Mathematical Modeling for Structured Textures

A Thesis Presented to

The Faculty of the Mathematics Program

California State University Channel Islands

In (Partial) Fulfillment

of the Requirements for the Degree

Masters of Science

by

Juan Apolinar Zuñiga-Olea

December, 2009

APPROVED FOR THE MATHEMATICS PROGRAM

Dr. Kathryn Leonard, Advisor — Date: 9 Dec 09

Dr. Jorge Garcia — Date: Dec 15 09

Dr. Cynthia J. Wyels — Date: 12/14/09

APPROVED FOR THE UNIVERSITY

Dr. Gary A. Berg — Date

To my parents, Ipolito Zuñiga and Celedonia Olea, to my brothers, and to my loving wife Laura Chacon-Angeles, in gratitude for their encouragement and support. Gracias por todo su apoyo y por todo su amor que siempre me han brindado. Los quiero mucho.

Acknowledgements

This dissertation could not have been written without the constant advice and encouragement of my advisor, Dr. Kathryn Leonard. Many thanks must also go to the members of my committee, Dr. Jorge Garcia and Dr. Cynthia J. Wyels. To my family for always believing in me. Specially to my parents for giving me all of their love. To my professors, Dr. K. Leonard, Dr. Jorge Garcia, Dr. Ivona G., Dr. Cynthia J. Wyels. Special thanks to my wife Laura Chacon-Angeles, to my friend Victor Moreno and to anybody I miss, thank you all for always being there when I needed a hand. I couldn't done it with all of your help.

Abstract

Mathematical Modeling for Structured Textures

by Juan Apolinar Zuñiga-Olea

In the 1950s, scientists predicted it would take ten years to develop robots capable of performing most human tasks. Today, 50 years later, scientists are still predicting it will take ten years. The primary obstacle to building such robots is the problem of interpreting visual data, or image processing. Computers process images as grids of grey-level values. The structures in those values are the structures in the image. In other words, image processing requires mathematical analysis of images, where the images are represented as grids of numbers.

At the most basic level, images consist of shapes and textures. These textures range from completely regular (e.g., wallpaper) to completely random (e.g., white noise).

The mathematical qualities of regular textures are significantly different from the qualities of random textures because of the high degree of structure present in regular textures and absent in random ones. This project examines structured textures,

those close enough to regular textures to be viewed as deformed versions of regular textures (e.g., Figure 1.1). In particular, we are interested in learning how to relate properties of the regular texture with properties of the structured texture (i.e., after deformation).

Ultimately, we want to answer the question: Given an unknown structured texture, can we recover a regular texture and the associated deformation?

Contents

1 Introduction **1**

2 Thin Plate Splines **5**

2.1 Thin-plate spline 8

3 Polynomial Deformations **18**

3.1 Piecewise Linear 19

3.2 Cubic 21

3.3 General Polynomial 22

3.4 Piecewise cubic or B-spline 23

4 Autocorrelation, Self Similarity and Deformations **46**

4.1 Background for Autocorrelation 48

4.2 Autocorrelation and periodic structures 54

4.3 Autocorrelation and deformations 56

5 Conclusion **59**

Chapter 1

Introduction

This thesis arises from an open question about finding original images and deformations from warped functions. We can see that if we know the original image we need to know only certain corresponding points of the deformed (target image) ahead of time in order for us to find the map that takes one to the other.

What is the difference between a human eye and any other form of sight? For one, our eye can distinguish between texture and shape. Our eyesight, as any other part of our body, is wonderfully created. We have seen it in movies, cartoons, and many other media; robots or computers replicating, our vision with no flaws. That is the magic of the media; everything seems believable. But, it will be years before we can replicate our vision and

integrate it into computers. Vision is very complex, so complex that today's computers are not able to distinguish between shape and texture.

Imagine if we put a pencil, pen, straw and a book on a table. A computer may distinguish between sizes, yet it cannot distinguish between objects. If you were to ask a human to pick up the pen it will do so with no hesitation. Yet, if a robot where to perform the same task, it will have a difficult time. Another task we can ask a human to perform is to tell which object is closer to them. Our eyes can perceive how close or far an object is. Computers are not able to perform this task without complications.

This project, texture modeling, arose from the fact that, in the 1950s, engineers predicted that within 10 years they would produce robots that could perform most of the tasks humans perform. The difficulty in doing so is the development of an artificial visual system that would allow robots to process visual information like humans do. Such a visual system should be able to take in an image (images are composed of shape and texture), determine contours of shapes and recognize the shapes, and determine areas of texture and recognize the textures.

The first category we want to understand is the *near-regular* category which contains textures that are close enough to regularly tiled textures. Below is an example of a near-regular texture.

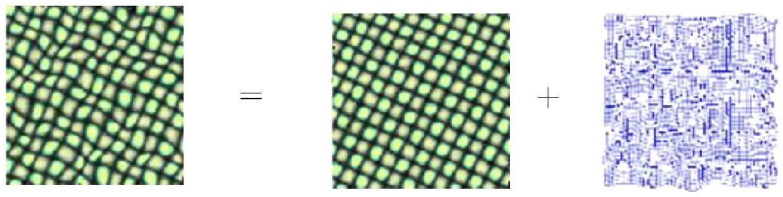

Figure 1.1: Near-regular textures = regular textures + deformation.

We would like to understand and model near-regular textures. In Figure 1.1, we can see that a near-regular texture is composed of a regular texture plus a deformation. The class of regular textures is simply wallpaper patterns or symmetry groups which are very well known. Since regular textures are completely understood, we want to understand the class of deformations. We explore useful classes of deformations. A deformation should have certain desirable properties. It must be bijective, differentiable, easily solvable, and flexible for both small and large deformations.

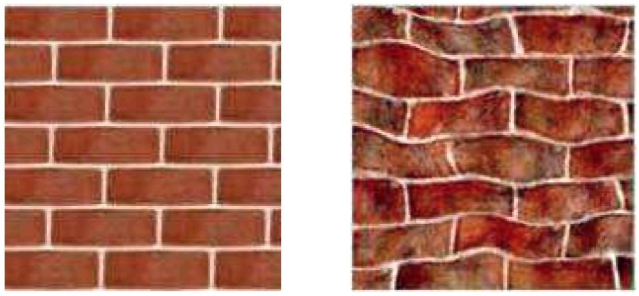

Figure 1.2: Example of a regular image and the image with a deformation.

The second category we want to understand is: given an image can we recover the underlying regular texture? One of the tools that we explore is the autocorrelation. We can use the autocorrelation to test the periodicity of an image helping us recover the underlying regular texture.

Chapter 2

Thin Plate Splines

In order for us to be able to answer the question: Are there any mathematical measurements that will allow us to distinguish stochastic from near-regular texture? we need to understand what happens to functions under different types of deformation. As periodic functions are deformed we would like to study how the periodicity is distorted. This will give insights into our goal of starting with a deformed texture and finding the underlying regular texture and the deformation that took the regular texture to the deformed texture. We work our way from the most simple deformations to the more complex.

Definition 1. A **landmark** is a point on a figure.

Definition 2. Consider two k landmark configuration matrices in $\mathbb{R}^m$,

$T = (t_1, \ldots, t_k)'$ and $Y = (y_1, \ldots, y_k)'$ both $k \times m$ matrices and we wish to deform T (the source) into Y (the target) where $t_j, y_j \in \mathbb{R}^m$. We use the notation that the m-vector t_j is written as $t_j = (t_j[1], \ldots, t_j[m])'$. A **deformation** is a mapping from $T \in \mathbb{R}^m$ to $Y \in \mathbb{R}^m$ defined by the transformation:

$$Y = \Phi(T) = (\Phi_1(T), \Phi_2(T), \ldots, \Phi_m(T))',$$

where $'$ denotes transformation of a matrix, T is the source and Y is the target.

The function $\Phi(t)$ should have certain desirable properties. In particular, we would like for it to be bijective, differentiable, easily solvable, flexible for both small and large deformations. The class of, say, differentiable deformations is immensely large. Hence, we would like to restrict ourselves to more manageable deformations. In particular, we would like to restrict our class of deformations to only those that would have most of the desired properties for a deformation.

Below we use deformation, transformation, and warping interchangeably to refer to a deformation. The following are several different types of deformations that we looked at.

The simplest deformation is a linear deformation. Let the map $\phi(x) : x \mapsto ax$ be a polynomial linear deformation. Hence, if we let $f(x)$

represent the original image, then the deformed image would be $f \circ \phi(x) = f(ax)$. For example, $f(x) = \sin(x)$ and $f \circ \phi(x) = f(ax) = \sin(ax)$ would be as follows:

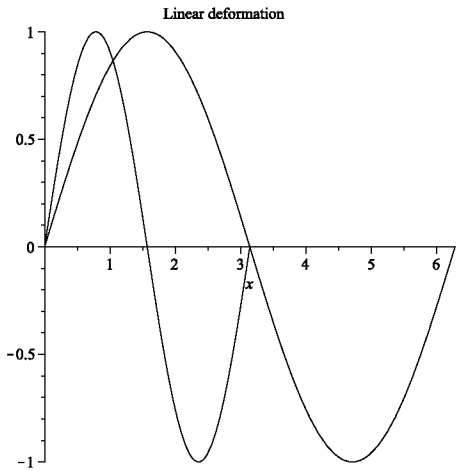

Figure 2.1: Example of a linear deformation $\sin(x)$ and $\sin(ax)$. Note that this restricts the subintervals. For example, if $x \in [0, 2\pi]$, then $ax \in [0, \frac{2\pi}{a}]$.

For simplicity, we consider the 1D (1 dimensional) case, but the theory can be generalized. With a linear approximation, any deformation that is approximated is a coarse approximation. Yet, we only need to find two unknowns to find a linear deformation, so the class is simple. The simplest deformations that we came across with are linear deformations, but not sufficiently rich. One way to enrich the class is to allow thin plate splines. This, as we will learn, is a linear deformation with a nonlinear part and a

shift.

2.1 Thin-plate spline

Thin-plate spline is a natural interpolating function for data in two dimensions. Why do we care about thin-plate spline? Well in practice, if there are more than $m+1$ landmarks (knots or points on a figure) in m dimensions, an affine transformation in general will not be able to find the deformation, so we need to use a non-affine deformation. The thin-plate spline consists of an affine and non-affine part, which allows for nonlinearity in deformations. In calculating a deformation grid we, do not want to see any more bending locally than necessary, and also do not want to see bending where there are no data. Thin-plate spline minimizes the amount of bending in transforming between two configurations. We make this more precise below.

Consider the (2×1) landmarks t_j, $j = 1, \ldots, k$, on the first figure mapped exactly into y_i, $i = 1, \ldots, k$, on the second figure, i.e. there are $2k$ interpolation constraints, $(y_j)_r = \Phi_r(t_j)$, $r = 1, 2$, $j = 1, \ldots, k$, and we write $\Phi(t_j) = (\Phi_1(t_j), \Phi_2(t_j))'$, $j = 1, \ldots, k$, for the two dimensional

deformation, see example 1. Let

$$T = \begin{bmatrix} t_1 \\ t_2 \\ \cdots \\ t_k \end{bmatrix}, \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_k \end{bmatrix}$$

so that T and Y are both $(k \times 2)$ matrices.

Definition 3. A pair of thin-plate splines(PTPS) is given by the bivariate function

$$\Phi(t) = (\Phi_1(t), \Phi_2(t)) = c + At + W's(t),$$

where W' is the warping function, $s(t)$ is the weight on the warping distance (how far away). Note that the warping increases when it is farther away from the landmark point shift. Also, t (the knots, landmarks, or points) is (2×1), $s(t) = (\sigma(t - t_1), \ldots, \sigma(t - t_k))'$, $(k \times 1)$, and

$$\sigma(h) = \begin{cases} \|h\|^2 \log(\|h\|), & \|h\| > 0, \\ 0, & \|h\| = 0 \end{cases}$$

where $h = t_i - t_j$, and σ measures how much warping is present in the interpolation.

As h is small $\sigma(h)$ is small, hence $s(t)$ also gets small. As h is large $\sigma(h)$ is large, and $s(t)$ gets large. In the definition of the thin-plate spline,

A is the linear part, W is the warping, and the coefficient of the warping function, $s(t)$, is nonlinear as you can see from the definition of $s(t)$ and $\sigma(t)$.

Note that c is (2×1), A is (2×2) and W is $(k \times 2)$, this makes a total of $2k + 6$ parameters that are being mapped. Remember that we said there are $2k$ interpolation constraints, and we introduce six more constraints in order for the bending energy equation B_e to be defined.

Definition 4. Bending energy is how much bending or warp is present on an image after being deformed. It provides six more constraints. Hence we defined the following equations in order for the bending equation to be defined:

$$1_k' W = 0, \;\; T'W = 0.$$

Combining equations $\Phi(t) = c + At + W's(t)$, $1_k' W = 0$, and $T'W = 0$, into a matrix we can write:

$$\begin{bmatrix} S & 1_k & T \\ 1_k' & 0 & 0 \\ T' & 0 & 0 \end{bmatrix} \begin{bmatrix} W \\ c' \\ A' \end{bmatrix} = \begin{bmatrix} Y \\ 0 \\ 0 \end{bmatrix},$$

where $(S)_{ij} = \sigma(t_i - t_j)$ and 1_k is the k-vector of ones. If we multiply out the matrix above we will obtain the thin plate spline equation and the two equations we defined.

Provided that the inverse of S exists, the inverse of the matrix

$$\begin{bmatrix} S & 1_k & T \\ 1_k' & 0 & 0 \\ T' & 0 & 0 \end{bmatrix}$$

exists. Therefore,

$$\begin{bmatrix} W \\ c' \\ A' \end{bmatrix} = \begin{bmatrix} S & 1_k & T \\ 1_k' & 0 & 0 \\ T' & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} Y \\ 0 \\ 0 \end{bmatrix} = \Gamma^{-1} \begin{bmatrix} Y \\ 0 \\ 0 \end{bmatrix}.$$

And,

$$\Gamma^{-1} = \begin{bmatrix} \Gamma^{11} & \Gamma^{12} \\ \Gamma^{21} & \Gamma^{22} \end{bmatrix} = \begin{bmatrix} S & Q \\ Q' & 0 \end{bmatrix}^{-1},$$

where Γ^{11} is $k \times k$ and $Q = \begin{bmatrix} 1_k, & T \end{bmatrix}$, it follows that

$$W = \Gamma^{11} Y$$

$$\begin{bmatrix} c' \\ A' \end{bmatrix} = \Gamma^{21} Y,$$

Hence, given the initial image and the deformed image we can obtain warping function, mainly $\Phi(t) = Y$. In other words, given matrix landmarks T and Y and assumptions, we can find a thin plate spline taking $T \mapsto Y$ minimizing the bending in the image.

Example 1. The following is an example of how we can apply the pair of thin-plate splines.

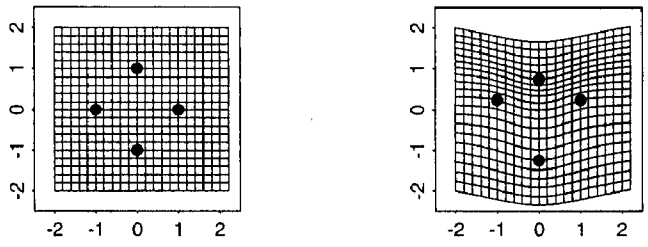

Figure 2.2: An original image and its deformed image by PTPS. The grid on the left is the source "T" image and the right is the target "Y" image.

From the images above we can obtain the matrices T and Y:

$$T = \begin{bmatrix} 0 & 1 \\ -1 & 0 \\ 0 & -1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & 0.75 \\ -1 & 0.25 \\ 0 & -1.25 \\ 1 & 0.25 \end{bmatrix},$$

Taking $(S)_{ij} = \sigma(\|t_i - t_j\|)$, for $i, j = 1 \ldots 4$, we obtain the following:

$$\begin{aligned} (S)_{11} &= \sigma(\|t_1 - t_1\|) = \sigma(0) = 0 \\ (S)_{12} &= \sigma(\|t_1 - t_2\|) = \sigma(\sqrt{2}) \\ (S)_{13} &= \sigma(\|t_1 - t_3\|) = \sigma(2) \end{aligned}$$

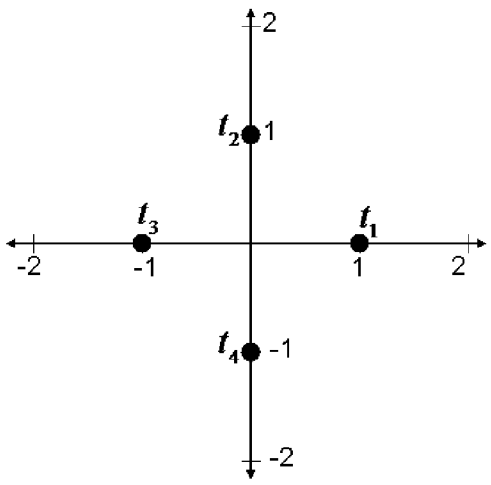

Figure 2.3: Kite. Defining the source landmarks the following way.

and so on. We then obtain the matrix

$$S = \begin{bmatrix} S_{11} & \cdots & S_{14} \\ \vdots & \ddots & \vdots \\ S_{41} & \cdots & S_{44} \end{bmatrix} = \begin{bmatrix} 0 & a & b & a \\ a & 0 & a & b \\ b & a & 0 & a \\ a & b & a & 0 \end{bmatrix},$$

where $a = \sigma(\sqrt{2}) \approx 0.6931$ and $b = \sigma(2) \approx 2.7726$. We now use

$$Q = \begin{bmatrix} 1_k, & T \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{bmatrix},$$

to find the inverse of Γ^{-1}:

$$\Gamma^{-1} = \begin{bmatrix} \Gamma^{11} & \Gamma^{12} \\ \Gamma^{21} & \Gamma^{22} \end{bmatrix} = \begin{bmatrix} S & Q \\ Q' & 0 \end{bmatrix}^{-1}.$$

We will find that:

$$\begin{aligned} \Gamma^{11} &= S^{-1} - S^{-1}Q(Q'S^{-1}Q)^{-1}Q'S^{-1}, \\ \Gamma^{21} &= (Q'S^{-1}Q)^{-1}Q'S^{-1}, \end{aligned}$$

$$S^{-1} = \begin{bmatrix} 0.0601 & -0.1202 & 0.4207 & -0.1202 \\ -0.1202 & 0.0601 & -0.1202 & 0.4207 \\ 0.4207 & -0.1202 & 0.0601 & -0.1202 \\ -0.1202 & 0.4207 & -0.1202 & 0.0601 \end{bmatrix},$$

$$Q' = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -1 & 0 & 1 \\ 1 & 0 & -1 & 0 \end{bmatrix},$$

we then obtain

$$\Gamma^{11} = \begin{bmatrix} 0.1803 & -0.1803 & 0.1803 & -0.1803 \\ -0.1803 & 0.1803 & -0.1803 & 0.1803 \\ 0.1803 & -0.1803 & 0.1803 & -0.1803 \\ -0.1803 & 0.1803 & -0.1803 & 0.1803 \end{bmatrix}.$$

Hence, we obtain the following results

$$B_e = \Gamma^{11} = 0.1803 \begin{bmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{bmatrix}$$

. The $k \times k$ matrix B_e is called the bending energy matrix. Why is the bending energy so important? Remember that we do not want to see any more bending locally than necessary or see bending where there are no data. Thin-plate spline minimizes bending between two configurations, where the total bending energy is given by:

$$J(\Phi) = \sum_{j=1}^{2} \int \int_{\mathbb{R}^2} (\frac{\partial^2 \Phi_j}{\partial x^2})^2 + 2(\frac{\partial^2 \Phi_j}{\partial x \partial y})^2 + (\frac{\partial^2 \Phi_j}{\partial y^2})^2 \, dxdy.$$

It has been shown that the minimized total bending energy is given by,

$$J(\Phi) = trace(W'SW) = trace(Y'\Gamma^{11}Y).$$

As a result, thin-plate spline minimizes the bending energy between two configurations.

$$W = \Gamma^{11}Y = \begin{bmatrix} 0 & -0.1803 \\ -1 \times 10^{-14} & 0.1803 \\ 0 & -0.1803 \\ 2 \times 10^{-14} & 0.1803 \end{bmatrix},$$

and using the fact that $\Gamma^{21} = (Q'S^{-1}Q)^{-1}Q'S^{-1}$, we can find the following result

$$\Gamma^{21} = \begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0 & -0.5 & 0 & 0.5 \\ 0.5 & 0 & -0.5 & 0 \end{bmatrix},$$

which implies that

$$\begin{bmatrix} c' \\ A' \end{bmatrix} = \Gamma^{21}Y = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \Longrightarrow c' = \begin{bmatrix} 0 & 0 \end{bmatrix} \Longrightarrow c = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

and

$$A' = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \Longrightarrow A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

From the example above we can see that there is no shift, and that the linear part is simply the identity matrix, which means there is no linear change, and the nonlinear part comes from the warping in the image. Note also, that in the example, we solved a nonlinear deformation, yet the solution was linear.

Using the pair thin-plate splines, we can see that if we know the original image, we need to know only certain corresponding points of the deformed

(target image) ahead of time in order for us to find the corresponding thin-plate spline map that takes one to the other. Thin-plate spline has all of the desired properties that we would like on a deformation. It is assumed to be differentiable. Yet, it does not generate the continuous function. Therefore, we did not continue using thin-plate spline. As a result, we started looking at other rich classes of deformations, primarily polynomial deformations.

Chapter 3

Polynomial Deformations

Thin-plate spline generalizes linear deformation to a class of deformations that can be expressed as $t \mapsto c+At+w's(t)$. Another generalized class comes from polynomial families.The simplest deformation is a linear deformation. Let the map

$\phi(x) : x \mapsto ax$ be a polynomial linear deformation. Hence, if we let $f(x)$ represent the original image, then the deformed image would be

$f \circ \phi(x) = f(ax)$. For example, $f(x) = \sin(x)$ and $f \circ \phi(x) = f(ax) = \sin(ax)$ would be as follows:

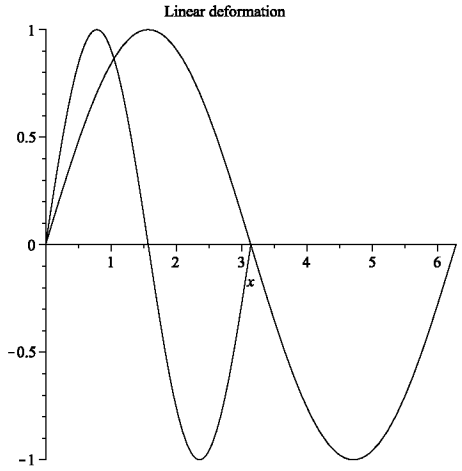

Figure 3.1: Example of a linear deformation $\sin(x)$ and $\sin(ax)$. Note that this restricts the subintervals. For example, if $x \in [0, 2\pi]$, then $ax \in [0, \frac{2\pi}{a}]$.

For simplicity, we consider the 1D (1 dimensional) case, but the theory can be generalized. With a linear approximation, any deformation that is approximated is a coarse approximation. Yet, we only need to find two unknowns to find a linear deformation, so the class is simple.

3.1 Piecewise Linear

To allow finer approximation, we apply a piecewise linear deformation. As you may note, the piecewise linear deformation is more general than the linear approximation. The reason is that a curve can now be approximated by more than one line, within an interval. This gives a more general, accurate

approximation than the linear.

Let the map

$$\phi(x) : x \mapsto \begin{cases} a_1 x & x \in [0, x_1] \\ a_2 x + b_2 & x \in [x_1, x_2] \\ \vdots & \\ a_n x + b_n & x \in [x_{n-1}, x_n] \end{cases}$$

be a piecewise linear deformation, where $a_i = \frac{y_i - y_{i-1}}{x_i - x_{i-1}}$ and $b_i = -a_i t_i + y_i$.

The following are examples of periodic functions before and after piecewise linear deformations. Piecewise linear deformation is flexible and simple, but it is restricted in the sense that it is not differentiable across subintervals.

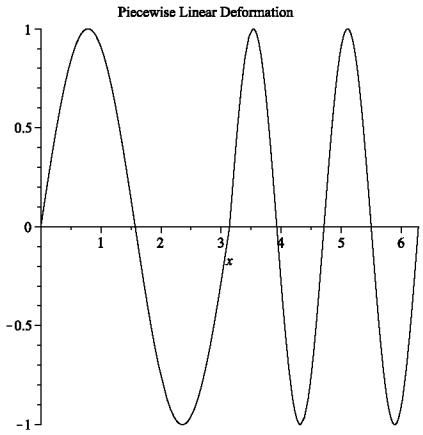

Figure 3.2: Example of a piecewise linear deformation.

3.2 Cubic

A quadratic and piecewise deformations are more general than a linear deformation, but problematic. Hence, we move directly to cubic deformations.

The map $\phi(x) : x \mapsto ax^3$ is a cubic deformation, where only the leading term is put to use to simplify the work and the complexity of the deformation. Just as we introduced this chapter, the higher the order of the polynomial the more accurate an approximation we get when approximating a curve by polynomial. We also get more control of differentiability with a higher degree polynomial. On the other hand, we need to find four unknowns in order to find the cubic polynomial map from a regular image to a deformed image.

The following is an example of a function and its deformed graph using cubic deformation.

The 2-Dimensional cubic deformation is defined as:

$$\phi(x, y) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} x^i y^j.$$

The map $x \mapsto ax^3 + bx^2 + cx + d$ is a general cubic deformation. As we know the higher the order of our polynomial approximation, the more accurate we will get. The only difference is that we need to find $n + 1$

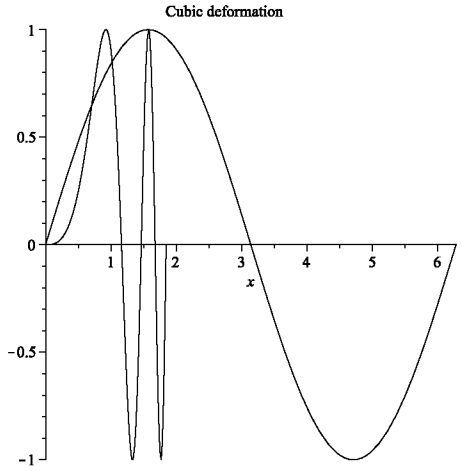

Figure 3.3: Example of a cubic deformation. Note that this restricts the subintervals. For example, if $x \in [0, 2\pi]$, then $ax \in [0, \sqrt[3]{a\pi}]$.

unknowns in order for us to find the map. So, in order to have more accurate approximation we need to work more.

3.3 General Polynomial

As we learned with the thin-plate spline, we needed to know the original image and certain corresponding points of the target image in order for us to find the map that takes one to the other. The more accurate you want to be, the higher the order of the polynomial map you have to use. The general polynomial map is the following: $x \mapsto a_n x^n + \cdots + a_0$ for 1D $\vec{x} \mapsto A_n \vec{x}^n + \cdots + A_0$ for $m - d$ dimensions.

The general polynomial will lead to a lot of work since the higher the order approximation, the more unknowns there are needed to be known. For a degree of n you need to find $n+1$ unknowns, or $n+1$ coefficients. The challenge is to choose n so that the class of deformations is sufficiently rich but not too complicated. Say, $n = 3$. So, why $n = 3$? Well, $n = 1$ is the simplest deformation, $n = 2$ is very problematic. Hence, we look at B-splines, since they are piecewise cubic, and they are differentiable.

3.4 Piecewise cubic or B-spline

B-spline is a piecewise cubic function that is twice differentiable. B-splines are constructed from cubic blending function over four spans. B-spline is differentiable with linear spans. It guarantees differentiability across spans. The piecewise cubic polynomial is more general than the cubic polynomial and allows for a broad class of deformations and more local control. The blending function, defined below, guarantees differentiability across spans. As we will see when approximating a curve with the cubic B-spline, a change, move or introduction of one point will not require a new B-spline function. B-splines give a better approximation when given a non B-spline deformation. It is a very rich class of deformations, and it is differentiable.

This gives us the best tool to use in case we need to introduce more points, or more points are known in our regular image to find the map that would take it to its deformed image. On the other hand, to gain this flexibility, we need to find four unknowns on each subintervals in order to find a given B-spline deformation. We will define the B-spline for curves (1D B-splines) first, and then generalize to B-splines for surfaces (2D B-splines). We encounter very similar problems for the 2D B-splines as we do for the 1D B-splines. Yet, we will learn that a 2D B-spline is nothing else but a 1D B-spline taken twice, where we introduce a new variable v.

Definition 5. The cubic B-spline blending function is given by

$$B(u) = \begin{cases} (2+u)^3/6, & -2 \le u \le -1, \\ (4-6u^2-3u^3)/6, & -1 \le u \le 0, \\ (4-6u^2+3u^3)/6, & 0 \le u \le 1, \\ (2-u)^3/6, & 1 \le u \le 2, \\ 0, & \textit{otherwise}, \end{cases}$$

To use cubic B-spline, we will use a set of points known as parametric knots $A_0, A_1, \ldots, A_N$ and two extra points known, as phantom knots A_{-1} and A_{N+1}. We selected the parametric knot set so that the B-spline interpolates the geometry knots, or the control points P_{i-2}, P_{i-1}, P_i, P_{i+1}, P_{i+2}. The phantom knots are introduced to allow flexibility when setting

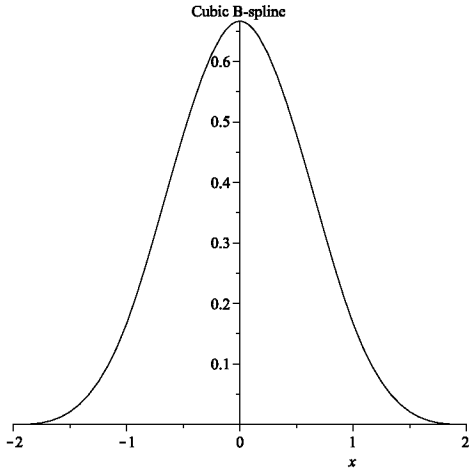

Figure 3.4: Graph of the Cubic B-spline blending function.

the derivative values at the ends of the spline.

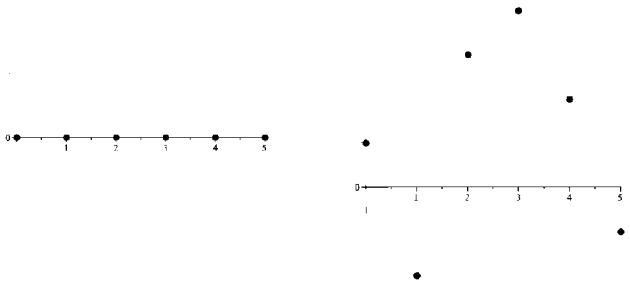

Figure 3.5: On the left are the A_i's and on the right are the P_i's.

The local control offered by B-splines means that if we introduce a new control point, we do not have to rework the whole problem. The B-spline expression only changes within the neighborhood of the new control point. By the definition of the cubic blending function, we can see that it has support 4, which is the number of spans over which it is non-zero. Therefore,

if we change, say, the point P_k, it will only affect the four spans $k-1$, k, $k+1$, and $k+2$. Hence, we have the opportunity for local control of the B-spline.

Formally, we assume a point correspondence deformation. Given that correspondence, we will construct a system of equations that will allow us to find the coefficients of the associated b-spline deformation. Assume that the point A_i has position vector $\mathbf{R}_i$, and that the parameter μ moves from 0 at P_0 to 1 at P_N and has value i/N at P_i.

Then the equation of the spline interpolating the points P_i is:

$$\mathbf{r}(\mu) = \sum_{i=-1}^{N+1} B(N\mu - i)\mathbf{R}_i.$$

We can then find the position vectors $\mathbf{R}_i$ by the fact that the geometric knots $P_0, \ldots, P_N$ occur when $\mu = \frac{i}{N}$ $(i = 0, \ldots, N)$. Therefore,

$$\mathbf{r}_j = \mathbf{r}(\frac{j}{N}) = \sum_{i=-1}^{N+1} B(N(\frac{j}{N}) - i)\mathbf{R}_i = \sum_{i=-1}^{N+1} B(j-i)\mathbf{R}_i$$

and by the definition of the cubic B-Spline, our u parameter is between -2 and 2, we have $-2 \leq j - i \leq 2 \Longrightarrow j - 2 \leq i \leq j + 2$. Hence,

$$\begin{aligned}\sum_{i=-1}^{N+1} B(j-i)\mathbf{R}_i &= B(2)\mathbf{R}_{j-2} + B(1)\mathbf{R}_{j-1} + B(0)\mathbf{R}_j + B(-1)\mathbf{R}_{j+1} + B(-2)\mathbf{R}_{j+2} \\ &= \frac{1}{6}(\mathbf{R}_{j-1} + 4\mathbf{R}_j + \mathbf{R}_{j+1}).\end{aligned}$$

This will give us a set of $N+1$ equations for $N+3$ unknowns $\mathbf{R}_j$. Hence, we need two more equations. They can be found by setting the gradient at

the ends to g_0 and g_N, for example, $\mathbf{r}'(0) = g_0$ and $\mathbf{r}'(1) = g_N$. So,

$$\begin{aligned}\mathbf{r}(\mu) &= \sum_{i=-1}^{N+1} B(N\mu - i)\mathbf{R}_i \\ &= B(N\mu+1)\mathbf{R}_{-1} + B(N\mu)\mathbf{R}_0 + B(N\mu-1)\mathbf{R}_1 \\ &+ \cdots + B(N\mu - N - 1)\mathbf{R}_{N+1}.\end{aligned}$$

Then,

$$\begin{aligned}\mathbf{r}'(\mu) &= NB'(N\mu+1)\mathbf{R}_{-1} + NB'(N\mu)\mathbf{R}_0 + NB'(N\mu-1)\mathbf{R}_1 \\ &+ \cdots + NB'(N\mu - N - 1)\mathbf{R}_{N+1}.\end{aligned}$$

Therefore,

$$\mathbf{r}'(0) = g_0 = N(B'(1)\mathbf{R}_{-1} + B'(0)\mathbf{R}_0 + B'(-1)\mathbf{R}_1).$$

Using the fact that

$$B'(u) = \begin{cases} (2+u)^2/2, & -2 \le u \le -1, \\ (-4u - 3u^2)/2, & -1 \le u \le 0, \\ (-4u + 3u^2)/2, & 0 \le u \le 1, \\ -(2-u)^2/2, & 1 \le u \le 2, \\ 0, & otherwise, \end{cases}$$

it follows that

$$\mathbf{r}'(0) = g_0 = N(-\frac{1}{2}\mathbf{R}_{-1} + 0\mathbf{R}_0 + \frac{1}{2}\mathbf{R}_1) = \frac{1}{2}N(\mathbf{R}_1 - \mathbf{R}_{-1}).$$

Similarly, $g_N = \frac{1}{2}N(\mathbf{R}_{N+1} - \mathbf{R}_{-1})$. We have obtained the following equation:

$Mx = h$, where

$$M = \begin{bmatrix} -N & 0 & N & 0 & 0 & \cdots & 0 \\ 1 & 4 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 4 & 1 & 0 & \cdots & 0 \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & \cdots & 1 & 4 & 1 \\ 0 & 0 & 0 & \cdots & -N & 0 & N \end{bmatrix}$$

x is the vector of unknowns

$$x = \begin{bmatrix} \mathbf{R}_{-1} \\ \mathbf{R}_0 \\ \vdots \\ \mathbf{R}_N \\ \mathbf{R}_{N+1} \end{bmatrix} = \begin{bmatrix} X_{-1} & Y_{-1} & Z_{-1} \\ \vdots & \vdots & \vdots \\ X_{N+1} & Y_{N+1} & Z_{N+1} \end{bmatrix}$$

and h is the vector of known values

$$h = 6 \begin{bmatrix} \frac{1}{3}g_0 \\ \mathbf{r}_0 \\ \vdots \\ \mathbf{r}_N \\ \frac{1}{3}g_N \end{bmatrix} = 6 \begin{bmatrix} \frac{1}{3}g_{x_0} & \frac{1}{3}g_{y_0} & \frac{1}{3}g_{z_0} \\ x_0 & y_0 & z_0 \\ \vdots & \vdots & \vdots \\ x_N & y_N & z_N \\ \frac{1}{3}g_{x_N} & \frac{1}{3}g_{y_N} & \frac{1}{3}g_{z_N} \end{bmatrix}$$

.

In practice, we would solve the system the following way: $Mx = h$.

$$\begin{bmatrix} -N & 0 & N & 0 & 0 & \cdots & 0 \\ 1 & 4 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 4 & 1 & 0 & \cdots & 0 \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & \cdots & 1 & 4 & 1 \\ 0 & 0 & 0 & \cdots & -N & 0 & N \end{bmatrix} \begin{bmatrix} X_{-1} & Y_{-1} & Z_{-1} \\ \vdots & \vdots & \vdots \\ X_{N+1} & Y_{N+1} & Z_{N+1} \end{bmatrix} =$$

$$= 6 \begin{bmatrix} \frac{1}{3}g_{x_0} & \frac{1}{3}g_{y_0} & \frac{1}{3}g_{z_0} \\ x_0 & y_0 & z_0 \\ \vdots & \vdots & \vdots \\ x_N & y_N & z_N \\ \frac{1}{3}g_{x_N} & \frac{1}{3}g_{y_N} & \frac{1}{3}g_{z_N} \end{bmatrix}$$

where $\mathbf{R}_i = (X_i, Y_i, Z_i)$ and $\mathbf{r}_i = (x_i, y_i, z_i)$

For the curve to pass through P_0, we require that $\mathbf{r}_{-1} = 2\mathbf{r}_0 - \mathbf{r}_1$. Similarly, we require that $\mathbf{r}_{N+1} = 2\mathbf{r}_N - \mathbf{r}_{N-1}$. This would force the curve to pass through P_N.

Using the fact that

$$B'(u) = \begin{cases} (2+u)^2/2, & -2 \le u \le -1, \\ (-4u-3u^2)/2, & -1 \le u \le 0, \\ (-4u+3u^2)/2, & 0 \le u \le 1, \\ -(2-u)^2/2, & 1 \le u \le 2, \\ 0, & otherwise, \end{cases}$$

We can see that B-spline is differentiable across control points:

$(2+u)^2/2$ at $u=-1$ it equals $\frac{1}{2}$.

$(-4u-3u^2)/2$ at $u=-1$ it equals $\frac{1}{2}$.

$(-4u-3u^2)/2$ at $u=0$ it equals 0.

$(-4u+3u^2)/2$ at $u=0$ it equals 0.

$(-4u+3u^2)/2$ at $u=1$ it equals $-\frac{1}{2}$.

$-(2-u)^2/2$ at $u=1$ it equals $-\frac{1}{2}$.

Example 2. Construct the natural cubic spline for the 7 points

$P_i \in \{(0,1),(1,0),(2,0),(3,1),(4,2),(5,2),(6,1)\}$ and

$A_i \in \{(0,0),(1,0),(2,0),(3,0),(4,0),(5,0),(6,0)\}$. For this construction we will use the fact that:

$R_{-1} = 2R_0 - R_1$

$R_{N+1} = 2R_N - R_{N-1}$

Solve the system of equations for the coefficients:

$R_{-1} = 2R_0 - R_1 \Rightarrow R_{-1} - 2R_0 + R_1 = 0$

$R_{-1} + 4R_0 + R_1 = 1$

$R_0 + 4R_1 + R_2 = 0$

$R_1 + 4R_2 + R_3 = 0$

$R_2 + 4R_3 + R_4 = 1$

$R_3 + 4R_4 + R_5 = 2$

$R_4 + 4R_5 + R_6 = 2$

$R_5 + 4R_6 + R_7 = 1$

$R_7 = -R_5 + 2R_6 \Rightarrow R_5 - 2R_6 + R_7 = 0$

We obtain the following matrix:

$$M = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 \end{bmatrix}$$

If we solve the following equation:

$$\begin{bmatrix} R_{-1} \\ R_0 \\ R_1 \\ R_2 \\ R_3 \\ R_4 \\ R_5 \\ R_6 \\ R_7 \end{bmatrix} = M^{-1} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 2 \\ 2 \\ 1 \\ 0 \end{bmatrix}$$

We obtain:

$$\begin{bmatrix} R_{-1} \\ R_0 \\ R_1 \\ R_2 \\ R_3 \\ R_4 \\ R_5 \\ R_6 \\ R_7 \end{bmatrix} = \begin{bmatrix} \frac{11}{30} \\ \frac{1}{6} \\ -\frac{1}{30} \\ -\frac{1}{30} \\ \frac{1}{6} \\ \frac{11}{30} \\ \frac{11}{30} \\ \frac{1}{6} \\ -\frac{1}{30} \end{bmatrix}$$

$$\mathbf{r}(\mu) = \sum_{i=-1}^{N+1} B(N\mu - i)\mathbf{R}_i = \sum_{i=-1}^{N+1} B(u - i)\mathbf{R}_i$$

For $0 \leq u \leq 1$. Therefore,

$$\begin{aligned}
\mathbf{r}(\mu) &= \frac{11}{30}B(u+1) + \frac{1}{6}B(u) \\
&- \frac{1}{30}B(u-1) - \frac{1}{30}B(u-2) + \frac{1}{6}B(u-3) \\
&+ \frac{11}{30}B(u-4) + \frac{11}{30}B(u-5) + \frac{1}{6}B(u-6) - \frac{1}{30}B(u-7)
\end{aligned}$$

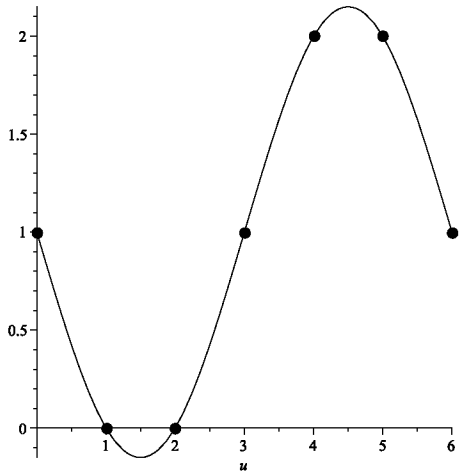

Figure 3.6: B-spline interpolation example of 7 points.

So for B-spline, we end up solving for coefficients by linear operations (i.e. invert matrix), but give nonlinear deformation because coefficients are coefficients of nonlinear polynomials.

The previous material was all for 1D B-splines. Now we can move to 2D, cubic B-splines, which will turn out to be 2 sets of 1D B-splines. We come across the same difficulties for the edges where we needed the curve to be forced to pass through P_0 and P_N. Again we will need a set of points called phantom knots along the boundaries. They are used in order for the B-spline to interpolate the control points. So, in order for us to control the surface at the edges, we introduce phantom knots so that the knot set becomes $\{P_{ij}\}$ where $i = -1, 0, \ldots, N+1$, and $j = -1, 0, \ldots, M+1$. We introduce these phantom knots to allow flexibility when setting the derivative values at the edges of the surface. Therefore, the blending function takes the following form

$$\mathbf{r}(u,v) = \sum_{i=-1}^{N+1} \sum_{j=-1}^{M+1} B(Nu-i)B(Nv-j)\mathbf{r}_{ij}$$

where $0 \leq u \leq 1$, and $0 \leq v \leq 1$.

At a knot, $u = \frac{i}{N}$, and $v = \frac{j}{M}$ so that

$$\mathbf{r}(\frac{i}{N}, \frac{j}{M}) = \begin{bmatrix} \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{bmatrix} \begin{bmatrix} \mathbf{r}_{i-1,j-1} & \mathbf{r}_{i-1,j} & \mathbf{r}_{i-1,j+1} \\ \mathbf{r}_{i,j-1} & \mathbf{r}_{i,j} & \mathbf{r}_{i,j+1} \\ \mathbf{r}_{i+1,j-1} & \mathbf{r}_{i+1,j} & \mathbf{r}_{i+1,j+1} \end{bmatrix} \begin{bmatrix} \frac{1}{6} \\ \frac{2}{3} \\ \frac{1}{6} \end{bmatrix}$$

. So to ensure that the surface passes through the corners and to have a

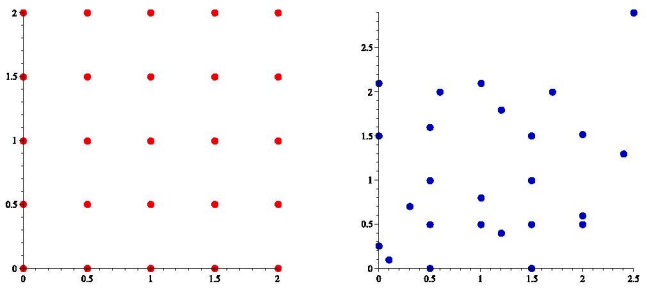

Figure 3.7: On the left are the $A_{i,j}$'s (source) and on the right are the $P_{i,j}$'s (target).

smooth surface, we can choose double knots so that

$$\mathbf{r}_{i,-1} = \mathbf{r}_{i,0} \qquad i = 0, 1, \ldots, N$$
$$\mathbf{r}_{-1,j} = \mathbf{r}_{0,j} \qquad j = 0, 1, \ldots, M$$

.

Then using equations $\mathbf{r}(\frac{i}{N}, \frac{j}{M})$, $\mathbf{r}_{i,-1}$, and $\mathbf{r}_{-1,j}$ we can find that

$$\mathbf{r}_{0,0} = \mathbf{r}(0,0) = \begin{bmatrix} \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{bmatrix} \begin{bmatrix} \mathbf{r}_{-1,-1} & \mathbf{r}_{0,0} & \mathbf{r}_{0,1} \\ \mathbf{r}_{0,0} & \mathbf{r}_{0,0} & \mathbf{r}_{0,1} \\ \mathbf{r}_{1,0} & \mathbf{r}_{1,0} & \mathbf{r}_{1,1} \end{bmatrix} \begin{bmatrix} \frac{1}{6} \\ \frac{2}{3} \\ \frac{1}{6} \end{bmatrix}$$

$$\Rightarrow \quad \mathbf{r}_{0,0} = \frac{1}{3}(\mathbf{r}_{-1,-1} + \mathbf{r}_{0,1} + \mathbf{r}_{1,0} + \mathbf{r}_{1,1}) + \frac{1}{9}(\mathbf{r}_{0,0} + \mathbf{r}_{1,0} + \mathbf{r}_{0,0} + \mathbf{r}_{0,1}) + \frac{1}{3}\mathbf{r}_{0,0})$$

$$\Rightarrow \quad \mathbf{r}_{-1,-1} = 12\mathbf{r}_{0,0} - 5\mathbf{r}_{0,1} - 5\mathbf{r}_{1,0} - \mathbf{r}_{1,1}$$

Similarly,

$$\begin{aligned}
\mathbf{r}_{-1,M+1} &= 12\mathbf{r}_{0,M} - 5\mathbf{r}_{0,M-1} - 5\mathbf{r}_{1,M} - \mathbf{r}_{1,M-1} \\
\mathbf{r}_{N+1,-1} &= 12\mathbf{r}_{N,0} - 5\mathbf{r}_{N-1,0} - 5\mathbf{r}_{N,1} - \mathbf{r}_{N-1,1} \\
\mathbf{r}_{N+1,M+1} &= 12\mathbf{r}_{N,M} - 5\mathbf{r}_{N,M-1} - 5\mathbf{r}_{N-1,M} - \mathbf{r}_{N-1,M-1}
\end{aligned}$$

To solve for cubic B-Spline surfaces is to solve using cubic B-Spline curves twice. Solve in the horizontal direction first, then in the vertical (or vice versa). Here is an example to illustrate the cubic B-Spline for surfaces.

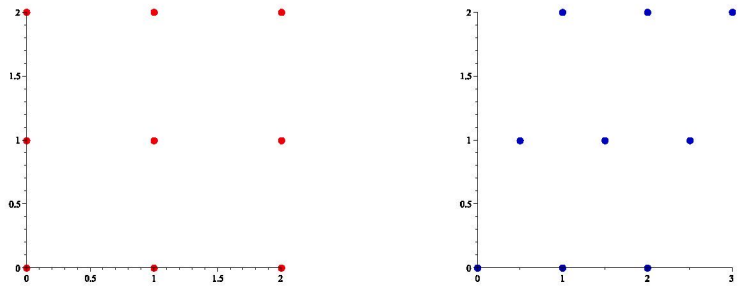

Figure 3.8: On the left is the source surface, and on the right is the target surface.

Example 3.

$$\mathbf{r}(u,v) = \sum_{i=-1}^{N+1}\sum_{j=-1}^{M+1} B(Nu-i)B(Mv-j)\mathbf{r}_{ij}$$

rewriting it the following way

$$\mathbf{r}(u,v) = \sum_{i=-1}^{N+1} B(Nu-i)[\sum_{j=-1}^{M+1} B(Mv-j)\mathbf{r}_{ij}\ ]$$

So, we let $Q_{ij} = \sum_{j=-1}^{M+1} B(Mv-j)\mathbf{r}_{ij}$ therefore,

$$\mathbf{r}(u,v) = \sum_{i=-1}^{N+1}\sum_{j=-1}^{M+1} B(Nu-i)B(Mv-j)\mathbf{r}_{ij}$$

then becomes

$$\mathbf{r}(u,v) = \sum_{i=-1}^{N+1} B(Nu-i)Q_{ij}$$

Fix $i = 0$:

$$\begin{bmatrix} Q_{-1,-1} \\ Q_{-1,0} \\ Q_{-1,1} \\ Q_{-1,2} \\ Q_{-1,3} \end{bmatrix} = \begin{bmatrix} Q_{0,-1} \\ Q_{0,0} \\ Q_{0,1} \\ Q_{0,2} \\ Q_{0,3} \end{bmatrix} = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & -2 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 2 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} -\frac{1}{6} & 0 \\ 0 & 0 \\ \frac{1}{6} & 0 \\ \frac{1}{3} & 0 \\ \frac{1}{2} & 0 \end{bmatrix}$$

Fix $i = 1$:

$$\begin{bmatrix} Q_{1,-1} \\ Q_{1,0} \\ Q_{1,1} \\ Q_{1,2} \\ Q_{1,3} \end{bmatrix} = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & -2 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 & 0 \\ \frac{1}{2} & 1 \\ \frac{3}{2} & 1 \\ \frac{5}{2} & 1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} -\frac{1}{12} & \frac{1}{6} \\ \frac{1}{12} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{6} \\ \frac{5}{12} & \frac{1}{6} \\ \frac{7}{12} & \frac{1}{6} \end{bmatrix}$$

Fix $i = 2$:

$$\begin{bmatrix} Q_{2,-1} \\ Q_{2,0} \\ Q_{2,1} \\ Q_{2,2} \\ Q_{2,3} \end{bmatrix} = \begin{bmatrix} Q_{3,-1} \\ Q_{3,0} \\ Q_{3,1} \\ Q_{3,2} \\ Q_{3,3} \end{bmatrix} = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & -2 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 & 0 \\ 1 & 2 \\ 2 & 2 \\ 3 & 2 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{3} \\ \frac{1}{6} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} \\ \frac{2}{3} & \frac{1}{3} \end{bmatrix}$$

So, now we know Q_i, for $i = -1, 0, 1, 2, 3$.

Now we would solve the other direction.

$$\mathbf{r}(u,v) = \sum_{j=-1}^{M+1} B(Mv-j)\mathbf{r}_{i,j}$$

Fix $j = -1$:

$$\begin{bmatrix} \mathbf{r}_{-1,-1} \\ \mathbf{r}_{0,-1} \\ \mathbf{r}_{1,-1} \\ \mathbf{r}_{2,-1} \\ \mathbf{r}_{3,-1} \end{bmatrix} = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & -2 & 1 \end{bmatrix}^{-1} \begin{bmatrix} Q_{-1,-1} \\ Q_{0,-1} \\ Q_{1,-1} \\ Q_{2,-1} \\ Q_{3,-1} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & -2 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & -2 & 1 \end{bmatrix}^{-1} \begin{bmatrix} -\frac{1}{6} & 0 \\ -\frac{1}{6} & 0 \\ -\frac{1}{12} & \frac{1}{6} \\ 0 & \frac{1}{3} \\ 0 & \frac{1}{3} \end{bmatrix} = \begin{bmatrix} -\frac{7}{48} & -\frac{1}{24} \\ 0 & 0 \\ -\frac{1}{48} & \frac{1}{24} \\ 0 & 0 \\ \frac{1}{48} & \frac{7}{24} \end{bmatrix}$$

Fix $j = 0$:

$$\begin{bmatrix} \mathbf{r}_{-1,0} \\ \mathbf{r}_{0,0} \\ \mathbf{r}_{1,0} \\ \mathbf{r}_{2,0} \\ \mathbf{r}_{3,0} \end{bmatrix} = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & -2 & 1 \end{bmatrix}^{-1} \begin{bmatrix} Q_{-1,0} \\ Q_{0,0} \\ Q_{1,0} \\ Q_{2,0} \\ Q_{3,0} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & -2 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & -2 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{12} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{3} \\ \frac{1}{6} & \frac{1}{3} \end{bmatrix} = \begin{bmatrix} -\frac{1}{48} & -\frac{1}{24} \\ 0 & 0 \\ \frac{1}{48} & \frac{1}{24} \\ 0 & 0 \\ \frac{7}{48} & \frac{7}{24} \end{bmatrix}$$

Fix $j = 1$:

$$\begin{bmatrix} \mathbf{r}_{-1,1} \\ \mathbf{r}_{0,1} \\ \mathbf{r}_{1,1} \\ \mathbf{r}_{2,1} \\ \mathbf{r}_{3,1} \end{bmatrix} = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & -2 & 1 \end{bmatrix}^{-1} \begin{bmatrix} Q_{-1,1} \\ Q_{0,1} \\ Q_{1,1} \\ Q_{2,1} \\ Q_{3,1} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & -2 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & -2 & 1 \end{bmatrix}^{-1} \begin{bmatrix} \frac{1}{6} & 0 \\ \frac{1}{6} & 0 \\ \frac{1}{4} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} \end{bmatrix} = \begin{bmatrix} \frac{5}{48} & -\frac{1}{24} \\ 0 & 0 \\ \frac{1}{16} & \frac{1}{24} \\ 0 & 0 \\ \frac{13}{48} & \frac{7}{24} \end{bmatrix}$$

Fix $j = 2$:

$$\begin{bmatrix} \mathbf{r}_{-1,2} \\ \mathbf{r}_{0,2} \\ \mathbf{r}_{1,2} \\ \mathbf{r}_{2,2} \\ \mathbf{r}_{3,2} \end{bmatrix} = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & -2 & 1 \end{bmatrix}^{-1} \begin{bmatrix} Q_{-1,2} \\ Q_{0,2} \\ Q_{1,2} \\ Q_{2,2} \\ Q_{3,2} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & -2 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & -2 & 1 \end{bmatrix}^{-1} \begin{bmatrix} \frac{1}{3} & 0 \\ \frac{1}{3} & 0 \\ \frac{5}{12} & \frac{1}{6} \\ \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} \end{bmatrix} = \begin{bmatrix} \frac{11}{48} & -\frac{1}{24} \\ 0 & 0 \\ \frac{5}{48} & \frac{1}{24} \\ 0 & 0 \\ \frac{19}{48} & \frac{7}{24} \end{bmatrix}$$

Fix $j = 3$:

$$\begin{bmatrix} \mathbf{r}_{-1,3} \\ \mathbf{r}_{0,3} \\ \mathbf{r}_{1,3} \\ \mathbf{r}_{2,3} \\ \mathbf{r}_{3,3} \end{bmatrix} = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & -2 & 1 \end{bmatrix}^{-1} \begin{bmatrix} Q_{-1,3} \\ Q_{0,3} \\ Q_{1,3} \\ Q_{2,3} \\ Q_{3,3} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & -2 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & -2 & 1 \end{bmatrix}^{-1} \begin{bmatrix} \frac{1}{2} & 0 \\ \frac{1}{2} & 0 \\ \frac{7}{12} & \frac{1}{6} \\ \frac{2}{3} & \frac{1}{3} \\ \frac{2}{3} & \frac{1}{3} \end{bmatrix} = \begin{bmatrix} \frac{17}{48} & -\frac{1}{24} \\ 0 & 0 \\ \frac{7}{48} & \frac{1}{24} \\ 0 & 0 \\ \frac{25}{48} & \frac{7}{24} \end{bmatrix}$$

Hence we can write the following $n \times m$, 5×5, matrix of coefficients $\mathbf{r}_{i,j}$:

$$\left[\ \mathbf{r}_{i,j}\ \right] = \begin{bmatrix} \mathbf{r}_{-1,-1} & \mathbf{r}_{-1,0} & \mathbf{r}_{-1,1} & \mathbf{r}_{-1,2} & \mathbf{r}_{-1,3} \\ \mathbf{r}_{0,-1} & \mathbf{r}_{0,0} & \mathbf{r}_{0,1} & \mathbf{r}_{0,2} & \mathbf{r}_{0,3} \\ \mathbf{r}_{1,-1} & \mathbf{r}_{1,0} & \mathbf{r}_{1,1} & \mathbf{r}_{1,2} & \mathbf{r}_{1,3} \\ \mathbf{r}_{2,-1} & \mathbf{r}_{2,0} & \mathbf{r}_{2,1} & \mathbf{r}_{2,2} & \mathbf{r}_{2,3} \\ \mathbf{r}_{3,-1} & \mathbf{r}_{3,0} & \mathbf{r}_{3,1} & \mathbf{r}_{3,2} & \mathbf{r}_{3,3} \end{bmatrix}$$

$$= \begin{bmatrix} (-\frac{7}{48}, -\frac{1}{24}) & (-\frac{1}{48}, -\frac{1}{24}) & (\frac{5}{48}, -\frac{1}{24}) & (\frac{11}{48}, -\frac{1}{24}) & (\frac{17}{48}, -\frac{1}{24}) \\ (0,0) & (0,0) & (0,0) & (0,0) & (0,0) \\ (-\frac{1}{48}, \frac{1}{24}) & (\frac{1}{48}, \frac{1}{24}) & (\frac{1}{16}, \frac{1}{24}) & (\frac{5}{48}, \frac{1}{24}) & (\frac{7}{48}, \frac{1}{24}) \\ (0,0) & (0,0) & (0,0) & (0,0) & (0,0) \\ (\frac{1}{48}, \frac{7}{24}) & (\frac{7}{48}, \frac{7}{24}) & (\frac{13}{48}, \frac{7}{24}) & (\frac{19}{48}, \frac{7}{24}) & (\frac{25}{48}, \frac{7}{24}) \end{bmatrix}$$

So to graph the 2D B-spline, we would graph the following: Picking out the x values from the $r_{i,j}$ we obtain:

$$\begin{aligned} x &= -\frac{7}{48}B(u+1)B(v+1) - \frac{1}{48}B(u+1)B(v) + \frac{5}{48}B(u+1)B(v-1) \\ &+ \frac{11}{48}B(u+1)B(v-2) + \frac{17}{48}B(u+1)B(v-3) - \frac{1}{48}B(u-1)B(v+1) \\ &+ \frac{1}{48}B(u-1)B(v) + \frac{1}{16}B(u-1)B(v-1) + \frac{5}{48}B(u-1)B(v-2) \\ &+ \frac{7}{48}B(u-1)B(v-3) + \frac{1}{48}B(u-3)B(v+1) + \frac{7}{48}B(u-3)B(v) \\ &+ \frac{13}{48}B(u-3)B(v-1) + \frac{19}{48}B(u-3)B(v-2) + \frac{25}{48}B(u-3)B(v-3). \end{aligned}$$

Similarly picking out the y values from the $r_{i,j}$ we obtain:

$$\begin{aligned}
y &= -\frac{1}{24}B(u+1)B(v+1) + \frac{1}{24}B(u-1)B(v+1) + \frac{7}{24}B(u-3)B(v+1) \\
&- \frac{1}{24}B(u+1)B(v) + \frac{1}{24}B(u-1)B(v) + \frac{7}{24}B(u-3)B(v) \\
&- \frac{1}{24}B(u+1)B(v-1) + \frac{1}{24}B(u-1)B(v-1) + \frac{7}{24}B(u-3)B(v-1) \\
&- \frac{1}{24}B(u+1)B(v-2) + \frac{1}{24}B(u-1)B(v-2) + \frac{7}{24}B(u-3)B(v-2) \\
&- \frac{1}{24}B(u+1)B(v-3) + \frac{1}{24}B(u-1)B(v-3) + \frac{7}{24}B(u-3)B(v-3)
\end{aligned}$$

Therefore, you would graph, (x, y) on the plane.

Chapter 4

Autocorrelation, Self Similarity and Deformations

Autocorrelation is a mathematical tool for finding repeating patterns such as the presence of a periodic signal which has been buried under noise, or identifying the missing fundamental frequency in a signal implied by its harmonic frequencies. It is used frequently in signal processing for analyzing functions or series of values, such as time domain signals. Informally, it is the similarity between observations as a function of the time separation between them. More precisely, it is the cross-correlation of a signal with itself. We are interested in it as a measure of self-similarity. Why do we care? For one, the values of function will tell us if we have near-regular

texture or not. If we do, the local maxima of the autocorrelation will help us detect what the periodic structure is. Chetverikov [3], takes a sequence of the local maxima values of the autocorrelation, and the maximum of the this sequence is denoted as the maximal regularity. The figure below (from [3, page 11]), displays patterns that are arranged in rows of four categories according to the maximal regularity: random [0,0.25), low regularity [0.25,0.50), medium regularity [0.50,0.75) and high regularity [0.75, 1.00].

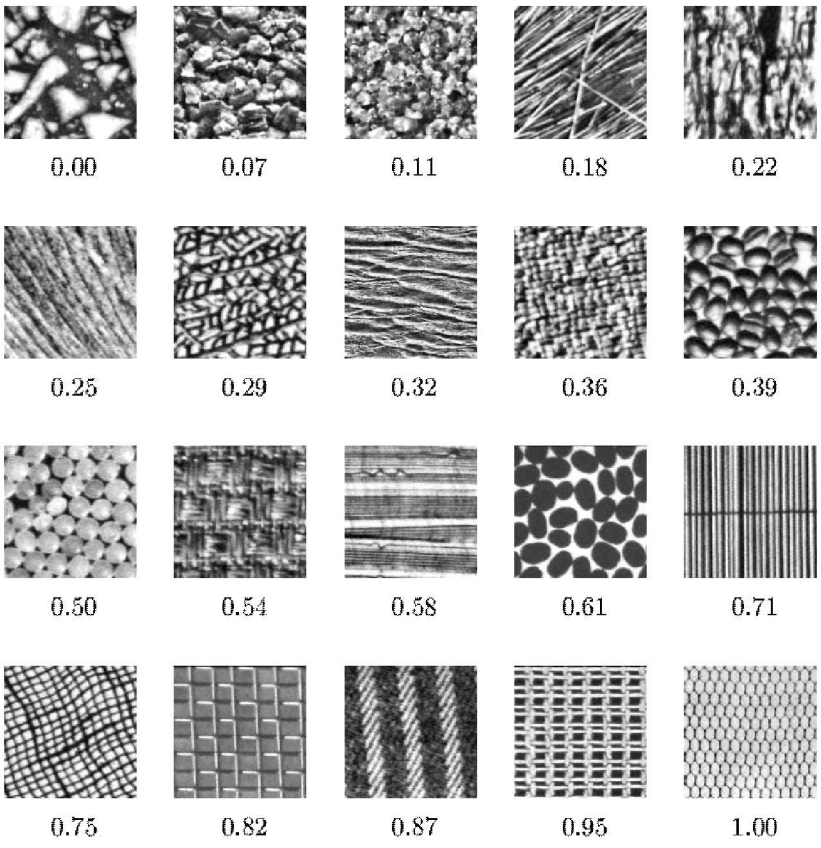

Figure 4.1: Patterns and their maximal regularity values. [3, page 11]

4.1 Background for Autocorrelation

First, consider the 1D setting.

Definition 6. The discrete normalized autocorrelation function $AC_I(d_x)$ is given by:

$$AC_I(dx) = \frac{\sum_{m=0}^{N-1} I(x_m)I(x_m + dx)}{\sum_{m=0}^{N-1} I^2(x_m)},$$

where $x_m \in D = \{x_0, x_1 + \Delta x, x_2 + \Delta x, \ldots, nx_{n-1} + \Delta x\}$, $I : D \mapsto \mathbb{R}$ and $I(x_m)$ is a discrete valued function. Also dx is a shift, the common difference in D and can be found by $dx = k\Delta x$, $k = 0, \ldots, n-1$. When $x_m + dx$ lies outside D, we extend $I(x_m + dx)$ in the standard way by gluing a copy to each end of the original $I(x_m)$.

Definition 7. For $f(t)$ a function defined for $t \in D$, a continuous domain, we define for $x \in \mathbb{R}$:

$$AC_f(x) = \frac{1}{\int_D f^2\, dt} \int_D f(t)f(t-x)\, dt$$

As in the discrete case, adding a shift x in the continuous autocorrelation may take you outside the domain, D, eventually. Again, we extend f periodically.

Note that the autocorrelation for discrete signals is just a discretization of the continuous autocorrelation. Let $I(x_m)$ be a sampling of the

continuous function $f(t)$. Then:

Example 4. Let $f(t) = \sin(t)$ and choose $D = \{-2\pi, \frac{-3\pi}{2}, -\pi, \frac{-\pi}{2}, 0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}, 2\pi\}$. Then $I(x_m) = \sin(x_m)$. Then possible values for dx are $dx = k \cdot \frac{\pi}{2}$ for $k = -4, \ldots, 4$. Using the trigonometric identity $\sin(x \pm y) = \sin(x)\cos(y) \pm \cos(x)\sin(y)$, we arrive to:

$$\begin{aligned}
AC_I(dx) &= \frac{\sum_{m=0}^{N-1} I(x_m)I(x_m + dx)}{\sum_{m=0}^{N-1} I^2(x_m)} \\
&= \frac{\sum_{m=0}^{8} \sin(x_m)\sin(x_m + dx)}{\sum_{m=0}^{8} \sin^2(x_m)} \\
&= \frac{\sin(x_0)\sin(x_0 + dx) + \cdots + \sin(x_8)\sin(x_8 + dx)}{\sin^2(x_0) + \sin^2(x_1) + \cdots + \sin^2(x_8)} \\
&= \frac{\sin(-2\pi)\sin(-2\pi + dx) + \cdots + \sin(2\pi)\sin(2\pi + dx)}{\sin^2(-2\pi) + \sin^2(\frac{-3\pi}{2}) + \cdots + \sin^2(2\pi)} \\
&= \frac{\sin(\frac{-3\pi}{2} + dx) - \sin(\frac{-\pi}{2} + dx) + \sin(\frac{\pi}{2} + dx) - \sin(\frac{3\pi}{2} + dx)}{(1)^2 + (-1)^2 + (1)^2 + (-1)^2} \\
&= \frac{4\cos(dx)}{4} \\
&= \cos(dx) = \begin{cases} 1 & dx = 0, 2\pi, 4\pi \ldots \\ 0 & dx = \frac{\pi}{2}, \frac{3\pi}{2}, \ldots \end{cases}
\end{aligned}$$

For the continuous case we will use the autocorrelation the following way:

$$AC_f(x) = \frac{1}{\int_D f^2 \, dt} \int_D f(t) f(t - x) \, dt$$

Let $f(x) = \sin(x)$. Then,

$$AC_f(x) = \frac{1}{\int_D f^2 \, dt} \int_D f(t) f(t - x) dt$$

$$
\begin{aligned}
&= \frac{1}{\int_{-2\pi}^{2\pi} \sin^2(x)} \int_{-2\pi}^{2\pi} \sin(t)\sin(t-x)dt \\
&= \frac{1}{\frac{1}{2}[x - \sin(x)\cos(x)]_{-2\pi}^{2\pi}} (\frac{1}{2}[t\cos(x) - \frac{1}{2}\sin(2t-x)]_{-2\pi}^{2\pi}) \\
&= \frac{2\pi\cos(x)}{2\pi} \\
&= \cos(x)
\end{aligned}
$$

When x corresponds to the values of $\{x_m + dx\}$

$$
\cos(x) = \begin{cases} 1 & x = 0, 2\pi, 4\pi \ldots \\ 0 & x = \frac{\pi}{2}, \frac{3\pi}{2}, \ldots \end{cases}
$$

Hence, in this case, we have shown that the discrete and the continuous autocorrelation agree on their intersection, and this is true in general. The autocorrelation is equal to 1 when $\sin(x)$ is equal to the shifted $\sin(x)$. When $x = dx$, the continuous autocorrelation is equal to the discrete autocorrelation. The autocorrelation is therefore capturing information about periodicity. We can use the autocorrelation to test the periodicity of an image. As we will learn below, the autocorrelation of a perfectly periodic function is 1 when the shift equals to the period, whereas the autocorrelation of a random pattern is 0. We see this in the following two theorems.

Theorem 1. For $f(t)$ a function on a continuous domain D and $p \in \mathbb{R}$ such that $f(t) = f(t-p)$, then $AC_f(p) = 1$.

Proof. We assume conventions of extending f as described in definition 7.

$$AC_f(x) = \frac{1}{\int_D f^2\ dt} \int_D f(t)f(t-x)\ dt$$

Then,

$$AC_f(p) = \frac{1}{\int_D f^2 dt} \int_D f(t)f(t-p)dt$$

using the fact that f is periodic, then $f(t) = f(t-p)$. Therefore,

$$\begin{aligned} \frac{1}{\int_D f^2\ dt} \int_D f(t)f(t-p)\ dt &= \frac{1}{\int_D f^2\ dt} \int_D f(t)f(t)\ dt \\ &= \frac{1}{\int_D f^2\ dt} \int_D f^2\ dt \\ &= 1 \end{aligned}$$

□

The following discrete process is "most" random possible.

Theorem 2. Suppose $I(x_m)$ is the realization of an independent identically distributed discrete random process, where

$$p(x_m = x) = \begin{cases} \frac{1}{2} & x = -1 \\ \frac{1}{2} & x = 1 \\ 0 & else \end{cases}$$

Then

$$E[AC_I(dx)] = \begin{cases} 1 & dx = 0 \\ 0 & dx \neq 0 \end{cases}$$

, where E denotes expected value.

Proof.

$$\begin{aligned} E(AC_I(dx)) &= E[\frac{\sum_{m=0}^{N-1} I(x_m)I(x_m+dx)}{\sum_{m=0}^{N-1} I^2(x_m)}] \\ &= \frac{\sum_{m=0}^{N-1} E[I(x_m)I(x_m+dx)]}{\sum_{m=0}^{N-1} E[I^2(x_m)]} \end{aligned}$$

Case 1: $dx = 0$:

$$\begin{aligned} E(AC_I(dx=0)) &= \frac{\sum_{m=0}^{N-1} E[I(x_m)I(x_m+0)]}{\sum_{m=0}^{N-1} E[I^2(x_m)]} \\ &= \frac{\sum_{m=0}^{N-1} E[I^2(x_m)]}{\sum_{m=0}^{N-1} E[I^2(x_m)]} \end{aligned}$$

$E[I^2(x_m)] = 1$, by definition of $I(x_m)$

$$p([I^2(x_m)]) = \begin{cases} 1 & I^2(x_m) = 1 \\ 0 & \text{else} \end{cases}$$

. Therefore,

$$\begin{aligned} E(AC_I(dx=0)) &= \frac{\sum_{m=0}^{N-1} E[I(x_m)I(x_m+0)]}{\sum_{m=0}^{N-1} E[I^2(x_m)]} \\ &= \frac{\sum_{m=0}^{N-1} E[I^2(x_m)]}{\sum_{m=0}^{N-1} E[I^2(x_m)]} \\ &= \frac{\sum_{m=0}^{N-1} 1}{\sum_{m=0}^{N-1} 1} \\ &= \frac{n}{n} \\ &= 1. \end{aligned}$$

Case 2: $dx \neq 0$:

Since, $dx \neq 0$ then $E[I(x_m)I(x_m + dx)] = E[I(x_m)]E[I(x_m + dx)]$ because of independence. So, using this fact we obtain:

$$\begin{aligned} E(AC_I(dx)) &= \frac{\sum_{m=0}^{N-1} E[I(x_m)I(x_m + dx)]}{\sum_{m=0}^{N-1} E[I^2(x_m)]} \\ &= \frac{\sum_{m=0}^{N-1} E[I(x_m)]E[I(x_m + dx)]}{\sum_{m=0}^{N-1} E[I^2(x_m)]}. \end{aligned}$$

By definition of $I(x_m)$ we find that

$E[I(x_m)] = (-1)p(-1) + (1)p(1) = (-1)(\frac{1}{2}) + (1)(\frac{1}{2}) = 0$. As a result,

$$\begin{aligned} E(AC_I(dx)) &= \frac{\sum_{m=0}^{N-1} E[I(x_m)I(x_m + dx)]}{\sum_{m=0}^{N-1} E[I^2(x_m)]} \\ &= \frac{\sum_{m=0}^{N-1} E[I(x_m)]E[I(x_m + dx)]}{\sum_{m=0}^{N-1} E[I^2(x_m)]} \\ &= \frac{\sum_{m=0}^{N-1} 0 \cdot E[I(x_m + dx)]}{\sum_{m=0}^{N-1} E[I^2(x_m)]} \\ &= 0. \end{aligned}$$

Hence, if the shift equals zero, then the autocorrelation is equal to zero. If the shift is not equal to zero, then the autocorrelation is not equal to zero. □

Because our interest is 2D images, we now define the autocorrelation for 2D. An image is an array of M rows by N columns, where $M \times N$ is the pixel size of the digital image and (M, N) is the gray value of the image.

Definition 8. The discrete normalized (two-dimensional) autocorrelation function $AC_I(dx, dy)$ is given by:

$$AC_I(dx, dy) = \frac{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(x_m, y_n) I(x_m + dx, y_n + dy)}{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(x_m, y_n) I(x_m, y_n)},$$

where $I : D \mapsto \mathbb{R}$ a discrete function $I(x_m, y_n)$, $x_m, y_n \in D \subset \mathbb{R}$, D is the domain of gray values, (dx, dy) is a shift, the common difference in D.

Definition 9. For a function $f(s, t)$ defined for $s, t \in D \subset \mathbb{R}$, a continuous domain, we define for $x, y \in \mathbb{R}$:

$$AC_f(x, y) = \int_D \int_D f(s, t) f(s - x, t - y) ds\, dt$$

4.2 Autocorrelation and periodic structures

We are interested in knowing what happens in between, in other words, in determining the autocorrelation for near-regular textures. The sensitivity of the autocorrelation to periodicity of functions allows us to determine the underlying grid periodicity. In other words, we can use the autocorrelation to find the period of a periodic function.

The following images illustrate the texture of a regular brick and a deformation of the brick. We look at what happens when we implement the autocorrelation to each image.

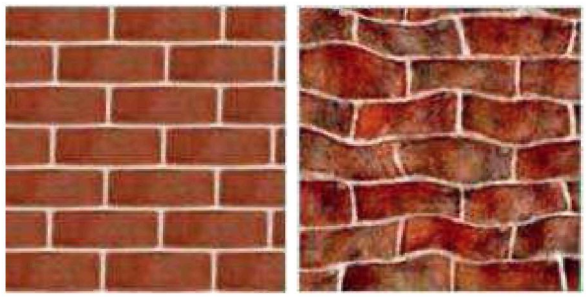

Figure 4.2: Regular bricks and deformed bricks texture.

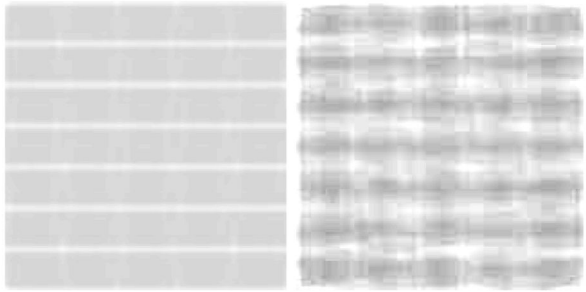

Figure 4.3: The images of the autocorrelation of the regular and deformed bricks.

Looking at the images above, we can easily see the white lines on the autocorrelation image of the regular brick. The white lines indicate the maximum values of the autocorrelation. Yet, when we focus our attention to the autocorrelation of the deformed texture, it is hard to see what is going on. The maxima are diffused because the deformation has disrupted the periodicity of the image. The maxima of the autocorrelation detecting the period for the original image are global maxima, yet for the deformed

image, they are local maxima. Therefore, we want to determine which of the local maxima correspond to the deformed period. One method used by Liu [4] is to take the maxima of the autocorrelation and find the region of dominance. The region of dominance is defined as the largest circle centered on the local maxima such that no other maxima are contained in the circle. What she does is takes all maxima and arranges them in descending order of maxima. Next, she computes the distances for each maxima j to each maxima i where $1 \leq i < j$ that comes before it in the list and denotes the minimum distance. Finally, she sorts the list of maxima again in descending order of the minimum distance to a higher maxima. The set of maxima are now arranged in decreasing order of dominance, where dominance is defined by distance to the next local maximum.

4.3 Autocorrelation and deformations

In order to more systematically understand the relationship between autocorrelation (image), autocorrelation (deformed image), we examine local maxima of the autocorrelation of the regular image to the local maxima of the autocorrelation of the deformed image. Begin with a simple case, where f is a function of one continuous variable case, and the deformation

is linear. Eventually, we would like to generalize the following result to B-spline deformations. So, as we did in earlier chapters we will start with linear deformations.

Theorem 3. Given a periodic function $f(t)$, and a deformation $\phi(t) = at$, for $a, t \in \mathbb{R}$, we define $g(t) = f \circ \phi = f(at)$. Then $AC_g(x) = AC_f(ax)$.

Proof.

$$\begin{aligned} AC_g(x) &= \frac{1}{\int_{-\infty}^{\infty} g^2 \, dt} \int_{-\infty}^{\infty} g(t)g(t-x) \, dt \\ &= \frac{1}{\int_{-\infty}^{\infty} f^2(at) \, dt} \int_{-\infty}^{\infty} f(at)f(a(t-x)) \, dt \end{aligned}$$

let $u = at$ then $\frac{1}{a} \, du = dt$ which implies

$$\begin{aligned} AC_g(x) &= \frac{1}{\int_{-\infty}^{\infty} f^2(at) \, dt} \int_{-\infty}^{\infty} f(at)f(a(t-x)) \, dt \\ &= \frac{1}{\frac{1}{a}\int_{-\infty}^{\infty} f^2(u) \, du} \frac{1}{a} \int_{-\infty}^{\infty} f(u)f(u-ax) \, du \\ &= \frac{1}{\int_{-\infty}^{\infty} f^2(u) \, du} \int_{-\infty}^{\infty} f(u)f(u-ax) \, du \\ &= AC_f(ax) \end{aligned}$$

□

We would like to move now to the piecewise linear case which is the next natural step to follow. Yet, since the autocorrelation is a global measure, it

is difficult to handle piecewise. We have to compare each piece with every other piece in our deformation.

Example 5. Say we have the following piecewise linear deformation. From $[0, a]$, $f(x)$ is 2-periodic and from $[a, b]$ its 4-periodic. Then, to find the autocorrelation we can use: $A(t) = \frac{1}{\|f\|^2} \int_0^b f(x)f(x+t)\ dt$. $A(t) = 1$ if and only if $f(x) = f(x+t)$ for every x. So now to find the autocorrelation of the piecewise function, we will do the following:

$$\begin{aligned} A(t) &= \frac{1}{\|f\|^2} \int_0^a f(x)f(t-x)\ dt + \frac{1}{\|f\|^2} \int_a^b f(x)f(t-x)\ dt \\ &= \frac{1}{\|f\|^2} \int_0^{a-t} f(x)f(t-x)\ dt + \frac{1}{\|f\|^2} \int_{a-t}^{b-t} f(x)f(t-x)\ dt \\ &+ \frac{1}{\|f\|^2} \int_{b-t}^b f(x)f(t-x)\ dt, \end{aligned}$$

where

$\frac{1}{\|f\|^2} \int_0^{a-t} f(x)f(x+t)\ dt$ is the 2-periodic; $\frac{1}{\|f\|^2} \int_{b-t}^b f(x)f(x+t)\ dt$ is the 4-periodic; $\frac{1}{\|f\|^2} \int_{a-t}^{b-t} f(x)f(x+t)\ dt$ is 1-periodic.

For a piecewise deformation with n pieces we will have $\binom{n}{2}$ distinct integrals.

As we increased the number of pieces, we saw that the combinations increased combinatorially. Therefore, we concluded that the autocorrelation may not be the most effective way to relate regular and deformed textures for a more general class of deformations such as B-splines which may have arbitrarily many pieces.

Chapter 5

Conclusion

Eyesight is one of the mysteries of science. How can we replicate it into machines (robots)? Scientists are still trying to find an answer that will bring to life an artificial visual system that would allow robots to process visual information like humans do. All of what we learned was modeling near-regular textures, and we learned of one tool that will allow us to detect near-regular textures, autocorrelation. As we saw with autocorrelation, when we introduced piecewise linear functions, the work was more involved, hence we need a better and faster tool than autocorrelation.

Ongoing work involves more tools for detection of near regular textures. One of the tools we are exploring is coefficients in orthonormal bases. Why? To see if we can obtain a relationship between the coefficients of function

f and the coefficients of $f \circ w$ (deformed function). The main focus is the exploration of fourier series and wavelets. Fourier series is an improvement over autocorrelation. It is very good in determining periodicity (frequency). Yet, there is a drawback. Fourier series gives us the frequency, yet it does not tell us when it happens in our original function. Therefore, we started to explore other tools to detect near regular textures. The wavelet transform, or wavelet analysis, is probably the most recent solution to overcome the problem of the fourier series. Not only does the wavelet transform give us scale (frequency), but it tells us the time (location where the frequency happens).

As we explore these tools more and more, we might one day arrive to the ultimate goal of the project: *to find mathematical measurements of textures that will categorize them as regular, stochastic, or in transitional.*

Bibliography

[1] I.L. Dryden, K.V. Mardia, J. Wiley, *Statistical Shape Analysis*, 1st edition, September 2, 1998.

[2] F.P. Preparata, M.I. Shamos, *Computational Geometry, and Introduction to Plane Curves*, Springer, August 6, 1993.

[3] D. Chetverikov, *Pattern Regularity as a visual Key*, Image and Vision Computing, Vol. 18, pg 2000, 1998.

[4] Y. Liu, R. Collins, Y. Tsin, *A computational Model for Periodic Pattern Perception based on Frieze and Wallpaper Groups*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, pg 354-371, 2004.