

Cell Shape Analysis Using Spherical Harmonics

Casper van Bavel

Thesis submitted for the degree of
Master of Science in Bioinformatics

Thesis supervisor:
Prof. Rob Jelier

Assessor:
Prof. Stein Aerts

Mentor:
Wim Thiels

© Copyright KU Leuven

Without written permission of the thesis supervisor and the author it is forbidden to reproduce or adapt in any form or by any means any part of this publication. Requests for obtaining the right to reproduce or utilize parts of this publication should be addressed to Faculteit Ingenieurswetenschappen, Kasteelpark Arenberg 1 bus 2200, B-3001 Heverlee, +32-16-321350.

A written permission of the thesis supervisor is also required to use the methods, products, schematics and programmes described in this work for industrial or commercial use, and for submitting this publication in scientific contests.

Preface

I would first like to thank my promotores, Prof. Rob Jelier and Prof. Bart Smeets. Without them, this work would not have been possible. I would also like to thank Wim Thiels, who supervised the writing of this thesis. He and Prof. Rob Jelier gave me many helpful comments and suggestions. Their feedback and involvement were invaluable. In addition, I am grateful to my parents for their continued support and encouragement throughout my study. Finally, I would like to thank my friends, especially Michael Bartlett, Lewis Saunders, Micah Johnston, Alex Iadicicco and Vincent Chang, for their support, helpful comments and interesting discussions.

Casper van Bavel

Contents

Preface	i
Abstract	iv
List of Figures	v
List of Abbreviations and Symbols	vii
I Literature study	1
1 Introduction	2
1.1 Shape descriptors	2
1.2 Spherical harmonics decomposition	3
1.3 Spherical harmonics	4
1.4 Spherical parameterization	6
1.5 Invariant descriptors & power spectrum	6
1.6 Existing pipelines	7
2 Discrete surfaces	9
2.1 Differential geometry of surfaces	9
2.2 Curvature	10
2.3 Laplacian	11
2.4 Discretization	12
2.5 Mass matrix	13
3 Spherical parameterization	14
3.1 Introduction	14
3.2 Optimization algorithms	14
3.3 Flow-based algorithms	15
3.4 Conformalized mean curvature flow	15
4 Spherical harmonics decomposition	18
4.1 Least squares solution	18
4.2 Iterative residual fitting	19
5 <i>C. elegans</i> embryology	21
5.1 Description	21
5.2 Embryology	21
5.3 Shape and cell mechanics	22

5.4	7-cell stage	22
5.5	Perturbations	22
6	Motivation & objectives	24
6.1	Motivation	24
6.2	Objectives	24
II	Implementation	26
7	Implementation	27
7.1	Triangle mesh data structures	28
7.2	Spherical parameterization	28
7.3	Möbius transformations	29
7.4	Canonical rotation	31
7.5	Spherical harmonics decomposition	32
7.6	Algorithm Speed	34
8	Analysis of spherical harmonics	35
8.1	First order ellipsoid	35
8.2	Smoothing and filtering	35
III	Application	38
9	Results	39
9.1	<i>dsh-2/mig-5</i> RNAi knockdowns	39
9.2	<i>fmi-1</i> mutants	42
10	Conclusion	44
10.1	Contributions	44
10.2	Limitations	44
10.3	Further work	44
Bibliography		46
Popularized summary		50
Appendices		51
Finite element method discretization		52
Basis functions		52
Weak formulation		53
Inner product		54

Abstract

Understanding how cells self-organize into multicellular systems is a fundamental challenge in biology. The complexity of this process arises from the dynamic interplay between molecular components and physical forces.

The shape of a cell reflects, among other things, actomyosin activity, adhesion properties and other interactions with its environment. Cell shape is typically tightly linked to the differentiation of the cells and can further be used to infer important cellular properties such as force generation, cortical tension and adhesion properties.

Current shape descriptors only look at simple geometric features such as volume and sphericity. In this work, a new framework is presented to analyze the shape of cells in developing *C. elegans* embryos. As a central tool, the spherical harmonics decomposition is used to quantify shape in a complete and generic way.

While the spherical harmonics decomposition has already been used in the description of shape, it has not been applied to *C. elegans* before. A new package is developed to calculate the spherical harmonics decomposition, which includes several innovations related to the calculation and interpretation of this decomposition. Further, this new analysis pipeline is both fast and modular.

Next, these new tools are applied to study the phenotype of two *C. elegans* mutants, *dsh-2/mig-5* RNAi knockouts and *fmi-1* mutants in the 7-cell stage. For the *dsh-2/mig-5* knockouts, a difference was found in the E cells compared to the wild type, by analyzing their power spectrum. For the *fmi-1* mutants, a difference was found in the development of the lamellipodium with a novel descriptor that detects protrusions.

The methods presented in this work allow for complete and accurate cell shape description, which can empower new observations and experiments in *C. elegans* as well as other organisms.

List of Figures

1.1	Progression of Alzheimer disease. Each image is taken 1 year apart and show to progression of hippocampal (H) atrophy. Figure from Johnson et al. [17]	3
1.2	Spherical angular coordinates (θ, φ) of a point P on the unit sphere.	5
1.3	Real spherical harmonics of degree 0 to 3.	5
1.4	<i>Left:</i> stellar shape. <i>Right:</i> non-stellar shape with overlapping area shaded in grey.	6
2.1	Local map of a surface around a point p	10
2.2	Tangent and normal vectors.	10
2.3	Normal curvature of surface at a point p	11
2.4	Map for a discrete surface.	12
3.1	Schematic representation of surface smoothing with geometric flow.	15
3.2	Evolution of the unmodified and conformalized mean curvature flows on Spot the cow. Timestep $h = 0.1$. For the unmodified flow the mesh collapses into a line.	17
5.1	Diagram of early cell stages. The dashed line shows the chosen timepoint. Figure adapted from Wormbook [21].	23
7.1	Conformal scale factor λ plotted on a wild-type ABpl cell and its spherical map. Red means $\lambda > 1$, blue means $\lambda < 1$	30
8.1	Heat kernel (blue) and its Laplacian (orange) plotted as a function of θ	37
8.2	Magnitude of Laplacian of Gaussian, as in eq. (8.8), plotted on the mesh of a wild-type ABpl cell.	37
9.1	Overview of all the E cells in the dataset. Wild type on the left and <i>dsh-2/mig-5</i> RNAi knockdowns on the right. The contribution of spherical harmonics of degree larger than 1 is highlighted in red.	39
9.2	Averaged shapes for the E cells of wild type and <i>dsh-2/mig-5</i> RNAi knockdowns. The contribution of spherical harmonics of degree larger than 1 is highlighted in red.	40
9.3	Average normalized power spectrum of the E cells for mutants and wild types. The y-axis is on a logarithmic scale with base 10.	40

LIST OF FIGURES

9.4	Relative contribution of the first degree in the power spectrum for cells of <i>dsh-2/mig-5</i> RNAi knockdowns versus wild type.	41
9.5	Overview of all the ABpl cells in the dataset. Wildtype on the left and <i>fmi-1</i> mutants on the right. The LoG magnitude is shown in red.	42
9.6	Averaged shapes for ABpl cells of wild type and <i>dsh-2/mig-5</i> RNAi knockdowns. The LoG magnitude is shown in red.	42
9.7	Integrated Laplacian of Gaussian (LoG) for cells of <i>fmi-1</i> mutants versus wild-type specimens. For the ABpl cells the result is significant ($p = 0.028$), for the ABal cells there is no significant difference ($p = 0.33$).	43
1	Hat function ϕ_i with associated vertex v_i	52

List of Abbreviations and Symbols

Abbreviations

SH	Spherical harmonics
FEM	Finite element method
MCF	Mean curvature flow
CMCF	Conformalized mean curvature flow
FOE	First order ellipsoid
LoG	Laplacian of Gaussian

Symbols

Y_ℓ^m	Complex spherical harmonics of degree ℓ and order m
\mathcal{Y}_ℓ^m	Real spherical harmonics of degree ℓ and order m
P_ℓ^m	The associated Legendre polynomials
θ	Colatitude
φ	Longitude
S^2	The unit 2-sphere
H	Mean curvature
Δ	Laplacian operator
M	Mass matrix
L	Stiffness matrix
$S(\ell)$	Power spectrum of degree ℓ
$\hat{\phi}_i$	Hat basis function

Part I

Literature study

Chapter 1

Introduction

In our daily lives we constantly analyze and compare shapes. We use shape to judge the nature of objects, and it allows us to make detailed assessments about that object, for example when we judge the age of an apple.

In medicine, imaging of anatomical structures has become a very important tool. For example, to determine brain pathologies by magnetic resonance imaging (MRI). By comparing images from a set of normal brains to that of a patient, atrophy of brain regions can be identified to aid in the diagnosis of diseases such as Alzheimer disease [17] (fig. 1.1). On the cellular level, pathologists study the morphology of cells from a biopsy to determine if cells are likely to be malignant [36]. Even though we readily observe differences in shape by eye, it remains challenging to handle information about shapes in an exact and quantifiable way.

When studying embryonic development and multicellular organization, cell shape is also of critical importance. In this thesis the focus is on the development of the roundworm *Caenorhabditis elegans*, an important model organism. During its fast and invariant embryogenesis, cells rapidly change shape, interact with each other, and migrate. These cell shapes can be accurately measured by fluorescently tagging the cell membrane and using high resolution fluorescent microscopy. In the *C. elegans* literature, cell shapes are frequently discussed, typically in the context of a change in cell shape in response to a genetic or chemical perturbation. However, there is no formal framework to describe and compare cell shapes in a generic fashion. In this thesis, a pipeline is developed to describe and compare cell shapes in a quantitative way and at a resolution of choice.

1.1 Shape descriptors

Shape descriptors are quantities that depend on shape while being invariant to its underlying representation. Cells are commonly represented using 3D triangle meshes, but these meshes can have different densities and vertex positions while describing the same shape. Such descriptors should also be invariant to translations and rotations, since those do not modify the actual shape. The descriptors should also not depend on the way the measurement is taken.

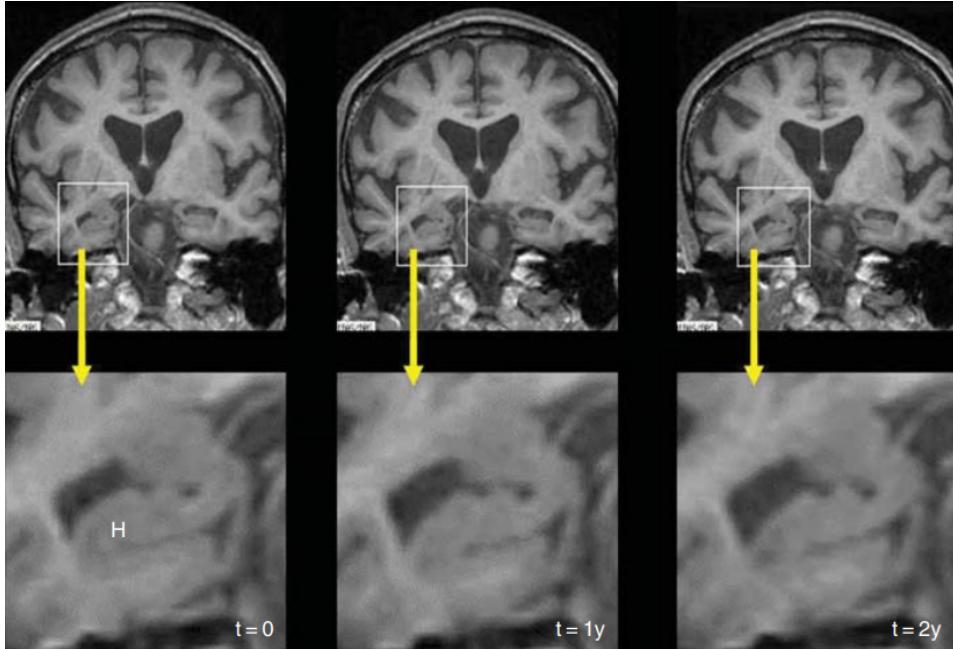


FIGURE 1.1: Progression of Alzheimer disease. Each image is taken 1 year apart and show to progression of hippocampal (H) atrophy. Figure from Johnson et al. [17]

A first approach, as in Andrews et al. [3] is to use geometrical parameters such as surface-volume ratio, sphericity, flatness, etc. This approach is very pragmatic: when there is a clear phenotype, usually some geometrical parameter can be found that corresponds to this phenotype. However, these descriptors can not fully and generically describe a shape.

A second approach is landmark-based morphometrics, often used in evolutionary biology [35]. Broadly, with this technique points are (manually) placed on anatomical landmarks. The distances and angles between points can then be compared between samples. It is a very powerful approach to talk quantitatively about the shape of, for example, bone structures. The problem is that cell shapes are fluid and highly variable over time, so it is rare to have such recognizable landmarks.

1.2 Spherical harmonics decomposition

An attractive choice for shape descriptors are the spherical harmonics (SH). They can be used as a compact representation of a shape and have some very favorable properties [12]. The main idea of spherical harmonics decomposition is that functions on a sphere can be described as a linear combination of basis functions (spherical harmonics). This composition can be found if the function f is square-integrable:

$$\int_{S^2} |f(x)|^2 dS < \infty \quad (1.1)$$

1. INTRODUCTION

The real spherical harmonics form an orthonormal basis of the space of square-integrable functions $L^2_{\mathbb{R}}(S^2)$. This means that every square-integrable function $f \in L^2_{\mathbb{R}}(S^2)$ can be decomposed as a linear combination of spherical harmonics. This is known as spherical harmonics decomposition. Usually, the spherical harmonics are given as complex functions, but they can be transformed into a real-valued form. For a real function, using the real SH has the added benefit that the coefficients are also guaranteed to be real. This decomposition can be seen as the spherical analogue to the Fourier expansion of one dimensional periodic functions [27].

Given a square-integrable function $f : S^2 \rightarrow \mathbb{R}$, the SH decomposition is:

$$f(\theta, \varphi) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} c_{\ell}^m Y_{\ell}^m(\theta, \varphi) \quad (1.2)$$

where c_{ℓ}^m are real coefficients.

While this is an infinite sum, in practice it has to be truncated to a maximum degree ℓ_{max} . This representation is hierarchical, with low degrees capturing coarse information and higher degrees capturing finer details. The more harmonics are used, the more details are captured. For a maximum degree ℓ_{max} there are $(\ell_{max} + 1)^2$ coefficients. If ℓ_{max} is large enough, the function can be recovered from the coefficients alone.

1.3 Spherical harmonics

The spherical harmonics are a set of functions defined on the surface of a sphere. The complex spherical harmonics $Y_{\ell}^m : S^2 \rightarrow \mathbb{C}$ of degree ℓ and order m are defined as:

$$Y_{\ell}^m(\theta, \varphi) = N_{\ell m} P_{\ell}^m(\cos(\theta)) e^{im\varphi} \quad (1.3)$$

Where S^2 is the unit sphere, (θ, φ) are spherical angular coordinates, with θ the colatitude ($0 \leq \theta \leq \pi$) and φ the longitude ($0 \leq \varphi < 2\pi$) (fig. 1.2). $N_{\ell m}$ is the normalization factor:

$$N_{\ell m} = \sqrt{\frac{2\ell + 1}{4\pi} \frac{(\ell - m)!}{(\ell + m)!}}$$

P_{ℓ}^m are the associated Legendre polynomials, which can be defined in terms of the ordinary Legendre polynomials $P_{\ell}(x)$:

$$P_{\ell}^m(x) = (-1)^m (1 - x^2)^{m/2} \frac{d^m}{dx^m} (P_{\ell}(x))$$

The ordinary Legendre polynomials can be expressed as:

$$P_{\ell}(x) = \frac{1}{2^{\ell} \ell!} \frac{d^{\ell}}{dx^{\ell}} (x^2 - 1)^{\ell}$$

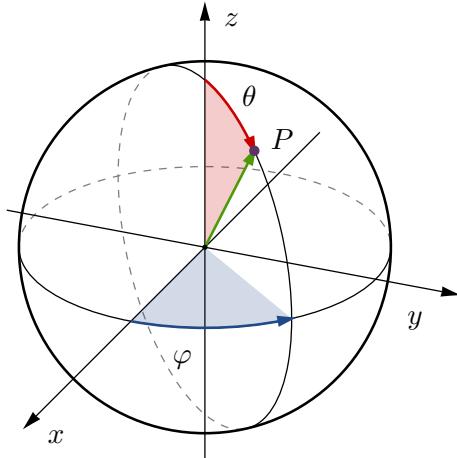


FIGURE 1.2: Spherical angular coordinates (θ, φ) of a point P on the unit sphere.

The real spherical harmonics $\mathcal{Y}_\ell^m : S^2 \rightarrow \mathbb{R}$ can then be derived:

$$\mathcal{Y}_\ell^m = \begin{cases} \sqrt{2} \operatorname{Im} [Y_\ell^{lm}] & \text{if } m < 0 \\ Y_\ell^0 & \text{if } m = 0 \\ \sqrt{2} \operatorname{Re} [Y_\ell^{lm}] & \text{if } m > 0 \end{cases} \quad (1.4)$$

For every degree ℓ there are $2\ell + 1$ such functions, with order $m = -\ell, \dots, +\ell$.

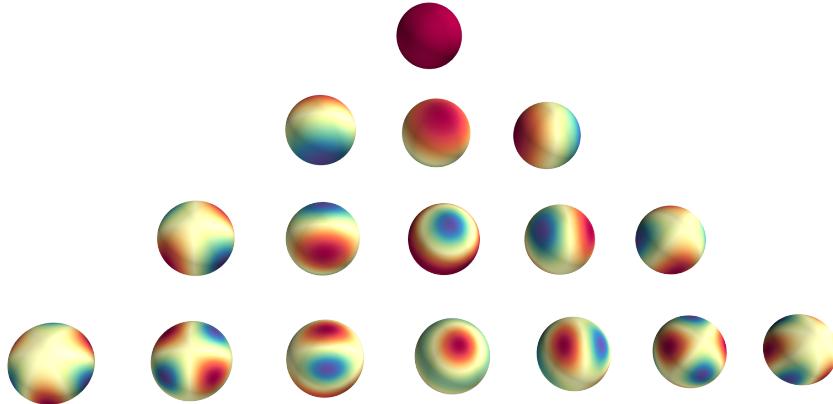


FIGURE 1.3: Real spherical harmonics of degree 0 to 3.

1.4 Spherical parameterization

Of course, to use this decomposition there needs to be a way to describe a shape using only functions on the sphere. One idea is to use the distance from the center. While this might seem like a good solution, it is only possible to represent the shape when it is *stellar*, meaning we can ‘see’ the whole surface from the center without obstructions [20] (see fig. 1.4). To be able to represent non-stellar objects, shapes

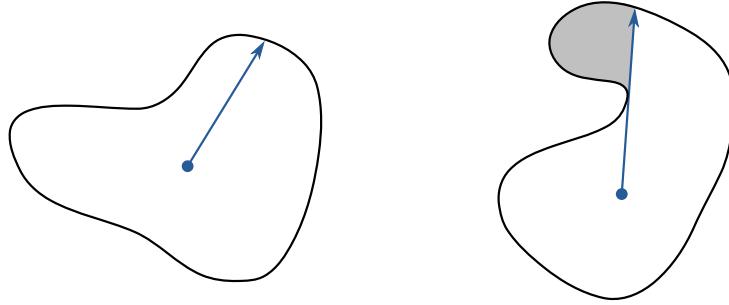


FIGURE 1.4: *Left:* stellar shape. *Right:* non-stellar shape with overlapping area shaded in grey.

can be described as 3 functions on the sphere, one for each coordinate. For a surface X and an invertible map $\omega : S^2 \rightarrow X$ from the unit sphere to the surface, we can describe X in spherical coordinates as:

$$X(\theta, \varphi) = \omega(\theta, \varphi) = \begin{pmatrix} x(\theta, \varphi) \\ y(\theta, \varphi) \\ z(\theta, \varphi) \end{pmatrix} \quad (1.5)$$

The SH decomposition then consists of the three separate decompositions for each coordinate function. The problem is then finding such a map. The procedure of finding the map of the shape on a sphere is called spherical parameterization.

Cell shapes can be described by closed surfaces of genus zero. For such surfaces there always exists a smooth, invertible map to the sphere, since they are topologically equivalent. Intuitively, the genus of a surface is the number of holes it has. The sphere has genus zero while a torus has genus one. Closed here means the surface has no boundary.

1.5 Invariant descriptors & power spectrum

A good set of shape descriptors should not change when we apply a rigid transformation (rotations or translation) on the whole shape. Translations are easy to take care of: before the analysis the shape is translated so that its center of mass is on

the origin. To normalize the scale, the whole shape can be scaled so that its surface area is the same for all the shapes in the dataset.

It is not immediately clear how to handle rotations. This problem can be solved using SH. We can decompose a function $f : S^2 \rightarrow \mathbb{R}$ into its frequency components by summing equation eq. (1.2) over each degree:

$$f(\theta, \varphi) = \sum_{\ell=0}^{\infty} f_{\ell}(\theta, \varphi) \quad (1.6)$$

with: $f_{\ell}(\theta, \varphi) = \sum_{m=-\ell}^{\ell} c_{\ell}^m Y_{\ell}^m(\theta, \varphi)$

and define the squared L^2 -norm as:

$$S_f(\ell) = \|f_{\ell}(\theta, \varphi)\|^2 = \sum_{m=-\ell}^{\ell} (c_{\ell}^m)^2 \quad (1.7)$$

This is known as the power spectrum. The key property is that applying an arbitrary rotation to the function preserves this norm [18]. In other words: a rotation ‘mixes’ the coefficients only within each order ℓ , while preserving the squared sum of the coefficients. This property follows from the fact that SH are the basis function for the irreducible representation of the rotation group in three dimensions.

This gives a set of shape descriptors that are invariant to rotations:

$$S_f = \left\{ \|f_0\|^2, \|f_1\|^2, \|f_2\|^2, \dots \right\} \quad (1.8)$$

By Parseval’s theorem, the total power of a function (the square integral) is equal to the sum of the spectrum:

$$\int_{S^2} |f(x)|^2 dS = \sum_{\ell=0}^{\infty} S_f(\ell) \quad (1.9)$$

We can imagine the total power being ‘distributed’ over the frequencies. Again these decompositions are infinite, but with truncation this gives $\ell_{max} + 1$ descriptors.

Another possibility is given by Brechbühler et al. [6] They rotate the shape into a canonical orientation. Using only the contribution of the SH of degree $\ell = 1$ results in an ellipsoid which can be rotated so that its shortest and longest axes are oriented consistently. This method has the benefit that all of the components can be used invariantly, instead of having to sum over them for each degree.

1.6 Existing pipelines

The spherical harmonics decomposition has been applied to cell shapes before by Agus et al. [2]. They use the Willmore flow method developed by Crane et al. [9] (as

1. INTRODUCTION

described in section 3.3) combined with the power spectrum. This power spectrum is used as the input to a support vector machine to classify nuclei of brain cells. Their pipeline is not publicly available.

A MATLAB [22] toolkit called SPHARM-MAT [1] is available. It uses methods described by Brechbühler et al. [6] and Shen et al. [29]. It is mostly focused on analyzing brain structures. While the toolkit itself is open-source, MATLAB is proprietary and non-free software.

Chapter 2

Discrete surfaces

We will represent the shape of a cell as a 2-dimensional regular surface in Euclidean space \mathbb{R}^3 . It is helpful to first think about the continuous setting before discretizing. In this chapter some necessary mathematical tools will be reviewed.

2.1 Differential geometry of surfaces

A regular surface \mathcal{M} is a subset of \mathbb{R}^3 for which there is local parameterization at every point of \mathcal{M} . A local parameterization around a point consists of an invertible map $f : V \rightarrow U$ where $V \subset \mathbb{R}^2$ and $U \subset \mathcal{M}$ is an open neighborhood around the point. (See fig. 2.1.) We can write this map as a function of the coordinates $(u, v) \in V$:

$$f(u, v) = \begin{bmatrix} x(u, v) \\ y(u, v) \\ z(u, v) \end{bmatrix} \quad (2.1)$$

And denote the partial derivatives as:

$$\mathbf{x}_u = \frac{\partial f}{\partial u}, \quad \mathbf{x}_v = \frac{\partial f}{\partial v} \quad (2.2)$$

For a regular surface we also require that \mathbf{x}_u and \mathbf{x}_v are linearly independent vectors in \mathbb{R}^3 .

Tangent and normal vectors

At a point $p \in \mathcal{M}$ the tangent vectors are any linear combination of \mathbf{x}_u and \mathbf{x}_v , evaluated at $f^{-1}(p)$. The unit normal vector at this point is:

$$\hat{\mathbf{n}} = \frac{\mathbf{x}_u \times \mathbf{x}_v}{\|\mathbf{x}_u \times \mathbf{x}_v\|} \quad (2.3)$$

Where $\|\cdot\|$ is the usual vector norm in \mathbb{R}^3 (fig. 2.2). Note that this construction of tangent and normal vectors does not depend on the particular local parameterization.

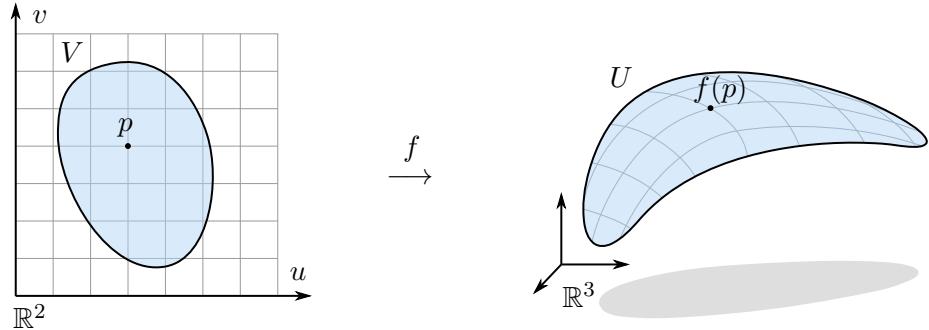
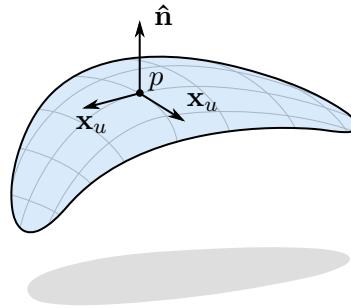

 FIGURE 2.1: Local map of a surface around a point p .


FIGURE 2.2: Tangent and normal vectors.

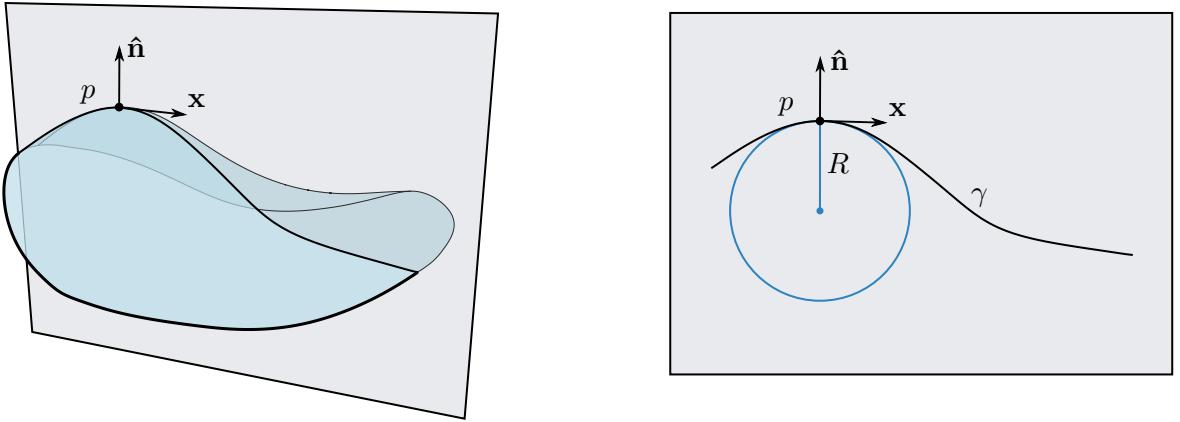
2.2 Curvature

Intuitively, the curvature of a surface is the amount by which a curve deviates from a straight line. For a circle of radius R the curvature κ is $1/R$. For every point of a general curve in the plane, we can find its osculating circle, the circle that best approximates the curve at that point. This definition is analogous to the tangent line. The radius of curvature R at the point is then the radius of the osculating circle. The curvature is the reciprocal of radius of curvature:

$$\kappa = \frac{1}{R} \quad (2.4)$$

Normal curvature

Let \mathbf{x} be any tangent vector and $\hat{\mathbf{n}}$ the normal vector at a point $p \in \mathcal{M}$. We can construct a plane going through p , containing \mathbf{x} and $\hat{\mathbf{n}}$. The intersection of this plane


 FIGURE 2.3: Normal curvature of surface at a point p .

and the surface is a plane curve γ . The normal curvature κ_n at p is the curvature of γ at p (fig. 2.3). We use eq. (2.4) again to find the curvature as the reciprocal of radius.

Principal and mean curvatures

The normal curvature at a point depends on the direction ϕ of the tangent vector we choose. Euler's theorem says that the curvature of a surface at a point is completely described by the minimum and maximum normal curvatures, together called the principal curvatures:

$$\begin{aligned}\kappa_1 &= \max_{\phi} \kappa_n(\phi) \\ \kappa_2 &= \min_{\phi} \kappa_n(\phi)\end{aligned}\tag{2.5}$$

The principal curvature directions are always orthogonal. The mean curvature H is the average of all the normal curvatures or equivalently, the average of the principal curvatures:

$$H = \frac{1}{2\pi} \int_0^{2\pi} \kappa_n(\theta) d\theta = \frac{1}{2}(\kappa_1 + \kappa_2)\tag{2.6}$$

2.3 Laplacian

The Laplace-Beltrami operator, denoted Δ , also known as simply the Laplacian, is a differential operator defined as the divergence of the gradient: $\Delta u = \nabla \cdot \nabla u$, where $u : \mathcal{M} \rightarrow \mathbb{R}$. It takes a scalar function on a surface to another scalar function. In a

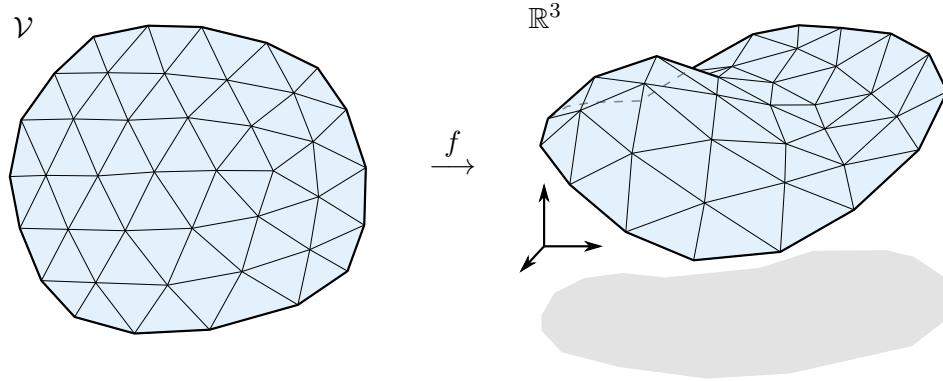


FIGURE 2.4: Map for a discrete surface.

Cartesian coordinate system it simplifies to the usual Laplace operator, given by the sum of second partial derivatives of the function.

The Laplacian is also related to the mean curvature. Denoting the coordinate function $\mathbf{p} = (x_1, x_2, x_3)$ which simply returns the coordinates of a point in \mathbb{R}^3 , and the unit normal $\hat{\mathbf{n}} = (n_1, n_2, n_3)$, we can write:

$$\Delta x_i = -2Hn_i \text{ for } i = 1, 2, 3 \quad (2.7)$$

Or more compactly:

$$\Delta \mathbf{p} = -2H\hat{\mathbf{n}} \in \mathbb{R}^3 \quad (2.8)$$

where we take the Laplacian independently over each coordinate. The resulting vector is the unit normal scaled by minus twice the mean curvature.

2.4 Discretization

To numerically solve geometrical problems we need to discretize them. The most widely used discretization of surfaces is the triangle mesh. An abstract triangle mesh \mathcal{K} is a set containing vertices V , edges E , and triangle faces F . Vertices are labeled points. Edges are 2-tuples of vertices that are connected. Faces are 3-tuples of vertices that form triangles. Two vertices are said to be neighbors if they are connected by an edge. This abstract mesh only defines connectivity and does not yet define a surface in space.

To get a surface in \mathbb{R}^3 we also need the discrete map $f : \mathcal{K} \rightarrow \mathbb{R}^3$ (fig. 2.4), analogous to the map used to define continuous surfaces. Instead of some subset of \mathbb{R}^2 , the discrete map takes the abstract set of vertices and their connectivity (edges and faces) to \mathbb{R}^3 . It is enough to only specify where the vertices go: edges can be found by the line segment between two neighboring vertices. Three points in \mathbb{R}^3 specify a plane, so triangle faces can also be reconstructed.

2.5 Mass matrix

The mass matrix M is a symmetric $n \times n$ matrix, where n is the number of vertices. It is important for the study of discrete surfaces as virtually every calculation done on a mesh involves the mass matrix. It accounts for the areas around each vertex and is defined as:

$$M_{ij} = \begin{cases} \frac{1}{6} \sum_{k \sim i} |T_{ik}|, & i = j \\ \frac{1}{12} (|T_1| + |T_2|), & i \sim j \\ 0, & \text{otherwise} \end{cases} \quad (2.9)$$

where $i \sim j$ means vertices that are connected by an edge, $|T_{ik}|$ are the areas of triangles around vertex i and $|T_1|$ and $|T_2|$ are the areas of the triangles sharing an edge (i, j) [25]. By taking the row sum of the mass matrix, we get the vertex area A_i , which is one third of the area of the triangles around a vertex.

$$A_i = \sum_{j \in V} M_{ij} = \frac{1}{3} \sum_{k \sim i} |T_{ik}| \quad (2.10)$$

Often, a ‘lumped’ mass matrix [15] is used, which just has the vertex areas on the diagonal:

$$M_{ij}^{(lumped)} = \begin{cases} \frac{1}{3} \sum_{k \sim i} |T_{ik}|, & i = j \\ 0, & \text{otherwise} \end{cases} \quad (2.11)$$

The mass matrix discretizes the inner product of two functions $f, g : \mathcal{M} \rightarrow \mathbb{R}$, sampled at each vertex:

$$\begin{aligned} \mathbf{f}_i &= f(\mathbf{v}_i) \\ \mathbf{g}_i &= g(\mathbf{v}_i) \\ \langle f, g \rangle &= \int_{\mathcal{M}} f g dS \approx \mathbf{f}^T M \mathbf{g} \end{aligned} \quad (2.12)$$

A short derivation is given in the final section of appendix A. The inner product directly leads to other useful calculations, such as the squared norm,

$$\|f\|^2 = \langle f, f \rangle \approx \mathbf{f}^T M \mathbf{f} \quad (2.13)$$

the integral,

$$\int_{\mathcal{M}} f dS = \langle 1, f \rangle \approx \sum_{i \in V} (M \mathbf{f})_i \quad (2.14)$$

and the total area A of a mesh:

$$A = \int_{\mathcal{M}} dS \approx \sum_{i \in V} \sum_{j \in V} M_{ij} = \sum_{i \in V} A_i \quad (2.15)$$

As expected, the total area is just the sum of the vertex areas.

Chapter 3

Spherical parameterization

3.1 Introduction

The objective of spherical parameterization is to find a global map $f : S^2 \rightarrow \mathbb{R}^3$ from the unit sphere to an arbitrary surface. There has been a lot of interest in finding spherical parameterizations. Many of the techniques come from the computer graphics community because such parameterizations are useful for e.g. texture mapping. These algorithms work on discrete meshes. In this setting, the parameterization gives us a map from every vertex to the sphere. This is equivalent to assigning spherical coordinates (θ, φ) to every vertex.

This mapping necessarily induces some distortion. Angles, distances and areas as measured on the original surface will not be the same when measured on the sphere. A map that preserves angles is called conformal.

Broadly, the algorithms can be divided into two groups. The first use some kind of nonlinear optimization routine to find a spherical map under various constraints. The second group are newer algorithms that use geometric flow. The intuition behind such flow is that the mesh is iteratively smoothed until all the irregularities are gone. The flow-based algorithms are generally more efficient.

3.2 Optimization algorithms

The first approach, taken by Brechbühler et al. [6] and Du et al. [11], consists of three steps. The first step determines the latitude θ for each vertex. Two vertices are selected as the north and south pole. They set $\theta_{north} = 0$ and $\theta_{south} = \pi$ at the poles. The other θ values are calculated by diffusing the values from the poles outwards over the surface. The second step determines the longitude φ . A date line is chosen where φ jumps from 0 to 2π and similarly the values are diffused over the surface. The final result of these steps is a spherical mapping, since each vertex is assigned coordinates (θ, φ) , but the distortion is still high. In the last step they optimize this initial parameterization using a non-linear constrained minimization.

A second approach, proposed by Friedel et al. [13] uses unconstrained non-linear minimization. They show that common distortion energies can be modified so that

the surfaces converges to a sphere when the energy is minimized. This works even when starting with an initial configuration where they project every vertex to the closest point on a sphere. By taking a weighted average of area and angle errors, the trade-offs can be balanced towards either area or angle preservation.

3.3 Flow-based algorithms

In a geometric flow, each point on the surfaces moves along the direction given by a gradient field. By choosing a direction that minimizes curvature, the surface is progressively smoothed (fig. 3.1).

Kazhdan et al. [19] describe what they call the conformalized mean curvature flow (cMCF). Mean curvature flow is a geometric flow that minimizes the mean curvature. Because the mean curvature flow is known to develop singularities, they derive a new method that modifies the mean curvature flow slightly, so it is conformal (angle preserving) and does not suffer from singularities.

A different kind of flow is proposed by Crane et al. [9]. They derive a robust method that finds conformal maps using Willmore flow. Willmore flow is quite similar to mean curvature flow but uses the squared mean curvature in its formulation. Crane et al. developed a fast method for solving this flow in the discrete setting. This algorithm has successfully been used for nuclear shape classification [2].

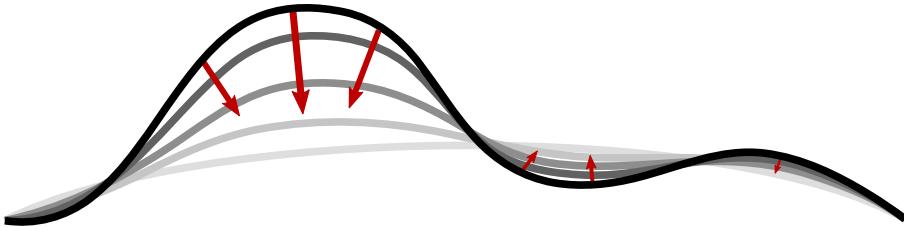


FIGURE 3.1: Schematic representation of surface smoothing with geometric flow.

3.4 Conformalized mean curvature flow

Because the cMCF is used in the final pipeline, it is discussed here in more detail.

3.4.1 Mean curvature flow

The idea of mean curvature flow (MCF) is to gradually smooth out the surface. As the surface details get smoothed out, this should converge to a sphere. In MCF, the surface moves along its normals, with speed proportional to the mean curvature. Specifically, let \mathbf{F}_t be a map from a surface \mathcal{M} to \mathbb{R}^3 at time t , then we evolve this map as:

$$\frac{\partial \mathbf{F}_t}{\partial t} = -2H\hat{\mathbf{n}} \quad (3.1)$$

3. SPHERICAL PARAMETERIZATION

Using eq. (2.8) we can rewrite this as:

$$\frac{\partial \mathbf{F}_t}{\partial t} = \Delta_t \mathbf{F}_t \quad (3.2)$$

Note the subscript t for the Laplacian, since it depends on the surface which is itself changing in the flow. We can see that this is analogous to the heat equation, $\frac{\partial u}{\partial t} = \Delta u$. Where the heat equation smooths out a function over time, the MCF smooths the geometry itself.

3.4.2 Discrete mean curvature flow

Discretizing the MCF is done with the vector of vertex coordinates \mathbf{v} , which are the coefficients for the coordinate function. A complete derivation is given in appendix A. Combining eq. (3.2) with discretized Laplacian:

$$\Delta \mathbf{F}_t \approx (\mathbf{M}_t)^{-1} \mathbf{L}_t \mathbf{v}_t \quad (3.3)$$

Time is discretized with an implicit (backwards) Euler scheme and timestep h :

$$\frac{\partial \mathbf{F}_t}{\partial t} \approx \frac{\mathbf{v}^{t+h} - \mathbf{v}^t}{h} = (\mathbf{M}_t)^{-1} \mathbf{L}_t \mathbf{v}_{t+h} \quad (3.4)$$

\Downarrow

$$(\mathbf{M}_t - h \mathbf{L}_t) \mathbf{v}_{t+h} = \mathbf{M}_t \mathbf{v}_t \quad (3.5)$$

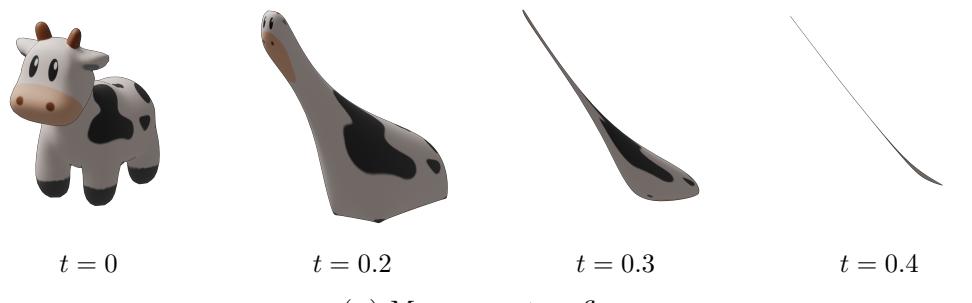
This equation actually denotes 3 systems, one for each coordinate, that can be solved independently. For the matrices, the subscript t denotes that they are recalculated every iteration. Since most of the entries in the mass and stiffness matrices are zero, the result is a sparse linear system. Sparse linear systems can be solved very efficiently and there are many numerical methods available that deal specifically with these kinds of problems.

3.4.3 Conformal modification

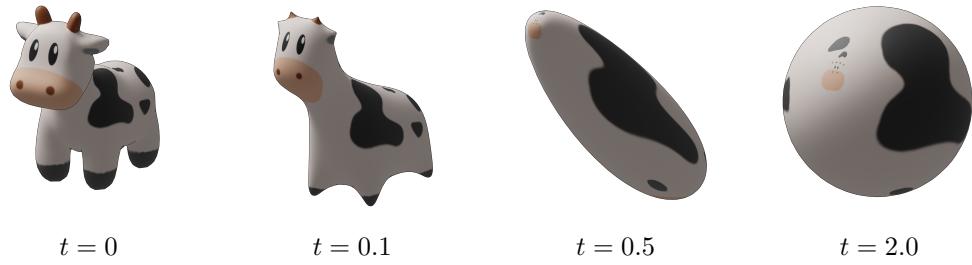
A problem with the MCF is that it can develop singularities over time. Features on the mesh collapse and the numerical scheme becomes unstable, as shown in fig. 3.2. The main contribution of Kazhdan et al. [19] is the development of a modification they call the conformalized mean curvature flow (cMCF), which has a very simple formulation when discretized. cMCF uses the same update rule as eq. (3.5), but instead of recalculating the stiffness matrix, we always re-use the matrix from the first timestep $t = 0$:

$$(\mathbf{M}_t - h \mathbf{L}_0) \mathbf{v}_{t+h} = \mathbf{M}_t \mathbf{v}_t \quad (3.6)$$

They also show that this flow is conformal (angle preserving). This is a useful property because it means that the aspect-ratio of triangles is preserved. A good rule of thumb for triangle meshes is that the triangles should be as close to equilateral as possible [30]. A conformal mapping will preserve these without distorting them. Results of this algorithm are shown in fig. 3.2.



(A) Mean curvature flow.



(B) Conformalized mean curvature flow.

FIGURE 3.2: Evolution of the unmodified and conformalized mean curvature flows on Spot the cow. Timestep $h = 0.1$. For the unmodified flow the mesh collapses into a line.

Chapter 4

Spherical harmonics decomposition

After obtaining the spherical parameterization, we can continue with the spherical harmonics decomposition. Given a spherical parameterization for every vertex:

$$\begin{aligned}\mathbf{v}_i &= (x_i, y_i, z_i) \\ \omega_i &= (\theta_i, \varphi_i)\end{aligned}\tag{4.1}$$

The vertex positions are just three functions, sampled at every vertex. The goal is now to recover the original mesh, using a linear combination of spherical harmonics. For each coordinate, we seek coefficients c_ℓ^m as in eq. (1.2):

$$x_i = \sum_{\ell=0}^N \sum_{m=-\ell}^{\ell} c_\ell^m \mathcal{Y}_\ell^m(\omega_i)\tag{4.2}$$

up to some maximum degree $N = \ell_{max}$. This procedure is repeated for each coordinate in \mathbf{v} .

4.1 Least squares solution

A simple way to estimate the coefficients is to solve a single least squares problem. This approach is used in the method of Agus et al. [2] Contracting the indices m and ℓ into a single index $k = ((0, 0), (1, -1), (1, 0), (1, 1), (2, -2), \dots)$ which has a range from 1 to $K = (N + 1)^2$, we can rewrite eq. (4.2):

$$x_i = \sum_{k=1}^K c_k \mathcal{Y}_k(\omega_i)\tag{4.3}$$

In matrix form this becomes:

$$x = \mathbf{Y}c\tag{4.4}$$

The $n \times K$ matrix \mathbf{Y} is defined:

$$\mathbf{Y} = \begin{bmatrix} \mathcal{Y}_1(\omega_1) & \dots & \mathcal{Y}_K(\omega_1) \\ \vdots & \ddots & \vdots \\ \mathcal{Y}_1(\omega_n) & \dots & \mathcal{Y}_K(\omega_n) \end{bmatrix} \quad (4.5)$$

x is the column vector of coordinates, with length n (the number of vertices), and c is a column vector coefficients with length $K = (N + 1)^2$. This is an overdetermined system, so we can minimize the squared residuals:

$$\hat{c} = \underset{c}{\operatorname{argmin}} \|x - \mathbf{Y}c\|^2 \quad (4.6)$$

This has a closed form solution using ordinary least squares:

$$\hat{c} = (\mathbf{Y}^\top \mathbf{Y})^{-1} \mathbf{Y}^\top x \quad (4.7)$$

While simple, this approach has some problems. It is not very efficient since the matrix $\mathbf{Y}^\top \mathbf{Y}$ has size $K \times K = (N + 1)^2 \times (N + 1)^2$, so it becomes expensive to invert for a high maximum degree. It also has the effect that the estimated coefficients will depend on which maximum degree we choose.

4.2 Iterative residual fitting

Chung et al. derive an iterative scheme based on least squares they call iterative residual fitting [7]. The least squares problem is split up into a number of smaller subproblems, exploiting the hierarchical nature of the spherical harmonics.

The problem is split into multiple subproblems, one per degree ℓ . Let c_ℓ be the column vector of estimates with length $2\ell + 1$, and \mathbf{Y}_ℓ a matrix with size $n \times (2\ell + 1)$:

$$\mathbf{Y}_\ell = \begin{bmatrix} \mathcal{Y}_\ell^{-\ell}(\omega_1) & \dots & \mathcal{Y}_\ell^\ell(\omega_1) \\ \vdots & \ddots & \vdots \\ \mathcal{Y}_\ell^{-\ell}(\omega_n) & \dots & \mathcal{Y}_\ell^\ell(\omega_n) \end{bmatrix} \quad (4.8)$$

Set the initial residuals $r_0 = x$. The first subproblem is then to find \hat{c}_0 that minimizes:

$$\hat{c}_0 = \underset{c_0}{\operatorname{argmin}} \|r_0 - \mathbf{Y}_0 c_0\|^2 \quad (4.9)$$

The estimated coordinates are $\mathbf{Y}_0 \hat{c}_0$ so that the new residual is:

$$r_1 = r_0 - \mathbf{Y}_0 \hat{c}_0 \quad (4.10)$$

This procedure continues iteratively. At step j we find \hat{c}_j that minimizes:

$$\hat{c}_j = \underset{c_j}{\operatorname{argmin}} \|r_j - \mathbf{Y}_j c_j\|^2 \quad (4.11)$$

4. SPHERICAL HARMONICS DECOMPOSITION

Updating the residual:

$$\begin{aligned} r_{j+1} &= r_0 - \mathbf{Y}_1 \hat{c}_1 - \mathbf{Y}_2 \hat{c}_2 - \dots - \mathbf{Y}_j \hat{c}_j \\ &= r_j - \mathbf{Y}_j \hat{c}_j \end{aligned} \quad (4.12)$$

At each iteration j , eq. (4.11) is solved using ordinary least squares:

$$\hat{c}_j = (\mathbf{Y}_j^\top \mathbf{Y}_j)^{-1} \mathbf{Y}_j^\top r_j \quad (4.13)$$

The final estimate \hat{c} is then just the concatenation of all the \hat{c}_j .

This approach provides a solution to the problems mentioned above for the ‘naive’ ordinary least squares. The matrix $\mathbf{Y}_j^\top \mathbf{Y}_j$ has size $(2j + 1) \times (2j + 1)$, which is much faster to invert. Also, the coefficients are fitted in order of increasing degree, making the procedure independent of how many degrees we wish to fit.

Chapter 5

C. elegans embryology

The embryogenesis of the roundworm *Caenorhabditis elegans* is a powerful model for animal development and multicellular self-organization. Three Nobel Prizes have been awarded for work on the roundworm in recent years. The first one for the study of apoptosis by Brenner, Sulston and Horvitz in 2002, a second one for the mechanism of RNA interference gene knockdown by Fire and Mello in 2006, and the last one for the work on the GFP protein by Chalfie in 2008.

5.1 Description

C. elegans is a small non-parasitic nematode. It is about 1mm in length and lives in the soil where it feeds on bacteria or yeast. They occur mostly as hermaphrodites that are self-fertile. Males are induced in stress conditions such as starvation. A number of reasons make it a very suitable model organism:

- They are cheap and easy to cultivate.
- Each worm lays around 300 eggs.
- Its lifespan is around 3 days.
- They can be frozen for long-term storage.
- The worms are fully transparent making microscopic observation easy.

5.2 Embryology

C. elegans is eutelic, meaning it has a fixed number of cells (959 in hermaphrodites and 1031 in males) as well as invariant cell lineage. This makes it easy to detect differences in development for mutants. Despite its low number of cells and relative simplicity, it is a complete animal with intestines, gonads, a nervous system, muscles, etc. It does not, however, have a circulatory or respiratory system. Crucially, many of the processes that are specific to multicellular systems are present in the worm, such as apoptosis, Wnt and Notch signalling.

5.3 Shape and cell mechanics

During embryonic development, cells divide and move in a way that is highly invariant between individuals. How this development is orchestrated is a fundamental question, particularly in the context of perturbations. By analyzing the shape of the cells during development, we can hopefully gain new insights into the processes and force generation that enables this self-organization. Using digital methods, we can screen for phenotypes that are not readily identified by human observation alone.

5.4 7-cell stage

Early development of the embryo is characterized by a number of asymmetric cell divisions [21]. Since the cell lineage is invariant, each cell can be given a name (fig. 5.1). The first progenitor (P0) cell divides to give the P1 and AB cells (2-cell stage). After this, the AB cell splits into the ABa and ABp cells, and the P1 cell into the EMS and P2 cells (4-cell stage). The ABa and ABp cells then give rise to the ABal, ABar; ABpl and ABpr cells. The EMS cell divides to give the MS and E cells. At this point there are 7 cells.

The 7-cell stage is of interest because at this point, all the major axes of the organism are decided (anterior-posterior, left-right and dorsal-ventral). Around this timepoint the ABpl cell also transiently makes a lamellipodium, a protrusion that is associated with movements that are part of the formation of the left-right asymmetry in the worm [24]. Later in this thesis we will examine this 7-cell stage more closely, specifically looking at the cell shapes immediately after the division of the EMS cell.

5.5 Perturbations

Two genetic perturbations of the worm were studied. The first is a *dsh-2/mig-5* double RNA interference knockdown, the second is a *fmi-1* mutant.

5.5.1 *dsh-2* and *mig-5*

DSH-2 and MIG-5 are part of the Dishevelled (DSH) family of proteins. These proteins are involved in canonical and non-canonical Wnt signalling pathways. Dishevelled plays important roles in cellular differentiation and cell polarity. Wnt signaling plays a role in determining cell fate at the 4-cell stage, but also has important roles later in development, such as in orienting cell division spindles in EMS cells, and neuronal cell migration [26].

5.5.2 *fmi-1*

FMI-1 (Flamingo) is an atypical adhesion protein related to both cadherins and G protein-coupled receptors. Apart from being involved in the development of neurons [23], not much is known about its role in embryonic development. Its ortholog in *Drosophila* is well studied [33, 31]. Flamingo is involved in the planar cell polarity

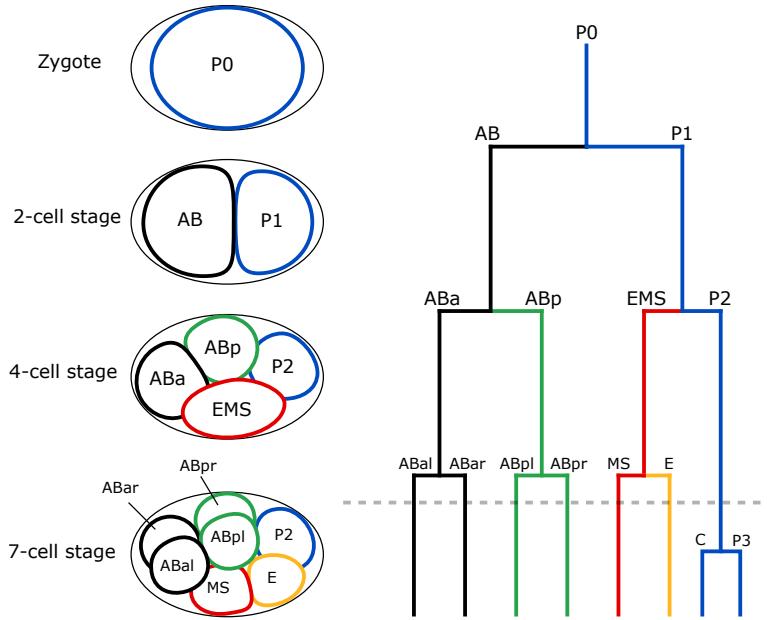


FIGURE 5.1: Diagram of early cell stages. The dashed line shows the chosen timepoint. Figure adapted from Wormbook [21].

(PCP) pathway, one of the non-canonical Wnt signaling pathways [26]. The PCP pathway establishes cell polarization, a way for cells to orient themselves spatially. This polarization plays a crucial role in developmental processes. It is also important in asymmetric cell division, such as when the EMS cell divides into the E and MS cells. This asymmetric division determines the fate of the daughter cells. E cells become endodermal and MS cells mesodermal [26].

Chapter 6

Motivation & objectives

6.1 Motivation

Understanding how cells self-organize into multicellular systems is a fundamental challenge in biology. Past research has focused mainly on molecular factors involved in signaling and cellular differentiation. However, how cells exert forces and interact is also central to understanding self-organization. Indeed, the shape of a cell contains a lot of information about its state. Shape can reflect forces such as adhesion to neighboring cells, contractility of the cortex. As cells differentiate, they also change shape, which can be used to identify cells in complex mixtures of cells such as in the brain, or to identify if a cell is malignant in cellular pathology examination.

During *C. elegans* embryogenesis cells rapidly change shape during development, reflecting the active processes of cellular signaling, differentiation and positioning. To classify these shapes in the normal state and to then compare them to perturbed states (e.g. due to gene knockdown) a formal shape description is required.

The spherical harmonics decomposition is a powerful tool that can be used to describe shape in an objective way. They have not been used to study *C. elegans* embryology before, which will be explored in this work. As no suitable toolbox was readily available to work with the SH decomposition for cell shape, a new pipeline was developed.

6.2 Objectives

The objective of this work is to derive a framework for the interpretation and quantification of cell shapes, using the spherical harmonics decomposition as a starting point. There are three problems that need to be solved:

- Finding a map from the shape to the sphere.
- Using this map to calculate the spherical harmonics decomposition.
- Interpreting the spherical harmonics

The goal is to develop a tool that can be used in the context of *C. elegans* research. This will require the integration of these methods into a software pipeline with a focus on modularity and speed and the ability to derive specific biological features of

6.2. Objectives

interest from these generic shape descriptors. Finally, the methods will be put to the test by applying them to shapes from perturbed *C. elegans* embryos. This work will build on the work of W. Thiels, who previously introduced a novel pipeline for extracting 3D meshes from confocal microscopy data for *C. elegans* embryos [32].

Part II

Implementation

Chapter 7

Implementation

The pipeline is implemented in the Python programming language. Python has many advantages: it is a high level programming language, the code is usually very human-readable, its features are easy to extend, it is cross-platform, and it has a rich ecosystem of modules. It is also completely free and open source. While high level languages can be slow for numerical computations, this can be largely resolved using the NumPy [14] and SciPy [34] packages. These provide a Python interface to efficient numerical schemes which are implemented in low-level languages such as C and Fortran. Additionally, libigl [16] is used to handle the loading of meshes, as well as calculating the mass and stiffness matrices. libigl is a cross-platform geometry processing library written in C++ and offers bindings for Python.

Python supports multiple programming paradigms, such as object-oriented and functional programming. For this pipeline a functional approach was chosen. In functional programming, the problem is decomposed into modular functions, each of which has no side effects. This makes it easy to reason about the program and verify that it is correct. The data structures used are multi-dimensional arrays which are directly interoperable with NumPy, SciPy and libigl. The Python package is open source and available at: <https://github.com/Sin-tel/spheremesh>. It includes some examples that show how to use the pipeline.

The overall steps that need to be taken to calculate the spherical harmonics decomposition are:

1. Read an input triangle mesh (section 7.1).
2. Run the spherical parameterization algorithm using the conformalized mean curvature flow (section 7.2).
3. Calculate the unique map using Möbius centering (section 7.3).
4. Find the canonical rotation (section 7.4).
5. Fit the spherical harmonics decomposition using iterative residual fitting (section 7.5).

In the following sections each of the steps in the pipeline are discussed in more detail.

7.1 Triangle mesh data structures

The loading of triangle meshes is handled by `libigl`. Supported formats are: Wavefront object (`.obj`), 3D Systems STL (`.stl`), polygon file format (`.ply`) and Geomview polyhedral file format (`.off`).

The mesh is represented by two simple data structures: the vertices and faces. The vertices are an array of triplets of numbers in arbitrary order, each representing a 3D coordinate in space. The faces are triplets of vertex indices that represent a connected triangle. Because all of the algorithms preserve the mesh connectivity, the faces are never updated, so the operations only operate update the vertices.

7.2 Spherical parameterization

7.2.1 Conformalized mean curvature flow

Of all the available methods to calculate the spherical parameterization (see chapter 3), the conformalized mean curvature flow (cMCF) was chosen. The two flow-based methods are much more efficient than the methods based on nonlinear optimization. The method by Crane et al. [9] (using Willmore flow) is more generally applicable as it also works on higher genus surfaces, but uses quaternionic operators that are not currently supported in `libigl`. While the cMCF method solves a linear system each timestep, in the Willmore flow an eigenvalue problem must be solved, which is generally slower.

Since both methods converge to conformal maps, they are equivalent for this application. That leaves the cMCF algorithm, which is relatively easy to implement. The core of the algorithm is eq. (3.6), repeated here for clarity:

$$(M_t - hL_0)\mathbf{v}_{t+h} = M_t\mathbf{v}_t \quad (7.1)$$

Each iteration, this equation must be solved. It requires the mass and stiffness matrices (which are provided by `libigl`) and a sparse matrix solver, provided by SciPy (`scipy.sparse.linalg.spsolve`). Since the flow evolves towards a single point, the mesh is uniformly scaled after each iteration such that the maximum distance from the center to the surface is one. This scaling also helps to keep the numbers in a good range, which is necessary to maintain numerical stability.

Because of the stability of the implicit scheme, we can take quite large timesteps (here we take $h = 0.1$) resulting in a map very close to a sphere after only 20 iterations, as shown in fig. 3.2.

7.2.2 Projection

To make sure the final map exactly corresponds to a sphere, the vertices $\mathbf{v}_i \in V$ are projected to the unit sphere around the origin:

$$\mathbf{v}'_i = \frac{\mathbf{v}_i - \bar{\mathbf{v}}}{\|\mathbf{v}_i - \bar{\mathbf{v}}\|} \quad (7.2)$$

where $\bar{\mathbf{v}}$ is the average vertex position.

7.2.3 Spherical coordinates

After obtaining a spherical parameterization, the spherical angular coordinates $\omega_i = (\theta_i, \varphi_i)$ can be calculated for each vertex:

$$\begin{aligned}\varphi &= \tan^{-1} \left(\frac{y}{x} \right) \\ \theta &= \cos^{-1} (z)\end{aligned}\tag{7.3}$$

with θ the colatitude ($0 \leq \theta \leq \pi$) and φ the longitude ($0 \leq \varphi < 2\pi$) (fig. 1.2). The arctangent is defined to take into account the correct quadrant (usually known as the `atan2` function).

7.3 Möbius transformations

In their paper on the conformalized mean curvature flow, Kazhdan et al. [19] give no proof of convergence. They do show that if the flow converges, it does so to a conformal map. The flow converged for each of the 269 cells in the dataset, so empirically there are no convergence issues.

By the uniformization theorem, a conformal map always exists, at least for the surfaces that are considered here (genus zero and without boundary). However, the uniformization theorem does not guarantee a unique map. Indeed, there are many conformal maps from the unit sphere to itself, known as the Möbius transformations. Applying a Möbius transformation to the map obtained from the cMCF gives another conformal map from the shape to a sphere. To deal with this, we will try to find a ‘canonical’ Möbius transformation by adding constraints.

7.3.1 Inversions

General Möbius transformations on the sphere can be found as compositions of inversions and rotations. We will deal with inversions first. Rotations will be handled later in section 7.4. An inversion $\eta_{\mathbf{c}} : S^2 \rightarrow S^2$ with center \mathbf{c} is defined as:

$$\eta_{\mathbf{c}}(\mathbf{v}) = (1 - \|\mathbf{c}\|^2) \frac{\mathbf{v} + \mathbf{c}}{\|\mathbf{v} + \mathbf{c}\|^2} + \mathbf{c}\tag{7.4}$$

with $\|\mathbf{c}\| < 1$. Inversions can be understood as fixing the poles that lie on the axis from the origin to \mathbf{c} and ‘pushing’ the geometry towards \mathbf{c} .

7.3.2 Möbius centering algorithm

Baden et al. [4] describe an algorithm to find a canonical inversion. They note that a conformal parameterization of a surface defines a local conformal scale factor λ (fig. 7.1). For a conformal map $f : \mathcal{M} \rightarrow S^2$ this factor is given by the local scaling of the area element dA on \mathcal{M} corresponding to the area element dA_{S^2} on the sphere:

$$dA = \lambda dA_{S^2}\tag{7.5}$$

7. IMPLEMENTATION

so that $\lambda : S^2 \rightarrow \mathbb{R}_{>0}$. This can be interpreted as a mass density on the sphere, so that the corresponding center of mass is:

$$\mu = \int_{S^2} x \lambda(x) dA_{S^2} \quad (7.6)$$

The canonical map is then $\eta_c \circ f$ such that it satisfies the constraint $\mu = 0$. To accomplish this they minimize the energy

$$E = \frac{1}{2} \|\mu(\mathbf{c})\|^2 \quad (7.7)$$

with respect to \mathbf{c} . The map resulting from the cMCF is usually already quite close to satisfying this constraint, so the algorithm does not need many iterations to converge.

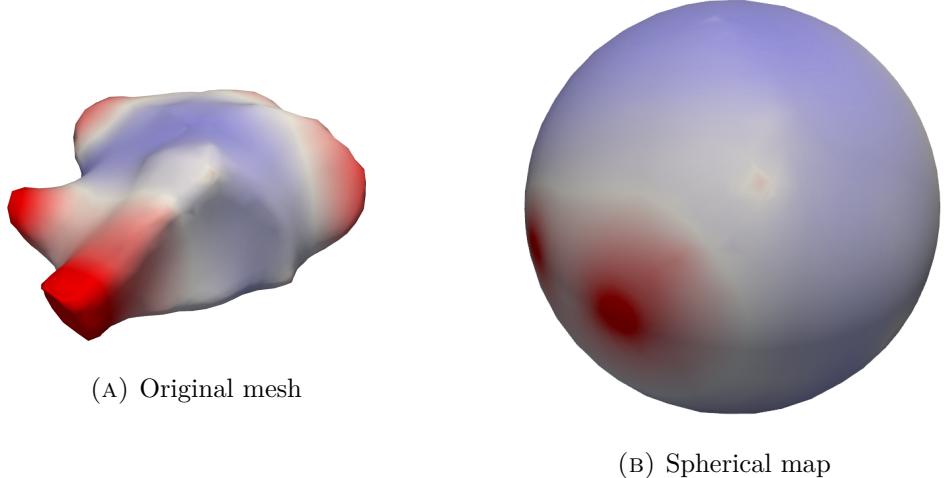


FIGURE 7.1: Conformal scale factor λ plotted on a wild-type ABpl cell and its spherical map. Red means $\lambda > 1$, blue means $\lambda < 1$.

7.4 Canonical rotation

To get shape descriptors that are independent of rotations, a canonical rotation is calculated. The idea is described by Brechbühler et al. [6], but since they use complex spherical harmonics, a new derivation is given here in terms of a singular value decomposition.

The first degree spherical harmonics have a particularly simple form when written in Cartesian coordinates:

$$\begin{aligned}\mathcal{Y}_1^{-1} &= -\sqrt{\frac{3}{4\pi}}y \\ \mathcal{Y}_1^0 &= \sqrt{\frac{3}{4\pi}}z \\ \mathcal{Y}_1^1 &= -\sqrt{\frac{3}{4\pi}}x\end{aligned}\tag{7.8}$$

Rewriting this gives:

$$\begin{aligned}x &= -\sqrt{\frac{4\pi}{3}}\mathcal{Y}_1^1 \\ y &= -\sqrt{\frac{4\pi}{3}}\mathcal{Y}_1^{-1} \\ z &= \sqrt{\frac{4\pi}{3}}\mathcal{Y}_1^0\end{aligned}\tag{7.9}$$

This means the first degree expansion is just a linear transformation of the sphere. The coefficients can be written as a matrix

$$C = \begin{bmatrix} c_{xx} & c_{yx} & c_{zx} \\ c_{xy} & c_{yy} & c_{zy} \\ c_{xz} & c_{yz} & c_{zz} \end{bmatrix}\tag{7.10}$$

where the rows are the expansion coefficients of each coordinate function with corrected signs, scaling and order as in eq. (7.9). Any linear transformation of a sphere results in an ellipsoid, so the first degree SH expansion is called the first order ellipsoid (FOE) [6]. This terminology is somewhat confusing, it should really be called the first degree ellipsoid, but the term has already caught on.

Ellipsoids have three principal axes that are orthogonal. We can use this fact to define a canonical rotation. With \mathbf{s} the Cartesian coordinates on the sphere and \mathbf{e} the coordinates on the ellipsoid:

$$\mathbf{e} = C\mathbf{s}\tag{7.11}$$

Taking the singular value decomposition of C :

$$C = U\Sigma V^T\tag{7.12}$$

$$U^T U = I$$

$$V^T V = I$$

$$\Sigma = \begin{bmatrix} a_x & 0 & 0 \\ 0 & a_y & 0 \\ 0 & 0 & a_z \end{bmatrix}$$

7. IMPLEMENTATION

U and V are rotation matrices, (a_x, a_y, a_z) are the principal axes of the ellipsoid and $a_x \geq a_y \geq a_z > 0$. The canonical rotation can then be found by rotating the sphere s and ellipsoid e :

$$\begin{aligned} e' &= U^T e \\ s' &= V^T s \end{aligned} \tag{7.13}$$

so that:

$$\begin{aligned} e &= Cs \\ e &= U\Sigma V^T s \\ U^T e &= \Sigma V^T s \\ e' &= \Sigma s' \end{aligned} \tag{7.14}$$

This canonical rotation rotates the FOE so that its largest principal axis is aligned with the x-axis, its second principal axis with the y-axis and its smallest principal axis with the z-axis.

There are two rotations that are taken care of by the calculations above: the first in the object space (of the original mesh) of the shape and the second in the parameter space (of the sphere). It is possible that we are not interested in a canonical rotation for the original mesh. As discussed in section 7.3.2, both the inversions and rotations of the spherical mapping need to be fixed. So to only take care of the rotation in parameter space the following can be done:

$$s' = UV^T s \tag{7.15}$$

so that:

$$\begin{aligned} e &= Cs \\ e &= U\Sigma V^T s \\ e &= U\Sigma U^T UV^T s \\ e &= U\Sigma U^T s' \end{aligned} \tag{7.16}$$

where the resulting transformation matrix $U\Sigma U^T$ is symmetric. Symmetric matrices contain no rotational component.

7.5 Spherical harmonics decomposition

While a regular least squares fit gives less global error, the overfitting is significant and the resulting coefficients are unstable. Because the meshes are discrete, the spherical harmonics are no longer exactly orthogonal when they are only evaluated at the vertices. This makes the convergence fail as we fit more degrees. So there is a trade-off where the IRF gives a worse global approximation, but is very stable compared to regular least squares.

The overfitting could also be mitigated by regularizing the least squares. Because of the hierarchical nature, the regularization parameter has to depend on the order. It is not easy to find the optimal tuning for the regularization, and it might need to change depending on the input shape.

With this in mind, the iterative residual fitting algorithm was chosen for this pipeline. As discussed in section 4.2, it is more efficient than the least squares formulation and the coefficients do not depend on the maximum degree we wish to fit.

7.5.1 Unequal meshing

There is one subtle problem with the algorithm, which is solved here. The IRF assumes that the original mesh has a roughly equal distribution of vertices. If this is not the case the more correct squared norm should be used as in eq. (2.13), using the mass matrix (dropping subscript j for clarity):

$$\|r - \mathbf{Y}c\|^2 = (r - \mathbf{Y}c)^\top M(r - \mathbf{Y}c) \quad (7.17)$$

This leads to a generalized (weighted) least squares problem:

$$\begin{aligned} \hat{c} &= \underset{c}{\operatorname{argmin}} (r - \mathbf{Y}c)^\top M(r - \mathbf{Y}c) \\ \hat{c} &= (\mathbf{Y}^\top M \mathbf{Y})^{-1} \mathbf{Y}^\top M r \end{aligned} \quad (7.18)$$

Using this formulation leads to a lower reconstruction error.

7.5.2 Final decomposition

The full decomposition is then obtained after fitting each of the coordinates:

$$\hat{\mathbf{c}} = (\hat{c}_x, \hat{c}_y, \hat{c}_z) \quad (7.19)$$

The reconstructed vertex positions are then:

$$\hat{\mathbf{v}} = \mathbf{Y}\hat{\mathbf{c}} = \sum_{\ell=0}^N \mathbf{Y}_\ell \hat{\mathbf{c}}_\ell = \sum_{\ell=0}^N \sum_{m=-\ell}^{\ell} \hat{\mathbf{c}}_\ell^m \mathcal{Y}_\ell^m(\omega) \quad (7.20)$$

Where \mathbf{Y} and $\hat{\mathbf{c}}$ are the concatenations of all the \mathbf{Y}_ℓ and $\hat{\mathbf{c}}_\ell$, respectively.

Note that since the spherical harmonics can be sampled at any ω , vertex positions can be reconstructed regardless of how the sphere is sampled. If enough spherical harmonics are used to capture the shape, the original mesh is no longer needed and any (regular) sampling of the sphere can be used to reconstruct the shape. This fact shows the strength of this method: the SH coefficients give a complete description of the original shape.

7.6 Algorithm Speed

The meshes in the *C. elegans* dataset contain around 1200 vertices per cell. For these meshes, the whole pipeline runs in less than a second on a consumer grade laptop. The spherical mapping using cMCF takes around 200ms. The speed of the SH decomposition using IRF depends on the number of degrees fitted. With a maximum degree of 16 it takes around 200ms and for a maximum degree of 32 it takes around 500ms.

In an article by Shen et al. [28] the nonlinear optimization method of Brechbühler et al. [6] was used. They report that the spherical parameterization with around 2500 vertices typically took 15 minutes to 3 hours per mesh, with a worst case of 8 hours. The new method gives a very significant speed gain.

Chapter 8

Analysis of spherical harmonics

8.1 First order ellipsoid

Looking at only the FOE can already give a good high-level description of the shape. The principal axes can be used to check if a shape is spherical, elongated, prolate or oblate.

8.2 Smoothing and filtering

Heat equation

To smooth a shape, the heat equation can be solved. For some function f on the sphere, the heat equation is

$$\frac{\partial f}{\partial k} = \Delta f \quad (8.1)$$

where k can be interpreted both as time and diffusivity. Using the fact that the spherical harmonics are eigenfunctions of the Laplacian:

$$\Delta \mathcal{Y}_\ell^m = -\ell(\ell+1) \mathcal{Y}_\ell^m \quad (8.2)$$

Denote f_ℓ^m the coefficients of the SH decomposition, the heat equation can be solved analytically:

$$\begin{aligned} f(0) &= \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} f_\ell^m \mathcal{Y}_\ell^m \\ \frac{\partial \mathcal{Y}_\ell^m}{\partial k} &= -\ell(\ell+1) \mathcal{Y}_\ell^m \\ f(k) &= \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} e^{-\ell(\ell+1)k} f_\ell^m \mathcal{Y}_\ell^m \end{aligned} \quad (8.3)$$

Smoothing a function using the heat equation becomes multiplying each coefficient with a factor that depends only on ℓ .

Convolution

By solving the heat equation in eq. (8.3) we showed that it becomes a product in the spherical harmonics. This is a general result and is known as the convolution theorem: convolution with a kernel in the spatial domain becomes a pointwise product in the frequency (harmonics) domain [10]. Given kernel h and a function f , the convolution (*) is:

$$(h * f)_\ell^m = \sqrt{\frac{2\ell+1}{4\pi}} h_\ell^0 f_\ell^m \quad (8.4)$$

Note that the kernel only has coefficients where $m = 0$, which is equivalent to saying it is axially symmetric. The heat kernel G_k is then (fig. 8.1):

$$(G_k)_\ell^m = \begin{cases} \sqrt{\frac{4\pi}{2\ell+1}} e^{-\ell(\ell+1)k} & \text{if } m = 0 \\ 0 & \text{otherwise} \end{cases} \quad (8.5)$$

This heat kernel is the spherical analogue of the well-known Gaussian distribution [5], which can be derived analogously in Euclidean space. This kind of filtering can be done with any axially symmetric kernel.

By doing the filtering on the spherical domain instead of the original mesh, the conformal scale factor is ignored. The result is that the filtering is locally scaled, but because the map is conformal, the filtering remains isotropic.

Laplacian of Gaussian

The Laplacian of Gaussian is often used as an edge detector filter for 2D images. It is now straightforward to derive an analogous filter for shapes:

$$(\Delta G_k)_\ell^m = \begin{cases} \sqrt{\frac{4\pi}{2\ell+1}} - \ell(\ell+1) e^{-\ell(\ell+1)k} & \text{if } m = 0 \\ 0 & \text{otherwise} \end{cases} \quad (8.6)$$

which can be normalized so that the maximum filter coefficient does not depend on k (fig. 8.1):

$$(\Delta \hat{G}_k)_\ell^m = \begin{cases} \sqrt{\frac{4\pi}{2\ell+1}} - \ell(\ell+1)k e^{1-\ell(\ell+1)k} & \text{if } m = 0 \\ 0 & \text{otherwise} \end{cases} \quad (8.7)$$

Applying this filter to the full decomposition of a shape, as in eq. (7.20), and taking the magnitude:

$$g_i = \left\| \sum_{\ell=0}^N \sum_{m=-\ell}^{\ell} -\ell(\ell+1)k e^{1-\ell(\ell+1)k} \hat{\mathbf{c}}_\ell^m \mathcal{Y}_\ell^m(\omega_i) \right\| \quad (8.8)$$

The resulting scalar function g is large for ridges and protrusions (fig. 8.2).

8.2. Smoothing and filtering

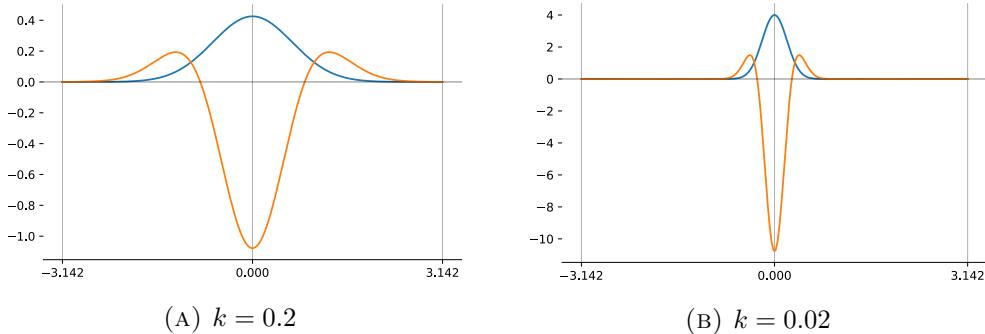


FIGURE 8.1: Heat kernel (blue) and its Laplacian (orange) plotted as a function of θ .

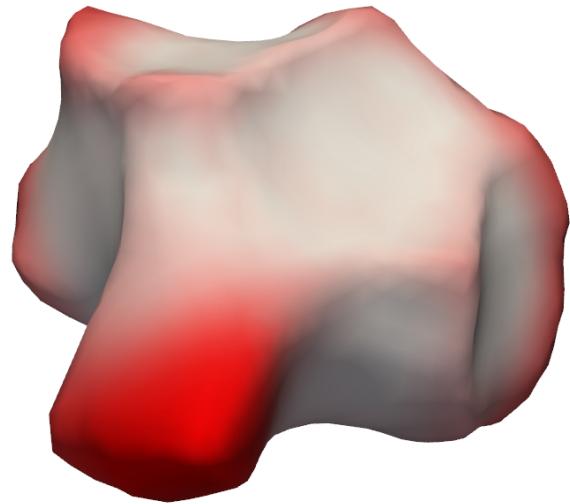


FIGURE 8.2: Magnitude of Laplacian of Gaussian, as in eq. (8.8), plotted on the mesh of a wild-type ABpl cell.

Part III

Application

Chapter 9

Results

The pipeline was used to study meshes of *C. elegans* embryo meshes in the 7-cell stage directly after EMS division. The dataset consists of meshes for 7 wild-type embryos, 3 *dsh-2/mig-5* RNA interference knockdowns and 2 *fmi-1* mutants.

9.1 *dsh-2/mig-5* RNAi knockdowns

In the *dsh-2/mig-5* RNAi knockdowns it was observed that the E cell shows little resistance to getting ‘dented’ by neighboring cells. This implies that there is a less cortical tension, resulting in a shape that is less round overall (fig. 9.1). Using the SH decomposition, the average cell shape can also be calculated, by averaging the

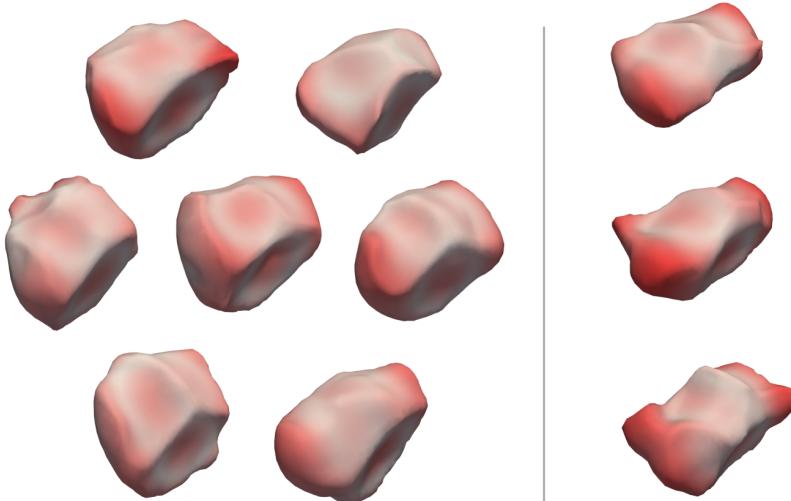


FIGURE 9.1: Overview of all the E cells in the dataset. Wild type on the left and *dsh-2/mig-5* RNAi knockdowns on the right. The contribution of spherical harmonics of degree larger than 1 is highlighted in red.

9. RESULTS

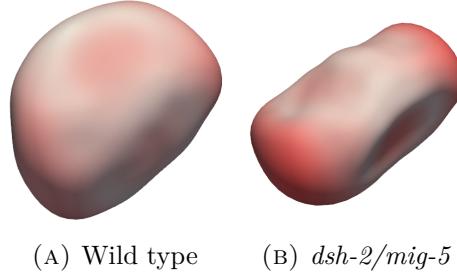


FIGURE 9.2: Averaged shapes for the E cells of wild type and *dsh-2/mig-5* RNAi knockdowns. The contribution of spherical harmonics of degree larger than 1 is highlighted in red.

coefficients and reconstructing the shape (fig. 9.2). The same thing can be seen there.

The spectral power (see eq. (1.7)) of the first degree was calculated and normalized relative to the total power:

$$\hat{S}(1) = \frac{S(1)}{\sum_{n=0}^{16} S(n)} \quad (9.1)$$

This descriptor captures how close the cell is to an ellipsoid. The average normalized power spectrum for the E cells is plotted on a logarithmic scale in fig. 9.3. Overall, the power decreases for higher degrees. Using 16 degrees seems more than enough as the average power drops by 5 orders of magnitude over this range. The difference in the first degree is hard to see because of the logarithmic scale. Since the normalized spectrum sums to 1, the power in the first degree can also be seen as the complement of the power contained in all the higher degrees.

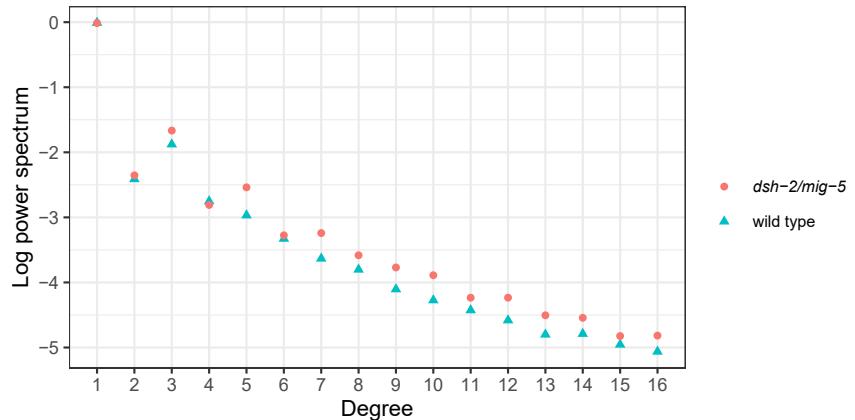


FIGURE 9.3: Average normalized power spectrum of the E cells for mutants and wild types. The y-axis is on a logarithmic scale with base 10.

The relative first degree power was compared between the wild type and *dsh-2/mig-5* RNAi knockdowns. This was done for every cell in the embryo to simulate a situation where we didn't know beforehand what to look for. For 2 of the *dsh-*

9.1. *dsh-2/mig-5* RNAi knockdowns

2/mig-5 RNAi knockdown embryos, only the 8-cell stage was available, so the P2 cell was left out of the analysis, as it has already divided there. The significance was tested using a permutation test and corrected for multiple comparisons using the Benjamini-Hochberg procedure. After adjustment, only the E cells were found to significantly differ in their first degree power. The differences between the cells are shown in fig. 9.4. Differences can also be seen in other cells, but the dataset is too small to say anything conclusive on those.

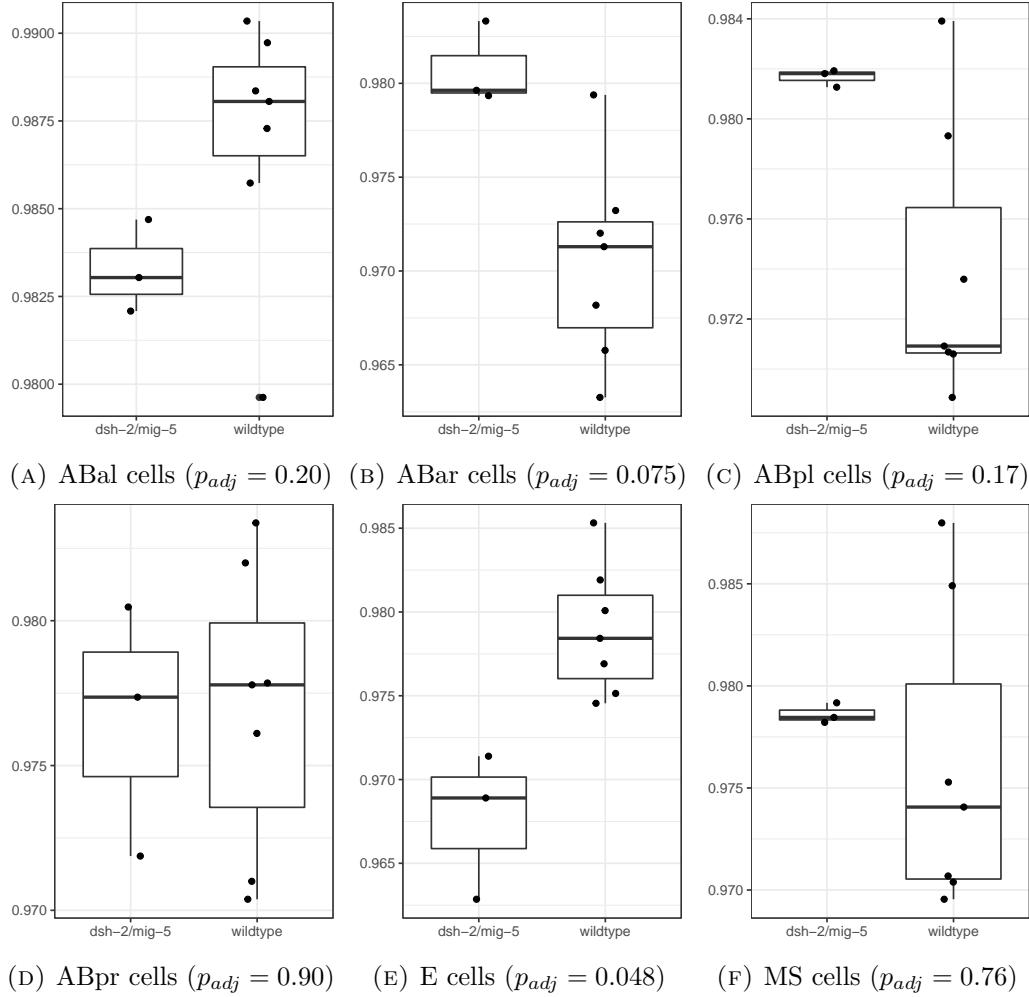


FIGURE 9.4: Relative contribution of the first degree in the power spectrum for cells of *dsh-2/mig-5* RNAi knockdowns versus wild type.

9. RESULTS

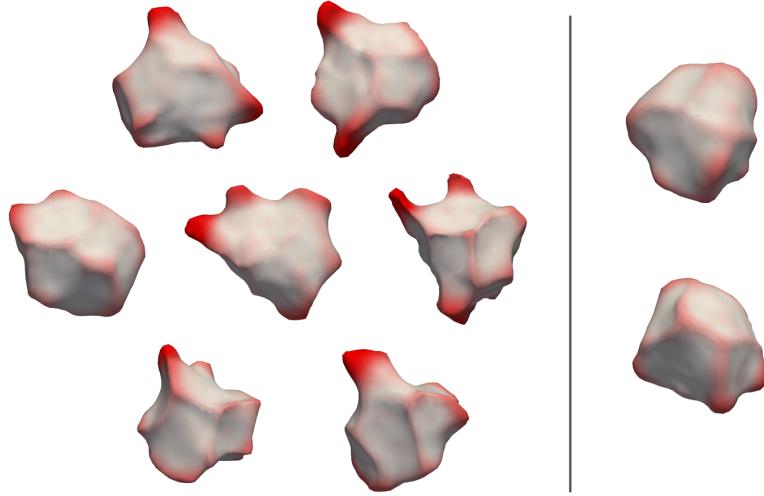


FIGURE 9.5: Overview of all the ABpl cells in the dataset. Wildtype on the left and *fmi-1* mutants on the right. The LoG magnitude is shown in red.

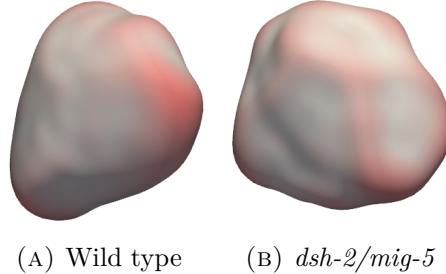


FIGURE 9.6: Averaged shapes for ABpl cells of wild type and *dsh-2/mig-5* RNAi knockdowns. The LoG magnitude is shown in red.

9.2 *fmi-1* mutants

In wild-type embryos, the ABpl cell develops a lamellipodium during the 7-cell stage. It seems that *fmi-1* mutants do not develop these to the same extent as the wild-type embryos (fig. 9.5). Like before, the average shape is also calculated for each group (fig. 9.6). The difference is not as clear here. As the lamellipodia get mapped to small areas on the sphere, they do not contribute much to the first order ellipsoid. This results in a misalignment of the lamellipodia, so they spread out in the average.

To test this hypothesis, the full SH decomposition was calculated for the ABpl cells up to degree 16. Then, the Laplacian of Gaussian filter was applied, and the results integrated over the whole surface of the cells (with $k = 0.005$). This gives a good indication of the total amount and size of protrusions on the cell. The results are summarized in fig. 9.7. Using a permutation test, a significant difference was found even though the sample size was very small. As a control, the same procedure

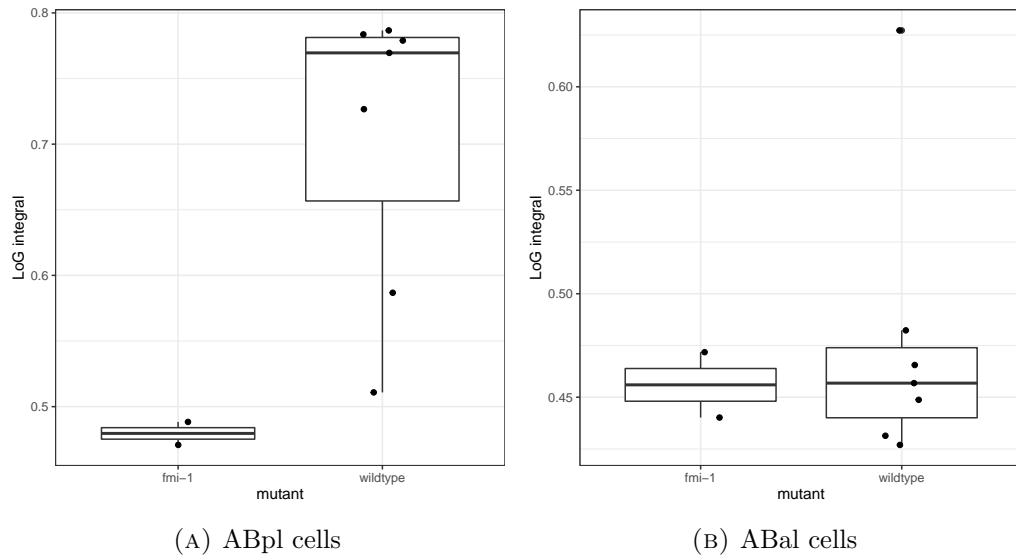


FIGURE 9.7: Integrated Laplacian of Gaussian (LoG) for cells of *fmi-1* mutants versus wild-type specimens. For the ABpl cells the result is significant ($p = 0.028$), for the ABal cells there is no significant difference ($p = 0.33$).

was repeated for the ABal cells, where there was no significant difference.

Chapter 10

Conclusion

10.1 Contributions

A framework is developed to study the shape of cells. Spherical harmonics have been used before in shape analysis, but in this work they are implemented in a very efficient way, and some new tools and interpretations are derived. The combination of conformal spherical parameterization and Möbius transformation provides a unique map to the sphere, without any parameters. Invariance to rotations can be accomplished through the power spectrum or first order ellipsoid alignment. Finally, the unique properties of spherical harmonics are used to derive efficient frequency filtering.

The derivation has strong theoretical foundations and the methods are shown to be a good fit for practical applications. The pipeline is open-source, easy to use, very fast and can be extended for other applications.

10.2 Limitations

While the SH decomposition is a powerful framework to talk about shape, it is not a drop-in replacement for simpler descriptors. Because of its completeness, many coefficients are necessary to describe the shape. While the spherical harmonics of first order are easy to interpret, it is harder to interpret to higher degrees.

The first order ellipsoid rotation can be unstable. Especially for roughly spherical shapes where the principal axes are of similar size. A better method that also takes into account higher order harmonics might be needed.

10.3 Further work

The application studied in this work is fairly limited due to time constraints. Since the available data are time-lapses, further work can be done to track changes in cell types over time.

The new pipeline could also be used to automatically detect new phenotypes or changes in shape when the embryo is subject to perturbations. Such perturbations

can be mechanical (e.g. compression) or genetic. Since the shape description is complete and generic, it could detect changes that might otherwise be missed by visual inspection alone.

Since SH decomposition can also be applied to scalar functions, it can be used to study the concentration of a protein that is located on the membrane or close to it. Cell shape can also reflect force generation and transmission by the cell, such as cortical tension and cell adhesion [37]. Further work would be necessary to link these to spherical harmonics and infer forces from the shape.

Finally, since a generic convolution operator is given for spherical harmonics, all the tools are in place to develop a convolutional neural network to classify cells, that operates with shapes as an input.

Bibliography

- [1] SPHARM-MAT. <https://www.med.upenn.edu/shenlab/spharm-mat.html>. Accessed: 04-06-2021.
- [2] M. Agus, E. Gobbetti, G. Pintore, C. Calì, and J. Schneider. WISH: efficient 3D biological shape classification through Willmore flow and Spherical Harmonics decomposition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4184–4194, 2020.
- [3] T. G. R. Andrews, W. Pönisch, E. Paluch, B. J. Steventon, and E. Benito-Gutierrez. Single-cell morphometrics reveals ancestral principles of notochord development. *bioRxiv*, 2020.
- [4] A. Baden, K. Crane, and M. Kazhdan. Möbius registration. *Computer Graphics Forum*, 37(5):211 – 220, August 2018.
- [5] S. Bigot, D. Kachi, and S. Durand. Spherical edge detector: Application to omnidirectional imaging. In J. Blanc-Talon, S. Bourennane, W. Philips, D. Popescu, and P. Scheunders, editors, *Advanced Concepts for Intelligent Vision Systems*, pages 554–565, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
- [6] C. Brechbühler, G. Gerig, and O. Kübler. Parametrization of Closed Surfaces for 3-D Shape Description. *Computer Vision and Image Understanding*, 61(2):154 – 170, 1995.
- [7] M. K. Chung, R. Hartley, K. M. Dalton, and R. J. Davidson. Encoding cortical surface by spherical harmonics. *Statistica Sinica*, 18(4):1269–1291, 2008.
- [8] K. Crane, F. de Goes, M. Desbrun, and P. Schröder. Digital geometry processing with discrete exterior calculus. In *ACM SIGGRAPH 2013 courses*, SIGGRAPH ’13, New York, NY, USA, 2013. ACM.
- [9] K. Crane, U. Pinkall, and P. Schröder. Robust fairing via conformal curvature flow. *ACM Trans. Graph.*, 32, 2013.
- [10] J. Driscoll and D. Healy. Computing fourier transforms and convolutions on the 2-sphere. *Advances in Applied Mathematics*, 15(2):202–250, 1994.

BIBLIOGRAPHY

- [11] C. J. Du, P. T. Hawkins, L. R. Stephens, and T. Bretschneider. 3D time series analysis of cell shape using Laplacian approaches. *BMC Bioinformatics*, 14:296, Oct 2013.
- [12] A. C. Dufour, T.-Y. Liu, C. Ducroz, R. Tournemenne, B. Cummings, R. Thibeaux, N. Guillen, A. O. Hero, and J.-C. Olivo-Marin. Signal Processing Challenges in Quantitative 3-D Cell Morphology: More than meets the eye. *IEEE Signal Processing Magazine*, 32(1):30 – 40, Dec. 2014.
- [13] I. Friedel, P. Schröder, and M. Desbrun. Unconstrained Spherical Parameterization, 2007.
- [14] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. Fernández del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. *Nature*, 585:357–362, 2020.
- [15] E. Hinton, T. Rock, and O. C. Zienkiewicz. A note on mass lumping and related processes in the finite element method. *Earthquake Engineering & Structural Dynamics*, 4(3):245–249, 1976.
- [16] A. Jacobson, D. Panozzo, et al. libigl: A simple C++ geometry processing library, 2018. <https://libigl.github.io/>.
- [17] K. A. Johnson, N. C. Fox, R. A. Sperling, and W. E. Klunk. Brain imaging in alzheimer disease. *Cold Spring Harbor Perspectives in Medicine*, 2(4), 2012.
- [18] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation Invariant Spherical Harmonic Representation of 3D Shape Descriptors. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, SGP 03, pages 156–164, Goslar, DEU, 2003. Eurographics Association.
- [19] M. Kazhdan, J. Solomon, and M. Ben-Chen. Can mean-curvature flow be made non-singular?, 2012.
- [20] K. Khairy and J. Howard. Minimum-energy vesicle and cell shapes calculated using spherical harmonics parameterization. *Soft Matter*, 7:2138–2143, 2011.
- [21] R. L. and G. P. Polarity establishment, asymmetric division and segregation of fate determinants in early *C. elegans* embryos. WormBook, <http://www.wormbook.org>, 2014.
- [22] MATLAB. *version 7.10.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts, 2010.

BIBLIOGRAPHY

- [23] E. H. Najarro, L. Wong, M. Zhen, E. P. Carpio, A. Goncharov, G. Garriga, E. A. Lundquist, Y. Jin, and B. D. Ackley. *Caenorhabditis elegans* flamingo cadherin fmi-1 regulates gabaergic neuronal development. *Journal of Neuroscience*, 32(12):4196–4211, 2012.
- [24] C. Pohl and Z. Bao. Chiral forces organize left-right patterning in *C. elegans* by uncoupling midline and anteroposterior axis. *Developmental Cell*, 19(3):402–412, 2010.
- [25] M. Reuter, S. Biasotti, D. Giorgi, G. Patané, and M. Spagnuolo. Discrete laplace-beltrami operators for shape analysis and segmentation. *Computers & Graphics*, 33(3):381–390, 2009. IEEE International Conference on Shape Modelling and Applications 2009.
- [26] K. H. C. Sawa H. Wnt signaling in *C. elegans*. WormBook, <http://www.wormbook.org>, 2013.
- [27] L. Shen, H. Farid, and M. A. McPeek. Modeling three-dimensional morphological structures using spherical harmonics. *Evolution*, 63(4):1003–1016, Apr 2009.
- [28] L. Shen, J. Ford, F. Makedon, and A. Saykin. A surface-based approach for classification of 3d neuroanatomic structures. *Intell. Data Anal.*, 8:519–542, 12 2004.
- [29] L. Shen and F. Makedon. Spherical mapping for processing of 3d closed surfaces. *Image Vis. Comput.*, 24:743–761, 2006.
- [30] J. R. Shewchuk. What is a good linear finite element? - interpolation, conditioning, anisotropy, and quality measures. Technical report, In Proc. of the 11th International Meshing Roundtable, 2002.
- [31] H. Strutt and D. Strutt. Differential stability of flamingo protein complexes underlies the establishment of planar polarity. *Current Biology*, 18(20):1555–1564, 2008.
- [32] W. Thiels. Automated Cell Shape Retrieval From 4D Microscopy Images of *C. elegans* Embryos. Master’s thesis, KU Leuven, sep 2019.
- [33] T. Usui, Y. Shima, Y. Shimada, S. Hirano, R. W. Burgess, T. L. Schwarz, M. Takeichi, and T. Uemura. Flamingo, a seven-pass transmembrane cadherin, regulates planar cell polarity under the control of frizzled. *Cell*, 98(5):585–595, 1999.
- [34] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0

BIBLIOGRAPHY

- Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.
- [35] M. Webster and H. D. Sheets. A practical introduction to landmark-based geometric morphometrics. *The Paleontological Society Papers*, 16:163–188, 2010.
 - [36] F. Xing and L. Yang. Robust nucleus/cell detection and segmentation in digital pathology and microscopy images: A comprehensive review. *IEEE Reviews in Biomedical Engineering*, 9:234–263, 2016.
 - [37] M. Xu, Y. Wu, H. Shroff, M. Wu, and M. Mani. A scheme for 3-dimensional morphological reconstruction and force inference in the early *c. elegans* embryo. *PLOS ONE*, 13(7):1–20, 07 2018.

Popularized summary

Understanding how embryos develop remains a fundamental challenge in biology. To gain insight into this development, a roundworm called *C. elegans* is often used. It is about 1mm in size, fully transparent and easy to cultivate. Another interesting feature of the worm is that its development is invariant: every embryo develops in exactly the same way, resulting in adult worms that have the same fixed number of cells. All these features make the worm an ideal organism to study how animals develop.

In the medical world, shape is already used in many ways. For example, Alzheimer disease can be diagnosed by taking a brain scan and looking at the shape of brain structures. In cells, shape can reflect forces between cells, the state of a cell, and cell-to-cell communication.

In this work, the shape of cells is analyzed using the spherical harmonics decomposition. This can be understood as splitting up the shape into a number of simple components. When all these components are added, the original shape can be recovered. The more of these components are used, the more detail of the shape is preserved.

As the name implies, this decomposition can only be calculated relative to a sphere. To be able to use this method on cells, an algorithm is described that smooths out the surface until it becomes a sphere. Based on these spherical harmonics, a new way to detect ridges and protrusions is derived. These methods have not been used to analyze *C. elegans* before and here they are implemented in a very efficient way.

These methods are applied to digital representations of *C. elegans* embryos, obtained by imaging the embryos at the stage where they only have 7 cells. The wild-type worm is compared to two types that differ from it in specific ways. In the first one, two genes called *dsh-1* and *mig-5* are suppressed using a technique known as RNA interference. The other carries a mutation in a gene called *fmi-1*. These genes are all known to be part of important mechanisms in the embryonic development of the worm. Using the new techniques, differences are found in the shape of specific cells in these embryos.

Appendices

Finite element method discretization

The finite element method (FEM) is a numerical method to transform partial differential equations into systems of equations that can be solved using linear algebra. To do this we discretize space into many small finite elements. In this case, these elements will be the triangles making up the mesh.

Basis functions

In FEM, functions on the surface are represented by linear combinations of basis functions. For a triangle mesh, the most natural basis functions are the piecewise linear hat functions ϕ_i (fig. 1). They are equal to one on their associated vertex v_i , zero on every other vertex and linearly interpolated in each triangle around v_i . We will approximate a function $u : \mathcal{M} \rightarrow \mathbb{R}$ with these elements as:

$$u \approx \sum_{i \in V} x_i \phi_i, \quad x_i \in \mathbb{R} \tag{1}$$

Where V is the set of vertices and x is a vector of coefficients. To find suitable coefficients we can minimize the difference between u and $\sum x_i \phi_i$ with, for example, least squares.

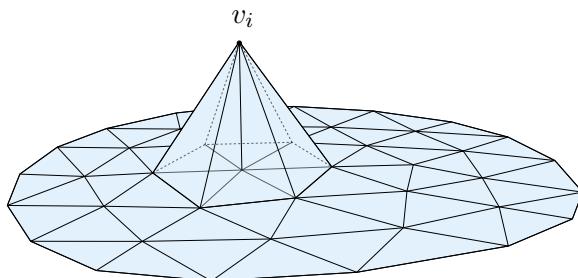


FIGURE 1: Hat function ϕ_i with associated vertex v_i .

Weak formulation

To discretize the Laplacian, it is instructive to start with a Poisson problem: [8]

$$\Delta u = f \quad (2)$$

To discretize this problem we use the so-called weak formulation: the equation should not hold absolutely, but only with respect to the basis functions. To do this we first introduce the inner product $\langle \cdot, \cdot \rangle$ for two scalar functions f, g on a surface \mathcal{M} :

$$\langle f, g \rangle = \int_{\mathcal{M}} fg dS \quad (3)$$

If the functions are vector valued, the usual dot product in \mathbb{R}^3 is used:

$$\langle \mathbf{p}, \mathbf{q} \rangle = \int_{\mathcal{M}} \mathbf{p} \cdot \mathbf{q} dS \quad (4)$$

The weak formulation of eq. (2) is then:

$$\langle \Delta u, \phi_j \rangle = \langle f, \phi_j \rangle \quad (5)$$

Which should hold for every ϕ_j . For the left-hand side we can apply Green's first identity:

$$\langle \Delta u, \phi_j \rangle = - \langle \nabla u, \nabla \phi_j \rangle + \oint_{\partial \mathcal{M}} (\mathbf{n} \cdot \nabla u) \phi_j ds \quad (6)$$

where the last term is a contour integral over the boundary of \mathcal{M} , where \mathbf{n} is the boundary normal. We can drop this term since we only consider closed surfaces, which have no boundary by definition. Using the approximation from eq. (1):

$$- \langle \nabla u, \nabla \phi_j \rangle = - \left\langle \nabla \sum_i x_i \phi_i, \nabla \phi_j \right\rangle = - \sum_i x_i \langle \nabla \phi_i, \nabla \phi_j \rangle \quad (7)$$

For the right-hand side of eq. (5) we use the same approximation with coefficients b_i and find:

$$\begin{aligned} f &\approx \sum_i b_i \phi_i \\ \langle f, \phi_j \rangle &= \left\langle \sum_i b_i \phi_i, \phi_j \right\rangle = \sum_i b_i \langle \phi_i, \phi_j \rangle \end{aligned} \quad (8)$$

The problem can now be stated in a matrix formulation:

$$\begin{aligned} L_{ij} &= - \langle \nabla \phi_i, \nabla \phi_j \rangle \\ M_{ij} &= \langle \phi_i, \phi_j \rangle \\ \mathbf{x} &= [x_1, \dots, x_n]^T \\ \mathbf{b} &= [b_1, \dots, b_n]^T \end{aligned} \quad (9)$$

with n the number of vertices. Finally, we can rephrase eq. (5) as:

$$-\sum_i x_i \langle \nabla \phi_i, \nabla \phi_j \rangle = \sum_i b_i \langle \phi_i, \phi_j \rangle \quad (10)$$

$$\Downarrow \\ \mathbf{Lx} = \mathbf{Mb} \quad (11)$$

$$\Downarrow \\ \mathbf{M}^{-1} \mathbf{Lx} = \mathbf{b} \quad (12)$$

Comparing to eq. (2), the discretized version of the Laplacian Δ is thus the matrix $\mathbf{M}^{-1} \mathbf{L}$.

The matrix \mathbf{L} is known as the stiffness matrix and using the hat basis it is equal to:

$$L_{ij} = \begin{cases} -\sum_{k \sim i} \frac{1}{2} (\cot \alpha_{ik} + \cot \beta_{ik}), & i = j \\ \frac{1}{2} (\cot \alpha_{ij} + \cot \beta_{ij}), & i \sim j \\ 0, & \text{otherwise} \end{cases} \quad (13)$$

where $i \sim j$ means vertices that are connected by an edge. α_{ij} and β_{ij} are the two angles opposite to an edge (i, j) [25].

The matrix \mathbf{M} is known as the mass matrix:

$$M_{ij} = \begin{cases} \frac{1}{6} \sum_{k \sim i} |T_{ik}|, & i = j \\ \frac{1}{12} (|T_1| + |T_2|), & i \sim j \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

where $|T_{ik}|$ are the areas of triangles around vertex i and $|T_1|$ and $|T_2|$ are the areas of the triangles sharing an edge (i, j) [25].

Inner product

Given two functions $f, g : \mathcal{M} \rightarrow \mathbb{R}$ and using the same approximation as above, with coefficient vectors b and c :

$$\begin{aligned} f &\approx \sum_i b_i \phi_i \\ g &\approx \sum_i c_i \phi_i \end{aligned} \quad (15)$$

To get their inner product

$$\begin{aligned} \langle f, g \rangle &\approx \sum_i \sum_j \langle b_i \phi_i, c_j \phi_j \rangle \\ &= \sum_i b_i \sum_j \langle \phi_i, \phi_j \rangle c_j \\ &= \sum_i b_i \sum_j M_{ij} c_j \\ &= b^T M c \end{aligned} \quad (16)$$

Lumped mass matrix

By using piecewise constant, non-overlapping basis functions instead, the mass matrix must be zero except on the diagonal [15]

$$\langle \phi_i, \phi_j \rangle = 0, \quad i \neq j \quad (17)$$

So that we find the diagonal lumped mass matrix:

$$M_{ij}^{(lumped)} = \begin{cases} \frac{1}{3} \sum_{k \sim i} |T_{ik}|, & i = j \\ 0, & \text{otherwise} \end{cases} \quad (18)$$

Where $\frac{1}{3} \sum_{k \sim i} |T_{ik}|$ are the vertex areas.