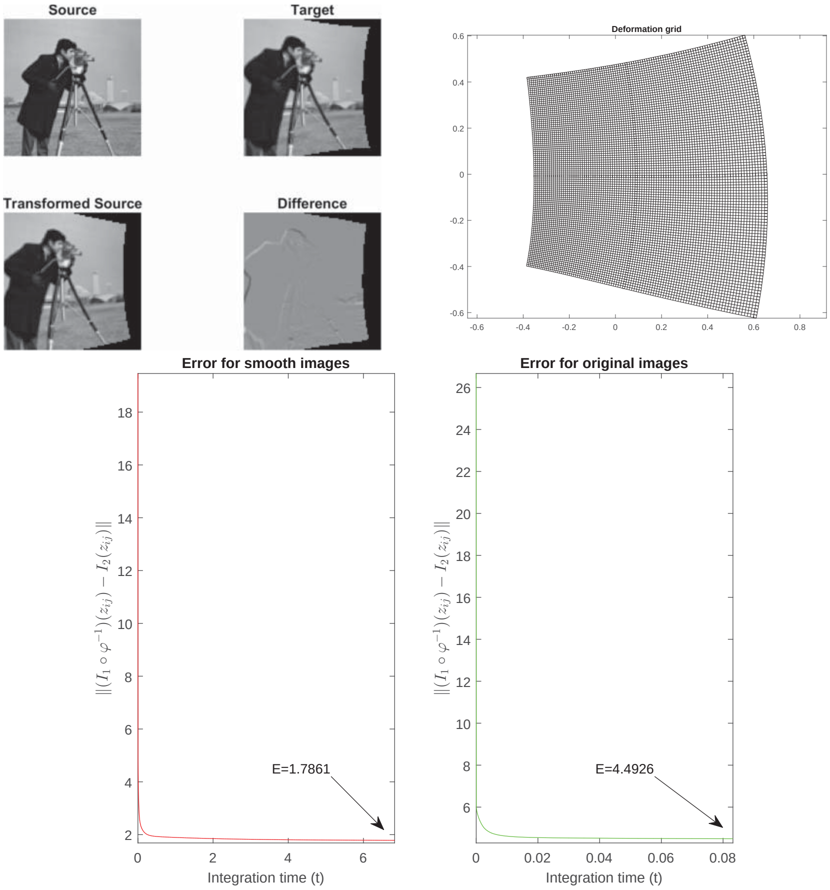


Figure 5.10: *Top:* Results of conformal registration using the gradient flow method (with four terms) for Example 5.3. *Bottom:* Progress of the gradient descent algorithm towards finding a minimum for Example 5.3.

The registration errors for the two methods are 1.0019 and 4.4926, respectively. It can also be seen that the mapping constructed by the control point method (which produced better results of registration, see Figure 5.9) more closely resembles the (non-conformal) map used to generate the target, while retaining the overall discrete conformal property.

Example 5.4 (Thompson's fishes). In this example, the images of Thompson's fishes are reconsidered for image registration. Recall that we already registered these images successfully with the control points method and the fourth penalty term. Here we compare those results to those obtained by the gradient flow method. In this section we have altered the source image (of Example 3.5) by scaling and translating it so that it occupies the whole domain. Both methods are essentially invariant under this transformation and it should not greatly affect the results. The source and target images are given in Figure 5.11.

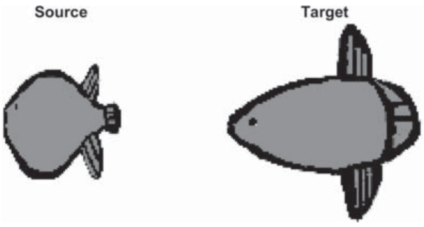
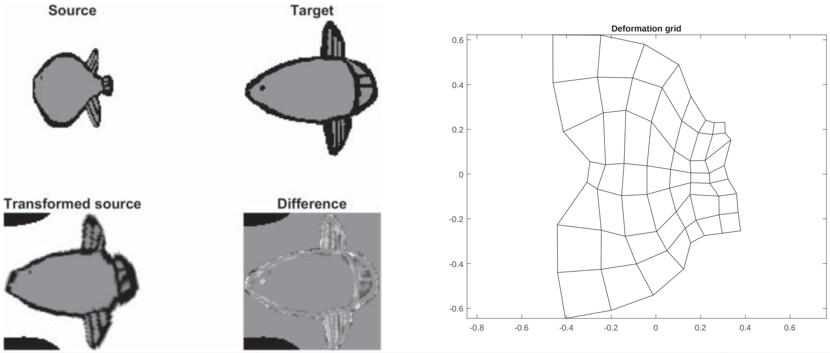


Figure 5.11: The source and target images for Example 5.4, the cartoon versions of Thompson's fish that he believed were 'isogonally' related.

The results of the control points method with the 4th penalty term and $\lambda = 150$ are shown in Figure 5.12; the effect of continuation for other values of λ is shown in Figures 5.13, 5.14, and L-curve is given in Figure 5.15.



n^2	λ	$\|\|(I_1 \circ \varphi^{-1})(\hat{x}_{ij}) - I_2(\hat{x}_{ij})\|\|$	$\|\|P_4\|\|$
16	150	15.1790	0.0624
64	150	7.6496	0.0483
64	150	13.5271	0.0701

Figure 5.12: Results of conformal registration using the control points method for Example 5.4. Although the shapes are very different, an extremely good registration is obtained, with an invertible mapping, at $\lambda = 150$.

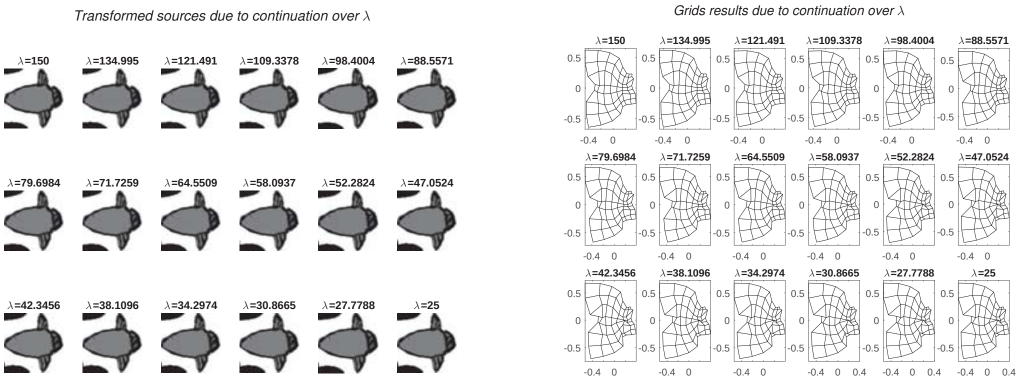


Figure 5.13: Results of the continuation method for Example 5.4 for $25 \leq \lambda \leq 150$.

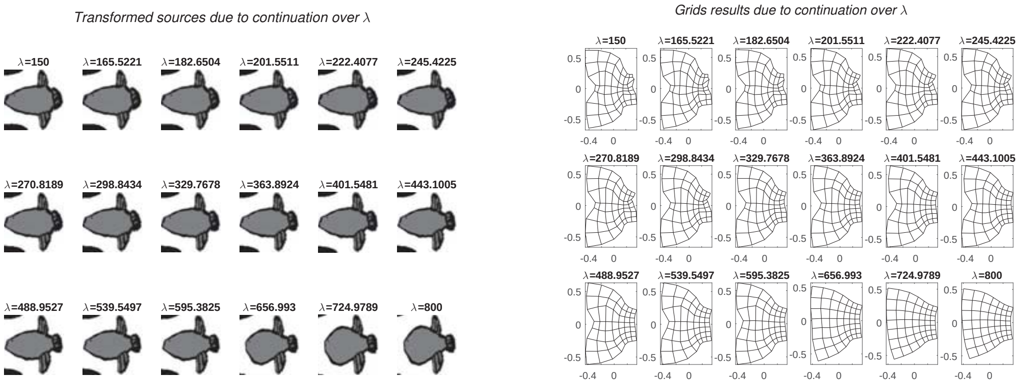


Figure 5.14: Results of the continuation method for Example 5.4 for $150 \leq \lambda \leq 800$. The basic shape of the mapping remains unchanged over a wide range of λ values.

The L-curve method selects a λ in the range 150 to 250. A good registration and a relatively smooth transformation is obtained.

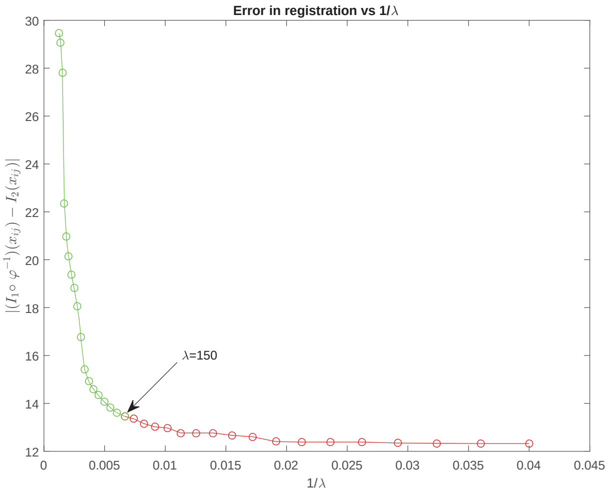


Figure 5.15: The L-curve for Example 5.4.

When we ran Algorithm 3 on this example, it appeared to operate normally and showed good convergence with respect to the number of terms in the Taylor series. However, a closer inspection of the optimal values of the Taylor coefficients computed with four, eight, and twelve terms (not shown) revealed that the matching coefficients were virtually identical in all three cases. To check that this result was correct, we adapted the algorithm in which first the smoothed images were registered with $n = 4$ terms in the Taylor series, and from this solution the original images were registered with $n = 4$. Second, the original images were registered with $n = 8$, using as initial guess the output of the $n = 4$ registration plus a small random perturbation to all coefficients. Third, the original images were registered with $n = 12$ using as initial guess the output of the $n = 8$ registration plus a small random perturbation. That is, we used continuation in n. We checked that the results did not depend greatly on the amount of smoothing in the final steps (here we used no smoothing). The random perturbations were included to ensure that the gradient flow would not simply stop extremely close to the initial guess.

The results of registration with four, eight and twelve terms are given in Figures 5.16, 5.17 and 5.18 respectively. The progress of the gradient flow is shown in Figure 5.20, while Figure 5.19 gives the grids of each attempt plotted on top of each so that the similarities and differences can be clearly seen.

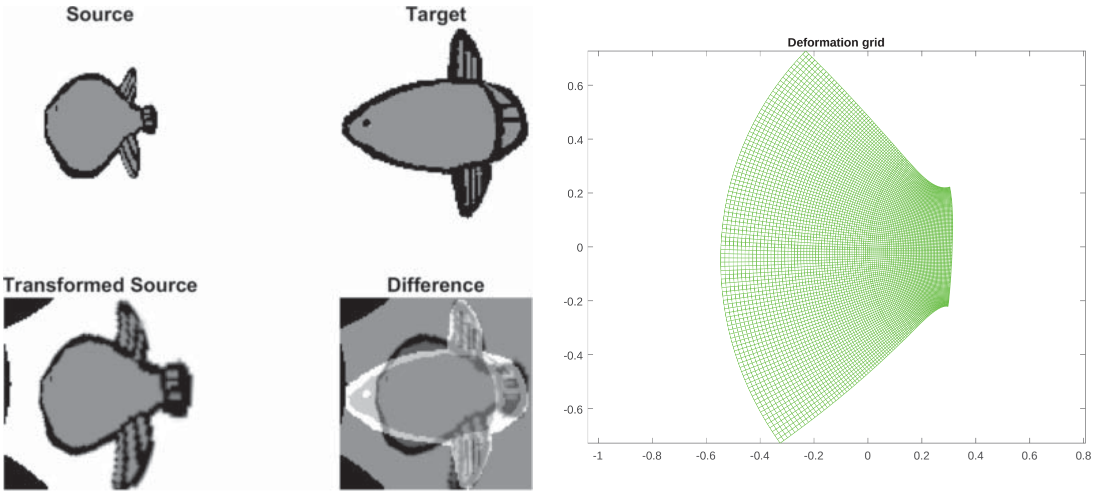


Figure 5.16: Registration results for Example 5.4 with the gradient flow method with four terms.

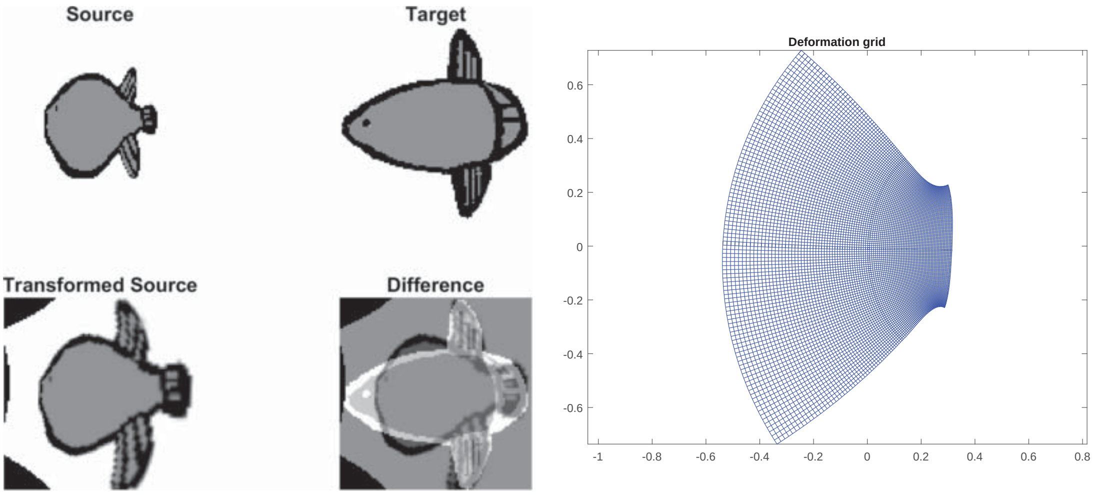


Figure 5.17: Registration results for Example 5.4 with the gradient flow method with eight terms.

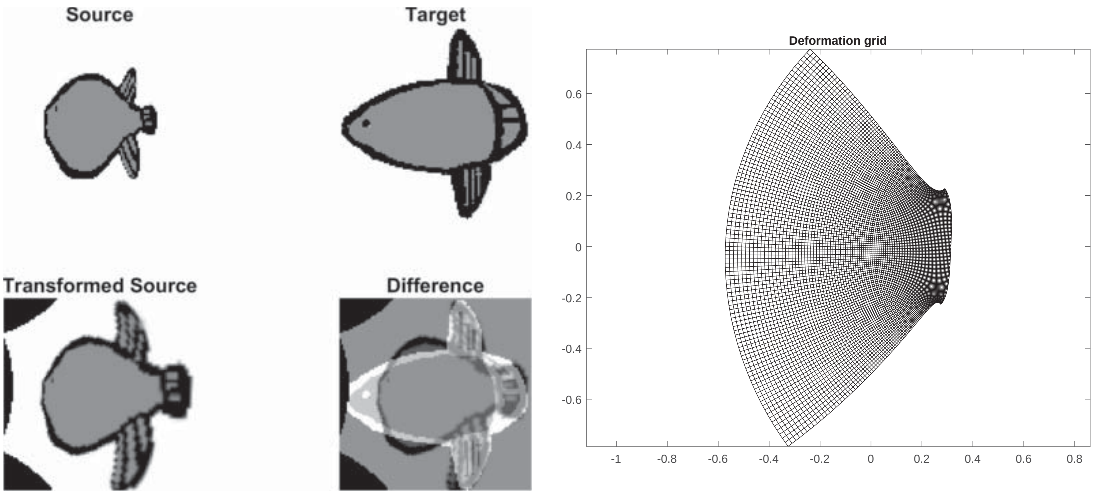


Figure 5.18: Registration results for Example 5.4 with the gradient flow method with twelve terms.

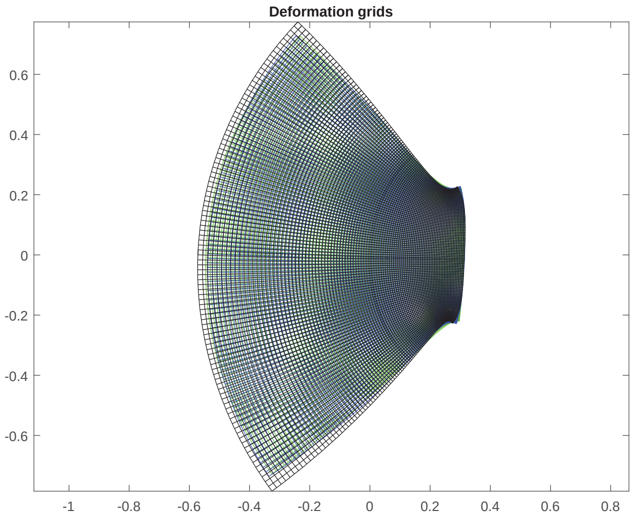


Figure 5.19: The three grids found by the gradient flow method for Example 5.4 are plotted on the same graph. It can be seen that all three grids (with 4 terms in green, 8 terms in blue and 12 terms in black) display slightly different deformations.

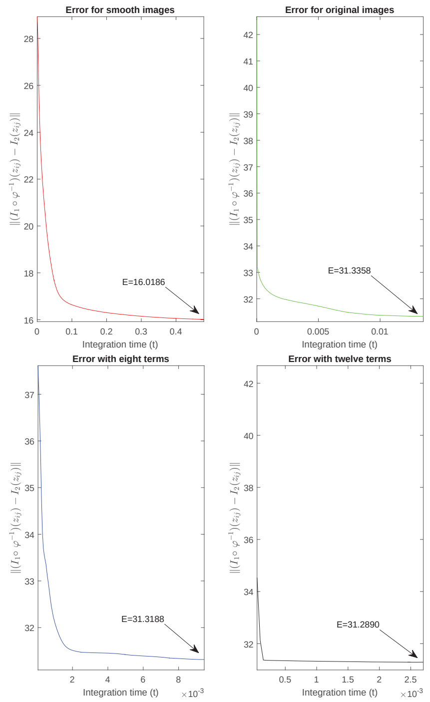


Figure 5.20: Convergence of the gradient flow for Example 5.4; note that the y axis is different for the smooth case. *Top left*: Smoothed images, 4 terms; *top right*: original images, 4 terms; *bottom left*: original images, 8 terms; *bottom right*: original images, 12 terms. In each case the gradient flow ran successfully to a local minimum.

All three grids shown confirm that invertiblity and conformality are secured successfully. The mappings φ^{-1} are shown in Figure 5.19, and all look fairly similar. However, there are some slight differences.

The registration errors with four, eight, and twelve terms in the Taylor series are 31.3358, 31.3188 and 31.2890 respectively. No significant improvement is found by adding more terms. As a final check, we repeated the 8-term gradient flow four times with four different random perturbations to the initial guess; the computed Taylor coefficients are given in Table 5.1. The coefficient a_1 is the same across the 4 runs to within 4%; a_2 to within 20%; a_3 to within 25%; and the higher coefficients are essentially random, and appear to be located very close to the initial guess. All runs have very similar values of the registration error. Our conclusion is that in the region of the a_i parameter space near the initial guesses there are many local minima that all offer very similar quality registrations and so the algorithm gets stuck there.

Coefficients of the Taylor series	***First run***	***Second run***	***Third run***	***Fourth run***
a_0	0.015-0.001i	0.012-0.001i	0.020-0.002i	0.017-0.002i
a_1	0.744-0.022i	0.760-0.027i	0.731-0.017i	0.735- 0.019i
a_2	-0.468+0.038i	-0.523+0.048i	-0.420+0.032i	-0.447+ 0.033i
a_3	0.337-0.053i	0.440-0.050i	0.273-0.055i	0.317-0.053i
a_4	0.001+0.002i	-0.041+0.004i	0.011+0.003i	0.010+ 0.004i
a_5	0.061+0.001i	0.007+0.004i	-0.080+0.003i	0.050+ 0.006i
a_6	-0.024+0.000i	-0.003+0.001i	-0.020+0.001i	-0.007+ 0.002i
a_7	-0.054+0.000i	-0.047-0.000i	0.020-0.000i	-0.005- 0.000i

Table 5.1: The Taylor series (for Example 5.4) found by four different runs. Each run used an initial guess given by the results of the $n = 4$ registration plus a small random perturbation (independent normal random variates with mean 0 and standard deviation 0.04). While the algorithm has located a local minimum in each case, the minima are all different, and the coefficients a_n for $n \geq 4$ carry essentially no information about the image registrations.

To further explore this example we repeated the entire process with 6, 9, and 12 terms. That is, the images were first smoothed and registered with 6 terms in the Taylor series; the rest of the algorithm followed as before. Surprisingly, some quite different grids were obtained. There are shown in Figures 5.21, 5.22, and 5.23 for 6, 9 and 12 terms respectively. The combined grids are given in Figure 5.24. For 6 and 12 terms the gradient flow was able to run to convergence at a local minimum; for 9 terms the mapping developed a singularity (i.e., it became non-invertible) and the optimisation was stopped at that point. The convergence of the gradient flow is shown in Figure 5.25. The registration errors for 6, 9, and 12 terms are 30.1266, 30.0629, 29.9483 respectively; note that, despite the different grids, the registration errors are very similar to the previous registration attempt. Repeated runs gave very similar results.

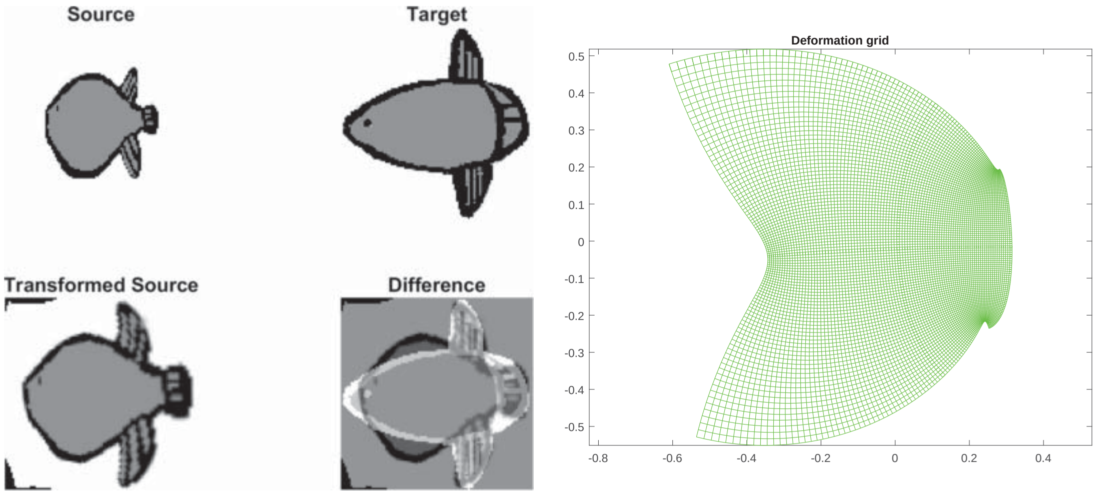


Figure 5.21: Registration results for Example 5.4 with the gradient flow method with six terms.

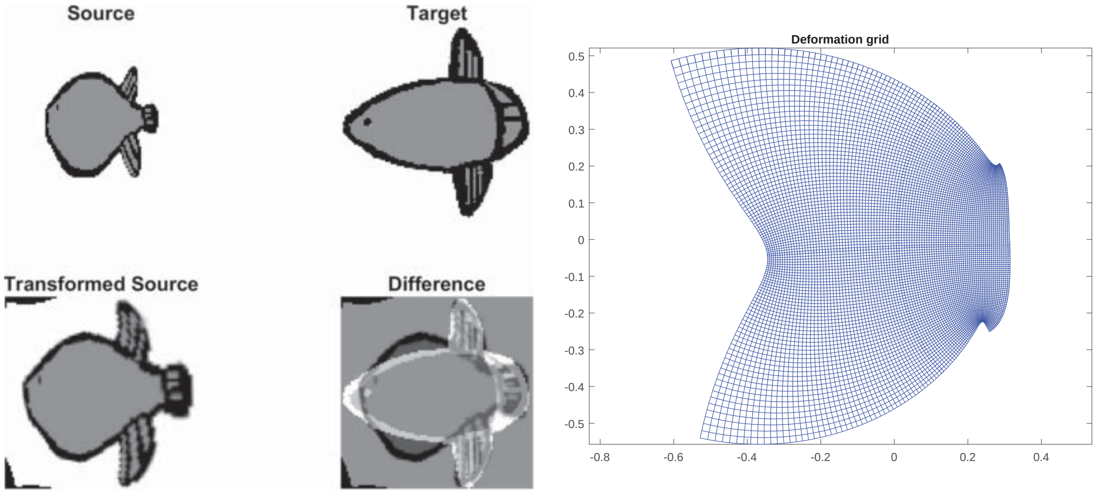


Figure 5.22: Registration results for Example 5.4 with the gradient flow method with nine terms.

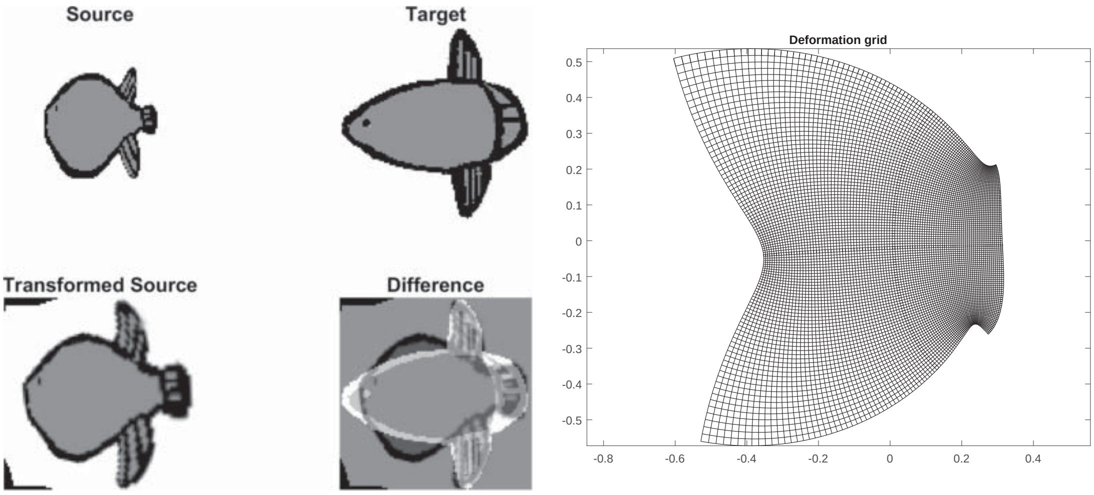


Figure 5.23: Registration results for Example 5.4 with the gradient flow method with twelve terms.

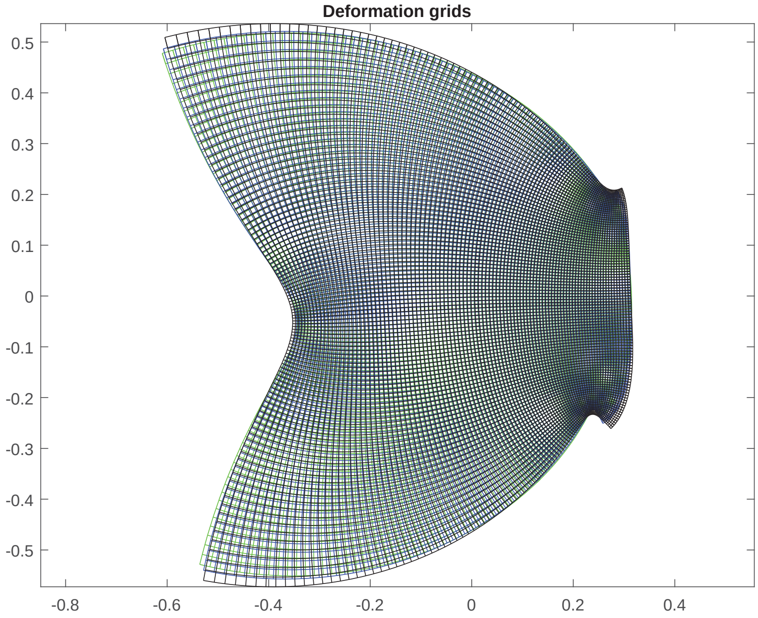


Figure 5.24: The three grids found with six, nine and twelve terms for Example 5.4; compare to the somewhat different registrations found with 4, 8, and 12 terms shown in Figure 5.19.

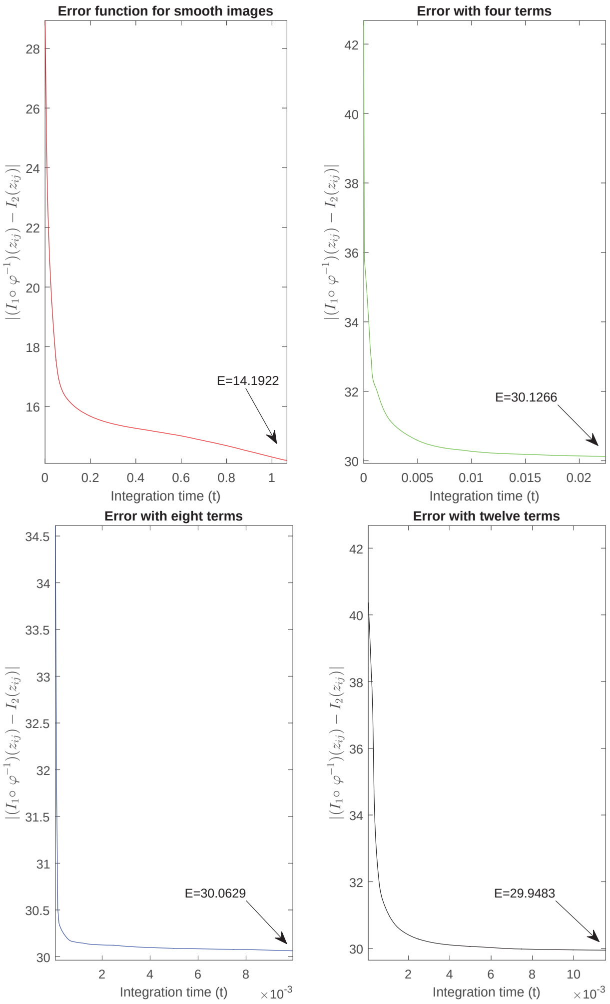


Figure 5.25: Convergence of the gradient flow for Example 5.4. *Top left*: Smoothed images, 6 terms; *top right*: original images, 6 terms; *bottom left*: original images, 9 terms; *bottom right*: original images, 12 terms. The two graphs on the left do not show convergence to a local minimum because the gradient flow was stopped due to non-invertibility of the mapping.

We tentatively conclude that, in addition to the very many closely-spaced local minima, the objective function has other more widely-spaced local minima, with different solutions, but with nearly equal values of the objective function. We have established that conformal registration of this difficult problem is possible, but that it is difficult to guarantee that we have located the global minimum or have found a high-quality registration. In fact, this example has only really been satisfactorily conformally registered with the control points method (Chapter 3), not with the gradient flow method (Chapter 4).

Example 5.5 (Cat to chicken registration). In [114], a version of the gradient flow method for diffeomorphisms is demonstrated on a binary image of a cat, which is transformed to that of a chicken. These images (Figure 5.26) are taken from a publicly available database [31]. Although this is an extreme example, partly for fun we attempt to conformally register these images.

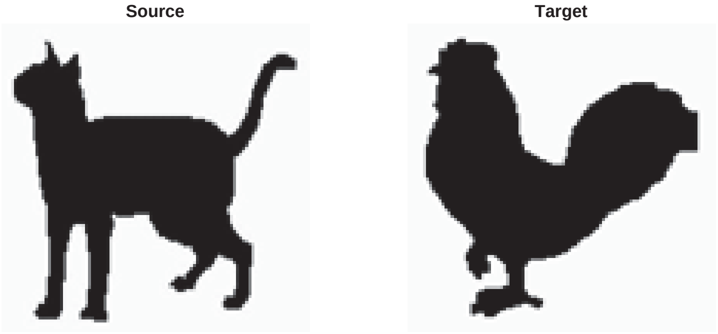
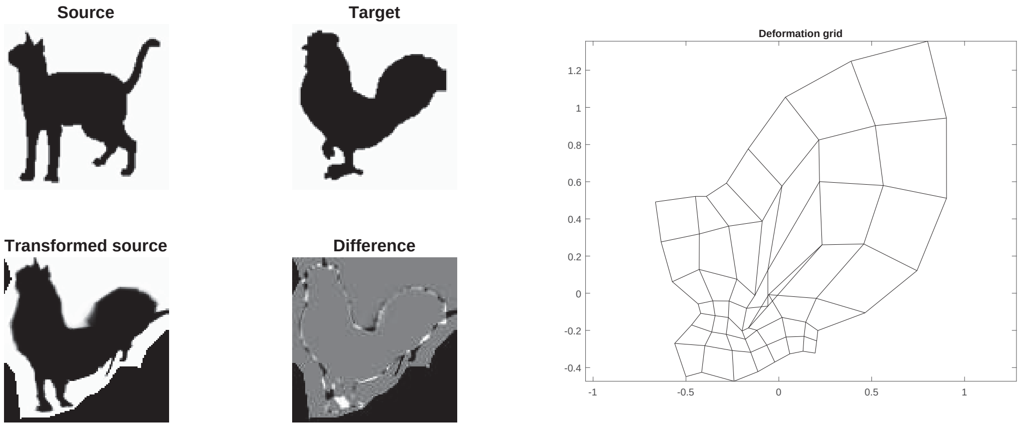


Figure 5.26: A pair of black and white images. *Left:* Image of a cat, which serves as the source. *Right:* Image of a chicken, which is used as the target.

Images like this define simply connected domains [100], [28]. According to the Riemann mapping theorem [107], [61] they can be mapped to an open disc with a conformal transformation. Therefore, regarding the boundary of the cat and chicken as continuous planar curves, there exists a conformal registration between them. The results of the control points method are given in Figures 5.27, 5.28 and 5.29 for different ways to choose λ.



n^2	λ	$\|\|(I_1 \circ \varphi^{-1})(\hat{x}_{ij}) - I_2(\hat{x}_{ij})\|\|$	$\|\|P_4\|\|$
16	150	21.4848	0.0812
64	150	6.6024	0.0690
64	150	16.4794	0.0856

Figure 5.27: Results of conformal registration using the control points method for Example 5.5. A good registration is obtained at the cost of an irregular grid. Note that the first two values in the table are with the smoothed images, while the last row is not.

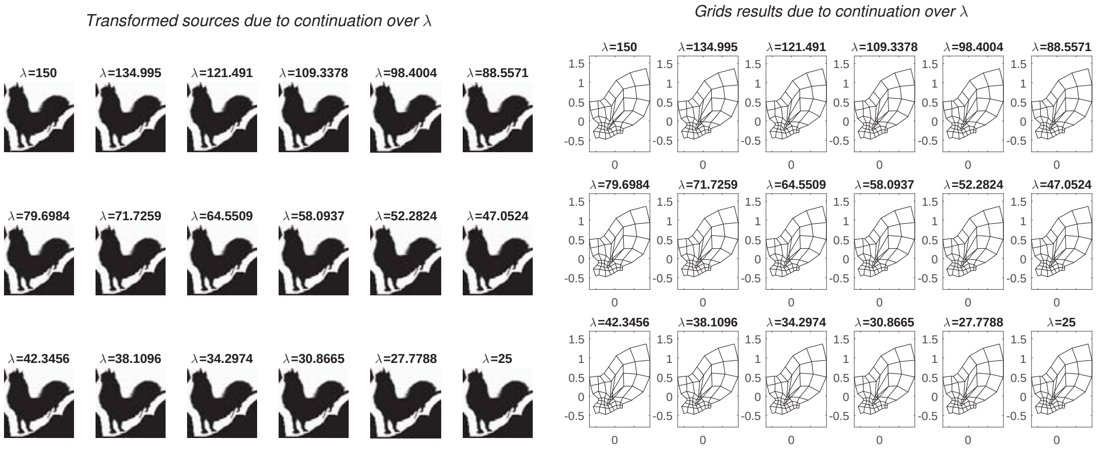


Figure 5.28: Results of the continuation method for Example 5.5 for $25 \leq \lambda \leq 150$.

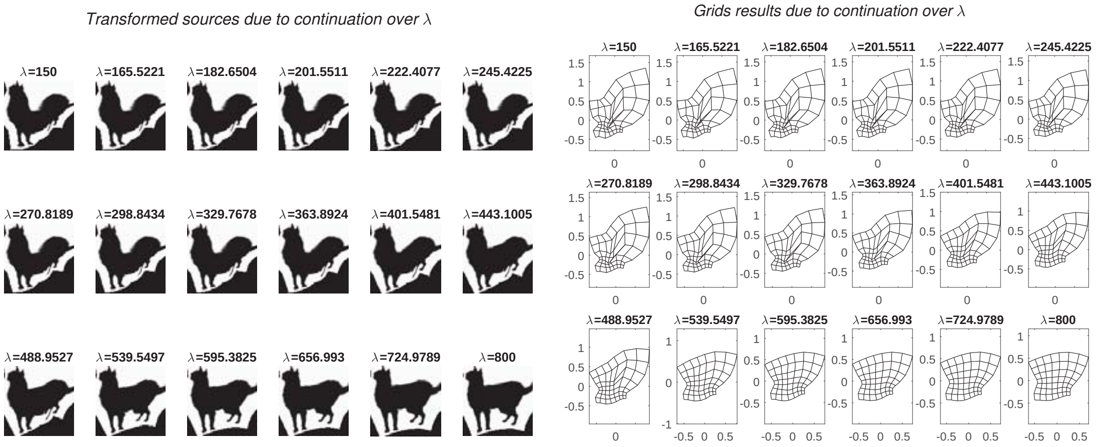


Figure 5.29: Results of the continuation method for Example 5.5 for $150le\lambda \leq 800$. When λ is sufficiently large (greater than about 240) the mapping is invertible.

The method has successfully aligned the overall body shapes of the animals, but at the cost of a somewhat irregular grid. For $\lambda \lesssim 240$ the constructed map is not invertible in one of the cells. This is also the approximate location of the corner of the L-curve (Figure 5.30).

The gradient flow algorithm was applied in the same way as in Example 5.4. The results for 4, 8, and 12 terms in the Taylor series are shown in Figures 5.31, 5.32 and 5.33 respectively. The combined grids are shown in Figure 5.34 and the error graphs in Figure 5.35.

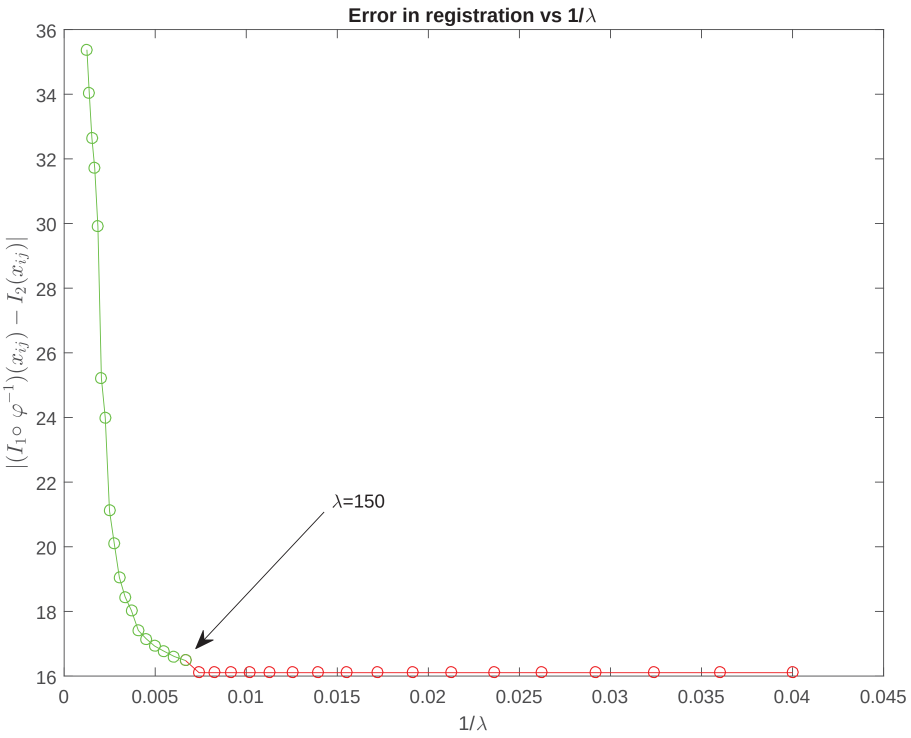


Figure 5.30: The L-curve for Example 5.5.

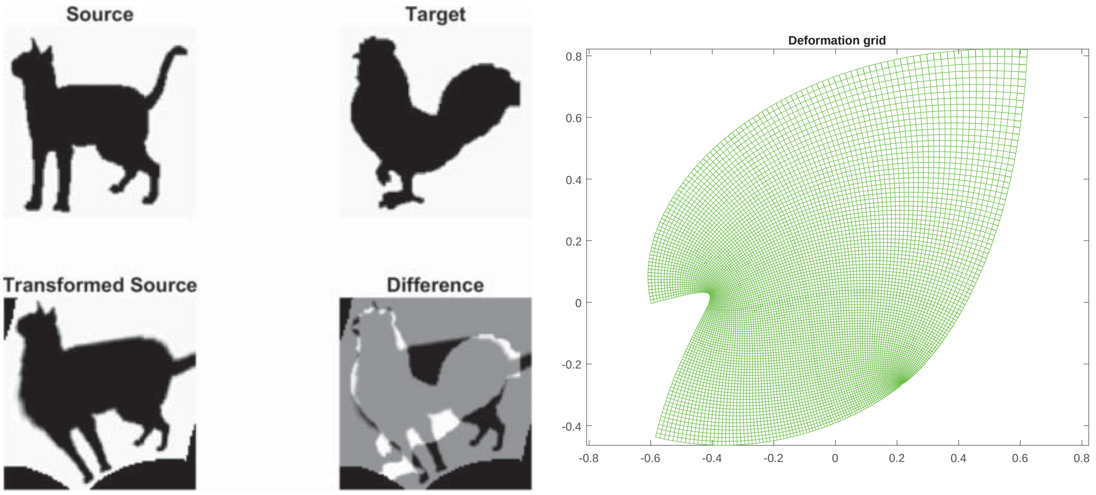


Figure 5.31: Registration results for Example 5.5 with the gradient flow method with four terms. The deformation is conformal, but the match is poor.

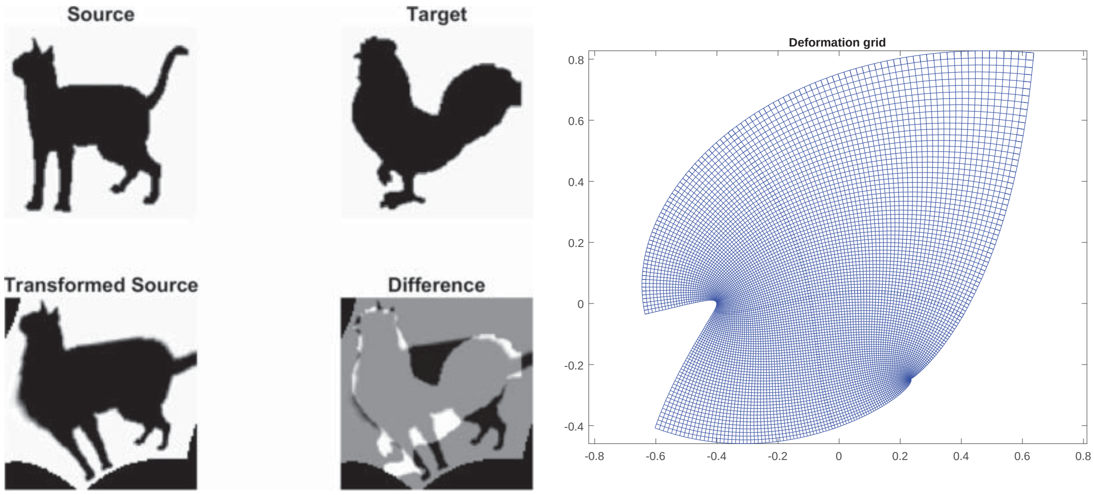


Figure 5.32: Registration results for Example 5.5 with the gradient flow method with eight terms. The registration is not markedly different to that with four terms.

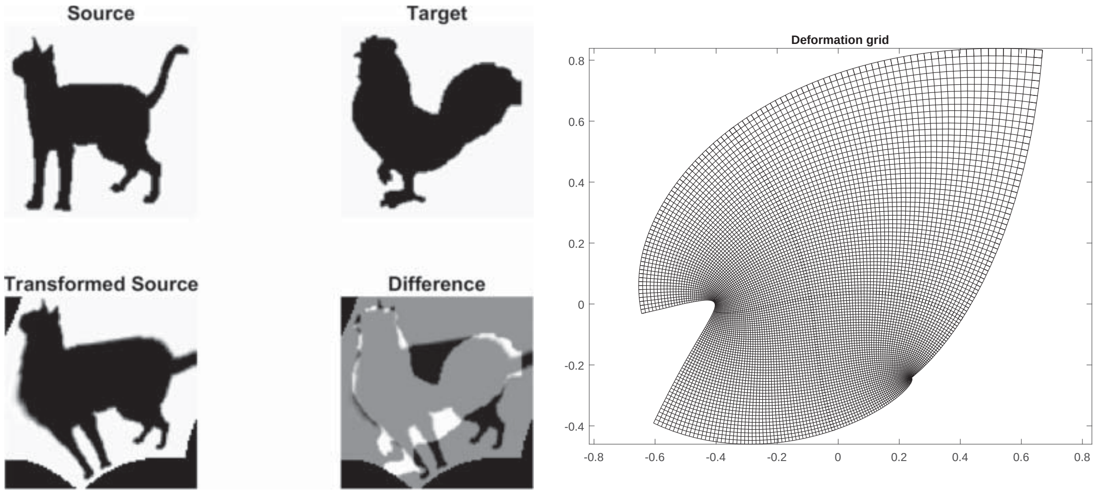


Figure 5.33: Registration results for Example 5.5 with the gradient flow method with twelve terms. The registration is not markedly different to that with four or eight terms.

The final registration errors are 38.3589, 38.0234, and 37.8034 respectively. For 4 and 8 terms, the gradient flow successfully located a local minimum, while for 12 terms the algorithm stopped at $t = 0.0115$ as the grid became non-invertible at this point. (See the convergence results in Figure 5.35; inspecting Figure 5.32 closely, it is clear that that solution is also very close to forming a singularity.)

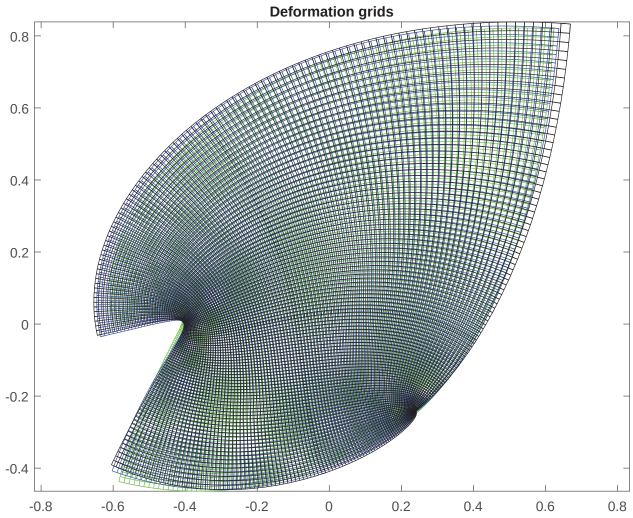


Figure 5.34: Three grids for Example 5.5 (with four, eight and twelve terms) are shown. Each grid shows a slightly different deformation.

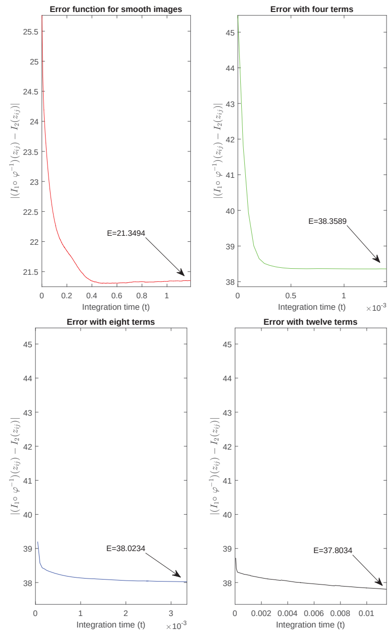


Figure 5.35: Convergence of the gradient flow for Example 5.5. *Top left*: Smoothed images, 4 terms; *top right*: original images, 4 terms; *bottom left*: original images, 8 terms; *bottom right*: original images, 12 terms. For 12 terms the gradient flow stopped at $t = 0.0115$ (before locating a local minimum), due to the emergence of a non-invertible grid.

Surprisingly, the control points method did a good job for this pair of images, at the cost of a less smooth transformation, while the gradient flow produces an exactly conformal map with worse registration.

Example 5.6 (Skull growth). We discussed in Chapter 1 the comparison of human skulls that was presented in [81]. In that paper, the author used Möbius transformations in the modelling of the growth of the human skull. As the Möbius group is a finite dimensional subgroup of the infinite dimensional conformal group, we became interested in the registration of these images by conformal mappings. The actual skull images, of 5 human skulls at different ontogenetic (i.e. since formation) times, are taken from [63]. These images of skulls are given in Figure 5.36.

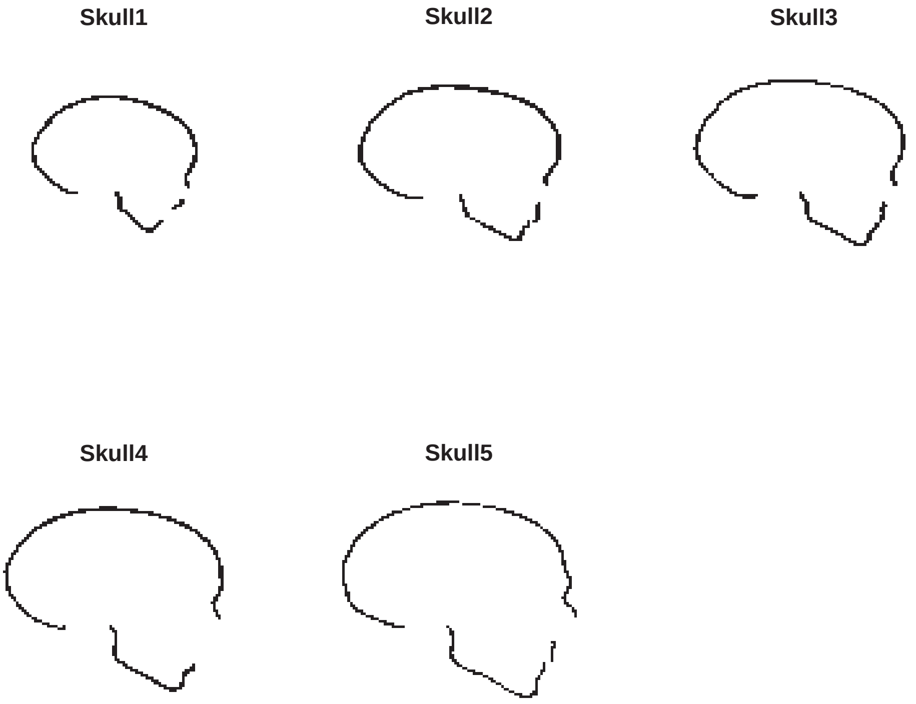
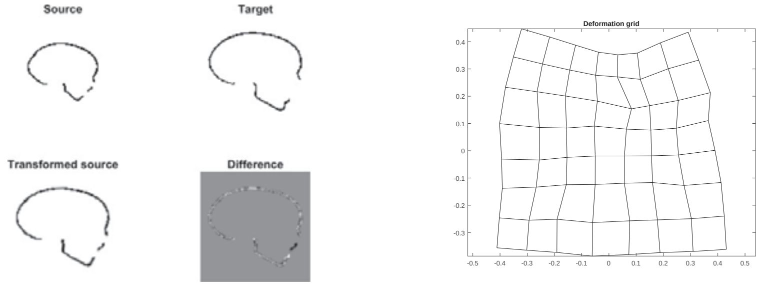


Figure 5.36: Five images of immature human skulls are presented at different ontogenetic times. These images of skulls are taken from [63].

The original skull outlines were very high resolution and contained lines of varying thickness. We resized them to 100 × 100 pixels and then set to black (i.e., to be part of the line) *any* nonwhite pixel. This simple method gave lines of roughly constant thickness.

We first consider 'Skull1' as the source and 'Skull4' as the target. Note that one feature of the algorithms is that they are both easily able to cope with line drawings; sufficient smoothing makes the images able to 'see' one another since the thickened curves overlap. The objective function essentially attempts to make the lines overlap as much as possible. The breaks in the lines (which here are natural, but which in other cases might be due to missing data) are also not a problem.

The results for the control point method are shown in Figure 5.37 for $\lambda = 150$ and for the entire range of λ in Figures 5.38, 5.39, and 5.40. The results for the gradient flow method with 12 terms in the Taylor series are shown in Figure 5.41.



n^2	λ	$\|\|(I_1 \circ \varphi^{-1})(\hat{x}_{ij}) - I_2(\hat{x}_{ij})\|\|$	$\|\|P_4\|\|$
16	150	2.3424	0.0019
64	150	2.0387	0.0049
64	150	9.1343	0.0301

Figure 5.37: Results of conformal registration using the control points for Example 5.6 at $\lambda = 150$, ‘Skull1’ (source) registered to ‘Skull4’ (target).

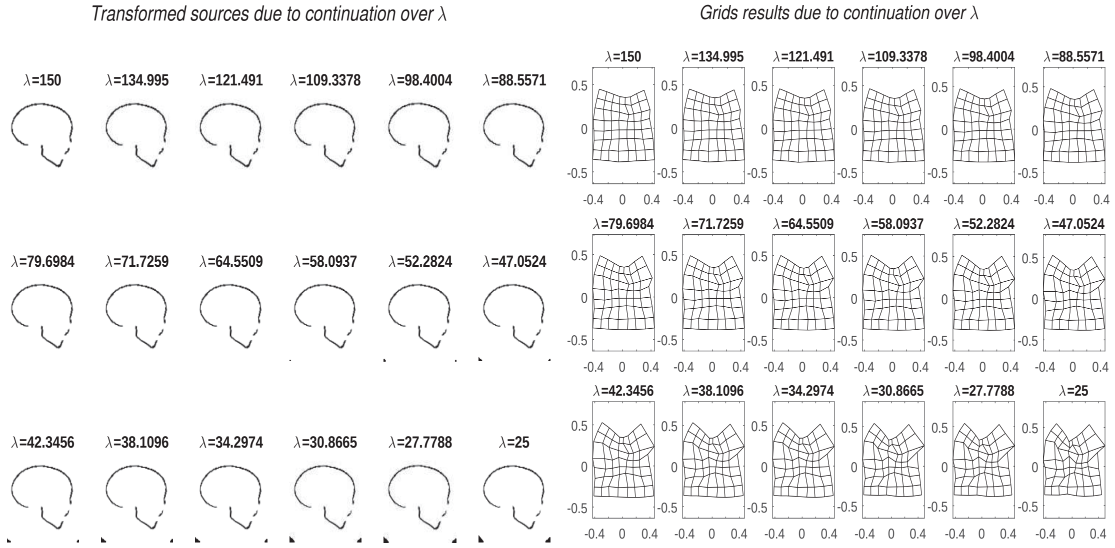


Figure 5.38: Results of the continuation method for Example 5.6 for $25 \leq \lambda \leq 150$.

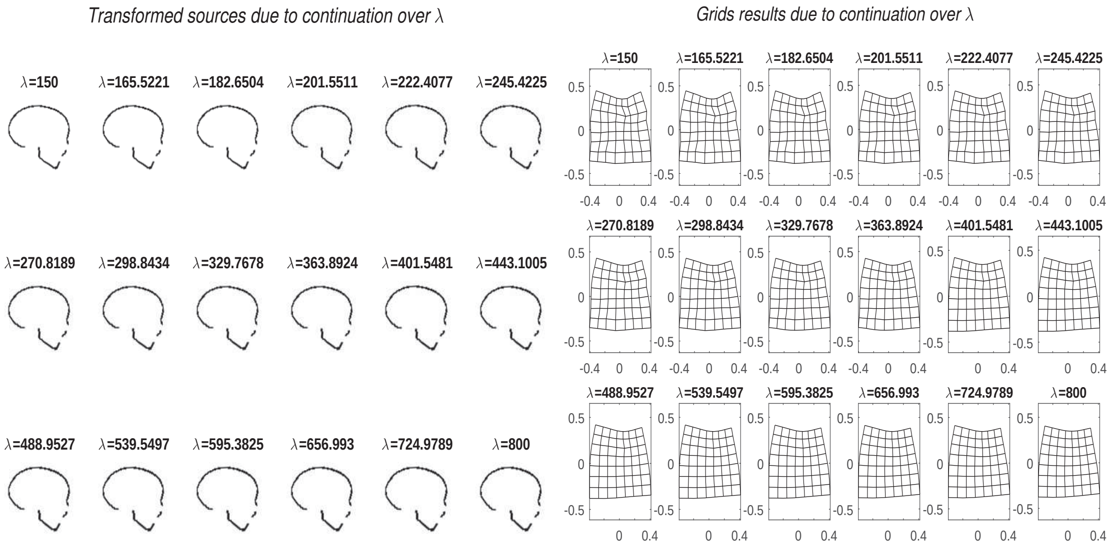


Figure 5.39: Results of the continuation method for Example 5.6 for $150 \leq \lambda \leq 800$.

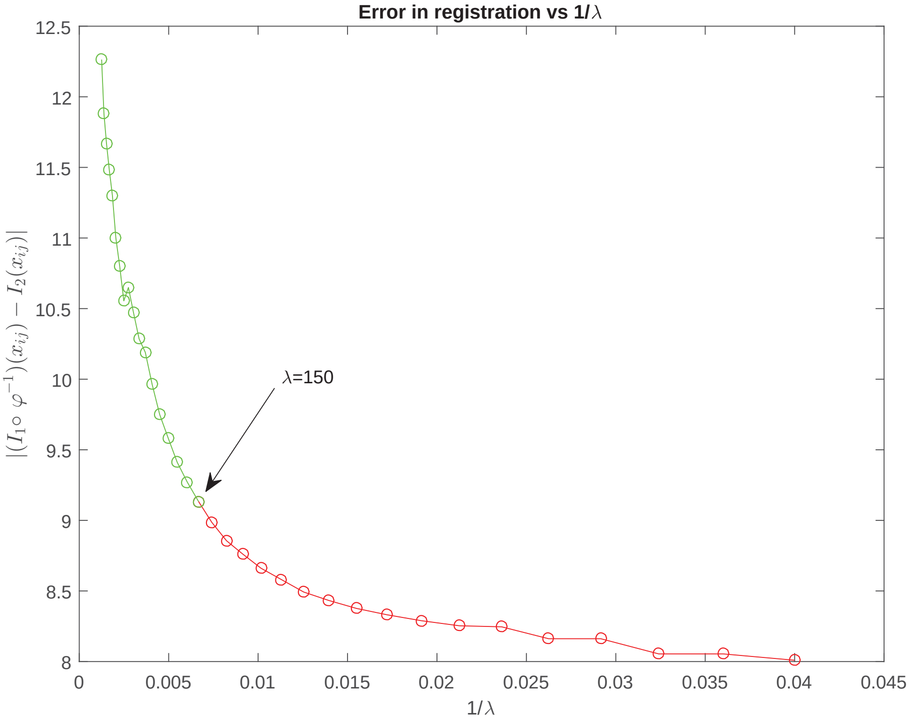


Figure 5.40: L-curve for Example 5.6, ‘Skull1’ registered to ‘Skull4’.

Both methods produce excellent registration of this pair of images. The control points method produces a better registration than the gradient flow method (error 9.1343 vs. 13.7248, see also the figures), at the cost of a less smooth transformation. A few local things remain unregistered, such as the portion between the nose and lips.

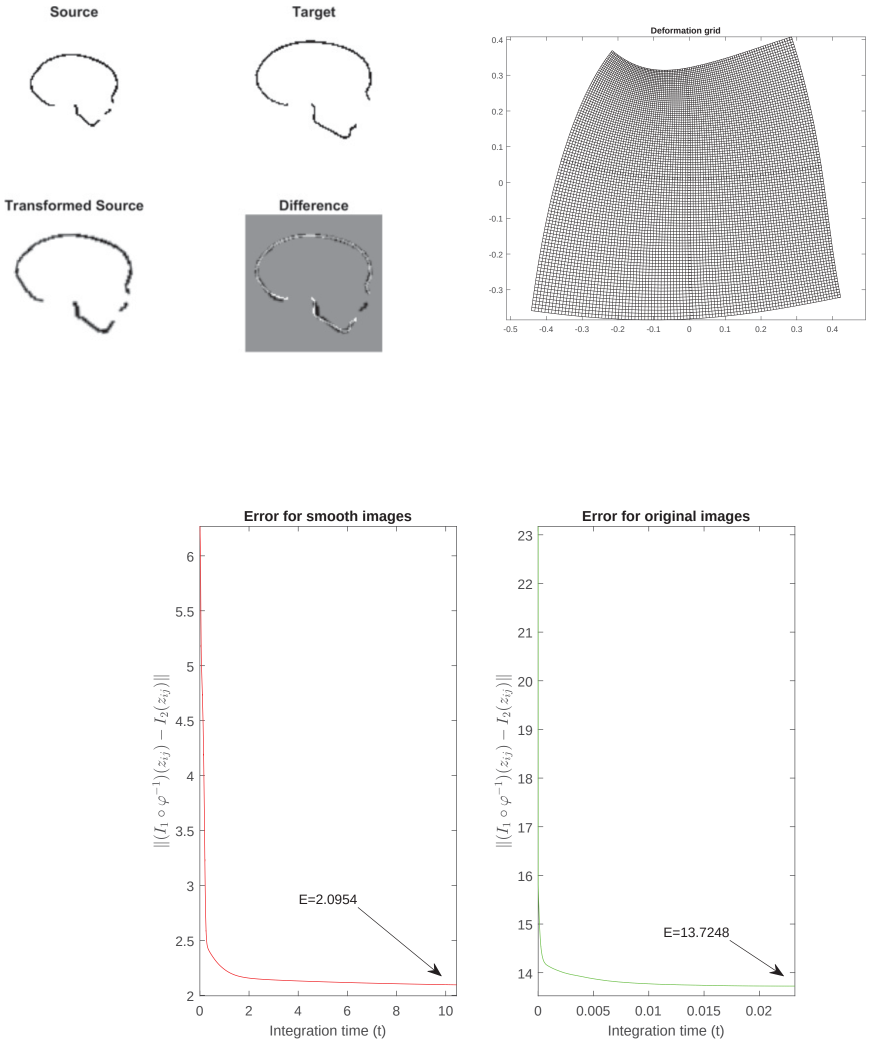
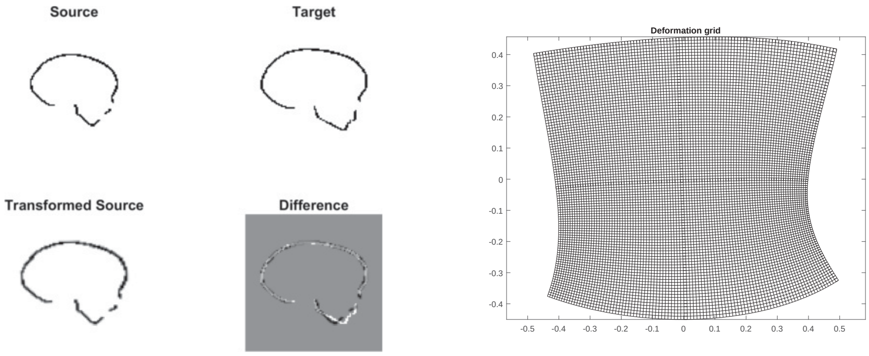


Figure 5.41: Results of conformal registration using the gradient flow method, ‘Skull1’ registered to ‘Skull4’.

However, in conformal registration our aim is to register global, not local, features and that aim is achieved successfully with both methods.

We also registered ‘Skull1’ to each of the other skulls with the gradient flow method. The results are given in Figures 5.42, 5.43, and 5.44. In all these experiments the number of terms for the Taylor series is 12.

Conformal image registration of a whole *family* of related images like this allows us to study the evolution of the family of shapes. Here this corresponds to the way the shape of a skull changes during its growth. The registration decomposes the shape evolution into two parts. One part, coming from the conformal maps and determining the overall smooth trend in the shape evolution, is represented as a curve in the space of conformal maps. The other part, the difference between the targets and the transformed source, is composed of the non-conformal part of the shape change, the small local shape changes,



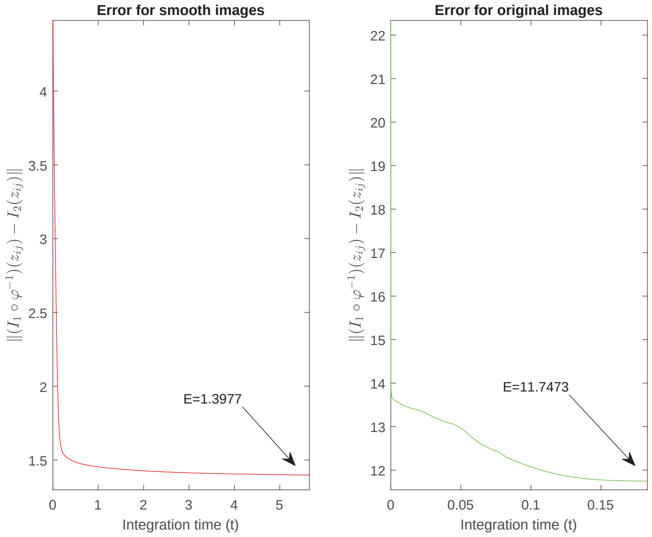
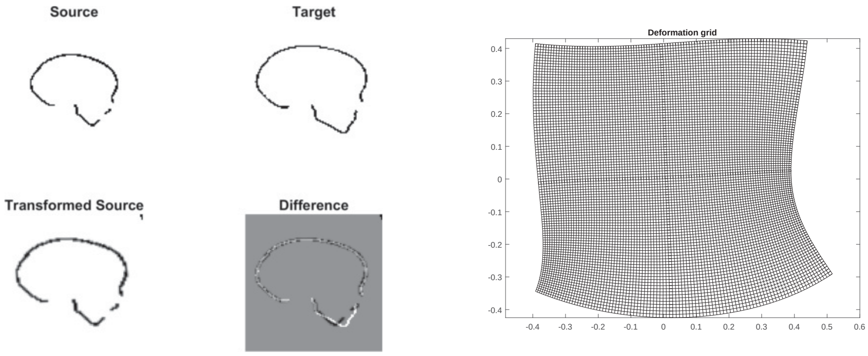


Figure 5.42: Results of registration (using gradient flow) when ‘Skull1’ is registered with ‘Skull2’.

and the noise in the images. In the present example the second part is quite small as the images are very well registered by conformal maps.

While the preceding discussion holds for any image registration technique for registration in any space of transformations, conformal registration by the gradient flow method has an important and significant feature not shared by other approaches. Namely, the conformal maps are represented as Taylor series. The invertible Taylor series form an open subset of the vector space of all Taylor series; here we are using the Taylor series coefficients a_i as coordinates on a finite-dimensional subset (the truncated Taylor series) of the conformal maps. This ability to put coordinates on a nonlinear shape space is a huge simplification and advantage that is not shared by other methods.

However, even more is true. The Taylor coefficients of lower powers of z are, in general, more informative than the Taylor coefficients of higher powers of z. The same



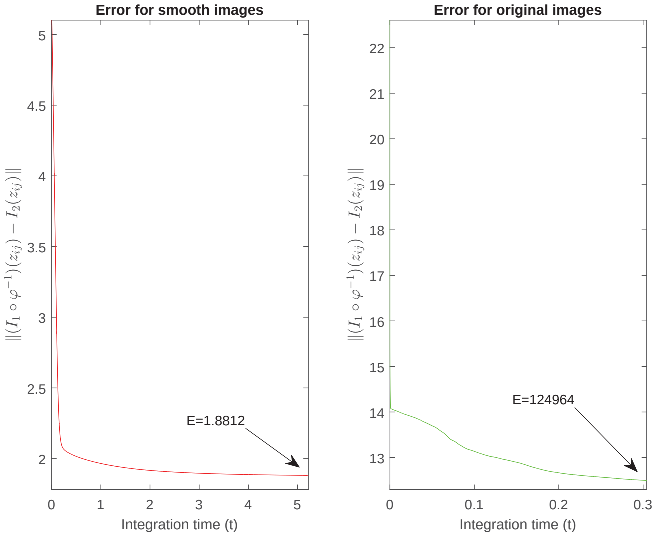


Figure 5.43: Results of registration (using gradient flow) when ‘Skull1’ is registered with ‘Skull3’.

is true of Fourier coefficients. (Putting $z = re^{i\theta}$ and fixing r, the Taylor series becomes exactly a (finite) Fourier series.) Thus, the Taylor coefficients form a kind of spectrum of the conformal map.

The coefficient a_0 is the translation part. In many applications, such as the skulls, this can be ignored. The coefficient $\arg a_1$ is the rotation part. Although not completely devoid of information, since the skull images are presumed to be captured in the same way, it still seems that the rotation part is not that relevant. The coefficient $|a_1|$ is the scaling part and captures the overall amount of growth between source and target—it is clearly important. The coefficient a_2 is the most important nonlinear part of the shape change. The higher coefficients are important too but are expected to be increasingly affected by smaller, local, shape changes.

Thus in Figure 5.45 we plot $|a_1|$ vs $|a_2|$ for the five skull registrations, where the

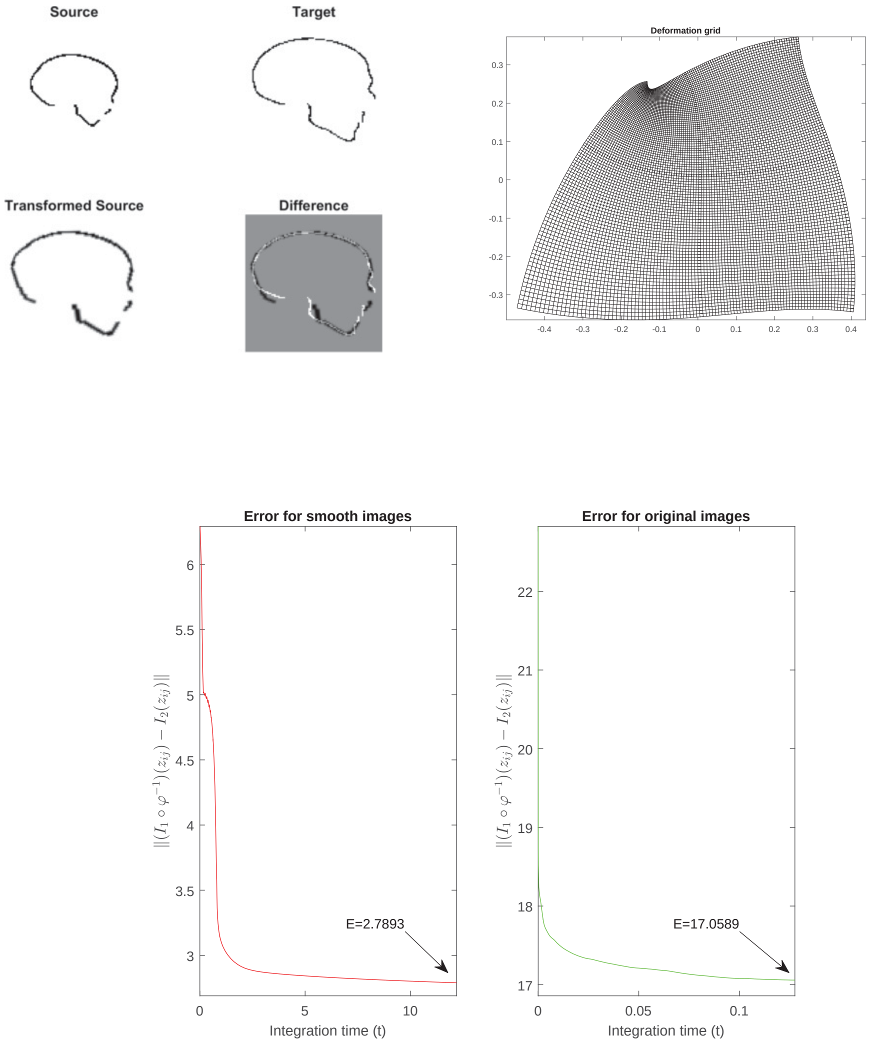
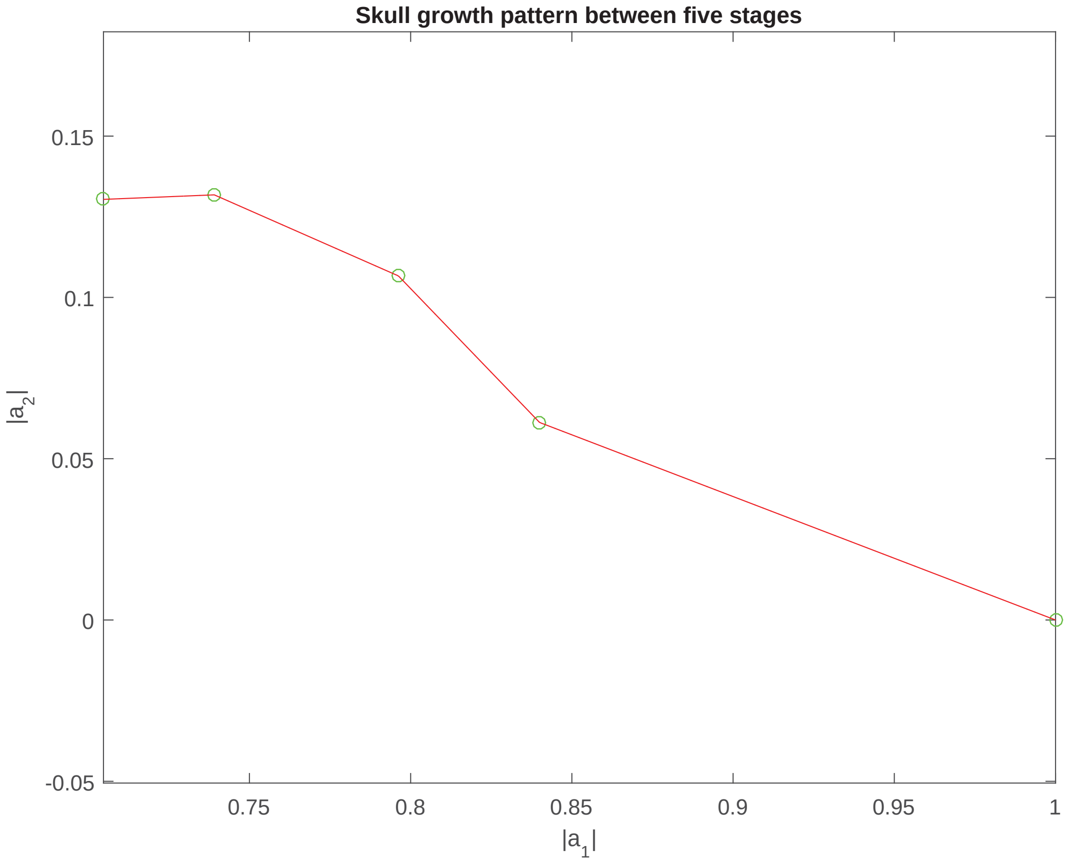


Figure 5.44: Results of registration (using gradient flow) when 'Skull1' is registered with 'Skull5'.

first data point is $a_1 = 1$, $a_2 = 0$ and corresponds to registering 'Skull1' to 'Skull1' by the identity. The hypothesis is that this set of data will lie on a smooth curve corresponding to growth by a smooth family of conformal maps. Examining the data and its graph, the hypothesis appears to be reasonable.

This kind of analysis opens up many possibilities for future research. The shape data is easy to interpolate or to fit with a simple model such linear growth. This allows the intermediate forms to be calculated or approximated smoothly. The statistical significance of the higher Taylor coefficients could be studied. This is a problem in statistical model selection. For larger shape data sets whose relationship is a priori unknown—for example images of hundreds of individuals of the same or closely related species—the shape relationships are now put in a linear space so that standard statistical techniques of variation (e.g. Principal Components Analysis, [95], [82]) and



	Skull1		**Skull2**		**Skull3**		**Skull4**		**Skull5**	
	$\lvert a_1\rvert$	$\lvert a_2\rvert$	$\lvert a_1\rvert$	$\lvert a_2\rvert$	$\lvert a_1\rvert$	$\lvert a_2\rvert$	$\lvert a_1\rvert$	$\lvert a_2\rvert$	$\lvert a_1\rvert$	$\lvert a_2\rvert$
Skull1	1	0	0.8400	0.0612	0.7962	0.1067	0.7391	0.1318	0.7047	0.1304

Figure 5.45: This graph indicates a smooth conformal pattern of skull growth in Example 5.6, with shape data relating the skull shapes (from the set of conformal mapping) projected to two dimensions. The data points are the absolute values of two Taylor coefficients in the conformal registration of ‘Skull1’ to each of the 5 skulls.

clustering can be applied. This would be one obvious way to extend this work and utilise it for morphometrics.

5.1 Comparing the Algorithms

In this chapter the two main algorithms (Algorithm 2, which is based on control points and a penalty term and was presented in Chapter 3 and Algorithm 3, which is based on gradient flow and was presented in Chapter 4) are compared. For simple problems both methods provide successful registrations. The application of these algorithms to more challenging examples also provided satisfactory outcomes most of the time. For Example 5.1, in which the registration of two circles is presented, the control points method appears more appropriate than the gradient flow method because the former algorithm gives a better error term than the later. These algorithms are also adequate for the registration of pair of non-conformally related images (see Example 5.2 for smooth pair of images and Example 5.3 for non-smooth pair of images). The reason for considering these two examples is to give the idea to a reader of how well the job of registration can be done; even on pairs of non-conformally related images. Both algorithms produced nearly identical results of registration for Example 5.2. However, in Example 5.3 the control points method produced better registration results than the gradient flow (see a perfect mid-grey screen for the difference image in Figure 5.9, but not in Figure 5.10).

In Example 5.4, a pair of Thompson's fish is considered for the registration. The control points method satisfactorily conformally registered the images but the gradient flow method did not, probably due to the closely-spaced local minima. Examples of registrations of a black and white image of a cat to a chicken are given in Example 5.5. The control points method (surprisingly) has performed an excellent job of registration, although unfortunately at the cost of conformality. In contrast, the gradient flow method was unable to register the images at all due to the existence of a singularity near to the boundary (this can be seen in the deformation grid in Figure 5.33). Finally, the registration of images of the ontogenetic growth of immature human skulls was presented and both the algorithms produced significant (although not perfect) results for registration. Our overall conclusion is that the control points method has better capability than the gradient flow method for image registration in general, but the desire to find conformal deformations brings with it computational costs.

Chapter 6

Conclusions and Future Work

A mathematical theory is not to be considered complete until you have made it so clear that you can explain it to the first man whom you meet on the street.

- David Hilbert (1862-1943)

6.1 Conclusion

We have developed and presented two algorithms for the construction of conformal diffeomorphisms, one based on constrained optimisation of a set of control points (Chapter 3), and one based on gradient flow (Chapter 4). For the first method we considered four penalty terms that aimed to enforce conformality, based either on discretisations of the Cauchy-Riemann equations, or geometric principles. The penalty term that worked best simply measured how square each pixel cell is. For the gradient flow method the conformal diffeomorphisms are represented as a finite Taylor series. The algorithms have been tested on a variety of datasets, including synthetic data (i.e., the target is generated from the source using a known conformal transformation; the easiest possible case), and real images, including some that are not actually conformally related.

We compare the results of using these algorithms on some particularly interesting datasets: Thompson's fish example, which was one of our primary motivations for studying conformal image registration, and a small dataset demonstrating the growth of a human skull (following the work of Petukhov, who also believed that some examples of human growth are conformal, or even Möbius). For Thompson's fish it is clear from Figure 6.1 that local angles are not preserved in the mapping between parts of the fish, particularly the tails, fins, and lips. This contradicts Thompson's argument about the existence of isogonal transformations between these pairs. For the skull growth data, which is a very small dataset, the results of using our algorithms suggests that the conformal growth model cannot be ruled out, although more data would be needed to test this fully.

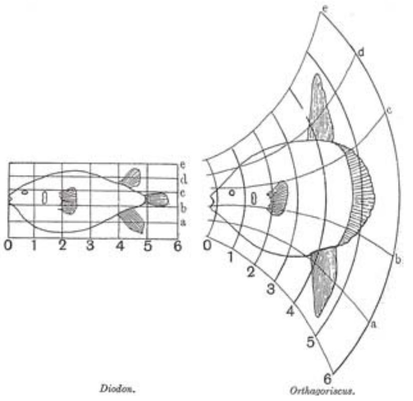


Figure 6.1: One of Thompson's isogonally related fish examples. These scanned images are taken from [101] (p 1064).

6.2 Future work

1. **Avoiding spurious local minima in the search space**

   The Algorithm 3 in Section 4.1.1 is based on the gradient descent approach. Although the registrations often worked well (and in a few of the examples where the global optimum was know, the algorithms successfully found it), gradient descent only guarantees the finding of a local minimum. It is likely that the search space has a lot of local minima in it, and so the gradient descent is getting stuck in shallow minima, some of which may be close to the initial condition. While it is obviously desirable to find the global minimum, this is extremely difficult without good knowledge of the search space. However, further investigation into how to avoid spurious, particularly shallow, local minima could provide better registrations. This may be particularly important for examples where the images are not exactly conformally related. It is possible that some form of stochastic gradient descent, such as simulated annealing [57], [71] may be useful. Further references on global optimisation include [40], [78], [111].

2. **Construction of conformal diffeomorphisms by composition**

   In Chapter 4 we used a Taylor series expansion to approximate $\psi = \varphi^{-1}$. It is possible to compose a set of Taylor expansions, so that for a set of truncated Taylor series $\psi_j = \sum_{i=0}^{n} a_{ji} z^i$, the conformal diffeomorphism is represented as:

   $$\psi = \psi_1 \circ \psi_2 \circ \psi_3 \circ \ldots \circ \psi_m.$$

   This changes the optimisation problem (because the initial guess for ψ_{i+1} would be the coefficients of ψ_i)and enables more complex functions to be approximated.

It would be interesting to determine experimentally whether or not it improved the computational results.

3. **Stationary vector field**

   In the LDDMM framework of image registration a diffeomorphism is constructed by solving the transport equation, meaning that the velocity field is time-dependent. It is also possible to construct diffeomorphisms using a stationary vector field, by using the exponential map instead of the transport equation. This provides vector fields that are independent of time (hence stationary) [6] and is computationally far cheaper [50]. The Stationary Vector Field (SVF) method has been used to perform diffeomorphic image registration of medical images in [49], [106] and [8]. An added benefit of this approach is that the parameters lend themselves to statistical analysis far more than those of LDDMM registrations [6]. It would be possible to use the SVF approach to construct conformal diffeomorphisms along the lines of Algorithm 3 that was presented in Section 4.1.1.

4. **Investigation with other objective functions**

   Our algorithms are based on the minimisation of the sum-of-squares difference between the images. This is a useful objective function for cases where the images are of the same modality, so that the regions where the image matches have the same intensity. However, in images where the lighting changes between different photographs, and more particularly in medical images, where different imaging modalities (such as CT and MRI) are used, objective functions that do not require intensity matches would be useful. Examples include *cross-correlation*, *Fourier transform* as referred to by Brown in [21], *Mutual Information* in [59] and [29], or *normalised gradient field* in [44]. Modifying the control point method to use these functions instead of the least-squares optimisation would enable the method to be used on these types of images.

5. **Study of Human growth**

   As was mentioned above, the experiment reported in Chapter 5 suggests that the growth of human skulls could follow a conformal path. However, this conclusion is based on the images of five skulls of a single individual. However, possible natural questions are (i) whether or not more than five (for example, 100 or 200) images follow a same conformal path, (ii) instead of checking conformality on images of individuals, it would be useful to compare the images of different people at a given time. We have developed the algorithms that could be used for the investigation of these questions on very large datasets.

6. **Geodesic equation for conformal image registration**

   In [66], the geodesic equation for a Riemannian metric on the space of conformal embeddings was derived. This means that the two point boundary problem for conformal diffeomorphisms can be solved, computing the geodesic path in the group of conformal diffeomorphisms between two images. This can be used as the basis for an image registration method, which would follow the LDDMM approach, finding the solution to:

   $$E(\varphi) = \int_0^1 \|v_t\|_V^2 dt + \|I_1 \circ \varphi^{-1} - I_2\|_{L_2}^2$$

   where φ^{-1} denotes a conformal diffeomorphism, the second term denotes the difference between two images under the L_2 norm and $\|v_t\|$ is a Sobolev norm on the time-dependent velocity field that penalises velocity fields that grow large. For a conformal geodesic path, we can represent φ^{-1} in the form of truncated Taylor series.

Bibliography

[1] K. M. Adal, R. M. Ensing, R. Couvert, P. van Etten, J. P. Martinez, K. A. Vermeer, and L. J. van Vliet. A hierarchical coarse-to-fine approach for fundus image registration. *Biomedical Image registration, Proceedings, vol. 8545, pp: 93-102*, 2014.

[2] M. J. Adrain, G. D. Edgecombe, and B. S. Lieberman. *Fossils, Phylogeny, and Form.* Springer-Verlag, 2001.

[3] S. Aggarwal. Principles of Remote Sensing. *in Satellite Remote Sensing and GIS Applications in Agricultural Meteorology, World Meteorological Organisation, pp: 23-38*, 2004.

[4] L. V. Ahlfors. *Complex Analysis.* USA: McGraw-Hill Book Company, 1966.

[5] J. W. Anderson. *Hyperbolic Geometry.* New York: Springer-Verlag, 1999.

[6] V. Arsigny, O. Commowick, X. Pennec, and N. Ayache. A log-Euclidean framework for statistics on diffeomorphisms. *Medical Image Computing and Computer-Assisted Intervention – MICCAI, pp: 924–931*, 2006.

[7] W. Arthur. D'Arcy Thompson and the theory of transformations. *Nature Reviews Genetics, vol. 7, pp: 401-406*, 2006.

[8] J. Ashburner. A fast diffeomorphic image registration algorithm. *NeuroImage, vol. 38, pp: 95-113*, 2007.

[9] C. Atzberger. Advances in remote sensing of agriculture: Context description, existing operational monitoring systems and major information needs. *Remote Sensing, vol. 5, pp: 949-981*, 2013.

[10] G. Bartoli. Image registration techniques: A comprehensive survey. *Universitá degli Studi di Siena Visual Information Processing and Protection Group.*, 2007.

[11] M. F. Beg, M. I. Miller, A. Trouvé, and L. Younes. Computational anatomy: Computing metrics on anatomical shapes. *Proceedings IEEE International Symposium on Biomedical Imaging, pp: 341-344*, 2002.

[12] M. F. Beg, M. I. Miller, A. Trouvé, and L. Younes. Computing large deformation metric mapping via geodesic flows of diffeomorphisms. *International Journal of Computer Vision, vol. 61, pp: 139–157*, 2005.

[13] D. P. Bertsekas. Projected Newton methods for optimization problems with simple constraints. *Journal on Control and Optimization, vol. 20, pp: 221-246*, 1982.

[14] A. I. Bobenko, C. Mercat, and Y. B. Suris. Linear and nonlinear theories of discrete analytic functions. Integrable structure and isomonodromic Green's function. *Journal für die reine und angewandte Mathematik, vol. 583, pp: 117-161*, 2005.

[15] A. J. Booker, J. E. Dennis, P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset. A rigorous framework for optimization of expensive functions by surrogates. *Structural Optimization, vol. 17, pp: 1-13*, 1999.

[16] F. L. Bookstein. *The Measurement of Biological Shape and Shape Change.* New York: Springer-Verlag, 1978.

[17] F. L. Bookstein. Foundations of morphometrics. *Annu. Rev. Ecol. Syst, vol. 13, pp: 451-470*, 1982.

[18] F. L. Bookstein. *Morphometric tools for Landmark Data: Geometry and Biology.* Cambridge University Press, 1991.

[19] L. Bottou. Large-scale machine learning with stochastic gradient descent. *in Proceedings of COMPSTAT'2010, Physica–Verlag HD, pp: 177-186*, 2010.

[20] J. M. Brown, A. Naylor, A. Filer, and E. Claridge. 3D articulated registration of the mouse hind limb for bone morphometric analysis in rheumatoid arthritis. *in International Workshop on Biomedical Image Registration. Springer International Publishing, pp: 41-50*, 2014.

[21] L.G. Brown. A survey of image registration techniques. *ACM Computing Surveys, vol. 24, No. 4, pp: 325-376*, 1992.

[22] V. V. Bunak. Changes in the relative length of human extremity skeleton segments during the growth period. *Izv. Akad. Nauk RSFSR 84. pp: 33-45*, 1957.

[23] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. *in Proceedings of the 22nd international conference on Machine learning, ACM, pp: 89-96*, 2005.

[24] L. Cattel, J. A. Schnabel, J. Declereck, and C. Hutton. Combined PET-MR brain registration to discriminate between Alzheimer's disease and healthy controls. *in Proceeding of Biomedical Image registration, vol. 8545, pp: 134-143*, 2014.

[25] G. Christensen, R. D. Rabbit, and M. I. Miller. Deformable templates using large deformation kinematics. *IEEE Transactions on Image Processing, 5(10), pp:1437–1447*, 1996.

[26] P. E. Clark and M. Rilee. *Remote Sensing Tools for Exploration.* Springer-Verlag, 2010.

[27] I. Cohen and G. Medioni. Detecting and tracking moving objects for video surveillance. *IEEE Conference on Computer Vision and Pattern Recognition, pp: 319–325*, 1999.

[28] J. B. Conway. *Functions of One Complex Variable II.* Springer-Verlag, 1995.

[29] E. D'Agostino, J. Modersitzki, F. Maes, D. Vandermeulen, B. Fischer, and P. Suetens. Free-form registration using mutual information and curvature regularization. *International Workshop on Biomedical Image Registration. Springer Berlin Heidelberg, pp: 11–20*, 2003.

[30] C. P. Dalmiya and V. S. Dharun. A survey of registration techniques in remote sensing images. *Indian Journal of Science and Technology, vol. 8, pp: 1-5*, 2015.

[31] Binary Shape Databases. Computer vision @ lems, available online. `https://vision.lems.brown.edu/content/available-software-and-databases`.

[32] S. Dawn, V. Saxena, and B. Sharma. Remote sensing image registration techniques: A survey. *Image and Signal Processing, vol. 6134, pp: 103-112*, 2010.

[33] I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis.* Chichester: John Wiley & Sons, 1998.

[34] P. Dupuis, U. Grenander, and M. Miller. Variational problems on flows of diffeomorphisms for image matching. *Quarterly of Applied Mathematics, vol. 56, pp: 587–600*, 1997.

[35] C. Echeverría, A. Newton, L. Nahuelhual, D. Coomes, and J. María. How landscapes change: Integration of spatial patterns and human processes in temperate landscapes of southern Chile. *Applied Geography, vol. 32, pp: 822-831*, 2012.

[36] A. M. T. Elewa. *Morphometrics, Applications in Biology and Paleontology.* New York: Springer-Verlag, 2004.

[37] P. T. Fletcher. Geodesic regression and the theory of least squares on Riemannian manifolds. *Springer: International Journal of Computer Vision, vol. 105, pp: 171-185*, 2013.

[38] L. M. G. Fonseca and M. H. M. Costa. Automatic registration of satellite images. *Brazilian symposium on Graphic Computation and Image Processing, vol. 10, pp: 219-226*, 1997.

[39] C. Frederick and E. L. Schwartz. Conformal image warping. *IEEE Computer Graphics and Applications, vol. 10, pp: 54-61*, 1990.

[40] J. Garloff and A. P. Smith. A comparison of methods for the computation of affine lower bound functions for polynomials. *Global Optimization and Constraint Satisfaction, vol. 3478, pp: 71-85*, 2003.

[41] G. V. Gerganov, K. K. Mitev, and I. Kawrakow. Iterative non-rigid image registration based on Möbius transformations. *IEEE Nuclear Science symposium conference*, 2011.

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

[43] J. Glaunés, M. Vaillant, and M. I. Miller. Landmark matching via large deformation diffeomorphisms on the sphere. *Journal of Mathematical Imaging and Vision, vol. 20, pp: 179–200*, 2004.

[44] A. A. Goshtasby. *Image Registration Principles, Tools and Methods.* London: Springer-Verlag, 2012.

[45] U. Grenander and M. I. Miller. Computational anatomy: An emerging discipline. *Quarterly of Applied Mathematics, vol. 56, pp: 617–694*, 1998.

[46] A. Gupta, M. Escolar, C. Dietrich, J. Gilmore, G. Gerig, and M. Styner. 3D tensor normalization for improved accuracy in DTI tensor registration methods. *International Workshop on Biomedical Image Registration. Springer Berlin Heidelberg, pp: 170-179*, 2012.

[47] E. Haber and J. Modersitzki. Numerical methods for volume preserving image registration. *Inverse Problems, vol. 20, pp: 1621-1638*, 2004.

[48] R. A. Heckemann, J. V. Hajnal, P. Aljabar, D. Rueckert, and A. Hammers. Automatic anatomical brain MRI segmentation combining label propagation and decision fusion. *NeuroImage, vol. 33, pp: 115-126*, 2006.

[49] M. Hernandez, M. N. Bossa, and S. Olmos. Registration of anatomical images using paths of diffeomorphisms parameterized with stationary vector field flows. *International Journal of Computer Vision, vol. 85, pp: 291-306*, 2009.

[50] M. Hernandez, S. Olmos, and X. Pennec. Comparing algorithms for diffeomorphic registration: Stationary LDDMM and diffeomorphic demons. *2nd MICCAI Workshop on Mathematical Foundations of Computational Anatomy, pp: 24-35*, 2008.

[51] M. Jenkinson, P. Bannister, M. Brady, and S. Smith. Improved optimization for the robust and accurate linear registration and motion correction of brain images. *NeuroImage, vol. 17, pp: 825-841*, 2002.

[52] M. Jenkinson and S. Smith. A global optimisation method for robust affine registration of brain images. *Medical Image Analysis, vol. 5, pp: 143-156*, 2001.

[53] S. Jingyong, S. Kurtek, E. Klassen, and A. Srivastava. Statistical analysis of trajectories on Riemannian manifolds: bird migration, hurricane tracking and video surveillance. *Annals of Applied Statistics, vol. 8, pp: 530-552.*, 2014.

[54] S. C. Joshi and M. I. Miller. Landmark matching via large deformation diffeomorphisms. *IEEE Transaction on Image Processing, vol. 9, No. 8, pp: 1357-1370.*, 2000.

[55] H. B. Keller. *Lectures on numerical methods in bifurcation problems.* Springer-Verlag, 1986.

[56] C. T. Kelley. *Iterative Methods for Optimization.* SIAM (Society for Industrial and Applied Mathematics), 1999.

[57] J. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science, vol. 220, pp: 671-680*, 1983.

[58] K. Knopp. *Theory of Functions Parts I and II.* New York: Dover, 1996.

[59] W. Kosiński, P. Michalak, and P. Gut. Robust image registration based on mutual information measure. *Journal of Signal and Information Processing, vol. 3, pp: 175-178*, 2012.

[60] S. G. Krantz. *Handbook of Complex Variables.* Springer, 1999.

[61] S. G. Krantz. *Geometric Function Theory: explorations in complex analysis.* Boston: Birkhäuser, 2006.

[62] G. V. Laurin, V. Liesenberg, Q. Chen, L. Guerriero, F. D. Frate, A. Bartolini, D. Coomes, B. Wilebore, J. Lindsell, and R. Valentini. Optical and SAR sensor synergies for forest and land cover mapping in a tropical site in west africa. *International Journal of Applied Earth observation and Geoinformation, vol. 21, pp: 7-16*, 2013.

[63] S. R. Lele and J. T. Richtsmeier. *An Invariant Approach to Statistical Analysis of Shapes.* USA: Chapman and Hall/CRC, 2001.

[64] L. F. Marcus, E. Bello, and A. García-Valdecasas. *Contribuitions to Morphometrics.* Editorial CSIC-CSIC Press, 1993.

[65] L. F. Marcus, M. Corti, A. Loy, G. J. P. Naylor, and D. E. Slice. *Advances in Morphometrics.* Springer-Verlag, 1996.

[66] S. Marsland, R. I. McLachlan, K. Modin, and M. Perlmutter. Geodesic warps by conformal mappings. *International Jounal of Computer Vision, vol. 105, pp: 144-154*, 2013.

[67] Mathworks. Available online. `https://au.mathworks.com/help/matlab/ref/nan.html`, 2016.

[68] Mathworks. Available online. `https://au.mathworks.com/help/images/ref/imfilter.html`, 2016.

[69] Mathworks. Available online. `https://au.mathworks.com/help/images/ref/imfilter.html`, 2016.

[70] E. Meijering. A chronology of interpolation: from ancient astronomy to modern signal and image processing. *Proceedings of the IEEE, vol. 90, pp: 319–342*, 2002.

[71] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, and A. H. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics, vol. 21, pp: 1087-1092*, 1953.

[72] P. W. Michor and D. Mumford. An overview of the Riemannian metrics on spaces of curves using the hamiltonian approach. *Applied and Computational Harmonic Analysis, vol. 23, pp: 74-113*, 2007.

[73] J. W. Milnor. *The Geometry of Growth and Form.* A lecture at The Institute for Advanced Study, Princeton, New Jersey. Available online at. `http://www.math.stonybrook.edu/~jack/`, 2010.

[74] J. Modersitzki. *FAIR: Flexible Algorithms for Image Registration.* Philadelphia: Society of Industrial and Applied Mathematics (SIAM), 2009.

[75] J. Modersitzki and B. Fischer. Optimal image registration with a guaranteed one-to-one point match. *Bildverarbeitung für die Medizin, pp: 1-5*, 2003.

[76] J. L. Moigne, N. S. Netanyahu, and R. D. Eastman. *Image Registration for Remote Sensing.* Cambridge University Press, 2011.

[77] N. Nkanza. *Image Registration and its application to computer Vision: Mosaicing and Independent Motion detection.* MSc Thesis, Department of Electrical Engineering, University of Cape Town., 2005.

[78] P. M. Pardalos and H. E. Romeijn. *Handbook of global optimization.* Kluwer Academic Publishers., 2002.

[79] B. M. Patten. *The Human Embryology (in Russian).* Medgiz, Moscow, 1959.

[80] X. Pennec. Intrinsic statistics on Riemannian manifolds: Basic tools for geometric measurements. *Journal of Mathematical Imaging and Vision, vol. 25, pp: 127-154*, 2006.

[81] S. V. Petukhov. Non-Euclidean geometries and algorithms of living bodies. *Computers and Mathematics with Applications, vol. 17, pp: 505-534*, 1989.

[82] P. S. Reel, L. S. Dooley, K. C. P. Wong, and A. Börner. Fast EM principal component analysis image registration using neighbourhood pixel connectivity. *in International Conference on Computer Analysis of Images and Patterns, pp: 270-277*, 2013.

[83] R. A. Reyment. *Multidimensional Paleobiology.* New York: Pergamon Press, 1991.

[84] F. J. Rohlf and L. F. Marcus. A revolution in morphometrics. *Trends in Ecology & Evolution, vol. 8, pp: 129-132*, 1993.

[85] D. G. Rokhlin. *X-ray Osteology and X-ray Anthropology.* Biommedgiz, Moscow, 1936.

[86] V. S. Roshni and K. Revathy. Using mutual information and cross correlation as metrics for registration of images. *Journal of Theoretical and Applied Information Technology, pp: 474–481*, 2008.

[87] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, M. O. Leach, and D. J. Hawkes. Nonrigid registration using free-form deformations: Application to breast MR images. *IEEE Transactions on Medical Imaging , vol. 18, pp: 712-721*, 1999.

[88] L. Ruthotto, E. Hodneland, and J. Modersitzki. Registration of dynamic contrast enhanced MRI with local rigidity constraint. *Biomedical Image registration, Proceedings, pp: 190-198*, 2012.

[89] T. Sabisch, A. Ferguson, and H. Bolouri. Automatic registration of complex images using a self organizing neural system. *IEEE International Joint Conference on Neural Networks*, 1997.

[90] S. Saxena and R. K. Singh. A survey of recent and classical image registration methods. *International Journal of Signal Processing, Image Processing and Pattern Recognition, vol. 7, No. 4, pp: 167-176*, 2014.

[91] M. Schmolesky. Available online. `http://webvision.med.utah.edu/book/part-ix-psychophysics-of-vision/the-primary-visual-cortex/`.

[92] C. E. Shannon. A mathematical theory of communication. *Bell Lab Technical Journal, vol. 27, pp: 623–656*, 1948.

[93] D. E. Slice. *Modern Morphometrics in Physical Anthropology.* New York: Kluwer Academic/ Plenum Publishers, 2005.

[94] C. Small. *The Statistical theory of Shape.* New York: Springer, 1996.

[95] L. I. Smith. *A tutorial on principal component analysis.* USA: Cornell University, vol. 51, 2002.

[96] S. Smith, M. Jenkinson, M. W. Woolrich, C. F. Beckmann, T. E. J. Behrens, H. Johansen-Berg, P. R. Bannister, M. D. Luca, I. Drobnjak, D. E. Flitney, R. K. Niazy, J. Saunders, J. Vickers, Y. Zhang, N. D. Stefano, J. M. Brady, and P. M. Matthews. Advances in functional and structural MR image analysis and implementation as FSL. *NeuroImage, vol. 23, pp: S208-S219*, 2004.

[97] S. Sommer, F. Lauze, S. Hauberg, and M. Nielsen. Manifold valued statistics, exact principal geodesic analysis and the effect of linear approximations. *European Conference on Computer Vision. Springer Berlin Heidelberg*, 2010.

[98] The Electromagnetic Spectrum. Available online. `http://imagine.gsfc.nasa.gov/science/toolbox/emspectrum1.html`.

[99] Smoothing Techniques. Available online. `http://www.solver.com/smoothing-techniques`.

[100] G. Theodore. *Complex Analysis.* Springer-Verlag, 2001.

[101] D. W. Thompson. *On Growth and Form.* Cambridge Univ. Press, 1942.

[102] A. Trouvé. An infinite dimensional group approach for physics based model in pattern recognition. Preprint. 1995.

[103] A. Trouvé. Diffeomorphisms groups and pattern matching in image analysis. *International Journal of Computer Vision, vol. 28, pp 213–221*, 1998.

[104] M. Uehara. Pictures by conformal mapping. *FORMA, vol. 16, pp: 1-15*, 2001.

[105] F. van der Meer, H. M. A. van der Werff, F. J. A. van Ruitenbeek, C. A. Hecker, Bakker. W. H., M. F. Noomen, M. van der Meijde, E. J. M. Carranza, J. B. de Smeth, and T. Woldai. Multi-and hyperspectral geologic remote sensing: A review. *International Journal of Applied Earth observation and Geoinformation, vol. 14, pp: 112-128*, 2012.

[106] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache. Non-parametric diffeomorphic image registration with the demons algorithm. *International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer Berlin Heidelberg*, 2007.

[107] J. L. Walsh. History of the Riemann mapping theorem. *The American Mathematical Monthly, vol. 80, pp: 270-276*, 1973.

[108] M. P. Wand and M. C. Jones. *Kernel Smoothing.* London: Chapman & Hall, 1995.

[109] Z. Wang and H. Wang. Image smoothing with generalized random walks: Algorithms and applications. *Applied Soft Computing, vol. 46, pp: 792-804*, 2016.

[110] H. J. Weber and G. B. Arfken. *Essential Mathematical Methods for Physicists.* Academic Press, 2003.

[111] T. Weise. *Global optimization algorithms-theory and application.* self-published, available online as e-book at. `http://www.it-weise.de/projects/book.pdf`, 2009.

[112] E. T. Whittaker and G. Robinson. *The Calculus of Observations: A Treatise on Numerical Mathematics.* New York: Dover, 1924.

[113] Y. Xiong, K. Zhu, D. Lin, and X. Tang. Recognize complex events from static images by fusing deep channels. *Computer Vision and Pattern Recognition (CVPR), pp: 1600-1609*, 2015.

[114] L. Younes. *Shapes and Diffeomorphisms.* Heidelberg: Springer-Verlag, 2010.

[115] L. Younes, P. W. Michor, J. Shah, and D. Mumford. A metric on shape space with explicit geodesics. *Rend. Lincei Mat. Appl., vol. 19, pp: 25-57*, 2008.

[116] M. L Zelditch, D. L Swiderski, and H. D. Sheets. *Geometric Morphometrics for Biologists. Second edition.* Elsevier, 2012.

[117] V. A. Zimmer and G. Piella. An Adaptive Multiscale Similarity Measure for Non-rigid Registration. *International Workshop on Biomedical Image Registration. Springer International Publishing*, 2014.

[118] B. Zitová and J. Flusser. Image registration methods: a survey. *Image and Vision Computing, vol. 21, pp: 977-1000*, 2003.