INTRODUCTION TO SPIN TRANSFORMATION
AND ITS APPLICATION ON SHAPE DESCRIPTOR

Derek Liu
May 10, 2016
CONTENTS

1 Introduction 1

2 Background 1
  2.1 Mean Curvature ...................................................... 2
  2.2 Conformal Transformation ........................................ 3
  2.3 Quaternions ........................................................... 4
  2.4 Quaternionic Dirac Operator ....................................... 5

3 Spin Transformation 6
  3.1 Smooth Spin Transformation ...................................... 6
  3.2 Mean Curvature Half-Density .................................... 8
  3.2 Discrete Spin Transformation .................................... 9

4 Applications - Shape Descriptor 11
  4.1 Shape Descriptor Overview ...................................... 12
  4.2 Spin-based Shape Descriptor .................................... 13

5 Conclusion and Outlook 18

6 Reference 18
1 INTRODUCTION

3D objects play a vital role in computer games, movies and engineering design. The advancement of scanning technology, modeling and big data has led to a huge growth of 3D shape repositories, such as Princeton Shape Benchmark [SMKF04]. With a collection of shapes, shape analysis tasks, such as classification and segmentation, rely on similarity measure between shapes. However, most of the existing shape data are represented in point cloud or polygonal meshes which are extremely difficult to compare. Therefore, to make shapes comparison tractable, we need to transform 3D shapes into different representations.

The research topic of mapping shapes to different feature spaces is called shape descriptor design. The most common approach utilizes surface properties, such as curvature and geodesic distance, to construct shape descriptors. Another popular approach uses shape operators, such as Laplace-Beltrami operator [WMKG07], to "describe" shape as a linear combination of basis functions. However, most descriptors were built heuristically and their performance is strongly tasks dependent. Therefore, the project asks which property may be the most proper way to represent a 3D shape? This article tries to find the answer and design a shape descriptor out of it.

Bonnet proposed that mean curvature and metric should suffice to determine the surface generically [Bo67]. The two geometric objects are building blocks of conformal transformation and spin transformation in differential geometry [KPP98]. Therefore, the article aims to introduce spin transformation and its application on shape descriptor. In Chapter 2, we describe background knowledge of spin transformation. Chapter 3 focuses on spin transformation in both smooth and discrete settings. Chapter 4 briefly reviews research in shape descriptors and puts emphasis on the idea of spin-based shape descriptor. Finally, Chapter 5 concludes shape descriptor design and future work on developing spin-based shape descriptor.

2 BACKGROUND

Spin Transformations are all about: conformal equivalence of immersed surfaces, and it can be studied by a mathematical tool Quaternions. Besides, the shape space of spin transformation is generically parameterized by the scale invariant version of mean curvature, or mean curvature half-density, and is related to the eigenvalues of the quaternionic Dirac Operator. In short, curvature and conformal transformation are building blocks of spin transformation. Quaternions and Quaternionic Dirac operator are proper mathematical tools to realize it. In this chapter, we will introduce some aspects of these topics relevant to spin transformation.
2.1 Mean Curvature

Mean curvature is a fundamental geometric property closely relevant to an essential object in spin transformation - mean curvature half-density. When we move along a curve $\gamma(X)$, curvature $\kappa$ expresses the infinitesimal change of tangent direction, $T^{t+\delta t} - T^t$. Because we know that direction of the change is the same as normal direction, we can define curvature as:

$$dT(X) = -\kappa N$$

To visualize curvature, we assume it takes $2\pi r$ time to go all the way around the circle and tangent direction changes $2\pi$ during the time. Since $T$ and $N$ have unit length, so we end up with:

$$|dT(X)| = \frac{2\pi}{2\pi r} = 1/r$$

$$| - \kappa N | = \kappa$$

$$\therefore \kappa = 1/r$$

Therefore, visualization of curvature could be a circle $S$ with radius $r$, which best locally approximates the curve $\gamma$ with the relationship $\kappa = 1/r$.

Source: Crane, K. Discrete Differential Geometry: An Applied Introduction
When it comes to define curvature on a surface, we can simply utilize the same idea by cutting the surface along each direction and fitting the curve on the cutting plane with a circle to compute curvature. And it can be computed using the following formula [Cr15]

\[ \kappa_n(X) = \frac{df(X) \cdot dN(X)}{|df(X)|^2} \]

\[ \rightarrow \kappa_i|df(X_i)|^2 = df(X) \cdot dN(X) \]

\[ \rightarrow \kappa_i df(X_i) = dN(X_i) \quad (1) \]

In which \( f \) is the map mapping plane region to curve surface, \( df(X) \) is a tangent vector on the surface, and \( dN(X) \) is a surface normal vector. One thing to mention is that the maximum and minimum curvatures \( \kappa_1 \) and \( \kappa_2 \) along the directions \( X_1 \) and \( X_2 \) are called principal curvatures. And mean curvature \( H \) is defined as arithmetic mean of principal curvatures

\[ H := \frac{\kappa_1 + \kappa_2}{2} \]

### 2.2 Conformal Transformation

Conformal transformation, also known as angle-preserving transformation, is a superclass of spin transformation. Conformal transformation only has rotation and scaling effect. Therefore, we can write conformal map as

\[ df(JX) = N \times df(X) \]

\( X \) is a vector on original surface, \( df(X) \) is a scaled version of vector \( X \) along tangent direction on transformed surface induced by \( f \). \( J \) is complex structure, similar to \( i \) in complex number system, which is a operation rotating vector 90-degree counter-clockwise. Because conformal transformation preserves angle, rotating \( X \) by 90-degree in original surface and map it to the transformed space should be the same as rotating \( df(X) \) by 90-degree on the transformed surface. And the rotated version of \( df(X) \) is written as the cross product of surface normal \( N \) and \( df(X) \) as shown in the picture below. Therefore, the equation above defines conformal transformation.

Source: Crane, K. Discrete Differential Geometry: An Applied Introduction
2.3 Quaternions

Quaternions is a number system which extends the idea of complex number system and it sim-
plifies rotation and scaling operation in 3D space. In this section, we are going to overview basic
operations in quaternions and introduce a nice property relevant to spin transformation.

Complex number, \( a + bi \), handles 2D. Quaternions are designed to manipulate quantities in 3D
or 4D, such as \( a + bi + cj + dk \). A fundamental rule of quaternions is this cycle shown below, which
leads to many operations similar to typical number system except Commutative.

\[
q = a + bi + cj + dk
\]
\[
\bar{q} = a - bi - cj - dk
\]
\[
i^2 = j^2 = k^2 = -1
\]
\[
ij = k \quad ji = -k
\]
\[
jk = i \quad kj = -i
\]
\[
ki = j \quad ik = -j
\]

Let’s assume quaternions \( q_1 = a_1 + b_1i + c_1j + d_1k = (a_1, \vec{v}_1) \) and \( q_2 = a_2 + b_2i + c_2j + d_2k = (a_2, \vec{v}_2) \).

We know that they act similar to the common number system in many operations, such as:

\[
q_1 + q_2 = (a_1 + a_2, \vec{v}_1 + \vec{v}_2)
\]
\[
q_1 - q_2 = (a_1 - a_2, \vec{v}_1 - \vec{v}_2)
\]
\[
q_1^{-1} = \left( \frac{a_1}{a_1^2 + |\vec{v}_1|^2}, \frac{-\vec{v}_1}{a_1^2 + |\vec{v}_1|^2} \right)
\]

The inverse leads to \( q_1q_1^{-1} = (1, \vec{0}) \) and \( a_1^2 + |\vec{v}_1|^2 \) is also known as the length of a quaternion.

Why quaternions is suitable for describing rotation comes from quaternions multiplication,
also known as Hamilton Product. When we multiply \( q_1q_2 \) term by term, it becomes:

\[
q_1q_2 = (a_1a_2 - \vec{v}_1 \cdot \vec{v}_2, a_1\vec{v}_2 + a_2\vec{v}_1 + \vec{v}_1 \times \vec{v}_2)
\]

As mentioned above, quaternions captures most of the basic properties of typical number system
besides commutative. We can prove it by

\[
q_1q_2 = (a_1a_2 - \vec{v}_1 \cdot \vec{v}_2, a_1\vec{v}_2 + a_2\vec{v}_1 + \vec{v}_1 \times \vec{v}_2)
\]
\[
q_2q_1 = (a_2a_1 - \vec{v}_2 \cdot \vec{v}_1, a_2\vec{v}_1 + a_1\vec{v}_2 + \vec{v}_2 \times \vec{v}_1) = (a_1a_2 - \vec{v}_1 \cdot \vec{v}_2, a_1\vec{v}_2 + a_2\vec{v}_1 + \vec{v}_2 \times \vec{v}_1)
\]

We can see the difference between \( q_1q_2 \) and \( q_2q_1 \) is the cross product at the end. This property
roughly captures the notion of orientation because \( \vec{v}_1 \times \vec{v}_2 = -\vec{v}_2 \times \vec{v}_1 \). Also, it captures one of the
main ideas in rotation - order matters. In other words, rotating a vector along y-axis first and then
rotating it along z-axis is not the same as rotating along z-axis first and then along y-axis. The non-
commutative property implies that quaternions may be an elegant mathematical tool to perform
rotation in 3D space. In the end, it is a mathematical tool simplifying rotation and our main
topic spin transformation. A concrete example of describing rotation of 3D geometry is *quaternions* trackball, please refer to [HSH04]. As a result, rotation in Euclidean space using quaternions can be simply written as [Chi98]:

$$x_{\text{rotate}} = \hat{x} \lambda$$  \hspace{1cm} (2)

$\lambda$ is quaternions and $x_{\text{rotate}}$ is the rotated version of $x$ in $\mathbb{R}^3$ space. If $|\lambda| = 1$, it is a pure rotation. If not, it is a rotation with scaling. Another thing worth mentioned is that *unit quaternions*, $|\lambda| = 1$, can be rewritten as $\lambda = (\cos(\theta/2), -\sin(\theta/2)u)$ which represents rotating by angle $\theta$ along axis $u$.

### 2.4 Quaternionic Dirac Operator

Quaternionic Dirac operator is another useful tool to compute spin transformation, and it is also widely applicable to general shape processing. According to [Cr13], Quaternionic Dirac Operator is defined as

$$D \lambda := -\frac{df \wedge d\lambda}{|df|^2}$$  \hspace{1cm} (3)

This operator is a *self-adjoint, elliptic* operator. A self-adjoint operator implies that it induces an orthonormal basis in *Hilbert space* and eigenvalues are real numbers. To get some intuition from the properties, we are going to discuss some basic concepts about hilbert space and self-adjoint operator.

Hilbert space is a vector space $\mathcal{H}$ with inner product $\langle \cdot, \cdot \rangle$ such that the norm is defined as $|f| = \langle f, f \rangle$. The norm defined here gives us the notion of “distance” in this space. A simple example would be $|x - y| = \langle x - y, x - y \rangle$. Assume $\{b_1, b_2, \ldots\}$ is a set of hilbert basis, then any vector $\vec{x} \in \mathcal{H}$ can be written as $\vec{x} = \sum_{i=1}^{\infty} c_i b_i$, in which $\sum |c_i| < \infty$ and the only way to get zero vector is with $c_1 = c_2 = \cdots = 0$. And the benefit of having orthonormal hilbert basis, which is what self-adjoint operator can give us, is that $c_i$ would be just the projection, inner product $\langle \vec{x}, b_i \rangle$, of $\vec{x}$ to the basis $b_i$. An example of hilbert space is infinite dimensional Euclidean space which basis $e_j = (0, \cdots, 0, 1, 0, \cdots, 0)$ and the 1 is in $j^{th}$ slot. Other example bases are like $\sin(jft/L)$ and $ti e^{-t^2/2}$. For more description about hilbert space, please refer to [Lax02].

In terms of self-adjoint operator, it is defined as

$$\langle v_2, Av_1 \rangle_2 = \langle A^* v_2, v_1 \rangle_1$$

In which $A : \mathcal{H}_1 \rightarrow \mathcal{H}_2$ and $A^* : \mathcal{H}_2 \rightarrow \mathcal{H}_1$. If it is in finite dimensional case, $A^*$ is the complex conjugate transpose of $A$. The two main properties are:

1. All the eigenvalues of $A$ are real.
2. Eigenvectors which corresponding to different eigenvalues are orthonormal.

A important fact from those properties is that if $A$ is a self-adjoint operator on the hilbert space $\mathcal{H}$, then $\mathcal{H}$ can be defined by orthonormal basis made by eigenvectors of $A$. From the above mentioned properties, the main benefit extracted from self-adjoint operator is that it exists a set...
of functions $\phi$ such that

$$D\phi = \lambda \phi$$

Where $\lambda$ is a diagonal matrix containing real eigenvalues and $\phi$ is a matrix of orthonormal eigenfunctions. These properties allow us to treat spin transformation problem (in chapter 3) as an eigenvalue problem, which can be solved efficiently.

Another term we haven’t discuss about is ”elliptic”. A self-adjoint, elliptic operator is relevant to boundary conditions in conformal deformation. However, because we are considering simply connected shape which has no boundary. Therefore, we won’t discuss it in detail. If interested, please refer to [BP13].

Besides, Dirac operator has another attractive feature which it can represent all the other basic differential operators concisely, such as curl, grad and so on [Cr13]. This property allows it to form a framework for shape analysis tasks called quaternionic shape analysis [FC13].

3 Spin Transformation

In this section, we introduce the main subject of this article - spin transformation. It is all about conformal equivalence of immersed surfaces. Firstly, we discuss it in smooth settings. Then we will dig into discretized spin transformation and relevant algorithms.

3.1 Smooth Spin Transformation

Source: Crane, K. Conformal geometry processing
Consider two conformal immersions \( f \) and \( \tilde{f} \) of a surface \( M \). If their exist a quaternions \( \lambda \) such that

\[
d\tilde{f} = \tilde{\lambda}df \lambda
\]  

(4)

We call \( \tilde{f} \) is a spin transformed of \( f \) and \( f \) and \( \tilde{f} \) are spin equivalent. Recalling Eq.2, this formula represents transformation which only has rotation and scaling. A transformation with only rotation and scaling is exactly the definition of conformal transformation. Therefore, we can say that spin transformation implies conformal transformation. To prove it mathematically, we can write

\[
\langle d\tilde{f}, \star d\tilde{f} \rangle = \langle \tilde{\lambda}df \lambda, \tilde{\lambda} \star df \lambda \rangle
\]

\[
= \text{Re}(\tilde{\lambda}df \lambda \bar{\lambda} \star df \lambda)
\]

\[
= |\lambda|^4 \langle df, \star df \rangle = 0
\]

following similar procedure, we can get

\[
\langle d\tilde{f}, df \rangle = |\lambda|^4 \langle df, df \rangle
\]

\( \star \) is hodge star in exterior algebra, which can be viewed as 90-degree rotation of the original vector for real 1-forms on surfaces. \( \langle \cdot, \cdot \rangle \) represents inner product. The formulas implies conformal equivalence because \( |\lambda|^4 \) means scaling. \( \langle d\tilde{f}, \star df \rangle = \text{scale} \) \( \langle df, \star df \rangle \) means that right angle in \( \tilde{f} \) remains right angle in \( f \). Therefore, spin equivalence implies conformal equivalence.

But not every pair of conformal immersions, \( f \) and \( \tilde{f} \) can be spin equivalent. In [KPP98], it states that

(1) When surface \( M \) is simply connected, any two conformal immersions \( f, \tilde{f} \) are spin equivalent.

(2) If \( M \) is not simply connected, \( f, \tilde{f} \) are spin equivalent if and only if they are regularly homotopic.

Regularly homotopy [Pin85] is a subset of homotopy. Given two maps \( f, g \), both maps \( \mathcal{X} \) to \( \mathcal{Y} \). If we said \( f \) and \( g \) is homotopic, it means there exists a continuous map, \( h \in C(\mathcal{X}, \mathcal{Y}) \), which \( h(x, 0) = f(x) \) and \( h(x, 1) = g(x) \). In other words, homotopy is a 1-parameter continuous transformation, which \( h(x, i) \) where \( 0 < i < 1 \) covers the colored area in \( \mathcal{Y} \) shown below.

Source: Crane, K. Discrete Differential Geometry: An Applied Introduction
Given the same condition, if we said \( f \) and \( g \) is regularly homotopic, there exists an immersion map, \( h \in Imm(\mathcal{X}, \mathcal{Y}) \), which \( h(x, 0) = f(x) \) and \( h(x, 1) = g(x) \). Another relevant concept is isotopy, which exists a embedding map, \( h \in Emb(\mathcal{X}, \mathcal{Y}) \), satisfying \( h(x, 0) = f(x) \) and \( h(x, 1) = g(x) \). The relationship between these concepts are \( Emb(\mathcal{X}, \mathcal{Y}) \subseteq Imm(\mathcal{X}, \mathcal{Y}) \subseteq C(\mathcal{X}, \mathcal{Y}) \). In other words, isotopy \( \subseteq \) regularly homotopy \( \subseteq \) homotopy. In terms of our application, we only consider shapes that are simply connected. So any pair of conformally transformed shapes in the following discussion satisfy spin equivalence.

Besides, not every \( \lambda \) is valid for spin transformation either. The necessary condition, integrability condition, on \( \lambda \) according to [LP13] is \( d(df) = 0 \) and it implies

\[
0 = d(df) = d(\bar{\lambda} df \lambda) = d\bar{\lambda} \wedge df \lambda - \bar{\lambda} df \wedge d\lambda = \bar{\lambda} df \wedge d\lambda - \bar{\lambda} df \wedge d\lambda = -2Im(\bar{\lambda} df \wedge d\lambda)
\]

Hence, we can assume \( \bar{\lambda} df d\lambda \) equals to a real value function \( \rho \) with some scaling terms

\[
\bar{\lambda} df \wedge d\lambda = -\rho |\lambda|^2 |df|^2 \\
df \wedge d\lambda = -\rho |df|^2 \\
\frac{-df \wedge d\lambda}{|df|^2} = \rho \lambda
\]

According to Eq.3, we can notice that

\[
D\lambda := \frac{-df \wedge d\lambda}{|df|^2} = \rho \lambda \\
\rightarrow (D - \rho)\lambda = 0
\]

Therefore if \( \lambda \) satisfies \( (D - \rho)\lambda = 0 \), \( \lambda \) fulfills integrability condition and it is valid for spin transformation. An geometric intuition is that if \( \lambda \) doesn’t satisfy the condition, edges of a triangle in the original shape will not connect to each other after spin transformation.

### 3.2 Mean Curvature Half-density

Another question to explore is what is \( \rho \) in eq.5? So far, we know \( H \) mentioned in chapter 2.1 is mean curvature. And \( H|df| \) is called mean curvature half-density because people usually call \( |df|^2 \) density and \( |df| \) half-density. The intuition behind \( H|df| \) is that \( |df| \) is a scale factor corresponding to the conformal immersion \( f \), so we can treat \( H|df| \) as a scale invariant version of mean curvature. The benefit can be felt by a simple example. Assume we have two spheres with different radius, if
we want to use some intrinsic properties to describe the spheres, using mean curvature is not a good idea because smaller sphere has higher mean curvature. However, if we use mean curvature half-density, we can find that the values of the two spheres are identical which is the results we want.

So, back to the original question - what is \( \rho \)? \( \rho \) is curvature potential and \( \rho|df| \) means the change of mean curvature half-density (proved in [LP13]). The mathematical formula is

\[
\tilde{H}|d\tilde{f}| = H|df| + \rho|df| \tag{6}
\]

Therefore, \( \rho \) can be viewed as a property which controls the change of mean curvature half-density.

### 3.3 Discrete Spin Transformation

So far, we have all tools for computing spin transformation in smooth settings. Here, we will describe spin transformation process and how to discretize them to make it able to run on computers. Our task is to spin transform a sphere into desired shape given \( \rho \) [CPS11]. This algorithm would be used for shape reconstruction after having our new spin-based shape descriptor. The overall process of spin transformation is:

1. Given initial sphere mesh \( f \) and curvature potential \( \rho \)
2. Compute valid \( \lambda \) for spin transformation by solving smallest eigenvalue problem \((D - \rho)\lambda = \gamma \lambda\)
3. Use spin equivalence \( d\tilde{f} = \lambda df \) to get new edge vectors which \( \tilde{e} = d\tilde{f} \)
4. Solve \( \Delta \tilde{f} = \nabla \cdot \tilde{e} \) to get new vertex positions

\( f \) is the triangular mesh of our initial sphere and \( \rho \) is curvature potential which is a scalar function defined on each face implying the change of mean curvature half density. The first object to discretize is Dirac operator.
**Discrete Dirac Operator**

According to Eq.3, Dirac operator is

\[
D\lambda := -\frac{df \wedge d\lambda}{|df|^2}
\]

\[
= -\frac{ddf \wedge \lambda + df \wedge d\lambda}{|df|^2}
\]

\[
= -\frac{d(df \lambda)}{|df|^2}
\]

Applying stokes’ theorem

\[
D\lambda = \frac{1}{A_i} \int_{t_i} D\lambda |df|^2
\]

\[
= -\frac{1}{A_i} \int_{\partial t_i} df \lambda
\]

\[
= -\frac{1}{A_i} \sum_{e_{ij} \in \partial t_i} \int_{e_{ij}} df \lambda
\]

\[
= -\frac{1}{A_i} \sum_{e_{ij} \in \partial t_i} (f_j - f_i) \frac{\lambda_i + \lambda_j}{2}
\]

\[
= -\frac{1}{A_i} [(e_k + e_j) \frac{\lambda_i}{2} + (e_i + e_k) \frac{\lambda_j}{2} + (e_i + e_j) \frac{\lambda_i}{2}]
\]

\[
= -\frac{1}{2A_i} (e_i \lambda_i + e_j \lambda_j + e_k \lambda_k)
\]

Therefore, we can write Dirac operator as a matrix which:

\[
D_{ij} = -\frac{1}{2A_i} e_j
\]

**Discrete Curvature Potential**

To discretize curvature potential, we derive it similar to what we did for Dirac operator, which is integrating \(\rho \lambda\) over a triangle.

\[
\frac{1}{A_i} \int_{t_i} \rho \lambda |df|^2 = \rho_i \left( \frac{1}{3} \sum_{f_j \in t_i} \lambda_j \right) \quad \text{(if } \rho \text{ is defined per face)}
\]

\[
= \frac{1}{3} \sum_{v_j \in t_i} \lambda_j \rho_j \quad \text{(if } \rho \text{ is defined per vertex)}
\]
**Quaternionic Matrix**

The last thing we need to make it in matrix format is quaternions $\lambda$. We can construct $D$ and $\rho$ in matrix format already. We need a proper matrix for us to perform quaternions operations in typical vector operations. According to [Cr13], $q = a + bi + cj + dk$ can be written as:

$$
\begin{bmatrix}
  a & -b & -c & -d \\
  b & a & -d & c \\
  c & d & a & -b \\
  d & -c & b & a
\end{bmatrix}
$$

So far, we already know how to discretize Dirac operator, curvature potential, and quaternions. These allow us to build matrix according to those formulas and solve a simple eigenvalue problem to get valid $\lambda$ for spin transformation. The next step is to solve $d\bar{f} = \lambda df \lambda$ to get new edge vectors. According to [CPS11], it can be discretized into

$$
e_{ij} = \frac{1}{3}\lambda_i e_{ij}\lambda_i + \frac{1}{6}\lambda_i e_{ij}\lambda_j + \frac{1}{6}\lambda_j e_{ij}\lambda_i + \frac{1}{3}\lambda_j e_{ij}\lambda_j
$$

Then, in order to get new vertex positions, we start with this relationship

$$
d\bar{f} = \bar{e}
\rightarrow (\star d\star) d\bar{f} = (\star d\star) \bar{e}
\rightarrow \Delta \bar{f} = \nabla \cdot \bar{e}
$$

It makes the problem become standard Poisson problem, which we can use famous cotan formula to discretize laplace operator to find final vertex positions in discrete setting

$$(\Delta p)_i = -\frac{1}{2} \sum_j (\cot \alpha_j + \cot \beta_j)(p_j - p_i)$$

**Shape Descriptors**

Shape descriptors map shapes from the space of 3D objects to finite-dimensional vector space with the aim of achieving similar representation between shapes. It also aims to preserve as much "target" data as possible and keep the resulting vector as low-dimensional as possible. By comparing...
their feature representations, we could preprocess raw input shape data and quantify the difference between shapes. Based on these reasons, a good shape descriptor tends to be (1) discriminative, (2) robust with respect to noise, (3) easy to compare, (4) transformation invariant.

Shape descriptor is important because it defines metric space for shape collections or within a shape. Based on the metric space, it allows us to perform segmentation, classification, matching, semantic meaning extraction and lots of interesting shape analysis tasks.

Shape descriptor is closely relevant to differential geometry because most of the existing ones try to "describe" shapes in terms of geometric properties, including geodesic distance, curvature, and so on. Therefore, differential geometry is the foundation of designing shape descriptors.

4.1 Shape Descriptor Overview

In the past three decades, there have been tens of shape descriptors designed for different objectives and capturing different shape features. Descriptors are usually categorized into local and global descriptors. Local shape descriptors measure local similarity within a shape or between a pair of shapes. On the other hand, global descriptors map whole 3D shapes to vector spaces which allow us to compare encoded information among shape collections. Although it is a reasonable categorization, I prefer to categorize them according to shape properties they build on in order to understand geometric ideas behind those descriptors. Here is a list of some examples, no exhaustive, to give a feeling about how to utilize geometric properties to design shape descriptors.

Euclidean Distance
An representative example of using Euclidean Distance is called Shape Distribution [OFCD02] which calculate all pairwise distance between vertices and form a distance distribution as shape descriptor. Shape distribution is suitable for classification tasks, robust to transformation, and very easy to compute.

Geodesic Distance
Geodesic Fans [GGGZ05][OS11] grows geodesic circles around vertices and samples surface properties from the circle in order to compare with other geodesic fans. This is a typical local shape descriptor because each geodesic fan represents local features. Therefore, it is suitable to determine similarity within a shape.

Surface Normal
The shape descriptor utilizing surface normal is called Point Descriptor [YF02] which took a line through the point on the ring and store the line length and angle with respect to the normal as ingredient for building shape descriptor. This descriptor is fast to compute so that it allows real-time recognition, which is beneficial to computer vision tasks.

Curvature
A shape descriptor using curvature information is Smoothing [LG05]. The design is based on the fact that high-curvature region tend to move quicker in smoothing process. The intuition behind
designing this descriptor is similar to this project, utilizing the change in curvature to design descriptor. However, many other descriptors directly encode curvature value without utilizing its properties.

**SHAPE**

*Spherical Harmonics* [FMK*03] uses spherical harmonic functions to map shape geometry into feature space and treats the mapped vector as shape descriptor. This is robust to noise, orientation, and its classification power outperforms most of the other descriptors. The other one is called *Lightfield* [CTSO03] which uses 2D projections of a shape from different angles to design descriptor. This descriptor is also powerful for classification tasks. Besides, it becomes more popular recently because the 2D projections are applicable to image classification using deep learning models.

**LAPLACE-BELTRAMI**

Using laplace operator may be the most popular approach recently. Eigenfunctions derived from laplace operator are intrinsic to each shape. If we project the shape onto these eigenfunctions, we can get a descriptor in terms of a list of numbers. Besides, we can adjust the size of the numbers by adjusting the number of eigenfunctions to use. As a results, we can manipulate those numbers to different shape analysis tasks. For example, *Functional Map* [OBS*12] is the current state-of-the-art method for finding correspondence between shapes.

**OTHERS**

Besides above mentioned popular approaches, there are some interesting directions for shape descriptor design. Firstly, we can see that most of the popular geometric properties used for shape descriptor design are differential properties, such as curvature. Therefore, an alternative approach is to use integrative features such as surface area [PWHY09] and this approach is proved to be more stable than using differential properties. Another idea is to apply persistent homology to capture topological information [COO15]. Another approach is to utilize shape collections and machine learning techniques to learn latent representation of shapes. An example is using autoencoder to learn features for procedural modeling [YAMK15].

**4.2 Spin-based Shape Descriptor**

In [Bo67], Bonnet proposed that mean curvature and metric should suffice to determine the surface. This idea builds up spin transformation which utilizes curvature potential to reconstruct shapes. However, to the best of my knowledge, there is no shape descriptor designed using curvature potential which is a property built upon mean curvature. Therefore, it is interesting to ask whether curvature potential is a good property for shape descriptor design because it may be the simplest way to define a shape theoretically.

This project aims to encode $\rho$ from an arbitrary simply connected shape given its conformally transformed sphere and use $\rho$ to perform shape reconstruction and classification.

**Modified Mean Curvature Flow**
Modified mean curvature flow is an algorithm which conformally transforms a shape to a sphere [KSB12]. We used this algorithm to derive conformally transformed sphere for each input shape. Basically, modified mean curvature flow uses original mean curvature flow framework, \((I - M^{-1}h\Delta_0)f_h = f_0\), without updating the laplacian matrix. Here is an implementation result.

**Encode Curvature Potential**

We have two methods to encode curvature potential. The first method uses quadratic surface to approximate a surface patch. Then computes the eigenvalues of its hessian matrix. Because the eigenvalues are principle curvatures, we can average them to get mean curvature. After finding approximated mean curvature, we can simply apply the definition \(\tilde{H}|d\tilde{f}| = H|df| + \rho|df|\) to get curvature potential, in which we treat \(|df|\) as the square root of dual vertex area. Dual vertex area is the dark blue region in the figure below.

\[
(d*du)_i
\]

Source: Crane, K. *Discrete Differential Geometry: An Applied Introduction*

The second method uses Dirac operator to encode curvature potential, we first recall Eq.6

\[
\tilde{H}|d\tilde{f}| = H|df| + \rho|df|
\]

\[
\rightarrow \rho = \frac{\tilde{H}|d\tilde{f}| - H|df|}{|df|} = \frac{\tilde{H}\sqrt{\tilde{A}} - H\sqrt{A}}{\sqrt{A}}
\]
Because we are working on triangular mesh, we can again approximate scale factor $|df|$ to be square root of dual vertex area $\sqrt{A}$. According to above formula, we can calculate $\rho$ once we have mean curvature $H$. By applying Dirac operator on surface normal, we can get

$$DN : = -\frac{df \wedge dN}{|df|^2}$$

$$= -\frac{df \wedge dN(X_1, X_2)}{|df|^2} \quad \text{(Apply on principal directions)}$$

$$= -\frac{df(X_1)dN(X_2) - df(X_2)dN(X_1)}{|df|^2}$$

$$= -\frac{\kappa_2 df(X_1)df(X_2) - \kappa_1 df(X_2)df(X_1)}{|df|^2} \quad \text{(Apply Eq.1)}$$

$$= \frac{(\kappa_1 + \kappa_2) df(X_1)df(X_2)}{|df|^2}$$

$$= -\frac{(2H) df(X_1) \times df(X_2)}{|df|^2}$$

$$= -\frac{(2H) N|df(X_1, X_2)|^2}{|df|^2}$$

$$= -2HN$$

After knowing $DN = -2HN$, we can derive curvature potential using

$$\langle -\frac{DN}{2}, N \rangle = H$$

$$\therefore \rho = \frac{\sqrt{A} \langle -\frac{DN}{2}, \tilde{N} \rangle - \sqrt{A} \langle -\frac{DN}{2}, N \rangle}{\sqrt{A}}$$

Here is an example of encoding curvature potential onto its conformally transformed sphere. The color on the sphere represents curvature potential values.
**Reconstruction using Spin Transformation**

After getting sphere geometry and curvature potential, we can perform spin transformation to reconstruct shape as we discussed in Chapter 3. Here are a successful example of surface reconstruction and some failed ones.

According to the experiments, we found that reconstruction using spin-based descriptor may be limited to simply connected shape with simple geometry. It may not be able to reconstruct shape with sharp corners and salient structures, such as the head of the cow. Two potential solutions are (1) performing multiple primitive fittings and segmentation first and then spin transforming primitives to each part, (2) modifying spin transformation discretization scheme in order to reconstruct salient shape parts.
Classification using Persistent Barcode

Persistent Barcode is widely used to capture topological information of shapes [ZC05]. We apply the idea of growing-ball and track the number of connected components with respect to different radius, as figure shown below.

However, we use the idea differently. We grow the ball from the center of conformally transformed sphere and track connected components of curvature potential, 2D intuition is showed in the figure below. Dot line represents the sphere, red circle represent the growing-ball, black curve represents curvature potential value, and the blue curve is a connected component.
With this algorithm, we can get persistent barcode for each shape, such as the figure below. Beyond this project, we hope to apply some methods to quantify the difference between the barcodes with the aim of achieving good shape classification results.

5 Conclusion and Outlook

Spin transformation describes conformal equivalence between immersion surfaces and is able to compute elegantly using quaternions and Dirac operator. After discretization, spin transformation can be treated as eigenvalue problem and poisson equation, which can be solved efficiently.

Motivated by Bonnet’s surface theory, we designed a shape descriptor based on mean curvature. We used modified persistent barcode to perform shape classification using this descriptor. Besides, we used spin transformation to reconstruct geometry. However, the reconstruction power of this descriptor seemed to be limited to genus-0 shape with simple geometry. In order to make it a powerful descriptor for reconstruction, different discretization schemes are needed. Therefore, digging further into which properties are / are not preserved by discretization would be helpful for designing discretization schemes.

If it succeeded, its discriminative and reconstructive power implies it is a suitable representation of shapes. Besides, because the representation is a scalar curvature potential field, transforming it into a matrix and do other applications is possible. For example, we can feed them to machine learning models to learn relationship between shapes.

6 Reference


