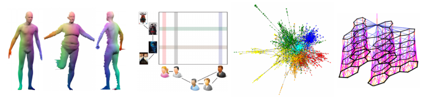
Analysis of 3D Shapes (IN2238)

Frank R. Schmidt
Matthias Vestner

Summer Semester 2017

Geometric deep learning on graphs and manifolds

Going beyond Euclidean data



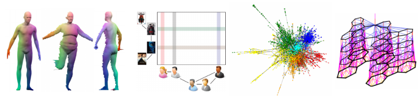
June 23rd, 14-17
June 30th, 14-17
MI HS 3

Michael Bronstein

- Associate Professor USI Lugano
- Associate Professor Tel Aviv University
- Principal Engineer at the Intel Perceptual Computing Group
- authored the first book on non-rigid shape analysis
- www.inf.usi.ch/bronstein/

Geometric deep learning on graphs and manifolds

Going beyond Euclidean data

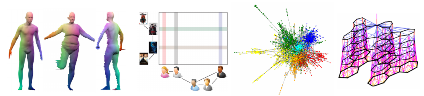


June 23rd, 14-17
June 30th, 14-17
MI HS 3

In the past decade, deep learning methods have achieved unprecedented performance on a broad range of problems in various fields from computer vision to speech recognition. However, so far research has mainly focused on developing deep learning methods for Euclidean-structured data. However, many important applications have to deal with non-Euclidean structured data, such as graphs and manifolds. Such geometric data are becoming increasingly important in computer graphics and 3D vision, sensor networks, drug design, biomedicine, recommendation systems, and web applications. The adoption of deep learning in these fields has been lagging behind until recently, primarily since the non-Euclidean nature of objects dealt with makes the very definition of basic operations used in deep networks rather elusive. The purpose of the proposed tutorial is to introduce the emerging field of geometric deep learning on graphs and manifolds, overview existing solutions and applications for this class of problems, as well as key difficulties and future research directions.

Geometric deep learning on graphs and manifolds

Going beyond Euclidean data



June 23rd, 14-17
June 30th, 14-17
MI HS 3

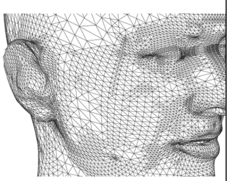
1. Introduction
2. Foundations of deep learning
3. Geometry of manifolds and graphs
4. Spectral domain geometric deep learning
5. Spatial domain geometric deep learning
6. Spatio-frequency geometric deep learning
7. Applications

8. Triangle Meshes, PL Functions

Triangle Meshes

Test A triangle mesh $\mathcal{M}$ is a pair $(\mathcal{V}, \mathcal{F})$ with

- $\mathcal{V} = \{v_1, \dots, v_V\}$ (vertices)
- $\mathcal{F} = \{f_1, \dots, f_F\}, \quad f_i \in \mathcal{V} \times \mathcal{V} \times \mathcal{V}$ (triangular faces)

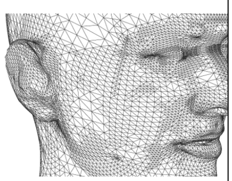


Test A triangle mesh $\mathcal{M}$ is a pair $(\mathcal{V}, \mathcal{F})$ with

- $\mathcal{V} = \{v_1, \dots, v_V\}$ (vertices)
- $\mathcal{F} = \{f_1, \dots, f_F\}, \quad f_i \in \mathcal{V} \times \mathcal{V} \times \mathcal{V}$ (triangular faces)

Implicitely given

- $\mathcal{E} = \{e_1, \dots, e_E\}, \quad e_i \in \mathcal{V} \times \mathcal{V}$ (edges)



Test A triangle mesh $\mathcal{M}$ is a pair $(\mathcal{V}, \mathcal{F})$ with

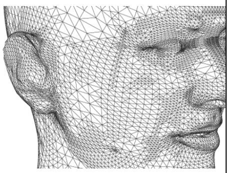
- $\mathcal{V} = \{v_1, \ldots, v_V\}$ (vertices)
- $\mathcal{F} = \{f_1, \ldots, f_F\}, \quad f_i \in \mathcal{V} \times \mathcal{V} \times \mathcal{V}$ (triangular faces)

Implicitely given

- $\mathcal{E} = \{e_1, \ldots, e_E\}, \quad e_i \in \mathcal{V} \times \mathcal{V}$ (edges)

Geometric embedding

- $\mathcal{P} = \{p_1, \ldots, p_V\},\ p_i := p(v_i) = \begin{pmatrix} x(v_i) \\ y(v_i) \\ z(v_i) \end{pmatrix}$



Test A triangle mesh $\mathcal{M}$ is a pair $(\mathcal{V}, \mathcal{F})$ with

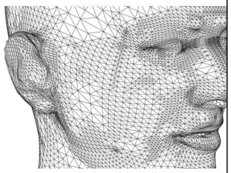
- $\mathcal{V} = \{v_1, \ldots, v_V\}$ (vertices)
- $\mathcal{F} = \{f_1, \ldots, f_F\}, \quad f_i \in \mathcal{V} \times \mathcal{V} \times \mathcal{V}$ (triangular faces)

Implicitely given

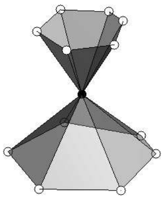
- $\mathcal{E} = \{e_1, \ldots, e_E\}, \quad e_i \in \mathcal{V} \times \mathcal{V}$ (edges)

Geometric embedding

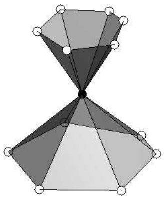
- $\mathcal{P} = \{p_1, \ldots, p_V\},\ p_i := p(v_i) = \begin{pmatrix} x(v_i) \\ y(v_i) \\ z(v_i) \end{pmatrix}$



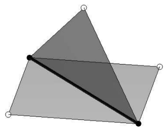
We will not be strict about the distinction of v_i's and p_i's.



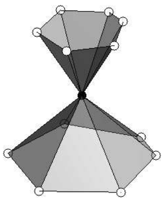
Non-manifold vertex



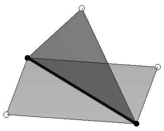
Non-manifold vertex



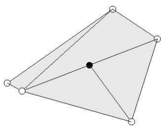
Non-manifold edge



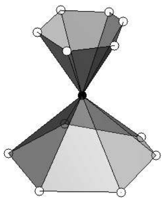
Non-manifold vertex



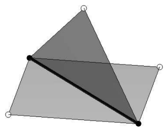
Non-manifold edge



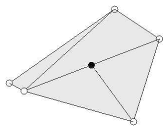
hanging vertex



Non-manifold vertex



Non-manifold edge



hanging vertex

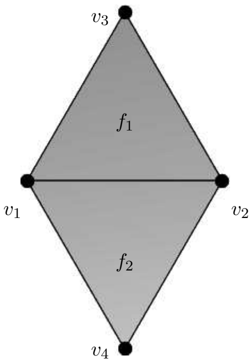
self intersection

Vertex List

0	0	0
1	0	0
.5	.866	0
.5	-.866	0

Triangle List

1	2	3
1	4	2

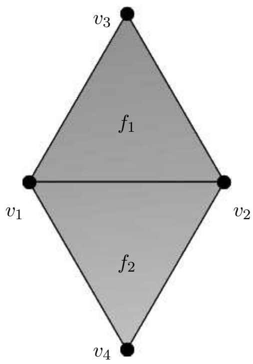


Vertex List

0	0	0
1	0	0
.5	.866	0
.5	-.866	0

Triangle List

1	2	3
1	4	2



Cyclic shifts do not change the mesh

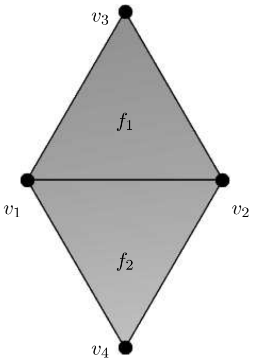
Example

\# Vertex List

0	0	0
1	0	0
.5	.866	0
.5	-.866	0

\# Triangle List

1	2	3
4	2	1



Cyclic shifts do not change the mesh

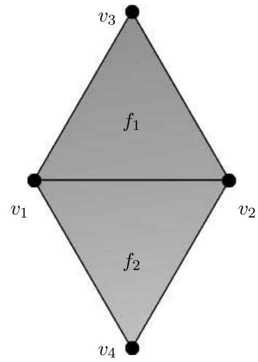
Example

\# Vertex List

0	0	0
1	0	0
.5	.866	0
.5	-.866	0

\# Triangle List

1	2	3
4	1	2

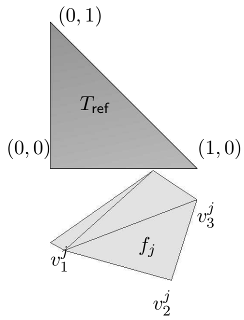


Triangles have to be consistently oriented

Triangle as surface patch

Each triangle $f_j = (v_1^j, v_2^j, v_3^j)$ comes with the coordinate map (x_j, T_{ref})

$$x_j(u) = v_1^j + u_1(v_2^j - v_1^j) + u_2(v_3^j - v_1^j)$$



Triangle as surface patch

Each triangle $f_j = (v_1^j, v_2^j, v_3^j)$ comes with the coordinate map (x_j, T_{ref})

$$x_j(u) = v_1^j + u_1(v_2^j - v_1^j) + u_2(v_3^j - v_1^j)$$

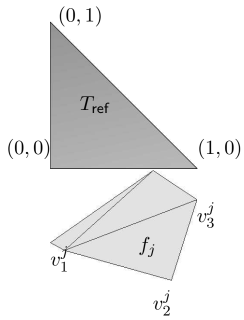
- each coordinate map is affine

Triangle as surface patch

Each triangle $f_j = (v_1^j, v_2^j, v_3^j)$ comes with the coordinate map (x_j, T_{ref})

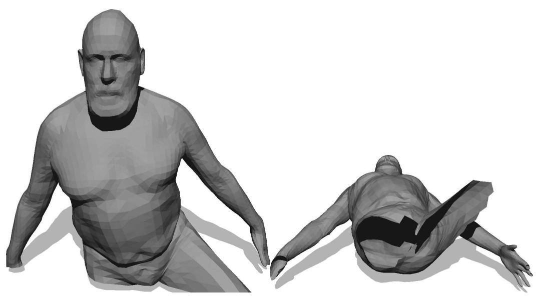
$$x_j(u) = v_1^j + u_1(v_2^j - v_1^j) + u_2(v_3^j - v_1^j)$$

- each coordinate map is affine
- constant differential Dx_j



Watertight Mesh

We will henceforth only consider meshes that are closed (watertight). This means they do not have a boundary and thus every edge is adjacent to exactly two faces.



Euler Formula

The famous Euler formula states an interesting relation between the number of verticec V, edges E and faces F in a closed and connected mesh:

$$V - E + F = 2(1 - g)$$

$g = 0$ $g = 1$ $g = 2$

Euler Formula

The famous Euler formula states an interesting relation between the number of verticec V, edges E and faces F in a closed and connected mesh:

$$V - E + F = 2(1 - g)$$

For most practical applications the genus g is small compared to the number of vertices, faces and edges.

- each triangle is bounded by three edges

Euler Formula

The famous Euler formula states an interesting relation between the number of verticec V, edges E and faces F in a closed and connected mesh:

$$V - E + F = 2(1 - g)$$

For most practical applications the genus g is small compared to the number of vertices, faces and edges.

- each triangle is bounded by three edges
- each edge is incident to two triangles

Euler Formula

The famous Euler formula states an interesting relation between the number of verticec V, edges E and faces F in a closed and connected mesh:

$$V - E + F = 2(1 - g)$$

For most practical applications the genus g is small compared to the number of vertices, faces and edges.

- each triangle is bounded by three edges
- each edge is incident to two triangles

$\Rightarrow$ Number of triangles is approx. twice the number of faces $F \approx 2V$

Euler Formula

The famous Euler formula states an interesting relation between the number of verticec V, edges E and faces F in a closed and connected mesh:

$$V - E + F = 2(1 - g)$$

For most practical applications the genus g is small compared to the number of vertices, faces and edges.

- each triangle is bounded by three edges
- each edge is incident to two triangles

$\Rightarrow$ Number of triangles is approx. twice the number of faces $F \approx 2V$
$\Rightarrow$ Number of edges is approx. three times the number of vertices $E \approx 3V$

Euler Formula

The famous Euler formula states an interesting relation between the number of verticec V, edges E and faces F in a closed and connected mesh:

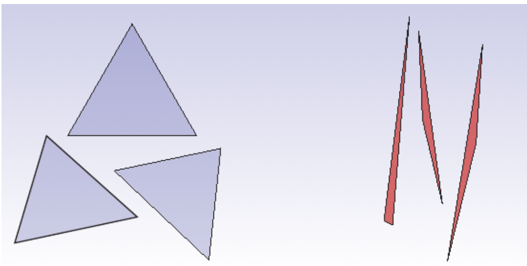
$$V - E + F = 2(1 - g)$$

For most practical applications the genus g is small compared to the number of vertices, faces and edges.

- each triangle is bounded by three edges
- each edge is incident to two triangles

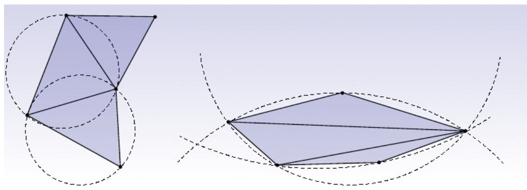
$\Rightarrow$ Number of triangles is approx. twice the number of faces $F \approx 2V$
$\Rightarrow$ Number of edges is approx. three times the number of vertices $E \approx 3V$
$\Rightarrow$ The average vertex valence (number of incident edges) is 6

Triangle quality - Intuition



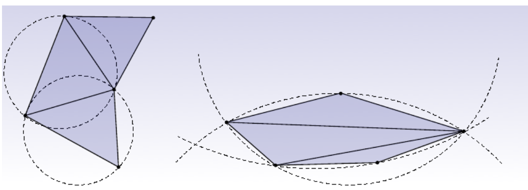
good triangles bad triangles

Triangle quality - Delaunay Property



Delaunay Not Delaunay

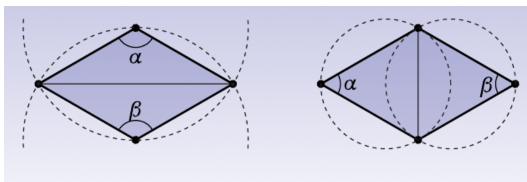
Triangle quality - Delaunay Property



Delaunay Not Delaunay

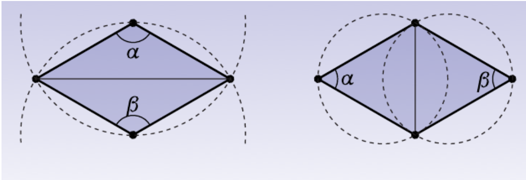
- *Delaunay:* no point is allowed to be inside the any triangles' circumcircle
- satisfied if no angle is obtuse ($> \frac{\pi}{2}$)

Local Mesh Improvement



- Some simple ways wo improve mesh quality
- e.g.: if $\alpha + \beta > \pi$ flip the edge; after enough flips, mesh will be Delaunay

Local Mesh Improvement

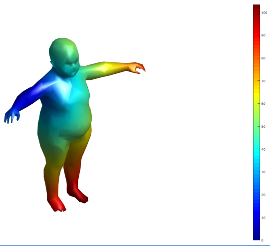


- Some simple ways wo improve mesh quality
- e.g.: if $\alpha + \beta > \pi$ flip the edge; after enough flips, mesh will be Delaunay
- other ways to improve mesh (edge collapse, edge split, ...)
- We assume our meshes are nice

PL Functions

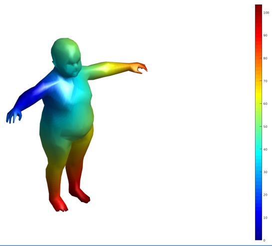
Functions on Manifolds

Let $\mathcal{S}$ be a manifold. A function $f : \mathcal{S} \supset V \to \mathbb{R}$ is **differentiable** at $p \in V$ if for some parametrization $x : \mathbb{R}^2 \subset U \to V$ the composition $f \circ x : U \to \mathbb{R}$ is differentiable.



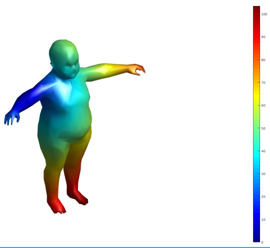
Functions on Manifolds

Let $\mathcal{S}$ be a manifold. A function $f : \mathcal{S} \supset V \to \mathbb{R}$ is **differentiable** at $p \in V$ if for some parametrization $x : \mathbb{R}^2 \subset U \to V$ the composition $f \circ x : U \to \mathbb{R}$ is differentiable.
This definition does **not depend** on a specific parametrization.



Functions on Manifolds

Let $\mathcal{S}$ be a manifold. A function $f : \mathcal{S} \supset V \to \mathbb{R}$ is **differentiable** at $p \in V$ if for some parametrization $x : \mathbb{R}^2 \subset U \to V$ the composition $f \circ x : U \to \mathbb{R}$ is differentiable.
This definition does **not depend** on a specific parametrization.
We will soon introduce **differential operators** such as gradient (∇) and Laplace-Beltrami-Operator (Δ) that can be applied to differentiable functions.



Discretizing functions

As we discretized two dimensional manifolds by a finite number V of vertices and F of triangles we are also only able to store a finite number of values to represent a function defined on a triangle mesh.

Discretizing functions

As we discretized two dimensional manifolds by a finite number V of vertices and F of triangles we are also only able to store a finite number of values to represent a function defined on a triangle mesh.
Given a function $f : \mathcal{M} \to \mathbb{R}$ defined on a triangle mesh $\mathcal{M}$, a standard way to discretize it is to only store its values at the vertices:

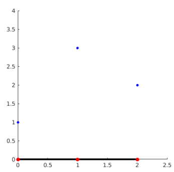
$$\begin{pmatrix} f(v_1) & \dots & f(v_V) \end{pmatrix}^T = \begin{pmatrix} \mathbf{f}_1 & \dots & \mathbf{f}_V \end{pmatrix}^T = \mathbf{f} \in \mathbb{R}^V$$

Discretizing functions

As we discretized two dimensional manifolds by a finite number V of vertices and F of triangles we are also only able to store a finite number of values to represent a function defined on a triangle mesh.
Given a function $f : \mathcal{M} \to \mathbb{R}$ defined on a triangle mesh $\mathcal{M}$, a standard way to discretize it is to only store its values at the vertices:

$$\begin{pmatrix} f(v_1) & \dots & f(v_V) \end{pmatrix}^T = \begin{pmatrix} \mathbf{f}_1 & \dots & \mathbf{f}_V \end{pmatrix}^T = \mathbf{f} \in \mathbb{R}^V$$

However many different functions f may have the same discretization.

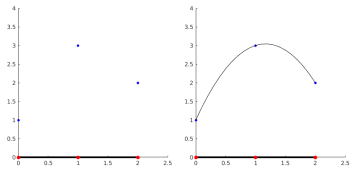


Discretizing functions

As we discretized two dimensional manifolds by a finite number V of vertices and F of triangles we are also only able to store a finite number of values to represent a function defined on a triangle mesh.
Given a function $f : \mathcal{M} \to \mathbb{R}$ defined on a triangle mesh $\mathcal{M}$, a standard way to discretize it is to only store its values at the vertices:

$$\begin{pmatrix} f(v_1) & \dots & f(v_V) \end{pmatrix}^T = \begin{pmatrix} \mathbf{f}_1 & \dots & \mathbf{f}_V \end{pmatrix}^T = \mathbf{f} \in \mathbb{R}^V$$

However many different functions f may have the same discretization.

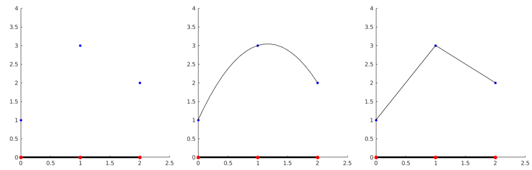


Discretizing functions

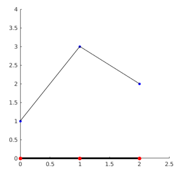
As we discretized two dimensional manifolds by a finite number V of vertices and F of triangles we are also only able to store a finite number of values to represent a function defined on a triangle mesh.
Given a function $f : \mathcal{M} \to \mathbb{R}$ defined on a triangle mesh $\mathcal{M}$, a standard way to discretize it is to only store its values at the vertices:

$$\begin{pmatrix} f(v_1) & \dots & f(v_V) \end{pmatrix}^T = \begin{pmatrix} \mathbf{f}_1 & \dots & \mathbf{f}_V \end{pmatrix}^T = \mathbf{f} \in \mathbb{R}^V$$

However many different functions f may have the same discretization.

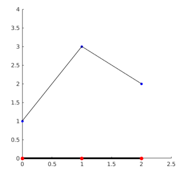


PL functions



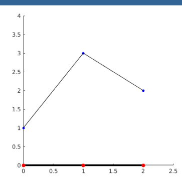
We will henceforth interpret vectors $\mathbf{f} \in \mathbb{R}^V$ as samplings of piecewise linear (PL) functions.

PL functions



We will henceforth interpret vectors $\mathbf{f} \in \mathbb{R}^V$ as samplings of piecewise linear (PL) functions. The space of PL functions is a vector space. Adding (scaling) the representing vectors is equivalent to adding (scaling) the represented functions.

PL functions



We will henceforth interpret vectors $\mathbf{f} \in \mathbb{R}^V$ as samplings of piecewise linear (PL) functions. The space of PL functions is a vector space. Adding (scaling) the representing vectors is equivalent to adding (scaling) the represented functions.
Notice: PL functions are not (classically) differentiable (at the vertices).

Hat functions

We have seen that the space of PL functions is a V-dimensional vectorspace (of functions).

Hat functions

We have seen that the space of PL functions is a V-dimensional vectorspace (of functions). We should be able to find V basis functions $\{\psi_1, \dots \psi_V\}$ that span this space.

Hat functions

We have seen that the space of PL functions is a V-dimensional vectorspace (of functions). We should be able to find V basis functions $\{\psi_1, \dots \psi_V\}$ that span this space. A standard choice are the so called **hat functions**, defined via

$$\psi_i(v_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

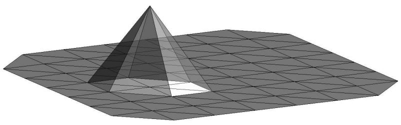
and their property of being PL functions (*i.e.*linear inside each triangle).

Hat functions

We have seen that the space of PL functions is a V-dimensional vectorspace (of functions). We should be able to find V basis functions $\{\psi_1, \dots \psi_V\}$ that span this space. A standard choice are the so called **hat functions**, defined via

$$\psi_i(v_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

and their property of being PL functions (*i.e.*linear inside each triangle).



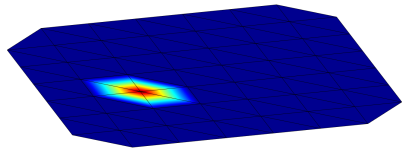
Triangle Meshes PL Functions

Hat functions

We have seen that the space of PL functions is a V-dimensional vectorspace (of functions). We should be able to find V basis functions $\{\psi_1, \dots \psi_V\}$ that span this space. A standard choice are the so called **hat functions**, defined via

$$\psi_i(v_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

and their property of being PL functions (*i.e.*linear inside each triangle).



IN2238 - Analysis of Three-Dimensional Shapes 8. Triangle Meshes, PL functions – 18 / 27

Triangle Meshes PL Functions

Hat functions

We have seen that the space of PL functions is a V-dimensional vectorspace (of functions). We should be able to find V basis functions $\{\psi_1, \dots \psi_V\}$ that span this space. A standard choice are the so called **hat functions**, defined via

$$\psi_i(v_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

and their property of being PL functions (*i.e.*linear inside each triangle).

Sampled values coincide with coefficients in this basis:

$$f(x) = \sum_{i=1}^{V} \alpha_i \psi_i(x)$$

IN2238 - Analysis of Three-Dimensional Shapes 8. Triangle Meshes, PL functions – 18 / 27

Triangle Meshes PL Functions

Hat functions

We have seen that the space of PL functions is a V-dimensional vectorspace (of functions). We should be able to find V basis functions $\{\psi_1, \dots \psi_V\}$ that span this space. A standard choice are the so called **hat functions**, defined via

$$\psi_i(v_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

and their property of being PL functions (*i.e.*linear inside each triangle).

Sampled values coincide with coefficients in this basis:

$$f(x) = \sum_{i=1}^{V} \alpha_i \psi_i(x) = \sum_{i=1}^{V} f(v_i) \psi_i(x) = \sum_{i=1}^{V} \mathbf{f}_i \psi_i(x)$$

IN2238 - Analysis of Three-Dimensional Shapes 8. Triangle Meshes, PL functions – 18 / 27

Triangle Meshes PL Functions

Hat functions

We have seen that the space of PL functions is a V-dimensional vectorspace (of functions). We should be able to find V basis functions $\{\psi_1, \dots \psi_V\}$ that span this space. A standard choice are the so called **hat functions**, defined via

$$\psi_i(v_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

and their property of being PL functions (*i.e.*linear inside each triangle).

- easy interpretation

IN2238 - Analysis of Three-Dimensional Shapes 8. Triangle Meshes, PL functions – 18 / 27

Triangle Meshes PL Functions

Hat functions

We have seen that the space of PL functions is a V-dimensional vectorspace (of functions). We should be able to find V basis functions $\{\psi_1, \dots \psi_V\}$ that span this space. A standard choice are the so called **hat functions**, defined via

$$\psi_i(v_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

and their property of being PL functions (*i.e.*linear inside each triangle).

- easy interpretation
- localized support (points where the function $\neq 0$) $\Rightarrow$ will lead to sparse matrices

IN2238 - Analysis of Three-Dimensional Shapes 8. Triangle Meshes, PL functions – 18 / 27

Triangle Meshes PL Functions

Hat functions

We have seen that the space of PL functions is a V-dimensional vectorspace (of functions). We should be able to find V basis functions $\{\psi_1, \dots \psi_V\}$ that span this space. A standard choice are the so called **hat functions**, defined via

$$\psi_i(v_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

and their property of being PL functions (*i.e.*linear inside each triangle).

- easy interpretation
- localized support (points where the function $\neq 0$) $\Rightarrow$ will lead to sparse matrices
- not orthogonal

IN2238 - Analysis of Three-Dimensional Shapes 8. Triangle Meshes, PL functions – 18 / 27

Triangle Meshes PL Functions

Scalar product

Definition. Let X be a vector space. A **scalar (or inner) product** $\langle \cdot, \cdot \rangle$ is a mapping $X \times X \to \mathbb{R}$ satisfying

1. Symmetry:

$$\langle x, y \rangle = \langle y, x \rangle \quad \forall\, x, y \in X$$

2. Linearity:

$$\langle \alpha x, y \rangle = \alpha \langle x, y \rangle \quad \forall x, y \in X, \alpha \in \mathbb{R}$$
$$\langle x + z, y \rangle = \langle x, y \rangle + \langle z, y \rangle \quad \forall x, y, z \in X$$

3. Positive definitness:

$$\langle x, x \rangle \geqslant 0$$
$$\langle x, x \rangle = 0 \Leftrightarrow x = 0$$

Scalar product - examples 1

Example 1. Every symmetric positive definit (spd) matrix $A \in \mathbb{R}^{n\times n}$ defines an inner product on the vectorspace $X = \mathbb{R}^n$ via

$$\langle x, y\rangle_A := x^T Ay = \langle x, Ay\rangle$$

Scalar product - examples 1

Example 1. Every symmetric positive definit (spd) matrix $A \in \mathbb{R}^{n\times n}$ defines an inner product on the vectorspace $X = \mathbb{R}^n$ via

$$\langle x, y\rangle_A := x^T Ay = \langle x, Ay\rangle$$

Example 2. For $X = C^0(\mathbb{S}^1)$ (a vector space of functions) we can define

$$\begin{aligned}\langle f, g\rangle_{L^2(\mathbb{S}^1)} &:= \int_{\mathbb{S}^1} f(p)g(p)dp\\ &= \int_0^{2\pi} f(\cos(t), \sin(t))g(\cos(t), \sin(t))\sqrt{\cos^2(t) + \sin^2(t)}dt\end{aligned}$$

Scalar product - examples 1

Example 1. Every symmetric positive definit (spd) matrix $A \in \mathbb{R}^{n\times n}$ defines an inner product on the vectorspace $X = \mathbb{R}^n$ via

$$\langle x, y\rangle_A := x^T Ay = \langle x, Ay\rangle$$

Example 2. For $X = C^0(\mathbb{S}^1)$ (a vector space of functions) we can define

$$\begin{aligned}\langle f, g\rangle_{L^2(\mathbb{S}^1)} &:= \int_{\mathbb{S}^1} f(p)g(p)dp\\ &= \int_0^{2\pi} f(\cos(t), \sin(t))g(\cos(t), \sin(t))\sqrt{\cos^2(t) + \sin^2(t)}dt\end{aligned}$$

Example 3. Let $\mathcal{M}$ be a manifold, then $X = C^0(\mathcal{M})$ can be equipped with the inner product

$$\langle f, g\rangle_{L^2(\mathcal{M})} := \int_{\mathcal{M}} f(p)g(p)dp = \ldots$$

Scalar product - examples 2

Are the following expressions inner products on $X = C^1((a, b))$?

Example 4.

$$\langle f, g\rangle = \int_a^b f'(t)g'(t)dt$$

Scalar product - examples 2

Are the following expressions inner products on $X = C^1((a, b))$?

Example 4.

$$\langle f, g\rangle = \int_a^b f'(t)g'(t)dt$$

Example 5.

$$\langle f, g\rangle = \int_a^b f(t)g(t)dt + \int_a^b f'(t)g'(t)dt$$

Inner product of PL functions

Let

$$f(p) = \sum_{i=1}^{V} \mathbf{f}_i\psi_i(p) \quad g(p) = \sum_{j=1}^{V} \mathbf{g}_j\psi_j(p)$$

be two PL functions defined on a discretized manifold $\mathcal{M}$.

Inner product of PL functions

Let

$$f(p) = \sum_{i=1}^{V} \mathbf{f}_i\psi_i(p) \quad g(p) = \sum_{j=1}^{V} \mathbf{g}_j\psi_j(p)$$

be two PL functions defined on a discretized manifold $\mathcal{M}$.
Making use of the linearity of integrals we observe:

$$\begin{aligned}\langle f, g\rangle_{L^2(\mathcal{M})} &= \int_{\mathcal{M}} f(p)g(p)dp\\ &= \int_{\mathcal{M}} \left(\sum_{i=1}^{V} \mathbf{f}_i\psi_i(p)\right)\left(\sum_{j=1}^{V} \mathbf{g}_j\psi_j(p)\right)dp\end{aligned}$$

Inner product of PL functions

Let

$$f(p) = \sum_{i=1}^{V} \mathbf{f}_i\psi_i(p) \quad g(p) = \sum_{j=1}^{V} \mathbf{g}_j\psi_j(p)$$

be two PL functions defined on a discretized manifold $\mathcal{M}$.
Making use of the linearity of integrals we observe:

$$\begin{aligned}\langle f, g\rangle_{L^2(\mathcal{M})} &= \int_{\mathcal{M}} f(p)g(p)dp\\ &= \int_{\mathcal{M}} \left(\sum_{i=1}^{V} \mathbf{f}_i\psi_i(p)\right)\left(\sum_{j=1}^{V} \mathbf{g}_j\psi_j(p)\right)dp\\ &= \sum_{i=1}^{V}\sum_{j=1}^{V} \mathbf{f}_i\mathbf{g}_j \int_{\mathcal{M}} \psi_i(p)\psi_j(p)dp\end{aligned}$$

Inner product of PL functions

Let

$$f(p)=\sum_{i=1}^{V}\mathbf{f}_i\psi_i(p)\quad g(p)=\sum_{j=1}^{V}\mathbf{g}_j\psi_j(p)$$

be two PL functions defined on a discretized manifold $\mathcal{M}$.
Making use of the linearity of integrals we observe:

$$\begin{aligned}\langle f,g\rangle_{L^2(\mathcal{M})}&=\int_{\mathcal{M}}f(p)g(p)dp\\&=\int_{\mathcal{M}}\left(\sum_{i=1}^{V}\mathbf{f}_i\psi_i(p)\right)\left(\sum_{j=1}^{V}\mathbf{g}_j\psi_j(p)\right)dp\\&=\sum_{i=1}^{V}\sum_{j=1}^{V}\mathbf{f}_i\mathbf{g}_j\underbrace{\int_{\mathcal{M}}\psi_i(p)\psi_j(p)dp}_{\mathbf{M}_{ij}}=\langle\mathbf{f},\mathbf{g}\rangle_{\mathbf{M}}\end{aligned}$$

Mass matrix

The matrix $\mathbf{M}$ with the entries $\mathbf{M}_{ij}=\int_{\mathcal{M}}\psi_i(p)\psi_j(p)$ is called **mass matrix**.

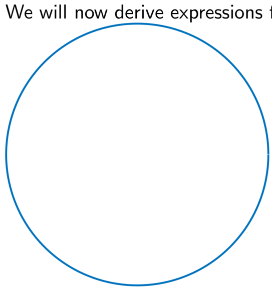
Mass matrix

The matrix $\mathbf{M}$ with the entries $\mathbf{M}_{ij}=\int_{\mathcal{M}}\psi_i(p)\psi_j(p)$ is called **mass matrix**.
It is a spd matrix and therefore induces an inner product on $\mathbb{R}^V$. Notice that this is not the standard inner product:

$$\begin{aligned}\langle f,g\rangle_{L^2(\mathcal{M})}&=\langle\mathbf{f},\mathbf{g}\rangle_{\mathbf{M}}\\&=\sum_{i=1}^{V}\sum_{j=1}^{V}\mathbf{f}_i\mathbf{g}_j\mathbf{M}_{ij}\\&\neq\sum_{i=1}^{V}\sum_{j=1}^{V}\mathbf{f}_i\mathbf{g}_j\\&=\langle\mathbf{f},\mathbf{g}\rangle\end{aligned}$$

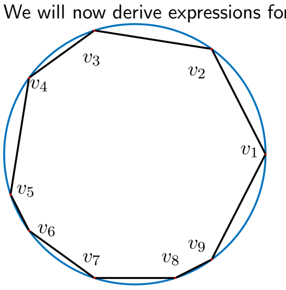
Mass matrix - 2D, 1/3

We will now derive expressions for the entries of the mass matrix (first in 2D).



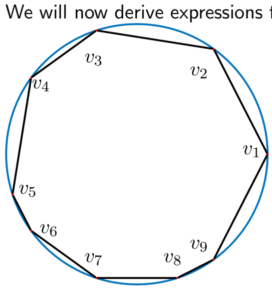
Mass matrix - 2D, 1/3

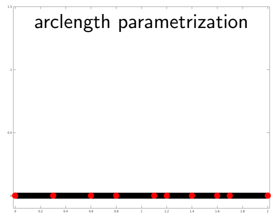
We will now derive expressions for the entries of the mass matrix (first in 2D).



Mass matrix - 2D, 1/3

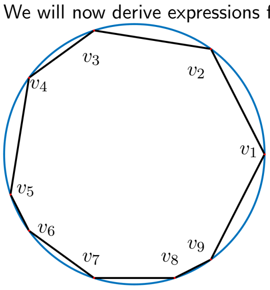
We will now derive expressions for the entries of the mass matrix (first in 2D).

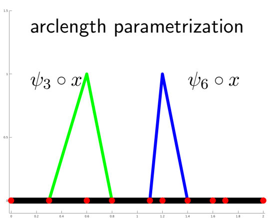




Mass matrix - 2D, 1/3

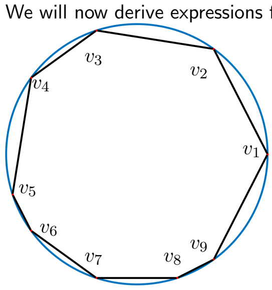
We will now derive expressions for the entries of the mass matrix (first in 2D).

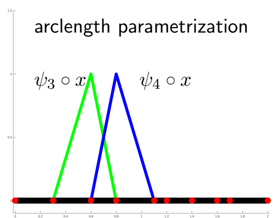




Mass matrix - 2D, 1/3

We will now derive expressions for the entries of the mass matrix (first in 2D).

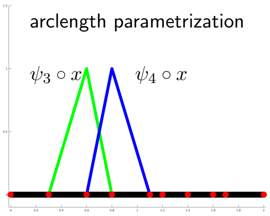




$$\mathbf{M}_{ij}=\int_{\mathcal{M}}\psi_i(p)\psi_j(p)dp=\begin{cases}0 & \text{if } (v_i,v_j)\notin\mathcal{E}\\ \int_{e_{ij}}\psi_i(p)\psi_j(p)dp & \text{if } (v_i,v_j)\in\mathcal{E}\\ \sum_{k\in\mathcal{N}(i)}\int_{e_{ik}}\psi_i^2(p)dp & \text{if } i=j\end{cases}$$

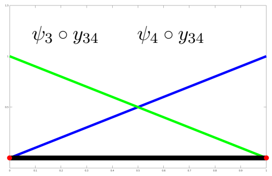
Mass matrix - 2D, 2/3

arclength parametrization



$$\int_{e_{ij}} \psi_i(p)\psi_j(p)dp = \int_{t_i}^{t_j} \psi_i(x(t))\psi_j(x(t)) \underbrace{\|\dot{x}(t)\|}_{=1} dt$$

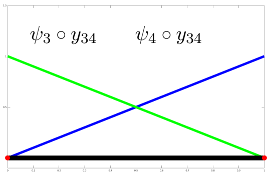
Mass matrix - 2D, 2/3



parametrization of $e_{ij} = (v_i, v_j)$ from reference interval $[0, 1]$
$y_{ij}(t) = (1 - t)v_i + tv_j$

$$\int_{e_{ij}} \psi_i(p)\psi_j(p)dp = \int_{t_i}^{t_j} \psi_i(x(t))\psi_j(x(t)) \underbrace{\|\dot{x}(t)\|}_{=1} dt$$
$$= \int_0^1 \psi_i(y_{ij}(t))\psi_j(y_{ij}(t))\|\dot{y}_{ij}(t)\|dt$$

Mass matrix - 2D, 2/3



parametrization of $e_{ij} = (v_i, v_j)$ from reference interval $[0, 1]$
$y_{ij}(t) = (1 - t)v_i + tv_j$

$$\int_{e_{ij}} \psi_i(p)\psi_j(p)dp = \int_{t_i}^{t_j} \psi_i(x(t))\psi_j(x(t)) \underbrace{\|\dot{x}(t)\|}_{=1} dt$$
$$= \int_0^1 \psi_i(y_{ij}(t))\psi_j(y_{ij}(t))\|\dot{y}_{ij}(t)\|dt$$
$$= \int_0^1 (1-t)t\|v_j - v_i\|dt = \frac{1}{6}\|e_{ij}\|$$

Mass matrix - 2D, 3/3

$$\mathbf{M}_{ij} = \int_{\mathcal{M}} \psi_i(p)\psi_j(p)dp = \begin{cases} 0 & \text{if } (v_i, v_j) \notin \mathcal{E} \\ \frac{1}{6}\|e_{ij}\| & \text{if } (v_i, v_j) \in \mathcal{E} \\ \frac{1}{3}\sum_{k \in \mathcal{N}(i)} \|e_{ik}\| & \text{if } i = j \end{cases}$$

Mass matrix - 2D, 3/3

$$\mathbf{M}_{ij} = \int_{\mathcal{M}} \psi_i(p)\psi_j(p)dp = \begin{cases} 0 & \text{if } (v_i, v_j) \notin \mathcal{E} \\ \frac{1}{6}\|e_{ij}\| & \text{if } (v_i, v_j) \in \mathcal{E} \\ \frac{1}{3}\sum_{k \in \mathcal{N}(i)} \|e_{ik}\| & \text{if } i = j \end{cases}$$

In the special case where all the edges have the same length $e_{ij} = s$, the mass matrix is given by:

$$\mathbf{M} = \frac{1}{6}s \begin{pmatrix} 4 & 1 & 0 & & & 1 \\ 1 & 4 & 1 & 0 & & 0 \\ 0 & 1 & 4 & 1 & 0 & 0 \\ \vdots & & & & & \vdots \\ 0 & & & & \ddots & 1 \\ 1 & 0 & \dots & & 1 & 4 \end{pmatrix}$$

Literature

- Polygon Mesh Processing (Botsch, Kobbelt, Pauly, Alliez, Levy)