SÃO JOSÉ DOS CAMPOS
SIBGRAPI '16
XXIX CONFERENCE ON GRAPHICS, PATTERNS AND IMAGES

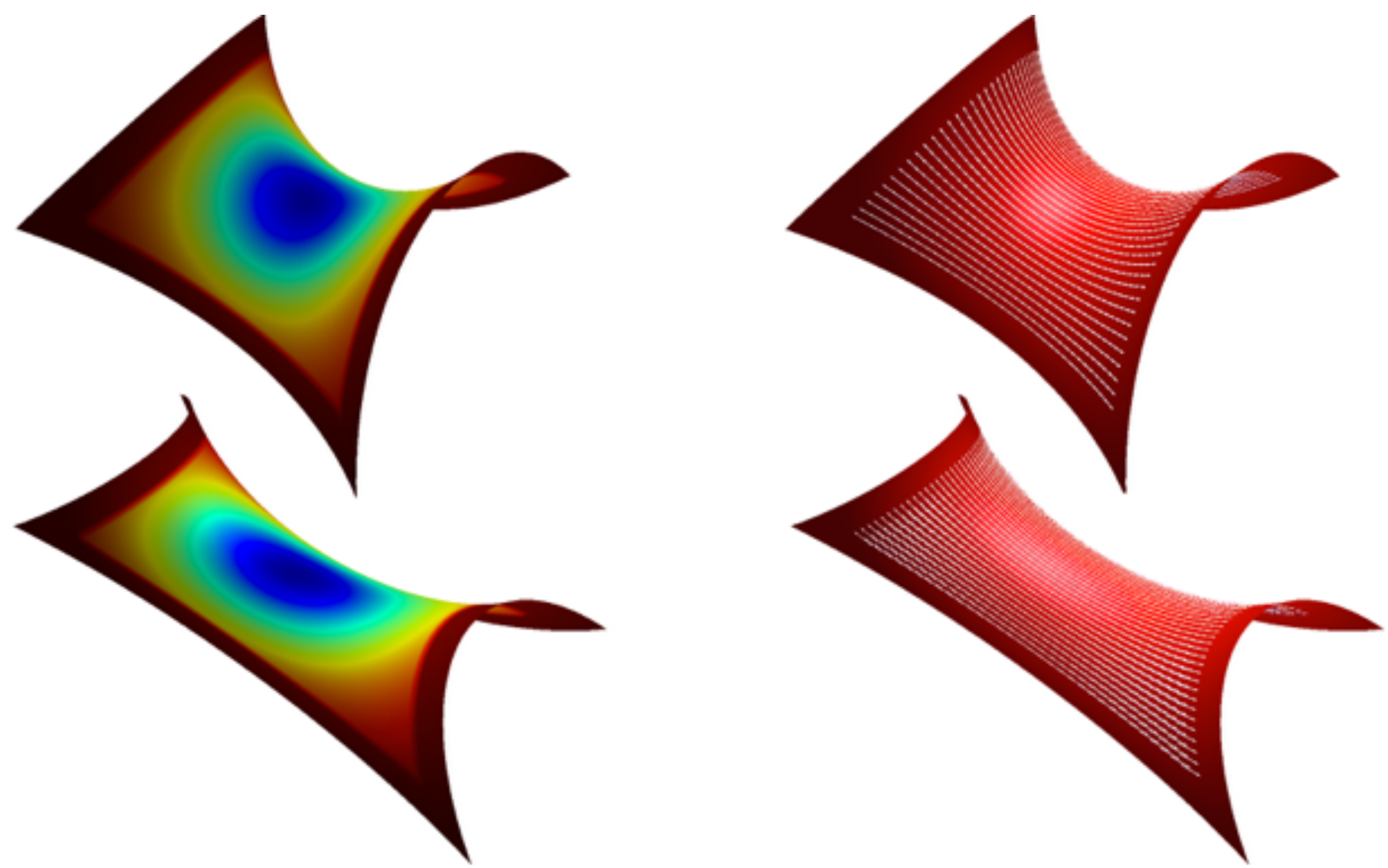
Estimating affine-invariant structures on triangle meshes

Thales Vieira
Mathematics, UFAL

Dimas Martinez
Mathematics, UFAM

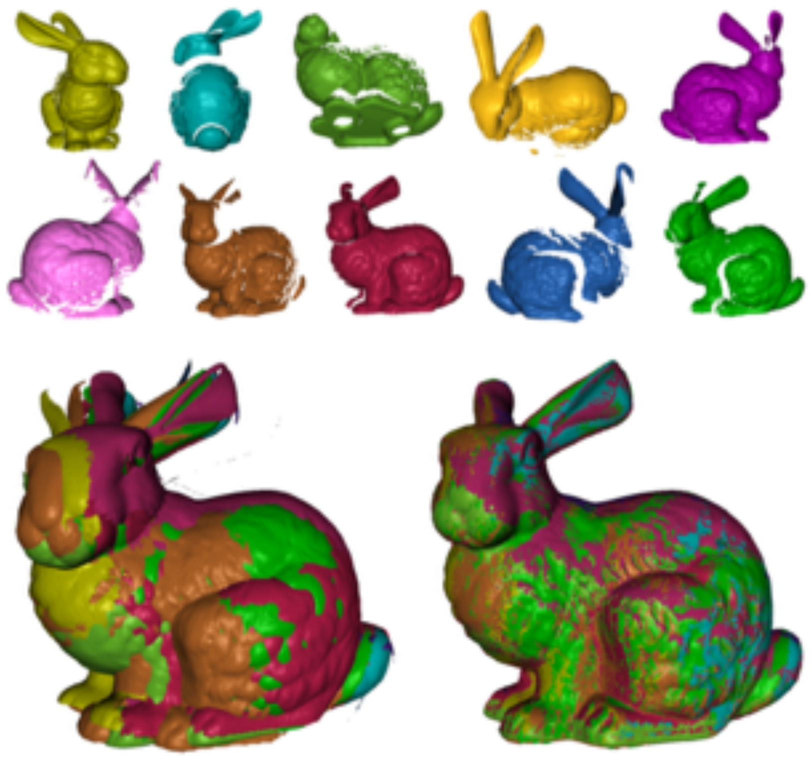
Maria Andrade
Mathematics, UFS

Thomas Lewiner
École Polytechnique



Invariant descriptors for shape analysis

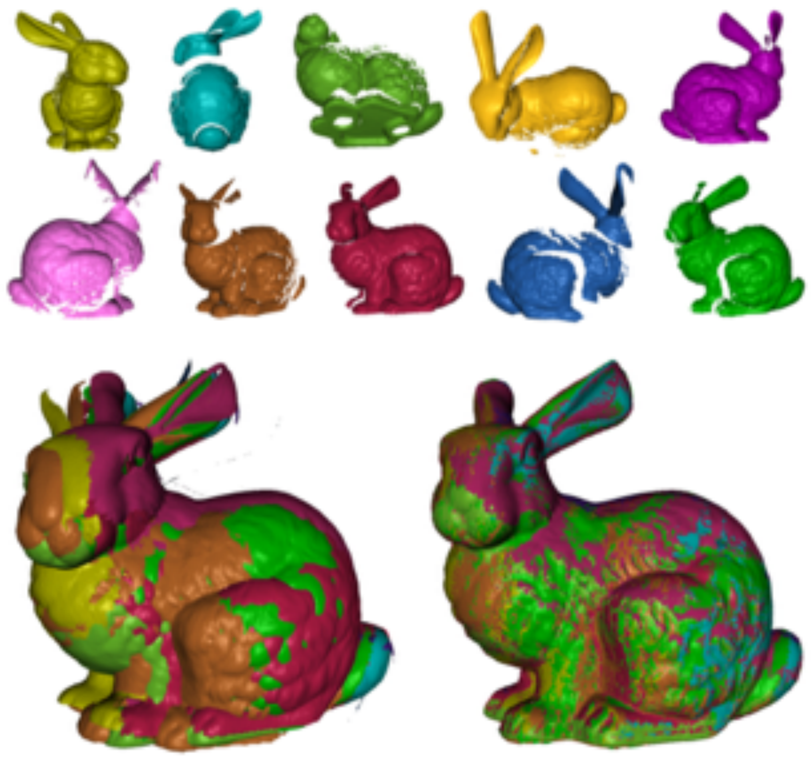
Rigid surface registration



Gelfand *et al* (2005)

Invariant descriptors for shape analysis

Rigid surface registration

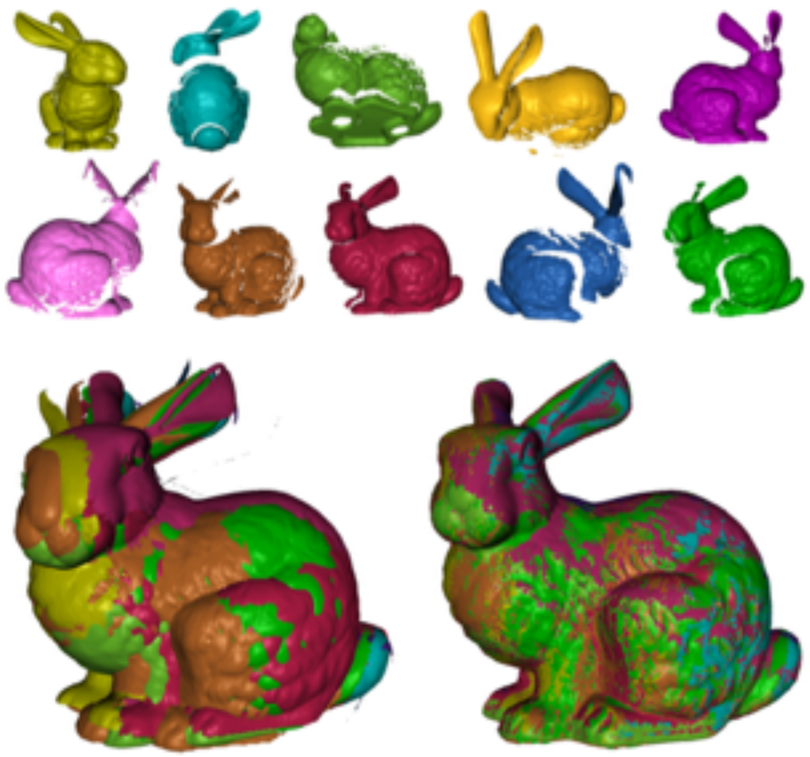


Gelfand *et al* (2005)

Rigid shape descriptors

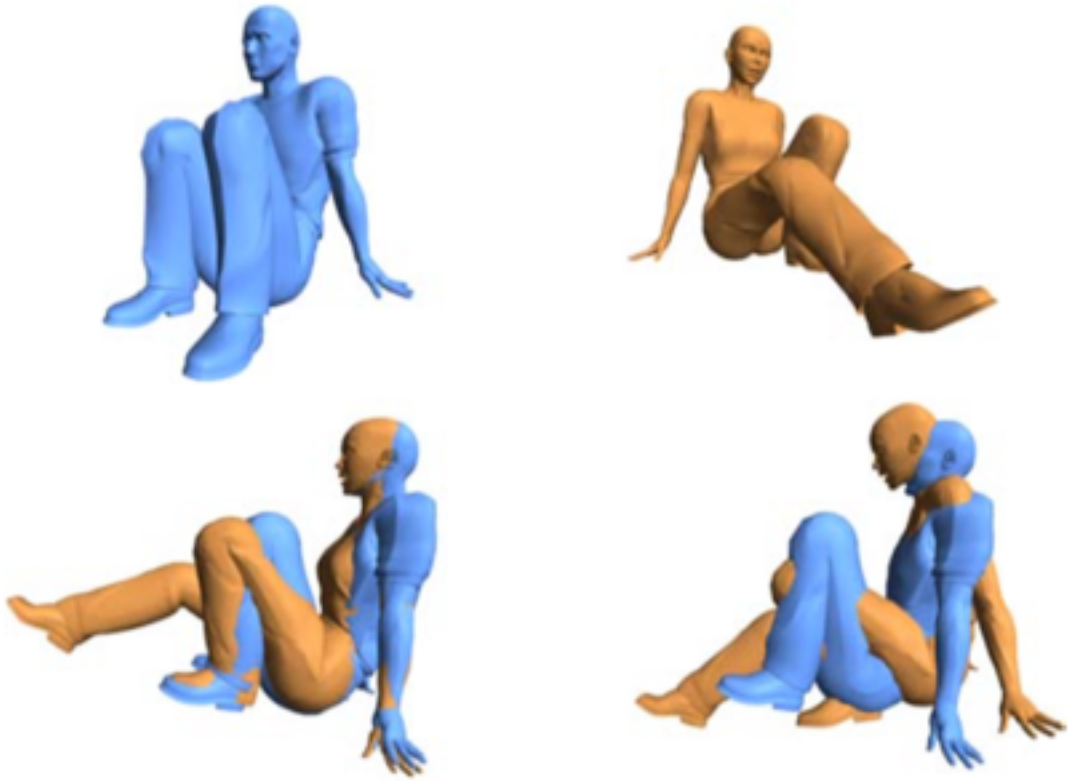
Invariant descriptors for shape analysis

Rigid surface registration



Gelfand *et al* (2005)

Non-rigid shape matching & similarity



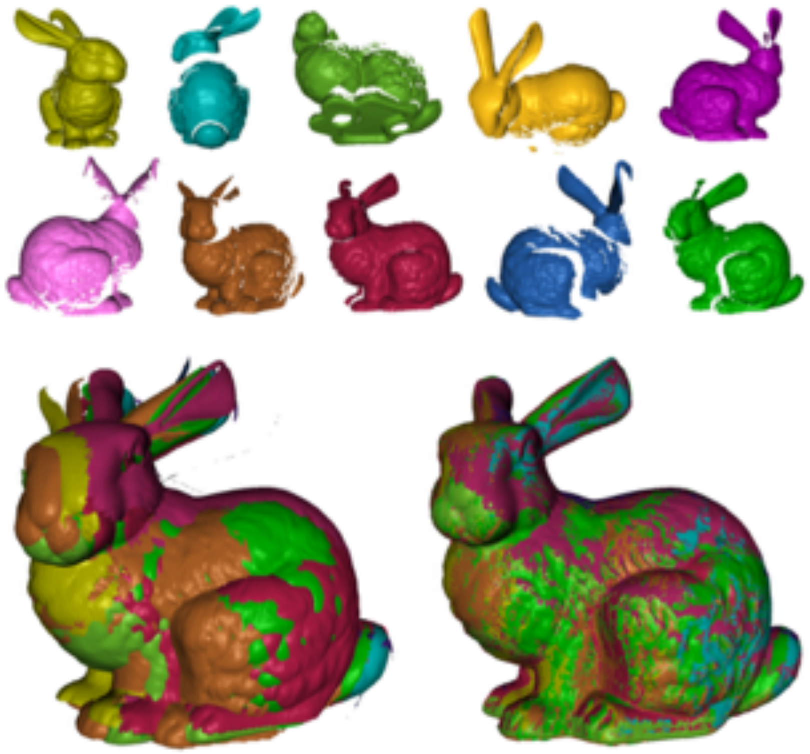
Gal and Cohen-Or (2006)

Rigid shape descriptors

Invariant descriptors for shape analysis

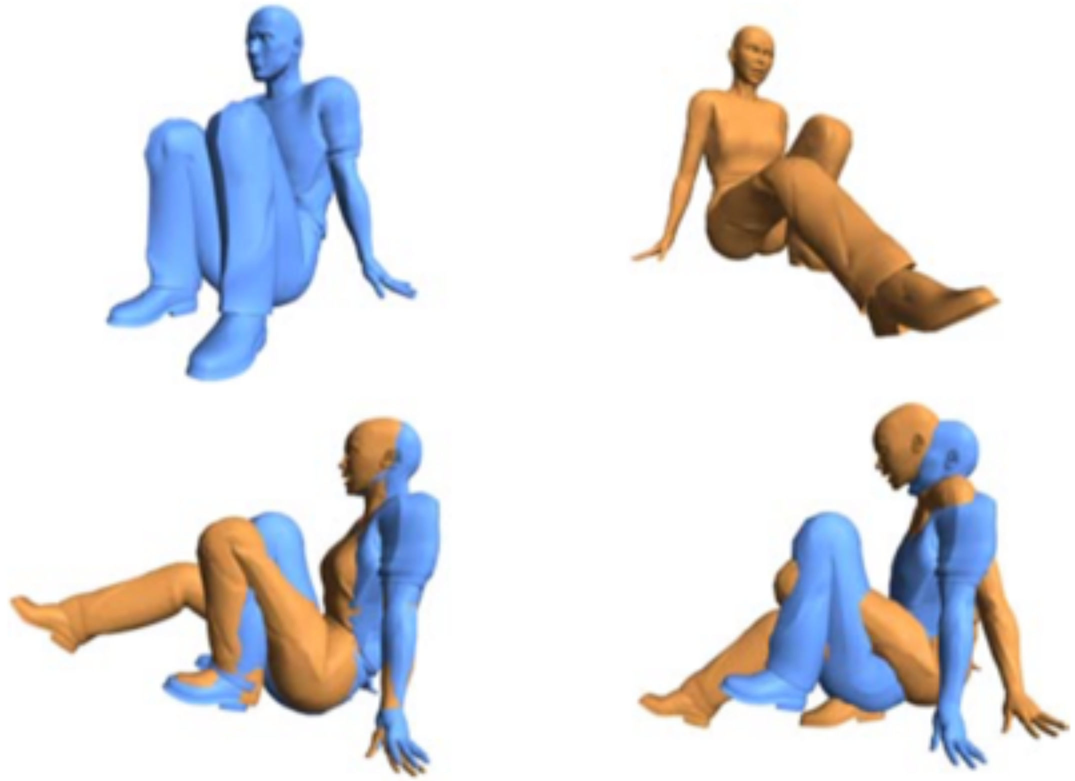
Local/rigid shape descriptors

Rigid surface registration



Gelfand *et al* (2005)

Non-rigid shape matching & similarity

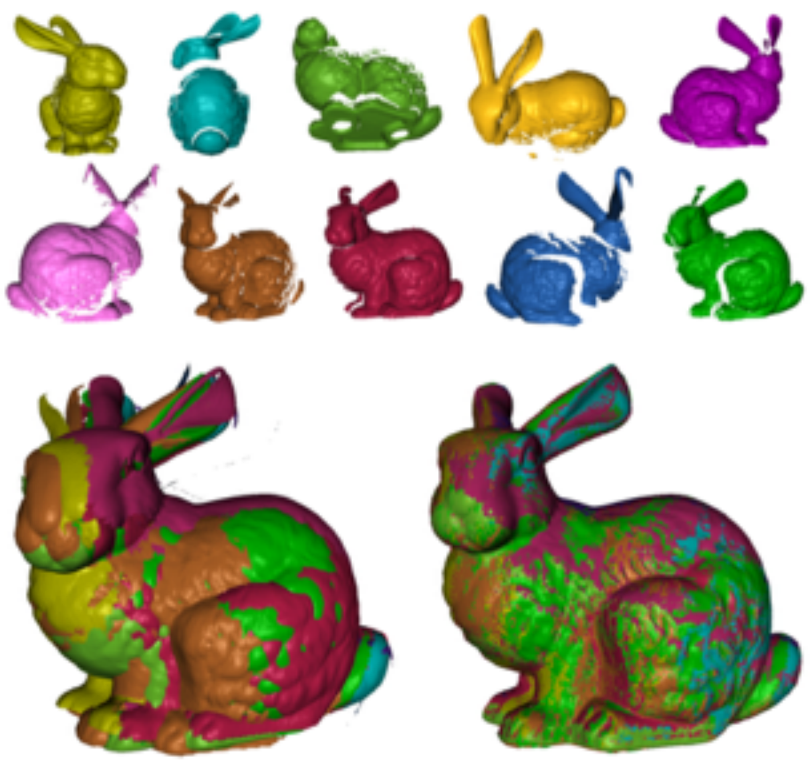


Gal and Cohen-Or (2006)

Rigid shape descriptors

Invariant descriptors for shape analysis

Rigid surface registration

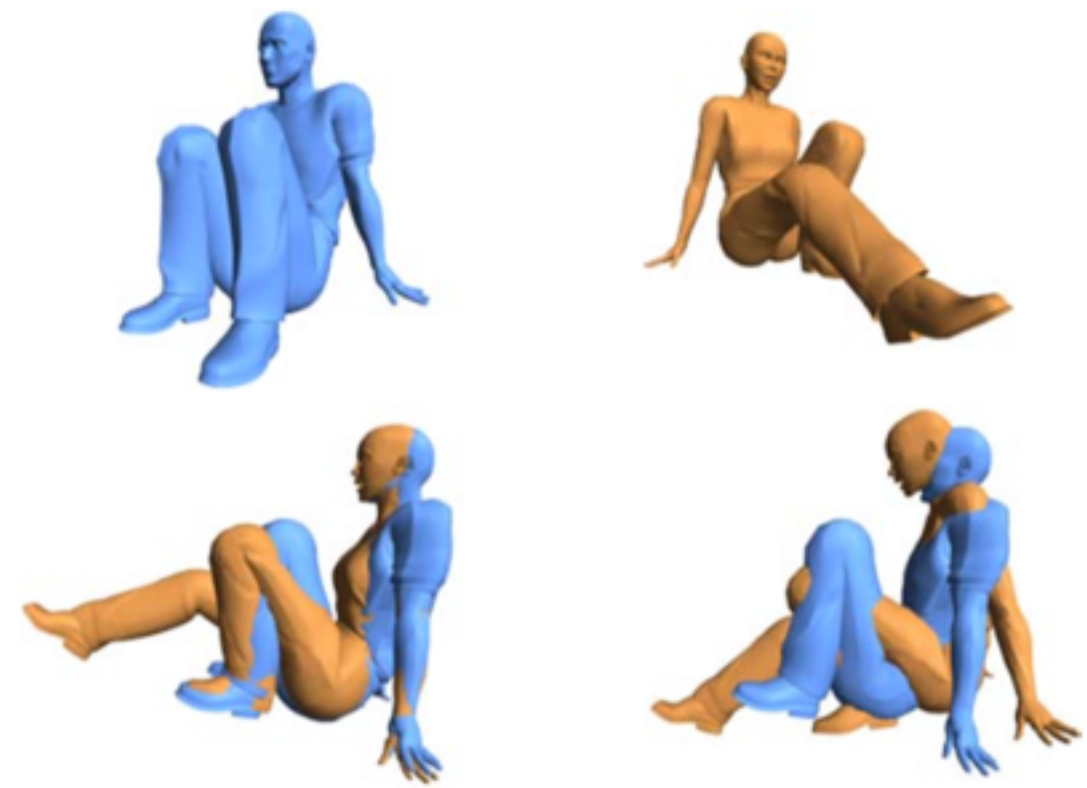


Gelfand *et al* (2005)

Rigid shape descriptors

Local/rigid shape descriptors

Non-rigid shape matching & similarity



Gal and Cohen-Or (2006)

How about non-rigid shape descriptors?

Space of transformations

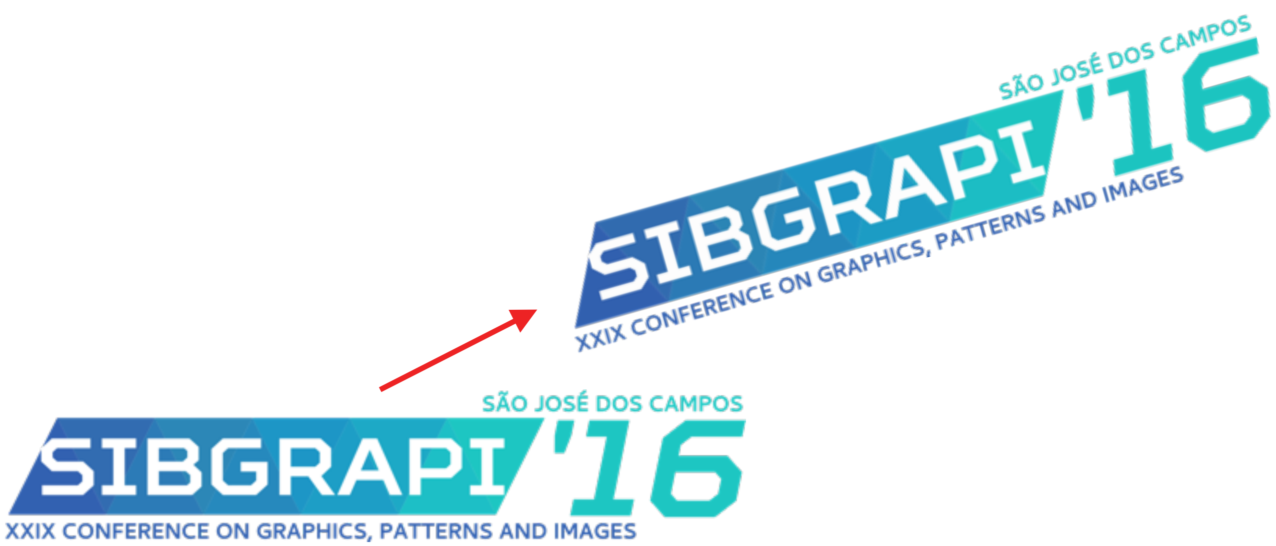
Let $A \in M_3(\mathbb{R})$ and $t \in \mathbb{R}^3$.

Space of transformations

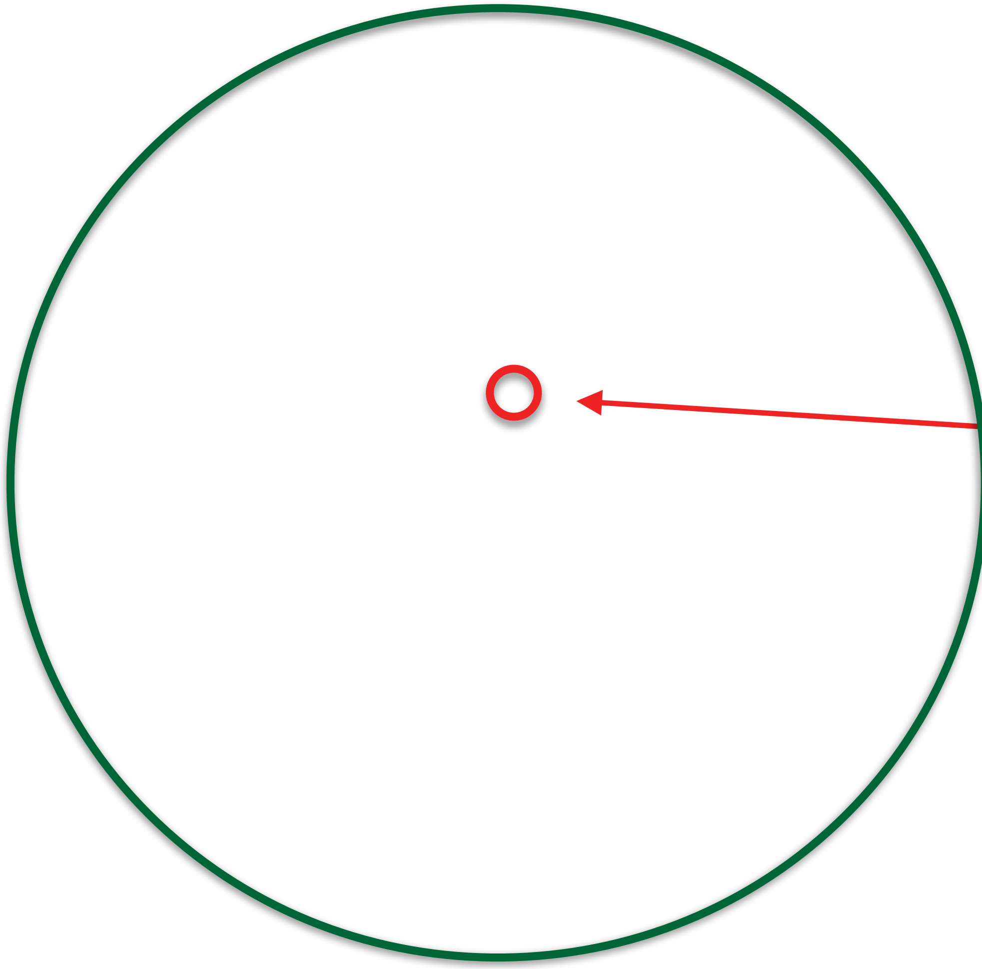
Let $A \in M_3(\mathbb{R})$ and $t \in \mathbb{R}^3$.

Rigid transformations
(Euclidean geometry)

$$T(p) = Ap + t, \quad A^T = A^{-1}.$$



Space of transformations



Let $A \in M_3(\mathbb{R})$ and $t \in \mathbb{R}^3$.

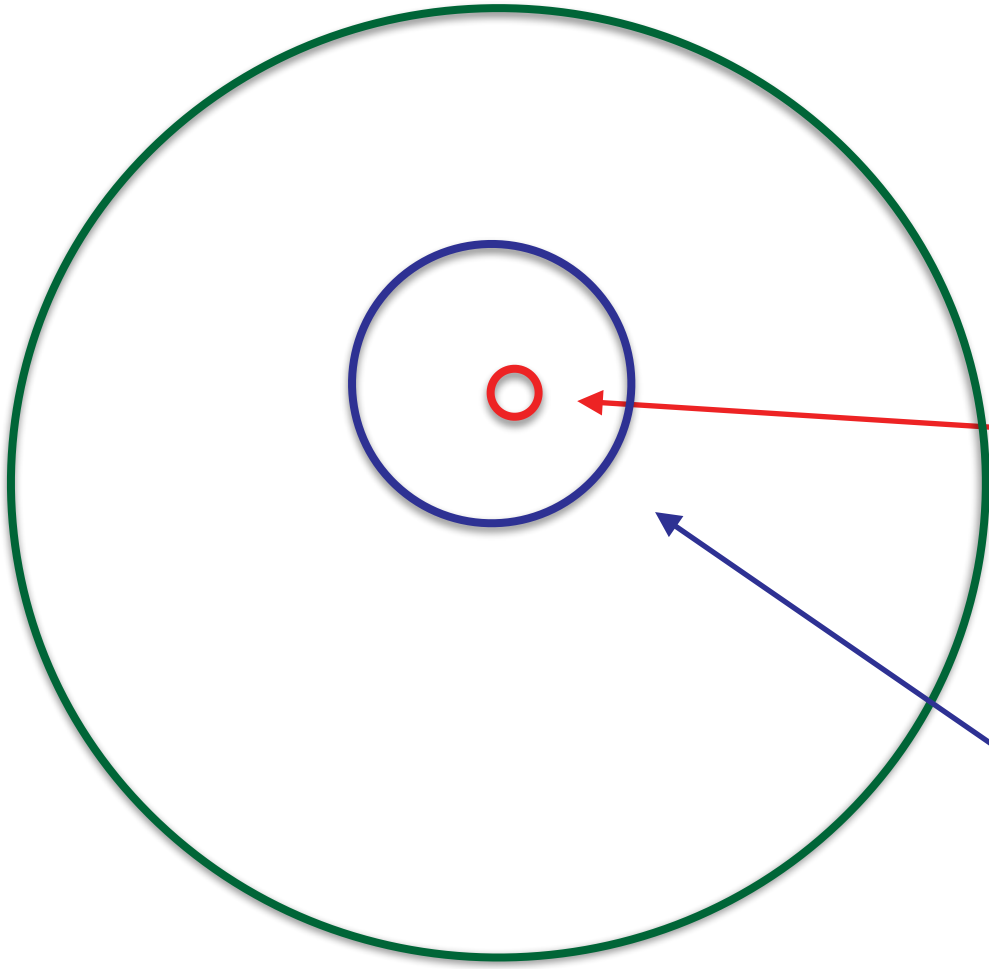
Rigid transformations
(Euclidean geometry)

$$T(p) = Ap + t, \quad A^T = A^{-1}.$$

Non-rigid
transformations

$$T(p) = ?$$

Space of transformations



Let $A \in M_3(\mathbb{R})$ and $t \in \mathbb{R}^3$.

Rigid transformations
(Euclidean geometry)

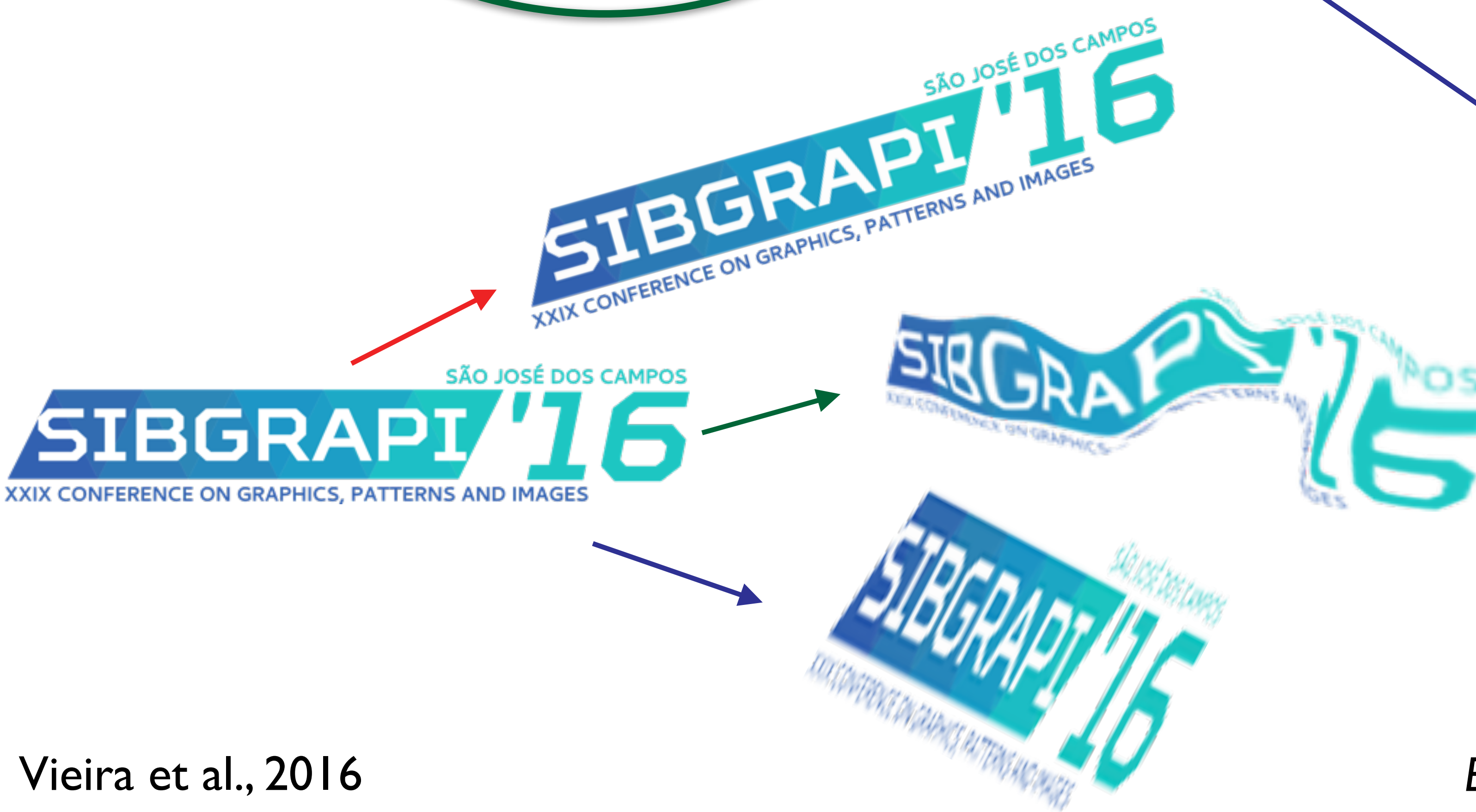
$$T(p) = Ap + t, \quad A^T = A^{-1}.$$

Non-rigid
transformations

$$T(p) = ?$$

Equi-affine
transformations
(Affine geometry)

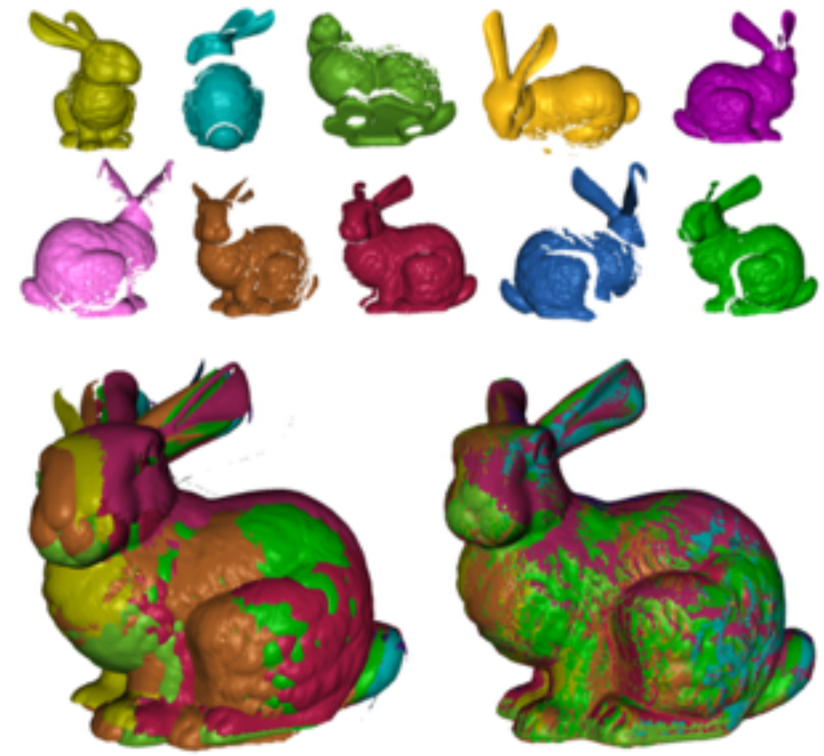
$$T(p) = Ap + t, \quad \det(A) = 1.$$



Related <u>Euclidean</u> Work

Measures from Euclidean geometry: invariant to rigid transformations

- Euclidean distance
- Mean / Gaussian curvature

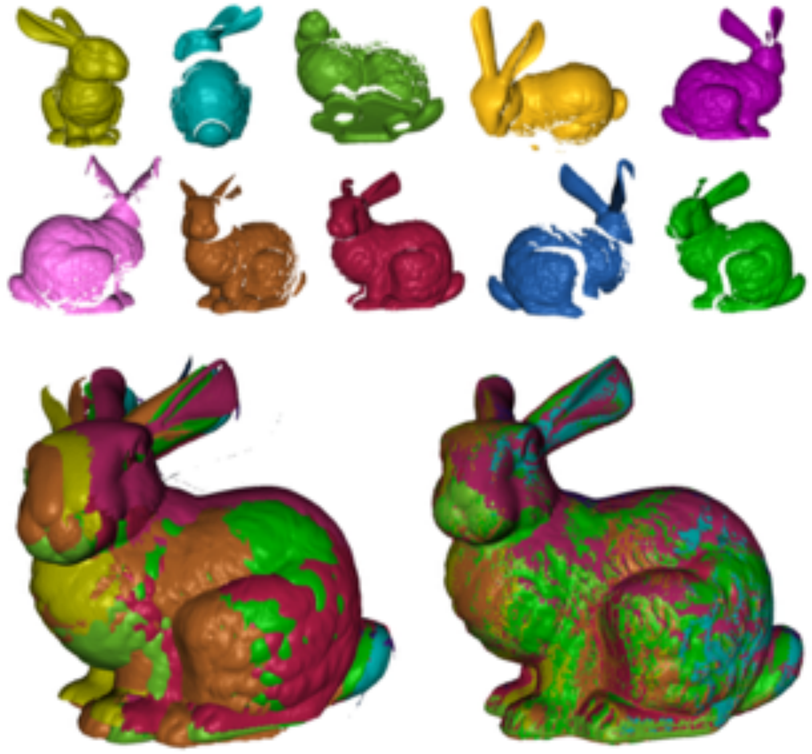


Gelfand *et al* (2005)

Related Euclidean Work

Measures from Euclidean geometry:
invariant to rigid transformations

- Euclidean distance
- Mean / Gaussian curvature



Gelfand *et al* (2005)

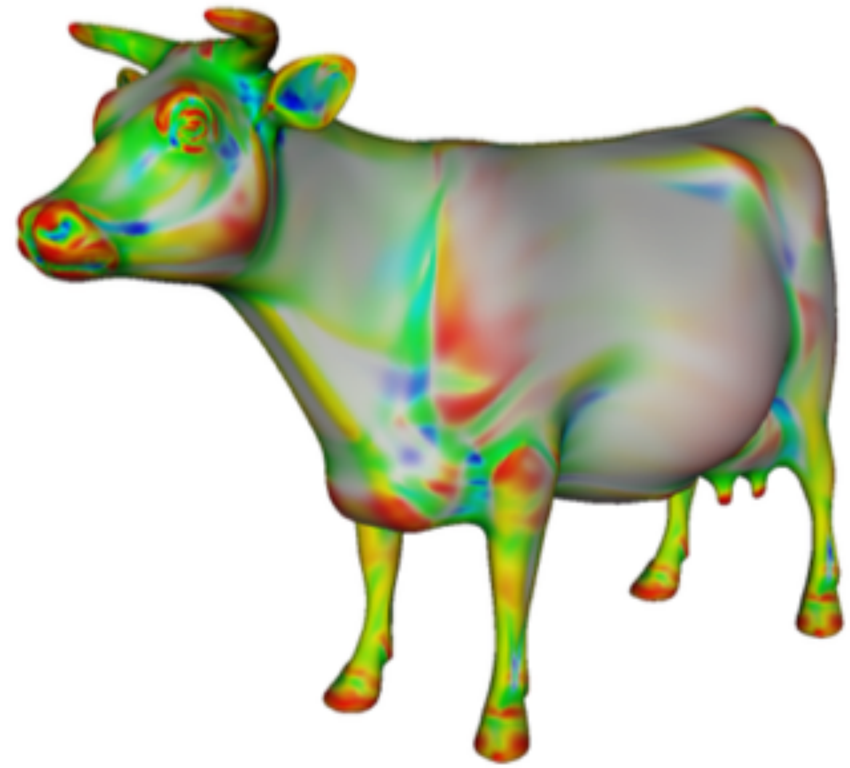
How to estimate these geometric measures on triangle meshes?

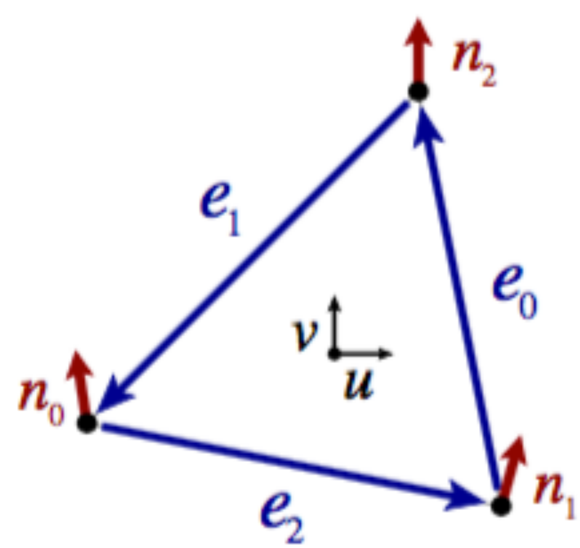
Related <u>Euclidean</u> Work

Estimating the curvature tensor from normal vectors

Taubin (1995)

Rusinkiewicz (2003)





$$\mathbf{II}\begin{pmatrix} e_0 \cdot u \\ e_0 \cdot v \end{pmatrix} = \begin{pmatrix} (n_2 - n_1)\cdot u \\ (n_2 - n_1)\cdot v \end{pmatrix}$$

$$\mathbf{II}\begin{pmatrix} e_1 \cdot u \\ e_1 \cdot v \end{pmatrix} = \begin{pmatrix} (n_0 - n_2)\cdot u \\ (n_0 - n_2)\cdot v \end{pmatrix}$$

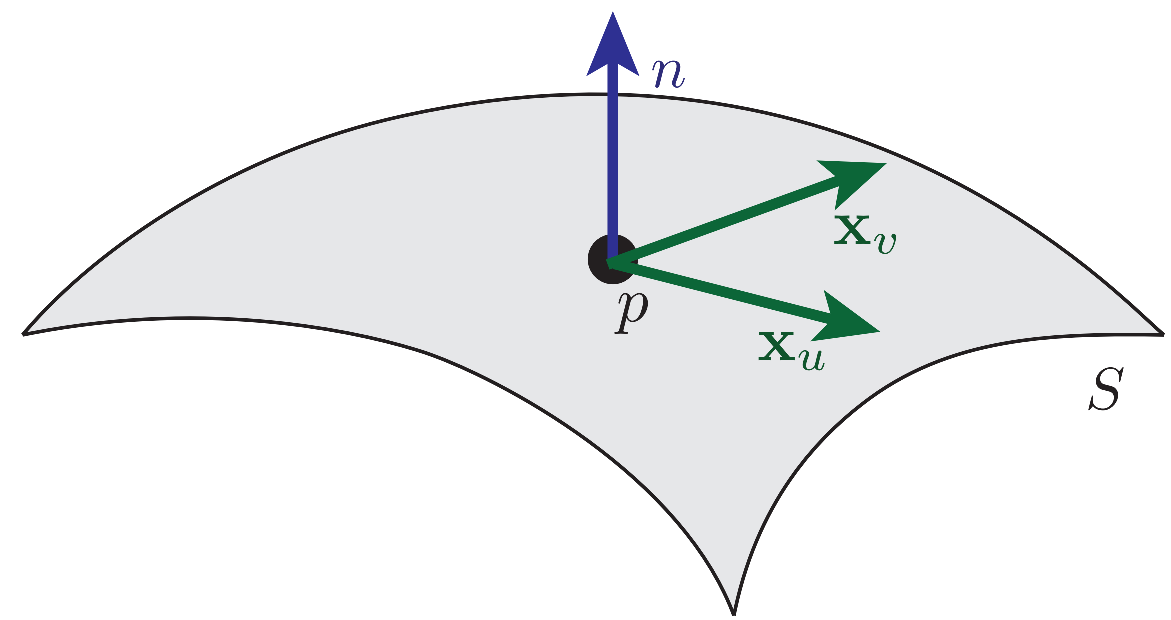
$$\mathbf{II}\begin{pmatrix} e_2 \cdot u \\ e_2 \cdot v \end{pmatrix} = \begin{pmatrix} (n_1 - n_0)\cdot u \\ (n_1 - n_0)\cdot v \end{pmatrix}$$

Euclidean geometry background

Shape operator

$$\mathcal{S}^{\mathrm{e}}\colon T_pS \to T_pS$$

$$\mathcal{S}^{\mathrm{e}}w = -D_w n$$

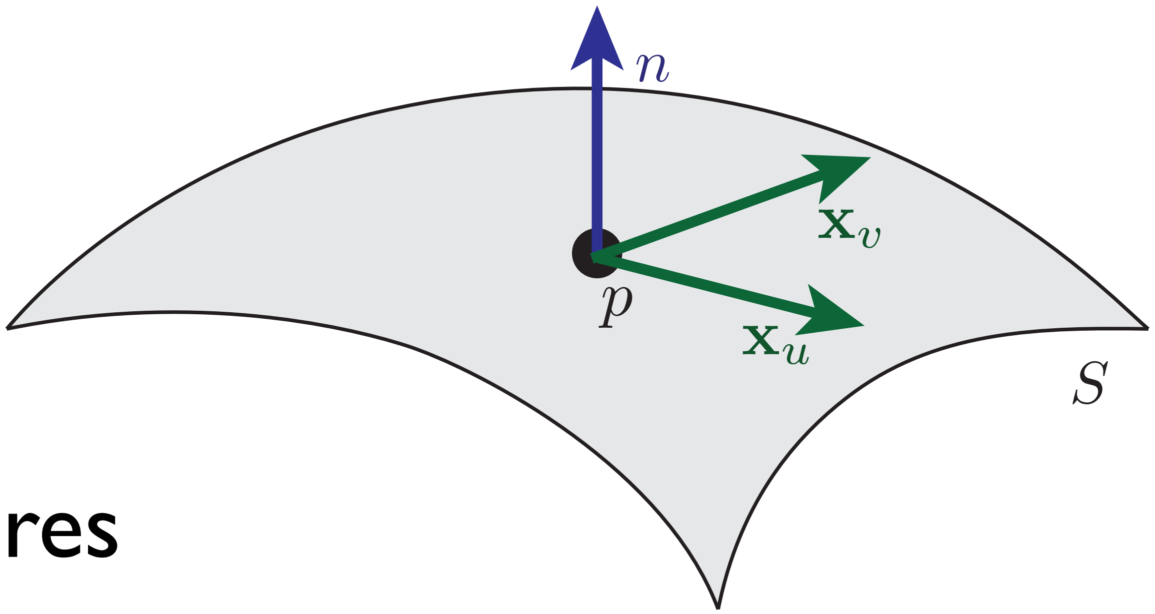


Euclidean geometry background

Shape operator

$$\mathcal{S}^{\mathrm{e}} : T_pS \rightarrow T_pS$$

$$\mathcal{S}^{\mathrm{e}} w = -D_w n$$



Principal directions and curvatures

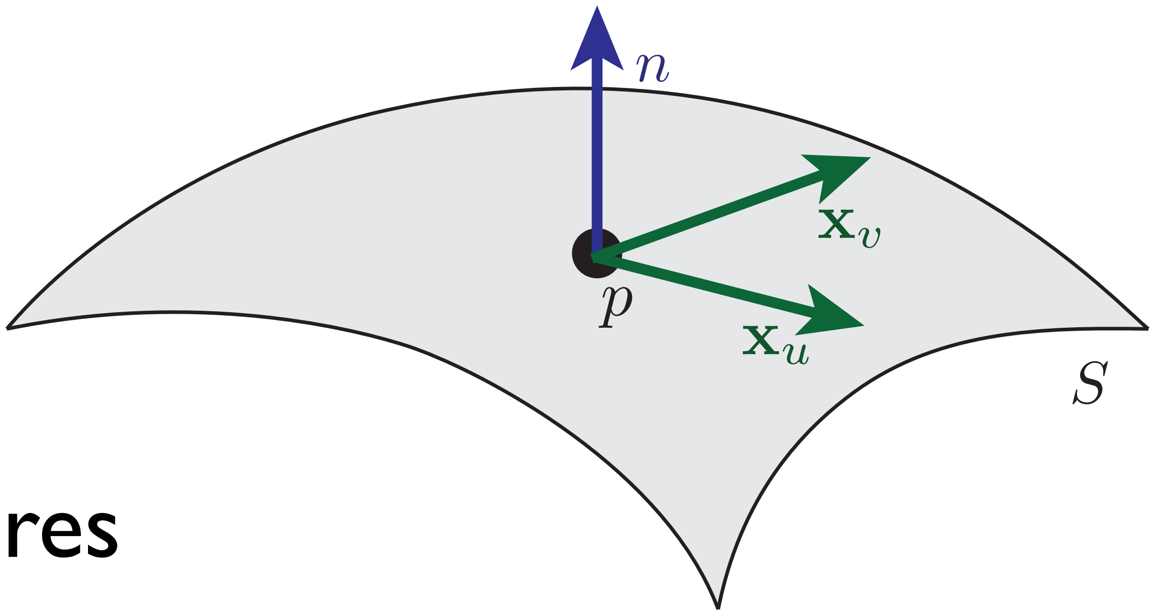
$$\mathcal{S}^{\mathrm{e}} e^{\mathrm{e}}_{\mathrm{min}} = k^{\mathrm{e}}_{\mathrm{min}} e^{\mathrm{e}}_{\mathrm{min}}$$
$$\mathcal{S}^{\mathrm{e}} e^{\mathrm{e}}_{\mathrm{max}} = k^{\mathrm{e}}_{\mathrm{max}} e^{\mathrm{e}}_{\mathrm{max}}$$
$$e^{\mathrm{e}}_{\mathrm{max}}, e^{\mathrm{e}}_{\mathrm{min}} \in T_pS$$

Gaussian and mean curvatures

$$K = \det(\mathcal{S}^{\mathrm{e}}) \qquad H = -\tfrac{1}{2}\mathrm{tr}(\mathcal{S}^{\mathrm{e}})$$

Euclidean geometry background

Shape operator



$$\mathcal{S}^{\mathrm{e}} \colon T_pS \to T_pS$$

$$\mathcal{S}^{\mathrm{e}} w = -D_w n$$

Principal directions and curvatures

$$\mathcal{S}^{\mathrm{e}} e^{\mathrm{e}}_{\min} = k^{\mathrm{e}}_{\min} e^{\mathrm{e}}_{\min}$$
$$\mathcal{S}^{\mathrm{e}} e^{\mathrm{e}}_{\max} = k^{\mathrm{e}}_{\max} e^{\mathrm{e}}_{\max}$$
$$e^{\mathrm{e}}_{\max}, e^{\mathrm{e}}_{\min} \in T_pS$$

Gaussian and mean curvatures

$$K = \det(\mathcal{S}^{\mathrm{e}}) \qquad H = -\tfrac{1}{2}\mathrm{tr}(\mathcal{S}^{\mathrm{e}})$$

Fixing an orthonormal frame $\{\mathbf{e}_1, \mathbf{e}_2\}$ of T_pS:

$$-\mathcal{S}^{\mathrm{e}} = (D_{\mathbf{e}_1} n, D_{\mathbf{e}_2} n) = \begin{pmatrix} \langle D_{\mathbf{e}_1} n, \mathbf{e}_1 \rangle & \langle D_{\mathbf{e}_2} n, \mathbf{e}_1 \rangle \\ \langle D_{\mathbf{e}_1} n, \mathbf{e}_2 \rangle & \langle D_{\mathbf{e}_2} n, \mathbf{e}_2 \rangle \end{pmatrix}$$

Euclidean geometry background

Fixing an orthonormal frame $\{\mathbf{e}_1, \mathbf{e}_2\}$ of $T_p S$:

$$-\mathcal{S}^{\mathrm{e}} = (D_{\mathbf{e}_1} n, D_{\mathbf{e}_2} n) = \begin{pmatrix} \langle D_{\mathbf{e}_1} n, \mathbf{e}_1 \rangle & \langle D_{\mathbf{e}_2} n, \mathbf{e}_1 \rangle \\ \\ \langle D_{\mathbf{e}_1} n, \mathbf{e}_2 \rangle & \langle D_{\mathbf{e}_2} n, \mathbf{e}_2 \rangle \end{pmatrix}$$

Rusinkiewicz tensor

Euclidean geometry background

Fixing an orthonormal frame $\{\mathbf{e}_1, \mathbf{e}_2\}$ of T_pS:

$$-\mathcal{S}^{\mathrm{e}} = (D_{\mathbf{e}_1}n, D_{\mathbf{e}_2}n) = \begin{pmatrix} \langle D_{\mathbf{e}_1}n, \mathbf{e}_1\rangle & \langle D_{\mathbf{e}_2}n, \mathbf{e}_1\rangle \\ \langle D_{\mathbf{e}_1}n, \mathbf{e}_2\rangle & \langle D_{\mathbf{e}_2}n, \mathbf{e}_2\rangle \end{pmatrix}$$

Rusinkiewicz tensor

1. Compute a normal vector field by averaging face normals

Euclidean geometry background

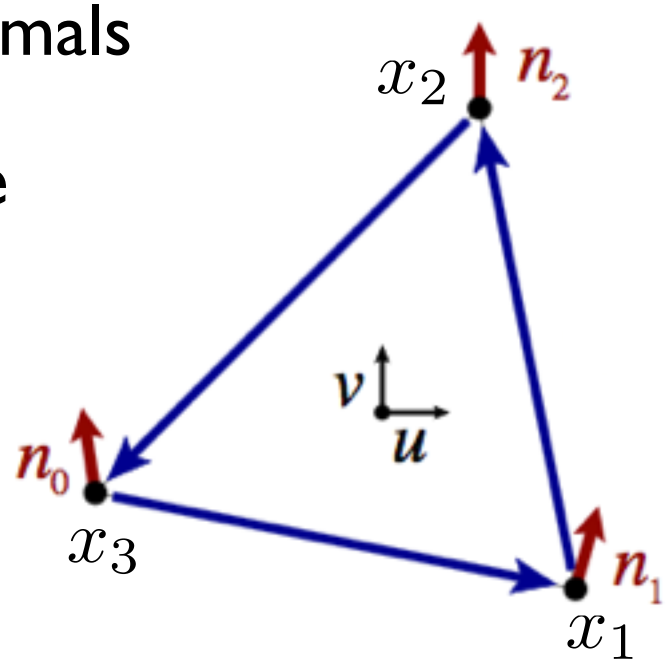
Fixing an orthonormal frame $\{\mathbf{e}_1, \mathbf{e}_2\}$ of T_pS:

$$-\mathcal{S}^{\mathrm{e}} = (D_{\mathbf{e}_1} n, D_{\mathbf{e}_2} n) = \begin{pmatrix} \langle D_{\mathbf{e}_1} n, \mathbf{e}_1 \rangle & \langle D_{\mathbf{e}_2} n, \mathbf{e}_1 \rangle \\ \langle D_{\mathbf{e}_1} n, \mathbf{e}_2 \rangle & \langle D_{\mathbf{e}_2} n, \mathbf{e}_2 \rangle \end{pmatrix}$$

Rusinkiewicz tensor

1. Compute a normal vector field by averaging face normals
2. Discretize the shape operator per face applying finite differences per edge

$$-\mathcal{S}^{\mathrm{e}}[e_{ij}] = \begin{pmatrix} \langle n_i - n_j, \mathbf{e}_1 \rangle \\ \langle n_i - n_j, \mathbf{e}_2 \rangle \end{pmatrix}, \quad e_{ij} = x_i - x_j$$



Vieira et al., 2016

Estimating affine-invariant structures on triangle meshes

Euclidean geometry background

Fixing an orthonormal frame $\{\mathbf{e}_1, \mathbf{e}_2\}$ of T_pS:

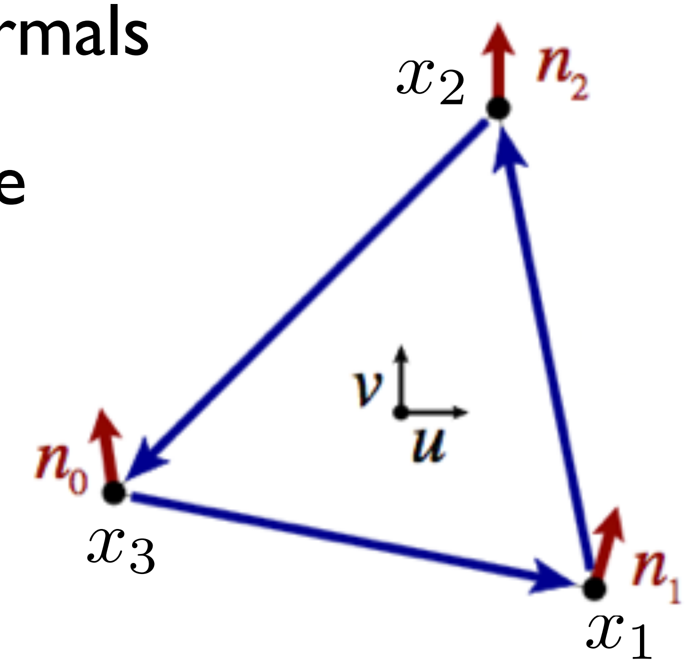
$$-\mathcal{S}^{\mathrm{e}} = (D_{\mathbf{e}_1} n, D_{\mathbf{e}_2} n) = \begin{pmatrix} \langle D_{\mathbf{e}_1} n, \mathbf{e}_1 \rangle & \langle D_{\mathbf{e}_2} n, \mathbf{e}_1 \rangle \\ \langle D_{\mathbf{e}_1} n, \mathbf{e}_2 \rangle & \langle D_{\mathbf{e}_2} n, \mathbf{e}_2 \rangle \end{pmatrix}$$

Rusinkiewicz tensor

1. Compute a normal vector field by averaging face normals

2. Discretize the shape operator per face applying finite differences per edge

$$-\mathcal{S}^{\mathrm{e}}[e_{ij}] = \begin{pmatrix} \langle n_i - n_j, \mathbf{e}_1 \rangle \\ \langle n_i - n_j, \mathbf{e}_2 \rangle \end{pmatrix}, \quad e_{ij} = x_i - x_j$$



3. Apply least-squares to find shape operators per face

Vieira et al., 2016

Estimating affine-invariant structures on triangle meshes

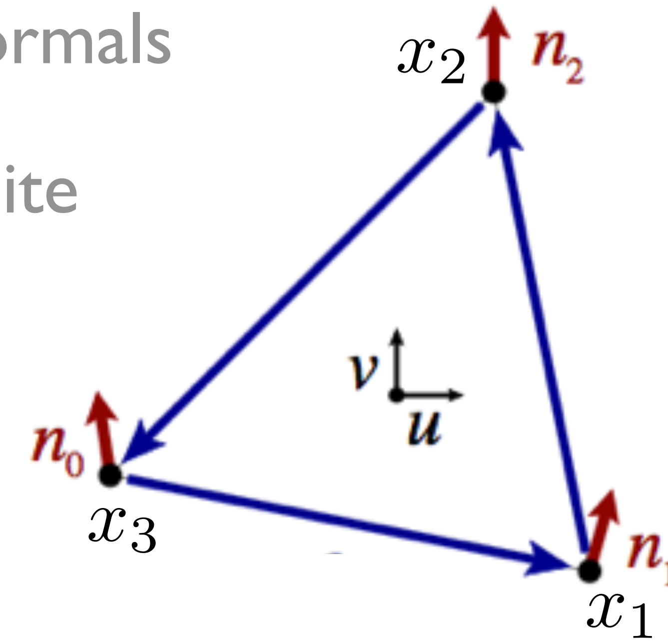
Euclidean geometry background

Rusinkiewicz tensor

1. Estimate a normal vector field by averaging face normals

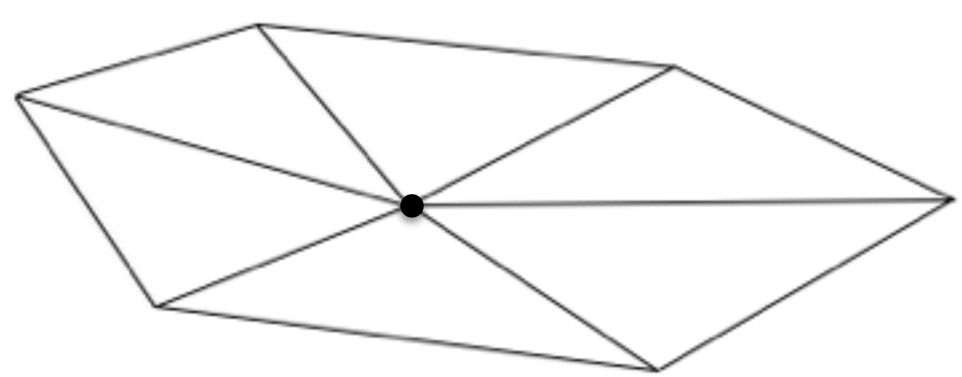
2. Discretize the shape operator per face applying finite differences per edge

$$-\mathcal{S}^{\mathrm{e}}[e_{ij}] = \begin{pmatrix} \langle n_i - n_j, \mathbf{e}_1 \rangle \\ \\ \langle n_i - n_j, \mathbf{e}_2 \rangle \end{pmatrix}, \quad e_{ij} = x_i - x_j$$



3. Apply least-squares to find shape operators per face

4. To find the shape operator at each vertex, average adjacent face operators after changing coordinates to a common coordinate system.



Vieira et al., 2016 *Estimating affine-invariant structures on triangle meshes*

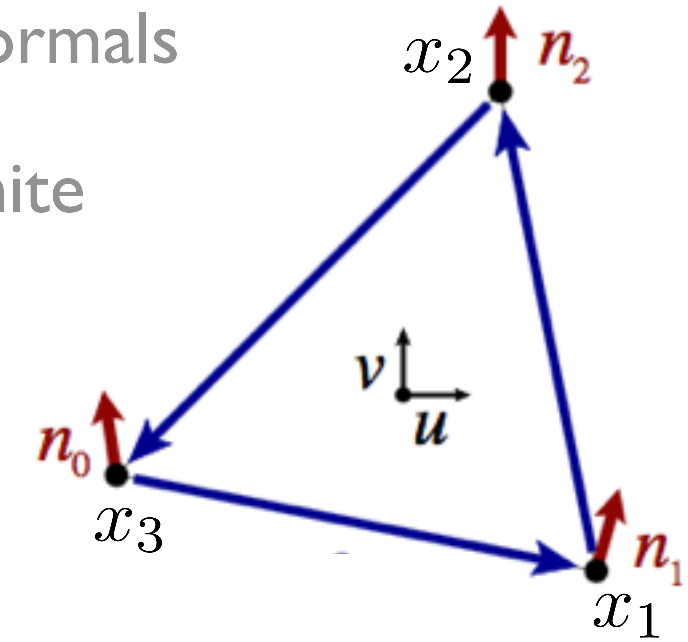
Euclidean geometry background

Rusinkiewicz tensor

1. Estimate a normal vector field by averaging face normals

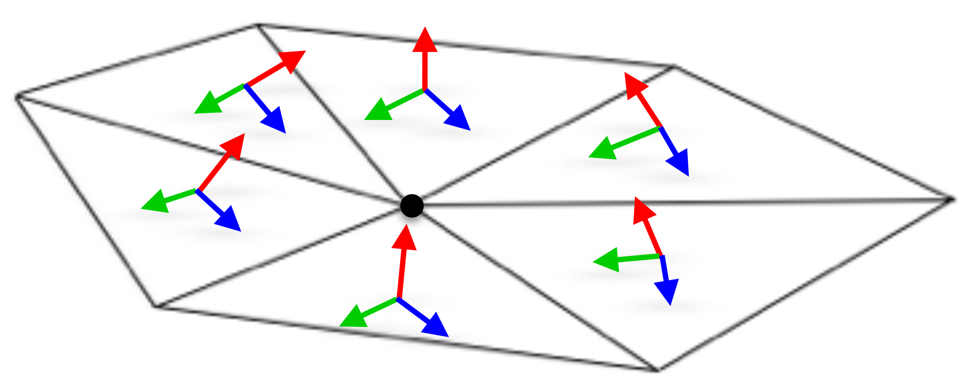
2. Discretize the shape operator per face applying finite differences per edge

$$-\mathcal{S}^{\mathrm{e}}[e_{ij}] = \begin{pmatrix} \langle n_i - n_j, \mathbf{e}_1 \rangle \\ \\ \langle n_i - n_j, \mathbf{e}_2 \rangle \end{pmatrix}, \quad e_{ij} = x_i - x_j$$



3. Apply least-squares to find shape operators per face

4. To find the shape operator at each vertex, average adjacent face operators after changing coordinates to a common coordinate system.



Vieira et al., 2016 *Estimating affine-invariant structures on triangle meshes*

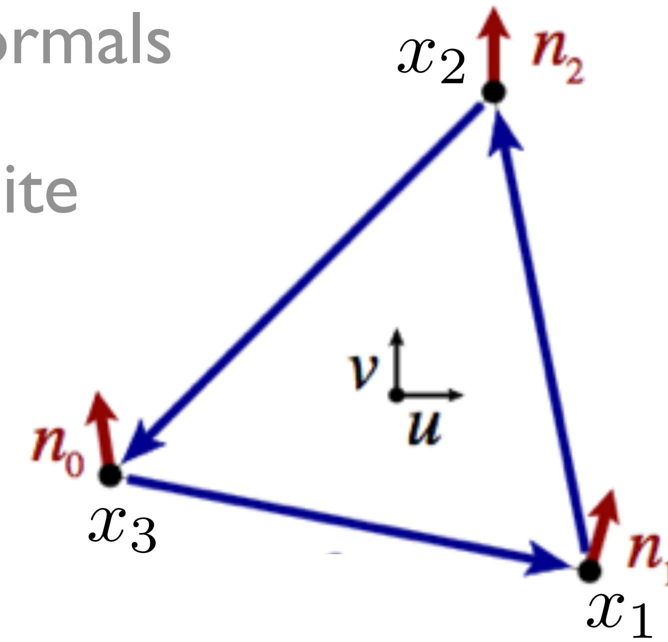
Euclidean geometry background

Rusinkiewicz tensor

1. Estimate a normal vector field by averaging face normals

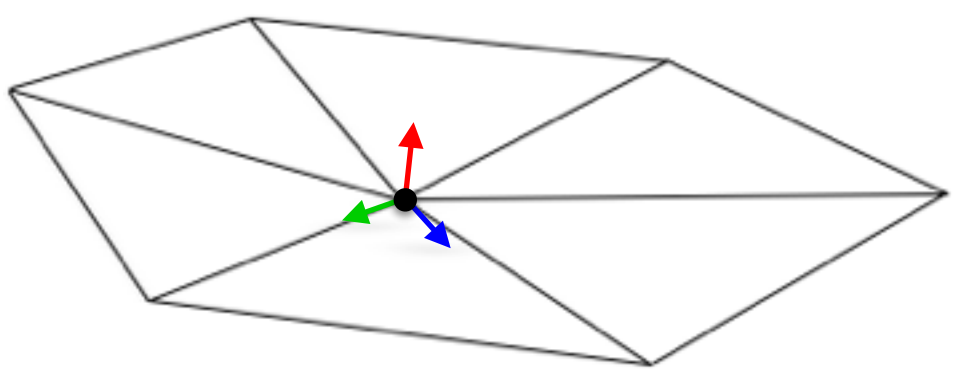
2. Discretize the shape operator per face applying finite differences per edge

$$-\mathcal{S}^{\mathrm{e}}[e_{ij}] = \begin{pmatrix} \langle n_i - n_j, \mathbf{e}_1 \rangle \\ \\ \langle n_i - n_j, \mathbf{e}_2 \rangle \end{pmatrix}, \quad e_{ij} = x_i - x_j$$



3. Apply least-squares to find shape operators per face

4. To find the shape operator at each vertex, average adjacent face operators after changing coordinates to a common coordinate system.



Vieira et al., 2016

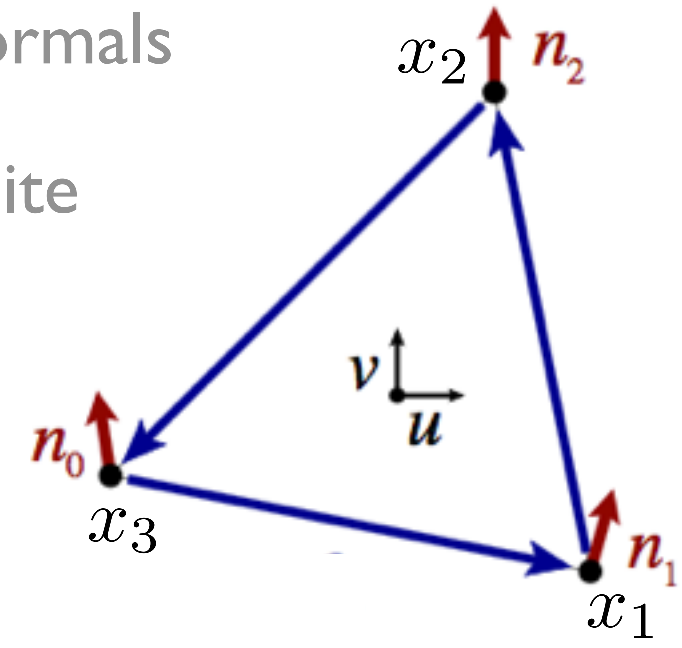
Estimating affine-invariant structures on triangle meshes

Euclidean geometry background

Rusinkiewicz tensor

1. Estimate a normal vector field by averaging face normals

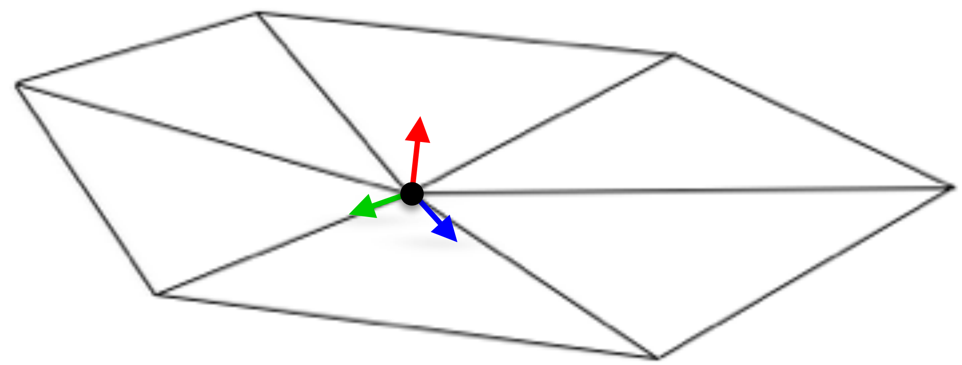
2. Discretize the shape operator per face applying finite differences per edge



$$-\mathcal{S}^{\mathrm{e}}[e_{ij}] = \begin{pmatrix} \langle n_i - n_j, \mathbf{e}_1 \rangle \\ \\ \langle n_i - n_j, \mathbf{e}_2 \rangle \end{pmatrix}, \quad e_{ij} = x_i - x_j$$

3. Apply least-squares to find shape operators per face

4. To find the shape operator at each vertex, average adjacent face operators after changing coordinates to a common coordinate system.



only normal vectors are required to discretize the shape operator!

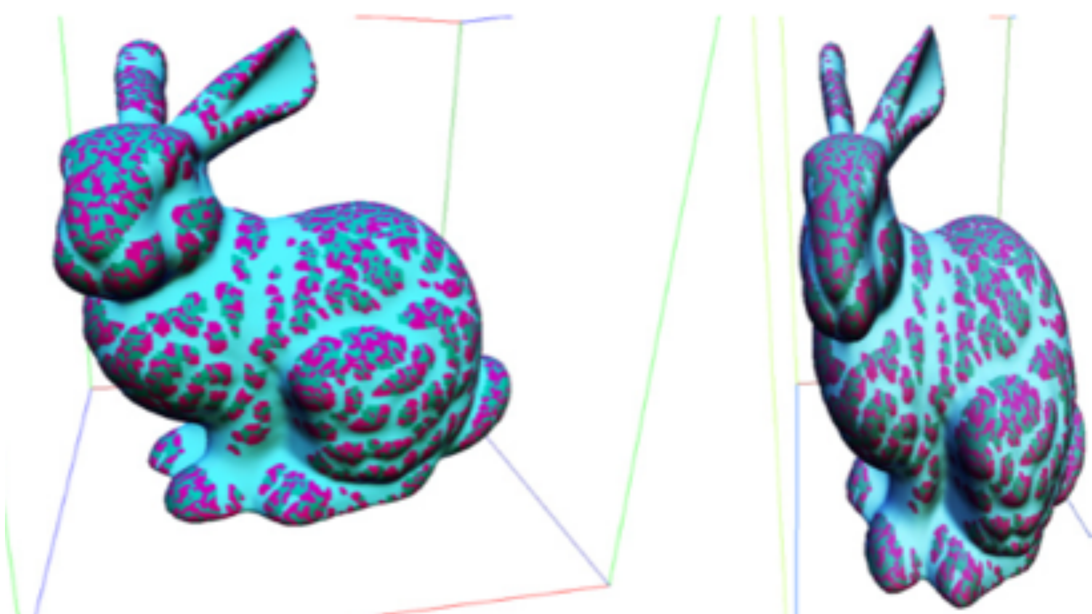
Vieira et al., 2016

Estimating affine-invariant structures on triangle meshes

Related Affine Work

Measures from affine geometry: invariance to equi-affine transformations (volume preserving)

- Affine Mean / Gaussian curvature
- Affine principal directions, …

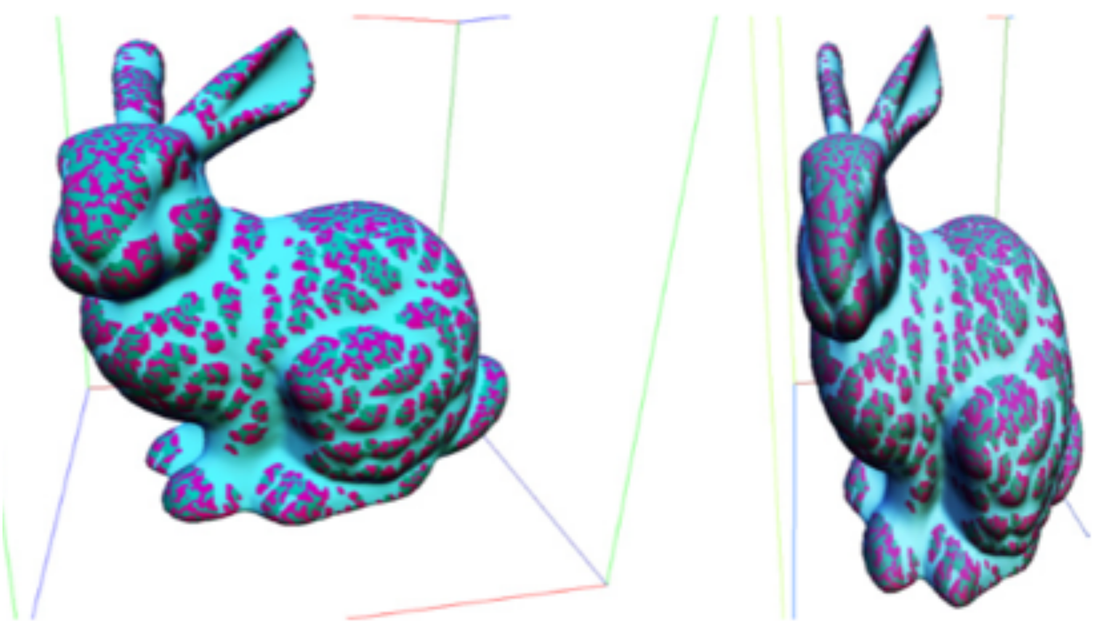


Andrade *et al* (2012)

Related Affine Work

Measures from affine geometry: invariance to equi-affine transformations (volume preserving)

- Affine Mean / Gaussian curvature
- Affine principal directions, …



Andrade *et al* (2012)

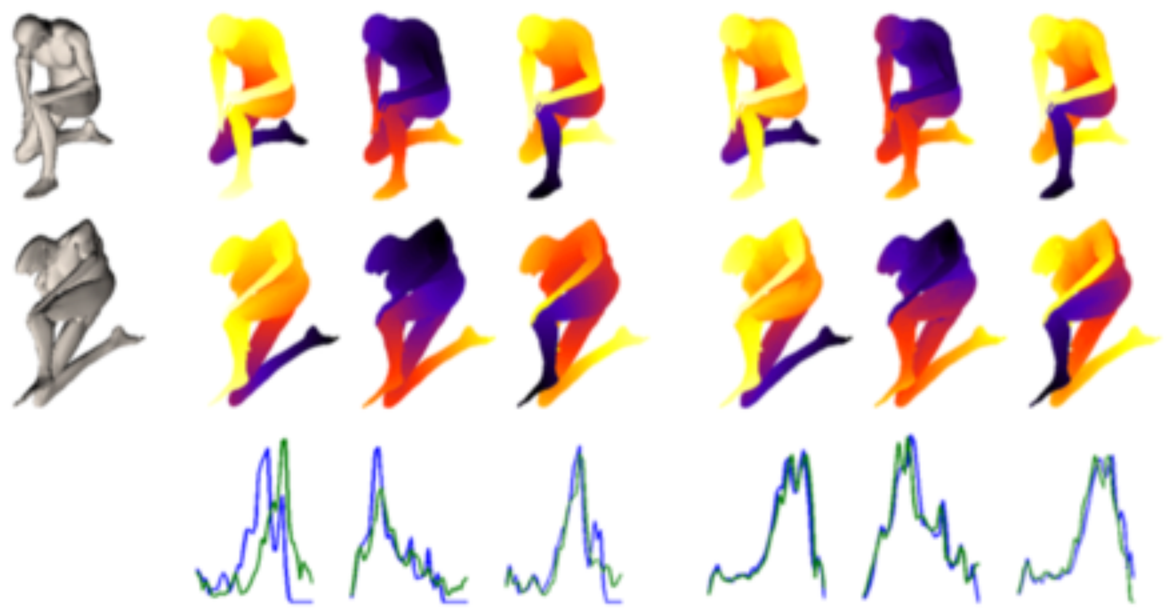
How to estimate these affine measures on triangle meshes?

Related Affine Work

Vieira et al., 2016 *Estimating affine-invariant structures on triangle meshes*

Related Affine Work

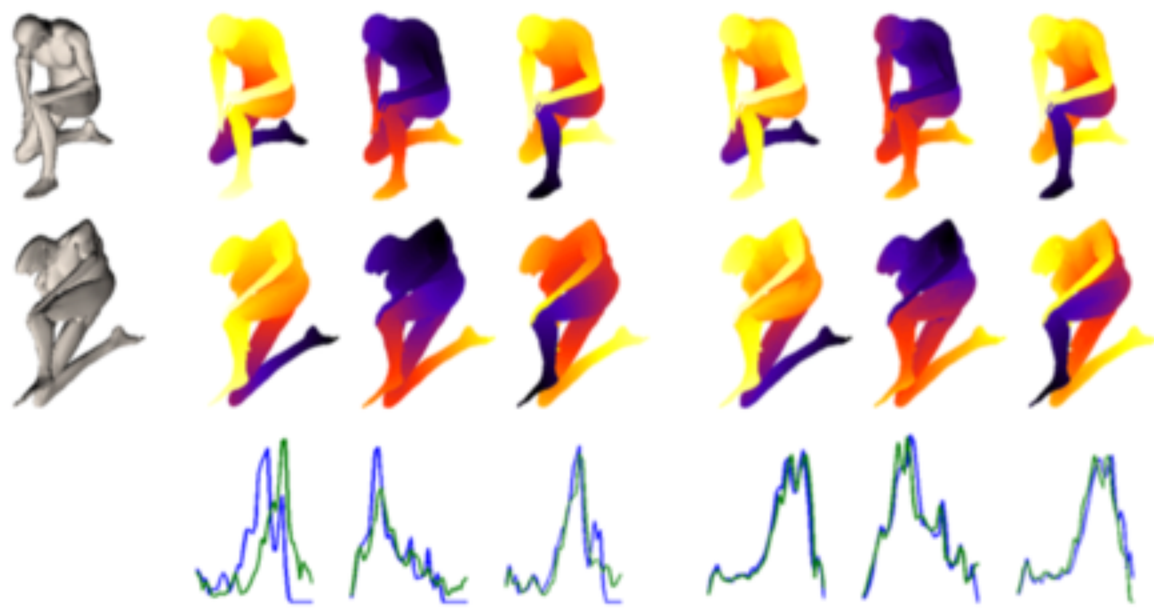
Equi-affine metric tensor



Raviv *et al* (2011)

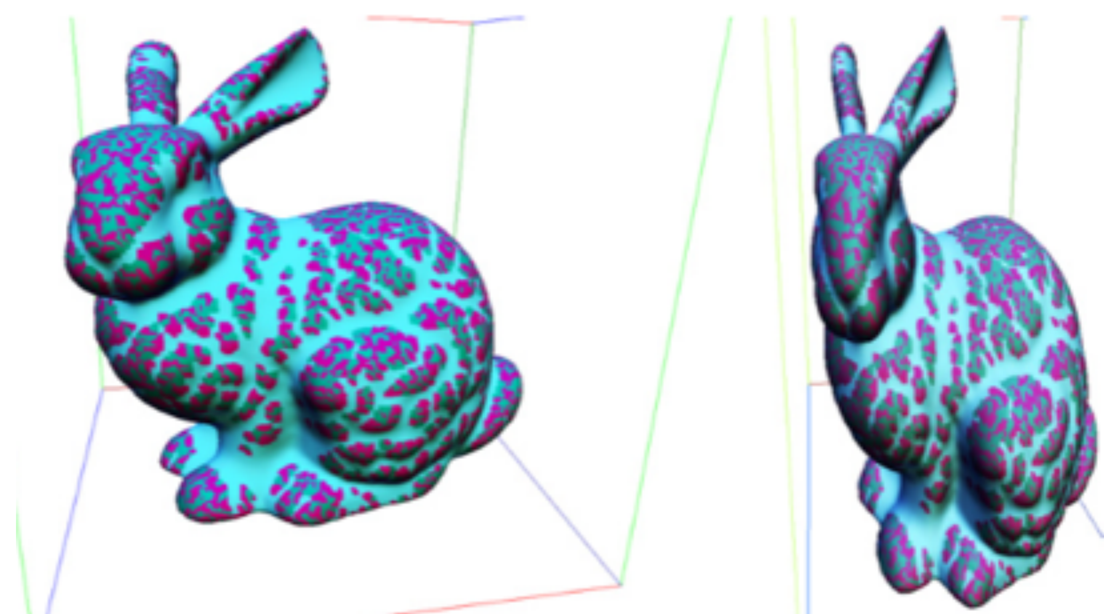
Related Affine Work

Equi-affine metric tensor



Raviv *et al* (2011)

Equi-affine curvature tensor on implicit surfaces



Andrade and Lewiner (2012)

Contributions

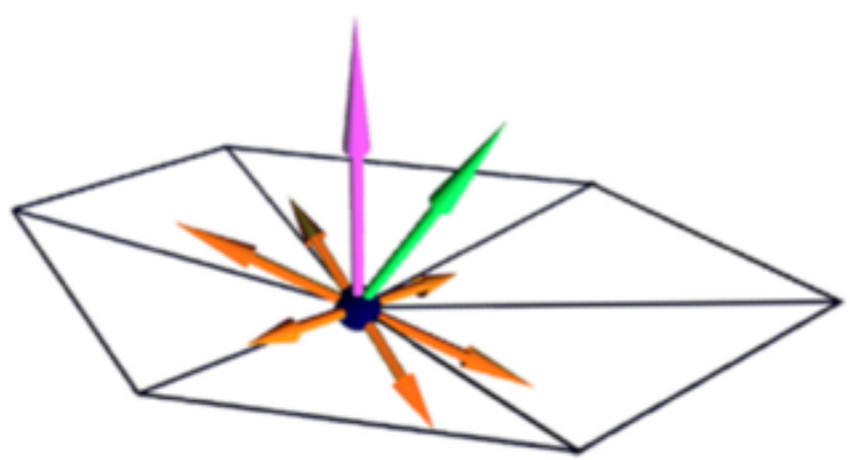
Estimators for equi-affine differential structures on triangle meshes:

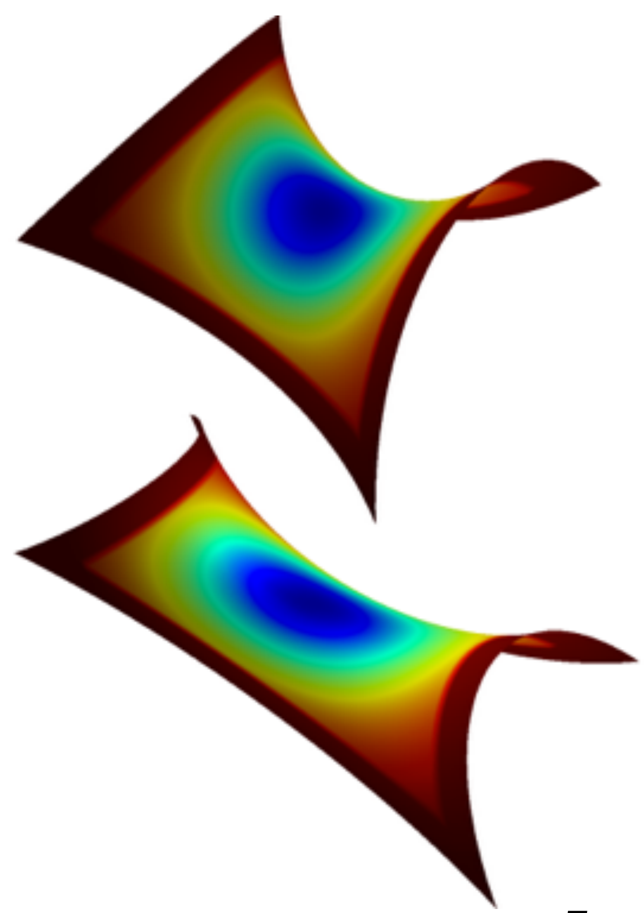
✓ Discrete affine normal vector

✓ Affine curvature tensor

✓ Affine Gaussian and mean curvatures

✓ Affine principal directions and curvatures





Vieira et al., 2016

Estimating affine-invariant structures on triangle meshes

Outline

1. Affine geometry background
2. Least-squares based affine normal estimator
3. Affine shape operator estimator
4. Experiments
5. Limitations & Future Work

Affine geometry background

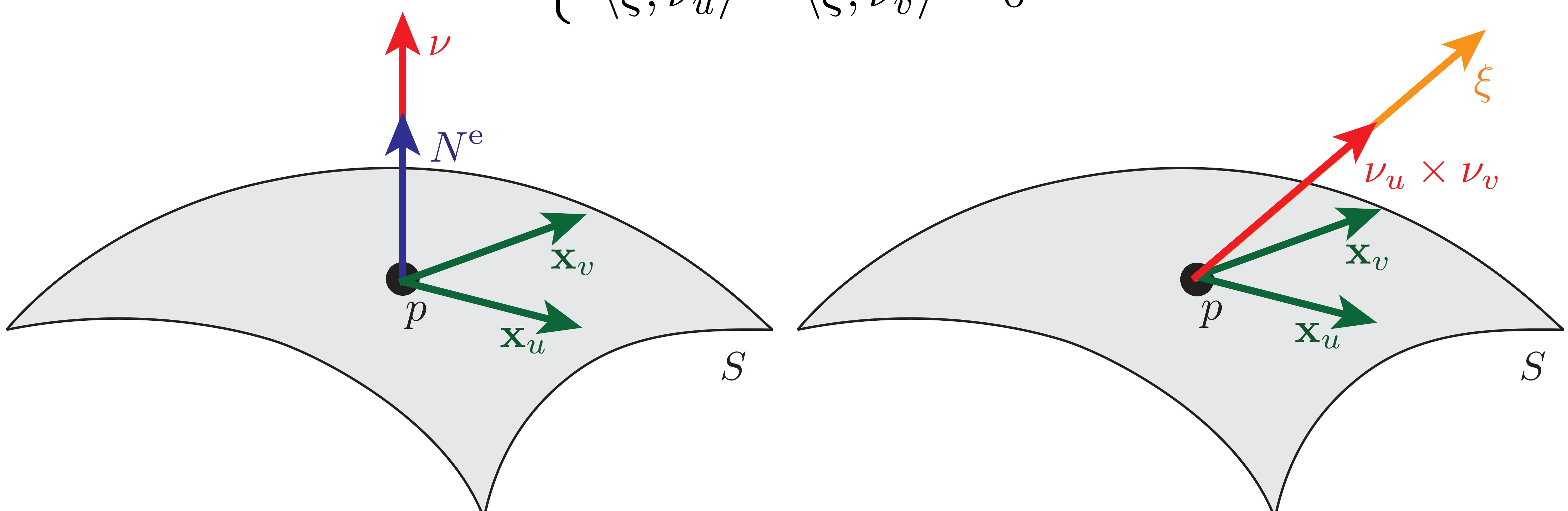
Affine co-normal vector

$$\nu = |K^{\mathrm{e}}|^{-1/4} N^{\mathrm{e}}$$

Affine normal vector

The affine normal vector ξ is implicitly defined by the equations

$$\begin{cases} \langle \nu, \xi \rangle = 1 \\ \langle \xi, \nu_u \rangle = \langle \xi, \nu_v \rangle = 0 \end{cases}$$



Vieira et al., 2016 *Estimating affine-invariant structures on triangle meshes*

Affine geometry background

Affine co-normal vector

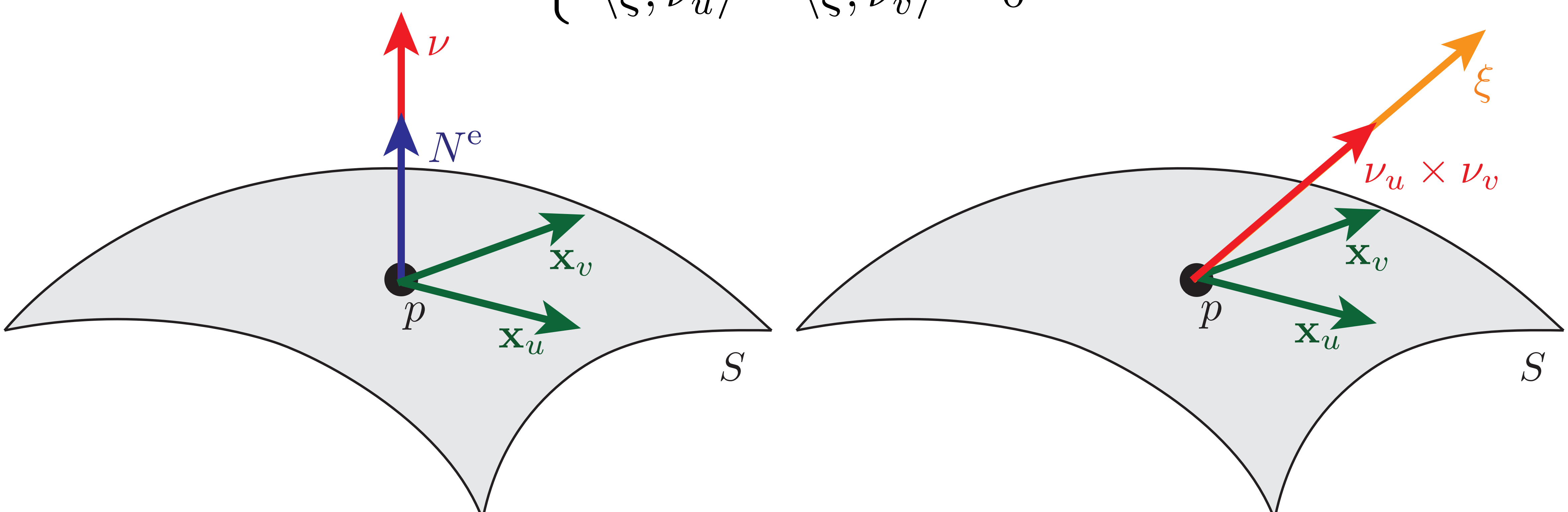
$$\nu = |K^{\mathrm{e}}|^{-1/4} N^{\mathrm{e}}$$

affine contravariant: $m(A(p)) = (A^{-1})^T(m(p))$

Affine normal vector

The affine normal vector ξ is implicitly defined by the equations

$$\begin{cases} \langle \nu, \xi \rangle = 1 \\ \langle \xi, \nu_u \rangle = \langle \xi, \nu_v \rangle = 0 \end{cases}$$



Vieira et al., 2016

Estimating affine-invariant structures on triangle meshes

Affine geometry background

Affine co-normal vector

$$\nu = |K^{\mathrm{e}}|^{-1/4} N^{\mathrm{e}}$$

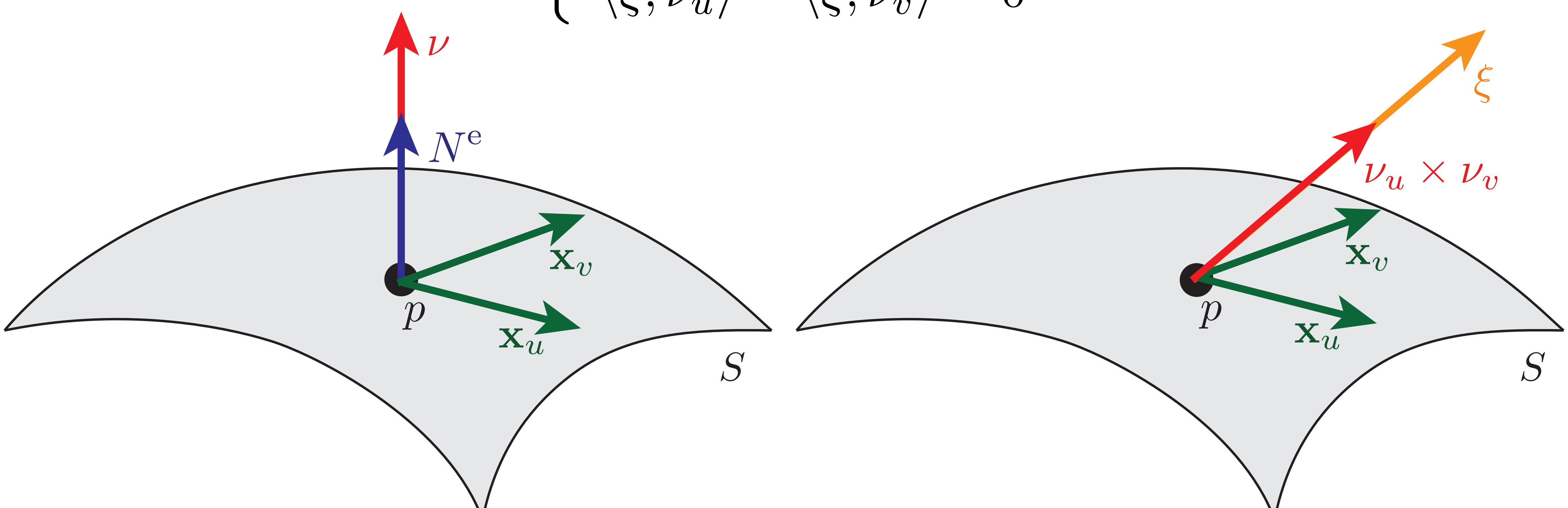
affine contravariant: $m(A(p)) = (A^{-1})^T(m(p))$

Affine normal vector

affine covariant: $m(A(p)) = A(m(p))$

The affine normal vector ξ is implicitly defined by the equations

$$\begin{cases} \langle \nu, \xi \rangle = 1 \\ \langle \xi, \nu_u \rangle = \langle \xi, \nu_v \rangle = 0 \end{cases}$$



Vieira et al., 2016

Estimating affine-invariant structures on triangle meshes

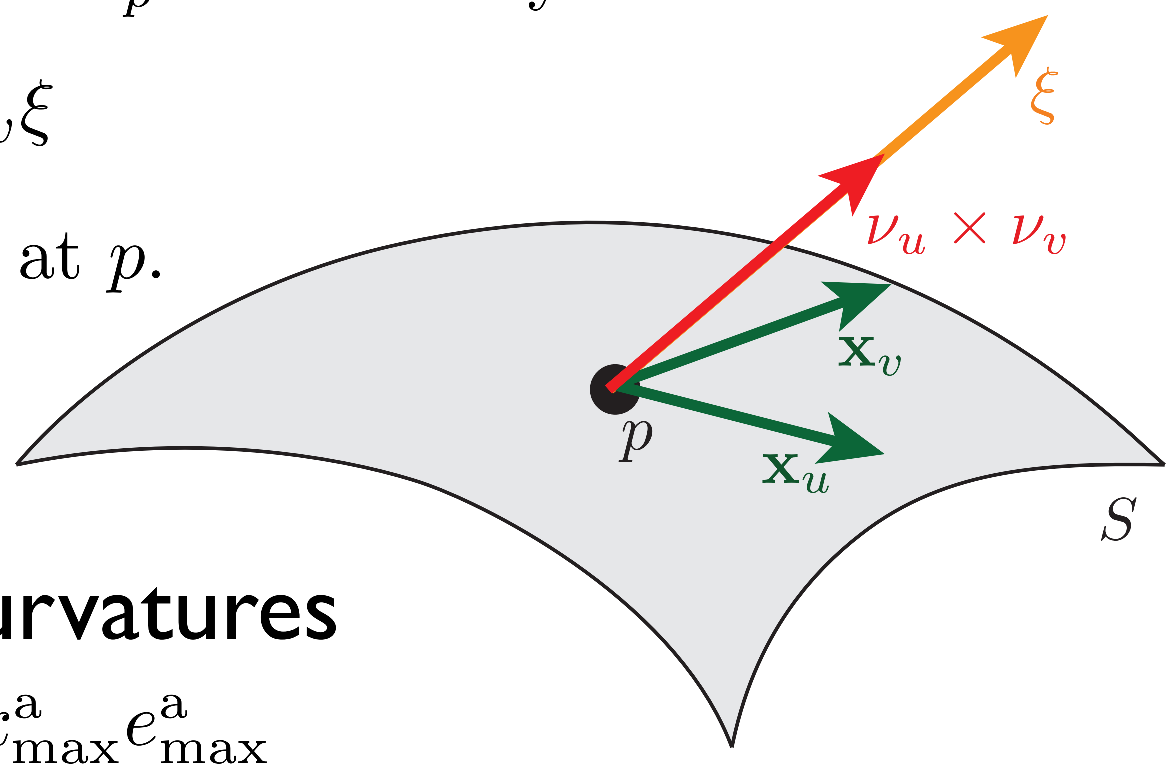
Affine geometry background

Affine shape operator

For each $p \in S$, the operator $\mathcal{S}^{\mathrm{a}} : T_pS \to T_pS$ defined by

$$\mathcal{S}^{\mathrm{a}} w = -D_w \xi$$

is called the affine shape operator of S at p.



Affine principal directions and curvatures

$$\mathcal{S}^{\mathrm{a}} e^{\mathrm{a}}_{\mathrm{max}} = k^{\mathrm{a}}_{\mathrm{max}} e^{\mathrm{a}}_{\mathrm{max}}$$

$$\mathcal{S}^{\mathrm{a}} e^{\mathrm{a}}_{\mathrm{min}} = k^{\mathrm{a}}_{\mathrm{min}} e^{\mathrm{a}}_{\mathrm{min}}$$

$$e^{\mathrm{a}}_{\mathrm{max}}, e^{\mathrm{a}}_{\mathrm{min}} \in T_pS$$

Affine Gaussian and mean curvatures

$$K^{\mathrm{a}} = \det(\mathcal{S}^{\mathrm{a}}) \qquad H^{\mathrm{a}} = -\tfrac{1}{2}\mathrm{tr}(\mathcal{S}^{\mathrm{a}})$$

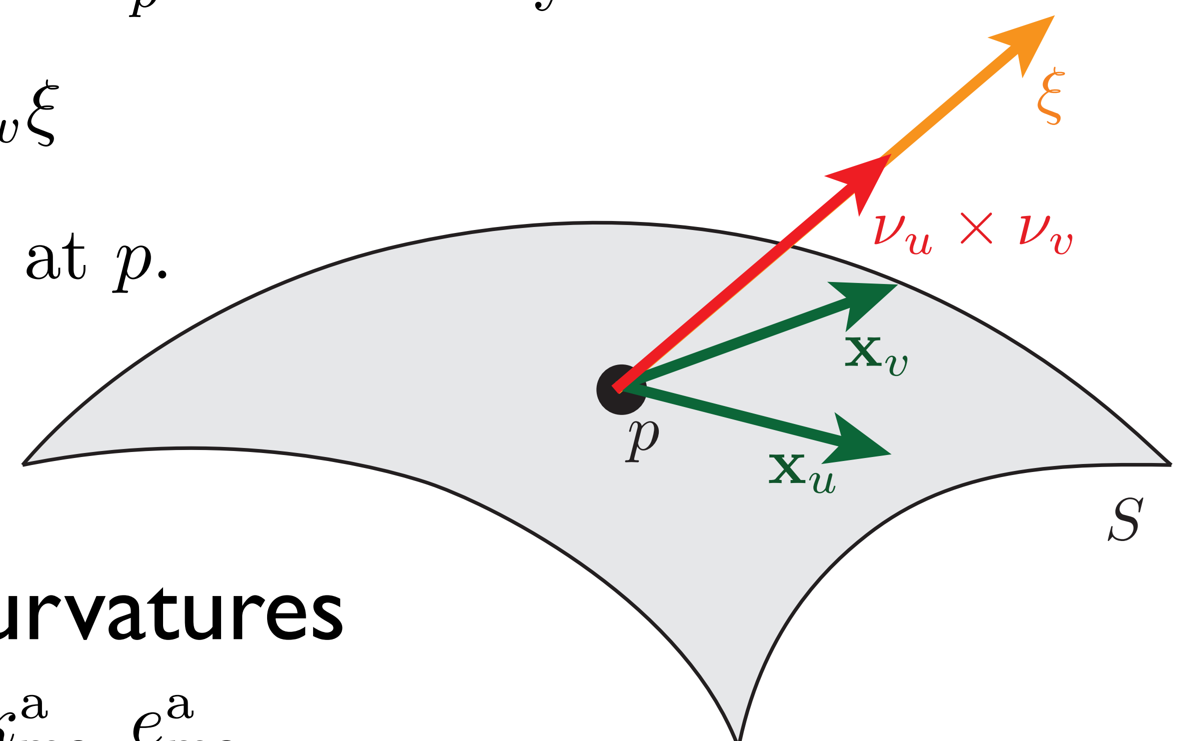
Affine geometry background

Affine shape operator

For each $p \in S$, the operator $\mathcal{S}^{\mathrm{a}} : T_pS \to T_pS$ defined by

$$\mathcal{S}^{\mathrm{a}} w = -D_w \xi$$

is called the affine shape operator of S at p.



Affine principal directions and curvatures

$$\mathcal{S}^{\mathrm{a}} e^{\mathrm{a}}_{\max} = k^{\mathrm{a}}_{\max} e^{\mathrm{a}}_{\max}$$

$$\mathcal{S}^{\mathrm{a}} e^{\mathrm{a}}_{\min} = k^{\mathrm{a}}_{\min} e^{\mathrm{a}}_{\min}$$

$$e^{\mathrm{a}}_{\max}, e^{\mathrm{a}}_{\min} \in T_pS$$

Affine Gaussian and mean curvatures

affine invariant: $m(A(p)) = m(p)$

$$K^{\mathrm{a}} = \det(\mathcal{S}^{\mathrm{a}}) \qquad H^{\mathrm{a}} = -\tfrac{1}{2}\mathrm{tr}(\mathcal{S}^{\mathrm{a}})$$

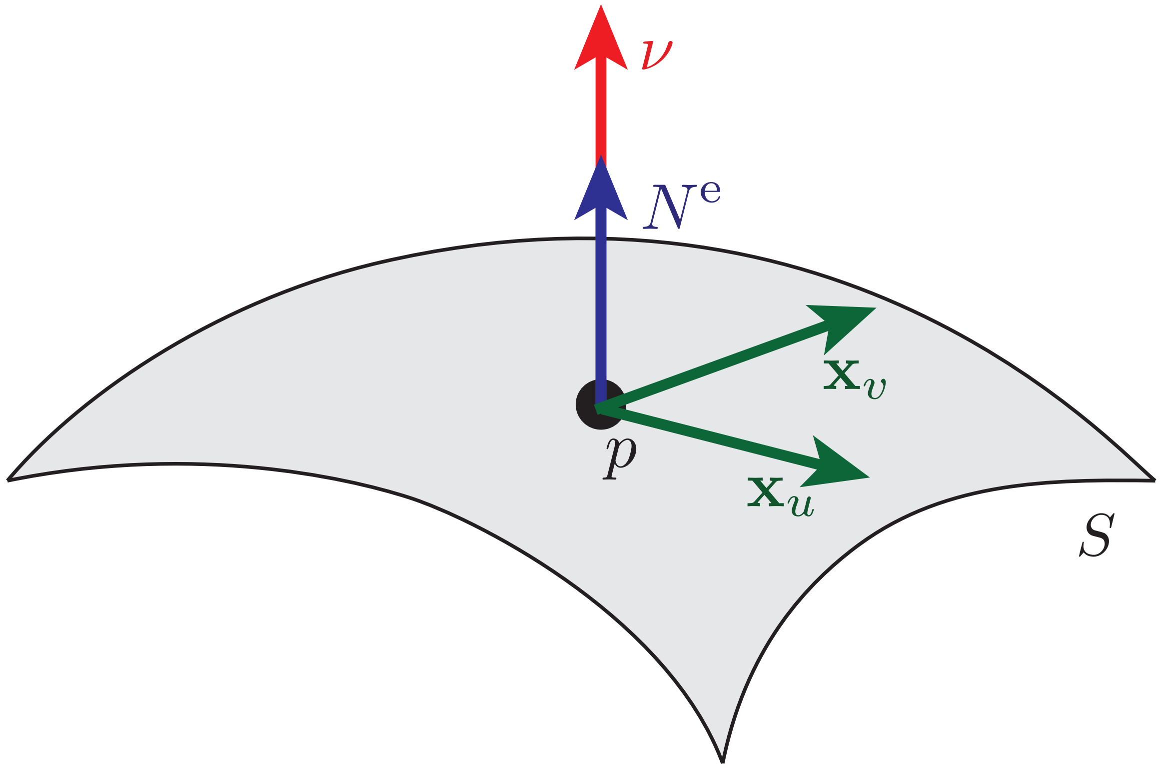
Outline

1. Affine geometry background
2. Least-squares based affine normal estimator
3. Affine shape operator estimator
4. Experiments
5. Limitations & Future Work

Least-squares based affine normal estimator

Affine co-normal vector

$$\nu = |K^{\mathrm{e}}|^{-1/4} N^{\mathrm{e}}$$



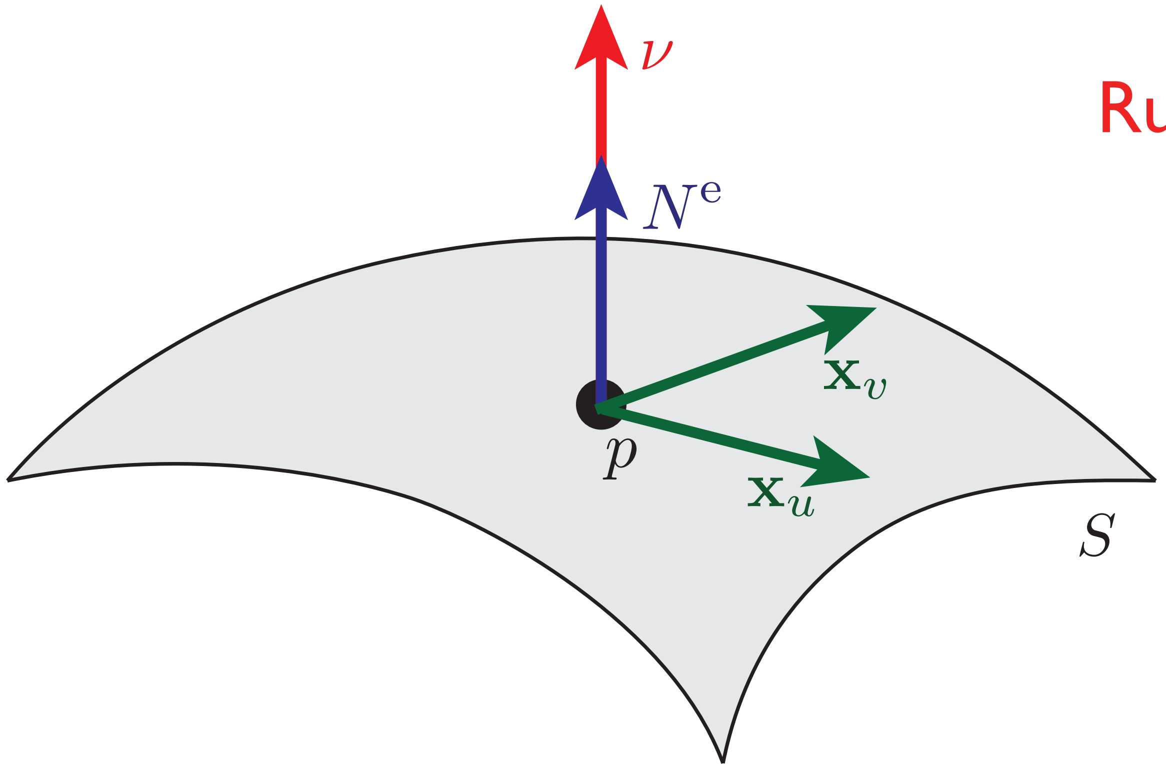
Least-squares based affine normal estimator

Affine co-normal vector

Estimated by averaging per-face normals

$$\nu = |K^{\mathrm{e}}|^{-1/4} N^{\mathrm{e}} \xrightarrow{\text{discretization}} \nu_i = |K_i^{\mathrm{e}}|^{-1/4} n_i$$

Estimated from Rusinkiewicz tensor



Vieira et al., 2016

Estimating affine-invariant structures on triangle meshes

Least-squares based affine normal estimator

Affine normal vector

$$\begin{cases} \langle \nu, \xi \rangle = 1 \\ \langle \xi, \nu_u \rangle = \langle \xi, \nu_v \rangle = 0 \end{cases}$$

Least-squares based affine normal estimator

Affine normal vector

$$\begin{cases} \langle \nu, \xi \rangle = 1 \\ \langle \xi, \nu_u \rangle = \langle \xi, \nu_v \rangle = 0 \end{cases}$$

Any directional derivative of the co-normal vector is orthogonal to the affine normal vector!

Least-squares based affine normal estimator

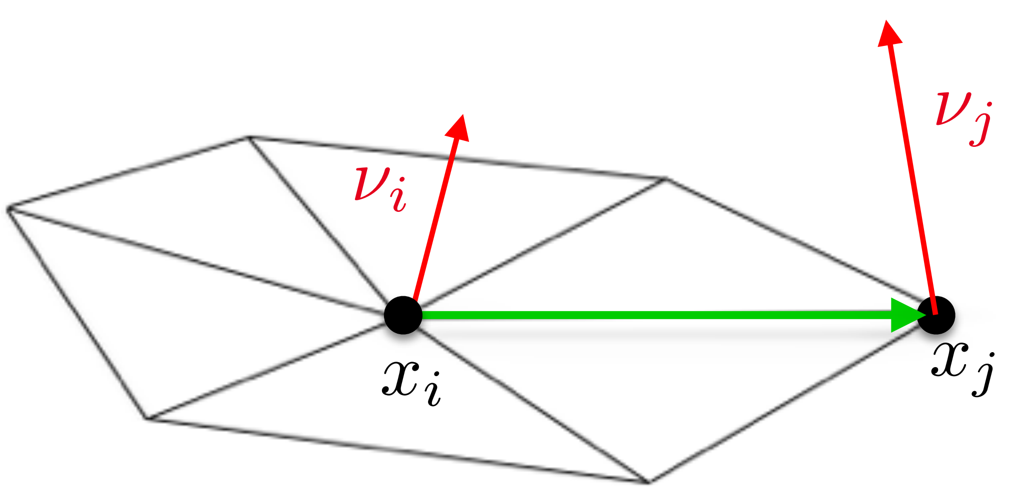
Affine normal vector

$$\begin{cases} \langle \nu, \xi \rangle = 1 \\ \langle \xi, \nu_u \rangle = \langle \xi, \nu_v \rangle = 0 \end{cases}$$

Any directional derivative of the co-normal vector is orthogonal to the affine normal vector!

Directional derivatives of the co-normal vector

$$D_j \nu_i \approx \frac{\nu_j - \nu_i}{\|x_j - x_i\|}$$



Vieira et al., 2016

Estimating affine-invariant structures on triangle meshes

Least-squares based affine normal estimator

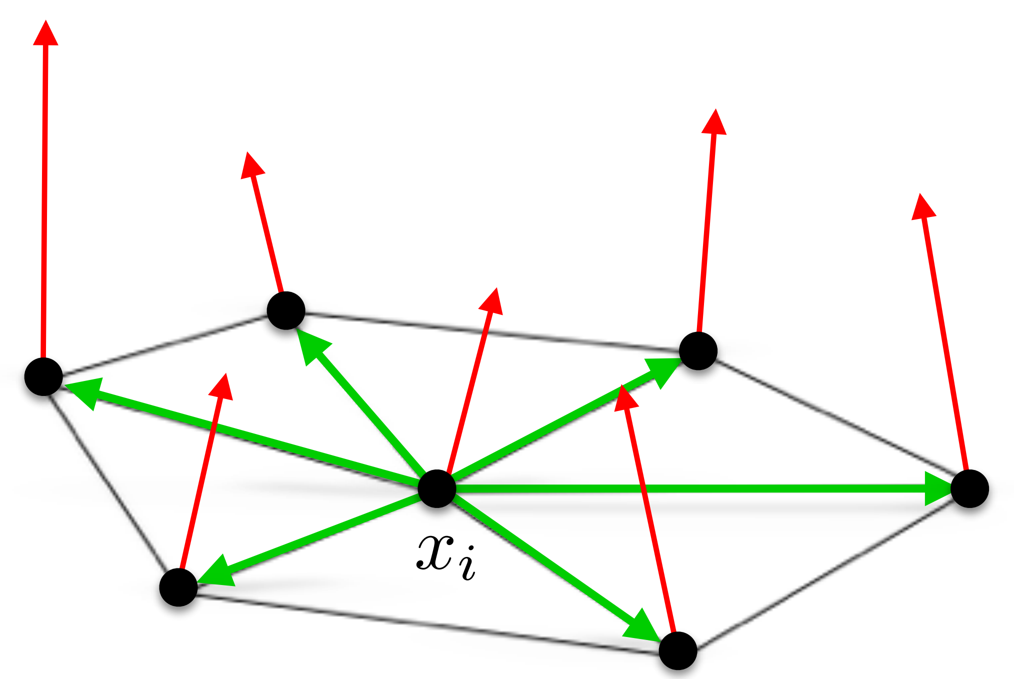
Affine normal vector

$$\begin{cases} \langle \nu, \xi \rangle = 1 \\ \langle \xi, \nu_u \rangle = \langle \xi, \nu_v \rangle = 0 \end{cases}$$

Any directional derivative of the co-normal vector is orthogonal to the affine normal vector!

Directional derivatives of the co-normal vector

$$D_j \nu_i \approx \frac{\nu_j - \nu_i}{\|x_j - x_i\|}$$



Vieira et al., 2016

Estimating affine-invariant structures on triangle meshes

Least-squares based affine normal estimator

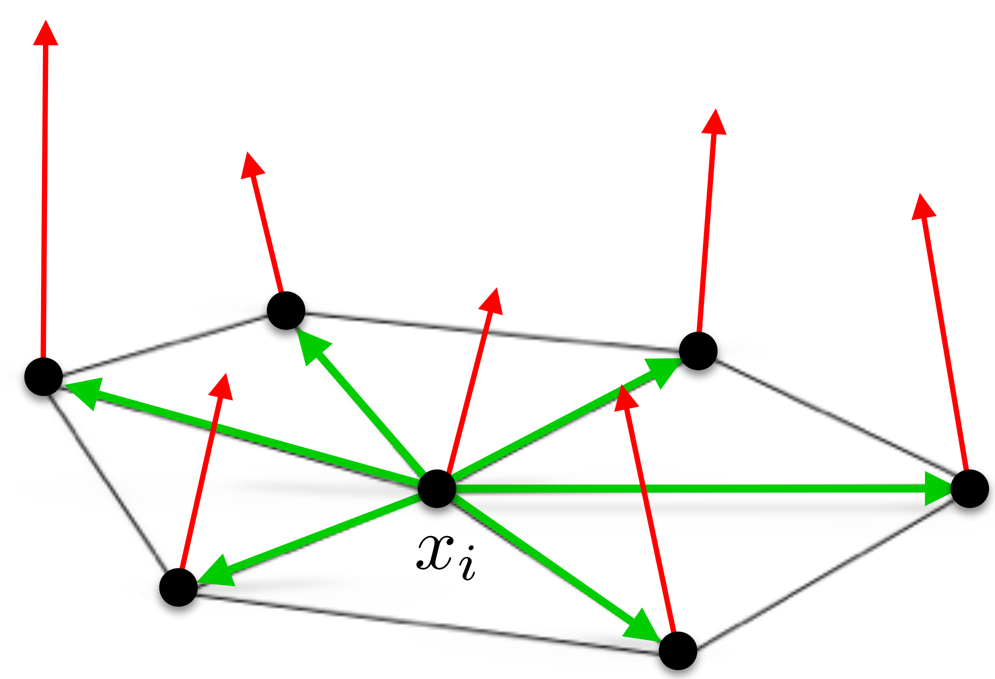
Affine normal vector

$$\begin{cases} \langle \nu, \xi \rangle = 1 \\ \langle \xi, \nu_u \rangle = \langle \xi, \nu_v \rangle = 0 \end{cases}$$

Any directional derivative of the co-normal vector is orthogonal to the affine normal vector!

Directional derivatives of the co-normal vector

$$D_j \nu_i \approx \frac{\nu_j - \nu_i}{\|x_j - x_i\|}$$



Estimated directional derivatives are prone to approximation error!

Least-squares based affine normal estimator

Affine normal vector

hard constraint

soft constraints

$$\begin{cases} \langle \nu, \xi \rangle = 1 \\ \langle \xi, \nu_u \rangle = \langle \xi, \nu_v \rangle = 0 \end{cases}$$

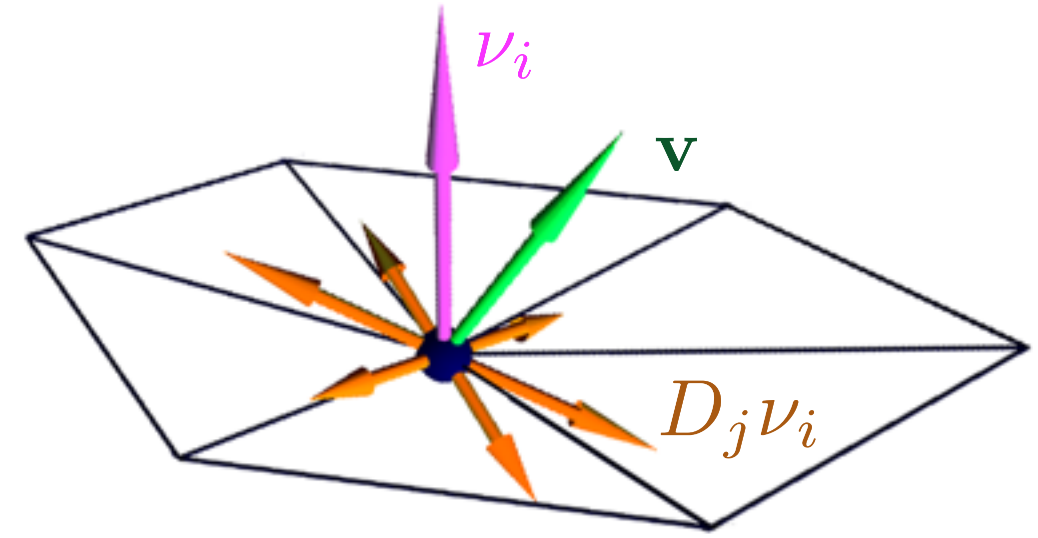
Least-squares based affine normal estimator

Affine normal vector

hard constraint

soft constraints

$$\begin{cases} \langle \nu, \xi \rangle = 1 \\ \langle \xi, \nu_u \rangle = \langle \xi, \nu_v \rangle = 0 \end{cases}$$



Least-squares minimization

Let $\mathbf{v} = (x, y, z)$, $\nu_i = (a, b, c)$ and $D_j\nu_i = (a_j, b_j, c_j)$.

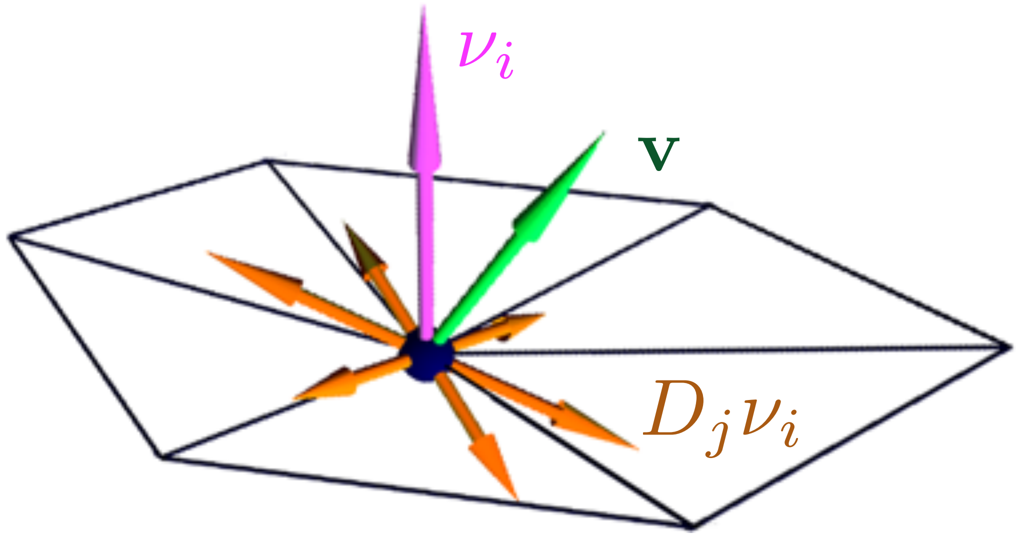
Least-squares based affine normal estimator

Affine normal vector

hard constraint

soft constraints

$$\begin{cases} \langle \nu, \xi \rangle = 1 \\ \langle \xi, \nu_u \rangle = \langle \xi, \nu_v \rangle = 0 \end{cases}$$



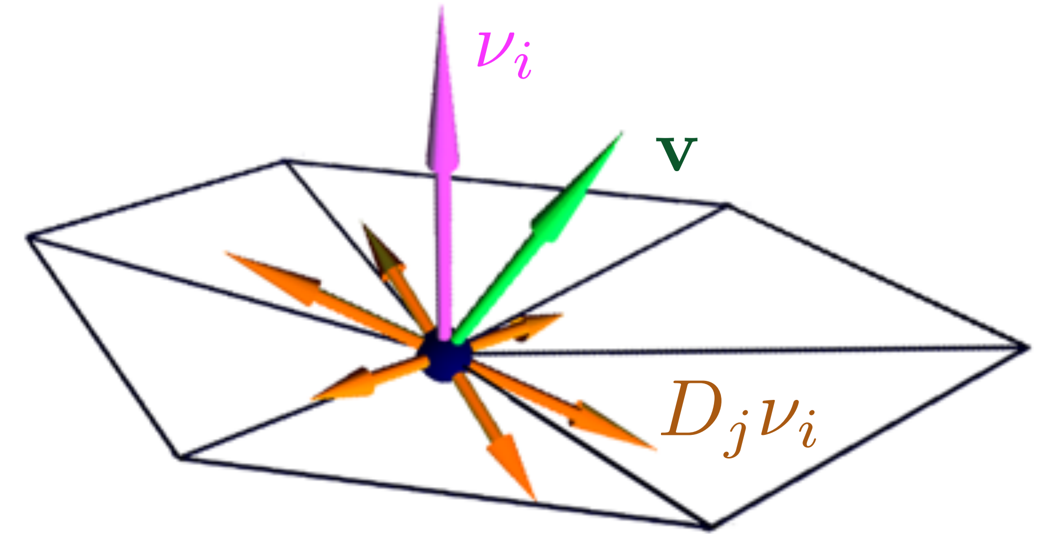
Least-squares minimization

Let $\mathbf{v} = (x, y, z)$, $\nu_i = (a, b, c)$ and $D_j\nu_i = (a_j, b_j, c_j)$.

The hard constraint $\langle \nu_i, \mathbf{v} \rangle = 1$ can be written as

$$x = \frac{1 - by - cz}{a}$$

Least-squares based affine normal estimator



Affine normal vector

hard constraint

soft constraints

$$\begin{cases} \langle \nu, \xi \rangle = 1 \\ \langle \xi, \nu_u \rangle = \langle \xi, \nu_v \rangle = 0 \end{cases}$$

Least-squares minimization

Let $\mathbf{v} = (x, y, z)$, $\nu_i = (a, b, c)$ and $D_j\nu_i = (a_j, b_j, c_j)$.

The hard constraint $\langle \nu_i, \mathbf{v} \rangle = 1$ can be written as

$$x = \frac{1 - by - cz}{a}$$

The soft orthogonality constraints can be minimized by least-squares:

$$\underset{y,z}{\text{minimize}} \sum_{j \in N_1(i)} w_j \left(\left(-\frac{b}{a} a_j + b_j \right) \cdot y + \left(-\frac{c}{a} a_j + c_j \right) \cdot z + \frac{a_j}{a} \right)^2$$

Vieira et al., 2016 *Estimating affine-invariant structures on triangle meshes*

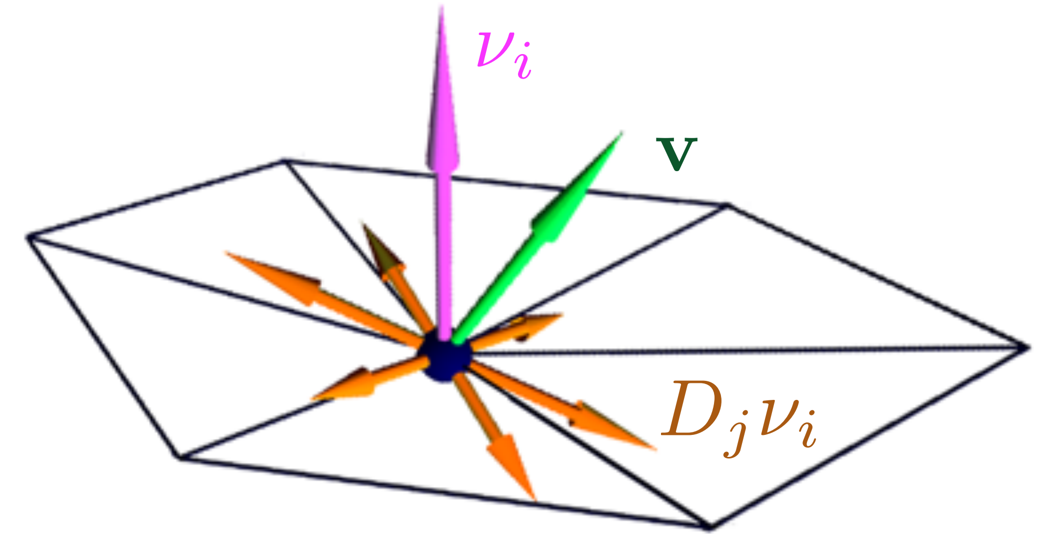
Least-squares based affine normal estimator

Affine normal vector

hard constraint

soft constraints

$$\begin{cases} \langle \nu, \xi \rangle = 1 \\ \langle \xi, \nu_u \rangle = \langle \xi, \nu_v \rangle = 0 \end{cases}$$



Least-squares minimization

Let $\mathbf{v} = (x, y, z)$, $\nu_i = (a, b, c)$ and $D_j \nu_i = (a_j, b_j, c_j)$.

The hard constraint $\langle \nu_i, \mathbf{v} \rangle = 1$ can be written as

$$x = \frac{1 - by - cz}{a}$$

sum of areas of adjacent triangles to the edge

The soft orthogonality constraints can be minimized by least-squares:

$$\underset{y,z}{\text{minimize}} \sum_{j \in N_1(i)} w_j \left(\left(-\frac{b}{a} a_j + b_j \right) \cdot y + \left(-\frac{c}{a} a_j + c_j \right) \cdot z + \frac{a_j}{a} \right)^2$$

Vieira et al., 2016 *Estimating affine-invariant structures on triangle meshes*

Outline

1. Affine geometry background
2. Least-squares based affine normal estimator
3. Affine shape operator estimator
4. Experiments
5. Limitations & Future Work

Affine shape operator estimator

Euclidean shape operator

$$\mathcal{S}^{\mathrm{e}}\colon T_pS \to T_pS$$

$$\mathcal{S}^{\mathrm{e}}w = -D_w n$$

Affine shape operator

$$\mathcal{S}^{\mathrm{a}}\colon TpS \to TpS$$

$$\mathcal{S}^{\mathrm{a}}w = -D_w \xi$$

Affine shape operator estimator

Euclidean shape operator

$$\mathcal{S}^{\mathrm{e}} \colon T_pS \to T_pS$$

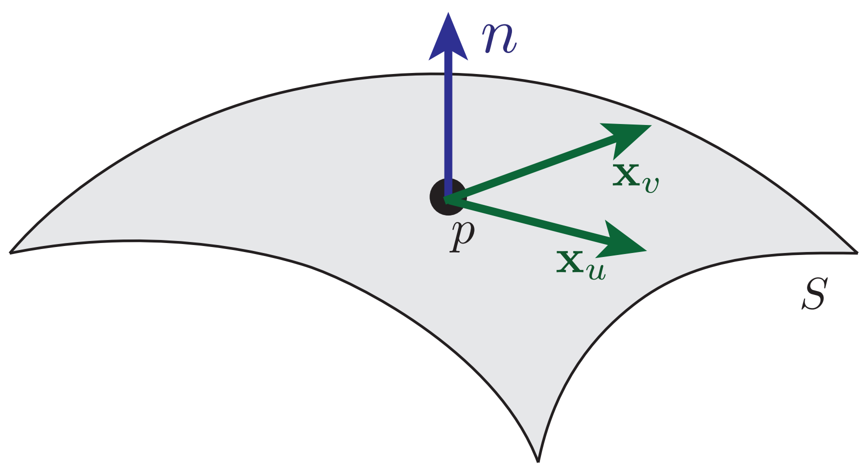
$$\mathcal{S}^{\mathrm{e}} w = -D_w n$$

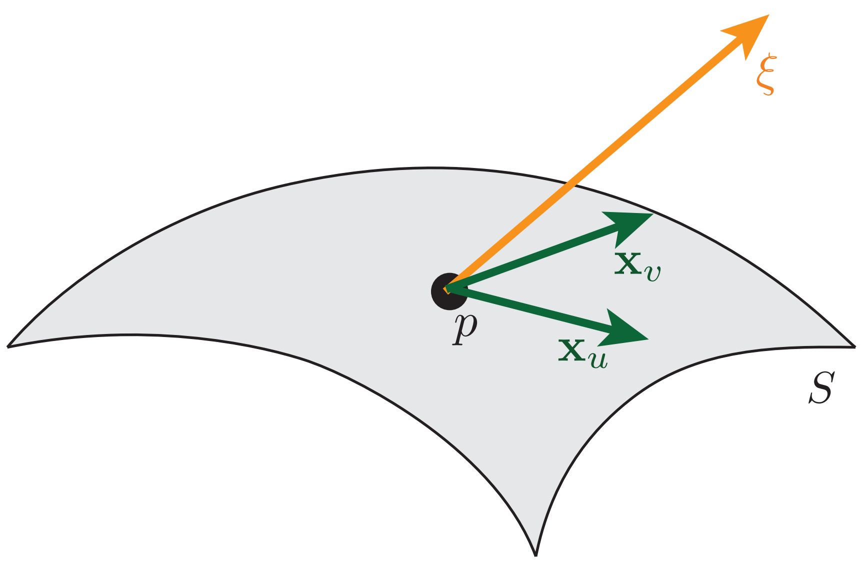
Affine shape operator

$$\mathcal{S}^{\mathrm{a}} \colon TpS \to TpS$$

$$\mathcal{S}^{\mathrm{a}} w = -D_w \xi$$

Derivatives of vector fields on surfaces





Vieira et al., 2016

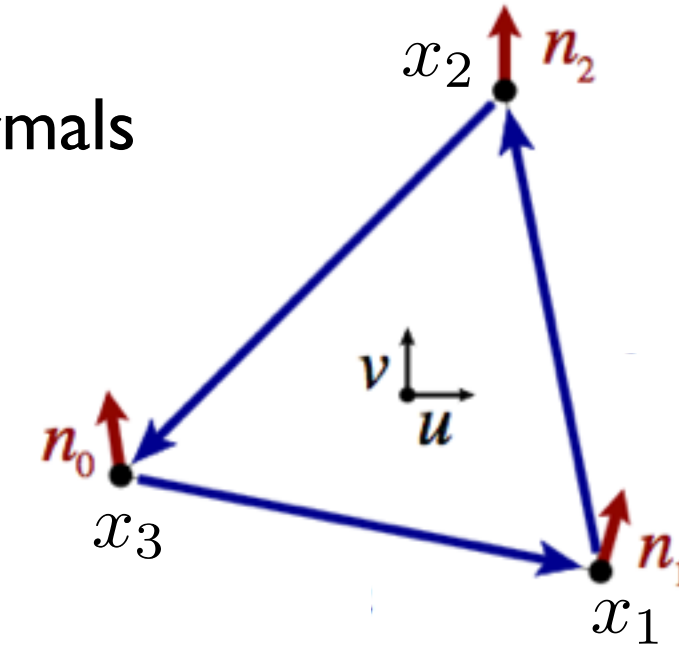
Estimating affine-invariant structures on triangle meshes

Back to the Euclidean Rusinkiewicz tensor…

Rusinkiewicz tensor

1. Estimate a normal vector field by averaging face normals

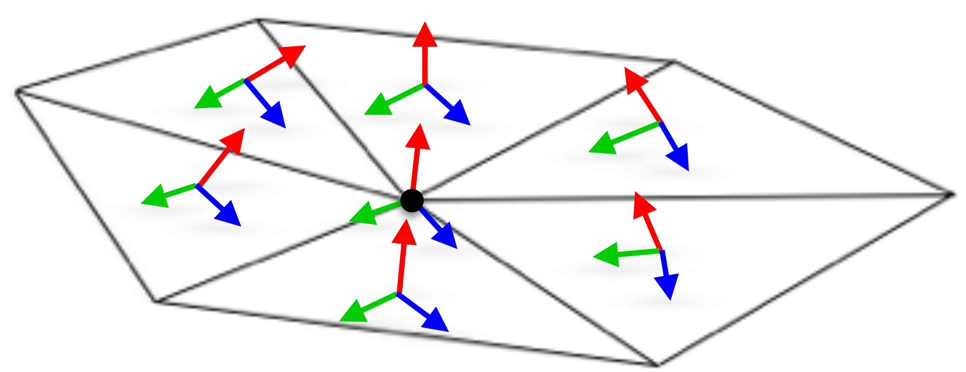
2. Discretize the shape operator per face using finite differences



$$-\mathcal{S}^{\mathrm{e}}[e_{ij}] = \begin{pmatrix} \langle n_i - n_j, \mathbf{e}_1 \rangle \\ \\ \langle n_i - n_j, \mathbf{e}_2 \rangle \end{pmatrix}, \quad e_{ij} = x_i - x_j$$

3. Apply least-squares to find shape operators per face

4. To find the shape operator at each vertex, average adjacent face operators after changing coordinates to a common coordinate system.



only normal vectors are required to discretize the shape operator!

Vieira et al., 2016

Estimating affine-invariant structures on triangle meshes

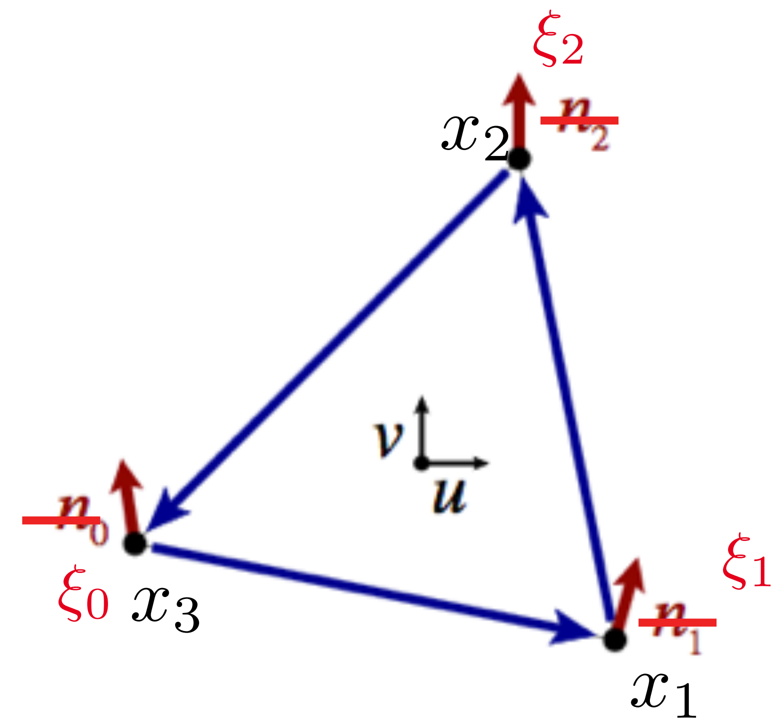
Rusinkiewicz tensor + affine geometry

Rusinkiewicz tensor

1. Estimate an affine normal vector field

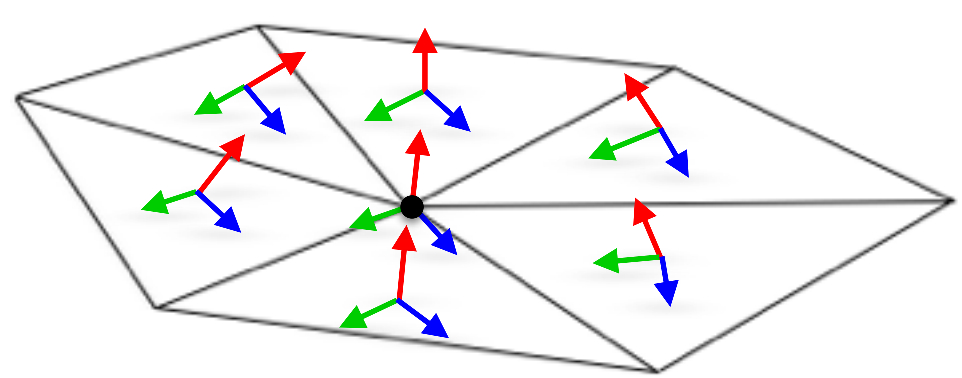
2. Discretize the shape operator per face using finite differences

$$-\mathcal{S}^{\mathrm{e}}[e_{ij}] = \begin{pmatrix} \langle \overset{\xi_i - \xi_j}{\cancel{n_i - n_j}}, \mathbf{e}_1 \rangle \\ \\ \langle \underset{\xi_i - \xi_j}{\cancel{n_i - n_j}}, \mathbf{e}_2 \rangle \end{pmatrix}, \quad e_{ij} = x_i - x_j$$



3. Apply least-squares to find shape operators per face

4. To find the shape operator at each vertex, average adjacent face operators after changing coordinates to a common coordinate system.



Vieira et al., 2016 *Estimating affine-invariant structures on triangle meshes*

Affine curvatures estimators

Affine principal directions and curvatures

$$\mathcal{S}^{\mathrm{a}} e^{\mathrm{a}}_{\max} = k^{\mathrm{a}}_{\max} e^{\mathrm{a}}_{\max}$$
$$\mathcal{S}^{\mathrm{a}} e^{\mathrm{a}}_{\min} = k^{\mathrm{a}}_{\min} e^{\mathrm{a}}_{\min}$$
$$e^{\mathrm{a}}_{\max}, e^{\mathrm{a}}_{\min} \in T_p S$$

Affine Gaussian and mean curvatures

$$K^{\mathrm{a}} = \det(\mathcal{S}^{\mathrm{a}}) \qquad H^{\mathrm{a}} = -\tfrac{1}{2}\mathrm{tr}(\mathcal{S}^{\mathrm{a}})$$

Affine curvatures estimators

Affine principal directions and curvatures

$$\mathcal{S}^{\mathrm{a}} e^{\mathrm{a}}_{\max} = k^{\mathrm{a}}_{\max} e^{\mathrm{a}}_{\max}$$

$$\mathcal{S}^{\mathrm{a}} e^{\mathrm{a}}_{\min} = k^{\mathrm{a}}_{\min} e^{\mathrm{a}}_{\min}$$

$$e^{\mathrm{a}}_{\max}, e^{\mathrm{a}}_{\min} \in T_p S$$

diagonalize estimated $\mathcal{S}^{\mathrm{a}}$

Affine Gaussian and mean curvatures

$$K^{\mathrm{a}} = \det(\mathcal{S}^{\mathrm{a}}) \qquad H^{\mathrm{a}} = -\tfrac{1}{2}\mathrm{tr}(\mathcal{S}^{\mathrm{a}})$$

straightforward from estimated $\mathcal{S}^{\mathrm{a}}$

Experiments

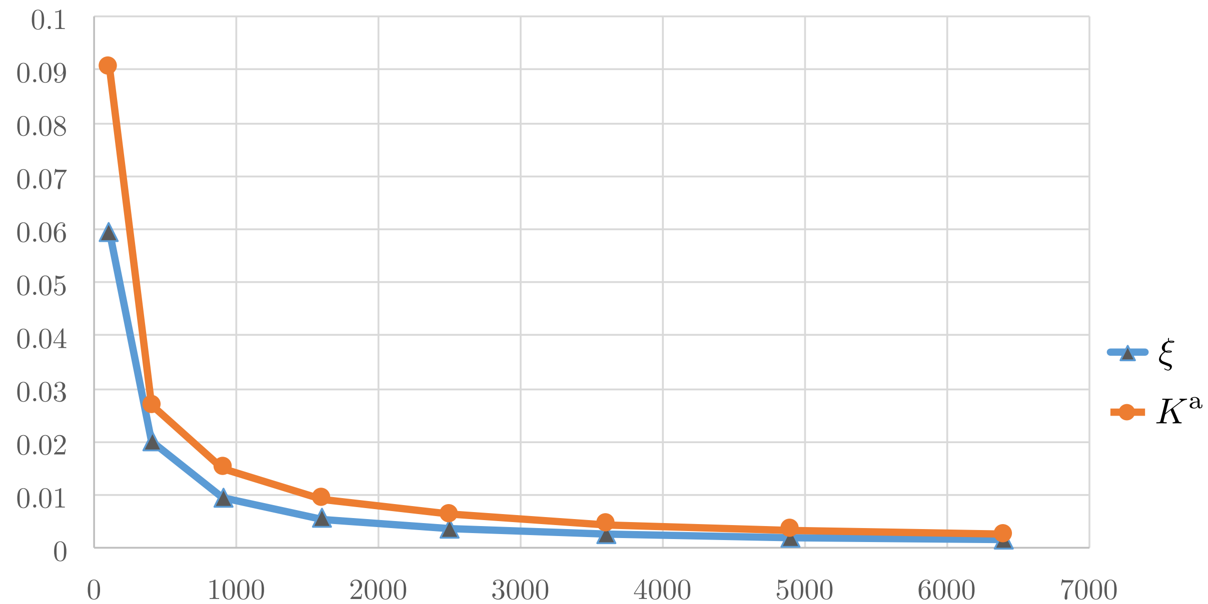
Convergence and scalability

Meshes uniformly tessellated from analytical models at different resolutions:

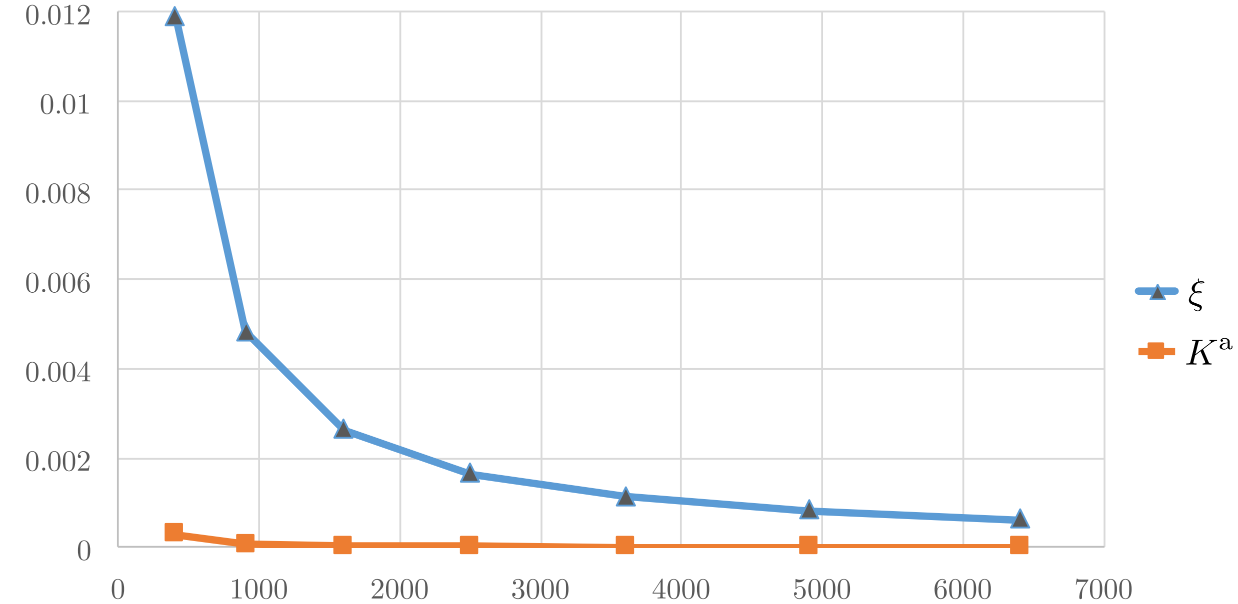
Model	**Equation**	**Range for u**	**Range for v**
Unit sphere	$\mathbf{x}(u,v) = (\cos(u)\sin(v),\ \sin(u)\sin(v),\ \cos(v)),$	$-\pi \leq u < \pi,$	$0 \leq v < \pi$
Ellipsoid	$\mathbf{x}(u,v) = (\frac{1}{2}\cos(u)\sin(v),\ \sin(u)\sin(v), 2\cos(v)),$	$-\pi \leq u < \pi,$	$0 \leq v < \pi$
Paraboloid	$\mathbf{x}(u,v) = (u, v, \frac{1}{2}u^2 + \frac{1}{2}v^2),$	$-2 \leq u \leq 2,$	$-2 \leq v \leq 2$
Torus	$\mathbf{x}(u,v) = (\cos(u)\sin(v),\ \sin(u)\sin(v),\ \cos(v)),$	$-\pi \leq u < \pi,$	$-\pi \leq v < \pi$
Enneper's surface	$\mathbf{x}(u,v) = (u - \frac{1}{3}u^3 + uv^2, v - \frac{1}{3}v^3 + vu^2, u^2 - v^2),$	$-1 \leq u < 1,$	$-1 \leq v < 1$

Comparison with exact measures from analytical models

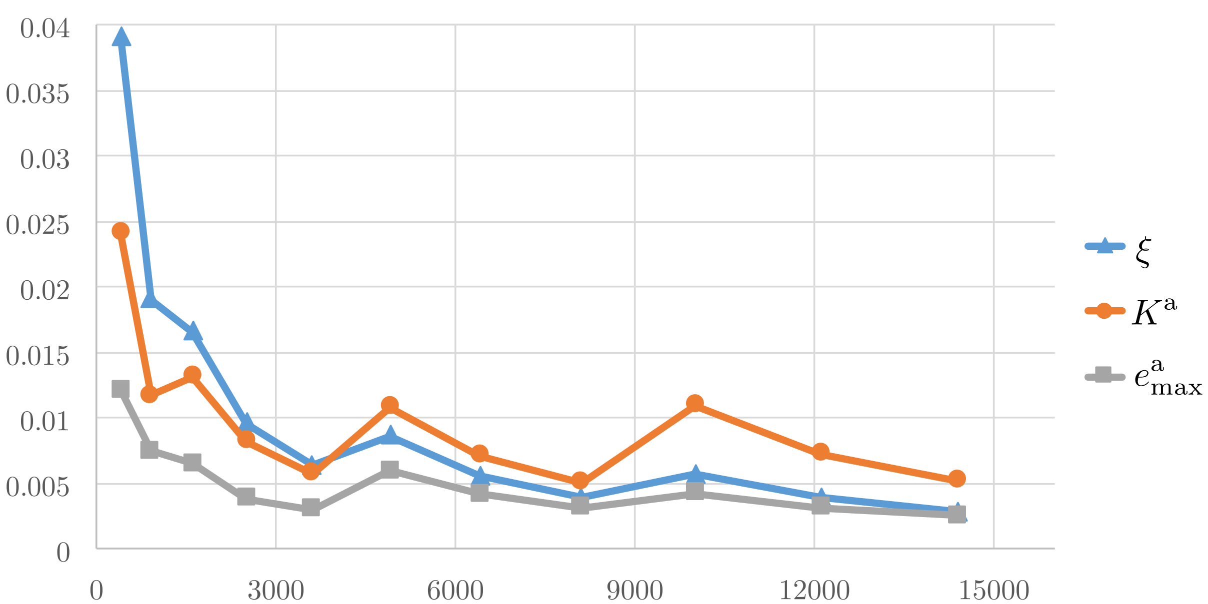
Average relative error as a measure of accuracy



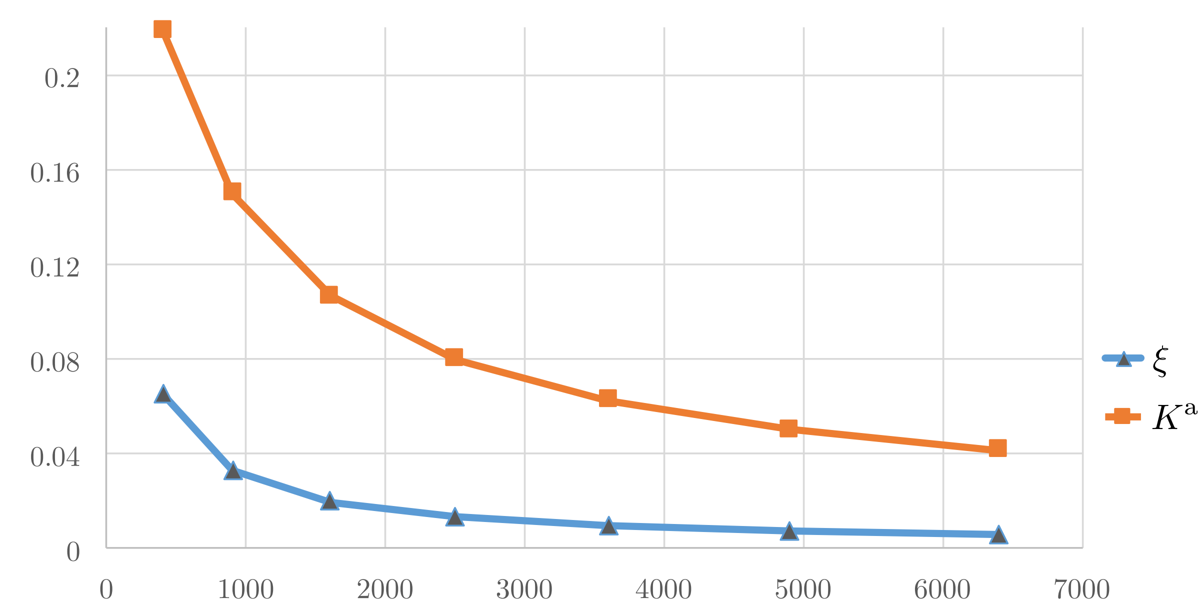
Sphere



Paraboloid

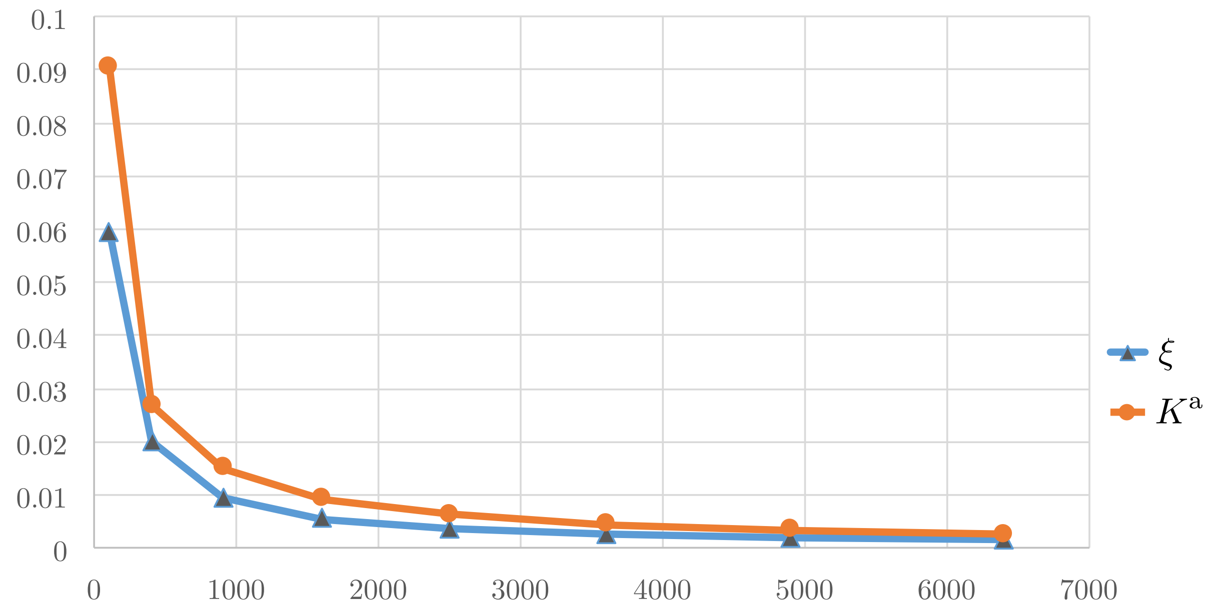


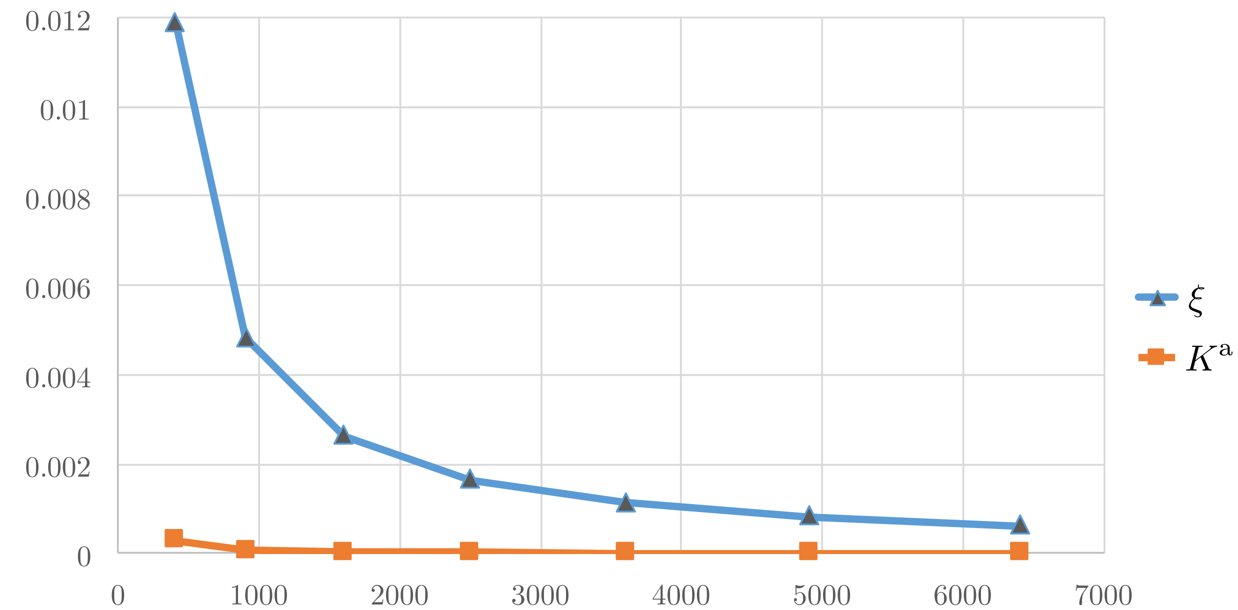
Torus



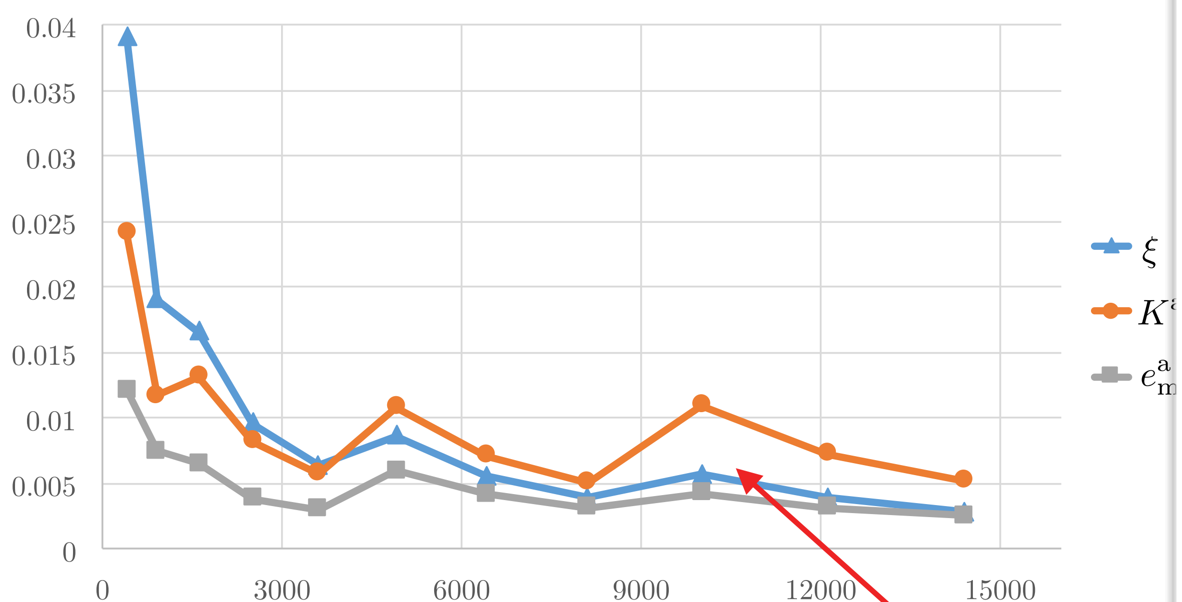
Ellipsoid

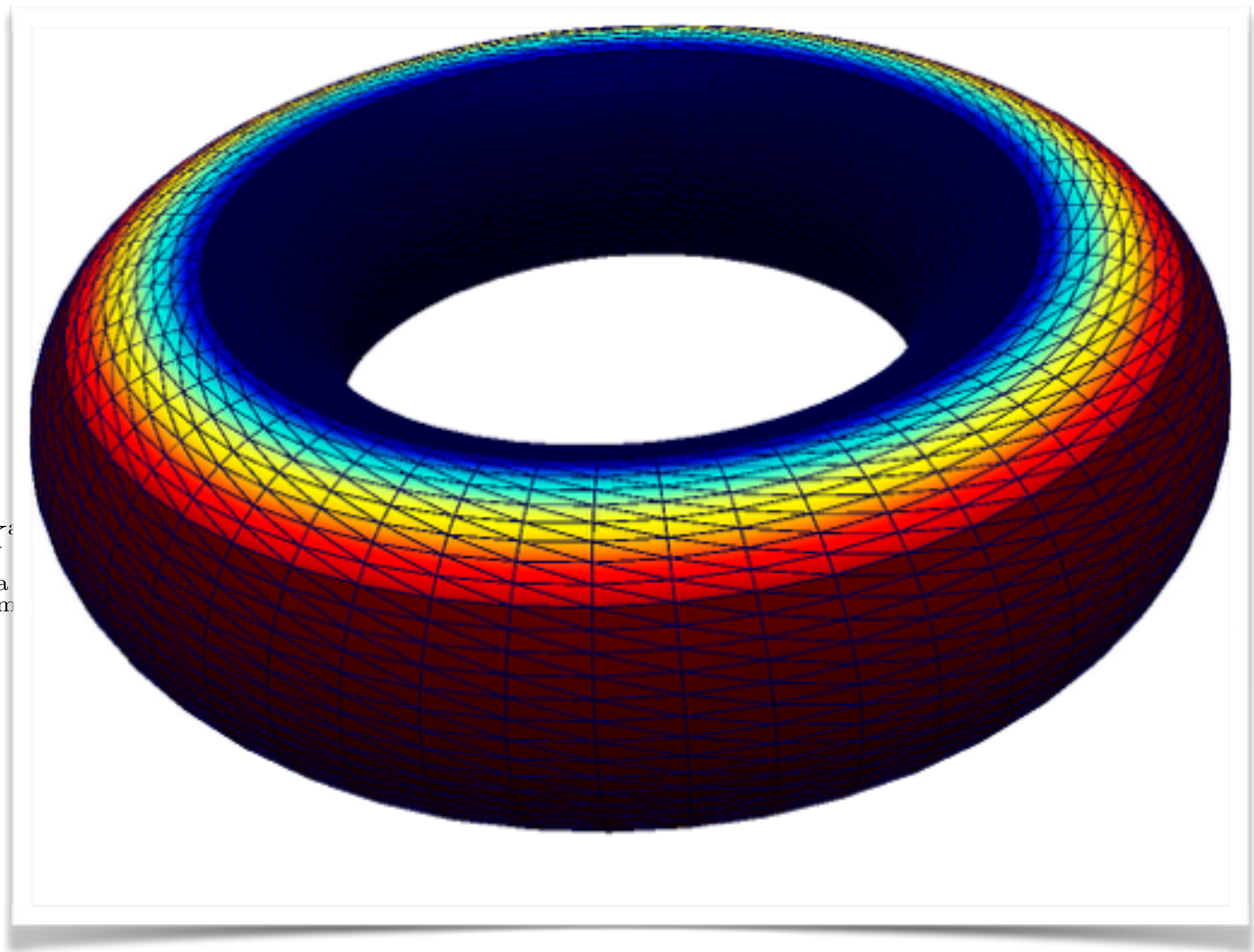
Vieira et al., 2016 *Estimating affine-invariant structures on triangle meshes*





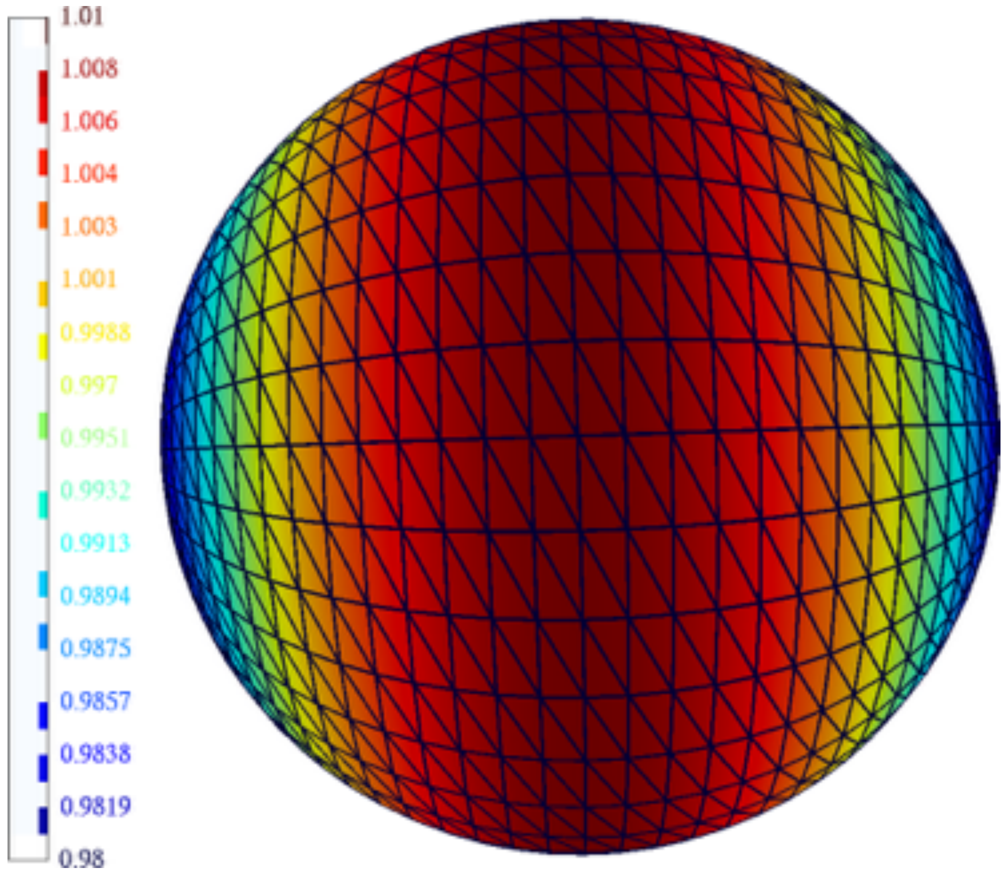
Sphere





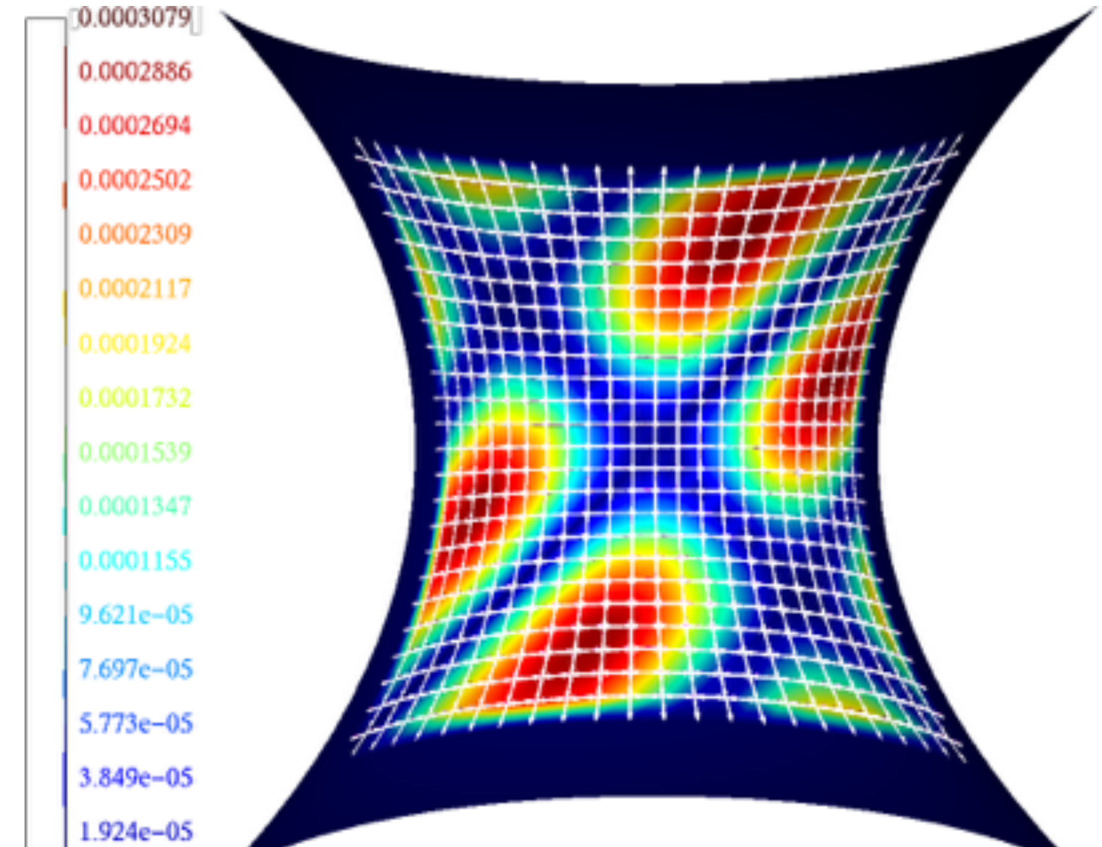
Torus

due to sampling on low $|K^{\mathrm{e}}|$ regions

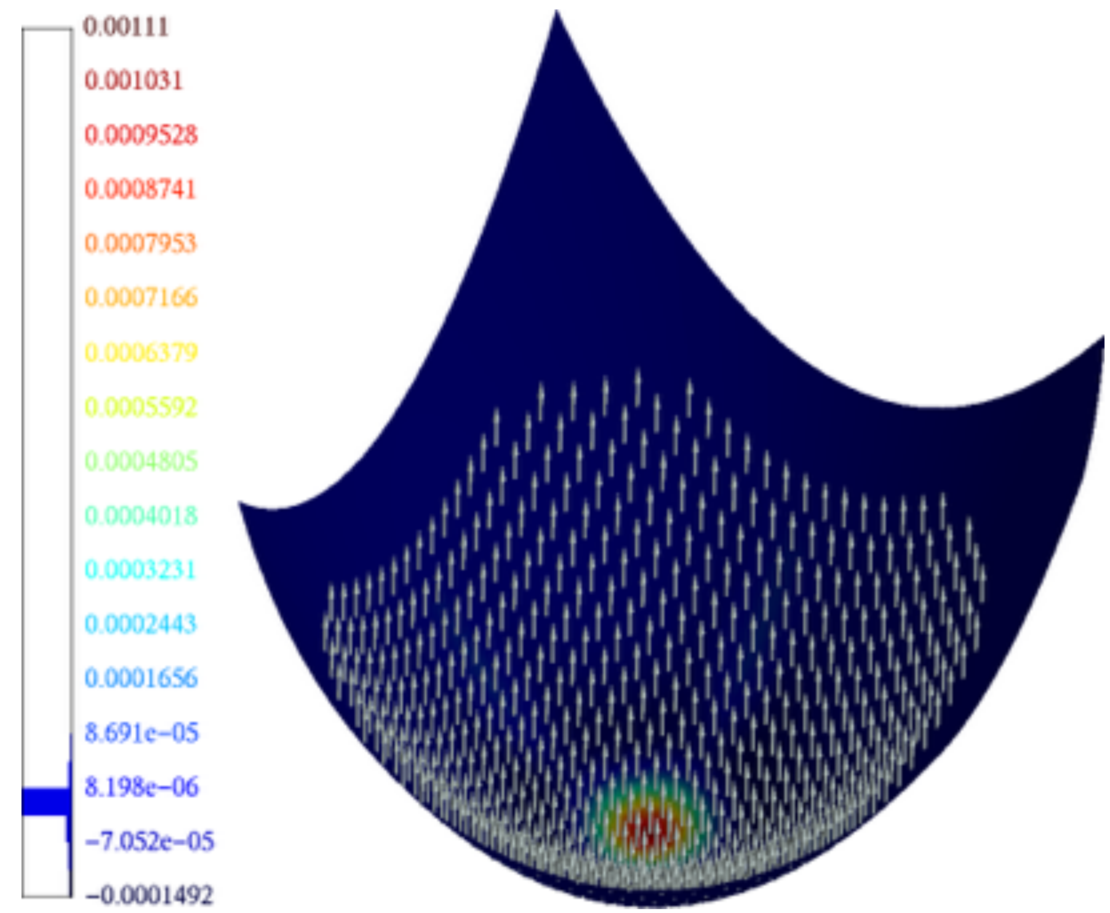


K^{a}

$e^{\mathrm{a}}_{\mathrm{min}}$



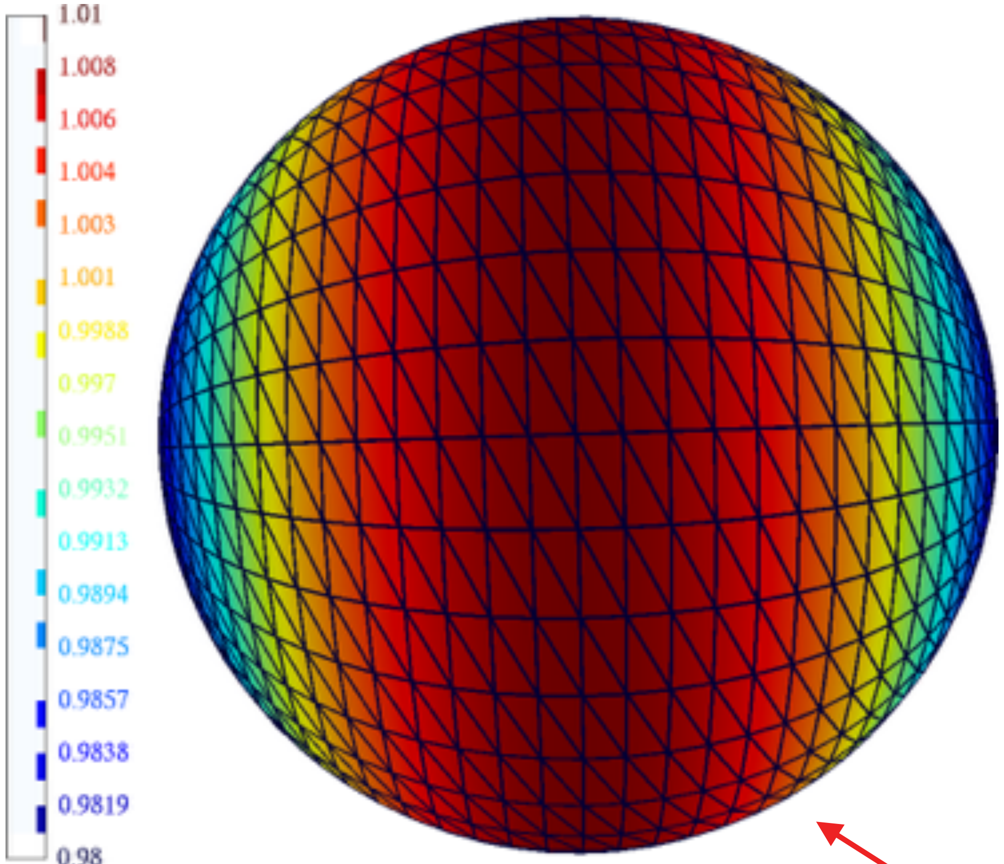
H^{a}



K^{a} and ξ

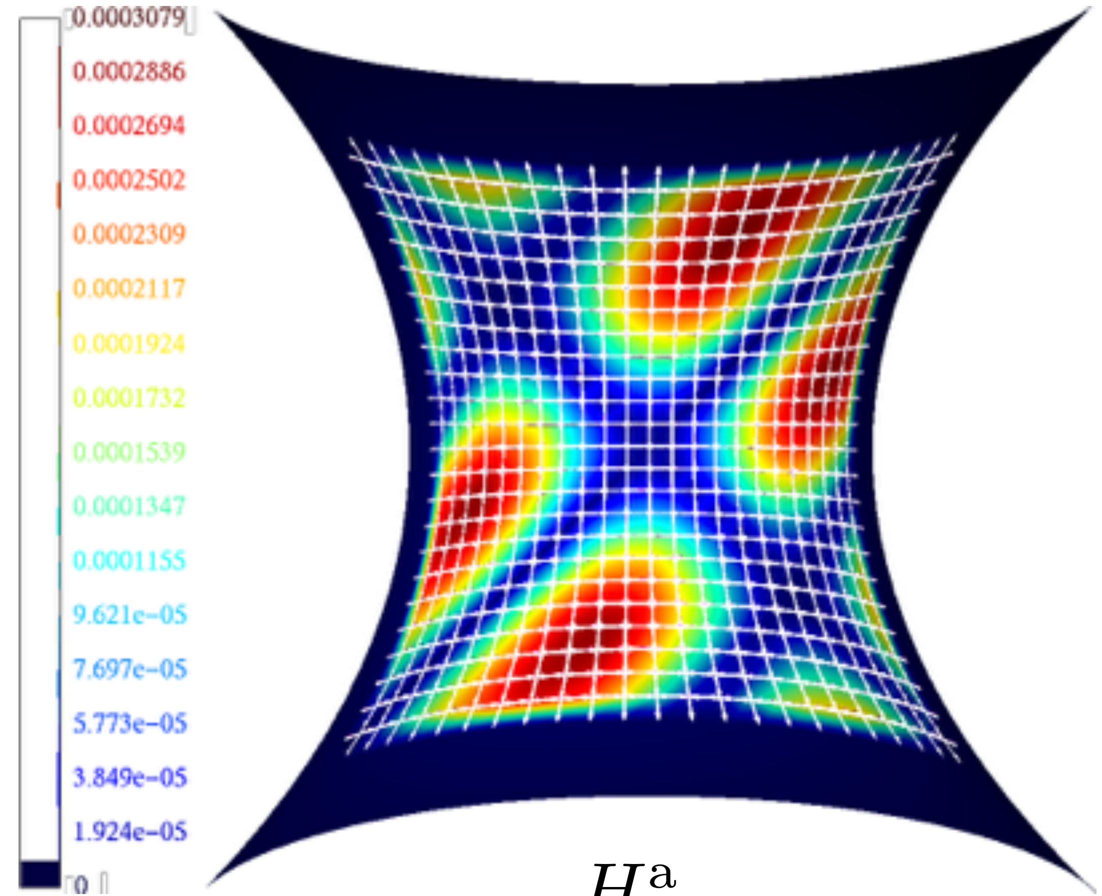
Vieira et al., 2016

Estimating affine-invariant structures on triangle meshes



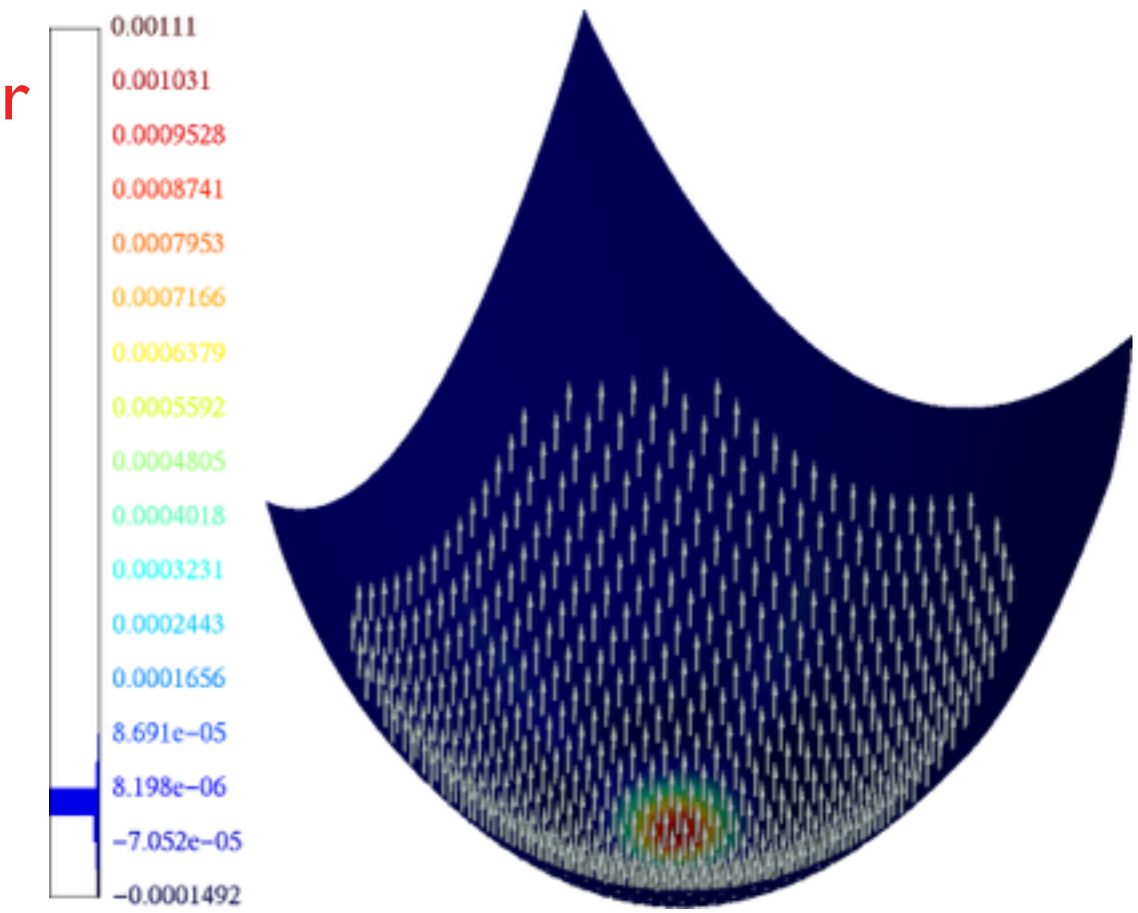
K^{a}

non-uniform color distribution



H^{a}

$e^{\mathrm{a}}_{\min}$

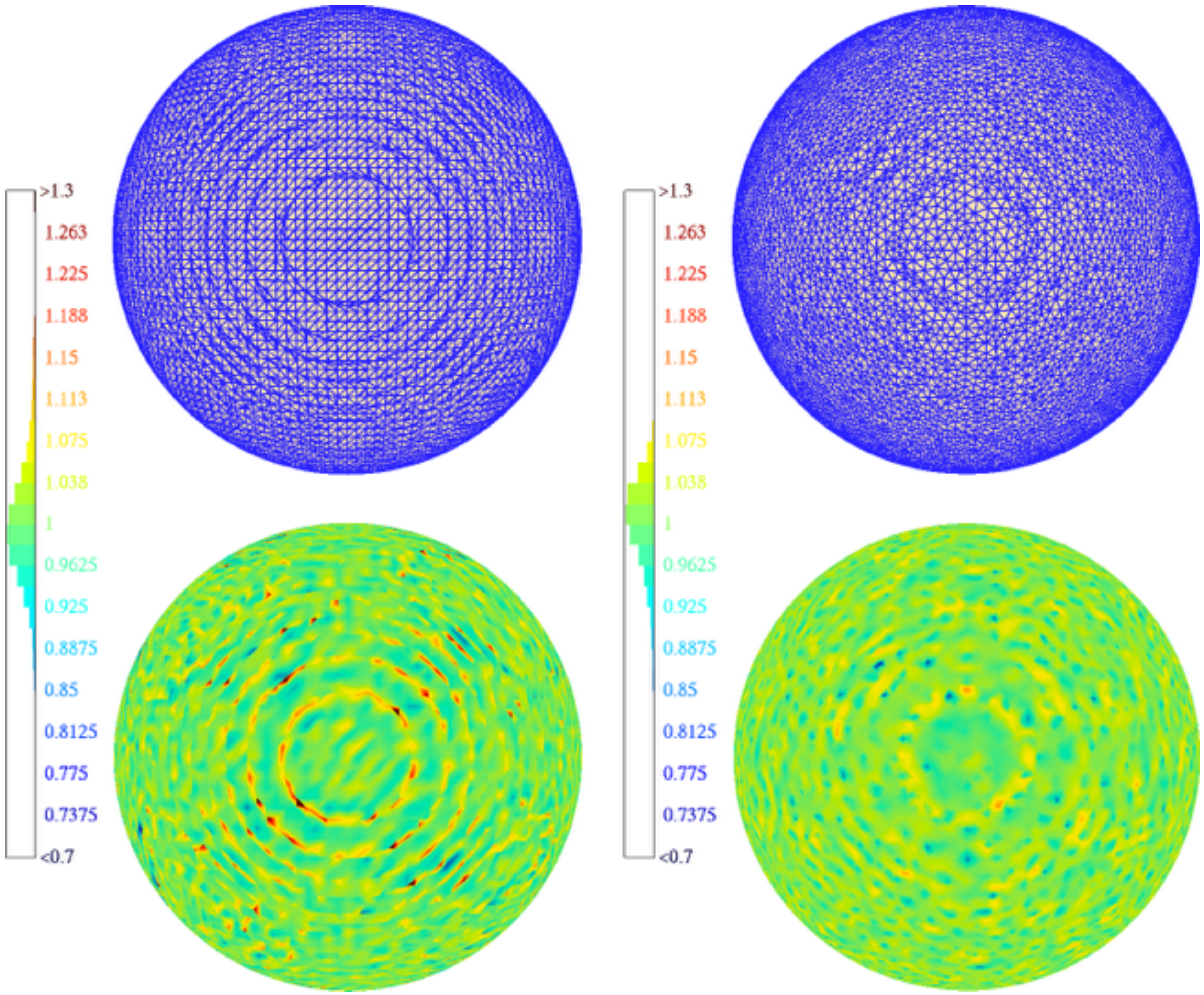


K^{a} and ξ

Vieira et al., 2016

Estimating affine-invariant structures on triangle meshes

Mesh regularity



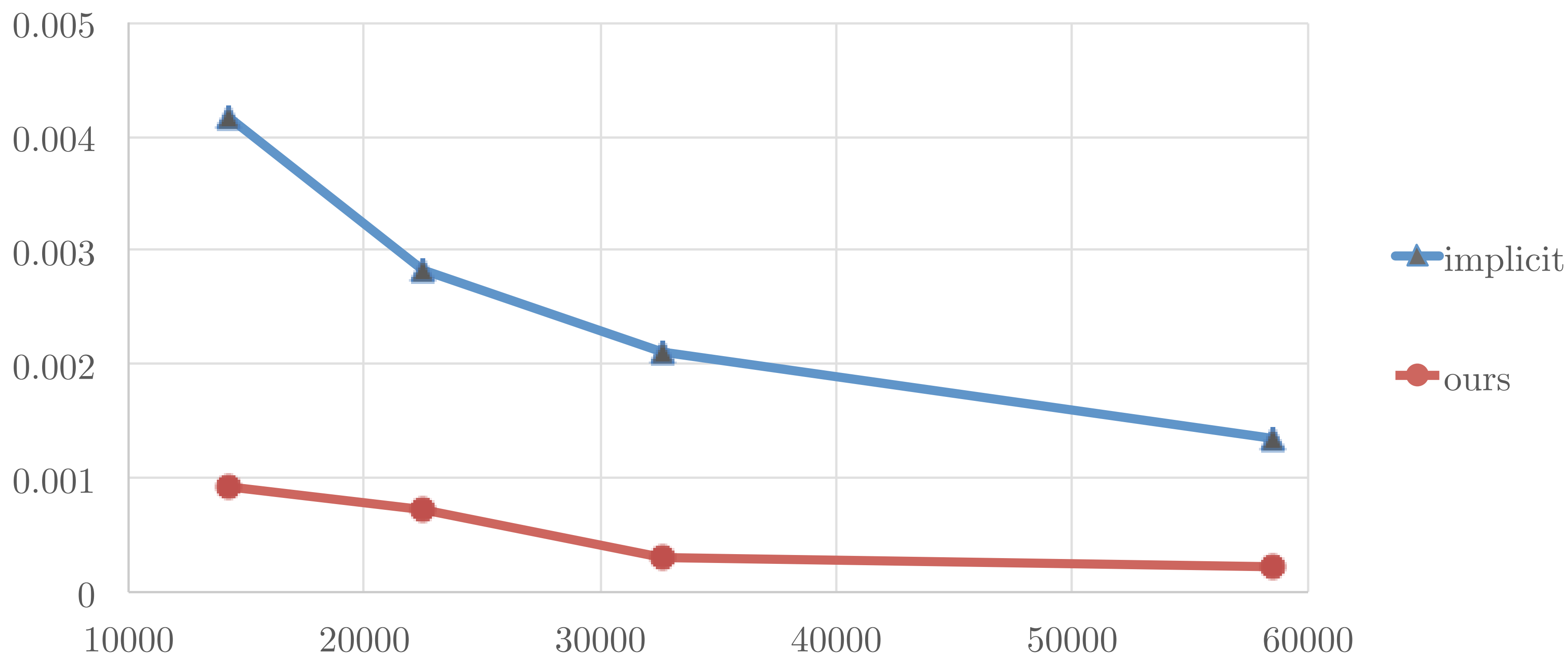
K^{a}, non-smoothed $\quad$ K^{a}, smoothed

Vieira et al., 2016 *Estimating affine-invariant structures on triangle meshes*

Comparison with Andrade and Lewiner (2012)

Not so fair: explicit vs. implicit: mean absolute error for meshes with 14.000 vertices.

Model	**Implicit**	**Ours**
Sphere	0.0041	0.0012
Ellipsoid	0.0035	0.0335
Paraboloid	0	8.96×10^{-7}
Torus	1.5265	0.0462



K^{a} error on the sphere

Vieira et al., 2016

Estimating affine-invariant structures on triangle meshes

Invariance

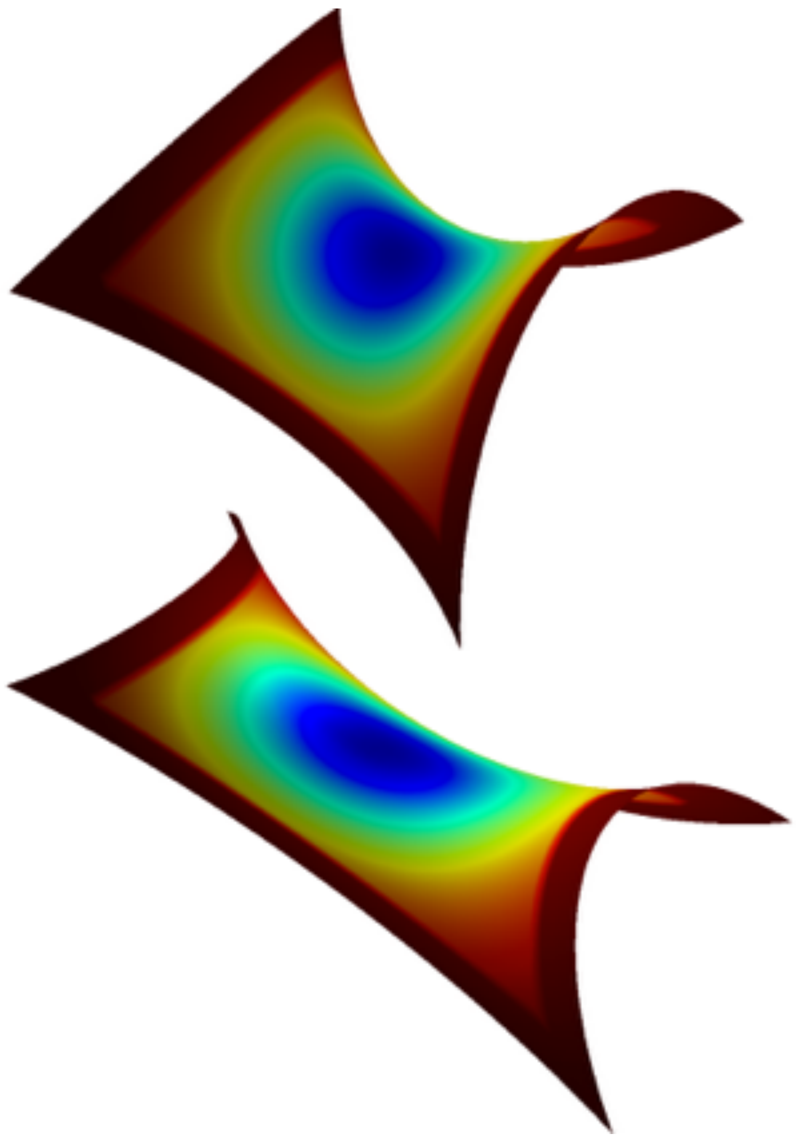
Experiments on equi-affine transformations with increasing spectral radius

Average relative covariance error:

$$E_{\xi}(\rho) = \sum_{i=1}^{n} \frac{\|\xi_i(A_\rho(\mathcal{M})) - A_\rho(\xi_i(\mathcal{M}))\|}{\|A_\rho(\xi_i(\mathcal{M}))\|}$$

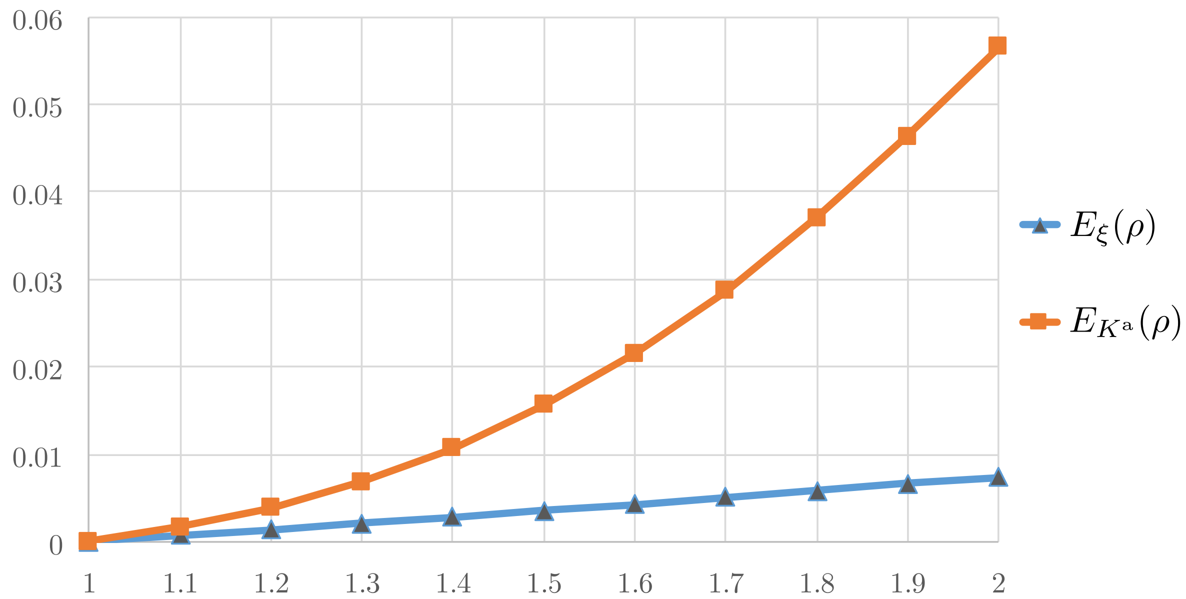
Average relative error:

$$E_{K^{\mathrm{a}}}(\rho) = \sum_{i=1}^{n} \frac{|K_i^{\mathrm{a}}(A_\rho(\mathcal{M})) - K_i^{\mathrm{a}}(\mathcal{M})|}{|K_i^{\mathrm{a}}(\mathcal{M})|}$$

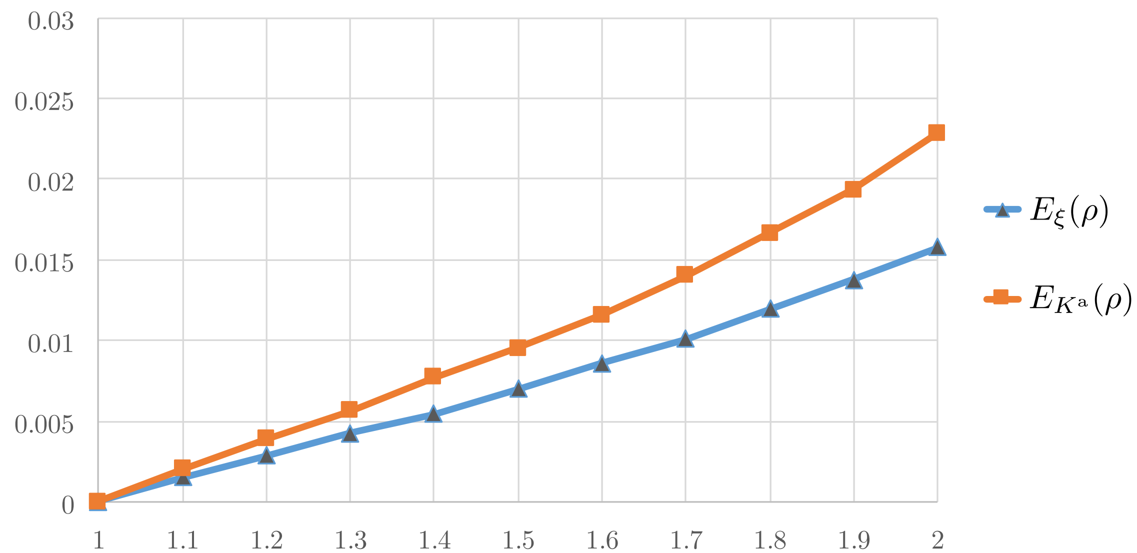


Vieira et al., 2016

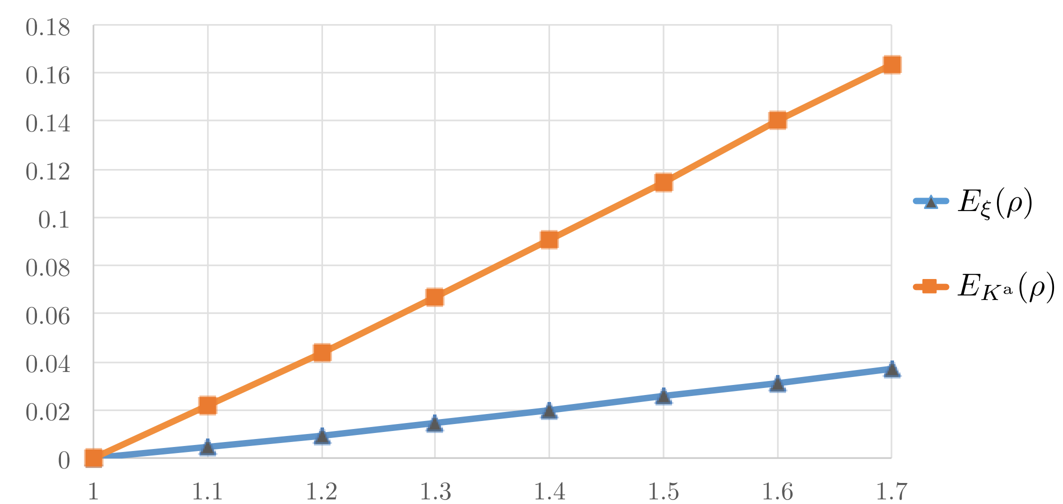
Estimating affine-invariant structures on triangle meshes



Sphere



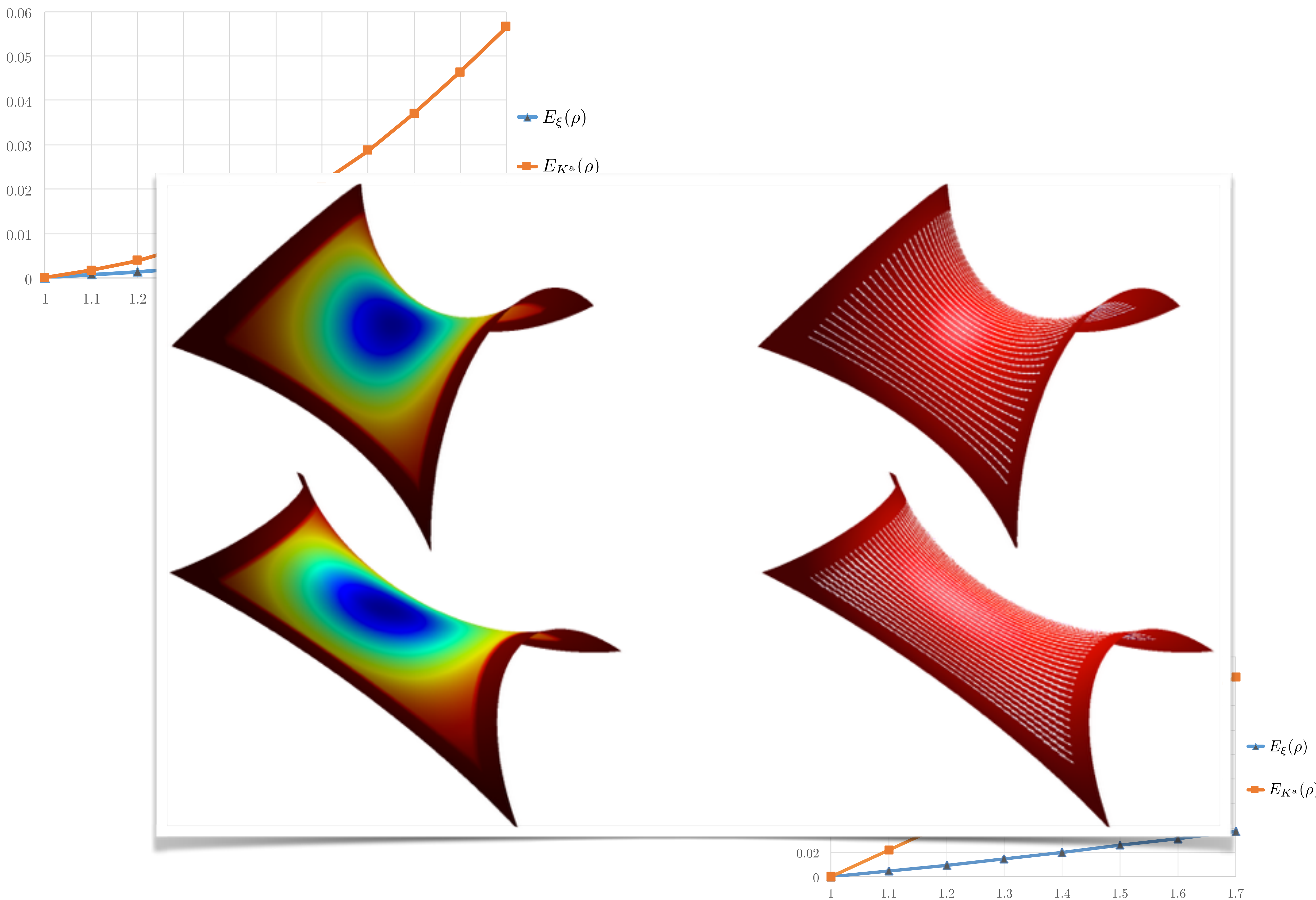
Torus



Horse

Vieira et al., 2016

Estimating affine-invariant structures on triangle meshes

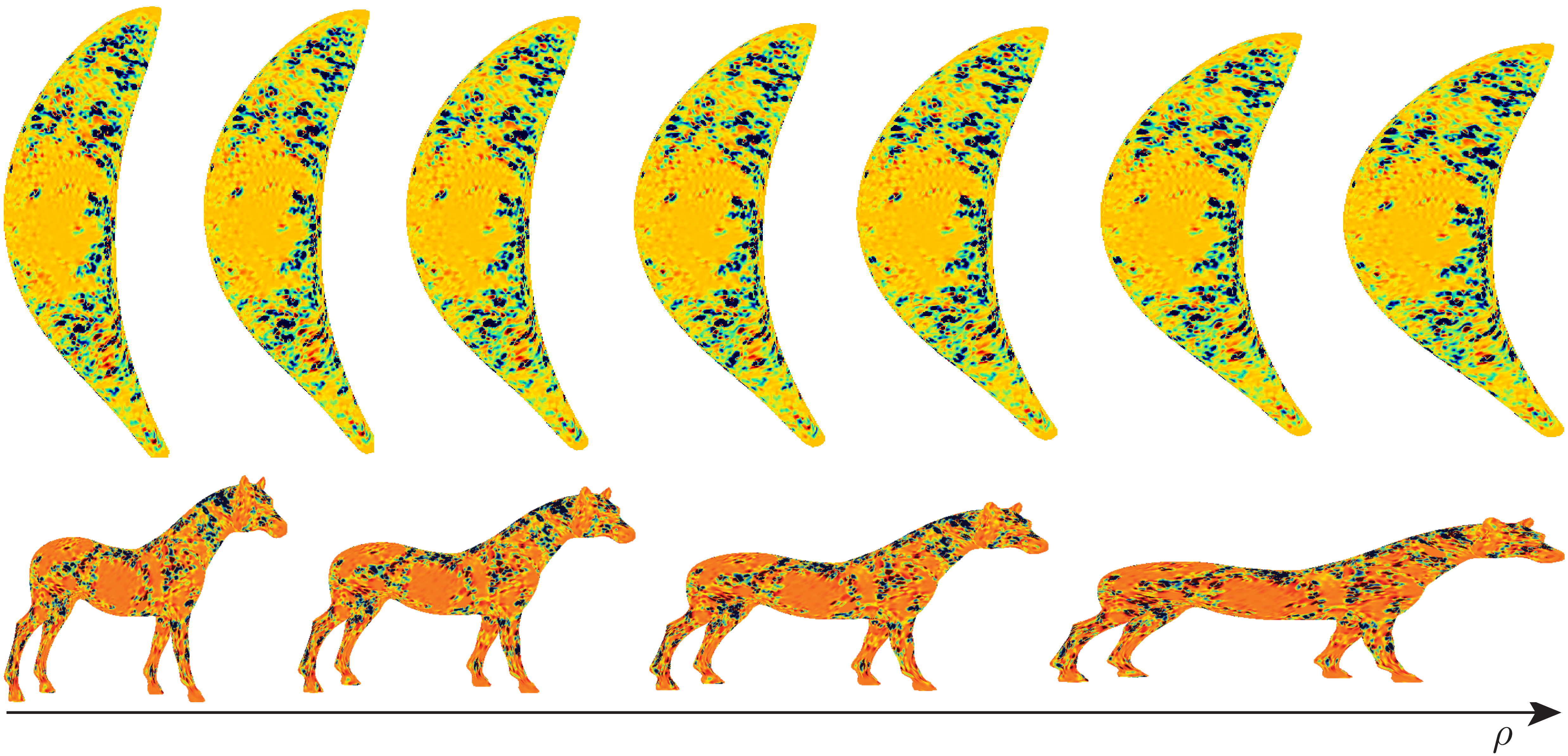


Horse

Vieira et al., 2016

Estimating affine-invariant structures on triangle meshes

Invariance on real-world meshes



Vieira et al., 2016

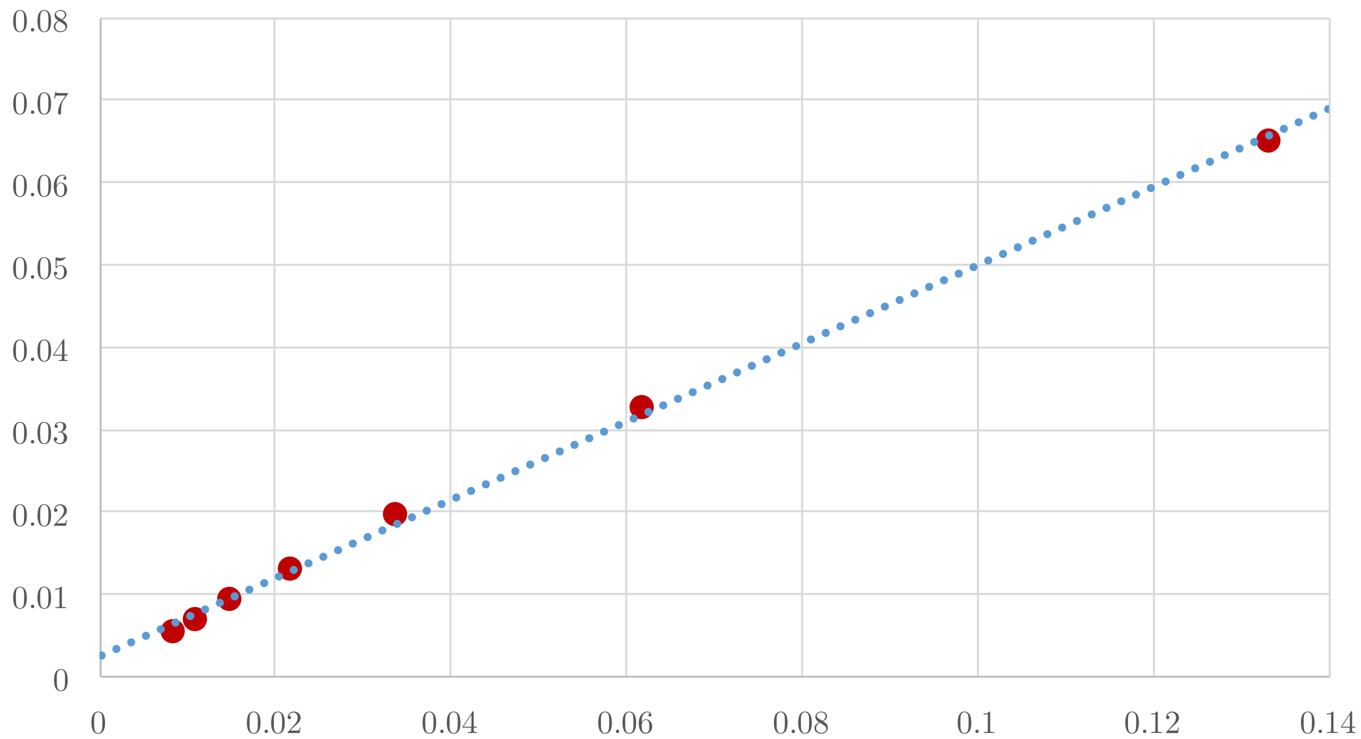
Estimating affine-invariant structures on triangle meshes

Error correlation

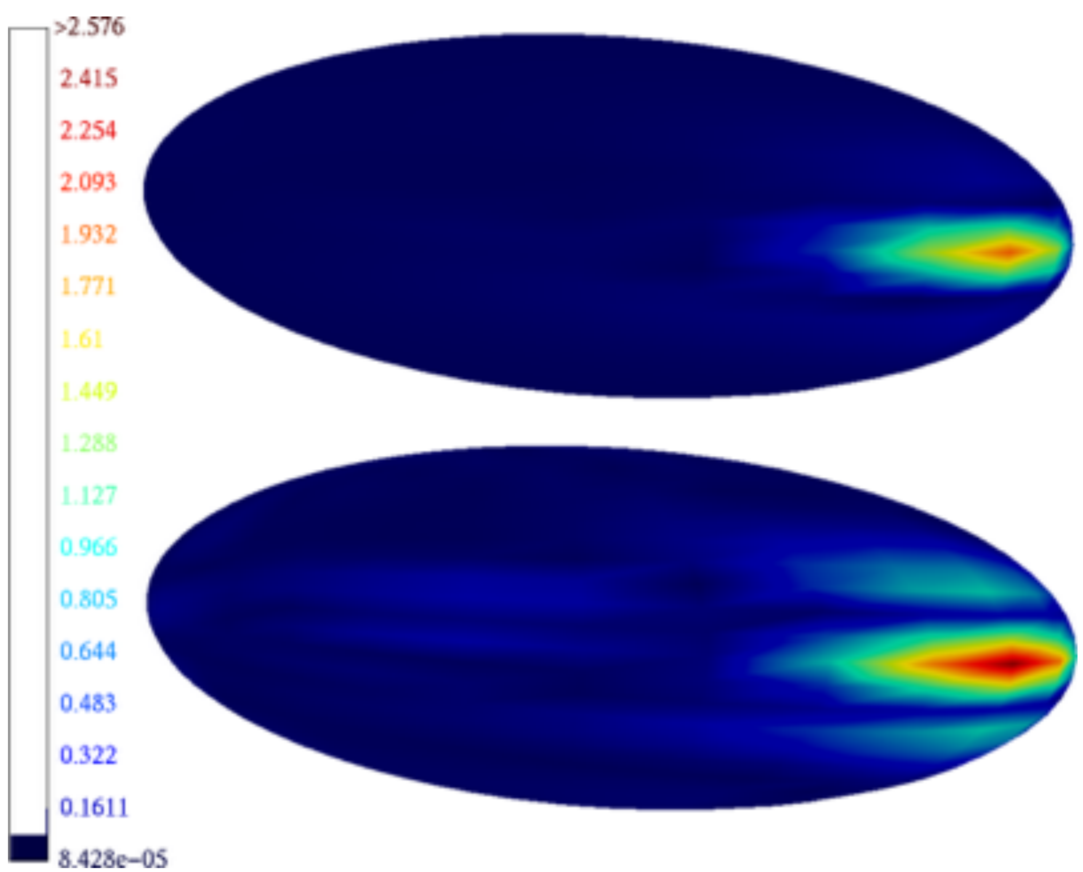
Affine estimators error depends on Euclidian Gaussian curvature estimators accuracy!

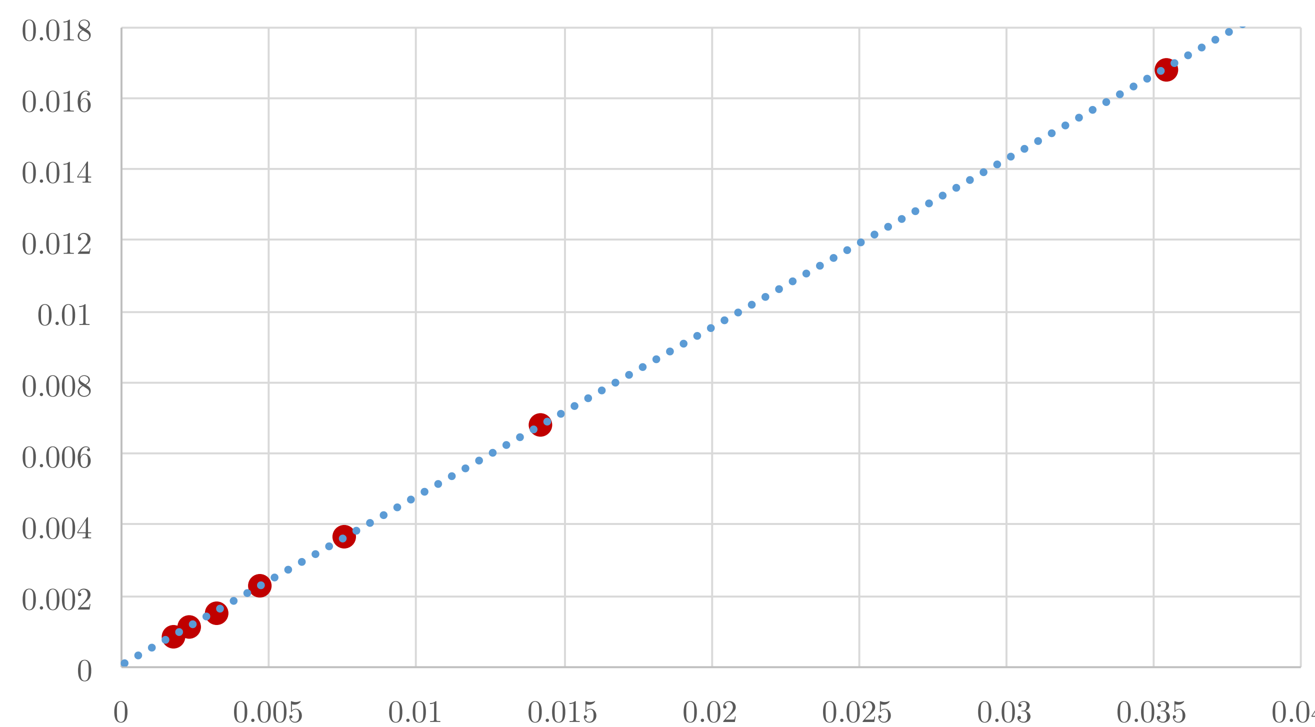
$$\nu_i = |K_i^{\mathrm{e}}|^{-1/4} n_i$$

Estimated from
Rusinkiewicz tensor



ellipsoid K^{e} error $\times$ ξ error





Enneper's surface K^{e} error $\times$ K^{a} error

Vieira et al., 2016

Limitations & Future Work

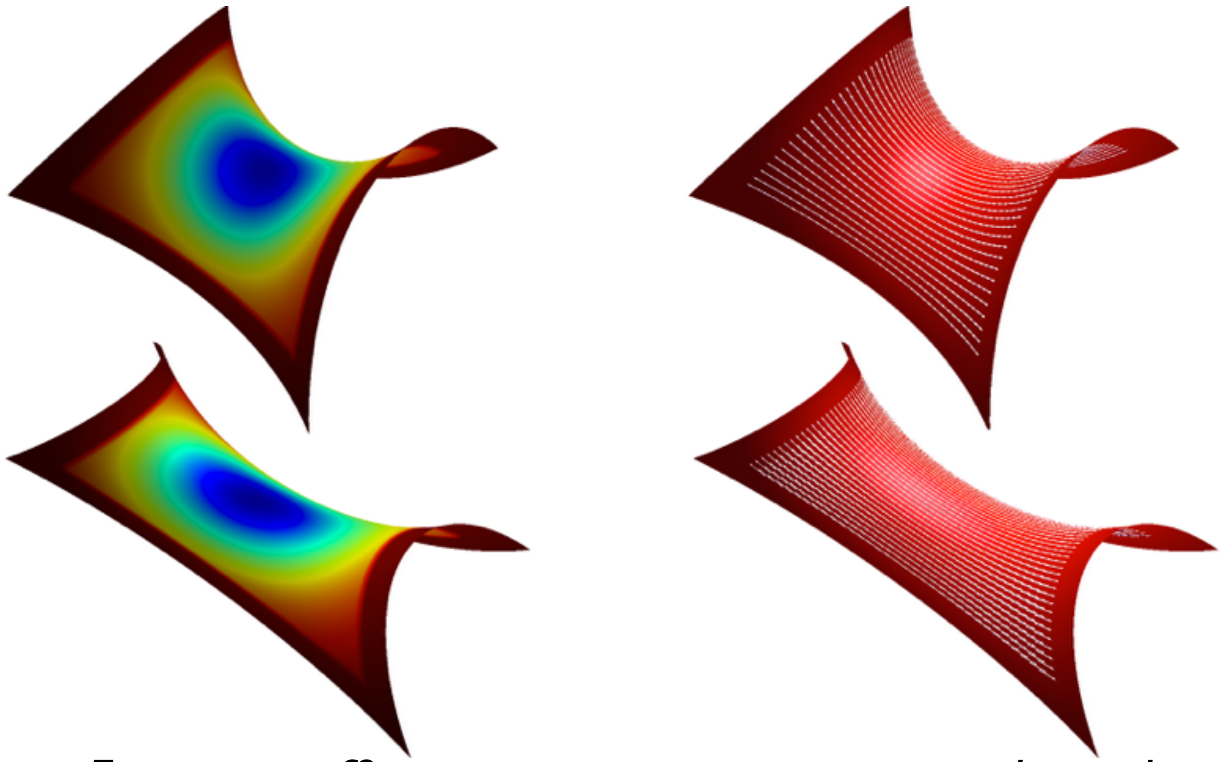
- ✓ Estimators accuracy depends on mesh regularity: better weighting schemes?

$$\underset{y,z}{\text{minimize}} \sum_{j \in N_1(i)} w_j \left(\left(-\frac{b}{a} a_j + b_j \right) \cdot y + \left(-\frac{c}{a} a_j + c_j \right) \cdot z + \frac{a_j}{a} \right)^2$$

- ✓ Estimators accuracy depends on Euclidean estimators

- ✓ Next step: develop practical affine invariant descriptors for shape analysis

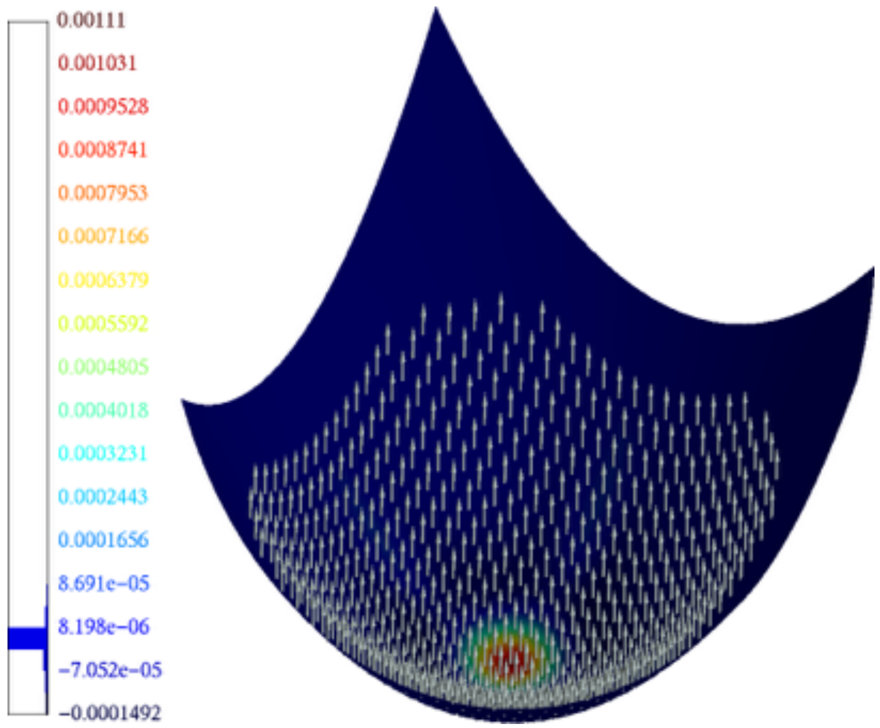
which kind of neighborhoods are more accurate for our estimators?

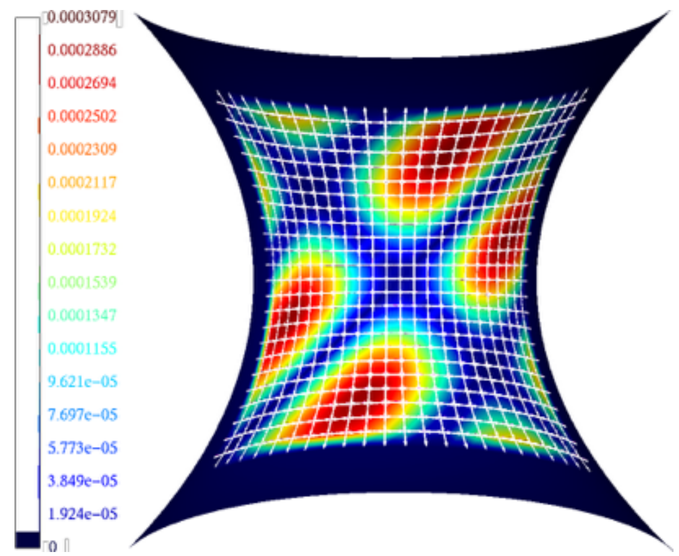


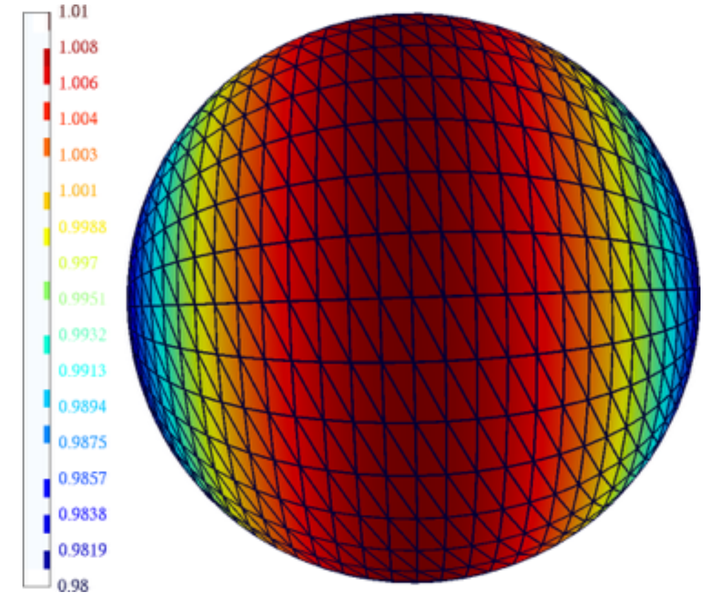
Vieira et al., 2016

Estimating affine-invariant structures on triangle meshes

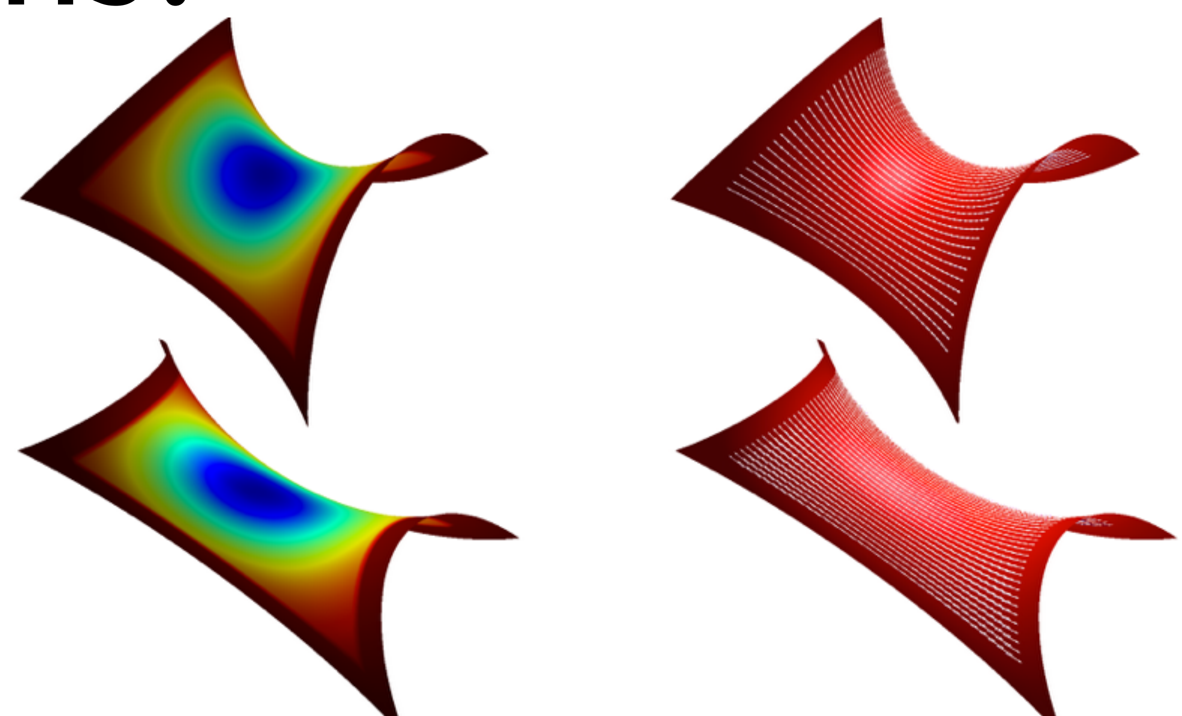
Thank you for your attention!







Questions?



Vieira et al., 2016

Estimating affine-invariant structures on triangle meshes