Euler Characteristic Transform of Shapes in 2D Digital Images as Cubical Sets

A Thesis

Presented in Partial Fulfillment of the Requirements for the Degree Master of Mathematical Sciences in the Graduate School of The Ohio State University

By

Qitong Jiang, B.S.

Graduate Program in Mathematical Sciences

The Ohio State University

2020

Master's Examination Committee:

James Fowler, Advisor

Sebastian Kurtek

Abstract

We discuss the Euler characteristic transform (ECT) as a method to model shapes in 2D digital images. We study the mathematical background of Euler characteristic transform based on cubical homology, and we review the recent works about Euler characteristic transform. We also exhibit the algorithms for Euler characteristic transform. Lastly, we conduct some distance-based clustering analysis, and we present some results in the end.

This is dedicated to the one I love

Acknowledgments

I would like to thank my advisor, Dr. Jim Fowler, for the guidance and advice he provided to me through numerous valuable conversations. I would like to thank Dr. Tom Needham for introducing this topic to me and for his kind help throughout the master's program. I would also like to thank Dr. Sebastian Kurtek for his significant insights to the project and for being on my defense committee. Additionally, I would like to thank my friend, Zichuan Wang. Thank you for taking time to listen to me.

Finally, a special thank you to my mom for the love and support!

Vita

2017 ..... B.S. Economics

2017 ..... B.S. Mathematics

Fields of Study

Major Field: Mathematical Sciences

Table of Contents

List of Figures

Chapter 1: Introduction

The paper will explore the uses of the Euler characteristic transform (ECT) on analyzing shapes in 2D digital images. Euler characteristic transform is a method to represent geometric shapes by using tools from topological data analysis. So in the first chapter, we will review the math concepts in applied algebraic topology. In particular, we will talk about cubical homology theory. In the same chapter, we will define the Euler characteristic transform for a cubical set and discuss a bit about persistent homology.

In the next chapter, we will see the precursor of the Euler characteristic transform, which is the persistent homology transform. And we will also review recent research about persistent homology transform and Euler characteristic transform to help readers know the short history of ECT.

After the literature review, we will first show the idea of how to convert a digital image to a cubical complex. And then, we will demonstrate how to construct a number of Euler curves for each cubical complex made from an image, that is, to execute the Euler characteristic transform. The detailed algorithms and explanation will also be presented. Moreover, after we use ECT to represent images, a distance between two ECTs of cubical sets will be defined. So, we will do some distance-based cluster analysis on some samples sets of image data. The cluster

analysis will help us further master the procedures and understand what kinds of information this technique can tell us.

Finally, we will address some directions of future works.

Chapter 2: Cubical Homology and Euler Characteristic Transform

2.1 Cubical Homology

In this section, we will review the concept of cubical sets and cubical homology. We follow the materials in chapter 2 of the book *Computational Homology* by Kaczynski et al. [5]. The notations we used mostly came from the book.

2.1.1 Elementary Cubes

Definition 2.1. An ***elementary interval*** is a closed interval $I \subset \mathbb{R}$ of the form $I = [v, v+1]$ or $I = [v, v]$ for some $v \in \mathbb{Z}$. We write $[v] = [v, v]$ for an interval that contains only one point. The elementary intervals of the form $[v]$ are said to be ***degenerate***, and the intervals of the form $[v, v]$ are ***nondegenerate***.

Definition 2.2. An ***elementary cube*** Q is a finite product of elementary intervals, that is, $Q = I_1 \times I_2 \times \cdots \times I_d \subset \mathbb{R}^d$, where each I_j is an elementary interval. The set of all elementary cubes in $\mathbb{R}^d$ is denoted by $\mathcal{K}^d$. The set of all elementary cubes is denoted by $\mathcal{K}$, namely $\mathcal{K} := \bigcup_{d=1}^{\infty} \mathcal{K}^d$.

Definition 2.3. Let $Q \in \mathcal{K}^d$. The ***dimension*** of Q is defined to be the number of nondegenerate elementary intervals in Q, denoted by $\dim Q$. We let $\mathcal{K}_k := \{Q \in \mathcal{K} : \dim Q = k\}$ and $\mathcal{K}_k^d := \mathcal{K}_k \cap \mathcal{K}^d$.

2.1.2 Cubical Sets

Definition 2.4. A set $X \subset \mathbb{R}^d$ is said to be ***cubical*** if X can be written as a finite union of elementary cubes.

If $X \subset \mathbb{R}^d$ is a cubical set, then we adopt the following notations:

$$\mathcal{K}(X) := \{Q \in \mathcal{K} : Q \subset X\}$$

and

$$\mathcal{K}_k(X) := \{Q \in \mathcal{K}(X) : \dim Q = k\}.$$

So, the cubical set X can be written as

$$X := \bigcup_{Q \in \mathcal{K}(X)} Q = \bigcup_{k=1}^{d} \bigcup_{Q \in \mathcal{K}_k(X)} Q.$$

The elements of $\mathcal{K}_0(X)$ are called the ***vertices*** of X, and the elements of $\mathcal{K}_1(X)$ are called the ***edges*** of X. More generally, the elements of $\mathcal{K}_k(X)$ are the ***k-cubes*** of X.

Proposition 2.1. *If $X \subset \mathbb{R}^d$ is cubical, then X is closed and bounded.*

Proof. By definition a cubical set X is the finite union of elementary cubes. An elementary cube is closed since it is a product of closed sets. A finite union of closed sets is also closed.

For $Q = I_1 \times I_2 \times \cdots \times I_d \in \mathcal{K}(X)$, where $I_j = [v_j]$ or $I_j = [v_j, v_j + 1]$, let $\rho(Q) = \max_{j=1,\ldots,d}\{|v_j| + 1\}$. By taking $r := \max_{Q \in \mathcal{K}(X)} \rho(Q)$, we see that $X \subset B(0, R)$, where $B(0, r)$ is the open ball centered at 0 with radius r. So, X is bounded.

□

2.1.3 Cubical Chains

Definition 2.5. For each $Q \in \mathcal{K}_k^d$, define $\widehat{Q} : \mathcal{K}_k^d \to \mathbb{Z}$ by

$$\widehat{Q}(P) := \begin{cases} 1 & \text{for } P = Q, \\ 0 & \text{else.} \end{cases}$$

$\widehat{Q}$ is called an ***elementary k-chain dual*** to the elementary cube Q. The set of all elementary k-chains of $\mathbb{R}^d$ is denoted by $\widehat{\mathcal{K}}_k^d := \left\{\widehat{Q} : Q \in \mathcal{K}_k^d\right\}$, and the set of all elementary chains of $\mathbb{R}^d$ is denoted by $\widehat{\mathcal{K}}^d := \bigcup_{k=0}^{\infty} \widehat{\mathcal{K}}_k^d$.

Definition 2.6. The group of ***k-dimensional chains*** or ***k-chains*** of $\mathbb{R}^d$ is the free abelian group generated by the elementary chains of $\mathcal{K}_k^d$ and is denoted by C_k^d. Thus the elements of C_k^d are functions $c : \mathcal{K}_k^d \to \mathbb{Z}$ such that $c(Q) = 0$ for all but finitely many $Q \in \mathcal{K}_k^d$. In particular, $\widehat{\mathcal{K}}_k^d$ is the basis for C_k^d. If $c \in C_k^d$, then $\dim c := k$.

Proposition 2.2. *The map $\phi : \mathcal{K}_k^d \to \widehat{\mathcal{K}}_k^d$ by $Q \mapsto \widehat{Q}$ is a bijection.*

Proof. $\widehat{\mathcal{K}}_k^d$ is defined to be the image of ϕ. To prove injectivity, assume that $P, Q \in \mathcal{K}_k^d$ and $\widehat{P} = \widehat{Q}$. Then $1 = \widehat{P}(P) = \widehat{Q}(P)$. It follows that $P = Q$. □

Definition 2.7. Let $c \in C_k^d$. The ***support*** of the chain c is the cubical set

$$|c| := \bigcup \left\{ Q \in \mathcal{K}_k^d : c(Q) \neq 0 \right\}.$$

Remark. Support of a chain satisfies nice geometric features. For example, if $Q \in \mathcal{K}$, then $\left|\widehat{Q}\right| = Q$.

Definition 2.8. Let $c_1, c_2 \in C_k^d$, where $c_1 = \sum_{i=1}^m \alpha_i \widehat{Q}_i$ and $c_2 = \sum_{i=1}^m \beta_i \widehat{Q}_i$. The ***scalar product*** of the chains c_1 and c_2 is defined as

$$\langle c_1, c_2 \rangle := \sum_{i=1}^m \alpha_i \beta_i.$$

Proposition 2.3. *The scalar product defines a bilinear mapping* $\langle \cdot, \cdot \rangle : C_k^d \times C_k^d \to \mathbb{Z}$.

Definition 2.9. Given two elementary cubes $P \in \mathcal{K}_k^d$ and $Q \in \mathcal{K}_l^e$, define

$$\widehat{P} \diamond \widehat{Q} := \widehat{P \times Q}.$$

This extends to arbitrary chains $c_1 \in C_k^d$ and $c_2 \in C_l^e$ by

$$c_1 \diamond c_2 := \sum_{\substack{P \in \mathcal{K}_k \\ Q \in \mathcal{K}_l}} \langle c_1, \widehat{P} \rangle \langle c_2, \widehat{Q} \rangle \widehat{P \times Q}.$$

The chain $c_1 \diamond c_2 \in C_{k+l}^{d+e}$ is called the ***cubical product*** of c_1 and c_2.

Proposition 2.4. *Let* a, b, c *be any chains.*

(i) $a \diamond 0 = 0 \diamond a = 0$.

(ii) $a \diamond (b + c) = a \diamond b + a \diamond c$ *if* $b, c \in C_k^d$.

(iii) $(a \diamond b) \diamond c = a \diamond (b \diamond c)$.

(iv) If $a \diamond b = 0$, *then* $a = 0$ *or* $b = 0$.

(v) $|a \diamond b| = |a| \times |b|$.

Proposition 2.5. *Let* $\widehat{Q}$ *be an elementary cubical chain of* $\mathbb{R}^d$ *with* $d > 1$. *Then there exist unique elementary cubical chains* $\widehat{I}$ *and* $\widehat{P}$ *with* $I \in \mathcal{K}^1$ *and* $P \in \mathcal{K}^{d-1}$ *such that* $\widehat{Q} = \widehat{I} \diamond \widehat{P}$.

Proof. Since $\widehat{Q}$ is an elementary cubical chain, Q is an elementary cube with $Q = I_1 \times I_2 \times \cdots \times I_d$. If we set $I := I_1$ and $P := I_2 \times \cdots \times I_d$, then $\widehat{Q} = \widehat{I} \diamond \widehat{P}$.

If $\widehat{Q} = \widehat{J} \diamond \widehat{R}$ for some $J \in \mathcal{K}^1$ and $R \in \mathcal{K}^{d-1}$, then $\widehat{I \times P} = \widehat{J \times R}$. By Proposition 2.2 we have $I \times P = J \times R$. Since $I, J \subset \mathbb{R}$ (the first copy of $\mathbb{R}$ in $\mathbb{R}^d$), it follows that $I = J$ and hence $P = R$. □

2.1.4 Cubical Chains in a Cubical Set

Definition 2.10. Let $X \subset \mathbb{R}^d$ be a cubical set. Let $\widehat{\mathcal{K}}_k(X) := \left\{\widehat{Q} : Q \in \mathcal{K}_k(X)\right\}$. The set of ***k-chains*** of X is the free subgroup of C_k^d generated by the elements of $\mathcal{K}_k(X)$ and is denoted by $C_k(X)$.

Remark. We see that

$$C_k(X) = \left\{c \in C_k^d : |c| \subset X\right\}.$$

Lemma 2.1. *$\widehat{\mathcal{K}}_k(X)$ is a basis of $C_k(X)$.*

Proof. By definition of $\widehat{Q}$, the set $\widehat{\mathcal{K}}_k(X)$ is linearly independent. □

Remark. Since $\mathcal{K}_k(X)$ is finite, $C_k(X)$ is a finite dimensional free abelian group. And for any $c \in C_k(X)$, we can write

$$c = \sum_{Q_i \in \mathcal{K}_k(X)} \alpha_i \widehat{Q}_i,$$

where $\alpha_i = c(Q_i)$.

2.1.5 The Boundary Map

Definition 2.11. Given $k \in \mathbb{Z}$, the ***cubical boundary map*** $\partial_k : C_k^d \to C_{k-1}^d$ is defined for an elementary chain $\widehat{Q} \in \widehat{\mathcal{K}}_k^d$ by induction on d as follows.

Define $\partial_0 = 0$. When $d = 1$, Q is an elementary interval. Define

$$\partial_k \widehat{Q} := \begin{cases} 0 & \text{for } Q = [v], \\ \widehat{[v+1]} - \widehat{[v]} & \text{for } Q = [v, v+1]. \end{cases}$$

Next, when $d > 1$, $Q = I_1 \times I_2 \times \cdots \times I_d$. Let $I = I_1$ and $P = I_2 \times \cdots \times I_d$. Then by Proposition 2.5 we have $\widehat{Q} = \widehat{I} \diamond \widehat{P}$. Define

$$\partial_k \widehat{Q} := \partial_{k_1} \widehat{I} \diamond \widehat{P} + (-1)^{\dim I} \widehat{I} \diamond \partial_{k_2} \widehat{P},$$

where $k_1 = \dim I$ and $k_2 = \dim P$.

Finally, we extend the definition to all chains in C_k^d by linearity. That is, if $c = \sum_{i=1}^n \alpha_i \widehat{Q}_i$, then

$$\partial_k c := \sum_{i=1}^n \alpha_i \partial_k \widehat{Q}_i.$$

Remark. Note that the domain of ∂_k is C_k^d consisting of the k-chains. Whenever the domain C_k^d is given, k is understood. So we simplify the notation ∂_k to ∂.

Proposition 2.6.

$$\partial^2 = 0.$$

Proof. It suffices to verify the property for elementary chains. And the proof is by induction on the dimension of the ambient space $\mathbb{R}^d$.

Let Q be an elementary interval. If $Q = [v]$, then $\partial\partial\widehat{Q} = 0$. If $Q = [v, v+1]$, then

$$\begin{aligned}
\partial(\partial\widehat{Q}) &= \partial(\partial\widehat{[v, v+1]}) \\
&= \partial(\widehat{[v+1]} - \widehat{[v]}) \\
&= \partial(\widehat{[v+1]}) - \partial(\widehat{[v]}) \\
&= 0.
\end{aligned}$$

Next assume that Q is an elementary cube in $\mathcal{K}^d$ for $d > 1$. Then $Q = I_1 \times I_2 \times \cdots \times I_d$. Let $I = I_1$ and $P = I_2 \times \cdots \times I_d$. Then $Q = I \times P$. By Proposition 2.5,

$$\begin{aligned}
\partial(\partial\widehat{Q}) &= \partial(\partial(\widehat{I \times P})) \\
&= \partial(\partial(\widehat{I} \diamond \widehat{P})) \\
&= \partial(\partial\widehat{I} \diamond \widehat{P} + (-1)^{\dim \widehat{I}} \widehat{I} \diamond \partial\widehat{P}) \\
&= \partial(\partial\widehat{I} \diamond \widehat{P}) + (-1)^{\dim \widehat{I}} \partial(\widehat{I} \diamond \partial\widehat{P}) \\
&= \partial\partial\widehat{I} \diamond \widehat{P} + (-1)^{\dim \partial\widehat{I}} \partial\widehat{I} \diamond \partial\widehat{P} + (-1)^{\dim \widehat{I}} \Big(\partial\widehat{I} \diamond \partial\widehat{P} + (-1)^{\dim \widehat{I}} \widehat{I} \diamond \partial\partial\widehat{P} \Big) \\
&= (-1)^{\dim \partial\widehat{I}} \partial\widehat{I} \diamond \partial\widehat{P} + (-1)^{\dim \widehat{I}} \partial\widehat{I} \diamond \partial\widehat{P},
\end{aligned}$$

where $\partial\partial\widehat{I} = 0$ and $\partial\partial\widehat{P}$ by the inductive hypothesis.

Observe that if $\dim \widehat{I} = 0$, then $\partial\widehat{I} = 0$. Then $\partial\widehat{I} \diamond \partial\widehat{P} = 0$ and hence $\partial\partial\widehat{Q} = 0$. If $\dim \widehat{I} = 1$, then $\dim \widehat{I} = 0$. Thus, the two terms cancel each other. It follows that $\partial\partial\widehat{Q} = 0$. □

Definition 2.12. The boundary map for the cubical set X is defined to be $\partial_k^X : C_k(X) \to C_{k-1}(X)$ obtained by restricting ∂_k to $C_k(X)$.

Remark. The definition may need further justification of well-definedness. Readers can refer to [5] for details.

Definition 2.13. The ***cubical chain complex*** for the cubical set $X \subset \mathbb{R}^d$ is

$$\mathcal{C}(X) := \left\{ C_k(X), \partial_k^X \right\}_{k \in \mathbb{Z}},$$

where $C_k(X)$ are the groups of cubical k-chains generated by $\widehat{\mathcal{K}}_k(X)$ and ∂_k^X is the cubical boundary map restricted to X.

Definition 2.14. Let $\mathcal{C}(X) = \{C_k(X), \partial_k\}_{k \in \mathbb{Z}}$ be a cubical chain complex for the cubical set $X \subset \mathbb{R}^d$. A cubical chain complex $\mathcal{D}(X) = \left\{ D_k(X), \partial'_k \right\}_{k \in \mathbb{Z}}$ is a ***cubical chain subcomplex*** of $\mathcal{C}(X)$ if

1. $D_k(X)$ is a subgroup of $C_k(X)$ for all $k \in \mathbb{Z}$.

2. $\partial'_k = \partial_k|_{D_k(X)}$.

Proposition 2.7. *Let $X \subset Y$ be cubical sets. Then $\mathcal{C}(X)$ is a chain subcomplex of $\mathcal{C}(Y)$.*

Proof. The proof requires only a careful inspection of definitions. $X \subset Y$ implies that $\mathcal{K}(X) \subset \mathcal{K}(Y)$, hence $\mathcal{K}_k(X) \subset \mathcal{K}_k(Y)$ for all k. Thus, $\widehat{\mathcal{K}}_k(X) \subset \widehat{\mathcal{K}}_k(Y)$. Since $\widehat{\mathcal{K}}_k(X)$ and $\widehat{\mathcal{K}}_k(Y)$ are bases of the free subgroups $C_k(X)$ and $C_k(Y)$ respectively, $C_k(X)$ is a subgroup of $C_k(Y)$.

Recall that the cubical boundary map was defined by linearly extending the definition on elementary cubes. Thus, the boundary maps ∂_k^X for $\mathcal{C}(X)$ and ∂_k^Y for $\mathcal{C}(Y)$ can be obtained by restricting ∂ to $C_k(X)$ and $C_k(Y)$ respectively. So $\partial_k^X = \partial_k^Y|_{C_k(X)}$ □

Remark. For cubical sets X, Y, we have $X \subset Y$ iff $K(X) \subset K(Y)$. Indeed, for the opposite direction, if there exists an $x \in X$ such that $x \notin Y$, then there exists a cube $Q \in X$ such that $Q \notin Y$. Hence, $K(X) \not\subset K(Y)$.

2.1.6 Homology of Cubical Sets

Definition 2.15. Let $X \subset \mathbb{R}^d$ be a cubical set. A k-chain $z \in C_k(X)$ is called a ***cycle*** in X if $\partial z = 0$. The set of all k-cycles in $C_k(X)$ is denoted by $Z_k(X)$.

A k-chain $b \in C_k(X)$ is called a ***boundary*** in X if there exists $c \in C_{k+1}(X)$ such that $\partial c = b$. The set of boundary elements in $C_k(X)$ is denoted by $B_k(X)$.

Remark. The set $Z_k(X) = \left\{z \in C_k(X) : \partial_k^X z = 0\right\}$ is the kernel of the boundary map. Thus, the set $Z_k(X)$ forms a subgroup of $C_k(X)$. The set $B_k(X)$ is by definition the image of the boundary map ∂_{k+1}^X. So $B_k(X)$ is also a subgroup of $C_k(X)$.

Moreover, by proposition 2.6, $\partial c = z$ implies $\partial z = \partial^2 c = 0$. Thus, every boundary is a cycle and hence $B_k(X)$ is a subgroup of $Z_k(X)$.

Definition 2.16. The kth ***cubical homology group*** or the kth ***homology group*** of X is the quotient group

$$H_k(X) := Z_k(X)/B_k(X).$$

The homology of X is the collection of all homology groups of X, denoted by $H_*(X) := \{H_k(X)\}_{k\in\mathbb{Z}}$.

Remark. We are interested in cycles that are not boundaries. So we treat cycles that are boundaries as trivial. Therefore, the elements of the quotient group $H_k(X) := Z_k(X)/B_k(X)$ are equivalence class of cycles under the equivalence relation that for any two cycles $z_1, z_2 \in Z_k(X)$, $z_1 \sim z_2$ if $z_1 - z_2$ is a boundary, i.e., $z_1 - z_2 \in B_k(X)$. We say that z_1 and z_2 are ***homologous*** if they differ by a boundary.

Definition 2.17. Let $z \in Z_k(X)$. We call the equivalence class of z in $H_k(X)$ the ***homology class*** of z. It is denoted by $[z]_X$ or $[z]$ if the cubical set X is clear from the context.

2.2 Euler Characteristic Curves

In this section, we want to discuss about the Euler characteristic curve, which is the main focus of this paper. The reader will also get some ideas of filtration and persistent homology from this section before we introduce them in the later section.

We will quote without proving the following theorem:

Theorem 2.1. *Any finitely generated abelian group G is isomorphic to a group of the form:*

$$\mathbb{Z}^r \oplus \mathbb{Z}/b_1\mathbb{Z} \oplus \mathbb{Z}/b_2\mathbb{Z} \oplus \cdots \oplus \mathbb{Z}/b_k\mathbb{Z},$$

where r is a non-negative integer, $b_i > 1$ whenever $k > 0$, and b_i divides b_{i+1} for $i \in \{1, 2, \ldots, k-1\}$ whenever $k > 1$. The numbers $b_1, b_2, \ldots, b_k$, and r are uniquely determined by G.

Definition 2.18. The number r is the ***rank*** of the free subgroup $\mathbb{Z}^r$ and is called the ***Betti number*** of G.

Definition 2.19. Let $X \subset \mathbb{R}^d$ be a cubical set, and β_k be the Betti number of the kth homology group $H_k(X)$ of X. The ***Euler characteristic*** of X is defined to be

$$\chi(X) := \sum_{k \in \mathbb{Z}} (-1)^k \beta_k.$$

Definition 2.20. Let $u \in \mathbb{R}^d$ be a vector and $h \in \mathbb{R}$ be a 'height' value. Let $Q = I_1 \times I_2 \times \cdots \times I_d$ be an elementary cube. We say that Q is ***with a height of*** $\leq h$ ***in direction*** u if for every I_j with $j \in \{1, \ldots, d\}$,

1. $(v_j + 1) \cdot u \leq h$, provided $I_j = [v_j, v_{j+1}]$, and
2. $v_j \cdot u \leq h$, provided $I_j = [v_j]$.

The set of all elementary cubes that are with a height of $\leq h$ in direction u is denoted by $\mathcal{K}^d_{(u,h)}$.

Definition 2.21. Let $X \subset \mathbb{R}^d$ be a cubical set and let $\mathcal{K}_{k(u,h)}(X) := \mathcal{K}_k(X) \cap \mathcal{K}^d_{(u,h)}$. We define

$$X_{(u,h)} := \bigcup_{Q \in \mathcal{K}_{(u,h)}(X)} Q,$$

where $\mathcal{K}_{(u,h)}(X) = \bigcup_{k=0}^{d} \mathcal{K}_{k(u,h)}(X)$.

Remark. By Proposition 2.7 the corresponding cubical chain subcomplex is given by

$$\mathcal{C}_{(u,h)}(X) := \left\{ C_k(X_{(u,h)}), \partial_{(u,h)} \right\}_{k \in \mathbb{Z}}.$$

Accordingly, we can compute homology groups with a cubical chain subcomplex. Also we can compute Euler characteristics by taking an alternating sum of the rank of the resulted homology group in each dimension. It leads to the following definition.

Definition 2.22. Let $X \subset \mathbb{R}^d$ be a cubical set and let u be a direction vector in R^d. The ***Euler curve*** of X in the direction u is a function $EC_{u,X} : \mathbb{R} \to \mathbb{Z}$ defined by

$$h \mapsto \chi\left(X_{(u,h)}\right).$$

Remark. The Figure 2.1 is an illustration of the Euler curve of a cubical set in the direction $(1,1) \in \mathbb{R}^2$.

Definition 2.23. Let $X \subset \mathbb{R}^d$ be a cubical set and let $S^1 \subset \mathbb{R}^2$ be the set of vectors of unit length in $\mathbb{R}^2$. The ***Euler characteristic transform*** of X is defined to be $ECT(X) : S^{d-1} \to \mathbb{Z}^{\mathbb{R}}$ by

$$u \mapsto EC_{u,X}.$$

At this point, we have established the definition of Euler characteristic transform.

2.3 Chain Maps

In the later Section 2.4, in order to define the persistent homology, we want to consider the effects of continuous functions between cubical sets on the homology

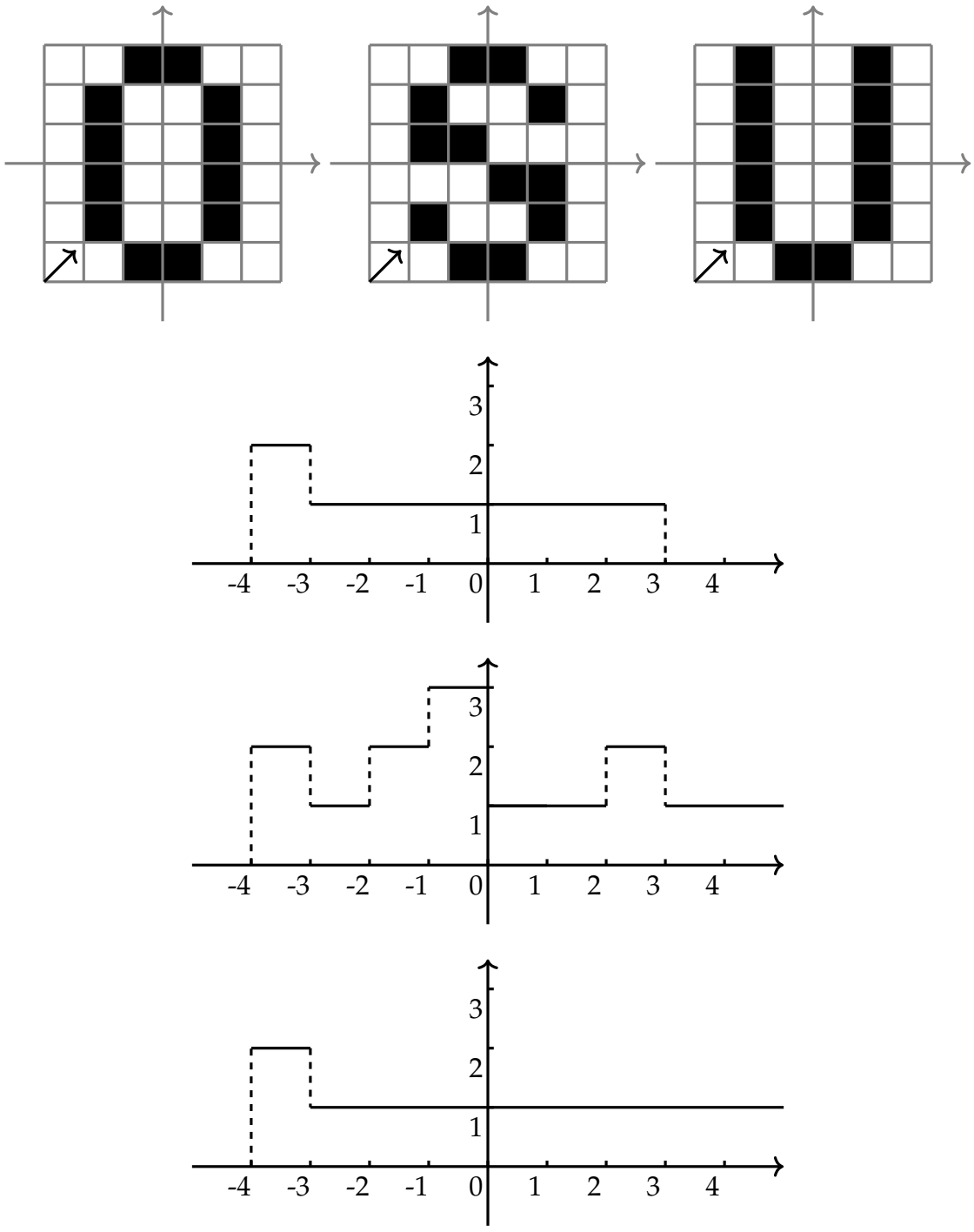


Figure 2.1: Euler curves of binary images in direction $(1, 1)$.

groups. In particular, if X_i and X_j are cubical sets with $X_i \subset X_j$, what will be the effect of the inclusion map $X_i \hookrightarrow X_j$ on the homology groups $H_k(X_j)$ for all $k \in \mathbb{Z}$. We will see that a continuous map between cubical sets X, Y induces a chain map between the cubical chain complexes of X and Y, hence induces homomorphisms between homology groups of X and Y.

Definition 2.24. Let $\mathcal{C}(X) = \{C_k(X), \partial_k^X\}_{k\in\mathbb{Z}}$ and $\mathcal{C}(Y) = \{C_k(Y), \partial_k^Y\}_{k\in\mathbb{Z}}$ be cubical chain complexes. A sequence of homomorphisms $\phi_k : C_k(X) \to C_k(Y)$ is a ***chain map*** if for every $k \in \mathbb{Z}$

$$\partial_k^Y \phi_k = \phi_{k-1} \partial_k^X.$$

We use the notation $\phi_\# : \mathcal{C}(X) \to \mathcal{C}(Y)$ to represent the collection of homomorphisms $\{\phi_k : k \in \mathbb{Z}\}$.

Remark. Given $z \in Z_k(X)$, since $\partial_k^X(z) = 0$, we have $\partial_k^Y \phi_k(z) = \phi_{k-1}\partial_k^X(z) = \phi_{k-1}(0) = 0$. So a chain map takes cycles to cycles. Since $\phi_{k-1}\partial_k^X = \partial_k^Y \phi_k$, a chain map also takes boundaries to boundaries. We have the following Lemma.

Lemma 2.2. *If $\phi : \mathcal{C}(X) \to \mathcal{C}(Y)$ is a chain map, then*

$$\phi_k(Z_k(X)) \subset Z_k(Y) := \ker \partial_k^Y$$

and

$$\phi_k(B_k(X)) \subset B_k(Y) := \operatorname{im} \partial_{k+1}^Y$$

for all $k \in \mathbb{Z}$.

Proposition 2.8. *A chain map $\phi : \mathcal{C}(X) \to \mathcal{C}(Y)$ between chain complexes induces homomorphisms between the homology groups of X and Y. That is, the homomorphism*

$\phi_{k*} : H_k(X) \to H_k(Y)$ *by*

$$\phi_{k*}([z]) := [\phi_k(z)],$$

where $z \in Z_k$ *for all* $k \in \mathbb{Z}$.

Proof. The Lemma 2.2 guarantees that the map ϕ_{k*} is a homomorphism from the quotient group $\ker \partial_k^X / \operatorname{im} \partial_{k+1}^X$ to $\ker \partial_k^Y / \operatorname{im} \partial_{k+1}^Y$, that is $\phi_{k*} : H_k(X) \to H_k(Y)$.

We also want to see the actual definition is independent of the choice of z. Assume that $[z] = [z']$ for some $z' \in Z_k$. Then $z' = z + b$ for some $b \in B_k$. Since $\phi_\#$ is a chain map, we have

$$\phi_{k*}\big([z']\big) = \big[\phi_k(z')\big] = [\phi_k(z+b)] = [\phi_k(z) + \phi_k(b)] = [\phi_k(z)] = \phi_{k*}([z]).$$

□

Remark. We use the notation $\phi_* : H_*(X) \to H_*(Y)$ to represent the collection of homomorphisms $\{\phi_{k*} : k \in \mathbb{Z}\}$.

The following example will be useful when we define the persistent homology groups in the section 2.4.

Example 2.1. Let $X \subset Y$ be cubical sets in $\mathbb{R}^d$. We know from the Proposition 2.7 that the cubical chain complex $\mathcal{C}(X)$ is a chain subcomplex of $\mathcal{C}(Y)$. The inclusion map $\iota : X \hookrightarrow Y$, given by $\iota(Q) := Q$, maps elementary cubes to elementary cubes. It defines an inclusion map of the chain complexes, $\iota : \mathcal{C}(X) \to \mathcal{C}(Y)$ by

$$\iota_k(\widehat{Q}) := \widehat{Q},$$

for all $Q \in \mathcal{K}_k(X)$. More specifically, in each dimension k, by taking the linear combination of the elementary k-chains, we get an inclusion map $\iota_k : C_k(X) \to$

$C_k(Y)$ defined by $\iota_k(c) := c$ for every $c \in C(X)$. Since $\partial_k \iota_k(c) = \partial_k(c) = \iota_{k-1}\partial(c)$, we have ι is a chain map. Therefore, it induces a homomorphism of homology groups

$$\iota_* : H_*(X) \to H_*(Y).$$

We will consider the image of a homology class $[z] \in H_k(X)$ under the map ι_* as well as the image of $H_k(X)$ for all $k \in \mathbb{Z}$ later.

2.4 Filtrations and Persistent Homology

In the above section 2.2, we defined the subcomplex determined by a directional vector and a height value, and we wanted to calculated the sublevel Euler characteristics. The idea behind these is to measure changes in the homology of a filtration of complexes over time. So in this chapter, we will briefly talk about the concept of filtrations and persistent homology. The definitions we used here came from [3, 12, 10]. It is helpful to know the concepts for the literature review chapter.

Definition 2.25. Let $X \subset \mathbb{R}^d$ be a cubical set. A ***filtration*** of X is a nested sequence of cubical subsets of X, that is $\emptyset = X_0 \subset X_1 \subset X_2 \subset \cdots \subset X_n = X$.

Remark. The cubical chain complex for X_i is denoted by

$$\mathcal{C}(X_i) := \left\{ C_k(X_i), \partial_k^{X_i} \right\}_{k \in \mathbb{Z}}.$$

Proposition 2.7 implies that $\mathcal{C}(X_i)$ is a subcomplex of $\mathcal{C}(X_{i+1})$ for every $i \geq 0$.

Example 2.2. In this example we introduce a kind of filtration that is related to the Euler characteristic transform. Given a cubical set $X \subset \mathbb{R}^d$ and a direction u, we can order the vertices of X by their "heights" in the direction u. For example, as

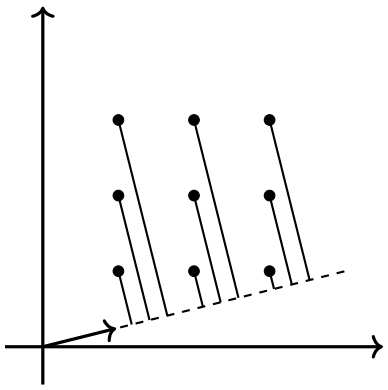


Figure 2.2: Order of vertices

shown in Figure 2.2, if the directional vector u is a unit vector, then we can compare the scalar projections of those vertex vectors onto u, which is the dot product of the vertex vector and u. We can partition the set of vertices into equivalence classes by the relation $v_1 \sim v_2$ iff $v_1 \cdot u = v_2 \cdot u$.

Suppose we have a set of equivalence classes of vertices of a cubical set X, and let $\{[v_1], [v_2], \ldots, [v_n]\}$ denote the set, ordered by comparing their scalar projections. We define $X_0 = \emptyset$ and X_i to be the cubical set such that

$$\mathcal{K}(X_i) := \{Q \in \mathcal{K}(X) : v \notin Q \text{ if } [v] > [v_i]\}.$$

The sequence $\emptyset = X_0 \subset X_1 \subset \cdots \subset X_n = X$ is a filtration. We can see an illustration of the filtration of a cubical set determined by the vector $(1, 1)$ in the Figure 2.3.

Definition 2.26. Let $X_0 \subset X_1 \subset \cdots \subset X_n = X$ be a filtration. The ***p-persistent kth homology group*** of X_i is

$$H_k\big(X_{i,i+p}\big) := Z_k(X_i) / \big(B_k(X_{i+p}) \cap Z_k(X_i)\big)$$

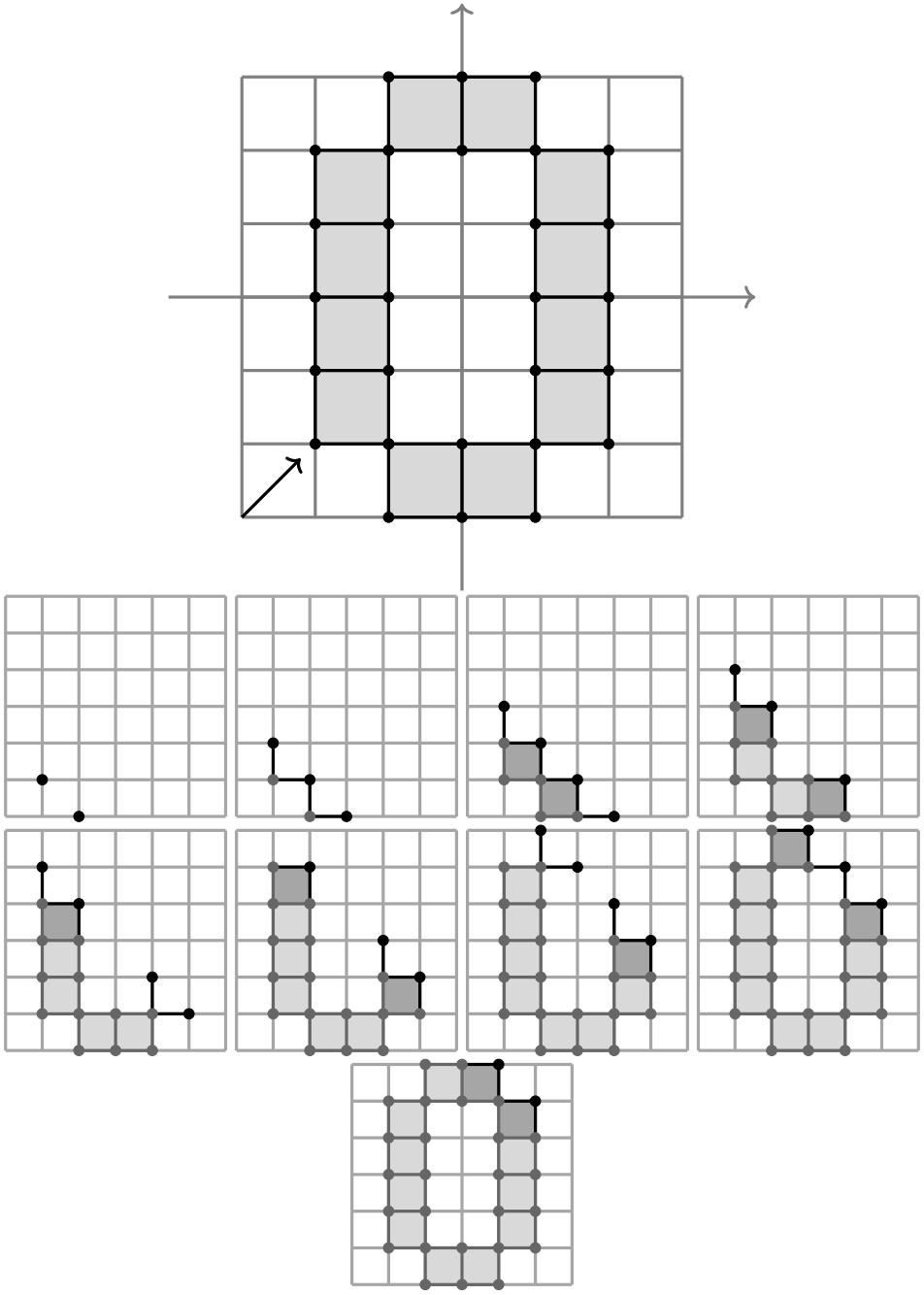


Figure 2.3: Filtration

The corresponding ***p*-persistent *k*th Betti number** is $\beta_k^{i,i+p} := \operatorname{rank} H_k(X_{i,i+p})$.

Remark. An equivalence definition is that the p-persistent kth homology groups of a cubical set X_i is the image of the homomorphism induced by the inclusion map $\iota_{i,j} : X_i \to X_j$ where $j = i + p$. That is $H_k(X_{i,j}) = \operatorname{im} \iota_{i,j}$ if we use the same notation for the induced map $H_k(X_i) \hookrightarrow H_k(X_{i+p})$.

We note that $H_k(X_{i,i}) = H_k(X_i)$.

Definition 2.27. Let $[z] \in H_k(X_i)$ for $i > 0$. We say that $[z]$ is ***born at*** X_i if $[z] \notin H_k(X_{i',i})$, that is $[z] \notin \operatorname{im} \iota_{i',i}$ for all $i' < i$. Moreover, for $[z]$ that is born at X_i, we say that $[z]$ ***dies entering*** X_j if for every $i' < i < j' < j$, we have $\iota_{i,j'}([z]) \notin H_k(X_{i',j'})$ and $\iota_{i,j}([z]) \in H_k(X_{i',j})$.

Definition 2.28. If $[z]$ is born at X_i and dies entering X_j, then the difference in index $j - i$ is said to be the ***index persistence*** of $[z]$. If $[z]$ never dies, then its index persistence is defined to be ∞.

We can visualize the collection of persistent Betti numbers.

Definition 2.29. Let $\mu_k^{i,j}$ be the number of k dimensional homology classes born at X_i and dying entering X_j, we have for every $i' < i < j' < j$,

$$\mu_k^{i,j} := \left(\beta_k^{i,j'} - \beta_k^{i,j}\right) - \left(\beta_k^{i',j'} - \beta_k^{i',j}\right).$$

For $i \leq j$, drawing each point (i, j) with multiplicity $\mu_k^{i,j}$ on the extended real plane $\overline{\mathbb{R}}^2$, we get the ***k*th persistence diagram** of the filtration, denoted by Dgm_k.

Chapter 3: Literature Review

In this chapter, we will review recent works about the Euler characteristic transform. Euler characteristic transform is derived from the persistent homology transform (PHT), which is designed to model surfaces in $\mathbb{R}^3$ and shapes in $\mathbb{R}^2$. Persistent homology is a popular tool in topological data analysis, and it is an algebraic method for detecting topological structures of data. The data being studied in TDA could be many types of data, such as point cloud data, time series data, or image data. Shape analysis is about statistical analysis of geometric shapes including shape matching and shape recognition. It mainly studies and processes geometric shapes extracted from 2D and 3D images. The idea behind persistent homology transform and Euler characteristic transform is to use methods from topological data analysis to study geometric shapes.

3.1 Euler Characteristic Transform

Turner et al. [10] introduce the persistent homology transform as a tool to perform shape analysis on objects in $\mathbb{R}^3$ and shapes in $\mathbb{R}^2$. In short, for a subset M of $\mathbb{R}^d$ (d=2 or 3), which can be written as a finite simplicial complex, the persistent homology transform of M is a function assigning to a directional vector $v \in S^{d-1}$ the collection of the kth dimensional persistence diagram corresponding to v for

$k \in \{1, \ldots, d-1\}$. One notable result of the paper is that such collections of persistent diagrams are sufficient statistics for shape and surface models. It is the first formal demonstration that using persistent homology would not result in loss of information. This has been demonstrated by proving the theorem that the persistent homology transform is injective when the domain is space of subsets of $\mathbb{R}^2$ or $\mathbb{R}^3$ that can be written as finite simplicial complexes (Theorem 3.1 and Corollary 3.1 in [10]).

Another important aspect of the injectivity theorem is that the proof of the theorem is actually constructive. It suggests that the transform is theoretically invertible and provides an algorithm to reconstruct the simplical complex hence the set from the persistent diagrams.

The space of persistent diagrams is a metric space [7]. The distances between persistent diagram may be defined. Nevertheless, the geometric structure of the space of persistent diagrams is complicated. In order to resolve this issue, they introduce a simplified variation of the PHT, which is the Euler characteristic transform. Although the Euler curves store relatively less geometric and topological information than the persistent diagrams, they live in a space with a much better geometric structure. As a corollary of the aforementioned theorem, the Euler characteristic transform is also injective. This new sufficient statistic is a collection of curves that can have an inner product structure which allows people to apply the smooth ECT to a broader set of statistical methodologies [1]. The original idea and the definitions can be found in [10]. This explains the motivation for inventing the Euler characteristic transform in addition to the persistent homology transform.

The definition are basically the same as we presented in section 2, except that they originally consider the sets that can be turned into finite simplicial complexes in $\mathbb{R}^d$ instead of cubical sets. Since most of the examinations we did were on pixelated images, we chose a version of complex that is closer to thresholded planar images.

Crawford et al. further explain the ECT and the smooth Euler characteristic transform (SECT) [1]. They restrict the Euler curve to a compact interval $[a, b]$ based on the simplicial complex. And then, they take the mean of an Euler curve over the interval $[a, b]$ the and subtract the mean from the Euler curve. This produces a centered Euler curve. Integrating the center Euler curve gives a continuous piecewise linear function that has value 0 at the endpoints of the interval $[a, b]$.

Recall the Euler curves involved in the definition of the Euler characteristic transform. A Euler curve of a cubical set $X \subset \mathbb{R}^d$ in a certain direction is a function well-defined on $\mathbb{R}$. The paper [10] also treats the Euler curves as functions defined on $\mathbb{R}$. For practical purpose, we may want to consider truncated Euler curves.

3.2 Reconstruction of Shapes from Euler Curves

In addition to classifying shapes using the Euler curves, people also think about the reconstruction of shapes from the information stored in the persistent diagram and Euler curves [2, 4, 10].

The proof in [10] of the injectivity of the persistent diagram transform and Euler characteristic transform when the domain is the collection of sets in $\mathbb{R}^2$ or $\mathbb{R}^3$ that can be turned into simplicial complexes is moreover an algorithm by which we can

read the information of simplicial complexes (vertices, edges, faces, etc.) from the collection of persistent diagrams produced by the PHT.

Curry et al. generalize the injectivity theorem [2]. The theorem is stated as

Theorem 3.1 (Theorem 3.4 in [2]). *Let $CS(\mathbb{R}^d)$ be the set of constructible sets, i.e., compact definable sets. The map $ECT : CS(\mathbb{R}^d) \to CF(S^{d-1} \times \mathbb{R})$ is injective. Equivalently, if M and M' are two constructible sets that determined the same association of directions to Euler curves, then they are the same set. Symbolically:*

$$ECT(M) = ECT(M') : S^{d-1} \to CF(\mathbb{R}) \quad \Rightarrow \quad M = M'$$

The proof of the Theorem 3.1 is based on an inversion theorem of Schapira [9]. And another major result of the paper shows that any shape in a certain set of non-axis aligned shapes can be characterized by finitely many Euler curves.

Fasy et al. [4] worked on the collection of planar graphs, which are equivalent to simplicial complexes in $\mathbb{R}^2$, and showed that planar graphs can be reconstructed using a finite number of ECs instead of infinitely many Euler curve, each corresponding to a direction on S^1.

Chapter 4: Applications

4.1 Digital Image and Cubical Homology

In this section, we will talk about the relation between the shapes in digital images and cubical sets, and use some simple example to demonstrate how to convert a digital image to a cubical set.

4.1.1 Digital Images as Cubical Sets

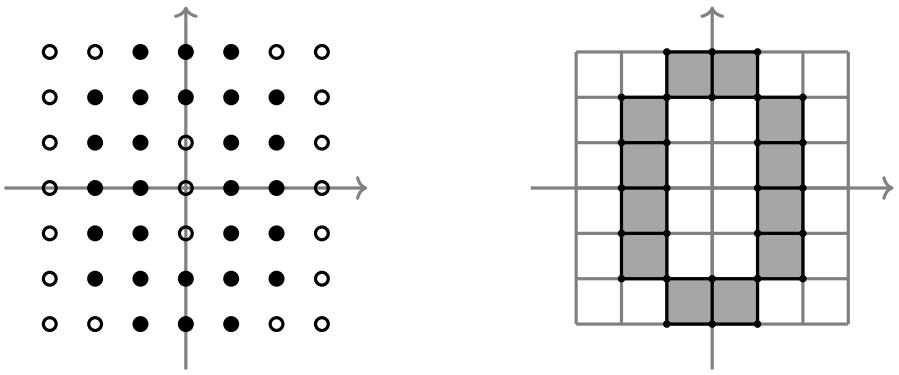
A ***digital image*** is a image made of pixels. The pixels can be stored in many different formats, such as an ordered long vector or a rectangular array, i.e., an m by n matrix. The image matrix has a natural coordinate system. We can say that the pixel stored at the ith row and the jth column has coordinate (i, j). The coordinate system is like a rotated coordinate system compared to the standard coordinate system in $\mathbb{R}^2$. For convenience, in this thesis we might want to think that we always store pixels in an m by n matrix.

If we rotate the coordinate system in mind and say that the coordinate (i, j) of a pixel is just the coordinate of a vector in $\mathbb{R}^2$ with respect to the standard basis, then it allows us to identify each pixel with a vector in $\mathbb{R}^2$ and put each pixel on the integer lattice $\mathbb{Z}^2$ in $\mathbb{R}^2$.

The information that a digital image carries is not only the position of the pixels, otherwise images are all dots arranged in rectangular shapes with differences only in size. Moreover, the grayscale images, which are the digital images we are caring about in this paper, carries also the intensity information. Each pixel of a grayscale image has a ***intensity value***, which is a real number ranged from 0 to 255. The number represents the amount of light. Therefore, 0 means completely dark and 255 means the brightest. The ***binary image*** which has only intensity values 0 and 255 is an extreme case of the grayscale image. We can always turn a grayscale image into a binary image by thresholding it, that is, if the intensity value is less than or equal to the threshold number, we change the intensity value to 0, and if the intensity value is greater than the threshold number, we change it to 255. We can further make those pixels have values only 0 and 1. If the pixel has value 1, we record the location information, the coordinate, of the pixel. Otherwise, we ignore the pixel completely.

We therefore obtain a set of pixels. The pixels depict the shape of the object in the digital image. We also obtain a set of vectors in $\mathbb{R}^2$. The vectors can be thought of as the set of vertices of some cubical set. Any two adjacent vertices determine an edge, or the elementary 1-cubes. The area enclosed by four edges are the squares or 2-cubes. This is the process of turning a digital image into a cubical set. Additionally, if we do not satisfy with the fact that cubical sets are only located in the first quadrant of the plane, we can translate the approximate center of the image to the origin or translate the cubical set.

Let us look at an example.



(a) Binary image (b) Cubical set

Figure 4.1: Turning a binary image into a cubical set

Example 4.1. Suppose we have a 7 by 7 binary image stored as the following matrix

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

We associate a Cartesian coordinate to each of the pixels in the way we described above. We then translate the pixels to make sure that the pixel that previously has coordinate $(4, 4)$ to be centered at the origin now. As shown in the Figure 4.1a, the filled dots represent pixels that have value 1, and the empty dots represents pixels that have value 0. We consider the filled dots as vertices. If any of the four neighbors (left, right, top, and bottom) of a certain vertex is also a filled dot, we connect those two vertices by an edge. And whenever we have four edges enclosing a square area of side length 1, we fill the area with a square. We obtain the cubical set illustrated in the Figure 4.1b. Denote the cubical set by O. Then

$$\mathcal{K}_0(O) = \{[-2]\times[-2], [-1]\times[-3], [-2]\times[-1], [-1]\times[-2], [0]\times[-3],$$
$$[-2]\times[0], [-1]\times[-1], [0]\times[-2], [1]\times[-3], [-2]\times[1],$$
$$[-1]\times[0], [0]\times[-1], [-2]\times[2], [-1]\times[1], [1]\times[-1],$$
$$[2]\times[-2], [-1]\times[2], [1]\times[0], [2]\times[-1], [-1]\times[3],$$
$$[0]\times[2], [1]\times[1], [2]\times[0], [0]\times[3], [2]\times[1],$$
$$[1]\times[2], [1]\times[3], [2]\times[2]\},$$

In this manner, we can write out the $\mathcal{K}_1(O)$ and $\mathcal{K}_2(O)$ sets, but it seems tedious. We just list some of the elements:

$$\mathcal{K}_1(O) = \{[-2]\times[-2,-1], [-2]\times[-1,0], [-2]\times[0,1], [-2]\times[1,2], \ldots$$
$$[-1,0]\times[-3], [0,1]\times[-3], \ldots\},$$
$$\mathcal{K}_2(O) = \{[-2,-1]\times[-2,-1], [-2,-1]\times[-1,0], [-2,-1]\times[0,1], \ldots\}.$$

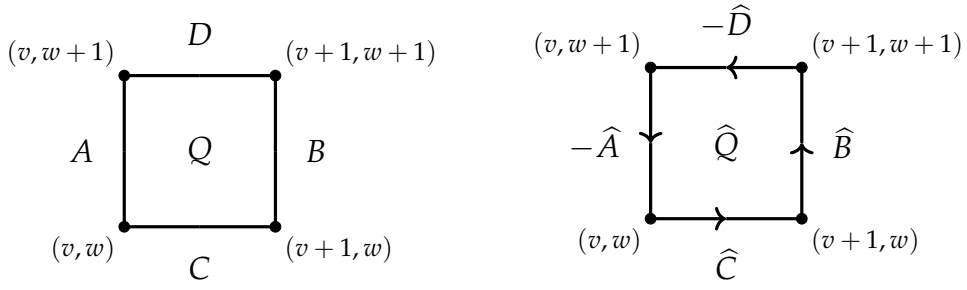
The cubical set O is the made out of these elementary cubes.

The way we exhibit above was just one way of turning a image into a cubical set. We will further demonstrate the above process and some variations in application contexts in subsection 4.2.2. The algorithms of producing cubical complexes is provided in Appendix A.

4.1.2 Algebra of Cubical Sets

We would also like to show an example of applying the boundary map to an elementary 2-cube.

Example 4.2. Let $Q = [v, v+1]\times[w, w+1]$ as shown in the Figure 4.2a. The



(a) Elementary cube (b) Geometric interpretation of a chain

Figure 4.2: Example of an elementary 2-cube and a chain dual to it

elements in $\mathcal{K}_1(Q)$ are

$$A = [v] \times [w, w+1],$$

$$B = [v+1] \times [w, w+1],$$

$$C = [v, v+1] \times [w],$$

$$D = [v, v+1] \times [w+1].$$

The corresponding elementary 1-chain dual to these elementary intervals are

$$\widehat{A} = [v] \times \widehat{[w, w+1]},$$

$$\widehat{B} = \widehat{[v+1] \times [w}, w+1],$$

$$\widehat{C} = [v, \widehat{v+1]} \times [w],$$

$$\widehat{D} = [v, \widehat{v+1] \times [w+1}].$$

Recall the definition of the boundary map ∂_2: for $\widehat{Q} = \widehat{I}_1 \diamond \widehat{I}_2$,

$$\partial_2 \widehat{Q} := \partial_1 \widehat{I}_1 \diamond \widehat{I}_2 + (-1)^{\dim I_1} \widehat{I}_1 \diamond \partial_1 \widehat{I}_2,$$

So we can calculate

$$\begin{aligned}
\partial_2\widehat{Q} &= \partial_1\widehat{[v, v+1]} \diamond \widehat{[w, w+1]} + (-1)^{\dim \widehat{[v,v+1]}} \widehat{[v, v+1]} \diamond \partial_1\widehat{[w, w+1]} \\
&= \left(\widehat{[v+1]} - \widehat{[v]}\right) \diamond \widehat{[w, w+1]} - \widehat{[v, v+1]} \diamond \left(\widehat{[w+1]} - \widehat{[w]}\right) \\
&= \widehat{[v+1]} \diamond \widehat{[w, w+1]} - \widehat{[v]} \diamond \widehat{[w, w+1]} - \widehat{[v, v+1]} \diamond \widehat{[w+1]} + \widehat{[v, v+1]} \diamond \widehat{[w]} \\
&= \widehat{B} - \widehat{A} - \widehat{D} + \widehat{C}.
\end{aligned}$$

From the above calculation, we see that the image of an elementary 2-chain under the boundary map ∂_2 is a 1-chain $c = \widehat{B} - \widehat{A} - \widehat{D} + \widehat{C}$. $|c|$ is the contour of the square Q. This might suggest that the algebraic and topological boundaries are closely related, that is $\left|\partial\widehat{Q}\right| = \mathrm{bd}\left|\widehat{Q}\right| = |c|$. The plus or minus sign in the expression for c has also a geometric interpretation: For example, we think of $\widehat{A}$ as indicating moving along the edge from (v, w) to $(v, w+1)$ while $-\widehat{A}$ means traversing the edge in the opposite direction. The direction is from the "lower vertex" to the "upper vertex" for positive elementary 1-chains and reverse for negative elementary 1-chains. So, c represents a counterclockwise closed path around the square as shown in 4.2b.

4.2 Clustering Random Sampling from MNIST Dataset

The MNIST database is a database of handwritten digits. It consists of 60,000 training images and 10,000 testing images [6]. There is a variation of the MNIST dataset called Fashion-MNIST, which is a dataset consisting of gray scale images of 70,000 fashion products [11]. We will choose a subset of the MNIST dataset and a subset of the Fashion-MNIST dataset to illustrate how to use the Euler characteristic transform to cluster shapes in image data.

4.2.1 Preprocessing Dataset

The first step is to process the images in dataset. The digital images in both MNIST and Fashion MNIST dataset are grayscale images, we want to turn them into binary images. There are many standardized functions or build-in functions in many different languages to load in and preprocess the dataset, for instance, the binarizing function IMBINARIZE in MATLAB. These tools help us convert the images in the MNIST dataset to desired binary images for building cubical complexes. By default, IMBINARIZE uses Otsu's method, which chooses the threshold value to minimize the intraclass variance of the thresholded black and white pixels [8]. And we can choose different threshold values as well. Different ways of binarizing the same digital image will result in different cubical complexes, hence different homology groups.

4.2.2 Turning Images to Cubical Sets

Having obtained binary images from the dataset, we then consider how to turn them into cubical sets. Recalling in subsection 4.1.1, we treated each nonzero pixel as a vector in $\mathbb{R}^2$ and as a vertex. Alternatively, each nonzero pixel can be thought of as a square. The coordinate of the pixel is the coordinate of the center of the square. To make sure that two squares can intersect only at edges or vertices, it is reasonable to choose the length of the sides of the squares to be 1. Hence, the coordinate of any vertex is of the form $(i + 1/2, j + 1/2)$ for some $i, j \in \mathbb{Z}$. In this paper, we think of pixels as vertices.

Given a binary image matrix, we can easily find a list of coordinates of the nonzero vertices, for example, by means of the FIND function in MATLAB. Next,

for a vertex that has coordinate (i, j) with i and j in this list, we consider the two pixels with coordinates $(i+1, j)$ and $(i, j+1)$ respectively. If say $(i+1, j)$ is nonzero, we want to connect (i, j) and $(i+1, j)$ to form an edge. We record the two coordinates in the list of edges. Always considering $(i+1, j)$ and $(i, j+1)$ is just one way to avoid repeated counting of edges. We do the process for each vertex in the list of vertices to get a list storing the coordinate information of all desired edges. Finally, for a vertex with coordinate (i, j) in this list, we consider the three pixels $(i+1, j)$, $(i, j+1$ and $(i+1, j+1)$. If $(i+1, j) = (i, j+1) = (i+1, j+1) = 1$, the four vertices form a square. We add the four coordinates to the list of squares. Considering for each vertex in the vertex list, we can get list of all squares. The details are in Algorithm 1.

4.2.3 Euler Curves

With the information of cubical complexes, we can compute combinatorial or topological invariants, such as Betti numbers and Euler characteristics. Recalling in Chapter 2 section 2.2, we defined the Euler characteristic and Euler curves for general cubical set in $\mathbb{R}^d$. The Euler characteristic of a cubical set is defined to be the alternating sum of the nth Betti numbers. In computation, calculating the rank of cubical homology groups involves reduction of some large matrices, which takes a lot of time. On the other hand, counting the elementary cubes is way faster than applying elementary operations to matrices. To easily acquire the Euler characteristic, we can use the following definition:

Definition 4.1. Let $X \subset \mathbb{R}^d$ be a cubical set, and n_k be the number of the kth elementary cubes. The ***Euler characteristic*** of X is defined to be

$$\chi(X) := \sum_{k \in \mathbb{Z}} (-1)^k n_k.$$

The definition of Euler curve indicates the algorithm of how to produce Euler curves from cubical sets. For a cubical set X, we obtained the lists of vertices, edges, and faces of X from Algorithm 1. Let the list of vertices V be a n_v by 2 matrix, where n_v is the number of vertices. Then each row of V stores the coordinate of a vertex. Similarly, let the list of edges E be a n_e by 4 matrix, where n_e is the number of edges, and let the list of squares F be a n_f by 8 matrix, where n_f is the number of squares. So, each row of E stores 2 coordinates of two vertex and each row of F stores 4 coordinates. Let $u = (u_1, u_2)^T$ be a direction vector as a column vector. A matrix multiplication of V and u can quickly yield a list of dot products of v and u. This is a list of 'heights' we assigned to each v in V. The list of edges actually has two lists of vertices, say V_1 and V_2. Multiplying V_1, V_2 with u, respectively, we find two lists of dot products, say A and B. The MATLAB code $\text{MAX}(A, B)$ will return the largest element from each row of A and B. In this way, we assign the 'height' value $\max\{v \cdot u : v \in e\}$ to e for every e in E. In a similar fashion, we assign height to each of the faces in F.

Next, we set an interval $[m, M]$. So the Euler curve will start at height m and end at height M. The number of increments n is also necessary. Once m, M and n are given, the increment on h-axis is of course $\Delta h = \frac{M-m}{n-1}$. So, at $h_i = m + i\Delta h$ for $i \in \{0, \ldots, n-1\}$, we want to find the Euler characteristic $\chi(X_{(u,h_i)})$. So, using the three lists of heights of elementary cubes, we can find the number nv of elementary 0-cubes with height of $\leq h_i$, the number ne of elementary 1-cubes with height of

$\leq h_i$, and the number nf of elementary 2-cubes with height of $\leq h_i$. It follows that $\chi(X_{(u,h_i)}) = nv - ne + nf$. A **for** loop can give $\chi(X_{(u,h_i)})$ for all i. The details can be found in Algorithm 2.

4.2.4 Distance Between ECT

We now know how to calculate the Euler curve of a cubical set X for a fixed direction u. We can then evenly sample a number of directions from S^1 and compute a bunch of Euler curves. We obtain the Euler characteristic transform of X. We want to further smooth the Euler curves, for example, using the MATLAB function SMOOTHDATA$(EC, '\mathit{gaussian}')$. After smoothing, we calculate the distance between the SECTs of two cubical sets, i.e., two images, by the way suggested in the paper [1]. The distance function is given by

$$\text{dist}(SECT(X_1), SECT(X_2)) := \left(\int_{S_1} \left\| EC_{u,X_1} - EC_{u,X_2} \right\|^2 du \right)^{1/2}. \tag{4.1}$$

With this distance function, we can try some distance-based clustering.

4.2.5 Clustering

At first, we randomly selected 30 samples from the digit MNIST dataset. We applied the Otsu's method in the thresholding process. We used Algorithm 1 to turn the sampled images into cubical complexes and used Algorithm 2 to compute 20 (corresponding to 20 directions evenly sampled from S^1) Euler curves for each sample. We smoothed them and calculated pairwise distance between any two $SECT(X_i)$ and $SECT(X_j)$ based on equation 4.1. We got a 30 by 30 distance matrix D.

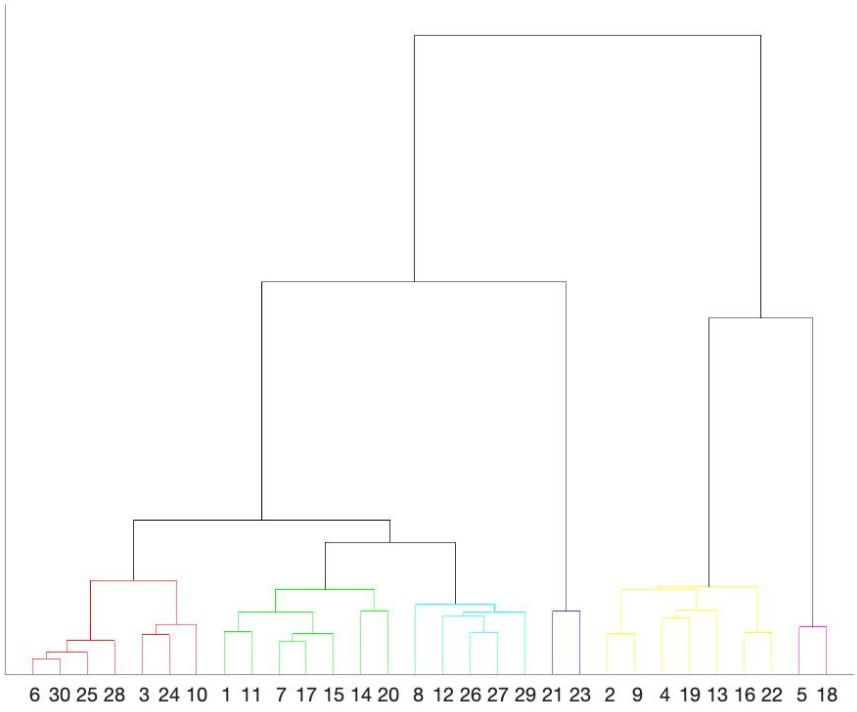


Figure 4.3: Colored dendrogram for 30 digits

The MATLAB function LINKAGE$(D,'ward')$ helps carry out the hierarchical cluster analysis with ward linkage on D. The clustering dendrogram is shown in Figure 4.3. The plot indicates that there are two main clusters. By reading the leaf nodes of the cluster on the right-hand side, we see that this cluster contains almost all of the digits 0 and 9, and all of the digits 6 and 8. On the other hand, the left cluster contains mostly the digits that do not have a cycle. However, this is a relatively less interesting result because by simply calculating the Euler characteristic of each digit, we can get very similar clustering result instead of doing the ECT. So we may want to consider sub-clusters of these two main clusters to see if

the ECT could tell more information. We chose to examine the 6 sub-clusters. In Figure 4.3, the six colored clusters from left to right are cluster 3, cluster 2, cluster 1, cluster 6, cluster 5, and cluster 4. We read these cluster numbers from the Figure 4.4. The orders of the digit images from left to right, top to bottom in Figure 4.4 are corresponding to the numbers on the leaf nodes of the dendrogram in Figure 4.3.

A reorganized list of digits is shown in the Figure 4.5. We can read off some interesting points from sub-clusters. First, we note that the break-up digit 0 has Euler characteristic 1. Clusters 1, 2 and 3 all contains digits with characteristic 1. The zero was not clustered into other clusters but the cluster 2 with five other digit 4s. This may be because the cut appears on the top side of the loop. So when we sweep from the top side of the image to bottom side, the filtration contains subcomplexes with two connected components in the beginning, like the situation that we sweep over digit 4 from top to bottom. We may suspect that if the cut of the loop of 0 appeared on the left side, then the zero might be clustered into cluster 1 together with those 3s.

Secondly, a narrow 3 was clustered with many other 1s, whereas wide 3s are more similar to 5. This might suggest that the ECT also carries some geometric information in addition to topological information. The third point was observed by a careful inspection on both the clusters and images. This process is sensitive to tiny pixels. A small cut on the loop of the zero changes the topological structure. By zooming in the middle digit 6 in cluster 5 in Figure 4.5, we can see a tiny empty pixel. Once this pixel gets filled, again the topology will suddenly change. And in cluster 7 in Figure 4.5, we can clearly see that the first digit 5 has two components,

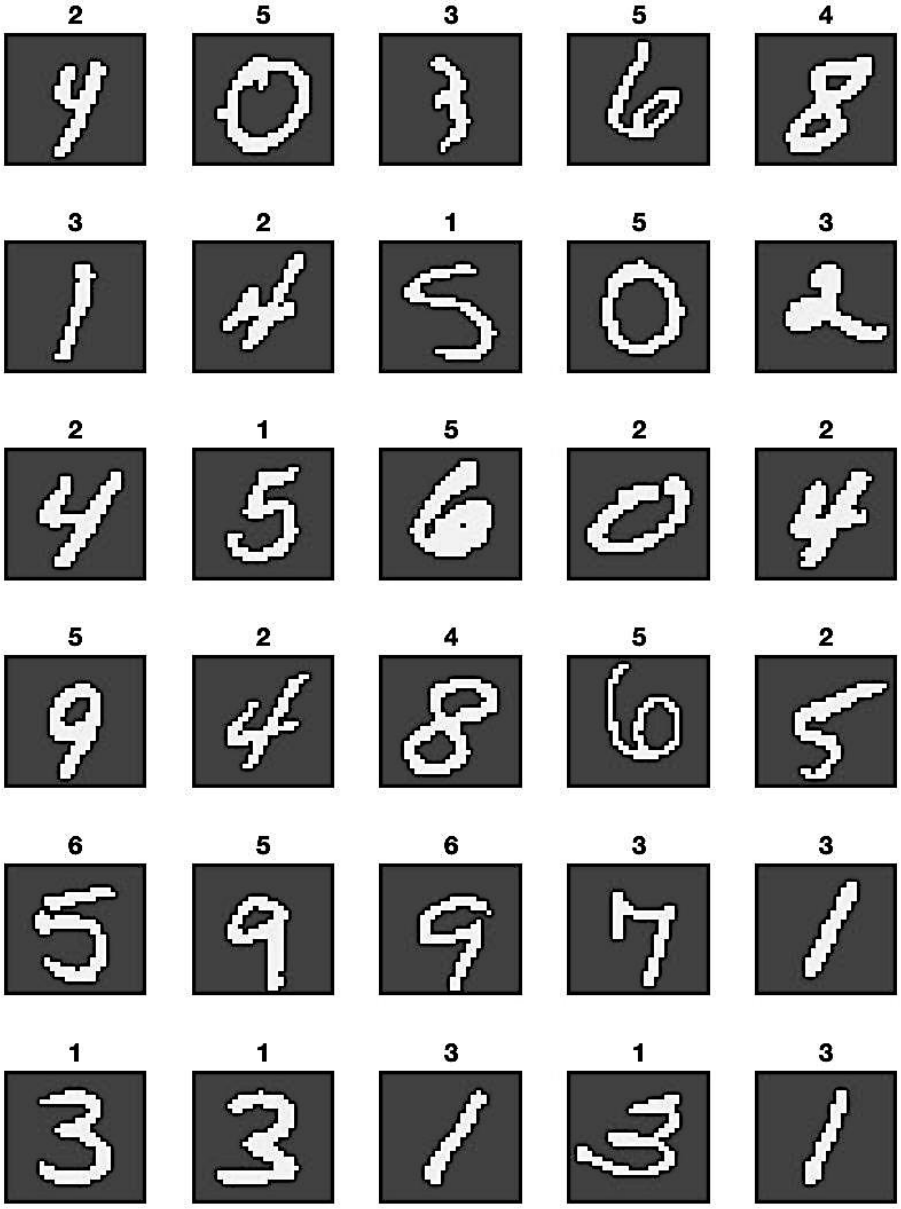


Figure 4.4: 30 samples from MNIST dataset with clusters labeled on top.

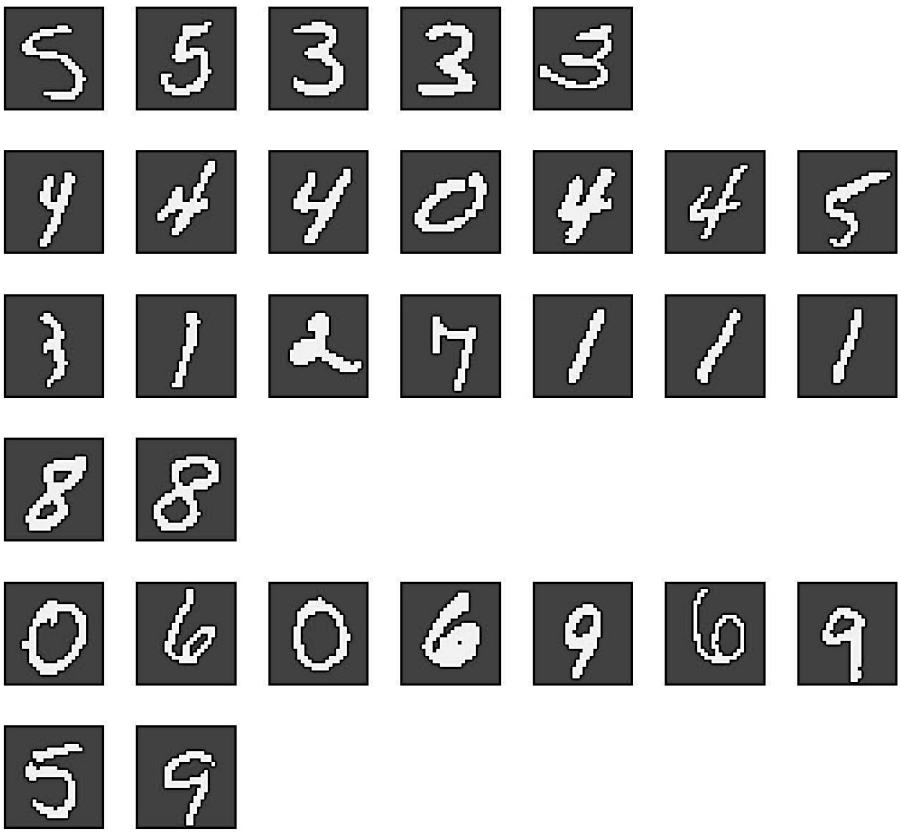


Figure 4.5: Cluster 1 to 6 listed from top to bottom

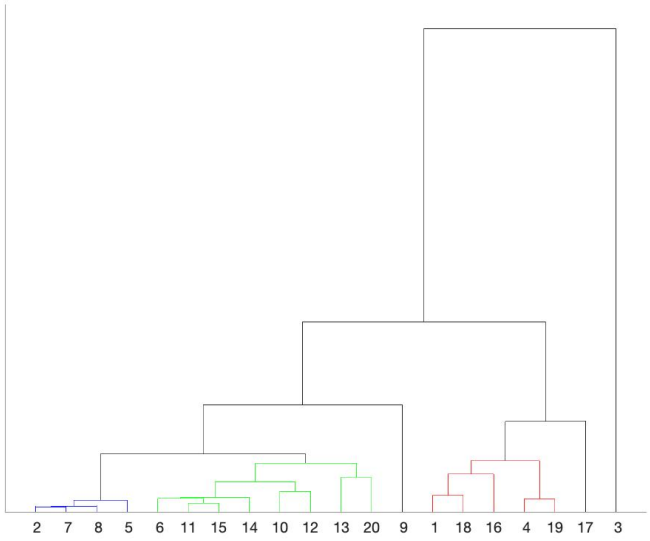
while the second connected component of the digit 9 is almost invisible to see. However, the Euler characteristic transform can detect this.

This experiment on MNIST handwriting dataset might help us understand the Euler characteristic transform somehow. We may also want to conduct another experiment on the Fashion MNIST dataset especially to see how different methods of binarizing will affect the clustering result.

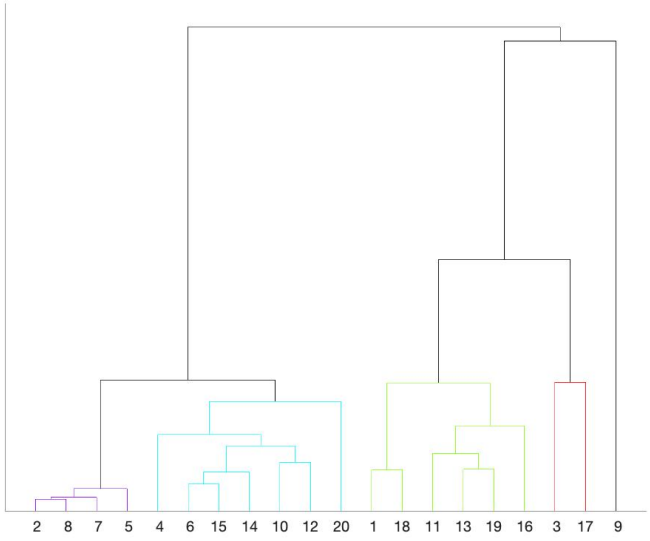
Figure 4.6: 20 images from Fashion MNIST dataset

We selected 20 samples from the Fashion MNIST dataset as shown in Figure 4.6. Since Otsu's method may result in some information loss in this dataset, we chose to threshold by some fixed numbers. How to determine a thresholding number is an important question. We tried to threshold by 0.05 and 0.1 (the images in the dataset have been normalized to have intensity values ranged from 0 to 1) to see some results. Since we did binirizing process twice differently, we label these two sub-experiments by 2A and 2B, respectively. We used Algorithms 1 and 2 to compute 20 Euler curves for each of the 20 cubical sets in each time. After computing distance, we got two distance matrices. We did the hierarchical cluster analysis using the same method. And the Figure 4.7a and 4.7b are the dendrograms of the clusterings from experiment 2A and 2B, respectively.

We still examine sub-clusters in details. In experiment 2A, we chose to see 6 sub-clusters, and in experiment 2B, we chose to see 5 clusters. In Figure 4.7a and 4.8b, we can see where each image got clustered.

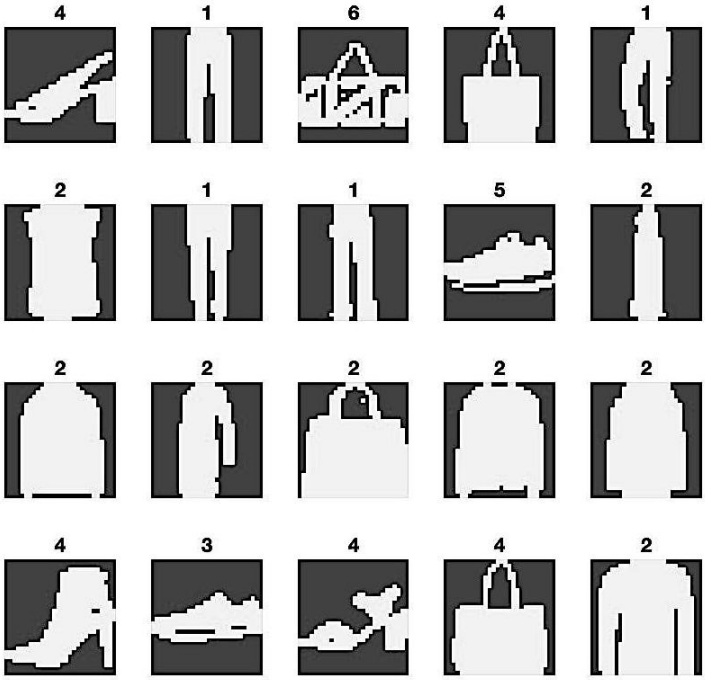


(a) Clustering dendrogram in experiment 2A

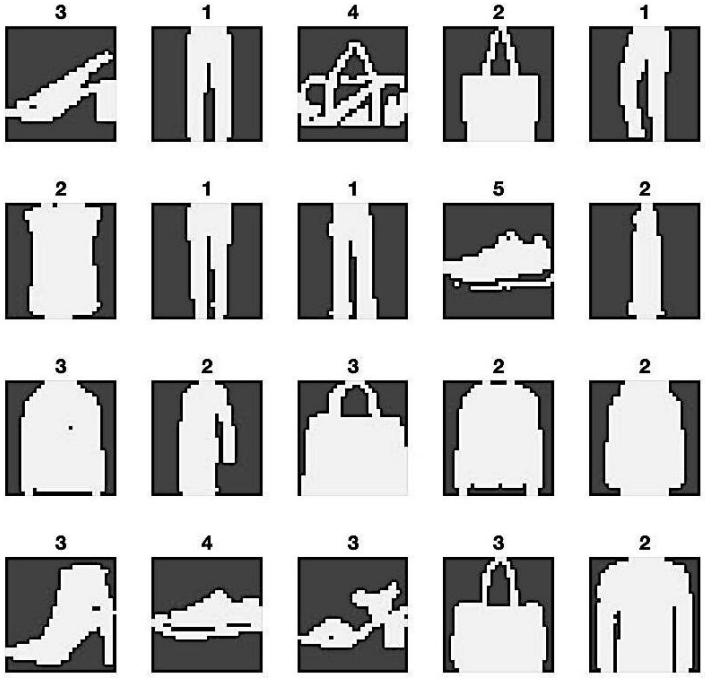


(b) Clustering dendrogram in experiment 2B

Figure 4.7: Colored dendrograms in experiment 2A and 2B



(a) Clustering result in experiment 2A



(b) Clustering result in experiment 2B

Figure 4.8: The binary images with cluster label on top in experiments 2A and 2B

There are some notable points. Increasing thresholding value from 0.05 to 0.1 changed the cubical complexes. The first example is the bag in the fourth column of the first row. Increasing threshold resulted in a cut on the band. So the bag does not have degree one homology in experiment 2B any more. It was clustered into the group in which other images clearly have characteristic 1. The second example is the jacket in the first column of the third row. Increasing threshold created a small hole in it. So it was clustered with the bags with a band as well as the shoes with a hole. The third example is the bag in the third column of the third row. When taking threshold value 0.05, there was a pixel in the background. So the cubical set had two components. While in experiment 2B, the pixel in the background got ignored. The above three examples are typical examples of homology changing owing to different thresholding values.

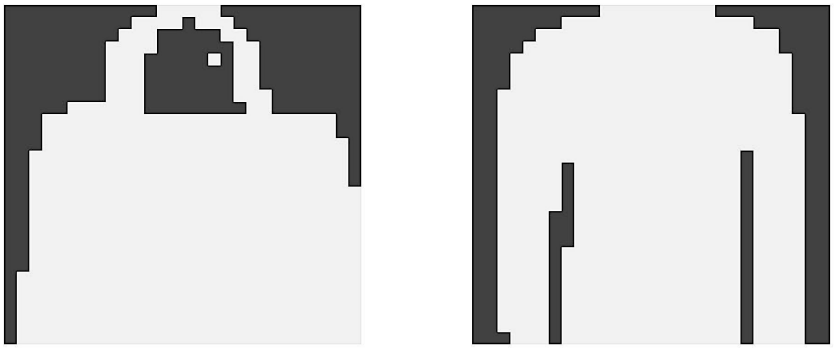
On the other hand, we still see some positive results. For example, the trousers and shirts are clustered into different clusters although they all have Euler characteristic 1. But we might expect a performance downgrade when we went from a relatively simple MNIST dataset to the Fashion MNIST dataset.

Chapter 5: Future Works

We found in experiment 2A that the handbag in the third column of the third row (the 13th sample) and the shirt in the last column of the last row (the 20th sample) were clustered into the same cluster, cluster 2. And we also talked about that when using the threshold value 0.05 in experiment 2A, it will result in an isolated pixel in the background of the image of the bag. Regarding this issue, we talk about an upgraded method in this final chapter.

5.1 Betti Number Curves

We can use what we have learned about cubical homology to get some Betti numbers. If we compute the kth Betti number of each subcomplex of a filtration of some cubical set, we can similarly produce the kth Betti number curve of a cubical set. For example, we take the two images from the experiment 2A in subsection 4.2.5, as shown in Figure 5.1. The first image is the 13th sample, and the second image is the 20th sample. If we keep subdividing the cluster 2 in experiment 2A, we can see from the dendrogram in the Figure 4.7a that these two images are clustered together into a finer sub-cluster, separated from other samples in cluster 2. Let us see some Euler characteristic curves in some directions for the two images in Figure 5.1a and Figure 5.1b.

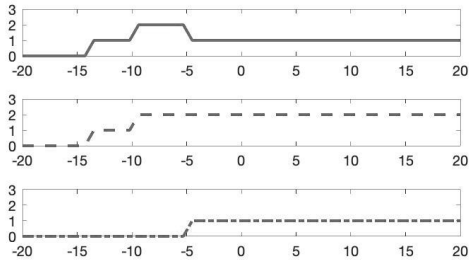


(a) The 13th image (b) The 20th image

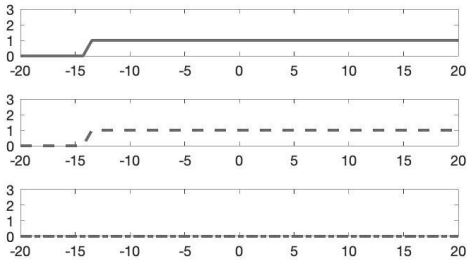
Figure 5.1: Two images from experiment 2A

In Figure 5.2, we present some Euler curves and Betti curves. The four sub-figures 5.2a, 5.2c, 5.2e, and 5.2g in Figure 5.2 contain the Euler curves, the degree 0 Betti curves, and the degree 1 Betti curves for the image in Figure 5.1a in directions $(0,-1)$, $(1,0)$, $(0,1)$, and $(-1,0)$, respectively. Within each sub-figure, the solid curve is the Euler curve, the dashed curve is the degree 0 Betti curve, and the dash-dotted curve is the degree 1 Betti curve. Similarly, the four sub-figures 5.2b, 5.2d, 5.2f, and 5.2h contains the same kinds of curves for the image in Figure 5.1b. Fix any of the four directions, we compare the two Euler curves for two images. And we can indeed see that the two Euler curves are similar in each of the four directions. On the other hand, we could see significant differences (in the sense that the $L2$-distance is large) when comparing the Betti number curves for a fixed direction.

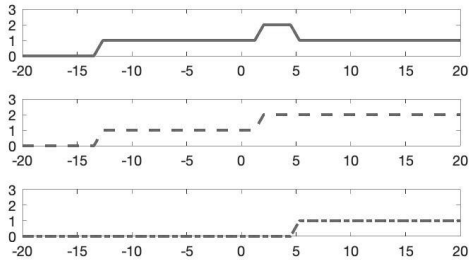
The future works may involve finding faster algorithms to produce Betti number curves and carrying out similar clustering analysis on the dataset.



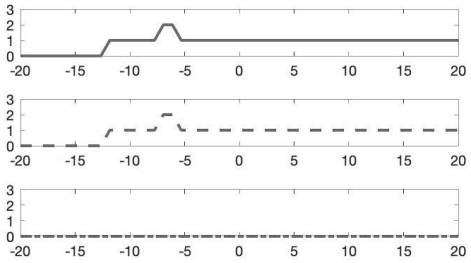
(a) top-down



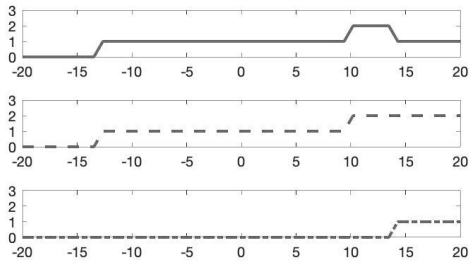
(b) top-down



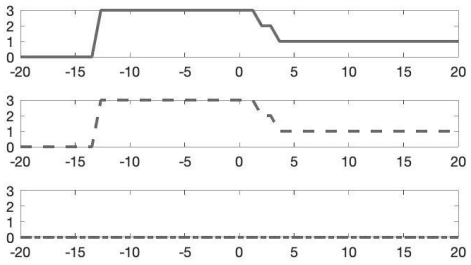
(c) left-right



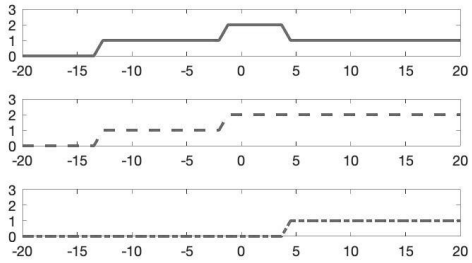
(d) left-right



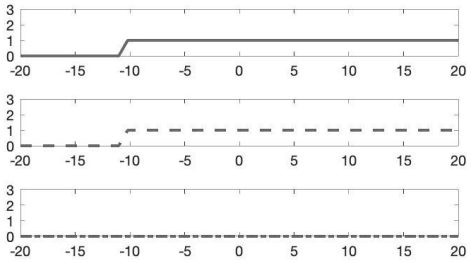
(e) bottom-up



(f) bottom-up



(g) right-left



(h) right-left

Figure 5.2: Comparison of Betti number curves and Euler number curves

Appendix A: Algorithms

Algorithm 1 Binary Image to Cubical Complex

function CUBICALCPLX(A) ▷ $A \in \mathbb{Z}_2^{m \times n}$
  $V = \text{FIND}(A \neq 0)$
  for all $v \in V$ **do** ▷ $v = (i, j)$
    if $(i+1, j) = 1$ **then**
      append e to E ▷ $e = [i, i+1] \times [j]$
    end if
    if $(i, j+1) = 1$ **then**
      append e to E ▷ $e = [i] \times [j, j+1]$
    end if
  end for
  for all $v \in V$ **do**
    if $(i+1, j) = 1$ & $(i, j+1) = 1$ & $(i+1, j+1) = 1$ **then**
      append f to F ▷ $f = [i, i+1] \times [j, j+1]$
    end if
  end for
  return V,E,F
end function

Algorithm 2 Cubical Complex to Euler Curve

1: **function** EULERCURVE(V, E, F, u, m, M, n) ▷ $V = n_V$ by 2 matrix
2: ▷ $E = n_E$ by 4 matrix
3: ▷ $F = n_F$ by 8 matrix
4: ▷ $u =$ column vector in $\mathbb{R}^2$
5: ▷ $[m, M] =$ domain of EC
6: ▷ $n =$ number of increments
7: $hV = V * u$ ▷ $*$ is matrix multiplication
8: $hE = \max([E_1\ E_2] * u, [E_3\ E_4] * u)$
9: $hF = \max(\max([F_1\ F_2] * u, [F_3\ F_4] * u), \max([F_5\ F_6] * u, [F_7\ F_8] * u))$
10: $N = m : (M - m)/(n - 1) : M$
11: **for** $s = 1 : n$ **do**
12: $v = hV(\text{FIND}(hV \leq N(s)))$; $nv = \text{LENGTH}(v)$
13: $e = hE(\text{FIND}(hE \leq N(s)))$; $ne = \text{LENGTH}(e)$
14: $f = hF(\text{FIND}(hF \leq N(s)))$; $nf = \text{LENGTH}(f)$
15: $EC(s) = nv - ne + nf$
16: **end for**
17: **return** EC
18: **end function**

Bibliography

[1] L. Crawford, A. Monod, A. X. Chen, S. Mukherjee, and R. Rabadán. Predicting clinical outcomes in glioblastoma: An application of topological and functional data analysis. *Journal of the American Statistical Association*, 00(0):1–12, 2019.

[2] J. Curry, S. Mukherjee, and K. Turner. How many directions determine a shape and other sufficiency results for two topological transforms, 2018.

[3] H. Edelsbrunner and J. Harer. *Computational Topology: An Introduction*. Applied Mathematics. American Mathematical Society, 2010.

[4] B. T. Fasy, S. Micka, D. L. Millman, A Schenfisch, and L. Williams. Challenges in reconstructing shapes from euler characteristic curves, 2018.

[5] T. Kaczynski, K. Mischaikow, and M. Mrozek. *Computational Homology*. Springer, New York, NY, First edition, 2004.

[6] Y. LeCun, C. Cortes, and C.J.C. Burges. The mnist database of handwritten digits.

[7] Y. Mileyko, S. Mukherjee, and J. Harer. Probability measures on the space of persistence diagrams. *Inverse Problems*, 27(12):124007, nov 2011.

[8] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, Jan 1979.

[9] P. Schapira. Tomography of constructible functions. In *Proceedings of the 11th International Symposium on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, AAECC-11, page 427–435, Berlin, Heidelberg, 1995. Springer-Verlag.

[10] K. Turner, S. Mukherjee, and D. M. Boyer. Persistent homology transform for modeling shapes and surfaces. *Information & Inference: A Journal of the IMA*, 3(4):310 – 344, 2014.

[11] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[12] A. Zomorodian and G. Carlsson. Computing persistent homology. In *Proceedings of the Twentieth Annual Symposium on Computational Geometry*, SCG '04, page 347–356, New York, NY, USA, 2004. Association for Computing Machinery.