Vessel Data Taking Shape:
Using persistent homology to analyze images of tumor vasculature

Christopher DuBois

April 7, 2006
# Contents

1 Introduction  
2 Background  
   2.1 Scientific Background:  
   2.2 Vascular Imaging  
   2.3 Quantifying Tortuosity of Blood Vessels  
   2.4 Mathematical Background  
3 Theory - Simplicial Homology  
   3.1 The Simplex  
   3.2 The Simplicial Complex  
   3.3 Orientation  
   3.4 Chain Groups  
   3.5 The Boundary Map  
   3.6 The Homology Group  
   3.7 Interpreting Betti Numbers  
   3.8 Filtrations  
4 Method  
   4.1 Overview  
   4.2 Image Preprocessing  
   4.3 Thresholding Procedure  
   4.4 Creating the Simplicial Complex  
   4.5 Creating the Filtration and Computing Persistence  
   4.6 Calculating Persistence of $\beta_0$  
   4.7 Visualizing Persistent Homology  
5 Results  
6 Conclusion  
A Vector Spaces  
B List of Variables  
C Matlab Code for pMovie.m  
D Matlab Code for persist_getI.m  
E Matlab Code for persist_makeC.m
Abstract

One way to distinguish between surfaces is to find a topological invariant for which they differ. The $k$-th Betti number for a space $X$, $\beta_k(X)$, provides one such property; $\beta_1$, for example, represents (in a rough sense) the number of holes in a 2D object. In this thesis, I present the underlying theory and methods needed for calculating Betti numbers of a given simplicial complex (i.e., a construction using vertices, edges, and triangles). I also review an approach to studying how this quantity changes as we consider different subsets of this complex. Finally, I apply these methods to gain insight into the morphology of the blood vessels by analyzing SEM images of vasculature. My hope is to elucidate a clinically known difference between images vasculature in normal and cancerous settings. Preliminary results are included.
Chapter 1

Introduction

Blood-borne treatments such as chemotherapy depend on the ability of vessels to deliver drugs to wherever cancer cells are located in the body [10]. However, in contrast to normal tissue, vessels near tumors often have irregular branching, greater twisting, and widely variable intervascular distances [2] [12]. It is known that the higher vessel permeability stimulates the growth of new blood vessels [5], but many suspect the abnormal qualities of the tumor vasculature also inhibit effective drug delivery [10]. A method that quantifies this irregular morphology may help reveal how vascular geometry influences tumor growth or a particular treatment protocol. Some work has already been done in this direction [4].

For my thesis, I hope to provide a method that can distinguish between images of normal and tumor vasculature. To represent the blood vessels we use a simplicial complex (i.e. a collection of vertices, edges, and triangles). Using methods from abstract topology, we can compute topological invariants that show if two objects are fundamentally different. We hope to apply these techniques to reveal topological differences between tumor vasculature and normal vasculature.

First, I will motivate and present the problem at hand. After reviewing the relevant algebra and simplicial homology theory, I propose a method which requires the calculation of Betti numbers, a particular invariant. I will use algorithms [14] which are well suited for computations on simplicial complexes and analysis of simplicial homology. Finally, I include preliminary experimental results and evaluate the method’s ability to provide information that helps distinguish between images of tumor vasculature and normal vasculature.
Chapter 2

Background

2.1 Scientific Background:

Among the many diseases that alter the physical structure of blood vessels, tumor growth appears to have caught the attention of both clinicians and researchers. Aggressive tumors reach a size at which the nearby vasculature cannot support the tumor’s high rate of nutrient consumption. This triggers certain chemical factors (e.g. VEGF - vascular endothelial growth factor) to be released from the tumor cells that cause new microvessels to grow towards the tumor [9].

The blood vessels that surround malignant tumors are well known for their abnormalities. In 2.1 we have two SEM images comparing healthy tissue (left) to cancerous tissue (right).

![Figure 2.1: The vasculature in detail [11]. Left: normal tissue. Right: tumor tissue. Note the sharp offshoots where new vessels are growing. Also note the differences in regularity and intravascular voids.](image)

Experimental studies have shown the vessel walls are weaker, but the vasculature also appears more tangled and the vessels more twisted. Many suspect this tortuosity affects the local circulation and nutrient delivery. The growth of the tumor relies on the local vasculature, so anti-angiogenic treatments try to restrict the growth of new vessels in an attempt to restrict the nutrient flow to the tumor. On the other hand, blood-borne therapies (e.g. chemotherapy) also depend on the microvasculature since these chemicals must reach the tumor cells in order to be effective. Thus, a better understanding of vessel morphology could provide for better diagnosis, prognosis, and monitoring of treatments.
2.2 Vascular Imaging

There are various vascular imaging technologies that reveal the structural qualities of the blood vessels: Magnetic resonance imaging (MRI), computed tomography (CT), multi-photon microscopy, positron emission tomography (PET), and optical imaging can provide noninvasive images of vessels in living animals and humans, but do not have the resolution to capture the microvessels [11]. A few of these technologies have recently been used to construct 3-dimensional models of a tumor’s local vasculature [11]. For 2-dimensional information, scanning electron microscope (SEM) images of preserved tissue specimens offer unmatched resolution of the physical structure of blood vessels. For an analytical method to be useful in a clinical setting, however, relying on SEM images is infeasible whereas using MRI or a similar technology is far more practical. However, the resolutions of these technologies may improve to a point where an analysis of in vivo vessel structure may be both possible and useful.

2.3 Quantifying Tortuosity of Blood Vessels

The most widely used metric for quantifying angiogenesis is known as microvessel density (MVD) which is simply a count of the number of microvessels per area as seen in a microscope slide, for example. Because of the role vessel morphology may play on tumor growth and treatment, an quantitative estimation of the tortuosity of the vessel clusters may provide a superior measure of angiogenesis.

Bullitt [4] reviews several tortuosity metrics which have been proposed for analyzing 2-D images, e.g. counting the number of inflection points or finding the magnitude of curvature and normalizing by path length. Bullitt’s own research extends a related metric to 3-D vessel data in order to study the oscillations of the vessel coils.

In SEM images the tangled nature of tumor vasculature is apparent. In these clusters of high tortuosity, the tangled web of vessels creates a larger number of small voids whereas in images of normal vessels the intravascular regions are few and large. With appropriate image analysis, we could count the number of these regions to provide a rough measure of how tangled the vessels are. This paper provides a method for performing this analysis locally throughout the image in a way that may provide insight into the nature of blood vessel morphology, both in normal and malignant tissue.

2.4 Mathematical Background

Our proposed analysis requires counting the number of intravascular voids throughout a sample SEM image. Ideally, we would like an image that clearly depicts the structure of the blood vessels, e.g. an image with a network of thick black lines against a white background.

We are not nearly so lucky, however. The SEM images sometimes have uneven lighting and use shading to help illustrate depth. We will apply basic image processing techniques to help highlight the vasculature.

Once we have this black-and-white image of vasculature, we hope to count the number of white holes, regardless of the shape or size of the hole, formed by the network of black lines. In other words, we wish to count the holes in a two-dimensional surface that represents the blood vessels. I will employ methods from topology, the mathematical study of spaces and how those spaces behave as they are deformed. This is ideal since we also do not care about small imperfections or deformations of the holes in the space, but simply the number of holes, which is invariant with respect to these small changes.

The primary topological tool I will use is simplicial homology. I will represent the space as a large collection of simple geometric structures called simplices. I will represent these collections using vector spaces with which the counting of holes in a space can be reduced to linear algebra. After developing the theory necessary for this analysis, I present a method that implements these mathematical tools in order to analyze SEM images of vasculature.
Chapter 3

Theory - Simplicial Homology

In this section, I introduce the structures and theory necessary to count the holes in 2-dimensional spaces. If we hope to know generally about a particular space, it may be best to work independently from its geometric description. To do this, we require a structure that has no dependence on a distance function.

First, we define a building block (simplex) for building structures (simplicial complexes) that we will use to represent our spaces. These structures allow us to use tools from algebra to gain insight into the topology of our space.

This section borrows heavily from [13], [15] and [1]. We will begin with a few definitions.

Definition. Consider the set \( P = \{p_0, p_1, \ldots, p_k\} \subseteq \mathbb{R}^n \). An affine linear combination is a linear combination \( x = \sum_{i=0}^{k} \lambda_i p_i \) for \( \lambda_i \in \mathbb{R} \) where \( \sum_{i=0}^{k} \lambda_i = 1 \). If \( \lambda_i \geq 0 \) for each \( i \), we call it a convex combination. The set of all convex combinations is the convex hull. [15]

Definition. The set \( P \) is affinely independent if no element in \( P \) is an affine linear combination of the other elements in \( P \).

Figure 3.1: The points \( p_1, p_2, p_3 \) are affinely independent. The convex hull is the triangle formed by these three points.

Example. Consider the four points of \( \mathbb{R}^2 \) in the figure below. The points are not affinely independent since \( p_3 \) can be written as an affine combination of the other three elements, that is \( p_4 = (1/2)p_1 + (1/2)p_2 + (1/2)p_3 \). The convex hull of these points is shown in Figure 3.1.
3.1 The Simplex

We now can formally define the “building blocks” we will use to represent our spaces.

**Definition.** A simplex of dimension *k* (or *k*-simplex) is the convex hull of *k*+1 affinely independent points \( V = \{v_0, v_1, ..., v_k\} \), where \( v_i \) are called the vertices of the simplex. [14]

![Figure 3.2: Examples of k-simplices for small k. From left to right: a 0-simplex, 1-simplex, 2-simplex, and 3-simplex.](image)

In Figure 3.2 we have a few examples of simplices for small *k*, which we know intuitively as a vertex, an edge, a triangle, and a tetrahedron. We know that the faces of this tetrahedron are triangles. This idea can be formally extended to all simplices.

**Definition.** Let \( \rho \) be a *k*-simplex defined by \( R = \{p_0, p_1, ..., p_k\} \). A face of \( \rho \) is a simplex \( \rho' \) defined by a subset \( R' \subseteq R \). This relationship is denoted \( \rho' \leq \rho \). Note \( \rho \leq \rho \). [14]

**Example.** The above definition agrees with our intuition of the concept of face. Since \( \{a, b\} \subseteq \{a, b, c\} \), the 1-simplex defined by \( \{a, b\} \) is a face of the 2-simplex defined by \( \{a, b, c\} \) by definition, agreeing with what we have in Figure 3.2. Similarly, the 2-simplex \( \rho_1 \) defined by \( R_1 = \{a, b, c\} \) is a face of the 3-simplex \( \rho_2 \) defined by \( R_2 = \{a, b, c, d\} \) since \( R_1 \subseteq R_2 \).

3.2 The Simplicial Complex

If we “glue” several simplices together, we can create larger structures that we can use to represent a given space. We will formalize this idea with the following definition.

**Definition.** [14] A simplicial complex \( L \) is a finite set of simplices such that

1. \( \rho \in L, \rho' \leq \rho \Rightarrow \rho' \in L \)
2. \( \rho_1, \rho_2 \in L \Rightarrow \rho_1 \cap \rho_2 \leq \rho_1, \rho_2 \) or \( \rho_1 \cap \rho_2 = \emptyset \)

In other words, a simplicial complex is a finite collection of simplices where two restrictions are true: for every simplex in the collection each of its faces is in the collection, and whenever two simplices of the collection intersect they do so along a common face or not at all. [1]
Figure 3.3: We see the triangles on the left are properly joined because they share a face; this construction forms a simplicial complex. The triangle on the right intersects the edge but not along a shared face, so this construction does not form a simplicial complex.

Figure 3.4: An example of how we might choose to represent a space as a simplicial complex made of vertices, edges, and triangles.
Example. We hope to build a simplicial complex to represent a 2-dimensional space. For this purpose we need a configuration of simplices that approximates a given space accurately enough to capture its important features, such as holes in the space. Suppose we are given the space in Figure 3.4 (the region of large orange pixels at left). One strategy is to impose a grid on the space, placing vertices systematically throughout this grid. These vertices will guide the placement of our other vertices, and in so doing we are assured the complex will closely mimic the underlying space. More specifically, we will place edges between neighboring vertices; for each small square of 4 vertices, we connect one of the pairs of vertices located diagonally from each other, as shown in the center illustration in Figure 3.4. If we make each of the triangles formed in this construction a 2-simplex in our simplicial complex, we are left with the structure on the right of Figure 3.4, which roughly approximates the space as desired.

We can redefine these structures in a way that is completely free of geometry by relying on set notation.

Definition. An **abstract simplicial complex** is a set $K$, together with a collection $U$ of subsets of $K$, the abstract simplices of $K$, where

1. For all $x \in K$, $\{x\} \in U$. We call the sets $\{x\}$ the **vertices** of $K$.
2. If $\sigma' \subseteq \sigma \in U$, then $\sigma' \in U$. Again, we call $\sigma'$ a **face** of $\sigma$.

These alternate definitions prove to be more flexible. Throughout the rest of this thesis, I will use the term simplicial complex to refer to an abstract simplicial complex. I will also assume the collection $U$ is clear in context and will refer to the set $K$ as a complex.

### 3.3 Orientation

**Definition.** Let $K$ be a simplicial complex. An **orientated simplex** is a simplex equipped with an orientation. Let $\sigma$ be the $k$-simplex $\{v_0, v_1, ..., v_k\} \in K$. The oriented simplex $[\sigma] = [v_0, v_1, ..., v_k]$ is equivalence class of orderings of the vertices of $\sigma$, where

\[
(v_0, v_1, ..., v_k) \sim (v_{\tau(0)}, v_{\tau(1)}, ..., v_{\tau(k)})
\]

are equivalent orderings if the parity of the permutation $\tau$ is even.

Example. The oriented line segment in Figure 3.5 is denoted $[a, b]$. When added to the 1-chain $[b, a]$ (the same line segment oriented in the opposite direction), the two cancel each other: $[a, b] + [b, a] = [a, b] - [a, b] = 0$. This result is simply the empty set $\emptyset$.

Example. The triangle in Figure 3.5 $[a, b, c]$ has its orientation denoted by the arrow in the figure, and we have $[a, b, c] = [b, c, a] = [c, a, b]$. The other orientation $[c, b, a]$ is shown and we see that $[c, b, a] = [b, a, c] = [a, c, b]$. Note that we have explicitly written the two equivalence classes of orientations.
3.4 Chain Groups

In the Example 3.2, we found it useful to consider collections of simplices. To do this, we will construct a vectorspace that will represent this collection.

**Definition.** Let $K$ be a simplicial complex, and let

$$W = \{ [\sigma_1], [\tau_1], [\sigma_2], [\tau_2], \ldots, [\sigma_n], [\tau_n] \} \quad (3.4.1)$$

be the set of oriented $k$-simplices in $K$ where $\sigma_i = \tau_i$ but $[\sigma_i]$ has a different orientation than $[\tau_i]$. Letting $[\tau_i] = -[\sigma_i]$, we define the $k$th chain group as the vectorspace formed by the oriented $k$-simplices $[\sigma_i]$ as a basis, where $1 \leq i \leq n$. Note each $k$-simplex of $K$ can be viewed as an element of the $k$th chain group ($W \leq C_k$).

An element of $c \in C_k$ is called a $k$-chain, and can be written as a linear combination of the $k$-simplices $[\sigma_i]$ in the basis: $\sum_i n_i [\sigma_i], n_i \in \mathbb{R}$.

![Figure 3.6: The edges $[d, c], [c, b] \in C_1$ are highlighted blue and respectively. The sum of these two elements gives a 1-chain $[d, c] + [c, b] \in C_1$, shown in purple.](image1)

![Figure 3.7: The 1-chains $[a, d] + [d, c] + [c, e], [e, c] + [c, b] \in C_1$ are highlighted blue and red respectively. The sum of these two elements gives a 1-chain $[a, d] + [d, c] + [c, b] \in C_1$, shown in purple.](image2)

**Example.** The 1-simplices $[a, d], [d, c], [c, e]$ are a few of the basis elements of $C_1$ for the simplicial complex in the left of Figure 3.6. Two examples of 1-chains are shown (blue and red), as well as their sum in purple.
These 1-chains are elements of the vectorspace $C_1$ for this complex. A similar example is shown in Figure 3.7.

### 3.5 The Boundary Map

We are now able to define an important tool in algebraic topology that lets us explore the relationship between two adjacent dimensions in a simplicial complex, e.g. between the second chain group and the first chain group. We define a map between chain groups that follows our intuition of the boundary of these structures.

**Definition.** Let $K$ be a simplicial complex and let $\sigma = [v_0, ..., v_k] \in K$. The boundary homomorphism $\partial_k : C_k \to C_{k-1}$ is

$$\partial_k \sigma = \partial_k[v_0, ..., v_k] = \sum_{i=0}^{k} (-1)^i[v_0, ..., \hat{v_i}, ..., v_k]$$

(3.5.1)

where the symbol $\hat{v_i}$ means the vertex $v_i$ is to be deleted from the array. [13]

**Example.** Consider the simplices in Figure 3.8. Using the above definition to compute the boundary of each, we have

\[
\begin{align*}
\partial_1[a, b] &= [b] - [a] \\
\partial_2[a, b, c] &= [b, c] - [a, c] + [a, b] \\
&= [b, c] + [c, a] + [a, b] \\
\partial_3[a, b, c, d] &= [b, c, d] - [a, c, d] + [a, b, d] - [a, b, c] \\
&= [b, c, d] + [a, d, c] + [a, b, d] + [a, c, b]
\end{align*}
\]

We see the results geometrically in the top half of Figure 3.8.

**Example.** We can take the boundary of each of the 1-chains in Figures 3.6 and 3.7. For example, we can compute the boundary of the 1-chain on the left of figure 3.6 as

$$\partial_1[d, c] = [c] - [d]$$

which is exactly what is shown on the left of Figure 3.9. Similarly, if we compute the boundary of the 1-chain on the right of 3.7, we find

$$\partial_1[a, d] + [d, c] + [c, b] = [d] - [a] + [c] - [d] + [b] - [c]$$

$$= [b] - [a]$$

just as is shown in Figure 3.10.

Note that these boundaries are elements of the vectorspace $C_0$. Also, note that adding the boundaries of two 1-chains gives us the boundary of the sum of those 1-chains, as is illustrated in Figures 3.9 and 3.10.

**Example.** Consider $c = \partial_2[v_0, v_1, v_2]$. Note that $c$ is a 1-chain (that is, $c \in C_1$) because it is the sum of the oriented simplices $[v_0, v_1]$, $[v_1, v_2]$, and $[v_2, v_0]$. Also note that the boundary of this 1-chain is 0 since

$$\partial_1([v_0, v_1] + [v_1, v_2] + [v_2, v_0]) = ([v_1] - [v_0]) + ([v_2] - [v_1]) + ([v_0] - [v_2]) = 0$$

(3.5.2)

This result is expected because this 1-chain is a cycle and thus has no boundary.
Figure 3.8: For a $k$-simplex $\sigma_k$, an illustration of the boundary map and the fact $\partial_k \partial_{k+1} \sigma_k = 0$. Left: $\partial_0 \partial_1 \sigma_1 = 0$. Center: $\partial_1 \partial_2 \sigma_2 = 0$. Right: $\partial_2 \partial_3 \sigma_3 = 0$
Figure 3.9: Boundaries of the respective 1-chains from Figure 3.6 and their sums. The colored vertices depict the vertices which are boundaries of the respective 1-chain. Note that + and - signify the orientation of that vertex.

Figure 3.10: Boundaries of the respective 1-chains from Figure 3.7 and their sums. Note the boundary of the sum of 1-chains is the same as the boundary of the sum: \((-a + e) + (-e + b) = b - a\), which is the boundary of the sum of those 1-chains.
In the example above, we have that \( \partial_1 \partial_2 [v_0, v_1, v_2] = 0 \). We may generalize this notion with the following lemma.

**Lemma 3.5.1.** \( \partial_k \partial_{k+1} = 0 \) for all \( k \).

**Proof.** For an arbitrary \( k \)-simplex \( [v_0, v_1, ..., v_k] \).

\[
\partial_k \partial_{k+1} [v_0, ..., v_k] = \sum_{i=0}^{k+1} (-1)^i \partial_k [v_0, v_1, ..., \hat{v}_i, ..., v_{k+1}] \\
= \sum_{i=0}^{k+1} (-1)^i \sum_{j=i+1}^{k+1} (-1)^{j-1} [v_0, ..., \hat{v}_i, ..., \hat{v}_j, ..., v_{k+1}] + \\
\sum_{i=0}^{k+1} (-1)^i \sum_{j=0}^{i-1} (-1)^j [v_0, ..., \hat{v}_i, ..., \hat{v}_j, ..., v_{k+1}] \\
= 0
\]

Each oriented \( (k-1) \) simplex \( [v_0, \hat{v}_i, ..., \hat{v}_j, ..., v_{k+1}] \) appears twice in the sum above, first with sign \( (-1)^{i+j-1} \) and then with sign \( (-1)^{i+j} \). Thus, the expression sums to zero. \( \square \)

In the example above, note that \( c \in \ker \partial_1 \). Going a step further, we see that the kernel of \( \partial_1 \) is in fact those 1-chains in \( C_1 \) that form cycles; we denote this subspace by \( Z_1 \). We define this space for all \( k \) below.

**Definition.** The \( k \)th cycle group of the simplicial complex \( K \) is \( Z_k = \ker \partial_k \). A chain that is an element of \( Z_k \) is a \( k \)-cycle.

The above lemma lets us explore the subgroup of \( Z_k(K) \) that is the boundary for \( (k+1) \)-chains in \( K \). Let us define this formally.

**Definition.** The \( k \)th bounding group of \( K \) is \( B_k = \operatorname{im} \partial_{k+1} \). A chain that is an element of \( B_k \) is a \( k \)-boundary.

From the definition of kernel and image in Appendix A, we know \( Z_k \leq C_k \) and \( B_k \leq C_k \). Since bounding cycles are cycles, we also note that \( B_k \leq Z_k \).

### 3.6 The Homology Group

Consider the simplicial complex in Figure 3.4. How might we systematically find the number of white holes present? The number of white holes is precisely the number of closed curves that trace the edge of these features. However, in practice we do not attempt to trace the edges of holes. Instead we use an approach which is algebraically more convenient. Our strategy is as follows: consider all the closed curves, and consider to be equivalent any pair of curves which enclose the same hole (or holes). In this way, we arrive at the number of features in our 2-dimensional space.

The cycles we desire are 1-chains: they are elements in the chain group \( C_1 \). Examples of 1-chains for the complex in question are shown in Figure 3.11. We will reject elements of this vector space until we are left with a set which represents the features we are looking for. We first reject the blue and green 1-chains because they are not cycles, leading us to consider 1-chains from the vectorspace \( C_1 \) like those in Figure 3.12. These cycles have no boundary, so they are elements of \( \ker \partial_1 = Z_1 \).

Which of the cycles in Figure 3.12 capture features? We see the pink and brown 1-chains do not surround a feature since they form the boundary of a collection of triangles (ie. a 2-chain), so we will reject them. Using our terminology defined above, the pink and brown 1-chains are elements of \( \operatorname{im} \partial_2 = B_1 \), and are appropriately called bounding cycles.

However, if our goal is to count the number of features, we must consider all elements in \( Z_1 \) which enclose a feature to be equivalent. This can be done by examining whether or not they differ by some piece of the
Figure 3.11: The above 1-chains are elements of $C_1$ for this simplicial complex. We are looking for cycles which enclose holes, so we reject the blue and green 1-chains. Note that the pink chain is a cycle because it forms a loop, and is thus an element of $Z_1$.

Figure 3.12: The 1-chains shown above are all cycles, and thus elements of $Z_1$ for this complex. The pink and brown cycle above do not surround features in the simplicial complex shown since they are boundaries of 2-chains, which have each been highlighted in orange. The other three cycles do, however, surround holes in the complex.
Figure 3.13: The pink cycle above is a bounding cycle (i.e., an element of $B_1$). The difference between the red and blue cycles also is a 1-chain in the subspace $B_1$ since it forms a boundary for the highlighted 2-chain on the right. We construct the quotient space $H_1 = Z_1/B_1$ in which the 1-chain formed by all three cycles shown is considered equivalent to the cycle tracing just the edge of the right hole.

surface. For example, we identify the red and blue cycles as equivalent because the difference of these two cycles is the boundary of a 2-chain. Algebraically they represent the same feature.

In doing this, we are considering a subspace of cycles where the pink cycle is a zero element and the difference of the red and blue cycles is also a zero element. In this subspace, all 1-chains which are bounding cycles are zero elements.

This equivalence class structure is captured by the quotient group $H_1 = Z_1/B_1$. For example, in Figure 3.13 there is one element in this space for the feature which both the red and blue cycles surround. Furthermore, there is a separate element in $H_1$ for the feature the green cycle surrounds in Figure 3.12.

This inspires the following definition.

**Definition.** We define the $k$th homology space of $K$ as the quotient space

$$H_k = Z_k/B_k$$

(3.6.1)

$H_k$ is a vector space which is generated by cycles surrounding each of the holes. As such, we count the holes by taking the dimension of $H_k$. Above, we noticed the dimension of $H_1$ was the number of holes in our space. This is the invariant we have been waiting for.

**Definition.** [15] The $k$-th Betti number $\beta_k$ of a simplicial complex $K$ is $\dim H_k$.

From A.0.2 we have that $\beta_k = \dim H_k = \dim Z_k - \dim B_k = \dim \ker \partial_k - \dim \text{im} \partial_{k+1}$. This will prove very helpful when we want to compute $k$th Betti numbers of a simplicial complex.

For the interested reader, much of this machinery can be thought of in the context of groups instead of vector spaces, where $Z_k$ is a subgroup of the chaingroup $C_k$. The homology group is the quotient group $H_k = Z_k/B_k$. The above structures can also be generalized using modules, where we work over rings instead of a field as I have done above.

### 3.7 Interpreting Betti Numbers

In $\mathbb{R}^2$, the only relevant Betti numbers are $\beta_0$ and $\beta_1$. Both have intuitive meanings for a space $X$: $\beta_0$ measures the number of connected components and $\beta_1$ counts the number of holes. Consider the space shown in Figure 3.14. For this space, $\beta_0 = 1$ and $\beta_1 = 2$. 

15
We can compute Betti numbers, a particular topological invariant, for the above 2-dimensional spaces; $\beta_0$ counts the number of components and $\beta_1$ counts the number of holes. Left: This space (call it $X$) has Betti numbers $\beta_0 = 1$ and $\beta_1 = 2$. Right: The complement of the space ($X^c$) has Betti numbers $\beta_0 = 3$ and $\beta_1 = 1$.

We can also find the Betti numbers of the complement, $X^c$, as shown in Figure 3.14: $\beta_0 = 3$ and $\beta_1 = 1$. It is visually apparent that these Betti values must be closely related to the Betti values for the original space $X$. We note that in the example above $\beta_1(X) = \beta_0(X^c) - 1$. We will use this fact in Section 4.

### 3.8 Filtrations

We want to be able to explore how this invariant changes as we consider larger portions of the complex surrounding a given vertex. By defining an ordering for how we construct the simplicial complex around a given vertex, we can compute Betti numbers for the space as it grows. This idea is called a *filtration*.

**Definition.** [16]. For a topological space $X$, a filtration $\{X^n\}_{n \geq 0}$, is a nested sequence of subspaces: $\emptyset = X^0 \subseteq X^1 \cdots \subseteq X^n = X$. Here we call $X$ a filtered space.

We could filter a simplicial complex in the same way.

**Definition.** A filtration of a complex $K$ is a nested subsequence of complexes $\emptyset = K^0 \subseteq K^1 \subseteq \cdots \subseteq K^n = K$. We call $K$ a filtered complex.

We should consider this filtered complex like a growing space, one where new simplices appear until the entire complex is present. In this view, topological attributes will appear and disappear. Persistent homology theory [8] provides the tools to track attributes such as $\beta_k$ throughout a filtration and calculate its lifetime: when in the filtration the attribute began and when it ceased.
Chapter 4

Method

Our goal is to analyze SEM (Scanning Electron Microscope) images of tumor vasculature. These images provide information about the 3-dimensional structure of the vessels. After some image preprocessing, as described below, we will be able to create a simplicial complex that represents a two dimensional image of the vessel structure. From this complex we will explore how $\beta_0$ differs for small neighborhoods throughout the image, as well as how this value varies as we increase the radii of the neighborhood under consideration. In doing so, any persistence in the homology data may provide insight into the topological structure of the image.

4.1 Overview

We plan to count the number of intravascular voids in two given SEM images for different sized neighborhoods for each pixel throughout the image. We hypothesize this may provide a rudimentary measure for how tangled the vessels are and how this complexity varies throughout the image. This information may be able to describe a certain quality of vessel or a certain prognosis.

![Figure 4.1: Left: Original SEM image of vasculature in normal tissue, Right: SEM image of vasculature in tumor tissue](image)

The counting of intravascular voids is equivalent to counting the number of holes in the 2D space formed by the intersecting network of blood vessels. We could represent this 2D space with pixels whose locations are determined by the image of blood vessels. For each pixel in the SEM image which we decide represents a piece of the blood vessel, we place a pixel in the corresponding location of our 2D space. Deciding whether or not a pixel is part of a blood vessel requires thresholding and segmentation techniques that will be covered.
in Section 4.3. Thresholding techniques will make this decision by considering the value of the hue of each pixel, whereas segmentation techniques utilize the relationship between the hue values of a pixel and those of its neighbors.

To count the holes in this space, we can take advantage of the tools we developed in Chapter 3. The generality of homology theory is attractive for our application in particular since we do not care about the size or shape of the holes, only the quantity. Thus, we would like to be able to compute the Betti numbers of the space. In order to perform these computations we will first create a simplicial complex to represent the space: there will be a vertex for every pixel in our space and an edge connecting vertices whose corresponding pixels are within a certain distance. This information can be represented as a graph, and the boundary operator can be represented as a matrix. As we saw in Chapter 3, computing the Betti numbers only requires calculating the rank of two boundary maps in matrix form, $\partial_k$ and $\partial_{k+1}$.

This construction allows us to explore the progression of the Betti numbers at a pixel as we consider more of the surrounding complex. Lastly, we will use algorithms well suited for computing and analyzing this persistence in the homology information at a given location.

An overview of this methodology is given in Figure 4.2.

### 4.2 Image Preprocessing

The SEM images are uncompressed TIFF files, and since our method requires a black-and-white image, we need to convert the image to greyscale. We also are hoping to discern the regions of the image representing blood vessels, so we increase the brightness and contrast of the images in order to emphasize the difference between vessel and the background. Furthermore, we want the blood vessels to be black, so we simply invert the image. All of these functions can be performed using a standard image manipulation software, such as GIMP. This progression is shown for a sample SEM image in Figure 4.3.

### 4.3 Thresholding Procedure

Each SEM image is imported into MATLAB as a matrix. We perform segmentation techniques to obtain an image that has binary values: the image vasculature will be shown by darkened pixels, which will also be the vertices of our simplicial complex.

Our greyscale SEM image `tumorvessels.tif` is in uncompressed TIF format. We import into MATLAB by

```matlab
>> A = imread('tumor_vessels','tif');
>> size(A)
ans =
    [ 300  400 ]
```

The image is now described by a matrix $A$ containing values between 0 and 1 corresponding to the intensity of the respective pixel, e.g. 0 calls for a black pixel, 1 calls for a white pixel. The size of this matrix corresponds to the number of pixels in the image, which had dimensions of 300 pixels by 400 pixels.

For our procedure, we want to alter this image so that the blood vessels are black and the rest of the image is white. One way to produce a binary image $A^*$ (one which has only two values - black and white) is to consider some threshold value, $\gamma$, and let

$$A_{ij}^* = \begin{cases} 
1 & \text{for } i, j \text{ st. } A_{ij} > \gamma \\
0 & \text{for } i, j \text{ st. } A_{ij} \leq \gamma 
\end{cases}$$

An example of this procedure is shown in 4.4.

While this method may be preferred for its simplicity, the resulting image often fails to distinguish features in dark or light regions. For example, dark features will be ignored and light features will be overemphasized. Consider the upper left regions of the two images in Figure 4.4. The vessels and intravascular voids have
not been properly recognized because the local region is too dark for this method to distinguish vessels in
the image.

There are many ways we might try to choose which pixels should represent blood vessels. To detect
objects in an image, one might use information about the intensity of surrounding of pixels to group or
“cluster” pixels that are part of the same object. We will use the k-means algorithm, a type of clustering
algorithm commonly used for image analysis. In this algorithm, each pixel is compared with the surrounding
pixels and put in a particular group according to some measure; the number of groups is defined by a user-
defined parameter \( \alpha \). For more information on the algorithm, see (Forsyth and Ponce p.317 Algorithm 14.5).
We use the algorithm as implemented in MATLAB by Michael Lazzareschi and Daniel Barcay [3].

Initial testing suggested that setting \( \alpha = 4 \) provided the most accurate depiction of the image and best
improvement over rudimentary thresholding techniques like the one outlined above. This technique is most
helpful in darker regions where vasculature information had previously ignored by the \( \gamma \) threshold.

The results of using this method on the original SEM image in Figure 4.1 are shown on the left of Figure
4.4. Note the difference in the upper left hand corner: the segmentation algorithm is able to discern the
blood vessels that we recognize in Figure 4.1, but are not present in the \( \gamma \) thresholded image.

The cluster of pixels which depicted the intervascular areas was chosen to be white, and the remaining
two clusters were chosen to be black. This provided a binary image with the vessels distinctly shown. In
matrix form, this image has 300 rows and 400 columns and each element is either 0 or 1, where each 1
represents one of the darkened pixels that form the blood vessels.

### 4.4 Creating the Simplicial Complex

We want to create a simplicial complex of vertices and edges whose structure mimics how the vasculature
as seen in our thresholded image. We can represent the darkened pixels in our thresholded image as a set of
a \( M \) points \( x_1, x_2, \ldots, x_m, \ldots, x_M \in \mathbb{R}^2 \). We consider these points to be lying on the subspace \( X \subset \mathbb{R}^2 \) that
represents the blood vessels in the image. We can find and store the locations of our pixels by finding the
subscripts of matrix components which are 1.

```matlab
>> [I,J] = ind2sub(size(A),find(A));
>> x = [I,J]';
```

Our dataset \( x \) is a \( 2 \times M \) matrix of subscripts where \( x_{mi}, x_{mj} \) are the row and column of the datapoint
\( x_m \), respectively.

We will construct a simplicial complex with a vertex for each point \( x_m \) and a 1-cell for each set \( \{x_{m1}, x_{m2}\} \)
such that \( ||x_{m1} - x_{m2}|| < \epsilon \). Note that distance is measured from center to center for each pixel in units of
pixels. Thus, our simplicial complex includes those edges which connect vertices that are within \( \epsilon \) distance
of each other. We denote this complex \( K_\epsilon \).

In order to represent this complex we require specialized data structures. Plex [6] is a collection of
MATLAB routines which provides a data structure for representing simplicial complexes and routines for
various related computations. One of these routines (px_euclid) allows us to find the distances between the
vertices of our data, \( x_m \). With this we decide which vertices are within \( \epsilon \); we will create our edges from this
information. This complex \( K_\epsilon \) can be represented in the form of a graph.

The px_euclid function, when provided one argument of size \( 2 \times M \), returns a \( M \times M \) distance matrix
where \( M_{ij} = ||x_i - x_j|| \). We use this function and store the resulting \( M \times M \) matrix as \( D \).

```matlab
>> D = px_euclid(x);
```

Defining \( \epsilon = 1.5 \), we can create a matrix representation of the graph of \( K_\epsilon \) by placing a 1 in components of
\( D \) with value less than \( \epsilon \) and placing a 0 where values are greater than \( \epsilon \). We do this by

```matlab
>> epsilon = 1.5;
>> GRAPH = (D < epsilon);
```
Our choice for $\epsilon$ will create a graph with connections between any datapoints that are immediate neighbors of each other (i.e. each is located in one of the nine surrounding components of the other).

To create a plex object as defined by the Plex library from this information, we use the `plex` function. In order to 1-dimensional simplicial complex with edges defined by the 1’s in `GRAPH`, we use the command

$$\text{>> } K = \text{plex}(`\text{GRAPH}`);$$

where $K$ is the plex object which contains information about the simplicial object $K(\epsilon)$. For example we can query the object $K$ for information about the simplicial complex it represents by

$$\text{>> display}(K)$$

1-dimensional simplicial complex

$$[1333 \text{ 4214}]$$

where the output vector is the number of vertices and edges, respectively.

This construction is known as the Rips complex for a set of points in Euclidean space. To construct a Plex object $K$ in the above way, we could instead just the command

$$\text{>> } [K,F] = \text{px_rips}(x,1,\text{epsilon});$$

where $K$ is the Plex object returned by the function. This object contains all the necessary information to describe the simplicial complex $K(\epsilon)$.

### 4.5 Creating the Filtration and Computing Persistence

If we chose, we could consider only simplices in the complex $K$ which contain vertices within a distance $r$ of a particular vertex, refering to this subcomplex as $K_r$. Note that we have the inclusion $K_r \rightarrow K_{r'}$ whenever $r < r'$. By definition this provides a filtered complex for every increasing sequence of non-negative distances $r \in \mathbb{R}$. That is, we will be producing a nested subsequence of complexes and examine the sequence $\beta_0(K^1_r), \ldots, \beta_0(K^n_r)$.

To calculate $\beta_0$ for $n$ different $r$ at a given component $(i,j)$, we first calculate a matrix of distances from this pixel to all our vertices $x$ using the command

$$\text{>> } D = \text{px_euclid}(x,[i;j]);$$

We will use these distances to govern our filtration: Simplices “close” to component $(i,j)$ will be included in early, smaller complexes. To induce these filtration values over our simplicial complex $K$ we utilize the command

$$\text{>> } G = \text{extend}(C,\{D'\},\text{dim});$$

The last parameter governs the maximal dimension `dim` for this operation, e.g. for `dim = 1` the script only considers 0-cells and 1-cells. The result is a filtered simplicial complex.

### 4.6 Calculating Persistence of $\beta_0$

Plex can compute the persistent homology of a persistence complex. The output is a set of intervals, each describing “when” a particular feature is found as the parameter $r$ is varied. We will use this interval data to calculate $\beta_0$ by counting the number of features at various $r$.

$$\text{>> } [\text{Int\_Data}] = \text{persistence}(K,\text{GRAPH});$$

$$\text{>> } Z = \text{Int\_Dat\{1\};}$$

The $Z$ variable has two rows: the first contains the values of $Z_i^{\text{start}} = \epsilon$ for which each component was first found, and the second contains values of $Z_i^{\text{end}} = \epsilon$ for which the respective component no longer existed in simplicial complex.

To calculate $\beta_0$ at $\epsilon = r$ we count the number of $i$ for which $Z_i^{\text{start}} < r < Z_i^{\text{end}}$. 

20
4.7 Visualizing Persistent Homology

Creating the simplicial complexes is the most computationally intensive part of our methodology. Therefore, we will divide our image into small regions of $5 \times 5$ pixels and calculate $\beta_0$ for each $(i, j)$ and a range of $k$ from a simplicial complex created from a neighborhood of $35 \times 35$ centered at this region. This scheme is illustrated in 4.5 below, where the maximum radius considered is $r = 15$. The highlighted region at left is the selected neighborhood and the highlighted region at center is the pixel, whose values for $\beta_0$ for various $r$ are given at right.

For a thresholded image of size $300 \times 400$ pixels, this scheme will allow us to only need to calculate $(270/5) \times (370/5) = 3996$ simplicial complexes, rather than $270 \times 370 = 99900$ if we calculated a simplicial complex for the neighborhood of each $(i, j)$ we planned to analyze.

The result is a matrix of size $N_1 \times N_2 \times k = 300 \times 400 \times 15$ where each element in layer $k$ represents the number of $\beta_0$ components found for each pixel within a radius between $k$ and $k - 1$. In this format, Matlab allows for easy visualization, in the form of a movie. This movie depicts how local homology information progresses throughout the image as $k$ increase from 1 to 15.
## Outline of Method

1. Begin with an SEM Image of Microvasculature  
   *Either from normal tissue or tumor tissue*

2. Image Preprocessing
   - **Concert to Greyscale**
   - **Adjust Brightness/Contrast**
   - **Segmentation Threshold**  
     Use region-finding algorithm to make a black-and-white version of greyscale image.

3. Computing Homology
   - For each 5x5 square of pixels in the segmented image:
     - Construct a simplicial complex from the surrounding 35x35 neighborhood
       - **Vertices created for each pixel**
       - **Edges created between vertices within distance of 1.5 pixels**
   - For each pixel in this focus region
     - Compute persistent homology:
       - *Find Betti 0 for subcomplex within given radius*
       - Save this Betti 0 for each pixel and each radii

4. Finish by Displaying Homology Information
   - For each radii,
     - Display an image whose color values represent the Betti 0 values at each pixel.

---

Figure 4.2: An overview of our methods.
Figure 4.3: Left: Original SEM image of tumor tissue, Center: After brightness and contrast adjustment, Right: After invert procedure

Figure 4.4: Left: Result of $\gamma$ threshold, Right: Result of segmentation thresholding
Figure 4.5: Example of neighborhood scheme for a vertex \((i, j) = (22, 17)\). The green square imposed on the thresholded image (top left) demarcates the boundary of the neighborhood for the vertex shown by the blue square (top right). The plot shows the number of white components \(\beta_0\) for various \(r\).
Chapter 5

Results

In Figure 4.1, we have the two original SEM images corresponding to normal tissue and tumor tissue. In Figures 5.1 we see the results of our performing our method on these two images. Each image depicts the local homology data at each pixel for various values of $r$.

As expected, the number of components found ($\beta_0$) is greater for larger values of $r$. More importantly, the results show regions where $\beta_0$ grows to values larger than surrounding regions. These “hotspots” roughly correspond to regions where there are more intersections between the vessels, and where the vessels appear more tangled. For the two sample images at the bottom of Figure 5.1, we notice the hotspots are more regular for the normal tissue than for the tumor tissue. This can be seen immediately from the SEM images themselves, but it is evident in the results as well.

The most interesting information may come from a closer analysis at the both the shape of these hotspots and the progression of their growth, since this contain information about the shape and size of the voids between vessels, and in turn information about the morphology of the vasculature.
Figure 5.1: Persistent homology for processed SEM images of vasculature in tissue. Left: $\beta_0$ values at each pixel for normal tissue for various radii. Right: $\beta_0$ values at each pixel for tumor tissue for various radii.
Chapter 6

Conclusion

In this thesis, I have introduced a method for analyzing SEM images of vasculature by using algorithms from computational topology. By looking at the persistent homology of thresholded images, we can consider how the local complexity of the vasculature changes throughout the image. Preliminary results suggest this method may allow for quantification of vessel morphology. This could provide a metric for distinguishing between normal blood vessels and the abnormal angiogenic vessels found near aggressive tumors.

For future study, two directions are important. First, using this method on a large dataset of images, both of normal and tumor tissue, could allow us to determine whether or not the differences in persistent homology that we have found are statistically significant. Second, because the method is so reliant on the thresholding process, it would be beneficial to examine the dependence on the thresholding parameters. This may make a $\gamma$-threshold or a contrast-brightness method more appealing since it is easily automated. In this way, we could study how robust our metric is.
Appendix A

Vector Spaces

A vector space \( V \) is a set that is closed under finite vector addition and scalar multiplication. For \( V \) to be a vector space, the following conditions must hold for all elements \( w, y, z \in V \) and any scalars \( a, b \in F \) where \( F \) is a field.

1. Commutativity: \( w + y = y + w \)
2. Associativity: \( (w + y) + z = w + (y + z) \)
3. Additive identity: for all \( w \), \( 0 + w = w + 0 = w \)
4. Additive inverses exist: \( w + (-w) = 0 \)
5. Associativity of scalar multiplication: \( a(bw) = (ab)w \)
6. Distributivity of scalar sums: \( (a + b)w = aw + bw \)
7. Distributivity of vector sums: \( a(w + y) = aw + ay \)
8. Scalar multiplicative identity: \( 1w = w \)

**Definition.** For any map \( \varphi : V \to W \) (where \( V \) and \( W \) are vector spaces), the **kernel** is defined by

\[
\ker(\varphi) = \{ v \mid v \in V \text{such that } \varphi(v) = 0 \}.
\] (A.0.1)

Since this gives the elements from the vector space \( V \) which are mapped to zero, \( \ker(T) \) is a subset of \( V \).

**Definition.** The **image** of \( T \) is defined by

\[
im(\varphi) = \{ \varphi(v) \mid v \in V \}.
\] (A.0.2)

**Definition.** Let \( Y \) be a subspace of a vector space \( W \). The quotient space \( W/Y \) is the set of equivalence classes \( [w] \) where \( w_1 \sim w_2 \) if \( w_1 - w_2 \in Y \).

Alternatively, if \( w_1 \) is equivalent to \( w_2 \) modulo \( Y \) \( (w_1 \sim w_2) \), it is also true that \( w_1 = w_2 + y \) for some \( y \in Y \). These equivalence classes could also be written as cosets \( w + Y \).

Let \( Y = \{(w_1, w_2, w_3)\} \) and \( Z = \{(w_1, 0, 0)\} \). The quotient space \( Y/Z \) is isomorphic to \( \{(w_2, w_3)\} = R^2 \).

**Theorem A.0.1.** \( \{y\} \) Let \( V \) be a vectorspace over \( F \) and let \( W \) be a subspace of \( V \). Then \( V/W \) is a vectorspace with \( \dim V = \dim W + \dim V/W \) (where if one side is infinite, both are).

**Corollary A.0.2.** Let \( \varphi : V \to U \) be a linear transformation of vectorspaces over \( F \). Then \( \ker \varphi \) is a subspace of \( V \), \( \varphi(V) \) is a subspace of \( U \) and \( \dim V = \dim \ker \varphi + \dim \varphi(V) \).
## Appendix B

### List of Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k$</td>
<td>dimension of a simplex or chain</td>
</tr>
<tr>
<td>$\rho, R$</td>
<td>simplex $\rho$ defined by the set $R$</td>
</tr>
<tr>
<td>$\sigma, S$</td>
<td>abstract simplex $\sigma$ defined by the set $S$</td>
</tr>
<tr>
<td>$v$</td>
<td>vertices of abstract simplices</td>
</tr>
<tr>
<td>$x$</td>
<td>vertices of abstract simplicial complexes</td>
</tr>
<tr>
<td>$K$</td>
<td>an abstract simplicial complex</td>
</tr>
<tr>
<td>$U$</td>
<td>a collection of abstract simplices</td>
</tr>
<tr>
<td>$C_k$</td>
<td>the $k$th chain space of the simplicial complex $K$, ie. $C_k(K)$</td>
</tr>
<tr>
<td>$\partial_k$</td>
<td>the boundary map $C_k \rightarrow C_{k+1}$</td>
</tr>
<tr>
<td>$B_k$</td>
<td>the $k$th boundary space</td>
</tr>
<tr>
<td>$Z_k$</td>
<td>the $k$th cycle space</td>
</tr>
<tr>
<td>$H_k$</td>
<td>the $k$th homology space</td>
</tr>
<tr>
<td>$\beta_k$</td>
<td>the $k$th Betti number</td>
</tr>
<tr>
<td>$x_m$</td>
<td>the $m$th datapoint of blood vessel pixel data</td>
</tr>
<tr>
<td>$M$</td>
<td>the number of datapoints $x_m$</td>
</tr>
<tr>
<td>$n$</td>
<td>the number of complexes in a filtered complex</td>
</tr>
<tr>
<td>$N_1 \times N_2$</td>
<td>size of matrix that represents the SEM image</td>
</tr>
<tr>
<td>$i, j$</td>
<td>subscripts for matrix component $(i,j)$ where $1 \leq i \leq N_1, 1 \leq j \leq N_2$, and $i, j \in \mathbb{Z}$</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>radius for Rips complex</td>
</tr>
<tr>
<td>$r$</td>
<td>filtration parameter</td>
</tr>
<tr>
<td>$A$</td>
<td>thresholded image</td>
</tr>
<tr>
<td>$X$</td>
<td>the space in $\mathbb{R}^2$ representing the blood vessels</td>
</tr>
</tbody>
</table>
Appendix C

Matlab Code for pMovie.m

% pMovie.m
% Input:
% 300x400 binary matrix (e.g. a thresholded SEM image)
% Output:
% (300-2*radius)x(400-2*radius)x20 matrix of local homology data for each pixel of the input image for 20 different sized neighborhoods.
% This Matlab routine uses persist_getI.m and the Plex library.
% Author: Chris DuBois
% Advisor: Professor Vin de Silva
% Last revised: March 22, 2006

function final = pMovie(T)

[height,width]=size(T);

% Initialize output matrix
final = zeros(270,370,20);

% Parameters
% comp=0; % 0 = count b0 where space is black (value 0)
nbrhWidth = 35; % width of square neighborhood
focusWidth = 5; % width of square focus region
%============================================================================

% Number of pixels on each side of focus region
radius = (nbrhWidth-focusWidth)/2;

% Number of rows of neighborhoods to evaluate
numOfNbrhdRows = (height-2*radius)/focusWidth;

% Number of columns of neighborhoods to evaluate
numOfNbrhdCols = (width-2*radius)/focusWidth;

% For each neighborhood block....
for NBR_i = 1:numOfNbrhdRows
  % Find the neighborhood’s range of vertical indices
  NBR_top = (NBR_i-1) * focusWidth + 1;
  NBR_bot = (NBR_i-1) * focusWidth + nbrhWidth;
  for NBR_j = 1:numOfNbrhdCols
    % Find the neighborhood’s range of horizontal indices
    NBR_lft = (NBR_j-1) * focusWidth + 1;
    NBR_rht = (NBR_j-1) * focusWidth + nbrhWidth;

    % Output current progress
    fprintf('%g %g progress: ',NBR_i,NBR_j)

    % Save current neighborhood
    A = T(NBR_top:NBR_bot,NBR_lft:NBR_rht);

    % Make simplicial complex from this neighborhood
    A = (A~=comp);
    [C,X] = persist_makeC(A,1.5);

    % Perform homology calculations for each pixel in the focus region
    for fi=(radius+1):(radius+focusWidth)
      for fj=(radius+1):(radius+focusWidth)
        % Get persistence interval data
        [I] = persist_getI(C,X,fi,fj);
        intrv = I(1);

        % Count number of b0 components within r using interval data
        for k = 1:radius
          r = k;
          s = sum((intrv(1,:)<r).*(intrv(2,:)>=r));
          final(NBR_top+fi-1,NBR_lft+fj-1,k) = s;
        end
      end
    end
  end
end
Appendix D

Matlab Code for persist_getI.m

% persist_getI.m

% Input:
% C: Simplicial complex (from persist_makeC)
% X: List of vertex subscripts (from persist_makeC)
% (i,j): current pixel under consideration

% Output:
% I: interval data from persistence algorithm

% Author: Chris DuBois
% Advisor: Professor Vin de Silva
% Last revised: March 22, 2006

function I = persist_getI(C,X,i,j)

% Compute distances to between all vertices and pixel at (i,j)
D = px_euclid(X,[i;j]);

% Make a filtration
G = extend(C,{D'},1);

% Apply the persistence algorithm
[I] = persistence(C,G);
intervals = I{1};
Appendix E

Matlab Code for persist_makeC.m

function [C,X] = persist_makeC(Z, r)

% Input:
% Z: region of thresholded image
% %
% Output:
% C: simplicial complex: vertices where v=(i,j)=1, edges where
% |v-u|<1.5
% X: (n x 2) matrix of subscripts [i j] for each of n vertices

list = find(Z);
[i_sub,j_sub] = ind2sub(size(Z),list);
X = [i_sub,j_sub]';
[C,F] = px_rips(X,1,r);
Bibliography


