IMAGE REGISTRATION USING FINITE DIMENSIONAL LIE GROUPS

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.

Raziyeh Zarredooghabadi

2016

Contents

List of Tables

List of Figures

Abstract

D'Arcy Thompson was a biologist and mathematician who, in his 1917 book 'On Growth and Form', posited a 'Theory of Transformations', which is based on the observation that a smooth, global transformation of space may be applied to the shape of an organism so that its transformed shape corresponds closely to that of a related organism. Image registration is the computational task of finding such transformations between pairs of images.

In modern applications in areas such as medical imaging, the transformations are often chosen from the infinite-dimensional diffeomorphism group. However, this differs from Thompson's approach where the groups are chosen to be as simple as possible, and are generally finite-dimensional. The main exception to this is the similarity group of translation, rotation, and scaling, which is used to pre-align images. In this thesis the set of planar Lie groups are investigated and applied to image registration of the types of images that Thompson considered. As these groups are smaller, successful registration in these groups provides more specific information about the relationship between the images than diffeomorphic registration does, as well as providing faster implementations. We build a lattice of the Lie groups showing which are subgroups of each other, and the groups are used to perform image registration by minimizing the L^2-norm of the difference between the group-transformed source image and the target image. A robust, practical, and efficient algorithm for registration in Lie groups is developed and tested on a variety of image types.

Each successful registration returns a point in a Lie group. Given several related images (such as the hooves of several animals) it is possible to find smooth curves that pass through the Lie group elements used to relate the various images. These curves can then be employed to interpolate points between the set of images or to extrapolate to new images that have not been seen before. We discuss the mathematics behind this and demonstrate it on the images that Thompson used, as well as other datasets of interest.

Finally, we consider using a sequence of the planar Lie groups to perform registration, with the output from one group being used as the input to the next. We call this multi-registration, and have identified two types: where the smallest group is a subgroup of the next smallest, and so on up a chain, and where the groups are not directly related, i.e., separated on the lattice. We demonstrate experimentally that multi-registration can provide more information about the relationship between images than simple registration. In addition, we show that transformations that cannot be obtained by a single registration in any of the groups considered can be successfully reached.

Acknowledgements

Firstly, I would like to express my sincere appreciation to my supervisors, Prof. Stephen Marsland and Prof. Robert McLachlan for their constant support, patience, motivation, and immense knowledge. Their guidance helped me to achieve my PhD project and to write my thesis, without their help it would have been impossible to accomplish my PhD. Their advice on the completion of my dissertation has taught me numerous lessons and insight about academic research in general.

I would like to thank the Marsden Fund of the Royal Society of New Zealand; without their financial support I could not have achieved my PhD. I would like to thank the administration staff of the School of Engineering and Advanced Technology (SEAT), the Massey Residential Services Office, the Graduate Research School, the International Office, the Medical Centre, and the Massey library staff for their immense support during my Ph.D.

Last but not the least, I would like to thank my family: my father, my mother, my sisters for supporting me spiritually throughout my PhD journey, my beloved husband for his immense patience, encouragement, understanding me, taking care of me, and his spiritual support.

Chapter 1

Introduction

Natural scientists classify organisms by their appearance and structure, known as form. The form of an organism is a characteristic of the organism that remains unchanged during translation and rotation [37]. Biologists study the form of an organism to understand the reasons that underlie the variations of the organism over various timescales, such as growth, disease or evolution. These variations are measured and quantified, which may help us to discover processes of biological phenomena that may not be identifiable from casual observation [11].

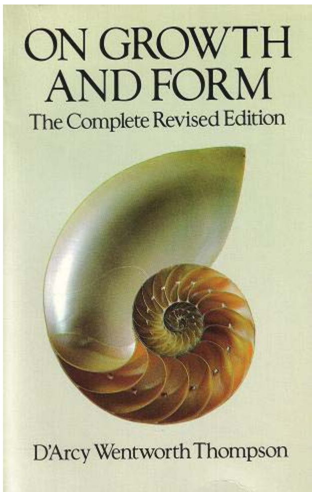
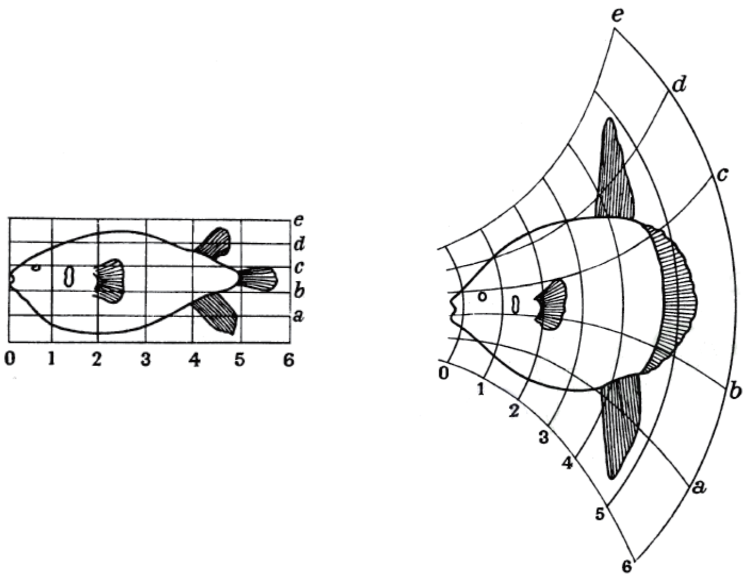


Figure 1.1: On Growth and Form

The quantitative study of form is known as *morphometrics* [11]. The first attempt to quantitatively study form was published by D'Arcy Thompson in his landmark book 'On Growth and Form' [67]. D'Arcy Thompson was a biologist and mathematician; he combined mathematics and biology into a significant work, his '*Theory of Transformations*', given in the last chapter of his book. The *Theory of Transformations* is based on the observation that a smooth, global transformation of space may be applied to the shape of an organism such that its transformation corresponds closely to that of other related organisms. For example, he showed that there is a transformation that transfers the *Diodon porcupine* fish shown in Figure 1.2a to *Orthagoriscus mola* shown in Figure 1.2b.



(a) *Diodon porcupine* (b) *Orthagoriscus mola*

Figure 1.2: *Diodon porcupine* fish is transferred to *Orthagoriscus mola* by deforming the grid isogonally. Taken from [67].

Thompson compared organisms belong to the same zoological taxa, which he called *related forms*. In the comparison of related forms, he said that most of the parts exhibit differences, which he called *independent variants*. For example, in the evolution of fish, the fish has several parts: head, body, tail, fin, etc., where each part exhibits some differences between the fish. Rather than focusing on individual variants, for which Thompson points out that the required transformation will be very localised and complex, he compares the entire body of the related forms, dealing with the independent variants only implicitly.

In fact, the idea of Thompson is to look at the global variation between forms. He said that comparing the entire body of forms without regard to the independent variants will guide us to see a **simple transformation** between related organisms. And finding the simple transformation will furnish us to some guidance as to the 'law of growth' or play of forces.

One modern method to compare related forms is *image registration*, which aligns two or more images so that their appearance matches as closely as possible. The idea of D'Arcy Thompson motivates us to study the shape of related forms using the modern tools of image registration. This research aims to find the simple transformations between the related forms given in Thompson's book mathematically. All the examples that Thompson gave in his book [67] are two dimensional representation of the organisms. Therefore, in this research we consider the 2D images. In the following section we give a review of image registration.

1.1 Review of Image Registration

The primary task in image registration is to find a geometric transformation between a pair of images, the source and target, so that they match as closely as possible [48]. Image registration has been applied in four major research areas: **Computer Vision**—including object recognition, shape reconstruction, motion tracking, and character recognition [6, 22, 31, 58]; **Remote Sensing**—where images are taken from a terrain by a sensor mounted on aircraft or satellite for different tasks, such as geology, oceanography, oil and mineral exploration [18, 33, 66]; **Medical Image Analysis**—used in diagnostic medical imaging, such as disease localization and tumour detection, and in biomedical research, such as in imaging blood cells [21, 40, 41, 46, 65, 70]; and **Morphometrics**—studying the form of an organism to understand the impact of growth, disease or evolution on the shape of the organism; for example, analysing the fossil records of organisms [11, 37].

In the mathematical setting, an image is considered as a mapping from a domain into the real numbers. The domain of the image is denoted by $\Omega \subset \mathbb{R}^d$, where d is the dimension of the image. In this research, we only deal with two-dimensional grey-scale images. A $2D$ grey-scale image I is defined as follows:

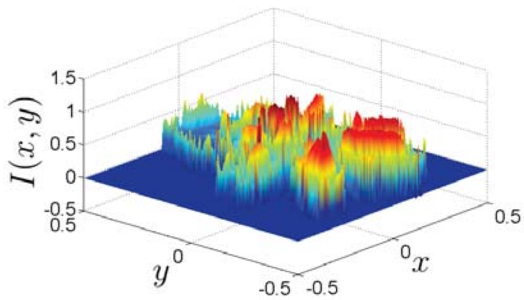
$$I : \Omega \to [0, 1]$$

where $\Omega \subseteq \mathbb{R}^2$ is the domain of the image and to each point $(x, y) \in \Omega$ an intensity $I(x, y) \in [0, 1]$ is assigned. Figure 1.3 shows a functional view of an image.

Of relevance to this thesis are two forms of registration:

Registration based on landmarks: Landmark-based registration is the general representative of feature-based schemes. In the landmark-based method, salient features of the images are extracted. Notable features could be: significant regions

(a)



(b)

Figure 1.3: (a) An image of a plant, (b) a functional view of the image, the z-axis is the range of the image (intensities), the domain of the image is in the xy-plane.

(a)

(b)

(c)

Figure 1.4: (a) An office image, (b) lines in the edge of image are detected, (c) corners are detected. Figures are taken from [27].

(forest, building, lakes), lines (roads, rivers, boundaries of an area) or points (corners of the region, line intersections, points of curves with high curvature) [76]. Figure 1.4 shows an office picture where the points and lines are detected. Distinctive features can be detected manually or automatically; there are various attempts to detect them automatically, see [55]. The challenge in this form of registration is to establish points that match in the two images. This is known as the correspondence problem.

In this method, the features of images are expressed by points. For example, a region is represented by its centre of gravity, a line by points at the ends or middle of the line. Therefore, corresponding representative points must be selected in both the source and target images. The idea is to transform the selected points in the source such that they become as close as possible to the corresponding points in the target. For instance, in $2D$ images, let $\boldsymbol{x}_i = (x_i^1, x_i^2)$, $i = 1, 2, \ldots, n$ be the selected points in the source image, and $\boldsymbol{y}_i = (y_i^1, y_i^2)$ be the corresponding points in the target image. Then landmark registration is:

$$\min_{\psi \in G} \|\psi(\boldsymbol{x}_i) - \boldsymbol{y}_i\|,$$

where $\|\cdot\|$ can be the L^2 distance between the sets of points, and G is a set of transformations. Then the transformation obtained from the landmark registration is applied to the source image. In the following it will be explained how an image is transformed.

Registration based on intensity: In contrast, intensity-based methods deal with the intensities of the pixels in the entire image, and no salient features are detected. The best transformation is found by minimising an intensity-based distance function between the target image and the transformed source image.

Therefore, in registration based on intensity, one image (the source) needs to be transformed to match the other image (the target). Let $\varphi \colon \Omega \to \mathbb{R}^2$ be a planar transformation. We consider how this transformation transforms an image I_0 to a new image I_1 when the pixel values of I_0 are unchanged but merely carried into new locations by φ. In order for this process to uniquely define a transformed image I_1 for all images I_0, it is necessary that φ be 1–1, i.e., it must be invertible on its range. Let $(x', y') = \varphi(x, y)$. Then the value of I_1 at (x', y') is equal to pullback of the value of I_0 at (x, y):

$$I_1(x', y') = I_0(x, y) = I_0 \circ \varphi^{-1}(x', y').$$

See Figure 1.5 for an example of the scaling transformation $\varphi(x, y) = \frac{1}{2}(x, y)$. Therefore, $I_1 = I_0 \circ \varphi^{-1}$.

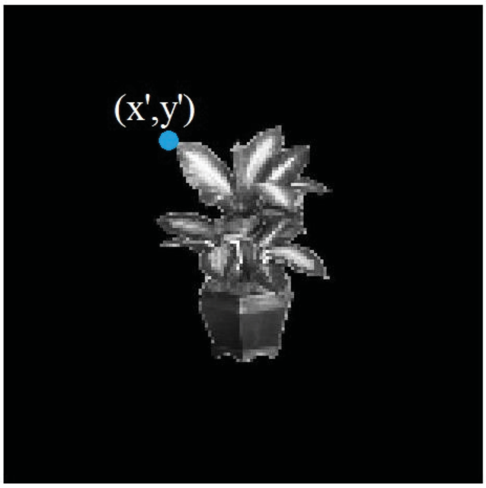
Let I be the source and J the target. Then, in the mathematical setting, intensity-based registration can be written as:

$$\min_{\varphi \in G} E(\varphi) = \min_{\varphi \in G} \|I \circ \varphi^{-1} - J\|, \tag{1.1}$$

where G is a transformation set, and $\|\cdot\|$ is a distance function.

There are three key components in image registration [63]: a distance function between the images that is to be minimised, a set of transformations that can be applied to the images, and an optimisation function that finds the transformation parameters that minimise the distance function. Each of these separate components are now described.

(a) I_0



(b) I_1

Figure 1.5: (a) I_0 is scaled by $\varphi(x, y) = \frac{1}{2}(x, y)$ to (b) I_1, and $I_1(x', y') = I_0 \circ \varphi^{-1}(x', y')$, where $(x', y') = \varphi(x, y)$.

1.1.1 Intensity-Based Distance Functions

There are a variety of types of intensity-based distance functions that can be applied depending on the type of image [49]. A prime example is L^2 distance:

$$\|I \circ \varphi^{-1} - J\|_2 = \left(\int_\Omega |I \circ \varphi^{-1}(x, y) - J(x, y)|^2 dxdy \right)^{\frac{1}{2}},$$

where Ω is the domain of the source. This function compares the intensity of the images. A disadvantage of this distance function is that matching objects in the images need to have similar intensities [49]. For example, images with different illuminations do not have similar intensities.

Cross-correlation (or more commonly, normalized cross-correlation) is a distance function that can be employed when images have similar intensities, but different illuminations, normally a monotonic change in the intensities [49]. If I and J are two images with similar intensities but different illumination then for some $\lambda > 0$, $I \circ \varphi^{-1}(x, y) = \lambda J(x, y)$. The cross-correlation of two images I and J is

$$-\int_\Omega \left\langle \frac{I \circ \varphi^{-1}(x, y) - \overline{I \circ \varphi^{-1}}}{\|I \circ \varphi^{-1}(x, y) - \overline{I \circ \varphi^{-1}}\|}, \frac{J(x, y) - \bar{J}}{\|J(x, y) - \bar{J}\|} \right\rangle dxdy.$$

Here $\overline{I \circ \varphi^{-1}}$ and $\bar{J}$ are the mean of the images, and $\langle , \rangle$ is the L^2 inner product. So,

Figure 1.6: Two images with different intensities. (a) Inside the ellipse intensities are zero and outside are 0.5, (b) inside of the ellipse intensities are 0.5 and outside are zero.

cross-correlation is minimum when $I \circ \varphi^{-1}(x, y) = \lambda J(x, y)$. However, the cross-correlation does not work well if objects in the images have different intensity relations to the rest of the image. For example, in the two images given in Figure 1.6 one image has zero intensity inside the ellipse and 0.5 intensity outside the ellipse, and the other one has 0.5 intensity inside and zero outside. Neither the L^2 distance or cross-correlation can tell us when the ellipses match up, because L^2 distance and cross-correlation try to match the similar intensities.

To register the images in Figure 1.6, *mutual information* (MI) can be employed as the distance function. This function was introduced in [72], and it is derived from the theory of information [59]. This function measures the statistical dependency between two images I and J as follows:

$$MI(I, J) = H(I) + H(J) - H(I, J),$$

where $H(I)$ is the entropy of the image intensities (a):

$$H(I) = -\sum_a p_I(a) \ln(p_I(a)),$$

and $p_I(a)$ is the probability distribution of a discrete image, computed as follows:

$$p_I(a) = \frac{N(a)}{S},$$

where $N(a)$ is the number of the pixels which have equal intensity a, and S is the total number of pixels. The entropy is maximised when each pixel has a different intensity:

$$H(I) = -\sum_{i=1}^{S} \left(\frac{1}{S}\right) \ln \left(\frac{1}{S}\right) = \ln(S).$$

The minimum entropy is zero when all pixels have equal intensity. $H(I, J)$ is called the joint entropy of I and J; it is:

$$H(I, J) = -\sum_{a,b} p_{I,J}(a, b) \ln(p_{I,J}(a, b)),$$

where $p_{I,J}$ is the joint probability distribution function of I and J,

$$p_{I,J}(a, b) = \frac{N(a, b)}{S},$$

and

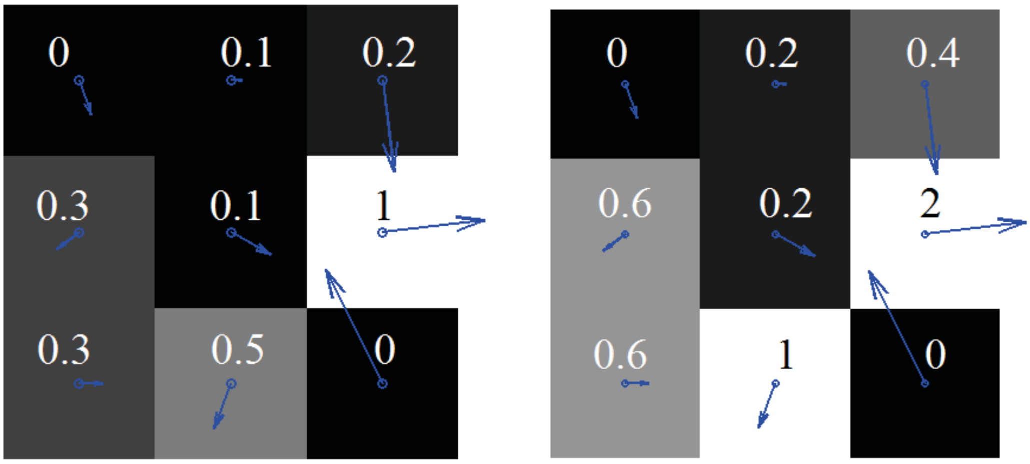
$$N(a, b) = \left|\{(I(i, j), J(i, j)) \mid I(i, j) = a, J(i, j) = b\}\right|.$$

If $I = J$, then $H(I, J) = H(I)$. In fact, the joint entropy is always greater than or equal to the entropy of I and J. So, to register images, it is necessary to minimise their joint entropy, which leads to maximisation of MI. In the example in Figure 1.6, registration with respect to translations and mutual information would lead to perfect alignment of the ellipses.

Normalised Gradient Fields (NGF) is another distance measure that is not sensitive to the change of illumination of intensities. It is as follows:

$$\int_{\Omega} 1 - \left\langle \frac{\nabla I}{\|\nabla I\|}, \frac{\nabla J}{\|\nabla J\|} \right\rangle^2 dxdy,$$

where $\langle , \rangle$ is the inner product. Two discrete images (see Section 3.1.1) are given in Figure 1.7, where $J = 2I$. The gradient fields of them have equal magnitude and the same direction at each point. Therefore, their inner product is 1 at every point and so NGF is zero. There are other distance functions, see [76] for more information, but these are the most commonly used. As we will only use images where the greyscale intensity values are similar, only the L^2 distance will be used in this thesis.



(a) I (b) J

Figure 1.7: The gradient field of two images I and J, where $J = 2I$. The gradient at each point $v(i, j)$ is calculated by *central difference*, i.e. $v(:, j) = 0.5(I(:, j+1) - I(:, j-1))$, $j = 1, 2, 3$, and at the edge it is calculated by *single-sided difference*, i.e. $v(:, N) = I(:, N+1) - I(:, N)$, $N = 3$, [44].

1.1.2 Optimisation

A search strategy to find the optimal transformation, that is the point at which the distance function is a minimum, is needed for image registration. Optimisation starts from an initial guess and progresses until it finds at least a local minimum. There are a variety of methods of optimisation. In this section we only give a brief introduction to some of the optimisation methods; for more sophisticated literature see [24, 34, 51].

Let f be a function of x. A numerical optimisation process to minimise $f(x)$ starts from an initial guess x_0, and generates a sequence of iterations $\{x_k\}_0^n$ such that $f(x_{k+1}) < f(x_k)$, which terminates when there is no more progress or the solution is accurate enough. In deciding how to move from a point x_k to the next iteration x_{k+1}, there are two fundamental methods, *line search* and *trust region*.

In line search, x_{k+1} is computed as follows,

$$x_{k+1} = x_k + \alpha_k p_k,$$

where p_k is the direction to move and α_k is the step length that decides how far to move. The success of this method depends on the choice of both direction and step length.

There are a variety of models for the line search method to choose a direction from x_k to move. One of the methods is *steepest descent*. In this method the direction is:

$$p = -\frac{\nabla f}{\|\nabla f\|}.$$

This is the direction in which f is most rapidly decreasing. For every direction p and step length parameter α, the Taylor expansion of f is,

$$f(x_k + \alpha p) \approx f(x_k) + \alpha p^T \nabla f_k + \frac{1}{2}\alpha^2 p^T H_k p, \tag{1.2}$$

where $p^T \nabla f_k$ is the rate of change of f along the direction p, and is most negative when $p = -\frac{\nabla f}{\|\nabla f\|}$. Here H_k is the Hessian of f at x_k. The advantage of this method is that it only needs to compute ∇f, but it converges slower than the *Newton* method on difficult problems, see [56]. In the *Newton* method the direction is computed as:

$$p_k = -H_k^{-1} \nabla f_k,$$

where it is assumed that the Hessian matrix is positive definite[1]. This direction is derived from the second order Taylor series of f. Taking the derivative of the second order Taylor series of f, given in Equation (1.2), where $\alpha = 1$ with respect to p:

$$\nabla f(x_k) + H_k p,$$

setting the derivative equal to zero, we obtain:

$$p = -H_k^{-1} \nabla f_k.$$

There are other methods such as *conjugate gradient* which are very effective, see [51], however we do not use them, and so do not describe them further.

The basic idea of the trust region method is that the function f is modelled around the current point x_k as a simple function m_k whose behaviour is similar to the function near to the point x_k. Because the model m_k may not be a good approximation of f at x when x is far from x_k, the model m_k is restricted to some region N around x_k. Then a trial step p is computed by minimisation of the model m_k on the region N,

$$\min_p m_k(x_k + p), \tag{1.3}$$

where $x_k + p$ lies inside the trust region N. The current point x_k is updated to $x_k + p$

[1] Positive definite: A positive definite matrix is a symmetric matrix with all positive eigenvalues.

if $f(x_k + p) < f(x_k)$, otherwise it is concluded that the trust region is too large, so the region N is shrunk and m_k is minimized again. The model m_k usually consists of the second order approximation of f. Therefore, the trust region subproblem is typically stated as follows:

$$\min_p(m_k(x_k + p)) = f_k + p^T \nabla f_k + \frac{1}{2} p^T B_k p, \tag{1.4}$$

where f_k is the value of objective function f at x_k, ∇f_k is the gradient of f at x_k and B_k is the Hessian matrix or its approximation.

Trust-region-reflective is an optimisation approach based on the trust region. The minimisation of Equation (1.4) is restricted to a two dimensional subspace of $S \subset N$. Therefore, the solution of Equation (1.4) is trivial, and the dominant work is shifted to the determination of the two dimensional subspace S. Therefore, trust-region-reflective is a faster approach than trust-region, see [13, 17] for details.

Another gradient descent optimisation method is *Levenberg–Marquardt*, which is used specifically for non-linear least squares (L^2 distance) problems. Write the distance function as

$$f(x) = \sum r_i^2(x),$$

where $r_i(x)$ are the residuals which depend on x non-linearly. In this method, it is assumed that the residuals are small and have zero mean. The gradient and the Hessian matrix of f are:

$$\begin{aligned} \nabla f &= 2 \sum r_i(x) \nabla r_i(x) \\ \nabla \nabla^T f &= 2 \sum r_i(x) \nabla^T \nabla r_i(x) + \nabla^T r_i(x) \nabla r_i(x). \end{aligned}$$

If H_k in the Newton method is set as $H_k = 2 \sum \nabla^T r_i(x) \nabla r_i(x)$ then we obtain the method called *Gauss–Newton.* If $H_k = 2 \sum \nabla^T r_i(x) \nabla r_i(x) + \lambda I$, where I is the identity matrix, we obtain the *Levenberg–Marquardt* method. If λ is small then the Levenberg–Marquardt method approximates the Gauss–Newton method, and if λ is large it approximates the steepest descent method.

These optimisation methods find a local minimum depends on the initial value. In registration the aim is to find the global minimum of the distance function. However, it is not guaranteed that the optimizer finds the global minimum. So, this is a significant issue in image registration. We will investigate this issue in Chapter 3.

There is a function in Matlab (2010) software [43] called lsqnonlin which performs least squares optimisation. As we are using the L^2 distance function, we use this code for the minimisation of the distance function. The algorithms that this code employs for the optimisation are trust-region-reflective and Levenberg–Marquardt. We choose to use trust-region-reflective as our optimisation method. Our registration algorithm is explained fully in Chapter 3.

1.1.3 Transformation Sets

Another key component in image registration is the class of transformations from which the optimal transformation is to be selected. The transformations sets may be classified into two categories.

The first category is the finite dimensional sets of transformations, in which a transformation is represented by a finite number of parameters. For example, translation in the plane has two parameters: translation along the x and the y directions.

The second category is the infinite dimensional sets of transformations, in which the possible transformations are represented by arbitrary functions, e.g. diffeomorphisms. In practice, for computational implementations there are a finite set of parameters, see Section 1.2.

Some finite dimensional transformation sets on the plane that have been used in registration are the similarity, rigid, affine, projective, and polynomial transformations [15, 49, 76]. We introduce these important sets of transformations now.

Similarity: One of the most common and simplest set of transformations that is used in image registration is the similarity transformations, which contain rotations, isometric scalings, x-translations and y-translations. The similarity transformation of a point $(x, y) \in \mathbb{R}^2$ is:

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto r \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \end{pmatrix},$$

where r is the scaling parameter, θ is the angle of rotation and t_i, $i = 1, 2$ are translations. This set of transformations is 4 dimensional. Similarities arise, for example, if a pin-hole similarity camera is set up perpendicular to the image plane so that it only rotates about its optical axis, but can be translated in the plane

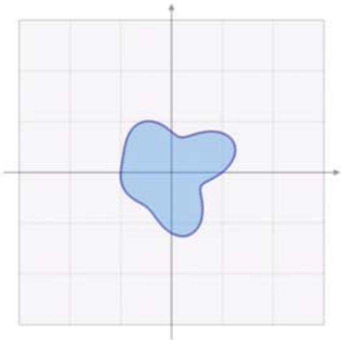


Figure 1.8: A sample image [73].

and the distance to the image plane can change [49]. Such camera motions result in similarity transformations of the acquired image.

A rigid transformation is a particular similarity transformation that allows only rotations and translations:

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}.$$

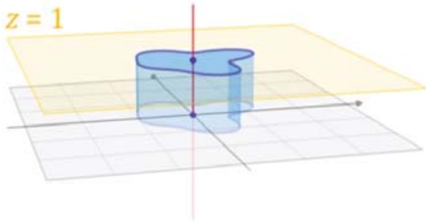
The set of rigid transformations is three dimensional. In image registration applications, a similarity and rigid transformation is used when the camera movement retains the shape of the object.

Affine: Affine transformations are more general than similarity transformations. In addition to rotation, translation and scaling, affine transformations allow for shearing. The affine transformation of a point (x, y) is:

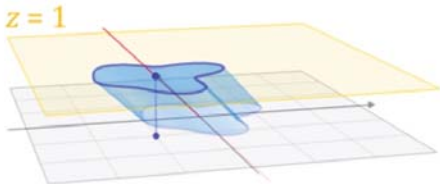
$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}, \quad ad - bc \neq 0.$$

Therefore, the set of affine transformations is 6 dimensional. The most common use of affine transformation in image registration is when images are taken from the same viewing angle, but from different positions. For example, Figure 1.9 shows two photos that are taken from the image given in Figure 1.8, where the camera is a pin-hole affine camera. As can be seen, the viewing angle of cameras is the same, but their positions are different.

Affine transformation is applied in medical applications when the position of a patient in the equipment is not identical each time, but the camera viewing angle is fixed [48].



(a)



(b)

Figure 1.9: The image in Figure 1.8 taken from the same viewing angle (red line) but two different positions of camera [73].

Figure 1.10: A projective photo of a railway. The camera optical axis is not perpendicular to the plane in which the railway is located.

Projective: Figure 1.10 shows a perspective image of a railway. It can be seen that the farther from the camera a point is the more oblique the transformation; the closer it is the more compressed. Projective transformations are applied to rectify the perspective image, such as in aerial photos [15]. Projective transformations are employed in image registration when the optical axis of the camera is not perpendicular to the image. Projective transformation is based on *homogeneous coordinates*, see Section 2.1.9. The projective transformation of $(x, y) \in \mathbb{R}^2$ is:

$$f(x,y) = \left(\frac{a_1x + b_1y + c_1}{a_3x + b_3y + c_3}, \frac{a_2x + b_2y + c_2}{a_3x + b_3y + c_3} \right).$$

Although in this form there are nine free parameters, this set of transformations is eight dimensional. To see this, let $A = \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix}$ be a matrix, where its entries are the parameters of f and $\det(A) = d$. We can scale the transformation by $\sqrt[3]{d}$ and the transformation does not change:

$$f(x,y) = \left(\frac{a_1x+b_1y+c_1}{a_3x+b_3y+c_3}, \frac{a_2x+b_2y+c_2}{a_3x+b_3y+c_3}\right) = \tag{1.5}$$

$$\left(\frac{\frac{a_1}{\sqrt[3]{d}}x+\frac{b_1}{\sqrt[3]{d}}y+\frac{c_1}{\sqrt[3]{d}}}{\frac{a_3}{\sqrt[3]{d}}x+\frac{b_3}{\sqrt[3]{d}}y+\frac{c_3}{\sqrt[3]{d}}}, \frac{\frac{a_2}{\sqrt[3]{d}}x+\frac{b_2}{\sqrt[3]{d}}y+\frac{c_2}{\sqrt[3]{d}}}{\frac{a_3}{\sqrt[3]{d}}x+\frac{b_3}{\sqrt[3]{d}}y+\frac{c_3}{\sqrt[3]{d}}}\right), \tag{1.6}$$

and $\det \begin{pmatrix} \frac{a_1}{\sqrt[3]{d}} & \frac{b_1}{\sqrt[3]{d}} & \frac{c_1}{\sqrt[3]{d}} \\ \frac{a_2}{\sqrt[3]{d}} & \frac{b_2}{\sqrt[3]{d}} & \frac{c_2}{\sqrt[3]{d}} \\ \frac{a_3}{\sqrt[3]{d}} & \frac{b_3}{\sqrt[3]{d}} & \frac{c_3}{\sqrt[3]{d}} \end{pmatrix} = 1$. So, we can suppose that the matrix of parameters of a projective transformation has determinant 1. Now, let $A = \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix}$, such that $\det(A) = 1$; one of the entries of A can be given as a function of other entries, for example $c_3 = \frac{1-(a_3(b_1c_2-c_1b_2)-b_3(a_1c_2-c_1a_2))}{a_1b_2-b_1a_2}$. Therefore, each transformation in the set of projective transformations depends on eight parameters.

Polynomial: If images have some non-linear distortion, then a polynomial of second or third order can be used for the registration.

In image registration we often desire that the transformation set be a group[2] under function composition. Being a group in many cases brings the following benefits [5]:

- Having an identity: having an identity is necessary to match source and target when they are identical.
- Having an inverse: If the source is mapped to the target by a transformation, then the target can be mapped to the source by the inverse of the transformation.
- Composing transformations: This allows us to combine transformations. This property can be used to connect images. A simple example is translations. Three photos of a pencil are taken, see Figure 1.11. Suppose we know the translation between figures 1.11a and 1.11b, and between figures 1.11b and 1.11c. Then we can obtain the translation between Figure 1.11c and 1.11a by composing (adding) two translations: from 1.11a to 1.11b, and from 1.11b to 1.11c.

[2] A group $(G, \circ)$ is a set G, closed under a binary operation $\circ$, such that the following axioms are satisfied [23]:

Associativity of $\circ$: For all $a, b, c \in G$ we have: $(a \circ b) \circ c = a \circ (b \circ c)$.

Identity element e for $\circ$: There is an element in G, such that for all $x \in G$: $x \circ e = e \circ x = x$.

Inverse: Corresponding to each $a \in G$, there exists $a' \in G$, such that: $a \circ a' = a' \circ a = e$.

(a) (b) (c)

Figure 1.11: Three photos of a pencil. The pencil is translated on the plane.

Moreover, let $(G, \circ)$ be a group of bijective transformations on $\Omega \subset \mathbb{R}^2$, and $\Gamma = \{I, I : \Omega \to [0,1]\}$ be an image space. Then the transformation of the image I by φ, which as discussed earlier is given by

$$\varphi \cdot I = I \circ \varphi^{-1}, \tag{1.7}$$

defines a group action on the image space Γ [75]. A *(left) group action* of $(G, \circ)$ on a set X is a function [20, 23]:

$$\begin{aligned} \phi : G \times X &\to X \\ (g, x) &\mapsto \phi(g, x), \end{aligned}$$

which satisfies the following axioms:

- $\phi(e, x) = x$, (e is the identity element of the group G)
- $\phi(g_1 \circ g_2, x) = \phi(g_1, \phi(g_2, x))$, $g_1, g_2 \in G$.

Example 1. Let $G = \left\{ \begin{pmatrix} r & 0 \\ 0 & r \end{pmatrix}, r \in \mathbb{R} \right\}$; $(G, \times)$ forms a group. G acts on $\mathbb{R}^2$ by matrix multiplication as follows:

$$\left(\begin{pmatrix} r & 0 \\ 0 & r \end{pmatrix}, \begin{pmatrix} x \\ y \end{pmatrix} \right) \mapsto \begin{pmatrix} r & 0 \\ 0 & r \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} rx \\ ry \end{pmatrix}.$$

This action scales every point by r. The axioms hold:

1. The identity element of the group is $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, which maps any point $\begin{pmatrix} x \\ y \end{pmatrix}$ to itself:
$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix}$$

2. Let $g_1 = \begin{pmatrix} r_1 & 0 \\ 0 & r_1 \end{pmatrix}$, $g_2 = \begin{pmatrix} r_2 & 0 \\ 0 & r_2 \end{pmatrix}$, $\bar{\boldsymbol{x}} = \begin{pmatrix} x \\ y \end{pmatrix}$, then:

$$g_1 g_2 = \begin{pmatrix} r_1 & 0 \\ 0 & r_1 \end{pmatrix} \begin{pmatrix} r_2 & 0 \\ 0 & r_2 \end{pmatrix} = \begin{pmatrix} r_1 r_2 & 0 \\ 0 & r_1 r_2 \end{pmatrix}$$

and:

$$(g_1 g_2)\bar{\boldsymbol{x}} = \begin{pmatrix} r_1 r_2 x \\ r_1 r_2 y \end{pmatrix} = \begin{pmatrix} r_1 & 0 \\ 0 & r_1 \end{pmatrix} \begin{pmatrix} r_2 x \\ r_2 y \end{pmatrix} = g_1(g_2 \bar{\boldsymbol{x}}).$$

It can be easily seen that Equation (1.7) obeys the axioms:

- $id \cdot I = I \circ id = I$, ($id$ is identity transformation)
- $(\varphi_1 \circ \varphi_2) \cdot I = I \circ (\varphi_1 \circ \varphi_2)^{-1} = I \circ \varphi_2^{-1} \circ \varphi_1^{-1} = \varphi_1 \cdot (I \circ \varphi_2^{-1}) = \varphi_1 \cdot (\varphi_2 \circ I)$

Image registration methodology is influenced by the transformation set. In the study of biological variability, it is common that the transformation is a diffeomorphism (a smooth invertible function). Diffeomorphisms are a natural choice to explore anatomy, since connected and disjoint sets are kept connected and disjoint respectively, and the smoothness of curves and surfaces of the anatomy are preserved [9]. The set of diffeomorphisms forms a group under composition. In the following we give a brief overview of diffeomorphic registration.

1.2 Diffeomorphic Registration

This section concerns the infinite dimensional diffeomorphism group. It is not strictly relevant to this thesis, but the diffeomorphism group is commonly used for image registration. The purpose of this section is to give an overview of diffeomorphic image registration for completeness.

The first attempt at computing a high-dimensional non-rigid registration was given by Broit, Bajcsy and co-workers in [7, 8, 14]. In this setting, the transformation φ is generated by its linear approximation in a neighbourhood of the source, $\varphi(x) = x + u(x)$ or $\varphi^{-1}(x) = x - u(x)$, where $u : \Omega \mapsto \mathbb{R}^2$ is the displacement vector field. The distance function E_2 measures the square of the L_2 difference between the images.

The optimal transformation is the one which minimizes E_2 among all possible solutions and has the highest smoothness, where the smoothness of u is measured by E_1 as follows,

$$E_1(u) = \|Lu\|_2^2,$$

where L is an operator on the space of vector fields. Commonly, L is chosen as $L = (-\alpha\triangle + \gamma)I_{n\times n}$, where $\triangle$ is the Laplacian operator and I is the identity matrix. In the variational setting the optimal vector field is given by the minimisation of

$$\arg\min_u E_1 + \frac{1}{\sigma^2}E_2,$$

where σ is a parameter. This approach is known as small deformation matching [2], because for sufficiently small u, φ is a diffeomorphism. One limitation of this approach is that the transformation does not necessarily lie in the diffeomorphism group for larger u. To overcome this limitation, the large deformation model was developed [19]. In this model, the transformation φ is the endpoint of a path ϕ_t in the space of transformations, where ϕ_t is the flow of time-dependent vector field $v_t : \Omega \to \mathbb{R}^n$, $t \in [0,1]$ and is specified by the ODE, $\dot{\phi}_t = v_t(\phi_t)$, with $\phi_0 = id$ (id is the identity map) and the endpoint $\varphi = \phi_1 = \phi_0 + \int_0^1 v_t(\phi_t)dt$. The large deformation algorithm is given as follows (see [75] for more details):

- *Start with* $\varphi_0 = id$.
- *Solve the evolution equation* :

$$\partial_t\varphi(t,y) = -2\int_\Omega (I\circ\varphi^{-1}(t,x) - J)\nabla I\circ\varphi^{-1}(t,x)K(\varphi(t,y),x)dx.$$

Here K is a *reproducing kernel of Hilbert space* and is commonly chosen to be a Gaussian, i.e. $K(x,y) = \exp(-\alpha\|x-y\|^2)$. This algorithm is also known as greedy image matching.

One of the principal aspects of diffeomorphic image registration is to measure the distance between images in the diffeomorphism group. The diffeomorphism group is also an infinite dimensional manifold and can be equipped with a Riemannian metric. The large diffeomorphic method introduced by [19] connects the source to the target, but the orbit is not the shortest path. A method called Large Deformation Diffeomorphic Metric Mapping (LDDMM) is introduced in [9] such that its solution is similar to the flow of the large deformation model, but in contrast to the large deformation method, the path connecting the source and target is the shortest path. The transformation is determined via the basic variational problem that in the space of smooth velocity vector fields V on domain Ω takes the form:

$$\hat{v} = \arg \min_{v:\dot{\varphi}_t = v_t(\varphi_t)} \left(\int_0^1 \|v_t\|_V^2 dt + \|I_0 \circ \varphi_1^{-1} - I_1\|_2^2 \right), \tag{1.8}$$

where $I_0 \circ \varphi_1^{-1} = I_1$, $\dot{\varphi}_t = v_t(\varphi_t)$ and $\|v_t\|_V = \|Lv\|_2$, where L is a differential operator to enforce the vector field to be smooth. The second term in Equation (1.8) enforces matching of the images with $\|.\|_2^2$, which is the squared-error norm. The length of the shortest path is $\inf \int_0^1 \|v_t\|_V dt$, which defines a metric on the image orbit.

Another method for diffeomorphic image registration is the *Stationary velocity field* (SVF) method. In this setting, the diffeomorphism is parameterized by the one-parameter subgroups generated by stationary velocity fields through the Lie group exponential. In contrast to LDDMM in which the diffeomorphism lies on a geodesic, in the SVF method the diffeomorphism may not lie on a geodesic, because one-parameter subgroups may not be geodesic. (On geodesics, the acceleration is zero.) To measure the acceleration on geodesics an *affine connection* (∇) between the tangent spaces is defined. If the velocity $X = \dot{\gamma}(0)$ is transported along a curve γ by an affine connection parallel, so that $\nabla_{\dot{\gamma}} X = 0$, then γ is a geodesic. An affine connection is called the *Levi-Civita connection* if the parallel transport is geodesic using the Riemannian metric. In [39] they investigated when one-parameter subgroups coincide with Riemannian geodesics. They found that with the *Cartan connection* one-parameter subgroups are Riemannian geodesics, and based on this, they proposed a diffeomorphic registration method.

Image registration with the infinite dimensional diffeomorphism group has been well studied during the last decade. However, relatively little attention has been devoted to image registration in finite dimensional groups other than the similarity group. This is the focus of this thesis. We present the following motivations for the study of image registration by finite dimensional groups.

Thompson's idea of simplest transformations: The remarkable idea of Thompson that simple groups are to be preferred is a strong justification to study finite dimensional groups for image registration.

Few groups: As far as we are aware, only a few finite dimensional groups (rigid, similarity, affine, projective) have been employed in image registration [15, 49, 76]. Are there other groups, either subgroups of the full diffeomorphism group or not, that can be usefully employed in registration?

More information: In the standard approach to diffeomorphic image registration, affine registration is performed before full diffeomorphism, in order to match gross features of the images and get the coordinate frames to line up. However, there may well be information in the affine part (e.g., growth). Are there other groups, either subgroups of the full diffeomorphism group, or not, that can provide useful information?

Occam's razor: Occam's razor states that among competing hypotheses, the one with the fewest assumptions should be selected. By Occam's razor, simpler theories are preferable to more complex ones because they are better testable and falsifiable. Finite dimensional groups are less complicated than infinite dimensional groups, so by Occam's razor, they are preferable.

Faster implementation: Diffeomorphic registration may not be quick enough for some uses, while finite dimensional registration is faster due to having fewer unknowns. Also for this reason, among the finite dimensional groups, groups with lower dimension are preferred for the registration.

1.3 Aims of The Research and Overview of The Thesis

As explained, one of the motivations of this research is the work of D'Arcy Thompson. Thompson found simple transformations between organisms that he presented in his book by $2D$ drawings. Therefore, this thesis is devoted on registration of $2D$ images with finite dimensional groups, where many of the images that we registrar are Thompson's drawings of organisms. Our principal research aims are as follows:

1. Compile a list of finite dimensional groups of transformations that may be useful in image registration. There are three main groups: $PSL(2, \mathbb{C})$, $PSL(3, \mathbb{R})$ and $PSL(2, \mathbb{R}) \times PSL(2, \mathbb{R})$ and their subgroups, which total 19 groups. They will be derived in Chapter 2.

2. Investigate the issues of image registration algorithm. There are different possible methodologies for image registration in finite dimensional groups (e.g. rigid, similarity, affine, projective, etc.). The methodology that is used in this research is image registration based on intensity, where the distance function is L^2 distance of images. However, we found that the naïve image registration algorithm based

on this function failed to find a good match a lot of the time. In Chapter 3, we investigate the issues mathematically and develop an effective image registration algorithm that uses finite dimensional groups.

3. Employ the finite dimensional planar groups given in Chapter 2 in registration. Using finite dimensional groups in image registration gives us some benefits, such as gaining more information about the geometric relationship between images that can be useful for different purposes, and seeing the global variations between images. These benefits are illuminated with many examples, some of which include Thompson's simple transformations between organisms and registrations of a dataset of side views of human skulls. In fact, no one apparently has reproduced Thompson's work. Therefore, the first approximations between the related organisms will be shown as well. Also, we use model selection to show that finite dimensional groups are preferred to infinite dimensional ones, moreover sometimes lower dimensional groups are enough to describe the relationship between the images. Chapter 4 covers these aims.

4. Fit curves in the Lie groups. Given some transformations from the registration of related images, it is possible to find curves that pass through them. In Chapter 5, we demonstrate how to fit such curves in the Lie groups through the transformations that are obtained from the registration. Such curves will give us a better insight about how a natural phenomenon (e.g. growth, evolution, disease) acts through time. This method also can be employed to interpolate or extrapolate between the set of images to get more information (new image) or find lost information (inbetween forms) that may be useful. We employ the method on a dataset of human skulls to find the growth curve, and on a dataset of hoofed mammal feet to find other related feet. Chapter 5 covers these aims.

5. Introduce a novel type of image registration that we call *Multi-registration*, where images are registered with a sequence of groups of transformations. We introduce two cases for the multi-registration: 1) Multi-registration on a chain of groups (groups and subgroups), 2) Multi-registration on a tree of groups (separated groups). Moreover, we show the benefits of multi-registration which are:

    - Multi-registration provides useful information, for example the significance or insignificance of a group of transformations, or the relationship between the images.
    - Multi-registration provides a bigger space of transformations (through composition of groups), which is still finite dimensional and for which the transformations are invertible.

- Multi-registration enables us to find a complex transformation that cannot be obtained by single registration.
- Multi-registration enables us to describe a complex deformation as a composition of simple deformations.
- Multi-registration provides an easier registration process. For example, registration by G_1 followed by registration by G_2 with $G_1 \subset G_2$ can be easier to compute than registration with G_2 alone, as G_1 has fewer parameters and yields a lower-dimensional optimisation problem.

Chapter 6 covers these aims.

Chapter 2

Finite Dimensional Planar Lie Groups

In this chapter, the finite dimension planar Lie groups are described. They will be used for registration in Chapters 4 and 6. We will focus on *Lie groups*, which bring some benefits for this research. First, a global group can be replaced by its local linearization, which is easier to work with. Second, a metric can be defined on a Lie group, which enables us to measure the distance between images.

This chapter has four main sections. Section 2.1 gives an introduction to Lie groups and their properties. In Section 2.2 the planar Lie groups are derived from their infinitesimal generators, and in Section 2.3 the lattice of planar Lie groups is given, where the lattice shows the group and subgroup relationship. As mentioned, one of the goals of this research is to explore the benefits of using finite dimensional groups for registration. Therefore we need to have knowledge about the geometric properties of the action of these groups on the plane to extract useful information; thus Section 2.4 explains the geometric properties of the action of groups on the plane.

2.1 Introduction to Lie Groups

A *Lie group* is a *smooth manifold* that is also a group and has differentiable group operations. A *manifold* is a topological space that resembles Euclidean space around each point. More precisely, a manifold is defined as follows [36]:

Definition 2.1.1. A topological space (M, τ) with topology τ is a *manifold* of dimension n if it satisfies the following axioms:

- It is a Hausdorff.[1]
- It is a second-countable[2] space.
- It is locally Euclidean.

The locally Euclidean axiom means that for any open set $U \in \tau$ there exists an open set $\widetilde{U} \subseteq \mathbb{R}^n$ such that there is a homeomorphism $\varphi : U \longrightarrow \widetilde{U}$ i.e., a continuous bijective map with continuous inverse. (U, φ) is called a chart on the manifold M. Let $\mathcal{A} = \{(U_\alpha, \varphi_\alpha)\}$ be a set of charts on M. We define $\mathcal{A}$ as an atlas if $\bigcup_\alpha U_\alpha$ covers M. An atlas $\mathcal{A}$ is smooth if for any two charts $(U_\alpha, \varphi_\alpha)$, (U_β, φ_β), then $\varphi_\alpha^{-1} \circ \varphi_\beta(U_\alpha \bigcap U_\beta)$ is smooth. A smooth structure on M is a smooth atlas. Thus, a smooth manifold is a pair $(M, \mathcal{A})$, where M is a manifold and $\mathcal{A}$ is a smooth structure on M [36].

Definition 2.1.2. [26] A Lie group G is a smooth manifold which is also a group, such that the group multiplication $(g, h) \mapsto g.h$ and inversion $g \mapsto g^{-1}$ define smooth maps.

As mentioned, one of the striking features of a Lie group is its local linearization, which is described next.

2.1.1 Linearization of a Lie Group

A key tool in the study of a smooth manifold is the idea of a *tangent space*, which can be used to approximate the manifold linearly. For example, a one variable function can be approximated by its derivative, which is the tangent line. An example of a manifold is Euclidian space $\mathbb{R}^n$. The tangent space at point $a \in \mathbb{R}^n$ is all the vectors with initial point at a, denoted by $T_a\mathbb{R}^n$:

$$T_a\mathbb{R}^n = \{(a, X), X \in \mathbb{R}^n\},$$

where X is a vector in $\mathbb{R}^n$. The notation X_a will be used for an element of $T_a\mathbb{R}^n$ [36].

[1] A topological space (X, τ) is *Hausdorff* if for every $a, b \in X$, there exists open sets $U_a, U_b \in \tau$ such that $a \in U_a$, $b \in U_b$ and $U_a \bigcap U_b = \emptyset$ [38].

[2] A topological space (X, τ) is called *second-countable* if τ has a countable base [38].

Let f be a function $f : U \to V$, such that $a \in U \subseteq \mathbb{R}^n$, $V \subseteq \mathbb{R}^n$ and U, V are open sets in $\mathbb{R}^n$. Then each X_a induces a directional derivative of f at point a along X_a. Now let M be a manifold, and $f : U \subseteq M \to V \subseteq M$ be a C^∞ function[3]; U, V are open sets. Similar to Euclidean space we can define directional derivatives of f at some point $p \in U$. Let $X : C^\infty(M) \to \mathbb{R}$ be a linear map[4]. Then X is a derivation at $p \in M$ if, for any $f, g \in C^\infty(M)$:

$$X(fg)(p) = f(p)X(g) + g(p)X(f).$$

The set of all derivations of $C^\infty(M)$ at p is the tangent space to M at p and is denoted by $T_p(M)$. It can be shown directly that T_pM forms a vector space over the field $\mathbb{R}$:

$$\begin{aligned} X_p(f) + Y_p(f) &= (X+Y)_p(f), \\ c(X_p(f)) &= (cX)_p(f), c \in \mathbb{R}. \end{aligned}$$

As a Lie group is also a manifold, a tangent space can be associated with a Lie group.

2.1.2 Lie Subgroups

In this section we will introduce the concept of a Lie subgroup. Lie subgroups are subgroups of a group where they are submanifolds. There are two types of submanifold, which will be explained in the following.

Let $F : N \to M$ be a differentiable map of smooth manifolds. Suppose (U, φ) and (V, ψ) are coordinate charts in $p \in N$ and $F(p) \in M$ respectively, such that $F(U) \subset V$.

Definition 2.1.3. The rank of F at p is defined as the rank of $\hat{F}$ at $\varphi(p)$, where $\hat{F}$ is:

$$\hat{F} = \psi \circ F \circ \varphi^{-1} : \varphi(U) \to \psi(V), \tag{2.1}$$

such that

$$\hat{F}(x^1, x^2, \ldots, x^n) = (f^1(x^1, x^2, \ldots, x^n), \ldots, f^m(x^1, x^2, \ldots, x^n)).$$

Thus, the rank of F at p is the rank of the Jacobian matrix:

[3] A C^∞ function is a function that is differentiable for all degrees of differentiation.

[4] $C^\infty(M)$ is the space of C^∞ functions from M to M.

$$\begin{pmatrix} \frac{\partial f^1}{\partial_x^1} & \cdots & \frac{\partial f^1}{\partial_x^n} \\ \vdots & & \vdots \\ \frac{\partial f^m}{\partial_x^1} & \cdots & \frac{\partial f^m}{\partial_x^n} \end{pmatrix} \tag{2.2}$$

Note that the rank of F is independent of the choice of coordinates [12].

2.1.2.1 Submanifolds

Let $F : N \to M$ be a continuous map, $\dim(M) = m, \dim(n) = n$.

Definition 2.1.4. $F : N \to M$ is an *immersion* if $rank(F) = dim(N)$ everywhere.

If F is an injective immersion then $F(N)$ is called the *immersed submanifold* of M. A question arises here: what is the topology of $F(N)$? Since F is a continuous map, so a set u is open in $F(N)$ if and only if $F^{-1}(u)$ is open in N. Therefore, all sets that are open on the relative topology in $F(N)$ are open, but there are some more open sets that are not open on the relative topology. In fact, the topology of $F(N)$ is finer than the relative topology in $F(N)$; it has more open sets, see Example 2.

Example 2. Let

$$F : \mathbb{R} \to \mathbb{R}^2,$$
$$F(t) = \left(2\cos\left(g(t) - \frac{\pi}{2}\right), \sin 2\left(g(t) - \frac{\pi}{2}\right)\right).$$

Let $g(t)$ be a monotone C^∞ function on $-\infty < t < \infty$ such that $g(0) = \pi$, $\lim_{t\to-\infty} g(t) = 0$ and $\lim_{t\to\infty} g(t) = 2\pi$. The image of F is shown in Figure 2.1; a is an open set in the image of F, but it is not an open set with respect to the relative topology in $\mathbb{R}^2$.

If F is a homeomorphism[5] and an immersion then $F(N)$ is called an *embedded submanifold*. The topology of $F(N)$ is matched to the relative topology of $F(N)$ in M, because F and F^{-1} are continuous.

An immersed submanifold is a locally embedded submanifold. If $F(N)$ is an immersed submanifold then function F from N to $F(N)$ is bijective. Since $rank(DF) = n$, so it is an isomorphism, and by the inverse mapping theorem, F is locally invertible and consequently $F(N)$ is a locally embedded submanifold.

[5] A homeomorphism is a bijective continuous map with continuous inverse.

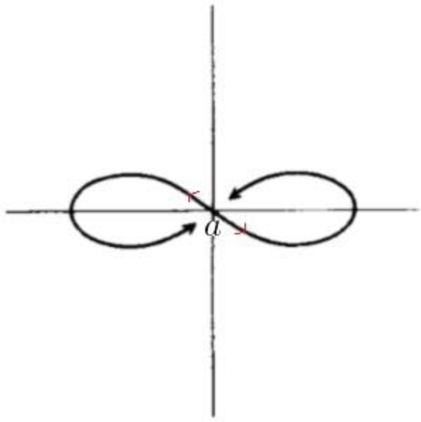


Figure 2.1: The image of the function F which is an immersed submanifold of $\mathbb{R}^2$. Figure in taken from [12].

Definition 2.1.5. Subgroup H is a Lie subgroup of G if the inclusion map $H \hookrightarrow G$ is an injective immersion and group homomorphism[6].

Therefore, a Lie subgroup is an immersed submanifold of G. The following proposition shows that embedded subgroups are automatically Lie subgroups:

Proposition 2.1.6. [36] Let G be a Lie group, and suppose $H \subset G$ is a subgroup that is also an embedded submanifold. Then H is a closed Lie subgroup of G.

Proof. See [36] page 124. □

2.1.3 Infinitesimal Transformations of Lie Group Actions on Manifolds

The action of a Lie group on a manifold, similarly to the action of a group on a set, needs to obey the following axioms [36]:

Definition 2.1.7. Let $(G, *)$ be a Lie group and M a manifold. A left action of G on M is a smooth map $\Phi : G \times M \to M$ denoted by $\Phi(g, x) = g \cdot x$ which satisfies:

- $e \cdot x = x$

[6]A map $\varphi : (H, \cdot) \to (G, *)$ is a group homomorphism if $\varphi(h_1 \cdot h_2) = \varphi(h_1) * \varphi(h_2)$, G and H are groups.

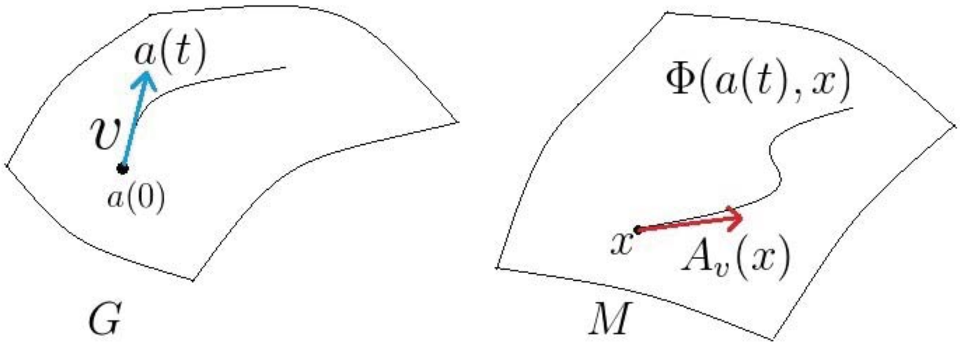


Figure 2.2: The group G acting on the manifold M. The flow $\Phi(a(t), x)$ is generated by the action of $a(t)$ on M. A_v is the infinitesimal generator of $\Phi(a(t), x)$, where $a(0) = e$ and $a'(0) = v$.

- $g \cdot (h \cdot x) = (g * h) \cdot x$

for all $x \in M, g, h \in G$.

Let $a(t)$ be a curve in G such that $a(0) = e$ and $a'(0) = v$, see Figure 2.2. Then, $\Phi(a(t), x)$ generates a flow on the manifold M. The infinitesimal transformation of the flow generated by $a(t)$ on M is:

$$x + A_v(x)\epsilon + o(\epsilon^2) = \Phi(e + \epsilon v, x),$$

where $A_v(x)$ is:

$$A_v(x) = \frac{\partial}{\partial t}\Phi(a(t), x))\bigg|_{t=0} = \lim_{\epsilon \to 0} \frac{\Phi(e + \epsilon v, x) - \Phi(e, x)}{\epsilon} \tag{2.3}$$

If $a(t) = \exp(tv)$, $v \in T_eG$, and $\Phi(a(t), x) = \exp(tv) \cdot x$, then $A_v(x)$ is known as an *infinitesimal generator* of this action. The curve $\exp(tv)$ is known as a *one-parameter* subgroup of the group. The following section covers these concepts.

2.1.4 Lie Algebras and One-Parameter Subgroups of a Lie Group

Lie algebras were introduced to study the infinitesimal transformations of Lie group actions on manifolds. The Lie algebra of a Lie group is the set of *left (or right) invariant vector fields* on the Lie group, together with an induced multiplication from the group. The set of all left/right invariant vector fields is isomorphic to the tangent space at the identity of the group, T_eG. In the following these concepts are explained in detail.

A vector field on a manifold M is a function from M to the tangent bundle $\bigcup_{p\in M} T_pM$ such that it assigns to each point $p \in M$ a vector $X_p \in T_pM$. A C^r-vector field is defined as follows.

Definition 2.1.8. [12] A vector field of class C^r on M is a function that assigns to each point p a vector $X_p \in T_pM$, such that its components are C^r with respect to any *coordinate frame.*

Roughly speaking, a coordinate frame is a basis of T_pM that is induced from the coordinate chart at p in the manifold. The following theorem describes coordinate frames precisely.

Theorem 2.1.9. [12] *To each coordinate neighbourhood U in M, there corresponds a basis $E_{1p}, E_{2p}, \ldots, E_{np}$ of T_pM for every $p \in U$.*

Proof. Let (U, φ) be a coordinate chart at p such that $\varphi : U \to \widetilde{U}$. We have $\hat{\varphi} = \varphi \circ \varphi^{-1} \circ i^{-1} : i(\widetilde{U}) \to \varphi(U)$ where $(\widetilde{U}, i)$ is a coordinate chart at $\varphi(p)$, and i is the identity map. So $\hat{\varphi}$ will be a homeomorphism such that its rank is $\dim(\widetilde{U}) = \dim(U)$. Hence $D\varphi$ is an isomorphism. Let $\left\{\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \ldots, \frac{\partial}{\partial x_n}\right\}$ be a basis for $T_{\varphi(p)}\mathbb{R}^n$, then $\hat{\varphi}(\frac{\partial}{\partial x_i}) = E_{ip}$ is a basis for T_pM. □

The basis $\{E_{ip}\}_{i=1,\ldots,n}$ is called a *coordinate frame.* So any X_p in T_pM can be expressed as:

$$X_p = \sum_i x^i E_{ip},$$

where x^i are the components of X_p with respect to this coordinate frame.

Now, let $\mathcal{V}$ be the space of all C^∞ vector fields on M. The space $(\mathcal{V}, +)$ forms a vector space over the field $\mathbb{R}$. A multiplication is defined on $\mathcal{V}$ such that for any $X, Y \in \mathcal{V}$,

$$[X, Y] = XY - YX.$$

This multiplication is called the commutator of X and Y. If $Z = [X, Y]$, and f is a C^∞ function from U to $\mathbb{R}$ and $p \in U$, then the directional derivative of f is as follows:

$$Z_p(f) = (XY - YX)_p(f) = X_p(Yf) - Y_p(Xf).$$

This multiplication obeys the following properties:

- if $X_1, X_2 \in \mathcal{V}$ then $[X_1, X_2] \in \mathcal{V}$ (Closure)
- $[X_1 + X_2, X_3] = [X_1, X_3] + [X_2, X_3]$ and $[X_1, X_2 + X_3] = [X_1, X_2] + [X_1, X_3]$ (Bilinearity)
- $[X_1, X_2] = -[X_2, X_1]$ (anti-symmetry)
- $[X_1, [X_2, X_3]] = [[X_1, X_2], X_3] + [X_2, [X_1, X_3]]$ (derivative property)

See [12] page 153, for the proof. A vector space $(\mathcal{V}, +, [,])$ that obeys the above conditions is called a *Lie algebra*[7].

A G-invariant vector field is defined as follows [52]:

Definition 2.1.10. A vector field v on M is called G-invariant if by the action of any group element it is unchanged, meaning that $dg(v\mid_x) = v\mid_{g\cdot x}$ for all $g \in G$ and $x \in M$ such that $g \cdot x$ is defined.

[7] A *linear algebra*,$(A, +, \cdot, \Box)$ is a vector space $(A, +, \cdot)$ over a field F, such that:

- $v_1, v_2 \in A$ then $v_1 \Box v_2 \in A$, (Closure)
- $(v_1 + v_2)\Box v_3 = v_1 \Box v_3 + v_2 \Box v_3$ and $v_1 \Box (v_2 + v_3) = v_1 \Box v_2 + v_1 \Box v_3$, (Bilinearity)

Different types of algebra may obtain depending on the additional properties:

- $(v_1 \Box v_2)\Box v_3 = v_1 \Box (v_2 \Box v_3)$, (associativity)
- $v_1 \Box 1 = v_1$, (existence of identity, in general this identity is not equal to the identity under $+$ and $\cdot$)
- $v_1 \Box v_2 = \pm v_2 \Box v_1$, ($+$ symmetric, $-$ anti-symmetric)
- $v_1 \Box (v_2 \Box v_3) = (v_1 \Box v_2)\Box v_3 + v_2 \Box (v_1 \Box v_3)$, (derivative property)

A linear algebra that is anti-symmetric and obeys the derivative property is called a *Lie algebra* [25] .

One of most important G-invariant vector fields is when G acts on itself by right multiplication:

$$R_a : G \to G$$
$$R_a(g) = g * a,\ a, g \in (G, *).$$

A vector field v is called a *right invariant vector* field if:

$$DR_a X_g = X_{ga},\ \text{for all } a, g \in G, \text{where } X_g \in v,$$

and $DR_a X_g$ is the derivative of R_a at X_g. If X_e is the value of a vector field at the identity, then $DR_a X_e = X_{ea} = X_a$, so every right invariant vector field can be determined uniquely by its value at the identity. The set of right invariant vector fields is denoted by $\mathfrak{g}_R$. As all right invariant vector fields can be identified by their value at the identity, so $\mathfrak{g}_R \cong T_e G$. $\mathfrak{g}_R$ is a subalgebra of the Lie algebra of all smooth vector fields on G, where a subalgebra is defined as follows:

Definition 2.1.11. Let K be a vector subspace of a Lie algebra L. K is a subalgebra of L if:

$$x, y \in K \Rightarrow [x, y] \in K,$$

or equivalently K is a subalgebra if it is closed under the multiplication of the Lie algebra. Right invariant vector field $\mathfrak{g}_R$ is a subalgebra of the Lie algebra of all smooth vector fields on G, because if $X, Y \in \mathfrak{g}_R$ then their commutator $[X, Y] \in \mathfrak{g}_R$:

$$\begin{aligned} DR_a(XY - YX)_g(f) &= DR_a(X_g(Yf) - Y_g(Xf)) \\ = DR_a X_g(Yf) - DR_a Y_g(Xf) &= X_{ga}(Yf) - Y_{ga}(Xf) \\ &= (XY - YX)_{ga} f. \end{aligned}$$

A left invariant vector field is defined similarly to a right invariant vector field. A left invariant vector field is invariant under the derivative of the left action of the group on itself. Both left and right invariant vector fields can be determined by their value at the identity, so both are isomorphic to $T_e G$. The right or left invariant vector fields is defined as the Lie algebra of a Lie group, and denoted by $\mathfrak{g}$.

Now we define one-parameter subgroups of a Lie group. Let $v \in \mathfrak{g}$ be a left invariant vector field on the Lie group G. Let $\exp(tv)$ be the flow of the vector field v. Suppose G acts on itself by left or right multiplication. Then the flow generated by $\exp(tv)$ at

e is a one-parameter subgroup of G:

$$\Phi(\exp(tv), e) = \exp(tv)e = \exp(tv)$$

and the vector field v is known as the infinitesimal generator of the subgroup.

2.1.5 Infinitesimal Generators of the Group Action on Manifolds

Suppose G is acting on a manifold M. The action induces a vector field on the manifold. Just as a one-parameter subgroup is the flow of a vector field, so the Lie group transformation can be generated by the set of vector fields on M known as infinitesimal generators. The flow of each infinitesimal generator coincides with the flow created by the action of a one-parameter subgroup on M. More precisely, suppose v generates the one-parameter subgroup $\exp(tv) \subset G$, $t \in \mathbb{R}$. If a one-parameter subgroup transformation is acting on M as $x \mapsto \exp(tv) \cdot x$, $x \in M$, then the infinitesimal generator of the one-parameter transformation is identified as $\hat{v}$, such that:

$$\hat{v}\Big|_x = \frac{d}{dt}\exp(tv)x\Big|_{t=0}, \quad x \in M, v \in \mathfrak{g}.$$

Consequently, $\hat{v}\Big|_x = d\Phi_x(v\Big|_e)$, where $\Phi_x : G \to M$ is given by $\Phi_x(g) = g.x$.

2.1.6 Lie Group and Lie Algebra Correspondence

There is a correspondence between a Lie group and its Lie algebra that allows us to replace or approximate the Lie group with its Lie algebra when helpful. The following theorem states this correspondence.

Theorem 2.1.12. There exists an open neighbourhood N_0 of $0 \in \mathfrak{g}$ and an open neighbourhood N_e of $e \in G$ such that $\exp : \mathfrak{g} \to G$ is an analytic and diffeomorphic mapping of N_0 onto N_e.

Proof. See [30] page 104. □

2.1.7 Distances in Lie groups

The distance function of two images is a way to measure the difference between images; examples of such functions were given in Chapter 1. Another way to know how far from or close to each other images are is to measure the distance between the transformations in the group that registers them. Lie groups are manifolds, and manifolds are often equipped with a *Riemannian* metric [12], which is a family of bilinear forms $\Phi_p : T_pM \times T_pM \to \mathbb{R}$, $p \in M$ that are:

- symmetric: $\Phi_p(v, w) = \Phi_p(w, v)$.
- positive definite: $\Phi_p(v, w) \geq 0$ and equality holds if and only if $v = w$.

A manifold with a Riemannian metric is called a *Riemannian manifold.* We often write $\Phi_p(v, w)$ as an inner product, $\Phi_p(v, w) = \langle v, w \rangle$. The length of a curve is the integration of the metric, i.e, the length of $\varphi : [0, 1] \to M$, $0 \leq t \leq 1$ is:

$$\int_0^1 \left\langle \frac{d\varphi}{dt}, \frac{d\varphi}{dt} \right\rangle^{\frac{1}{2}} dt.$$

So, the distance between two points, $d(p, q)$, can be calculated by the integration of the metric along the curve connecting them. The shortest path connecting the points is called a *geodesic.* If $d(p, q)$ is defined to be the geodesic distance, then a Riemannian manifold is a metric space.

2.1.8 Matrix Lie Groups

The *General Linear Group* $GL(2, \mathbb{C})$ is the set of invertible 2×2 complex matrices. It forms a group under matrix multiplication. It is described by the multiplication

$$\begin{aligned} G \times G &\to G \\ (g, h) &\mapsto g \times h. \end{aligned}$$

Both multiplication and the inverse map

$$\begin{aligned} G &\to G \\ g &\mapsto g^{-1}, \end{aligned}$$

are smooth functions. Therefore $GL(2,\mathbb{C})$ is a Lie group. The set of such matrices with determinant 1 is a subgroup of $GL(2,\mathbb{C})$, and is known as the *Special Linear group*, denoted by $SL(2,\mathbb{C})$.

The Lie algebra of $GL(2,\mathbb{C})$ is denoted by $\mathfrak{gl}(2,\mathbb{C})$. It is the space of all 2×2 complex matrices. In $GL(2,\mathbb{C})$, the logarithm map log, and exponential map exp give the correspondence between a neighbourhood of $I \in G$ and neighbourhood of $\mathbf{0} \in \mathfrak{g}$, where I is the identity matrix and $\mathbf{0}$ is the zero matrix, see [64] for the proof. For matrix Lie groups the exponential map is given as follows:

$$\exp(X) = I + X + \frac{1}{2}X^2 + \frac{1}{6}X^3 + \ldots,$$

where $X \in \mathfrak{g}$. Also, the function log is given as follows:

$$\log(I + A) = A - \frac{1}{2}A^2 + \frac{1}{3}A^3 - \frac{1}{4}A^4 + \ldots,$$

where $A \in G$.

In $\mathfrak{gl}(2,\mathbb{C})$ we have:

$$\det(\exp(X)) = \exp(tr(X)),$$

where $X \in \mathfrak{gl}(2,\mathbb{C})$ and tr is the trace of a matrix, see [25] for the proof. So, if $A \in SL(2,\mathbb{C})$ we have:

$$\begin{aligned} 1 = \det(\exp(\log(A))) &= \exp(tr(\log(A))) \\ &\Leftrightarrow \exp(tr(\log(A))) = 1 \\ &\Leftrightarrow tr(\log(A)) = 0. \end{aligned}$$

Now $\log(A)$ belongs to the Lie algebra of $SL(2,\mathbb{C})$ which, as has just been shown, is traceless. Therefore, the Lie algebra of $SL(2,\mathbb{C})$ is the set of matrices of zero trace. The standard basis of $\mathfrak{sl}(2,\mathbb{C})$ over the field $\mathbb{R}$ is:

$$v_1 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, v_2 = \begin{pmatrix} 0 & i \\ 0 & 0 \end{pmatrix}, v_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, v_4 = \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix}, v_5 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, v_6 = \begin{pmatrix} 0 & 0 \\ i & 0 \end{pmatrix}.$$

The one-parameter subgroups corresponding to each basis element are:

$$\exp(tv_1) = \begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix}$$
$$\exp(tv_2) = \begin{pmatrix} 1 & it \\ 0 & 1 \end{pmatrix}$$
$$\exp(tv_3) = \begin{pmatrix} \exp(t) & 0 \\ 0 & \exp(-t) \end{pmatrix}$$
$$\exp(tv_4) = \begin{pmatrix} \exp(it) & 0 \\ 0 & \exp(-it) \end{pmatrix}$$
$$\exp(tv_5) = \begin{pmatrix} 1 & 0 \\ t & 1 \end{pmatrix}$$
$$\exp(tv_6) = \begin{pmatrix} 1 & 0 \\ it & 1 \end{pmatrix}$$

Another Lie subgroup of $GL(2,\mathbb{C})$ is the *Special Unitary* group, $SU(2,\mathbb{C})$. This subgroup is the group of unitary matrices[8] with determinant one:

$$\left\{ A : A = \begin{pmatrix} a & b \\ -\bar{b} & \bar{a} \end{pmatrix}, a\bar{a} + b\bar{b} = 1 \right\},$$

where the over line stands for the complex conjugate. It is also a subgroup of $SL(2,\mathbb{C})$. Let $A \in SU(2,\mathbb{C})$ then:

$$\begin{aligned} \log(AA^*) &= \log(I) \\ \log(A) + \log(A^*) &= \mathbf{0} \\ \log(A) &= -\log(A)^*, \end{aligned}$$

where $\mathbf{0}$ is the zero matrix in $\mathfrak{gl}(2,\mathbb{C})$. Now $\log(A)$ is an element of the Lie algebra of $SU(2,\mathbb{C})$. Therefore, $\mathfrak{su}(2,\mathbb{C})$ is the space of skew-hermitian[9] matrices of zero trace. A basis of $\mathfrak{su}(2,\mathbb{C})$ is:

$$w_1 = \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix}, w_2 = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, w_3 = \begin{pmatrix} 0 & i \\ i & 0 \end{pmatrix}.$$

[8]Matrix A is unitary if and only if $AA^* = I$, where A^* is conjugate transpose.
[9]A matrix A is *skew-hermitian* if $A^* = -A$.

The one-parameter subgroups corresponding to each basis element are:

$$\exp(tw_1) = \begin{pmatrix} \exp(it) & 0 \\ 0 & \exp(-it) \end{pmatrix}$$
$$\exp(tw_2) = \begin{pmatrix} \cos(t) & \sin(t) \\ -\sin(t) & \cos(t) \end{pmatrix}$$
$$\exp(tw_3) = \begin{pmatrix} \cos(t) & i\sin(t) \\ i\sin(t) & \cos(t) \end{pmatrix}$$

The *Real Special Linear* group, $SL(2,\mathbb{R})$, the real 2×2 matrices of determinant 1, is also a Lie subgroup of $SL(2,\mathbb{C})$. It acts on the real line as follows:

$$\Phi : SL(2,\mathbb{R}) \times \mathbb{R} \to \mathbb{R}$$
$$\Phi\left(\begin{pmatrix} a & b \\ c & d \end{pmatrix}, x\right) = \frac{ax+b}{cx+d},$$

The Lie algebra of $SL(2,\mathbb{R})$ is the space of real traceless matrices. Let

$$u_1 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, u_2 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, u_3 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$$

be a basis for $\mathfrak{sl}(2,\mathbb{R})$. The one-parameter subgroups generated by u_i, $i = 1,2,3$ are:

$$\exp(tu_1) = \begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix}$$
$$\exp(tu_2) = \begin{pmatrix} \exp(t) & 0 \\ 0 & \exp(-t) \end{pmatrix}$$
$$\exp(tu_3) = \begin{pmatrix} 1 & 0 \\ t & 1 \end{pmatrix}$$

The infinitesimal generators are:

$$\frac{d}{dt}\exp(tu_1)x\Big|_{t=0} = \frac{d}{dt}(x+t)\Big|_{t=0} = 1$$
$$\frac{d}{dt}\exp(tu_2)x\Big|_{t=0} = \frac{d}{dt}(\exp(2t)x)\Big|_{t=0} = 2x$$
$$\frac{d}{dt}\exp(tu_3)x\Big|_{t=0} = \frac{d}{dt}\left(\frac{x}{tx+1}\right)\Big|_{t=0} = -x^2$$

Ignoring coefficients, the infinitesimal generators are: $1\partial_x, x\partial_x$ and $x^2\partial_x$, where ∂_x is the basis of the vector field on $\mathbb{R}$, and infinitesimal transformations at the identity of the group $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ are:

$$\Phi(x, e + \epsilon v) = \Phi(x, e) + D\Phi(x, e)\Big|_e \epsilon v.$$

Let $\epsilon v = (\epsilon_1, \epsilon_2, \epsilon_3)$, then:

$$\Phi(x, e + \epsilon v) = x + \epsilon_1 + x\epsilon_2 + x^2\epsilon_3.$$

The following section is about the *Projective Linear* group and its relation with the General Linear group.

2.1.9 Projective Linear Group

Definition 2.1.13. Let V be a vector space over some field K such as $\mathbb{R}$ or $\mathbb{C}$. An equivalence relation on $V - \{0\}$ is defined as follows:

$$x \sim y \Leftrightarrow y = \lambda x, \text{for some } \lambda \in K.$$

Let $P(V)$ be the set of nonzero equivalence classes,

$$P(V) = \{[x], [x] = \{y; y = \lambda x\}, x, y \in V, \lambda \neq 0\}.$$

The set $P(V)$ is called projective space. When $K = \mathbb{R}$ it is called real projective space and when $K = \mathbb{C}$ it is called complex projective space. If the dimension of V is n then the dimension of $P(V)$ is $n - 1$.

The set of linear transformations from $P(V)$ to $P(V)$ under composition of maps forms a group and is called the projective group, which is denoted by $PSL(V)$.

Example 3. Let $V = \mathbb{R}^2$, then points in projective space $P(\mathbb{R}^2)$ are lines passing through the origin:

$$P(\mathbb{R}^2) = \{[x]; [x] = \{y; y = \lambda x\}, x, y \in \mathbb{R}^2\}.$$

The equivalence class can be considered as the slope λ of the lines. Then every line can be mapped to its slope, $(x, y) \mapsto (1, \frac{y}{x}) = (1, \lambda)$.

Let $GL(V)$ be the group of linear transformations from V to V. The following lemma states the relationship between $PSL(V)$ and $GL(V)$.

Lemma 2.1.14. *$PSL(V) \cong GL(V)/K_*I$, where $K_* = K - \{0\}$ and I is the identity map in $GL(V)$.*

Proof. See [10]. □

In the following we show two specific relationships that will be used later in Sections 2.3 and 2.4.

1. $PSL(2, \mathbb{C}) \cong SL(2, \mathbb{C})/ \pm I$.
2. $PSL(3, \mathbb{R}) \cong SL(3, \mathbb{R})/ \pm I$.

So:

1. $PSL(2, \mathbb{C}) \cong SL(2, \mathbb{C})/ \pm I$:

   As mentioned, the special linear group, $SL(2, \mathbb{C})$, is the group of matrices with determinant one:

   $$SL(2, \mathbb{C}) = \left\{ A : A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \det(A) = 1, a, b, c, d \in \mathbb{C} \right\}.$$

   This group acts on $\mathbb{C}^2$ by matrix multiplication. Let $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in SL(2, \mathbb{C})$. Then

   $$\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \mapsto \begin{pmatrix} az_1 + bz_2 \\ cz_1 + dz_2 \end{pmatrix}.$$

   Matrix A maps $\begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$ to $\begin{pmatrix} az_1 + bz_2 \\ cz_1 + dz_2 \end{pmatrix}$. The following diagram shows that every transformation in $A \in SL(2, \mathbb{C})$ induces a transformation in $PSL(2, \mathbb{C})$.

$$\begin{tikzcd}
(z_1,z_2) \arrow[r, "SL(2{,}\mathbb{C})"] \arrow[d, "P(\mathbb{C}^2)"'] & (az_1+bz_2,cz_1+dz_2) \arrow[d, "P(\mathbb{C}^2)"] \\
{[\frac{z_1}{z_2},1]} \arrow[r, "PSL(2{,}\mathbb{C})"'] & {[\frac{a\frac{z_1}{z_2}+b}{c\frac{z_1}{z_2}+d},1]}
\end{tikzcd}$$

The space $P(\mathbb{C}^2)$ is isomorphic to the Riemann sphere. Let $z = \frac{z_1}{z_2}$; therefore, if $P \in PSL(2,\mathbb{C})$ then by the above diagram we have:

$$P(z) = \frac{az+b}{cz+d}.$$

2. $PSL(3,\mathbb{R}) \cong SL(3,\mathbb{R})/\pm I$:

   The group $SL(3,\mathbb{R})$ is:

$$\left\{ A : A = \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix} ; \det(A) = 1 \right\}.$$

   The group $SL(3,\mathbb{R})$ acts on $\mathbb{R}^3$ by matrix multiplication. If $A \in SL(3,\mathbb{R})$ then it maps (x,y,z) to $(a_1x+b_1y+c_1z, a_2x+b_2y+c_2z, a_3x+b_3y+c_3z)$. The following diagram shows that every transformation in $SL(3,\mathbb{R})$ induces a transformation belonging to $PSL(3,\mathbb{R})$.

$$\begin{tikzcd}
(x,y,z) \arrow[r, "SL(3{,}\mathbb{R})"] \arrow[d, "P(\mathbb{R}^3)"'] & (a_1x+b_1y+c_1z,a_2x+b_2y+c_2z,a_3x+b_3y+c_3z) \arrow[d, "P(\mathbb{R}^3)"] \\
{\left[\frac{x}{z},\frac{y}{z},1\right]} \arrow[r, "PSL(3{,}\mathbb{R})"'] & {\left[\frac{a_1\frac{x}{z}+b_1\frac{y}{z}+c_1}{a_3\frac{x}{z}+b_3\frac{y}{z}+c_3},\frac{a_2\frac{x}{z}+b_2\frac{y}{z}+c_2}{a_3\frac{x}{z}+b_3\frac{y}{z}+c_3},1\right]}
\end{tikzcd}$$

2.2 Planar Lie Groups

As mentioned in Chapter 1, only a few finite dimensional groups (similarity, affine, projective) have been used in image registration. In this section, we will give a list of finite dimensional Lie groups that, as far as we know, have never been used in image registration. These planar Lie groups will be employed to achieve the third aim of this thesis: to show the benefits of using finite dimensional groups in image registration, and the fifth aim: to introduce a novel type of image registration that we call *multi-registration*. In [26], Olver classified finite dimensional real Lie algebras of vector fields

	Generators
1	$\{\partial_x, \partial_y, x\partial_x + y\partial_y, y\partial_x - x\partial_y, (x^2-y^2)\partial_x + 2xy\partial_y, 2xy\partial_x + (y^2-x^2)\partial_y\}$
2	$\{\partial_x, x\partial_x + y\partial_y, (x^2-y^2)\partial_x + 2xy\partial_y\}$
3	$\{y\partial_x - x\partial_y, (1+x^2-y^2)\partial_x + 2xy\partial_y, 2xy\partial_x + (1+y^2-x^2)\partial_y\}$
4	$\{\partial_x, \partial_y, x\partial_x + y\partial_y, y\partial_x - x\partial_y\}$
5	$\{\partial_x, \partial_y, \alpha(x\partial_x + y\partial_y) + y\partial_x - x\partial_y\}, \alpha \geq 0\}$
6	$\{\partial_x, \partial_y, x\partial_x, y\partial_x, x\partial_y, y\partial_y, x^2\partial_x + xy\partial_y, xy\partial_x + y^2\partial_y\}$
7	$\{\partial_x, \partial_y, x\partial_x, y\partial_x, x\partial_y, y\partial_y\}$
8	$\{\partial_x, \partial_y, x\partial_x - y\partial_y, y\partial_x, x\partial_y\}$
9	$\{\partial_x, 2x\partial_x + y\partial_y, x^2\partial_x + xy\partial_y\}$
10	$\{\partial_x, x\partial_x, y\partial_y, x^2\partial_x + xy\partial_y\}$
11	$\{\partial_x, \partial_y, x\partial_x, y\partial_y, x^2\partial_x, y^2\partial_y\}$
12	$\{\partial_x + \partial_y, x\partial_x + y\partial_y, x^2\partial_x + y^2\partial_y\}$
13	$\{\partial_x, \partial_y, x\partial_x, y\partial_y, x^2\partial_x\}$
14	$\{\partial_x, \partial_y, x\partial_x, x^2\partial_x\}$
15	$\{\partial_x, \partial_y, x\partial_x, y\partial_y\}$
16	$\{\partial_x, \partial_y, x\partial_x + \alpha y\partial_y\}$, $0 < \alpha \leq 1$

Table 2.1: Infinitesimal generators of planar Lie groups [26].

on $\mathbb{R}^2$ (up to changes of local coordinates) completely. In total, 28 sets of generators for the Lie algebras are given. We used 16 of them: 9 include functions of the generators and were excluded as unnecessary, while 3 are one dimension, and therefore removed. The set of 16 are sufficient to demonstrate the proof of concept that we aim for in this thesis. The generators that we use are given in Table 2.1.

In the following sections we derive the transformation relating to each set of generators, and then we put the groups in a lattice to show the subgroup and group relationships. The lattice will be employed for multi-registration in Chapter 6.

2.2.1 Transformations Relating to Each Generator

As explained in Section 2.1.5, the generators of a group action on a manifold are calculated by taking the derivative of a flow of the vector field. Therefore, to obtain the group action from the generators, it is enough to integrate the generators. In this section we derive the transformations relating to each set of generators given in Table 2.1. Some generators are common between several groups, so to avoid repeating the integration in each set of generators, first we derive the transformation relating to each generator individually, then in the next section we derive the transformations relating to each set of generators.

Let $z(0) = z_0 = x_0 + iy_0$ at $t = 0$.

1. Generator is ∂_x. ∂_x, which is a vector field that assigns to each point (x, y) on the plane a constant vector $(1, 0)$. Hence, the differential equation is:

$$\begin{cases} \frac{dx}{dt} = 1 \Rightarrow dx = dt \Rightarrow x(t) = t + x_0 \\ \frac{dy}{dt} = 0 \Rightarrow y(t) = y_0 \end{cases}.$$

   Therefore, ∂_x is the generator of

$$x(t) = t + x_0, y(t) = y_0, \tag{2.4}$$

   which is x-translation. Similar to ∂_x, ∂_y is the generator of $y(t) = t+y_0, x(t) = x_0$ which is y-translation.

2. Generator is $x\partial_x$, which assigns to each point (x, y) on the plane a vector $(x, 0)$. Hence, the differential equation is:

$$\begin{cases} \frac{dx}{dt} = x \Rightarrow x(t) = \exp(t + c) \\ \frac{dy}{dt} = 0 \Rightarrow y(t) = y_0 \end{cases},$$

   where $x_0 = \exp(c)$. Therefore $x\partial_x$ is the generator of

$$x(t) = \exp(t)x_0, y(t) = y_0, \tag{2.5}$$

   which is scaling along the x-axis.

   Similarly $y\partial_y$ is the generator of $y(t) = \exp(t)y_0, x(t) = x_0$ which is scaling along the y-axis.

3. Generator is $x\partial_y$, so:

$$\begin{cases} \frac{dy}{dt} = x \Rightarrow y(t) = tx + c \\ \frac{dx}{dt} = 0 \Rightarrow x(t) = x_0 \end{cases},$$

   where $y_0 = c$. Therefore, $x\partial_y$ is the generator of the shear

$$y(t) = tx(t) + y_0, x(t) = x_0. \tag{2.6}$$

   Similarly, $y\partial_x$ is the generator of $x(t) = ty(t) + x_0, y(t) = y_0$.

   From here on we write $\dot{x} = \frac{dx}{dt}$, $\dot{y} = \frac{dy}{dt}$.

4. Generator is $x\partial_x + y\partial_y$, so,

$$\begin{cases} \dot{x} = x \\ \\ \dot{y} = y \end{cases} \implies \dot{x} + i\dot{y} = x + iy \Rightarrow \dot{z} = z$$

and,

$$\dot{z} = z \Rightarrow z(t) = \exp(t)\exp(c). \tag{2.7}$$

Therefore, $x\partial_x + y\partial_y$ is the generator of

$$z(t) = \exp(t)z_0, \tag{2.8}$$

where $z_0 = \exp(c)$, which is a scaling.

5. Generator is $x\partial_y - y\partial_x$, so,

$$\begin{cases} \dot{y} = x \\ \\ \dot{x} = -y \end{cases} \implies \dot{x} + i\dot{y} = -y + ix \Rightarrow \dot{z} = iz$$

and,

$$\dot{z} = iz \Rightarrow z(t) = \exp(it + c).$$

Therefore, $x\partial_y - y\partial_x$ is the generator of

$$z(t) = \exp(it)z_0, \tag{2.9}$$

where $z_0 = \exp(c)$, so it is rotation about the origin.

6. Generator is $\alpha(x\partial_x + y\partial_y) + y\partial_x - x\partial_y$, $\alpha \geq 0$, so,

$$\begin{cases} \dot{x} = \alpha x + y \\ \\ \dot{y} = \alpha y - x \end{cases} \implies \dot{x} + i\dot{y} = \alpha(x + iy) - i(x + iy) \Rightarrow \dot{z} = (\alpha - i)z$$

and,

$$\dot{z} = (\alpha - i)z \Rightarrow z(t) = \exp((\alpha - i)t + c).$$

Therefore, $\alpha(x\partial_x + y\partial_y) + y\partial_x - x\partial_y$ is the generator of

$$z(t) = \exp((\alpha - i)t)z_0, \tag{2.10}$$

where $z_0 = \exp(c)$. This describes a spiral.

7. Generator is $(1+x^2-y^2)\partial_x + 2xy\partial_y$.

$$\begin{cases} \dot{x} = 1+x^2-y^2 \\ \\ \dot{y} = 2xy \end{cases} \implies \dot{x}+i\dot{y} = 1+x^2-y^2+i2xy \Rightarrow \dot{z} = 1+z^2$$

and,

$$\begin{aligned} \dot{z} = 1+z^2 \Rightarrow \frac{dz}{1+z^2} = dt \Rightarrow & \\ \int \frac{dz}{1+z^2} = \int dt \Rightarrow \tan^{-1}(z) = t+c \Rightarrow & \\ z(t) = \tan(t+\tan^{-1}(z_0)) = \frac{\tan(t)+z_0}{1-z_0\tan(t)} = \frac{\cos(t)z_0+\sin(t)}{-\sin(t)z_0+\cos(t)}, & \end{aligned}$$

where $\tan^{-1}(z_0) = c$.

8. Generator is $(1+y^2-x^2)\partial_y + 2xy\partial_x$, so,

$$\begin{cases} \dot{x} = 2xy \\ \\ \dot{y} = 1+y^2-x^2 \end{cases} \implies \dot{x}+i\dot{y} = 2xy+i(1+y^2-x^2) \Rightarrow \dot{z} = i(1-z^2)$$

and,

$$\begin{aligned} \dot{z} = i(1-z^2) \Rightarrow & \\ \int \frac{dz}{1-z^2} = \int idt \Rightarrow \frac{1}{2}\int \frac{1}{z+1} + \frac{1}{1-z} = \int idt \Rightarrow & \\ \frac{1}{2}\ln\frac{1-z}{1+z} = it+c \Rightarrow \frac{1-z}{1+z} = \exp(2it+2c), & \end{aligned}$$

and $\exp(2c) = \frac{1-z_0}{1+z_0}$. Therefore $(1+y^2-x^2)\partial_y + 2xy\partial_x$ is the generator of

$$\begin{aligned} z(t) =& \frac{(1+\exp(2it))z_0 + (1-\exp(2it))}{(1-\exp(2it))z_0 + (1+\exp(2it))} \\ =& \frac{2\exp(it)\cos(t)z_0 - 2i\exp(it)\sin(t)}{-2i\exp(it)\sin(t)z_0 + 2\exp(it)\cos(t)} \\ =& \frac{\cos(-t)z_0 + i\sin(-t)}{i\sin(-t)z_0 + \cos(-t)}. \end{aligned}$$

9. Generator is $(x^2 - y^2)\partial_x + 2xy\partial_y$, so,

$$\begin{cases} \dot{x} = x^2 - y^2 \\ \\ \dot{y} = 2xy \end{cases} \implies \dot{x} + i\dot{y} = x^2 - y^2 + i2xy \Rightarrow \dot{z} = z^2$$

and,

$$\dot{z} = z^2 \Rightarrow \tag{2.11}$$

$$\int \frac{dz}{z^2} = \int dt \Rightarrow -\frac{1}{z(t)} = t + c, \tag{2.12}$$

and $c = -\frac{1}{z(0)}$. Therefore, $(x^2 - y^2)\partial_x + 2xy\partial_y$ is the generator of

$$z(t) = \frac{z_0}{1 - tz_0}. \tag{2.13}$$

10. Generator is $2xy\partial_x + (y^2 - x^2)\partial_y$, so,

$$\begin{cases} \dot{x} = 2xy \\ \\ \dot{y} = y^2 - x^2 \end{cases} \implies \dot{x} + i\dot{y} = 2xy + i(y^2 - x^2) \Rightarrow \dot{z} = -iz^2$$

and,

$$\dot{z} = -iz^2 \Rightarrow \frac{dz}{z^2} = -idt \Rightarrow \tag{2.14}$$

$$\int \frac{dz}{z^2} = \int -idt \Rightarrow -\frac{1}{z(t)} = -it + c, \tag{2.15}$$

and, $c = -\frac{1}{z(0)}$. Therefore $2xy\partial_x + (y^2 - x^2)\partial_y$ is the generator of

$$z(t) = \frac{z_0}{1 + itz_0}. \tag{2.16}$$

11. Generator is $x^2\partial_x + xy\partial_y$, so,

$$\begin{cases} \dot{x} = x^2 \\ \dot{y} = xy \end{cases}.$$

Solving the first equation,

$$\dot{x} = x^2 \Rightarrow \frac{dx}{x^2} = dt \Rightarrow x(t) = \frac{x_0}{1 - tx_0}.$$

Solving the second equation,

$$\dot{y} = \frac{x_0}{1-tx_0}y \Rightarrow \frac{dy}{y} = \frac{x_0}{1-tx_0}dt \Rightarrow$$
$$\int \frac{dy}{y} = \int \frac{x_0}{1-tx_0}dt \Rightarrow \ln(y) = -\ln(1-tx_0) + c,$$

and $c = \ln(y_0)$, so,

$$y(t) = \frac{y_0}{1-tx_0}.$$

Therefore $x^2\partial_x + xy\partial_y$ is the generator of

$$z(t) = x(t) + iy(t) = \frac{x_0}{1-tx_0} + i\frac{y_0}{1-tx_0} = \frac{z_0}{1-tx_0}.$$

12. Generator is $y^2\partial_y + xy\partial_x$, so,

$$\begin{cases} \dot{y} = y^2 \\ \dot{x} = xy \end{cases}.$$

Solving the first equation,

$$\dot{y} = y^2 \Rightarrow \frac{dy}{y^2} = dt \Rightarrow y(t) = \frac{y_0}{1-ty_0}.$$

Solving the second equation,

$$\dot{x} = \frac{y_0}{1-ty_0}x \Rightarrow \frac{dx}{x} = \frac{y_0}{1-ty_0}dt \Rightarrow$$
$$\int \frac{dx}{x} = \int \frac{y_0}{1-ty_0}dt \Rightarrow \ln(x) = -\ln(1-ty_0) + c$$

and $c = \ln(x_0)$, so,

$$x(t) = \frac{x_0}{1-ty_0}.$$

Therefore, $y^2\partial_y + xy\partial_x$ is the generator of

$$z(t) = x(t) + iy(t) = \frac{y_0}{1-ty_0} + i\frac{x_0}{1-ty_0} = \frac{z_0}{1-ty_0}.$$

13. Generator is $x\partial_x - y\partial_y$, so,

$$\begin{cases} \dot{x} = x \\ \dot{y} = -y \end{cases}.$$

Solving the equations, we have

$$\dot{x} = x \Rightarrow x(t) = \exp(t)x_0$$
$$\dot{y} = -y \Rightarrow y(t) = \exp(-t)y_0.$$

Therefore, $x\partial_x - y\partial_y$ is the generator of

$$(x(t), y(t)) = (\exp(t)x_0, \exp(-t)y_0). \tag{2.17}$$

14. Generator is $x\partial_x + \alpha y\partial_y$, $0 < \alpha \leq 1$, so,

$$\begin{cases} \dot{x} = x \\ \dot{y} = \alpha y \end{cases}.$$

Solving equations, we have,

$$\dot{x} = x \Rightarrow x(t) = \exp(t)x_0$$
$$\dot{y} = -y \Rightarrow y(t) = \exp(\alpha t)y_0.$$

Therefore, $x\partial_x + \alpha y\partial_y$ is the generator of

$$(x(t), y(t)) = (\exp(t)x_0, \exp(\alpha t)y_0). \tag{2.18}$$

15. Generator is $\partial_x + \partial_y$, so,

$$\begin{cases} \dot{x} = 1 \\ \\ \dot{y} = 1 \end{cases} \implies \dot{x} + i\dot{y} = 1 + i \Rightarrow \dot{z} = 1 + i$$

and,

$$\dot{z} = 1 + i \Rightarrow dz = (1 + i)dt \Rightarrow z = z_0 + (1 + i)i$$

Therefore, $\partial_x + \partial_y$ is the generator of

$$(x(t), y(t)) = (x_0 + t, y_0 + t) \tag{2.19}$$

16. Generator is $x\partial_x + y\partial_y$, so,

$$\begin{cases} \dot{x} = x \\ \\ \dot{y} = y \end{cases} \implies \dot{x} + i\dot{y} = x + iy \Rightarrow \dot{z} = z$$

and,

$$\dot{z} = z \Rightarrow \frac{dz}{z} = dt \Rightarrow z = \exp(t)z_0$$

Therefore, $x\partial_x + y\partial_y$ is the generator of

$$(x(t), y(t)) = (\exp(t)x_0, \exp(t)y_0) \tag{2.20}$$

17. Generator is $x^2\partial_x + y^2\partial_y$, so,

$$\begin{cases} \dot{x} = x^2 \\ \dot{y} = y^2 \end{cases}.$$

Solving the equations, we have,

$$\dot{x} = x^2 \Rightarrow x(t) = \frac{x_0}{1 - tx_0},$$
$$\dot{y} = y^2 \Rightarrow y(t) = \frac{y_0}{1 - ty_0}.$$

Therefore, $x^2\partial_x + \alpha y^2\partial_y$ is the generator of

$$(x(t), y(t)) = \left(\frac{x_0}{1 - tx_0}, \frac{y_0}{1 - ty_0}\right). \tag{2.21}$$

18. Generator is $x^2\partial_x$, so,

$$\begin{cases} \dot{x} = x^2 \Rightarrow x(t) = \frac{x_0}{1-tx_0} \\ \dot{y} = 0 \Rightarrow y(t) = y_0 \end{cases}.$$

Therefore, $x^2\partial_x$ is the generator of

$$x(t) = \frac{x_0}{1 - tx_0}, y(t) = y_0. \tag{2.22}$$

Similarly, $y^2\partial_y$ is the generator of $y(t) = \frac{y_0}{1-ty_0}, x(t) = x_0$.

As mentioned, in the following section we use the above transformations of each generator to derive the group of transformations from the generator set.

2.2.2 Planar Lie Groups

The set of generators are:

1) $\{\boldsymbol{\partial_x, \partial_y, x\partial_x + y\partial_y, y\partial_x - x\partial_y, (x^2 - y^2)\partial_x + 2xy\partial_y, 2xy\partial_x + (x^2 - y^2)\partial_y}\}$:

The transformations relating to each generator are:

$$\begin{aligned}
\boldsymbol{\partial_x} &: x+iy \mapsto x+t_1,\\
\boldsymbol{\partial_y} &: x+iy \mapsto y+t_2,\\
\boldsymbol{x\partial_x + y\partial_y} &: z \mapsto \exp(t_3)z,\\
\boldsymbol{y\partial_x - x\partial_y} &: z \mapsto \exp(it_4)z,\\
\boldsymbol{(x^2-y^2)\partial_x + 2xy\partial_y} &: z \mapsto \frac{z}{1-t_5z},\\
\boldsymbol{2xy\partial_x + (x^2-y^2)\partial_y} &: z \mapsto \frac{z}{1+it_6z},
\end{aligned}$$

where $z = x + iy$. These transformations correspond to transformations in $SL(2,\mathbb{C})$ as follows:

$$f_1 = \begin{pmatrix} 1 & t_1 \\ 0 & 1 \end{pmatrix}, f_2 = \begin{pmatrix} 1 & it_2 \\ 0 & 1 \end{pmatrix},$$
$$f_3 = \begin{pmatrix} \exp(\frac{t_3}{2}) & 0 \\ 0 & \exp(\frac{-t_3}{2}) \end{pmatrix}, f_4 = \begin{pmatrix} \exp(\frac{it_4}{2}) & 0 \\ 0 & \exp(\frac{-it_4}{2}) \end{pmatrix},$$
$$f_5 = \begin{pmatrix} 1 & 0 \\ -t_5 & 1 \end{pmatrix}, f_6 = \begin{pmatrix} 1 & 0 \\ it_6 & 1 \end{pmatrix}.$$

They are one-parameter subgroups relating to the following vectors in the Lie algebra of $\mathfrak{sl}(2,\mathbb{C})$, see Section 2.1.8.

$$v_1 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, v_2 = \begin{pmatrix} 0 & i \\ 0 & 0 \end{pmatrix}, v_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$
$$v_4 = \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix}, v_5 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, v_6 = \begin{pmatrix} 0 & 0 \\ i & 0 \end{pmatrix}.$$

As can be seen v_i, $i = 1, 2, \ldots, 6$ are the basis of $\mathfrak{sl}(2,\mathbb{C})$. Therefore combinations of f_i, $i = 1, 2, \ldots, 6$ give $SL(2,\mathbb{C})$ and the above generators are the generators of $PSL(2,\mathbb{C})$. This group is also known as the **Möbius** group, the group of transformations of the form

$$z \mapsto \frac{az+b}{cz+d}, \quad a, b, c, d, z \in \mathbb{C}, \quad ad - bc \neq 0.$$

2) $\boldsymbol{\{\partial_x, x\partial_x + y\partial_y, (x^2-y^2)\partial_x + 2xy\partial_y\}}$:

The transformations relating to each generator are:

$$\begin{aligned}
\boldsymbol{\partial_x} &: x+iy \mapsto x+t_1,\\
\boldsymbol{x\partial_x + y\partial_y} &: z \mapsto \exp(t_2)z,\\
\boldsymbol{(x^2-y^2)\partial_x + 2xy\partial_y} &: z \mapsto \frac{z}{1-t_3 z}.
\end{aligned}$$

As mentioned these transformations correspond to transformation f_1, f_3, f_5 respectively in $SL(2,\mathbb{C})$, and they are one-parameter subgroups of:

$$v_1 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, v_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, v_5 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}.$$

These v_i are the basis of $\mathfrak{sl}(2,\mathbb{R})$, a subalgebra of $\mathfrak{sl}(2,\mathbb{C})$. So, these generators are the generators of a group $PSL(2,\mathbb{R})$:

$$f(z) = \frac{az+b}{cz+d}, ad-bc \neq 1, a,b,c,d \in \mathbb{R}. \tag{2.23}$$

We also call this group of transformations the **real Möbius** group.

3) $\boldsymbol{\{y\partial_x - x\partial_y, (1+x^2-y^2)\partial_x + 2xy\partial_y, 2xy\partial_x + (1+y^2-x^2)\partial_y\}}$:

The transformations relating to each generator are:

$$\begin{aligned}
\boldsymbol{y\partial_x - x\partial_y} &: z \mapsto \exp(it_1)z,\\
\boldsymbol{(1+x^2-y^2)\partial_x + 2xy\partial_y} &: z \mapsto \frac{\cos(t_2)z + \sin(t_2)}{-\sin(t_2)z + \cos(t_2)},\\
\boldsymbol{2xy\partial_x + (1+y^2-x^2)\partial_y} &: z \mapsto \frac{\cos(-t)z + i\sin(-t)}{i\sin(-t)z + \cos(-t)}.
\end{aligned}$$

These transformations correspond to transformation in $SU(2,\mathbb{C})$ as follows:

$$g_1 = \begin{pmatrix} \exp(\frac{it_1}{2}) & 0 \\ 0 & \exp(\frac{-it_1}{2}) \end{pmatrix}, g_2 = \begin{pmatrix} \cos(t_2) & \sin(t_2) \\ -\sin(t_2) & \cos(t_2) \end{pmatrix},$$
$$g_3 = \begin{pmatrix} \cos(-t_3) & i\sin(-t_3) \\ i\sin(-t_3) & \cos(-t_3) \end{pmatrix}.$$

They are one-parameter subgroups relating to the following vectors in the Lie algebra of $\mathfrak{su}(2,\mathbb{C})$, see Section 2.1.8.

$$w_1 = \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix}, w_2 = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, w_3 = \begin{pmatrix} 0 & i \\ i & 0 \end{pmatrix}.$$

As can be seen, w_i, $i = 1, 2, 3$, are the basis of $\mathfrak{su}(2, \mathbb{C})$. Therefore combinations of g_i, $i = 1, 2, 3$, give $SU(2, \mathbb{C})$ and the above generators are the generators of $PSU(2, \mathbb{C})$. This group is also known as the **Projective Special Unitary** group:

$$f(z) = \frac{az + b}{-\bar{b}z + \bar{a}}, a, b \in \mathbb{C}, \bar{a}a + \bar{b}b = 1. \tag{2.24}$$

4) $\boldsymbol{\{\partial_x, \partial_y, x\partial_x + y\partial_y, y\partial_x - x\partial_y\}}$:

These generators produce the transformations:

$$\begin{aligned}
\boldsymbol{\partial_x} &: x + iy \mapsto x + t_1,\\
\boldsymbol{\partial_y} &: x + iy \mapsto y + t_2,\\
\boldsymbol{x\partial_x + y\partial_y} &: z \mapsto \exp(t_3)z,\\
\boldsymbol{y\partial_x - x\partial_y} &: z \mapsto \exp(it_4)z.
\end{aligned}$$

As mentioned, these transformations correspond to transformations f_1, f_2, f_3, f_4 respectively in $SL(2, \mathbb{C})$, where they are one-parameter subgroups of:

$$v_1 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, v_2 = \begin{pmatrix} 0 & i \\ 0 & 0 \end{pmatrix}, v_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, v_4 = \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix}.$$

These v_i, are the basis of a subalgebra of $\mathfrak{sl}(2, \mathbb{C})$, and a subgroup in general form of:

$$f(z) = az + b, a, b \in \mathbb{C}. \tag{2.25}$$

This group of transformations is known as the **Similarity** group.

5) $\boldsymbol{\{\partial_x, \partial_y, \alpha(x\partial_x + y\partial_y) + y\partial_x - x\partial_y\}, \alpha \geq 0}$: These generators generate the transformations:

$$\begin{aligned}
\boldsymbol{\partial_x} &: x + iy \mapsto x + t_1,\\
\boldsymbol{\partial_y} &: x + iy \mapsto y + t_2,\\
\boldsymbol{\alpha(x\partial_x + y\partial_y) + y\partial_x - x\partial_y} &: z \mapsto \exp((\alpha - i)t_3)z.
\end{aligned}$$

Combining these transformations leads to the general form of:

$$f(z) = \exp(r(\alpha - i))z + b, r \in \mathbb{R}, b \in \mathbb{C}. \tag{2.26}$$

We call this group **Spiral** because the orbit of a point in the plane has a spiral shape. If $\alpha = 0$ then the transformation is known as **Rigid**. Also, the set of rigid transformations forms a group.

6) $\{\boldsymbol{\partial_x}, \boldsymbol{\partial_y}, \boldsymbol{x\partial_x}, \boldsymbol{y\partial_y}, \boldsymbol{x\partial_y}, \boldsymbol{y\partial_x}, \boldsymbol{x^2\partial_x + xy\partial_y}, \boldsymbol{xy\partial_x + y^2\partial_y}\}$:

These generators generate the transformations:

$$\begin{aligned}
\boldsymbol{\partial_x} &: (x, y) \mapsto (x + t_1, y),\\
\boldsymbol{\partial_y} &: (x, y) \mapsto (x, y + t_2),\\
\boldsymbol{x\partial_x} &: (x, y) \mapsto (\exp(t_3)x, y),\\
\boldsymbol{y\partial_y} &: (x, y) \mapsto (x, \exp(t_4)y),\\
\boldsymbol{x\partial_y} &: (x, y) \mapsto (x, t_5x + y),\\
\boldsymbol{y\partial_x} &: (x, y) \mapsto (x + t_6y, y),\\
\boldsymbol{x^2\partial_x + xy\partial_y} &: (x, y) \mapsto (\frac{x}{1 - t_7x}, \frac{y}{1 - t_7x}),\\
\boldsymbol{xy\partial_x + y^2\partial_y} &: (x, y) \mapsto (\frac{x}{1 - t_8y}, \frac{y}{1 - t_8y}).
\end{aligned}$$

These transformations correspond to transformation in $SL(3, \mathbb{R})$ as follows:

$$h_1 = \begin{pmatrix} 1 & 0 & t_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, h_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & t_2 \\ 0 & 0 & 1 \end{pmatrix},$$

$$h_3 = \begin{pmatrix} \exp(\frac{2t_3}{3}) & 0 & 0 \\ 0 & \exp(\frac{-t_3}{3}) & 0 \\ 0 & 0 & \exp(\frac{-t_3}{3}) \end{pmatrix}, h_4 = \begin{pmatrix} \exp(\frac{-t_4}{3}) & 0 & 0 \\ 0 & \exp(\frac{2t_4}{3}) & 0 \\ 0 & 0 & \exp(\frac{-t_4}{3}) \end{pmatrix},$$

$$h_5 = \begin{pmatrix} 1 & 0 & 0 \\ t_5 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, h_6 = \begin{pmatrix} 1 & t_6 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$h_7 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -t_7 & 0 & 1 \end{pmatrix}, h_8 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -t_8 & 1 \end{pmatrix}.$$

They are one-parameter subgroups relating to the following vector in the Lie algebra of $\mathfrak{sl}(3,\mathbb{R})$:

$$v_1 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, v_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix},$$
$$v_3 = \begin{pmatrix} \frac{2}{3} & 0 & 0 \\ 0 & \frac{-1}{3} & 0 \\ 0 & 0 & \frac{-1}{3} \end{pmatrix}, v_4 = \begin{pmatrix} \frac{-1}{3} & 0 & 0 \\ 0 & \frac{2}{3} & 0 \\ 0 & 0 & \frac{-1}{3} \end{pmatrix},$$
$$v_5 = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, v_6 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$
$$v_7 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, v_8 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

The v_i, $i = 1, 2, \ldots, 8$ are the basis of $\mathfrak{sl}(3,\mathbb{R})$. Therefore combinations of h_i, $i = 1, 2, \ldots, 8$ give $SL(3,\mathbb{R})$ and the above generators are the generators of $PSL(3,\mathbb{R})$ in general form:

$$f(x,y) = \left(\frac{a_1x + b_1y + c_1}{a_3x + b_3y + c_3}, \frac{a_2x + b_2y + c_2}{a_3x + b_3y + c_3}\right), \det \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix} = 1. \qquad (2.27)$$

In image registration, this group of transformations is called the projective group, see Chapter 1.

7) $\boldsymbol{\{\partial_x, \partial_y, x\partial_x, y\partial_y, x\partial_y, y\partial_x\}}$:

These generators generate the transformations:

$$\begin{aligned} \boldsymbol{\partial_x} &: (x,y) \mapsto x + t_1, \\ \boldsymbol{\partial_y} &: (x,y) \mapsto y + t_2, \\ \boldsymbol{x\partial_x} &: (x,y) \mapsto \exp(t_3)x, \\ \boldsymbol{y\partial_y} &: (x,y) \mapsto \exp(t_4)y, \\ \boldsymbol{x\partial_y} &: (x,y) \mapsto t_5x + y, \\ \boldsymbol{y\partial_x} &: (x,y) \mapsto t_6y + x. \end{aligned}$$

As mentioned, these transformations correspond to transformations h_1, h_2, h_3, h_4, h_5 and h_6 in $SL(3,\mathbb{R})$, where they are one-parameter subgroups of v_i, $i = 1, 2, \ldots, 6$. The v_i, $i = 1, 2, \ldots, 6$ are the basis of a subalgebra of $\mathfrak{sl}(3,\mathbb{R})$, and the subgroup has a general form of:

$$A = \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ 0 & 0 & c_3 \end{pmatrix}, \det(A) = 1.$$

The corresponding transformation in $PSL(3,\mathbb{R})$ has a general form of:

$$f\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} T_1 \\ T_2 \end{pmatrix}, a, b, c, d, T_1, T_2 \in \mathbb{R}, ad - bc \neq 0.$$

This group of transformations has six parameters and is known as the **Affine** group.

8) $\{\boldsymbol{\partial_x}, \boldsymbol{\partial_y}, \boldsymbol{x\partial_x - y\partial_y}, \boldsymbol{y\partial_x}, \boldsymbol{x\partial_y}\}$:

These generators generate the transformations:

$$\begin{aligned} \boldsymbol{\partial_x} &: (x,y) \mapsto x + t_1, \\ \boldsymbol{\partial_y} &: (x,y) \mapsto y + t_2, \\ \boldsymbol{x\partial_x - y\partial_y} &: (x,y) \mapsto (\exp(t_3)x, \exp(-t_3)y), \\ \boldsymbol{y\partial_x} &: (x,y) \mapsto t_4 y + x, \\ \boldsymbol{x\partial_y} &: (x,y) \mapsto t_5 x + y. \end{aligned}$$

The transformation $(x,y) \mapsto (\exp(t_3)x, \exp(-t_3)y)$ corresponds to a transformation in $SL(3,\mathbb{R})$ that is the one-parameter subgroup associated with

$$v_9 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Therefore, the transformations correspond to transformations in $SL(3,\mathbb{R})$ that are the one-parameter subgroups associated with v_1, v_2, v_9, v_5, v_6. These vectors are a basis of a subalgebra in $\mathfrak{sl}(3,\mathbb{R})$, where it is the Lie algebra of a subgroup of $SL(3,\mathbb{R})$ as follows:

$$A = \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ 0 & 0 & 1 \end{pmatrix}, \ a_1 b_2 - a_2 b_1 = 1.$$

The group of transformations corresponding to this subgroup of $PSL(3,\mathbb{R})$ is:

$$f\begin{pmatrix}x\\y\end{pmatrix}=\begin{pmatrix}a&b\\c&d\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}+\begin{pmatrix}T_1\\T_2\end{pmatrix}, a,b,c,d,T_1,T_2\in\mathbb{R}, ad-bc=1.$$

This group of transformations is known as the **Special Affine** group.

9) $\{\boldsymbol{\partial_x, 2x\partial_x + y\partial_y, x^2\partial_x + xy\partial_y}\}$:

These generators generate the transformations:

$$\begin{aligned}\boldsymbol{\partial_x}&:(x,y)\mapsto x+t_1,\\ \boldsymbol{2x\partial_x+y\partial_y}&:(x,y)\mapsto(\exp(2t_2)x,\exp(t_2)y),\\ \boldsymbol{x^2\partial_x+xy\partial_y}&:(x,y)\mapsto(\frac{x}{1-t_3x},\frac{y}{1-t_3x}).\end{aligned}$$

The transformation $(x,y)\mapsto(\exp(2t_2)x,\exp(t_2)y)$ corresponds to a transformation in $SL(3,\mathbb{R})$:

$$h_{10}=\begin{pmatrix}\exp(t_2)&0&0\\0&1&0\\0&0&\exp(-t_2)\end{pmatrix},$$

where it is the one-parameter subgroup associated with:

$$v_{10}=\begin{pmatrix}1&0&0\\0&0&0\\0&0&-1\end{pmatrix}.$$

Therefore, the transformations correspond to transformations in $SL(3,\mathbb{R})$ that are the one-parameter subgroups associated with v_1, v_{10}, v_7. These vectors are a basis of a subalgebra in $\mathfrak{sl}(3,\mathbb{R})$, where it is the Lie algebra of a subgroup of $SL(3,\mathbb{R})$ as follows:

$$A=\begin{pmatrix}a_1^2&0&c_1\\0&a_1&0\\a_3&0&c_3\end{pmatrix}, \det(A)=1.$$

The group of transformations corresponding to this subgroup of $PSL(3,\mathbb{R})$ is:

$$f(x,y)=(\frac{a_1^2x+c_1}{a_3x+c_3},\frac{a_1y}{a_3x+c_3}), a_1,c_1,a_3,c_3\in\mathbb{R}, \det(A)=1.$$

We denote this group with $SPSL(3,\mathbb{R})$.

10) $\{\boldsymbol{\partial_x}, \boldsymbol{x\partial_x}, \boldsymbol{y\partial_y}, \boldsymbol{x^2\partial_x + xy\partial_y}\}$:

These generators generate the transformations:

$$\begin{aligned}
\boldsymbol{\partial_x} &: (x,y) \mapsto x + t_1, \\
\boldsymbol{x\partial_x} &: (x,y) \mapsto (\exp(t_2)x, y), \\
\boldsymbol{y\partial_y} &: (x,y) \mapsto (x, \exp(t_3)y), \\
\boldsymbol{x^2\partial_x + xy\partial_y} &: (x,y) \mapsto (\frac{x}{1-t_4x}, \frac{y}{1-t_4x}).
\end{aligned}$$

These transformations correspond to transformations h_1, h_3, h_4, h_7 in $SL(3,\mathbb{R})$, where they are one-parameter subgroups of v_i, $i = 1,3,4,7$. The v_i, $i = 1,3,4,7$ are the basis of a subalgebra of $\mathfrak{sl}(3,\mathbb{R})$, and the subgroup has a general form of:

$$A = \begin{pmatrix} a_1 & 0 & c_1 \\ 0 & b_2 & 0 \\ a_3 & 0 & c_3 \end{pmatrix}, \det(A) = 1.$$

The group of transformations corresponding to this subgroup of $PSL(3,\mathbb{R})$ is:

$$f(x,y) = (\frac{a_1x + c_1}{a_3x + c_3}, \frac{b_2y}{a_3x + c_3}), a_1, c_1, b_2, a_3, c_3 \in \mathbb{R}, \det(A) = 1. \tag{2.28}$$

We denote this group with $GPSL(3,\mathbb{R})$.

The transformations relating to the following generators are obtained in a similar way, so we just give the transformations.

11) $\{\boldsymbol{\partial_x}, \boldsymbol{\partial_y}, \boldsymbol{x\partial_x}, \boldsymbol{y\partial_y}, \boldsymbol{x^2\partial_x}, \boldsymbol{y^2\partial_y}\}$: These generators generate the transformations:

$$\begin{aligned}
\boldsymbol{\partial_x} &: (x,y) \mapsto x + t_1, \\
\boldsymbol{\partial_y} &: (x,y) \mapsto y + t_2, \\
\boldsymbol{x\partial_x} &: (x,y) \mapsto \exp(t_3)x, \\
\boldsymbol{y\partial_y} &: (x,y) \mapsto \exp(t_4)y, \\
\boldsymbol{x^2\partial_x} &: (x,y) \mapsto \frac{x}{1-t_5x}, \\
\boldsymbol{y^2\partial_y} &: (x,y) \mapsto \frac{y}{1-t_6y}.
\end{aligned}$$

So, they are the generators of:

$$f(x,y) = \left(\frac{a_1x + b_1}{c_1x + d_1}, \frac{a_2y + b_2}{c_2y + d_2}\right), \qquad a_id_i - b_ic_i = 1, i = 1,2. \tag{2.29}$$

We denote this group with $PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$[10].

12) $\boldsymbol{\{\partial_x + \partial_y, x\partial_x + y\partial_y, x^2\partial_x + y^2\partial_y\}}$: These generators generate the transformations:

$$\begin{aligned}
\boldsymbol{\partial_x + \partial_y} &: (x,y) \mapsto (x+t_1, y+t_1),\\
\boldsymbol{x\partial_x + y\partial_y} &: (x,y) \mapsto (\exp(t_2)x, \exp(t_2)y),\\
\boldsymbol{x^2\partial_x + y^2\partial_y} &: (x,y) \mapsto \left(\frac{x}{1-t_3x}, \frac{y}{1-t_3y}\right).
\end{aligned}$$

Combining these transformations and dividing the coefficients by the determinant to eliminate the scale factor leads to a general form of:

$$f(x,y) = \left(\frac{ax+b}{cx+d}, \frac{ay+b}{cy+d}\right), \qquad ad-bc=1. \tag{2.30}$$

We denote this group by:

$EPSL(2,\mathbb{R})$.

13) $\boldsymbol{\{\partial_x, \partial_y, x\partial_x, y\partial_y, x^2\partial_x\}}$: These generators generate the transformations:

$$\begin{aligned}
\boldsymbol{\partial_x} &: (x,y) \mapsto x+t_1,\\
\boldsymbol{\partial_y} &: (x,y) \mapsto y+t_2,\\
\boldsymbol{x\partial_x} &: (x,y) \mapsto \exp(t_3)x,\\
\boldsymbol{x^2\partial_x} &: (x,y) \mapsto \left(\exp(t_4)y, \frac{x}{1-t_5x}\right).
\end{aligned}$$

Combining these transformations and dividing the coefficients by the determinant to eliminate the scale factor leads to a general form of:

$$f(x,y) = \left(\frac{a_1x+b_1}{c_1x+d_1}, a_2y+b_2\right), \qquad a_1d_1 - b_1c_1 = 1, a_1, a_2, b_1, b_2, c_1, d_1 \in \mathbb{R}. \tag{2.31}$$

We denote this group with $PSL(2,\mathbb{R}) \times Sim$ (Sim stands for similarity).

[10]Let $(G,\circ)$ and $(H,*)$ be groups. The product of G and H is $G \times H$, which is a group, where:

- $\{(g,h), g \in G, h \in H\}$ is the element of $G \times H$.
- $\square$ is a group operation such that $(g_1,h_1)\square(g_2,h_2) = (g_1 \circ g_2, h_1 * h_2)$.

14) $\{\boldsymbol{\partial_x}, \boldsymbol{\partial_y}, \boldsymbol{x\partial_x}, \boldsymbol{x^2\partial_x}\}$: These generators generate the transformations:

$$\begin{aligned}\boldsymbol{\partial_x} &: (x,y) \mapsto x + t_1,\\ \boldsymbol{\partial_y} &: (x,y) \mapsto y + t_2,\\ \boldsymbol{x\partial_x} &: (x,y) \mapsto \exp(t_3)x,\\ \boldsymbol{x^2\partial_x} &: (x,y) \mapsto \frac{x}{1-t_4x}.\end{aligned}$$

Combining these transformations and dividing the coefficients by the determinant to eliminate the scale factor leads to a general form of:

$$f(x,y) = \left(\frac{a_1x+b_1}{c_1x+d_1}, y+b_2\right), \quad a_1d_1 - b_1c_1 = 1, a_1, b_1, b_2, c_1, d_1 \in \mathbb{R}. \tag{2.32}$$

We denote this group with $PSL(2,\mathbb{R}) \times Trans$ ($Trans$ stands for translation).

15) $\{\boldsymbol{\partial_x}, \boldsymbol{\partial_y}, \boldsymbol{x\partial_x}, \boldsymbol{y\partial_y}\}$: These generators generate the transformations:

$$\begin{aligned}\boldsymbol{\partial_x} &: (x,y) \mapsto x + t_1,\\ \boldsymbol{\partial_y} &: (x,y) \mapsto y + t_2,\\ \boldsymbol{x\partial_x} &: (x,y) \mapsto \exp(t_3)x,\\ \boldsymbol{y\partial_y} &: (x,y) \mapsto \exp(t_4)y.\end{aligned}$$

Combining these transformations leads to a general form of:

$$f(x,y) = (a_1x + b_1, a_2y + b_2). \tag{2.33}$$

We denote this group with $Sim \times Sim$.

16) $\{\boldsymbol{\partial_x}, \boldsymbol{\partial_y}, \boldsymbol{x\partial_x + \alpha y\partial_y}, \quad \boldsymbol{0 < |\alpha| \leq 1}\}$: These generators generate the transformations:

$$\begin{aligned}\boldsymbol{\partial_x} &: (x,y) \mapsto x + t_1,\\ \boldsymbol{\partial_y} &: (x,y) \mapsto y + t_2,\\ \boldsymbol{x\partial_x + \alpha y\partial_y} &: (x,y) \mapsto (\exp(t_3)x, \exp(\alpha t_3)y).\end{aligned}$$

Combining these transformations leads to a general form of:

$$f(x,y) = (\exp(t)x + b_1, \exp(\alpha t)y + b_2), t, b_1, b_2 \in \mathbb{R}. \tag{2.34}$$

We denote this group with $\exp \times Trans$.

2.3 Groups and Subgroups

In the previous section a list of planar Lie groups were given. Notice that there are three main groups: $PSL(2,\mathbb{C})$, $PSL(3,\mathbb{R})$ and $PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$ and their subgroups.

2.3.1 Subgroups of $PSL(2,\mathbb{C})$

In the previous section, we observed that some of the given groups are the subgroups of $SL(2,\mathbb{C})$ and so subgroups of $PSL(2,\mathbb{C})$; they are listed as follows:

- $PSU(2,\mathbb{C}) = \{\frac{az+b}{-\bar{b}z+\bar{a}}, a, b, \in \mathbb{C}, \bar{a}a + \bar{b}b = 1\}$.
- $PSL(2,\mathbb{R}) = \{\frac{az+b}{cz+d}, a, b, c \in \mathbb{R}, ad - bc = 1\}$.
- **Similarity**$=\{az + b, a, b \in \mathbb{C}\}$. The following groups are also subgroups of similarity:
  - **Rigid**: $\{\exp(i\theta)z + b, \theta \in [0, 2\pi], b \in \mathbb{C}\}$.
  - **Rotation**: $\{\exp(i\theta)z, \theta \in [0, 2\pi]\}$.
  - **Translation**: $\{z + b, b \in \mathbb{C}\}$.
  - **Scale**: $\{rz, r \in \mathbb{R}\}$.
- **Spiral**$=\{\exp(\alpha - i\theta)z + b, \theta \in [0, 2\pi], b \in \mathbb{C}\}$, $\alpha \in \mathbb{R}$ is a constant.
- Another group which is not listed as a group in [26], and that we are interested in, is the group with generators: $\boldsymbol{\{(x^2 - y^2)\partial_x + 2xy\partial_y, 2xy\partial_x + (y^2 - x^2)\partial_y\}}$. The general form of this transformation is:

  $$f(z) = \frac{z}{az + 1}, a \in \mathbb{C}. \tag{2.35}$$

  We call this group the **Non-linear part of Möbius** and denote it by $NPSL(2,\mathbb{C})$. It is obviously a subgroup of $PSL(2,\mathbb{C})$. Also, any transformation $k(z) = \frac{z}{cz+1}$ corresponds to A:

  $$\left\{A; A = \begin{pmatrix} 1 & 0 \\ c & 1 \end{pmatrix}\right\},$$

  and the set of such A is a subgroup of $SL(2,\mathbb{C})$.

  The interesting thing that this group brings to us is that it does not include any similarities. In other words, similarity and $NPSL(2,\mathbb{C})$ are two disjoint subgroups of $PSL(2,\mathbb{C})$, and each Möbius transformation can be given as a composition of them.

2.3.2 Subgroups of $PSL(3,\mathbb{R})$

Subgroups of $PSL(3,\mathbb{R})$ are listed as follows:

- **Affine**$= \{f : f(x,y) = (ax+by+t_1, cx+dy+t_2), ad-bc \neq 0\}$.
- **Special affine**$= \{f : f(x,y) = (ax+by+t_1, cx+dy+t_2), ad-bc = 1\}$.
- $SPSL(3,\mathbb{R})$$= \{f : f(x,y) = (\frac{a^2x+b}{cx+d}, \frac{ay}{cx+d})\}$.
- $GPSL(3,\mathbb{R})$$= \{f : f(x,y) = (\frac{a_1x+b}{cx+d}, \frac{a_2y}{cx+d})\}$.

2.3.3 Subgroups of $PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$

The group $PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$ acts on each axis independently. This group is isomorphic to $SL(2,\mathbb{R})/\pm I \times SL(2,\mathbb{R})/\pm I$. So its subgroups are:

- $PSL(2,\mathbb{R}) \times Sim$ is a subgroup of $PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$. Because each transformation
$$(x,y) \mapsto \left(\frac{a_1x+b_1}{c_1x+d_1}, a_2y+b_2\right),$$
corresponds to a matrix in $SL(2,\mathbb{R})/\pm I \times SL(2,\mathbb{R})/\pm I$,
$$\left(\begin{pmatrix} a_1 & b_1 \\ c_1 & d_1 \end{pmatrix}, \begin{pmatrix} \sqrt{a_2} & \frac{b_2}{\sqrt{a_2}} \\ 0 & \frac{1}{\sqrt{a_2}} \end{pmatrix}\right), \quad a_1d_1 - b_1c_1 = 1,$$
and the set of all such matrices is also a subgroup of $SL(2,\mathbb{R})/\pm I \times SL(2,\mathbb{R})/\pm I$.
- $EPSL(2,\mathbb{R})$ is a subgroup of $PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$, because each transformation
$$(x,y) \mapsto \left(\frac{ax+b}{cx+d}, \frac{ay+b}{cy+d}\right),$$
corresponds to a matrix in $SL(2,\mathbb{R})/\pm I \times SL(2,\mathbb{R})/\pm I$,
$$\left(\begin{pmatrix} a & b \\ c & d \end{pmatrix}, \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \quad ad-bc = 1\right),$$

and the set of such matrices is also a subgroup of $SL(2,\mathbb{R})/\pm I \times SL(2,\mathbb{R})/\pm I$.

- $PSL(2,\mathbb{R}) \times Trans$ is a subgroup of $PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$, because each transformation
$$(x,y) \mapsto \left(\frac{ax+b}{cx+d}, y+e\right)$$
corresponds to a matrix in $SL(2,\mathbb{R})/\pm I \times SL(2,\mathbb{R})/\pm I$,
$$\left(\begin{pmatrix} a & b \\ c & d \end{pmatrix}, \begin{pmatrix} 1 & e \\ 0 & 1 \end{pmatrix}\right), \quad ad-bc=1,$$
and the set of all such matrices is also a subgroup of $SL(2,\mathbb{R})/\pm I \times SL(2,\mathbb{R})/\pm I$.

- $Sim \times Sim$ is a subgroup of $PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$, because each transformation
$$(x,y) \mapsto (a_1x+b_1, a_2y+b_2)$$
corresponds to a matrix in $SL(2,\mathbb{R})/\pm I \times SL(2,\mathbb{R})/\pm I$,
$$\left(\begin{pmatrix} \sqrt{a_1} & \frac{b_1}{\sqrt{a_1}} \\ 0 & \frac{1}{\sqrt{a_1}} \end{pmatrix}, \begin{pmatrix} \sqrt{a_2} & \frac{b_2}{\sqrt{a_2}} \\ 0 & \frac{1}{\sqrt{a_2}} \end{pmatrix}\right),$$
and the set of all such matrices is also a subgroup of $SL(2,\mathbb{R})/\pm I \times SL(2,\mathbb{R})/\pm I$.

- $\exp \times Trans$ is a subgroup of $PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$, because each transformation
$$(x,y) \mapsto (\exp(t)x+b_1, \exp(\alpha t)y+b_2).$$
corresponds to a matrix in $SL(2,\mathbb{R})/\pm I \times SL(2,\mathbb{R})/\pm I$,
$$\left(\begin{pmatrix} \sqrt{\exp(t)} & \frac{b_1}{\sqrt{\exp(t)}} \\ 0 & \frac{1}{\sqrt{\exp(t)}} \end{pmatrix}, \begin{pmatrix} \sqrt{\exp(\alpha t)} & \frac{b_2}{\sqrt{\exp(\alpha t)}} \\ 0 & \frac{1}{\sqrt{\exp(\alpha t)}} \end{pmatrix}\right),$$
and the set of all such matrices is also a subgroup of $SL(2,\mathbb{R})/\pm I \times SL(2,\mathbb{R})/\pm I$.

$$\begin{tikzcd}
 & & PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R}) \arrow[dl] \arrow[dr] & \\
 & PSL(2,\mathbb{R}) \times Sim \arrow[dl] \arrow[dr] & & EPSL(2,\mathbb{R}) \\
Sim \times Sim \arrow[d] & & PSL(2,\mathbb{R}) \times Trans & \\
\exp \times Trans & & &
\end{tikzcd}$$

Figure 2.3: The lattice of $PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$ and its subgroups. Here $G_1 \longrightarrow G_2$ means G_2 is subgroup of G_1.

We now give the lattice of group and subgroups. A lattice is a partially ordered set[11], where every two elements in the lattice have a unique supremum[12] and a unique infimum[13]. In the lattice of subgroups and groups, elements are subgroups and the relation is set inclusion. The supremum of two subgroups is the group generated by their union and the infimum is their intersection. Two lattices are shown in Figures 2.3 and 2.4, where the edge $G_1 \longrightarrow G_2$ means that G_2 is a subgroup of G_1.

2.4 Properties of Planar Lie Group Actions

The action of each group on the plane has some special properties that are explained in the following sections.

[11] A partial order [60] is a binary relation '$\leq$' over a set P which is reflexive, antisymmetric, and transitive, i.e., which satisfies for all a, b, and c in P:

- $a \leq a$ (reflexivity);
- if $a \leq b$ and $b \leq a$, then $a = b$ (antisymmetry);
- if $a \leq b$ and $b \leq c$, then $a \leq c$ (transitivity).

[12] Least upper bound.

[13] Greatest lower bound.

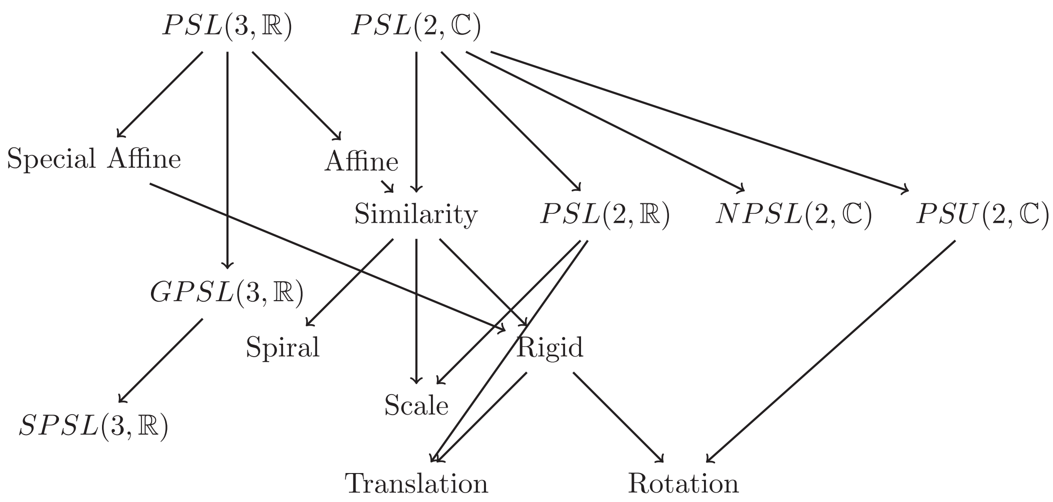


Figure 2.4: The lattice of groups $PSL(2,\mathbb{C})$ and $PSL(3,\mathbb{R})$ and their subgroups. $G_1 \longrightarrow G_2$ means G_2 is subgroup of G_1.

2.4.1 $PSL(2,\mathbb{C})$ Action on the Plane

The action of $PSL(2,\mathbb{R})$ on the plane has the following properties:

1. Under the $PSL(2,\mathbb{C})$ action on the plane, infinitesimal angles are preserved, because $z \mapsto \frac{az+b}{cz+d}$ is conformal. For example, in Figure 2.5 the rectangular grid (left) is mapped by $f(z) = \frac{(1+2i)z+0.2+0.1i}{(0.2-0.1i)z+0.21-0.42i}$ to another grid (right). It can be seen that the transformation maps pairs of lines intersection at right angles to pairs of curves still intersecting at 90 degrees.

2. The *cross-ratio* is constant under $PSL(2,\mathbb{C})$ transformations. Let $z_1, z_2, z_3, z_4 \in \mathbb{C}$. The cross-ratio of them is defined as:

$$\frac{(z_1-z_2)(z_3-z_4)}{(z_1-z_3)(z_2-z_4)}. \tag{2.36}$$

   Suppose z_i, $i = 1,2,3,4$ are transformed by $f(z) = \frac{az+b}{cz+d}$. Then

$$f(z_1) - f(z_2) = \frac{az_1+b}{cz_1+d} - \frac{az_2+b}{cz_2+d} = \frac{z_1-z_2}{(cz_1+d)(cz_2+d)}$$

   and their cross-ratio after transformation is:

$$\frac{\frac{z_1-z_2}{(cz_1+d)(cz_2+d)}\frac{z_3-z_4}{(cz_3+d)(cz_4+d)}}{\frac{z_1-z_3}{(cz_1+d)(cz_3+d)}\frac{z_2-z_4}{(cz_2+d)(cz_4+d)}} = \frac{(z_1-z_2)(z_3-z_4)}{(z_1-z_3)(z_2-z_4)}.$$

Therefore, the cross-ratio of points is constant under the action of $PSL(2, \mathbb{C})$.

3. $PSL(2, \mathbb{C})$ transformations map lines and circles to lines and circles. Let $\varphi(z) = \frac{az+b}{cz+d}$ be a $PSL(2, \mathbb{C})$ transformation. $\varphi(z)$ can be written as:

$$\varphi(z) = \frac{az+b}{cz+d} = \frac{c(az + \frac{1+ad}{c})}{c(cz+d)} = \frac{a}{c} + \frac{1}{c(cz+d)}.$$

So, φ is the composition of three transformations: similarity, inversion ($\frac{1}{z}$), and translation. Similarity and translation map circles to circles and lines to lines. It is enough to study the action of inversion on the plane.

Inversion is a function on the extended complex plane $f : \mathbb{C} \bigcup \{\infty\} \to \mathbb{C} \bigcup \{\infty\}$, where $f(0) = \infty$ and $f(\infty) = 0$. f maps lines and circles to lines and circles.

- A line passing through the origin consists of all points $z = x + iy$ where $y = mx$ for some m. By inversion it is mapped to:

$$\frac{1}{x+imx} = \frac{x-imx}{x^2+(mx)^2} = \frac{x}{x^2+(mx)^2} - i\frac{mx}{x^2+(mx)^2}.$$

  It follows that:

$$(x, y) \mapsto \left(\frac{1}{x+m^2x}, -\frac{m}{x+m^2x}\right).$$

  Let $X = \frac{1}{x+m^2x}$ and $Y = -\frac{m}{x+m^2x}$, then $Y = mX$, which defines a straight line. Note that zero is mapped to ∞ and ∞ is mapped to zero.

- A line that does not pass through the origin consists of all the points $z = x + iy$, where $ax + by = c$, $a, b, c \in \mathbb{R}$, $c \neq 0$. A point $z = x + iy$ is mapped by inversion to $(\frac{x}{x^2+y^2}, -\frac{y}{x^2+y^2})$ as follows:

$$\frac{1}{z} = \frac{1}{x+iy} = \frac{x-iy}{x^2+y^2} = \frac{x}{x^2+y^2} - i\frac{y}{x^2+y^2}.$$

  Therefore, a line $ax + by = c$ is mapped to:

$$a\frac{x}{x^2+y^2} - b\frac{y}{x^2+y^2} = c,$$

  which can be written as:

$$x^2 + y^2 - \frac{a}{c}x + \frac{b}{c}y = 0,$$

  which is the equation of a circle that goes through the origin.

- A circle that goes through the origin is mapped to a line that does not pass

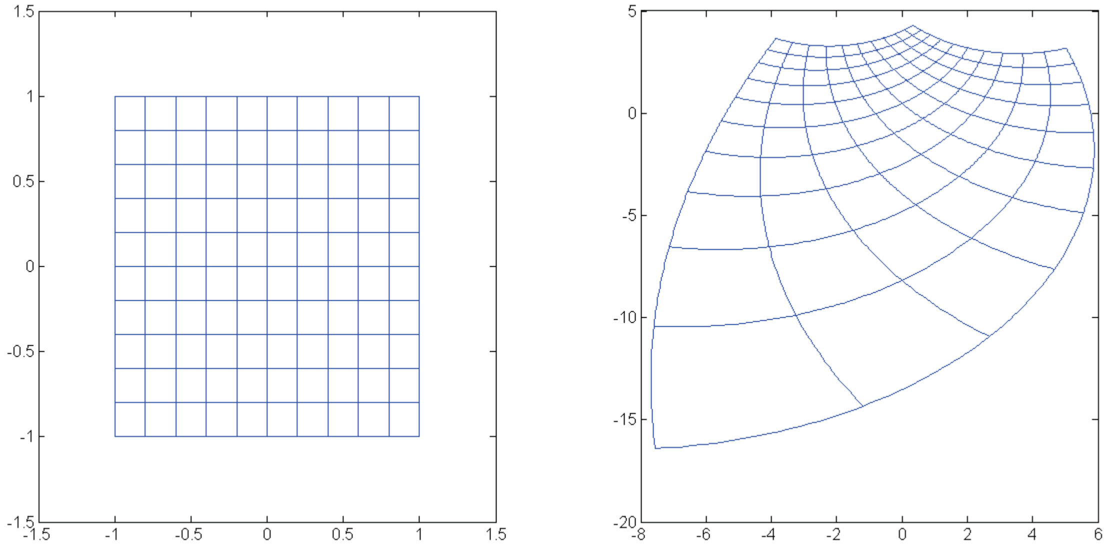


Figure 2.5: A rectangular grid (left) is mapped by $f(z) = \frac{(1+2i)z+0.2+0.1i}{(0.2-0.1i)z+0.21-0.42i} \in PSL(2, \mathbb{C})$ to another grid (right). Infinitesimal angles are preserved.

through the origin by inversion, because the inverse of an inversion is an inversion.

- A circle that does not pass through the origin is mapped to a circle that does not pass through the origin. A circle that does not pass through the origin consists of all points $z = x + iy$, where $x^2 + y^2 + ax + by = c$, $a, b, c \in \mathbb{R}$, $c \neq 0$. By inversion it is mapped to:

$$\frac{x^2}{x^2+y^2} + \frac{y^2}{x^2+y^2} + a\left(\frac{x}{x^2+y^2} - b\frac{y}{x^2+y^2}\right) = c,$$

which can be written as:

$$x^2 + y^2 - \frac{a}{c}x + \frac{b}{c}y = \frac{1}{c},$$

which is the equation of a circle that does not pass through the origin.

Therefore, under $PSL(2, \mathbb{C})$ transformations, lines and circles map to lines and circles.

2.4.1.1 Similarity

The similarity group acts on the plane by:

$$z \mapsto az + b, \quad a, b \in \mathbb{C}.$$

In polar coordinates, $a = a_1 + ia_2$ can be written as $r\exp(i\theta)$, where $r = \sqrt{a_1^2 + a_2^2}$ and $\theta = \tan^{-1}(\frac{a_2}{a_1})$. So $a = r\exp(i\theta)$; r is the scale parameter and θ is the rotation parameter. Therefore, the similarity group action on the plane includes: rotation, scale and two translations. Under the similarity action the ratio of distances is invariant:

$$\frac{|az_1 + b - az_2 - b|}{|az_3 + b - az_4 - b|} = \frac{a|z_1 - z_2|}{a|z_3 - z_4|} = \frac{|z_1 - z_2|}{|z_3 - z_4|}.$$

2.4.1.2 $PSU(2, \mathbb{C})$

There is an onto homomorphism map between the groups $SO(3, \mathbb{R})$ and $SU(2, \mathbb{C})$, where the kernel is $\pm I$ (I is the identity matrix) [68]. The group $SO(3, \mathbb{R})$ is the group of orthogonal matrices with determinant one. The action of this group on $\mathbb{R}^3$ is rotation, so the length is preserved. In fact $SU(2, \mathbb{C})$ is the universal double cover[14] of $SO(3)$, see [68] for the proof. Each matrix in $SO(3)$ can be parameterized by three Euler angles: $R(\psi, \phi, \psi')$. So, if:

$$SU(2, \mathbb{C}) = \left\{ \begin{pmatrix} a & b \\ -\bar{b} & \bar{a} \end{pmatrix} : a, b \in \mathbb{C}, \quad \bar{a}a + \bar{b}b = 1 \right\}, \tag{2.37}$$

then

$$a = \pm\cos\left(\frac{\phi}{2}\right)\exp\left(i\frac{\psi + \psi'}{2}\right), \qquad b = \pm i \sin\left(\frac{\phi}{2}\right)\exp\left(i\frac{\psi - \psi'}{2}\right);$$

see [69] for more detail.

[14] A covering space of X is the space C with a continuous surjective map $P : C \mapsto X$, such that, for every $x \in X$ there exists a neighbourhood U of x such that $P^{-1}(U)$ is the union of disjoint open sets in C [61].

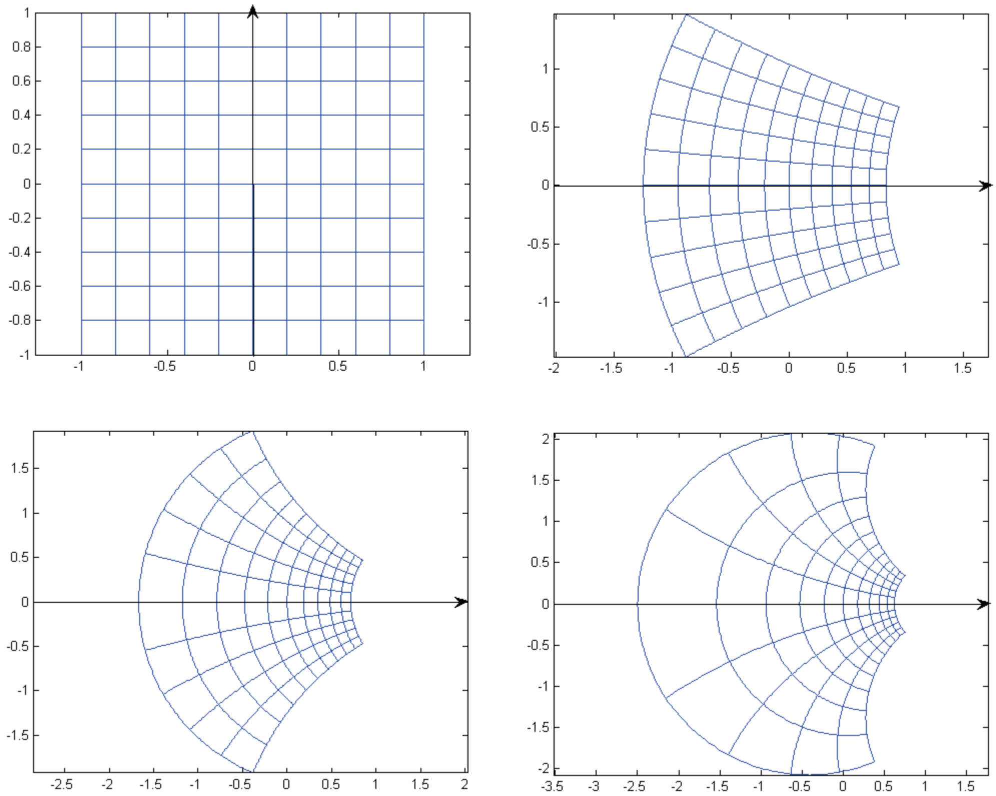


Figure 2.6: The action of G_1, Equation (2.38), on a rectangular grid (top-left), where $r = 0.2$ (top-right), $r = 0.4$ (bottom-left), $r = 0.6$ (bottom-right).

2.4.1.3 $NPSL(2, \mathbb{C})$

The action of $NPSL(2, \mathbb{C})$ on the plane is:

$$z \mapsto \frac{z}{az+1}, \quad a \in \mathbb{C}.$$

It has two subgroups:

$$G_1 = \left\{ \frac{z}{rz+1}, \quad r \in \mathbb{R} \right\} \tag{2.38}$$

$$G_2 = \left\{ \frac{z}{irz+1}, \quad r \in \mathbb{R} \right\} \tag{2.39}$$

The action of G_1 and G_2 are shown in Figures 2.6 and 2.7 respectively. As can be seen in Figure 2.6, transformation of the unit square by G_1 is symmetric with respect to the real axis, and by G_2 it is symmetric with respect to the imaginary axis.

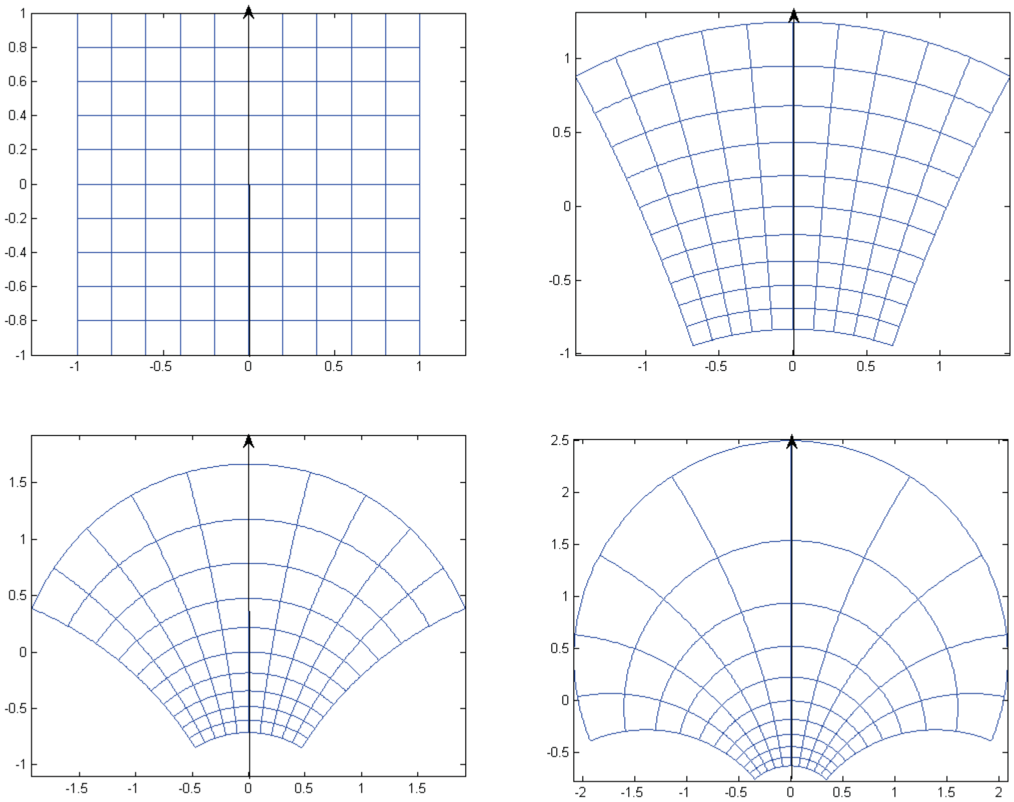


Figure 2.7: The action of G_2, Equation (2.39), on a rectangular (top-left), where $r = 0.2$ (top-right), $r = 0.4$ (bottom-left), $r = 0.6$ (bottom-right).

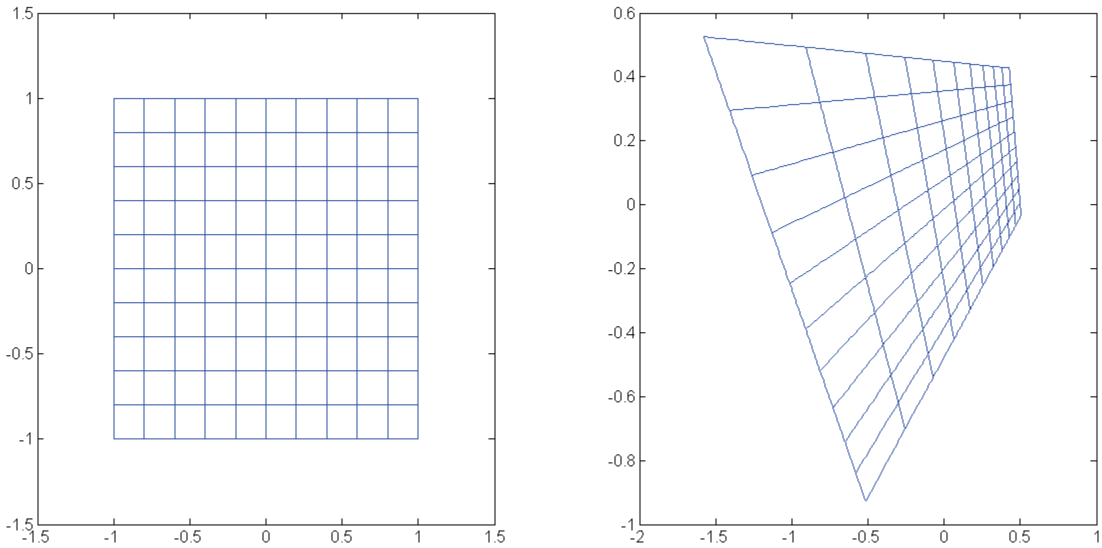


Figure 2.8: A rectangular grid (left) is mapped by a function $f(x, y) = \left(\frac{x-0.2y+0.3}{x-0.2y+1.7706}, \frac{0.4x+0.6y+0.1}{x-0.2y+1.7706}\right) \in PSL(3, \mathbb{R})$ to another grid (right).

2.4.2 $PSL(3, \mathbb{R})$ Action on the Plane

The action of $PSL(3, \mathbb{R})$ on the plane is as follows:

$$(x, y) \mapsto \left(\frac{a_1x + b_1y + c_1}{a_3x + b_3y + c_3}, \frac{a_2x + b_2y + c_2}{a_3x + b_3y + c_3}\right).$$

Figure 2.8 shows the action of $PSL(3, \mathbb{R})$ on a rectangular grid. The corresponding matrix in $SL(3, \mathbb{R})$ to $PSL(3, \mathbb{R})$ is:

$$\begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix}.$$

As was mentioned previously, $PSL(3, \mathbb{R})$ is isomorphic to $SL(3, \mathbb{R})/\pm I$. So the sub-transformations of $PSL(3, \mathbb{R})$ can be figured out from sub-transformations of $SL(3, \mathbb{R})$, which include translations, scales, shears and rotations. The translation matrices are:

$$\begin{pmatrix} 1 & 0 & t_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & t_2 \\ 0 & 0 & 1 \end{pmatrix}.$$

The scale matrices are:

$$\begin{pmatrix} t_1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} 1 & 0 & 0 \\ 0 & t_2 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

The shear matrices are:

$$\begin{pmatrix} 1 & t_1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} 1 & 0 & 0 \\ t_2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

The rotations about the x and y axes in $\mathbb{R}^3$ are:

$$\begin{pmatrix} \cos(t_1) & 0 & \sin(t_1) \\ 0 & 1 & 0 \\ -\sin(t_1) & 0 & \cos(t_1) \end{pmatrix}, \quad \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(t_2) & \sin(t_2) \\ 0 & -\sin(t_2) & \cos(t_2) \end{pmatrix},$$

respectively. Rotation about the z axis can be written as a multiplication of shear and scale matrices,

$$\begin{pmatrix} \cos(t) & \sin(t) & 0 \\ -\sin(t) & \cos(t) & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \cos(t) & 0 & 0 \\ 0 & \cos(t) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & \tan(t) & 0 \\ -\tan(t) & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

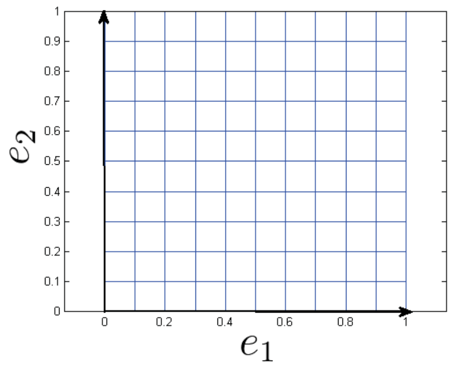
In the following, the action of $PSL(3, \mathbb{R})$ subgroups are explained. They include some of the $PSL(3, \mathbb{R})$ sub-transformations.

2.4.2.1 Affine

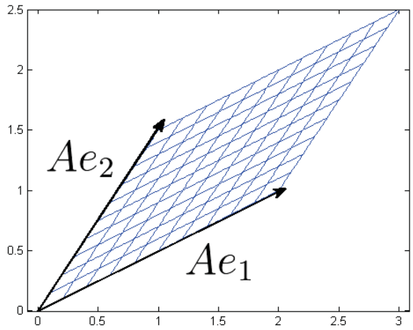
The affine group action on the plane is as follows:

$$(x, y) \mapsto (a_1 x + b_1 y + c_1, a_2 x + b_2 y + c_2), \qquad a_1 b_2 - b_1 a_2 \neq 0.$$

In matrix form the affine group is: $\begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ 0 & 0 & 1 \end{pmatrix}$. It can be figured out easily that affine action on the plane includes two scalings (in x and y directions), two shears and two translations.



(a) D before transformation.



(b) D after transformation.

Figure 2.9: D (left) is a square, which is transformed by a special affine transformation with matrix A, where $\det(A) = 1$, to the right figure. The area of D before and after transformation is equal.

2.4.2.2 Special Affine

The special affine group action is the same as the affine group action, but the determinant of the matrix is one. The transformation is area preserving. Let e_1 and e_2 be two unit basis vectors on $\mathbb{R}^2$, and A is a 2 by 2 matrix. The area of D in Figure 2.9 before transformation is $|e_1 \times e_2|$ and after the transformation is $|Ae_1 \times Ae_2| = \det(A).e_3$, where e_3 is a third unit vector in $\mathbb{R}^3$. If the determinant of A is one, then the area of D before and after the transformation is equal. Therefore, special affine transformations preserve area. The group action includes two shears, two translations, rotation about the z axis, and scale in x and y axes, such that area is preserved.

2.4.3 $PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$ Action on the Plane

Since the action of $PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$ on each one dimensional space is similar to the action of $PSL(2,\mathbb{C})$, all the mentioned properties of the action of $PSL(2,\mathbb{C})$ and its subgroups hold for the action of $PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$ and its subgroups on each one dimensional space. The action of a $PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$ transformation on $[-1, 1] \times [-1, 1]$ is shown in Figure 2.10.

So far, some finite dimensional planar Lie groups have been introduced. As mentioned, only four finite groups (rigid, similarity, affine, projective) have been applied to image registration in the past. In this chapter we have introduced more groups:

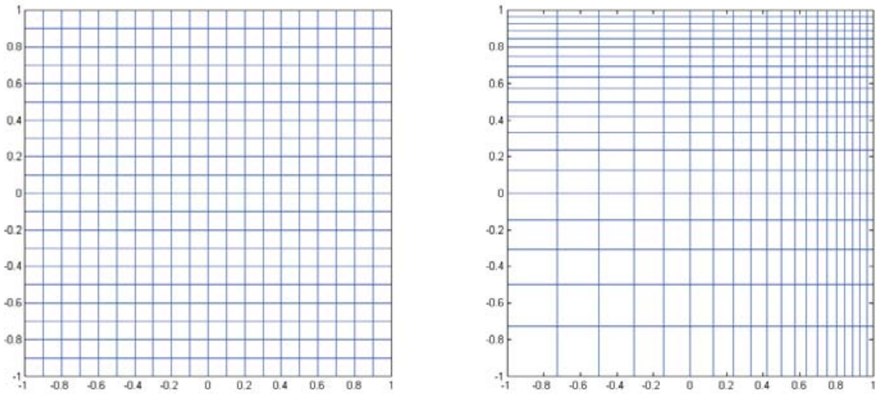


Figure 2.10: A rectangular grid (left) is transformed by a $PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$ transformation (right).

$PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$ and its subgroups, which are in total six groups, $PSL(2,\mathbb{C})$ and its subgroups: $PSU(2,\mathbb{C})$, $NPSL(2,\mathbb{C})$, and $PSL(2,\mathbb{R})$, which are in total four groups, and special affine, which is a subgroup of $PSL(3,\mathbb{R})$. As explained in Chapter 1, finite dimensional groups have primarily been used only as pre-registration for diffeomorphic registration. But in Chapters 4 and 6, we will employ these groups and show they provide benefits that justify their use as more than pre-registration.

Chapter 3

Image Registration Methodology

In the previous chapter we introduced finite dimensional Lie groups. In order to use the groups as the transformation sets for image registration we need a robust registration algorithm. The objective of this chapter is to develop such an algorithm, considering the practical and mathematical issues that arise in registration. We then present the algorithm that will be used in the rest of the thesis, and demonstrate it on several different examples.

3.1 Registration Methodology of This Research

The literature on image registration has used several different transformation sets (both rigid and non-rigid) and a variety of distance functions, as was discussed in Chapter 1. We aim to develop an algorithm that works robustly on a wide variety of images that can be used with any finite-dimensional group as the transformation set. Our algorithm is based on matching images of similar intensity, and in particular, we only consider images that are on a plain background, and that tend towards that constant intensity towards the edges of their domain. We also assume that the images to be registered have similar intensities for the matching objects. This means that the distance function we use is the L^2 distance function.

The implementation of our algorithm is code written by the author using Matlab (2010) software [43]. The L^2 optimisation is performed using the Matlab function lsqnonlin. For a vector-valued function $f(x) = (f_1(x), f_2(x), ..., f_n(x))$, lsqnonlin takes $f(x)$ as

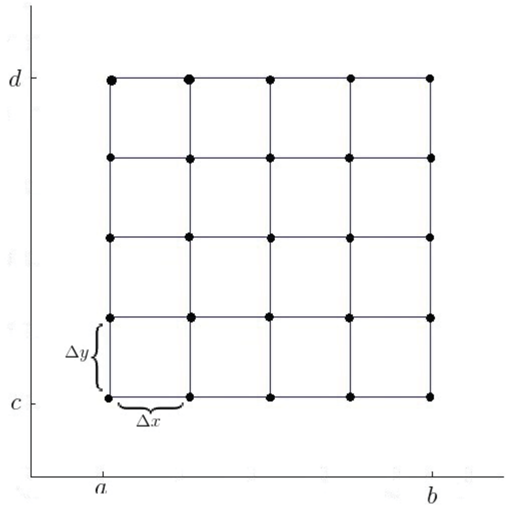


Figure 3.1: The discrete domain of an image.

input and finds the minimum of $\|f(x)\|_2^2$:

$$\mathsf{lsqnonlin}(f, x_0) = \min_x \|f(x)\|_2^2 = \min_x (f_1(x)^2 + f_2(x)^2 + ... + f_n(x)^2), \tag{3.1}$$

where x_0 is an initial value of the optimisation chosen by user. The algorithm that lsqnonlin employs for minimisation of the least square problem is based on trust-region-reflective and Levenberg-Marquardt optimisation. If there is no constraint or only bound constraints, then trust-region-reflective is a good choice, and if the number of equations are less than the dimension of the space then Levenberg-Marquardt can be used [45]. Therefore, in this research the trust-region-reflective method is employed for the minimisation of the distance function. Our methods could be applied to other distance functions if necessary, with a change of optimisation function.

Computationally, images are represented as a discrete functions. So, before showing an example of registration, we first explain how calculation is done on discrete images.

3.1.1 Image Discretization

A discretized function has a discrete domain that is a subset of the continuous domain. Let $I : \Omega \to [0, 1]$ be a continuous image such that:

$$\Omega = [a, b] \times [c, d] = \{(x, y) \in \mathbb{R}^2, a \leq x \leq b, c \leq y \leq d\}.$$

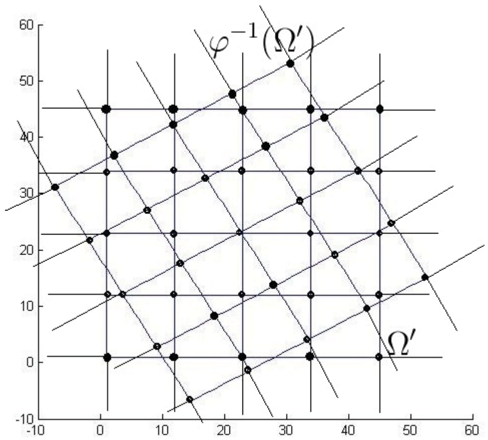


Figure 3.2: Ω' is the discrete domain. After transformation by φ^{-1}, Ω' is not matched to itself.

Choose positive real numbers $\triangle x$ and $\triangle y$. The discretized domain is :

$$\Omega' = \{(x_j, y_k) \in \Omega, x_j = j\triangle x, y_k = k\triangle y,\ j,k \in \mathbb{Z},\ \text{s.t. } a \leq j\triangle x \leq b, c \leq k\triangle y \leq d\}.$$

An example of a discretized domain is shown in Figure 3.1. Each square in the discrete domain is called a pixel. It can be represented by points on its corners or at the centre. The position of each pixel (x_j, y_k) can be represented by an integer pair (j,k). Thus the discrete domain is isomorphic to a subset of $\mathbb{Z}^2$.

The grey-scale image intensity range is typically encoded in 256 values: $0, \frac{1}{255}, \frac{2}{255}, ..., \frac{255}{255}$ ranging from black (0) to white (1). Hence a discretized image on Ω' can be represented by an $m \times n$ matrix, where its entries are $a_{jk} = I(x_j, y_k)$ such that $a_{jk} \in \{0, \frac{1}{255}, \frac{2}{255}, ..., \frac{255}{255}\}$ [71]. In this research, the discretized image space is denoted by Γ' such that $\Gamma' = \{I, I : \Omega' \to [0,1]\}$, and $[0,1]$ is considered as the discretized range.

3.1.2 Transformation of a Discretized Image

Suppose, $I \in \Gamma'$ is transformed to $I' = I \circ \varphi^{-1} \in \Gamma'$ where $id \neq \varphi \in G$. If, for $(x', y') \in \Omega'$, $(x, y) = \varphi^{-1}(x', y')$, then the value of $I'(x', y') = I \circ \varphi^{-1}(x', y') = I(x, y)$. But (x, y) may not belong to Ω', since a discrete domain after transformation is not matched to itself, see Figure 3.2.

In order to estimate the value of $I(x, y)$, we use interpolation from the neighbourhood pixels that *are* defined. The most popular interpolation methods, which trade off computational efficiency and approximation quality, are nearest neighbour, linear and cubic interpolation [74]. In this research, bilinear interpolation (a two-dimensional version of linear interpolation) is used. This method is explained in the following section. For more information about other interpolation methods, see [49].

3.1.2.1 Bilinear Interpolation

Linear interpolation is a method that interpolates a piece-wise linear function $p(x)$ through the data set $\{(x_i, f(x_i)), i = 1, \ldots, n\}$ such that:

$$\begin{cases} p\Big|_{[x_i, x_{i+1}]} \text{ is linear} \\ p(x_i) = f(x_i), i = 1, 2, \ldots, n \end{cases}.$$

Example 4. A data set of $\{(x_i, f(x_i)), i = 1, 2, 3\}$ is given in Figure 3.3. The value at point x between x_i and x_{i+1} can be approximated by linear interpolation by:

$$p(x) = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}(x - x_{i+1}) + f(x_{i+1}). \tag{3.2}$$

Equation (3.2) is a line passing through the points x_{i+1} and x_i such that $x_i < x < x_{i+1}$.

Linear interpolation through two-dimensional data is performed by bilinear interpolation. Bilinear interpolation through the set of points $\{(x_i, y_i, f(x_i, y_i)), i = 1, 2, \ldots, n\}$, where (x_i, y_i) form a rectangular mesh, is a piece-wise bilinear function $p(x, y)$ such that for $i = 1, 2, \ldots, n$:

$$\begin{cases} p\Big|_{([x_i, x_{i+1}], y)} \text{ and } p\Big|_{(x, [y_i, y_{i+1}])} \text{ are linear when } y \text{ and } x \text{ respectively are fixed variables} \\ p(x_i, y_i) = f(x_i, y_i) \end{cases}.$$

Example 5. A data set of $\{(x_i, y_i, f(x_i, y_i)), i = 1, 2\}$ is given in Figure 3.4. The value of (x', y') can be approximated by bilinear interpolation.

We have four points $(x_1, y_1), (x_1, y_2), (x_2, y_1), (x_2, y_2)$ on the vertices of a quadrilateral, see Figure 3.4. Each pairs of points are on a one-dimensional space; for example

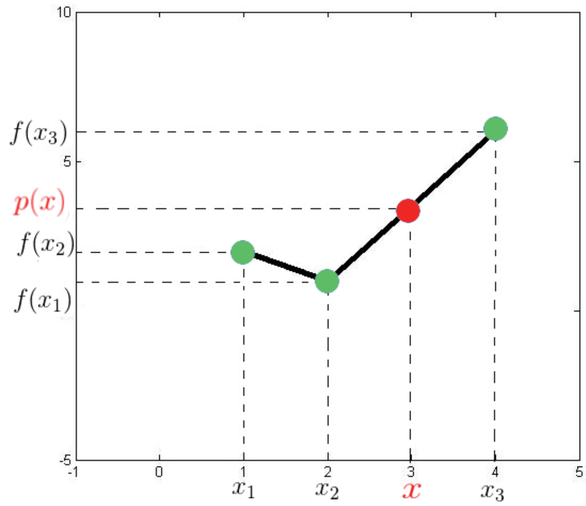


Figure 3.3: Linear interpolation through the data: x_1, x_2, x_3; $p(x)$ is an approximation at point x within the data set.

$\{(x_1, y_1), (x_2, y_1)\}$ are on the line $y = y_1$. So, the idea is to employ linear interpolation in the one-dimensional space that each pair of points lie in. We consider the pairs as $\{(x_1, y_1), (x_2, y_1)\}$, and $\{(x_1, y_2), (x_2, y_2)\}$. Then we evaluate $p(x', y_1)$ and $p(x', y_2)$ by linear interpolation:

$$p(x', y_1) = \frac{f(x_1, y_1) - f(x_2, y_1)}{x_1 - x_2}(x' - x_1) + f(x_1, y_1),$$
$$p(x', y_2) = \frac{f(x_1, y_2) - f(x_2, y_2)}{x_1 - x_2}(x' - x_1) + f(x_1, y_2).$$

Now we have two points $\{(x', y_1), (x', y_2)\}$ which lie on a one-dimensional space $x = x'$. Therefore, we again use linear interpolation for this pair and $p(x', y')$ is computed as:

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

Bilinear interpolation is a reasonable method for image registration because of the low computational costs; also, it is piece-wise continuous and differentiable almost everywhere. It is not differentiable at the grid points, so when no derivative is required, it is a good choice of interpolation [49].

The value of $I(x, y)$ between four pixels $\{(x_1, y_1), (x_1, y_2), (x_2, y_1), (x_2, y_2\}$ can be approximated by bilinear interpolation by treating I as a function. First the value of $I(x_1, y)$ and $I(x_2, y)$ are calculated. Then the value of $I(x, y)$ is calculated such that

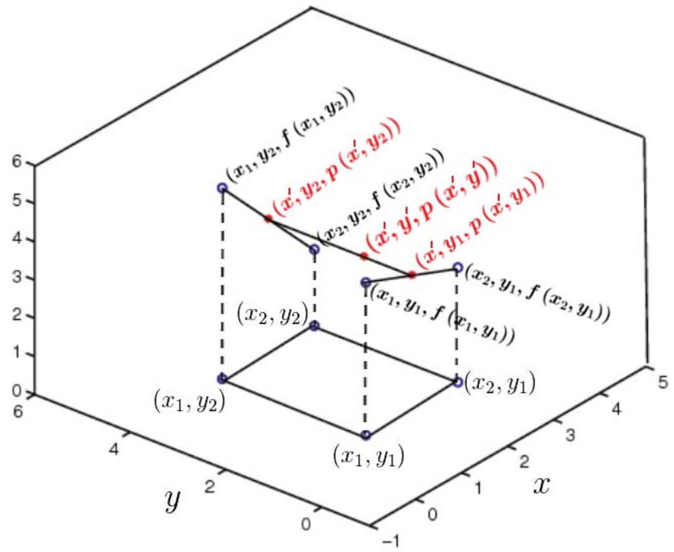


Figure 3.4: Four points $\{(x_1, y_1), (x_1, y_2), (x_2, y_1), (x_2, y_2)\}$ and their values are given. The value at point (x', y') can approximated by bilinear interpolation.

(x, y) is between (x_1, y) and (x_2, y). Figure 3.5 shows the bilinear interpolation procedure for an image I.

3.1.3 Example of Registration and Issues

In this section we show a first example of registration. We took a photo of a cup using an ordinary digital camera, as is shown in Figure 3.6a. Then we zoomed out and rotated the camera and took another photo of the cup, see Figure 3.6b.

As mentioned before, we only deal with images that have constant background. As can be seen in Figure 3.6, these images do not have constant background. So, we cut the images of the cups and put them in a solid black background using a graphics editor. The result is shown in Figure 3.7.

A registration is performed between the source and the target, where the distance function is L^2 distance. The group of transformations is similarity, because the second photo of the cup is taken by a rotation and zooming the camera lens. The similarity group depends on four real parameters: $z \mapsto az + b$, $a = t_1 + it_2, b = t_3 + it_4$. So, we

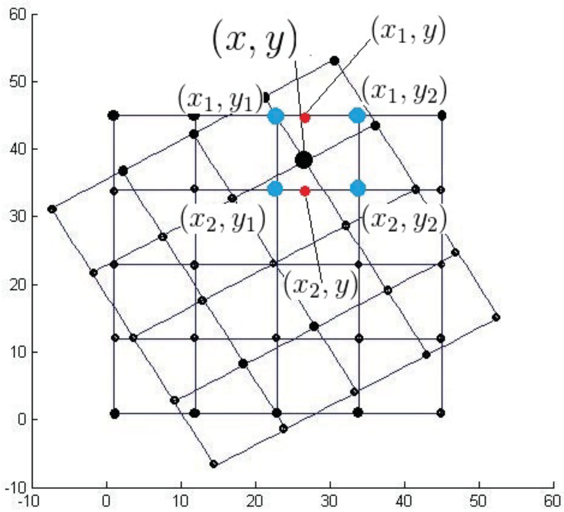


Figure 3.5: Bilinear interpolation in an image to approximate the value of $I(x, y)$.

(a) (b)

Figure 3.6: Two photos of a cup, with a camera rotation and zoom between them.

(a) Source (b) Target

Figure 3.7: The same as in Figure 3.6 after the cup is cut out and placed on a solid black background for the registration. These images can be registered with a similarity transformation.

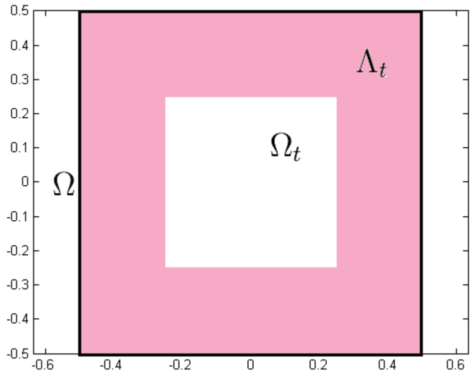


Figure 3.8: Ω is transformed by $\varphi(\Omega, \boldsymbol{t}) = \boldsymbol{t}z$ to Ω_t, where $\boldsymbol{t} = \frac{1}{2}$; $\Lambda_t = \Omega - \Omega_t$.

denote a transformation in the similarity group by $\varphi(\Omega, \boldsymbol{t})$, where $\boldsymbol{t} = (t_1, t_2, t_3, t_4)$ is the vector of parameters.

In registration, the distance function is calculated on Ω. However, some transformations are not surjective on Ω. Therefore, for some set of transformation parameters $\boldsymbol{t}$, the domain of φ^{-1} (which is $\Omega_t = \varphi(\Omega) \bigcap \Omega$) is a proper subset of Ω and is not defined on $\Lambda_t = \Omega - \Omega_t$, see Figure 3.8 for an example. Therefore, the transformed source is defined only on Ω_t. This causes an issue in the calculation of the distance function, because one image (the source) is defined only on a proper subset of Ω and the other image (the target) is defined on Ω. There are two ways to overcome this and calculate the distance function [32]:

1. The distance function can be calculated on Ω, by defining the missing values of the transformed image by $I \circ \varphi^{-1}(\Lambda_t, \boldsymbol{t}) = 0$.
2. The distance function can be calculated strictly on $\Omega_t = \varphi(\Omega) \bigcap \Omega = \Omega - \Lambda_t$.

As a comparison, we employ both methods to calculate the distance function in a registration of the cups. The initial value for optimisation is set as the identity transformation $\boldsymbol{t} = (1, 0, 0, 0)$. Note that from here on, when we say the value of distance function, we mean the square of the L^2 norm, $\|I \circ \varphi^{-1} - J\|_2^2$. The value of the distance function before registration is 3527.93.

First method: Transformed source is zero on Λ_t. The optimizer stops at $\mathbf{t} = (1.3248,$ $-0.4965, -0.0337, -0.0273)$, and the images are not well registered. Figure 3.9b

(a) (b) (c)

Figure 3.9: (a) Discrepancy between the source and the target before registration, (b) discrepancy between the transformed source and the target after registration using the first method (transformed source is zero at points where the transformation is not defined), (c) discrepancy between the transformed source and the target after registration using second method (both the transformed source and the target are zero at the points where the transformation is not defined).

shows the discrepancy[1] between the transformed source and the target. The value of the distance function after registration is 1233.4.

Second method: Calculating the distance function strictly on Ω_t is equivalent to considering the source and the target to be zero on Λ_t. The images are not well registered, and the optimizer stops at exactly the same point: $\mathbf{t} = (1.3248, -0.4965, -0.0337, -0.0273)$. Figure 3.9c shows the discrepancies. The value of the distance function after registration is 1233.4.

So far we used two existing methods in the registration of the cups. These methods have some limitations and problems.

The limitation of the first method is that the images need to be zero toward their edges or on their background. For example, in Figure 3.10 I is taken as source (left) and target J is generated by scaling the source by $\varphi(z) = \frac{1}{2}z$ (right). The distance function is calculated for scaling from 0.2 to 1, see Figure 3.11. We expect the distance function to be minimum when the scaling is $\frac{1}{2}$, but it is not. The reason is that the background of the images should also match. Not only is the distance function not minimum at $\frac{1}{2}$, but it is not smooth either.

[1]When two images are subtracted, some intensities are out of the range: $[0, 1]$. For example, if image I at some point is zero and J is one then $I - J = -1$ at that point, and in the computer all intensities that are less than zero are shown as black and larger than one as white. Therefore, for a better representation of the discrepancy image, either the absolute value of the difference $|I - J|$, or $0.5(I - J + 1)$ can be used. In this research, we used the latter to show the discrepancy image. Hence, a perfect match appears as mid-grey.

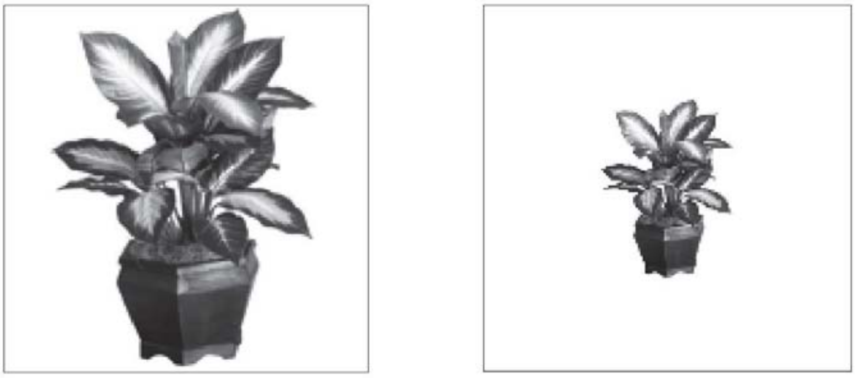


Figure 3.10: A different pair of images. Here, the target (right) is generated by applying a known transformation to the source (left). This is an easier problem.

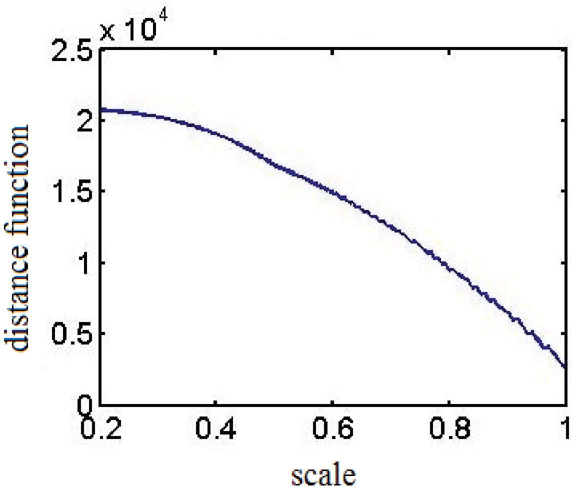


Figure 3.11: Value of the distance function between the source and target given in Figure 3.10, versus scale, where the points without pre-image have assigned zero intensity (i.e., the second method is employed to treat pixels in Λ_t).

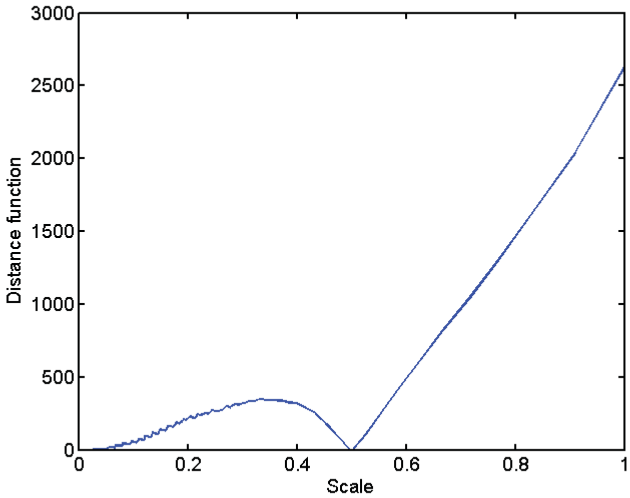


Figure 3.12: Distance function of source and target in Figure 3.10 versus scale, where the distance function is calculated on the overlap region of target and source, i.e., on $\Omega - \Lambda_t$.

The limitation of the second method is that for some $\boldsymbol{t}$, $\varphi(\Omega, \boldsymbol{t})$ does not have any overlap with Ω, i.e. $\varphi(\Omega) \bigcap \Omega = \varnothing$, so $\Lambda_t = \Omega$. In the second method the distance function is calculated on $\varphi(\Omega) \bigcap \Omega$. So, in this case the distance function is zero, which is an incorrect solution. Figure 3.12 shows the distance function of the images I and J given in Figure 3.10. It can be seen that when the scaling is very large the distance function is almost zero, and the optimizer may find these incorrect solutions. This limitation will be addressed in Section 3.4.

In the previous example the background was black, but the registration was not good. In fact, both methods produced exactly the same result for this example. Although the registration seems to be easy, the cups did not register well. In the following sections, we investigate what caused the registration to fail, and how to address the problems.

3.2 Difficulties with Image Registration

One of the most common and severe problems for image registration is the presence of local minima in the distance function, which causes the optimizer to get stuck at a local minimum and hence fail to find the desired global minimum. A method that is employed to solve this problem is the 'Multi-resolution' technique [49]. In this technique, registration starts with low-resolution images (containing only gross features) and continues progressively through higher resolutions using the previous solution as the initial guess. Low-resolution images contain fewer pixels than higher-resolution ones, so the distance function requires less computation when images have low-resolution. Also, because only gross features of the images remain in the low-resolution image, there should be fewer local minima in the distance function. Starting with low resolution images provides a good starting initial value for the registration of images at higher resolution. One of the key points of this technique is to get a smooth representation of images at low resolution, which yields a smooth objective function and help prevent the optimizer from getting stuck at a local optima.

In [32] they investigated image registration using the second method, and they claimed that the multi-resolution technique also sometimes fails to find the global minimum, and that the low-resolution images that are employed in this method are not sufficient to prevent the optimizer from getting stuck at a local minimum because the distance function still has some local minima and is discontinuous at some points. They explained that the presence of the discontinuity in the distance function is because images are discrete sets of pixels and the distance function is calculated on the overlap region and the number of pixels in the overlap changes during registration. They also mentioned that the discontinuity is not due to the interpolation methods, as all interpolation methods (except nearest neighbour, which is used rarely) are continuous. An apodization function is used to overcome this lack of continuity; this is a function which brings the edge of a function smoothly to zero. The apodization function de-weights the contribution of locations that are near to the edge of the overlapping area. The weighting was chosen so that the input of such locations drops continuously until it reaches zero at the edge of the overlapping region.

Based on the literature, the causes of the registration failing are the presence of local minima and discontinuity points in the distance function. Therefore, we study the distance function carefully and give a precise mathematical understanding of issues of the distance function in registration in the following sections.

3.3 Critical Points in the Distance Function

There are three types of points in the distance function that are an obstacle for the optimizer to find the global minimum. They are discussed in the following sections.

3.3.1 Discontinuity Points

The first type of point is one where the distance function is not continuous. We show that the discontinuity in the distance function is the result of discontinuity in the transformed source.

Theorem 3.3.1. *If two functions f and g are continuous, then their product fg, sum $f+g$ and composition $f \circ g$ are continuous.*

Proof. See [57]. $\square$

Theorem 3.3.2. *Let $u(x)$ be a positive function. Then we have the following statements:*

- *If $u(x)$ is continuous at x_0 then $\sqrt{u}$ is continuous at x_0. Because:*

$$\lim_{x \to x_0} \sqrt{u(x)} = \sqrt{\lim_{x \to x_0} u(x)} = \sqrt{u(x_0)}.$$

- *If $u(x)$ is differentiable at x_0 then $\sqrt{u(x)}$ is differentiable at x_0. The derivative of $\sqrt{u(x)}$ is $\frac{u'(x)}{2\sqrt{u(x)}}$. We show that it is continuous at x_0:*

$$\lim_{x \to x_0} \frac{u'(x)}{2\sqrt{u(x)}} = \frac{\lim_{x \to x_0} u'(x)}{\lim_{x \to x_0} 2\sqrt{u(x)}} = \frac{u'(x_0)}{2\sqrt{u(x_0)}}.$$

  Therefore, $\sqrt{u(x)}$ is differentiable at x_0.

Theorem 3.3.3. *Suppose $\varphi : \mathbb{R}^2 \to \mathbb{R}^2$ is a smooth function on $\mathbb{R}^2$. If distance function $E(t) = \sqrt{\sum (I \circ \varphi^{-1}(t) - J)^2}$ is discontinuous at t_0 then there exists $(x_0, y_0) \in \mathbb{R}^2$ such that I is discontinuous at $(x_0, y_0) \in \mathbb{R}^2$.*

Proof. Proof by contradiction: Suppose I is continuous for every $(x, y) \in \mathbb{R}^2$. Since I is a continuous function we have:

$$\begin{aligned} \lim_{t \to t_0} E(t)^2 &= \sum \lim_{t \to t_0} (I \circ \varphi^{-1}(t) - J)^2 \\ &= \sum (I \circ (\lim_{t \to t_0} \varphi^{-1}(t)) - J)^2 \\ &= \sum (I \circ \varphi^{-1}(t_0) - J)^2 = E(t_0)^2. \end{aligned}$$

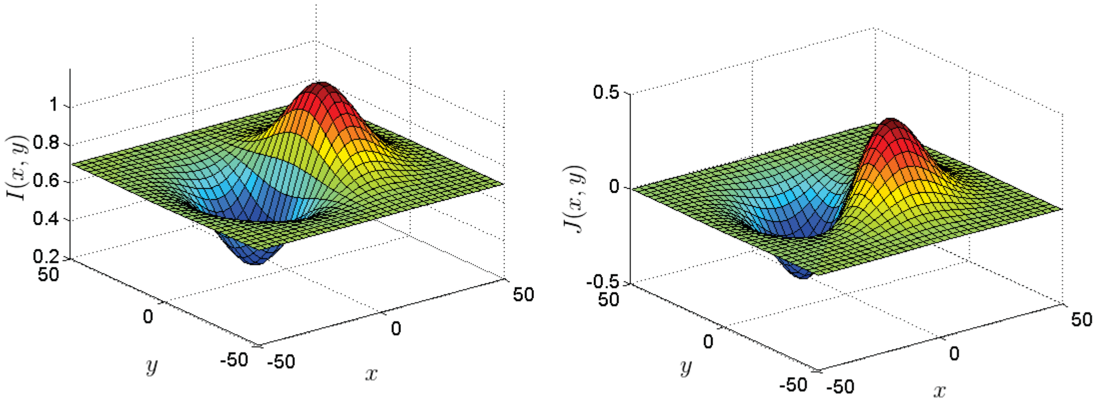


Figure 3.13: Two continuous image functions. Left: source. Right: target image.

Therefore, E^2 is continuous at t_0, and by Theorem 3.3.2 E is continuous at t_0, which is a contradiction to the assumption. □

Example 6. Two smooth images are given as source I and target J. The target is a rotation of function I by $\frac{\pi}{4}$.

$$
\begin{aligned}
I(x,y) =& \frac{x}{30} \exp\left(-(\frac{x}{30})^2 - (\frac{y}{30})^2)\right) \\
J(x,y) =& \frac{\cos(\frac{\pi}{4})x + \sin(\frac{\pi}{4})y}{30} \exp\left(-\left(\frac{\cos(\frac{\pi}{4})x + \sin(\frac{\pi}{4})y}{30}\right)^2 - \left(\frac{-\sin(\frac{\pi}{4})x + \cos(\frac{\pi}{4})y}{30}\right)^2\right)
\end{aligned}
$$

The functions are shown in Figure 3.13. They are smooth functions, but have different background values, tending to 0.7 in the source and to zero in the target. The distance function is calculated on Ω and plotted in Figures 3.14 and 3.15 versus rotation, where:

- $I(\mathbb{R}^2 - \Omega) = 0$ so $I \circ \varphi^{-1}(\Lambda_t) = 0$ (the first method given in Section 3.1.3).

  It can be seen in Figure 3.14 that the distance function E is discontinuous for some t_0, which is the result of some discontinuity of the source on $\mathbb{R}^2$ by Theorem 3.3.3.

- $I(\mathbb{R}^2 - \Omega) = 0.7$, so $I \circ \varphi^{-1}(\Lambda_t) = 0.7$. This makes the source into a continuous image on $\mathbb{R}^2$. Therefore, by Theorem 3.3.3 the distance function will be continuous, see Figure 3.15.

So, in this example, we observed that the value of Λ_t is equal to the value of the source

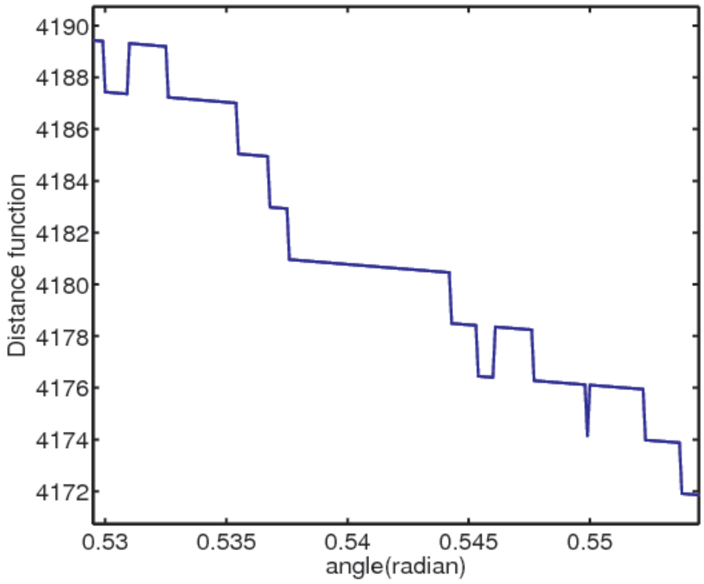


Figure 3.14: The distance function of the source and the target given in Figure 3.13 versus angle of rotation (the applied transformation is rotation about the origin), where the source is zero on its extended domain: $I(\mathbb{R}^2 - \Omega) = 0$.

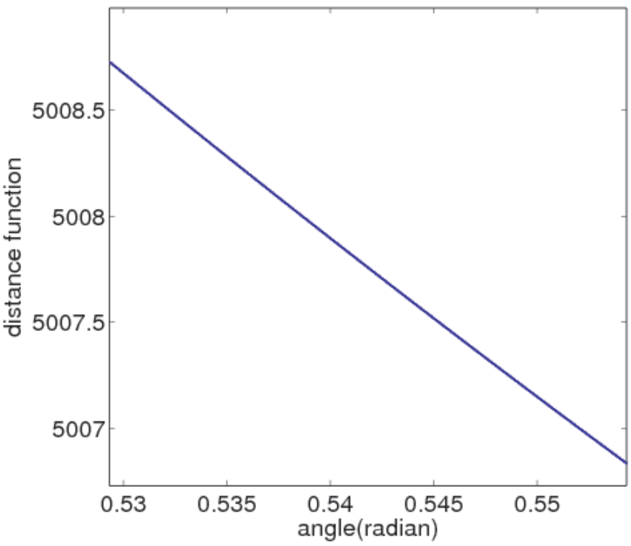


Figure 3.15: The distance function of the source and target given in Figure 3.13 versus angle of rotation (the applied transformation is rotation about the origin), where the source is 0.7 on its extended domain: $I(\mathbb{R}^2 - \Omega) = 0.7$, and so it is continuous.

on its extended domain, because the Λ_t are the points that are mapped from $\mathbb{R}^2 - \Omega$ into Ω. We extend the domain of the source so that the source is continuous, because if we have a continuous source then the distance function is continuous by Theorem 3.3.3.

However, note that just because I is discontinuous for some (x_0, y_0) it cannot be concluded that E is discontinuous, i.e., discontinuous images might have a continuous distance function, because the sum of discontinuous functions can be continuous. Theorem 3.3.3 only says that to have a continuous distance function, it is sufficient to have a continuous source.

In this example, the image was continuous everywhere except on the edge of Ω (when $I(\mathbb{R}^2 - \Omega) = 0$), and we easily removed the discontinuity in the source by putting $I(\mathbb{R}^2 - \Omega) = 0.7$. But some images may be discontinuous on their domain also. In Section 3.3.3 it will be explained how to make continuous approximations to them.

3.3.2 Non-Differentiable Critical Points

The second type of potentially problematic points that occur in the distance function are those at which the distance function is not differentiable. Here we show that non-differentiable points in the distance function are the result of non-differentiability of the source.

Theorem 3.3.4. *If function $f \neq 0$ is discontinuous and $g \neq 0$ is continuous then their product fg is also discontinuous.*

Proof. Suppose fg is continuous, then because g is continuous, by Theorem 3.3.1 $f = \frac{fg}{g}$ is continuous, which is a contradiction to the assumption. $\square$

Theorem 3.3.5. *Suppose $\varphi : \mathbb{R}^2 \to \mathbb{R}^2$ is a smooth function on $\mathbb{R}^2$. If $E(t) = \sqrt{\sum (I \circ \varphi^{-1}(t) - J)^2}$ is not differentiable at t_0, then there exists $(x_0, y_0) \in \mathbb{R}^2$ such that I is not differentiable at (x_0, y_0) .*

Proof. The derivative of E^2 with respect to t is:

$$\frac{\partial E^2}{\partial t} = 2 \sum_{\mathbb{R}^2} (I \circ \varphi^{-1}(t) - J) \langle \partial_t \varphi^{-1}(t) \cdot \nabla I(\varphi^{-1}(t)) \rangle. \tag{3.3}$$

Proof by contradiction: Suppose I is differentiable for every $(x, y) \in \mathbb{R}^2$. Then I and ∇I are continuous functions, so:

$$\begin{aligned}
\lim_{t \to t_0} \frac{\partial E^2}{\partial t} &= \\
&= \sum \lim_{t \to t_0} (I \circ \varphi^{-1}(t) - J) \langle \partial_t \varphi^{-1}(t) \cdot \nabla I(\varphi^{-1}(t)) \rangle \\
&= \sum (I \circ \lim_{t \to t_0} \varphi^{-1}(t) - J) \langle \lim_{t \to t_0} \partial_t \varphi^{-1}(t) \cdot \nabla I(\lim_{t \to t_0} \varphi^{-1}(t)) \rangle \\
&= \sum (I \circ \varphi^{-1}(t_0) - J) \langle \partial_t \varphi^{-1}(t_0) \cdot \nabla I(\varphi^{-1}(t_0)) \rangle.
\end{aligned}$$

Therefore, E^2 is differentiable at t_0, and so by Theorem 3.3.2, E is differentiable at t_0, which is a contradiction to the assumption. □

Therefore, to have a differentiable function it is enough to have a differentiable source. In the following section we explain a method of making a differentiable image.

3.3.3 Construction of a Differentiable Image

First, it is illustrated by an example how to make a differentiable function from a non-differentiable function, where the domain of the function is one dimensional.

Example 7. Let the function f be given as:

$$f(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}.$$

Here f is discontinuous at zero, as shown in Figure 3.16. To create a continuous and differentiable function, f is convolved with a Gaussian function. The one dimensional Gaussian function is:

$$g_\sigma(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-h)^2}{\sigma^2}\right) \tag{3.4}$$

where h is the mean and σ is the standard deviation. The convolution function is:

$$(f * g_\sigma)(x) = \int_{\mathbb{R}} f(h) \times \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-h)^2}{\sigma^2}\right) dx. \tag{3.5}$$

$f * g_\sigma$ is shown in Figure 3.17 for $\sigma = 1$. As can be seen in Figure 3.17, $f * g_\sigma$ and f

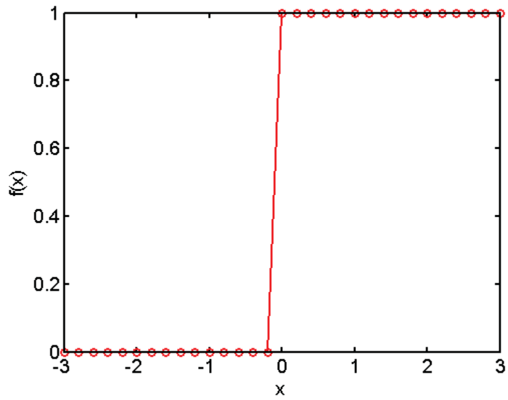


Figure 3.16: The function f is discontinuous at zero.

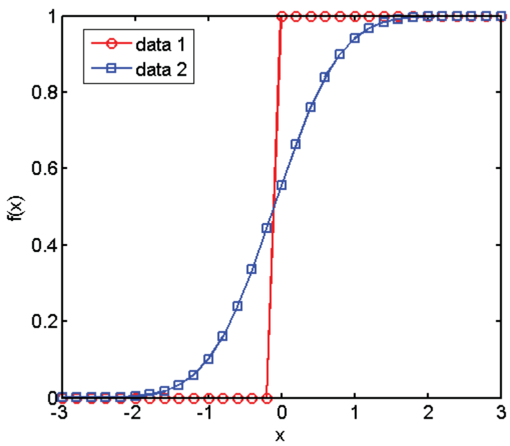


Figure 3.17: Convolution. Here data 1 is f and data 2 is the convolution of f with the Gaussian given in Equation (3.5) with $\sigma = 1$ and $h = 0$.

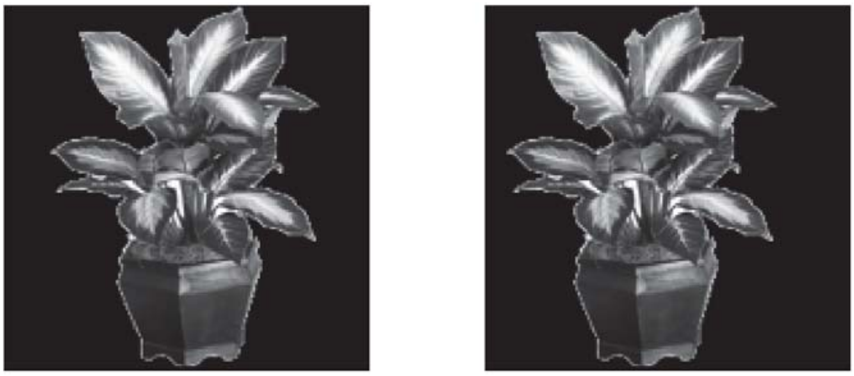


Figure 3.18: Left: source. Right: target. There is a translation between source and target.

are equal everywhere except in a neighbourhood of zero. The convolved function $f * g_\sigma$ is differentiable at zero.

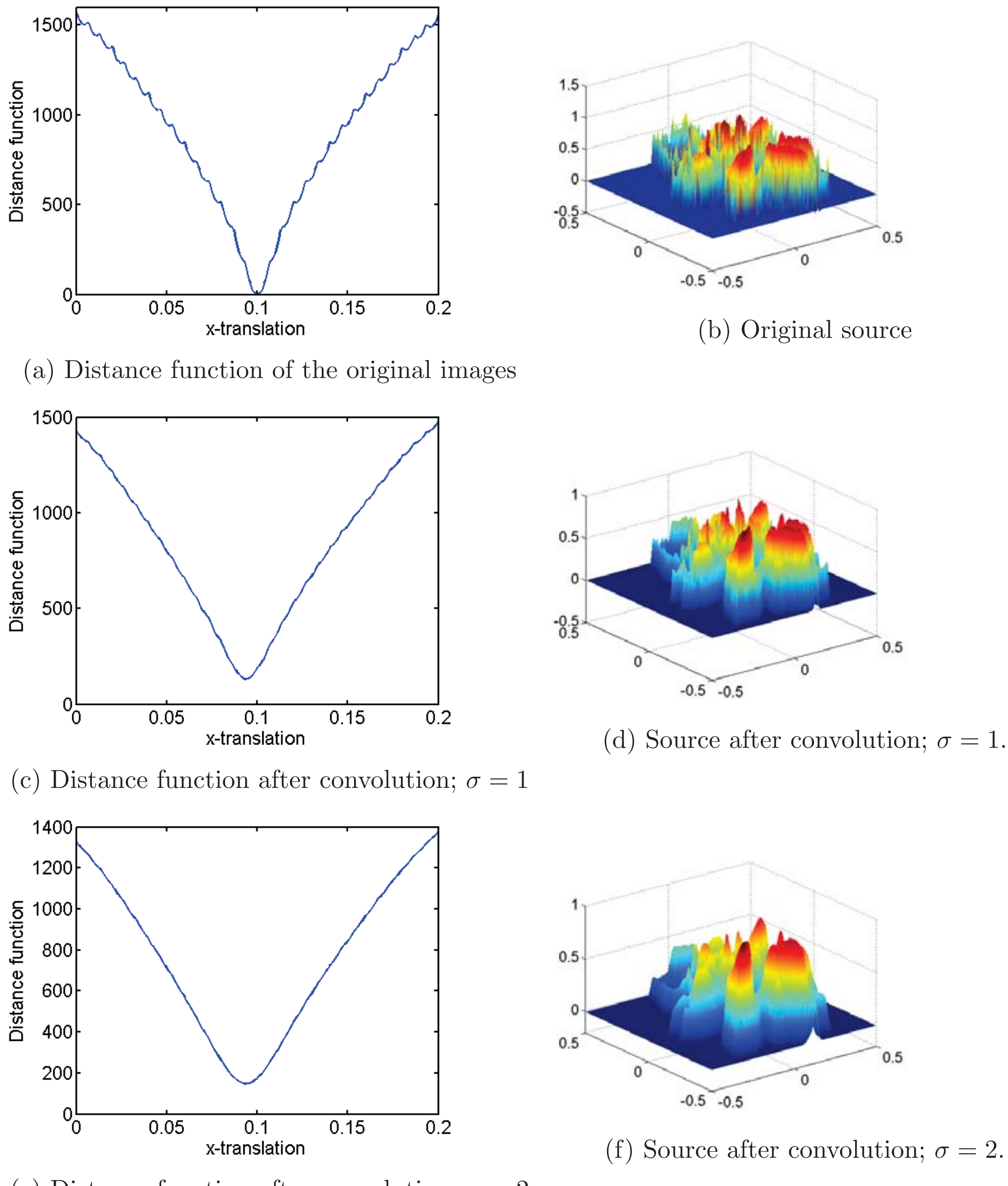
The same method can be applied to the image functions, but a two dimensional spherical Gaussian function is used for the convolution:

$$(I * g_\sigma)(h_1, h_2) = \int_{\mathbb{R}^2} I(h_1, h_2) \times \frac{1}{\sqrt{2\pi\sigma^2}} \exp -\frac{(x-h_1)^2 + (y-h_2)^2}{\sigma^2} dxdy. \qquad (3.6)$$

Example 8. Two images as source and target are given in Figure 3.18. The size of the images is 150×150 pixels, and the domain of the images is $\Omega = [-0.5, 0.5] \times [-0.5, -0.5]$. The target is an x-translation of I by 0.1. The value of the target on $\Lambda_{0.1} = 0$. The distance function is calculated where the group of transformation is x-translation and $I \circ \varphi^{-1}(\Lambda_t, \boldsymbol{t}) = 0$. The distance function versus translation is shown in Figure 3.19a. It can be seen in Figure 3.19a that the distance function is not differentiable at many points.

To have a differentiable distance function it is enough to have a differentiable source for every $(x, y) \in \mathbb{R}^2$ (by Theorem 3.3.3). To see the effect of different widths, we convolved the source with two different Gaussians with $\sigma = 1$ and 2 pixels. The convolved source and the distance function after source convolution are shown in Figure 3.19 (c-f).

The value of σ is chosen such that the source after convolution is differentiable at every point. As can be seen in Figure 3.19, when $\sigma = 2$ the distance function and convolved source are smooth.



(a) Distance function of the original images

(b) Original source

(c) Distance function after convolution; $\sigma = 1$

(d) Source after convolution; $\sigma = 1$.

(e) Distance function after convolution; $\sigma = 2$

(f) Source after convolution; $\sigma = 2$.

Figure 3.19: Left column: Distance function of the source and the target given in Figure 3.18, where the group of transformations is x-translation and the source is convolved with Gaussians with $\sigma = 1, 2$. The right column shows the functional view of the images.

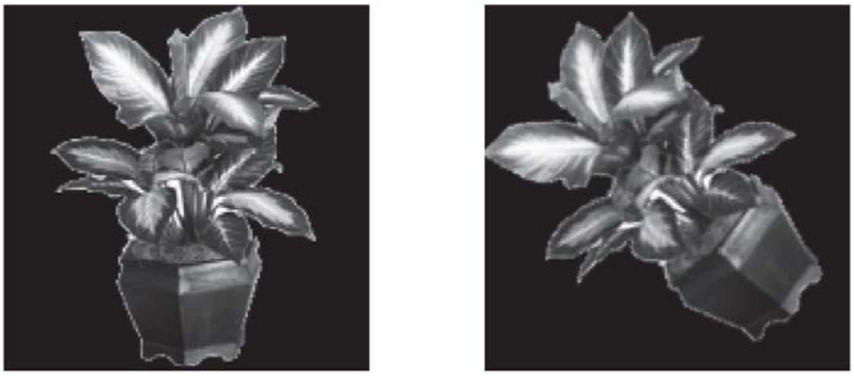


Figure 3.20: Left: source, Right: target. The target is generated from source by a known rotation.

3.3.4 Differentiable Critical Points

Sections 3.3.1 and 3.3.2 explained why discontinuity and non-differential critical points occur in the distance function, and how we remove them is explained in Section 3.3.3. Although after convolution the distance function is differentiable, there are still some local minima in the distance function that the optimizer may get stuck at.

Example 9. Two images, the source and the target of size 150×150 pixels, are given in Figure 3.20. The target is a rotation of the source by $\frac{\pi}{4}$. On $\Lambda_{\frac{\pi}{4}}$ in the target we put zero. The images are convolved with a Gaussian with $\sigma = 4$. The distance function versus angle (in radians) is shown in Figure 3.21, where $I(\mathbb{R}^2 - \Omega) = 0$. As can be seen, there are still some local minima in the distance function. The discrepancy between the transformed source and the target: $\frac{1}{2}\left(I \circ \varphi^{-1}(\Omega, \boldsymbol{t}_i) - J + 1\right)$ corresponding to each $\boldsymbol{t}_i$, $i = 1, 2, 3, \ldots, 6$, is shown in Figure 3.22. It can be seen in Figure 3.22 that at:

- t_1 the overlap is a maximum and the intensities match perfectly.
- t_4 the overlap is a minimum.
- t_6 the overlap is a maximum, but the intensities do not match.

It is reasonable that these three points are maxima or minima, but points t_2, t_3 and t_5 are also local optima. The reason for the presence of t_2, t_3 and t_5 is because there is too much detail in the images.

As was mentioned in Section 3.2, a method that has been used in image registration to keep only gross features is to construct a low resolution version of the images. We make

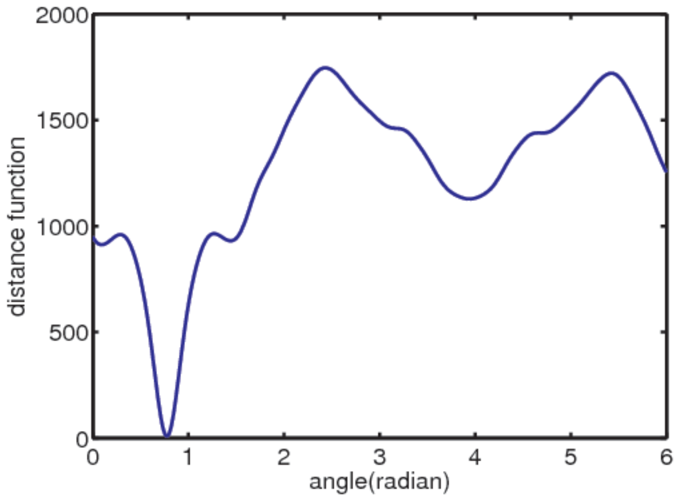


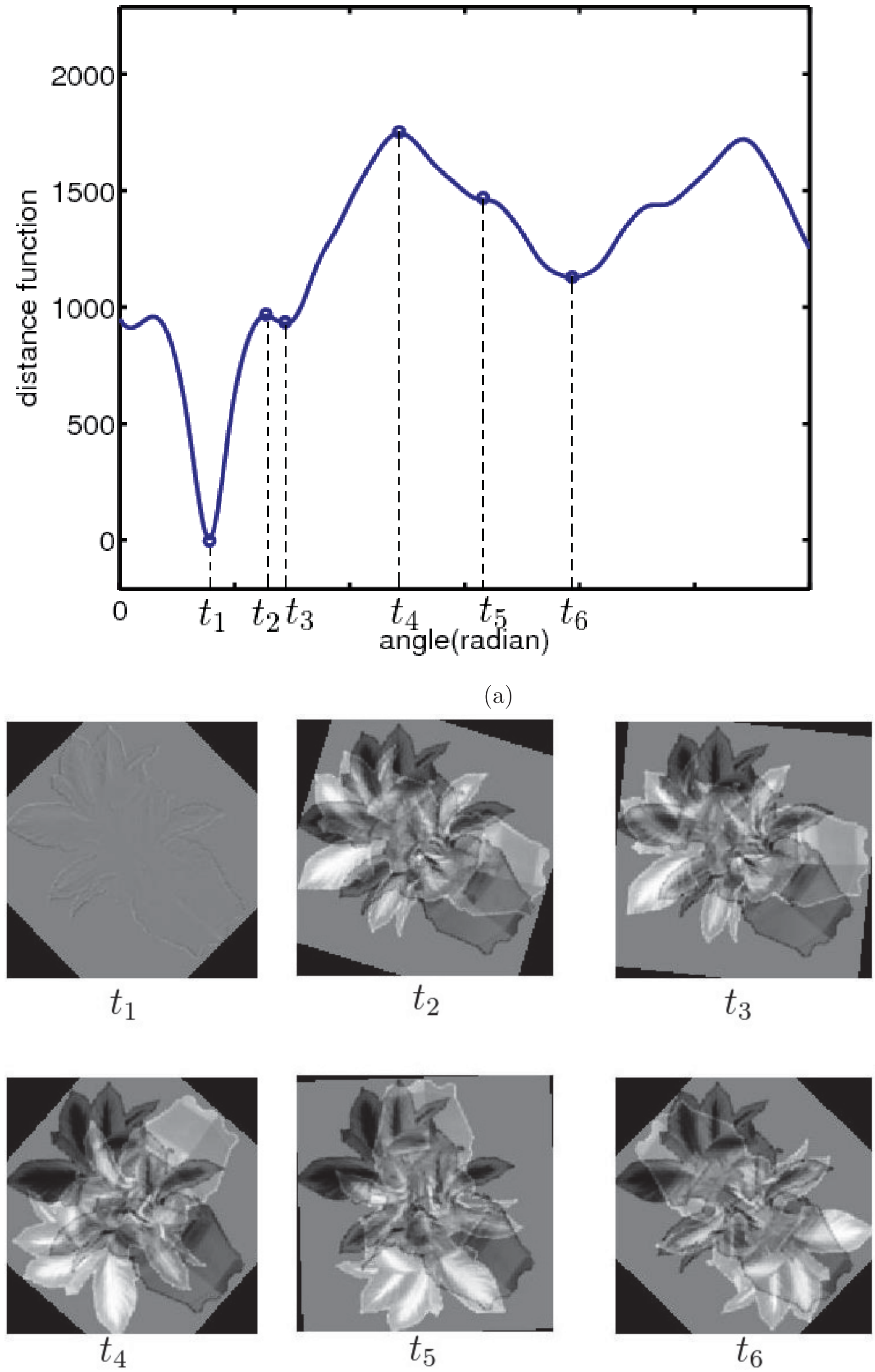
Figure 3.21: The distance function of source and target given in Figure 3.20, where the images are convolved with a Gaussian with $\sigma = 4$, and also the value of the source on the extended domain is zero: $I(\mathbb{R}^2 - \Omega) = 0$.

a low resolution of the source and the target from Figure 3.20, where each pixel in the low resolution image is the mean average of intensities on 10×10 blocks of pixels. In this way, only gross features remain and so the unreasonable local minima or maxima, like t_2, t_3 and t_5, should be eliminated. Figure 3.23 shows the low-resolution versions of the source and target. The size of the source and target in the low-resolution version is 15×15. Their distance function is shown in Figure 3.24.

It can be seen in Figure 3.24 that local minima t_2, t_3 and t_5 are eliminated, but many non-differentiable and discontinuous points occur in the distance function, which is the result of the discontinuous and non-differentiable source. Therefore, the source needs to be smoothed. We find that when the original images are convolved with Gaussian kernels with $\sigma = 10$, all discontinuities, non-differentiable critical and unreasonable differentiable critical points are removed, see Figures 3.25 and 3.26. Only one spurious local minimum remains.

3.4 Treating the Pixels Without Pre-image

In the previous sections we identified the reasons for critical points in the distance function. Using that knowledge we now try to improve the methods of treating pixels without pre-image, i.e. pixels in Λ_t.



(b)

Figure 3.22: (a) The distance function for the source and the target in Figure 3.20 versus angle (in radians). The group of transformations is rotations. Six local optima t_i, $i = 1, 2, \ldots, 6$ in the distance function are highlighted, and (b) shows the discrepancy between the transformed source and target for t_i, $i = 1, 2, \ldots, 6$.

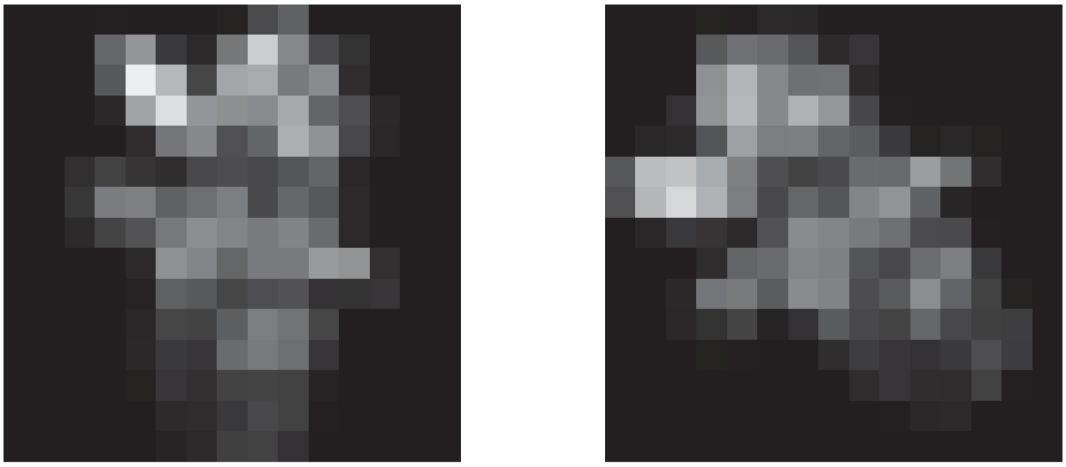


Figure 3.23: Low resolution versions of the source and the target in Figure 3.20. Each pixel is the mean average of intensities on 10×10 blocks of pixels, so the size of the images is 15×15 pixels.

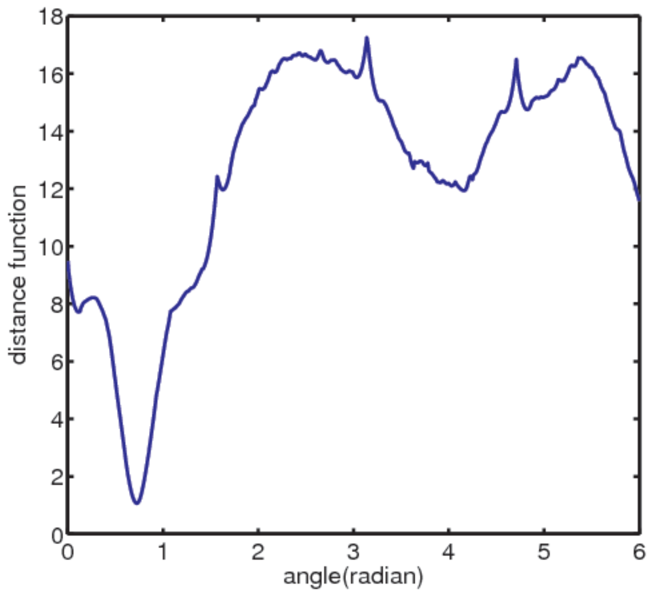


Figure 3.24: The distance function of the low-resolution versions of the source and the target given in Figure 3.23 versus angle (in radians), where the group of transformations is rotations.

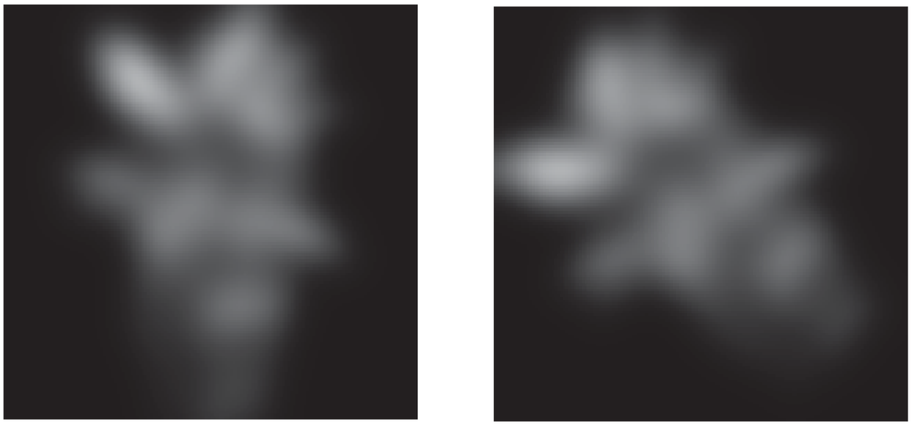


Figure 3.25: The source and the target in Figure 3.20 are convolved with a Gaussian with $\sigma = 10$.

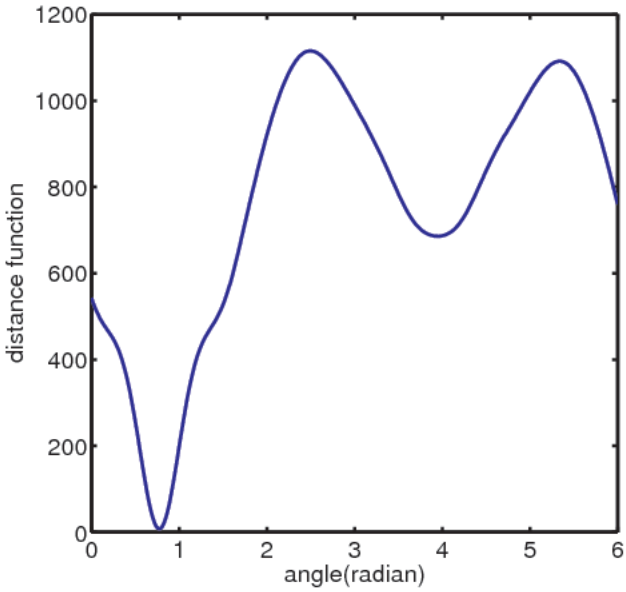


Figure 3.26: The distance function of the source and the target in Figure 3.25 versus rotation. The group of transformations is rotations.

- First method: This method can be applied only when images have constant backgrounds. Suppose that an image I is continuous and is equal to value a on its background. So, to have a continuous distance function we have to assign a value equal to a to the points in Λ_t: $I \circ \varphi^{-1}(\Lambda_t) = a$. If we extend the domain of I to $\mathbb{R}^2$, then
$$I(x,y) = \begin{cases} I(x,y) & (x,y) \in \Omega \\ b & b \in \mathbb{R}, (x,y) \in \mathbb{R}^2 - \Omega \end{cases},$$
where b is a constant. Then by Theorem 3.3.3, b should be equal to a, and so $I \circ \varphi^{-1}(\Lambda_t) = a$.

  Therefore, the first method should only be used when images have constant backgrounds, and $I \circ \varphi^{-1}(\Lambda_t) = a$, where a is the value of I on its background.

- Second method: Theorem 3.3.3 tells us why the first method causes problems in registration, but it does not explain why the second method also fails. When using the second method, on the extended domain the target is also changing with t, $J_t(\Lambda_t) = 0$:
$$J_t(x,y) = \begin{cases} J(x,y) & (x,y) \in \Omega - \Lambda_t \\ 0 & (x,y) \in \Lambda_t \end{cases},$$
while in Theorem 3.3.3, the target is supposed to be constant. Similarly, Theorem 3.3.5 does not explain the failure of registration, as images are discontinuous and so non-differentiable on their extended domain. So the second method needs further study to identify the mathematical reasons for its failure. As we will only use the first method in the rest of this thesis, this is left as future work.

Another issue that was mentioned about the second method was $\varphi(\Omega) \bigcap \Omega = \varnothing$. In this case, the distance function is zero and so a minimum. To overcome this problem, we multiply the cost function by an coefficient:
$$\left(\frac{S}{S-N}\right) \Sigma_{\Omega-\Lambda_t} (I \circ \varphi^{-1}(\Omega, \boldsymbol{t}) - J)^2, \tag{3.7}$$
where S is the total number of pixels of Ω and N is the number of pixels of Λ_t. So, the coefficient controls the overlaps. For example, when there is not any overlap, it is ∞.

The distance function of the images given in Figure 3.10 is calculated again using Equation (3.7), see Figure 3.27.

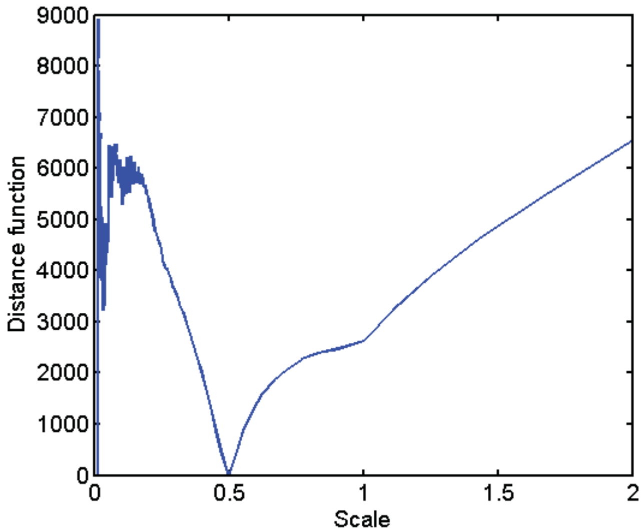


Figure 3.27: Distance function versus scale. Distance function is computed by Equation (3.7).

As mentioned before, because images in this research have constant background, we use the first modified method to treat points in Λ_t. Our registration algorithm is given in the following section.

3.5 Modified Registration Methodology of This Research

The image registration code used in this thesis consists of a main program, algorithm 1, and an objective function, algorithm 2. The smoothing parameter σ is chosen experimentally, but typically varies between 1 and 10 in our experiments.

Algorithm 1 Image registration

1: Input images I the source and J the target.
2: Pick an initial value $\boldsymbol{t}_0$ for the optimizer.
3: $\boldsymbol{t}_1 = \mathsf{lsqnonlin}(\boldsymbol{t}_0, E(I * g_\sigma, J * g_\sigma, \boldsymbol{t}))$ $\triangleright$ E is defined in algorithm 2; g_σ is a Gaussian function with standard deviation σ as smoothing kernel.
4: $\boldsymbol{t}_2 = \mathsf{lsqnonlin}(\boldsymbol{t}_1, E(I, J, \boldsymbol{t}))$ $\triangleright$ E is defined in algorithm 2; initial value is t_1, which is the output of the registration of smooth images from the previous step.

Algorithm 2 Function $E(I_1, J_1, \boldsymbol{t})$

1: Transform the discrete domain Ω' by φ^{-1}.
2: Construct the discrete image $I_1 \circ \varphi^{-1}(\Omega', \boldsymbol{t})$ by bilinear interpolation.
3: Set $I_1 \circ \varphi^{-1}(\Lambda_{\boldsymbol{t}}, \boldsymbol{t}) = a$, where a is chosen such that source is continuous on $\mathbb{R}^2$.
4: Compute the error $E(I_1, J_1, \boldsymbol{t}) = \|I_1 \circ \varphi^{-1}(\Omega', \boldsymbol{t}) - J_1\|_2$ (which is squared by the optimizer).

3.6 Testing the Algorithm

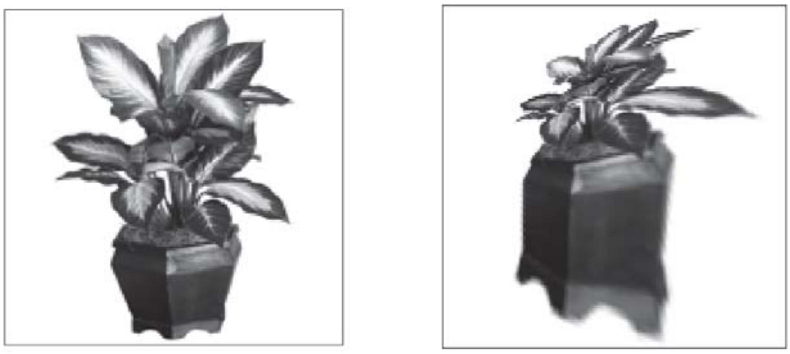
In this section we test the algorithm given in the previous section to register images. We give six examples of two classes of images: synthetic images, i.e., the target is generated by a known transformation of the source, and real images. The synthetic image examples are pictures of a plant and a brain. The real image examples are the cups given in Section 3.1.3, a side view of human skulls that is actually a line drawing, two galaxies and two bananas. They will be registered with different groups.

Example 10. An image of a plant of size 150×150 pixels is taken as source I, see Figure 3.28a. The target $J = I \circ \phi_1^{-1}$ is given in Figure 3.28b that is generated from the source by a $PSL(3, \mathbb{R})$ transformation as follows:

$$\phi_1(x, y) = \left(\frac{x + 0.1y + 0.3}{1.2x + 1.5y + 1.2008}, \frac{0.3x + 1.3y + 0.3}{1.2x + 1.5y + 1.2008} \right).$$

Since the source has a white background, and we want the source and the target to have similar intensities, we put $J(\Lambda_T) = 1$, where $T = [1, 0.1, 0.3, 1.2008, 0.3, 1.3, 1.2, 1.5]$ is the vector of parameters of ϕ_1. The domain of the images is $[-0.5, 0.5] \times [-0.5, 0.5]$. A transformation $\phi(x, y) = (\frac{t_1x+t_2y+t_3}{t_7x+t_8y+t_9}, \frac{t_4x+t_5y+t_6}{t_7x+t_8y+t_9})$ in $PSL(3, \mathbb{R})$ has eight parameters: $\boldsymbol{t} = [t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8]$. We perform a registration, where $\sigma = 3$, and the optimizer is initialized with the identity transformation $\boldsymbol{t}_0 = [1, 0, 0, 0, 1, 0, 0, 0]$. The images are registered perfectly; Figure 3.29 shows the images before and after registration.

Tables 3.1 and 3.2 give the outputs of the registration. As can be seen in the tables, the majority of registration has been fulfilled by the first step of the registration, i.e., with the smooth images.



(a) Source (b) Target

Figure 3.28: (a) An image of a plant that is taken as the source. (b) Transformation of source with a $PSL(3, \mathbb{R})$ transformation (details in the text); this image is taken as the target.

(a) Before registration (b) After registration

Figure 3.29: Registration of plants given in Figure 3.28 in $PSL(3, \mathbb{R})$, where the images are convolved with a Gaussian with $\sigma = 3$ in the algorithm given in Section 3.5

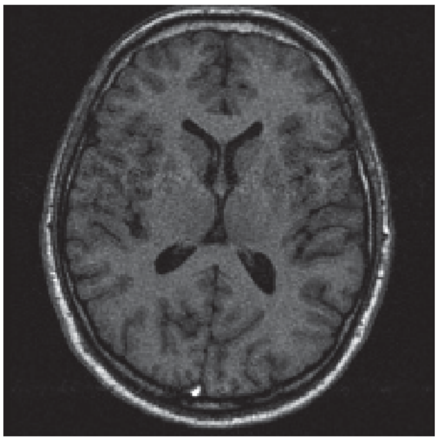
.

	Distance function before registration	Distance function after registration
Smooth images	1252.68	143.35
Original images	35.1179	0

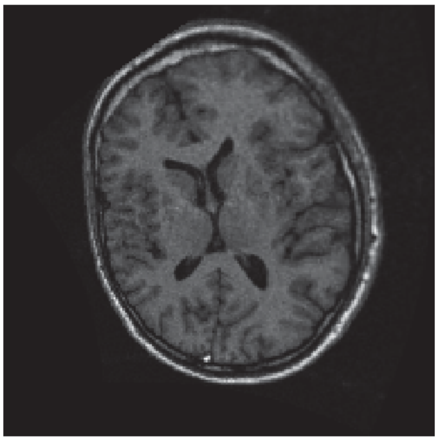
Table 3.1: The output of the registration of the plants given in Figure 3.28 in $PSL(3, \mathbb{R})$, where the images are convolved with Gaussian with $\sigma = 3$ in the algorithm given in Section 3.5.

	Output transformations
Smooth images	$(0.992, 0.0895, 0.099, 0.2908, 1.2883, 0.3019, 1.1917, 1.4902)$
Original images	$(1, 0.1, 0.1, 0.3, 1.3, 0.3, 1.2, 1.5)$

Table 3.2: The output transformation from the registration of the plants given in Figure 3.28 in $PSL(3, \mathbb{R})$, where images are convolved with Gaussian with $\sigma = 3$ in the algorithm given in Section 3.5.



(a) Source



(b) Target

Figure 3.30: (a) An image of a brain which is taken as the source. (b) Transformation of source with a $PSL(2,\mathbb{C})$ transformation (details in the text), this image is taken as target.

In this example the dimension of the group was eight. Registration in higher dimensional groups is usually more difficult.

Example 11. An image of a brain is taken as source, see Figure 3.30a. The size of the source is 192×192 pixels and its domain is $[-0.5, 0.5] \times [-0.5, 0.5]$. A transformation in $PSL(2,\mathbb{C})$ has the form $\phi(z) = \frac{az+b}{cz+d}$, $ad - bc = 1$, so it has three complex parameters: $a = t_1 + it_2, b = t_3 + it_4, c = t_5 + it_6$, and it depends on six real parameters. The target is given in Figure 3.30b and was generated from the source by a $PSL(2,\mathbb{C})$ transformation as follows:

$$\phi_2(z) = \frac{(1.1 + 0.1i)z + 0.01 - 0.03i}{(0.4 + 0.3i)z + 0.9126 - 0.0911i}.$$

The source is zero almost everywhere on its background, as mentioned we want the source and the target to have similar intensities. Therefore, we assign zero to the pixels in Λ_T in the target, where $T = (1.1, 0.1, 0.01, -0.03, 0.4, 0.3)$ are the parameters of ϕ_2.

Now we register the source and the target with the algorithm, where $\sigma = 2$ and $I \circ \phi^{-1}(\Lambda_t) = 0$, $\phi \in PSL(2,\mathbb{C})$. We initialize the optimisation with the identity transformation $\boldsymbol{t}_0 = (1, 0, 0, 0, 0, 0)$. The source and the target are registered perfectly. The outputs of the registration are given in Tables 3.3 and 3.4. As can be seen in the tables, again the majority of the registration has been done by the first step of the registration.

	Output transformation
Smooth images	$(1.0995, 0.0963, 0.01, -0.028, 0.3994, 0.2996)$
Original images	$(1.1, 0.1, 0.01, -0.03, 0.4, 0.3)$

Table 3.3: The output of transformations from the registration of the brains given in Figure 3.30 in $PSL(2,\mathbb{C})$. The images are convolved with a Gaussian with $\sigma = 2$.

	Distance function before registration	Distance function after registration
Smooth images	697.9	9
Original images	23.6	0

Table 3.4: Output of the registration of the brains given in Figure 3.30 in $PSL(2,\mathbb{C})$. The images are convolved with a Gaussian with $\sigma = 2$.

The next four examples are registration of real images.

Example 12. We register the cups given in Section 3.1.3 in the similarity group. The size of the source and the target is 200×200 and they are convolved with a Gaussian with $\sigma = 2$. A similarity transformation $z \mapsto az + b$ depends on four real parameters, $[t_1, t_2, t_3, t_4]$, where $a = t_1 + it_2$ and $b = t_3 + it_4$. We initialize the optimisation with the identity transformation $\boldsymbol{t}_0 = (1, 0, 0, 0)$, and $I \circ \varphi^{-1}(\Lambda_t) = 0$, φ belongs to the group. The output of the registration is given in Table 3.5. Figure 3.31 shows the discrepancy between the transformed source and the target after the registration.

In the registration in Section 3.1.3, the cups were not registered well, but using our algorithm they are registered very well. The reason that they did not register well the first time was because of some local minima in the distance function, which have been removed by smoothing the images.

Example 13. Two human skulls representing people of different ages are taken as source and target; the images are taken from [37]. Those are shown in Figure 3.32. The size of the images is 164×123. We register them in the $PSL(2,\mathbb{C})$ group. The

	Output transformation	Distance function before registration	Distance function after registration
Smooth images	$(1.1591, -0.8324,$ $-0.0408, -0.0639)$	2857.96	382.7
Original images	$(1.1498, -0.8253,$ $-0.0455, -0.0652)$	761.5	739.9215

Table 3.5: Output of registration of the cups given in Figure 3.7 in the similarity group. The images are convolved with a Gaussian with $\sigma = 2$.

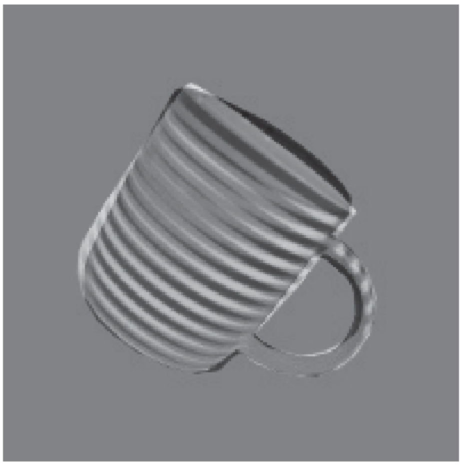
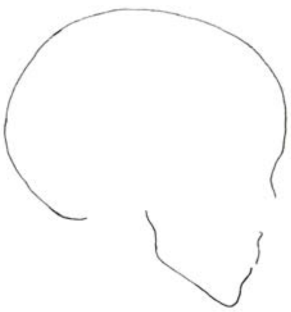


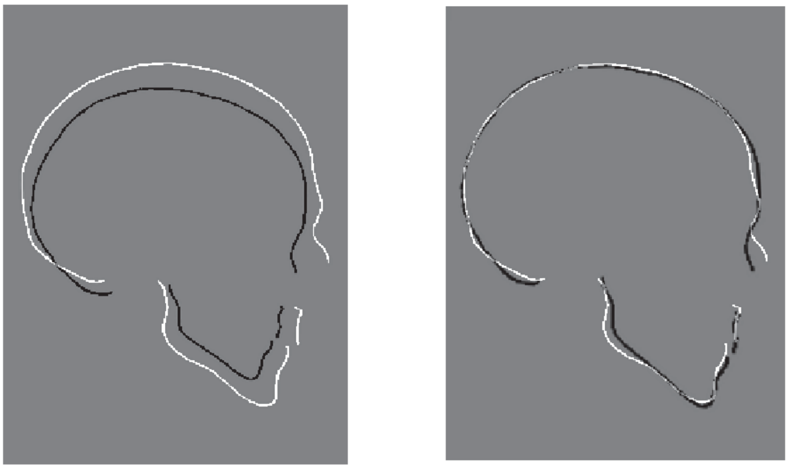
Figure 3.31: Discrepancy image of two cups given in Figure 3.7 after registration.



(a) Source

(b) Target

Figure 3.32: Two side views of human skulls at different ages, taken from [37].



(a) Before registration (b) After registration

Figure 3.33: Registration of the human skulls given in Figure 3.32 in $PSL(2,\mathbb{C})$, where the images are convolved with a Gaussian with $\sigma = 4$ in the algorithm given in Section 3.5.

	Distance function before registration	Distance function after registration
Smooth images	2.7	0.85
Original images	35.4	18.3

Table 3.6: The output of the registration of the human skulls given in Figure 3.32 in $PSL(2,\mathbb{C})$, where the images are convolved with a Gaussian with $\sigma = 4$ in the algorithm given in Section 3.5.

optimizer was initialized with $(1, 0, 0, 0, 0, 0)$, and $\sigma = 4$. Since the background of images is white, $I \circ \varphi^{-1}(\Lambda_t) = 1$. The source and target before and after registration are shown in Figure 3.33. The outputs of the registration also is given in Tables 3.6 and 3.7. As can be seen, the images are line drawings, but the algorithm still registered them well. This is a harder problem as there is less overlap between the original images.

Example 14. We register two photos of different galaxies. The source and the target are shown in Figure 3.34. The size of the images is 310×310 pixels and their domain is

	Output transformations
Smooth images	$(0.939, -0.016, 0.0093, 0.0186, 0.074, -0.1677)$
Original images	$(0.9393, -0.0207, 0.0078, 0.0205, 0.0905, -0.1722)$

Table 3.7: The output transformation from the registration of the human skulls given in Figure 3.32 in $PSL(2,\mathbb{C})$, where the images are convolved with a Gaussian with $\sigma = 4$ in the algorithm given in Section 3.5.

(a) Source (b) Target

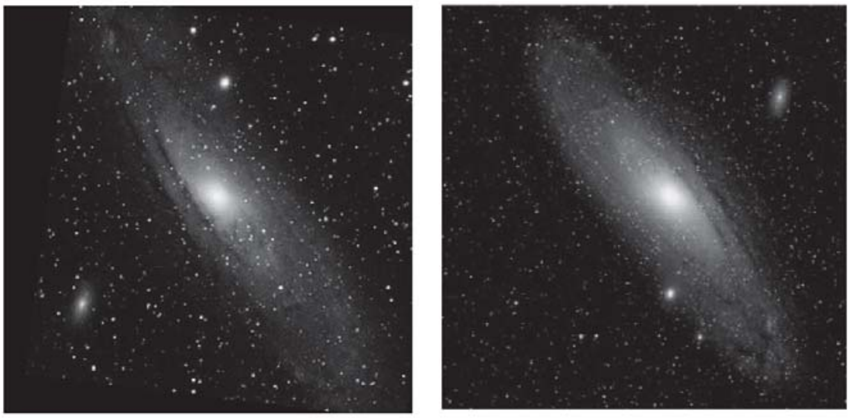
Figure 3.34: Two photos of galaxies.

	Output transformation	Distance function before registration	Distance function after registration
Smooth images	$(-0.1307, -1.0516, 0.0162, 0.0784)$	1802.6	290.7
Original images	$(-0.132, -1.0514, 0.0159, 0.0784)$	998.2	995.47

Table 3.8: Output of the registration of the galaxies given in Figure 3.34 in the similarity group. The images are convolved with a Gaussian with $\sigma = 6$.

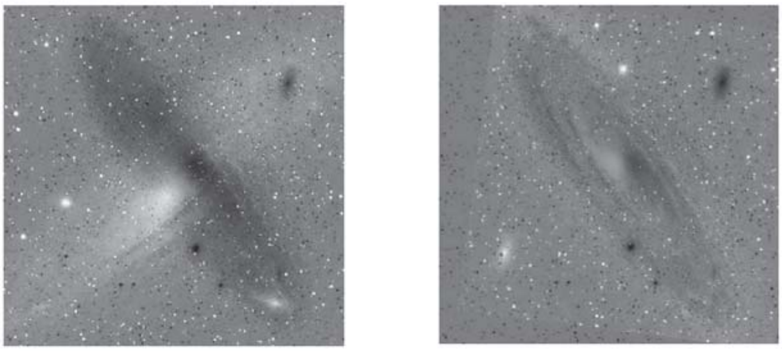
$[-0.5, 0.5] \times [-0.5, 0.5]$. We register the source and the target in the similarity group. We put $\sigma = 6$ and $I \circ \varphi^{-1}(\Lambda_t) = 0$. We initialize the optimisation with $\boldsymbol{t}_0 = (1, 0, 0, 0)$. The output of the registration is given in Table 3.8. Figure 3.35 shows the source after transformation and the target. Figure 3.36 shows the discrepancy between the source and target before and after registration.

Example 15. We register images of two different bananas in $PSL(2, \mathbb{C})$. The source and the target are shown in Figure 3.37. The size of the images is 216×216 pixels and their domain is $[-0.5, 0.5] \times [-0.5, 0.5]$. In the algorithm $\sigma = 2$ and $I \circ \varphi^{-1}(\Lambda_t) = 1$. We initialize optimisation with $\boldsymbol{t}_0 = (1, 0, 0, 0, 0, 0)$. The source and the target are well registered. The outputs of the registration are given in Tables 3.9 and 3.10. Figure 3.38 shows the source after transformation and Figure 3.39 shows the discrepancy between the source and the target before and after registration.



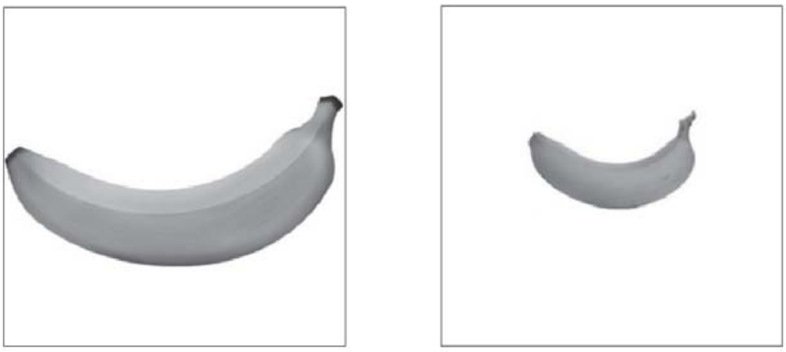
(a) Transformed source (b) Target

Figure 3.35: (a) Transformed source (the source is given in Figure 3.34) after registration with target in the similarity group. (b) The target.



(a) Before registration (b) After registration

Figure 3.36: Discrepancy images of two the galaxies given in Figure 3.34 (a): before, (b): after registration, where the images are convolved with a Gaussian with $\sigma = 6$.



(a) Source (b) Target

Figure 3.37: Two bananas.

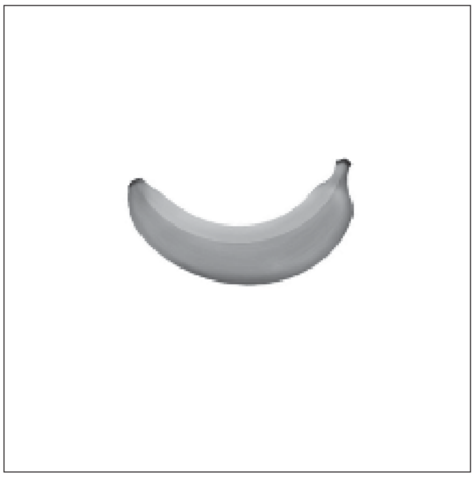


Figure 3.38: Transformed source (source and target are given in Figure 3.37) after registration with the target in $PSL(2, \mathbb{C})$.

	Distance function before registration	Distance function after registration
Smooth images	1865.5	29.2
Original images	53.6	44.1

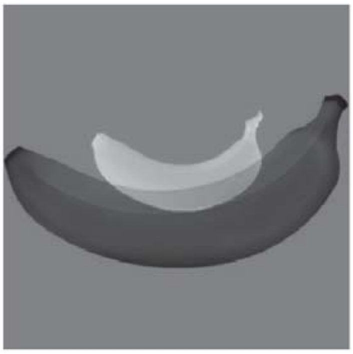
Table 3.9: Output of the registration of the bananas given in Figure 3.37 in $PSL(2, \mathbb{C})$. The images are convolved with a Gaussian with $\sigma = 2$.

	Output transformations
Smooth images	$(1.3859, -0.0496, 0.0025, 0.0719, 0.0012, -0.3324)$
Original images	$(1.39, -0.0519, 0.0018, 0.067, 0.0236, -0.3317)$

Table 3.10: Output of the registration of the bananas given in Figure 3.37 in $PSL(2, \mathbb{C})$. The images are convolved with a Gaussian with $\sigma = 2$.

3.7 Conclusion and Future Work

In this chapter, we investigated various image registration issues mathematically, and addressed some of them. In summary, to have a differentiable distance function it is enough to have a differentiable image on $\mathbb{R}^2$. The images are made smoother using Gaussian convolution. Moreover, there are some spurious local minima in the smooth distance function, which are caused by too much detail in the images. They can also



(a) Before registration

(b) After registration

Figure 3.39: Discrepancy image of the two bananas given in Figure 3.37 (a): before, (b): after registration in $PSL(2,\mathbb{C})$, where the images are convolved with a Gaussian with $\sigma = 2$.

Figure 3.40: An image that does not tend to a constant intensity toward its edges.

be eliminated from the distance function by a Gaussian convolution of the images with σ large enough. Therefore, we developed an image registration method based on this knowledge. So, the important factors which influence the registration are:

- Background of images: If images tend to a constant value toward their edges, then their domain easily can be extended to $\mathbb{R}^2$ continuously. But images that do not tend to a constant toward their edges, for example Figure 3.40, may need to be convolved with a function to bring their edges continuously to a constant value on $\mathbb{R}^2$; we leave this as future work.
- Smoothness of the images.
- Dimension of the group of transformations: presence of spurious local minima

is higher when the dimension of the group is higher. We suggest convolving the image with σ large enough to remove most of them.

- How good the match is.
- Real images versus synthetic images: Registration of real images is more difficult than synthetic images, because in the synthetic image there is a point where the images match perfectly, while is unlikely in real images.

In this research we choose the size of convolution kernel (σ) experimentally. Finding a general way to determine the σ, if it is possible at all, we leave as future work.

Chapter 4

Image Registration Using Finite Dimensional Lie Groups

In this chapter, the finite dimensional Lie groups that were discussed in Chapter 2, and the registration algorithm that was described in Chapter 3, will be employed to register images. The objectives of this chapter are:

- To review Thompson's *Theory of Transformation* in light of modern understanding of image registration.
- To reproduce Thompson's work using registration in finite dimensional Lie groups.
- To study the benefits of using finite dimensional groups, and to study the benefits of using subgroups of a higher dimensional group.

4.1 On the Theory of Transformations

As was explained in Chapter 1, Thompson introduced the idea of the *Theory of Transformations* which is about using simple transformations to transfer the appearance of one organism into another, such that their prominent features are matched. In this thesis we reproduce and study Thompson's work mathematically for the following reasons:

1. No one has apparently reproduced Thompson's work in the 100 years since it was first published. Two important steps in this direction, that we discuss below and in Sections 4.2, 4.5, and 4.6, are those of Milnor [47] and Petukhov [53]. Petukhov shows the importance of non-Euclidean transformation by applying them to describe organisms. He says:

   > "Symmetry in the form of biological bodies has always attracted the attention of natural scientists as one of the most remarkable and mysterious natural phenomena. School curricula in biology include numerous instances of rotational, translational and mirror symmetries, and also symmetries of similarities of scale in biological bodies.... Let us note once more that the group of similarity transformations is the core of the Euclidean geometry and so such symmetries may be referred to as Euclidean as opposed to non-Euclidean symmetries which represent transformations from non-Euclidean groups. We should not overlook the fact that from the geometrical (i.e. group invariant) viewpoint the entire classical biomorphology is essentially an extension of the group of similarity transformations. This is the case for morphological studies and theories of mirror symmetry and asymmetry of biological bodies...."

   The idea of Thompson can easily extend to cover ontogeny[1] rather than only comparing the adult stages of organisms; Petukhov [53] applied this idea to human growth (see Section 4.6).

2. The biologist Arthur Wallace wrote a review of Thompson's work [4] in light of modern biology. He says: D'Arcy Thompson explained natural phenomena in terms of physical and mathematical laws. His theory of transformations suggest coordinated as opposed to piecemeal changes to shape during evolution, an important matter that is ignored in 'modern synthesis' of evolutionary theory. In regard to embryology and inheritance Thompson put emphasis on the forces that shape the organisms, in contrast to the common view of biologists which concern the material that makes up an organism, such as a piece of embryonic tissue. But now we can knit the two different point of views, as Arthur says:

   > "A gene is indeed a material thing. But its pattern of expression during development is dynamic. And the forces that cause this dynamism include such things as transcription factors and morphogens, which are themselves material things.... All the tools are now in place to examine

[1] Ontogeny pertains to the developmental history of an organism within its own lifetime.

Figure number in Thompson [67]	Transformation group
515	$x \mapsto ax$, $y \mapsto y$
513.2	$x \mapsto ax$, $y \mapsto by$
509,510,518	$x \mapsto ax$, $y \mapsto cx + dy$ (shears)
521-522,513.5	$x \mapsto ax + by$, $y \mapsto cx + dy$ (affine)
506,508	$x \mapsto ax$, $y \mapsto g(y)$
511	$x \mapsto f(x)$, $y \mapsto g(y)$
517-520, 523, 513.1, 513.3, 513.4, 513.6, 514, 525	Conformal
524	'Peculiar'

Table 4.1: Transformation groups between the related forms used in [67]. Table is given in [42].

> the mechanistic basis of transformations. Not only do we have phylogenetic systematics and evo-devo[2], but, so obvious that it is easy to forget, we have computers, and especially, in this context, advanced computer graphics. It seems almost incredible that D'Arcy Thompson achieved what he did without this modern aid to morphology, working in an era in which the forms of animals were all individually hand-drawn."

3. In the modern era, the evolutionary development of organisms is studied to discover the relationship between them, or to find how an organism itself evolved. In this study, one seeks the genetic variations for distinct and independent parts of a body. But in Thompson's theory of transformation, one organism evolves into another not by successive small changes in individual body parts, but by large-scale transformation of the whole body. Possibly, that transformation carries significant information itself, in contrast to the modern view of evo-devo.

4. Thompson appears to be using simple classes of transformation [42]. In [42] the authors say:

   > "We draw attention to two key aspects of Thompson's examples: (i) the transformations are as simple as possible to achieve what he considers a good enough match (see Table 4.1); and (ii) the classes of transformations that he considers all forms groups (or pseudogroups), either finite or infinite dimensional."

   So here we classify Thompson's examples based on the Lie groups given in Chapter 2, see Table 4.2. We also show some of the related forms given in Figures 4.1, 4.2, and 4.3. References to the Figure numbers in [67] are given in Table 4.2.

[2] evolutionary development.

Figure number in Thompson [67]	Transformation group
515,513.2	$Sim \times Sim$
509,510,517-518	shears
513.5	affine
521-522	$PSL(3,\mathbb{R})$
508,511-512	$PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$
506	$Sim \times PSL(2,\mathbb{R})$
513.1,513.3,513.4,513.6, 523, 525-526, 550-551	$PSL(2,\mathbb{C})$

Table 4.2: Transformation groups between the related forms used in [67], 2.

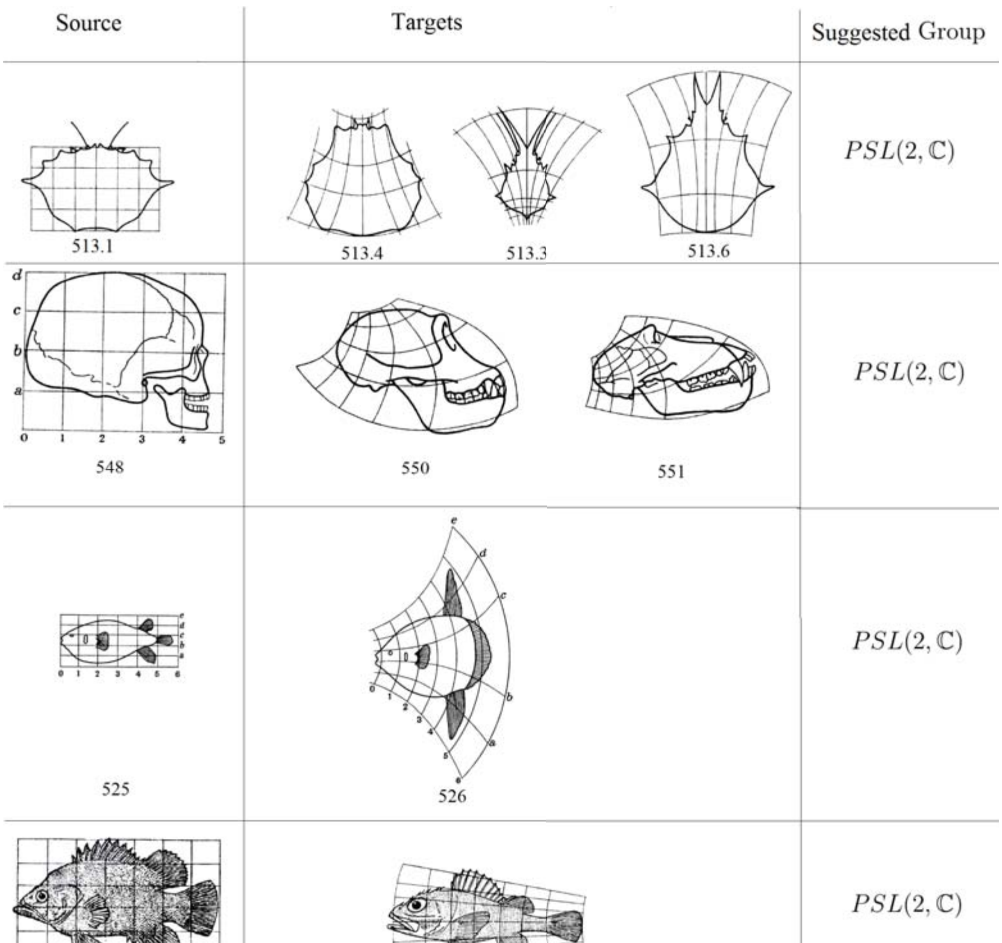


Figure 4.1: Thompson took the outline of a form (source) and drew it against a Cartesian grid, then transformed the grid into another, and overlaid the image of another to form the target. The deformed grids resemble the action of $PSL(2,\mathbb{C})$. Figures are taken from [67]. The groups are our suggestions as possible candidates that will be explained later in this chapter. Numbers at the bottom of the images refer to their number in [67].

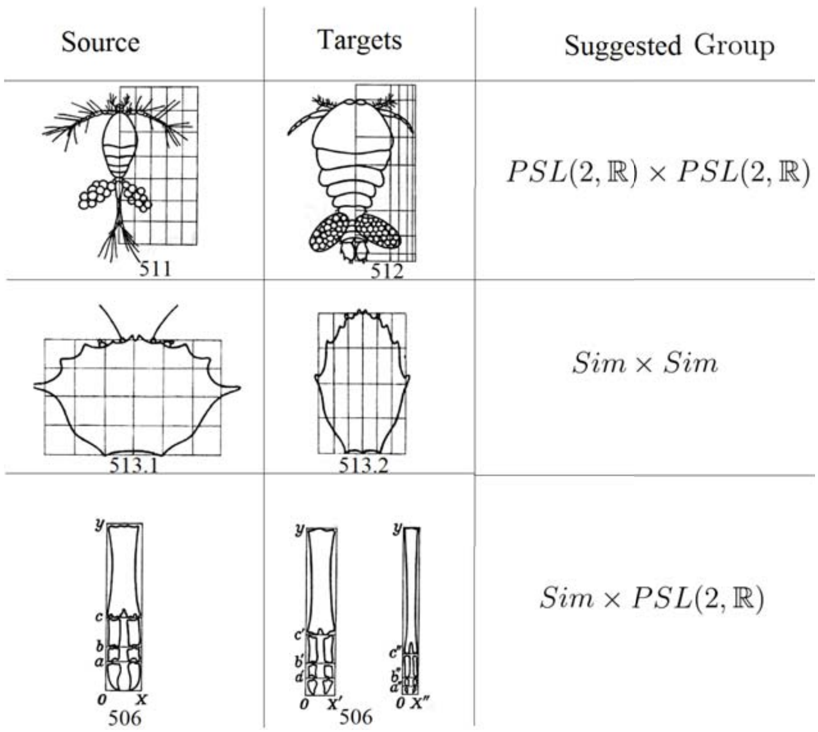


Figure 4.2: Thompson took the outline of a form (source) and drew it against a Cartesian grid, then transformed the grid into another, and overlaid the image of another form, the target. The first row can be related by $PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$, the second row by a subgroup of the group $Sim \times Sim$, and the last row by a subgroup of the group $Sim \times PSL(2,\mathbb{R})$. Figures are taken from [67]. The groups are our suggestions as possible candidates that will be explained later in this chapter. Numbers at the bottom of the images refer to [67].

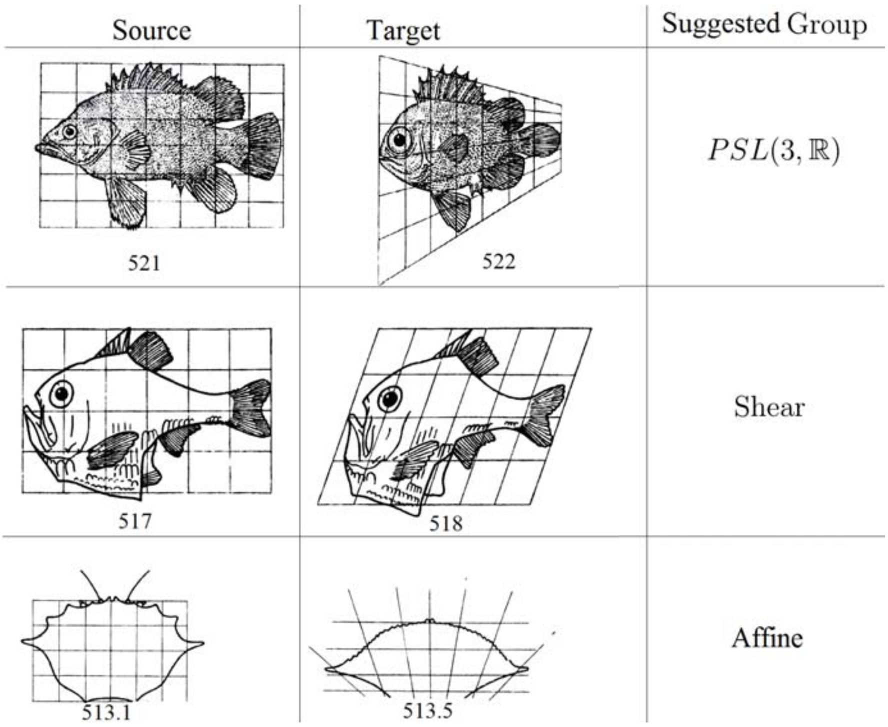


Figure 4.3: Thompson took the outline of a form (source) and drew it against a Cartesian grid, then transformed the grid into another, and overlaid the image of another form, the target. The deformed grids in the first and last row resemble the action of $PSL(3, \mathbb{R})$ on identity grids on source and the middle row simply is a shearing which is a subgroup of $PSL(3, \mathbb{R})$. Figures are taken from [67]. The groups are our suggestions as possible candidates that will be explained later in this chapter. Numbers at the bottom of the images refer to [67].

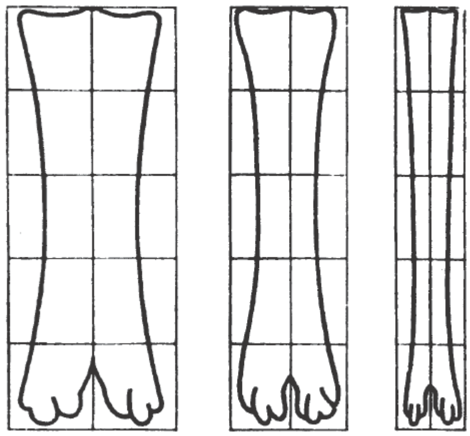


Figure 4.4: Left to right: Cannon-bones of ox, sheep and giraffe.

5. His book is descriptive rather than experimental science. And in some of his examples, he said things which are not at all obvious. For example, in the comparison of fishes in Figure 1.2, he said: '*...it is symmetrical to the eye, and obviously approaches to an isogonal system under certain conditions of friction or constraint'*, (page 1064 in [67]), which merits futher mathematical description. Or in his comparison of the skull of the chimpanzee and baboon (see Figure 4.1, human skull in source column, chimpanzee and baboon skulls in target column.), he said: '*...in case of a baboon, it is obvious that the transformation is of precisely the same order, and differs only in an increased intensity or degree of deformation* (page 1083 in [67]). Again, this needs further mathematical investigation.

6. In some examples, Thompson did not produce the transformed source. By taking the corresponding points in source and target, he sketched a new grid on the target, which shows how the source can be transformed to the target. But, we need to see the transformed source for a better comparison; we also need to see the difference between the transformed source and the target to see how closely they resemble each other.

4.2 The Relationship Between Hoofed Mammals's Feet

Thompson compared the cannon-bones[3] of the ox, sheep and giraffe, see Figure 4.4. He explained that the fundamental difference between the bones is their relative length and breadth. He scaled the feet to a common length of 100, and calculated the breadth

[3] The cannon-bone is a bone in hoofed mammals that extends from the knee or hock to the fetlock.

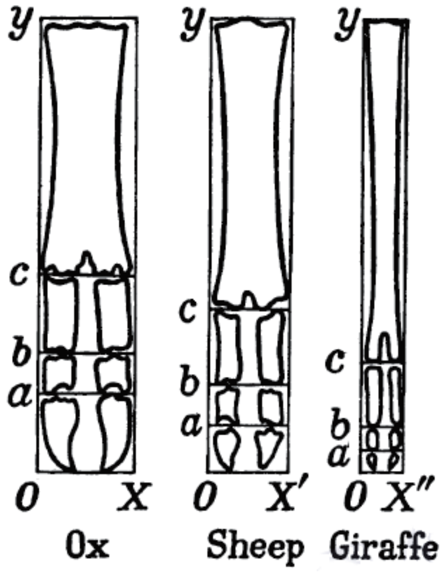


Figure 4.5: Foot of the ox, sheep and giraffe. Feet are scaled to a unit length, then corresponding points are chosen between them to compare the parts. Figure is taken from [67].

	o	oa	ob	oc	oy
ox length	0	18	27	42	100
sheep length	0	10	19	36	100
giraffe length	0	5	10	24	100

Table 4.3: The measurements of oa, ob, oc and oy parts in the ox, sheep and giraffe feet in Figure 4.5.

of the sheep as two-thirds of the breadth of the ox and the breadth of the giraffe as one-third of the breadth of the ox. Next he took the entire foot of the ox, sheep and giraffe; see Figure 4.5.

He said comparing the entire foot is more difficult, since there are several parts and each part elongates differently. He chose corresponding points o, a, b, c in each foot, see Figure 4.5, and then measured oa, ob, oc and oy, which are given in Table 4.3.

Thompson sketched three curves as a function of the ox length passing through: {(0, 0), (18, 18), (27, 27), (42, 42), (100, 100)} which is the ox length curve (line), {(0, 0), (18, 10), (27, 19), (42, 36), (100, 100)} the sheep length curve, and {(0, 0), (18, 5), (27, 10), (42, 24), (100, 100)} the giraffe length curve; see Figure 4.6. In fact, these curves are an example of landmark registration in $\mathit{Diff}([0,1]) \times \mathit{Sim}(1)$, where $\mathit{Sim}(1)$

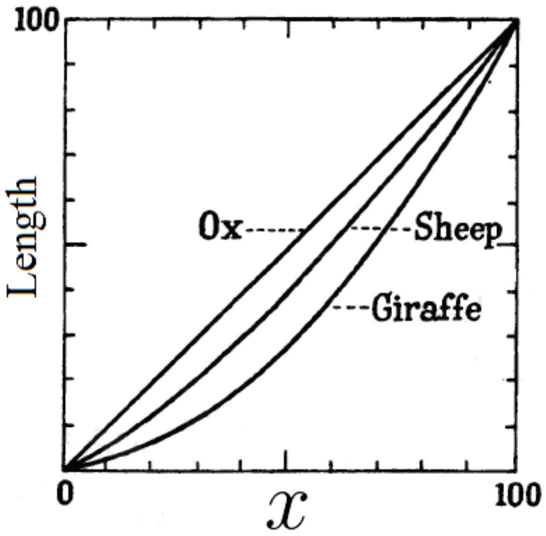


Figure 4.6: The curves of feet (the ox, sheep and giraffe) length as functions of the ox length, taken from [67].

stands for the one dimensional similarity transformation. He said that the graphs show us that there is a comparatively simple equation between the length of the ox, sheep and giraffe, which will enable us to draw a sheep or giraffe foot from an ox foot.

Milnor tried to find these transformations in [47]. He took two sets of four marked points: o, a, b, c and a, b, c, y and calculated the cross-ratio[4]; in the vertical direction for the ox, sheep and giraffe as follows. Let $[o, a, b, c]$ be the cross-ratio of points. He obtained:

Cross-ratio	$[o, a, b, c]$	$[a, b, c, y]$
Ox	$[0, 18, 27, 42] = 2.40$	$[18, 27, 42, 100] = 3.36$
Sheep	$[0, 10, 19, 36] = 2.91$	$[10, 19, 36, 100] = 3.66$
Giraffe	$[0, 5, 10, 24] = 2.71$	$[5, 10, 24, 100] = 4.50$

This indicates that the cross-ratio is almost constant in the vertical direction. He also calculated the two-dimensional cross-ratios. The cross-ratio of vertices of a rectangle with $\triangle x$ length and $\triangle y$ width is $(\frac{\triangle y}{\triangle x})^2$. Milnor calculated the cross-ratios of four points on the vertices of the rectangles drawn by Thompson, which are given in the following table.

[4]The one-dimensional cross-ratio is the restriction of the two-dimensional cross ratio to real numbers.

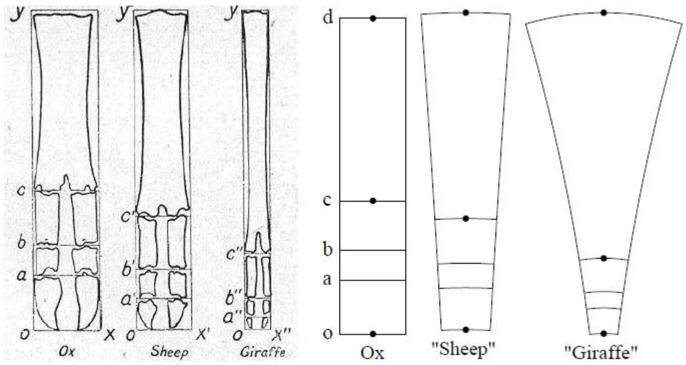


Figure 4.7: A $PSL(2, \mathbb{C})$ transformation that carries the three marked points on the ox foot to the corresponding points on the sheep and giraffe feet, taken from [47].

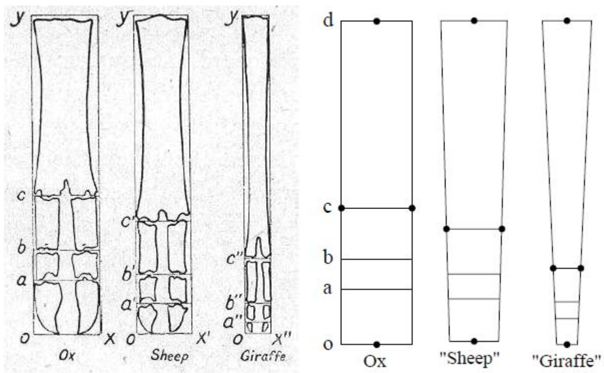


Figure 4.8: A $PSL(3, \mathbb{R})$ transformation that carries the four marked points on the ox to the corresponding points on the sheep and giraffe feet, taken from [47].

	Ox	Sheep	Giraffe
2D cross-ratio	23	34	135

This shows that the feet are not related by $PSL(2, \mathbb{C})$ transformations. Nevertheless, Milnor marked three landmarks in the ox, the sheep and the giraffe and found a $PSL(2, \mathbb{C})$ transformation that exactly registered the three points; see Figure 4.7. He also marked four landmarks and found a $PSL(3, \mathbb{R})$ transformation that exactly registered the four points; see Figure 4.8. It can be seen in Figures 4.7 and 4.8 that the transformations blow up the top part of the ox foot, while the breadth of the sheep and giraffe feet are consistent along the length of the feet. Therefore, the feet are not well registered by $PSL(2, \mathbb{C})$ or by $PSL(3, \mathbb{R})$. Yet, there is another relevant group (actually, product-group) of transformations, $PSL(2, \mathbb{R}) \times PSL(2, \mathbb{R})$, that Milnor did not test. We consider this group.

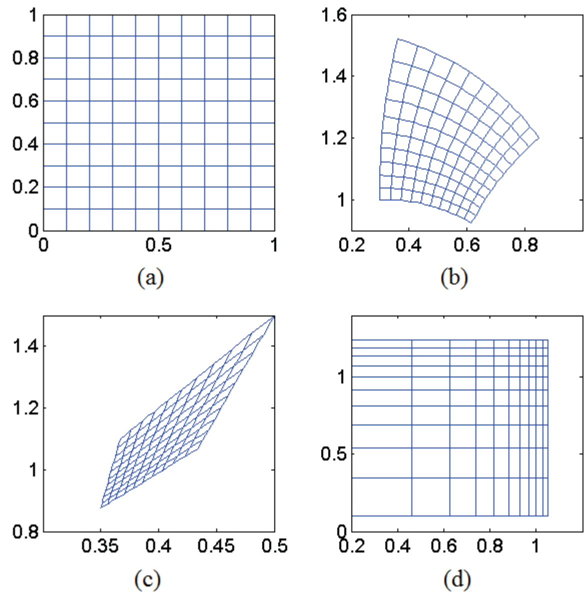


Figure 4.9: (a) A rectangular grid in the plane. The grid is transformed by (b): $PSL(2, \mathbb{C})$, (c): $PSL(3, \mathbb{R})$, (d): $PSL(2, \mathbb{R}) \times PSL(2, \mathbb{R})$ transformations.

Looking at the ox, sheep and giraffe feet, it can be observed that the transformations need to elongate the length of parts of the feet unevenly. Also, Milnor showed that the cross-ratio is almost constant vertically. Therefore, the feet can be related by a $PSL(2, \mathbb{R})$ transformation vertically, and by scaling horizontally.

Figure 4.9 shows the action of three main groups. Also, comparing the action of the groups in Figure 4.9 shows that the action of $PSL(2, \mathbb{R}) \times PSL(2, \mathbb{R})$ on the grid is close to what is described about the relationship between the feet. Therefore, we register the feet with $PSL(2, \mathbb{R}) \times PSL(2, \mathbb{R})$.

We copied the feet from Thompson [67], pasted them into a graphics editor, and filled their inside with black and outside with white; see Figure 4.10. The size of the images is 114×454 pixels and their domain is $\Omega = [0, 0.25] \times [0, 1]$. Note that it is not necessary to fill the inside of the feet, as our algorithm can also register images which are line drawings, see Section 3.6, although the shading makes the registration easier. For the registration of the feet, we need to ensure that each corresponding part matches well, and there are small gaps between the parts. So registration of the feet considered as line drawings may have larger errors than registration of the feet with filled interiors, although we did not test this.

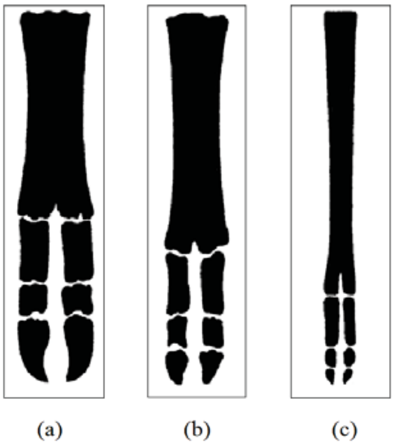


Figure 4.10: Three feet given in Figure 4.5 with their inside filled with black and outside with white, (a) the ox, (b) the sheep, (c) the giraffe.

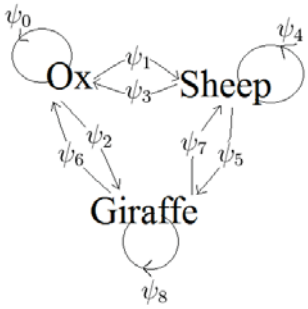


Figure 4.11: Diagram of the transformations between the ox, sheep and giraffe feet. ψ_i, $i = 0, 2, \ldots, 8$ belongs to $PSL(2, \mathbb{R}) \times PSL(2, \mathbb{R})$.

Let the ox, sheep and giraffe feet be J_0, J_1, J_2, respectively. Each pair of images is registered with smoothing parameter $\sigma = 5$ pixels. The values of the distance function (square of L^2-distance) of the images before registration are:

$$\|J_0 - J_1\|_2^2 = 6771$$
$$\|J_0 - J_2\|_2^2 = 16165$$
$$\|J_1 - J_2\|_2^2 = 10697.$$

Nine transformations $\Psi_i(x, y)$, $i = 0, 1, 2, \ldots, 8$ are obtained from the registrations. Let $\Psi_i^{-1} = \psi_i$. Figure 4.11 shows the diagram of transformations that map feet together, and Table 4.4 gives the transformations and the residuals after registration.

Source \ Target	Ox	Sheep	Giraffe
Ox	$\psi_0(x,y) = (x,y)$	$\psi_1(x,y) = \left(\frac{1.1753x-0.0256}{0.5428x+0.8390}, \frac{0.8258y-0.0001}{-0.3949y+1.2110}\right)$, Residual=2059	$\psi_2(x,y) = \left(\frac{1.4322x-0.0976}{-0.9076x+0.7601} \frac{0.6106y+0.0183}{-1.0103y+1.6075}\right)$, Residual=846
Sheep	$\psi_3(x,y) = \left(\frac{0.7951x+0.0301}{-0.8905x+1.2239} \frac{1.1979y+0.0016}{0.364y+0.8353}\right)$, Residual=2833	$\psi_4(x,y) = (x,y)$	$\psi_5(x,y) = \left(\frac{1.1310x-0.0567}{-2.2869x+0.9989} \frac{0.7307y+0.0223}{-0.6278y+1.3494}\right)$, Residual=770
Giraffe	$\psi_6(x,y) = \left(\frac{0.7563x+0.0976}{0.8817x+1.4359} \frac{1.6211y-0.0143}{1.0522y+0.6075}\right)$, Residual=2933	$\psi_7(x,y) = \left(\frac{0.962x+0.0597}{1.9607x+1.1611} \frac{1.3577y-0.0208}{0.6356y+0.7268}\right)$, Residual=1886	$\varphi_8(x,y) = (x,y)$

Table 4.4: Output of the registration in $PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$ between each pair of feet.

These feet are related by $PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$ if and only if the diagram of the transformations given in Figure 4.11 commutes, i.e., if a transformation ψ_i maps J_k to J_h, then the inverse of ψ_i maps J_h to J_k, and if ψ_i maps J_k to J_h, and ψ_j maps J_h to J_l, then $\psi_j \circ \psi_i$ should map J_k to J_l. We can check this for the diagram by comparing the inverses and a composition:

- ψ_1^{-1}, ψ_2^{-1}, ψ_5^{-1} are approximately equal to ψ_3, ψ_6 and ψ_7 respectively. This is confirmed as follows:

$$\begin{aligned}
\psi_1^{-1}(x,y) &= \left(\frac{0.8390x+0.0256}{-0.5428x+1.1753}, \frac{1.2110y+0.0001}{0.3949y+0.8258}\right) \cong \\
\psi_3(x,y) &= \left(\frac{0.7951x+0.0301}{-0.8905x+1.2239}, \frac{1.1979y+0.0016}{0.364y+0.8353}\right),
\end{aligned}$$

$$\begin{aligned}
\psi_2^{-1}(x,y) &= \left(\frac{0.7601x+0.0976}{0.9076x+1.4322}, \frac{1.6075y-0.0183}{1.0103y+0.6106}\right) \cong \\
\psi_6(x,y) &= \left(\frac{0.7563x+0.0976}{0.8817x+1.4359}, \frac{1.6211y-0.0143}{1.0522y+0.6075}\right),
\end{aligned}$$

$$\begin{aligned}
\psi_5^{-1}(x,y) &= \left(\frac{0.9989x+0.0567}{2.2869x+1.1310}, \frac{1.3494y-0.0223}{0.6278y+0.7307}\right) \cong \\
\psi_7(x,y) &= \left(\frac{0.962x+0.0597}{1.9607x+1.1611}, \frac{1.3577y-0.0208}{0.6356y+0.7268}\right),
\end{aligned}$$

- ψ_1 and ψ_5 transform the ox to the sheep and the sheep to the giraffe respectively; so $\psi_5 \circ \psi_1(x,y)$ should transform the ox to the giraffe, and be equal to ψ_2:

$$\begin{aligned}
\psi_5 \circ \psi_1(x,y) &= \left(\frac{1.2985x-0.0765}{-2.1456x+0.8966}, \frac{0.5946y+0.0249}{-1.0513y+1.5127}\right) \cong \\
\psi_2(x,y) &= \left(\frac{1.4322x-0.0976}{-0.9076x+0.7601}, \frac{0.6106y+0.0183}{-1.0103y+1.6075}\right).
\end{aligned}$$

  It can be seen that the non-linear part of $\psi_5 \circ \psi_1(x,.)$ (which is -2.1456) differs markedly from the non-linear part of $\psi_2(x,.)$, which is -0.9076. The reason for this is because these transformations do not make a perfect match between the feet. However, the action of $\psi_5 \circ \psi_1(x,.)$ and $\psi_2(x,.)$ on $[0, 0.25]$ are almost equal in the interval in which the feet are located, see Figure 4.12. It is only over a larger interval that the difference is appreciable.

This suggests that the feet are related by $PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$.

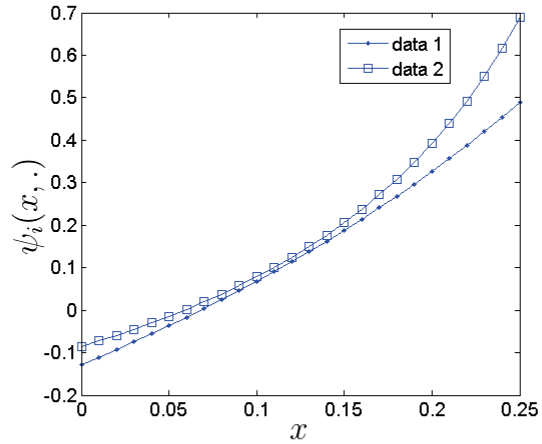
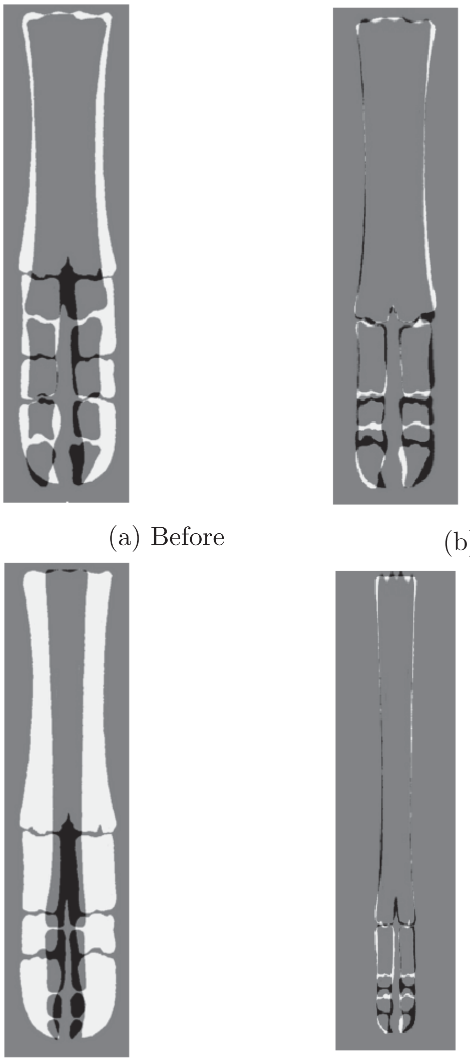


Figure 4.12: Data 1: composition of $\psi_5 \circ \psi_1(x, .)$ where ψ_5 maps the ox to the sheep, and ψ_1 maps the sheep to giraffe. Data 2: $\psi_2(x, .)$, which maps the ox to the giraffe.

Comparing the residuals before and after registration shows that the registration works well in $PSL(2, \mathbb{R}) \times PSL(2, \mathbb{R})$. In Figure 4.13, examples of discrepancies between the ox and sheep, and the ox and giraffe before and after registration are shown.

So far, we have seen that the feet are transformed to each other by $PSL(2, \mathbb{R}) \times PSL(2, \mathbb{R})$ quite well. However, there may be a smaller group that could also be used. As mentioned, the width of the feet seems to be scaled. Therefore, the $PSL(2, \mathbb{R})$ transformation in the x-axis can be replaced by a similarity transformation. There is a subgroup of $PSL(2, \mathbb{R}) \times PSL(2, \mathbb{R})$, which is $Sim \times PSL(2, \mathbb{R})$. Its action on the y-axis is linear-fraction as in the previous group, and on the x-axis it is similarity (scale and translation). Therefore, we register the feet in $Sim \times PSL(2, \mathbb{R})$. Let $\Phi_i^{-1} = \varphi_i$. Figure 4.14 shows the diagram of the transformations which map the feet to each other. Table 4.5 gives these transformations and the residuals of the registration.



(a) Before (b) After

(c) Before (d) After

Figure 4.13: Discrepancy between the ox and sheep before (a) and after (b) registration in $PSL(2,\mathbb{R})) \times PSL(2,\mathbb{R})$. Discrepancy between the ox and giraffe before (c) and after (d) registration in $PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$.

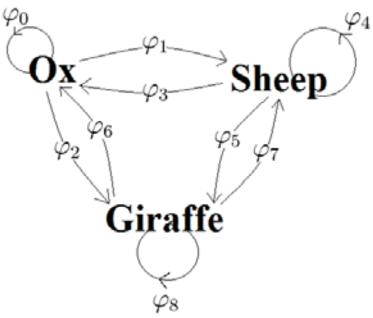


Figure 4.14: The diagram of the transformations in $Sim \times PSL(2,\mathbb{R})$, which maps the feet to each other.

Source \ Target	Ox	Sheep	Giraffe
Ox	$\varphi_0(x,y) = (x,y)$	$\varphi_1(x,y) = \begin{pmatrix} 1.2373x - 0.0238, \\ \frac{0.8246y+0.0007}{-0.3967y+1.2124} \end{pmatrix}$, Residual=2121	$\varphi_2(x,y) = \begin{pmatrix} 2.3678x - 0.1671 \\ \frac{0.6067y+0.0185}{-1.0295y+1.6169} \end{pmatrix}$, Residual=863
Sheep	$\varphi_3(x,y) = \begin{pmatrix} 0.8000x + 0.0192 \\ \frac{1.2013y+0.0015}{0.3759y+0.8329} \end{pmatrix}$, Residual=3049	$\varphi_4(x,y) = (x,y)$	$\varphi_5(x,y) = \begin{pmatrix} 1.9395x - 0.1168 \\ \frac{0.7228y+0.0237}{-0.6433y+1.3624} \end{pmatrix}$, Residual=927
Giraffe	$\varphi_6(x,y) = \begin{pmatrix} 0.4224x + 0.0701 \\ \frac{1.6054y-0.0171}{1.0108y+0.6121} \end{pmatrix}$, Residual=2997	$\varphi_7(x,y) = \begin{pmatrix} 0.5151x + 0.0612 \\ \frac{1.3580y-0.0238}{0.6319y+0.7253} \end{pmatrix}$, Residual=2081	$\varphi_8(x,y) = (x,y)$

Table 4.5: Output of the registration in $Sim \times PSL(2,\mathbb{R})$ between each pair of feet.

Similarly to the diagram of the transformations in $PSL(2,\mathbb{R})\times PSL(2,\mathbb{R})$, the diagram of transformations in $Sim \times PSL(2,\mathbb{R})$ approximately commutes:

- $\varphi_1^{-1} = \varphi_3$, $\varphi_2^{-1} = \varphi_6$ and $\varphi_5^{-1} = \varphi_7$:

$$\begin{aligned}\varphi_1^{-1}(x,y) &= \left(0.8082x + 0.0192, \frac{1.2124y - 0.0007}{0.3967y + 0.8246}\right) \cong \\ \varphi_3(x,y) &= \left(0.8000x + 0.0192\frac{1.2013y + 0.0015}{0.3759y + 0.8329}\right).\end{aligned}$$

$$\begin{aligned}\varphi_2^{-1}(x,y) &= \left(0.4223x + 0.0706, \frac{1.6169y - 0.0185}{1.0295y + 0.6067}\right) \cong \\ \varphi_6(x,y) &= \left(0.4224x + 0.0701, \frac{1.6054y - 0.0171}{1.0108y + 0.6121}\right).\end{aligned}$$

$$\begin{aligned}\varphi_5^{-1}(x,y) &= \left(0.5156x + 0.0602, \frac{1.3624y - 0.0237}{0.6433y + 0.7228}\right) \cong \\ \varphi_7(x,y) &= \left(0.5151x + 0.0612, \frac{1.3580y - 0.0238}{0.6319y + 0.7253}\right).\end{aligned}$$

- $\varphi_5 \circ \varphi_1(x,y) = \varphi_2$:

$$\begin{aligned}\varphi_5 \circ \varphi_1(x,y) &= \left(2.3997x - 0.1630, \frac{0.5956y + 0.0205}{-1.0667y + 1.6424}\right) \cong \\ \varphi_2(x,y) &= \left(2.3678x - 0.1671, \frac{0.6067y + 0.0185}{-1.0295y + 1.6169}\right).\end{aligned}$$

Comparing the residuals of the registration in the subgroup and group shows that the subgroup registers the images as well as the group. In the following section we compare the group and subgroup output in detail.

4.2.1 Comparing The Group and Subgroup Results

We have shown that the diagrams of transformations, Figures 4.11 and 4.14, approximately commute, so the choice of source and target does not matter. Let J_0 (the ox foot) be the source and J_i, $i = 1, 2$, the sheep and giraffe feet be the targets. Figure 4.15 shows the outline of the ox after transformation by the group and subgroup.

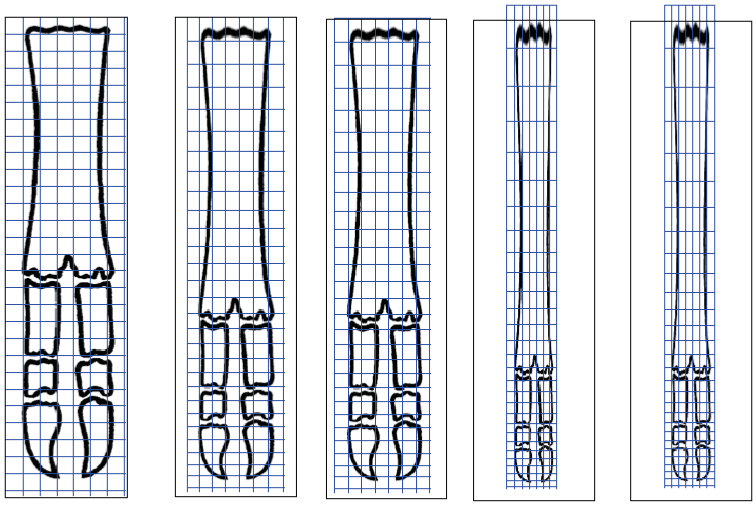


Figure 4.15: (a) The ox foot with a rectangular grid. (b) Transformation of the ox and the rectangular grid to the sheep by group $(PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R}))$, (c) by subgroup $Sim \times PSL(2,\mathbb{R})$. (d) Transformation of the ox and the rectangular grid to the giraffe by group $(PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R}))$, (f) by subgroup $Sim \times PSL(2,\mathbb{R})$.

The group and subgroup have the same action in the y direction, whereas in the x direction they have different actions. However, the grid is deformed in the x direction by the group and subgroup similarly, as we can see in Figure 4.15. The group does not contract or expand the x-axis unevenly. In fact, the action of the group on $[0, 0.25]$ on the x-axis is very close to the subgroup action. Figure 4.16a shows the action of $\psi_1(x,.)$ and $\varphi_1(x,.)$ on $[0, 0.25]$ and Figure 4.16b shows the action of $\psi_2(x,.)$ and $\varphi_2(x,.)$ on $[0, 0.25]$.

Figure 4.17 shows the discrepancy between the transformed ox and the sheep and giraffe in the group and subgroup, which are very similar.

In Figure 4.17, there are some mismatches on the bottom parts of the feet, where they are not matched perfectly. To check whether or not the registration was stuck in a local minimum we performed a one-dimensional landmark registration of the ox and sheep feet along the y-axis. Five corresponding points along the length of them were chosen as shown in Figure 4.18. A transformation along the y-axis is obtained by landmark registration as follows:

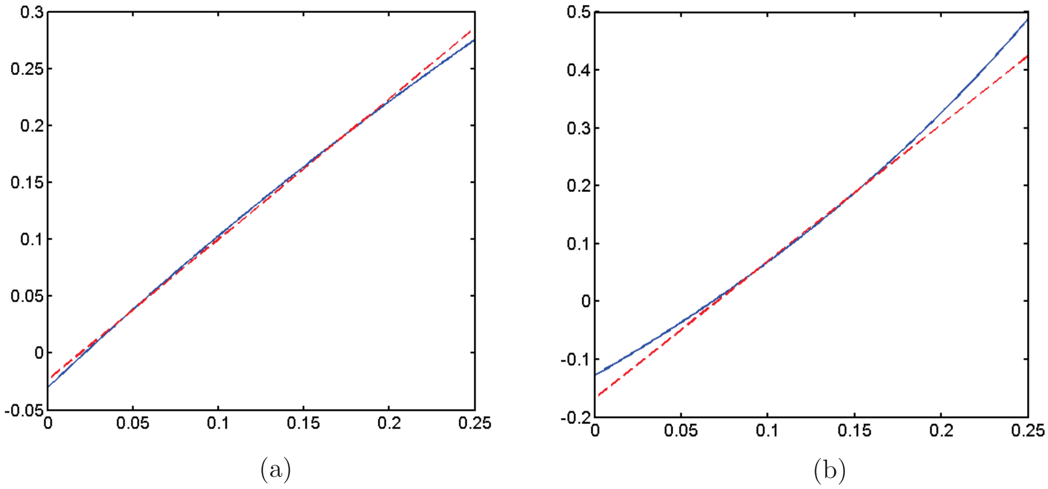


Figure 4.16: The action of two transformations on $[0, 0.25]$ by the group (solid line) and the subgroup (dashed line). Here the transformations match: (a) the ox foot to the sheep foot, (b): the ox foot to the giraffe foot horizontally. The group is $PSL(2, \mathbb{R}) \times PSL(2, \mathbb{R})$ and the subgroup is $Sim \times PSL(2, \mathbb{R})$.

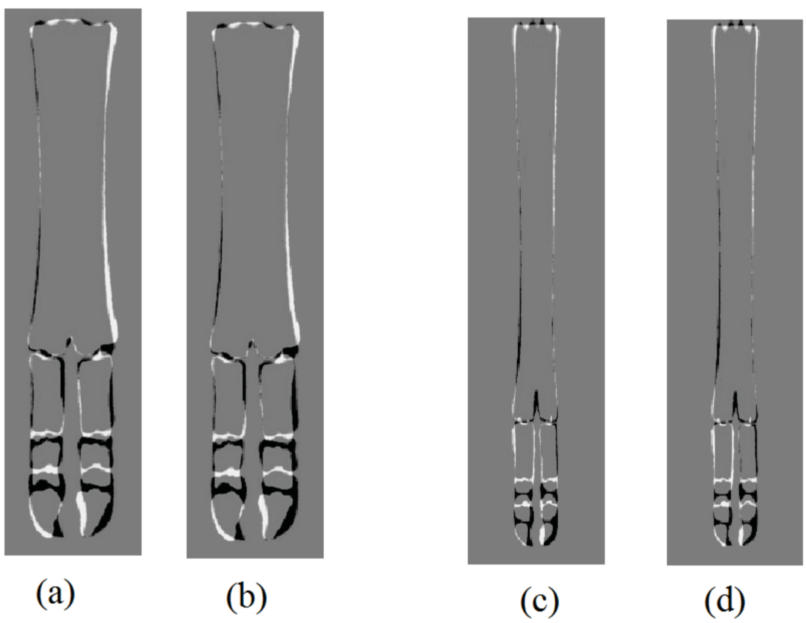


Figure 4.17: Discrepancy between the transformed ox and the sheep, where the ox is transformed by: (a) group $(PSL(2, \mathbb{R}) \times PSL(2, \mathbb{R}))$, (b) subgroup $(Sim \times PSL(2, \mathbb{R}))$. Discrepancy between the transformed ox and the giraffe, where the ox is transformed by: (c) group $(PSL(2, \mathbb{R}) \times PSL(2, \mathbb{R}))$, (d) subgroup $(Sim \times PSL(2, \mathbb{R}))$.

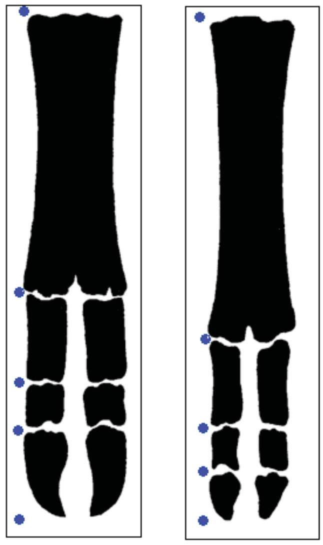


Figure 4.18: Five corresponding points along the length of (a) the ox, and (b) sheep are selected for landmark registration.

$$\phi(.,y) = \frac{0.8105y - 0.0014}{-0.4236y + 1.2346},$$

which is close to $\psi_1(.,y)$ and $\varphi_1(.,y)$. The value of the distance function at $(\varphi_1(x,.), \phi(.,y))$ is 2219, which is greater than its value 2121, at $\varphi_1(x,y) = (\varphi_1(x,.), \varphi_1(.,y))$.

The distance function is not scale invariant. For example, the residual of the transformed ox (to giraffe) and giraffe is smaller that the residual of the transformed giraffe (to ox) and ox, see the residuals in Tables 4.5 and 4.4. Also, the value of the L^2 distance function changes from image to image and group to group. Therefore, for a better comparison of the residuals we normalize the distance function and calculate the geometric mean for each pair of the images.

Let I be the source, J the target and φ_{IJ} a transformation that approximately registers I to J, and φ_{JI} a transformation that approximately registers J to I. The geometric mean of the normalized similarity measure is calculated as:

$$D = 1 - \sqrt{\frac{\|I \circ \varphi_{IJ}^{-1} - J\|_2^2}{\|I - J\|_2^2} \frac{\|J \circ \varphi_{JI}^{-1} - I\|_2^2}{\|I - J\|_2^2}}. \tag{4.1}$$

If I and J match perfectly with φ then $D = 1$, and if $\|I \circ \varphi_{IJ}^{-1} - J\|_2^2 = \|I - J\|_2^2$ and

Pair of images	(Ox, Sheep)	(Ox, Giraffe)	(Sheep, Giraffe)
Group	0.6433	0.9025	0.8873
Subgroup	0.6244	0.9005	0.8701

Table 4.6: The normalized similarity measure D for each pair of images (ox, sheep, giraffe) in the group $(PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R}))$ and the subgroup $(Sim \times PSL(2,\mathbb{R}))$.

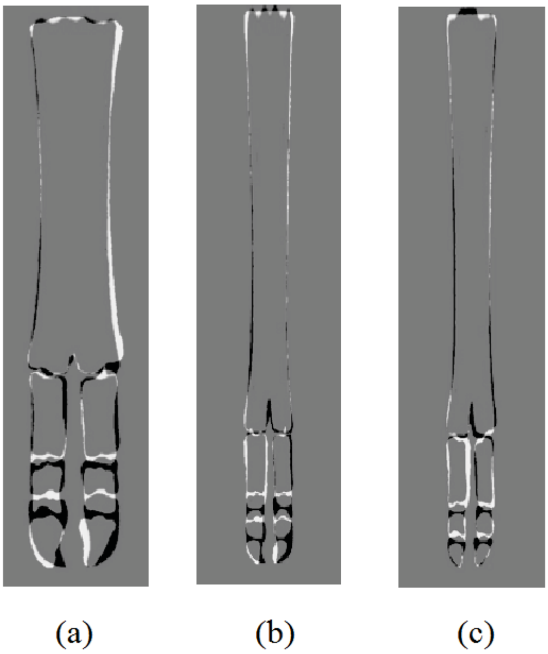


Figure 4.19: Discrepancy between (a) transformed ox (to sheep) and sheep, (b) transformed ox (to giraffe) and giraffe, (c) transformed sheep (to giraffe) and giraffe.

$\|J \circ \varphi_{JI}^{-1} - I\|_2^2 = \|I - J\|_2^2$ then $D = 0$. In black and white images, the value of D can tell us the amount of the overlap and matching between images.

The values of D in the registrations of feet in the group and the subgroup are given in Table 4.6. Because the images are black and white, the value of D can tell us the amount of the matching; for example 90% of the ox is matched to the giraffe by registration, and only 10% does not match. Comparing the values of D given in Table 4.6 between each pair of images for the group and the subgroup tells us that the registration in the subgroup is essentially as good as the registration in the group. Also, the discrepancy between the transformed ox and giraffe, and between the transformed sheep and giraffe is less than the transformed ox and sheep, see Figure 4.19.

This first finite dimensional registration supports Thompson's idea of simple transformation between related forms. We have also compared the group and subgroup registration outputs. It has been shown that registration in a subgroup is as good as registration in a group in this example.

This last point is an example of *model selection*. This is the task of choosing the best model among a sets of models [16]. Some criteria used in model selection are:

- The goodness of fit.
- The simplicity of the model.
- The capability of the model to describe the phenomena that underlie the data.

In the present example, the models are nested. That is, the residuals from registration in the group must necessarily be smaller than the residuals from registration in the subgroup, since there are more degrees of freedom to allow a better match. This is confirmed in Table 4.6. So the question becomes, how much reduction in residual is needed to justify a more complex model? This is the key question in the statistical theory of model selection [16]. Two of the most popular criteria are the Akaike information criterion (AIC),

$$\mathrm{AIC} = -2 \ln \mathcal{L} + 2p,$$

and the Bayesian information criterion (BIC),

$$\mathrm{BIC} = -2 \ln \mathcal{L} + p \ln n.$$

Here $\mathcal{L}$ is the likelihood of the estimated model, p is the number of parameters in the model, and n is the sample size. In the case that the model errors are independent and normally distributed, $\mathcal{L} = n|\ln(\|x - \hat{x}\|^2/n)|$. A lower value of AIC (resp. BIC) is preferred. A rule of thumb is that a difference of AIC or BIC between two models of $2 - 6$ is positive, $6 - 10$ is strong, and more than 10 is very strong.

A review of model selection in ecology [1] found that 84% of studies used AIC, 14% used BIC, and 2% used some other criterion. They argue that AIC is preferred when the 'true' model is extremely complex and essentially unknowable, and prediction errors are to be minimized, while BIC is preferred when the 'true' model is simple and can be in principle determined given enough data. This, to some extent, explains why ecologists prefer AIC to BIC.

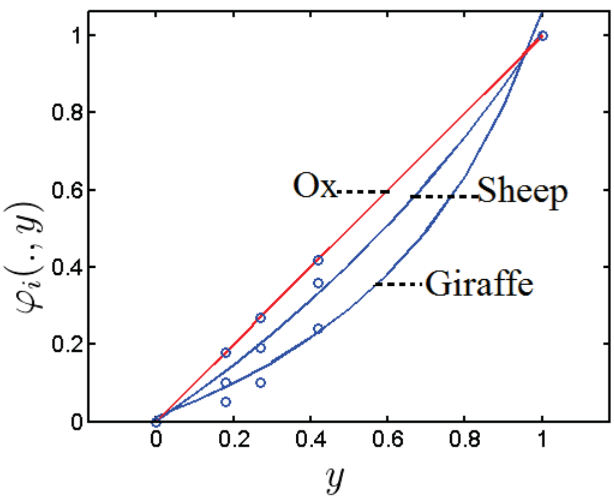


Figure 4.20: Vertical dimension of the ox, sheep and giraffe feet as functions of the ox vertical dimension. Dots are the Thompson data in Table 4.3.

Our context, however, is a little different in that it may actually be true that organisms are (very closely) related by simple models. In the present example, these complications are not really relevant as the more complex model scarcely reduces the residuals and the simpler model is clearly preferred even though $\Delta\text{AIC} \approx 2$. However, in a more detailed study (ideally with a much larger data set) it would be interesting to carry our statistical model selection. In that case a fundamental difficulty to address would be our present complete lack of knowledge as to the distribution from which the data are drawn, and how to weight the different model functions.

Hence, we choose subgroup registration for further investigation of the relationships between the feet. Thompson sketched approximate functions of the lengths of the ox, sheep and giraffe feet, see Figure 4.6. Now we use our transformations $\varphi_i(.,y)$, $i = 0, 1, 2$ to reproduce those curves. They are drawn along with the Thompson data given in Table 4.3, see Figure 4.20. It can be seen they are very close to Thompson's curves. However, the bottom parts are not matched as well as the top parts. However, it was shown by landmark registration that these transformations are the best for the group considered. Of course, it would be possible to obtain a better registration using a larger group. But the only available groups are infinite dimensional, namely *Diff*$\times$*Diff* or *Diff*($\mathbb{R}^2$). Given that a 5-dimension group has already explained 90% of the difference between the feet, it seems unlikely from the point of view of model selection that such drastically more complex models would be preferred.

4.3 The Relationship Between Crabs

Thompson compared the outline of the carapaces of various crabs.

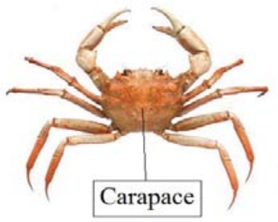


Figure 4.21: The carapace of a crab *Geryon*.

Thompson said that, apart from the details, such as number and situation of marginal spines, which are independent variants, the comparison of carapaces is easy. He put the *Geryon* in a uniform rectangle grid and showed how coordinate changes gave other crabs; see Figure 4.22.

He explained that it is more difficult to compare the entire body of forms than to simply compare corresponding parts. However, in an organism there is one particular mode and direction of variation, which is often more prominent throughout the entire body. So, taking the whole body may give us a better picture of the actual phenomenon.

In the following sections, Thompson's crab forms are registered with different groups. For the registrations, the carapaces of the crabs in Figure 4.22 are copied and pasted into a graphics editor, their grids are deleted and their inside is filled with black.

4.3.1 $PSL(2,\mathbb{C})$ Transformations Between Related Crabs

Thompson's drawings show that there are different transformations that relate crab (a) to the others. In the following, we try to find which crabs can be related by $PSL(2,\mathbb{C})$ and which by $PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$.

First, which crabs can be related by $PSL(2,\mathbb{C})$ transformations? We know that the cross-ratio of four points on the plane is invariant under $PSL(2,\mathbb{C})$ transformations, therefore we calculate the cross-ratios of the four marked corresponding points on the

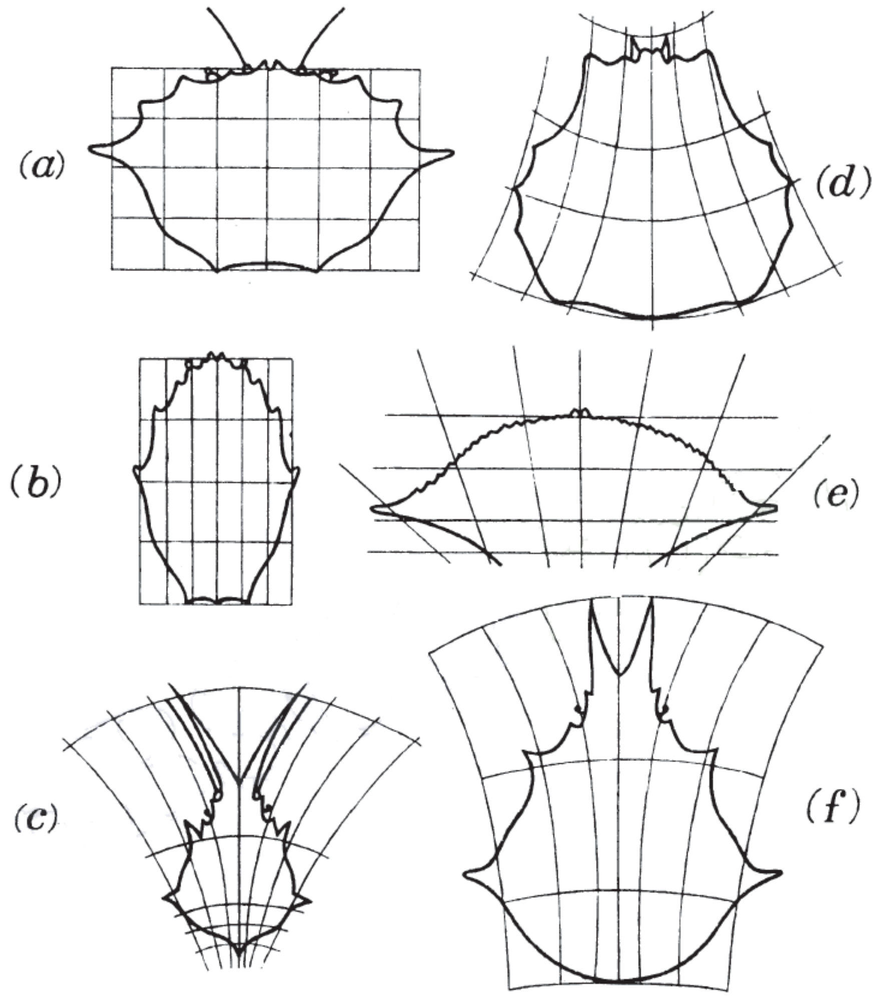
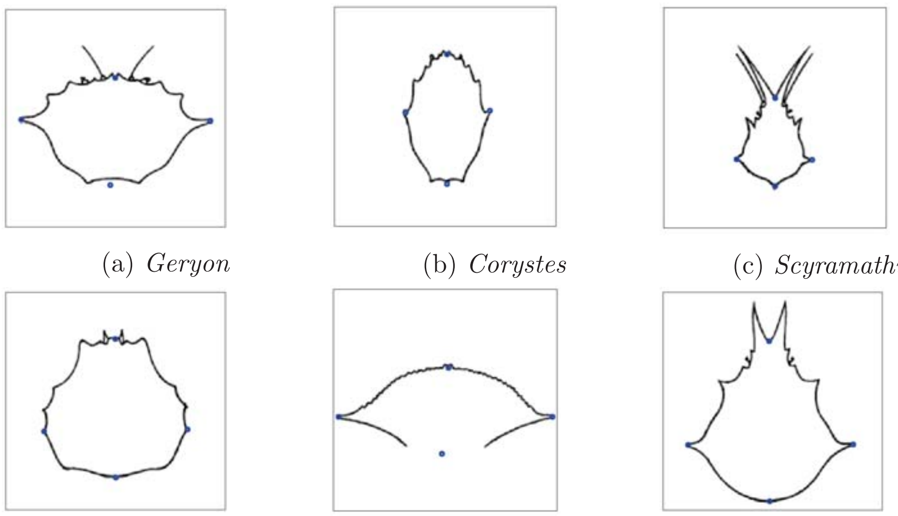


Figure 4.22: Carapace of various crabs, (a) *Geryon*; (b) *Corystes*; (c) *Scyramathis*; (d) *Paralomis*; (e) *Lupa*; (f) *Chorinus*, taken from [67]



(a) *Geryon* (b) *Corystes* (c) *Scyramathis*

(d) *Paralomis* (e) *Lupa* (f) *Chorinus*

Figure 4.23: Four corresponding points in crabs are marked to calculate the cross-ratios.

Crabs	(a)	(b)	(c)
Cross-ratio	$0.52 + 0.3i$	$0.51 - 0.2i$	$0.49 - 0.03i$
Crabs	(d)	(e)	(f)
Cross-ratio	$0.5 + 0.04i$	$0.55 + 0.5i$	$0.5 + 0.03i$

Table 4.7: Cross-ratios of four marked points in the crabs given in Figure 4.23.

crabs. The marked points are shown in Figure 4.23.

Let C_a be the cross-ratio of marked points in *Geryon*, C_b in *Corystes*, C_c in *Scyramathis*, C_d in *Paralomis*, C_e in *Lupa*, and C_f in *Chorinus*. They are given in Table 4.7. As can be seen in the table, C_c, C_f and C_d are almost equal. Therefore, it is possible that there are $PSL(2,\mathbb{C})$ transformations between *Scyramathis*, *Chorinus*, *Paralomis*. So, in the following we take these three crabs and register them in $PSL(2,\mathbb{C})$.

Let *Chorinus* be I_1, *Paralomis* I_2, and *Scyramathis* I_3. If we suppose that these crabs can be related by $PSL(2,\mathbb{C})$, then by using the method given in Section 4.2, all transformations between each pairs of the images can be obtained by transformations between the two other pairs of images, and the choice of source does not matter. So, let I_1 be the source and I_2, I_3 the targets. Two transformations are obtained from the

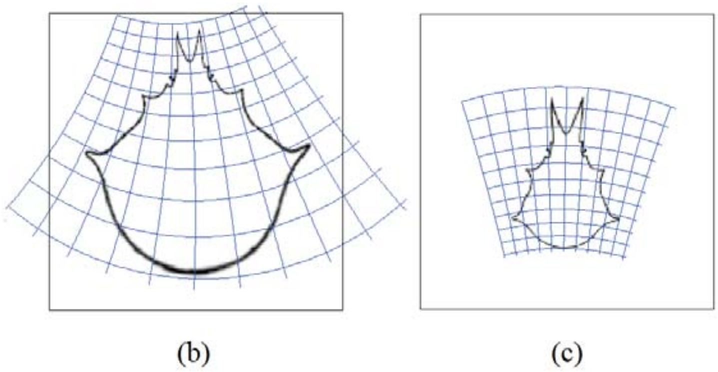


Figure 4.24: (a)*Chorinus* as source (I_1) with a rectangular grid, (b) *Chorinus* after registration with *Paralomis*(I_2) in $PSL(2,\mathbb{C})$ along with the deformed rectangular grid, (c) *Chorinus* after registration with *Scyramathis*(I_3) in $PSL(2,\mathbb{C})$ along with the deformed rectangular grid.

Target	$i=2$	$i=3$
$\|I_1 - I_i\|_2^2$	3272	6607
$\|I_1 \circ \varphi_i^{-1} - I_i\|_2^2$	1096	330

Table 4.8: The residuals of *Chorinus* as source (I_1) and *Paralomis* as target (I_2), and the residuals of *Chorinus* and *Lupa* as target (I_3), before and after registration in $PSL(2,\mathbb{C})$.

registrations, as follows:

$$\varphi_2(z) = \frac{(1.0891 + 0.0335i)z - 0.0112 + 0.1486i}{(0.0781 - 0.6153i)z + 1.0009 - 0.0138i},$$
$$\varphi_3(z) = \frac{(1.3509 - 0.0181i)z + 0.0063 - 0.0932i}{(-0.0027 + 0.3964i)z + 0.7675 + 0.0086i}.$$

φ_2 maps I_1 to I_2, and φ_3 maps I_1 to I_3. Figure 4.24 shows the transformed I_1 along with the deformed grid, and Figure 4.25 shows the discrepancy between transformed I_1 with I_2 and I_3. Note that for a better representation we applied the obtained transformations from the registration on the outline of the crabs in Figures 4.24 and 4.25. Table 4.8 gives the residuals before and after registration. It can be seen that *Chorinus* is well registered with *Scyramathis* and *Paralomis*.

(a) (b)

Figure 4.25: *Chorinus* as source (I_1) is transformed by $PSL(2, \mathbb{C})$ transformations. (a) Discrepancy between transformed *Chorinus* and *Paralomis* as target (I_2), (c) discrepancy between transformed *Chorinus* and *Lupa* as target (I_3).

Crab	(a)	(b)	(c)
Cross-ratio	0.2639	0.287	0.1227
Crab	(d)	(e)	(f)
Cross-ratio	0.4797	0.5163	0.1763

Table 4.9: The one-dimensional cross-ratios of the y-coordinates of the marked points given in Figure 4.26.

4.3.2 $PSL(2, \mathbb{R}) \times PSL(2, \mathbb{R})$ Transformations Between Related Crabs

In this section, we try to find which crabs can be related by $PSL(2, \mathbb{R}) \times PSL(2, \mathbb{R})$. This group preserves the cross-ratio on each axis. We mark four corresponding points on the marginal spine of the crabs. Figure 4.26 shows the marked points whose y-coordinates are taken to calculate the cross-ratios. Figure 4.27 shows the marked points whose x-coordinates are taken to calculate the cross-ratios. Note that the marked points may not correspond exactly, because some crabs do not have distinguished marginal spines, e.g. *Lupa*. The cross-ratios are given in Tables 4.9 and 4.10 respectively.

Comparing the cross-ratios in Table 4.9 and 4.10, indicates that only (a) and (b) may possibly be related by $PSL(2, \mathbb{R}) \times PSL(2, \mathbb{R})$. Nevertheless, we register all the crabs in $PSL(2, \mathbb{R}) \times PSL(2, \mathbb{R})$, where *Geryon* is I_0 (source), *Corystes* is I_1, *Scyramathis* is I_2, *Paralomis* is I_3, *Lupa* is I_4, and *Chorinus* is I_5 (which are the targets). Six

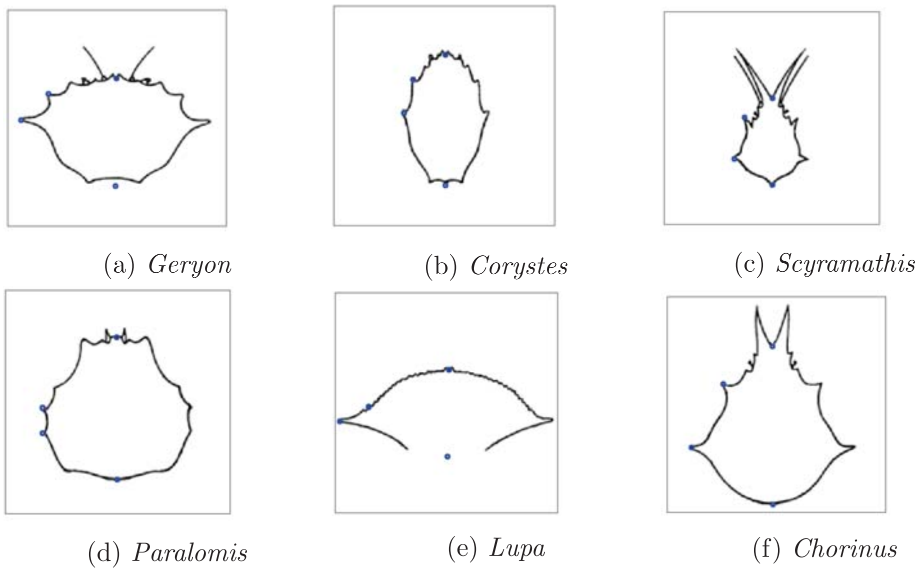


Figure 4.26: Four corresponding points in the crab are marked. The y-coordinates of marked points are taken to calculate the one-dimensional cross-ratios.

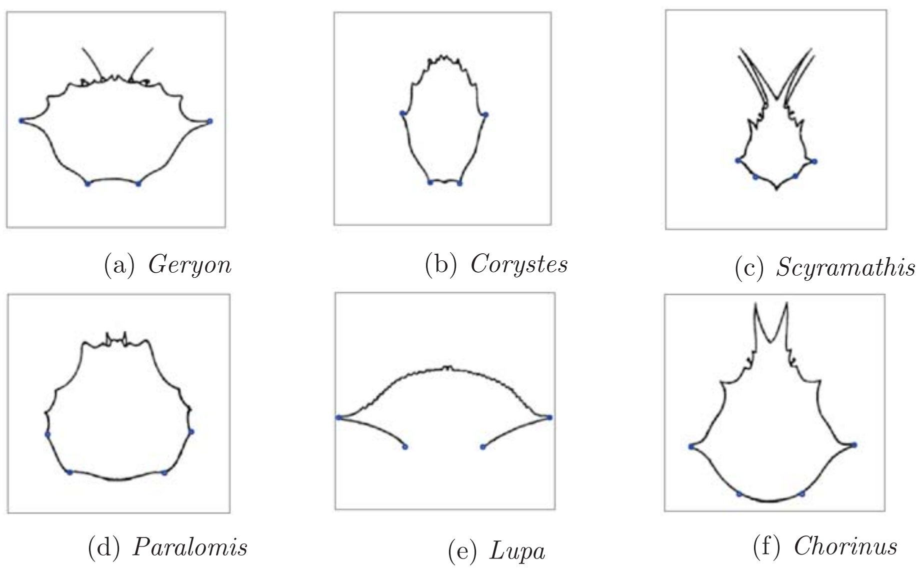


Figure 4.27: Four corresponding points in the crabs are marked to calculate the cross-ratios. The x-coordinates of the marked points are taken to calculate the one-dimensional cross-ratios.

Crab	(a)	(b)	(c)
Cross-ratio	0.3137	0.2285	0.0997
Crab	(d)	(e)	(f)
Cross-ratio	0.0417	0.2118	0.2444

Table 4.10: The one-dimensional cross-ratios of the x-coordinates of the marked points given in Figure 4.27.

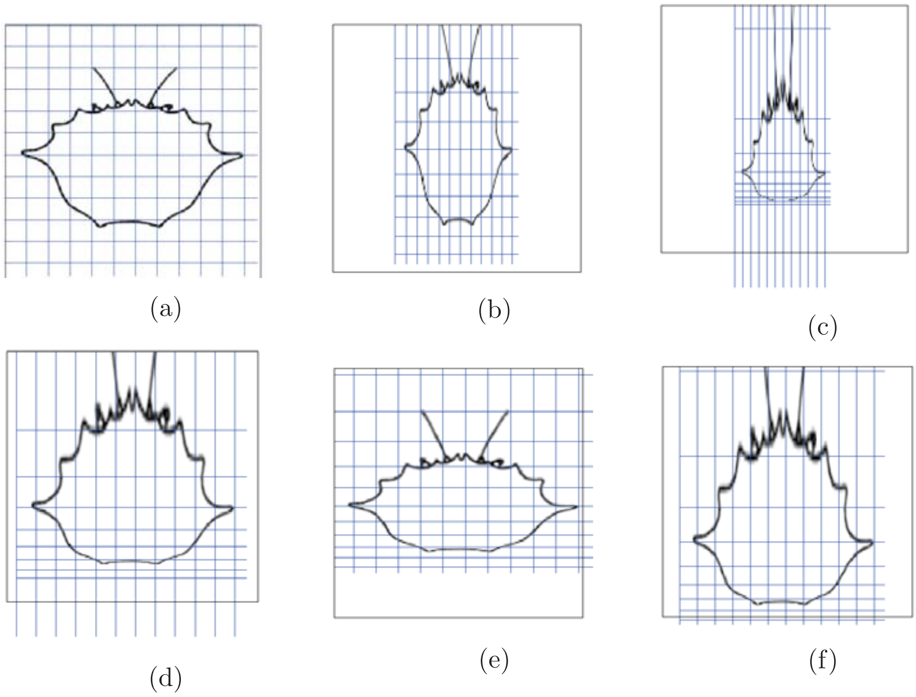


Figure 4.28: (a) *Geryon* as source (I_0). Transformed source after registration in $PSL(2, \mathbb{R}) \times PSL(2,\mathbb{R})$ with: (b) *Corystes* (I_1), (c) *Scyramathis* (I_2), (d) *Paralomis* (I_3), (e) *Lupa* (I_4), (f) *Chorinus* (I_5).

transformations are obtained from the registration as follows:

$$\begin{aligned}
\psi_1(x,y) &= \left(\frac{1.4165x - 0.0167}{-0.081x + 0.7069}, \frac{0.8964y + 0.0117}{-0.5048y + 1.1091}\right), \\
\psi_2(x,y) &= \left(\frac{1.604x + 0.0148}{0.0692x + 0.6241}, \frac{1.1704y - 0.196}{-3.3203y + 1.4106}\right), \\
\psi_3(x,y) &= \left(\frac{1.03888x - 0.003}{-0.0064x + 0.9627}, \frac{0.8834y - 0.0976}{-1.7763y + 1.328}\right), \\
\psi_4(x,y) &= \left(\frac{0.9706x - 0.0157}{0.015x + 1.03}, \frac{1.1103y - 0.0488}{-0.9424y + 0.942}\right), \\
\psi_5(x,y) &= \left(\frac{1.099x + 0.0196}{-0.0213x + 0.9093}, \frac{0.8464y - 0.1571}{-1.7474y + 1.5058}\right).
\end{aligned}$$

Here ψ_i transforms I_0 to I_i, $i = 1,2,3,4,5$. Figure 4.28 shows the transformed source along with the deformed rectangular grid, and Figure 4.29 shows the discrepancy between the transformed source and the targets after registration. Again, we applied the transformations on the outline of the crabs to show a better representation of the discrepancy images and the transformed source.

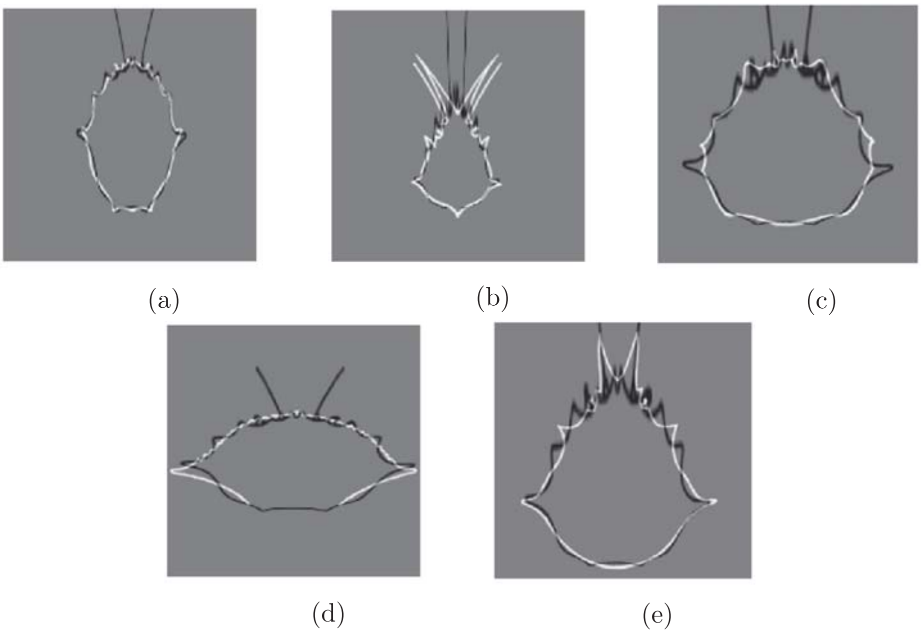
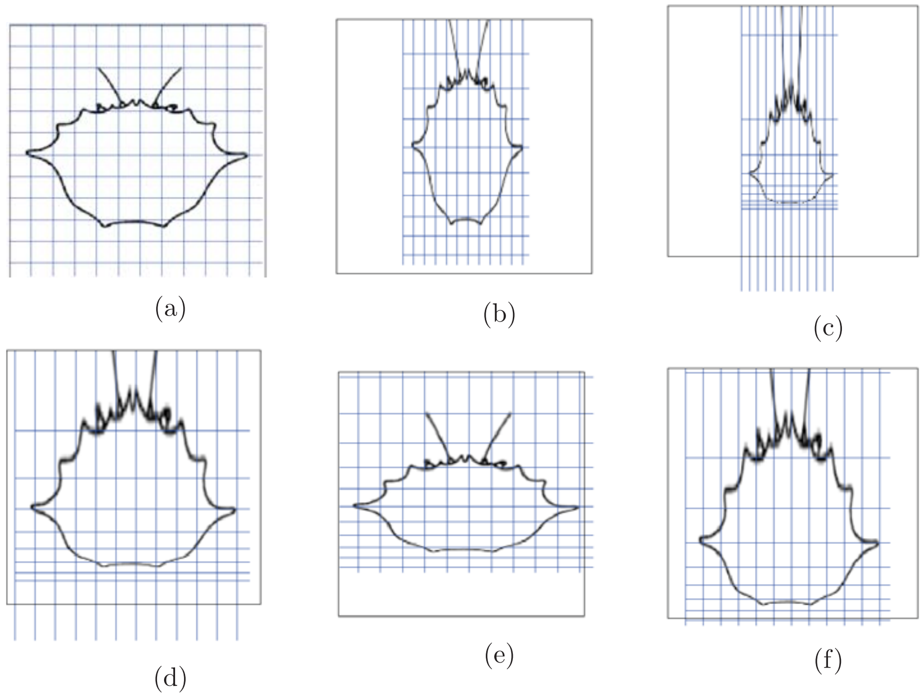


Figure 4.29: *Geryon* as source (I_0) is registered with *Corystes* (I_1), *Scyramathis* (I_2), *Paralomis* (I_3), *Lupa* (I_4), *Chorinus* (I_5) in $PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$. Discrepancy between transformed I_0 and (a) I_1, (b) I_2, (c) I_3, (d) I_4, (f) I_5.

The cross-ratios indicated that only (a) and (b) can be related by a $PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$ transformation, but the overall body of other crabs are also well registered with the transformed crab (a). Therefore, ignoring the small variations on the marginal spine of the crabs, there are simple transformations between them. This again confirms the idea of Thompson's of simple transformations between organisms. In addition, this example illustrates that the approach of detecting group relationships using invariants (such as the cross-ratio) of landmarks, used by Milnor and by us, is not very reliable. It is too susceptible to errors caused by the mis-placement of the landmarks. Image registration, which attempts to match the *entire* body of the image, does not have this weakness, and appears to be more in the spirit of Thompson's theory.

Looking at $\psi_i(x,.)$ it can be seen that the non-linear parameters of $\psi_i(x,.)$ are very small. Therefore, the images can be related by a smaller group, a group whose action on x-axis is only similarity. As we discussed in Section 4.2, the action of $Sim \times PSL(2,\mathbb{R})$ on the x-axis is similarity. Therefore, we register I_0 to I_i, $i = 1, 2, 3, 4, 5$ in this



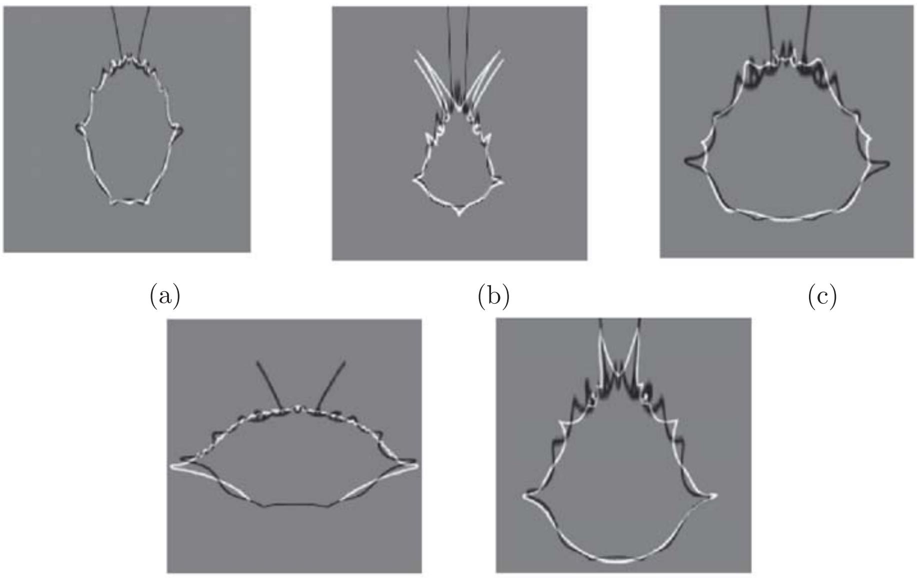
(a) (b) (c) (d) (e) (f)

Figure 4.30: (a) *Geryon* as source (I_0). Transformed source after registration in $Sim \times PSL(2, \mathbb{R})$ with: (b) *Corystes* (I_1), (c) *Scyramathis* (I_2), (d) *Paralomis* (I_3), (e) *Lupa* (I_4), (f) *Chorinus* (I_5).

group, and five transformations are obtained from the registration as follows:

$$\varphi_1(x, y) = \left(2x - 0.0178, \frac{0.8962y + 0.0116}{-0.5065y + 1.1092}\right),$$
$$\varphi_2(x, y) = \left(2.58x + 0.0229, \frac{1.1527y - 0.1884}{-3.149y + 1.3835}\right),$$
$$\varphi_3(x, y) = \left(1.079x - 0.003, \frac{0.8834y - 0.097}{1.079y - 0.003}\right),$$
$$\varphi_4(x, y) = \left(0.9425x - 0.016, \frac{1.1102y - 0.0488}{-0.9418y + 0.9421}\right),$$
$$\varphi_5(x, y) = \left(1.2126x + 0.0222, \frac{0.8405y - 0.1544}{-1.6732y + 1.4972}\right).$$

Here φ_i maps I_0 to I_i, $i = 1, 2, 3, 4, 5$. Figure 4.30 shows the transformed source along with the deformed grids, and Figure 4.31 shows the discrepancy between the transformed source and the targets, where the transformations are applied to the outline of the crabs. The residuals after registration in the group and the subgroup are given in Table 4.11.



(a) (b) (c)

(d) (e)

Figure 4.31: *Geryon* as source (I_0) is registered with *Corystes*(I_1), *Scyramathis*(I_2), *Paralomis*(I_3), *Lupa*(I_4), *Chorinus*(I_5) in $PSL(2,\mathbb{R})\times PSL(2,\mathbb{R})$. Discrepancy between transformed I_0 and: (a) I_1, (b) I_2, (c) I_3, (d) I_4, (f) I_5.

Target	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$
$\|I_0 - I_i\|_2^2$	190460	269480	42854	30305	121150
$\|I_0 \circ \psi_i^{-1} - I_i\|_2^2$	206	528	750	545.	933
$\|I_0 \circ \varphi_i^{-1} - I_i\|_2^2$	216	601	778	546	934

Table 4.11: The residual of source (I_0) and the targets, I_i, $i = 1,2,3,4,5$ before and after registration in the group and in the subgroup. First row: Residuals before registration. Second row: Residuals after registration in the group. Last row: Residual after registration in the subgroup, where the group is $PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$, and the subgroup is $Sim \times PSL(2,\mathbb{R})$.

In this section we gave a second example of registration of related forms in finite dimensional groups that supports the idea of Thompson of using simple transformations. Although the marginal spines of the crabs are not matched perfectly with the groups, the overall shape of their body is well matched. We could get a perfect match with diffeomorphic registration, but should not expect a large reduction in the residuals, as the discrepancy images show. So, it is unlikely that a complicated group (the diffeomorphism group) is preferred by model selection. However, as mentioned before in Section 4.2, for more sophisticated analysis of the model selection we need to have knowledge of the distribution of the data.

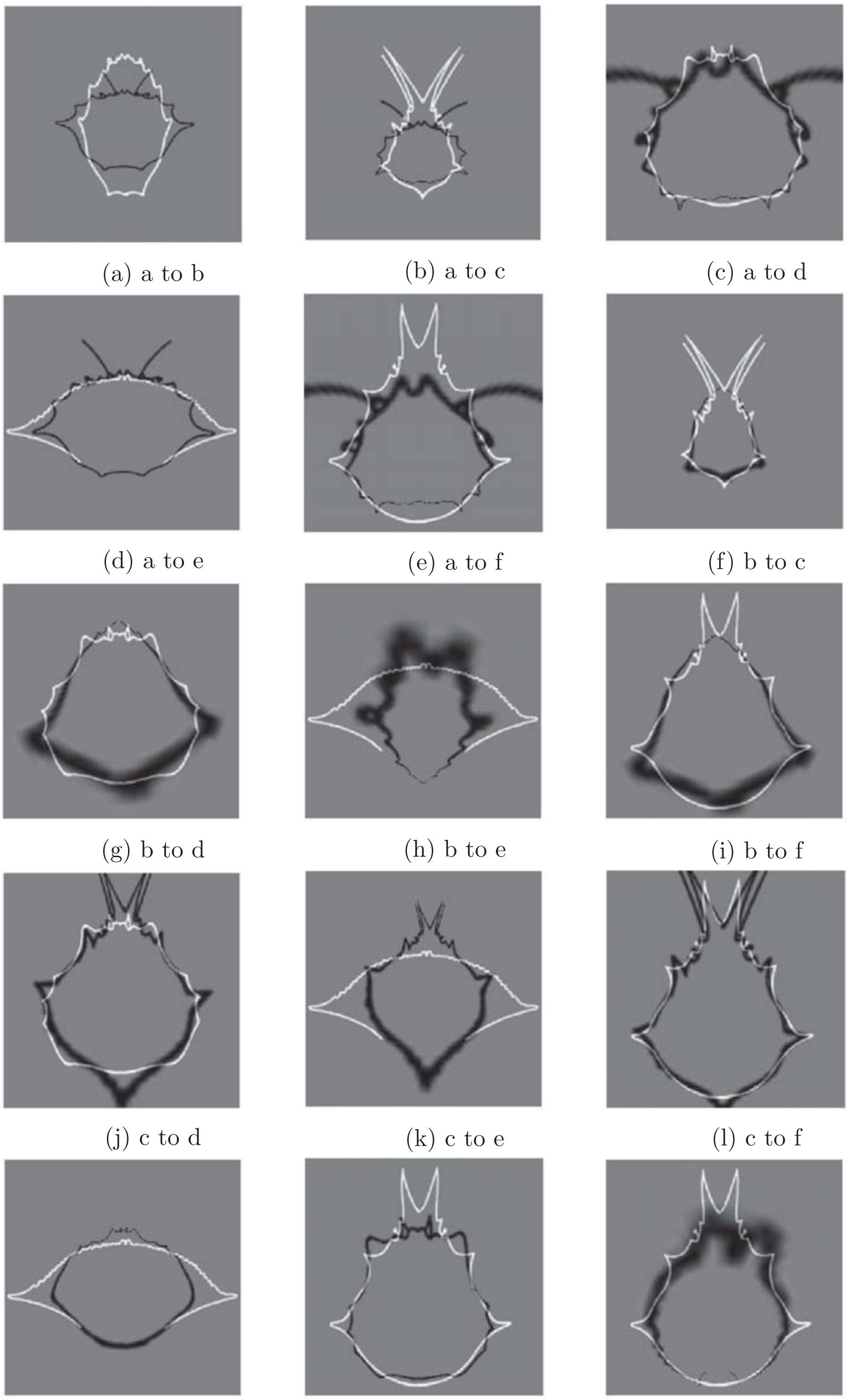
Moreover, we registered the crabs with the group $PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$ and its subgroup $Sim \times PSL(2,\mathbb{R})$. They are nested models. Therefore, the residuals from the group must be smaller than residuals in subgroups, but they are almost equal (see Table 4.11). Also, the non-linear part of $\psi_i(x,.)$ is very small. Therefore, by model selection criteria the subgroup may be preferable to describe the relationship between the crabs.

4.3.3 Registration of Each Pairs of Crabs in $PSL(2,\mathbb{C})$

In light of what we have learned about the unreliability of using invariants of landmarks (caused by positioning of the landmarks) to identify possible groups, we decided to register all pairs of crabs against each other using the $PSL(2,\mathbb{C})$ group. Figure 4.32 shows the discrepancy images and Figure 4.33 shows the transformed sources along with the deformed grids. As can be seen in the figures the overall body of (a) well registered with (d), (e) and (f), and (b) registered well with (c), (d), and (f). This is different to the cross-ratios, which suggested that only (d), (e) and (f) were related by $PSL(2,\mathbb{C})$.

4.4 The Relationship Between Fishes

Thompson suggested a great variety of deformations between fishes. Some of them are simple, while other are more complicated. For example, he illustrated that between *Argyropelecus olfersi* and *Sternoptyx diaphana* in Figure 4.34, there is a simple shear.



(a) a to b (b) a to c (c) a to d

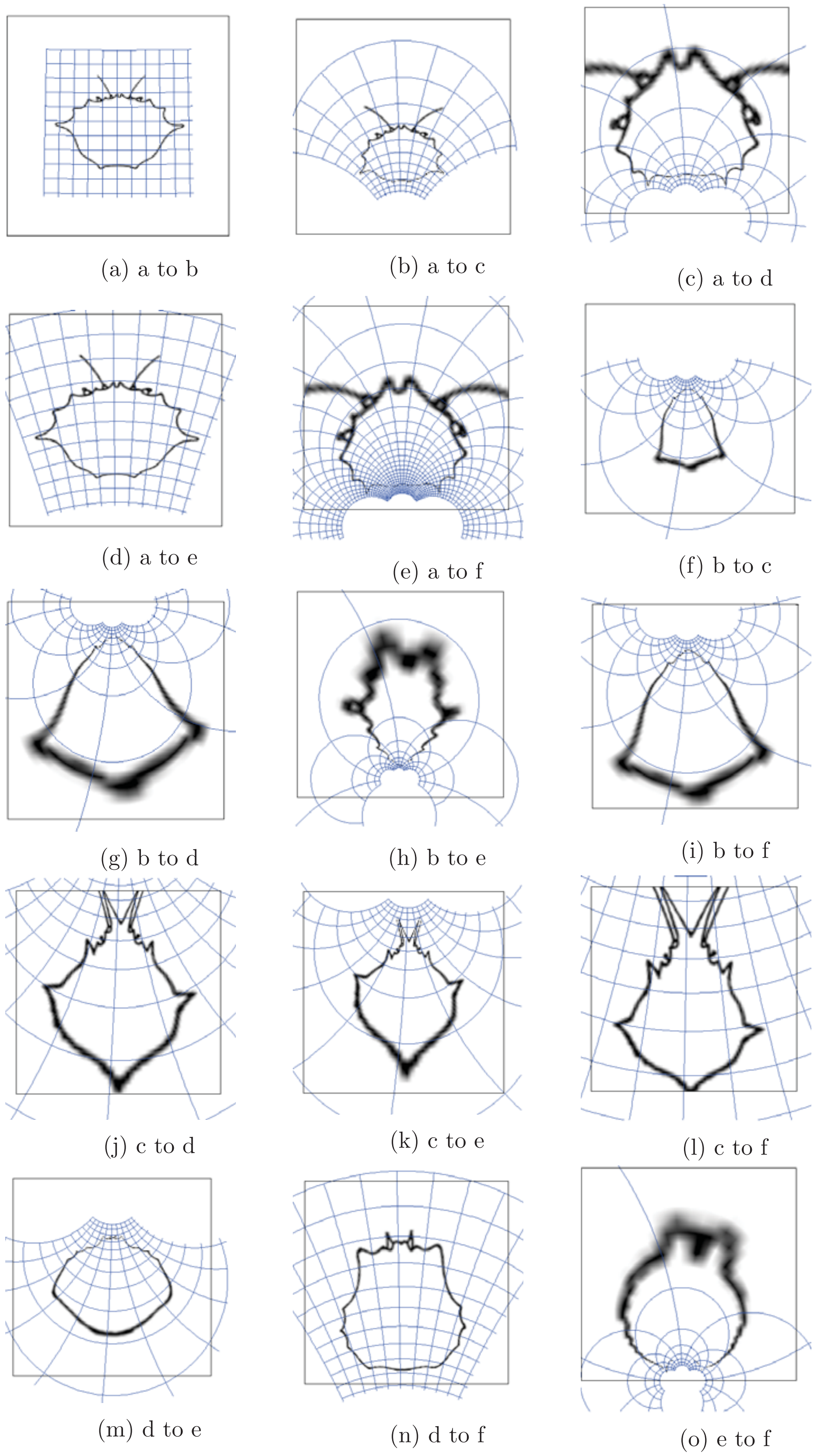
(d) a to e (e) a to f (f) b to c

(g) b to d (h) b to e (i) b to f

(j) c to d (k) c to e (l) c to f

(m) d to e (n) d to f (o) e to f

Figure 4.32: Discrepancy image of each pair of crabs (given in Figure 4.22) after registration in $PSL(2, \mathbb{C})$, where for example, 'a' to 'b', means 'a' is source, 'b' is target.



(a) a to b (b) a to c (c) a to d

(d) a to e (e) a to f (f) b to c

(g) b to d (h) b to e (i) b to f

(j) c to d (k) c to e (l) c to f

(m) d to e (n) d to f (o) e to f

Figure 4.33: Transformed source along with the deformed grid, which is the output of registration of each pairs of crabs (given in Figure 4.22) in $PSL(2,\mathbb{C})$, where for example, ‘a’ to ‘b’, means ‘a’ is source, ‘b’ is target.

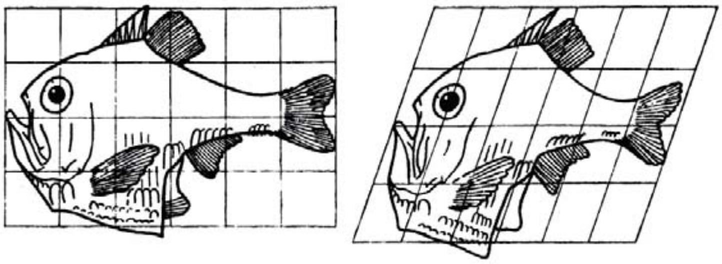
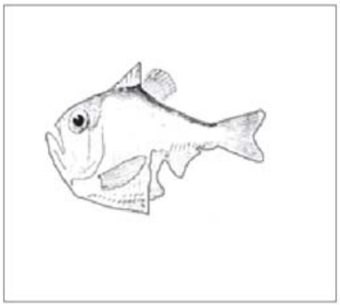
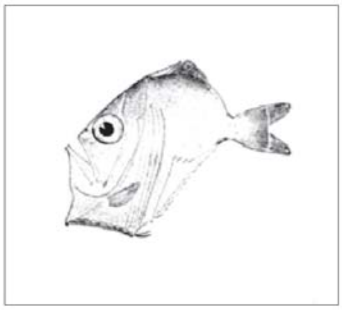


Figure 4.34: Left: *Argyropelecus olfersi*. Right: *Sternoptyx diaphana*. Images are taken from [67].



(a) *Argyropelecus olfersi*



(b) *Sternoptyx diaphana*

Figure 4.35: Images of the same two fishes taken from [50].

In Figure 4.34 Thompson did not show the *Sternoptyx diaphana*, he just transformed *Argyropelecus olfersi*. Instead, we take the real *Argyropelecus olfersi* as source and real *Sternoptyx diaphana* as targets, using the images shown in Figure 6.3.

Shears are a subgroup of $PSL(3, \mathbb{R})$, therefore we register the fishes in the group $PSL(3, \mathbb{R})$. We also register the images with affine and special affine, to study the possible benefit of using lower dimensional groups. Let *Argyropelecus olfersi* be the source I_0, and *Sternoptyx diaphana* the target I_1. The domain of the images is $[-0.5, 0.5] \times [-0.5, 0.5]$. The registration outputs are given in Table 4.12. The residuals before and after registration in the group and subgroups are given in the Table 4.13.

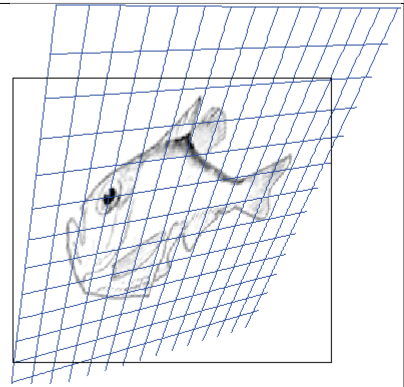
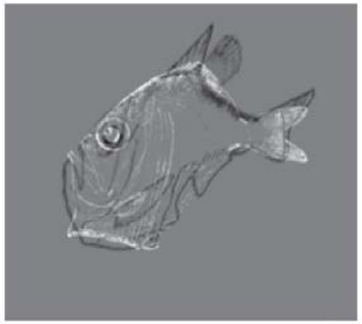
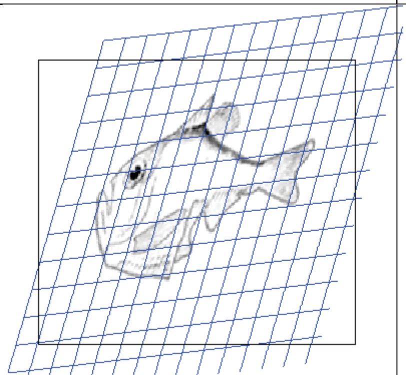
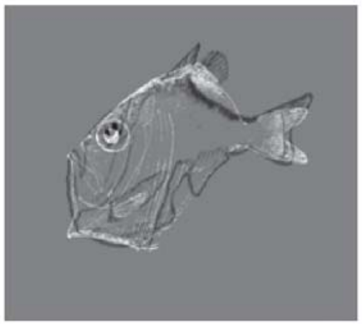
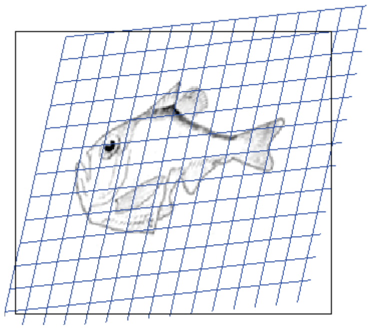
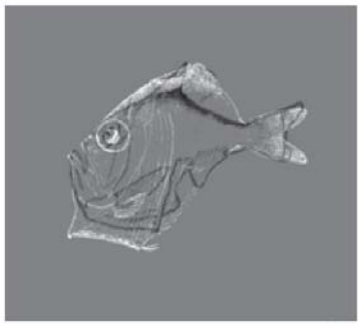
Group	Transformed source on grid	Discrepancy image	Transformation
$PSL(3,\mathbb{R})$			$\psi_1(x,.) = \frac{1.1624x + 0.3472y - 0.0671}{-0.3891x - 0.4897y + 1.0740}$ $\psi_1(.,y) = \frac{0.1872x + 0.8781y - 0.0221}{-0.3891x - 0.4897y + 1.0740}$
Affine			$\psi_2(x,.) = 1.0352x + 0.2683y - 0.0420$ $\psi_2(.,y) = 0.1056x + 0.8135y - 0.0064$
Special Affine			$\psi_3(x,.) = 1.0537x + 0.2062y - 0.0411$ $\psi_3(.,y) = 0.1138x + 0.9713y + 0.0025$

Table 4.12: This table shows the registration result in the group $PSL(3,\mathbb{R})$ and its two subgroups, affine and special affine. The source is *Argyropelecus olfersi* as given in Figure 4.35a, and the target is *Sternoptyx diaphana* as given in Figure 4.35b.

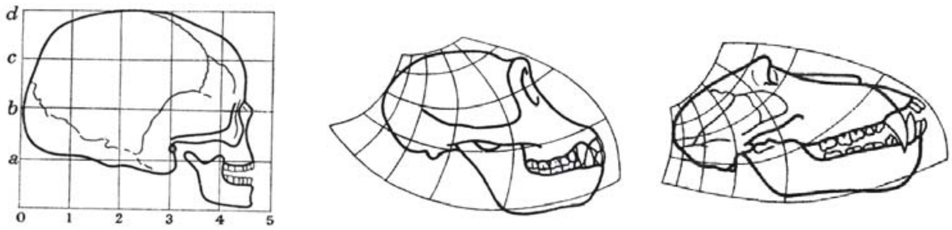
Group	$PSL(3, \mathbb{R})$	Affine	Special Affine
$\|I_0 - I_1\|_2^2$	606.21	606.21	606.21
$\|I_0 \circ \psi_i^{-1} - I_1\|_2^2$	373	407	414

Table 4.13: *Argyropelecus olfersi* is the source (I_0); *Sternoptyx diaphana* is the target (I_1). First row are the residuals before registration and second row after registration in the group $PSL(3, \mathbb{R})$ and its two subgroups: affine and special affine.

Between these three groups, the affine group may be preferred to describe the relationship between fishes. First we compare the result from the registration in the affine and the special affine group. Comparing the residuals of registration in affine (407) and special affine (414), they are very close. The residuals in registration in the group (affine) is not reduced significantly in comparison to the subgroup (special affine). As mentioned in Chapter 3 one of the important factor in registration is to determine how good the match is. Comparing the residuals tells us that special affine makes as good a match as the affine group. But the goodness of the match cannot only be determined by the residuals. We need also to compare the discrepancy images. In fact, comparing the discrepancy images, it can be seen that affine makes a better match. We believe the weakness of the residuals as a criterion in this example is due to the grey-scale nature of the images and the overall lower quality of the match. In biological examples like this, the human operator is able to make judgements about which body parts are inessential to the match (e.g the fins) and which are essential (the body).

Now between the affine and the projective group, affine is preferred, because, first neither the residual of the group (373) is significantly smaller, nor does it make a significantly better match in comparison to the affine, which has fewer parameters than the group.

In this section we have given a third example of simple transformations between related forms (fishes), in which the registration results support Thompson's idea. In the following section we reproduce another of Thompson's examples, which is about the relationship between human skulls and simian skulls. The information from this section will be used in Chapter 6. In Section 4.6 we approximate human skull growth by two main groups and compare the results. The results from this section will be used in Chapter 5.



(a) human (b) chimpanzee (c) baboon

Figure 4.36: Human, chimpanzee and baboon skulls, taken from [67].

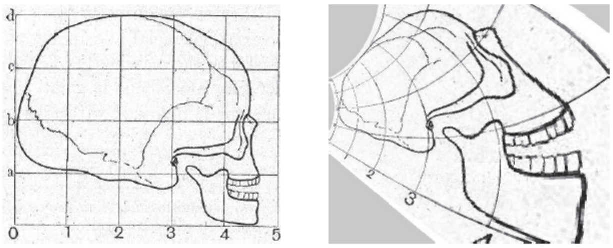


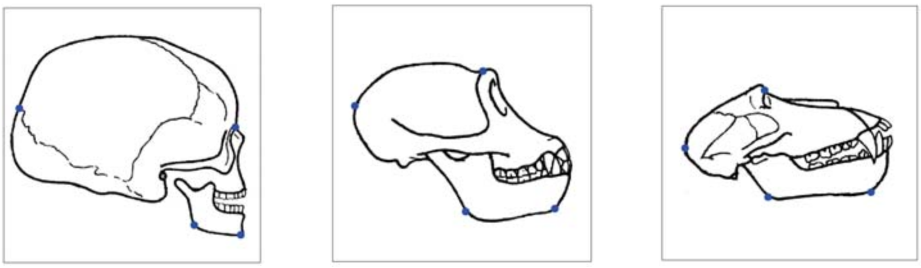
Figure 4.37: A $PSL(2,\mathbb{C})$ transformation of human skull to chimpanzee skull by Milnor [47].

4.5 The Relationship Between Human and Simian Skulls

Thompson compared a human skull with simian skulls. He explained that the main differences between the human head and simian types are the enlargements of the brain and the braincase in human, the relative diminution of his jaws, and that the facial angle increases from an oblique angle to nearly a right angle in man. He put the human skull in a grid and marked out some corresponding points on the skulls and showed what the grid of chimpanzee and baboon look like in comparison. These are shown in Figure 4.36.

In this section, we will show that these images are related by the $PSL(2,\mathbb{C})$ (Möbius) group. Milnor also attempted to transform the human skull to chimpanzee skull with $PSL(2,\mathbb{C})$ as given in Figure 4.37.

First we show why we choose this group. We mark out four corresponding points on the skulls, see Figure 4.38 and calculate the cross-ratios, see Table 4.14. As can be seen in Table 4.14, the cross-ratios are almost equal.



(a) (b) (c)

Figure 4.38: Four corresponding points are marked in skulls to calculate the cross-ratios. Cross-ratios are given in Table 4.14.

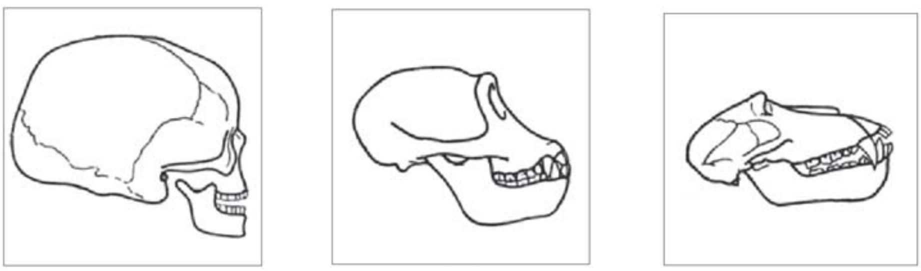
	Human	Chimpanzee	Baboon
Cross-ratio	$-0.1097 - 0.4382i$	$-0.27 - 0.4108i$	$-0.2822 - 0.6555i$

Table 4.14: Cross-ratios of four corresponding marked point in the human, chimpanzee and baboon skulls in Figure 4.38.

Let the human skull be I_0, chimpanzee I_1 and baboon I_2, see Figure 4.39. The inside of the images are filled with black for the registration. The domain of the images is $[-0.5, 0.5] \times [-0.5, 0.5]$. The output of the registration is ϕ_i^{-1}, let $\phi_i^{-1} = \varphi_i$. From the registration of the human skull and the chimpanzee skull we obtained transformation φ_1 as follows:

$$\varphi_1(z) = \frac{(1.2049 + 0.0608i)z + 0.2363 + 0.0793i}{(0.6641 - 0.7029i)z + 0.9992 - 0.1446i}.$$

As explained in Chapter 3, registration with our algorithm may get stuck at some non-removable local minimum depending on the initial value. Starting the registration of the human skull and baboon from the identity, the algorithm got stuck at a local minimum.



(a) human (b) chimpanzee (c) baboon

Figure 4.39: Human, chimpanzee and baboon skulls, taken from [67].

	$i=1$	$i=2$
$\|I_0 - I_i\|_2^2$	6330	7724
$\|I_0 \circ \varphi_i^{-1} - I_i\|_2^2$	1818	1546

Table 4.15: I_0 is human, I_1 chimpanzee and I_2 baboon skulls. First row: The residuals of human skull and chimpanzee, human and baboon before registration. Second row: The residual of human skull and chimpanzee, human and baboon after registration in $PSL(2,\mathbb{C})$.

If we suppose that there are $PSL(2,\mathbb{C})$ transformations between the images, then using the method in Section 4.2, two transformations enable us to obtain other transformations. Therefore, we register the chimpanzee skull with the baboon skull, and compose the transformations to get a good initial guess for the registration of human and baboon. Let the chimpanzee be the source and the baboon the target. The output of the registration is the following transformation:

$$\varphi_3(z) = \frac{(1.0778 + 0.2414i)z + 0.1310 - 0.0002i}{(0.5947 - 0.0367i)z + 0.9513 - 0.2176i}.$$

Then we take $\varphi_1 \circ \varphi_3$ as an initial guess for the registration of human skull and baboon skull:

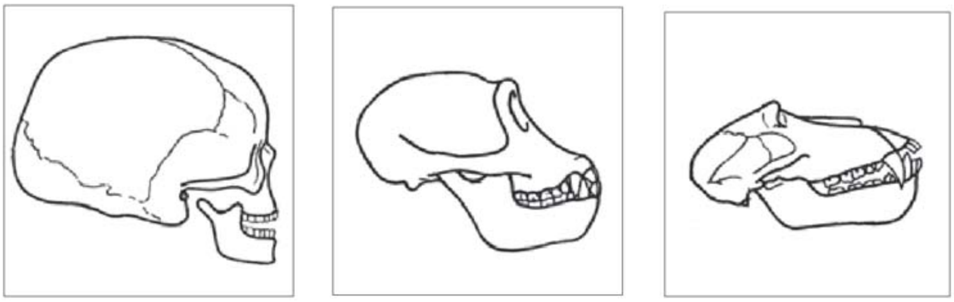
$$\varphi_1 \circ \varphi_3(z) = \frac{(1.4252 + 0.4028i)z + 0.3978 + 0.0344i}{(1.4453 - 0.7212i)z + 0.9959 - 0.4479i}$$

Starting from there, the output of registration of the human and baboon skulls is:

$$\varphi_2(z) = \frac{(1.4409 + 0.4166i)z + 0.4044 + 0.0346i}{(1.4371 - 0.8288i)z + 0.9782 - 0.4809i}.$$

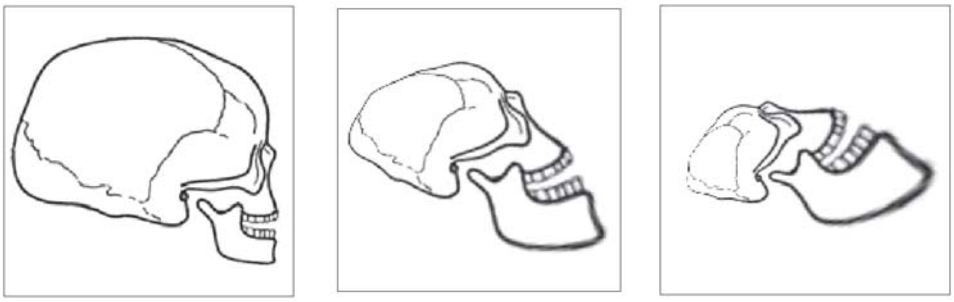
The residuals before and after deformation are given in Table 4.15. Thompson compared the two transformations between the human and the chimpanzee, and the human and the baboon; he concluded that the transformations were of the same order and they only differ in their degree of deformation. We can see this here by comparing φ_1 and φ_3. The non-linear part of φ_3 is bigger than the non-linear part of φ_1, which can be considered as the degree of deformation, which Thompson talked about. Figures 4.40 and 4.41 show skulls before registration and after registration respectively. Also, their discrepancy after registration is shown in Figure 4.42.

As can be seen in Figure 4.42, $PSL(2,\mathbb{C})$ gives a good approximation of the transformation of human skull to chimpanzee and baboon skulls. The transformations enlarge the jaw and shrink the braincase, also changing the facial angle as Thompson described. Figure 4.43 shows how the grid on the human skull is deformed by the transformations.



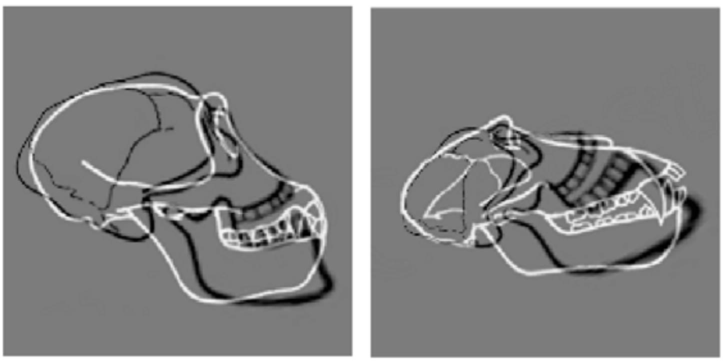
(a) Human (b) Chimpanzee (c) Baboon

Figure 4.40: Before registration.



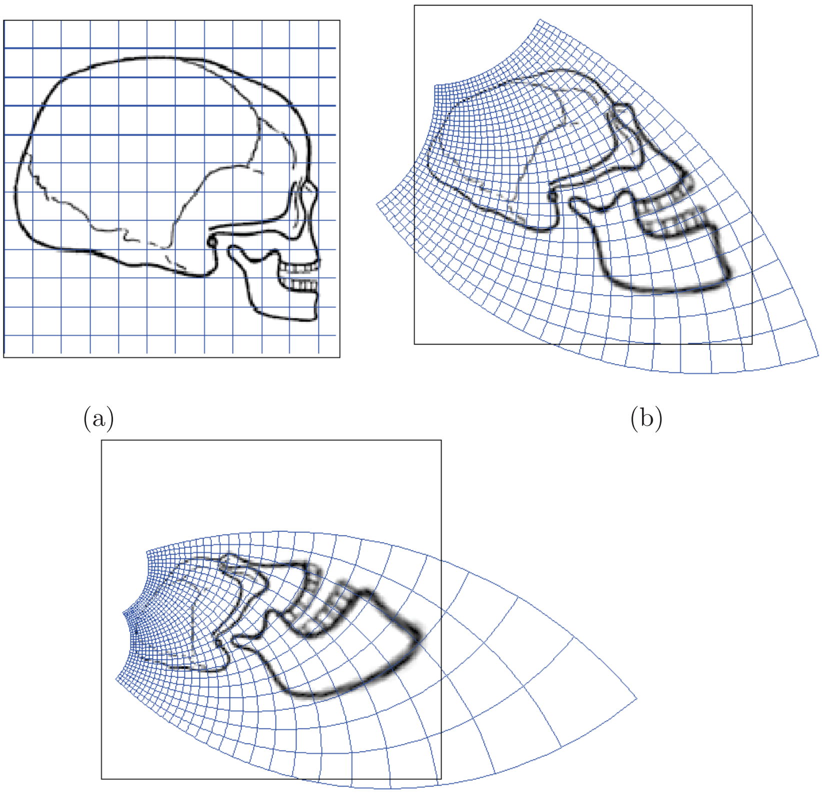
(d) Human (e) Chimpanzee (f) Baboon

Figure 4.41: After registration. (e) and (f) are the transformation of human skull to chimpanzee and baboon skulls respectively.



(a) (b)

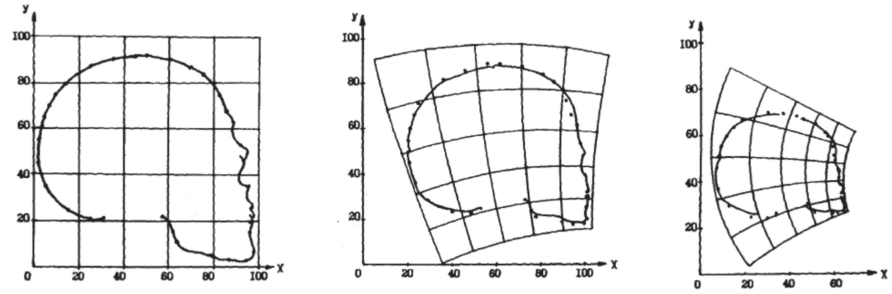
Figure 4.42: (a) Discrepancy between the transformed human and chimpanzee after registration in $PSL(2,\mathbb{C})$. (b) Discrepancy between the transformed human and baboon after registration in $PSL(2,\mathbb{C})$.



(a) (b)

(c)

Figure 4.43: (a) Human skull along with a rectangular grid. Human skull and rectangular grid after registration and transformation to (b) chimpanzee, (c) baboon.



(a) (b) (c)

Figure 4.44: $PSL(2,\mathbb{C})$ transformation of the human skull growth, (a) adult, (b) 5 years old, (c) newborn. Taken from [53].

It can be seen in Figure 4.43 that the human grid after deformation to chimpanzee and baboon are very similar to those that Thompson sketched (shown in Figures 4.36b and 4.36c).

4.6 Human Skull Growth

The study of the growth of a living body is an important subject in biology development and morphology. Petukhov [53] studied the growth of the human body mathematically. He explained that when a living body is growing the scale of the body is changing. This scale can be equal or different in each direction for each local zone in the body. In [53], he asked a question: 'Does nature employ simpler type of growth changes, for example, equal scale in each direction for each local zone in the body?'

Suppose that growth for every local region of the body is equal in each direction. However, if we look at the whole body scaling then it may not scale equally. In other words, similarity of a small area does not mean similarity of the whole. The actions of conformal transformations on the plane have this property, for this reason in [53] the author approximated the growth of the human skull with $PSL(2,\mathbb{C})$ transformations, see Figure 4.44. These are conformal, but are not the full set of conformation transformations, which is infinite dimensional.

The author did not give the transformations, nor did he explain what method was used to find them. So, in the following, we give more than three human skulls and perform an image registration between them in two groups: $PSL(2,\mathbb{C})$ and $PSL(3,\mathbb{R})$.

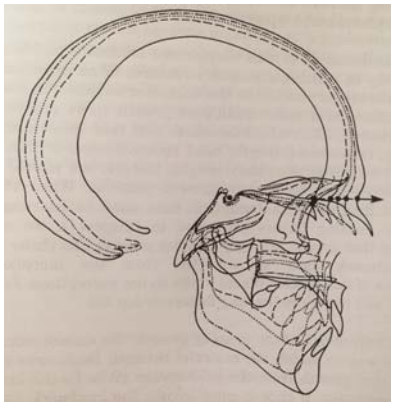


Figure 4.45: Traditional growth series based on longitudinal cephalometric radiographs taken from [37].

A picture of some human skulls over developmental time is shown in Figure 4.45. The outline of the skulls are taken for the registration, see Figure 4.46. The size of the images is 164×123, and the domain is $[-0.5, 0.5] \times [-0.5, 0.5]$. The skull I_3 is taken as source and registered with I_i, $i = 1, 2, 3, 4, 5$ in the $PSL(2, \mathbb{C})$ group and $PSL(3, \mathbb{R})$, with $\sigma = 4$ pixels. Figure 4.47 shows the output of the registration in $PSL(2, \mathbb{C})$ and $PSL(3, \mathbb{R})$, and Figure 4.48 shows the discrepancy between transformed I_3 and the targets. The algorithm performs a good registration even though the images were line drawings. As the algorithm works on images rather than curves, it can cope with missing data, and with images with poor overlaps originally.

It can be seen in Figure 4.48 that $PSL(2, \mathbb{C})$ has done a better matching of the jaw of the human skull than $PSL(3, \mathbb{R})$, and $PSL(3, \mathbb{R})$ has done a better job of matching the head.

If two curves do not have any overlap, then their L^2 distance cannot tell how close they are. For this reason, we give the registration error on the smoothed images as well in Table 4.16. Images are convolved with a Gaussian with $\sigma = 4$ pixels.

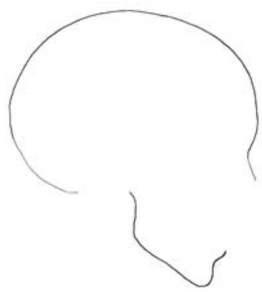
Comparing the registration error shows that $PSL(3, \mathbb{R})$ results are as good as $PSL(2, \mathbb{C})$, so human skull growth can be described by $PSL(3, \mathbb{R})$ also. Table 4.17 gives the transformations, which are the output of the registration in $PSL(2, \mathbb{C})$ and $PSL(3, \mathbb{R})$.

So far, the human skulls are registered with both groups. Here we compare the corresponding transformations φ_i and ψ_i to see if there is any relationship between them.

(a) I_1

(b) I_2

(c) I_3



(d) I_4

(e) I_5

Figure 4.46: Five human skulls.

	Before Registration		After Registration			
			$PSL(2,\mathbb{C})$		$PSL(3,\mathbb{R})$	
Target	Smooth	Non-smooth	Smooth	Non-smooth	Smooth	Non-smooth
I_1	5.8	56	1.1	20.6	1.08	20.1
I_2	1.9	50.3	0.7	17.3	0.66	14.2
I_3	0	0	0	0	0	0
I_4	2.6	54	0.57	19.8	0.58	17.4
I_5	2.7	35.4	0.85	18.3	0.87	17.2

Table 4.16: Registration error in $PSL(2,\mathbb{C})$ and $PSL(3,\mathbb{R})$ for smooth and non-smooth images. Source is I_3 and the targets are I_i, $i = 1, 2, 3, 4, 5$ given in Figure 4.46.

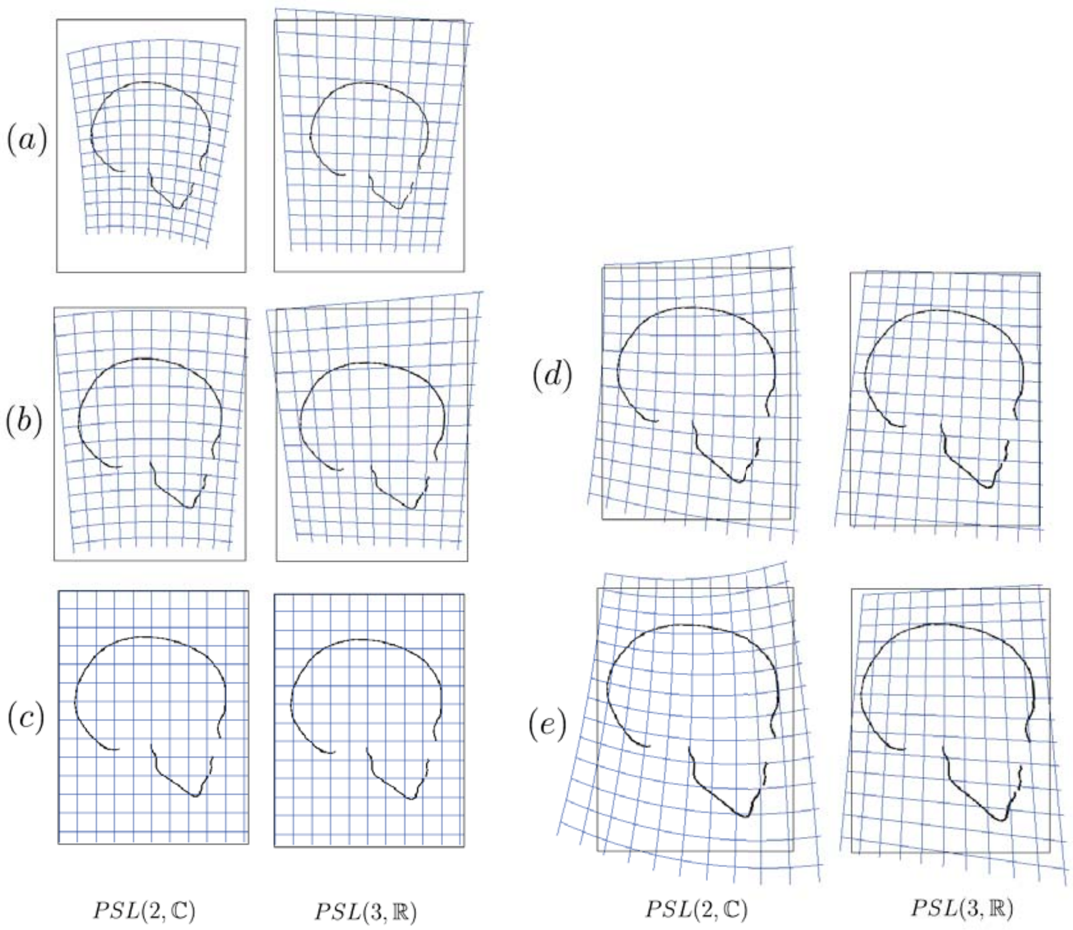


Figure 4.47: Skull I_3 is registered with I_i, $i = 1, 2, 3, 4, 5$ (given in Figure 4.46) in groups: $PSL(2, \mathbb{C})$ and $PSL(3, \mathbb{R})$. Transformed I_3 after registration with (a): I_1, (b): I_2, (c): I_3, (d): I_4, (e): I_5.

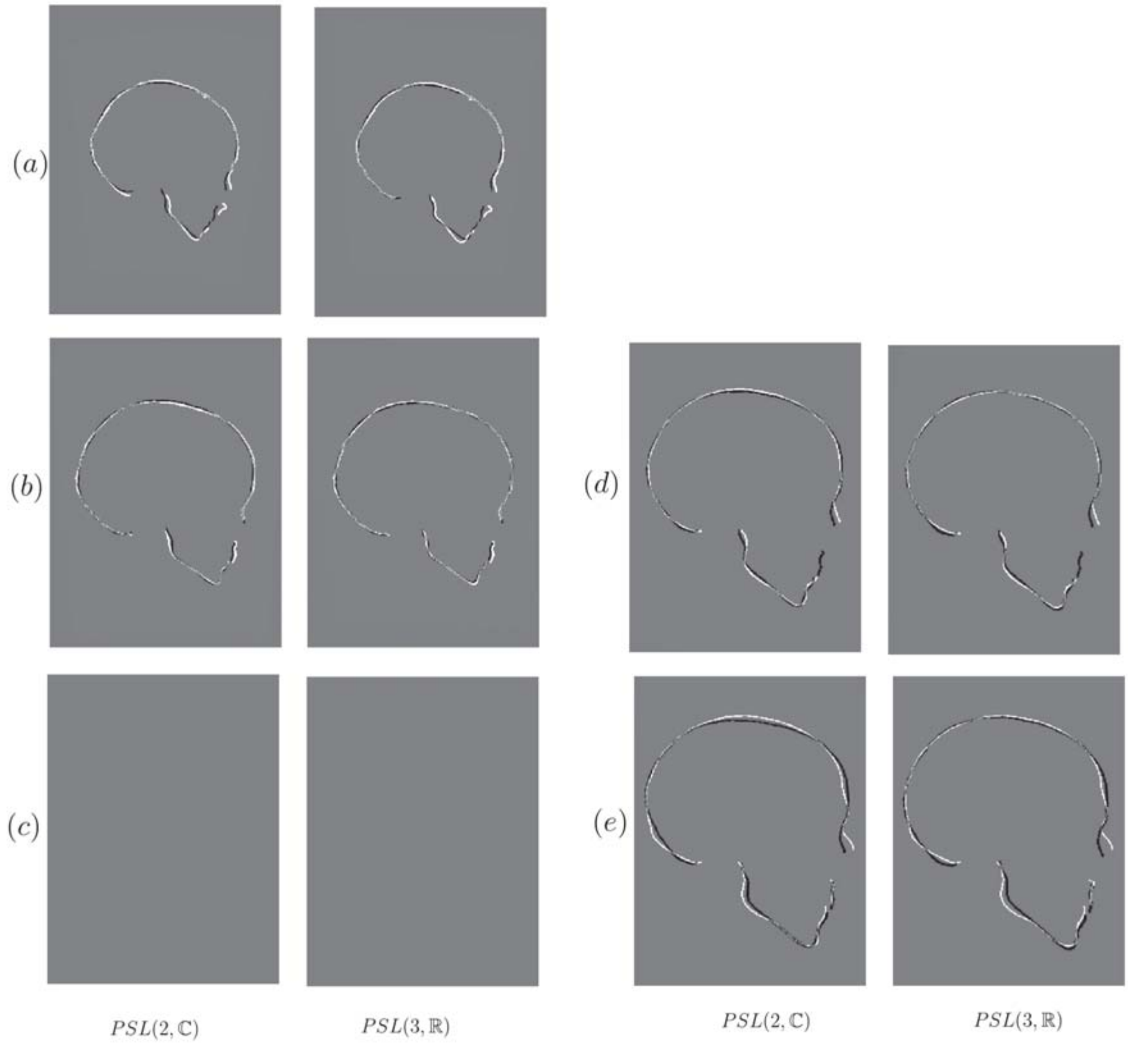


Figure 4.48: Skull I_3 is registered with I_i, $i = 1,2,3,4,5$ (given in Figure 4.46) in the groups: $PSL(2,\mathbb{C})$ and $PSL(3,\mathbb{R})$. Discrepancy between transformed I_3 after registration with (a): I_1, (b): I_2, (c): I_3, (d): I_4, (e): I_5.

Target	$PSL(2,\mathbb{C})$	$PSL(3,\mathbb{R})$
I_1	$\varphi_1(z) = \frac{(1.1402-0.0199i)z-0.0089i}{(0.0719+0.1999i)z+0.8783+0.0148i}$	$\psi_1(x,y)_1 = \frac{1.1033x+0.0621y-0.0149}{-0.0413x-0.3120y+0.8461}$ $\psi_1(x,y)_2 = \frac{-0.0305x+1.0793y-0.0244}{-0.0413x-0.3120y+0.8461}$
I_2	$\varphi_2(z) = \frac{(1.0349-0.0006i)z-0.0197-0.0097i}{(-0.01+0.1166i)z+0.9676-0.0016i}$	$\psi_2(x,y)_1 = \frac{1.0080x-0.039y-0.0052}{0.1030x-0.3020y+0.9525}$ $\psi_2(x,y)_2 = \frac{0.006x+1.0484y-0.0245}{0.103x-0.302y+0.9525}$
I_3	$\varphi_3(z) = z$	$\psi_3(x,y) = (x,y)$
I_4	$\varphi_4(z) = \frac{(0.9688-0.0183i)z+0.0026+0.0117i}{(0.104-0.0561i)z+1.0328+0.0206i}$	$\psi_4(x,y)_1 = \frac{0.9883x+0.0755y-0.0048}{0.0297x+0.1787y+1.0533}$ $\psi_4(x,y)_2 = \frac{-0.0399x+0.9609y+0.021}{0.0297x+0.1787y+1.0533}$
I_5	$\varphi_5(z) = \frac{(0.9398-0.0195i)z+0.0084+0.0187i}{(0.0797-0.1891i)z+1.0681+0.0220i}$	$\psi_5(x,y)_1 = \frac{0.9704x+0.0049y+0.0174}{0.1425x+0.2351y+1.0969}$ $\psi_5(x,y)_2 = \frac{-0.0255x+0.9478y+0.0291}{0.1425x+0.2351y+1.0969}$

Table 4.17: Output transformations of the registration of the human skulls in $PSL(2, \mathbb{C})$ and $PSL(3,\mathbb{R})$. The source is I_3 and the targets are I_i, $i = 1,2,3,4,5$ given in Figure 4.46.

For example, the numerators of $\psi_1(x, y)$ are:

$$\begin{pmatrix} 1.1033x + 0.0621y - 0.0149 \\ -0.0305x + 1.0793y - 0.0244 \end{pmatrix} \cong 1.1 \begin{pmatrix} 1 & 0.056 \\ -0.03 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} -0.0149 \\ -0.0244 \end{pmatrix} \tag{4.2}$$

And the numerator of $\varphi_1(z)$ is:

$$(1.1402 - 0.0199i)z - 0.0089i \cong 1.14 \begin{pmatrix} 1 & 0.02 \\ -0.02 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ -0.0089 \end{pmatrix} \tag{4.3}$$

Comparing Equations (4.2) and (4.3) shows that φ_1 and ψ_1 have almost equal scale. The dominator of φ_1 is:

$$(0.07 + 0.2i)z + 0.89 = 0.07x + 0.07iy + 0.2ix - 0.2y + 0.89,$$

and the dominator of ψ_1 is:

$$-0.04x - 0.3y + 0.85.$$

They have almost equal parts: $-0.2y + 0.89$ and $-0.3y + 0.85$, and different parts: $0.07x + 0.07iy + 0.2ix$, and $-0.04x$. Although φ_1 and ψ_1 are not globally equal, they are almost equal on $[-0.5, 0.5] \times [-0.5, 0.5]$. Other transformations can be compared similarly. Comparing their numerators shows that they have almost equal scaling on the plane. So here we can discuss model selection between the two groups. The goodness of fit, which can be compared by the residual in both groups, is almost the same. If simplicity is compared by the number of parameters, $PSL(2, \mathbb{C})$ is the simpler one. And the last criteria is, which group describes the human growth? We cannot answer this, because we have only five skulls, and our information is not sufficient. This can be considered as future work. However, our study has extended and partly refuted Petukhov by showing that an apparently good match by one group, namely $PSL(2, \mathbb{C})$, is not by itself definitive; a completely different group, $PSL(3, \mathbb{R})$, may also be relevant and can give equally good registration.

4.7 Conclusion

In this chapter we reproduced Thompson's work using image registration with finite dimensional groups. We could get a perfect match between images using the diffeomorphism group. However, registering in finite dimensional groups has provided a good match, and so model selection is unlikely to choose the more complex group.

Chapter 5

Curve Fitting in a Lie Group

In this chapter we introduce a method to find curves in Lie groups fitting the transformations that are obtained by registration. Such curves will give us a better insight about how a natural phenomenon (e.g. growth, evolution, disease) carries on through time.

In Section 4.6, we observed that four skulls can be generated by transformations of one skull. Now we would like to know: Can we generate all the human skulls during growth only by having these five skulls? For example, in registration of skulls in $PSL(2,\mathbb{C})$, we have found five transformations φ_i, $i = 1,2,3,4,5$. Now, suppose other skulls are generated by some transformations φ_i, $i = 6,7,8,\ldots$. This set of deformations are all elements of a particular group, and can parameterize a curve in the group. Finding the curve may help us to discover a law of growth or the law of nature that relates different organisms. In the following, we introduce a method that can be applied to find the curve passing through the given data in the group.

Let G be a Lie group and $\mathfrak{g}$ be its Lie algebra. A Lie group is diffeomorphic to its Lie algebra in a neighbourhood of its identity, and this diffeomorphism is given by the exponential map, see Section 2.1.6. We use this property of the matrix Lie group to find the curve passing through the given data. The idea is to fit a curve in the Lie algebra and then map the curve into the group using the exponential map. The method is as follows:

Let $\{\varphi_i, i = 1,2,3,\ldots\}$ be the given data in the group.

- Map the φ_i to the Lie algebra: $\log(\varphi_i) = v_i$.
- Find a parameterized curve $E : \mathbb{R} \mapsto \mathfrak{g}$ passing near the v_i:

$$\min_{E \in \mathcal{E}, t_i} \Sigma_i \|E(t_i) - v_i\|^2, \tag{5.1}$$

  where $\mathcal{E}$ is some set of curves, $\|\cdot\|$ is some metric on the Lie algebra, and t the parameters of E.

- Map the curve $E(t)$ into the group using the exponential map: $\exp(E(t))$.

There are a variety of choices for the norm $\|\cdot\|$ on the Lie algebra. One option for a matrix group is as follows [28]:

$$\|v\|_{\mathcal{M}} = tr(\mathcal{M}vv^*)^{\frac{1}{2}},$$

where $\mathcal{M}$ is a positive definite matrix, v^* is the conjugate transpose, and tr is the trace of a matrix.

5.1 Curve of the Human Skull Growth

In Section 4.6, five transformations φ_i, ψ_i, $i = 1, 2, 3, 4, 5$ were obtained by registration in $PSL(2, \mathbb{C})$ and $PSL(3, \mathbb{R})$ respectively. The groups $PSL(2, \mathbb{C})$ and $SL(2, \mathbb{C})$, and $PSL(3, \mathbb{R})$ and $SL(3, \mathbb{R})$, are homeomorphic. So, if $\phi(z) = \frac{az+b}{cz+d}$ is a transformation in $PSL(2, \mathbb{C})$, then the matrix corresponding to it in $SL(2, \mathbb{C})$ is $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$. Therefore, the matrices in $SL(2,\mathbb{C})$ corresponding to φ_i are as follows:

$$\varphi_1 = \begin{pmatrix} 1.1402 - 0.0199i & -0.0089i \\ 0.0719 + 0.1999i & 0.8783 + 0.0148i \end{pmatrix},$$

$$\varphi_2 = \begin{pmatrix} 1.0349 - 0.0006i & -0.0197 - 0.0097i \\ -0.01 + 0.1166i & 0.9676 - 0.0016i \end{pmatrix},$$

$$\varphi_3 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

$$\varphi_4 = \begin{pmatrix} 0.9688 - 0.0183i & 0.0026 + 0.0117i \\ 0.104 - 0.0561i & 1.0328 + 0.0206i \end{pmatrix},$$

$$\varphi_5 = \begin{pmatrix} 0.9398 - 0.0195i & 0.0084 + 0.0187i \\ 0.0797 - 0.1891i & 1.0681 + 0.0220i \end{pmatrix}.$$

The matrices corresponding to ψ_i in $SL(3,\mathbb{R})$ are:

$$\psi_1 = \begin{pmatrix} 1.1033 & 0.0621 & -0.0149 \\ -0.0305 & 1.0793 & -0.0244 \\ -0.0413 & -0.312 & 0.8461 \end{pmatrix},$$

$$\psi_2 = \begin{pmatrix} 1.008 & -0.039 & -0.0052 \\ 0.006 & 1.0484 & -0.0245 \\ 0.103 & -0.302 & 0.9525 \end{pmatrix},$$

$$\psi_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$\psi_4 = \begin{pmatrix} 0.9883 & 0.0755 & -0.0048 \\ -0.0399 & 0.9609 & 0.021 \\ 0.0297 & 0.1787 & 1.0533 \end{pmatrix},$$

$$\psi_5 = \begin{pmatrix} 0.9704 & 0.0049 & 0.0174 \\ -0.0255 & 0.9478 & 0.0291 \\ 0.1425 & 0.2351 & 1.0969 \end{pmatrix}.$$

The Lie algebra elements corresponding to φ_i are as follows:

$$\zeta_1 = \begin{pmatrix} 0.1305 - 0.0172i & -0.0089i \\ 0.0715 + 0.1993i & -0.1306 + 0.0172i \end{pmatrix},$$
$$\zeta_2 = \begin{pmatrix} 0.0337 + 0.0005i & -0.0197 - 0.0097i \\ -0.0100 + 0.1165i & -0.0336 - 0.0005i \end{pmatrix},$$
$$\zeta_3 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix},$$
$$\zeta_4 = \begin{pmatrix} -0.0320 - 0.0194i & 0.0026 + 0.0117i \\ 0.1040 - 0.0561i & 0.0320 + 0.0194i \end{pmatrix},$$
$$\zeta_5 = \begin{pmatrix} -0.0641 - 0.0207i & 0.0084 + 0.0187i \\ 0.0795 - 0.1889i & 0.0641 + 0.0207i \end{pmatrix}.$$

And to ψ_i are:

$$\xi_1 = \begin{pmatrix} 0.0988 & 0.0547 & -0.0147 \\ -0.0285 & 0.0732 & -0.0258 \\ -0.0475 & -0.3252 & -0.1720 \end{pmatrix},$$
$$\xi_2 = \begin{pmatrix} 0.0084 & -0.0388 & -0.0058 \\ 0.0071 & 0.0438 & -0.0245 \\ 0.1062 & -0.3008 & -0.0522 \end{pmatrix},$$
$$\xi_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$
$$\xi_4 = \begin{pmatrix} -0.0101 & 0.0779 & -0.0055 \\ -0.0413 & -0.0402 & 0.0208 \\ 0.0327 & 0.1765 & 0.0502 \end{pmatrix},$$
$$\xi_5 = \begin{pmatrix} -0.0312 & 0.0031 & 0.0168 \\ -0.0287 & -0.0570 & 0.0288 \\ 0.1414 & 0.2307 & 0.0882 \end{pmatrix}.$$

We consider a standard basis for the Lie algebras, $\mathfrak{sl}(2, \mathbb{C})$ and $\mathfrak{sl}(3, \mathbb{R})$ as follows. The $\mathfrak{sl}(2, \mathbb{C})$ basis is:

$$v_1 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, v_2 = \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix}, v_3 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, v_4 = \begin{pmatrix} 0 & i \\ 0 & 0 \end{pmatrix}, v_5 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, v_6 = \begin{pmatrix} 0 & 0 \\ i & 0 \end{pmatrix}.$$

	φ_1	φ_2	φ_3	φ_4	φ_5
Scale	0.1847	0.0480	0	0.0447	0.0906
Rotation	0.0243	0	0	0.0274	0.0293
Translation	0.0089	0.022	0	0.012	0.0205
Non-linear	0.2119	0.117	0	0.1183	0.2049

Table 5.1: The norm of scale, rotation, translation and non-linear component in φ_i, $i = 1, 2, 3, 4, 5$; φ_i are the transformations between the human skulls, which belong to the $PSL(2, \mathbb{C})$ group.

The $\mathfrak{sl}(3, \mathbb{R})$ basis is:

$$w_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, w_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix}, w_3 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, w_4 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$
$$w_5 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}, w_6 = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, w_7 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, w_8 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

We calculate the norm of sub-transformations: scale, rotation, translation and non-linear transformation ($\frac{z}{cz+1}$) in φ_i by calculating the norm of their corresponding vectors in the Lie algebra. For example, let $\zeta_i = a_1v_1+a_2v_2+a_3v_3+a_4v_4+a_5v_5+a_6v_6$. Then the norm of scale is $tr((a_1v_1)(a_1v_1)^*)^{\frac{1}{2}}$. Table 5.1 gives these values. As can be seen from Table 5.1, the norms of rotation and translation are very close to zero. The 'shape' of an object (in the sense of shape space [35]) is invariant under rotation and translation. Therefore, we ignore the rotation and translation parts of ζ_i. This yields:

$$\zeta_1 = \begin{pmatrix} 0.1305 & 0 \\ 0.0715 + 0.1993i & -0.1306 \end{pmatrix}, \zeta_2 = \begin{pmatrix} 0.0337 & 0 \\ -0.0100 + 0.1165i & -0.0336 \end{pmatrix},$$
$$\zeta_3 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix},$$
$$\zeta_4 = \begin{pmatrix} -0.0320 & 0 \\ 0.1040 - 0.0561i & 0.0320 \end{pmatrix}, \zeta_5 = \begin{pmatrix} -0.0641 & 0 \\ 0.0795 - 0.1889i & 0.0641 \end{pmatrix}.$$

Similarly, we calculate the norm of the sub-transformation of translation in $\mathfrak{sl}(3, \mathbb{R})$ and we find that translations also have very small norm, and also under the translation the shape of an object is invariant. Therefore, we ignore the translations in ψ_i, and ξ_i are considered as follows:

$$\xi_1 = \begin{pmatrix} 0.0988 & 0.0547 & 0 \\ -0.0285 & 0.0732 & 0 \\ -0.0475 & -0.3252 & -0.1720 \end{pmatrix}, \xi_2 = \begin{pmatrix} 0.0084 & -0.0388 & 0 \\ 0.0071 & 0.0438 & 0 \\ 0.1062 & -0.3008 & -0.0522 \end{pmatrix},$$

$$\xi_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$\xi_4 = \begin{pmatrix} -0.0101 & 0.0779 & 0 \\ -0.0413 & -0.0402 & 0 \\ 0.0327 & 0.1765 & 0.0502 \end{pmatrix}, \xi_5 = \begin{pmatrix} -0.0312 & 0.0031 & 0 \\ -0.0287 & -0.0570 & 0 \\ 0.1414 & 0.2307 & 0.0882 \end{pmatrix}.$$

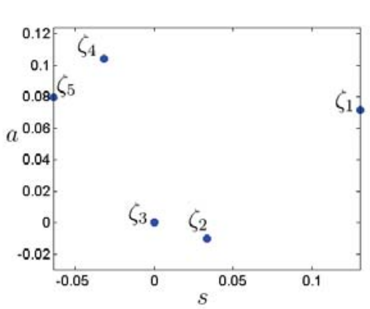
This projection is equivalent to using a metric on the Lie algebra which is zero in the similarity components.

Now, we need to choose a model to fit through the data in the Lie algebra. Figure 5.1 shows a two dimensional representation of the data $\zeta_k \in \mathfrak{sl}(2, \mathbb{C})$ in $\mathbb{R}^2$. Let $\zeta_k = \begin{pmatrix} s_k & 0 \\ a_k + ib_k & -s_k \end{pmatrix}$. Figure 5.1a shows the data (s_k, a_k), Figure 5.1b shows (s_k, b_k), and Figure 5.1c shows (a_k, b_k), $k = 1, 2, 3, 4, 5$.

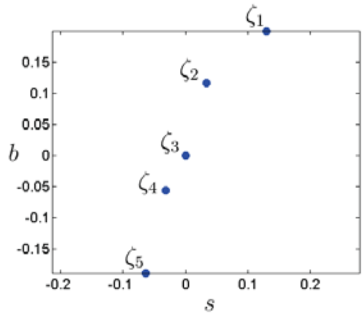
As can be seen in Figure 5.1 there is an approximately linear representation between the data (s_k, b_k). A polynomial of degree four can be fitted perfectly through the data. But, by model selection criteria, we should choose a simple model. Therefore, in the following we fit the simplest model, which is a straight line. We will also fit a line that passes through the origin, because the lines which pass through the origin correspond under the exponential map to one-parameter subgroups of the group. One-parameter subgroups are the simplest model (group) that can describe the relationship between the human skulls of different ages.

Therefore, we fit two lines fitted through the data $\{\zeta_i\}$, $\zeta_i \in \mathfrak{sl}(2, \mathbb{C})$.

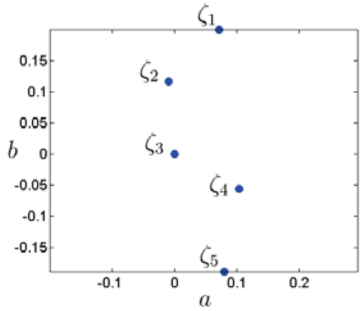
Line passing through the origin: A parameterized line $\Gamma_1(t) = tA$ is obtained by



(a)



(b)



(c)

Figure 5.1: Two-dimensional projection of ζ_k, where $\zeta_k = \log(\varphi_k) = \begin{pmatrix} s_k & 0 \\ a_k + ib_k & -s_k \end{pmatrix} \in \mathfrak{sl}(2,\mathbb{C})$, and φ_k are the outputs of the registration of the human skulls. I_3 is the source and I_k, $k = 1,2,3,4,5$ the targets given in Figure 4.46 in $PSL(2,\mathbb{C})$.

minimisation of

$$\sum_i \|\zeta_i - t_i A\|^2 = \sum_i tr((\zeta_i - t_i A)(\zeta_i - t_i A)^*)),$$

where A is found to be

$$\begin{pmatrix} -0.4406 & 0 \\ 0.0603 - 0.8869i & 0.4406 \end{pmatrix}.$$

The residual is 0.0254, and t_i corresponding to each ζ_i is: $t_1 = -0.2440$, $t_2 = -0.1134$, $t_3 = 0$, $t_4 = 0.0715$, $t_5 = 0.1941$.

Line not passing through origin: A parameterized line $\Gamma_2(t) = tA + B$ is obtained by minimisation of

$$\sum_i \|\zeta_i - (t_i A + B)\|^2 = \sum_i tr((\zeta_i - (t_i A + B))(\zeta_i - (t_i A + B))^*)),$$

where A and B are found to be

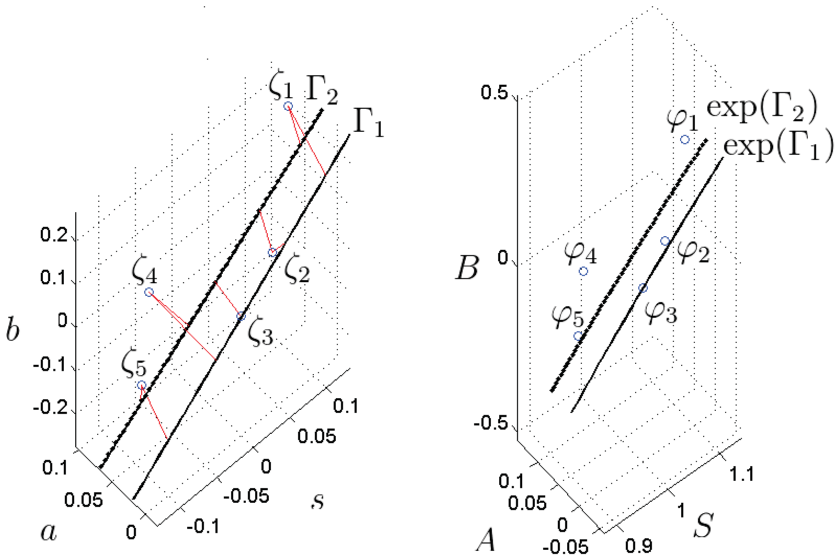
$$A = \begin{pmatrix} -0.4342 & 0 \\ 0.0772 - 0.8898i & 0.4342 \end{pmatrix}, B = \begin{pmatrix} 0.0023 & 0 \\ 0.0471 - 0.0086i & -0.0023 \end{pmatrix}.$$

The residual is 0.0123, and t_i corresponding to each ζ_i is: $t_1 = -0.2507$, $t_2 = -0.1217$, $t_3 = -0.0079$, $t_4 = 0.0651$, $t_5 = 0.1877$.

The exponentials $\exp(\Gamma_1)$ and $\exp(\Gamma_2)$ are the curves fitting through the points representing the skulls. Let $\zeta_k = \begin{pmatrix} s_k & 0 \\ a_k + ib_k & -s_k \end{pmatrix}$, $\varphi_k = \begin{pmatrix} S_k & 0 \\ A_k + iB_k & \frac{1}{S_k} \end{pmatrix}$ be the general form of ζ_k and φ_k, $k = 1, 2, 3, 4, 5$ respectively. Figure 5.2a show the data (s_k, a_k, b_k) and the fitted lines Γ_1 and Γ_2, and Figure 5.2b show the data (S_k, A_k, B_k) and the curves $\exp(\Gamma_1), \exp(\Gamma_2)$ respectively.

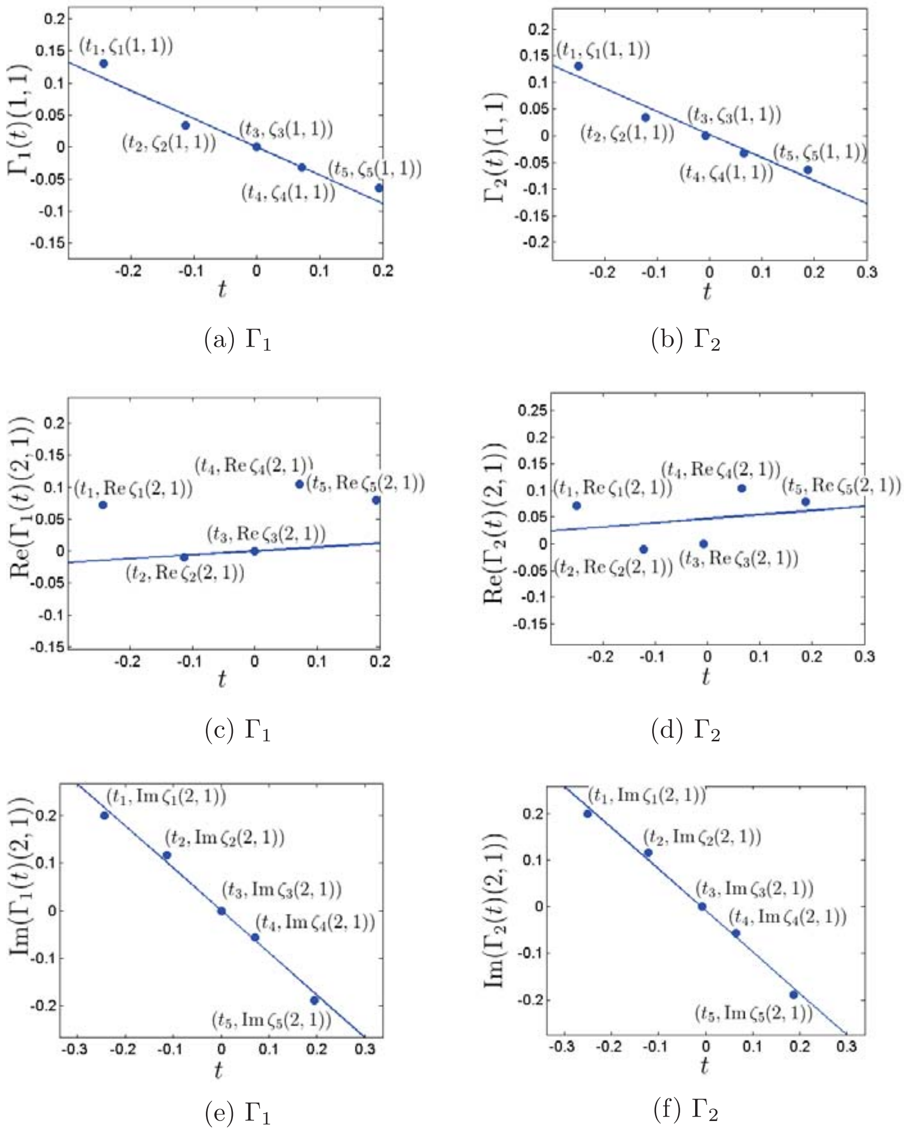
Figure 5.3 shows plots of each of the three parameters of the lines $\Gamma_1(t)(k, h)$ and $\Gamma_2(t)(k, h)$ against t, along with $(t_i, \zeta_i(k, h))$ where (k, h) is the kh array of Γ_j, $j = 1, 2$, and kh array in ζ_i, $i = 1, 2, 3, 4, 5$ respectively.

Similarly, two lines are fitted through the data ξ_i, $i = 1, 2, 3, 4, 5$.



(a) Fitted Lines in the Lie algebra (b) Exponential of lines in the group

Figure 5.2: $\varphi_k = \begin{pmatrix} S_k & 0 \\ A_k + iB_k & \frac{1}{S_k} \end{pmatrix}$, where φ_k are the outputs of the registration of human skull, I_3 is the source and I_k, $k = 1, 2, 3, 4, 5$ the targets, given in Figure 4.46 in $PSL(2, \mathbb{C})$. $\zeta_k = \log(\varphi_k) = \begin{pmatrix} s_k & 0 \\ a_k + ib_k & -s_k \end{pmatrix} \in \mathfrak{sl}(2, \mathbb{C})$. (a): Γ_1 is a line passing through the origin fitted through ζ_k and Γ_2 a line that does not pass through the origin. They map by the exponential to the group (b).



(a) Γ_1 (b) Γ_2

(c) Γ_1 (d) Γ_2

(e) Γ_1 (f) Γ_2

Figure 5.3: Plots of each of the three parameters of lines $\Gamma_i(t)$ against t, $i = 1, 2$. Γ_1 is a line in $\mathfrak{sl}(2, \mathbb{C})$ that passes through the origin and Γ_2 is a line that does not pass through the origin. They are fitted by least squares. The data are the elements $\zeta_k = \log(\varphi_k)$, where φ_k are the transformations between the human skulls in $PSL(2, \mathbb{C})$.

Line passing through the origin: A parameterized line Σ_1 is obtained by minimisation of

$$\sum_i \|\xi_i - t_i A\|^2 = \sum_i tr((\xi_i - t_i A)(\xi_i - t_i A)^*)),$$

where A is found to be

$$\begin{pmatrix} 0.0162 & -0.0019 & 0 \\ 0.0021 & 0.0204 & 0 \\ -0.01 & -0.0983 & -0.0366 \end{pmatrix}.$$

The residual is 0.0552, and t_i corresponding to each ξ_i is: $t_1 = 3.5346$, $t_2 = 2.6745$, $t_3 = 0$, $t_4 = -1.7583$, $t_5 = -2.4647$.

Line not passing through the origin: A parameterized line Σ_2, is obtained by minimisation of

$$\sum_i \|\xi_i - (t_i A + B)\|^2 = tr((\xi_i - (t_i A + B))(\xi_i - (t_i A + B))^*)),$$

where A and B are found to be

$$A = \begin{pmatrix} 0.0089 & -0.0019 & 0 \\ 0.002 & 0.0117 & 0 \\ -0.0078 & -0.0555 & -0.0206 \end{pmatrix}, B = \begin{pmatrix} 0.0043 & 0.0213 & 0 \\ -0.0203 & -0.0077 & 0 \\ 0.0544 & 0.0117 & 0.0034 \end{pmatrix}.$$

The residual is 0.0376, and t_i corresponding to each ξ_i is: $t_1 = 6.5489$, $t_2 = 4.9852$, $t_3 = 0.3369$, $t_4 = -2.7967$, $t_5 = -4.0790$.

Figures 5.4 and 5.5 show plots of each of the six parameters of the lines $\Sigma_1(t)$ and $\Sigma_2(t)$ against t, i.e. $\Sigma_i(k,h)$, $i = 1, 2$, along with $(t_i, \xi_i(k,h))$ where (k,h) is the kh array of Σ_i, $i = 1, 2$, and kh array in ξ_i, $i = 1, 2, 3, 4, 5$, respectively.

It is not always easy to see visually how well the linear model fits the data, particularly as the scales of the axes of the graphs given in Figures 5.4 and 5.5 are not equal. We therefore calculated the *coefficient of determination* (R^2) and the *adjusted coefficient of determination* ($\bar{R}^2$) [54], which are measures that determine how well a fitted model represents the data. The coefficient of determination is calculated as follows:

$$R^2 = 1 - \frac{\sum_i e_i^2}{\sum_i (Y_i - \bar{Y})^2},$$

where $\sum_i e_i^2$ are the sum of the square of the errors after fitting a model on the data Y_i,

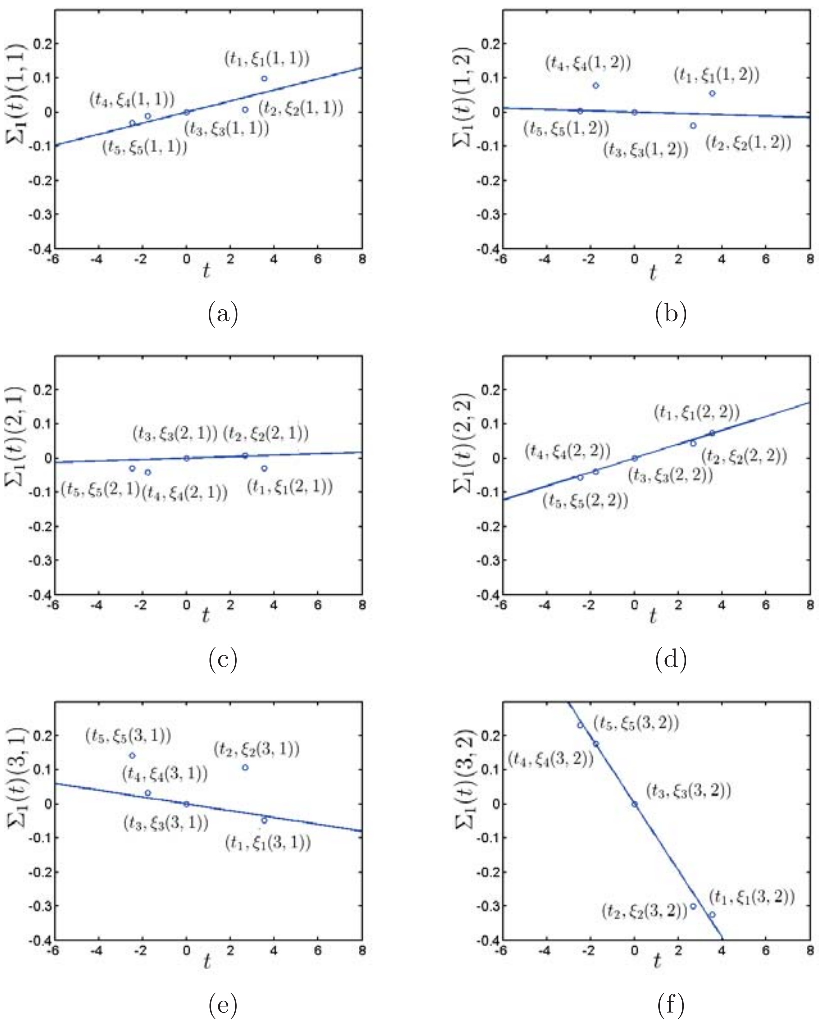


Figure 5.4: Plots of each of the six parameters of line $\Sigma_1(t)$ against t. Σ_1 is a line in $\mathfrak{sl}(3,\mathbb{R})$ that passes through the origin fitted by least squares. The data are the elements $\xi_k = \log(\psi_k)$, where ψ_k are the transformations between the human skulls in $PSL(3,\mathbb{R})$.

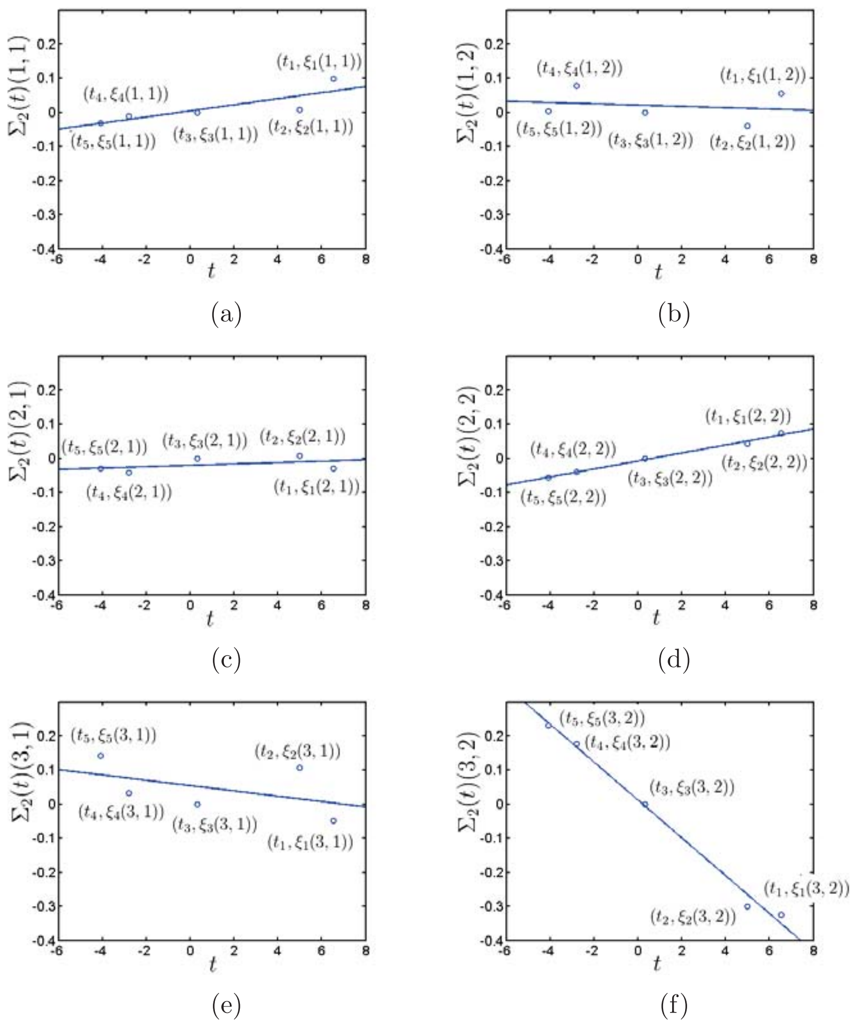


Figure 5.5: Plots of each of the six parameters of line $\Sigma_2(t)$ against t. Σ_2 is a line in $\mathfrak{sl}(3,\mathbb{R})$ that does not pass through the origin fitted by least squares. The data are the elements $\xi_k = \log(\psi_k)$, where ψ_k are the transformations between the human skulls in $PSL(3,\mathbb{R})$.

$\mathfrak{sl}(2,\mathbb{C})$	R^2	$\bar{R}^2$	$\mathfrak{sl}(3,\mathbb{R})$	R^2	$\bar{R}^2$
Γ_1	0.79	0.73	Σ_1	0.85	0.80
Γ_2	0.90	0.80	Σ_2	0.90	0.80

Table 5.2: The coefficient of determination R^2, and adjusted coefficient of determination $\bar{R}^2$ of the lines Γ_i and Σ_i, $i = 1, 2$, where the lines are fitted through the data in $\mathfrak{sl}(2,\mathbb{C})$ and $\mathfrak{sl}(3,\mathbb{R})$ respectively. The data in $\mathfrak{sl}(2,\mathbb{C})$ are the elements $\zeta_k = \log(\varphi_k)$, where φ_k are the transformations between the human skulls in $PSL(2,\mathbb{C})$, and the data in $\mathfrak{sl}(3,\mathbb{R})$ are the elements $\xi_k = \log(\psi_k)$, where ψ_k are the transformations between the human skulls in $PSL(3,\mathbb{R})$. Γ_1 and Σ_1 are the lines that pass through the origin, and Γ_2 and Σ_2 are the lines that do not pass through the origin.

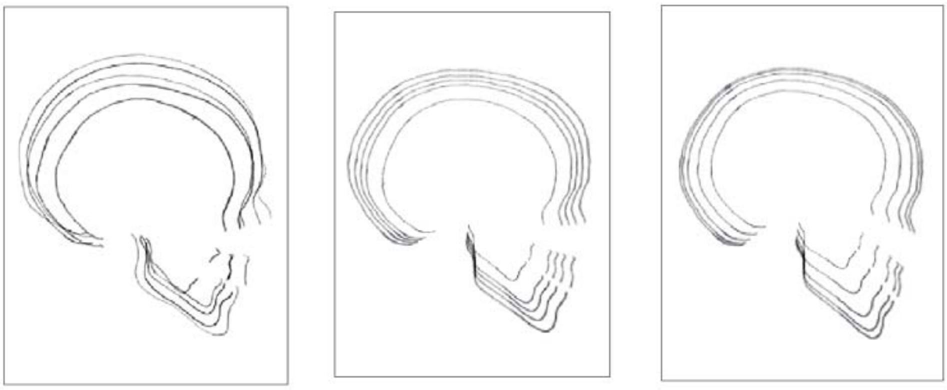
and $\bar{Y}$ is the mean of the data. R^2 is a number between zero and one, with higher values showing better fits. Models improve as we increase the degree of the model (since they include more independent variables), and so the residual will be smaller, and therefore R^2 is increasing. To deal with this we use the adjusted coefficient of determination ($\bar{R}^2$), which is calculated as follows:

$$\bar{R}^2 = 1 - \frac{(1 - R^2)(n - 1)}{n - k - 1},$$

where n is the number of datapoints, k is the degree of the model, and R^2 is the coefficient of determination of the model. This penalises models with more degrees of freedom that do not make large effects on the residuals.

So, in the following we calculate the R^2 and $\bar{R}^2$ for the fitted lines in $\mathfrak{sl}(2,\mathbb{C})$ and $\mathfrak{sl}(3,\mathbb{R})$. For the line that does not pass through the origin $k = 2$ (two independent variables, the slope of the line and the constant) and for the line that passes through the origin $k = 1$ (one independent variable, which is the slope of the line). Note that the origin is included as a datapoint in both cases. Table 5.2 gives the values of R^2 and $\bar{R}^2$ of the fitted lines in $\mathfrak{sl}(2,\mathbb{C})$ and $\mathfrak{sl}(3,\mathbb{R})$. As can be seen in Table 5.2, the value of $\bar{R}^2$ for the three models is 0.8, which suggests that the fitted lines represent the data well, although there are very few datapoints.

So far, we have showed that the linear models represent the data well. Therefore, we generate some possible human skulls by interpolation using both $\exp(\Gamma_2(t))$ and $\exp(\Sigma_2(t))$, where a new transformation is generated by changing the t, then the transformation is applied to the source (I_3) and a new skull is generated, see Figures 5.6b and 5.6c respectively.



(a) (b) (c)

Figure 5.6: (a) The original skulls, inside to outside: I_1, I_2, I_3, I_4, I_5. (b) Generating skulls using the curve $\exp(\Gamma_2)$ in $PSL(2,\mathbb{C})$. (c) Generating skulls using the curve $\exp(\Sigma_2)$ in $PSL(3,\mathbb{R})$.

5.2 Prediction of the Models in Human Skull Growth

In the previous section we obtained curves fitted through the transformations in the groups $PSL(2,\mathbb{C})$ and $PSL(3,\mathbb{R})$. We chose the line as a model to fit because it is the simplest and also fits well (the residuals are small). It is obvious that if we increase the degree of the model, i.e. fit a polynomial of a higher degree then the residual is smaller and so the fit is better. But the prediction of model may get worse as the dimension of the model is higher. In this section we compare four models in $\mathfrak{sl}(2,\mathbb{C})$,

We fitted two models in the previous section. Now we fit another two more models through the data $\zeta_i \in \mathfrak{sl}(2,\mathbb{C})$:

Quadratic: The curve Γ_3 is:

$$\begin{pmatrix} 0.3518 & 0 \\ -0.3496+1.0548i & -0.3518 \end{pmatrix} + t \begin{pmatrix} 0.4427 & 0 \\ -0.4279+1.2600i & -0.4427 \end{pmatrix} + \\ t^2 \begin{pmatrix} -0.5148 & 0 \\ 0.6382-1.6183i & 0.5148 \end{pmatrix},$$

where t_i corresponding to ζ_i are: $t_1 = 1.2140$, $t_2 = -0.4661$, $t_3 = -0.5032$, $t_4 = -0.5323$, $t_5 = -0.5674$, and the residual is 0.0037.

Cubic: The curve Γ_4 is:

$$\begin{pmatrix} -0.0132 & 0 \\ 0.0134+0.0032i & 0.0132 \end{pmatrix} + t \begin{pmatrix} 0.2040 & 0 \\ -0.5302+0.5689i & -0.2040 \end{pmatrix} +$$
$$t^2 \begin{pmatrix} 0.7084 & 0 \\ 1.9198+0.2628i & -0.7084 \end{pmatrix} + t^3 \begin{pmatrix} 0.3782 & 0 \\ 1.4040-0.0123i & -0.3782 \end{pmatrix},$$

where t_i corresponding to ζ_i are: $t_1 = 0.3078$, $t_2 = 0.1672$, $t_3 = 0.0158$, $t_4 = -0.1194$, $t_5 = -1.5867$, and the residual is zero.

The following table gives the residuals of the four models. As can be seen, as the degree of the model increases their residual decreases.

	Cubic	Quadratic	Line(non-zero intercept)	Line (zero intercept)
Residuals	0	0.0037	0.0123	0.0254

We generate the skulls corresponding to ζ_i by $\exp(\Gamma_j(t_i))$, $i = 1,2,3,4,5$, $j = 1,2,3,4$, see Figure 5.7.

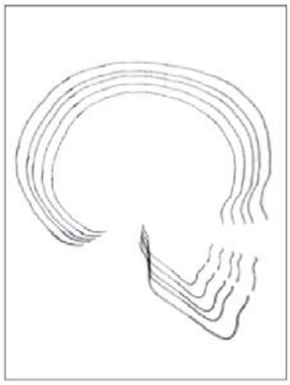
Now we generate more skulls in between the generated skulls given in Figure 5.7, see Figure 5.8.

As can be seen some strange skulls are generated by the cubic and quadratic models. Therefore, it seems that cubic and quadratic are not good models to describe human skull growth. We also use a statistical method called *Leave-One-Out* [3] to check the reliability of the models in prediction of new data. The steps of the leave-one-out method are as follows: Let ϕ_i, $i = 1,2,\ldots n$ be data.

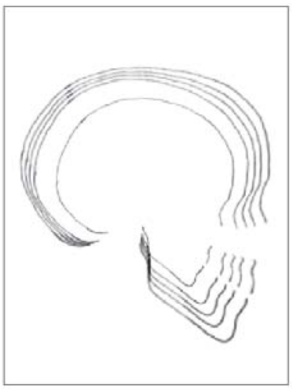
1. Leave the data ϕ_i out, and fit a model to the remaining data: ϕ_k, $k \neq i$, (for example fit a line L_k to ϕ_k, $k \neq i$).
2. Calculate the error e_i, (For example the distance of ϕ_i to L_k).
3. Repeat the first and second steps for $i = 1,2,\ldots,n$.
4. Calculate the mean of the square of the errors: MSE.

Smaller values of MSE means the model is more accurate in its predictions. We compute MSE for the models, where $e_i = \min_{t_i}(tr(\zeta_i - (\Gamma_j(t_k)))(\zeta_i - (\Gamma_j(t_k)))^*)^{\frac{1}{2}}$, $k \neq i$; ζ_i is



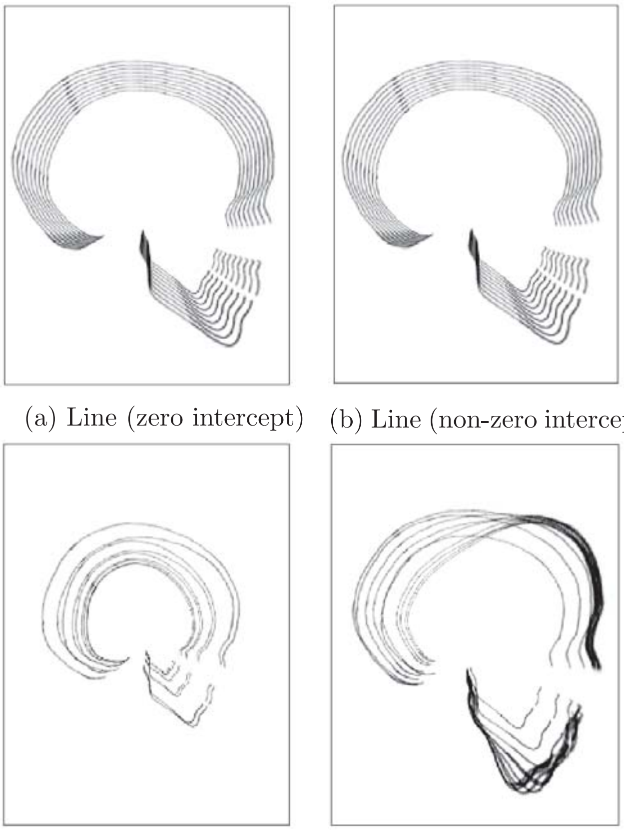


(a) Real skulls (b) Line (zero intercept) (c) Line (non-zero intercept)




(d) Quadratic (e) Cubic

Figure 5.7: (a) The real skulls, inside to outside: I_1, I_2, I_3, I_4, I_5. Generating skulls using the curve (b) $\exp(\Gamma_1)$, where Γ_1 is a line with zero intercept, (c) $\exp(\Gamma_2)$, where Γ_2 is a line with non-zero intercept, (d) $\exp(\Gamma_3)$, where Γ_3 is a quadratic curve, (e) $\exp(\Gamma_4)$, where Γ_4 is a cubic curve.



(a) Line (zero intercept) (b) Line (non-zero intercept)

(c) Quadratic (d) Cubic

Figure 5.8: Skull generating by the curve in $PSL(2, \mathbb{C})$: (a) $\exp(\Gamma_1)$, where Γ_1 is a line with zero intercept, (b) $\exp(\Gamma_2)$, where Γ_2 is a line with non-zero intercept, (c) $\exp(\Gamma_3)$, where Γ_3 is a quadratic curve, (d) $\exp(\Gamma_4)$, where Γ_4 is a cubic curve.

	e_1	e_2	e_3	e_4	e_5	MSE= $\frac{\sum_{i=1}^{5} e_i^2}{5}$
Cubic	0.1137	0.0483	0.0666	0.0738	0.1198	0.0079
Quadratic	0.1758	0.1537	0.1702	0.0743	0.1335	0.0214
Line non-zero intercept	0.1503	0.0843	0.0645	0.0661	0.0663	0.0085
Line zero intercept	0.1859	0.0313	0	0.105	0.1174	0.0119

Table 5.3: Leave-one-out method is employed for the data ζ_k, $k = 1,2,3,4,5$, (ζ_k are the Lie algebra element corresponding to the transformation between the human skull I_3 and I_k, $k = 1,2,3,4,5$ in $PSL(2,\mathbb{C})$) where a line passing through the origin is fitted through the data except ζ_i; e_i are the distance of data ζ_i from the fitted line. MSE is the mean of the square of e_i, $i = 1,2,3,4,5$.

omitted from the data $\{\zeta_k\}$, $k = 1,2,3,4,5$. Mean of the square of the errors (e_i) or MSE is given in the Table 5.3.

As can be seen from the Table 5.3, MSE of the quadratic model is the largest, which means it is the worse model to predict the growth. The two models line with non-zero intercept and cubic have almost equal MSE, but we observed in Figure 5.8 that the cubic model generates some strange skulls. Therefore, the cubic model also is not reliable. Between the two lines, MSE of the line with zero intercept is close to the MSE of line with non-zero intercept, so both models are reliable for prediction. We may prefer the line with zero intercept or one-parameter subgroup, but we need more data to confirm this; we leave this for future work.

5.3 Curves Describing The Hoofed Mammals Feet

In Section 4.2, we observed that the sheep and giraffe feet can be generated by a transformation of the ox foot. Now we would like to know whether or not we can generate other hoofed mammals' feet using these three feet. So, similarly to the curves that were fitted through the human skulls' transformations, we fit a curve through the transformations between the feet. That curve may generate other hoofed mammals' feet.

Three transformations are obtained from the registration of feet as follows:

$$\varphi_0^{-1}(x,y) = (x,y),$$
$$\varphi_1^{-1}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1.2373x - 0.0238 \\ \frac{0.8246y+0.0007}{-0.3967y+1.2124} \end{pmatrix},$$
$$\varphi_2^{-1}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 2.3678x - 0.1671 \\ \frac{0.6067y+0.0185}{-1.0295y+1.6169} \end{pmatrix}.$$

Here φ_0 maps the ox to the ox, φ_1 maps the ox to the sheep and φ_2 maps the ox to the giraffe. These transformations in the y-direction should preserve two points: zero and one ('o' and 'y' in Figure 4.5). Let $\varphi_i(.,y) = \frac{a_i y + b_i}{c_i y + d_i}$ be the general form of the transformations on the y-axis. Then:

$$\frac{a_i(0)+b_i}{c_i(0)+d_i} = 0 \Rightarrow b_i = 0,$$
$$\frac{a_i(1)+0}{c_i(1)+\frac{1}{a_i}} = 1 \Rightarrow c_i = a_i - \frac{1}{a_i}.$$

It can be seen that b_i is almost zero and $c_i = a_i - \frac{1}{a_i}$. Therefore, $\varphi_i(.,y)$ can be written as a function of only one variable as follows:

$$\frac{a_i y + b_i}{c_i y + d_i} = \frac{a_i y}{(a_i - \frac{1}{a_i})y + \frac{1}{a_i}}. \tag{5.2}$$

The translation part of $\varphi_i(x,.)$ can be ignored, because it only relates to the position of the image in the plane. Therefore, φ_i is considered as a function of only two variables:

$$\varphi_i(x,y) = \left(s_i x, \frac{a_i y}{(a_i - \frac{1}{a_i})y + \frac{1}{a_i}} \right).$$

As explained in Section 2.1.9, there is a homeomorphism between $PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$ and the matrix group $SL(2,\mathbb{R}) \times SL(2,\mathbb{R})$. Therefore, each transformation $\phi(x,y) = (ex+f, \frac{ay+b}{cy+d})$ is in correspondence with a matrix in $SL(2,\mathbb{R}) \times SL(2,\mathbb{R})$ as follows:

$$ex + f \leftrightarrow \begin{pmatrix} \frac{s}{\sqrt{s}} & \frac{f}{\sqrt{s}} \\ 0 & \frac{1}{\sqrt{s}} \end{pmatrix},$$
$$\frac{ay+b}{cy+d} \leftrightarrow \begin{pmatrix} a & b \\ c & d \end{pmatrix}.$$

Therefore, the matrix corresponding to φ_i is

$$\varphi_i(x,y) \leftrightarrow \left(\begin{pmatrix} s_i & 0 \\ 0 & \frac{1}{s_i} \end{pmatrix}, \begin{pmatrix} a_i & 0 \\ a_i - \frac{1}{a_i} & \frac{1}{a_i} \end{pmatrix} \right).$$

We use the same notation φ_i for the corresponding matrix. The general form of a Lie algebra element corresponding to φ_i is:

$$v_i = \log(\varphi) = \left(\begin{pmatrix} \beta_i & 0 \\ 0 & -\beta_i \end{pmatrix}, \begin{pmatrix} \alpha_i & 0 \\ 2\alpha_i & -\alpha_i \end{pmatrix} \right).$$

So:

$$v_0(x,y) = \log(\varphi_0^{-1}) = \left(\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \right),$$

$$v_1(x,y) = \log(\varphi_1^{-1}) = \left(\begin{pmatrix} 0.1065 & 0 \\ 0 & -0.1065 \end{pmatrix}, \begin{pmatrix} -0.1927 & 0 \\ -0.3943 & 0.1927 \end{pmatrix} \right),$$

$$v_2(x,y) = \log(\varphi_3^{-1}) = \left(\begin{pmatrix} 0.4310 & 0 \\ 0 & -0.4310 \end{pmatrix}, \begin{pmatrix} -0.4871 & 0 \\ -0.9928 & 0.4871 \end{pmatrix} \right).$$

Since we have only three points in the Lie algebra, we fit a line. We only fit a line passing through the origin. The line L is obtained by minimisation of:

$$\sum_i \|v_i - t_i A\|^2 = \sum_i tr((v_i - t_i A)(v_i - t_i A)^*)),$$

where A is :

$$A = \left(\begin{pmatrix} -0.646 & 0 \\ 0 & 0.646 \end{pmatrix}, \begin{pmatrix} 0.7633 & 0 \\ 2(0.7633) & -0.7633 \end{pmatrix} \right), \tag{5.3}$$

and the residual is 0.0057, and t_i corresponding to each v_i is: $t_1 = 0$, $t_2 = -0.2356$, $t_3 = -0.6437$. Figure 5.9 shows the data $\{(\beta_i, \alpha_i), i = 0, 1, 2\}$ in $\mathbb{R}^2$ along the fitted line $t \mapsto (-0.6460t, 0.7633t)$.

To find the curve in the group, we need to map the line L into the group using the exponential function. Figure 5.10 shows the data $\{(s_i, a_i), i = 0, 1, 2\}$ and the curve $t \mapsto (\exp(-0.6460t), \exp(0.7633t))$.

We also calculate the MSE by the leave-one-out method for this line. Table 5.4 gives the errors and MSE. The value of MSE is 0.018 which is small, so this model may be reliably used to predict the new data. So, we produce more new feet by the transformation of

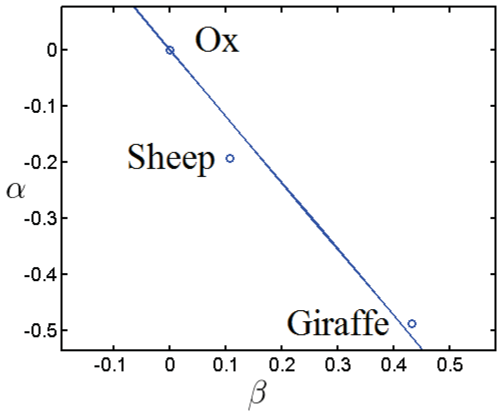


Figure 5.9: A two-dimensional representation of the line L fitted through $v_i = \log(\varphi_i) = \left(\begin{pmatrix}\beta_i & 0\\ 0 & \beta_i\end{pmatrix}, \begin{pmatrix}\alpha_i & 0\\ 2\alpha_i & \alpha_i\end{pmatrix}\right)$, where φ_0 maps ox to ox, φ_1 maps ox to sheep and φ_2 maps ox to giraffe.

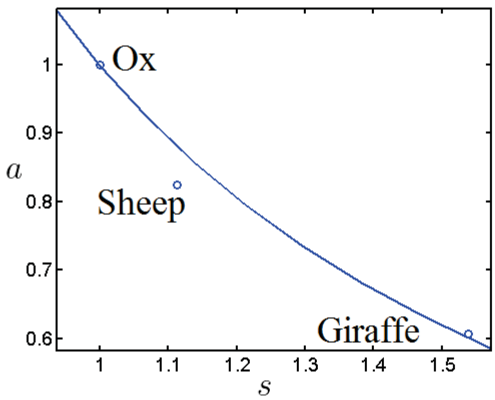


Figure 5.10: A two-dimensional representation of the exponential curve $\exp(L)$ (L is shown in Figure 5.9) fitted through $v_i = \left(\begin{pmatrix}\beta_i & 0\\ 0 & \beta_i\end{pmatrix}, \begin{pmatrix}\alpha_i & 0\\ 2\alpha_i & \alpha_i\end{pmatrix}\right) = \log(\varphi_i) = \log\left(\begin{pmatrix}s_i & 0\\ 0 & \frac{1}{s_i}\end{pmatrix}, \begin{pmatrix}a_i & 0\\ a_i - \frac{1}{a_i} & \frac{1}{a_i}\end{pmatrix}\right)$, where φ_0 maps ox to ox, φ_1 maps ox to sheep and φ_2 maps ox to giraffe, (see Figure 5.9).

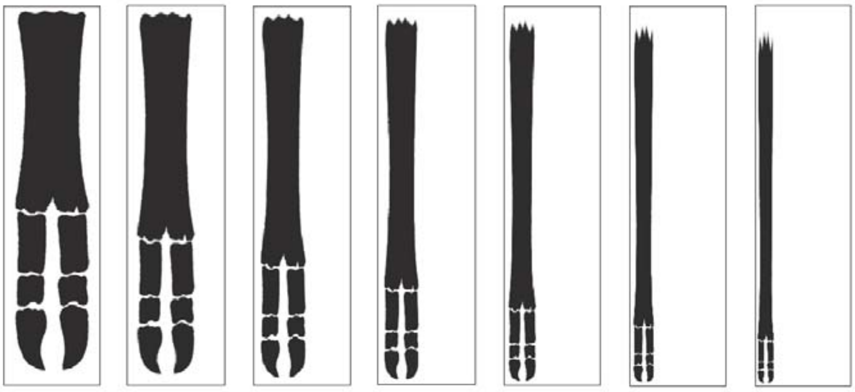


Figure 5.11: Transformation of ox foot by $\exp(L(t))$, where t is from left to right: $0, -0.2, -0.4, -0.6, -0.8, -1, -1.2$; $E(t)$ is the line fitted through $\log(\varphi_i)$, $i = 0, 1, 2$, where φ_0 maps ox to ox, φ_2 maps ox to sheep and φ_2 maps ox to giraffe.

	$i = 1$	$i = 2$	$i = 3$	MSE= $\frac{\sum_{i=1}^{3} e_i^2}{3}$
e_i	0	0.0806	0.2182	0.0180

Table 5.4: Leave-one-out method is employed for the data v_k, $k = 1, 2, 3$. Here v_k are the Lie algebra elements corresponding to the transformation of the ox foot to the ox, sheep and giraffe feet in $PSL(2, \mathbb{R}) \times PSL(2, \mathbb{R})$, where a line passing through the origin is fitted through the data except v_i; e_i are the distance of data v_i from the fitted line. MSE is the mean of the square of e_i, $i = 1, 2, 3$.

the ox by $\exp(L(t))$, where $t = 0, -0.2, -0.4, -0.6, -0.8, -1, -1.2$, see Figure 5.11.

5.4 Conclusion and Future Work

In this chapter we introduced a method to find a parameterized curve in the Lie group passing through the data. We employed the method for two examples, human skulls and hoofed mammal feet. For the human skull, we fitted lines in the Lie algebras of the groups $PSL(2, \mathbb{C})$ and $PSL(3, \mathbb{R})$. Then we calculated the coefficient of determination of the fitted lines to see if the data are linear or not. The coefficient of determination of the lines showed that the data may well be linear, although there are very few datapoints. Moreover, for the human skulls we calculated the MSE by leave-one-out method for four models, we found that lines can be reliably used to describe the human skull growth. In order to compare between the lines, simplicity, a good match (measured as a small residual), and the fact that it is a group may convince us to choose the one-parameter subgroup as a model to describe the human skull growth. Also, MSE of the one-parameter subgroups curve of feet was small, so the other hoofed mammal feet may be generated from the fitted one-parameter subgroup.

The fitted curve depends on the choice of the metric, so as future work we can consider other choices of metric. Also the number of our data was small, if we have more data we can test the accuracy of the model better. So as future work we can collect more data to find a more accurate model.

Chapter 6

Multi-Registration of Images

Thompson says:

> "Growth and Form are throughout of this composite nature; therefore the laws of mathematics are bound to underlie them, and her methods to be peculiarly fitted to interpret them." (page 1028 in [67]).

Thompson's point of view that every phenomenon is a composite of simple actions inspires us to introduce a new idea to the field of image registration. We call it *multi-registration*. Here images are registered with a sequence of simple groups of transformations to derive simple actions. One case of multi-registration is about using progressively larger groups. An idea that is slightly related is to use progressively smaller deformation, for example by starting with a low resolution smoothed image, and iteratively refining the registration as the image resolution is increased [62]. Another model is the GRID (Growth by Random Iterated Diffeomorphisms), which is focused on deriving local elementary deformations (small diffeomorphisms) successively. Each transformation that is a diffeomorphism is specified in polar coordinate systems at the centre of growth using two functions: angular and radial deformation functions [29].

In the mathematical setting, multi-registration can be written as:

Definition 6.0.1. *Let $G_1, G_2, ..., G_n$ be groups acting on the space of images, $\|\cdot\|$ the* *distance function, I the source and J the target. A* multi-registration *is the solution to*

$$\varphi_i = \arg\min_{\varphi_i \in G_i} \|(I_{i-1} \circ \varphi_i^{-1} - J)\|,$$
$$\textit{such that } I_0 = I \ \textit{ and } \quad I_i = I_{i-1} \circ \varphi_i^{-1}, i = 1, 2, ..., n. \tag{6.1}$$

In Chapter 2 the lattice of finite dimensional planar Lie groups was given. Groups on the lattice either have a group–subgroup relationship, or they do not. Therefore, two different cases can be considered in multi-registration, as follows:

Case one: Multi-registration on a chain of groups, where $G_i \subset G_{i+1}$. Registration is preformed first in G_i then in G_{i+1}.

$$G_1 \longrightarrow G_2 \longrightarrow G_3 \dots$$

Figure 6.1: The edge $G_i \to G_{i+1}$ means that G_i is a subgroup of G_{i+1}, and multi-registration is performed first in G_i and then in G_{i+1}.

Case two: Multi-registration on a tree of groups. Registration is performed in a different order that in the lattice diagram.

$$\begin{tikzcd} & G_1 \arrow[dl] \arrow[dr] & \\ G_3 \arrow[d] & & G_2 \\ G_4 & & \end{tikzcd}$$

Figure 6.2: The edge $G_i \to G_j$ means that G_i is a subgroup of G_j. On a tree, multi-registration can be performed in several different orders.

Multi-registration of images gives us the following information:

- A set of partially transformed images I_i.
- The transformations φ_i that partially register in G_i
- The discrepancies $\|I_{i-1} \circ \varphi_i^{-1} - J\|$.
- The norms of the partial transformations $d_i = tr(\log(\varphi_i) \log(\varphi_i)^*)^{\frac{1}{2}}$.

- Another measure of the size of φ_i is a length L. Let $\phi(t)$ be a curve in the diffeomorphism group, with $\phi(0) = id$, $\phi(1) = \varphi_i$. Let

$$L = \frac{1}{\mathrm{Vol}(\Omega)} \int_0^1 \int_{(x,y)\in\Omega} \| \frac{d}{dt}\phi_t(x,y)\|_2 dxdydt \tag{6.2}$$

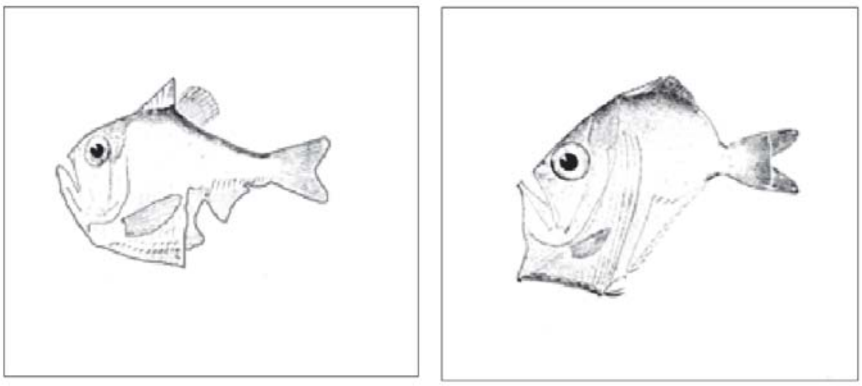
  where $\| \cdot \|_2$ is the L^2-norm of the gradient field, and $\mathrm{Vol}(\Omega)$ is the volume of Ω. The value of L provides us the with information about the amount of deformation of the domain of the image, in a way that is independent of the choice of norm in d.

This information can be useful to understand the relationship between images better.

6.1 Multi-registration in a Chain of Groups

This section is about multi-registrations in a chain of groups. We give two different examples. The first example is multi-registration with $G_3 = PSL(3,\mathbb{R})$ and its two subgroups G_2 = affine and G_1 = special affine, and the second example is multi-registration with $G_1 = PSL(2,\mathbb{C})$ and G_2 = diffeomorphism group.

Example 16. In Section 4.4, *Argyropelecus olfersi* as source (I) is registered with *Sternoptyx diaphana* as target (J) in $G_3 = PSL(3,\mathbb{R})$ and its two subgroups, G_2 = affine and G_1 = special affine.



(a) *Argyropelecus olfersi* (b) *Sternoptyx diaphana*

Figure 6.3: Two fish species images taken from [50].

Three images I_i and three transformations ϕ_i^{-1}, $i = 1, 2, 3$ are obtained by multi-registration of I and J. Let $\phi_i^{-1} = \varphi_i$, then

$$I_i = I_{i-1} \circ \varphi_i, \;\; \varphi_i \in G_i,$$

and we find

$$\varphi_1(x, y) = (1.0537x + 0.2062y - 0.0411, 0.1138x + 0.9713y + 0.0025),$$

$$\varphi_2(x, y) = (0.9648x + 0.0876y - 0.0076, 0.017x + 0.8221y - 0.0048),$$

and

$$\varphi_3(x, y) = \left(\frac{1.0485x + 0.0505y - 0.0172}{-0.3293x - 0.4784y + 0.9811}, \frac{0.0578x + 0.9864y - 0.0138}{-0.3293x - 0.4784y + 0.9811} \right).$$

Let Ω' be a rectangular grid on the image domain Ω, $\Omega'_0 = \Omega'$, and $\Omega'_i = \varphi_i^{-1} \circ \varphi_{i-1}^{-1} \circ \cdots \circ \varphi_1^{-1}(\Omega')$. Figure 6.4 shows images I_i, $i = 1, 2, 3$ along with the deformed grid Ω'_i, and Figure 6.5 the sequences of output images during the multi-registration.

We map transformations into their Lie algebras to calculate the distances d_i.

$$v_1 = \log(\varphi_1) = \begin{pmatrix} 0.0410 & 0.2053 & -0.0406 \\ 0.1133 & -0.0410 & 0.0049 \\ 0 & 0 & 0 \end{pmatrix},$$

$$v_2 = \log(\varphi_2) = \begin{pmatrix} -0.0367 & 0.0983 & -0.0075 \\ 0.0191 & -0.1969 & -0.0052 \\ 0 & 0 & 0 \end{pmatrix},$$

$$v_3 = \log(\varphi_3) \begin{pmatrix} 0.0435 & 0.0458 & -0.0167 \\ 0.0548 & -0.0183 & -0.0136 \\ -0.3124 & -0.4804 & -0.0251 \end{pmatrix}.$$

And:

$$d_1 = tr(v_1 v_1^*)^{\frac{1}{2}} = 0.2449,$$
$$d_2 = tr(v_2 v_2^*)^{\frac{1}{2}} = 0.2241,$$
$$d_3 = tr(v_3 v_3^*)^{\frac{1}{2}} = 0.5803.$$

In Section 4.4, images are registered with the $PSL(3, \mathbb{R})$ group, and the following

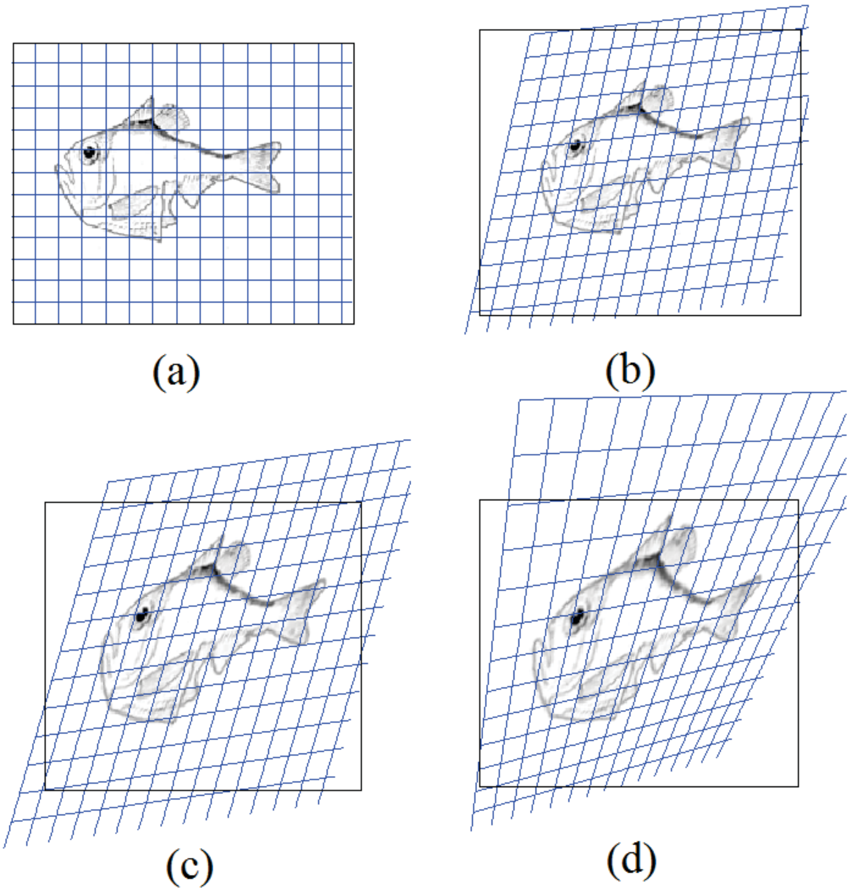
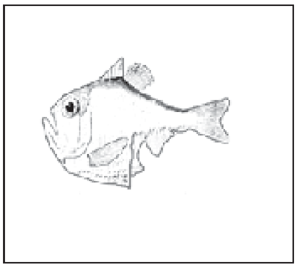
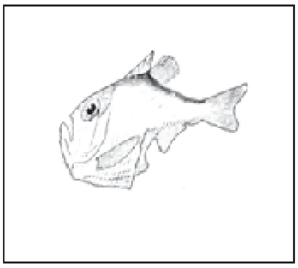
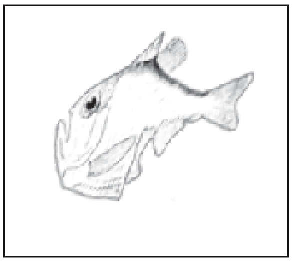


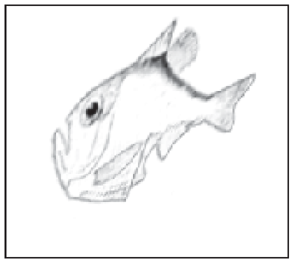
Figure 6.4: (a) Source I_0 and rectangular grid Ω', (b) source after registration in the special affine group, where it is transformed to $I_1 = I_0 \circ \varphi_1$, along with the transformed grid $\Omega_1{}' = \varphi_1^{-1}(\Omega')$, (c) second step of multi-registration in the affine group, where the source is I_1 and it is transformed to $I_2 = I_1 \circ \varphi_2$, along with the transformed grid $\Omega_2{}' = \varphi_2^{-1}(\Omega_1{}')$, (d) third step of multi-registration in $PSL(3, \mathbb{R})$, where the source is I_2 and it is transformed to $I_3 = I_2 \circ \varphi_3$, along with the transformed grid $\Omega_3{}' = \varphi_3^{-1}(\Omega_2{}')$.



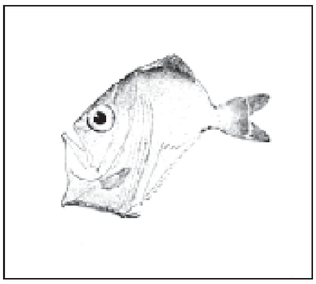
(a) I_0: Source







(b) $I_1 = I_0 \circ \varphi_1$ (c) $I_2 = I_1 \circ \varphi_2$ (d) $I_3 = I_2 \circ \varphi_3$



(e) J: Target

Figure 6.5: Image I_0 is multi-registered with J. (a) I_0 (source) (b) $I_1 = I_0 \circ \varphi_1$, (c) $I_2 = I_1 \circ \varphi_2$, (d) $I_3 = I_2 \circ \varphi_3$, where φ_1 belongs to special affine, φ_2 belongs to affine and φ_3 belongs to $PSL(3, \mathbb{R})$.

transformation is obtained,

$$f(x, y) = \left(\frac{1.1624x + 0.3472y - 0.0671}{-0.3891x - 0.4897y + 1.0740}, \frac{0.1872x + 0.8781y - 0.0221}{-0.3891x - 0.4897y + 1.0740} \right),$$

which is almost equal to

$$\varphi_3 \circ \varphi_2 \circ \varphi_1(x, y) = \left(\frac{1.1696x + 0.3657y - 0.0720}{-0.4231x - 0.5158y + 1.0790}, \frac{0.1830x + 0.8729y - 0.0215}{-0.4231x - 0.5158y + 1.0790} \right).$$

where $\varphi_3 \circ \varphi_2 \circ \varphi_1$ is scaled by its determinant, which is 0.7917. They are not exactly equal, because the source at each step of multi-registration is the output of the registration in the previous step, which is affected by interpolation. Also, as mentioned in Chapter 3, registration of real images is more difficult than registration of synthetic images, because in synthetic images there is exactly one point where the images match perfectly, which is unlikely to exist at all in real images. In fact, this interpolation problem could be avoided by concatenating the actions of the groups rather than using the deformed image from the previous step, although this was not done here.

So, with multi-registration the transformation is decomposed into several transformations (three in this example), each of which gives us some information that cannot be obtained from the final transformation alone. In the following, this information is examined further.

Since $\varphi_i \cong id + v_i$, then $\exp(v_i t)$ is a shortest path connecting φ_1 and id because:

$$\int_0^1 \|\frac{d}{dt}\exp(tv)\|dt \cong \int_0^1 \|\frac{d}{dt}(id + tv)\|dt = \int_0^1 \|v\|dt = \|v\|.$$

Therefore, L_i is calculated as follows,

$$L_i = \frac{1}{\mathrm{Vol}(\Omega)} \int_0^1 \int_{(x,y)\in\Omega} \|\frac{d}{dt}\exp(v_i t)(x, y)\|_2 dxdydt. \tag{6.3}$$

Because the domain of the images is discretized, $\mathrm{Vol}(\Omega)$ is taken as the total number of pixels. The following Table 6.1 gives all the information that we have measured about the multi-registration. Each φ_i gives us a better intuition about the geometrical relationship between the fish. This is summarized in Table 6.2.

Note here that when registering in a group we may not get all (or indeed any part) of the sub-transformation of the group. For example, shearing can be obtained by the action of special affine on the plane, but comparing the φ_i shows that there is still some shearing in the affine and $PSL(3, \mathbb{R})$ transformations. However, the significant

	$i=0$	$i=1$	$i=2$	$i=3$
$\|I_i - J\|_2^2$	606.2144	414.2554	384.4359	332.5374
$D_i = 1 - \frac{\|I_i - J\|_2^2}{\|I_{i-1} - J\|_2^2}$	0	0.3167	0.072	0.135
d_i	-	0.2449	0.2241	0.5803
L_i	-	0.0726	0.0635	0.0569

Table 6.1: Information of multi-registration of source and target in Figure 6.3 by three groups: G_1 = special affine ($i = 1$) , G_2 = affine ($i = 2$), and $G_3 = PSL(3, \mathbb{R})$ ($i = 3$).

	Transformation	Interpretation of the information
Special Affine	$x \mapsto 1.0537x + 0.2062y - 0.0411$ $y \mapsto 0.1138x + 0.9713y + 0.0025$	• Shears are: 0.2062 and 0.1138. • Determinant is one $\Rightarrow$ volume of I_1 = volume of I.
Affine	$x \mapsto 0.9648x + 0.0876y - 0.0048$ $y \mapsto 0.017x + 0.8221y - 0.0076$	• Shears are small (since they are already taken out by special affine). • $\frac{\text{Volume of Source}}{\text{Volume of Target}} = 0.7917$
Projective	$x \mapsto \dfrac{1.0485x + 0.0505y - 0.0172}{-0.3293x - 0.4784y + 0.9811}$ $y \mapsto \dfrac{0.0578x + 0.9864y - 0.0138}{-0.3293x - 0.4784y + 0.9811}$	• Scale and shear parts of the transformation are close to the identity. • There are some projective parts, which squeeze and expand the image to make a better alignment.

Table 6.2: Information of multi-registration of two fishes given in Figure 6.3 in three groups: special affine, affine, and $PSL(3, \mathbb{R})$.

part of the shear is in special affine. The main part of the scaling is seen in the affine group. The ratio of their size is $\frac{\text{Volume of Source}}{\text{Volume of Target}} = 0.7917$, see Table 6.2. Registration in the biggest group $PSL(3, \mathbb{R})$ makes a better alignment. The d_1, d_2 and d_3 tells us how big the transformations are. In order of impact, the projective parameters, shear and scale are the most important. And L_i tells us about the amount of deformation that each group performs on the domain of the image; it can be seen that they make almost equal deformations.

Therefore, this example shows us that with multi-registration a complex transformation can be given as a composition of simple sub-transformations: shear, scale and projective. Note also that the multi-registration $G_1 \to G_2 \to G_3$ with $G_1 \subset G_2 \subset G_3$ can be easier to compute than registration with G_3 alone, as the individual transformations are smaller and the dimensions (for G_1 and G_2) are lower.

The next example is to multi-register the human and chimpanzee skulls by two groups $G_1 = PSL(2, \mathbb{C})$ and $G_2 =$ diffeomorphism group. For the registration the same images as in Chapter 4 are used (the curves, with their inside filled with black).

Example 17. In Section 4.5 the human skull is deformed to the chimpanzee by $\phi^{-1} = \varphi_1 \in PSL(2, \mathbb{C})$.

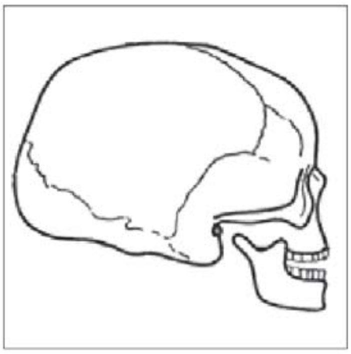
$$\varphi_1(z) = \frac{(1.2049 + 0.0608i)z + 0.2363 + 0.0793i}{(0.6641 - 0.7029i)z + 0.9992 - 0.1446i}.$$

However, the transformed human skull did not match the chimpanzee skull particularly well. Let the human skull be source I_0, chimpanzee J as target, and transform the human skull by $PSL(2, \mathbb{C})$ to make I_1, see Figure 6.6b.

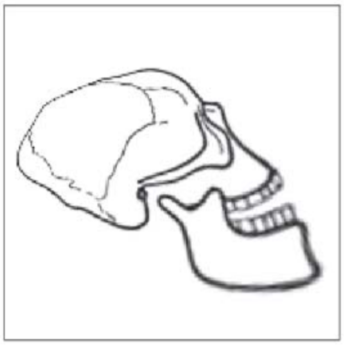
Registration is carried out in the full diffeomorphism group G_2, with source I_1 and target J. The greedy image matching algorithm is employed for the diffeomorphic registration (see Chapter 2) for 5 steps, with $\triangle t = 0.003$. The output of registration which transforms I_1 to $I_2 = I_1 \circ \varphi_2^{-1}$ is the diffeomorphism φ_2. Figure 6.6c shows I_2. Registration in G_2 makes an almost perfect match, Figure 6.7 shows the discrepancy between J with I_1 and I_2. Figure 6.8 shows the rectangular grid Ω' on the human skull which is deformed during multi-registration in G_1 and G_2.

We can calculate d_1 but not d_2:

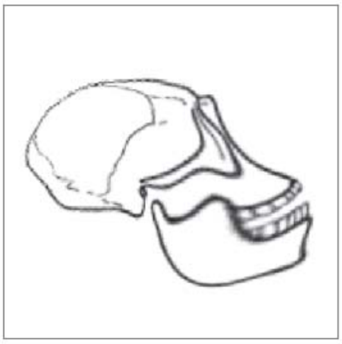
$$d_1 = tr(\log(\varphi_1)\log(\varphi_1)^*)^{\frac{1}{2}} = 0.9861,$$



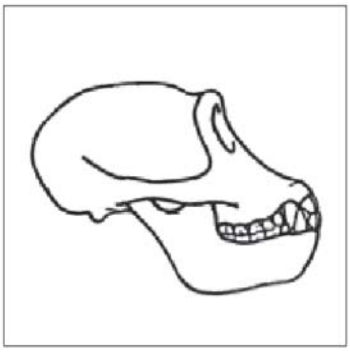
(a) Source: I_0



(b) I_1

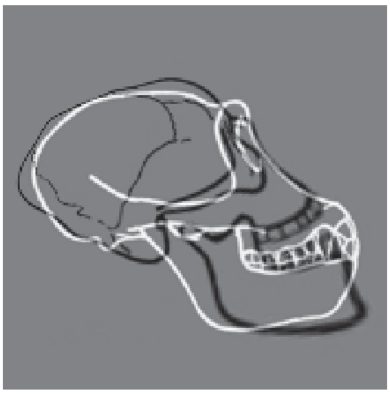


(c) I_2



(d) Target: J

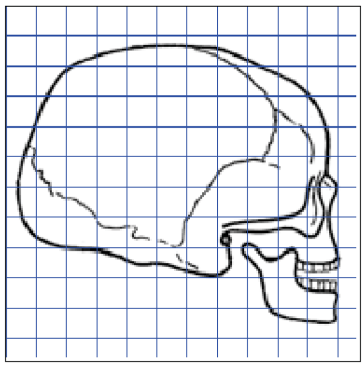
Figure 6.6: Multi-registration of the human skull as source and the chimpanzee skull as target in two groups: $G_1 = PSL(2, \mathbb{C})$, $G_2 =$ diffeomorphism. (a) Human skull as source. (b) I_1 which is transformed source and output of registration in $PSL(2, \mathbb{C})$. (c) I_2 which is transformed I_1 and the output of the registration in the diffeomorphism group, (d) chimpanzee skull (the target).



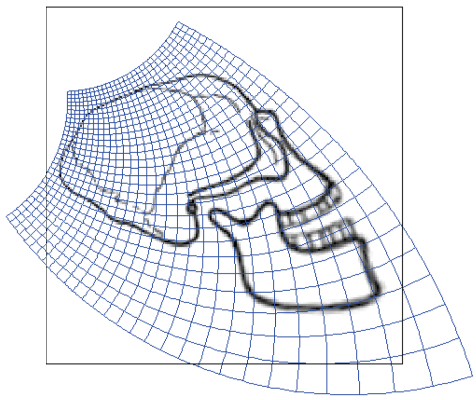
(a)

(b)

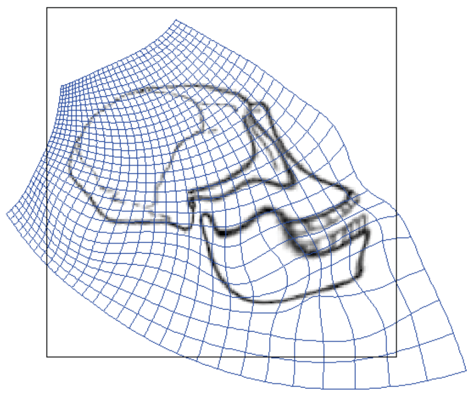
Figure 6.7: Discrepancy between the chimpanzee skull (J) the target and the transformed human skulls (I_0) in multi-registration, where $G_1 = PSL(2, \mathbb{C})$ and $G_2 =$ diffeomorphism group. (a) Discrepancy between chimpanzee skull and human skull after registration in $PSL(2, \mathbb{C})$, where the human skull is transformed to I_1. (b) Discrepancy between the chimpanzee skull and I_1 after registration in the diffeomorphism group.



(a) I_0



(b) I_1



(c) I_2

Figure 6.8: The output of multi-registration of the human skull as source and the chimpanzee skull as target in $PSL(2,\mathbb{C})$ and the diffeomorphism group. (a) The human skull as source along with the rectangular grid $\Omega_1{}'$. (b) Transformation of the human skull and $\Omega_1{}'$ by $PSL(2,\mathbb{C})$ to I_1 and $\Omega_2{}'$. (c) Transformation of I_1 and $\Omega_1{}'$ by diffeomorphism group to I_2 and $\Omega_2{}'$.

where

$$\log(\varphi_1) = \begin{pmatrix} 0.0982 + 0.1006i & 0.2275 + 0.0797i \\ 0.6513 - 0.6712i & -0.0981 - 0.1007i \end{pmatrix}.$$

Since

$$\varphi_1 = \begin{pmatrix} 1.2049 + 0.0608i & 0.2363 + 0.0793i \\ 0.6641 - 0.7029i & 0.9992 - 0.1446i \end{pmatrix} \cong id + \log(\varphi_1) =$$

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 0.0982 + 0.1006i & 0.2275 + 0.0797i \\ 0.6513 - 0.6712i & -0.0981 - 0.1007i \end{pmatrix} =$$

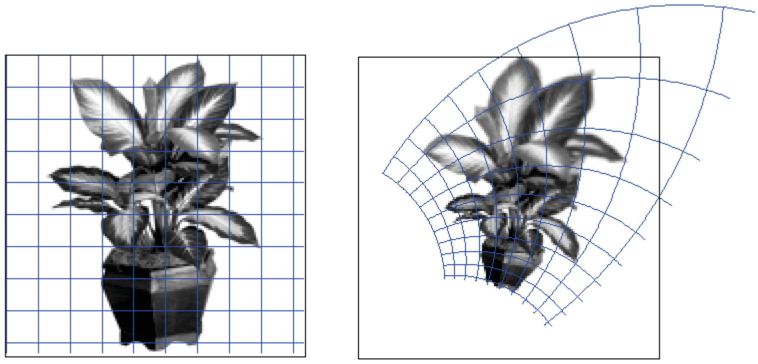
$$\begin{pmatrix} 1.0982 + 0.1006i & 0.2275 + 0.0797i \\ 0.6513 - 0.6712i & 0.9019 - 0.1007i \end{pmatrix},$$

L_1 can be calculated by Equation (6.3) and it is 0.2479. In diffeomorphic registration by the greedy image matching algorithm a path ϕ_t is generated, where $\phi_0 = id$ is the start point and φ_2 is the end point. Using Equation (6.3) $L_2 = 0.03$. Although ϕ_t is not a geodesic, since the deformation is small, the length of geodesic and the length of ϕ_t should not be significantly different. Otherwise the LDDMM algorithm needs to be employed, see Section 1.2. The residuals, d_i and L_i are given in Table 6.3.

	$i = 0$	$i = 1$	$i = 2$
Transformation	-	$\varphi_1(z) = \frac{(1.2049+0.0608i)z+0.2363+0.0793i}{(0.6641-0.7029i)z+0.9992-0.1446i}$	-
$\lVert I_i - J\rVert_2^2$	6330.2	1818.8	512.4584
$D_i = 1 - \frac{\lVert I_i-J\rVert_2^2}{\lVert I_{i-1}-J\rVert_2^2}$	0	0.7128	0.7182
d_i	-	0.9861	-
L_i	-	0.2479	0.03

Table 6.3: Information from multi-registration of human skull and chimpanzee skull.

As can be seen in Table 6.3, D_1 and D_2 are almost equal, which means that they make similar amounts of alignment between the images. But an important point is that $PSL(2, \mathbb{C})$ is a six dimensional group, while the diffeomorphism group is infinite dimensional. This suggests that $PSL(2, \mathbb{C})$ has a more significant role than the diffeomorphism group on the deformation of human skull to chimpanzee skull. Also, L_1 is much bigger than L_2, which indicates that $PSL(2, \mathbb{C})$ makes a bigger deformation in comparison to diffeomorphism. Therefore, we draw the conclusion that the human skull is well registered by a Möbius transformation to the chimpanzee skull. Note that one important difference between the groups is that the finite dimensional groups are



(a) (b)

Figure 6.9: (a) Source, I_0, (b) target, J. The target is generated by a $PSL(2, \mathbb{C})$ transformation of the source.

in some sense global, since the same matrix is applied to all points, while the diffeomorphisms are local, and can deform different parts of the image differently.

6.2 Multi-registration on a Tree of Groups

In this section, two examples of multi-registration on trees of groups are given. In the first example, the groups are $G_3 = PSL(2, \mathbb{C})$ and its two disjoint subgroups, $G_2 = NPSL(2, \mathbb{C})$ and $G_1 =$ similarity, and in the second example the groups are $PSL(2, \mathbb{C})$ and $PSL(2, \mathbb{R}) \times PSL(2, \mathbb{R})$.

Example 18. In this example, we take an image of a plant as source I_0, shown in Figure 6.9. The domain of the source is taken as $[-0.5, 0.5] \times [-0.5, 0.5]$. We generate a target J from the source by a $PSL(2, \mathbb{C})$ transformation of I_0, where the transformation is:

$$\phi^{-1}(z) = \frac{(1.2 - 0.01i)z - 0.01 - 0.02i}{(0.6 + 0.7i)z + 0.8401 - 0.0088i}.$$

In contrast to chain multi-registration where there is one natural order of groups, in tree multi-registration there are different possible orders. In this example two different orders of groups can be considered.

First case: Source and target are registered in $G_1 = NPSL(2, \mathbb{C})$, $G_2 =$ similarity, and $G_3 = PSL(2, \mathbb{C})$ respectively. Three transformations ϕ_i^{-1} and three images are obtained from multi-registration as follows, let $\phi_i^{-1} = \varphi_i$ (where

$\varphi_1 \in NPSL(2,\mathbb{C})$, $\varphi_2 \in$ similarity, and $\varphi_3 \in PSL(2,\mathbb{C})$):

$$\varphi_1(z) = \frac{z}{(0.611+0.5365i)z+1},$$
$$\varphi_2(z) = (1.4593 - 0.0679i)z - 0.0155 - 0.0244i,$$
$$\varphi_3(z) = \frac{(0.9875+0.0127i)z+0.0022+0.0008i}{(-0.158+0.0997i)z+1.0121-0.0129i}.$$

$$I_1 = I_0 \circ \varphi_1,$$
$$I_2 = I_1 \circ \varphi_2,$$
$$I_3 = I_2 \circ \varphi_3.$$

Images I_0, I_1, I_2, and I_3 are shown in Figure 6.10. The discrepancies between I_0, I_1, I_2 and I_3 with J are shown in Figure 6.11.

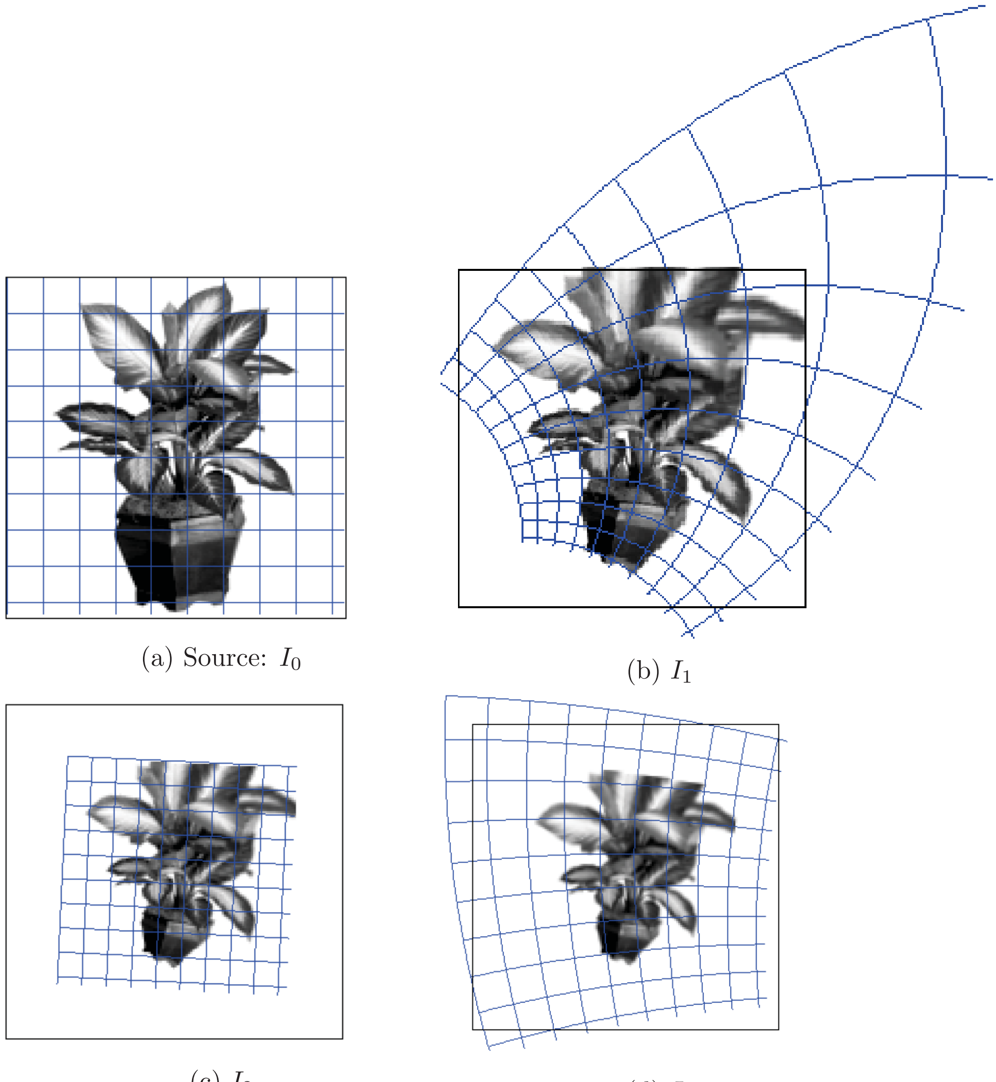
The residuals, d_i, and L_i, $i = 1, 2, 3$ are given in Table 6.4, where L_i is calculated using Equation (6.3).

	$i=0$	$i=1$	$i=2$	$i=3$
$E_i = \|I_i - J\|_2^2$	2139.6	1915.2	419.9	278.58
$D_i = 1 - \frac{\|I_i - J\|_2^2}{\|I_{i-1} - J\|_2^2}$	0	0.1049	0.78	0.34
d_i	-	0.8131	0.2711	0.1884
L_i	-	0.1443	0.1239	0.0340

Table 6.4: Output of multi-registration of the source and target given in Figure 6.9 in three groups $G_1 = NPSL(2,\mathbb{C})$, $G_2 =$ similarity and $G_3 = PSL(2,\mathbb{C})$.

As can be seen in Table 6.4, D_2 is the biggest value, so the majority of alignment is made by the similarity group, this can also be seen in Figure 6.11. Although $NPSL(2,\mathbb{C})$ matches only 10% of the images, this does not mean this group performs less deformation: L_1 shows that this group makes the biggest deformation on the images. Moreover, the non-linear parameter of φ_1 is $0.611+0.5365i$, which is almost close to the non-linear parameter of ϕ^{-1}, $0.6+0.7i$. This shows that registration in $NPSL(2,\mathbb{C})$ produces almost all of the non-linearity. After two steps of registration the images are almost matched. In other words, most of the deformation is fulfilled by the two subgroups, so registration in $PSL(2,\mathbb{C})$ does not produce a big deformation; L_3 is very small. Comparing d_i tells us about the size of transformations: in order of decreasing size the transformations are φ_1, φ_2, φ_3.

Second case: The source and the target are registered in $G_1 =$ similarity, $G_2 =$



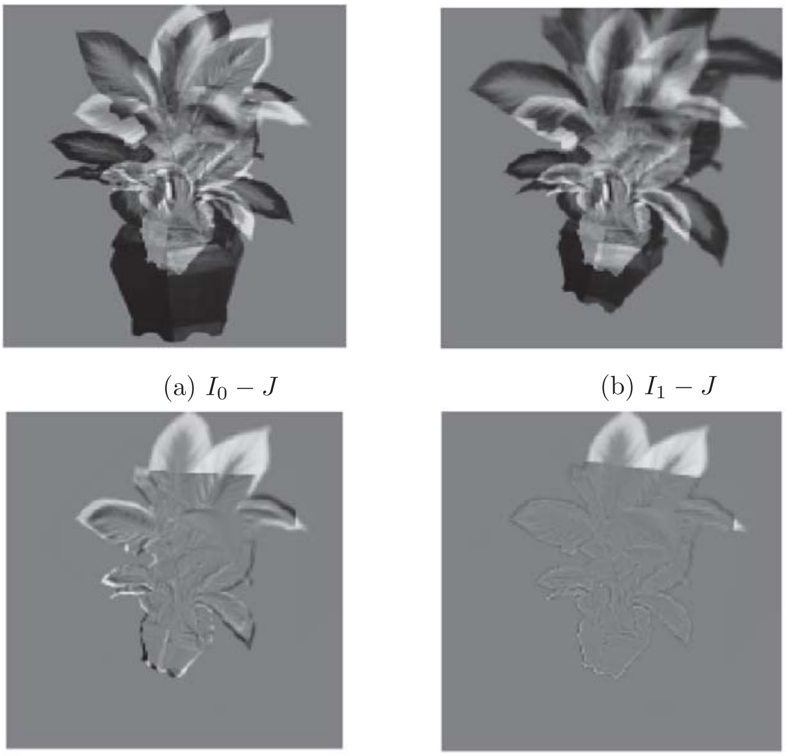
(a) Source: I_0

(b) I_1

(c) I_2

(d) I_3

Figure 6.10: Multi-registration of the source (I_0) and the target (J) given in Figure 6.10 in three groups: $G_1 = NPSL(2, \mathbb{C})$, $G_2 =$ similarity, and $G_3 = PSL(2, \mathbb{C})$. (a) Source along with rectangular grid Ω'. (b) Output of multi-registration in G_1: Transformation of I_0 and Ω' to I_1 and Ω'_1. (c) Output of multi-registration in G_2: Transformation of I_1 and Ω'_1 to I_2 and Ω'_2. (d) Output of multi-registration in G_3: Transformation of I_2 and Ω'_2 to I_3 and Ω'_3.



(a) $I_0 - J$ (b) $I_1 - J$

(c) $I_2 - J$ (d) $I_3 - J$

Figure 6.11: Multi-registration of the source (I_0) and target (J) given in Figure 6.9 in three groups: $G_1 = NPSL(2, \mathbb{C})$, $G_2 =$ similarity, and $G_3 = PSL(2, \mathbb{C})$. Discrepancy between target (J) and: (a) source, (b) I_1 that is output of multi-registration in G_1, (c) I_2 that is output of multi-registration in G_2, (d) I_3 that is output of multi-registration in G_3.

$NPSL(2,\mathbb{C})$, and $G_3 = PSL(2,\mathbb{C})$ respectively. Three transformations ϕ_i^{-1} are obtained from multi-registration as follows (let $\phi_i^{-1} = \varphi_i$):

$$\varphi_1(z) = (1.2654 - 0.1061i)z - 0.0403 + 0.1529,$$
$$\varphi_2(z) = \frac{z}{(-0.0468 - 0.0231i)z + 1},$$
$$\varphi_3(z) = \frac{(1.1831 - 0.0160i)z + 0.0252 - 0.1311}{(0.7651 + 0.7863i)z + 0.9495 - 0.0551i}.$$

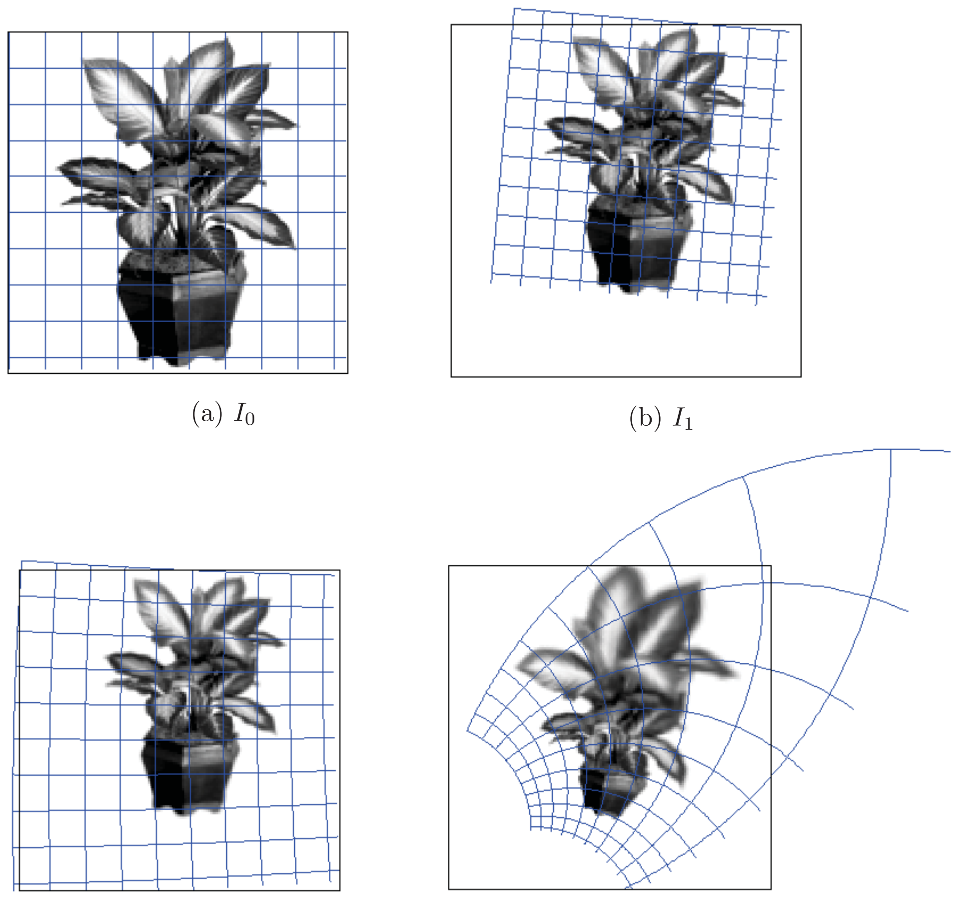
In this list, $\varphi_1 \in$ similarity, $\varphi_2 \in NPSL(2,\mathbb{C})$, and $\varphi_3 \in PSL(2,\mathbb{C})$. The three output images are I_1, I_2, and I_3; they are shown in Figure 6.12. The discrepancy between I_0, I_1, I_2, I_3 and J are shown in Figure 6.13. The residuals, d_i, L_i, $i = 1, 2, 3$ are given in Table 6.5, where L_i is calculated using Equation (6.3).

	$i = 0$	$i = 1$	$i = 2$	$i = 3$
$E_i = \|I_i \circ \varphi_i - J\|$	2139.6	940.24	918.37	14.73
$D_i = 1 - \frac{\|I_i \circ \varphi_i - J\|}{\|I_{i-1} - J\|}$	0	0.5606	0.0237	0.9840
d_i	-	0.2272	0.0520	1.0341
L_i	-	0.1420	0.0087	0.1893

Table 6.5: Output of multi-registration of the source and target given in Figure 6.9 in three groups $G_1 =$ similarity, $G_2 = NPSL(2,\mathbb{C})$, and $G_3 = PSL(2,\mathbb{C})$.

As mentioned before, when registering in a group we may not get all or any part of the sub-transformation of the group. In this example, we observe that $NPSL(2,\mathbb{C})$ does not give the non-linearity part, also D_2, d_2 and L_2 are very small and φ_2 is very close to identity.

Example 19. Thompson compared the *Diodon porcupine* fish and the *Orthagoriscus mola* fish, also known as a *Sunfish*. He described the *sunfish* as a non-linear transformation of *Diodon*. He deformed the vertical lines on *Diodon* fish to circular lines and the horizontal lines to approximately hyperbolic lines. Then the outline of the *Diodon* fish was transferred to the new grid, and the resulting figure resembled the *Sunfish*, as is shown in Figure 6.14.



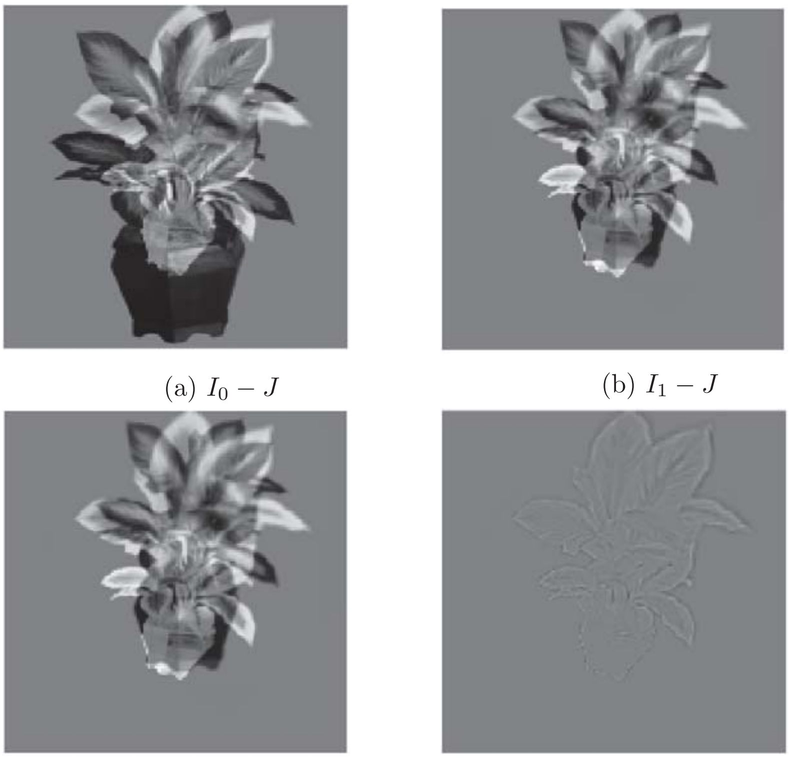
(a) I_0

(b) I_1

(c) I_2

(d) I_3

Figure 6.12: Multi-registration of the source (I_0) and target (J) given in Figure 6.9 in three groups: G_1 = similarity, $G_2 = NPSL(2, \mathbb{C})$, and $G_3 = PSL(2, \mathbb{C})$. (a) Source along with rectangular grid Ω'. (b) Output of multi-registration in G_1: Transformation of I_0 and Ω' to I_1 and Ω'_1. (c) Output of multi-registration in G_2: Transformation of I_1 and Ω'_1 to I_2 and Ω'_2. (d) Output of multi-registration in G_3: Transformation of I_2 and Ω'_2 to I_3 and Ω'_3.



(a) $I_0 - J$ (b) $I_1 - J$

(c) $I_2 - J$ (d) $I_3 - J$

Figure 6.13: Multi-registration of the source (I_0) and target (J) given in Figure 6.9 in three groups: G_1 = similarity, $G_2 = NPSL(2, \mathbb{C})$, and $G_3 = PSL(2, \mathbb{C})$. Discrepancy between the target (J) and: (a) source, (b) I_1, the output of multi-registration in G_1, (c) I_2, the output of multi-registration in G_2, (d) I_3, the output of multi-registration in G_3.

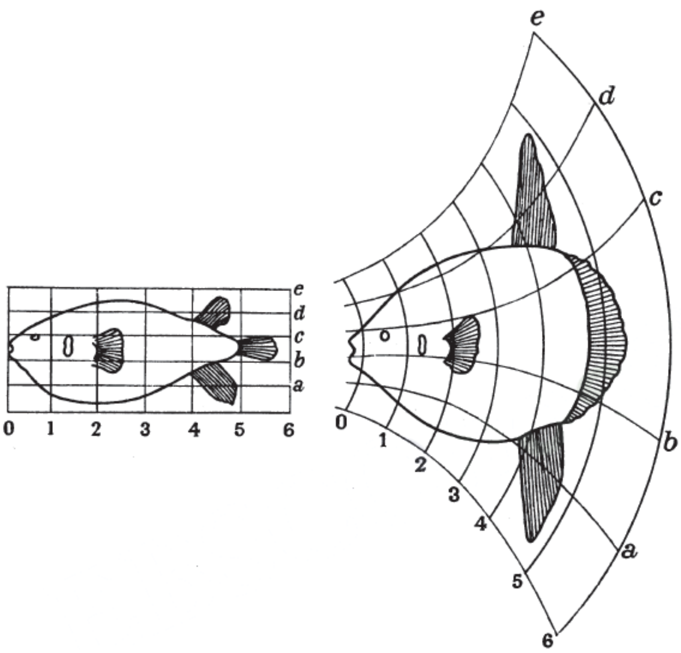


Figure 6.14: left: *Diodon porcupine*, right: *Sunfish*, figures are taken from [67].

He explained that:

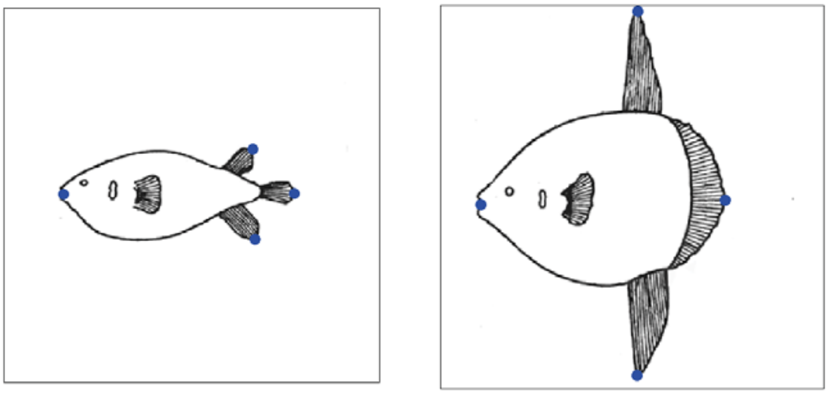
> "In a mathematical sense, it is not a perfectly satisfactory or perfectly regular deformation for the system is no longer isogonal[1]; but nevertheless, it is symmetrical to the eye, and obviously approaches to an isogonal system under certain conditions of friction or constraint." (Page 1064 in [67])

It can be seen in Figure 6.14 that he did not sketch the transformed *Diodon*, but simply showed the grid. This grid resembles the action of $PSL(2, \mathbb{C})$ because vertical lines are mapped to circles; this group is a subgroup of the conformal group.

Therefore, we register *Diodon* with *Sunfish* in $PSL(2, \mathbb{C})$ group, where *Diodon* is taken as the source I_0, and *Sunfish* is taken as the target J. The grid lines are deleted, and the inside of the fishes are filled with black for the registration. However, we found that the registration algorithm did not produce any reasonable result.

In our previous experiments the registration algorithm always succeeded, even when the images were line drawings, see Section 4.6. One reason why registration fails can be

[1]Conformal



(a) *Diodon* (b) *Sunfish*

Figure 6.15: Four corresponding points are selected to calculate the cross-ratios.

	Sunfish	*Diodon*
Cross-ratio	$0.5043 - 0.2179i$	$0.4945 + 0.2714i$

Table 6.6: Cross-ratios of four marked points on fishes, see Figure 6.15.

because there is no transformation in $PSL(2, \mathbb{C})$ that describes a good match between the fishes. To check this, we calculated the cross-ratios. Four corresponding points are marked, see Figure 6.15, and their cross-ratios are calculated. These are given in Table 6.6.

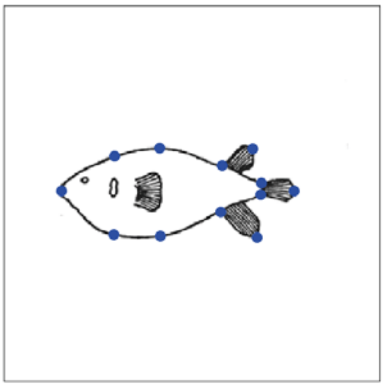
As can be seen in Table 6.6, the cross-ratios of four marked points are not equal. Moreover, we employed a landmark registration between fishes, to provide independent evidence that there is no element of $G_1 = PSL(2, \mathbb{C})$ that gives a good transformation between these fish. Twelve corresponding points are marked in I_0 and J, see Figure 6.16. Let z_1 and w be the marked points in I and J respectively; they are given in Table 6.7.

Transformation φ_1 is obtained from the landmark registration as follows:

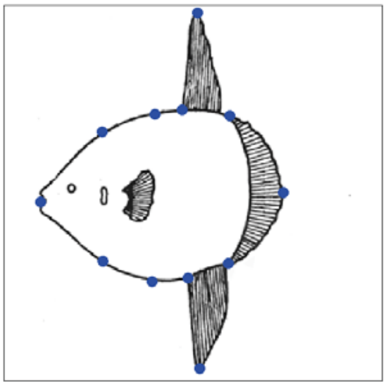
$$\varphi_1(z) = \frac{(1.2298 - 0.01i)z - 0.1177 + 0.0067i}{(-1.3579 - 0.0937i)z + 0.9435 + 0.0092i}.$$

The residual of points before registration is 0.48 and after registration is 0.3357. Figure 6.17a shows marked points z_1, w, and $z_2 = \varphi_1(z_1)$. Also, image I_0 is transformed by φ_1. Figure 6.18b shows I_0 and $I_1 = I_0 \circ \varphi_1^{-1}$. Figure 6.19a shows the discrepancy between I_1 and J.

As can be seen in Figure 6.17a the transformed points do not match, nor are they even close. Also in Figure 6.19a, I_1 is not similar to J at all. Moreover the residuals before



(a) Source

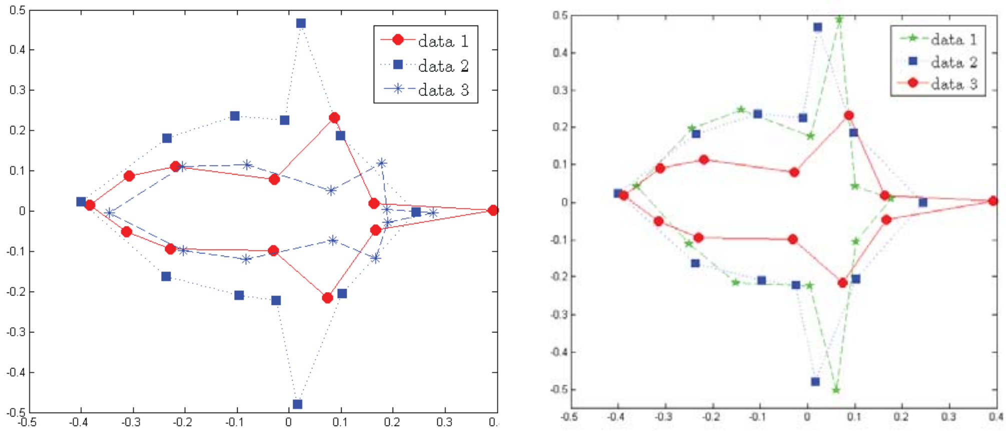


(b) Target

Figure 6.16: Twelve corresponding points are marked in source (I_0) and target(J) for landmark registration.

z_1	w
$(-0.3457, -0.0053)$	$(-0.4002, 0.0239)$
$(-0.2037, -0.0987)$	$-0.2368, -0.1629)$
$(-0.0831, -0.1201)$	$(-0.0967, -0.2096)$
$(0.0842, -0.0734)$	$(-0.0247, -0.2212)$
$(0.1659, -0.1181)$	$(0.0161, -0.48)$
$(0.1893, -0.0267)$	$(0.1017, -0.2057)$
$(0.2768, -0.0053)$	$(0.2438, -0.0014)$
$(0.1873, 0.0045)$	$(0.0978, 0.1873)$
$(0.1776, 0.1192)$	$(0.022, 0.4675)$
$(0.0803, 0.0511)$	$(-0.0092, 0.2262)$
$(-0.0811, 0.1154)$	$(-0.1045, 0.236)$
$(-0.2057, 0.1115)$	$(-0.2348, 0.1815)$
$(-0.3457, -0.0053)$	$(-0.4002, 0.0239)$

Table 6.7: The coordinates of twelve marked points on the source and the target in Figure 6.16 for landmark registration.



(a) (b)

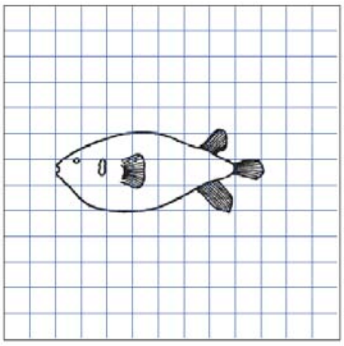
Figure 6.17: (a) Landmark registration of data1 and data2 in $PSL(2,\mathbb{C})$, where data1 (z_1) are the set of marked points on the source (I_0); data2 (w) are the set of marked points on the target (J) (see Figure 6.16) and data3 are the transformation of data1 after registration. (b) Landmark registration of data3 and data2 in $PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$, where data3 ($z_2$) are the transformation of data1 in the left figure; data2 (w) are the set of marked points on the target (J), and data1 are the transformation of data3 after registration.

and after registration do not change too much. Therefore, there does not seem to be any $PSL(2,\mathbb{C})$ transformation between these fish. However, looking at Figure 6.18b, it can be seen that the deformed grid looks similar to what Thompson sketched. The only difference is that Thompson drew the circles from the mouth to the tail of fish evenly, whereas in our grid the distances between the circles increases from the mouth to the tail. It seems that some contraction horizontally might make the grid similar to Thompson's grid and transfer *Diodon* to *Sunfish*. The action of $PSL(2,\mathbb{R})\times PSL(2,\mathbb{R})$ on the plane has this property. Hence, we continue the landmark registration in this group between the transformed points: $z_2 = \varphi_1(z_1)$ and w. We obtained the following transformation φ_2:

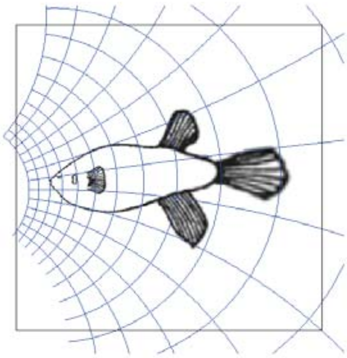
$$\varphi_2(x,y) = \left(\frac{0.7853x + 0.0308}{1.4788x + 1.3313}, \frac{1.4897y + 0.0013}{0.172y + 0.6714}\right).$$

The residual before registration is 0.3357 and after registration is 0.054. This indicates that points are matched well; this can be seen in Figure 6.17b also. Also, image I_1 is transformed by φ_2; see Figure 6.18c. Figure 6.19b shows the discrepancy between transformed I_1 and J.

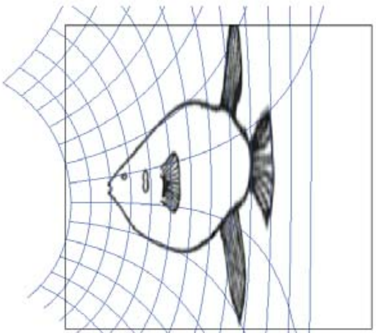
So far, we have multi-registered the *diodon* and *sunfish* using landmark registration.



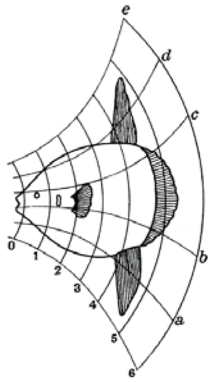
(a) I_0: Source



(b) I_1: Transformed I_0

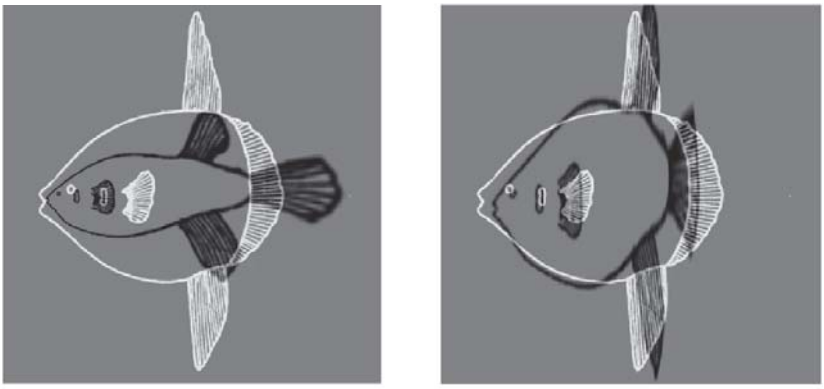


(c) I_2: Transformed I_1



(d) J: Target

Figure 6.18: *Diodon* as source (I_0) is landmark multi-registered with target (J) in $G_1 = PSL(2,\mathbb{C})$ and $G_2 = PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$. (a) Source($I_0$) together with a rectangular grid Ω'. (b) Transformation of source and Ω' to I_1 and Ω'_1 after landmark registration in G_1. (c) Transformation of I_1 and Ω'_1 to I_2 and Ω'_2 after landmark registration in G_2. (d) Target: *Sunfish* along with the deformed grid that Thompson sketched.



(a) (b)

Figure 6.19: (a) Discrepancy between transformed source I_1 and target J after landmark registration in $G_1 = PSL(2,\mathbb{C})$. (b) Discrepancy between I_2 (the transformed I_1) and target J after landmark registration in $G_2 = PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$.

Comparing Figures 6.18c and 6.18d, our deformed grid is very similar to the *Sunfish* that Thompson sketched.

Similarly to the previous examples, we calculate the d_i and L_i, $i = 1, 2$; they are given in Table 6.8.

	$i = 1$	$i = 2$	$i = 3$
$E_i = \|z_i - w\|_2^2$	0.49	0.3357	0.054
d_i	-	1.3427	2.2755
L_i	-	0.3474	0.3398

Table 6.8: The output of multi-registration of marked points on the fishes given in Figure 6.16 in two groups: $G_1 = PSL(2,\mathbb{C})$, $G_2 =$ diffeomorphism.

If $\Psi(x,y) = (\frac{a_1x+b_1}{c_1x+d_1}, \frac{a_2y+b_2}{c_2y+d_2})$ is a transformation in $PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$ then d_2 is calculated as follows. Let $\log(\Psi(x,y)) = (\log(\Psi(x,.)), \Psi(.,y)) = (v_1, v_2)$. Then

$$d_2 = \sqrt{tr(v_1v_1^*) + tr(v_2v_2^*)}.$$

It can be seen in Table 6.8 that d_2 is large, which means that the transformation φ_2 is far from the identity. Also, the L_i show that both groups make equal-sized deformations, therefore both groups play a significant role in the deformation of the fish.

6.3 Conclusion and Future Work

In this chapter we showed various examples of multi-registrations of images. Two different cases of multi-registrations are considered: multi-registration on a chain of groups and multi-registration on a tree of groups.

Multi-registration provides us with lots of useful information. For example, multi-registration of the human skull and chimpanzee skull in $G_1 = PSL(2, \mathbb{C})$ and $G_2 =$ diffeomorphism, showed the significance of $PSL(2, \mathbb{C})$ and insignificance of the diffeomorphism. Moreover, multi-registration on trees of groups provides a new space of transformations which are invertible. That space is composition of groups, for example, multi-registration of *Diodon* and *Sunfish* in two groups: $G_1 = PSL(2, \mathbb{C})$ and $G_2 = PSL(2, \mathbb{R}) \times PSL(2, \mathbb{R})$. The fish are related by a transformation that belongs to a bigger space $G_2 \circ G_1$, which contains the inverses of each product transformation. Without multi-registration, we had to employ a diffeomorphic registration between fishes, which is very complex and an infinite dimensional group. Therefore, multi-registration enables us to find simple and finite dimensional transformations which have inverses. We summarize the benefits of multi-registration:

- Multi-registration provides more information than standard registration, for example the significance or insignificance of a group of transformations.
- Multi-registration provides a bigger space of transformations which is finite dimensional and for which the transformations are invertible.
- Multi-registration enables us to find a transformation between images that can not be obtained by single registration.
- Multi-registration enables us to describe a complex deformation as a composition of simple transformations.
- Multi-registration provide an easier registration process. For example, registration $G_1 \to G_2$ with $G_1 \subset G_2$ can be easier to compute than registration with G_2 alone, as G_1 is smaller and the dimension is less.

In multi-registration the choice of the groups for the registration is important. We choose the group based on the knowledge about the properties of the groups such as invariance. However, we do not have any systematic way to determine them. Therefore, the choice of the groups for multi-registration is an open problem.

Chapter 7

Conclusion

This thesis has revisited Thompson's original idea that global transformations of space can transfer the appearance of one animal to that of another. This has been done in the mathematical context of planar Lie groups, and most notably, the combination of those groups, which we termed *multi-registration*.

While there has been a huge amount of interest in image registration over the past 20 years or so, much of it has been local (non-rigid, often based on diffeomorphisms), and only a small number of finite-dimensional groups have been used, principally similarity, affine, or projective. Instead, we have considered the planar Lie groups, which were classified up to changes of coordinate in [26]. From this we identified three main groups: $PSL(2,\mathbb{C})$, $PSL(3,\mathbb{R})$, and $PSL(2,\mathbb{R}) \times PSL(2,\mathbb{R})$. Together with their subgroups, this gave us a total of 19 groups.

In order to use any of these groups as the transformation set for image registration, we studied the issues of image registration based on the L^2 distance function mathematically, and this enabled us to propose a robust algorithm that avoids some of the common issues of registration algorithms, primarily the presence of many spurious critical points and the lack of continuity in the image. By applying smoothing through Gaussian convolution we were able to remove these issues. However, there are still some open issues for further study in finite-dimensional image registration, including:

- We have assumed that the images tend to a constant value at their edges, i.e., that they are on a plain background. This enables their domain to be extended to $\mathbb{R}^2$ continuously. However, images where this is not the case are common, and we have not studied them at all. Some form of convolution function could be used, and we leave this as future work.

- We have assumed that the images are reasonably smooth, so that simple Gaussian convolution smoothes them well. However, the width of the Gaussian was chosen experimentally, and we have not considered methods by which this could be chosen automatically.
- As the dimensionality of the group increases, so does the chance of spurious local optima appearing. While wider Gaussians can help with this, we have not been able to remove all local optima, and this leads to points where the optimizer can get stuck.

Despite these points, we have demonstrated that our algorithm can be used successfully on a wide variety of images and groups. In particular, we have reproduced many of the examples given by Thompson, and using computational methods we have been able to see where his hypothesized relationships are correct, and where they are not.

These registrations in finite-dimensional groups generally provide less perfect matches than a diffeomorphic image registration with an affine pre-registration (in theory, the only reasons why they would not be is if the diffeomorphic registration got stuck in a local minimum, although this is not quite true in practice, since the diffeomorphic registration is local rather than global). However, they do provide rather more information about the nature of the transformation between the images, as well as simplicity and fast implementation. In the context of model selection, these lower dimensional groups are to be preferred if they provide accurate registrations. Continuing the theme of model selection, we also considered subgroups of any of our groups, in order to find the smallest groups that gave good registrations, reasoning that the fewer the number of parameters, the better.

One benefit of working in Lie groups is the fact that they are also manifolds, and we were able to use this to consider curves between sets of images that were registered together. We demonstrated how to fit such curves in the Lie groups through the transformations that are obtained from the registration. As demonstrations of this idea, we considered two small datasets: a growth curve between five human skulls at different ages, and Thompson's data of three hooved mammal feet. We considered the possibility that such curves could be used to infer other possible points in between the datapoints, or extrapolated to new forms. By using a leave-one-out method we were able to show that the human skull dataset that we had seems to be well fitted by a one-parameter subgroup of $PSL(2, \mathbb{C})$. Unfortunately, the limited size of our datasets made this inconclusive, but we have described the method by which others could study larger datasets relatively easily.

We then moved on to introduce the idea of multi-registration, where the lattice of planar Lie groups suggests possible sequences of groups of transformations. The benefits of multi-registration are that it:

- provides useful information, such as suggesting the significance or insignificance of a group of transformations;
- provides a bigger space of transformations (but still finite dimensional) whose transformations are invertible;
- enables us to find a transformation between images that can not be obtained by single registration;
- enables us to describe a complex deformation as a composition of simple transformations; and
- provides an easier registration process. For example, registration $G_1 \to G_2$ with $G_1 \subset G_2$ can be easier to compute than registration with G_2 alone, as G_1 is lower-dimensional, and so has fewer spurious local minima.

It is clear that the choice of groups for multi-registration is important, and we do not have any systematic way to choose this. It might be the case that invariants of the various groups could be used to highlight possible groups to try; we have used this to check whether groups were correct choices when the registration failed.

In summary, we have provided the methods that will enable the study of related forms using finite-dimensional image registration and demonstrated that, just as Thompson hypothesized, many of the biological forms he looked at do indeed have some relatively simple relationships.

In this thesis we have considered 2D images of 3D objects. The extension to true 3D surfaces would be biologically interesting. While this would have computational challenges, the mathematics should not be radically different.

Bibliography

[1] K. Aho, D. Derryberry, and T. Peterson, *Model selection for ecologists: the worldviews of AIC and BIC*, Ecology, 95 (2014), pp. 631–636.

[2] Y. Amit, *A nonlinear variational problem for image matching*, SIAM Journal on Scientific Computing, 15 (1994), pp. 207–224.

[3] S. Arlot and A. Celisse, *A survey of cross-validation procedures for model selection*, Statistics Surveys, 4 (2010), pp. 40–79.

[4] W. Arthur, *D'Arcy Thompson and the theory of transformations*, Nature Review Genetics, 7 (2006), pp. 401–406.

[5] B. B. Avants, P. T. Schoenemann, and J. C. Gee, *Lagrangian frame diffeomorphic image registration: Morphometric comparison of human and chimpanzee cortex*, Medical Image Analysis, 10 (2006), pp. 397–412.

[6] P. Azad, T. Gockel, and R. Dillmann, *Computer Vision: Principles and Practice*, Elektor International Media, 2008.

[7] R. Bajcsy and C. Broit, *Matching of deformed images*, in Sixth International Conference on Pattern Recognition, 1982, pp. 351–353.

[8] R. Bajcsy, R. Lieberson, and M. Reivich, *A computerized system for the elastic matching of deformed radiographic images to idealized atlas images*, Journal of Computer Assisted Tomography, 7 (1983), pp. 618–625.

[9] M. F. Beg, A. Miller, M. I. Trouvé, and L. Younes, *Computing large deformation metric mappings via geodesic flows of diffeomorphisms*, Computer Vision, 61 (2005), pp. 139–157.

[10] M. Berger, *Geometry I*, Springer Science & Business Media, 2009.

[11] F. L. Bookstein, *Morphometric Tools for Landmark Data: Geometry and Biology*, Cambridge University Press, 1997.

[12] W. M. Boothby, *An Introduction to Differentiable Manifolds and Riemannian Geometry*, Elsevier Science, 1975.

[13] M. A. Branch, T. F. Coleman, and Y. Li, *A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems*, SIAM Journal on Scientific Computing, 21 (1999), pp. 1–23.

[14] C. Broit, *Optimal Registration of Deformed Images*, PhD thesis, Pennsylvania University, 1981.

[15] L. G. Brown, *A survey of image registration techniques*, ACM Computing Surveys (CSUR), 24 (1992), pp. 325–376.

[16] K. P. Burnham and D. R. Anderson, *Model Selection and Multimodel Inference, A Practical Information-Theoretic Approach*, Springer Science & Business Media, 2013.

[17] R. H. Byrd, R. B. Schnabel, and G. A. Shultz, *Approximate solution of the trust region problem by minimization over two-dimensional subspaces*, Mathematical Programming, 40 (1988), pp. 247–263.

[18] S. Cao, J. Jiang, G. Zhang, and Y. Yuan, *An edge-based scale-and affine-invariant algorithm for remote sensing image registration*, International Journal of Remote Sensing, 34 (2013), pp. 2301–2326.

[19] G. E. Christensen, R. D. Rabbitt, and M. I. Miller, *Deformable templates using large deformation kinematics*, IEEE Transactions on Image Processing, 5 (1996), pp. 1435–1447.

[20] T. T. Dieck, *Transformation Groups and Representation Theory*, Springer, 2006.

[21] B. Fischer and J. Modersitzki, *Ill-posed medicine-an introduction to image registration*, Inverse Problems, 24 (2008), p. 034008.

[22] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Pearson Education Limited, 2015.

[23] J. Fraleigh and V. Katz, *A First Course in Abstract Algebra*, Addison-Wesley, 2003.

[24] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, 1981.

[25] R. Gilmore, *Lie Groups, Lie Algebras, and Some of Their Applications*, Krieger Publishing Company, 1994.

[26] A. González-López, N. Kamran, and P. J. Olver, *Lie algebras of vector fields in the real plane*, Proceedings of the London Mathematical Society, s3-64 (1992), pp. 339–368.

[27] A. A. Goshtasby, *2-D and 3-D Image Registration: for Medical, Remote Sensing, and Industrial Applications*, John Wiley & Sons, 2005.

[28] U. Grenander and M. I. Miller, *Pattern Theory: From Representation to Inference*, Oxford University Press, 2007.

[29] U. Grenander, A. Srivastava, and S. Saini, *A pattern-theoretic characterization of biological growth*, IEEE Transactions on Medical Imaging, 26 (2007), pp. 648–659.

[30] S. Helgason, *Differential Geometry, Lie Groups, and Symmetric Spaces*, Academic Press, 1979.

[31] B. Horn, *Robot Vision*, MIT Press, 1986.

[32] 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, 17 (2002), pp. 825–841.

[33] J. R. Jensen, *Introductory Digital Image Processing: A Remote Sensing Perspective*, Pearson Education, 2015.

[34] C. T. Kelley, *Iterative Methods for Optimization*, SIAM, 1999.

[35] D. G. Kendall, *Shape manifolds, Procrustean metrics, and complex projective spaces*, Bulletin of the London Mathematical Society, 16 (1984), pp. 81–121.

[36] J. M. Lee, *Introduction to Smooth Manifolds*, Springer, 2003.

[37] S. R. Lele and J. T. Richtsmeier, *An Invariant Approach to Statistical Analysis of Shapes*, CRC Press, 2001.

[38] S. Lipschutz, *Schaum's Outline of Theory and Problems of General Topology*, Schaum Pub. Co., 1965.

[39] M. Lorenzi and X. Pennec, *Geodesics, parallel transport & one-parameter subgroups for diffeomorphic image registration*, International Journal of Computer Vision, 105 (2013), pp. 111–127.

[40] J. A. Maintz and M. A. Viergever, *A survey of medical image registration*, Medical Image Analysis, 2 (1998), pp. 1–36.

[41] V. Mani and D. Rivazhagan, *Survey of medical image registration*, Journal of Biomedical Engineering and Technology, 1 (2013), pp. 8–25.

[42] S. Marsland, R. I. McLachlan, K. Modin, and M. Perlmutter, *Geodesic warps by conformal mappings*, International Journal of Computer Vision, 105 (2013), pp. 144–154.

[43] MathWorks, 2016. `http://www.mathworks.com/products/matlab`.

[44] MathWorks, *Gradient*, 2016. `http://au.mathworks.com/help/matlab/ref/gradient.html`.

[45] MathWorks, *lsqnonlin*, 2016. `http://au.mathworks.com/help/optim/ug/lsqnonlin.html`.

[46] T. Mcinerney and D. Terzopoulos, *Deformable models in medical image analysis: a survey*, Medical Image Analysis, 1 (1996), pp. 91–108.

[47] J. Milnor, *Geometry of growth and form: Commentary on D'Arcy Thompson*, 2010. A lecture at Stony Brook University, on the book On Growth and Form by D'Arcy Thompson,`https://video.ias.edu/milnor-80th`.

[48] J. Modersitzki, *Numerical Methods for Image Registration*, Oxford University Press, 2004.

[49] J. Modersitzki, *Fair: Flexible Algorithms for Image Registration*, SIAM, 2009.

[50] National Oceanic and Atmospheric Adminstration (NOAA), 2016. `http://www.photolib.noaa.gov/htmls/figb0616.htm`.

[51] J. Nocedal and S. Wright, *Numerical Optimization*, Springer Science & Business Media, 2006.

[52] P. Olver, *Equivalence, Invariants and Symmetry*, Cambridge University Press, 1995.

[53] S. V. Petukhov, *Non-Euclidean geometries and algorithms of living bodies*, Computers & Mathematics with Applications, 17 (1989), pp. 505–534.

[54] K. Ramachandran and C. Tsokos, *Mathematical Statistics with Applications*, Elsevier Science, 2009.

[55] K. Rohr, *Landmark-Based Image Analysis: Using Geometric and Intensity Models*, Springer Science & Business Media, 2001.

[56] H. Rosenbrock, *An automatic method for finding the greatest or least value of a function*, The Computer Journal, 3 (1960), pp. 175–184.

[57] W. Rudin, *Functional Analysis*, McGraw-Hill, 1991.

[58] M. Shah and R. Kumar, *Video Registration*, Springer US, 2013.

[59] C. E. Shannon and W. Weaver, *The Mathematical Theory of Information*, University of Illinois Press, 1949.

[60] D. Simovici and C. Djeraba, *Mathematical Tools for Data Mining: Set Theory, Partial Orders, Combinatorics*, Springer Science & Business Media, 2014.

[61] I. Singer and J. Thorpe, *Lecture Notes on Elementary Topology and Geometry*, Springer New York, 1976.

[62] S. Sommer, M. Nielsen, F. Lauze, and X. Pennec, *A multi-scale kernel bundle for lddmm: towards sparse deformation description across space and scales*, in Biennial International Conference on Information Processing in Medical Imaging, 2011, pp. 624–635.

[63] A. Sotiras, C. Davatzikos, and N. Paragios, *Deformable medical image registration: A survey*, IEEE Transactions on Medical Imageing, 32 (2013), pp. 1153–1190.

[64] J. Stillwell, *Naive Lie Theory*, Springer Science & Business Media, 2008.

[65] M. R. Stytz, G. Frieder, and O. Frieder, *Three-dimensional medical imaging: algorithms and computer systems*, ACM Computing Surveys (CSUR), 23 (1991), pp. 421–499.

[66] I. L. Thomas, V. M. Benning, and N. P. Ching, *Classification of remotely sensed images*, Geocarto International, 2 (1987), pp. 77–77.

[67] D. W. Thompson, *On Growth and Form*, Cambridge Univ. Press, 2nd ed., 1942.

[68] J. Turski, *Harmonic analysis on* $sl(2,\mathbb{C})$ *and projectively adapted pattern representation and projectively adapted pattern representation*, Journal of Fourier Analysis and Applications, 4 (1998), pp. 67–91.

[69] J. Turski, *Geometric Fourier analysis of the conformal camera for active vision*, SIAM Review, 46 (2004), pp. 230–255.

[70] P. A. Van den Elsen, E. J. Pol, and M. A. Viergever, *Medical image matching-a review with classification*, Engineering in Medicine and Biology Magazine, IEEE, 12 (1993), pp. 26–39.

[71] L. Velho, A. C. Frery, and J. Gomes, *Image Processing for Computer Graphics and Vision*, Springer London, 2009.

[72] P. Viola and W. M. Wells III, *Alignment by maximization of mutual information*, International Journal of Computer Vision, 24 (1997), pp. 137–154.

[73] Wikipedia, *Affine transformations — Wikipedia and the free encyclopedia*, 2016. `https://en.wikipedia.org/wiki/File:Affine_transformations.ogv`.

[74] G. Wolberg, *Digital Image Warping*, IEEE Computer Society Press, 1990.

[75] L. Younes, *Shape and Diffeomorphisms*, Springer, 2010.

[76] B. Zitova and J. Flusser, *Image registration methods: a survey*, Image and Vision Computing, 21 (2003), pp. 977 – 1000.