Robust and Resistant 2D Shape alignment

Rasmus Larsen & Hrafnkell Eiriksson
Informatics and Mathematical Modelling
Technical University of Denmark, DK–2800 Kgs. Lyngby, Denmark
rl@imm.dtu.dk

Technical report 17/2001

Abstract

We express the alignment of 2D shapes as the minimization of the norm of a linear vector function. The minimization is done in the l_1, l_2 and the l_∞ norms using well known standard numerical methods. In particular, the l_1 and the l_∞ norm alignments are formulated as linear programming problems. The linear vector function formulation along with the different norms results in alignment methods that are both resistant from influence from outliers, robust wrt. errors in the annotation and capable of handling missing datapoints. Another reason for using other norms than the l_2 norm is to minimize the effect of the choice of landmarks. Examples that illustrate the properties of the different norms are given on simulated as well as real datasets.

1. Introduction

The study of the geometry of classes of biological objects represented by sets of (corresponding) landmark points is usually concerned with the variation up to a given transformation. The most common set of transformations applied are the Euclidean similarity transformations, i.e. translation, rotation, and isotropical scale. Following [1] we will denote the geometrical information up to a Euclidean similarity transformation the shape of an object. Analysis of the shape variablity across a set of examples requires the alignment of the shapes to a common frame of reference. This is usually obtained by means of a generalized Procrustes analysis (GPA) [2, 3]. The GPA consists of minimizing the sum of squared distances between corresponding landmarks on all examples and a reference shape with respect to the reference shape and similarity transformations of all example shapes, i.e. GPA is a least squares estimator.

Several problems may occur when aligning a set of shapes. Methods for extracting landmarks - be they manual, semi-automated, or fully automated - may result in missing points on some shapes, landmark outliers, and even errors in the correspondence between landmarks. Furthermore, object outliers may also occur.

Least squares methods are not particularly good at handling these situations. Other alignment procedures that handle these problems are therefore necessary. The insensitivity to outliers - landmark or object - is usually referred to as resistance, and insensitivity to correspondence errors, i.e. model breakdown, is called robustness [1].

Other work on robust and resistant alignment has been done. Siegel [4, 5] used double repeated medians to achieve robust alignment in order to study the differences in shapes. Dryden and Walker [6] used S-estimators.

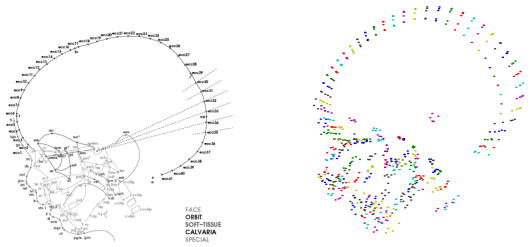
Here we present algorithms to align shapes based on well understood and efficiently implemented numerical methods. They align shapes in the sense of l_1, l_2 and l_∞ norms. The formulation presented allows for dealing with missing observations and some of the norms are robust when dealing with errors. The real power of the methods comes from using standard well understood algorithms.

2 Data – Profiles of crania

A common source of images for medical diagnostics is radiographs. Biological objects seen in radiographs are also an important and interesting class of objects in medical image analysis.

Fine details are difficult to discern in radiographs. Images from radiographs usually have low contrast so it is often difficult do precisely mark the boundaries of objects. Another source of ambiguity for radiographs stems from projecting three dimensional structures down onto a two dimensional image. This can result in an overlap of objects in the radiograph. All this leads to annotation errors. It is important to minimize the annotation errors as much as possible so that they do not obscure the real variations in the shapes.

A dataset consisting of annotated radiographs showing the crania of 2 month old children with cleft lip and palate was supplied by M.Sc. Tron Darvann and Dr. Nuno Vibe Hermann of the 3D Laboratory, School of Dentistry, University of Copenhagen. It contains 138 landmarks taken



(a) (b)

Figure 1: (a) An illustration of where the points were annotated on the crania data set (see page 31 in [7]). (b) Five typical profiles of a cranium superimposed. .

from profile radiographs of the heads of 2 months old children. The radiographs were obtained in a clinical context for patient treatment purposes. The landmarks were placed as part of a study on craniofacial morphology and growth in children with cleft lip and palate [7]. In Figure 1 the location of landmarks and a few typical example shapes are shown. The subset of the dataset considered here contains 48 shapes.

3 The alignment task

Let there be given L training examples for a given shape class, and let each example be represented by a set of k 2D landmark points (x_{ij}, y_{ij}), $i = 1, \ldots, L$ and $j = 1, \ldots, k$. Then each example is given by a $2n$ vector

$$\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ik}, y_{i1}, \ldots, y_{ik})^T. \tag{1}$$

The alignment problem in 2D consists of estimating an average shape, $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_k, \nu_1, \ldots, \nu_k)^T$, and a set of Euclidean similarity parameters parameters for each shape. Let these parameters be

$$\begin{aligned} \text{scale:} \quad & \beta_i \in R_+ \\ \text{rotation:} \quad & \Gamma_i \in \mathrm{SO}(2) = \begin{bmatrix} \cos\psi_i & \sin\psi_i \\ -\sin\psi_i & \cos\psi_i \end{bmatrix} \\ \text{translation:} \quad & \boldsymbol{\gamma}_i = (\gamma_{xi}, \gamma_{yi})^T \in R^2 \end{aligned}$$

Then using a multiple linear regression formulation as described in [3] the alignment problem consists of the minimisation of a vector function

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{\mu} - \boldsymbol{Z}_1\boldsymbol{\Theta}_1 \\ \boldsymbol{\mu} - \boldsymbol{Z}_2\boldsymbol{\Theta}_2 \\ \vdots \\ \boldsymbol{\mu} - \boldsymbol{Z}_L\boldsymbol{\Theta}_L \end{bmatrix}$$

where $\boldsymbol{\Theta}_i = [\beta_i \cos\psi_i, \beta_i \sin\psi_i, \gamma_{xi}, \gamma_{yi}]^T$ and

$$\boldsymbol{Z}_i = \begin{bmatrix} x_{i1} & -y_{i1} & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ x_{ik} & -y_{ik} & 1 & 0 \\ y_{i1} & x_{i1} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ y_{ik} & x_{ik} & 0 & 1 \end{bmatrix}$$

wrt. all $\boldsymbol{\Theta}_i$ and $\boldsymbol{\mu}$. The ith aligned shape is given by $\boldsymbol{Z}_i\boldsymbol{\Theta}_i$. Rewriting $\mathbf{F}$, it is obvious, that it is linear in the parameters, $\phi = [\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \ldots, \boldsymbol{\Theta}_L, \boldsymbol{\mu}]^T$,

$$\mathbf{F}(\phi) = \begin{bmatrix} \mathbf{Z}_1 & \mathbf{0} & \ldots & \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & \mathbf{Z}_2 & \ldots & \mathbf{0} & -\mathbf{I} \\ \vdots & & \ddots & & \vdots \\ \mathbf{0} & \ldots & \ldots & \mathbf{Z}_L & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Theta}_1 \\ \boldsymbol{\Theta}_2 \\ \vdots \\ \boldsymbol{\Theta}_L \\ \mu \end{bmatrix} \tag{2}$$

An obvious choice of parameters that minimizes any norm of $\mathbf{F}$ is to choose $\phi = \mathbf{0}$, that is, collapse everything to a point. The minimization needs to be constrained to not allow such degenerate solutions. One way of doing that is to specify that the meanshape should be aligned to an arbitrary shape (say the last) from the set of shapes. This enters the alignment vector function linearly

$$\mathbf{F}(\phi) = \begin{bmatrix} \mathbf{Z}_1 & \mathbf{0} & \ldots & \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & \mathbf{Z}_2 & \ldots & \mathbf{0} & -\mathbf{I} \\ \vdots & & \ddots & & \vdots \\ \mathbf{0} & \ldots & \ldots & \mathbf{Z}_L & -\mathbf{I} \\ \mathbf{0} & \ldots & \ldots & \ldots & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Theta}_1 \\ \boldsymbol{\Theta}_2 \\ \vdots \\ \boldsymbol{\Theta}_L \\ \mu \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{z}_L \end{bmatrix} \tag{3}$$

The main advantage of using the linear vector function formulation is that it is easy to deal with missing observations, that is, missing points in shapes. This is done just by leaving out the rows from the coefficient matrix that correspond to the missing points.

3.1 Aligning in different norms

The three most common norms used in numerical analysis are the l_2, l_1 and l_∞ norms. The l_2 norm is the generalization of the Euclidian size and is the sum of squares of the elements of a vector.

$$\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{N} x_i^2} \tag{4}$$

Besides being the generalization of what is commonly understood as size and length calculating with it is generally easy (compared to other norms). It is also known as the least squares norm.

The l_1 norm is the sum of the absolute values of the elements of the vector.

$$\|\mathbf{x}\|_1 = \sum_{i=1}^{N} |x_i| \tag{5}$$

It is insensitive to errors and outliers and is often the norm of choice if a data set is believed to be corrupted by errors or contain outliers.

The l_∞ norm is the value of the largest absolute value of the vector.

$$\|\mathbf{x}\|_\infty = \max_i |x_i| \tag{6}$$

It is the opposite of the l_1 norm as it is very sensitive to the individual elements. It is often used when the size of the individual elements of a vector must be uniform in size. The l_∞ norm is also known as the minimax norm because minimizing in it corresponds to minimizing the maximum of the vector.

All these norms can be used to align a set of shapes using the linear vector function formulation from eq. (2) and (3).

3.1.1 The l_2 norm

As has been stated earlier, the l_2 norm alignment is the well known Procrustes alignment. It is usually performed by either using an iterative algorithm proposed in [8] or by calculating the mean shape by solving an eigenvector problem and then aligning each shape to the mean shape using singular value decomposition [1].

It can of course also be done by minimizing the l_2 norm of the function $\mathbf{F}$ in eq. (3). Call the coefficient matrix $\mathbf{C}$, the vector with the parameters ϕ and the vector with the last shape $\mathbf{d}$, that is

$$\mathbf{F}(\phi) = \mathbf{C}\phi + \mathbf{d}. \tag{7}$$

The l_2 norm of $\mathbf{F}$ can be minimized using QR-Factorization of the coefficient matrix $\mathbf{C}$ and then using back-substitution with the $\mathbf{R}$ matrix to get the desired least squares solution. Missing points are handled by leaving out the corresponding rows of the coeeficient matrix, $\boldsymbol{C}$, and vector, $\boldsymbol{d}$.

3.1.2 The l_∞ norm

Here we wish to make the deviation of the shapes from the mean shape uniform by minimizing the maximum deviation, that is

$$\min_\phi \|\mathbf{F}(\phi)\|_\infty = \min_\phi \max |\mathbf{F}(\phi)| = \min_\phi \max |\mathbf{C}\phi + \mathbf{d}| \tag{8}$$

By casting this minimization task into the framework of linear programming [9], see App. A, it can be easily solved. The task is now to find the smallest number o that bounds all of the elements in $\mathbf{F}(\phi) = [F_1(\phi)\ F_2(\phi)\ \ldots\ F_{2N(L+1)}(\phi)]^T$ such that

$$o \geq |F_m(\phi)|, m = 1 \ldots 2N(L+1). \tag{9}$$

or equivalently

$$-o \leq F_m(\phi) \leq o, m = 1 \ldots 2N(L+1). \tag{10}$$

This can be seen as the linear constraints to the linear programming task of minimizing o, that is finding the smallest number that bounds all the elements in $\mathbf{F}(\phi)$.

The coefficient matrix is very sparse and has only 4 entries at most in each row, most of the rows have only 3 entries.

For each point we get four constraints

$$-o \leq x_{ij}(\beta_i \cos\psi_i) - y_{ij}(\beta_i \sin\psi_i) + \gamma_{xi} - \mu_j \leq o \tag{11}$$

$$-o \leq y_{ij}(\beta_i \cos\psi_i) + x_{ij}(\beta_i \sin\psi_i) + \gamma_{yi} - \nu_j \leq o \tag{12}$$

where i is the number of a shape (corresponds to a block in $\mathbf{C}$) and j is the number of the point within the shape. Because one shape (say the last one) is used to lock the pose a few extra constraints are needed

$$-o \leq x_{Lj} - \mu_j \leq o \tag{13}$$

$$-o \leq y_{Lj} - \nu_j \leq o \tag{14}$$

The constraints are linear in the parameters $\beta_i \cos\psi_i$, $\beta_i \sin\psi_i$, γ_{xi}, γ_{yi}, μ_j and ν_j.

If a point (or points) on a shape are missing then the constraints are just not included in the linear program.

3.1.3 The l_1 norm

If there are reasons to believe that the annotations of the shapes being aligned contain errors (e.g. errors in correspondence between points in different shapes) or outliers the l_1 norm should be used.

The task can again be formulated as a linear programming task. The task is to find a vector $\mathbf{o} = [o_1\ o_2\ \ldots\ o_{2NL}]$ whose elements bound the values of the alignment function, that is

$$o_m \geq |F_m(\phi)|, m = 1 \ldots 2NL. \tag{15}$$

or equivalently

$$-o_m \leq F_m(\phi) \leq o_m, m = 1 \ldots 2NL. \tag{16}$$

This can be stated as linear constraints

$$-o_{xij} \leq x_{ij}(\beta_i \cos\psi_i) - y_{ij}(\beta_i \sin\psi_i) + \gamma_{xi} - \mu_j \leq o_{xij} \quad (17)$$

$$-o_{yij} \leq y_{ij}(\beta_i \cos\psi_i) + x_{ij}(\beta_i \sin\psi_i) + \gamma_{yi} + \nu_j \leq o_{yij}. \quad (18)$$

The objective function of the linear programming task is to find a vector $\mathbf{o}$ that contains bounds for each element of $\mathbf{F}(\phi)$ such that

$$\sum_i o_i. \quad (19)$$

is minimized. The values of o_i take on the role of the absolute values of the residuals in each coordinate.

If there are many shapes being aligned there is still a possibility for degenerate solutions, even though a shape is included to lock the pose. This is because the l_1 alignment uses as an object function the accumulated size of the residuals from the shapes to the estimated mean shape. If the shapes are many it might be "cheaper" to collapse all the shapes to a point and estimate the mean shape as the same point. That way the residual is 0 from all the shapes except one. The only contribution to the l_1 norm then comes from the difference from the mean shape (that is now a point) and the shape used to lock the pose of the alignment. Other constraints are needed. One possibility is to constrain the average rotation and position, that is

$$\sum_{i=1}^{L} \beta_i \cos\psi_i = K_1 \quad (20)$$

$$\sum_{i=1}^{L} \beta_i \sin\psi_i = K_2 \quad (21)$$

$$\sum_{i=1}^{L} \gamma_{xi} = 0 \quad (22)$$

$$\sum_{i=1}^{L} \gamma_{yi} = 0 \quad (23)$$

where K_1 and K_2 are arbitrary constants. e.g. $K_2 = L$ and $K_2 = 0$. This has the advantage of allowing the specification of overall scale, orientation and position. This should of course be used together with Eq. (2).

The reason for the possibility of a collapsed solution is the linear relation between the size of the residuals and the size of the l_1 norm. In the l_2 norm this does not happen as many small residuals are favored over few larger because of the quadratic relation. If the accumulated contribution to the l_2 norm from the residuals suggests that it would be beneficial to collapse to a point in the l_2 norm it is sufficient to scale the shapes a bit (done automatically by the algorithm) to change the ratio between the residuals when collapsing and when estimating a proper shape. This ratio stays constant in the l_1 norm when the overall scale is changed.

If a point (or points) on a shape are missing then the constraints are just not included in the linear program as in the l_∞ alignment.

4 Repeated medians alignment

For comparison the repeated medians alignment algorithm is described here. It was described by Siegel and Benson [4]. Its main use is to detect localized differences in shapes. The least squares methods allows large, but local, differences in shapes to have big influence and thereby obscure it, that is, least squares is not resistant to outliers.

The algorithm estimates the similarity transformation parameters successively. It starts by estimating the scale β for each shape, then finds the rotation ψ and finally finds the translation γ_x and γ_x.

The algorithm aligns two shapes, say $\mathbf{z}$ and $\mathbf{z}^+$. For each pair of homologous points (point i and j for example) on the shapes a scale can be estimated

$$\beta_{ij} = \frac{\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}}{\sqrt{(x_j^+ - x_i^+)^2 + (y_j^+ - y_i^+)^2}}. \quad (24)$$

Taking the double repeated median of the β_{ij} values gives an estimate for scale

$$\hat{\beta} = \underset{i}{\text{median}}(\underset{i \neq j}{\text{median}}\beta_{ij}). \quad (25)$$

The rotation between the shapes can be estimated in a similar manner. The value of ψ_{ij} expresses the rotation needed to cause the ray from point i to j in shape $\mathbf{z}$ to point in the same direction as the corresponding ray in $\mathbf{z}^+$. The rotation estimate is then

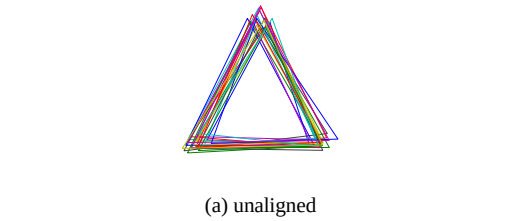
$$\hat{\psi} = \underset{i}{\text{median}}(\underset{i \neq j}{\text{median}}\psi_{ij}). \quad (26)$$

Translation can be defined for a point, pairs of points are not needed. Therefore the translation parameters γ_x and γ_y are estimated using simple (nonrepeted) medians

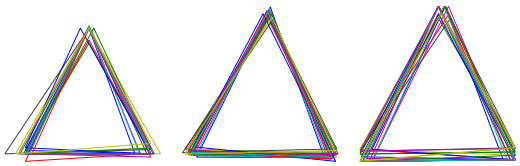
$$\hat{\gamma_x} = \underset{i}{\text{median}}[x_i - \hat{\beta}(x_i' \cos\hat{\psi} - y_i' \sin\hat{\psi})] \quad (27)$$

$$\hat{\gamma_y} = \underset{i}{\text{median}}[y_i - \hat{\beta}(y_i' \cos\hat{\psi} + x_i' \sin\hat{\psi})] \quad (28)$$

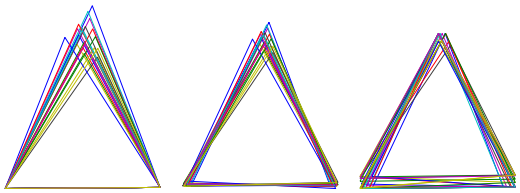
A set of shapes can be aligned using the iterative algorithm shown in [8] where the above repeated median method is used to align the set to the current guess at the mean shape. A 3D variant of the algorithm is demonstrated in [10, p179]



(a) unaligned



(b) l_1 norm (c) l_2 norm (d) l_∞ norm



(e) l_1 norm (f) l_2 norm (g) l_∞ norm

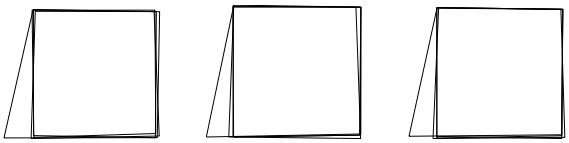
Figure 2: (a) 10 triangles. (b-d) alignment based on the 3 corner points using the l_1, l_2 and l_∞ norms, respectively. (e-g) aligned as (b-d) but with an addtional 18 landmarks distributed equidistantly on the lower side included.

5 Examples

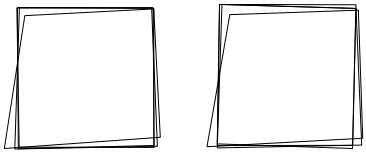
5.1 Triangles and squares

The choice of landmarks can have high influence on the result of the alignment of a set of shapes. This is illustrated in Figure 2 where the alignment of triangles with different number of landmarks is compared. The triangles were generated by creating a reference triangle and then adding i.i.d. Gaussian noise to each corner. They were then aligned in the three norms. Then between the bottom two points of the triangles additional 18 points were added by linear interpolation. The triangles were then realigned. The extra 18 points are added to illustrate and study the influence of different annotations of the same shapes.

We see that the l_2 norm alignment is sensitive to the chosen representation, whereas the l_∞ alignment is insensitive. Th l_1 alignment regards the top corner point as an outlier and disregards thereby achieving perfect alignment of the the lower sides.



(a) unaligned (b) l_1 (c) repeated medians



(d) l_2 (e) l_∞

Figure 3: A simulated dataset consisting of four squares. To one of the squares a large error has been added (shown in a)). The shapes were then aligned using different alignment methods to compare how visible the outlier was. As can be seen, the l_1 alignment and repeated medians method are least sensitive to the outlier and show it quite well. The l_2 and l_∞ try its best to smooth out the difference.

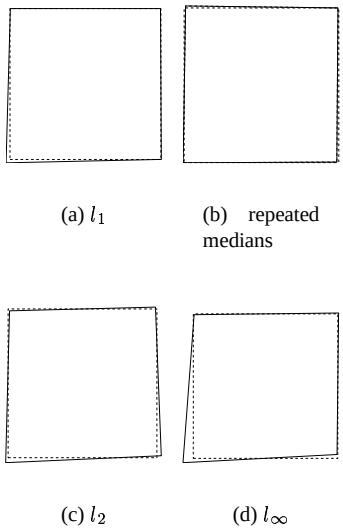
In some applications of shape alignment it is important that the real shape variations are not obscured [4]. An outlier point in the dataset can have big influence on the alignment and the estimation of the mean shape. To illustrate this, four squares were created in the same way the triangles were. A large deviation was then added to the bottom left corner of one of the squares to create an outlier shape. See Figure 3 for the unaligned set and the different alignment results.

The l_1 and repeated medians perform best when trying to find the real shape variations and both estimate a mean shape that is quite close to the true mean shape (the reference shape the squares were created from), see Figure 4.

5.2 Profiles of crania

Figure 1 shows a few profiles of the crania of 2 months old infants. The landmarks were annotated from radiographs in order to study the morphology and growth of infants with cleft lip and palate [7]. All the points have an anatomical meaning. The whole set contains 48 shapes. The shapes are fairly regular (no obvious outliers) but some of them have missing points. Two ways of dealing with the missing points using the different norms are demonstrated.

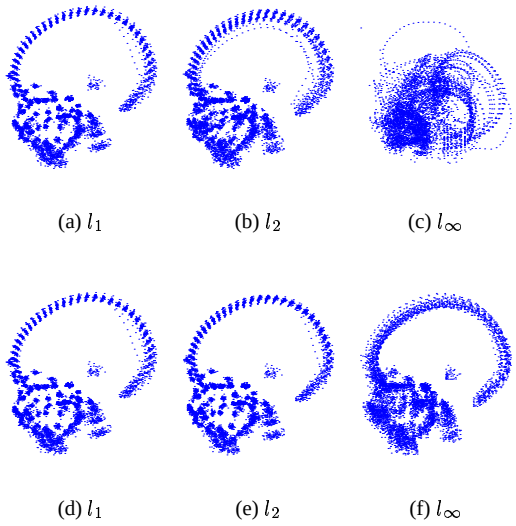
The missing points in the dataset have been given the value $(0, 0)$. A much better and educated guesses about the true value of the missing points can be done but the purpose



(a) l_1 (b) repeated medians

(c) l_2 (d) l_∞

Figure 4: Comparison of the estimated mean shapes (solid) to the original reference shape (dashed). The repeated medians and l_1 alignment methods estimate the best mean shapes. Notice the influence on the commonly used l_2 alignment method.



(a) l_1 (b) l_2 (c) l_∞

(d) l_1 (e) l_2 (f) l_∞

Figure 5: (a-c) alignment of shapes with errors (that is missing points set to $(0,0)$. (d-f) alignment when the missing points are left out of the minimization. The influence of the error is greatest in the l_∞ case and least in the l_1 case. Notice how stable the l_1 alignment is in the presence of errors and the hypersensitivity of the l_∞ alignment to errors. Obviously the l_2 norm is not resistant to the errors.

is to show the influence of having some points with wrong values when aligning. $(0,0)$ is just as good error as any other. This is the first way to deal with the missing point. The other way is to just leave out the rows corresponding to those missing coordinates in the coefficient matrix of the alignment function. The results of aligning in the different norms are shown in Figure 5

6 Conclusion

The alignment of shapes into a common reference frame is a fundamental task in shape modelling. A natural extension to the general Procrustes alignment usually employed is to use other norms than the l_2 norm. Here the l_1 norm and the l_2 norm have been explored.

One of the main attractions of the formulation of the alignment problem as set forth here is the capability to work in the presence of missing observations, no matter which norm is used.

The different norms also have some nice properties. The l_1 norm is robust when dealing with point outliers and errors. Such errors might stem from wrong correspondence or a severe error in annotation of a landmark. It also shows resistance against outlier shapes, though in a different manner than the repeated median method. The main property of the l_∞ norm is how insensitive it is to different annotations of the same shapes.

The alignment task has been formulated as a linear function. This allows the usage of standard, well understood numerical methods such as linear programming. Lots of research has been done in the field of linear programming and the properties of the algorithms used to solve linear programming tasks, such as convergence, are well understood. The coefficient matrix is very structured and sparse. That can be exploited when implementing the algorithms to speed up the alignment process.

7 Discussion

When solving linear programming problems with the Simplex algorithm the algorithm does not only yield the solution but also a sensitivity analysis. For each constraint the Simplex algorithm returns its so called marginal value (or shadow price). It expresses how much the value of the objective function would change if that constraint was moved. If the marginal value for a constraint is 0 it expresses that the constraint is not active (i.e. it does not limit the solution). This gives rise to an interesting interpretation for the minimax alignment. The constraints with non-zero marginal values tell which points in which shape are dominating the alignment. This might be used for outlier detection.

The mean shape μ estimated simultaneously with the

alignment is the shape that all the shapes in a given shape set deviate the least from. In the l_2 case this equals the mean of the point distribution for each point. This is also the case in the l_1 case unless there are some outliers, they do not affect the estimate of the mean shape μ (which is a good thing!). For the l_∞ alignment the estimated shape μ is not necessarily the shape that is the mean of the point distribution for each point. It is just the shape that minimizes the maximum deviation from it in each point.

A Linear programming

Linear programming [9] – programming should be understood here as planning – is an optimization technique mainly studied in operations research. It is a way of formulating and solving optimization problems where the goal is to minimize or maximize a function that is a linear combination of the unknown variables subject to constraints that also are linear. One way of formulating such problems in terms of matrices and vectors is

$$\text{Solve } \mathbf{A}\mathbf{x} \geq \mathbf{v}_1 \tag{29}$$

$$\text{with } \mathbf{v}_2^T\mathbf{x} \text{ minimal.} \tag{30}$$

Note that a constraint on the form $C \geq D$ can always be converted into one on the form $E \leq F$ or even $G = H$ by adding an extra variable. All the following constraints are equivalent in linear programming

$$x \geq y \tag{31}$$

$$-x \leq -y \tag{32}$$

$$x + s = y. \tag{33}$$

In the last case the extra variable s (called slack variable) has been added. It is what must be added or subtracted from x to convert a constraint like the first two to a constraint of the last type.

A good way to illustrate linear programming is to look at a problem in two dimensions (see figure 6). Consider a problem where a and b are two real numbers that need to be determined such that an objective is minimized. Linear constraints are placed on a and b. Each constraint in 2D can be interpreted as a line. On one side of the line are values of a and b that fulfill the constraint. All the constraints together then define an area in a, b-space that contains values that fulfill all the constraints. This area is called the feasible region. The task is then to find the value in the feasible region that minimizes the objective. As the objective is a linear combination $o = c_2a + c_1b + c_0$ of the variables it too can be interpreted as a line in a, b-space. By changing the value of o the line can be moved along its normal vector. Moving it to the intersection of constraint lines where the value of o is smallest is the solution.

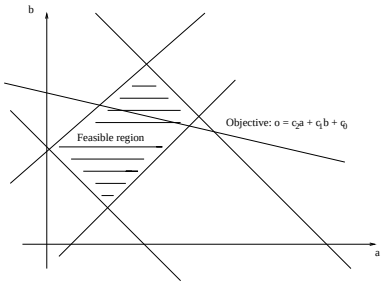


Figure 6: A linear programming problem in 2D. The unknowns are a and b. The objective function corresponds to the thick line and the constraints correspond to the thinner lines. The constraints define the feasible region in a, b-space that contains values that fulfill all constraints.

In higher dimension there exists good algorithms for solving linear programming problems. The best know of them is the Simplex algorithm. Various implementations of linear programming are available ranging from self contained packages with modelling languages for describing linear programming problems to add on tools for spreadsheets. The author has used the GAMS – General algebraic modelling system [11] package with good success.

References

[1] I. L. Dryden and K. Mardia, *Statistical Shape Analysis*. Chichester: John Wiley & Sons, 1998. xx + 347 pp.

[2] J. C. Gower, "Generalized Procrustes analysis," *Psychometrika*, vol. 40, pp. 33–50, 1975.

[3] C.Goodall, "Procrustes methods in the statistical analysis of shape," *Journal of the Royal Statistical Society, Series B*, vol. 53, no. 2, pp. 285–339, 1991.

[4] A. F. Siegel, "A robust comparison of biological shapes," *Biometrics*, vol. 38, pp. 341–350, June 1982.

[5] A. F. Siegel, "Robust regression using repeated medians," *Biometrika*, vol. 69, no. 1, pp. 242–244, 1982.

[6] I. L. Dryden and G. Walker, "Highly resistant regression and object matching," *Biometrics*, vol. 55, pp. 820–825, Sept. 1999.

[7] N. V. Hermann, *Craniofacial morphology and growth in infants and young children with cleft lip and palate*. PhD thesis, Dept. of Pediatric Dentistry, School of Dentistry, University of Copenhagen, Denmark, 2000.

[8] J. M. F. T. Berge, "Orthogonal Procrustes rotation for two or more matrices," *Psychometrika*, vol. 42, pp. 267–276, 1977.

[9] F. S. Hillier and G. J. Lieberman, *Introduction to Operations Research*. McGraw Hill, 1995.

[10] L. F. Marcus, M. Cortia, A. L. G. J. Naylor, and D. E. Slice, eds., *Advances in Morphometrics*. Plenum press, 1993.

[11] A. Brooke, D. Kendrick, A. Meeraus, and R. Raman, *GAMS - A Users Guide*. GAMS Development Corporation, http://www.gams.com/, Dec. 1998.