

國立交通大學

資訊科學系

碩士論文

形態變形法與等位函數變形法之比較

A Comparison between Image Morphing
by Morphological Interpolations
and Level Set Methods

研究生：王蕙綾

指導教授：薛元澤 教授

中華民國九十四年六月

形態變形法與等位函數變形法之比較
A Comparison between Image Morphing by
Morphological Interpolations and Level Set Methods

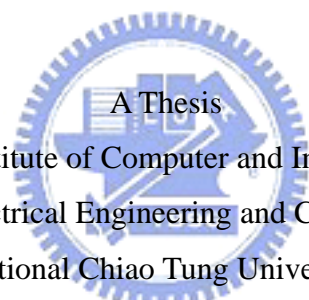
研究生：王蕙綾

Student : Hui-Ling Wang

指導教授：薛元澤

Advisor : Dr. Yuang-Cheh Hsueh

國立交通大學
資訊科學系
碩士論文



Submitted to Institute of Computer and Information Science
College of Electrical Engineering and Computer Science
National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer and Information Science

June 2005

Hsinchu, Taiwan, Republic of China

中華民國九十四年六月

形態變形法與等位函數變形法之比較

研究生：王蕙綾

指導教授：薛元澤教授

國立交通大學資訊科學研究所

摘要

現在有許多方法可以達到影像變形的目的。但是，這些方法在影像變形過程中常常伴隨著不適當的現象——“鬼影”。在我的論文中，我們將會討論兩種可避免鬼影的方法。這兩種方法分別為數學型態差補法 (Morphological interpolations) 以及等位函數集合法 (Level Set Methods)。更詳盡的說，在數學型態學中，我們比較數種型態差補法，同時我們以 blending 來代表等位函數法。在比較過型態差補法和 blending 的結果後，我們嘗試利用其中一種型態差補的影像來強化 blending 的影像。實際上，我們可以發現到強化過的影像明顯的比原本的blending 影像明亮許多且變形過程一樣流暢。

A comparison between Image Morphing by Morphological Interpolations and Level Set Methods

Student : Hui-Ling Wang

Advisor : Dr. Yuang-Cheh Hsueh

Department of Computer and information Science

National Chiao Tung University



Abstract

There are several ways to achieve image morphing. But it always goes along with some improper phenomenon – “ghost effect” in morphing process. To avoid this phenomenon, in this paper, we will study on morphological interpolations and morphing by level set methods. More detailed, in our experiments, we implement several different interpolating ways to achieve morphological interpolations; and we implement blending to represent level set methods. After comparing the implement results of several kinds of morphological interpolations and blending, we make an attempt to enhance the blending morphed images reusing distance-based interpolation. Actually, we can observe that the results of enhanced blending are more brighten than original blending morphed images.

誌謝

首先對於我的指導教授 薛元澤教授獻上最誠摯的感謝，感謝他對我的論文細心的指導。在這兩年的學習中，他細心的教導與照顧，對於學生的身教與言教，不僅讓學生領略到作學問的樂趣，更學習到生活的道理。

此外，我也要感謝我的論文口試委員：黃仲陵老師、莊榮宏老師，對於本論文提供許多寶貴的意見。

另外還要感謝實驗室的王聖博學長、石永靖學長，以及同學何昌憲、高薇婷、莊逢軒，他們在論文方面的建議及幫助讓我獲益匪淺。

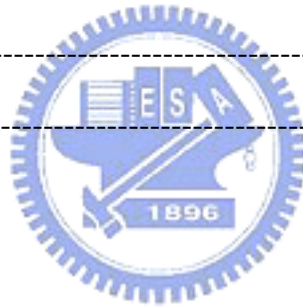
謹將此論文獻給我親愛的家人，感謝他們支持我完成學業，祝福他們永遠健康快樂。



Contents

I.	Introduction	1
1.1	Motivation -----	1
1.2	Previous Work -----	2
1.3	Organization of this thesis -----	5
II.	Mathematical Morphology	7
2.1	Fundamentals -----	7
2.1.1	Minkowski Algebra -----	7
2.1.2	Binary Morphological operations -----	8
2.1.3	Gray-level Morphological operations -----	10
2.2	Distance Transformation and Morphology -----	11
2.3	Morphological interpolation -----	15
2.3.1	Hausdorff distance interpolation -----	16
2.3.2	Distance-based interpolation -----	17
2.3.3	Median set interpolation -----	22
2.3.4	Shape-based interpolation -----	26
III.	Level Set Methods	28
3.1	Implicit functions -----	28
3.2	Signed distance function -----	29
3.3	Level set formula -----	30
3.4	Deforming Methods -----	37
3.4.1	Image Blending -----	38
3.4.2	Radial Basis function -----	40
IV.	Experimentation	43

4.1	Mathematical morphology interpolation -----	43
4.1.1	Shape-based morphology interpolation -----	43
4.1.2	Haudroff distance interpolation -----	45
4.1.3	Distance-based interpolation -----	46
4.1.4	Median interpolation -----	48
4.1.5	Discussion -----	48
4.2	Level Set Methods -----	49
4.2.1	Blending -----	49
4.3	Enhanced morphing -----	52
4.4	Combination of the two methods -----	55
V.	Conclusions	58
5.1	Discussion -----	58
5.2	Feature work -----	59
References		61



List of Figures

Figure 1-1 (a)	The original 128x128 source image -----	2
Figure 1-1 (b)	The result of linear interpolation -----	2
Figure 1-1 (c)	The original 128x128 target image -----	2
Figure 1-2	The relation of P , Q and X in source image -----	3
Figure 1-3	The relation of P , Q and X in target image -----	3
Figure 2-1 (a)	The set A and structuring element B -----	8
Figure 2-1 (b)	The result of $A \oplus B$ -----	8
Figure 2-1 (c)	The result of $A \ominus B$ -----	8
Figure 2-2	The upper left is the original set A , the right one is the structuring element S and show S^{\vee} in the same time; the lower left is after dilating A , the right one is after eroding A -----	9
Figure 2-3 (a)	The original 256x256 house -----	9
Figure 2-3 (b)	The dilated house -----	10
Figure 2-3 (c)	The eroded house -----	10
Figure2-4 (a)	The original 128x128 source image -----	17
Figure2-4 (b)	Hausdorff distance interpolation of $Z_{0.25}$ -----	17
Figure2-4 (c)	Hausdorff distance interpolation of $Z_{0.5}$ -----	17
Figure2-4 (d)	Hausdorff distance interpolation of $Z_{0.75}$ -----	17
Figure2-4 (e)	The original 128x128 target image -----	17
Figure2-5 (a)	The original images P and Q -----	18
Figure2-5 (b)	The distance function d_1 in $P/(P \cap Q)$ -----	19
Figure2-5 (c)	The distance function d_2 in $P/(P \cap Q)$ -----	19
Figure2-6 (a)	The original image P -----	19
Figure2-6 (b)	The original image Q -----	19

Figure2-6 (c)	The distance function $d = \frac{d_1}{d_1 + d_2}$ in P	20
Figure2-6 (d)	The distance function $d = \frac{d_1}{d_1 + d_2}$ in Q	20
Figure2-7 (a)	The distance function $d = \frac{d_1}{d_1 + d_2}$ in P	20
Figure2-7 (b)	The threshold of distance function $Thr_{0.5}(d)$ in P	20
Figure2-8 (a)	The original source image P	21
Figure2-8 (b)	The morphed image $Z_{0.25}$	21
Figure2-8 (c)	The morphed image $Z_{0.5}$	21
Figure2-8 (d)	The morphed image $Z_{0.75}$	21
Figure2-8 (e)	The original target image Q	21
Figure2-9 (a)	The first input image X	23
Figure2-9 (b)	The second input image Y	23
Figure2-9 (c)	The median $M(X, Y)$	23
Figure2-10 (a)	The original source image X	24
Figure2-10 (b)	The $M_{1/4}(X, Y)$	24
Figure2-10 (c)	The $M_{1/2}(X, Y)$	24
Figure2-10 (d)	The $M_{3/4}(X, Y)$	24
Figure2-10 (e)	The original target image Y	24
Figure3-1	The implicit function ϕ	29
Figure3-2	The relation of signed mean curvature κ and the appearance	32
Figure3-3	The static formulation	34
Figure3-4 (a)	The original shape	35
Figure3-4 (b)	The change of the shape	35
Figure3-5	The dynamic formulation	35
Figure3-6 (a)	The one stuff to do in-between images	38
Figure3-6 (b)	The other stuff to do in-between images	38

Figure3-7 (a)	Show that it start from source and move toward to target in one way -----	39
Figure3-7 (b)	It move start source and target, then it encounter in one position ---	39
Figure4-1 (a)	The original 128x128 source image -----	45
Figure4-1 (b)	The 20% shape-based interpolation morphed image -----	45
Figure4-1 (c)	The 40% shape-based interpolation morphed image -----	45
Figure4-1 (d)	The 60% shape-based interpolation morphed image -----	45
Figure4-1 (e)	The 80% shape-based interpolation morphed image -----	45
Figure4-1 (f)	The original target image -----	45
Figure4-2 (a)	The original 128x128 source image -----	46
Figure4-2 (b)	The 20% Haudroff distance interpolation morphed image -----	46
Figure4-2 (c)	The 40% Haudroff distance interpolation morphed image -----	46
Figure4-2 (d)	The 60% Haudroff distance interpolation morphed image -----	46
Figure4-2 (e)	The 80% Haudroff distance interpolation morphed image -----	46
Figure4-2 (f)	The original target image -----	46
Figure4-3 (a)	The original 128x128 source image -----	47
Figure4-3 (b)	The 20% distance-based interpolation morphed image -----	47
Figure4-3 (c)	The 40% distance-based interpolation morphed image -----	47
Figure4-3 (d)	The 60% distance-based interpolation morphed image -----	47
Figure4-3 (e)	The 80% distance-based interpolation morphed image -----	47
Figure4-3 (f)	The original target image -----	47
Figure4-4 (a)	The original 128x128 source image -----	48
Figure4-4 (b)	The $M_{1/4}$ of median interpolation morphed image -----	48
Figure4-4 (c)	The $M_{1/2}$ of median interpolation morphed image -----	48
Figure4-4 (d)	The $M_{3/4}$ of median interpolation morphed image -----	48
Figure4-4 (e)	The $M_{7/8}$ of median interpolation morphed image -----	48

Figure4-4 (f)	The original target image -----	48
Figure4-5 (a)	The original 128x128 source image -----	50
Figure4-5 (b)	The 20% blending morphed image -----	50
Figure4-5 (c)	The 40% blending morphed image -----	51
Figure4-5 (d)	The 60% blending morphed image -----	51
Figure4-5 (e)	The 80% blending morphed image -----	51
Figure4-5 (f)	The original target image -----	51
Figure4-6 (a)	The original 128x128 source image -----	53
Figure4-6 (b.1)	The 20% blending morphed image -----	53
Figure4-6 (b.2)	The 20% enhanced blending morphed image -----	53
Figure4-6 (c.1)	The 40% blending morphed image -----	53
Figure4-6 (c.2)	The 40% enhanced blending morphed image -----	53
Figure4-6 (d.1)	The 60% blending morphed image -----	54
Figure4-6 (d.2)	The 60% enhanced blending morphed image -----	54
Figure4-6 (e.1)	The 80% blending morphed image -----	54
Figure4-6 (e.2)	The 80% enhanced blending morphed image -----	54
Figure4-6 (f)	The original target image -----	54
Figure4-7	An example for the original input image -----	55
Figure4-8 (a)	The original 128x128 source image -----	55
Figure4-8 (b)	The decompose by the Figure4-7 -----	55
Figure4-8 (c)	The decompose by the Figure4-7 -----	55
Figure4-9 (a)	The original 128x128 source image -----	56
Figure4-9 (b)	The morphing image sequence -----	56
Figure4-9 (c)	The morphing image sequence -----	56
Figure4-9 (d)	The morphing image sequence -----	56
Figure4-9 (e)	The morphing image sequence -----	56

Figure4-9 (f)	The original target image -----	56
Figure4-10 (a)	The original 128x128 source image -----	56
Figure4-10 (b)	The morphing image sequence -----	56
Figure4-10 (c)	The morphing image sequence -----	56
Figure4-10 (d)	The morphing image sequence -----	57
Figure4-10 (e)	The morphing image sequence -----	57
Figure4-10 (f)	The original target image -----	57



List of Table

Table 2-1 (a)	City-block distance -----	11
Table 2-1 (b)	City-block distance -----	11
Table2-2 (a)	Chessboard distance -----	12
Table2-2 (b)	Chessboard distance -----	12
Table2-3	Euclidean distance -----	13
Table2-4 (a)	The forward chamfer distance -----	13
Table2-4 (b)	The backward chamfer distance -----	13
Table2-5	Show the distance transformation values from the boundary, the heavy line is the boundary -----	26
Table4-1	Shows the time consuming and the results -----	49



Chapter 1

Introduction

1.1 Motivation

Image morphing has received much attention in recent years. It is the animated transformation from one digital image to another, therefore image morphing is a method we use to construct a smooth, natural sequence of images in order to “morph” one image into another. Since it is a morphing process, the sequence constructed usually has the effect of gradually fading out the first image (“source”) then gradually fading in the second one (“target”). The sequence may be constructed in several different ways, unfortunately there are currently no “set” method to construct such sequence for every pair of images.



Traditionally, image morphing can be achieved by cross-dissolves. Linear interpolation is the simplest method in interpolation. The equation for linear interpolation can be written as $m = \alpha u + \beta v$, where m is the middle image, u is the source image, v is the target image, and α and β are real numbers satisfied the constraint $\alpha + \beta = 1$. Morphing by linear interpolation is very simple and fast but the result is not good enough. The most seriously problem is the double-exposure effect in misaligned regions. This effect also called “ghost effect” is due to averaging grayscale pixels in source image and in target image. We can observe this effect in Figure1-1. Therefore, this approach is not suitable for general applications.

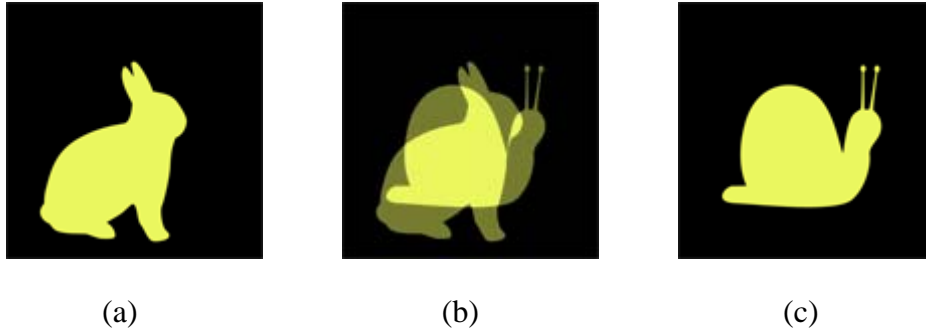


Figure1-1. Linear interpolation; (a) is the source image; (b) is the middle image with obvious ghost effect; (c) is the target image.

For the above reason, we need more powerful methods to accomplish image morphing. In this regard, mathematical morphological interpolations and level set methods are preferable to linear interpolation. Before we start to introduce our emphasis, we will survey some previous work.



1.2 Previous work

Image morphing, also called image metamorphosis, can be roughly classified into two groups, landmark-based approaches and image-based approaches. The former, landmark-based approaches, are sometimes named feature-based approaches. To perform landmark-based approach, we need specify the landmarks (features) manually, those landmarks are defined by pairs of points or line segments lied in both images.

For example, in Feature-Based Image Metamorphosis [1], proposed by T. Beier and S. Neely, it defines a pair of points X and X' , that X is in the destination image while X' is located in the corresponding position in the source image, and a pair of lines PQ and $P'Q'$. Here, the pair of points and the pair of lines are usually

chosen with the corresponding characteristics manually. Suppose the implementation is reverse mapping, which is from the destination image to the source image. Line $P^{\wedge}Q^{\wedge}$ is in the source image and line PQ in the target image. Then, we can calculate the X^{\wedge} in source image by using PQ , $P^{\wedge}Q^{\wedge}$ and X with the mapping functions. To perform image-based approach, the features are always provided by image alone, such as pixel intensities or pixel grayscales.

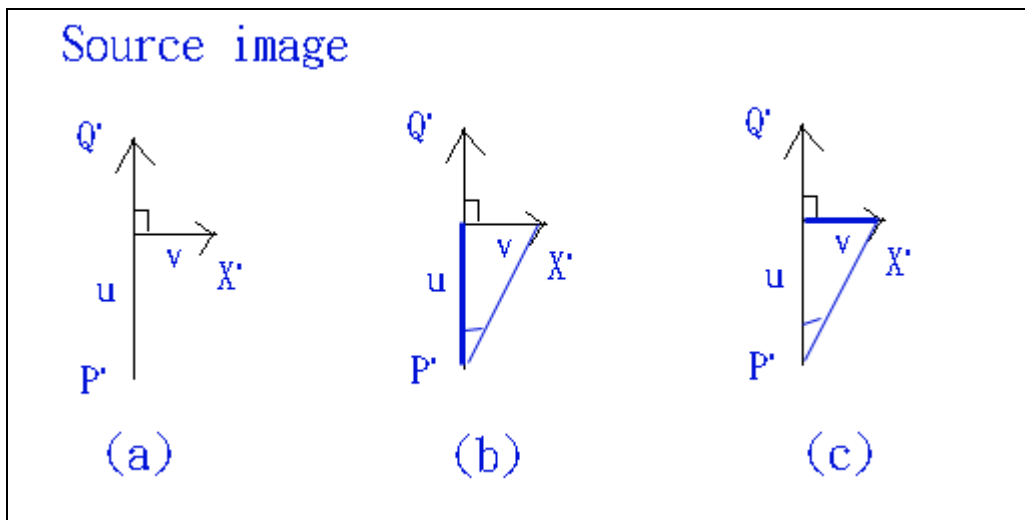


Figure1-2 (a) shows the relation of P^{\wedge} , Q^{\wedge} , and X^{\wedge} ; (b) the heavy line indicates u ; (c) the heavy line indicates v in source image

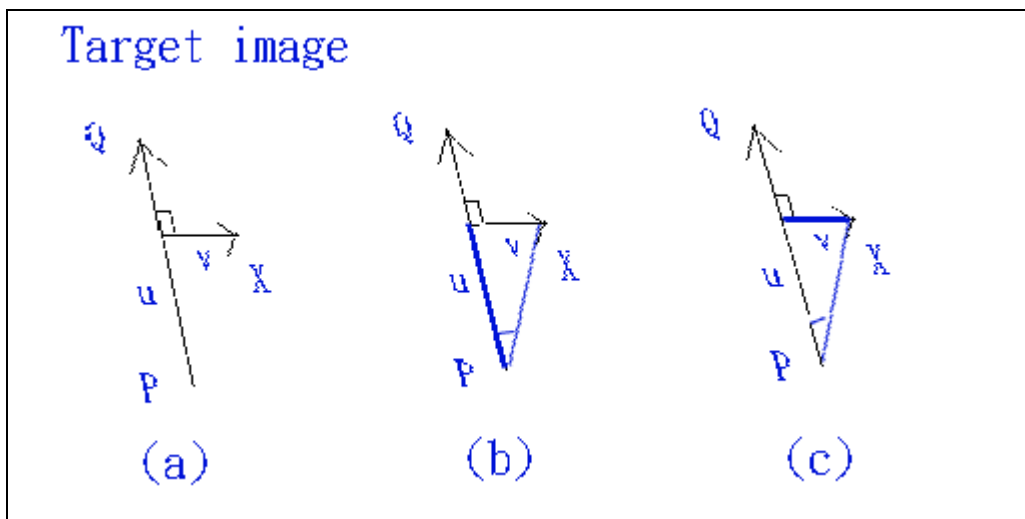


Figure1-3 (a) shows the relation of P , Q , and X ; (b) the heavy line indicates u ;

(c) the heavy line indicates v in source image

The mapping functions of landmark-based approaches are defined on a relatively small space while those of image-based approaches are defined on a much larger space. In the landmark-based approach proposed in [1], the mapping function is defined as:

$$X' = P' + u \cdot (Q' - P') + \frac{v \cdot \text{Perpendicular}(Q' - P')}{\|Q' - P'\|} \quad (0.1)$$

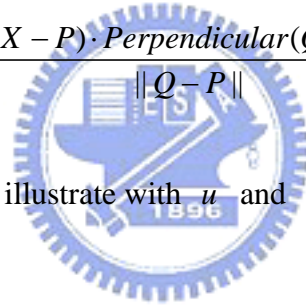
where

$$u = \frac{(X - P) \cdot (Q - P)}{\|Q - P\|^2} \quad (0.2)$$

and

$$v = \frac{(X - P) \cdot \text{Perpendicular}(Q - P)}{\|Q - P\|} \quad (0.3)$$

Also, we can use Figure1-2 to illustrate with u and v .



In an image-based approach, the linear interpolation that is described above is only one of the methods that are used. The morphology interpolation and level set methods are also used for image-based approaches, and since exhaustive description of those methods will take too much unnecessary space, we will skip them and continue on.

In Recent Advances in image Morphing [2], Wolberg lists some famous image morphing approaches. They are mesh warping, field morphing [3], radial basis functions [1], thin plate splines [4,5], energy minimization [6], and multilevel free-form deformations [7]. We can find that almost all of them are landmark-based except energy minimization.

1.3 Organization of this thesis

Up to now, we have mentioned some traditional image morphing approaches. In this paper, we will introduce mathematical morphology, level set methods, their implementations, discuss experimental results, and make our conclusion.

In chapter 2, the contents include mathematical morphological fundamentals, distance transformations, and morphological interpolation methods. Dilation and erosion are two basic operations in mathematical morphology. Distance transformations play important roles in mathematical interpolations. Morphological interpolations contain Hausdroff distance interpolation, proposed by Serra [14] [15], Distance-based interpolation function, proposed by Meyer [16], Median interpolation, proposed by Beucher [19], and Shape-based morphological interpolation, proposed by Udupa [20].

In chapter 3, the contents include the theory of level set methods and the application of them. Osher and Sethian introduce level sets for computing moving wave front. Since that time, level set methods have been proved useful for curve and surface reconstruction, shape reconstruction, image processing, and geometric modeling. Especially, level sets have been proven a robust morphing approach. Moreover, we will affix a blending (morphing) method proposed by Whitaker [32] [33] and a radial basis function using level set ideas proposed by Mullan, et. Al [37].

In chapter 4, after introducing mathematical morphology and level sets in details, we will implement those methods step by step. There, we will present our

results with figures. In chapter 5, we will make a conclusion about this paper.



Chapter 2

Mathematical Morphology

To attain the goal that image morphing should be satisfied with the human perception, mathematical morphology and level set methods both provide good theoretical frameworks for shape morphing. Mathematical morphology is a powerful methodology for the quantitative analysis of geometrical structures. Mathematical morphology [8] [9] was initiated in the last 1960s by Matheron and Serra at the Fontainebleau School of Mines in France. Originally, it was applied to analyze images from geological or biological specimens. In chapter 2, the basic theory and the know-how of mathematical morphological interpolations will be presented.



2.1 Fundamentals

Dilation and Erosion are two basic morphologic operations. They are close related to the Minkowski set addition and subtraction.

2.1.1 Minkowski Algebra

Let A and B be two sets in $P(E)$, where E is an n -dimensional Euclidean space and $P(E)$ is the power set of E .

The *Minkowski addition* of set A and B , written as $A \oplus B$, is defined by

$$A \oplus B = \{a + b \mid a \in A \text{ and } b \in B\} = \bigcup_{b \in B} A_b \quad (2.1)$$

where $A_b = \{a + b \mid a \in A\}$ is the translation of A by the vector $b \in E$.

The *Minkowski subtraction* of B from A , written as $A \ominus B$, is defined by

$$A \ominus B = \bigcap_{b \in B} A_{-b} = \bigcap_{b \in B^\vee} A_b \quad (2.2)$$

where the $B^\vee = \{-b \mid b \in B\}$ represents the reflected set of B .

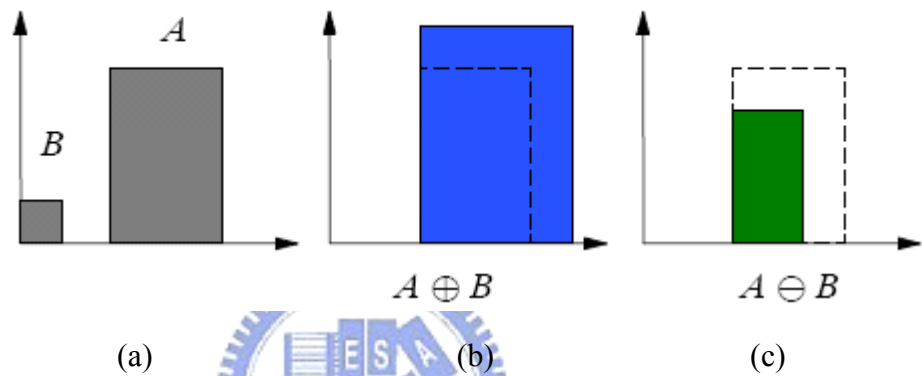


Figure 2-1. (a) is the set A and the structuring element B ; (b) is $A \oplus B$; (c) is $A \ominus B$

2.1.2 Binary Morphological operations

Let A and B be two sets in $P(E)$. The dilation of A by B , written as $\mathcal{D}_B(A)$, is given by

$$\mathcal{D}_B(A) = A \oplus (-B) = A \oplus B^\vee = \{x \in E \mid B_x \cap A \neq \emptyset\} \quad (2.3)$$

The erosion of A by B , written as $\mathcal{E}_B(A)$, is given by

$$\mathcal{E}_B(A) = A \ominus B = \{x \in E \mid B_x \subset A\} \quad (2.4)$$

In here, B is usually called a structuring element.

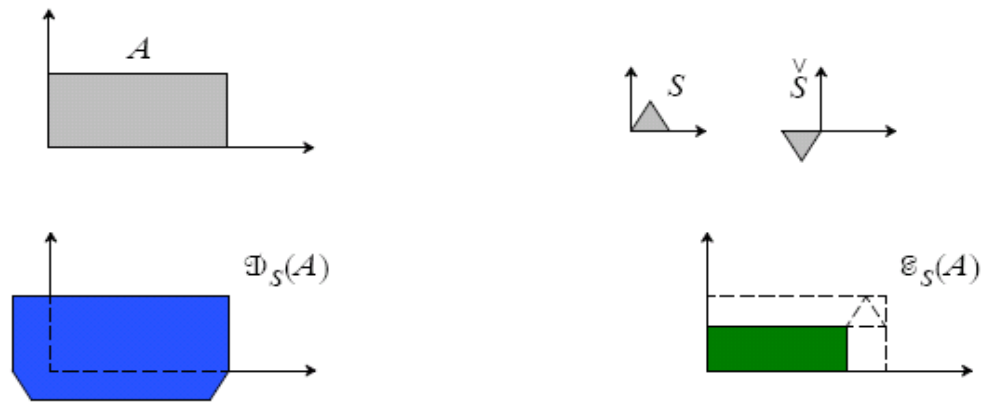


Figure2-2. The upper left is the original set A , the right one is the structuring element S and show S^v in the same time; the lower left is after dilating A , the right one is after eroding A

Another example for binary dilation and erosion with structuring element being the 3 by 3 square is shown in Figure 4.



Figure2-3. (a) Original image



Figure2-3. (b) after dilated



Figure2-3. (c) after eroded

2.1.3 Gray-level Morphological operations

In grayscale morphology, we will denote the set of all grayscale images on E by $F(E)$.

Definition.

The *dilation* \mathcal{D}_g with structuring element $g \in F(E)$ is a unary operation on $F(E)$ defined by

$$(\mathcal{D}_g(f))(x) = \sup_{y \in E} \{f(x+y) + g(y)\} \quad (2.5)$$

for all grayscale images f in $F(E)$.

Definition.

The *erosion* \mathcal{E}_g with structuring element g is a unary operation on $F(E)$ defined by

$$(\mathcal{E}_g(f))(x) = \inf_{y \in E} \{f(x+y) - g(y)\} \quad (2.6)$$

for all grayscale images f in $F(E)$.

2.2 Distance Transformation and Morphology

Distance transformations play major roles in the morphological interpolations. A distance transformation [10] converts a digital binary image into a graytone image consisting of feature and non-feature pixels. A few of common distance transformations will be presented here.

◆ City-block distance

A city block distance [11], also called D_4 distance, is defined as

$$D_4(a, b) = \sum_{i=1}^n |b_i - a_i| \quad (2.7)$$

where a and b are vectors in R^n with $a = (a_1, \dots, a_n)$ and $b = (b_1, \dots, b_n)$. The city-block distance does not allow traveling the distance between center and corner directly. In this case, all pixels with city-block distances from a less than or equal to some radius r form a diamond centered at a . The pixels b with $D_4(a, b) = 1$ are the 4-neighbors of the center a , as shown in the following table.

2	1	2
1	0	1
2	1	2

	b	
b	a	b
	b	

Table 2-1 (a)

Table 2-1 (b)

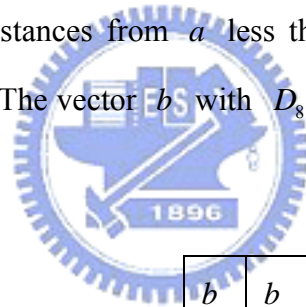
Table2-1. (a) (b) City-block distance

◆ Chessboard distance

A Chessboard distance [11], also called D_8 distance, is defined as

$$D_8(a, b) = \max_{1 \leq i \leq n} \{|b_i - a_i|\} \quad (2.8)$$

where a and b are vectors in R^n with $a = (a_1, \dots, a_n)$ and $b = (b_1, \dots, b_n)$. The chessboard distances allow a vector to go to its four corners at one step. In this case, all pixels with chessboard distances from a less than or equal to some radius r form a square centered at a . The vector b with $D_8(a, b) = 1$ are the 8-neighbors of the center a .



1	1	1
1	0	1
1	1	1

Table 2-2 (a)

b	b	b
b	a	b
b	b	b

Table 2-2(b)

Table2-2. (a) (b) Chessboard distance

◆ Euclidean distance

An Euclidean distance [11] is defined as

$$D_e(a, b) = \left(\sum_{i=1}^n (b_i - a_i)^2 \right)^{\frac{1}{2}} \quad (2.9)$$

where a and b are vectors in R^n with $a = (a_1, \dots, a_n)$ and $b = (b_1, \dots, b_n)$. The Euclidean distance can easily be calculated by using Pythagoras' theorem. $a^2 + b^2 = c^2$. This is the usual metric widely used in many areas. The disadvantages of it are the results are not always correct and the process is slow in some cases. Due to the complex feature geometry errors can be occurred, but they are always small.

$\sqrt{2}$	1	$\sqrt{2}$
1	0	1
$\sqrt{2}$	1	$\sqrt{2}$

Table2-3. Euclidean distance

◆ Chamfer distance (weighted distance)



Gunilla Borgefors [12][13] original chamfer-distance method calculates distances of non-object pixels from the nearest object pixel. Borgefors reviews a number of metrics in 2 and 3 dimensions. Chamfer distance transformations are produced in two raster scans over the image. In the forward scan, the mask starts in the upper left corner of the picture, moves from left to right and from top to bottom. In the backward scan, it starts in the lower right corner, moves from right to left and from bottom to top. The local distances, d_1 and d_2 , in the mask pixels are added to the pixel values in the distance map and the new value of the zero pixel is the minimum of the five sums.

d_2	d_1	d_2
d_1	0	

	0	d_1
d_2	d_1	d_2

Table2-4. (a)

Table2-4. (b)

To show in equations as:

Forward:

$$v_{i,j} = \min(v_{i-1,j-1} + d_2, v_{i-1,j} + d_1, v_{i-1,j+1} + d_2, v_{i,j-1} + d_1, v_{i,j}) \quad (2.10)$$

Backward:

$$v_{i,j} = \min(v_{i,j}, v_{i+1,j} + d_1, v_{i-1,j+1} + d_2, v_{i,j+1} + d_1, v_{i+1,j+1} + d_2) \quad (2.11)$$

For instance, for $d_1 = 1$ and $d_2 = \infty$, we have the city block metric; for $d_1 = d_2 = 1$, the chess board metric; for $d_1 = 1$ and $d_2 = \sqrt{2}$, Montanari's metric ; and for $d_1 = 2$ and $d_2 = 3$, Barrow's approximation; the most common is 3-4 chamfer.

◆ Hausdorff distance

In generally, the “distance” means the shortest path, if a point $a \in A$ is said to be at distance d to set B , we usually assume that d is the distance from a to the nearest point of $b \in B$. Regularly, this is called a *minimin* function, because the distance between a single point a and an object (compact set) B is defined as:

$$d(a, B) = \inf\{d(a, b) \mid b \in B\} \quad (2.12)$$

And this distance equation can also written by using the morphological notion of dilation, such as:

$$d(a, B) = \inf\{\lambda \mid a \in \mathcal{D}_\lambda(B)\} \quad (2.13)$$

hence we can get

$$\sup_{a \in A} d(a, B) = \inf\{\lambda \mid A \subseteq \mathcal{D}_\lambda(B)\} \quad (2.14)$$

in the same way, we can write

$$\sup_{b \in B} d(b, A) = \inf\{\lambda \mid B \subseteq \mathcal{D}_\lambda(A)\} \quad (2.15)$$

The Hausdorff distance is named by Felix Hausdorff (1868-1942), Hausdorff distance is the *maximum distance of a set to the nearest point in the other set*. More formally, Hausdorff distance between sets A and B is defined as

$$\rho(A, B) = \max\left\{\sup_{a \in A} d(a, B), \sup_{b \in B} d(b, A)\right\} \quad (2.16)$$

where a and b are points of sets A and B respectively, and $d(a, b)$ is any metric between these points. Changing the last equation to morphological notation:

$$\rho(A, B) = \inf\{\lambda \mid A \subseteq \mathcal{D}_\lambda(B), B \subseteq \mathcal{D}_\lambda(A)\} \quad (2.17)$$

2.3 Morphological Interpolation

To go into particulars, we will discuss some chief interpolating methods in the following :

- Hausdorff distance interpolation(Jean Serra)
- Distanced-based interpolation function (Frenand Meyer)
- Median interpolation (S.Beucher)
- Shape-based morphological interpolation (J. K. Udapa)

2.3.1 Hausdorff distance interpolation [14] [15] (Jean Serra)

The theoretical background of this interpolation method is the Hausdorff distance, and this method was first proposed by Serra in 1994. Hausdorff distance interpolation has one important disadvantage - the resulting interpolated objects are relatively large - even considerably larger than the input ones. The interpolator is a function which produces the interpolated objects between images in the interpolation sequence. It has three arguments: two input objects and the interpolation level. The interpolation level, denoted by α (a real number $0 \leq \alpha \leq 1$) indicates a position of the interpolated object in the interpolation sequence. If $\alpha = 0$, the interpolated object equals the initial object, denoted by P . If $\alpha = 1$, the interpolated object equals the final object, denoted by Q . Then, we denote an intermedium of interpolation level $0 \leq \alpha \leq 1$ by the following statement :

$$Z_{\alpha} = \mathcal{D}_{\alpha\rho}(P) \cap \mathcal{D}_{(1-\alpha)\rho}(Q) , \quad 0 \leq \alpha \leq 1 \quad (2.18)$$

where set Z_{α} turns out to be the intersection of dilates of P and of Q by the structuring element of radii $\alpha\rho$ and $(1-\alpha)\rho$ respectively.



Figure2-4 (a) source image



Figure2-4 (b) $Z_{0.25}$



Figure2-4 (c) $Z_{0.5}$



Figure2-4 (d) $Z_{0.75}$



Figure2-4 (e) target image

Figure2-4.(a) to (e) The example of Hausdorff distance interpolation



2.3.2 Distance-based interpolation (interpolation function)

“Distance-based interpolation” also named “interpolation function” was first proposed by Frenand Meyer [16]. This is a method of the binary morphological interpolation which makes use of interpolation function. It is based on the function which describes the relative distance between the objects and can be applied to binary and mosaic images. This method has a simple constraint: the two input objects must have non-empty intersection.

Let using P denote the source image and Q denote the target image, see Figure2-5 (a). We begin with overlapping the two images to get their intersection,

denoted by $R = P \cap Q$. And we will use the intersection R as a temporary target. It means that we will begin with the source image P and transform it into intersection R while on the other hand begin with intersection R and transform it into the target image Q .

Let P and Q be two sets (objects), $(R = P \cap Q) \subset P$. Interpolation function of P and Q is obtained from two base geodesic distance functions d_1 and d_2 . Distance function d_1 describes a distance from the complement of P , denotes P^c , in the area P/R , and the distance function d_2 describes a distance from R in the area P/R where P/R means the area in P but not in R . An example is shown in the following figure.

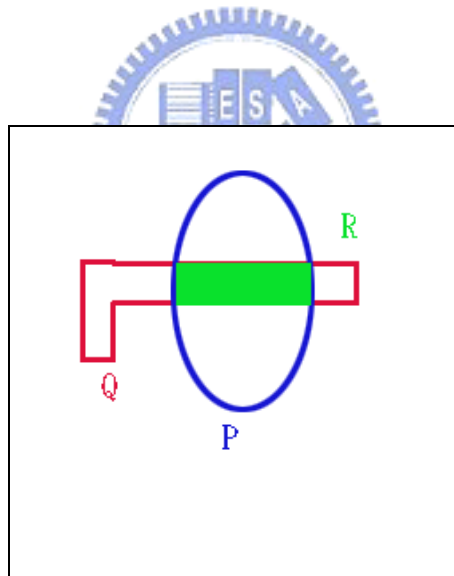


Figure2-5. (a)

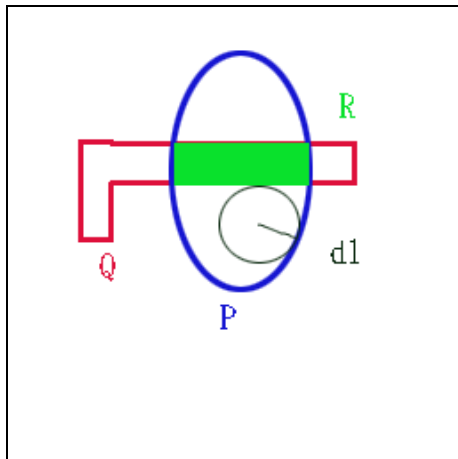


Figure2-5. (b)

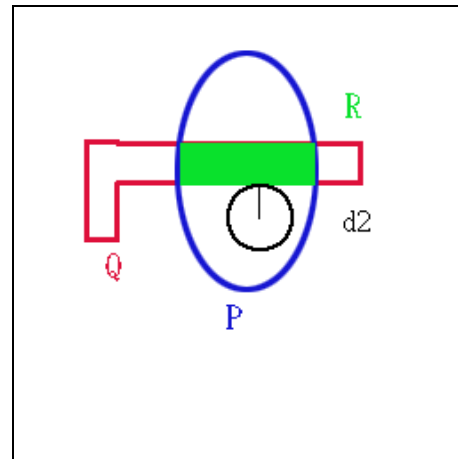


Figure2-5. (c)

In Figure2-5. (a) the ellipse presents P , the horizontal shape presents Q , and the coloration presents $P \cap Q$; (b) shows d_1 ; (c) shows d_2

After we find the geodesic distance functions d_1 and d_2 , we define another distance function d by

$$d = \frac{d_1}{d_1 + d_2} \quad (2.19)$$

Suppose that we choose a fixed distance $d = \lambda$ for instance that means $\lambda = \frac{d_1}{d_1 + d_2}$, then we can get the ratio $\frac{d_2}{d_1} = \frac{1 - \lambda}{\lambda}$. In this case the ratio $\frac{d_1}{d_2}$ forms a constant, the boundaries of the interpolation sets are all arranging to lines, as shown in the Figure2-6 (f).

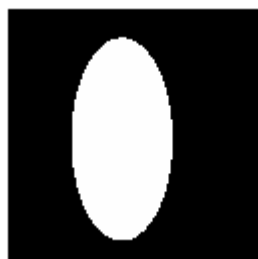


Figure2-6 (a) source P

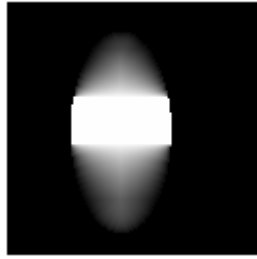


Figure2-6 (b) target Q



Figure2-6 (c) d in source P

Figure2-6 (d) d in target Q

Figure2-6. This example is morphing from P to Q . (a) the source image P ; (b) the target image Q ; (c) shows $d = \frac{d_1}{d_1 + d_2}$ in P , (d) shows $d = \frac{d_1}{d_1 + d_2}$ in Q

For $\lambda = 1$ is a special case which is skeleton by zone of influence between P^c and Q . Then we define an interpolation function int_Q^P as :

- ★ 0 on P (the start image) and 1 on Q (the target image)
- ★ all distance functions $d = \frac{d_1}{d_1 + d_2}$ are defined in the space $P/(P \cap Q)$
- ★ define $+\infty$ on P^c , where P^c is the complement of P

The interpolation sets between P and Q is obtained by a simple threshold:

$$\text{Thr}_k(\text{int}_Q^P) = \{x \mid \text{int}_Q^P \geq k\} \quad (2.20)$$

- ★ 0 on P and 1 on Q

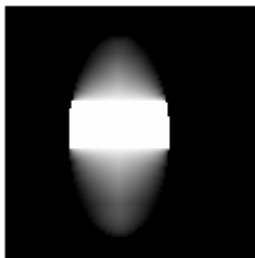


Figure2-7 (a) d in source P Figure2-7 (b) $Thr_{0.5}(\text{int}_Q^P = d)$ in source P

In the interpolation step, the start image P is transformed to the target image Q . The action can be separated into two small actions. The first step begins with P shrinking to $P \cap Q$ and the second one begins with $P \cap Q$ growing to Q . Those two steps can run at the same time. The interpolation set Z_k at distance k from P and $(1-k)$ from Q can be represented by the union of those two terms:

$$Z_k = Thr_k(\text{int}_{P \cap Q}^P) \cup Thr_{1-k}(\text{int}_{P \cap Q}^Q) \quad (2.21)$$



Figure 2-8 (a) P



Figure 2-8 (b) $Z_{0.25}$



Figure 2-8 (c) $Z_{0.5}$



Figure 2-8 (d) $Z_{0.75}$



Figure 2-8 (e) Q

Figure 2-8.(a) to (e) Show the step by step that P morphing to Q ;

2.3.3 Median Set interpolation (J.Serra, S.Beucher)

Morphological median is one of the useful morphological tools for image interpolation. The basic theory was primarily developed for sets. Median interpolation [19] is applicable to binary, mosaic, graytone and color images.

© Binary image

The overture of this approach is the *influence zones* [17] [18] of sets. Let P_1, P_2, \dots, P_n be disjoint sets, then the influence zones of P_i is the locus of those points which are closer to P_i than any other sets. To cite an instance, if we have two sets P and Q with $Q \subset P$, the influence zone of set Q in set P is

$$IZ_p(Q) = \{x \mid d(x, Q) < d(x, P^c)\}, \quad Q \subset P \quad (2.22)$$

where $d(x, E)$ indicates a geodesic distance between point x and set E . By the way, the influence zone defined by equation(2.22) is also represents a median set between two sets, so we can get

$$M(P, Q) = IZ_p(Q) \quad (2.23)$$

Morphological median of nested sets is defined using dilation and erosion, and it has been proved that the median set satisfies the following equation

$$M(P, Q) = \bigcup_{\forall \lambda} \{(Q \oplus \lambda B) \cap (P \ominus \lambda B)\} \quad (2.24)$$

Where $\oplus \lambda B$ represents a dilation of size λ and $\ominus \lambda B$ represents an erosion of size λ , both with the elementary structuring element B

More generally, we extend the simple case $Q \subset P$ to the partial inclusion of X inside Y with $X \cap Y \neq \emptyset$. Morphological median of non-empty intersection is define as the influence zone of $X \cap Y$ in $X \cup Y$:

$$M(X, Y) = IZ_{(X \cup Y)}(X \cap Y) = \bigcup_{\forall \lambda} \{[(X \cap Y) \oplus \lambda B] \cap [(X \cup Y) \ominus \lambda B]\} , \quad X \cap Y \neq \emptyset$$

(2.25)



Figure2-9(a)

Figure2-9(b)

Figure2-9(c)

Figure2-9 (a) first input set; (b) second input set; (c) Median of X, Y .

Interpolation sequence contains the change of shape of the first set into shape of the second one. It is produced by the iterative generation of new medians:

1. First iteration:

1. $M_{1/2} = M(X, Y)$

2. Second iteration:

1. $M_{1/4} = M(X, M_{1/2})$

2. $M_{3/4} = M(M_{1/2}, Y)$

3. Third iteration:

1. $M_{1/8} = M(X, M_{1/4})$

2. $M_{3/8} = M(M_{1/4}, M_{1/2})$

$$3. M_{5/8} = M(M_{1/2}, M_{3/4})$$

$$4. M_{7/8} = M(M_{3/4}, Y)$$

4. ...and so on...



Figure2-10 (a)



Figure2-10 (b)



Figure2-10 (c)



Figure2-10 (d)

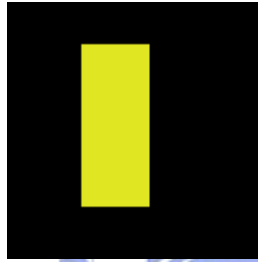


Figure2-10 (e)

Figure2-10 (a) to (e) interpolation sequence

© Gray-level image

One can observe the equation that we just talked about the median element turns out to be an *increasing mapping* of its two operands. Hence it extends to a unique graytone mapping (where the transforms of the cross sections are the cross sections of the transforms). The implementation of this graytone operator is similar to that of the set case. For this reason, we input a pair of initial gray-level images (f, g) and we can formula it in the following way

$$m(f, g) = \sup \{ \forall \lambda : \inf [\mathcal{D}^\lambda (\inf(f, g)), \mathcal{E}^\lambda (\sup(f, g))] \} \quad (2.26)$$

where $\lambda = 1, 2, \dots$ are increasing values, and the dilation and erosion of object X of a given size λ are defined by, respectively:

$$\mathcal{D}^\lambda(X) = \underbrace{(\mathcal{D}_B(\mathcal{D}_B(\dots\mathcal{D}_B(X))))}_{\lambda\text{-times}} \quad (2.27)$$

and

$$\mathcal{E}^\lambda(X) = \underbrace{(\mathcal{E}_B(\mathcal{E}_B(\dots\mathcal{E}_B(X))))}_{\lambda\text{-times}} \quad (2.28)$$

\mathcal{D}^λ stands for dilation of size λ and \mathcal{E}^λ stands for erosion of size λ both performed with non-flat (cylindrical, cone, etc.) structuring element.

We will use the last equation to construction the algorithm of median image generation. So we start from two initial input images (f, g) and define three working images z_0, w_0, m_0 as

$$\begin{aligned} z_0 &= \inf(f, g) \\ w_0 &= \sup(f, g) \\ m_0 &= \inf(f, g) \end{aligned} \quad (2.29)$$

Then we compute the iterated value as follows

$$\begin{aligned} z_i &= D(z_{i-1}) \\ w_i &= E(w_{i-1}) \\ m_i &= \sup(\inf(z_i, w_i), m_{i-1}) \end{aligned} \quad (2.30)$$

where \mathcal{D} and \mathcal{E} are respectively dilation and erosion with the non-flat structuring element. The iteration will execute until idempotence and finally

$$m(f, g) = m_\infty = m_i, \quad m_i = m_{i+1} \quad (2.31)$$

where $m(f, g)$ is a new image of morphological median images of images f and g .

2.3.4 Shape-based interpolation

Shape-based interpolation comes from numerical analysis literature. It was first proposed by J. K. Udapa [20]. Generally, shape-based interpolation algorithms are widespread employed on binary images [21] [22] [23]. These interpolation methods consider shape features extracted from the object sets. We will consider one-to-one object interpolation at the beginning.

Shape-based interpolation converts binary images into distance maps by distance transformation functions such as chamfer or city-block distance template [21]. Another illustration is converting the segmented slice image into gray-level images. These templates are used to efficiently approximate the *shortest distance* between the pixel and the contour of the object.

In the following figure, it is part from an object, we estimate the distance-from-boundary values assigned to pixels. We take b as the interval length. The heavy line indicates the boundary of the object thresholded at 0.

$+(a+b)^{\oplus}$	$+a^{\oplus}$	$-a^{\oplus}$	$-(a+b)^{\oplus}$	$-(a+2*b)^{\oplus}$	$-(a+3*b)^{\oplus}$	$-(a+4*b)^{\oplus}$	$-(a+5*b)^{\oplus}$	$-(a+6*b)^{\oplus}$
$+(a+5*b)^{\oplus}$	$+(a+4*b)^{\oplus}$	$+(a+3*b)^{\oplus}$	$+(a+2*b)^{\oplus}$	$+(a+b)^{\oplus}$	$+a^{\oplus}$	$-a^{\oplus}$	$-(a+b)^{\oplus}$	$-(a+2*b)^{\oplus}$

Table2-5. Show the distance transformation values from the boundary, the heavy line is the boundary

The distance function Raya and Udupa used is a version of the city-block distance in the past. City-block distance can be used to calculate this distance efficiently, but it is a relatively bad approximation to Euclidean distance. For this reason, some improvement can be achieved by using chamfer distance. That is obviously better performance of shape-based interpolation by using a distance function that approximates Euclidean distance more closely. In interpolation step, we can calculate these by two consecutive chamfering processes for the inside and then for the outside. One method of interpolating between slices is linear interpolation.

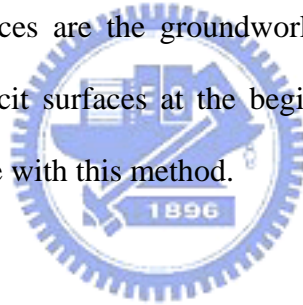


Chapter 3

Level Set Method

In Chapter 2, we have discussed the theory and interpolation method of mathematical morphology completely. In this chapter, we will continue deliberating the other morphing method – level set and discussing its theory and the application by the same token.

The level set method was first introduced by Stanley Osher and James A. Sethian in 1987 [24] [25] [26]. It has been proven to be a robust method to do morphing. The implicit surfaces are the groundwork of level set method. So it is essential to present the implicit surfaces at the beginning. Besides, there are some brief introductions to associate with this method.



3.1 Implicit functions

In n-dimensional space. There are two ways to define functions, *implicitly* [26] [27] [28] and *explicitly*. Most of the equations we have dealt with have been explicit equations that are forms of $y = f(x_1, x_2, \dots, x_n)$. In the meantime, the implicit definition of y as a function of x_1 to x_n is the same as a relation which expressed indirectly by an equation such as $f(x_1, \dots, x_n, y) = k$, where k is a function or a constant.

Definition

An equation of the form $f(x, y) = 0$ (where f is smooth) often defines a

curve in the plane. When this is the case, we say that the curve is defined *implicitly* (as opposed to explicitly) because we may not be able to solve for y in terms of x . Nevertheless, assuming that a local solution exists, the method of implicit differentiation often allows us to solve for implicit differentiation $y' = \frac{dy}{dx}$ as a function of x and y at any point (x, y) along this curve.

In general, an implicit function defined in 2-dimension space divides the space into two parts, *inside region* and *outside region*, and the border between the inside and the outside is called the *interface* or *isocontour*. In n-dimensional space, the inside and outside regions are n-dimensional object, while the interface is less than one-dimension. By the way, interface curves are usually limited to be closed.

The following figure is an example of the implicit function.

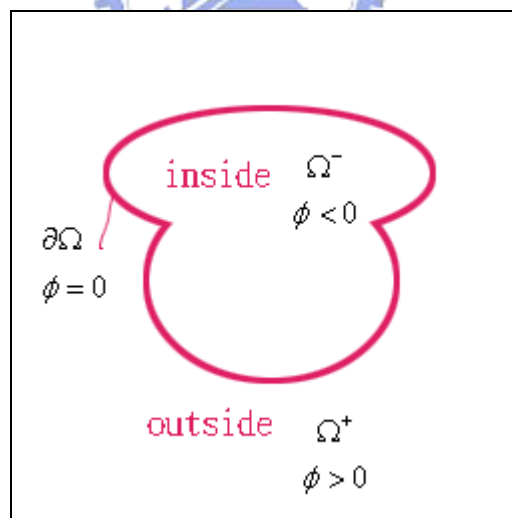


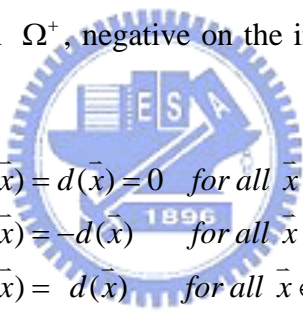
Figure3-1. The implicit function ϕ to identify inside region and outside

3.2 Signed distance function

In general, a *distance function* [26] [29] [30] $d(\bar{x})$ is defined as $d(\bar{x}) = \min(|\bar{x} - \bar{x}_l|)$ for all $\bar{x}_l \in \partial\Omega$ implies that $d(\bar{x}) = 0$ on the boundary and that there are several ways to calculate distance $d(\bar{x})$ such as Euclidean distance, chessboard distance....., it can be referred to chapter 2.2 Distance transform.

A *signed distance function* is an implicit function ϕ with $|\phi(\bar{x})| = d(\bar{x})$ for all \bar{x} and an extra condition $|\nabla\phi(\bar{x})| = 1$ is imposed.

To go a step further, a signed distance function defines a signed distance to be positive on the exterior region Ω^+ , negative on the interior region Ω^- , and zero on the boundary $\partial\Omega$.



$$\begin{cases} \phi(\bar{x}) = d(\bar{x}) = 0 & \text{for all } \bar{x} \in \partial\Omega \\ \phi(\bar{x}) = -d(\bar{x}) & \text{for all } \bar{x} \in \Omega^- \\ \phi(\bar{x}) = d(\bar{x}) & \text{for all } \bar{x} \in \Omega^+ \end{cases} \quad (3.1)$$

Also, the opposite definition to negative on the exterior region, positive on the interior region, and zero on the boundary is allowed.

3.3 Level set Formula

After that, level set method using implicit surfaces. The original idea of level set [26] [31] can be traced back to Hamilton - Jacobi approach to numerical solutions of a time-dependent equation for moving implicit surfaces. In other words, level set

method gives an interface Γ in R^n of codimension one, bounding a region Ω , we want to analyze and compute its subsequent motion under a velocity field v as the next position. More clearly, the idea behind level set methods is: *instead of evolving a curve in a two dimensional plane, which requires additional parameters of the curve, evolve a 2D function (surface) in 3D*. And this is a much easier problem.

For analyzing the level set method, we should introduce various basically know-how at first.

◆ The interface boundary $\Gamma(t)$ is defined by

$$\Gamma(t) = \{\bar{x} \mid \phi(\bar{x}, t) = 0\} \quad (3.2)$$

The region $\Omega(t)$ is confined of $\Gamma(t)$. It is always to define isocontour Ω as the boundary, the interior region Ω^- is defined by $\{\bar{x} \mid \phi(\bar{x}, t) < 0\}$, and the exterior region Ω^+ is defined by $\{\bar{x} \mid \phi(\bar{x}, t) > 0\}$

◆ The gradient of the implicit function is defined by

$$\nabla \phi = \left(\frac{\partial \phi}{\partial x}, \frac{\partial \phi}{\partial y}, \frac{\partial \phi}{\partial z} \right) \quad (3.3)$$

in three dimension.

◆ The unit (outward) normal \bar{N} to $\Gamma(t)$ is defined by

$$\bar{N} = \frac{\nabla \phi}{|\nabla \phi|} \quad (3.4)$$

◆ The mean curvature κ of the interface $\Gamma(t)$ is defined by divergence of the normal \bar{N}

$$\kappa = \nabla \cdot \bar{N} = \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \quad (3.5)$$

so that $\kappa > 0$ for convex region

$\kappa < 0$ for concave region

$\kappa = 0$ for a plane

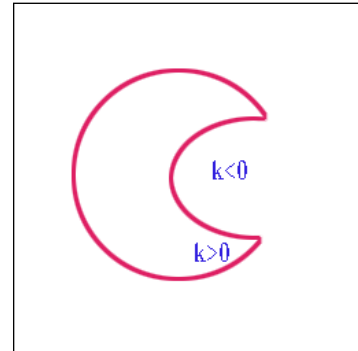


Figure3-2. Show the relation of signed mean curvature κ and the appearance

◆ The one dimension Heaviside function is defined by

$$H_e(\phi) = \begin{cases} 1 & \text{if } \phi > 0 \\ 0 & \text{if } \phi \leq 0 \end{cases} \quad (3.6)$$

◆ The characteristic function χ^- of the interior region $\Omega(t)^-$ and χ^+ of the exterior region $\Omega(t)^+$ is

$$\begin{aligned} \chi^-(\bar{x}) &= 1 - H_e(\phi(\bar{x})) \\ \chi^+(\bar{x}) &= H_e(\phi(\bar{x})) \end{aligned} \quad (3.7)$$

where H means the Heaviside function just described above.

◆ The Dirac delta function is the directional derivative of the Heaviside function H_e in the normal direction \bar{N}

$$\delta(\bar{x}) = \nabla H(\phi(\bar{x})) \cdot \bar{N} \quad (3.8)$$

where the distribution is nonzero only on the interface $\partial\Omega$ that $\phi = 0$.

◆ The Hamilton-Jacobi equation

$$H(q_1, \dots, q_n; \frac{\partial F}{\partial q_1}, \dots, \frac{\partial F}{\partial q_n}; t) + \frac{\partial F}{\partial t} = 0 \quad (3.9)$$

or we can express another formula

$$H(\nabla \phi) + \phi_t = 0 \quad (3.10)$$

where F is called Hamilton principle function.

After introducing the all above notations, we will carry on the main equation of level set. In order to avoid problems with instabilities, deformation of surface elements and complicated surgical procedures for topological repair of interfaces, the implicit function ϕ both represents the interface and the moving interface.

Level set method provides the mathematical and numerical mechanisms for computing surface (image) deformation as time-varying values of ϕ by solving a partial differential equation on the 3D grid. To describing the movement of the surface, we denote the point of $\bar{x} \in \Gamma$ lies on the surface and it deforms as $\frac{d\bar{x}}{dt}$. In this case, there are generally two options for representing such surface movement implicitly, those are static form and dynamic form.

★ Static :

The static formulation is represented as $F(x) = k(t)$, the F function is single while the k is varied.

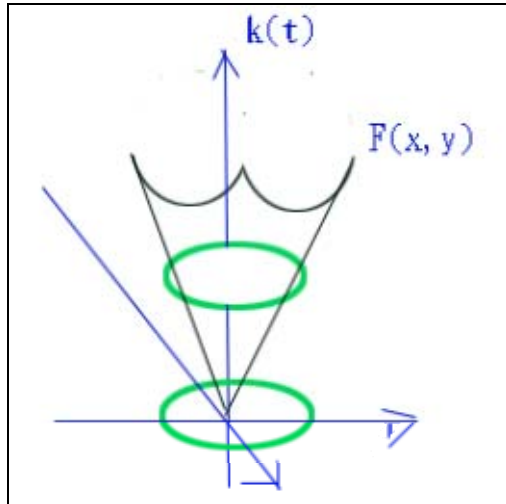


Figure3-3. The static formulation $F(x, y) = k(t)$

A single, static function $\phi(x_t)$ denotes a level set formula corresponding to the surface as different time t . That is at time t

$$\begin{aligned}
 & \phi(\bar{x}_t) = k(t) \\
 \Rightarrow & \phi'(\bar{x}_t) = k'(t) \\
 \Rightarrow & \frac{d\phi(\bar{x}_t)}{dt} = \frac{dk(t)}{dt} \\
 \Rightarrow & \nabla \phi(\bar{x}_t) \cdot \frac{\partial \bar{x}_t}{\partial t} = \frac{dk(t)}{dt}
 \end{aligned} \tag{3.11}$$

This equation can be solved efficiently by starting with a single surface using the fast marching method. Nevertheless, the static level set method has a unfavorable problem : it's motion strictly inward or outward. In other words, the surface can not pass back over itself overtime, and we can observe this phenomenon in the next figure.

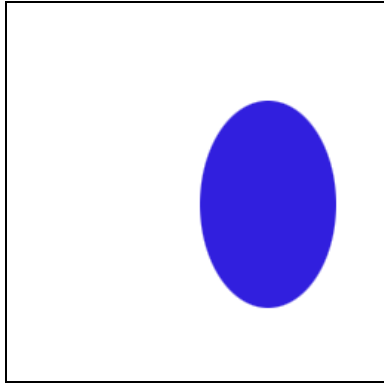


Figure3-4 (a)

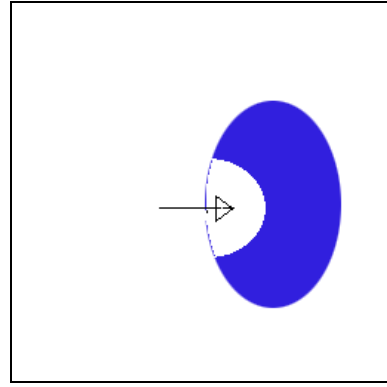


Figure3-4. (b)

Figure 3-4(a) the original shape and (b) the change of the shape and notice that the motion can not pass back

★ Dynamic

On the other side, the dynamic formulation is indicated that $F(x,t) = k$, the $F(x,t)$ is evolved while the k is fixed.

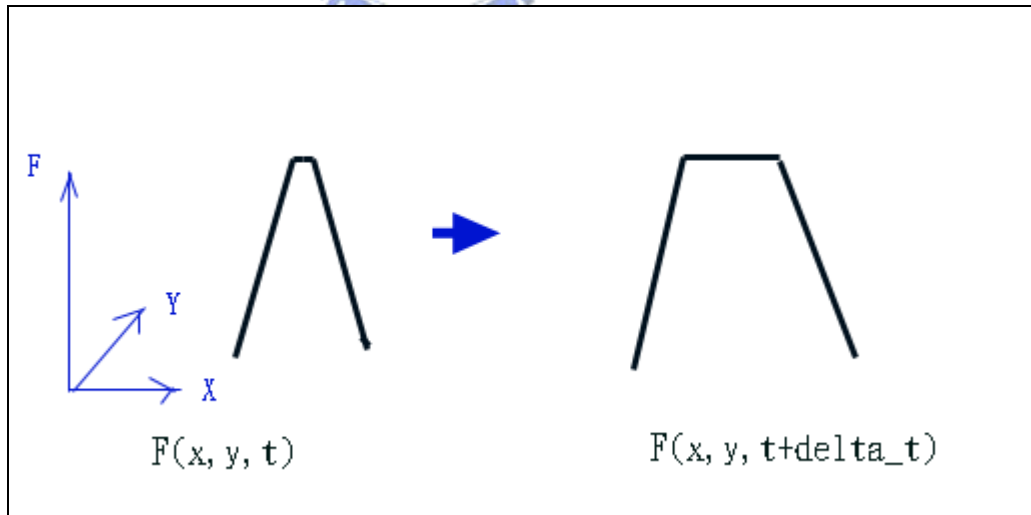


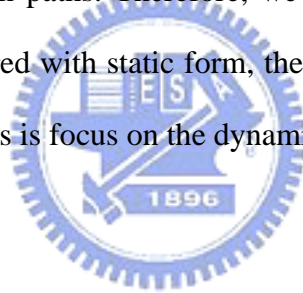
Figure. 3-5 The dynamic formulation of $F(x, y, t) = k$

Then evolving $\phi(\bar{x}, t)$ with embedding time t automatically gives the

evolution. The term \bar{x} remains on the k level set of ϕ as it moves, and k is a constant. The behavior of ϕ is obtained by setting the total derivative of $\phi(\bar{x}, t) = k$ to zero.

$$\begin{aligned}
& \phi(x_t, t) = k \\
\Rightarrow \frac{d\phi}{dt} &= \frac{1}{dt} \left[\left(\frac{\partial \phi}{\partial x} \right)_t dx + \left(\frac{\partial \phi}{\partial t} \right)_x dt \right] = 0 = \frac{dk}{dt} \\
\Rightarrow \nabla \phi \cdot \frac{dx}{dt} + \frac{\partial \phi}{\partial t} &= 0 \\
\Rightarrow \frac{\partial \phi}{\partial t} &= -\nabla \phi \cdot \frac{dx}{dt}
\end{aligned} \tag{3.12}$$

This dynamic form provides arbitrarily motions which move forward and backward and cross back over their own paths. Therefore, we using front tracking scheme to solving this equation. Compared with static form, the dynamic form is more flexible, so the remainder of this discuss is focus on the dynamic case.



There is a widespread use that choosing the simple convection (or advection) equation

$$\phi_t + \bar{V} \cdot \nabla \phi = 0 \tag{3.13}$$

to define the evolution of the implicit function where the t subscript denotes a temporal partial derivative in the time variable t and the \bar{V} is a speed function of the velocity of each point on the implicit surface. The level set equation also can rewrite as a well-known form

$$\begin{aligned}
& \phi_t + V_n |\nabla \phi| = 0 \\
\Leftrightarrow \phi_t + V_n (\bar{N} \cdot \nabla \phi) &= 0 \\
\Leftrightarrow \phi_t + \bar{V} \cdot \nabla \phi &= 0
\end{aligned} \tag{3.14}$$

where V_n is a constant component of velocity in the normal direction, it also wise known as the normal velocity. Let us return to our main subject, the level set equation is a kind of Hamilton-Jacobi equation where $H(\nabla\phi) = \vec{V} \cdot \nabla\phi$ or $H(\nabla\phi) = Vn|\nabla\phi|$.

To avoid oscillation, we can use the characteristic function to identify the shapes and boundaries. We calculate the volume integral of a function f over the interior region Ω^- at first :

$$\int_{\Omega} f(\bar{x}) \chi^-(\bar{x}) d\bar{x} \quad (3.15)$$

where the region of integral is all of Ω , but the χ^- prunes out the exterior region Ω^+ automatically. We rewrite the volume integral of a function f over the interior region Ω^- using the Heaviside function:

$$\int_{\Omega} f(\bar{x}) (1-H(\phi(\bar{x}))) d\bar{x} \quad (3.16)$$

The δ prunes out every thing except $\partial\Omega$ automatically. In according with the volume integral, the surface integral of the function f over the boundary $\partial\Omega$ is defined as :

$$\int_{\Omega} f(\bar{x}) \delta(\bar{x}) d\bar{x} \quad (3.17)$$

3.4 Deforming Method

In this section, we will talk about some image morphing using level set method.

3.4.1 Image blending

Ross T. Whitaker [32] [33] was proposed a method about the post-processing after image morphing. Once one image has been morphed, it needs a low-level post-processing to calculating the detail and color that are not captured by the coordinate transforms.

In advance, it is said that the image morphing need to input two images- source and target. To produce intermediums of morphing sequence, it was always use two input images in traditional. In the mean while, a convenient skill is proposed that using the new manufactured middle images as the materials in place of the two original input images.

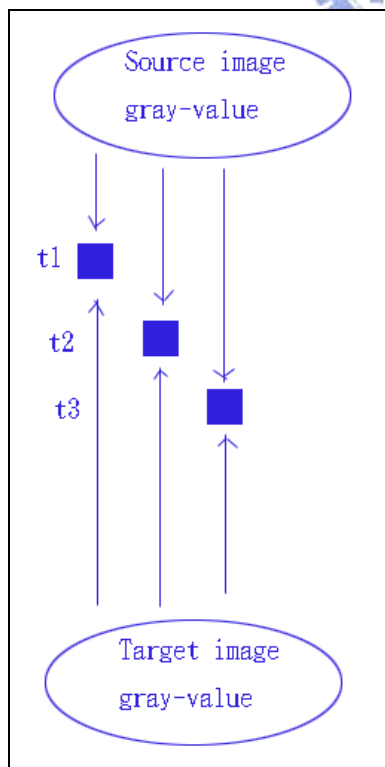


Figure3-6. (a)

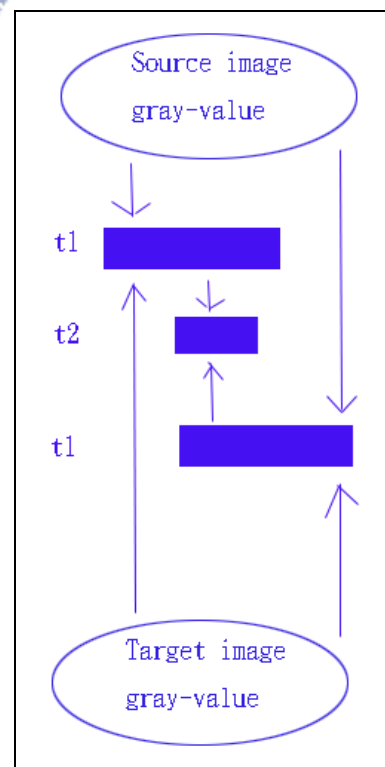


Figure3-6. (b)

Figure3-6 (a) (b) show the in-between images manufacture with difference kinds stuff

The two input images are regarded as two continuous functions $T: \mathfrak{R}^2 \rightarrow \mathfrak{R}$, the domain \mathfrak{R}^2 denotes the image domain and the range \mathfrak{R} denotes the set of grayscale values of the image. Let $F(x,y)$ and $G(x,y)$ represent the source and target function respectively, and $(x,y) \in \mathfrak{R}^2$ denotes the position of image. On the side, let an extra parameter α ($0 \leq \alpha \leq 1$) express the indexes of image sequence. Usually, $\alpha = 0$ is the source image and the larger α the closer to target image.

For reasons mentioned above, the image blending path can be described using $f(x,y,t)$ and $g(x,y,t)$, where $f(x,y,t=0) = F(x,y)$ the set of source image and $g(x,y,t=0) = G(x,y)$ the set of target image. A blend function $b(x,y,\alpha)$ can be generated by $f(x,y,t)$ forward in time and $g(x,y,t)$ backward in time. $f(x,y,t)$ and $g(x,y,t)$ will contact in a single α .

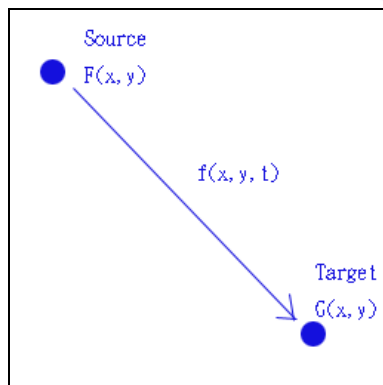


Figure3-7. (a)

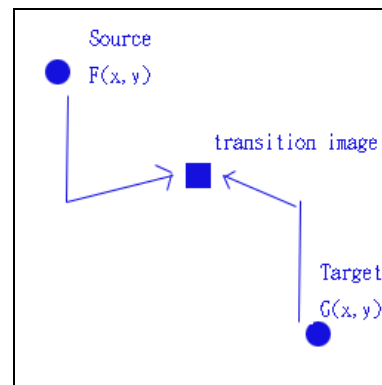


Figure3-7. (b)

Figure3-7. (a) show that it start from source and move toward to target in one way; (b) is the reform that it move start source and target, then it encounter in one position.

Rather than taking the image as a pixel by pixel, it can take the image as a

collection of isocontours. The shape of each isocontour (level set) is not only simply a local property of an image, but also depends on relations between pixels on images.

The k level set of f is a set of points identified by the Heaviside function, and defined as

$$L_k = \left\{ \begin{pmatrix} x \\ y \end{pmatrix} \mid f(x, y) = k \right\} \quad (3.18)$$

This binary function can be defined the same as above that 0 for inside region and 1 for outside region, also, if we were changing the representation absolutely opposite does not affect the result.

Because the image is treated as a collection of isocontours, thus it can calculate ϕ_i for each k independently, and the ϕ_i can be defined as

$$f_i = \frac{\partial f}{\partial t} = \frac{g - f}{|g - f|} |\nabla f| = \text{sgn}(g - f) |\nabla f| \quad (3.19)$$

where $\text{sgn}(g - f)$ defined as

$$\text{sgn}(g - f) = \begin{cases} -1 & (g - f) < 0 \\ 0 & (g - f) = 0 \\ 1 & (g - f) > 0 \end{cases} \quad (3.20)$$

and $|\nabla f|$ is the gradient magnitude.

3.4.2 Radial Basis function (RBF)

The Radial basis functions have become popular for implicit surface modeling [34], scattered point interpolation [35] and even morphing [36] [37]. In this section, we will focus on the part of morphing.

Radial basis functions (RBF) have been used since 1970 for different ways. In the early stage, the RBF are used for cartographic data. Therefore, the RBF have been used to approximate solutions to partial differential equations lately, but this area is still relatively unexplored.

A radial basis function (RBF) $\mathfrak{I}:\mathfrak{R} \rightarrow \mathfrak{R}$ defines a radially symmetric function. This function takes at a given point depends on the distance between that point and the center point. The basic function is define as

$$\mathfrak{I}(\|x - x_c\|) \tag{3.21}$$

where the $\|\square\|$ denotes Euclidean vector norm, and the x_c means the center point.

Usually, we define RBF $\mathfrak{I}(r) = r^2 \log r^2$ in 2-D, and it results in a thin plate spline that minimizes the bending energy of the interpolant. For the same definition, it defines $\mathfrak{I}(r) = r^3$ in 3-D, and it results in the isosurface that minimizes the bending energy of the field but not necessarily.

RBF method is highly flexible since it uses points which are scattered instead of requiring a grid. The only geometric property used is the pairwise distance between points in the domain. The RBFs surface collected by center points x_c can be defined as

$$\phi(X) = P(X) + \sum_{i=1}^n a_i \mathfrak{I}(\|x_c - X\|) \tag{3.22}$$

where $P(X) = n \cdot X + d$ is a plane equation of unknown parameters and a_i are unknown weighted numbers of the radial basis function.

Return to the main topic, the level set propagation performs on the RBF surface by moving each RBF dipole based on its speed function along its surface normal direction, then solving to get new surface. In the case of morphing, V is given as the distance transform to some target, and ∇V is the gradient of the distance transform. Assume we only consider motion by mean curvature, because of the interface moves in the normal direction with a velocity proportional to its mean curvature. According to propagating by mean curvature, the speed function has a special form, the same as κ , the divergence of normal

$$\kappa = \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \quad (3.23)$$

To summarize the above discussion, we can find it can do morphing by only control the speed function. Besides, this method enables level set algorithms to avoid a costly voxel representation and efficient procedural implicit surfaces instead.

Chapter 4

Experimentation

So far, we have seen how to do morphing by morphological interpolations [10] [20] [22] [23] and level set methods [32] [33] [36] [37]. In this chapter, we will continue to implement and compare them. We will divide our experiments into three parts. In the first part, we will present morphological interpolations and blending – a morphing by level set methods. In the second part, we will show their applications. And in the last part, we will show the morphing process in one image.



4.1 Mathematical morphology interpolations

This section, we will examine the implementation of those methods in detail. For fair comparison, we use the same source image and target image for all methods.

4.1.1 Shape-based morphology interpolation

In chapter 2, we have talked about several ways to carry out morphological interpolations. Here, we decide to adopt a simple way to achieve our goal using another kind of “shaped-based interpolation”. This idea is proposed by Adrian G. Bors and described in [22] [23]. Let us define P as the source image and Q as the target image in the beginning. The two input images can be grayscale or color images. In general, it has an intersection area, denoted as $R = P \cap Q$. We just begin with eroding P and taking union with R until the area equals R . While on the other hand, we begin with dilating R and taking intersection with Q until the area equals Q . The

above steps can be executed simultaneously, and the last action is to combine the results produced by above two steps. In case of the intersection area R is empty, that is $R = P \cap Q = \emptyset$, we try to calculate the minimum distance between source object and target object. Then we define a small area in target object which is closest to source object. Therefore, we can rewrite our algorithm as:

Begin

Set intersection $R = P \cap Q$

if ($R \neq \emptyset$); // the intersection is not empty

{

do $P_{i+1} = \mathcal{E}_B(P_i) \cup R$ until equal R ;

$Q_0 = R$

$Q_1 = \mathcal{D}(Q_0) \cap Q$

do $Q_{i+1} = \mathcal{D}_B(Q_i) \cap Q$ until equal Q ;

combine above results

}

else {

Set the area R in target object closest to source object

do $P_{i+1} = \mathcal{E}_B(P_i) \cup R$ until equal R ;

$Q_0 = R$

$Q_1 = \mathcal{D}(R) \cap Q$

do $Q_{i+1} = \mathcal{D}_B(Q_i) \cap Q$ until equal Q ;

combine above results

}

End Begin



Here, the image morphing process will be shown in the following. The image sizes are all 128*128.

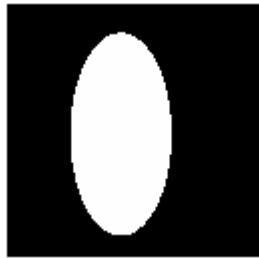


Figure4-1 (a) original source image;

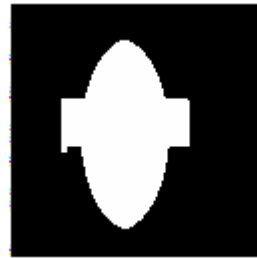


Figure4-1 (b) 20% morphing;

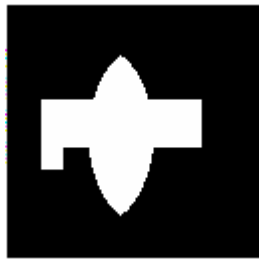


Figure4-1 (c) 40% morphing;

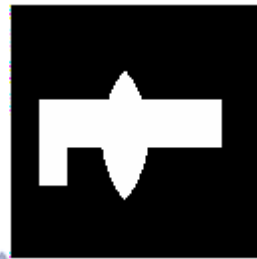


Figure4-1 (d) 60% morphing;

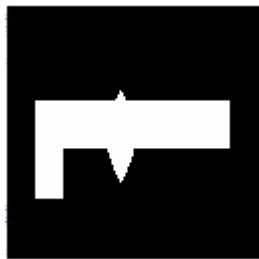


Figure4-1 (e) 80% morphing;



Figure4-1 (f) original target image.

Figure4-1 (a) to (f) show the image morphing processing by the morphological shape-based interpolation

4.1.2 Hausdorff distance interpolation [14] [15]

In this section, we will present some morphed results by using Hausdorff distance interpolation. The Hausdorff distance interpolation was discussed detailed in the chapter 2. Here, we just present the resulting images. The image sizes are all

128*128.

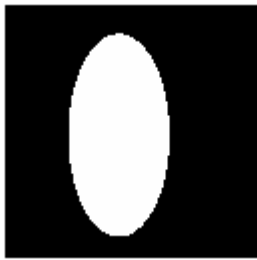


Figure4-2 (a) original source image;



Figure4-2 (b) 20% morphing;



Figure4-2 (c) 40% morphing;



Figure4-2 (d) 60% morphing;



Figure4-2 (e) 80% morphing;



Figure4-2 (f) original target image.

Figure4-2 (a) to (f) show the image morphing processing by the morphological Hausdorff distance interpolation

We have talked about that the morphed images by Hausdorff distance interpolation are larger than original images, we can aware this phenomenon by observing the width of the objects.

4.1.3 Distanced-based interpolation [16].

The distance-based interpolation also called “function interpolation”. The particular morphing processing is stated in the chapter 2. Then, we will show the morphed images in the Figure 4-3. The image sizes are all 128*128, too.



Figure4-3 (a) original source image;



Figure4-3 (b) 20% morphing;



Figure4-3 (c) 40% morphing;

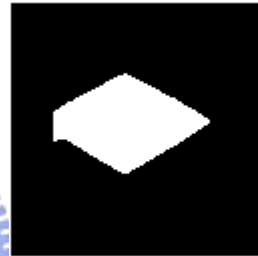


Figure4-3 (d) 60% morphing;



Figure4-3 (e) 80% morphing;



Figure4-3 (f) original target image.

Figure4-3 (a) to (f) show the image morphing processing by the distance-based interpolation

In our experiments, the distance-based interpolation leads to a not bad result. But, it uses more time to calculate the distance function defined in morphed images.

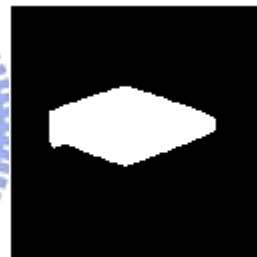
4.1.4 Median interpolation

The median interpolation is based on the influence zone. In the same way, the method of median interpolation is described in detail in chapter 2. We put the results by the same input source image and target image. The image sizes are all 128*128, too.



Figure4-4 (a) original source image;

(b) $M_{1/4} = M(\text{source}, M_{1/2})$



(c) $M_{1/2} = M(\text{source}, \text{target})$

(d) $M_{3/4} = M(M_{1/2}, \text{target})$



(e) $M_{7/8} = M(M_{3/4}, \text{target})$

(f) original target image.

Figure4-4 (a) to (f) show the image morphing processing by the median interpolation; (c) shows the first step, (b) (d) show the second step, (e) shows the third step

4.1.5 Discussion

In our experiments, we find the distance-based interpolation has good outcome, but needs more time to calculating the distance function. The results by distance-based interpolation are similar to the results by median interpolation, and they are better than the shape-based interpolation and Haudroff distance interpolation.

Methods Kinds	Shape-based interpolation	Haudroff interpolation	Distance-based interpolation	Median interpolation
Results (smooth)	3	2	1	1
Speed (fast)	1	2	4	3

Table4-1 Shows the time consuming and the results. The size of test images is 128X128



4.2 Level set Methods

4.2.1 Blending

For level set methods, we adopt the blending method proposed by Ross T. Whitaker [32] [33], the method just stated in chapter3. Now, the middle images are produced by the following equation

$$\begin{aligned} u_i^{k+1} &= u_i^k + \Delta u_i^k \Delta t^k \\ v_i^{k+1} &= v_i^k + \Delta v_i^k \Delta t^k \end{aligned} \quad (4.1)$$

This equation is simplified to one dimension for easier understanding. We use u to present the source information while it is defined as function f in chapter3. By the same token, v presents the target information while it is defined as function g in chapter3. Furthermore, Δu_i^k indicates $\frac{\partial f}{\partial t}$, the partial differential, and Δv_i^k

indicates $\frac{\partial g}{\partial t}$. Here, we use $\text{sgn}(g - f)$ to point out the moving direction. It should also refer to an extra scheme, the upwind method. The upwind method was proposed by Osher and Sethian, too. On Cartesian grid, it uses finite difference to approximate derivatives. They are divided into two types – forward (upwind) and backward. A first-order accurate forward difference is defined by

$$D^+ \phi = \frac{\partial \phi}{\partial x} \approx \frac{\phi_{i+1} - \phi_i}{\Delta x} \quad (4.2)$$

A first-order accurate backward difference is defined by

$$D^- \phi = \frac{\partial \phi}{\partial x} \approx \frac{\phi_i - \phi_{i-1}}{\Delta x} \quad (4.3)$$

Here, if we compute from source to target, then ϕ_i represents f (or u_i^k). On the contrary, ϕ_i represents g (or v_i^k). It is popular to use upwind method. On the side, we may assume without a loss of generality that $\Delta x = 1$, because the pixel distance can be fixed in Δt^k . Note that it needs resample the results whenever the blending program executed, since we usually restricted our parameters in 0 to 1. Ross T. Whitaker also mentioned that we can also use intersection area as morphological interpolation. But, we find the result is not good in our experiment.

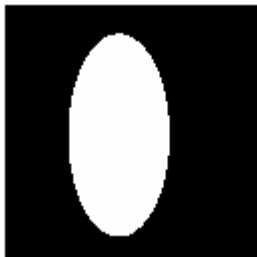


Figure4-5. (a) original source image



Figure4-5. (b) 20% morphing;



Figure4-5. (c) 40% morphing;



Figure4-5. (d) 60% morphing;



Figure4-5. (e) 60% morphing;



Figure4-5. (f) original target image

Figure4-5 (a) to (f) show the image morphing processing by blending

All these pictures make it obvious that middle images are blurred in right side while that are plain in the left side. According to our experiment, the blending case spends more time in producing one image, because blending needs to calculate numerical analysis. Although the mathematical morphological interpolation cases spend fewer times to produce an image in averaging, sometimes it spends more memory to recode all images. Analyzing those results, we can't conclude which one is better immediately, but we can say that which one may be suitable for some special cases. For example, blending may be more suitable for morphing nature scenes, like the variety of candle, cloud, or snow,...,ect. Mathematical morphological interpolation may be more suitable for medical applications, like face-lift, tooth reconstruction,...,ect.

4.3 Enhanced morphing

According to our experiments, the distance-based interpolation is good enough in the results. Hence, we attempt to morphing two images by strengthened level set methods. Because we can observe the blending brings about smear phenomenon, and the other character –the distance-based interpolation in this paper may well keep off smear form. For this reason, we try to use the distance-based interpolation in-between images to intensify the blending morphed image. Since we have considering the blending images as principal part and median interpolation images as a minor part. The intensifying way we use takes the grayscale of the distance-based interpolation morphed image to substitute the grayscale of blending morphed image if there is an object. For preserving the characteristic of blending images that the edges looks flowing appearance, so we use morphological operation – erosion first to keep the edge.



Algorithm:

Begin

do level set method - blending

do morphological interpolation- the distance-based interpolation

erode the blending morphed image to keep the edges

overlap relatively morphed image using two methods

if (the intersection is not empty)

we use the grayscale of median interpolation to substitute blending image

grayscale

else

we keep the blending images

End Begin

Figure4-6 (a) is the original source image, Figure4-6 (b.1) and Figure4-6 (b.2) are about 20% morphing, Figure4-6 (c.1) and Figure4-6 (c.2) are about 40% morphing, Figure4-6 (d.1) and Figure4-6 (d.2) are about 60% morphing, Figure4-6 (e.1) and Figure4-6 (e.2) are about 80% morphing, and Figure4-6 (f) is the original target image. The left middle image is using level set method and the right ones are using enhanced level set method. The image sizes are all 128*128, too.

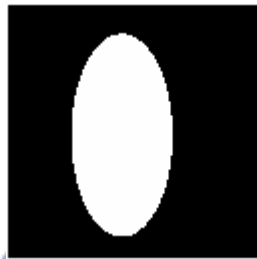


Figure4-6. (a) the original image



Figure4-6. (b.1) 20% morphing



Figure4-6 (b.2) 20% morphing



Figure4-6 (c.1) 40% morphing



Figure4-6 (c.2) 40% morphing



Figure4-6. (d.1) 60% morphing



Figure4-6. (d.2) 60% morphing



Figure4-6. (e.1) 80% morphing



Figure4-6. (e.2) 80% morphing



Figure4-6. (f) the target image

Figure4.6. (a) to (f) show the image morphing processing; the images in the left column are the level set method and in the right column are enhanced level set method

In those pictures, we can find the enhanced blending morphed images are much brighter than the original blending images. In the meantime, we can observe that the morphed images by enhanced blending are not so blur.

4.4 Combination of the two methods

For advanced comparison, we put the morphing process in one image. In this image, we put more than two objects in it. Then, we slice the image according to the objects. The following figure will show it.



Figure4-7. An example for the original input image

the original above image will be sliced into the below three images.



Figure4-8. (a)

Figure4-8. (b)

Figure4-8. (c)

Figure4-8. (a) (b) (c) are the decompose by the Figure4-7

After we sliced the image, we can choose one method which is fitted the object property manually. By following the rules we have mentioned about, we can classify the first and second images (cloud) to blending and third image to shape-based

interpolation. The left up one is the source, and the right down one is target. In this example, we choose two clouds as one image to morphing with blending, and the tank morphing with shape-based interpolation.

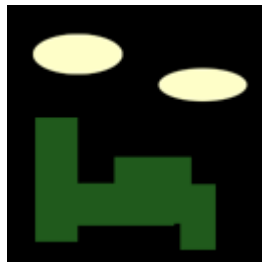


Figure4-9. (a)



Figure4-9. (b)



Figure4-9. (c)



Figure4-9. (d)



Figure4-9. (e)



Figure4-9. (f)

Figure4-9 (a) to (f) presents the image processing combined with two methods

The choice is not unique indeed. We can choose the left cloud as one object, the right cloud as the other object, needless to say, the tank as the third object. The different choices make different results.

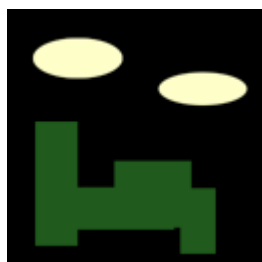


Figure4-10. (a)

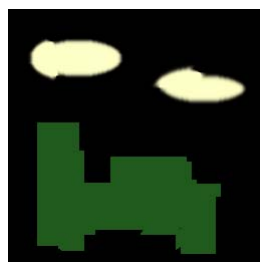


Figure4-10. (b)



Figure4-10. (c)



Figure4-10. (d)



Figure4-10. (e)



Figure4-10. (f)

Figure4-6 (a) to (f) presents the image processing combined with two methods by another choice



Chapter 5

Conclusions

5.1 Discussion

In this paper, our purpose is to compare the methods of morphological interpolations and level set methods - blending in morphing. In morphological interpolations, we have try to carry out shape-based interpolation, Hausdroff distance interpolation, distance-based interpolation, and median interpolation. We observe the in-between images in morphing process. The purpose of morphing is to construct a natural, smooth image sequence. To achieve this goal, we should confirm which effect we want at first. As described above, the results show the two methods cause completely different outcomes.



Mathematical interpolations are easy to understand and implement. A morphological morphing algorithm considers an image as a set, and the morphing transformation depends on operating two sets into the other one. In shape-based interpolation, it just uses two basic algebra operations – dilation and erosion, the in-between images can be obtained by idempotency of the morphed sets from the two side extremities. In Hausdroff distance interpolation, it just achieved by an operation – dilation. In distance-based interpolation, the morphed images are controlled by the distance functions. In median interpolation, the in-between images are just composed by influence zones.

On the contrary, level set methods are more complex. Level set methods based on the Partial Differential Equation. It is a novel approach for image morphing. Here,

we use image blending to achieve image morphing. Image blending is a low-level process to construct a set of image transition. The most important thing about using level set methods to do morphing is its ability of avoiding ghosting. The other additional advantage is that it is not restricted on empty intersection. By the way, in our experiment, we can find it spends more times than morphological shape-based interpolation does.

5.2 Future work

In this paper, we describe the detail of theory and application of mathematical morphology in chapter 2. Then, we discuss the detail of theory and application of level set methods in chapter 3. And, we present all the experimental results in chapter 4. But it still has several aspect should be enhanced.

Generally speaking, mathematical morphological interpolations cut the edge sharp. Besides, for empty intersection, except for Hausdroff distance interpolation is practicable, we just set the area by calculating the minimized distance between two original images.

In blending, unlike mathematical morphology, even though the source image and the target image are two quite different colors, the grayscale of pixels in morphed color image show in a more true form. Nevertheless, the whole morphed image stares blurred. Here, we try to clear the morphed image by using median interpolation morphed image to intensify the correspondent location pixels in blending morphed image in chapter 4 section 2. The impression on the in-between image are tolerable but could be advanced.

Collecting those disadvantages above all, we can do enhancement in two ways:

◆ For mathematical morphology:

Find some extra skill to make sure the grayscale of pixels in morphed image appear more real and may try another approach to solve empty intersection

◆ For blending method:

Find some image enhancement to strengthen the grayscale, let the image clear



References :

- [1] T. Beier and S. Neely, “Feature-based image metamorphosis”. *Computer Graphics (Proc. SIGGRAPH '92)*, Vol. 26, No.2, PP.35–42, 1992.
- [2] G. Wolberg, “Recent Advances in Image Morphing”. *Computer Graphics International, 1996*. Proceedings 24–28 June 1996 Page(s):64 – 71, 1996
- [3] G. Wolberg, “Digital Image Warping”. *IEEE Computer Society Press*, Los Alamitos, CA, 1990.
- [4] P. Litwinowicz and L. Williams, “Animating images with drawings”. *Computer Graphics (Proc. SIGGRAPH '94)*, pages 409–412, 1994.
- [5] S.-Y. Lee, K.-Y. Chwa, J. Hahn, and S. Y. Shin, “Image morphing using deformable surfaces”. *Proc. Computer Animation '94*, pages 31–39, 1994. IEEE Computer Society Press.
- [6] S.-Y. Lee, K.-Y. Chwa, J. Hahn, and S. Y. Shin, “Image morphing using deformation techniques”. *J. Visualization and Computer Animation*, 7(1):3–23, 1996.
- [7] S.-Y. Lee, K.-Y. Chwa, S. Y. Shin, and G. Wolberg, “Image metamorphosis using snakes and free-form deformations”. *Computer Graphics (Proc. SIGGRAPH '95)*, pages 439–448, 1995.
- [8] J. Serra, “Image Analysis and Mathematical Morphology” vol.1, Academic Press, 1982.
- [9] J. Serra, “Image Analysis and Mathematical Morphology” vol.2, Academic Press, 1988.
- [10] B. Luo and E. R. Hancock, “Slice interpolation using the distance transform and morphing”. *Digital Signal Processing Proceedings, 1997 13th International*

Conference on Publication Date: 2-4 Jul 1997 Volume: 2, page:1083-1086

- [11] R.-C. Gonzalez and R.-E. Woods, "Digital image processing second edition"
- [12] G. Borgefors, "Distance transformations in digital images". *Computer Vision, Graphics, and Image Processing* Pages: Volume 34 ,Issue 3 (June 1986)344 - 371
- [13] G. Boreefors, "Distance Transformations in Arbitrary Dimensions". *Computer Vision, Graphics, and Image Processing* Vol. 27, NO. 3, Sept. 1984, pp. 321.345
- [14] J. Serra, "Intepolations et distance de Hausdorff", *Tech.Rep. N-15/94/MM Centre de Morphologie Mathematique*, Ecole des Mines de Paris.
- [15] J. Serra, "Hausdorff distance and interpolations". *H. Heijmans and J. Roerdink in Mathematical morphology and its application to image and signal processing*, Kluwer, 1998.
- [16] F. Meyer, "A morphological interpolation method for Mosaic Image", Frence
- [17] S. Brucher, "Interpolation of sets, of partitions and of functions". *H. Heijmans and J. Roerdink in Mathematical morphology and its application to image and signal processing*, Editors, Kluwer, 1998.
- [18] S. Brucher, "Interpolation d'ensembles, de partitions ec de fonctions". *Tech Rep N-18/94/MM, Centre de Morphologie Mathematique, Ecole des Mines de paris,1994*
- [19] Marcin Iwanowski and Jean Serra, "Morphological Interpolation and Color Images". *Proceedings of the 10th International Conference on Image Analysis and Processing*, Page: 50, 1999
- [20] S. P. Raya and J.K. Udupa, "Shape-based interpolation of multidimensional objects," *IEEE Trans. on Medical Imagining*, vol. 1, no. 9, pp. 32–42, 1990.
- [21] Gabor T. Herman, Jingsheng Zheng, and Carolyn A. Bucholtz, "Shape-based Interpolation". *IEEE Computer Graphics & Applications*, 1992
- [22] Adrian G. Bors, Lefteris Kechagias, and Ioannis, "Shape-based interpolation

- using morphological morphing”. *Proc. IEEE Int. Conf. Image Processing, Vol. 2, Page 1661-1664, 2001*
- [23] Adrian G. Bors, Lefteris Kechagias, and Ioannis, “Binary Morphological Shape-Based Interpolation Applied to 3-D Tooth Reconstruction”. *IEEE TRANSACTIONS ON MEDICAL IMAGING, VOL. 21, NO. 2, FEBRUARY 2002*
- [24] Stanley Osher and James A. Sethian, “Fronts propagating with curvaturedependent speed: algorithms based on Hamilton-Jacobi formulations”. *J. Comput. Phys.*, 79(1):12–49, 1988.
- [25] Richard Tsai and Stanley Osher, “Level Set Methods in Image Science”. *IEEE Image Processing, 2003. Proceedings. 2003 International Conference on Volume 2, 14-17*
- [26] Stanley Osher and Ronald Fedkiw, “Level Set Methods and Dynamic Implicit Surfaces”, Springer; 1 edition, New York, November 1, 2002
- [27] Paul Burchard, Li-Tien Cheng, Barry Merriman, and Stanley Osher. “Motion of curves in three spatial dimensions using a level set approach”. *J. Comput. Phys.*, 170:720–741, 2001.
- [28] Li-Tien Cheng, Paul Burchard, Barry Merriman, and Stanley Osher. “Motion of curves constrained on surfaces using a level set approach”. *J. Comput. Phys.*, 175:604–644, 2002.
- [29] Baris Sumengen, “A Guide to Implementing Level Set Methods for Curve Evolution”. October 8, 2004
- [30] J. A. Sethian, “Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science”, Cambridge University Press, 1999.
- [31] Stanley Osher and Nikos Paragios, “Geometric Level Set Methods in imaging, vision, and Graphics”, Springer; 1 edition , July 17, 2003

- [32] Ross T. Whitaker, “A Level-Set Approach to Image Blending”. *IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 9, NO. 11, NOVEMBER 2000*
- [33] David E. Breen, and Ross T. Whitaker, “A Level-Set Approach for the Metamorphosis of Solid Models”. *IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, VOL. 7, NO. 2, APRIL-JUNE 2001*
- [34] TURK, G., AND O’BRIEN, J. F. “Modelling with implicit surfaces that interpolate”. *ACM TOG 21(4), page 855–873. 2002.*
- [35] CARR, J. C., BEATSON, R. K., CHERRIE, J. B., MITCHELL, T. J., FRIGHT, W. R., MCCALLUM, B. C., AND EVANS, T. R. “Reconstruction and representation of 3D objects with radial basis functions”. *In Proc. SIGGRAPH 2001, 67–76. 2001.*
- [36] TURK, G., AND O’BRIEN, J. F. 1999. “Shape transformation using variational implicit functions”. *In Proc. SIGGRAPH 99, 335–342.*
- [37] Michael Mullan, Ross Whitaker, and John C. Hart, “Procedural Level Sets”. *CARGO Remix - May 2004*