The Alignment of Arbitrary Contours Using Area Difference Distance Measurement

Jessica De Silva, Karen Murata, and Jung-ha An, Ph.D.

Department of Mathematics
California State University Stanislaus
One University Circle, Turlock, CA 95382, USA
jdesilva1@csustan.edu, kmurata@csustan.edu, jan@csustan.edu

Abstract. Advancements in medical imaging have allowed for analyzing the anatomy of the human body without the risks of surgery. One can use the methods in this paper to determine the health of a patient's brain or heart by comparing the anatomical contour to its ideal shape. Registration is an approach towards imaging which determines an optimal alignment between multiple images. In particular, the Area Difference Distance measurement is applied towards image registration to numerically determine an optimal alignment between arbitrary contours. The purpose of this paper is to propose the proof that the Area Difference Distance measurement[4] is a metric, illustrate optimal alignment using the Area Difference Procrustes method[2], and to show numerical simulations. This distance function takes into account two sets containing data points which represent their respective contours. The Area Difference Procrustes method is applied by aligning the arbitrary contour to a fixed contour. This optimal alignment requires minimizing the distance function in terms of rotating, scaling, and translating the arbitrary contour. The succeeding proof presented validates that these values, which optimize the distance between two contours, can be found with the sets of data points representing each contour. Once aligned, we can calculate the optimal Area Difference Distance between the contours. Through the use of MATLAB, synthetic data is applied to test the effectiveness of the optimization of the Area Difference Distance measurement.

Keywords: distance, Area Difference Distance, Area Difference Procrustes method

1 Introduction

Various metric functions representing a distance measurement can be implemented towards quantifying the difference between two shapes. The Area Difference Distance measurement presented can be utilized for determining differences between two arbitrary contours. The Euclidean distance formula is useful for measuring distances between two points. But in order to calculate the difference between two contours in real life, the finite sum of the Euclidean distance between corresponding points will not cover the entire area of the difference. The Area Difference Distance measurement calculates the area between

two subsequent points on one contour and those corresponding points on the other contour. By utilizing the Area Difference Distance measurement, the entire distance between any two arbitrary contours can be calculated with a finite sum.

In order to ensure that the distance between two arbitrary shapes is a minimum, the shapes must be optimally aligned with respect to the distance measurement. When working with arbitrary contours, this alignment can be difficult without a systematic approach. We can use Procrustes superimposition, also referred to as the Procrustes method, together with with the Area Difference Distance measurement as an efficient approach to this alignment problem. The method of using the Procrustes method to minimize the Area Difference Distance is called the Area Difference Procrustes method[2]. Before applying the Area Difference Procrustes method, a fixed and floating contour must be defined. The Area Difference Procrustes method then scales, rotates, and positions the floating contour to ensure optimal alignment relative to the fixed contour [1]. The results from this registration process will allow for the calculation of the minimum Area Difference Distance between these contours. In this paper, a formal proof is developed to validate the optimization of Area Difference Distance measurement by means of the Area Difference Procrustes method. Section 2 defines Area Difference Distance measurement and introduces the proof that this is a metric, Section 3 presents an optimization of this measurement and introduces a proof for the optimization, and Section 4 provides numerical implementation in MATLAB to illustrate the effectiveness of the distance optimization.

2 Distance Functions

In this section we define distance and prove its validity. An example which illustrates the distance definition is also provided.

Definition: Distance represents the amount of space between two objects. The distance function, referred to as the metric, defines the distance between elements of a set. A distance function on a set M is a function d:M×M→ $\mathbb{R}$, which satisfies the following properties $\forall(x, y) \in$ M×M:

i) $d(x, y) \geq 0$
ii) $d(x, y) = 0$ if and only if $x = y$
iii) $d(x, y) = d(y, x)$
iv) $d(x, y) \leq d(x, z) + d(z, y)$

2.1 Absolute Value Distance Function

Definition: Let $d(x, y) := |x - y|$.

Proposition 1: Show that $d(x, y) := |x - y|$ is a distance.

Proof: i) $d(x, y) = |x - y|$.

Then $|x-y| \geq 0$ by the definition of absolute value.

ii) $d(x,y) = |x-y| = 0$
Then $x - y = 0$ by definition of absolute values.
Therefore $x = y$.

Let $x = y$.
Hence $x - y = 0$.
Furthermore, $|x-y| = 0$.

$\therefore d(x,y) = |x-y| = 0$ if and only if $x = y$.

iii) $d(x,y) = |x-y|$
$$\begin{aligned} |x-y| &= |-(y-x)| \\ &= |-1||y-x| \\ &= |y-x| \\ &= d(y,x) \end{aligned}$$
$\therefore d(x,y) = d(y,x)$.

iv) $d(x,y) = |x-y|$.
$$\begin{aligned} |x-y| &= |x-z+z-y| \\ &\leq |x-z| + |z-y| \text{ by triangle inequality} \\ &= d(x,z) + d(z,y). \end{aligned}$$
$\therefore d(x,y) \leq d(x,z) + d(z,y)$.

$\therefore d(x,y) = |x-y|$ is a distance function. □

2.2 Area Difference Distance Measurement

Definition: Area Difference Distance refers to the symmetric difference between two sets; each set holds data points of a distinct contour. Specifically, Area Difference Distance is the difference between the union of the two sets and their intersection. For any two sets, A and B, representing distinct shapes, the Area Difference Distance measurement is defined as:

$$\text{d(A,B)=Int(A}\triangle\text{B) =Int((A}\cup\text{B)} \setminus \text{(A}\cap\text{B))}$$

where Int is the interior, or area, of the set[2, 4].

Proposition 2: Show that d(A,B)=Int(A$\triangle$B) =Int((A$\cup$B) $\setminus$ (A$\cap$B)) is an area difference distance.

Proof of Area Difference Distance:
Let d(A,B) := Int(A $\triangle$ B).

i) d(A,B)=Int(A$\triangle$B)

$=\text{Int}((A\cup B)\backslash(A\cap B))$.
Notice that $(A\cap B)\subset(A\cup B)$.

Then, $(A\cup B)\backslash(A\cap B)\supset\varnothing$.

So, $\text{Int}(A\triangle B)\geq 0$.
$\therefore d(A,B)\geq 0$.

ii) $\Rightarrow$Assume $A=B$ and $A\neq\varnothing$.
Then, $(A\triangle B)=(A\cup A)\backslash(A\cap A)=(A\backslash A)=\varnothing$.
Furthermore, $\text{Int}(\varnothing)=0$.
$\therefore$ if $A=B$, then $d(A,B)=0$.
$\Leftarrow$Assume $A\neq B$ and $d(A,B)=0$.
Then, without loss of generality, $\exists\ x\in\mathbf{R}^2\ni(x\in A\wedge x\notin B)$.
Hence, $(A\cap B)\subsetneq(A\cup B)$.
Therefore, $\text{Int}(A\triangle B)\neq 0$ and it follows that $d(A,B)\neq 0$.
Thus we have reached a contradiction.
$\therefore$ if $d(A,B)=0$, then $A=B$.
$\therefore d(A,B)=0$ if and only if $A=B$.

iii) $d(A,B)=\text{Int}(A\triangle B)$
$=\text{Int}((A\cup B)\backslash(A\cap B))$.
Notice that $(A\cup B)=(B\cup A)$ and $(A\cap B)=(B\cap A)$.

Thus, $\text{Int}((A\cup B)\backslash(A\cap B))=\text{Int}((B\cup A)\backslash(B\cap A))$
$=\text{Int}(B\triangle A)$
$=d(B,A)$.
$\therefore d(A,B)=d(B,A)$.

iv) $d(A,C)=\text{Int}(A\triangle C)$
$=\text{Int}((A\triangle B)\triangle(B\triangle C))$
$\leq\text{Int}(A\triangle B)+\text{Int}(B\triangle C)$
$=d(A,B)+d(B,C)$.
$\therefore d(A,C)\leq d(A,B)+d(B,C)$

$\therefore d(A,B)=\text{Int}(A\triangle B)$ is a metric. □

3 Alignment of Two Arbitrary Contours Using Area Difference Distance Measurement

The purpose of this section is to find an optimal alignment of two contours using Area Difference Distance measurement. Optimizing the Area Difference Distance between two data sets determines parameters for aligning these data sets. These parameters can then be used to determine the contours of optimal alignment.

3.1 Alignment of Two Arbitrary Contours

Proposition 3: Determine an optimization of the Area Difference Distance measurement by means of the Area Difference Procrustes method and show that this method is valid.

The goal is to find $\tilde{\text{A}}$ such that $d(\tilde{\text{A}}, B)$ is minimized. In other words, align A onto B. We try to find the best μ, R, T which solves the following optimization problem:

$\min_{\mu,R,T}$ d($\tilde{\text{A}}, B$) , where $\tilde{\text{A}} := \mu RA + T$

and $d(\tilde{\text{A}},B)$:= Int($\tilde{\text{A}} \triangle B$)
=Int(($\tilde{\text{A}} \cup B$) \ ($\tilde{\text{A}} \cap B$)).

Here, R= $\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$, T= $\begin{bmatrix} t_1 & t_1 & \dots & t_1 \\ t_2 & t_2 & \dots & t_2 \end{bmatrix}$, and μ is a scaling constant.

Let $S_i = \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$, $\forall S_i \in$ T, i= 1, 2,..., n.

Now define $\tilde{\text{A}} = \mu RA + T$

Then $\tilde{a}_i \in \tilde{\text{A}}$, $\tilde{a}_i = \begin{bmatrix} \tilde{a}_{i,1} \\ \tilde{a}_{i,2} \end{bmatrix}$

$$= \begin{bmatrix} a & -b \\ b & a \end{bmatrix} \begin{bmatrix} a_{i,1} \\ a_{i,2} \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}.$$

$$= \begin{bmatrix} aa_{i,1} - ba_{i,2} + t_1 \\ ba_{i,1} + aa_{i,2} + t_2 \end{bmatrix}.$$

So $b_{i+1} = (b_{i+1,1}\ b_{i+1,2})$,
$b_i = (b_{i,1}\ b_{i,2})$,

$\tilde{\text{a}}_i = (\tilde{a}_{i,1}\ \tilde{a}_{i,2})$,
and $\tilde{a}_{i+1} = (\tilde{a}_{i+1,1}\ \tilde{a}_{i+1,2})$.

3.2 Solving the Optimization Problem

Furthermore, the Area Difference Distance between any two points of each set A and B can be defined as:

$$d_i(A,B) = \{(a_{i,1} - b_{i,1})^2 + (a_{i,2} - b_{i,1})^2\}\{(b_{i+1,1} - b_{i,1})^2 + (b_{i+1,2} - b_{i,2})^2\}$$

to solve the optimization problem of $\min_{a,b,t_1,t_2}$ d($\tilde{\text{A}}, B$), where

$$d_i = \{(aa_{i,1} - ba_{i,2} + t_1 - b_{i,1})^2 + (ba_{i,1} + aa_{i,2} + t_2 - b_{i,2})^2\}\{(b_{i+1,1} - b_{i,1})^2 + (b_{i+1,2} - b_{i,2})^2\}.$$

In the process of determining values for a, b, t_1, and t_2 which best optimize $\min_{a,b,t_1,t_2} d(\tilde{A}, B)$, we first must find the critical points of our distance equation. These critical points are calculated by taking partial derivatives of d_i with respect to each variable in our optimization problem, and then setting the derivatives equal to zero. The following equations represent partial derivatives of the distance function, d_i, with respect to a, b, t_1, and t_2:

$$\begin{aligned}\frac{\partial d_i}{\partial a} &= \{2a_{i,1}(aa_{i,1} - ba_{i,2} + t_1 - b_{i,1}) + 2a_{i,2}(ba_{i,1} + aa_{i,2} + t_2 - b_{i,2})\}\{(b_{i+1,1} - b_{i,1})^2 + (b_{i+1,2} - b_{i,2})^2\} \\ &= \{2a(a_{i,1})^2 - 2ba_{i,1}a_{i,2} + 2a_{i,1}t_1 - 2a_{i,1}b_{i,1} + 2ba_{i,1}a_{i,2} + 2a(a_{i,2})^2 + 2a_{i,2}t_2 - 2a_{i,2}b_{i,2}\}\{(b_{i+1,1} - b_{i,1})^2 + (b_{i+1,2} - b_{i,2})^2\} \\ &= \{2a((a_{i,1})^2 + (a_{i,2})^2) + 2t_1(a_{i,1}) + 2t_2(a_{i,2}) - 2a_{i,1}b_{i,1} - 2a_{i,2}b_{i,2}\}\{(b_{i+1,1} - b_{i,1})^2 + (b_{i+1,2} - b_{i,2})^2\}\end{aligned}$$

$$\begin{aligned}\frac{\partial d_i}{\partial b} &= \{2a_{i,2}(aa_{i,1} - ba_{i,2} + t_1 - b_{i,1}) - 2a_{i,1}(ba_{i,1} + aa_{i,2} + t_2 - b_{i,2})\}\{(b_{i+1,1} - b_{i,1})^2 + (b_{i+1,2} - b_{i,2})^2\} \\ &= \{2aa_{i,1}a_{i,2} - 2aa_{i,1}a_{i,2} - 2b(a_{i,2})^2 - 2b(a_{i,1})^2 + 2t_1a_{i,2} - 2t_2a_{i,1} - 2a_{i,2}b_{i,1} + 2a_{i,1}b_{i,2}\}\{(b_{i+1,1} - b_{i,1})^2 + (b_{i+1,2} - b_{i,2})^2\} \\ &= \{-2b((a_{i,1})^2 + (a_{i,2})^2) + 2t_1a_{i,2} - 2t_2a_{i,1} - 2a_{i,2}b_{i,1} + 2a_{i,1}b_{i,2}\}\{(b_{i+1,1} - b_{i,1})^2 + (b_{i+1,2} - b_{i,2})^2\}\end{aligned}$$

$$\begin{aligned}\frac{\partial d_i}{\partial t_1} &= \{2(aa_{i,1} - ba_{i,2} + 2t_1 - 2b_{i,1})\}\{(b_{i+1,1} - b_{i,1})^2 + (b_{i+1,2} - b_{i,2})^2\} \\ &= \{2a(a_{i,1}) - 2ba_{i,2} + 2t_1 - 2b_{i,1})\}\{(b_{i+1,1} - b_{i,1})^2 + (b_{i+1,2} - b_{i,2})^2\}\end{aligned}$$

$$\begin{aligned}\frac{\partial d_i}{\partial t_2} &= \{2(ba_{i,1} + aa_{i,2} + t_2 - b_{i,2})\}\{(b_{i+1,1} - b_{i,1})^2 + (b_{i+1,2} - b_{i,2})^2\} \\ &= \{2aa_{i,2} + 2ba_{i,1} + 2t_2 - 2b_{i,2}\}\{(b_{i+1,1} - b_{i,1})^2 + (b_{i+1,2} - b_{i,2})^2\}\end{aligned}$$

The partial derivatives above are only taking into account the set of points at a particular value i. In order to reach the total distance, we must take the summation of all partial derivatives from $i = 1, 2, ..., n$.
For simpler notation, let $c_i = (b_{i+1,1} - b_{i,1})^2 + (b_{i+1,2} - b_{i,2})^2$,
$C1 = \sum_{i=1}^{n} c_i((a_{i,1})^2 + (a_{i,2})^2)$,

$C2 = \sum_{i=1}^{n} c_i(a_{i,1}),$
$C3 = \sum_{i=1}^{n} c_i(a_{i,2}),$
and $C4 = \sum_{i=1}^{n} c_i.$

Now we can rewrite the partial derivatives for our optimization problem as follows:

$$\frac{\partial d}{\partial a} = 2a(C1) + 2t_1(C2) + 2t_2(C3) - \sum_{i=1}^{n}(2a_{i,1}b_{i,1} + 2a_{i,2}b_{i,2})c_i,$$

$$\frac{\partial d}{\partial b} = 2b(-C1) + 2t_1(C3) + t_2(-C2) - \sum_{i=1}^{n}(2a_{i,2}b_{i,1} - 2a_{i,1}b_{i,2})c_i,$$

$$\frac{\partial d}{\partial t_1} = 2a(C2) + 2b(-C3) + 2t_1(C4) - \sum_{i=1}^{n}(2b_{i,1})c_i,$$

and $\frac{\partial d}{\partial t_2} = 2a(C3) + 2b(C2) + 2t_2(C4) - \sum_{i=1}^{n}(2b_{i,2})c_i$.

Next, we set this partial differential equation equal to zero:

$\frac{\partial d}{\partial a} = 0:$
$2a(C1) + 2t_1(C2) + 2t_2(C3) - \sum_{i=1}^{n}(2a_{i,1}b_{i,1} + 2a_{i,2}b_{i,2})c_i = 0$
$a(C1) + t_1(C2) + t_2(C3) - \sum_{i=1}^{n}(a_{i,1}b_{i,1} + a_{i,2}b_{i,2})c_i = 0$
$a(C1) + t_1(C2) + t_2(C3) = \sum_{i=1}^{n}(a_{i,1}b_{i,1} + a_{i,2}b_{i,2})c_i.$
For simpler notation, set $B1 = \sum_{i=1}^{n}(a_{i,1}b_{i,1} + a_{i,2}b_{i,2})c_i$
$\implies B1 = a(C1) + t_1(C2) + t_2(C3).$

$\frac{\partial d}{\partial b} = 0:$
$2b(-C1) + 2t_1(C3) + 2t_2(-C2) - \sum_{i=1}^{n}(2a_{i,2}b_{i,1} - 2a_{i,1}b_{i,2})c_i = 0$
$b(-C1) + t_1(C3) + t_2(-C2) - \sum_{i=1}^{n}(a_{i,2}b_{i,1} - a_{i,1}b_{i,2})c_i = 0$
$b(-C1) + t_1(C3) + t_2(-C2) = \sum_{i=1}^{n}(a_{i,2}b_{i,1} - a_{i,1}b_{i,2})c_i.$
For simpler notation, set $B2 = \sum_{i=1}^{n}(a_{i,2}b_{i,1} - a_{i,1}b_{i,2})c_i$

$$\implies B2 = b(-C1) + t_1(C3) + t_2(-C2).$$

$\frac{\partial d}{\partial t_1} = 0$:

$2a(C2) + 2b(-C3) + 2t_1(C4) - \sum_{i=1}^{n}(2b_{i,1})c_i = 0$

$a(C2) + b(-C3) + t_1(C4) - \sum_{i=1}^{n}(b_{i,1})c_i = 0$

$a(C2) + b(-C3) + t_1(C4) = \sum_{i=1}^{n}(b_{i,1})c_i.$

Let $B3 = \sum_{i=1}^{n}(b_{i,1})c_i$

$$\implies B3 = a(C2) + b(-C3) + t_1(C4).$$

$\frac{\partial d}{\partial t_2} = 0$:

$2a(C3) + 2b(C2) + 2t_2(C4) - \sum_{i=1}^{n}(2b_{i,2})c_i = 0$

$a(C3) + b(C2) + t_2(C4) - \sum_{i=1}^{n}(b_{i,2})c_i = 0$

$a(C3) + b(C2) + t_2(C4) = \sum_{i=1}^{n}(b_{i,2})c_i.$

Let $B4 = \sum_{i=1}^{n}(b_{i,2})c_i$

$$\implies B4 = a(C3) + b(C2) + t_2(C4).$$

Matrix Display of System

We will then use matrices to better display the system of four equations derived from the partial derivatives of d in terms of a, b, t_1, and t_2[4, 2]:

$$\begin{gathered}(C1)a + (C2)t_1 + (C3)t_2 = B1\\ (-C1)b + (C3)t_1 + (-C2)t_2 = B2\\ (C2)a + (-C3)b + (C4)t_1 = B3\\ (C3)a + (C2)b + (C4)t_2 = B4\end{gathered}$$

$$\begin{bmatrix} C1 & 0 & C2 & C3 \\ 0 & -C1 & C3 & -C2 \\ C2 & -C3 & C4 & 0 \\ C3 & C2 & 0 & C4 \end{bmatrix} \begin{bmatrix} a \\ b \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} B1 \\ B2 \\ B3 \\ B4 \end{bmatrix}$$

4 Numerical Implementation

Optimization of the Area Difference Distance measurement is numerically tested through the generation of contours in MATLAB. This testing requires code which

outputs a matrix of coordinates representing any desired shape, such as an ellipse, triangle, or rectangle. Since the rigid registration process is a point-based system, these matrices must be well-ordered. They then become inputs for the following code, which aligns the contours they represent.

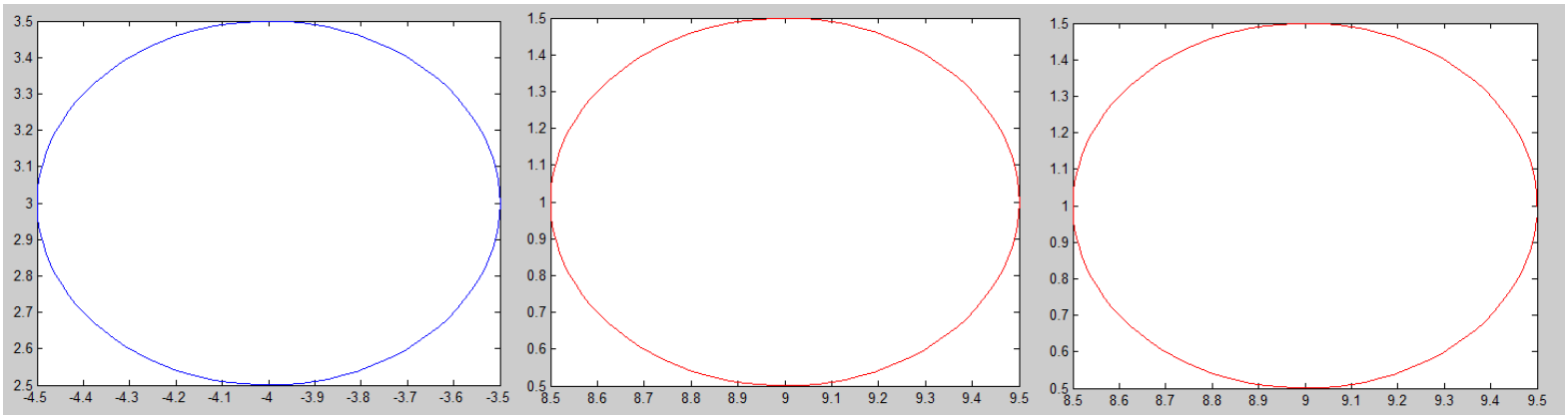
Proposition 4: Create code in MATLAB which uses two well-ordered matrices, representing fixed(**B**) and floating contours(**A**), for the optimal alignment and distance calculation between the two contours.

```
function Newcontour = AligningContours(A,B)
hold on;
ci=[size(100,1)];
for i=1:100
ci(i,1)= (B(i+1,1)-B(i,1))^2+ (B(i+1,2)-B(i,2))^2;
end
c1matrix=[size(101,1)];
c2matrix=[size(101,1)];
c3matrix=[size(101,1)];
c4matrix=[size(101,1)];
for i= 1:100
c1matrix(i,1)= ci(i,1)*(A(i,1)^2+A(i,2)^2);
c2matrix(i,1)= ci(i,1)*(A(i,1));
c3matrix(i,1)= ci(i,1)*(A(i,2));
c4matrix(i,1)= ci(i,1);
end
C1= sum(c1matrix);
C2= sum(c2matrix);
C3= sum(c3matrix);
C4= sum(c4matrix);
b1matrix=[size(101,1)];
b2matrix=[size(101,1)];
b3matrix=[size(101,1)];
b4matrix=[size(101,1)];
for i=1:100
b1matrix(i,1)= ci(i,1)*(A(i,1)*B(i,1)+A(i,2)*B(i,2));
b2matrix(i,1)= ci(i,1)*(A(i,2)*B(i,1)-A(i,1)*B(i,2));
b3matrix(i,1)= ci(i,1)*B(i,1);
b4matrix(i,1)= ci(i,1)*B(i,2);
end
B1= sum(b1matrix);
B2= sum(b2matrix);
B3= sum(b3matrix);
B4= sum(b4matrix);
C= [C1, 0, C2, C3; 0, -C1, C3, -C2; C2, -C3, C4, 0; C3, C2, 0, C4];
J= [B1;B2;B3;B4];
```

```
if det(j)~=0
V=C \ J; % C*inv(J)
rotatescale= [V(1,1), V(2,1); -V(2,1), V(1,1)];
T= [size(101,2)];
for i=1:101
T(i,1)=V(3,1);
T(i,2)=V(4,1);
end
Newcontour= A*rotatescale+T;
hold off;
plot(Newcontour(:,1), Newcontour(:,2));
hold on;
plot(B(:,1),B(:,2),'r');
Diff=[size(100,1)];
for i=1:100
Diff(i,1)=((Newcontour(i,1)-B(i,1))^ 2 + (Newcontour(i,2)-B(i,2))^2)^(1/2)
end
Distance=sum(Diff);
Distance
end
end
```

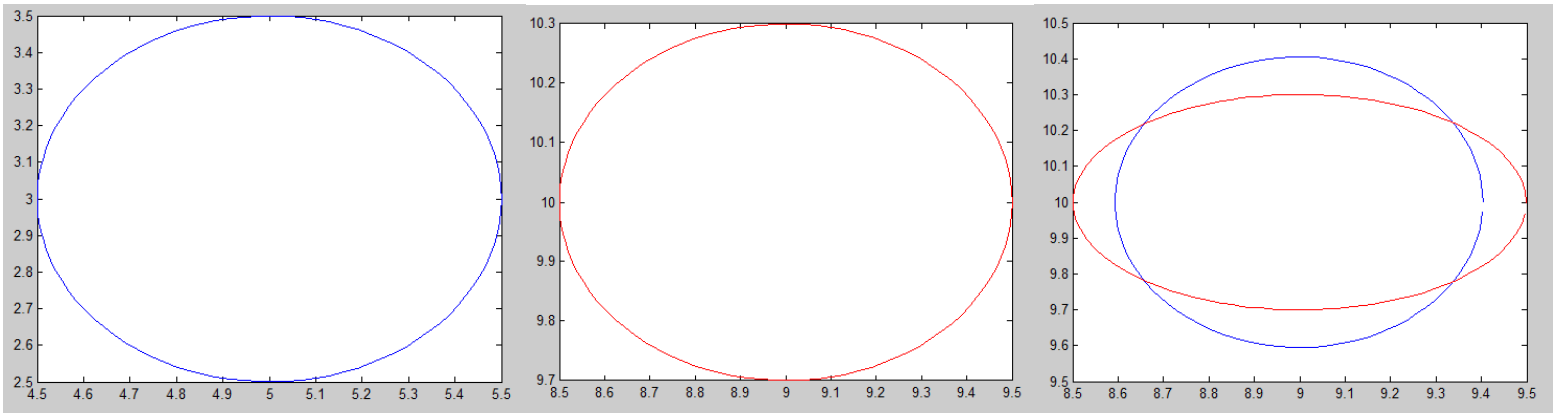
4.1 Experimental Results

The following figures represent various test runs in MATLAB of the alignment procedure. The graphs have plotted $\tilde{A}$ (3.1) in blue, and B in green in the final image of each figure.

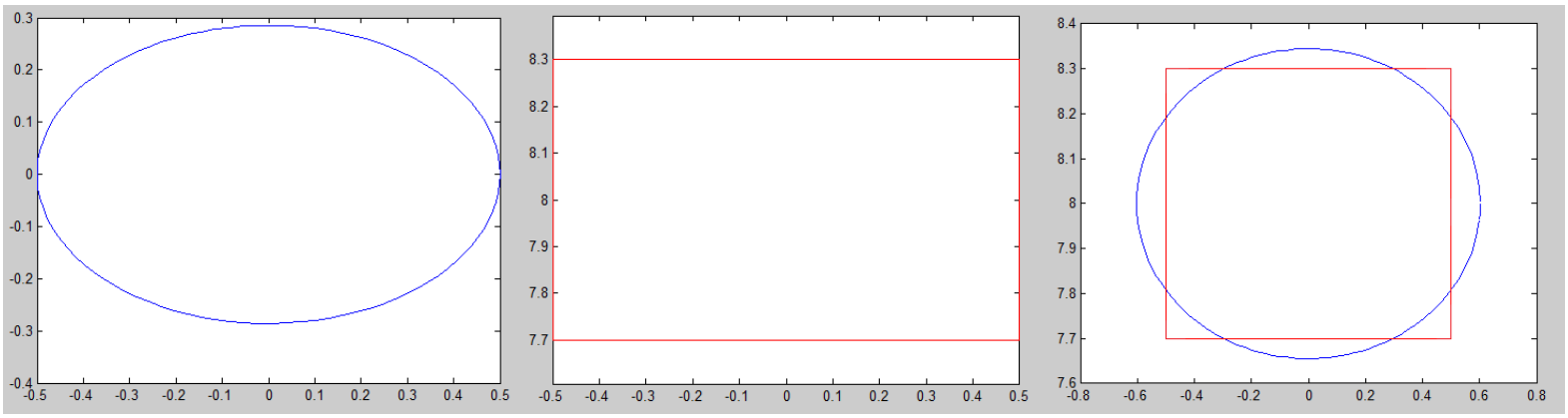


(a) (b) (c)

Fig. 1: A Circle(a) Aligned with a Circle(b)

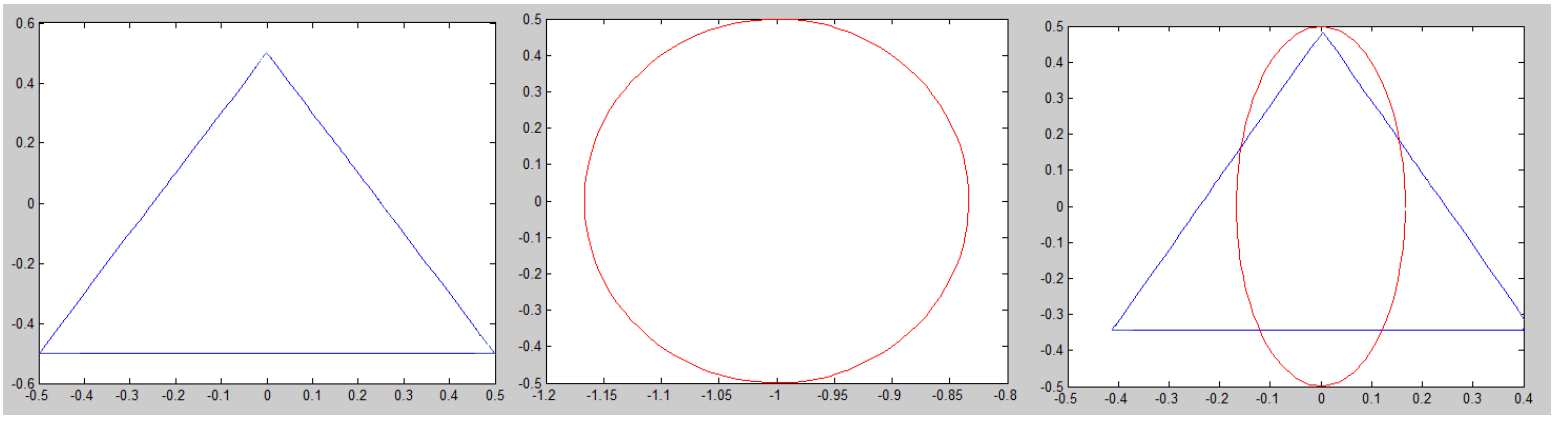


(a) (b) (c)



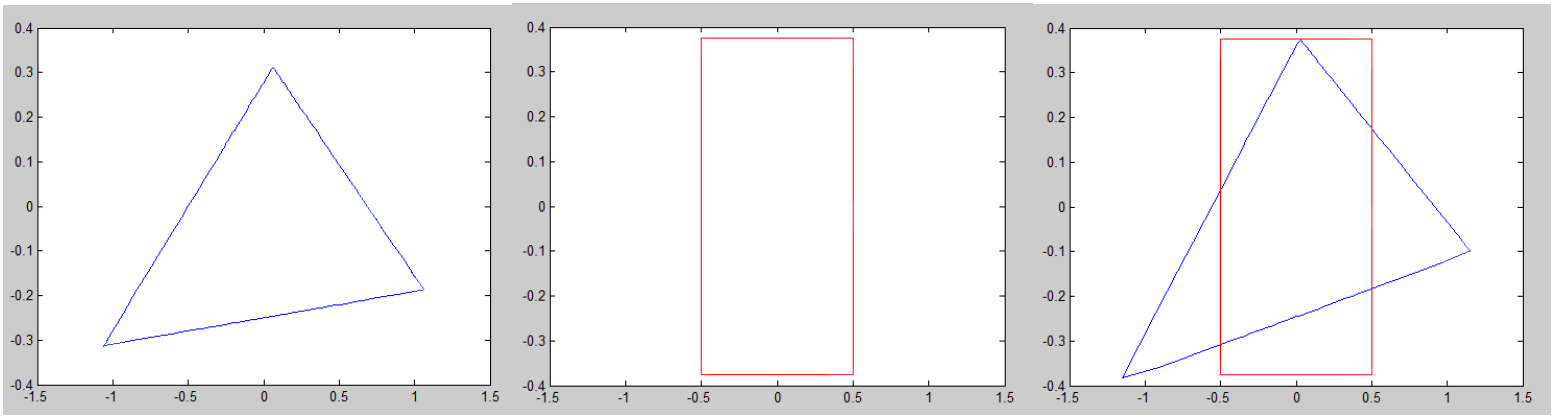
(a) (b) (c)

Fig. 3: An Ellipse(a) Aligned with a Rectangle(b)



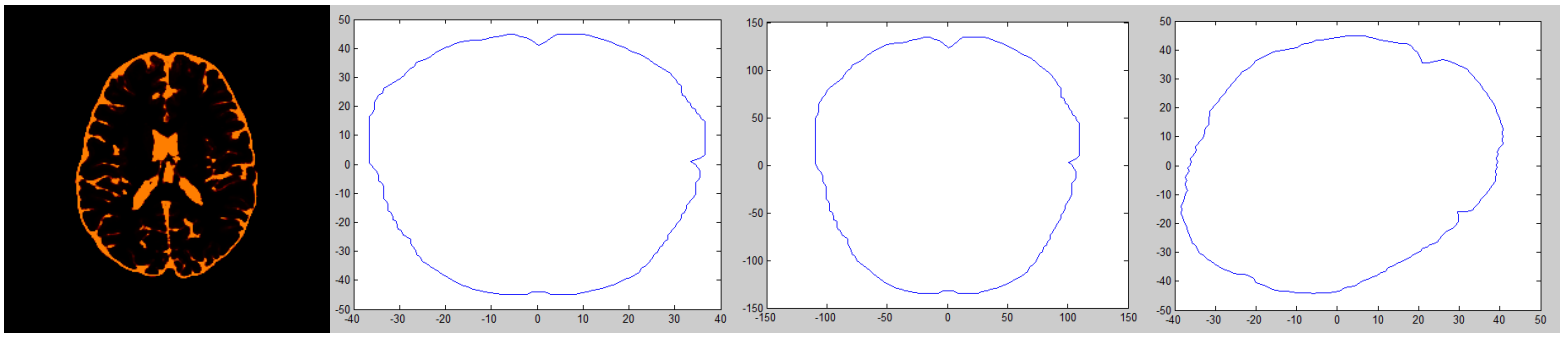
(a) (b) (c)

Fig. 4: A Triangle(a) Aligned with an Ellipse(b)



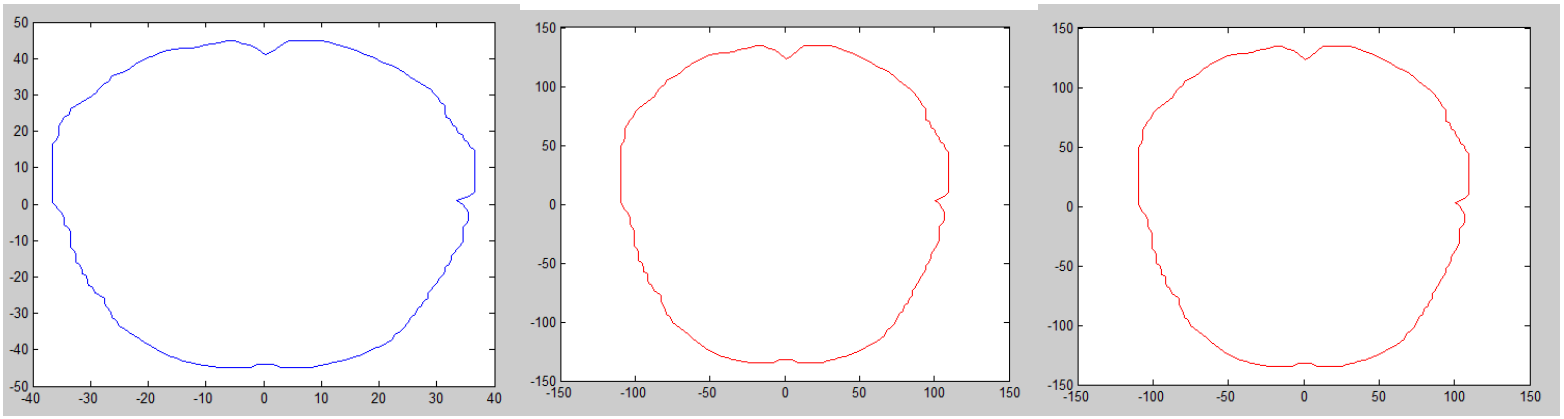
(a) (b) (c)

Fig. 5: A Triangle(a) Aligned with a Rectangle(b)



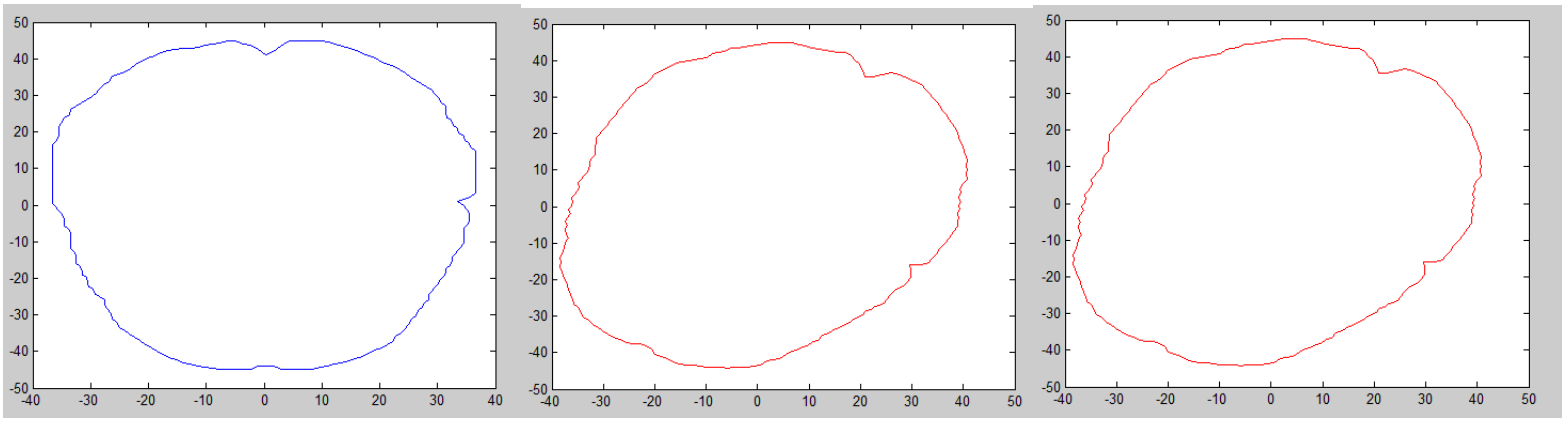
(a) (b) (c) (d)

Fig. 6: http://mouldy.bic.mni.mcgill.ca/cgi/brainweb1?alias=subject04_csf Brain Contour, Brain Contour Scaled(x3) and Rotated($\pi/6$) [3]



(a) (b) (c)

Fig. 7: Brain Contour aligned with Brain Contour Scaled(x3)



(a) (b) (c)

Fig. 8: Brain Contour aligned with Brain Contour Rotated($\pi/6$)

Figure	Floating Contour	Fixed Contour
1	circle; radius: .5, center:(-4,3)	circle; radius: .5, center: (9,1)
2	circle; radius: .5, center:(5,3)	ellipse; radii: (.5,.3), center:(9,10)
3	ellipse; radii: (.5,.2857), center:(0,0)	rectangle; length: 5, width: .3, center:(0,8)
4	triangle; vertices: (-.4876,-.3317), (.5124,-.3317), (.0124,.6683)	ellipse; radii: (.1667,.5), center: (-1,0)
5	triangle; vertices: (-1.0569,-.2475), (.0681,.3775), (1.0681,-.1225)	rectangle; length: .5, width: .375, center: (2,1)
7	Brain Contour	Scaled Brain Contour
8	Brain Contour	Rotated Brain Contour

Figure	$\tilde{\mathrm{R}}$	S_i	Distance
1	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	[13 -2]	$1.7395e$-30≈ 0
2	$\begin{bmatrix} .8101 & 0 \\ 0 & .8101 \end{bmatrix}$	[5.217 7.745]	0.0124
3	$\begin{bmatrix} 1.2066 & 0 \\ 1.2066 & 0 \end{bmatrix}$	[.0071 8]	0.0084
4	$\begin{bmatrix} .8301 & .002 \\ .8301 & -.002 \end{bmatrix}$	[-1.0277 $9.7243e^{-4}$]	0.0258
5	$\begin{bmatrix} 1.0986 & .0721 \\ 1.0986 & -.0721 \end{bmatrix}$	[2.0295 .9885]	.1440
7	$\begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}$	[0 0]	$5.0295e$-27≈ 0
8	$\begin{bmatrix} 0.8660 & -0.5 \\ 0.5 & 0.8660 \end{bmatrix}$	[0 0]	$5.2592e$-26≈ 0

5 Conclusion

This paper illustrates an optimization of the Area Difference Distance measurement. We can numerically interpret the minimum difference between any two arbitrary contours after translating, rotating, and scaling a floating contour in relation to the fixed contour. Figures 1-5 and 7-8 show promising results of optimal alignment between two contours, with relatively small outcomes of distance. Our future focus will be applying the alignment method with various brain contours. By calculating the minimum distance between a fixed and floating contour, classification of these shapes, relative to their difference to the fixed contour, will be represented more accurately. Prior anatomical shape information in medical imaging can be generated through the application of this metric function. For example, abnormalities of the contour in certain areas of a brain may suggest serious defects. Advancements such as this, which rely on the Area Difference Distance metric, will be the focus of our future work.

References

1. F. Rohlf. Shape Statistics: Procrustes Superimposition and tangent Spaces, Journal of Classification 16, pp. 197-223 (1999).
2. J. An, Y. Chen, M. Chang, D. Wilson, and E. Geiser. Generating Geometric Models through Self-Organizing Maps, Multiscale optimization methods and applications, Nonconvex Optim. Appl., 82 pp. 241-250, (2006).
3. McConnell Brain Imaging Center. BrainWeb: Simulated Brain Database, http://mouldy.bic.mni.mcgill.ca/brainweb/.
4. Y. Chen, D. Wilson and F. Huang. A New Procrustes Methods for Generating Geometric Models, Proceedings of World Multiconference on Systems, Cybernetics and Informatics, pp. 227-232, (2001).