

Version dated: July 16, 2012

Shape Variation in Outline Shapes

BRENDAN MCCANE

Department of Computer Science, University of Otago, Dunedin, New Zealand; Email:

mccane@cs.otago.ac.nz.

Appendices

These are the online appendices for the above titled paper which has or will appear in Systematic Biology.

APPENDIX 1: BASIC FACTS FOR COMPLEX NUMBERS

A complex number, z , is represented by two real numbers:

$$z = a + ib,$$

where $i = \sqrt{-1}$, and $a, b \in \mathbb{R}$. Given two complex numbers, $z_1 = a_1 + ib_1$, and $z_2 = a_2 + ib_2$:

$$z_1 + z_2 = (a_1 + a_2) + i(b_1 + b_2)$$

$$z_1 z_2 = (a_1 a_2 - b_1 b_2) + i(a_1 b_2 + b_1 a_2)$$

$$\overline{z_1} = a_1 - ib_1$$

$$\arg z_1 = \theta_z = \tan^{-1} b_1/a_1$$

$$|z_1| = r_z = (a_1^2 + b_1^2)^{0.5}$$

$$z_1 \overline{z_1} = a_1^2 + b_1^2$$

$$\frac{z_1}{z_2} = \frac{z_1 \overline{z_2}}{z_2 \overline{z_2}} = \frac{(a_1 a_2 - b_1 b_2) + i(b_1 a_2 - a_1 b_2)}{a_2^2 + b_2^2}$$

APPENDIX 2: DISTANCE BETWEEN CURVES

Given two plane finite smooth curves, $z'(t) \in \mathbb{C}^1$ and $w'(t) \in \mathbb{C}^1$, $0 \leq t \leq 1$, the goal is to compute a distance between the two curves based on shape only, and therefore independent of location, scale and rotation.

To factor out location, transform each curve according to:

$$z^*(t) = z'(t) - \int_0^1 z'(t) dt. \quad (1)$$

Location is factored out in the sense that the integral along the entire curve is zero and therefore the curve is zero-centred:

$$\begin{aligned} \int_0^1 z^*(t) dt &= \int_0^1 \left(z'(t) - \int_0^1 z'(t) dt \right) dt \\ &= 0 \end{aligned}$$

To factor out scale, apply the following transformation:

$$z(t) = \frac{z^*(t)}{\left(\int_0^1 z^*(t) \overline{z^*(t)} dt \right)^{0.5}}. \quad (2)$$

The resulting curve has a “norm” of one:

$$\begin{aligned} \int_0^1 z(t) \overline{z(t)} dt &= \int_0^1 \frac{z^*(t)}{\left(\int_0^1 z^*(t) \overline{z^*(t)} dt \right)^{0.5}} \frac{\overline{z^*(t)}}{\left(\int_0^1 z^*(t) \overline{z^*(t)} dt \right)^{0.5}} dt \\ &= \frac{\int_0^1 z^*(t) \overline{z^*(t)} dt}{\int_0^1 z^*(t) \overline{z^*(t)} dt} \\ &= 1 \end{aligned}$$

For the case of z or w equal to the zero curve, the arbitrary substitution $\int_0^1 z(t)\overline{z(t)}dt = 1$ is made.

The distance, $d(z, w)$, is then defined as:

$$d^2(z, w) = \min_{r, \theta} \int_0^1 (z(t) - re^{i\theta}w(t))\overline{(z(t) - re^{i\theta}w(t))}dt, \quad (3)$$

where $r, t, \theta \in \mathbb{R}^1$.

Proposition 1 (Positivity). $d(z, w) \geq 0$

Proof. It is known for all complex numbers that:

$$\begin{aligned} c\bar{c} &\in \mathbb{R}^1 \text{ and} \\ c\bar{c} &\geq 0 \quad \forall c \in \mathbb{C}^1. \end{aligned}$$

Since $d(z, w)$ consists of the square root of products or sums of elements in the form $c\bar{c}$, it follows that $d(z, w) \geq 0$ and $d(z, w) \in \mathbb{R}^1 \quad \forall z, w$. □

Proposition 2 (Equality). $d(z, w) = 0 \iff z(t) = re^{i\theta}w(t) \quad \forall t$.

Proof.

$$(z(t) - re^{i\theta}w(t))\overline{(z(t) - re^{i\theta}w(t))} \geq 0 \quad \forall t$$

therefore $d(z, w) = 0$

$$\text{iff } (z(t) - re^{i\theta}w(t))\overline{(z(t) - re^{i\theta}w(t))} = 0 \quad \forall t$$

$$\text{or } \int_0^1 z(t)\overline{z(t)}dt = \infty$$

$$\text{or } \int_0^1 w(t)\overline{w(t)}dt = \infty.$$

The latter two conditions violate the assumption of finite curves.

$$c\bar{c} = 0 \text{ iff } c = 0$$

$$\text{therefore } d(z, w) = 0 \text{ iff } z(t) - re^{i\theta}w(t) = 0,$$

$$\text{therefore } z(t) = re^{i\theta}w(t).$$

□

Proposition 3 (Symmetry). $d(z, w) = d(w, z)$

Proof. This is shown in two parts. First, it is shown that $\theta_1 = -\theta_2$ and $r_1 = r_2$ where r_1, θ_1 are the scale, rotation pair that result in minimum $d(z, w)$ and r_2, θ_2 are the scale, rotation pair that result in the minimum $d(w, z)$. Then that result is used to show that $d(z, w) = d(w, z)$.

$$\begin{aligned} d^2(w, z) &= \min_{r_2, \theta_2} \int_0^1 (w(t) - r_2 e^{i\theta_2} z(t)) \overline{(w(t) - r_2 e^{i\theta_2} z(t))} dt \\ &= \min_{r_2, \theta_2} \left[\int_0^1 w(t) \overline{w(t)} dt + r_2^2 \int_0^1 z(t) \overline{z(t)} dt - r_2 \left(\int_0^1 e^{i\theta_2} z(t) \overline{w(t)} + e^{-i\theta_2} w(t) \overline{z(t)} dt \right) \right] \\ &= \min_{r_2, \theta_2} \left[1 + r_2^2 - r_2 \left(\int_0^1 e^{i\theta_2} z(t) \overline{w(t)} + e^{-i\theta_2} w(t) \overline{z(t)} dt \right) \right] \end{aligned} \quad (4)$$

$$\frac{\partial d^2(w, z)}{\partial \theta_2} = -r_2 \left(\int_0^1 i e^{i\theta_2} z \overline{w(t)} - i e^{-i\theta_2} w(t) \overline{z(t)} dt \right) = 0$$

$$0 = e^{2i\theta_2} \int_0^1 z(t) \overline{w(t)} - w(t) \overline{z(t)} dt$$

$$e^{2i\theta_2} = \frac{\int_0^1 w(t) \overline{z(t)} dt}{\int_0^1 z(t) \overline{w(t)} dt} \quad (5)$$

$$\frac{\partial d^2(w, z)}{\partial r_2} = 2r_2 - \left(\int_0^1 e^{i\theta_2} z(t) \overline{w(t)} + e^{-i\theta_2} w(t) \overline{z(t)} dt \right) = 0$$

$$r_2 = \frac{1}{2} \int_0^1 e^{i\theta_2} z(t) \overline{w(t)} + e^{-i\theta_2} w(t) \overline{z(t)} dt \quad (6)$$

By similar reasoning, it is straightforward to see that:

$$\begin{aligned}
e^{2i\theta_1} &= \frac{\int_0^1 z(t)\overline{w(t)}dt}{\int_0^1 w(t)\overline{z(t)}dt} \\
0 &= 2r_1 - \left(\int_0^1 e^{i\theta_1}w(t)\overline{z(t)} + e^{-i\theta_1}z(t)\overline{w(t)}dt \right) \\
r_1 &= \frac{1}{2} \int_0^1 e^{i\theta_1}w(t)\overline{z(t)} + e^{-i\theta_1}z(t)\overline{w(t)}dt
\end{aligned}$$

It is easy to see that:

$$\begin{aligned}
e^{2i\theta_2} &= \frac{1}{e^{2i\theta_1}} \\
e^{2i\theta_2} &= e^{-2i\theta_1} \\
\theta_2 &= -\theta_1
\end{aligned} \tag{7}$$

and substituting Equation 7 into Equation 6:

$$\begin{aligned}
r_2 &= \frac{1}{2} \int_0^1 e^{-i\theta_1}z(t)\overline{w(t)} + e^{i\theta_1}w(t)\overline{z(t)}dt \\
&= r_1
\end{aligned}$$

Since the optimal scaling forwards and backwards is equal ($r_1 = r_2$) it follows that:

$$r_1 = r_2 = 1 \tag{8}$$

Finally, using the relationships between θ_1 and θ_2 and r_1 and r_2 , Equation 4

becomes:

$$\begin{aligned} d^2(w, z) &= 1 + r_1^2 - r_1 \left(\int_0^1 e^{-i\theta_1} z(t) \overline{w(t)} + e^{i\theta_1} w(t) \overline{z(t)} dt \right) \\ &= d^2(z, w) \end{aligned}$$

□

Proposition 4 (Triangle Inequality). $d(z, w) \leq d(z, y) + d(y, w)$

Proof. Let $\theta_{zy}, \theta_{yw}, \theta_{zw}$ be the rotations required to optimally align the curves z and y , y and w , and z and w respectively. Without loss of generality, let us assume that z and y are already aligned ($\theta_{zy} = 0$), and also that y and w are also already aligned ($\theta_{yw} = 0$). If this is not the case, then it is trivial to define new curves which are. The value of θ_{zw} is unknown, however, we can set it to a constant and calculate an upper bound on $d(z, w)$. For simplicity, let us choose $\theta_{zw*} = 0$ (where the asterisk indicates that this might not be the optimal value). If we can show that:

$$d_*(z, w) \leq d(z, y) + d(y, w) \tag{9}$$

$$d_*^2(z, w) \leq d^2(z, y) + 2d(z, y)d(y, w) + d^2(y, w) \tag{10}$$

$$d_*^2(z, w) \leq d^2(z, y) + d^2(y, w) \tag{11}$$

then the proposition necessarily follows. For a particular value of t from $d_*^2(z, w)$:

$$(z(t) - w(t)) \overline{(z(t) - w(t))}, \tag{12}$$

which is the square of the Euclidean distance between $z(t)$ and $w(t)$. From the right hand

side of Equation 11:

$$(z(t) - y(t))\overline{(z(t) - y(t))} + (y(t) - w(t))\overline{(y(t) - w(t))}, \quad (13)$$

which is the sum of the squares of the relevant Euclidean distances. Using the triangle inequality for Euclidean distances, we can conclude that:

$$(z(t) - w(t))\overline{(z(t) - w(t))} \leq (z(t) - y(t))\overline{(z(t) - y(t))} + (y(t) - w(t))\overline{(y(t) - w(t))}, \quad (14)$$

for all values of t . Since each individual element of Equation 11 satisfies the inequality, it follows that the integral must also satisfy it. Therefore Equation 11 must be true, and therefore the proposition follows. \square

Lemma 1. $d(z, w)$ is a distance.

Proof. This follows immediately from Propositions 1 to 4. \square

APPENDIX 3: OPTIMAL TRANSFORMATIONS FOR GROUPS OF POINTS AND CURVES

Consider two groups of corresponding structures $z \equiv \{z_j\}$ and $w \equiv \{w_j\}$ with each z_j, w_j representing two homologous structures in two individuals represented either as a landmark or a curve. The goal is to find a single r, θ pair that will minimise the total weighted squared distance between the two groups:

$$D^2(z, w) = \min_{r, \theta} \sum_{j=1}^N \alpha_j d^2(z_j, r e^{i\theta} w_j), \quad (15)$$

where $d(., .)$ is either the previously defined curve distance, or the Euclidean distance between two points, and α_j is a weight giving the importance of the specified homology. In the absence of other evidence, it would be usual to set $\alpha_j = 1$ for all j . Assume there are N_1 curve correspondences and N_2 landmark correspondences. Expanding and taking derivatives leads to:

$$\begin{aligned} D^2(z, w) &= \min_{r, \theta} \sum_{j=1}^{N_1} \alpha_j \int_0^1 (z_j(t) - r e^{i\theta} w_j(t)) (\overline{z_j(t) - r e^{i\theta} w_j(t)}) dt \\ &\quad + \sum_{j=N_1+1}^{N_1+N_2} \alpha_j (z_j - r e^{i\theta} w_j) (\overline{z_j - r e^{i\theta} w_j}) \\ &= \min_{r, \theta} \sum_{j=1}^{N_1} \alpha_j \int_0^1 z_j(t) \overline{z_j(t)} - r e^{i\theta} w_j(t) \overline{z_j(t)} - \overline{r e^{i\theta} w_j(t)} z_j(t) + r^2 w_j(t) \overline{w_j(t)} dt \\ &\quad + \sum_{j=N_1+1}^{N_1+N_2} \alpha_j (z_j \overline{z_j} - r e^{i\theta} w_j \overline{z_j} - \overline{r e^{i\theta} w_j} z_j + r^2 w_j \overline{w_j}) \end{aligned} \quad (16)$$

$$\begin{aligned}
\frac{\partial D^2}{\partial \theta} &= 0 \\
0 &= \sum_{j=1}^{N_1} \alpha_j \int_0^1 -ire^{i\theta} w_j(t) \overline{z_j(t)} + ire^{-i\theta} z_j(t) \overline{w_j(t)} dt \\
&\quad + \sum_{j=N_1+1}^{N_1+N_2} \alpha_j (-ire^{i\theta} w_j \overline{z_j} + ire^{-i\theta} z_j \overline{w_j})
\end{aligned} \tag{17}$$

$$e^{2i\theta} = \frac{\sum_{j=1}^{N_1} \alpha_j \left[\int_0^1 \overline{w_j(t)} z_j(t) dt \right] + \sum_{j=N_1+1}^{N_1+N_2} \alpha_j \overline{w_j} z_j}{\sum_{j=1}^{N_1} \alpha_j \left[\int_0^1 w_j(t) \overline{z_j(t)} \right] + \sum_{j=N_1+1}^{N_1+N_2} \alpha_j w_j \overline{z_j}} \tag{18}$$

$$\begin{aligned}
\frac{\partial D^2}{\partial r} &= 0 \\
0 &= \sum_{j=1}^{N_1} \alpha_j \int_0^1 -e^{i\theta} w_j(t) \overline{z_j(t)} - \overline{e^{i\theta} w_j(t)} z_j(t) + 2rw_j(t) \overline{w_j(t)} dt \\
&\quad + \sum_{j=N_1+1}^{N_1+N_2} \alpha_j (-e^{i\theta} w_j \overline{z_j} - e^{-i\theta} \overline{w_j} z_j + 2rw_j \overline{w_j})
\end{aligned} \tag{19}$$

$$r = \frac{\sum_{j=1}^{N_1} \alpha_j \left[\int_0^1 e^{i\theta} w_j(t) \overline{z_j(t)} + e^{-i\theta} \overline{w_j(t)} z_j(t) dt \right] + \sum_{j=N_1+1}^{N_1+N_2} \alpha_j (e^{i\theta} w_j \overline{z_j} + e^{-i\theta} \overline{w_j} z_j)}{2 \sum_{j=1}^{N_1} \alpha_j \left[\int_0^1 w_j(t) \overline{w_j(t)} dt \right] + 2 \sum_{j=N_1+1}^{N_1+N_2} \alpha_j (w_j \overline{w_j})} \tag{20}$$

APPENDIX 4: OPTIMISATION PROCEDURE

The equation to minimise is:

$$\min_{r,\theta,g} d(z, w, g) = \min_{r,\theta,g} \int_0^1 (z(t) - re^{i\theta}w(g(t)))\overline{(z(t) - re^{i\theta}w(g(t)))}dt \quad (21)$$

$$= \min_{r,\theta,g} \int_0^1 z(t)\overline{z(t)} + r^2w(g(t))\overline{w(g(t))} - re^{i\theta}w(g(t))\overline{z(t)} - re^{-i\theta}z(t)\overline{w(g(t))}dt \quad (22)$$

The function g can be any monotonic function that maps from $[0, 1]$ to $[0, 1]$. In this work differentiable monotonic spline interpolants are used to represent g (Fritsch and Carlson (1980); Hyman (1983)). These have the advantage of being smooth (hence diffeomorphic) and having simple representations which allows the use of several different optimisation strategies including gradient descent and simulated annealing. To construct the monotonic spline, a list of non-decreasing data points $\{g_i\}$ defined on the non-decreasing partition $\{t_i\}$, where $t_0 = 0$ and $t_{n_g} = 1$. For this work, the $\{t_i\}$ form a uniform partition (equally spaced), although non-uniform partitions are also possible. Given the $\{g_i\}$, the monotonic spline is generated according to (taken from Hyman (1983)):

$$g(t) = c_1 + (t - t_i)c_2 + (t - t_i)^2c_3 + (t - t_i)^3c_4, \quad (23)$$

where:

$$t_i \leq t \leq t_{i+1} \quad (24)$$

$$c_1 = g_i \quad (25)$$

$$c_2 = \frac{dg_i}{dt} \quad (26)$$

$$c_3 = \frac{3S_{i+1/2} - \frac{dg_{i+1}}{dt} - 2\frac{dg_i}{dt}}{t_{i+1} - t_i} \quad (27)$$

$$c_4 = \frac{2S_{i+1/2} - \frac{dg_{i+1}}{dt} - \frac{dg_i}{dt}}{(t_{i+1} - t_i)^2} \quad (28)$$

$$S_{i+1/2} = \frac{g_{i+1} - g_i}{t_{i+1} - t_i}. \quad (29)$$

There are several methods for estimating $\frac{dg_i}{dt}$ and the parabolic method has been used here:

$$\frac{dg_i^*}{dt} = \frac{(t_i - t_{i-1})S_{i+1/2} + (t_{i+1} - t_i)S_{i-1/2}}{t_{i+1} - t_{i-1}}. \quad (30)$$

Finally, $\frac{dg_i}{dt}$ must be constrained to lie inside the monotonicity region:

$$\frac{dg_i}{dt} = \begin{cases} \min[\max(0, \frac{dg_i^*}{dt}), 3S_{\min}^i] & \text{if } 0 < S_{\min}^i, \\ \max[\min(0, \frac{dg_i^*}{dt}), 3S_{\max}^i] & \text{if } 0 > S_{\max}^i, \\ 0 & 0 \geq S_{i-1/2}S_{i+1/2}, \end{cases} \quad (31)$$

where $S_{\min}^i = \min(S_{i-1/2}, S_{i+1/2})$, and $S_{\max}^i = \max(S_{i-1/2}, S_{i+1/2})$.

For optimising equation 22 it is sufficient to choose the optimal $\{g_i\}$ used to define the monotonic spline. This is a constrained optimisation problem with the constraint that $g_i \leq g_{i+1}$. There are many possible optimisation strategies. Simple gradient descent can be used but may produce non-feasible solutions (that is, the resulting $\{g_i\}$ need not be monotonic). Simulated annealing is an obvious choice because it can find global solutions

and constraints are easily incorporated into the search. Extensive experimentation with simulated annealing has been performed, but it proved to be very slow (it could take several minutes to match two curves on a standard desktop computer) and developing an appropriate cooling schedule was difficult and error-prone. Instead a modified gradient descent algorithm was used as described below.

Rather than explicitly incorporate constraints using a constrained optimization routine, the problem was reparameterised to implicitly incorporate the constraints. To ensure a non-decreasing bounded sequence of numbers, the g_i can be written as:

$$\{g_0, g_1, g_2, \dots\} = \{0, e^{\delta_1}, e^{\delta_1} + e^{\delta_2}, e^{\delta_1} + e^{\delta_2} + e^{\delta_3}, \dots\} / (e^{\delta_1} + \dots + e^{\delta_{n_g}}), \quad (32)$$

where $-\infty \leq \delta_i \leq \infty$. Note that $0 \leq e^{\delta_i} \leq \infty$, and therefore $g_i \leq g_{i+1}$ thus satisfying the monotonicity constraint. Also, the sequence is normalised by the last item and therefore $g_{n_g} = 1$. Standard gradient descent can then be applied over the parameter set $\{\delta_1, \dots, \delta_{n_g}\}$ where partial derivatives are calculated numerically.

The resultant algorithm is very fast, typically taking seconds to match two curves, and is at least an order of magnitude faster than the simulated annealing algorithm. In many cases, the matching found is similar with both algorithms, although in some, the modified gradient descent version produces a smoother mapping.

APPENDIX 5: ALGORITHM 1 CONVERGENCE

Algorithm 1 is a greedy optimisation method with guaranteed convergence, but may not converge to the optimal solution. The convergence proof requires the following lemma:

Lemma 2. $\hat{\mu} = \frac{1}{N} \sum_{i=1}^N z_i$, minimises the sample sum squared error, $D(\mu, Z)$:

$$D(\mu, Z) = \sum_{i=1}^N d^2(\mu, z_i) \quad (33)$$

Proof.

$$D(\mu, Z) = \sum_{i=1}^N d^2(\mu, z_i) \quad (34)$$

$$= \sum_{i=1}^N \int_0^1 (\mu(t) - z_i(t))(\overline{\mu(t) - z_i(t)}) dt \quad (35)$$

$$= \sum_{i=1}^N \int_0^1 \mu(t)\overline{\mu(t)} + z_i(t)\overline{z_i(t)} - \mu(t)\overline{z_i(t)} - \overline{\mu(t)}z_i(t) dt \quad (36)$$

$$= N + \int_0^1 \left[\sum_{i=1}^N \mu(t)\overline{\mu(t)} - \mu(t)\overline{z_i(t)} - \overline{\mu(t)}z_i(t) \right] dt \quad (37)$$

To find the minimum of Equation 37 with respect to μ , the calculus-of-variations can be applied. A necessary condition is that the Euler equation is satisfied and for a problem of

this type, the Euler equation is particularly simple:

$$\frac{\partial}{\partial \mu} \left[\sum_{i=1}^N \mu \bar{\mu} - \mu \bar{z}_i - \bar{\mu} z_i \right] = 0 \quad (38)$$

$$\sum_{i=1}^N \bar{\mu} - \bar{z}_i = 0 \quad (39)$$

$$N \bar{\mu} = \sum_{i=1}^N \bar{z}_i \quad (40)$$

$$\mu = \frac{1}{N} \sum_{i=1}^N z_i \quad (41)$$

Note that for the step from Equation 38 to 39, the Cauchy-Riemann equation $\frac{\partial \mu}{\partial \bar{\mu}} = 0$ is used. In using this fact, I am assuming that μ is a holomorphic or complex differentiable function, which is a reasonable assumption since the z_i are all piecewise polynomial splines. □

Alternative proofs are also possible. In particular, one can note that D is a real function and treat the real component directly. The result is the same however.

Convergence can now be proved as follows.

Lemma 3. *Algorithm 1 converges in finite steps.*

Proof. To establish the lemma, it is sufficient to show that the sum of distances from the curves to the estimated mean is non-decreasing.

First, define the sum squared distance for iteration j of the algorithm:

$$D_j(\mu_{j-1}, Z) = \sum_{i=1}^N d^2(\mu_{j-1}, z_i(g_{ij})). \quad (42)$$

where μ_j is defined as:

$$\mu_{j-1}(t) = 1/N \sum_{i=1}^N z_i(g_{i,j-1}(t)), \quad (43)$$

and a notational abuse is made so that $z_i(g_{i,j-1})$ should be interpreted as the curve z_i reparameterised by $g_{i,j-1}$, and g is indexed by curve number i and iteration number j .

From the following reasoning:

1. at iteration j , g_{ij} are computed to minimize $D_j(\mu_{j-1}, Z)$
2. μ_j is then computed that minimizes $D_j(\cdot, Z)$ by Lemma 2
3. if $g_{i,j+1} = g_{ij}$ then $D_{j+1}(\mu_j, Z) \leq D_j(\mu_{j-1}, Z)$
4. Since $g_{i,j+1}$ is the g that minimises $D_{j+1}(\mu_j, Z)$ by definition, it also follows that $D_{j+1}(\mu_j, Z) \leq D_j(\mu_{j-1}, Z)$
5. therefore D_j is a strictly non-increasing series and since it is also bounded below (by 0), it follows that the algorithm converges.

□

REFERENCES

- Fritsch, F. and R. Carlson. 1980. Monotone piecewise cubic interpolation. *SIAM Journal on Numerical Analysis* Pages 238–246.
- Hyman, J. 1983. Accurate monotonicity preserving cubic interpolation. *SIAM Journal on Scientific and Statistical Computing* 4:645.