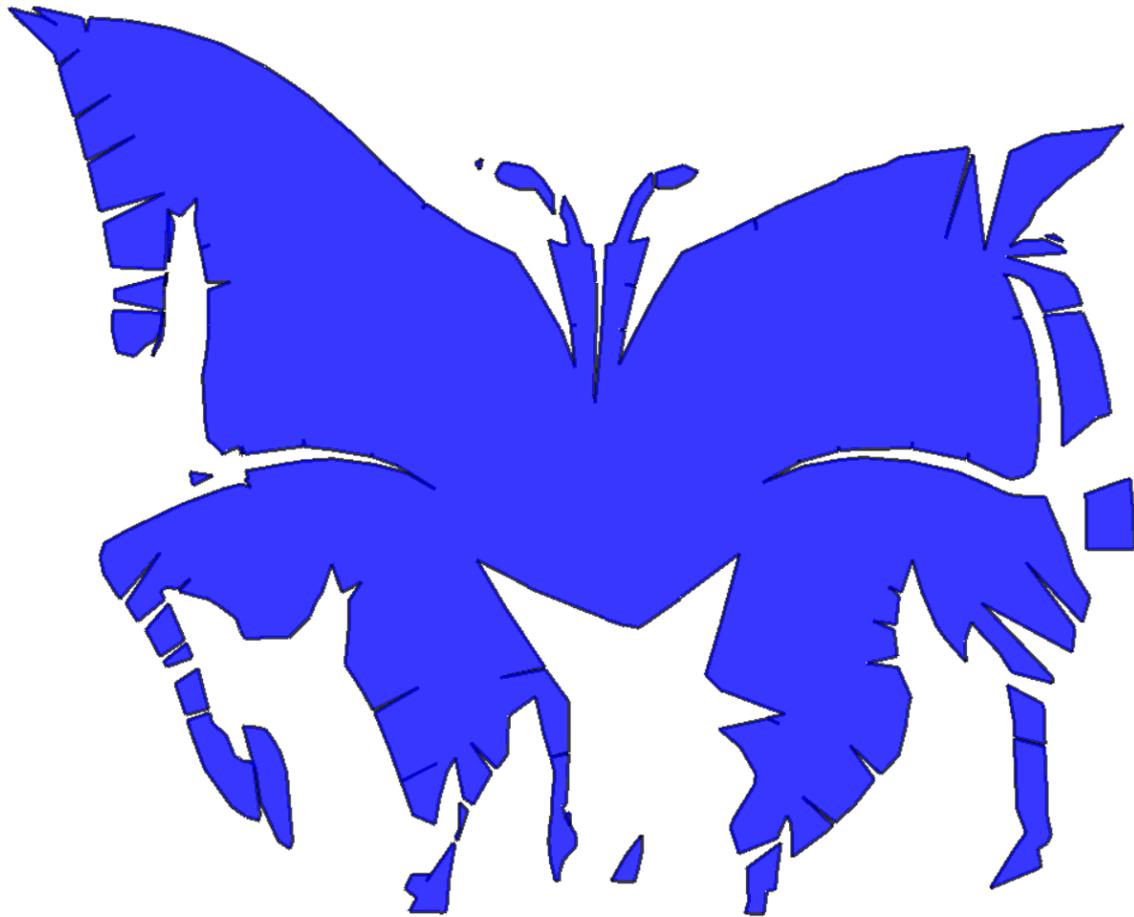


The Closest Point Morph



Utrecht University

Master's Thesis
ICA-5655331

Author:
Lex de Kogel

First Examiner:
Jordi L. Vermeulen

Date:
12-07-2021

Second Examiner:
Marc van Kreveld

1 Introduction

Given two or more shapes, a sequence of intermediate shapes can be generated which gradually change from a source to a target shape. This process is called morphing, also known as shape interpolation or blending.

Many different types of techniques exist with different purposes. For example, in the field of animation, morphing techniques can be used to smoothly transform a characters pose into another. Morphing techniques can also be used to reconstruct 3D shapes from 2D slices. These problems have very different requirements. For example when working on an animation of an animal transforming into another, it would be sensible to use a technique that allows the user to

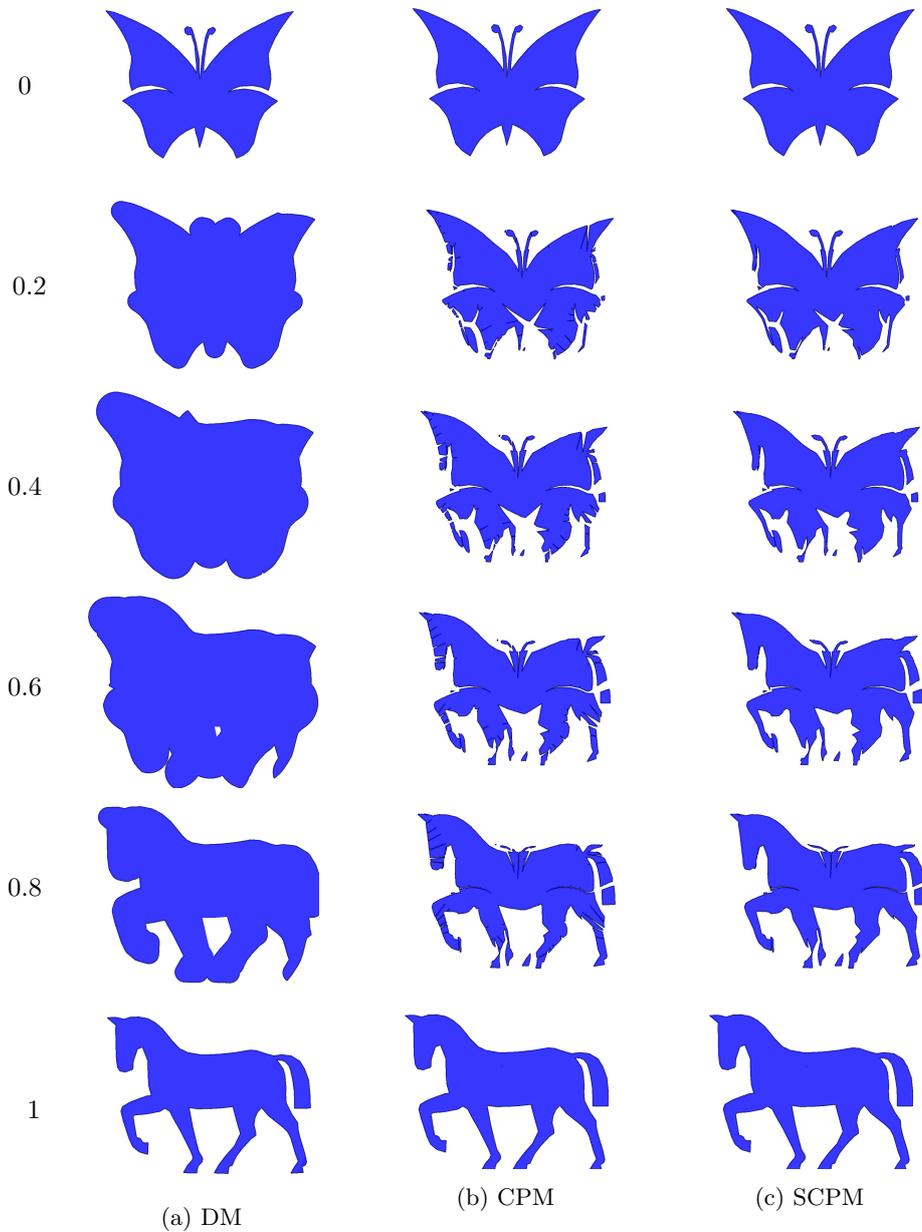


Figure 1: The results of the Dilation Morph (DM), Closest Point Morph (CPM) and the Special Closest Point Morph (SCPM), with a butterfly and horse shape as input.

define the correspondence between features of the shapes. This way you can be sure that the hind legs of the source shape transform into the hind legs of the target shape, even if they are in very different positions. Doing something like this would be difficult without user input, since it would require you to teach the computer to recognise what a hind leg is first. But when reconstructing a 3D shape from hundreds of slices, it would take too much time if user input is needed for each slice. And since the slices would likely be quite similar to their neighbors, it wouldn't be as useful.

One morph that does not require any input is introduced in a recent paper, Kreveld et al. [13]. They prove that given any two input shapes with Hausdorff distance 1, it is always possible to generate a shape that has a Hausdorff distance of α to the source shape and $1 - \alpha$ to the target shape, for any $0 \leq \alpha \leq 1$. They introduce S_α , the *Dilation Morph*, which gives the maximal shape that satisfies this property. This morph works on any input shapes and can be generalised to any dimension. But the actual shape (Figure 1.a) leaves some room for improvement. As S_α is the maximal set with these properties, its total area is quite a lot larger than that of the source and target and a lot of details are faded out by excessive dilation. A more logical morph might be one that doesn't grow any bigger than either A or B .

The Hausdorff distance is a very simple way of measuring how different two shapes are from each other. One of the usages of the Hausdorff distance is in computer vision, to find certain templates in images. The reason we might want to make a Hausdorff middle, is because it makes intuitive sense that the Hausdorff distance would change linearly during the morph. We have not found any morphs from previous research that satisfy this Hausdorff property. If a morph is not a Hausdorff morph, then the Hausdorff distance to the source or target shape must be higher than a linear interpolation. One reason we might want to use the Hausdorff distance over for example the Fréchet distance, which is a similar measure of how different shapes are, is largely because the Hausdorff distance is much simpler to use.

We introduce the *Closest Point Morph*, C_α , which also satisfies the *Hausdorff property*. Given two input shapes with an area of 1, the area of C_α will always be between 0.5 and 1 (Figure 1.b).

This thesis contains a definition and analysis of the Closest Point Morph and a comparison with the Dilation Morph. Additionally an experimental variant of C_α , the *Special Closest Point Morph* (SCPM), is introduced and analysed (Figure 1.c). The SCPM improves certain aspects of the standard Closest Point Morph when applied to polygons, but it is not properly defined to handle all possible simple polygons.

2 Related works

Many different techniques for shape morphing exist, this section provides an overview of some of these techniques and some related shape deformation techniques.

The shapes can be represented in different ways, the use of polygons is most common, but shapes can for example also be represented as points on a grid, curves or more abstractly as an implicit function. A lot of research on image morphing and deforming exists as well, however these are outside of our scope. A lot of morphing techniques work in both 2 dimensions and 3 dimensions, with some modifications. We will mostly focus on these techniques when applied to 2D shapes.

As this thesis works directly on the morph defined by Van Kreveld et al.[13], we will treat it separately in more detail in Section 3.

2.1 Morph using Wasserstein distance

A related problem is that of the optimal transportation with respect to the Wasserstein distance. The Wasserstein distance measures the minimal amount of work needed to move all mass of a source shape to a target shape. Calculating the optimal transport plan is quite complex and usually requires approximations. A fast scalable solution is provided by Solomon et al. [11].

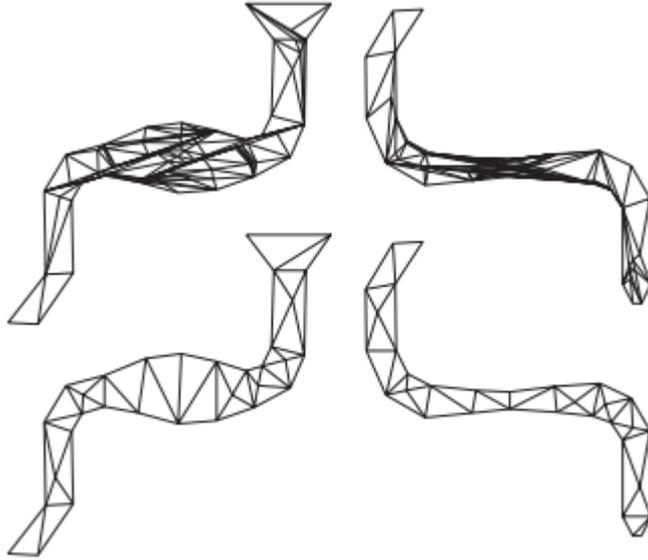


Figure 2: An example of different possible triangulations, with triangulations that would be good (bottom) and bad (top) for shape interpolation. Adapted from [1].

2.2 Morph using compatible triangulations

Polygon triangulation is the process of decomposing a polygon into a set of triangles. A popular process for creating morphs between two shapes is to triangulate both shapes and then morph the triangles from one shape into the corresponding triangle of the other. To do this the triangulations of both shapes need to be compatible, each vertex in one triangulation corresponds with exactly one point in the other triangulation and if an edge exists connecting two points in one triangulation the corresponding points must also be connected by an edge in the other triangulation [2]. To make a high quality morph, it's not only important to create high quality compatible triangulations but also to use a good algorithm to create a smooth interpolation from one triangulation to the other. As such a lot of research exists for both of these problems separately.

For the creation of compatible triangulation Aronov et al. have shown that it's always possible to create a compatible triangulation between simple polygons, but additional points may be needed inside the polygon [2]. Their algorithm takes a triangulation of shape A and shape B and maps both of them to a regular n-gon, then it overlays these triangulations, where new points are created where lines of the triangulations cross, so that it can be mapped to both shapes. While this is a fairly simple solution it usually does not give optimal results as it can for example create long thin triangles, which generally do not give good morphing results (Figure 2). Alexa et al. [1] solve this issue by using Delaunay triangulation for the initial triangulation step. This maximizes the minimum interior angle, thus avoiding thin triangles for the source and target triangulations. However the combined triangulation might still have thin triangles, so then they move the inner vertices and flip edges to further maximize the minimum interior angle.

A naive way to interpolate between compatible triangulations would be a linear transformation. While this gives acceptable results in some cases, it can't properly deal with rotation and tends to create smaller shapes compared to the source and target with a lot of local distortion. The as-rigid-as-possible technique by Alexa et al. [1] solves this problem by blending the interiors of the shapes, not the boundaries, minimizing the local distortion. To get good results this technique also requires the user to define anchor points, for point-to-point correspondence between the source and target shapes.

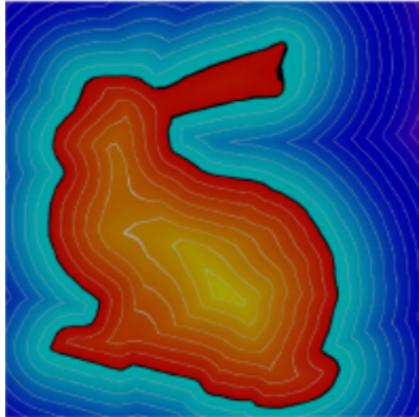


Figure 3: A visual representation of the signed distance field of the Stanford bunny. On the inside of the bunny the distance function gives the negative distance to the border, on the outside it gives the positive distance. At the black border the distance function gives 0 [9].

2.3 Using an implicit function

Shapes can be defined using implicit functions, for example a shape could be defined as all points for which an implicit function equals 0. Defining the shape in such a way allows for different types of morphs. It's possible to make an implicit function for both the source and target shape, then interpolate between the functions to get an intermediate shape. An advantage of using such a function is that you are not as constrained by having to find direct correspondence between the shapes and even shapes with very different topology can be morphed. A disadvantage is that such techniques are usually rather slow [12].

One type of implicit function is a signed distance function, which allows for a technique called distance field interpolation (DFI) [9]. For a shape A the signed distance function gives the Euclidean distance to the boundary of A, if the point is not part of A, and the negative distance to the boundary of A otherwise. A distance field stores the result of the distance function for any given point (Figure 3). Therefore the distance field will give positive values for every point not in A, zero for any point on the boundary of A and negative values for any other point in A. Then, instead of interpolating between the shapes themselves, the distance fields are interpolated. For any distance field the shape can be reconstructed: if the distance value for any given position in the field is smaller than or equal to zero then that point is part of the corresponding shape.

A naive linear interpolation can give reasonable results if the target and source shape mostly overlap, otherwise the shape will in most cases end up being rather small and strangely shaped. To solve this, Cohen-Or et al. [6] developed a technique that guides the interpolation of the distance fields by user defined warps. This allows for smooth translation and rotation of parts of the shape or the whole shape, but requires more work for the user. More specifically the warps are created by the user by making corresponding anchor points on the source and target shapes, then the part of the shape around that anchor point of the source shape will be morphed in the direction of the corresponding anchor point on the target shape.

Another technique described by Turk et al. uses a variational implicit function[12]. For shapes in n dimensions an implicit function is created in $n + 1$ dimensions, for this section we will assume a 2D shape with a 3D implicit function, the added dimension is called t . Consider that both the source and target shape have an implicit function, which is made using user defined constraint points. The constraint points of the implicit functions of the source shape and target shape are combined in the new implicit function, where the t value of the constraint points is set to 0 for the source shape and t_{max} for the target shape, where t_{max} is some positive value. Variational interpolation is used to create a scalar valued function over \mathbb{R}^3 . Now the 2D slices that exist for $0 < t < t_{max}$ smoothly interpolate between the source and target shape (Figure 4).

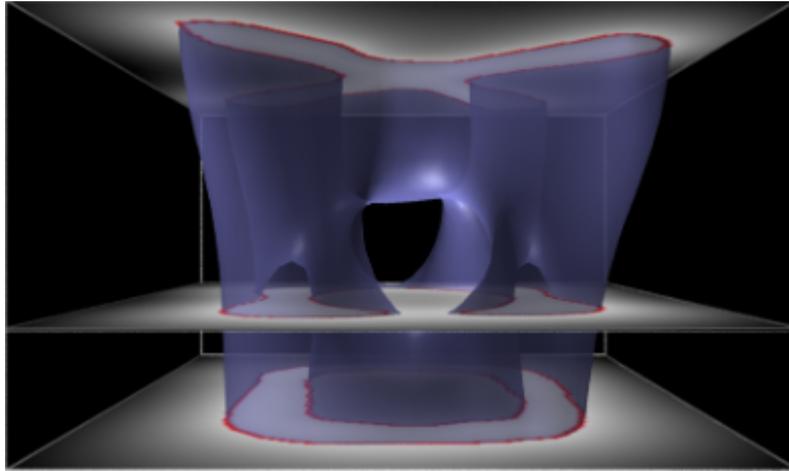


Figure 4: A visualisation of the variational implicit function technique from [12].

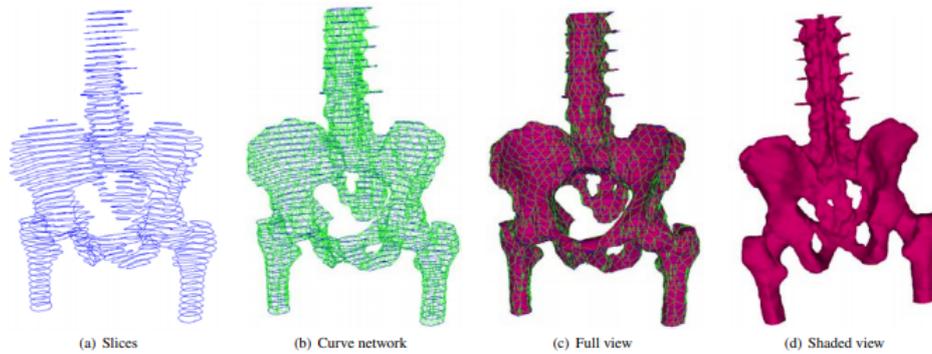


Figure 5: A 3D model of a pelvis that has been reconstructed from 2D slices [3].

2.4 3D reconstruction

Reconstructing 3D objects from 2D slices is an important problem in the medical world. An MRI scanner can be used to create 2D images of specific slices of the body. Using these slices as input, body parts can be turned into 3D models with morphing techniques (Figure 5). The surface of the object can be reconstructed from polygon slices by filling up the space between them with morphs of these slices. However, simply applying any morphing technique with compatible triangulations to fill in the details would not create a smooth mesh. In a technique by Borek and Vaxman[3], a curve network is created which ensures smooth arcs between the slices. Vertices on each slice are matched with the neighboring slices, this creates a flow graph. The flow graph is used to create the curve network, a network of cubic Bézier curves, which is then turned into the final mesh.

2.5 General shape deformation techniques

Shape morphing is a type of shape deformation. While morphing is about finding a shape in between a source and target shape, deformation is more generally about reshaping a source shape, for example with the goal of animating a cartoon character. In such cases it might be sensible to set up a skeleton such as in [15]. In these methods a user can define bones of a shape, such that if the bone moves a corresponding part of the shape moves with it, for example for moving the arms of a cartoon character. A type of morphing can easily be done using skeletons by morphing between different poses of the skeleton, then the result of the bone positions on the shape can be

outputted. However, this will still be limited to the original source shape, so this is not fit for morphing totally different shapes.

One way to make the bones have an impact on the shape is a process called weight painting, where weights are added to each vertex for every bone, this way the movement of a bone can change each vertex a different amount based on the weights. This, however, requires a lot of user input.

Numerous techniques exist to automate the process of weight painting, such as [7], where weights are chosen to minimize the Laplacian energy. Weight painting can also be avoided entirely, for example by transforming the triangles instead of the vertices such as in [15]. Similar techniques are possible without even defining a skeleton. In [14] vertices are added inside the polygon, which are then connected to the vertices of the boundary. Then the user changes the characters pose by dragging any of the points.

2.6 Morphs with curves

Some techniques for curve-based shape morphing exists as well. For example [8], where curves are used as the input. The curve is then reduced to line segments at the pixel level and for each line segment a correspondence is found. Based on this correspondence, the curves are subdivided into smaller curves and these are approximated with Bézier curves, which are then interpolated for the final result.

3 Preliminaries

This thesis builds on the work of Kreveld et al. [13] concerning the creation of morphs between shapes using the Hausdorff distance. The paper introduces S_α , which will also be referred to as the *dilation morph*.

The directed Hausdorff distance $d_{\vec{H}}$ from set A to set B in \mathbb{R}^2 is defined as

$$d_{\vec{H}}(A, B) = \sup_{a \in A} \inf_{b \in B} d(a, b),$$

where d is the Euclidean distance. In other words, it is the largest shortest distance to B over all points in A. The undirected Hausdorff distance is defined as

$$d_H(A, B) = \max(d_{\vec{H}}(A, B), d_{\vec{H}}(B, A)).$$

A Hausdorff middle of A and B is a shape that minimizes the undirected Hausdorff distance to both A and B. In most cases multiple Hausdorff middles are possible. When the undirected Hausdorff distance between A and B is 1, it is always possible to find a shape with undirected Hausdorff distance 1/2 to both A and B. We call such a shape a Hausdorff middle. The maximal Hausdorff middle is unique and it's defined as

$$S = (A \oplus D_{1/2}) \cap (B \oplus D_{1/2}),$$

where \oplus denotes the Minkowski sum, and D_r is a disk of radius r centered at the origin. A more generic version of this set is defined as

$$S_\alpha = (A \oplus D_\alpha) \cap (B \oplus D_{1-\alpha}),$$

for $\alpha \in [0, 1]$. S_α has some desirable properties with respect to the Hausdorff distance such as $d_H(A, S_\alpha) = \alpha$ and $d_H(B, S_\alpha) = 1 - \alpha$, assuming $d_H(A, B) = 1$. And for $\alpha \leq \beta$ then $d_H(S_\alpha, S_\beta) = \beta - \alpha$. In simpler terms, the Hausdorff distance changes linearly with α . This will be referred to as the *Hausdorff property* and any morph that satisfies this property will be known as a *Hausdorff morph*.

Using S_α a smooth blend can be created from A to B by increasing the value of α from 0 to 1 in small steps. It has also been proven that S_α will be connected for $\alpha \in [0, 1]$, if A and B are connected and at least one of A and B is convex.

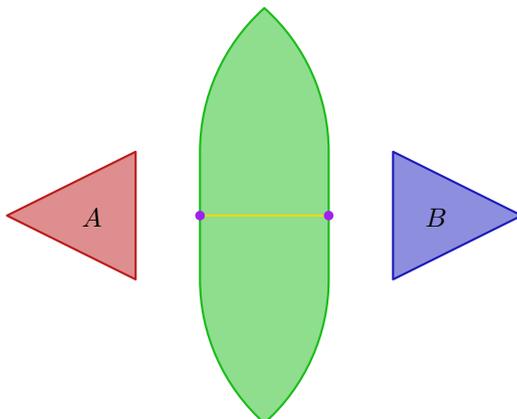


Figure 6: An example of S , the maximal Hausdorff middle, for two triangles [13].

The shape can become very large and details are faded out because of the dilation (Figure 1). We will try to improve on these issues with our own Hausdorff morph, the Closest Point Morph.

4 Closest Point Morph

For two sets A and B in \mathbb{R}^2 , we define the set $C_{\frac{1}{2}}^A$ that is found by taking each point in A and moving it halfway to its closest point in B .

$$\vec{C}_{\frac{1}{2}}(A, B) = \left\{ \frac{x + c_B(x)}{2} \mid x \in A \right\},$$

where $c_B(x)$ is the closest point to x in B . If x has multiple closest points in B then a new point is created for every option.

Similarly $\vec{C}_{\frac{1}{2}}(B, A)$ is the set that is found by taking each point in B and moving it halfway to its closest point in A . By combining these two sets we can define a Hausdorff middle C ,

$$C_{\frac{1}{2}} = \vec{C}_{\frac{1}{2}}(A, B) \cup \vec{C}_{\frac{1}{2}}(B, A).$$

A generalised version can be made by introducing α . Instead of moving each point halfway to its closest point in the other shape, the points are moved by a certain α . We define

$$\vec{C}_{\alpha}(A, B) = \{x + (c_B(x) - x) * \alpha \mid x \in A\}$$

$$\vec{C}_{1-\alpha}(B, A) = \{x + (c_A(x) - x) * \alpha \mid x \in B\}.$$

Using these generalised versions of $\vec{C}_{\alpha}(A, B)$ and $\vec{C}_{1-\alpha}(B, A)$ we can also create a generalised version C_{α} ,

$$C_{\alpha} = \vec{C}_{\alpha}(A, B) \cup \vec{C}_{1-\alpha}(B, A),$$

for $\alpha \in [0, 1]$. We will use $d(x, y)$ to denote the distance from point x to point y and $dist_A(p)$ will be used to denote the distance from p to $c_A(p)$.

Theorem 4.1. *Let A and B be two compact sets in the plane with $d_H(A, B) = 1$. Then $d_H(A, C_{\alpha}) = \alpha$ and $d_H(B, C_{\alpha}) = 1 - \alpha$.*

Proof. We will prove $d_H(A, C_{\alpha}) = \alpha$, the proof for $d_H(B, C_{\alpha}) = 1 - \alpha$ is analogous and therefore omitted.

By definition any point $x \in A$ has a corresponding point $v_A \in \vec{C}_{\alpha}(A, B)$, $v_A = x + (c_B(x) - x) * \alpha$, where $d(v_A, x) = dist_B(x) * \alpha$. If $x' \in A$ is the point that realises the directed Hausdorff

distance from A to B , with the corresponding point $v'_A \in \vec{C}_\alpha(A, B)$, then $d(v'_A, x') = d_{\vec{H}}(A, B) * \alpha$. Since x' is the furthest point from B , there can not be any point closer to x' than v'_A in $\vec{C}_\alpha(A, B)$. Thus $d_{\vec{H}}(A, \vec{C}_\alpha(A, B)) = d_{\vec{H}}(A, B) * \alpha$.

Using the same logic we can also see that the point that realises the directed Hausdorff distance from B to A has a corresponding point $v_B \in \vec{C}_{1-\alpha}(B, A)$ with $dist_A(v_B) = d_{\vec{H}}(B, A) * \alpha$. Therefore $d_{\vec{H}}(\vec{C}_{1-\alpha}(B, A), A) = d_{\vec{H}}(B, A) * \alpha$.

Consider any point $x \in A$ with corresponding point $v_A \in \vec{C}_\alpha(A, B)$. v_A is x linearly interpolated by α to the closest point in B . Any point $v_B \in \vec{C}_{1-\alpha}(B, A)$ is a linear interpolation by $1 - \alpha$ from some point $b \in B$ to it's closest point in A . It's clear that it's not possible for any point v_B to be closer to x than v_A . That would imply b is somehow closer to x than $C_A(x)$ or v_B is interpolated to some point behind x , both of which are by definition not possible. Therefore $d_{\vec{H}}(A, \vec{C}_{1-\alpha}(B, A)) \geq d_{\vec{H}}(A, \vec{C}_\alpha(A, B))$. Using the same logic we can also say that for any point in $\vec{C}_{1-\alpha}(B, A)$ there is a point in $\vec{C}_\alpha(A, B)$ that is closer to or equally close to A , thus $d_{\vec{H}}(\vec{C}_{1-\alpha}(B, A), A) \geq d_{\vec{H}}(\vec{C}_\alpha(A, B), A)$.

We can use this, combined with the directed Hausdorff distances we found earlier, to find

$$d_{\vec{H}}(A, C_\alpha) = d_{\vec{H}}(A, \vec{C}_\alpha(A, B)) = d_{\vec{H}}(A, B) * \alpha$$

and

$$d_{\vec{H}}(C_\alpha, A) = d_{\vec{H}}(\vec{C}_{1-\alpha}(B, A), A) = d_{\vec{H}}(B, A) * \alpha.$$

Therefore

$$d_H(A, C_\alpha) = \max(d_{\vec{H}}(A, B) * \alpha, d_{\vec{H}}(B, A) * \alpha).$$

We defined $d_H(A, B) = 1$, thus $d_H(A, C_\alpha) = \alpha$ ■

4.1 Basic algorithm for polygons

For this thesis we will consider the Closest Point Morph as it works on the interiors of polygons. When calculating C_α we need to find out for each part of the input shapes what the closest point is in the other input shape. A Voronoi diagram divides the space up into cells that all have the same closest point or edge. We can use this to cut both input shapes up using the Voronoi diagram of the other input shape. Since each Voronoi cell has a corresponding closest point or closest edge, this allows us to divide the shape into chunks that will all move to the same closest point or edge. The basic algorithm for sets of input polygons A and B works as follows:

1. Find the between the two input shapes. This will be called the *Overlapping Input*
2. Create a Voronoi diagram for the edges of both input shapes. For each Voronoi cell cut off the part that is inside the input shape.
3. For each cell of the Voronoi diagram, find the intersection between the cell and the other input shape. Save this together with the closest point or edge in the other input shape that corresponds with that Voronoi cell. These cut up shapes with a corresponding closest point or edge will be called *morphing chunks*.
4. Move each point in each morphing chunk closer to the corresponding closest point or to the closest point in the corresponding edge. Move it by α of the way to the closest point for morphing chunks coming from A and by $1 - \alpha$ for the chunks coming from B . So if $\alpha = 0.5$ the point should be exactly in between its original position and its closest point in the other shape.
5. Combine all morphing chunks and the overlapping input together to create the final morph.

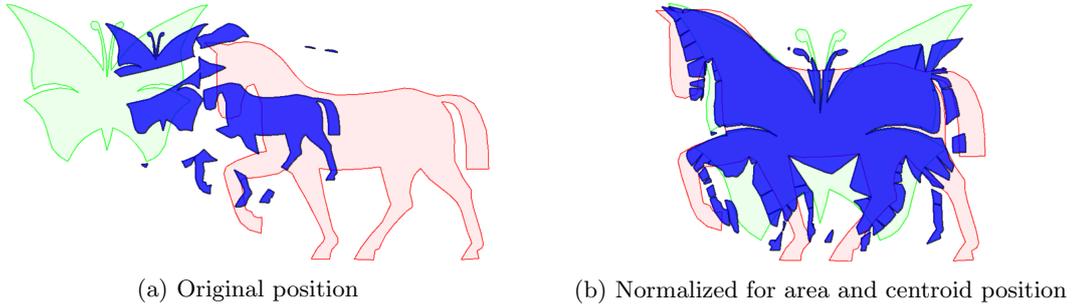


Figure 7: An example that shows the morph generally looks better when the shapes overlap more. (a) shows the morph in the original position, where both shapes just get cut up and the morph looks like big slices of both shapes. (b) shows the morph with the normalized area and centroid position. Now it looks more like a combination of a horse and a butterfly.

4.1.1 Voronoi diagram for polygons

When creating the Voronoi diagram the edges and their endpoints should be considered as separate entities. In other words, for each edge there should be three cells in the Voronoi diagram, one for the line segment and one for each endpoint.

4.2 Closest Point Morph with position and area normalization

The Closest Point Morph can create unnatural looking results when shapes are further apart, such as in Figure 7. When the shapes are far away from each other, the morph often ends up just being a small version of both shapes, or small, mostly disconnected slices of both shapes. This happens because large parts of both shapes are encapsulated entirely in a single Voronoi cell of the other shape, so that entire shape will be morphed towards a single point. Similar problems can arise when one of the shapes is much larger than the other. In general the Closest Point Morph seems to work best when the input shapes largely overlap.

This is why in most cases we will be normalizing both the area and centroid position of the input shapes. Some other way of configuring the input shapes might give slightly better results, but this solution is fairly simple and improves the quality of the morph quite a lot.

If the original size and position are not important we can simply normalize the area and centroid position of the target to the source and be done with it. If the area and position need to be preserved we can still make a smooth morph by generating the morph with normalized area and position. First do the morph with the target shape normalised to the source shape, then translate the result such that the centroid position linearly interpolates between the shapes. And finally scale it by a factor of $\frac{\alpha * TargetArea + (1 - \alpha) * SourceArea}{SourceArea}$ to interpolate the area.

This usually gives a much nicer looking smooth morph between two shapes that would otherwise not overlap much. However, the result is not necessarily also a Hausdorff Morph (see Figure 8).

4.3 Other ways of implementing a Closest Point Morph

For this thesis we will mostly look at the Closest Point Morph applied to 2D polygons. The Closest Point Morph could also be implemented in other ways. For example a 3D or pixel based implementation, these were however not implemented for this thesis.

Another possibility is to consider a set of edges, such as the boundary of a polygon. This variant was also implemented, and it will be briefly discussed. No in depth analysis has been done on the edge variant as the results didn't seem as promising as the version that works on polygon interiors. The algorithm for this variant is very similar to the algorithm for polygon interiors, but slightly simpler. The inputs are sets of edges A and B . It works as follows:

1. Create a Voronoi diagram for the edges of both input shapes.

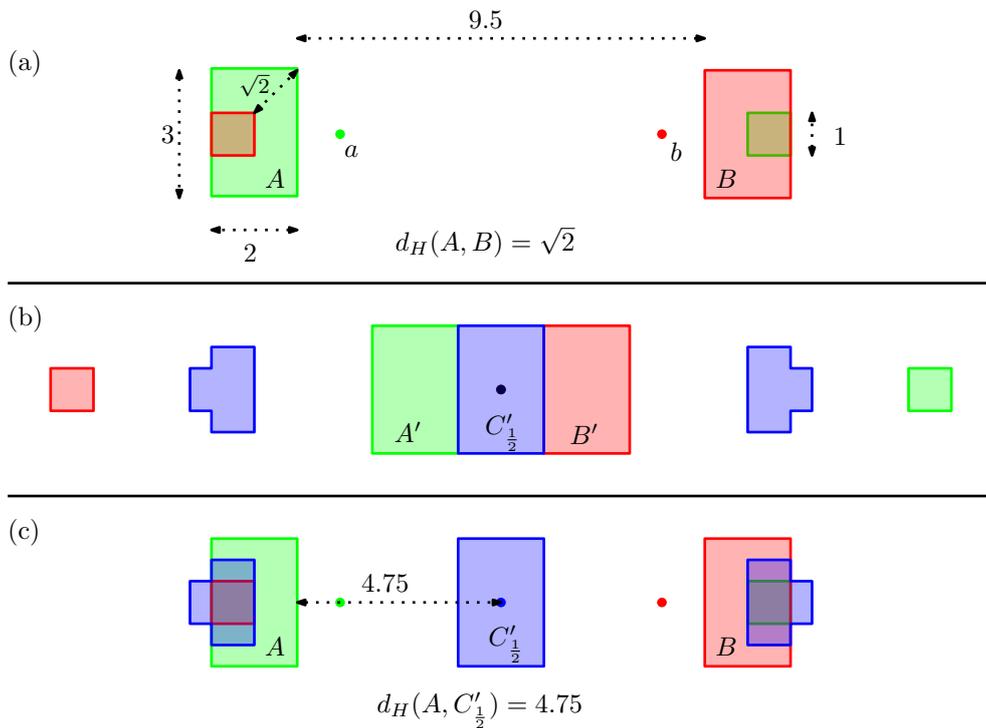


Figure 8: An example that shows the Hausdorff property is not necessarily preserved when the morph is created with a normalized centroid position. (a) shows the input shapes A and B , where point a is the centroid of A and b is the centroid of B . (b) shows A' and B' , which are the input shapes translated to have the same centroid position. $C'_{\frac{1}{2}}$ is created as a closest point morph between A' and B' . (c) shows A and B with $C'_{\frac{1}{2}}$ placed such that the centroid is half way between the centroids of A and B . To retain the Hausdorff property $d_H(A, C'_{\frac{1}{2}})$ should be $d_H(A, B) * \frac{1}{2}$, this is clearly not the case.

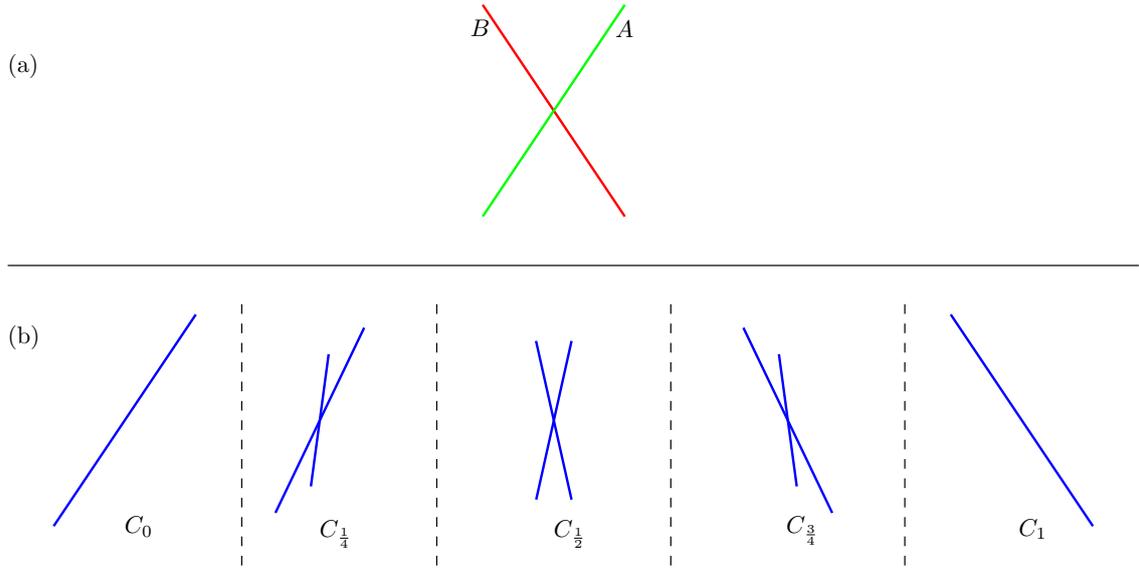


Figure 9: The Closest Point Morph applied to two edges. (a) shows the input shapes, (b) shows the results with different values for α .

2. For each cell of the Voronoi diagram, find the intersection between the cell and the other input shape. Save this together with the closest point or edge in the other input shape that corresponds with that Voronoi cell. These will be called *Morphing edges*.
3. Move each point in each *morphing edge* closer to the corresponding closest point or to the closest point in the corresponding edge. Move it by α to the closest point for morphing chunks coming from A and by $1 - \alpha$ for the chunks coming from B .
4. Combine all morphing chunks together to create the final morph.

When an edge is morphed to another edge in this way, its angle does not change linearly with α . This means that when two nonparallel *morphing edges* are morphed towards each other they will never be parallel if $0 < \alpha < 1$. This creates an unnatural looking morph even for a very basic case, as seen in Figure 9.

5 Properties of the Closest Point Morph

For the rest of this thesis we will consider the Closest Point Morph as it applies to the interiors of polygons. In this section we will discuss some of its properties.

5.1 Area

The area of C_α is the combination of the area of the overlapping input and the area of all morphing chunks at the given α . For target and source input polygons T and S we know the following attributes:

- I , the area of the overlapping input.
- T^P and S^P , the area of all morphing chunks moving towards a point for the target and source respectively.
- T^L and S^L , the area of all morphing chunks moving towards a line for the target and source respectively.

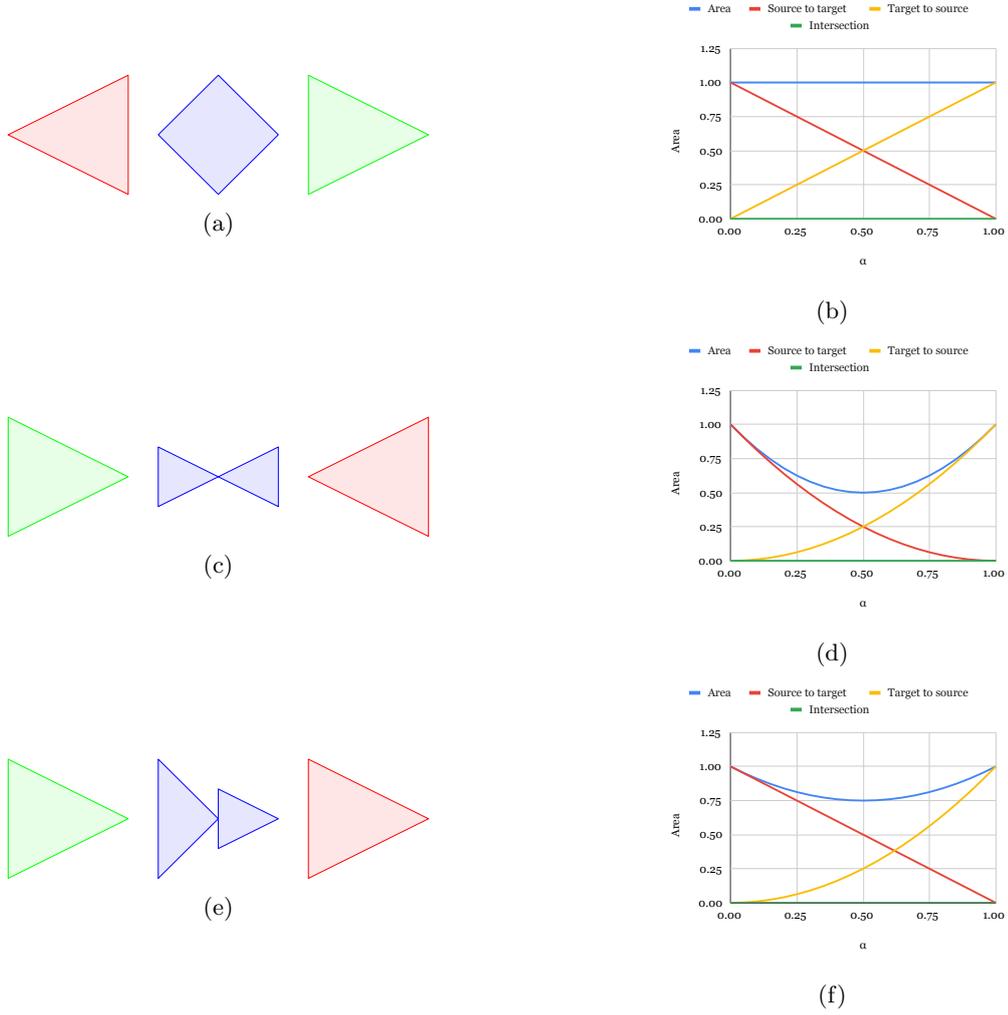


Figure 10: Some examples of basic cases and their area with different values for α . The green shape is the source, the red shape is the target and the blue shape is the result of the CPM with $\alpha = \frac{1}{2}$.

T_α^P , S_α^P , T_α^L and S_α^L are the areas of the same morphing chunks at a given α . When shrinking a polygon by moving each point to its closest point on a line the area reduction is linear, so $T_\alpha^L = T^L * \alpha$ and $S_\alpha^L = S^L * (1 - \alpha)$. When shrinking a polygon by moving each point to a single point the area reduction is quadratic, so $T_\alpha^P = T^P * \alpha^2$ and $S_\alpha^P = S^P * (1 - \alpha)^2$. I is constant so no further calculations are necessary. The total area at α is

$$\begin{aligned}
 Area(\alpha) &= I + T_\alpha^P + S_\alpha^P + T_\alpha^L + S_\alpha^L \\
 Area(\alpha) &= I + T^P * \alpha^2 + S^P * (1 - \alpha)^2 + T^L * \alpha + S^L * (1 - \alpha) \\
 Area(\alpha) &= (T^P + S^P) * \alpha^2 + (T^L - 2S^P - S^L) * \alpha + (S^P + S^L + I).
 \end{aligned}$$

As you can see the relation between the area and α is parabolic. See Figure 10 for some examples of what this looks like for basic cases. This formula allows us to determine precise boundaries for the area.

Lemma 5.1. *Let A and B be a source and target shape, both with the same area. The area is lowest when $\alpha = \frac{1}{2}$.*

Proof. We can find the lowest point of the parabola when the derivative of the parabola is 0. The derivative of the area is $(T^P + S^P) * 2 * \alpha + (T^L - 2S^P - S^L)$. So we need to find α for

$$(T^P + S^P) * 2 * \alpha + (T^L - 2S^P - S^L) = 0.$$

Since the area of A and B is the same we know that $T^P + T^L = S^P + S^L$ and therefore also $T^L - 2S^P - S^L = -S^P - T^P$.

$$(T^P + S^P) * 2 * \alpha + (T^L - 2S^P - S^L) = 0.$$

$$(T^P + S^P) * 2 * \alpha + (-S^P - T^P) = 0$$

$$(T^P + S^P) * 2 * \alpha = T^P + S^P$$

$$\alpha = \frac{(T^P + S^P)}{2 * (T^P + S^P)} = \frac{1}{2}$$

■

Theorem 5.2. *Let A and B be a source and target shape, both with Area = 1. The lowest possible area for a middle shape is $\frac{1}{2}$ and the highest possible area is 1.*

Proof. It's clear that the middle shape only shrinks when *morphing chunks* are morphed to points. The morph to lines is linear and the area of the overlapping input doesn't change. So consider the situation where $S^P = 1$, $T^P = 1$ and all other areas are 0 (for example, Figure 10.c). The area is lowest when $\alpha = \frac{1}{2}$ (lemma 5.1), so when we put that in the formula it becomes

$$Area\left(\frac{1}{2}\right) = (1 + 1) * \frac{1}{2} + (-2) * \frac{1}{2} + 1 = \frac{1}{2}$$

So putting area in S^P or T^P makes the area go down when $0 < \alpha < 1$ and putting area in I , S^L or T^L keeps the area constant. It's clear that there is no way to make the area go higher than the area of the input shapes, so the highest possible area is 1 (for example, Figure 10.a). ■

5.2 Perimeter

The perimeter is a bit harder to predict. The Closest Point Morph can create large cuts when a shape is morphed to different edges from a concave part of the other shape. When a couple of connected morphing chunks are morphed towards different edges in a concave part of the other shape, the Closest Point Morph will create a split between those morphing chunks. This happens because the points of these chunks are moved directly to the closest point on the corresponding line, which is not the same as the line that divides them.

Theorem 5.3. *The perimeter of $C_{\frac{1}{2}}$ can increase linearly with the number of vertices in the concave part of the boundary of the input shapes, while the perimeter of the input shapes is capped.*

Proof. Consider a source shape Q_n , which is a square with a hole in the middle, the hole has $n > 2$ vertices evenly spaced out on a circle with radius 1 (Figure 11). The target shape is B , which is the same as Q_n except without the hole. The perimeter of the square is 10. Note that the perimeter of the hole is not the same for each n , but it approaches 2π as n grows.

Because the points are evenly spread out over the circle, each morphing chunk is an identical triangle. q is the length of the base of the triangle, the part directly touches Q_n . At $\alpha = 1$ the other sides of the triangle are 1 by definition. The altitude of the triangle will be called t (Figure 11.e). The points are moved perpendicularly to the closest line. The morphing chunks will never touch each other with $\alpha < 1$, except at the base, since the lines that divide the morphing chunks are never perpendicular.

At $\alpha = 0.5$, we will call the length of the two edges other than the base s (Figure 11.f). Note that s is not the same for all values of n . Since none of these edges touch, each triangle adds

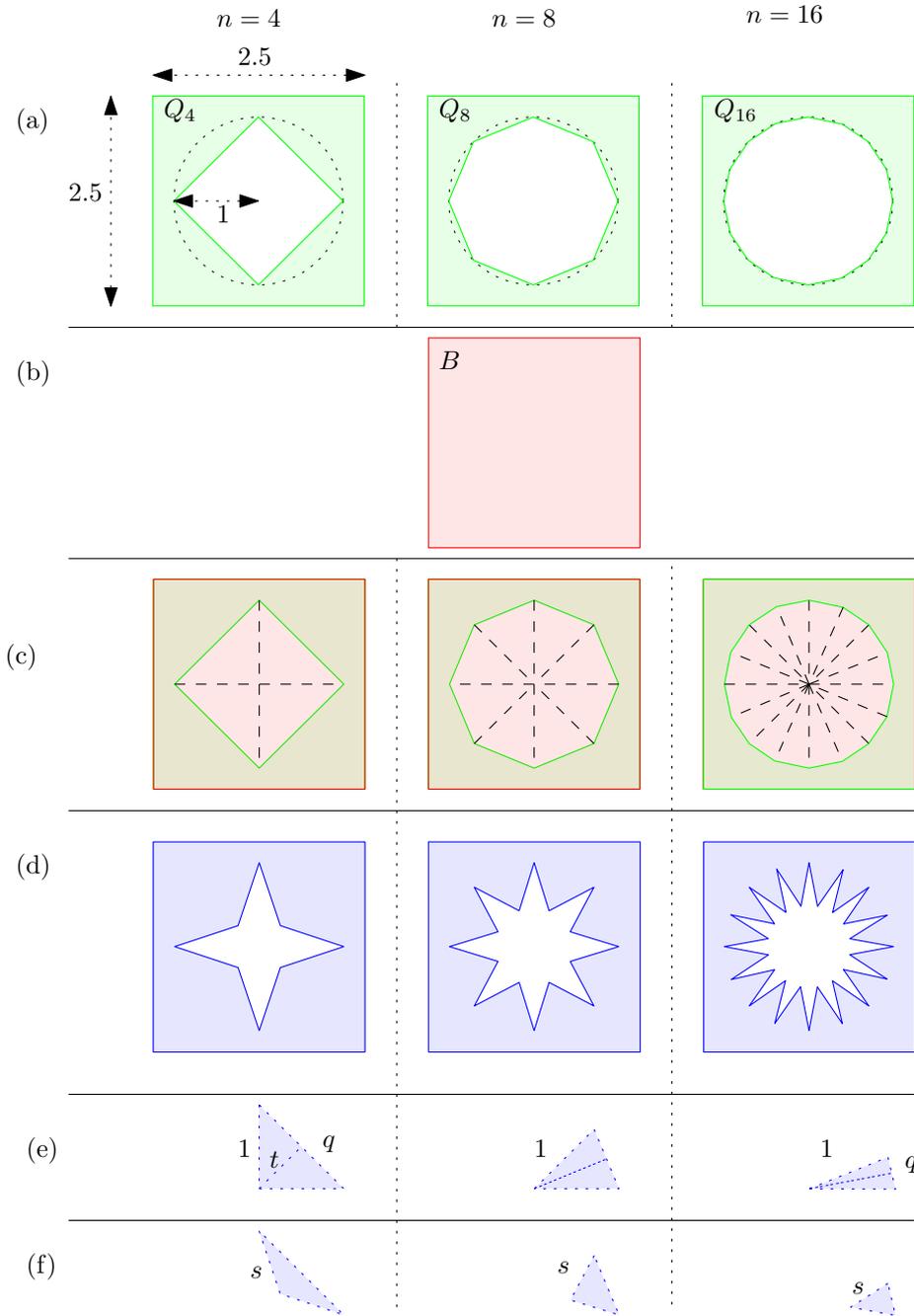


Figure 11: This shows an example of how the perimeter can grow in the CPM by increasing the vertex count. (a) shows the source shape Q_n for different values of n . (b) shows the target shape B . (c) shows the source and target shapes overlapped with the relevant voronoi cells. (d) shows $C_{\frac{1}{2}}(Q_n, B)$. (e) shows one of the morphing chunks at $\alpha = 1$. (f) shows one of the morphing chunks at $\alpha = 0.5$.

$2s$ to the total perimeter of the morph at $\alpha = 0.5$. And since there are n morphing chunks, the perimeter of the hole will be $2 * n * s$.

We can find $t = \sqrt{1 - (q/2)^2}$ and $s = \sqrt{(t/2)^2 + (q/2)^2}$ at $\alpha = 0.5$. q approaches 0 and s approaches $1/2$ as n grows. Assume n is so high that the difference between s and $1/2$ becomes insignificantly small, then each vertex adds 1 to the perimeter. So the perimeter of the hole would be n . Including the perimeter of the exterior, that makes the perimeter of $C_{\frac{1}{2}}$ $n + 10$. So the perimeter of $C_{\frac{1}{2}}$ can grow linearly with the vertex count, while the perimeter of the input is capped at $2\pi r + 10$. ■

5.3 Connected Components

The number of connected components, or pieces, for the Closest Point Morph is usually higher than that of the input. In this section some of the proofs are given related to the number of pieces.

Theorem 5.4. *The number of detached pieces in C_α is the same for any α with $0 < \alpha < 1$.*

Proof. Consider source and target shapes A and B . For any given morphing chunk we will look at three different situations: (i) its connectivity with the overlapping input, (ii) its connectivity with another morphing chunk coming from the same shape, and (iii) its connectivity with a morphing chunk not from the same shape.

(i) The points in the morphing chunk that touch the overlapping input in their initial position will not move at all and will be connected for any α . All of the points in the morphing chunk are moved linearly with α towards the closest point in the other shape, so if a point in a chunk coming from A does not start connected to the overlapping input, then it will only touch the other shape at $\alpha = 1$. The same is true for morphing chunks coming from B at $\alpha = 0$. Therefore, if a morphing chunk is touching the overlapping input, then it will stay connected.

(ii) Consider a chunk from A and a chunk from B . Assume there is a point in the A-chunk and point in B-chunk have each other as the closest point in the other shape. Because of the way the points are moved linearly towards each other with α and $1 - \alpha$ respectively, they will overlap for any α . Therefore if two chunks from different input shapes contain two points that have each other as the closest points, they will be connected for any α . In any other situation, even if the points are moved in each others direction, they will not overlap because at least one of the points will be moved a shorter distance towards some closer point, which for a linear transformation like this means they will never touch for $0 < \alpha < 1$. If one of the points coming from A is moving towards another point in B but that point is moving towards some other closer point, then these points will only overlap when $\alpha = 1$. The same is true for the opposite situation with points coming from B at $\alpha = 0$.

(iii) Since each point of the morphing chunk is moved in a straight line to the closest point in the other shape, it is not possible for any of the points to leave the corresponding Voronoi cell. This means the only way two chunks from the same source shape can be connected, is through the edges of these cells. If two points are connected through an edge of a cell in their initial positions, and are then both are moved along the direction of this edge, then this means they will move towards the same point and will therefore stay connected. In the case of a morphing chunk that is morphed towards an edge, a point will only be moved along a Voronoi edge if that edge is perpendicular to the edge in the input that it is morphed towards. In the case of a morphing chunk that is morphed towards a vertex, a point will only move along a Voronoi edge if that edge is pointed directly towards the corresponding point in the input. This situation only happens for two connected Voronoi cells when one cell belongs to an edge and the other belongs to one of the endpoints of that edge. In this situation the morphing chunks are connected for any α . If they are not moving towards the same point, then they will not follow the edge of the Voronoi cell and they will be disconnected for any α other than $\alpha = 0$ for morphing chunks from A . In this situation these points will stay disconnected from the Voronoi edges until the destination point is reached. The same is true for morphing chunks coming from B at $\alpha = 1$. In either case, the connectivity of these morphing chunks is the same for any $0 < \alpha < 1$.

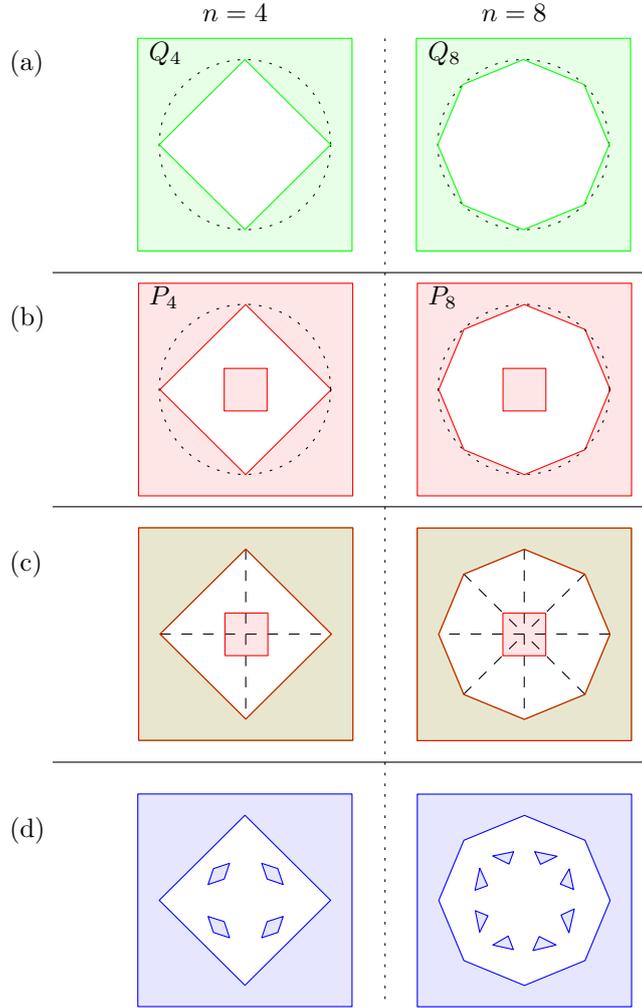


Figure 12: An example to show how the number of disconnected components can grow by increasing the vertex count. (a) shows the source shape Q_n for different values of n . (b) shows the target shape P_n . (c) shows the source and target overlapping with the relevant voronoi cells. (d) shows $C_{\frac{1}{2}}(Q_n, P_n)$.

So the connectivity between the morphing chunks is the same for any $0 < \alpha < 1$, both when the morphing chunks are from the same source shape, and when they are from another. And the connectivity between the morphing chunks and the overlapping input stays the same for any $0 < \alpha < 1$. Additionally the connectivity between different parts of the overlapping input is always the same since it never moves. Since there are no other combinations of parts that make up the morph, we can conclude that the number of connected components is the same for any $0 < \alpha < 1$. ■

The number of connected components with $0 < \alpha < 1$ can also grow linearly with the number of vertices in concave parts of one of the input shapes. The proof is similar to that of Theorem 5.3.

Theorem 5.5. *The number of connected components of $\vec{C}_\alpha(A, B)$ can be grown linearly by increasing the number of vertices in the concave part of the boundary of the input shapes, while the number of connected components of the input shapes stays the same.*

Proof. This proof uses Q_n from theorem 5.3 as the source shape and P_n as the target shape, which

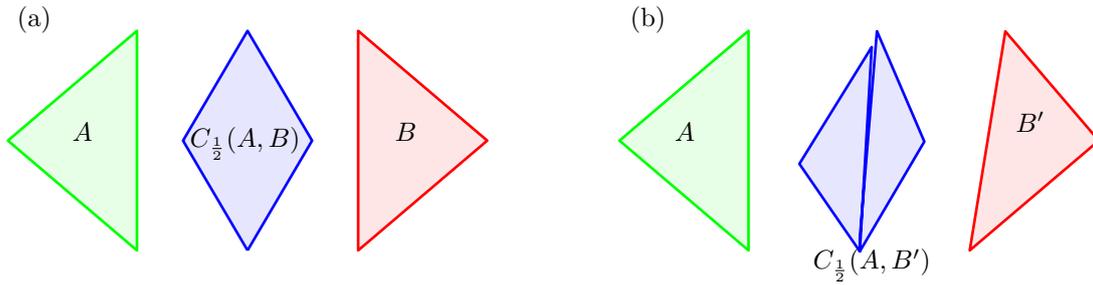


Figure 13: (a) shows triangles A and B and the CPM result $C_{\frac{1}{2}}(A, B)$. The edges that face each other are parallel, so the morph is nicely connected in the middle. (b) shows triangles A and B' and the CPM result $C_{\frac{1}{2}}(A, B')$. B' is the same as B , except for the highest vertex which was moved to the right slightly. The edges that face each other are not parallel, which means $C_{\frac{1}{2}}^A$ and $C_{\frac{1}{2}}^B$ are only connected in one point and a long, very thin gap exists between them.

is the same as Q_n except with a small square in the middle of the hole (Figure 12).

Since the square is the only part that does not overlap, all of the morphing chunks will be from the intersections between the square and the Voronoi cells of the hole. Just like in Theorem 5.3 the morphing chunks never touch each other, so for $0 < \alpha < 1$ each morphing chunk adds one connected component. There are as many morphing chunks as there are edges in the hole and the overlapping input is an additional connected component, so in total there are $n + 1$ connected components. Which means in this setup the number of connected components grows linearly with the vertex count. ■

5.4 Miscellaneous artifacts

The problem of the edge based CPM shown in Figure 9 also happens in the polygon based CPM, but in slightly more specific circumstances. When two nonoverlapping pieces of the source and target shape have parallel edges that are morphed towards each other, then the morph will be fully connected along those edges. This can give a nice natural feeling morph such as in Figure 13.a. If these edges have even slightly different angles however, then they will only overlap on their closest points, which means a very thin gap will exist between the morphing chunks. This can give some weird looking results such as in Figure 13.b.

In Figure 12 an example was shown where the morph disconnects if a part of a shape is morphed to a concave part of the other shape. More specifically, when that part of the shape is not connected to that concave boundary, and the boundary itself does not morph back to that shape. These splits can of course also happen when pieces are morphed to edges that are further apart. In Figure 14 a simple example is shown where most of the pointy bit of B is morphed down but a tiny piece in another Voronoi cell and is morphed up. In more complicated shapes these types of tiny floating pieces can be quite common.

6 Special Closest Point Morph

As previously mentioned the Closest Point Morph can create a lot of cuts in previously connected morphing chunks when morphing towards a concave part of the other shape. This happens because the points are moved directly to the closest point on the corresponding edge, which is not the same as the line that divides them. A more intuitive morph would in this case move these points according to the dividing lines (Figure 15.e), that is what the *Special Closest Point Morph* aims to do.

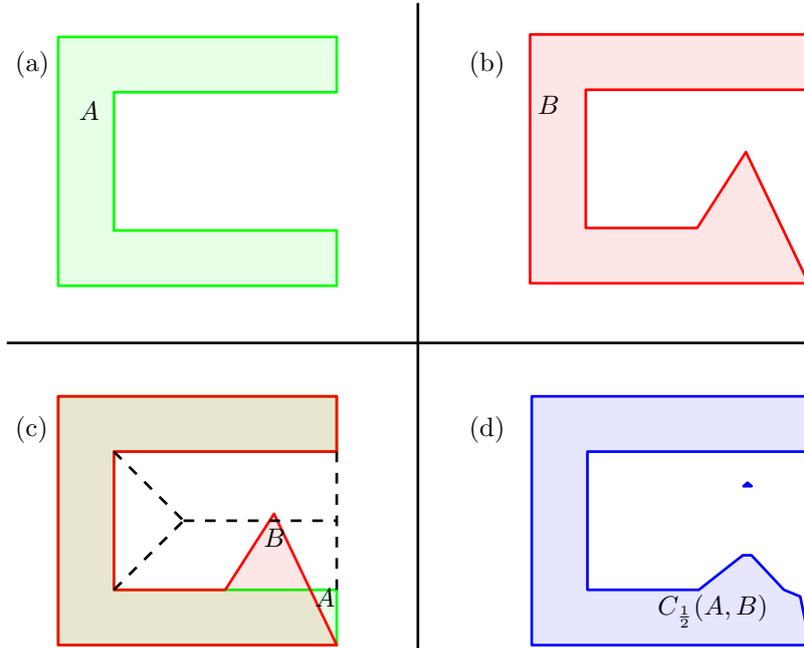


Figure 14: Another example where the number of connected components increases. (a) and (b) show input shapes A and B . (c) shows how they overlap, and how a tiny part of the pointy bit in B is in another Voronoi cell. (d) shows how this results in a small strange looking floating piece.

The Special Closest Point Morph (SCPM) only changes the way the morph works in morphing chunks that are moved to a line, not the ones moved to a point. Consider a Voronoi cell with a corresponding closest edge. Each cell has the part that intersects with the original shape cut off, so one of the edges of this cell is the closest edge that corresponds with the Voronoi cell. The first step is to find this edge in the cell, the *original edge* o , and then find the two edges next to it, the *protruding edges* p_1 and p_2 (Figure 15.c). In the standard closest point morph each vertex in the morphing chunk would be moved towards its closest point on the edge (Figure 15.d). The idea of the SCPM is that the points will be moved along the protruding edges instead (Figure 15.e).

We define the vectors P_1 and P_2 as the normalised direction vectors of the protruding edges (Figure 16). For each point in the Morphing Chunk we find the distance to both of the protruding edges, d_1 and d_2 . Now we find the direction vector $P = P_1 * (1 - \frac{d_1}{d_1+d_2}) + P_2 * \frac{d_1}{d_1+d_2}$. For any point directly on p_1 and p_2 this will be equal to P_1 and P_2 respectively. For any other point it will be something in between depending on how close to the lines it is. Now we find the intersection between the original edge and the line defined by the point of the morphing chunk and the direction vector P , this intersection point will be the point to morph towards. If the point from the morphing chunk is on the intersection of the protruding lines, then the point is split up and one of these points follows P_1 and the other follows P_2 .

The angle between these protruding edges and the original edge is always lower than or equal to 90 degrees. It can't be any higher because then part of the Voronoi cell would be closer to one of the points of the edge instead of the line segment, in which case it would not be a valid Voronoi cell. For a convex shape the edges will always be 90 degrees (such as Figure 17.a). In this case the SCPM and the CPM will give the same result as they both move the points towards the closest point on the line. In a Voronoi cell like the one in Figure 17.c, the vertices on P_1 would move the same in both the CPM and SCPM, but the ones closer to P_2 would be very different.

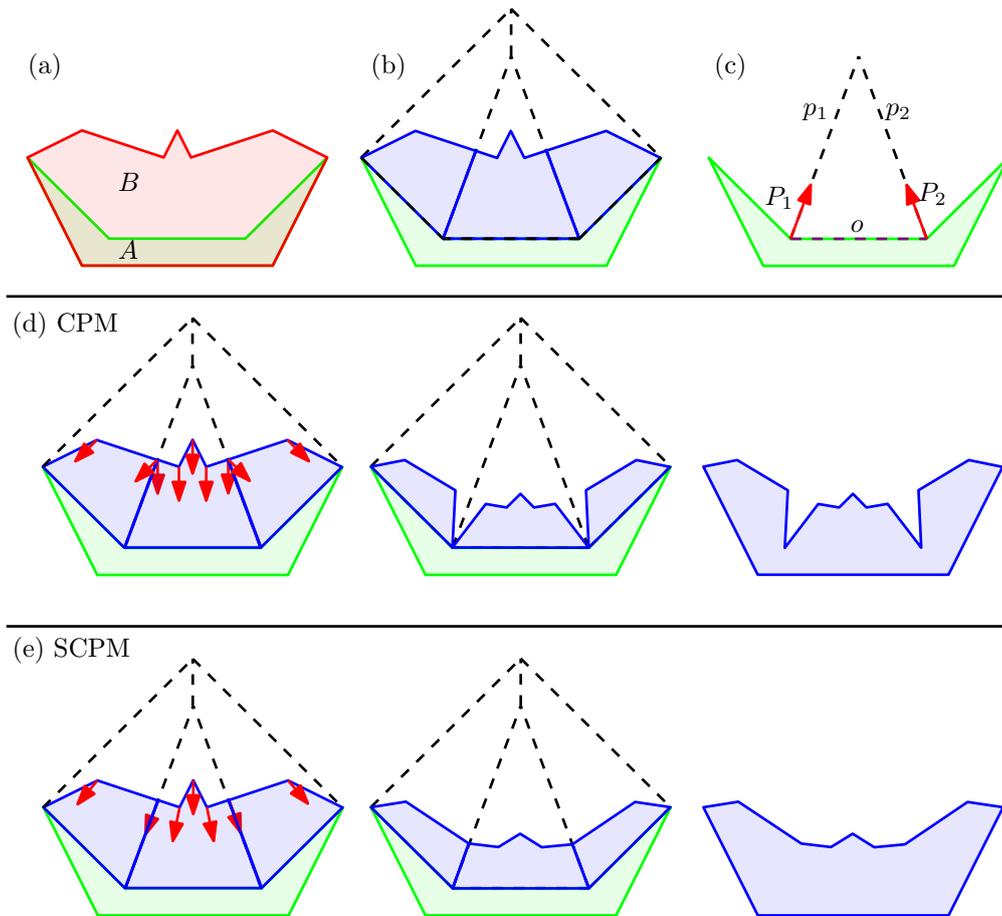


Figure 15: (a) shows the input shapes. (b) shows the untransformed morphing chunks. (c) shows the middle Voronoi cell with the corresponding original line o . It also shows the lines p_1 and p_2 protruding out of o and the corresponding vectors P_1 and P_2 , which will be used in the SCPM. (d) and (e) show the morphing process for the CPM and SCPM respectively. The first picture shows which direction the points will be moved in, the second shows the morphing chunks at $\alpha = 0.5$ and the third shows the full combined result at $\alpha = 0.5$.

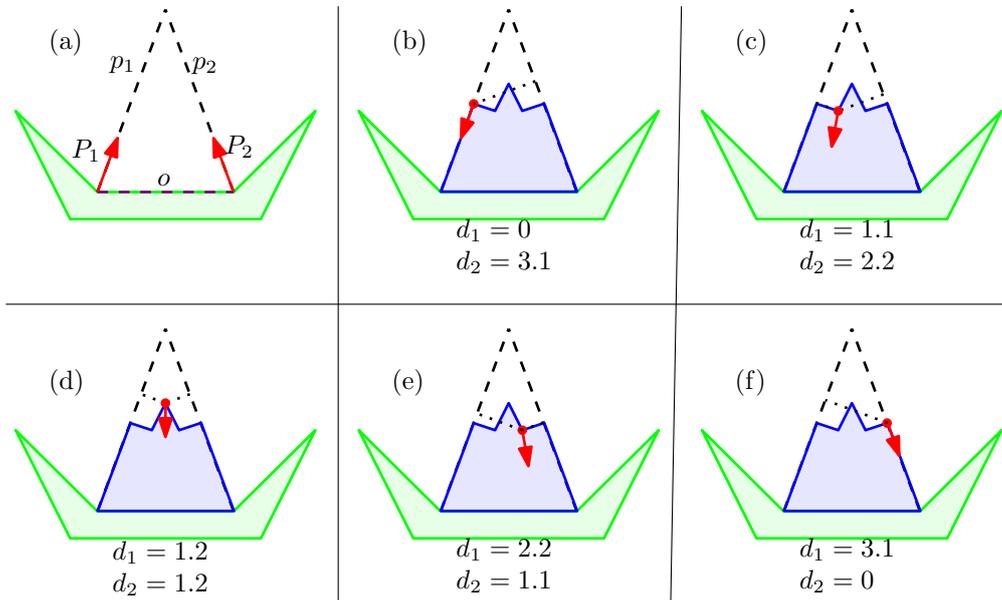


Figure 16: (a) shows a Voronoi cell with its original edge o and the protruding edges p_1 and p_2 . (b), (c), (d), (e) and (f) show which direction the different points will be moved in, with d_1 and d_2 , the distances of the point to p_1 and p_2 respectively. The direction vector is calculated as $P_1 * (1 - \frac{d_1}{d_1+d_2}) + P_2 * \frac{d_1}{d_1+d_2}$, the result is shown with the red arrows. The red point shows the initial position of the vertex, the green point shows where this vertex will be moved towards.

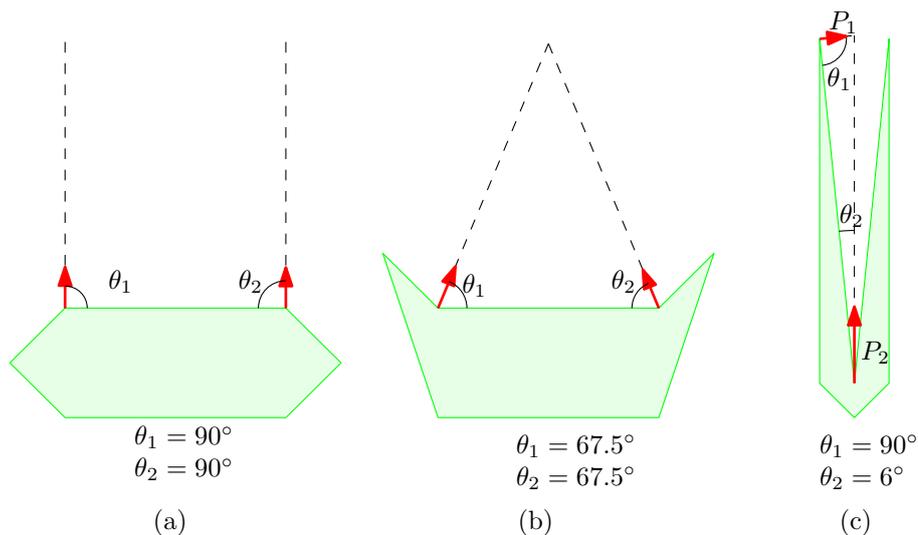


Figure 17: A couple of examples for the types of angles that are possible for the protruding edges used in the SCPM.

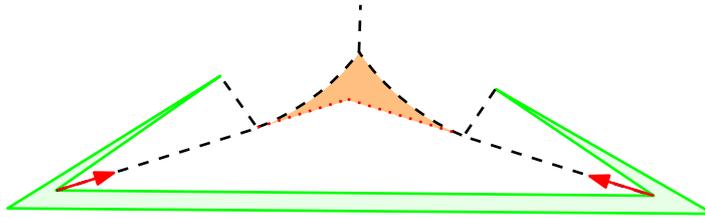


Figure 18: An example shape and part of its Voronoi diagram that can give unexpected results in the SCPM. The part that is colored orange is outside of the triangle created by the original edge and the protruding edges. If a morphing chunk has points in that area, then the directions those points will move into during the morph might give unexpected results.

6.1 Limitations of the Special Closest Point Morph

The implementation assumes that all points in the Voronoi cell are inside the triangle created by the original edge and the extended protruding lines. This is not always the case however, such as in Figure 18. This can create some strange unexpected results, including self-intersecting shapes. A better implementation of the SCPM would also work in these cases and move the points following the concave curve.

In practice however this seems to be a fairly rare problem, especially when shapes are fairly simple and normalised by area and centroid position.

6.2 Artifacts of the Special Closest Point Morph

The SCPM decreases the number of strange artifacts of the CPM, however some still exist. The SCPM does improve on situations such as Figure 11 and 12, the same morph in the SCPM can be seen in Figure 19.c. However the artifacts discussed in Section 5.4 are unaffected.

The SCPM can create holes that don't exist in either the CPM or the input shapes. This can happen when disconnected parts of both input shapes are morphed towards each other, such as in Figure 20

Figure 21 shows another limitation of the SCPM. The SCPM essentially works at the Voronoi cell level, which means the effectiveness is dependent on how long the Voronoi cells are, among other things. In the left example in the figure the SCPM can make a big difference, since the edges are long and there is only one vertex in the corner the big cut that exists in the CPM gets smoothed out. If you add a single small additional vertex to the corner, like on the right side of the image, then the SCPM will only smooth out the tiny cuts in the corner, but the large cut stays.

6.3 Hausdorff property of the Special Closest Point Morph

The Special Closest Point Morph is not a Hausdorff morph for all shapes as shown in the example of figure 22. As you can see, for this shape, $d_H(C_{\frac{1}{2}}, B) = d_H(A, B)$ instead of $d_H(C_{\frac{1}{2}}, B) = d_H(A, B) * \frac{1}{2}$, which shows the SCPM is not a Hausdorff Morph.

6.4 Area of the Special Closest Point Morph

The area of the Special Closest Point Morph is the same for the morphing chunks that morph to a point. But for morphing chunks that morph to an edge in a concave part of the other shape the area is higher than a linear interpolation. How much higher depends on how concave and how far away the shape is that the chunk is morphed towards. In practice this increase in area is usually fairly small, but in theory the area is unbounded.

Theorem 6.1. *The area of the Special Closest Point Morph can be arbitrarily high, given an input shape with an arbitrarily high perimeter and distance from the other input shape.*

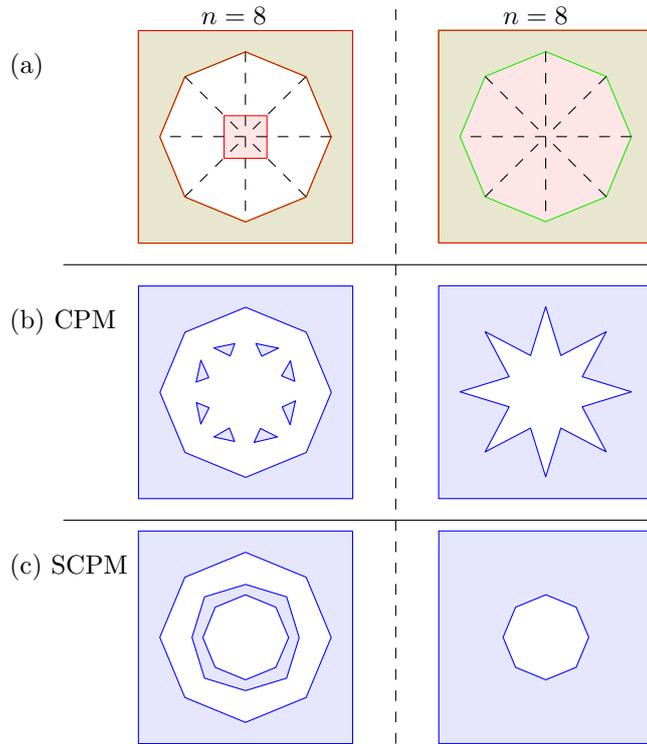


Figure 19: This shows how the SCPM handles the morphs between Q_n and P_n from Figure 12 (left) and the morph between Q_n and B from Figure 11 (right). The CPM result is also shown for comparison. All morphs are shown with $\alpha = 0.5$.

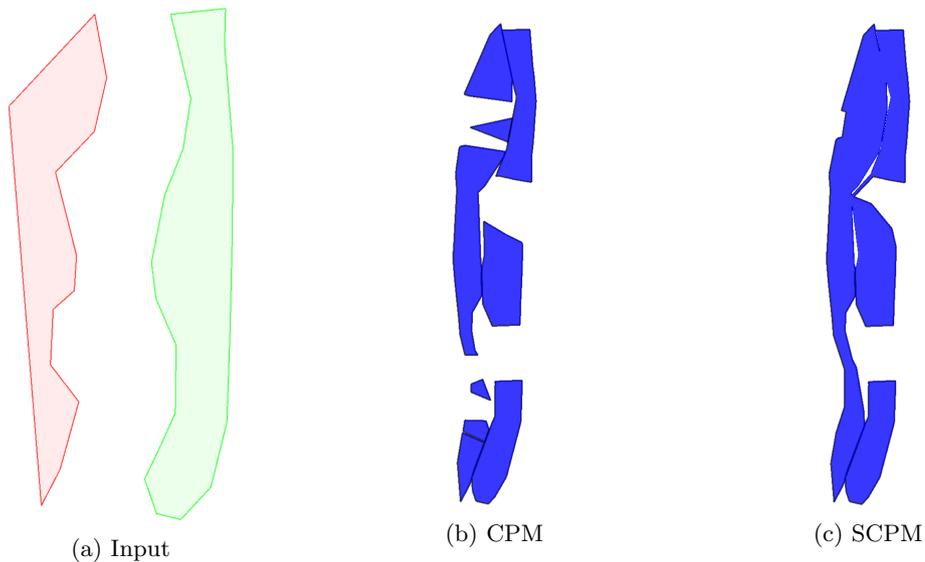


Figure 20: An example that shows how the CPM and SCPM handle two somewhat close to parallel sets of edges morphing towards each other. The morphs are shown at $\alpha = 0.5$. As you can see the SCPM gets some holes that the CPM doesn't have, and in both morphs the morphing chunks don't connect to the morphing chunks from the other shape properly.

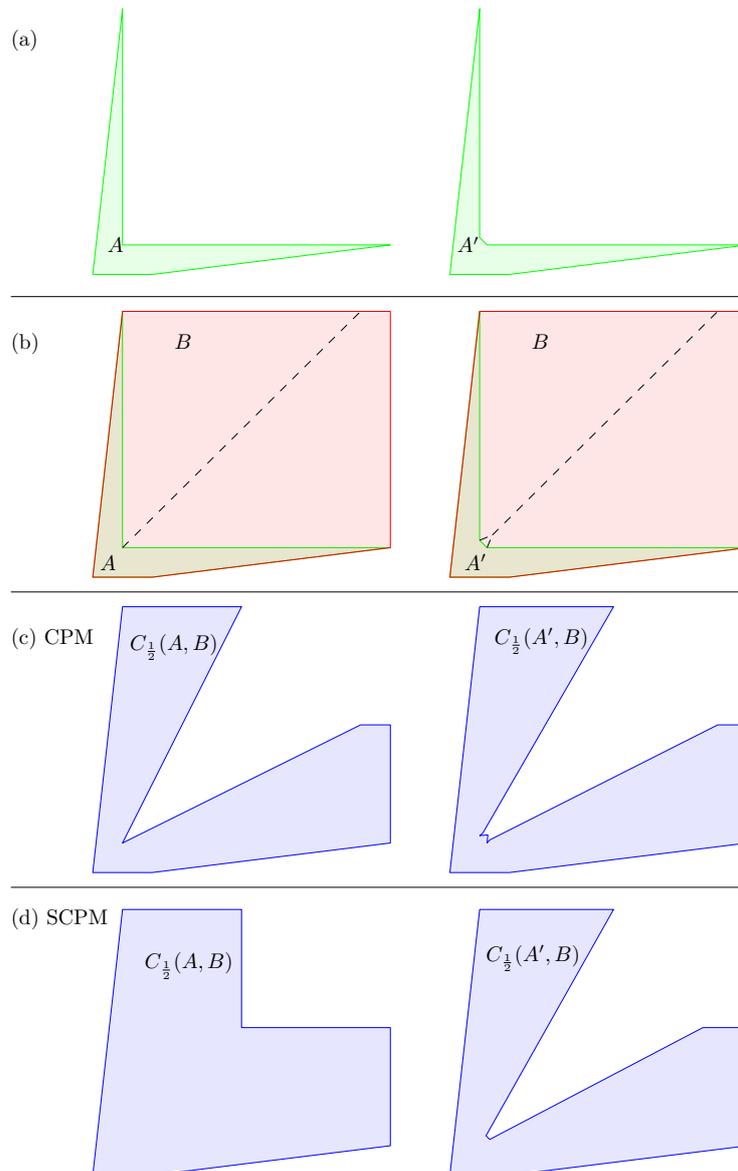


Figure 21: (a) shows the source shapes A and A' , A' is the same as A , except it has an additional vertex in the inner corner. (b) shows input shapes A , A' and B . (c) shows how these morphs look in the CPM. (d) shows how SCPM makes a big improvement when there is only one vertex in the corner, but doesn't create a much better result when there are two vertices.

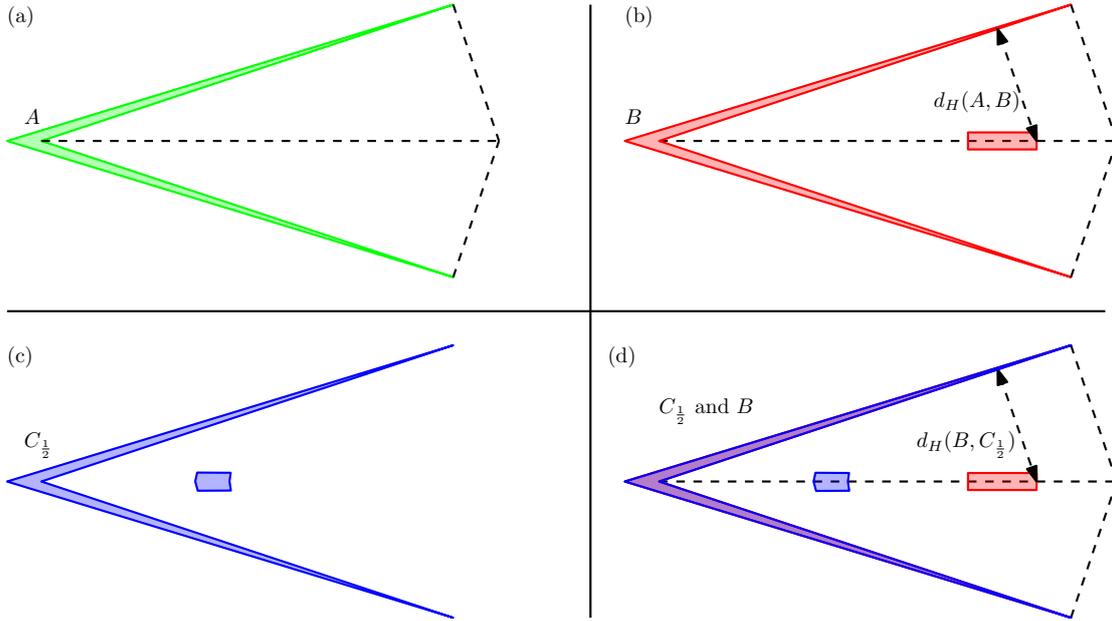


Figure 22: (a) shows input shape A . (b) shows input shape B which is the same as A except it has an added block in the middle that is very close to the middle protruding line in the voronoi diagram of A . It also shows the Hausdorff distance between B and A . (c) shows the result of $C_{\frac{1}{2}}$ in the SCPM. (d) shows $C_{\frac{1}{2}}$ and B , it also shows the Hausdorff distance between $C_{\frac{1}{2}}$ and B .

Proof. Consider a situation such as in figure 23.a with a shape A with two edges on the concave part of the boundary and a shape B entirely encapsulated within the cells of these two edges. Another shape A' can be created which is the same as A but moved further away, with the points of the edges on the concave part of the boundary following the direction of the protruding lines. A' is also made thinner to keep the area consistent. The directions of the protruding lines are still the same, but now the morphing chunks coming from B will be stretched out much further. As you can see in Figure 23.b and 23.c distances between the points traveling on the same line stays the same but the distance between the lines becomes much bigger when using A' and so does the area.

A new shape A'' can be created that is transformed in the same way as A' . The further away A'' is the larger the area of $C_{\frac{1}{2}}^B$, while the area of the input shapes is kept constant by making A'' thinner. Since A'' can be placed arbitrarily far away, $C_{\frac{1}{2}}^B$ can get an arbitrarily large area. ■

7 Experiments

7.1 Implementation details

The collected data comes from a polygon based implementation created in C++ using Boost [4]. Boost/Polygon is used to set up the Voronoi diagram and Boost/Geometry is used for polygon intersections.

Voronoi diagrams can have curves, in our implementation these are approximated with line segments, this can create a very small bias but it shouldn't have a big effect on the results. Additionally, in the final step where all of the individual morphing chunks are combined, each chunk is dilated very slightly to ensure neighbouring chunks are combined properly. This very slightly increases the area and in some very specific situations where there should be a tiny distance between two chunks, they might be incorrectly combined. This doesn't matter much for the area or visual result, but it can have a small impact on the perimeter and number of pieces.

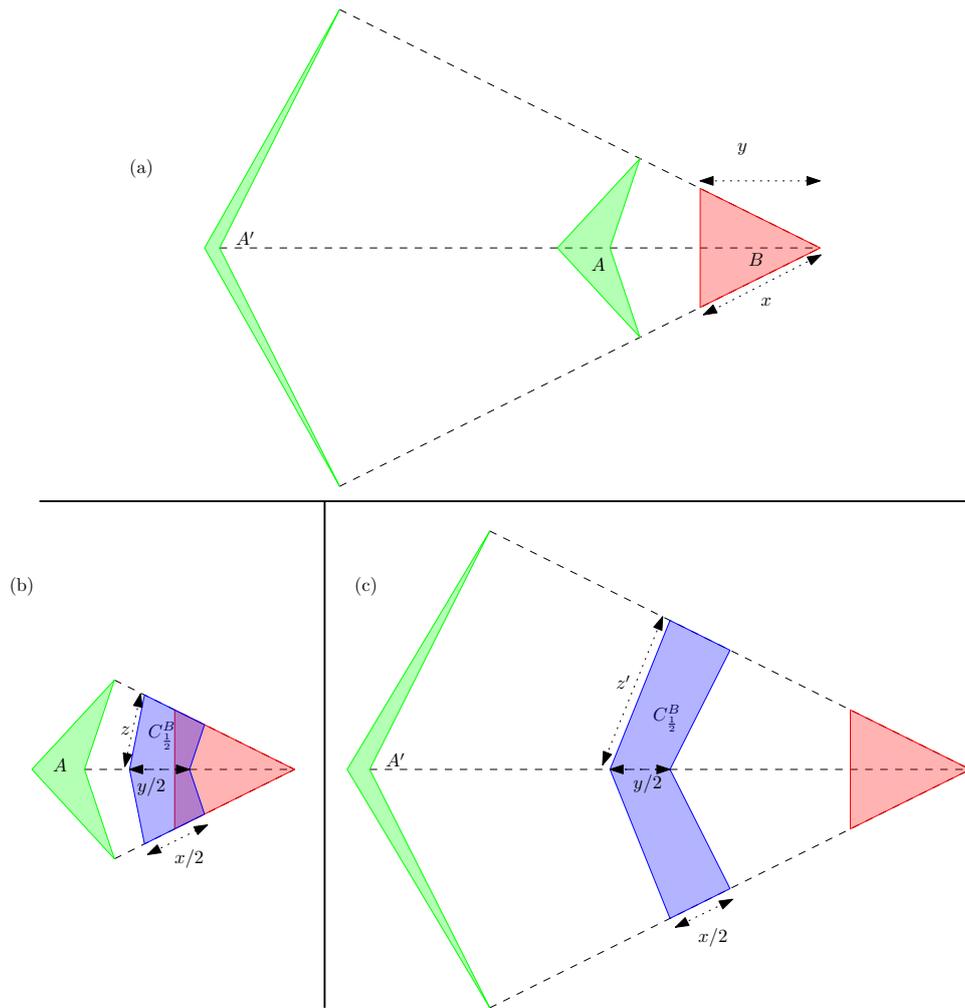


Figure 23: (a) shows the input shapes A , A' and B . (b) shows the morphing chunks coming from B when morphing to A in the SCPM with $\alpha = \frac{1}{2}$. (c) shows the morphing chunks coming from B when morphing to A' in the SCPM with $\alpha = \frac{1}{2}$. Note that the morphing chunks coming from A and A' to B are omitted.

7.2 Datasets

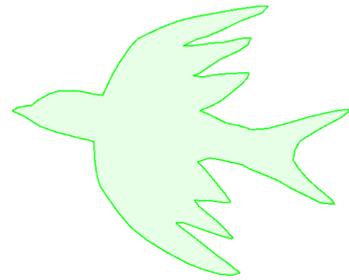
Tests were performed on two datasets, one containing 6 European countries from the Thematic Mapping World Borders Dataset[10] and another containing 9 simple drawings of animals, which was taken from previous work by Bouts et al. [5]. The animal dataset has a relatively low polygon count and a quite a lot of the vertices follow smooth curves. This could mean that the SCPM will do quite well as it can smooth out some of the cuts created by the curves. For comparison, the country dataset is much more complex, both because it has a much higher vertex count, and because the vertices don't tend to follow smooth curves.

The area has been normalised within each dataset, so the area for all Animals is the same and the area for all countries is the same. In the code this is quite a large area, 13911500 and 2869830 for the animal and country datasets respectively, however in the results this will be normalised to 1 to make the numbers easier to read. Keep in mind that the perimeter is not normalised in any way, since the animals and countries are calculated at different scales that means we cannot make a direct comparison between the perimeters. While the numbers are smaller, in general the country dataset has a much larger perimeter for its scale than the animal dataset. In the following subsections the input shapes will be shown, the images are not to scale and the perimeter is based on the perimeter after normalizing the area.

7.2.1 Animal dataset

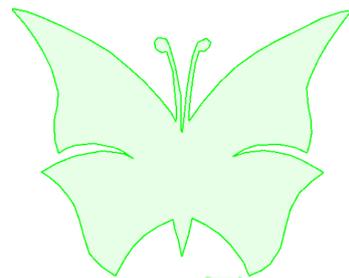
Bird

Number of vertices	138
Perimeter	34890.1
Pieces	1
Expected result	The bird has a fairly distinctive shape with its large wings, the sharp points of the feathers will probably be visible in a lot of the combinations. There are some concave curves that will probably do quite well in the SCPM, but they will likely create a lot of cuts in the CPM.



Butterfly

Number of vertices	131
Perimeter	34545.8
Pieces	1
Expected result	The butterfly is fairly compact, which likely means it will overlap with a lot of shapes. The large convex curve will probably do quite well in the CPM, the antenna's, and the cuts in the middle of both wings will likely be recognisable in a lot of the combinations.



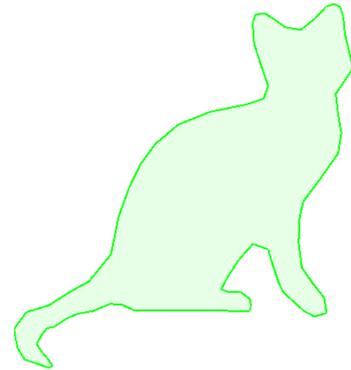
Cat

Number of vertices 76

Perimeter 24709.6

Pieces 1

Expected result The cat shape might be a bit harder to recognise in a morph since the features aren't as distinctive and might blend into features of the other shape. The large convex curve on the back and the low vertex count could help the perimeter stay fairly low in many combinations.

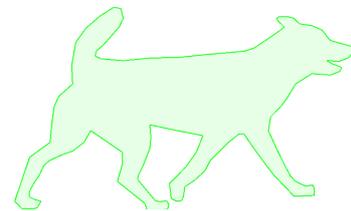
**Dog**

Number of vertices 125

Perimeter 34750.8

Pieces 1

Expected result The dog is fairly long, so it might not overlap as much with shapes that aren't also long. The head and tail will likely not overlap much in many combinations, which might keep them recognisable. The long legs will likely cause some awkward results like in Figure 20.

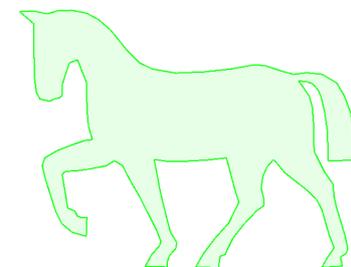
**Horse**

Number of vertices 148

Perimeter 39127.6

Pieces 1

Expected result The horse is also somewhat long, the legs are long and will cause some unnatural looking splits and floating pieces.

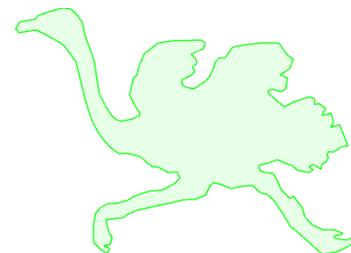
**Ostrich**

Number of vertices 191

Perimeter 37692.2

Pieces 1

Expected result The head and neck of the ostrich are long and thin, it likely won't overlap much with the other shapes, and could stay relatively intact throughout the morph.



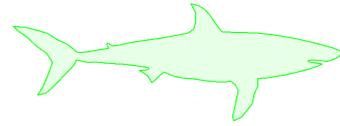
Shark

Number of vertices 100

Perimeter 32974.3

Pieces 1

Expected result The shark is the longest shape and will likely overlap the least out of all shapes. It does have fairly long convex curves, which could mean the CPM will not split very much.

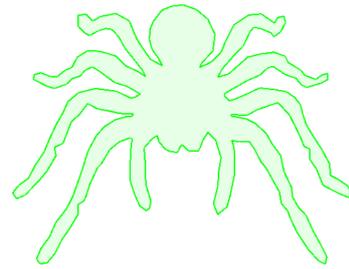
**Spider**

Number of vertices 262

Perimeter 65962.9

Pieces 1

Expected result The spider has by far the highest perimeter and it is the most complex shape in the animal dataset. The combinations containing the spider will likely split a lot too and have by far the highest perimeter out of all combinations.

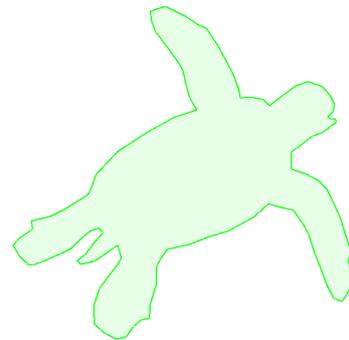
**Turtle**

Number of vertices 117

Perimeter 28279.6

Pieces 1

Expected result The turtle has some fairly distinctive features such as the long arms and little tail which could help to keep it recognisable in the morph. The shell gives quite a lot of area in the middle which could make for a good amount of overlap.



7.2.2 Country dataset

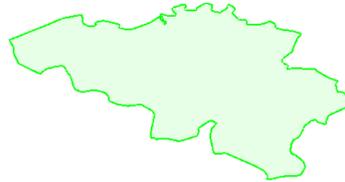
Germany

Number of vertices 2080
Perimeter 15199.7
Pieces 16
Expected result Relative to some other countries, Germany doesn't have a lot of islands, which could help keep the morph together a bit. There is a lot of detail in the border and there aren't many long curves, which likely means the SCPM will not do much better than the CPM.



Belgium

Number of vertices 381
Perimeter 10503.8
Pieces 1
Expected result Belgium is by far the simplest shape in this dataset, it doesn't have any islands either. This likely means that it will cause relatively few splits in the CPM.



France

Number of vertices 1990
Perimeter 14387.6
Pieces 8
Expected result There are a couple of small islands on the west coast, which will likely cause some trouble. Any part of the other shape close to the islands will morph to the small island instead of the large mainland, which could look unnatural.



Ireland

Number of vertices 1016
Perimeter 19745.1
Pieces 4
Expected result There are already quite a lot of 'cuts' on the west coast, the CPM will add some more. Perhaps the new cuts created by the morph could blend in with the real cuts.



Italy

Number of vertices	2186
Perimeter	21104.2
Pieces	15
Expected result	Italy is the most complex shape, it will likely not produce great results. It has a lot of small islands near the west coast, which could cause problems as stated before. It is also quite long and likely will not overlap much.



Spain

Number of vertices	1634
Perimeter	14357.1
Pieces	14
Expected result	Spain is an interesting case because it has some small islands very far away. This is likely so far away that there will never be a morphing chunk that is morphed towards it, so in the CPM we would likely only see it as a couple of islands being morphed towards the rest of the morph. The DM will be very large because those small islands cause a large increase in the Hausdorff distance.



7.3 Main experiments

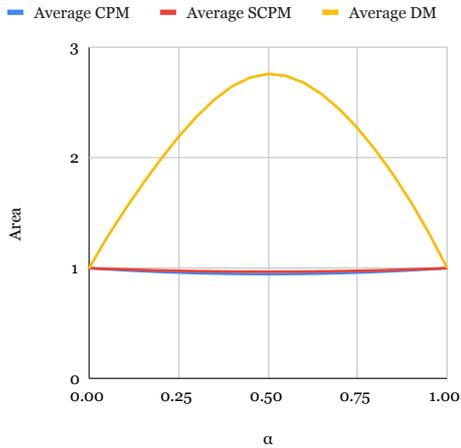
The following main experiments were done:

- For all combinations of both datasets the effect of changing α was tested with the centroid position and area normalised.
- For all combinations of animal shapes the effect of changing their relative position was tested with $\alpha = 0.5$. The centroids start on top of each other, then the target shape is moved to the right in equal steps until the shapes are far apart.

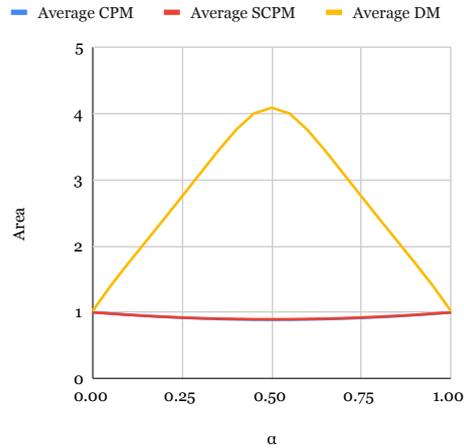
In these experiments the *Dilation Morph*, the *Closest Point Morph* and the *Special Closest Point Morph* were all tested and compared against each other. The main metrics which were collected are the *area*, *perimeter* and *number of connected components*.

The first experiment was chosen because it is the simplest most intuitive way to test whether the morph looks good, and the area, perimeter and number of connected components are useful simple metrics for the quality of a morph. The second experiment was chosen to compare the quality of the morphs at different distances, and to see if the CPM will quickly converge to a certain shape when it is moved apart, or if it will keep changing.

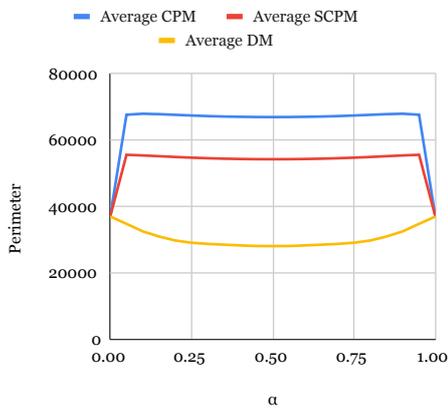
The area of the dilation morph likely depends mostly on the Hausdorff distance between the input shapes, since the amount of dilation depends on the Hausdorff distance.



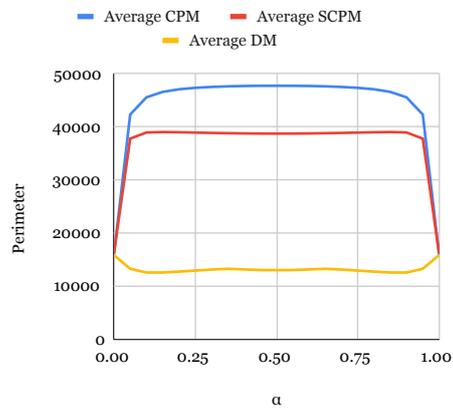
(a) Area Animal dataset



(b) Area Country dataset



(c) Perimeter Animal dataset



(d) Perimeter Country dataset

Figure 24: The effect of changing the alpha value on the area and perimeter, as tested on all combinations of shapes from the animal dataset and all combinations of the country dataset.

7.4 Crescent experiment

We also did an experiment similar to the one described in section 5.2. In this case with a circle and a crescent as input shapes instead of a square with a hole. The concave part of the crescent is inside the circle. This experiment was made to show the effect that the number of vertices can have on the perimeter in a worst case scenario, and how this differs between the Special Closest Point Morph and the Closest Point Morph. The Dilation Morph is left out for this experiment since it is not relevant for the purpose of the experiment and it does not give interesting results.

8 Results

In this section we will discuss the results of the experiments. Not every single combination will be discussed, first some general observations will be made and then we will look at several interesting combinations in more detail.

8.1 Changing α with normalised area and position

In this section we will show the effect of changing α for all combinations of the animal dataset and all combinations of the countries dataset. In general the animal dataset gave more appealing results, a comparison of the average area and perimeter can be found in Figure 24.

8.1.1 Closest Point Morph General Comments

As for the standard CPM, the relation between α and the perimeter seems to show an interesting pattern, the highest perimeter is almost always found with $\alpha = 0.05$ or $\alpha = 0.95$, the highest and lowest tested values. The few combinations where this is not the case could be due to precision issues when α gets too close to 0 or 1, where parts of the morph get combined when they theoretically should stay separated. It seems plausible that, for a theoretical perfect implementation, the highest perimeter would always be found at the highest or lowest α with $0 < \alpha < 1$. This however remains unproven.

In all combinations in both datasets there wasn't a single combination that stayed connected in the morph. The lowest number of connected components is 4, in the cat butterfly combination.

The relation between α and the area is parabolic, as predicted in section 5.1. For the animal dataset, the area at $\alpha = 0.5$, the lowest point, is between 0.8867 and 0.9768 for each combination, with an average area of 0.9467. The shape with the lowest average area is the shark, this is likely because it is the longest shape and it does not overlap as much as the other shapes. The shape with the highest average area is the butterfly, likely because it is a fairly compact shape that overlaps with the other shapes a lot.

For the country dataset the area at $\alpha = 0.5$ is between 0.7555 and 0.9537, with an average area of 0.8890. These areas are significantly lower than with the animal dataset, which can mostly be attributed to the shape of Italy. The average lowest area for all combinations including Italy is 0.7988 and the average lowest area for all combinations without Italy is 0.9318. Italy is very long and has some larger disconnected islands, this means most combinations do not have much overlap. Some of these combinations will be looked at more thoroughly.

8.1.2 Dilation Morph General Comments

The Dilation Morph is of course always much bigger than the Closest Point Morph. As for the animal shapes, the shark gave the lowest average area in the CPM, but it gives the highest area in the dilation morph. The highest recorded area is 4.224 in the morph between the dog and shark at $\alpha = 0.45$. The most important factor that determines how much the shape will grow in size is likely the initial Hausdorff distance, as this dictates how big the dilation is. The morph tends to look fine when the area stays at least below 2, albeit still quite big and low in detail. Above that the shapes lose almost all recognisable features. Although the morph does look better in motion than it does in static images, as the morph does of course smoothly changes between the shapes.

Despite the large increase in area, the perimeter usually gets lower, this is because a lot of detail is smoothed out to make the morph.

It's possible for the Dilation Morph to break up into multiple pieces [13], in practice the number of pieces for the dilation morph was equal to or lower than the number of connected components in source shapes for all combinations, except for the one with the bird and horse (Section 8.1.9).

All combinations containing Spain grow very large, this is because Spain has some small islands very far away which increases the Hausdorff distance, and therefore also the amount of dilation, by a lot.

8.1.3 Special Closest Point Morph General Comments

The SCPM makes some noticeable improvements over the CPM in the animal dataset. Every combination looks a bit more natural in the SCPM and the area and perimeter are more reasonable. The difference is much smaller for the country dataset, both in numbers and in subjective naturalness. In many of the combinations the difference is much harder to see. The SCPM works

best when the vertex count is low or when the vertices follow a long smooth curve, both are rarely the case for the country dataset. Compare for example the back of the horse and the coast line of Italy. This results in an average difference in area between the SCPM and CPM of 0.02140 for the animal dataset but only 0.01116 for the country dataset.

Additionally the problem where the SCPM doesn't always give valid results, which was explained in Section 6.1, occurs more with the countries than it does with the animals. This makes the results less reliable for that dataset.

8.1.4 Bird Butterfly combination

The results of this combination can be found in Figure 25. This is a fairly average case for the area in the CPM and SCPM. The Dilation Morphs area is fairly low relative to the other combinations, and it maintains the rough shapes of the bird and the butterfly fairly well. This is mostly because the Hausdorff distance of the input is pretty low and the shapes are easy to read.

The CPM and SCPM also preserve features of the original shape pretty well, but they do split up into quite a few pieces. This is largely because of the way the butterfly's wings are morphed towards the birds pointy feathers. The SCPM does a good job at smoothing out the small cuts in the CPM, since these cuts are all fairly small it doesn't end up making a big difference in the area.

8.1.5 Cat Ostrich combination

The results of this combination can be found in Figure 26. This is a fairly average case for the area and perimeter.

The CPM and SCPM preserve the original shapes fairly well, both heads are intact which makes it pretty easy to recognise. The way the heads morph is interesting, the cat head is slightly wrapped around one of the wings of the ostrich, this combined with the fact that it has a fairly simple shape allows the cat head to nicely blend. At $\alpha = 0.75$ (Figure 26.f) the head is already quite nicely blended in with the wing and the SCPM looks very similar to the ostrich input. The ostrich head doesn't blend as nicely with the cat however. At $\alpha = 0.25$ it still looks quite a lot like an ostrich head sticking out of the cat, this is because it is a slightly more complicated shape and it is further away.

8.1.6 Cat Turtle combination

The results of this combination can be found in Figure 27. This combination has the lowest area for the DM, mostly because the initial Hausdorff distance isn't very large. As expected the DM still grows quite a lot and loses quite a few details, however even at an area of 1.7859 the results still look quite natural, especially in motion. While the shapes are a bit abstract, the cat and turtle shapes can still be vaguely recognised at $\alpha = 0.5$. Because the Hausdorff distance is a bit lower less detail is lost, in this case the lost detail and gained area add up to a perimeter that is quite close to a linear interpolation between the two shapes.

The SCPM and CPM don't look as natural here, the turtle shape is still fairly recognisable, but the cat shape is not. The head of the cat is split up in a way that makes it hard to recognise. The SCPM does not fix this issue because of the vertices between the head and arm of the turtle, this problem is the same as the one shown in Figure 21. The cats tail also isn't recognisable as it is split unnaturally between the hind legs and tail of the turtle.

8.1.7 Cat Dog Combination

The results of this combination can be found in Figure 28. The Dilation Morph loses pretty much all features and only very rough shapes are maintained.

The CPM has a lot of small cuts and 13 disconnected components, but it maintains some features of the original shapes. The heads of both animals are partially morphed towards each other and partially towards the other animals body, this creates strange looking floating pieces

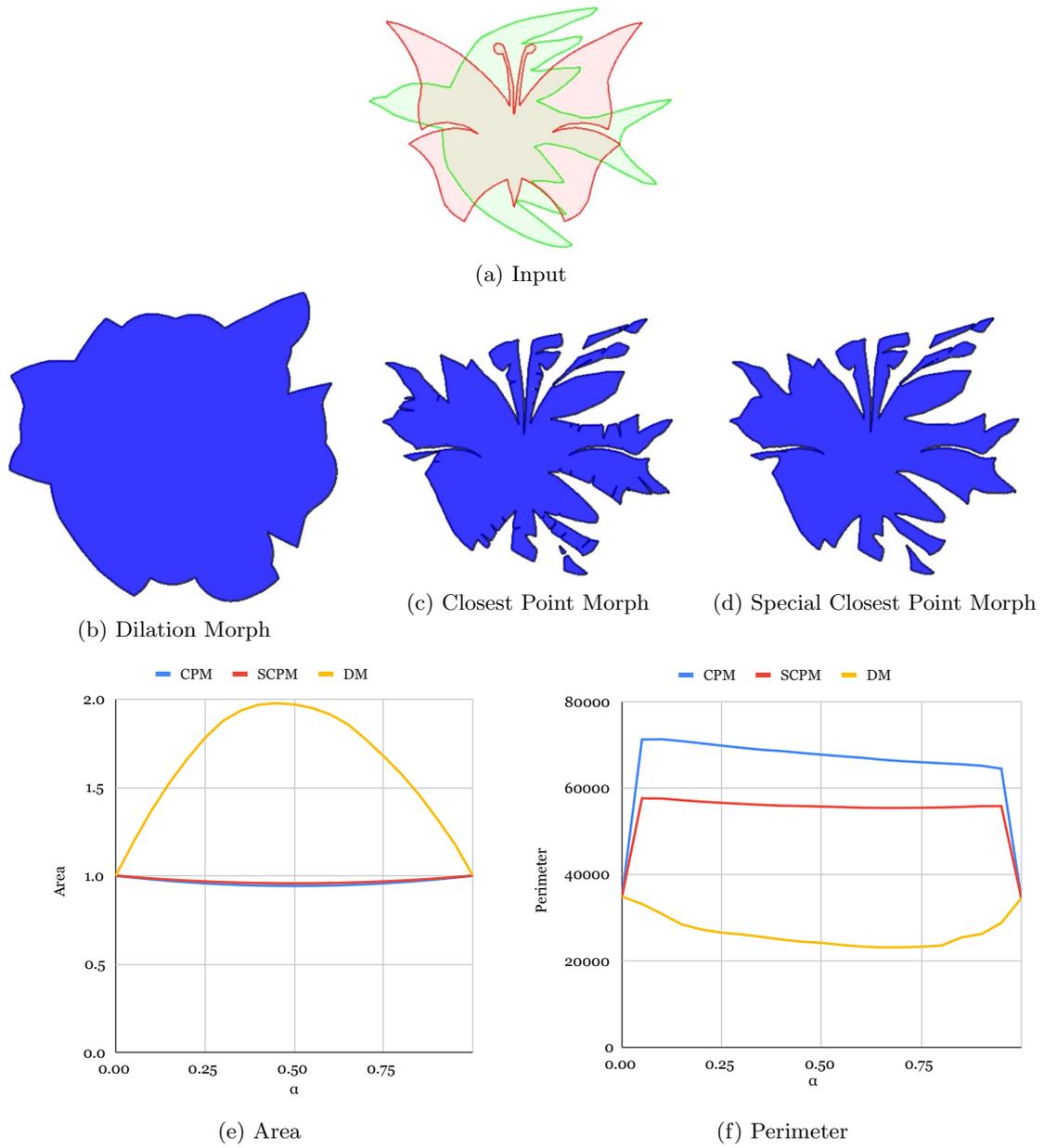


Figure 25: The results of the morph of the bird to the butterfly with normalised scale and centroid position. The images for each of the morphs was made with $\alpha = 0.5$. The lowest area is 0.9422 for the CPM and 0.9569 for the SCPM and the highest area for the DM is 1.97054.

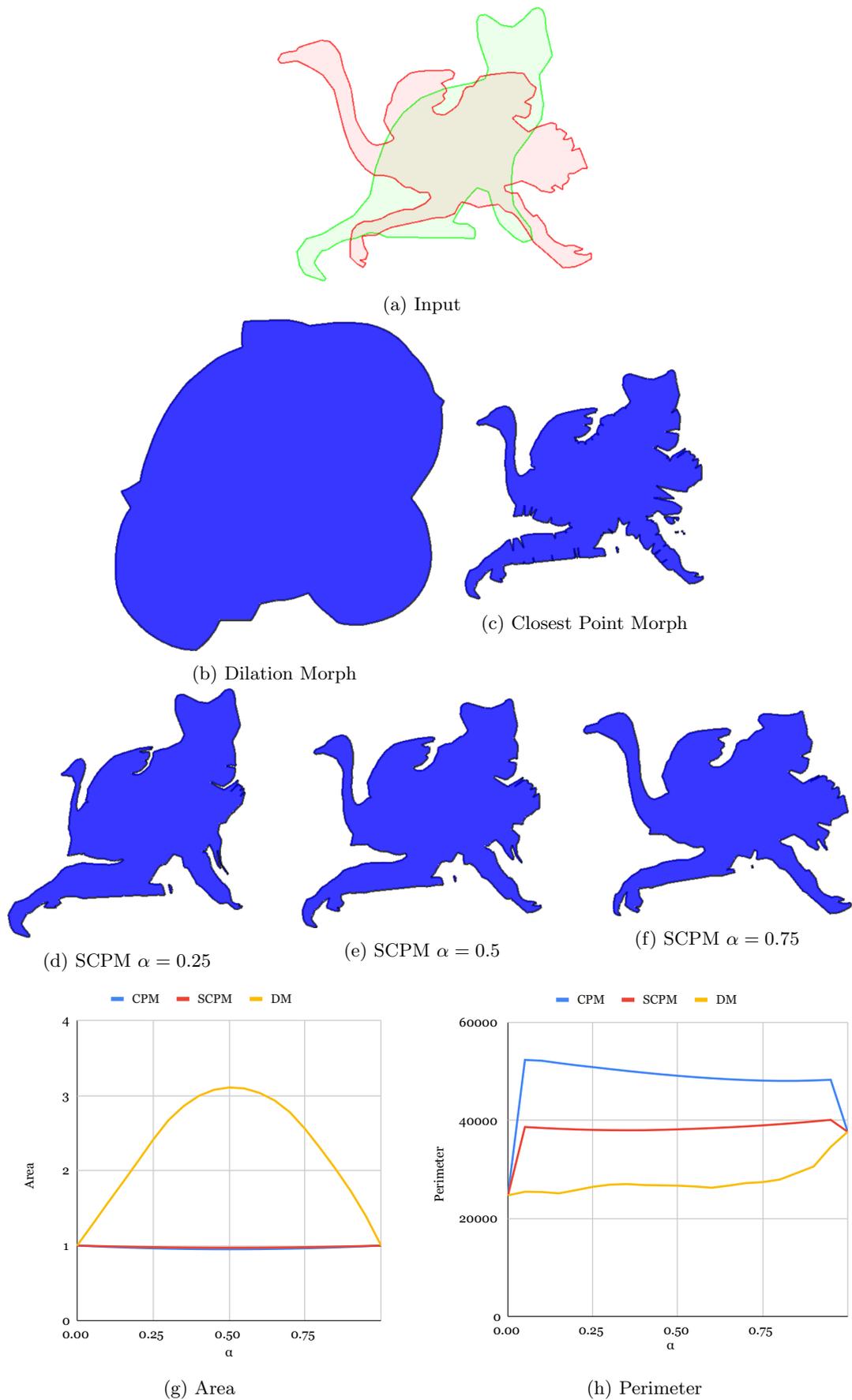


Figure 26: The results of the morph of the cat to the ostrich with normalised scale and centroid position. The images for the DM and CPM were made with $\alpha = 0.5$. The lowest area is 0.9508 for the CPM and 0.9709 for the SCPM.

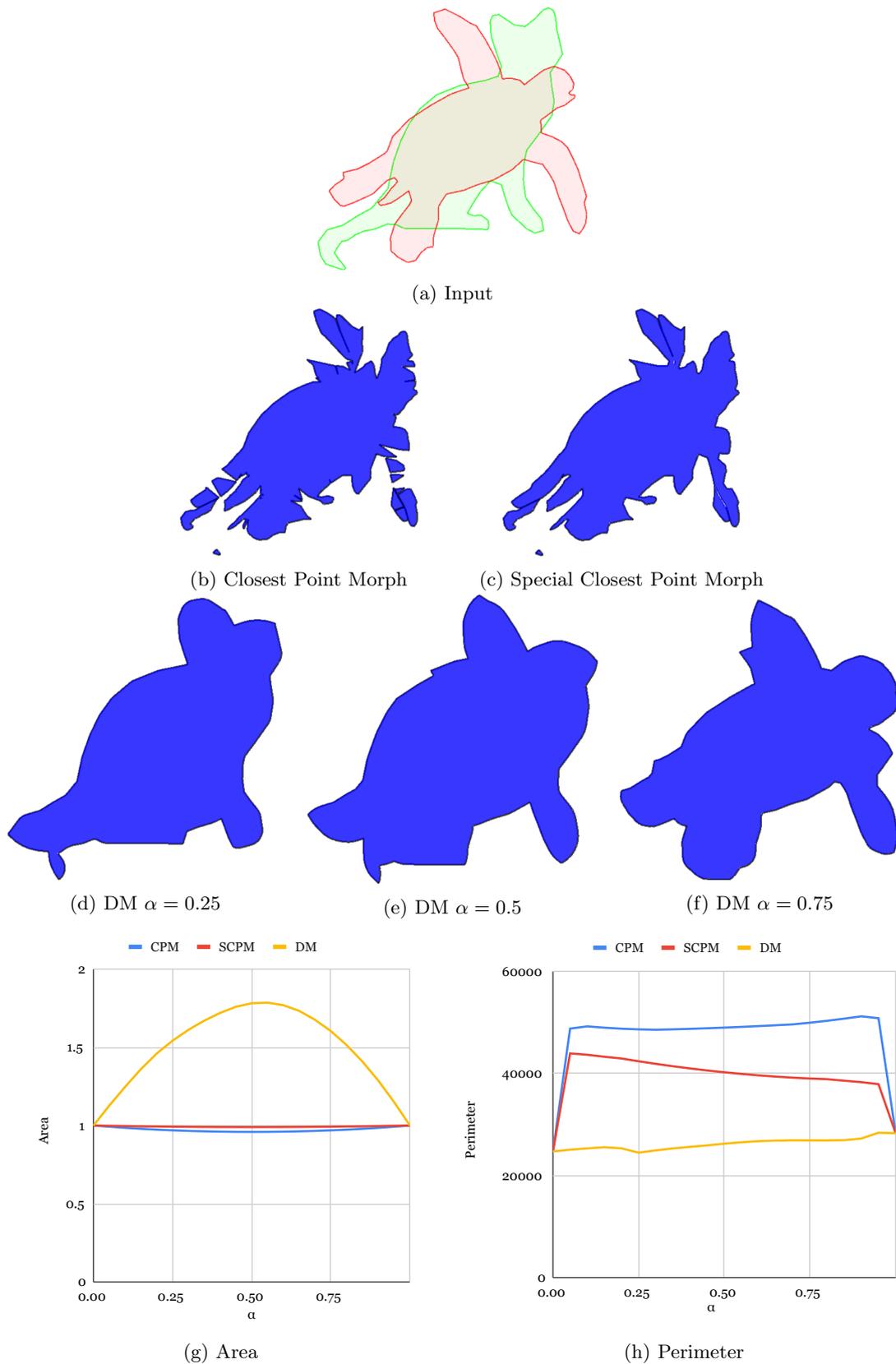


Figure 27: The results of the morph of the cat to the turtle with normalised scale and centroid position. The images for the SCPM and CPM were made with $\alpha = 0.5$. The lowest area is 0.9593 for the CPM and 0.9903 for the SCPM. The highest area for the DM is 1.7859.

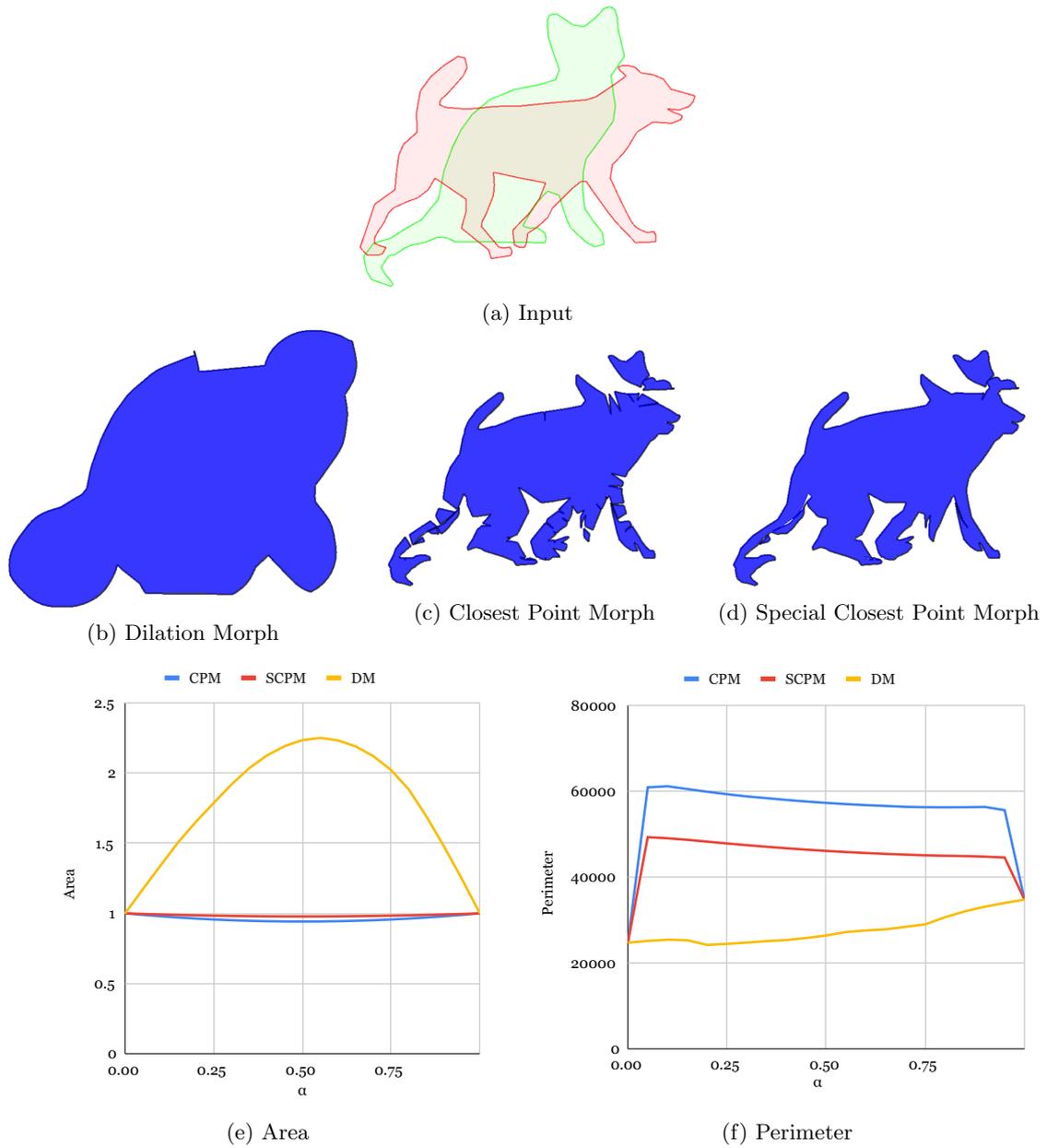


Figure 28: The results of the morph of the cat to the dog with normalised scale and centroid position. The images for each of the morphs was made with $\alpha = 0.5$. The lowest area is 0.9420 for the CPM and 0.9782 for the SCPM and the highest area for the DM is 2.2509.

of head. This combination has the biggest difference in area between the CPM and the SCPM out of all of the combinations in the animal dataset. By looking at the input you can see that for large parts of the both shapes the closest point in the other shape is on concave sets of edges. The SCPM does a good job of smoothing out the morph, and keeping it together. The heads are still cut in halve like in the CPM, but now it is held together a bit better, although the cut off cat head is still only connected in a single point to the rest of the morph. Additionally a long thin hole is created between the tail of the cat and the hind leg of the dog, which looks a bit odd. The SCPM still has some problems here but it is a big improvement over the CPM.

8.1.8 Horse Spider combination

The results for this combination can be found in Figure 29. The DM retains a rough shape of the horse, but is not very recognisable.

The horse and spider are both quite recognisable in the CPM and SCPM. The head of the spider deforms somewhat strangely because of the curve in the horses neck. Due to the long thin legs of both animals there are quite a lot of floating pieces. These legs also create some long, thin holes in the SCPM, as discussed in Section 6.2. The morph would look more natural if these holes were simply filled up. However that doesn't entirely fix the issue since a lot of these long thin cuts are not true holes, since they are not fully surrounded. In Figure 29 the holes X, Y and Z could easily be removed this way, but W is not fully surrounded so it would still look strange.

8.1.9 Bird Horse combination

The results for this combination can be found in Figure 30. This combination is special for two reasons, firstly it is the combination with the biggest perimeter difference between the CPM and SCPM, and secondly it is the only combination in the animal dataset where the Dilation Morph splits up into more than one connected component.

The bird horse combination has two separate components for $0.55 \leq \alpha \leq 0.70$, It is very rare for a shape to split up in the Dilation Morph because of the amount of dilation, but in this case the combination of the tip of the horses back hoof and the lowest part of the birds wing create a barely disconnected component (Figure 30.b).

The difference in area and perimeter between the SCPM and CPM is quite high. This is because large areas of both shapes have a concave part of the other shape, such as the large upper wing of the bird which is moved to the large concave back of the horse.

The head and front leg of the horse are split somewhat unnaturally, both in the CPM and SCPM. It might have looked more natural if this split had been smoothed out, similarly to how the SCPM smooths out the smaller cuts. This is the type of split that the SCPM does not do anything about, because the vertex count is too high in that area. If there would have been fewer vertices on that part of the wing the morph might have looked more natural.

8.2 Germany Italy combination

The results for this combination can be found in Figure 31. This is the combination with the lowest recorded area in the CPM and SCPM, this is mostly because of how little overlap there is between the shapes. The difference between the SCPM and CPM is also very small both visually and in terms of area. The perimeter is still quite a bit lower since the SCPM does slightly smooth out the morph at the vertex level. The morph also looks fairly strange because it splits up so much. It would already split quite a lot if it were only mainland Italy, but the small islands at the west coast make this quite a lot worse. This is also reflected in the perimeter that got very large.

While this is in no way specific to this combination, Figure 31.b is also a good example to show how the perimeter shoots up if α is even slightly lower than 1. The cracks appear immediately, there is no smooth transition from the relatively low perimeter of the source shape to the high perimeter of the morph.

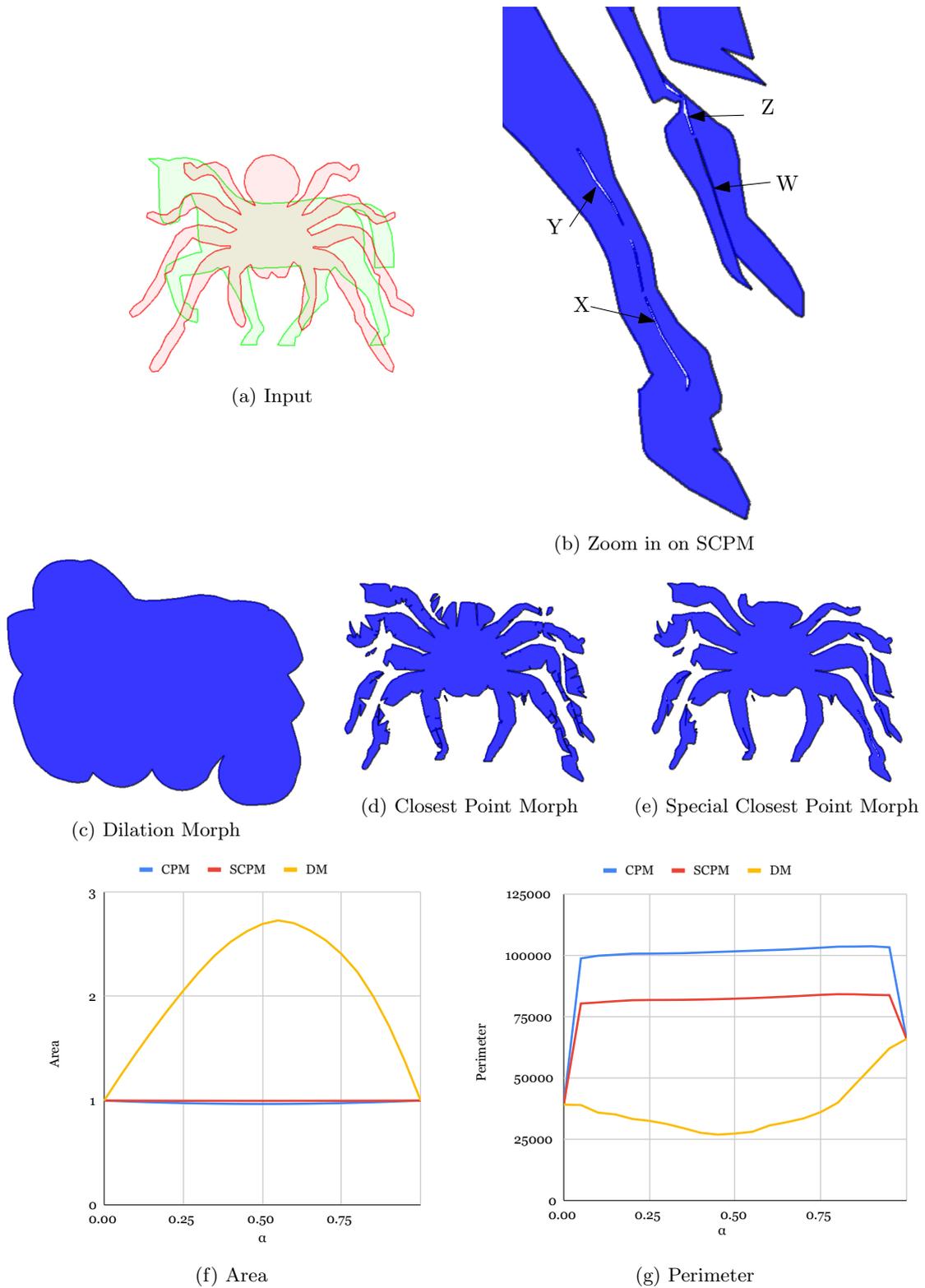


Figure 29: The results of the morph of the horse to the spider with normalised scale and centroid position. The images for each of the morphs was made with $\alpha = 0.5$. The lowest area is 0.9681 for the CPM and 0.9973 for the SCPM and the highest area for the DM is 2.7274.

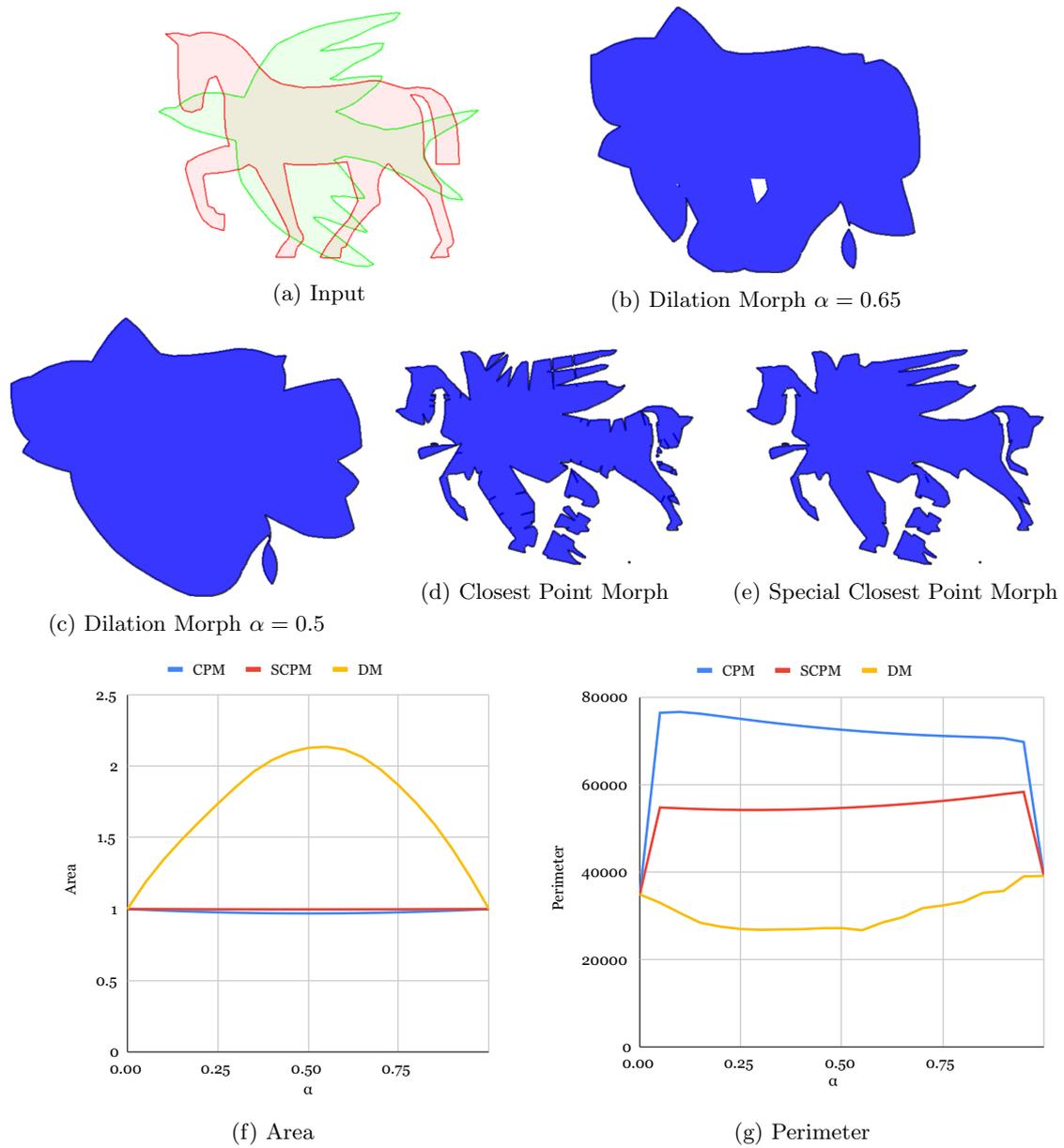


Figure 30: The results of the morph of the cat to the dog with normalised scale and centroid position. The images for each of the morphs was made with $\alpha = 0.5$. The lowest area is 0.9697 for the CPM and 0.9978 for the SCPM and the highest area for the DM is 2.1361.

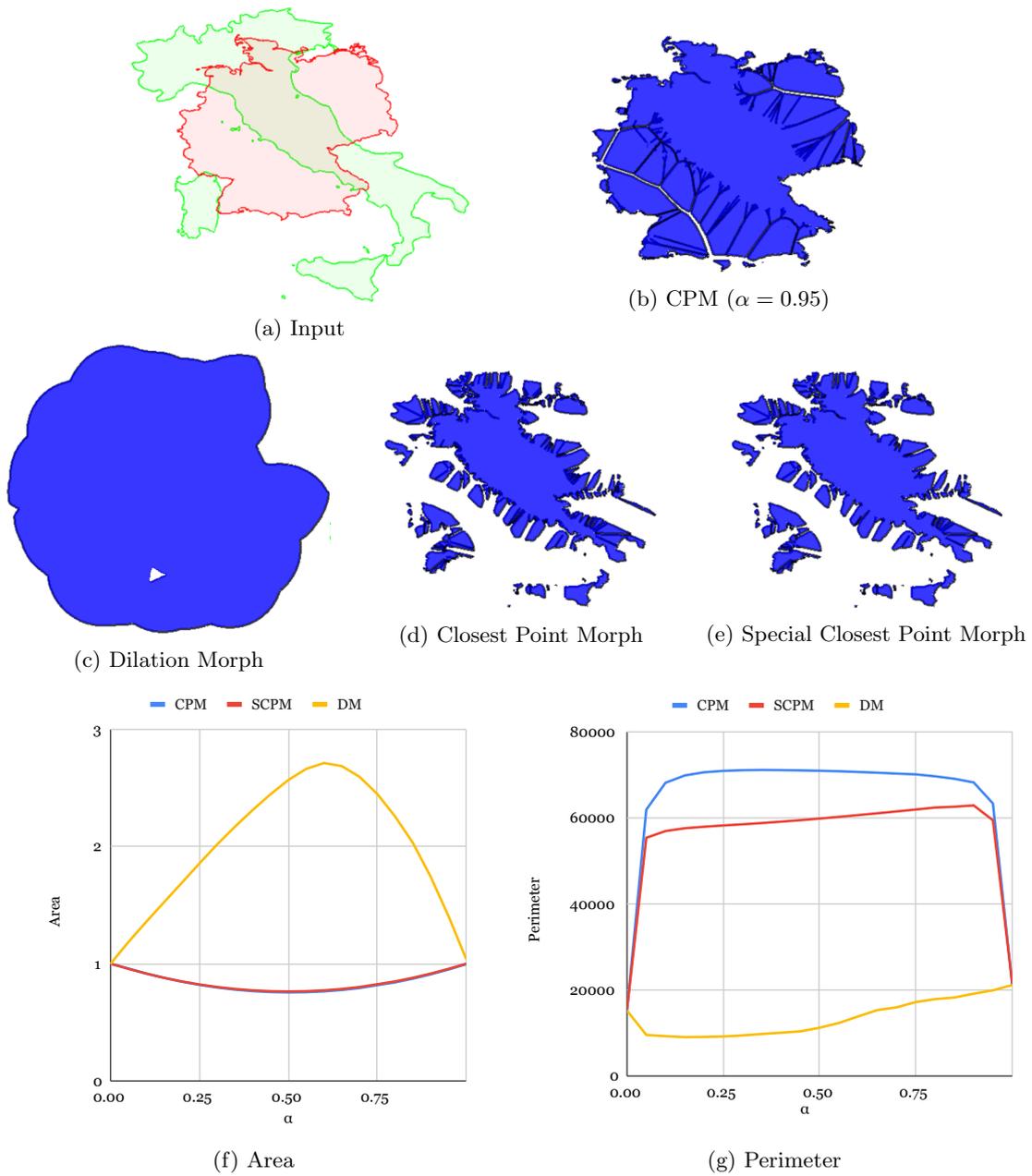


Figure 31: The results of the morph of Germany to Italy with normalised scale and centroid position. The images for each of the morphs was made with $\alpha = 0.5$. The lowest area is 0.7556 for the CPM and 0.7635 for the SCPM and the highest area for the DM is 2.7134.

Since the shapes overlap so little the DM becomes unrecognisable. Italy has land from the north-west to the south-east and Germany is slightly north-east to south-west. Neither of those really comes through and the small details are also lost because of the dilation.

8.2.1 Ireland Spain Combination

The results for this combination can be found in Figure 32. The most obvious observation is that the dilation morph is huge. This is because Spain has a couple of small islands quite far away, which makes the Hausdorff distance and therefore the amount of dilation very big. This is true for all combinations containing Spain. The perimeter is interesting as well, the dilation causes the morph to quickly lose detail even at low α which makes the perimeter go down sharply, then it starts going up again as the morph becomes bigger and then down again for the same reason.

The CPM and SCPM give fairly natural results. There is a lot of overlap between mainland Spain and Ireland, and both shapes retain their features quite well. The islands are not as big of a problem for the CPM and SCPM, they simply morph towards the mainland mostly keeping their original shape. As we will see in the section 8.3, disconnected components like this often don't give great results, but for tiny islands like this it actually works quite well.

8.3 Moving source and target apart

For this section we will not look at as many different combinations, since they mostly follow the same pattern. The average results of the animal dataset can be found in Figure 33. The difference between the CPM and SCPM quickly becomes smaller as the distance increases. At larger distances the situations where the SCPM makes a difference usually do not occur, so the morphs end up being the same. The area of the Closets Point Morph seems to be approaching 0.5, which is the lowest the area can be (Theorem 5.2). This can be explained by the fact that the further away the shapes are from each other, in general, more of both shapes is in voronoi cells that correspond to a point in the input shapes instead of an edge. And if all morphing chunks are morphed to a point then the area would be 0.5 with $\alpha = 0.5$.

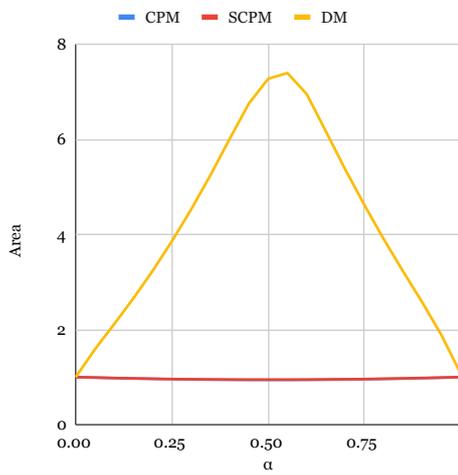
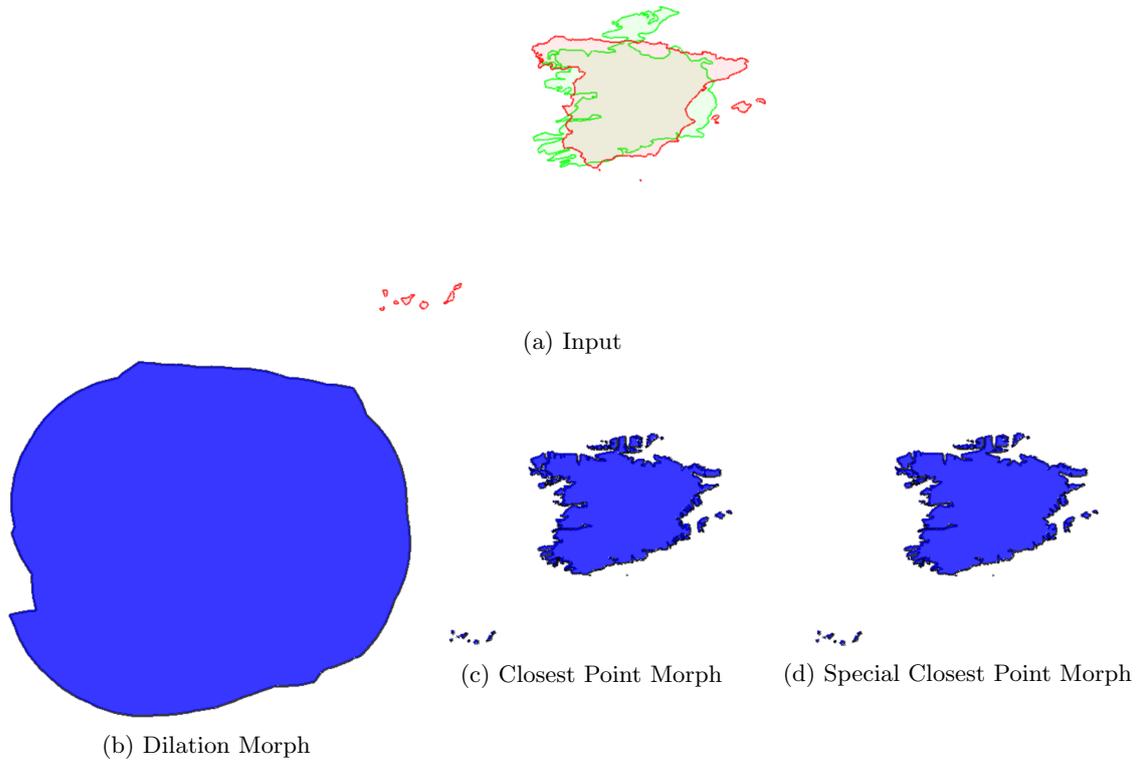
As the shapes are pulled further apart the Hausdorff distance will of course increase, and so the Dilation Morph continuously becomes bigger and bigger as well. The perimeter and area of the Dilation Morph will continue growing if we continue pulling the shapes further apart. We will look at a couple of examples of shapes being pulled apart.

8.3.1 Dog Ostrich combination

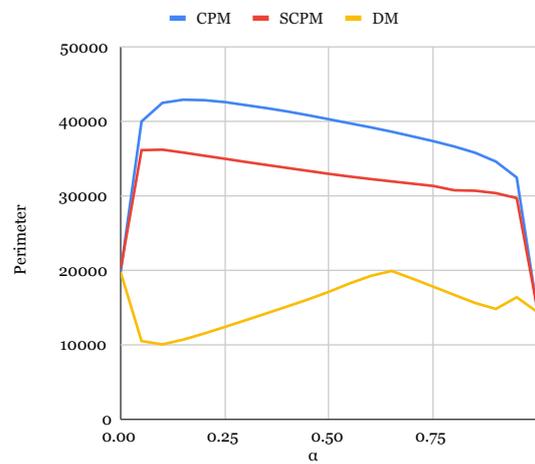
The results for this combination can be found in Figure 34. With a small distance between the centroids the area is actually bigger than when there is no distance at all, as the shapes actually overlap slightly better. When the distance = 4000 the morph starts looking quite strange, for example, the butt of the dog is entirely morphed towards one vertex in the beak of the ostrich. This creates a large barely attached piece that goes up a lot higher than feels natural. When the shapes are far enough apart, the resulting SCPM is simply a smaller version of both shapes attached to each other.

8.3.2 Bird Horse combination

The results of this combination are shown in Figure 35. This combination shows the type of fragmenting behaviour that is common when shapes are far apart, where there are only a few voronoi cells that split the shapes and large parts of both shapes are fully intact, recognisable. Usually these shapes are either not connected to the rest of the morph, or connected in only one single point.

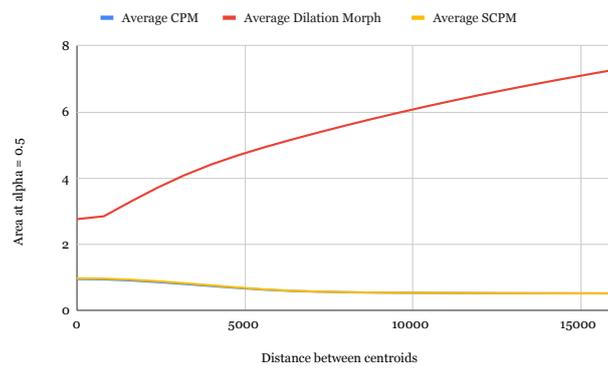


(e) Area

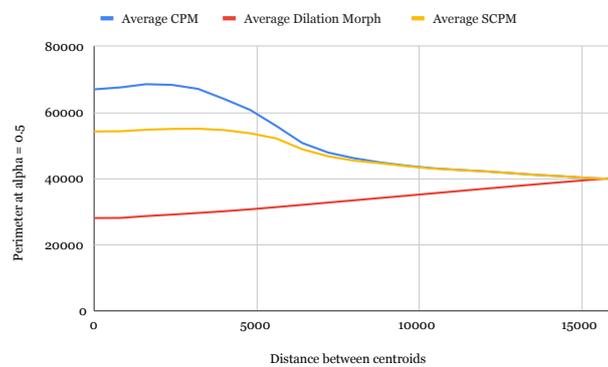


(f) Perimeter

Figure 32: The results of the morph of Germany to Italy with normalised scale and centroid position. The images for each of the morphs was made with $\alpha = 0.5$. The lowest area is 0.9419 for the CPM and 0.9479 for the SCPM and the highest area for the DM is 7.3950.



(a) Area



(b) Perimeter

Figure 33: The average effect of moving the centroid positions apart on the area (a) and perimeter (b), as tested on all combinations of shapes from the animal dataset with $\alpha = 0.5$.

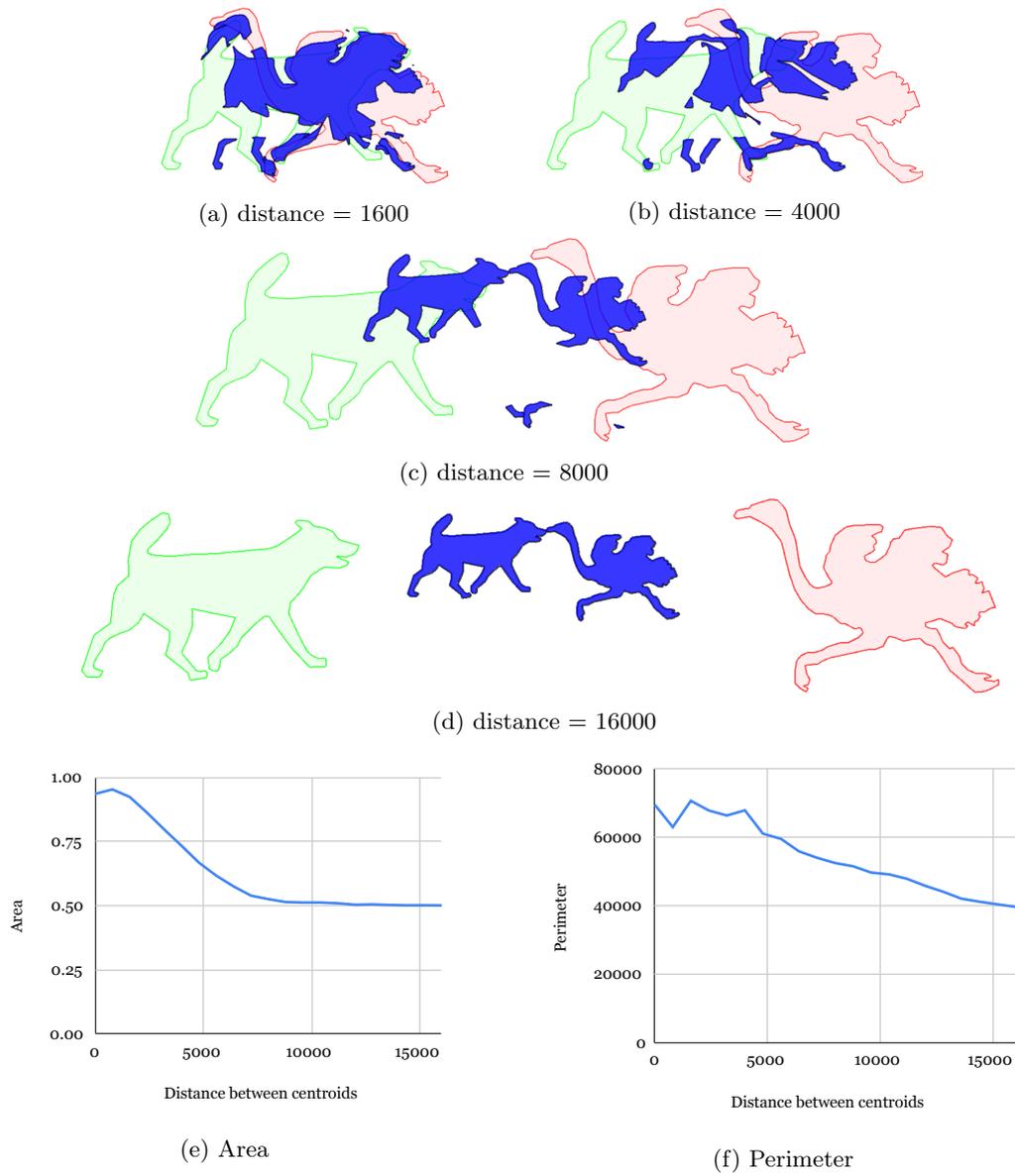
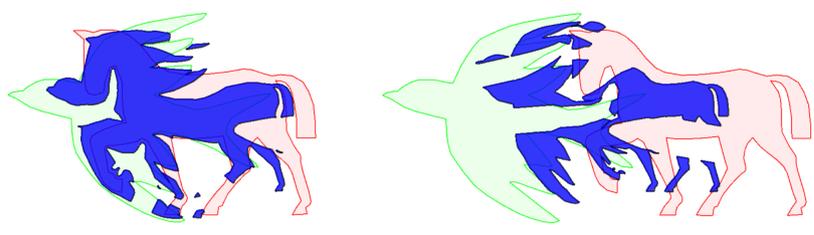
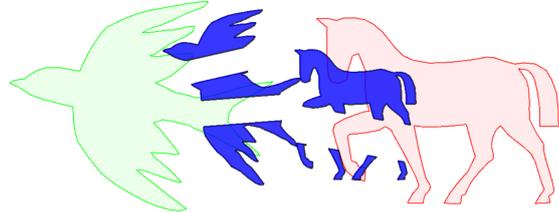


Figure 34: The results of moving the centroid distance between the dog and ostrich shapes. The morph shown is the SCPM with $\alpha = 0.5$.

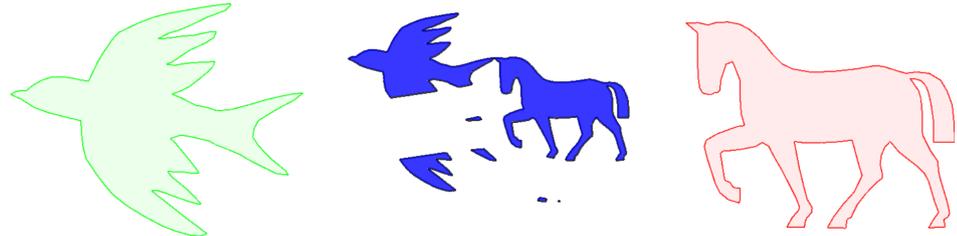


(a) distance = 1600

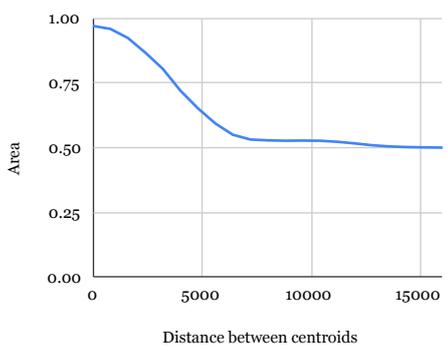
(b) distance = 4800



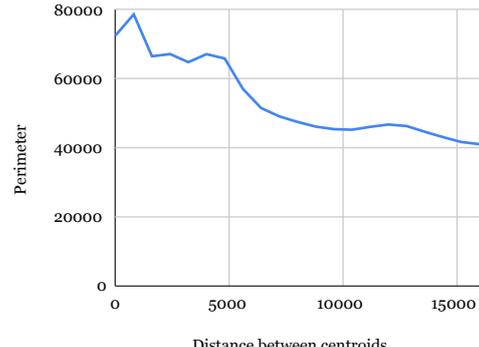
(c) distance = 8000



(d) distance = 16000

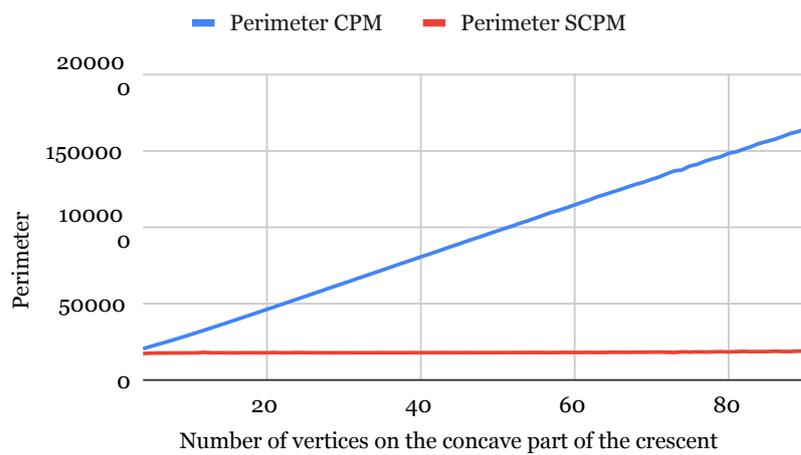
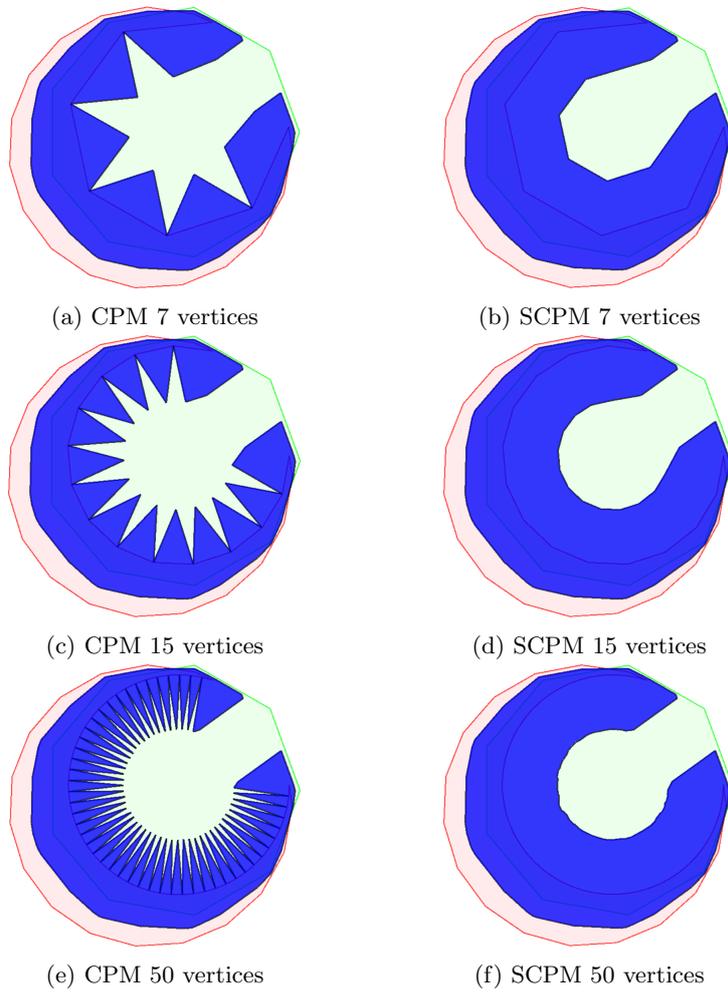


(e) Area



(f) Perimeter

Figure 35: The results of moving the centroid distance between the bird and horse shapes. The morph shown is the SCPM with $\alpha = 0.5$.



(g) The perimeter for different values of numbers of vertices on the concave part of the crescent.)

Figure 36: The results of the crescent test, α is always 0.5.

8.4 Crescent Experiment

The effect of number of vertices in the concave part of the crescent on the perimeter and the visual results can be seen in Figure 36. As you can see the line linearly goes up for the CPM, the total perimeter is about 7.8 times higher for 90 vertices as it is for 4 vertices, while the SCPM stays roughly consistent. The morphs do end up looking a bit jagged when the number of vertices becomes too high, likely do to problems with precision, this does not seem to have a big impact on the perimeter however.

9 Conclusion

We looked at the Dilation Morph, Closest Point Morph and the more experimental Special Closest Point Morph and compared their effectiveness in different combinations for the animal dataset and the countries dataset.

The Dilation Morph generally gave results that were way too big to seem natural, especially in the country dataset where the Hausdorff distance was bigger on average. It does fine when the Hausdorff distance is small, but this was rarely the case in the combinations we tested.

The CPM tends to work best when there is a lot of overlap between the input shapes and the vertex count is quite low. It tends to create quite a lot of big cuts and floating pieces otherwise. Given two input shapes with area 1 the lowest possible area in the CPM is $\frac{1}{2}$ and the highest is 1. The number of connected components stays the same for any $0 < \alpha < 1$.

The SCPM can be quite an improvement over the CPM, however its effectiveness depends quite a lot on the input. It works well in situations with low vertex counts and when the vertices follow a long smooth curve. When this is not the case, such as in the Country dataset, it still tends to make small improvements over the CPM, but you will have to look a lot closer to find them. Additionally the shape can get a bigger area than 1 in the SCPM, it can also create holes that do not exist in the input shapes or the CPM and it is not a Hausdorff Morph.

10 Future work

There are several open problems left which will be briefly discussed here.

The most obvious problem to be solved is that, while the Special Closest Point Morph improves the quality of the morph in many cases, it does not always give valid results. As it was a fairly late addition to the project there was no time to fully work out a definition that works well for all simple polygons.

One of the most noticeable artifacts in the CPM is that it can create small disconnected chunks, such as in figure 14. This happens when an input shape intersects with an edge of a Voronoi cell, where the corresponding points or lines of the cells are not directly attached, and a large part of the shape is on one side of the edge and a small part is on the other side. It would feel more natural if that small part that goes the other way would instead move together with the larger chunk. It would be interesting to see if a solution could be made that leaves the morph more connected, perhaps by deforming the Voronoi diagram before using the cells. This would however probably not be possible as a Hausdorff morph.

The perimeter of the standard Closest Point Morph can get very high and the area somewhat low. The SCPM is an improvement, but alternative untested solutions are possible. The Dilation morph has the opposite problem. One interesting idea is to take the result of the CPM and dilate it until the area is a linear interpolation of the areas of the source and target shapes. This would not only keep the area more consistent, but it would also lower the perimeter. It could also be dilated to have a slightly higher area, which could smooth out some other issues such as the thin holes that can exist in the SCPM (Figure 29.b). This by itself would not result in a Hausdorff morph, however it could be easily turned into one by taking the dilated result and intersecting it with the Dilation Morph. Since the dilation morph is the maximal Hausdorff morph, this would allow us to keep as much of the dilation as possible while still keeping it as a Hausdorff morph.

One big way we can improve the quality of the CPM is by normalizing the area and position. However, this is not necessarily the optimal position for the morph. Other orientations could be tested, for example one that maximizes overlap between the shapes, perhaps with some rotation as well.

References

- [1] Marc Alexa, Daniel Cohen-Or, and David Levin. As-rigid-as-possible shape interpolation. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '00*, page 157–164, 2000.
- [2] Boris Aronov, Raimund Seidel, and Diane Souvaine. On compatible triangulations of simple polygons. *Computational Geometry: Theory and Applications*, 3:27–35, 1993.
- [3] Gill Barequet and Amir Vaxman. Nonlinear interpolation between slices. *IJSM*, 14:97–107, January 2007.
- [4] Boost. Boost C++ Libraries. <http://www.boost.org/>, 2020. Version used Boost 1.73.0.
- [5] Quirijn W. Bouts, Irina Irina Kostitsyna, Marc van Kreveld, Wouter Meulemans, Willem Sonke, and Kevin Verbeek. Mapping Polygons to the Grid with Small Hausdorff and Fréchet Distance. In *24th Annual European Symposium on Algorithms (ESA 2016)*, volume 57, pages 22:1–22:16, 2016.
- [6] Daniel Cohen-Or, Amira Solomovic, and David Levin. Three-dimensional distance field metamorphosis. *ACM Trans. Graph.*, 17(2):116–141, April 1998.
- [7] Alec Jacobson, Ilya Baran, Jovan Popović, and Olga Sorkine-Hornung. Bounded biharmonic weights for real-time deformation. *Commun. ACM*, 57(4):99–106, April 2014.
- [8] Henry Johan, Yuichi Koiso, and Tomoyuki Nishita. Morphing using curves and shape interpolation techniques. In *Proceedings the Eighth Pacific Conference on Computer Graphics and Applications*, pages 348–454, 2000.
- [9] David Levin. Multidimensional Reconstruction by Set-valued Approximations. *IMAJNA*, 6(2):173–184, April 1986.
- [10] Bjorn Sandvik. World borders dataset, 2009. Data retrieved from Thematic Mapping, https://thematicmapping.org/downloads/world_borders.php.
- [11] Justin Solomon, Fernando de Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, Andy Nguyen, Tao Du, and Leonidas Guibas. Convolutional Wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Trans. Graph.*, 34(4):66:1–66:11, July 2015.
- [12] Greg Turk and James F. O’Brien. Shape transformation using variational implicit functions. In *ACM SIGGRAPH 2005 Courses*, 2005.
- [13] Marc van Kreveld, Tillmann Miltzow, Tim Ophelders, Willem Sonke, and Jordi L. Vermeulen. Between Shapes, Using the Hausdorff Distance. In *31st International Symposium on Algorithms and Computation (ISAAC 2020)*, volume 181, pages 13:1–13:16, 2020.
- [14] Yanlin Weng, Weiwei Xu, Yanchen Wu, Kun Zhou, and Baining Guo. 2d shape deformation using nonlinear least squares optimization. *The Visual Computer*, 22(9-11):653–660, 2006.
- [15] Han-Bing Yan, Hu Shimin, Ralph Martin, and Yong-Liang Yang Yang. Shape deformation using a skeleton to drive simplex transformations. *IEEE TVCG*, 14(3):693–706, 2008.