FFTs in Graphics and Vision

Invariance of Shape Descriptor

Outline

- Math Overview
  - Translation and Rotation Invariance
  - The 0th Order Frequency Component
- Shape Descriptors
- Invariance

Translation Invariance

Given a function *f* in 2D / 3D, we can obtain a translation invariant representation of the function by expressing the function in terms of its Fourier decomposition and storing just the magnitudes of the frequency components:

Translation Invariance

Given a function *f* in 2D / 3D, we can obtain a translation invariant representation of the function by expressing the function in terms of its Fourier decomposition and storing just the magnitudes of the frequency components:

$$f(x,y) = \sum_{l,m=-\infty}^{\infty} \hat{f}(l,m) \frac{e^{i(lx+my)}}{\sqrt{2\pi}}$$

$$\Big\downarrow$$

$$\left\{ \left\| \hat{f}(l,m) \right\| \right\}_{l,m=-\infty}^{\infty}$$

Rotation Invariance (Circle)

Given a function *f*(θ) on a circle, we can obtain a rotation invariant representation of the function by expressing the function in terms of its Fourier decomposition and storing just the magnitudes of the frequency components:

$$f(\theta) = \sum_{l=-\infty}^{\infty} \hat{f}(l) \frac{e^{il\theta}}{\sqrt{2\pi}}$$

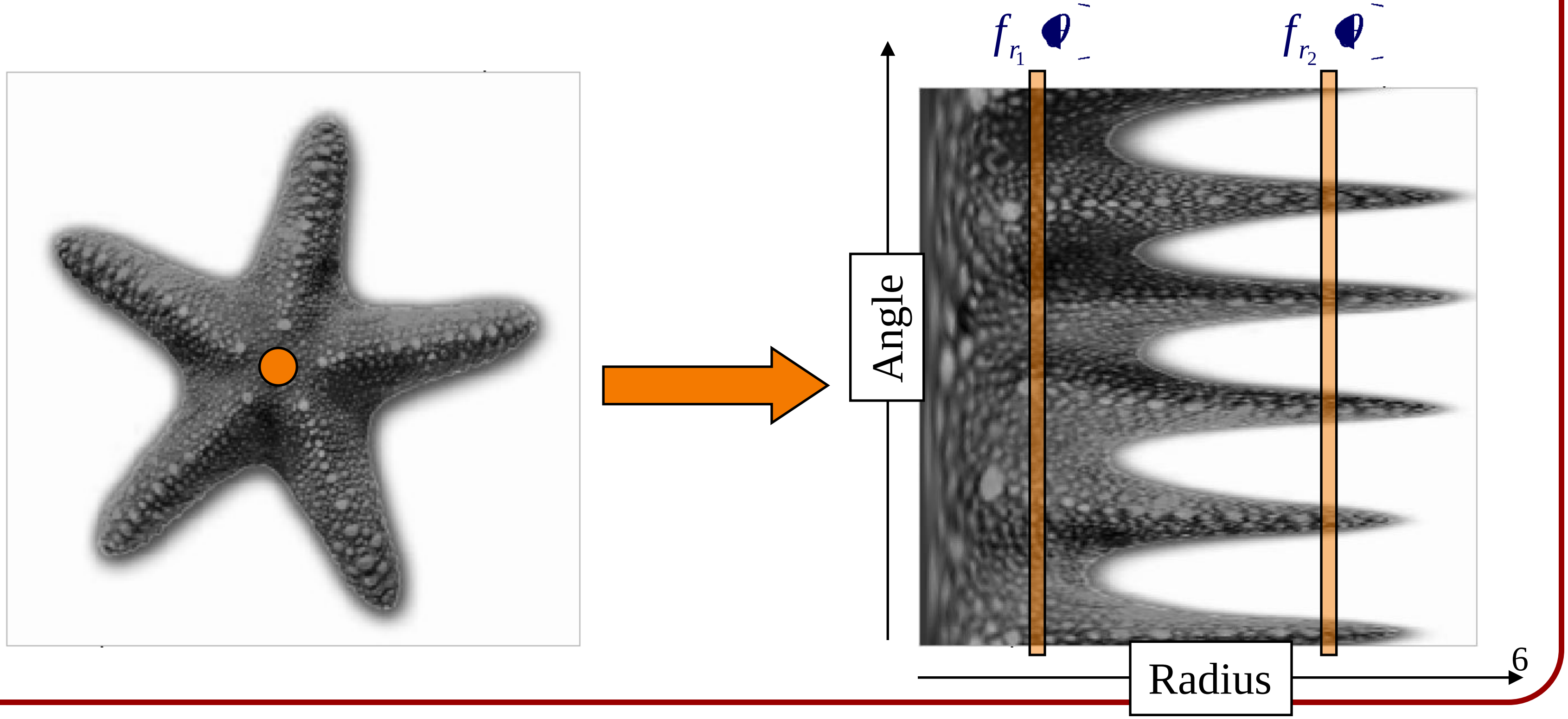
$$\left\{ \left\| \hat{f}(l) \right\| \right\}_{l=-\infty}^{\infty}$$

Rotation Invariance (2D)

Given a function *f*(*x*,*y*) in 2D, we can obtain a rotation invariant representation of *f* by:

- Expressing *f* in polar coordinates:

$$f_r(\theta) = f(r\cos\theta, r\sin\theta)$$



Rotation Invariance (2D)

Given a function *f*(*x*,*y*) in 2D, we can obtain a rotation invariant representation of *f* by:

- Expressing *f* in polar coordinates:

$$f_r(\theta) = f(r\cos\theta, r\sin\theta)$$

- Expressing each radial restriction in terms of its Fourier decomposition:

$$f_r(\theta) = \sum_{l=-\infty}^{\infty} \hat{f}_r(l)\frac{e^{il\theta}}{\sqrt{2\pi}}$$

Rotation Invariance (2D)

Given a function *f*(*x*,*y*) in 2D, we can obtain a rotation invariant representation of *f* by:

- Expressing *f* in polar coordinates:

$$f_r(\theta) = f(r\cos\theta, r\sin\theta)$$

- Expressing each radial restriction in terms of its Fourier decomposition:

$$f_r(\theta) = \sum_{l=-\infty}^{\infty} \hat{f}_r(l) \frac{e^{il\theta}}{\sqrt{2\pi}}$$

- Storing the magnitude of the Fourier coefficients of the different radial restrictions

$$\left\{ \left\| \hat{f}_r(l) \right\| \sqrt{2\pi r} \right\}_{l=-\infty}^{\infty}$$

Rotation Invariance (Sphere)

Given a function *f*(θ,ϕ) on a sphere, we can obtain a rotation invariant representation of the function by expressing the function in terms of its spherical harmonic decomposition and storing just the magnitudes of the frequency components:

$$f(\theta,\phi)=\sum_{l=0}^{\infty}\sum_{k=-l}^{l}\hat{f}(l,k)Y_l^k(\theta,\phi)$$

↓

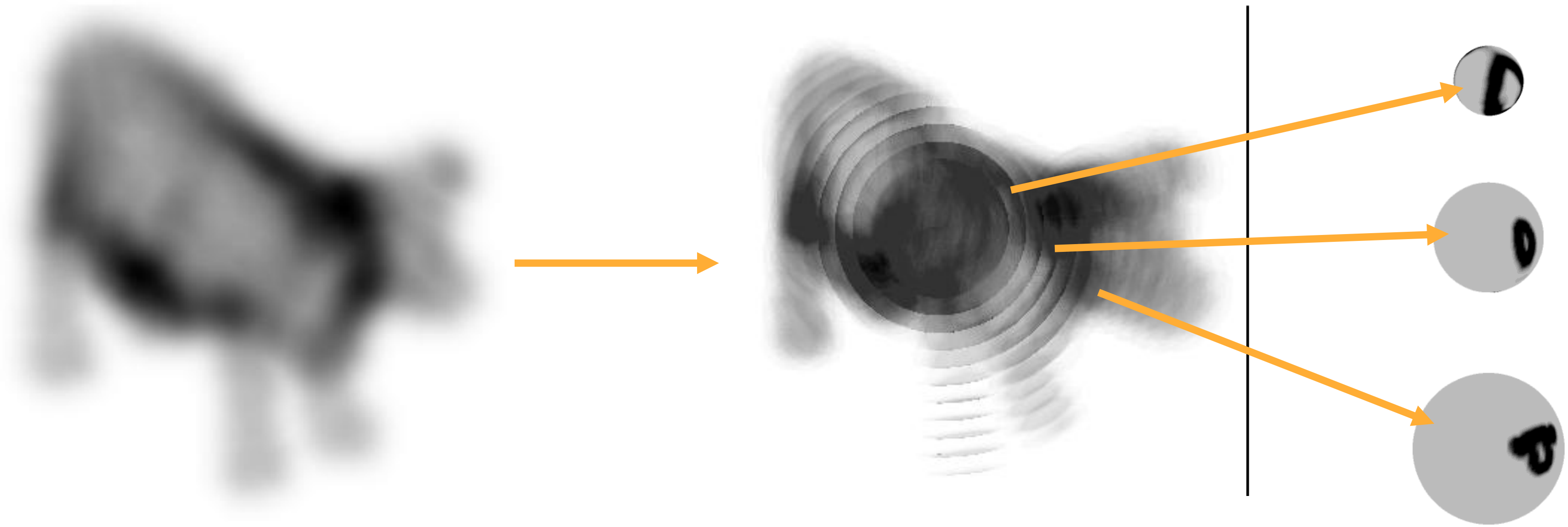
$$\left\{\sqrt{\sum_{k=-l}^{l}\left\|\hat{f}(l,k)\right\|^2}\right\}_{l=0}^{\infty}$$

Rotation Invariance (3D)

Given a function *f*(*x*,*y*,*z*) in 3D, we can obtain a rotation invariant representation of *f* by:

- Expressing *f* in spherical coordinates:

$$f_r(\theta,\phi) = f(r\cos\theta\sin\phi, r\cos\phi, r\sin\theta\sin\phi)$$



Rotation Invariance (3D)

Given a function *f*(*x*,*y*,*z*) in 3D, we can obtain a rotation invariant representation of *f* by:

- Expressing *f* in spherical coordinates:

$$f_r(\theta,\phi) = f(r\cos\theta\sin\phi, r\cos\phi, r\sin\theta\sin\phi)$$

- Expressing each radial restriction in terms of its spherical harmonic decomposition:

$$f_r(\theta,\phi) = \sum_{l=0}^{\infty}\sum_{k=-l}^{l}\hat{f}_r(l,k)Y_l^k(\theta,\phi)$$

Rotation Invariance (3D)

Given a function *f*(*x*,*y*,*z*) in 3D, we can obtain a rotation invariant representation of *f* by:

- Expressing *f* in spherical coordinates:

$$f_r(\theta,\phi) = f(r\cos\theta\sin\phi, r\cos\phi, r\sin\theta\sin\phi)$$

- Expressing each radial restriction in terms of its spherical harmonic decomposition:

$$f_r(\theta,\phi) = \sum_{l=0}^{\infty}\sum_{k=-l}^{l}\hat{f}_r(l,k)Y_l^k(\theta,\phi)$$

- Storing the size of the frequency components coefficients of the different radial restrictions

$$\left\{\sqrt{\sum_{k=-l}^{l}\left\|\hat{f}_r(l,k)\right\|^2}\sqrt{4\pi r^2}\right\}_{l=0}^{\infty}$$

The 0th Order Frequency Component

Given a function on the circle *f*(θ), we can express the function in terms of its Fourier decomposition:

$$f(\theta)=\sum_{l=-\infty}^{\infty}\hat{f}(l)\frac{e^{il\theta}}{\sqrt{2\pi}}$$

The 0th Order Frequency Component

Given a function on the circle *f*(θ), we can express the function in terms of its Fourier decomposition:

$$f(\theta) = \sum_{l=-\infty}^{\infty} \hat{f}(l) \frac{e^{il\theta}}{\sqrt{2\pi}}$$

What is the meaning of the 0th order frequency component?

The 0th Order Frequency Component

In general, the *l*th frequency component can be computed by taking the dot product of the function with the *l*th complex exponential:

$$\hat{f}(l) = \left\langle f(\theta), \frac{e^{il\theta}}{\sqrt{2\pi}} \right\rangle = \int_0^{2\pi} f(\theta) \frac{e^{-il\theta}}{\sqrt{2\pi}} d\theta$$

The 0th Order Frequency Component

In general, the *l*th frequency component can be computed by taking the dot product of the function with the *l*th complex exponential:

$$\hat{f}(l) = \left\langle f(\theta), \frac{e^{il\theta}}{\sqrt{2\pi}} \right\rangle = \int_0^{2\pi} f(\theta) \frac{e^{-il\theta}}{\sqrt{2\pi}} d\theta$$

Thus, the 0th frequency component is:

$$\hat{f}(0) = \frac{1}{\sqrt{2\pi}} \int_0^{2\pi} f(\theta) d\theta$$

The 0th Order Frequency Component

Up to a normalization term, the 0th frequency component of a function *f*(θ) is just the integral of the function over the circle:

$$\hat{f}(0) = \frac{1}{\sqrt{2\pi}} \int_0^{2\pi} f(\theta)d\theta$$

The 0th Order Frequency Component

Given a function on the sphere $f(\theta,\phi)$, we can express the function in terms of its spherical harmonic decomposition:

$$f(\theta,\phi)=\sum_{l=0}^{\infty}\sum_{k=-l}^{l}\hat{f}(l,k)Y_l^k(\theta,\phi)$$

The 0th Order Frequency Component

Given a function on the sphere $f(\theta,\phi)$, we can express the function in terms of its spherical harmonic decomposition:

$$f(\theta,\phi)=\sum_{l=0}^{\infty}\sum_{k=-l}^{l}\hat{f}(l,k)Y_l^k(\theta,\phi)$$

What is the meaning of the 0th order frequency component?

The 0th Order Frequency Component

In general, the (*l*,*k*)th frequency component can be computed by taking the dot product of the function with the (*l*,*k*)th spherical harmonic:

$$\hat{f}(l,k) = \left\langle f(\theta,\phi), Y_l^k(\theta,\phi) \right\rangle$$

The 0th Order Frequency Component

In general, the (*l*,*k*)th frequency component can be computed by taking the dot product of the function with the (*l*,*k*)th spherical harmonic:

$$\hat{f}(l,k) = \left\langle f(\theta,\phi), Y_l^m(\theta,\phi) \right\rangle$$

Thus, the 0th frequency component is:

$$\hat{f}(0,0) = \frac{1}{\sqrt{4\pi}} \int_{|p|=1} f(p)dp$$

The 0th Order Frequency Component

Up to a normalization term, the 0th frequency component of a function *f*(θ,φ) is just the integral of the function over the sphere:

$$\hat{f}\left(0,0\right)=\frac{1}{\sqrt{4\pi}}\int_{|p|=1} f(p)dp$$

The 0th Order Frequency Component

Note:

In the case that the function *f* is positive the 0th frequency coefficient will also be positive:

$$\left\|\hat{f}(0)\right\| = \hat{f}(0)$$

$$\left\|\hat{f}(0,0)\right\| = \hat{f}(0,0)$$

Outline

- Math Overview

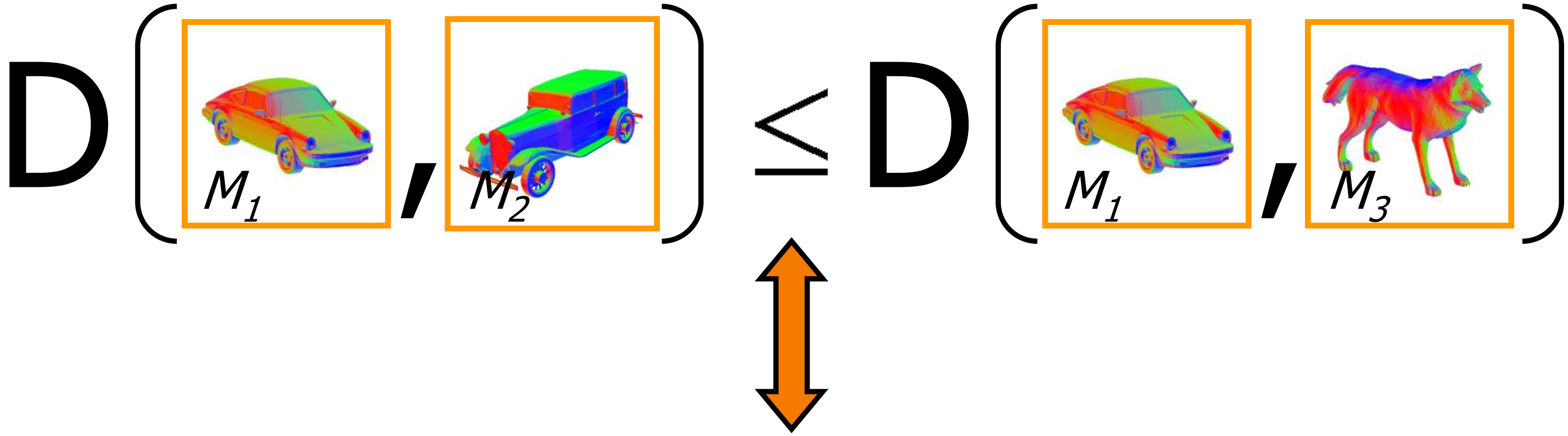
- Shape Descriptors
  - Shape Histograms (Ankerst *et al*.)
  - Shape Distributions (Osada *et al*.)
  - Extended Gaussian Images (Horn)

- Invariance

Shape Matching

General Approach

Define a function that takes in two models and returns a measure of their proximity.



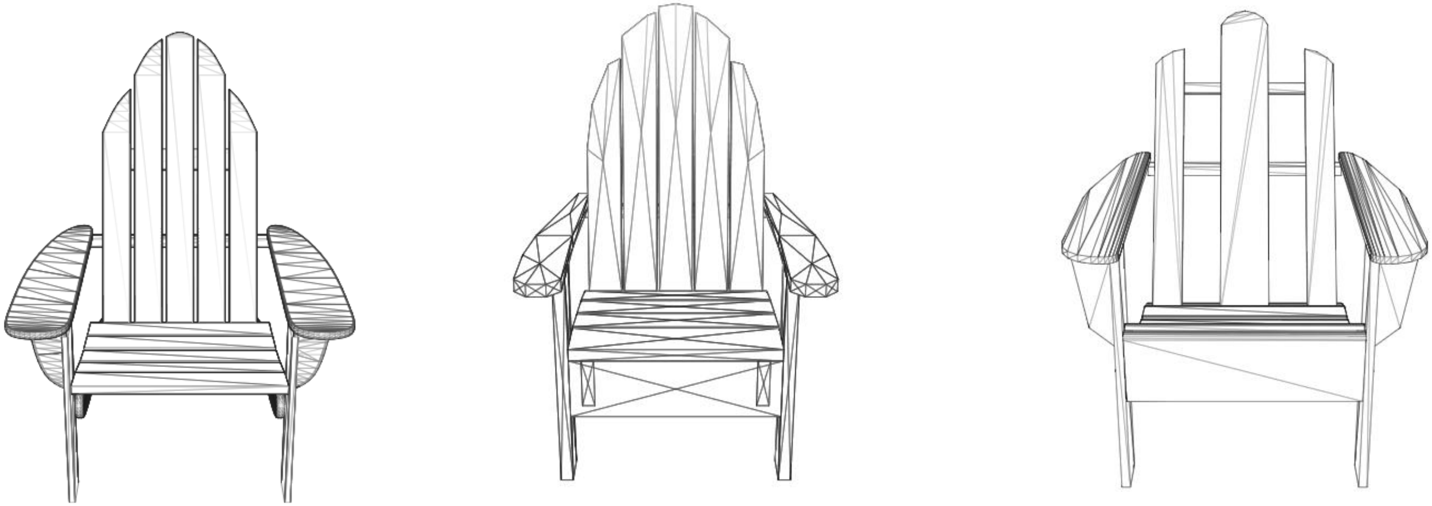
M_1 is closer to M_2 than it is to M_3

Shape Descriptors

Challenge

It is difficult to match shapes directly:

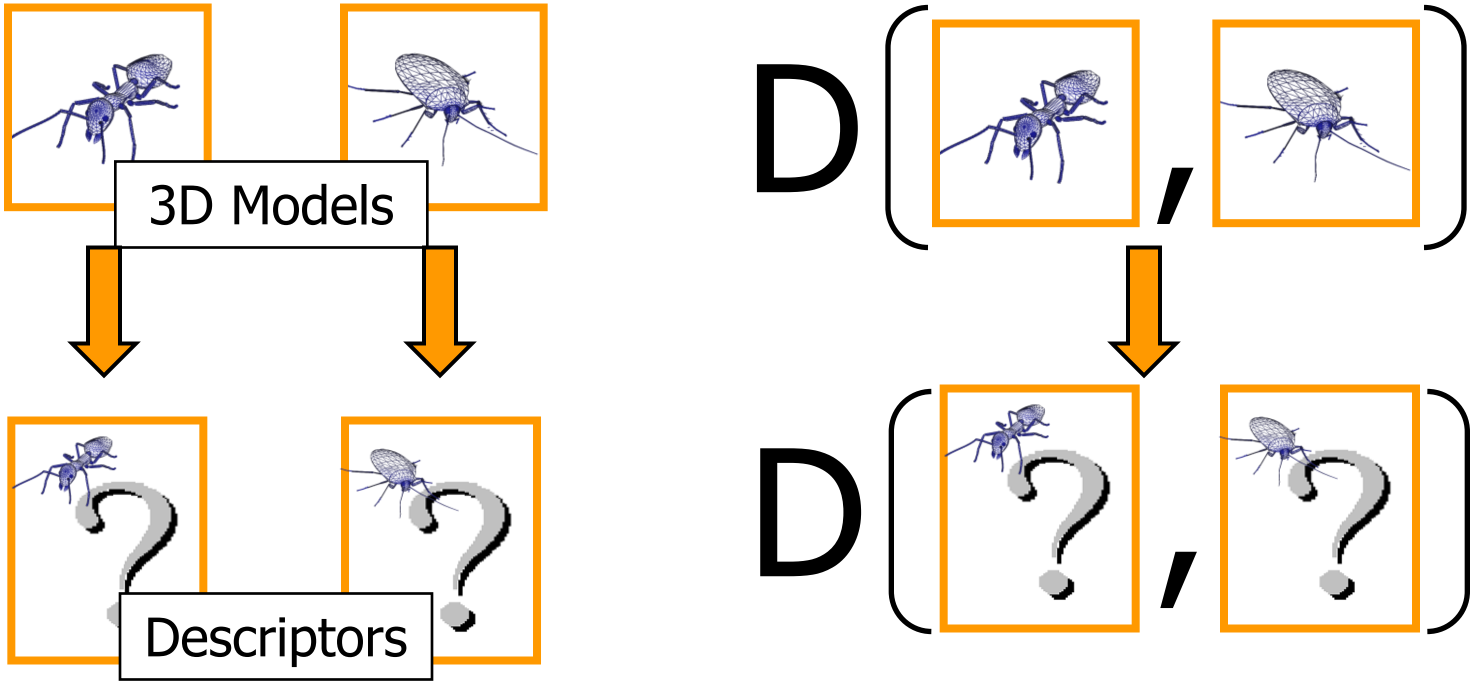
- Different triangulations of the same shape
- Different shapes have different genus
- The same shape may be in different poses
- Etc.



Shape Descriptors

Solution

Represent shapes by a structured abstraction that represents every shape in the same domain.



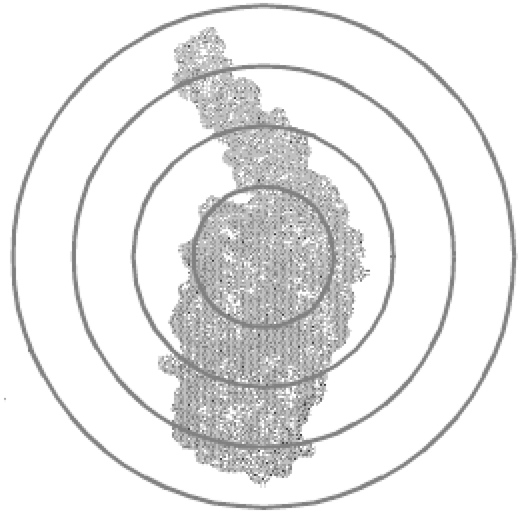
Outline

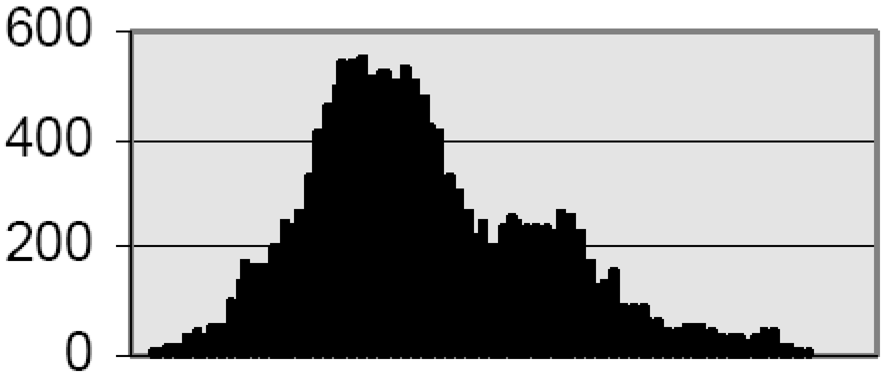
- Math Overview
- Shape Descriptors
  - Shape Histograms (Ankerst *et al*.)
  - Shape Distributions (Osada *et al*.)
  - Extended Gaussian Images (Horn)
- Invariance

Shape Histograms

Approach

- Decompose space into concentric shells
- Store how much of the shape falls into each of the shells

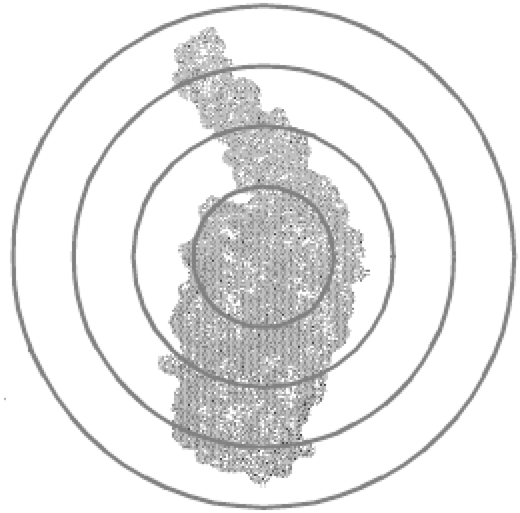


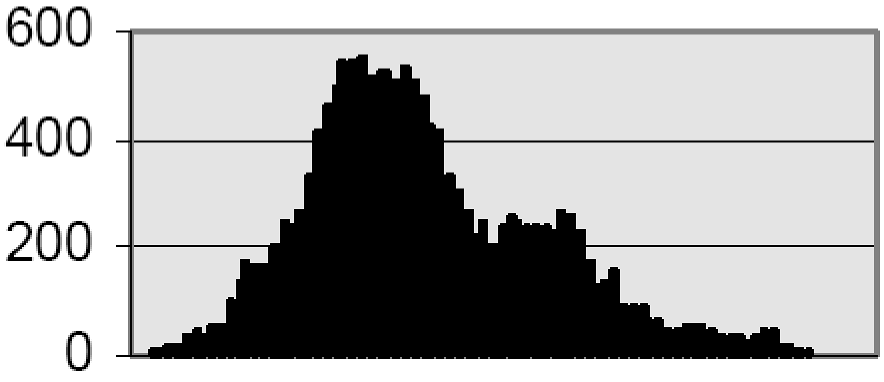


Shape Histograms

Properties

- Each shape is represented by 1D array of values.
- The representation is invariant to rotation





Outline

- Math Overview

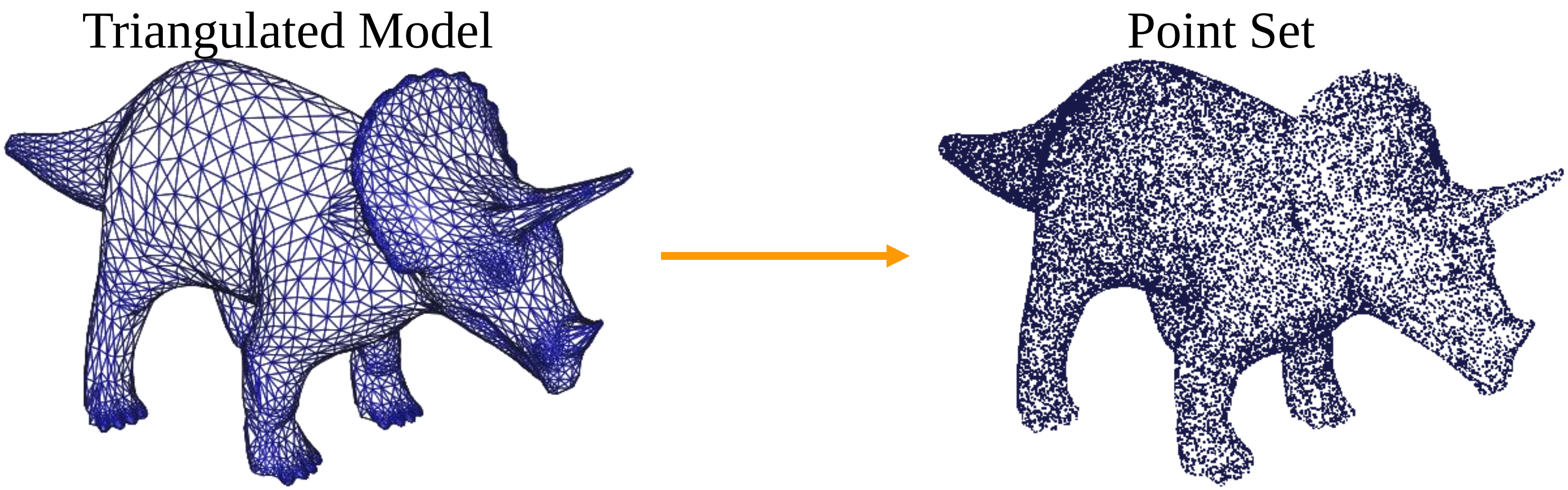
- Shape Descriptors
  - Shape Histograms (Ankerst *et al*.)
  - Shape Distributions (Osada *et al*.)
  - Extended Gaussian Images (Horn)

- Invariance

D2 Shape Distributions

Approach

Avoid the whole problem of tesselation, genus, etc. by building the shape descriptor from random samples from the surface of the model:



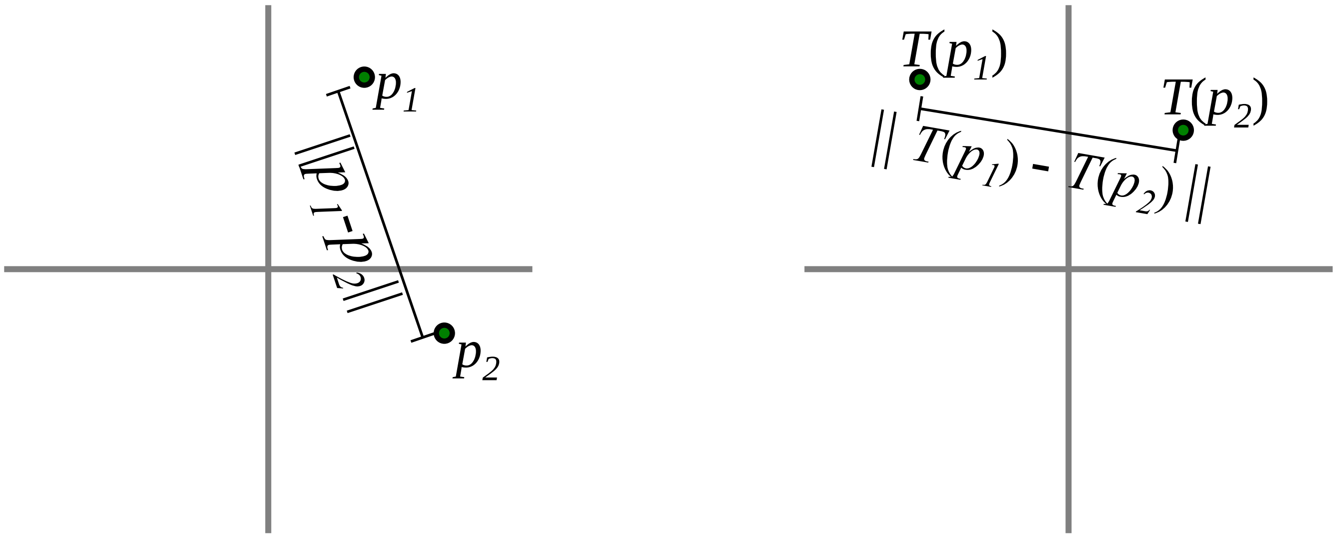
Triangulated Model

Point Set

D2 Shape Distributions

Key Idea

Use the fact that the distance between pairs of points on the model does not change if the model is translated and/or rotated.

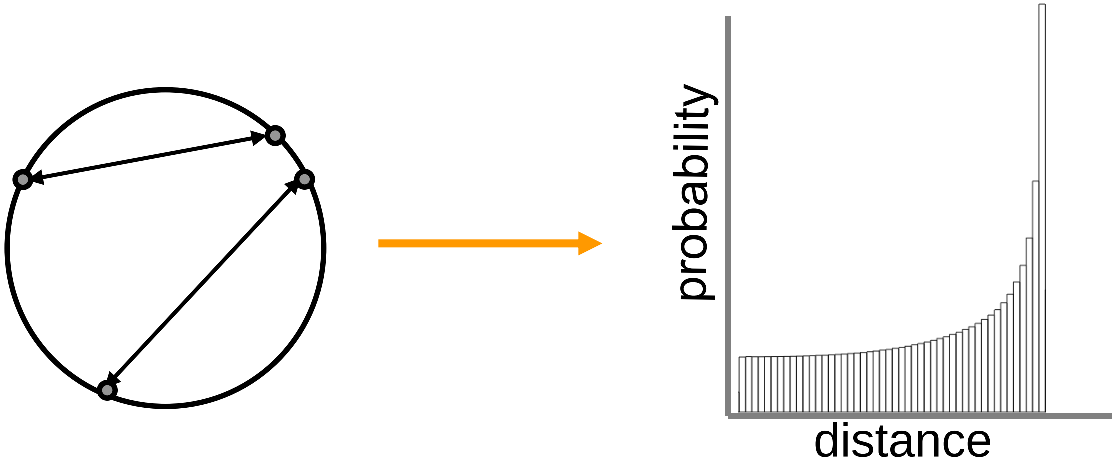


D2 Shape Distributions

Descriptor

Represent shapes by binning point-pairs from the surface by distance:

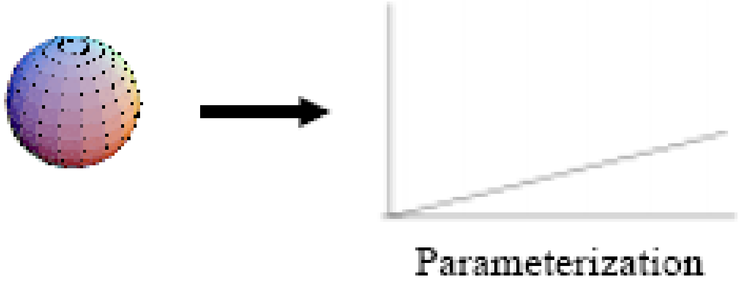
$$\mathrm{D2}_P(d) = \frac{\left|\left\{ p,q \in P \quad \text{s.t.} \quad \|p-q\| = d \right\}\right|}{|P|^2}$$

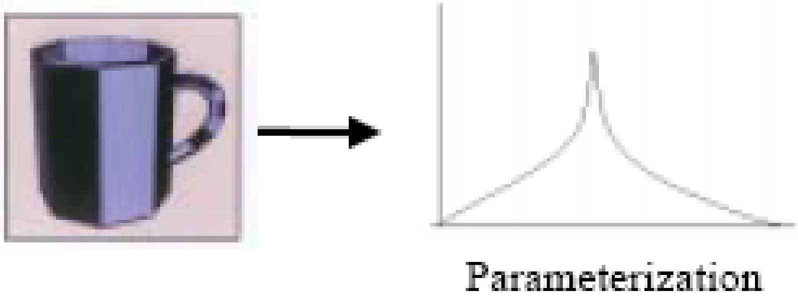


D2 Shape Distributions

Properties

- Each shape is represented by 1D array of values.
- The representation is invariant to translations and rotations





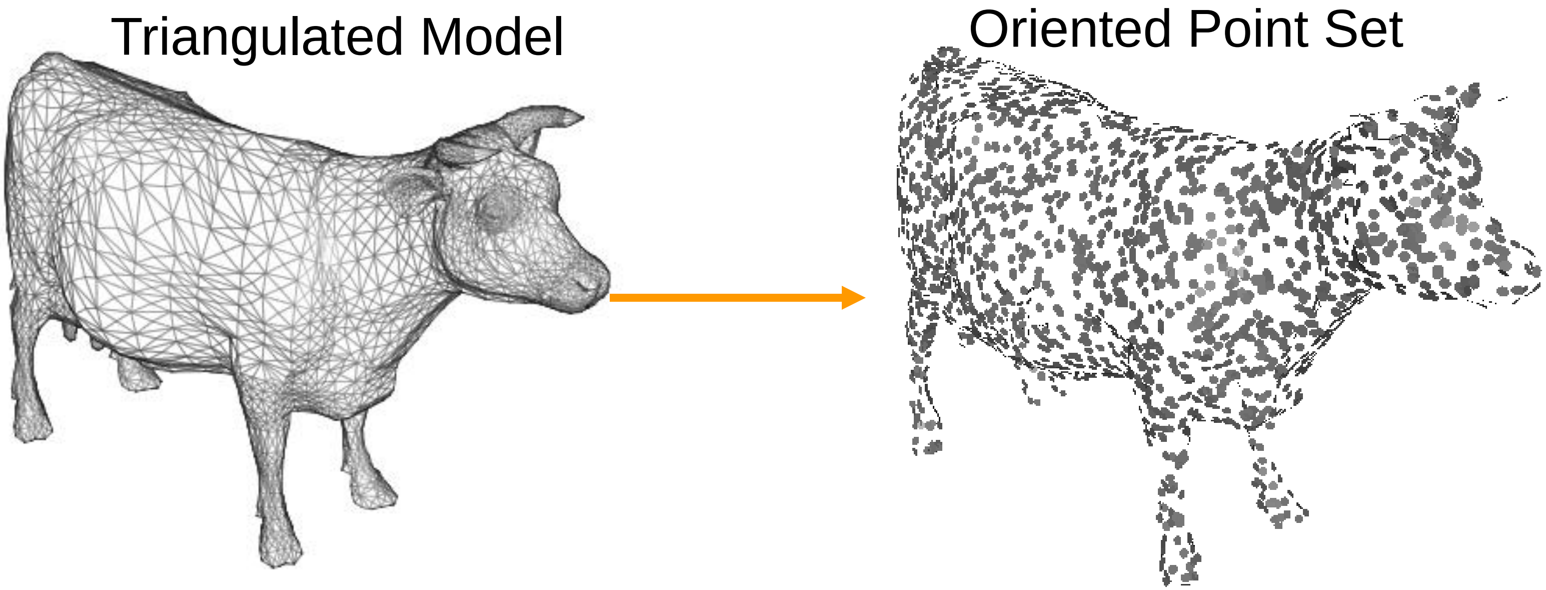
Outline

- Math Overview
- Shape Descriptors
  - Shape Histograms (Ankerst *et al*.)
  - Shape Distributions (Osada *et al*.)
  - Extended Gaussian Images (Horn)
- Invariance

Extended Gaussian Images

Approach

Use the fact that every point on the surface has a position and a normal.



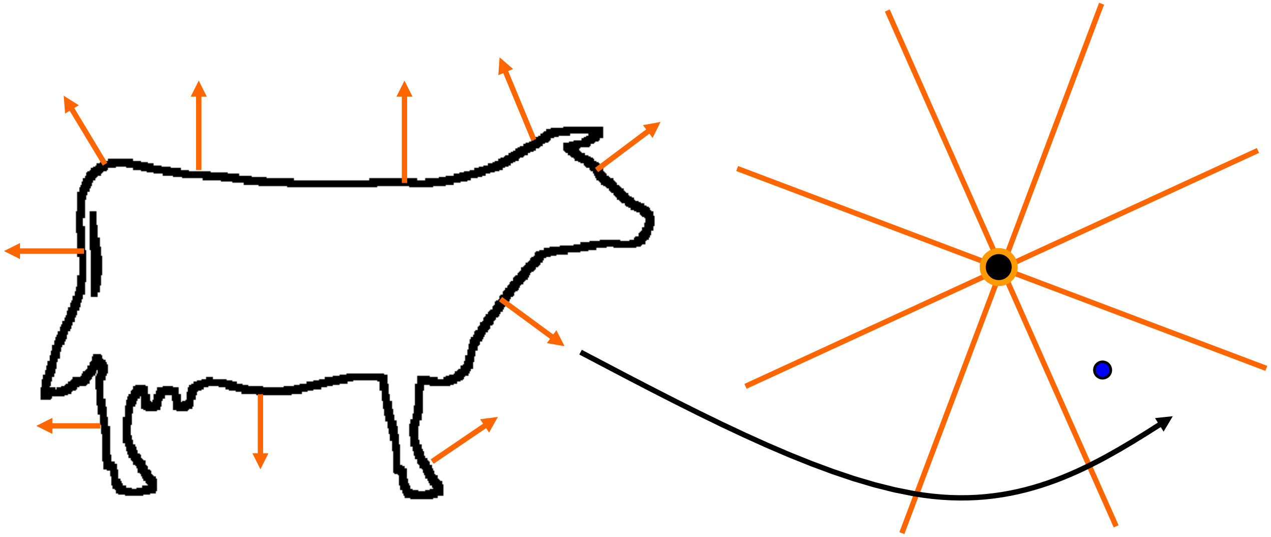
Triangulated Model

Oriented Point Set

Extended Gaussian Images

Descriptor

Represent a model by binning surface normals



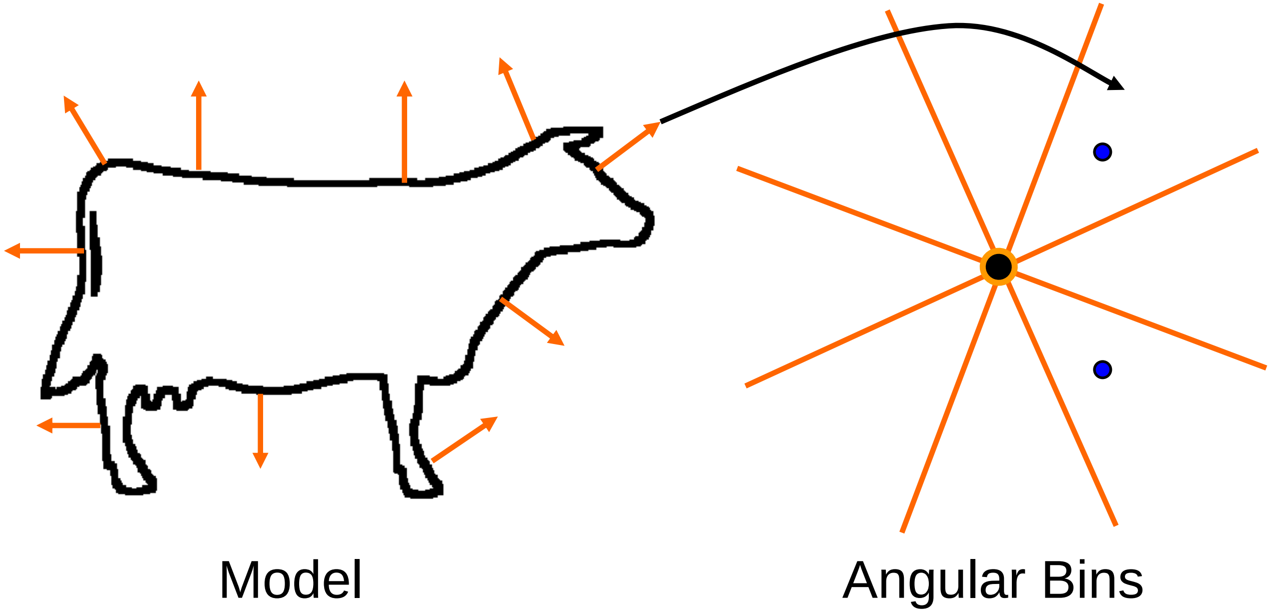
Model

Angular Bins

Extended Gaussian Images

Descriptor

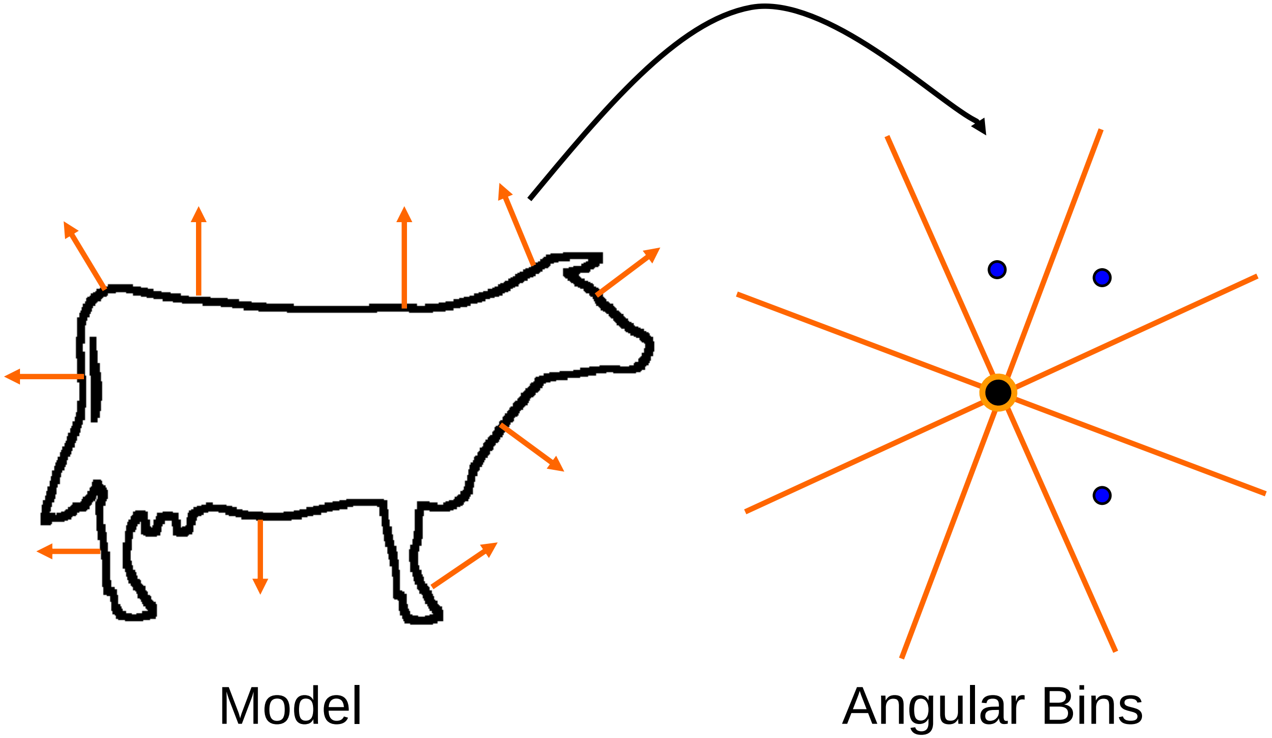
Represent a model by binning surface normals



Extended Gaussian Images

Descriptor

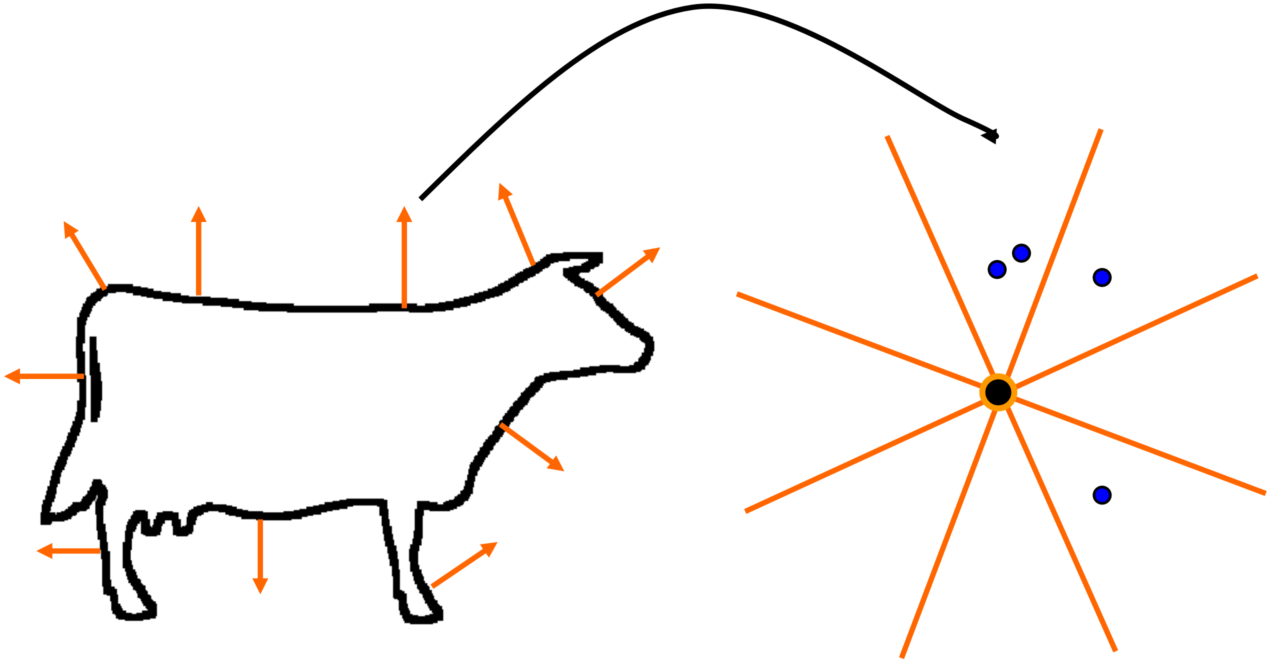
Represent a model by binning surface normals



Extended Gaussian Images

Descriptor

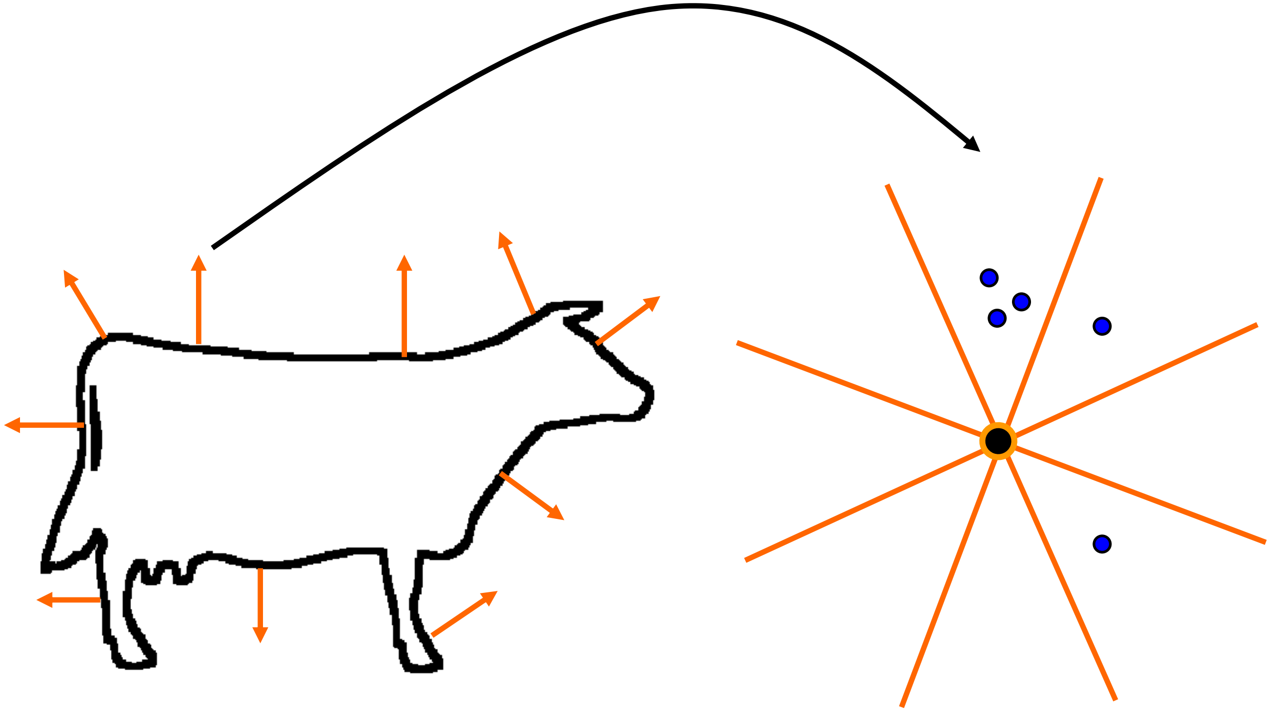
Represent a model by binning surface normals



Extended Gaussian Images

Descriptor

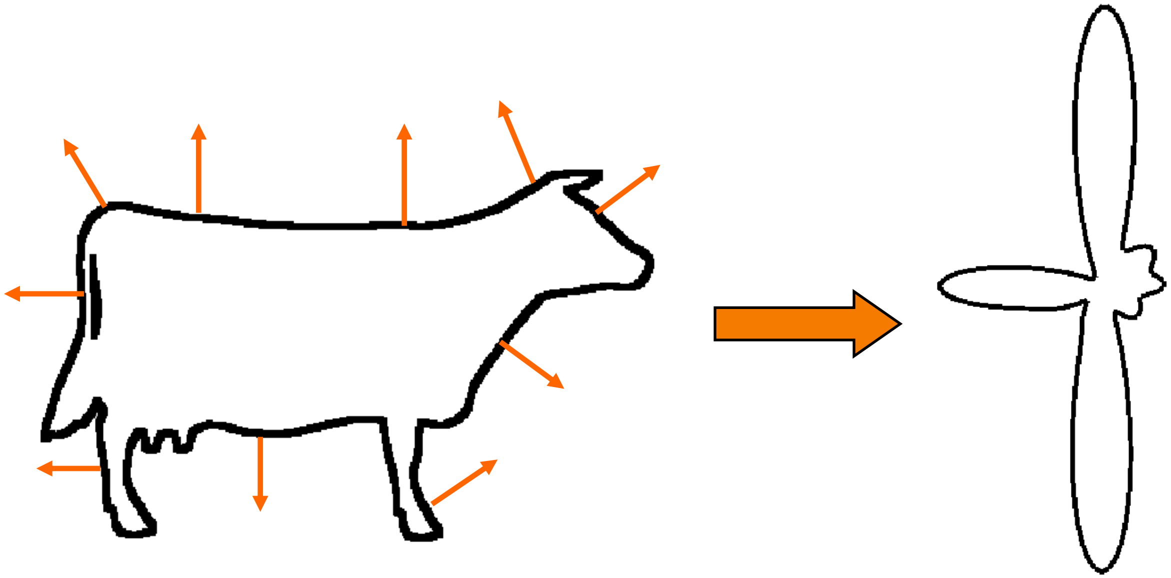
Represent a model by binning surface normals



Extended Gaussian Images

Descriptor

Represent a model by binning surface normals



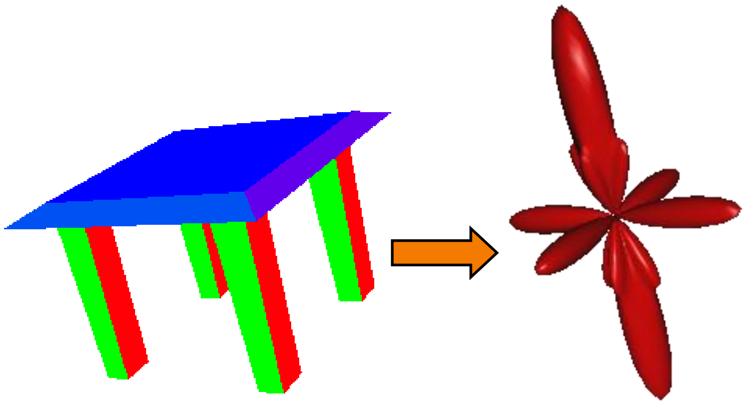
Model

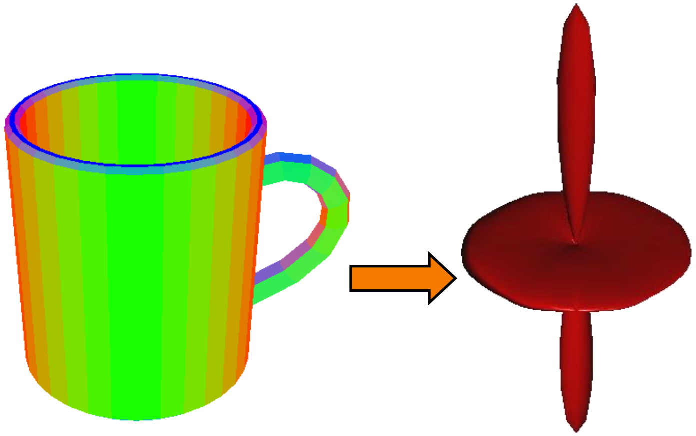
EGI

Extended Gaussian Images

Properties

- A 2D curve / 3D surface is represented by a histogram over a circle / sphere.
- The representation is invariant to translations.





Outline

- Math Overview
- Shape Descriptors
- Invariance

Normalization vs. Invariance

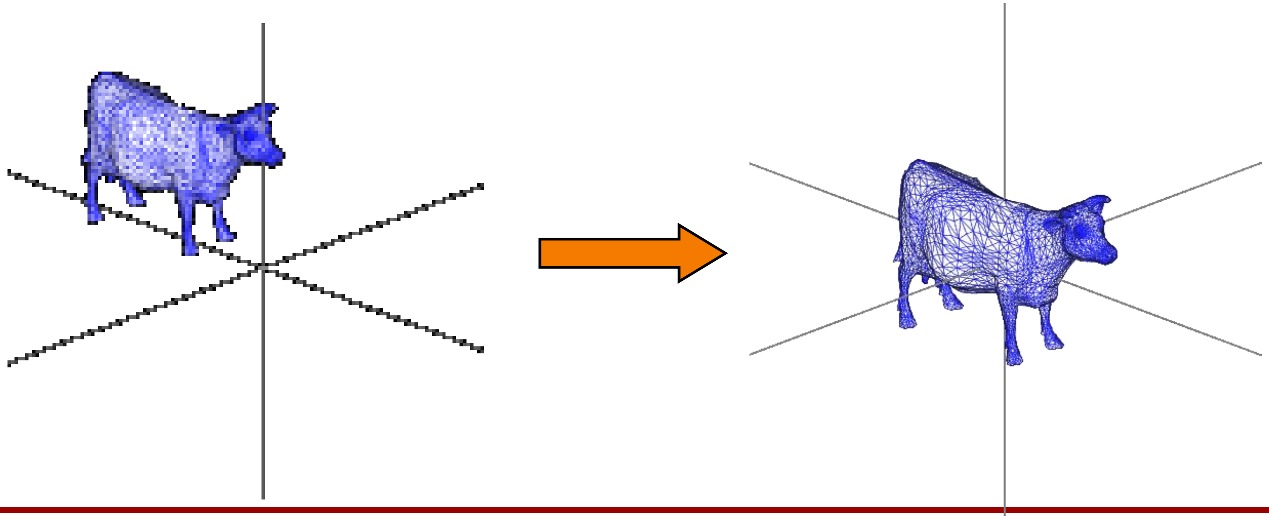
We say that a shape representation is normalized with respect to translation / rotation if the shape is placed into a canonical pose.

Normalization vs. Invariance

We say that a shape representation is normalized with respect to translation / rotation if the shape is placed into a canonical pose.

Example:

We can normalize for translation by moving the surface so that the center of mass is at the origin.



Normalization vs. Invariance

We say that a shape representation is invariant with respect to translation / rotation if the representation discards information that depends on translation / rotation.

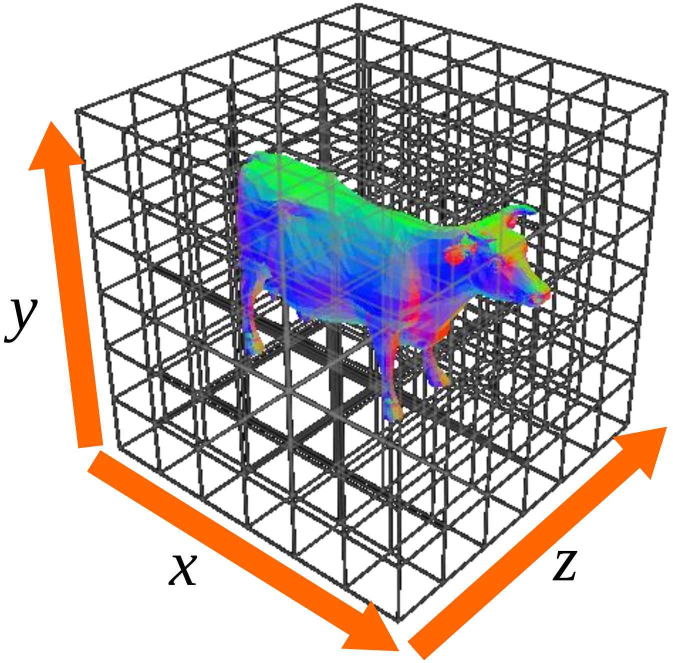
Invariance

We have seen a general method for making functions invariant to translation and rotation.

Invariance

Translation:

Compute the Fourier decomposition and store just the magnitudes of the Fourier coefficients.



Cartesian Coordinates

$$f(x,y,z)=\sum_{l,m,n} f_{l,m,n}e^{i(lx+my+zn)}$$

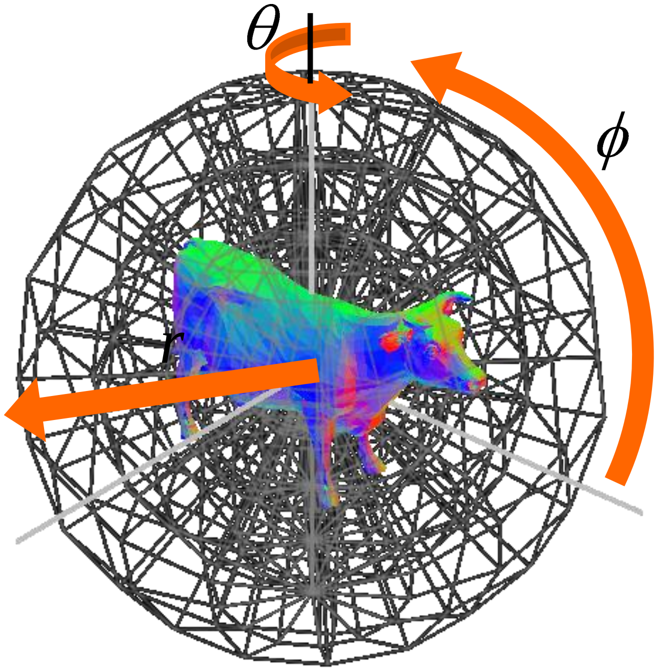
$$\left\|f_{l,m,n}\right\|_{l,m,n}$$

Translation Invariant Representation

Invariance

Rotation:

Compute the spherical harmonic decomposition and store just the sizes of the different frequency components of the different radial restrictions.



Spherical Coordinates

$$f_r(\theta,\phi)=\sum_{l=0}^{\infty}\sum_{k=-l}^{l}\hat{f}_r(l,k)Y_l^k(\theta,\phi)$$

$$\left\{\sqrt{\sum_{k=-l}^{l}\left\|\hat{f}_r(l,k)\right\|^2}\sqrt{4\pi r^2}\right\}_{l=0}^{\infty}$$

Rotation Invariant Representation

Overblown Claim

All methods that represent 3D shapes in either a translation-invariant or rotation-invariant method implicitly use these invariance approaches.

Goal

Given the three shape descriptors:

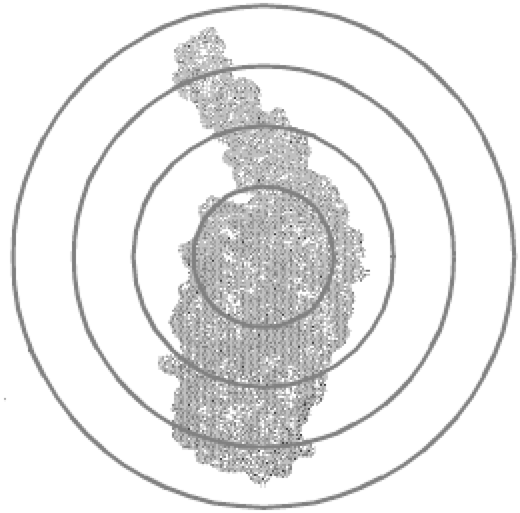
- ○ Shape Histograms
- ○ Shape Distributions
- ○ Extended Gaussian Images

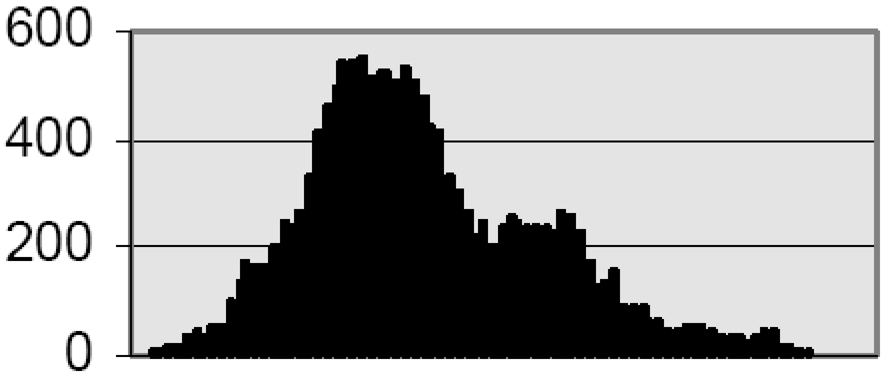
- How does the descriptor obtain its invariance?
- How can the descriptiveness of the descriptor be improved while maintaining the invariance property?

Shape Histograms

This shape descriptor represents a 3D shape by a 1D histogram.

It is obtained by binning points by their distance from the center and is rotation invariant.

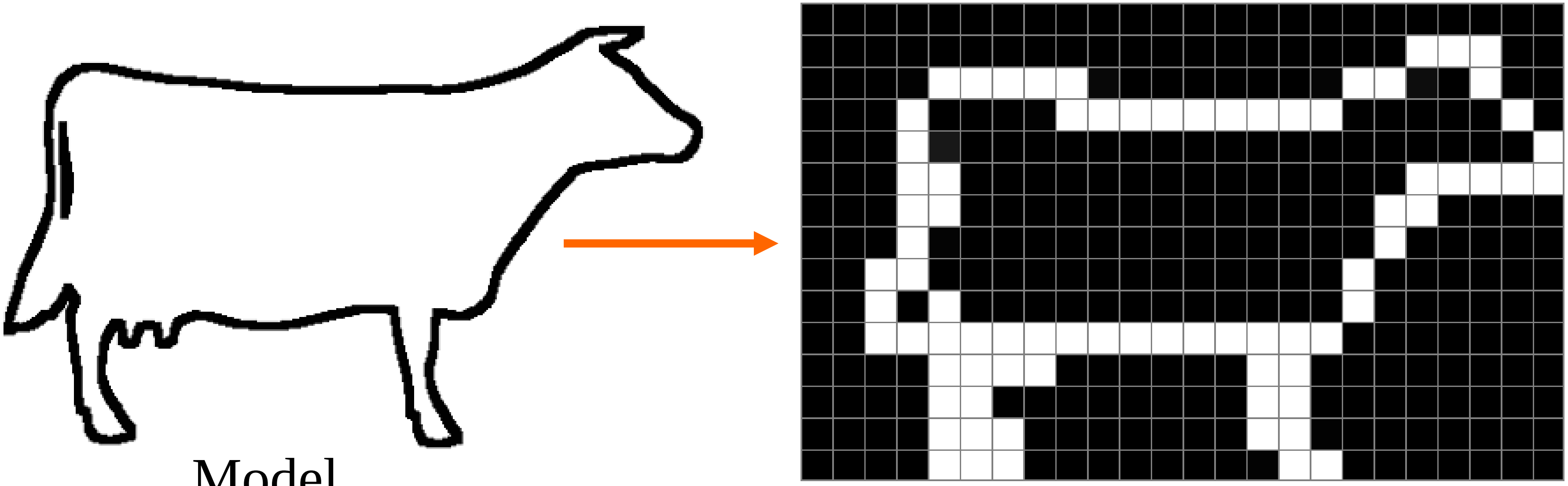




Shape Histograms

The shape histogram starts by representing the surface by a 3D function, obtained by rasterizing the boundary into a voxel grid:

- A voxel has value 1 if intersects the boundary
- A voxel has value 0 otherwise.



Model

Rasterization

Shape Histograms

The shape histogram can then be obtained by setting the value of the bin corresponding to radius *r* equal to the "size" of the rasterization restricted to the sphere of radius *r*:

$$\text{ShapeHistogram}(r) = \int_{|p|=r} \text{Raster}(p)\,dp$$

Shape Histograms

We can express the rasterization in spherical coordinates:

$$R(r,\theta,\phi) = \text{Raster}\left(r\cos\theta\sin\phi, r\cos\phi, r\sin\theta\sin\phi\right)$$

Shape Histograms

We can express the rasterization in spherical coordinates:

$$R(r,\theta,\phi) = \text{Raster}\left(r\cos\theta\sin\phi, r\cos\phi, r\sin\theta\sin\phi\right)$$

Then, for each radius *r*, we get a spherical function:

$$R_r(\theta,\phi) = R(r,\theta,\phi)$$

Shape Histograms

We can express the rasterization in spherical coordinates:

$$R(r,\theta,\phi)=\mathrm{Raster}\left(r\cos\theta\sin\phi, r\cos\phi, r\sin\theta\sin\phi\right)$$

Then, for each radius *r*, we get a spherical function:

$$R_r(\theta,\phi)=R(r,\theta,\phi)$$

Which we can express in terms of the spherical harmonics:

$$R_r(\theta,\phi)=\sum_{l=0}^{\infty}\sum_{k=-l}^{l}\hat{R}_r(l,k)Y_l^k(\theta,\phi)$$

Shape Histograms

In this formulation, the value of the shape histogram at a radius of *r* is just the value of the 0th spherical harmonic coefficient:

$$\text{ShapeHistogram}(r) = \hat{R}_r(0,0)\sqrt{4\pi r^2}$$

where the scale factor of the square root of $4\pi r^2$ accounts for the fact that the area of the sphere of radius *r* is $4\pi r^2$.

Shape Histograms

In sum, the shape histogram obtains its rotation invariance by storing the (size of the) 0th order frequency component:

$$\text{ShapeHistogram}(r) = \hat{R}_r(0,0)\sqrt{4\pi r^2}$$

Shape Histograms

In sum, the shape histogram obtains its rotation invariance by storing the (size of the) 0th order frequency component:

$$\text{ShapeHistogram}(r) = \hat{R}_r(0,0)\sqrt{4\pi r^2}$$

<u>Extension</u>:

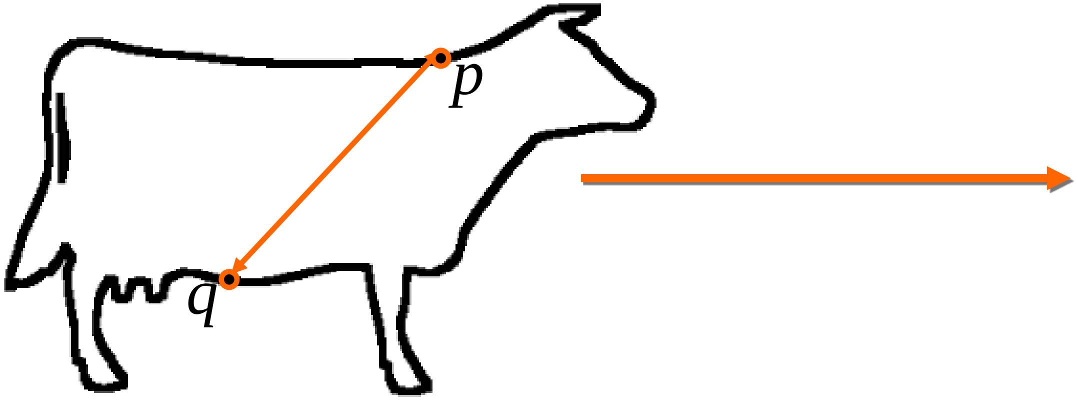
We can obtain a more descriptive representation, without giving up rotation invariance, by storing the size of <u>every</u> frequency component.

$$\text{EShapeHist ogram}(r,l) = \sqrt{\sum_{k=-l}^{l} \left(R_r(l,k) \right)^2}\sqrt{4\pi r^2}$$

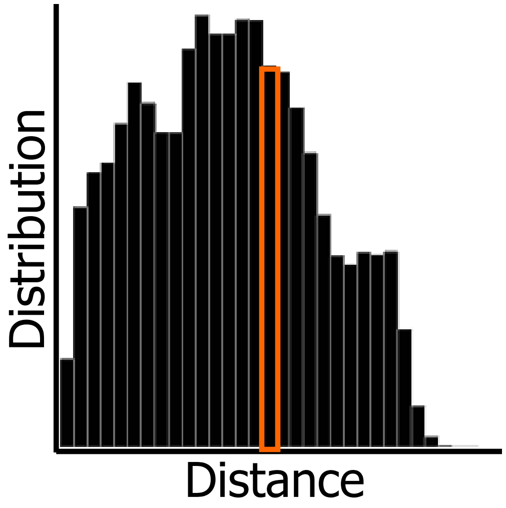
D2 Shape Distribution

This shape descriptor represents a 3D shape by a 1D histogram.

It is obtained by binning point-pairs by their distance, and is both translation and rotation invariant.



3D Model



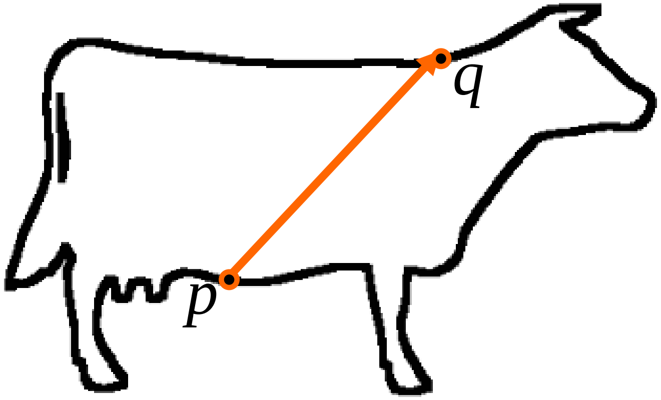
D2 Distribution

D2 Shape Distribution

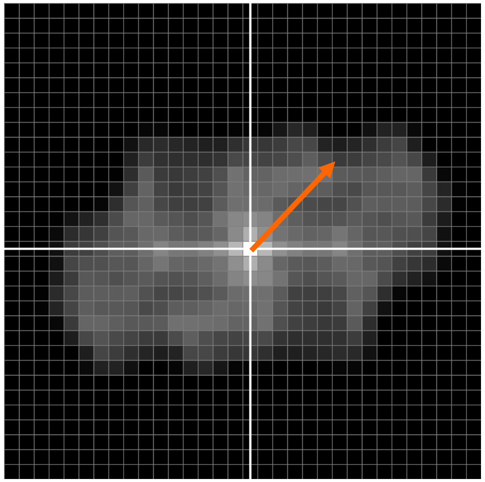
Let’s consider the rotation invariance first.

D2 Shape Distribution

One way to think of the D2 shape descriptor is by binning the difference vector between pairs of points on the surface:



3D Model

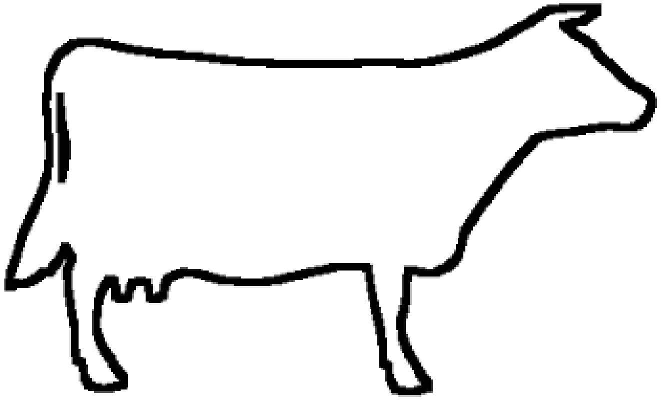


Binned Difference Vectors

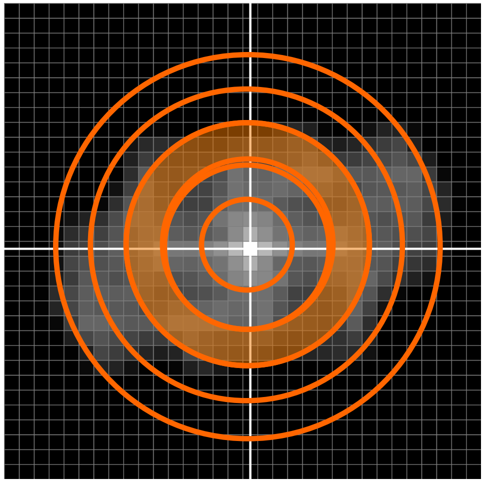
D2 Shape Distribution

One way to think of the D2 shape descriptor is by binning the difference vector between pairs of points on the surface.

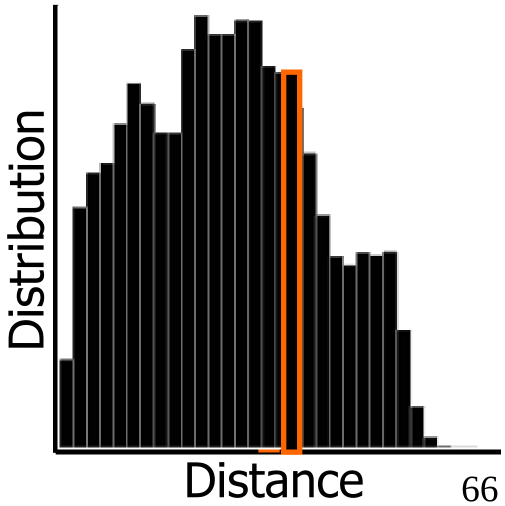
Then the shape distribution can be obtained by computing the Shape Histogram of the binning:



3D Model



Binned Difference Vectors



D2 Shape Distribution

As with the Shape Histogram, the D2 Shape Distribution can be realized by storing 0th order frequency components of the spherical harmonic decomposition.

D2 Shape Distribution

As with the Shape Histogram, the D2 Shape Distribution can be realized by storing 0th order frequency components of the spherical harmonic decomposition.

Extension:

As with the Shape Histogram the representation can be made more descriptive, without sacrificing rotation invariance, by storing the size of every frequency component.

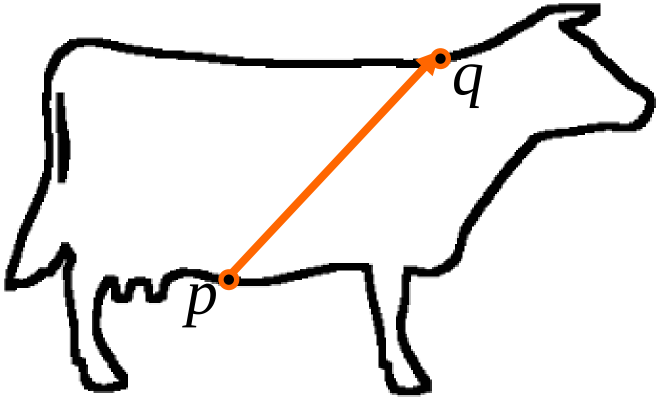
D2 Shape Distribution

This accounts for the rotation invariance of the D2 Shape Distribution.

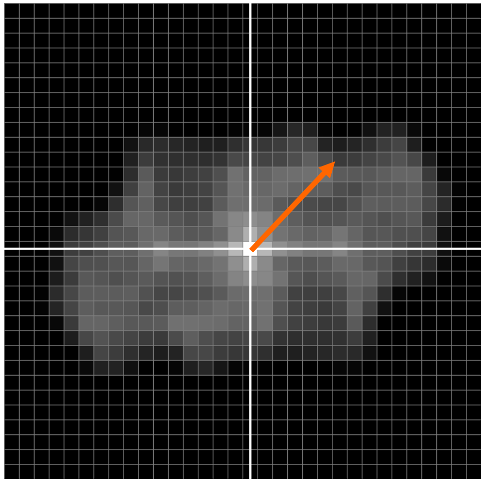
What makes it translation invariant?

D2 Shape Distribution

The Shape Distribution is computed from the binning of point-pair differences. How is this function computed?



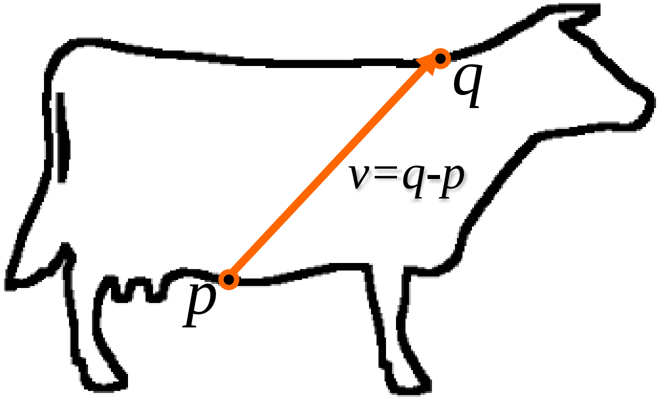
3D Model



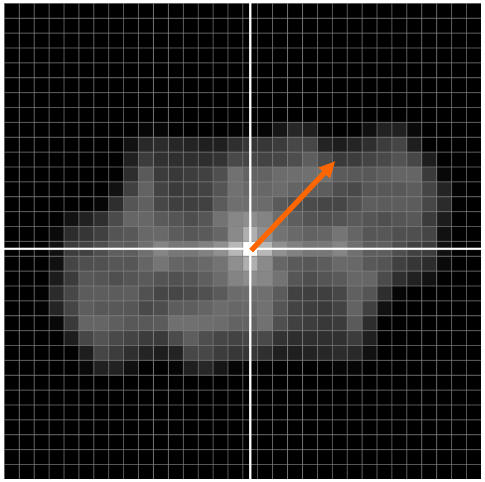
Binned Difference Vectors

D2 Shape Distribution

A point *q* on the surface will contribute to bin *v* if the point *q*-*v* is also on the surface.



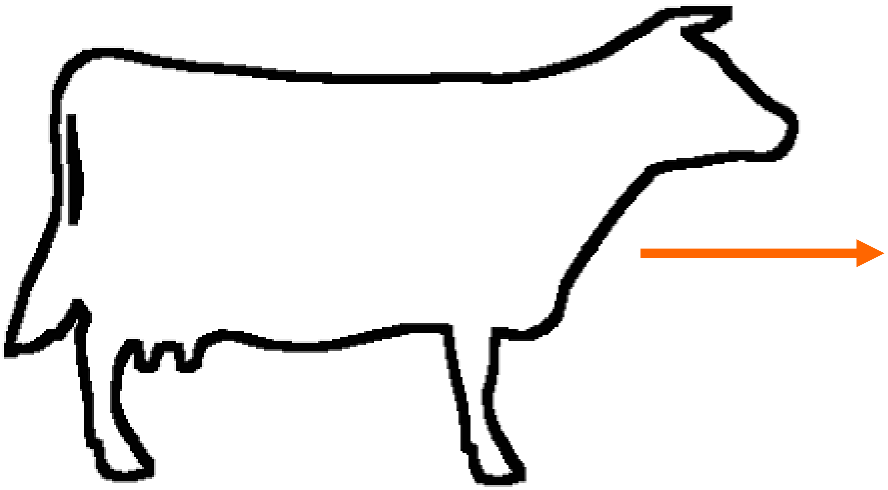
3D Model



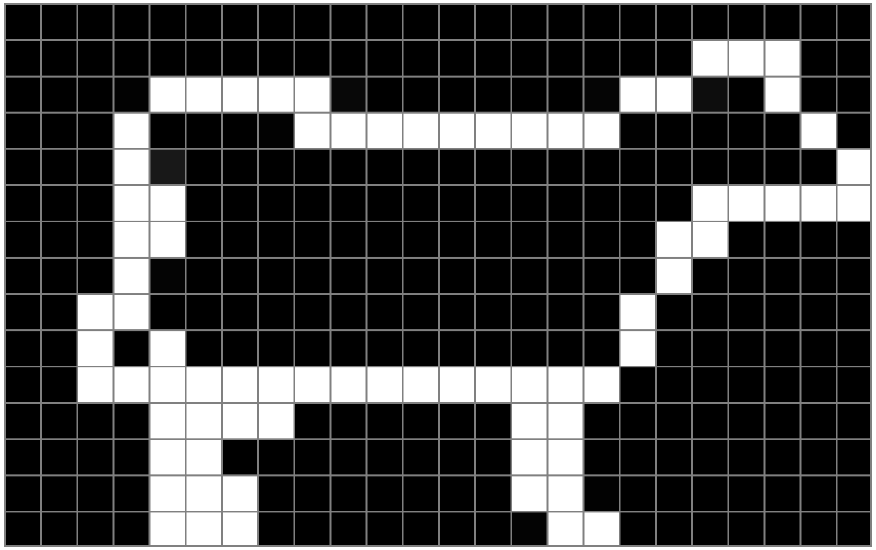
Binned Difference Vectors

D2 Shape Distribution

Once again, we consider the rasterization of the surface into a regular voxel grid.



Model



Rasterization

D2 Shape Distribution

A point *q* on the surface will contribute to bin *v* if the point *q*-*v* is also on the surface.

But a point *q*-*v* is on the surface if and only if:

$$\text{Raster}(q-v)=1$$

D2 Shape Distribution

A point *q* on the surface will contribute to bin *v* if the point *q-v* is also on the surface.

But a point *q-v* is on the surface if and only if:

$$\text{Raster}(q-v)=1$$

Since the value of the rasterization is 0 when *q-v* is not on the surface, this implies that:

$$\text{DBin}(v)=\int_{q\in\text{Surface}}\text{Raster}(q-v)\,dq$$

D2 Shape Distribution

For an arbitrary point in space, *q*, the point will only contribute to bin *v* if both *q* and *q*-*v* are on the surface.

D2 Shape Distribution

For an arbitrary point in space, *q*, the point will only contribute to bin *v* if both *q* and *q*-*v* are on the surface.

That, is *q* will contribute to bin *v* if and only if:

$$\text{Raster}(q)\cdot\text{Raster}(q-v)=1$$

D2 Shape Distribution

For an arbitrary point in space, *q*, the point will only contribute to bin *v* if both *q* and *q-v* are on the surface.

That, is *q* will contribute to bin *v* if and only if:

$$\mathrm{Raster}(q)\cdot\mathrm{Raster}(q-v)=1$$

Since this product is 0 when either *q* is not on the surface, or *q-v* is not on the surface, this gives:

$$\mathrm{DBin}(v)=\int\mathrm{Raster}(q)\cdot\mathrm{Raster}(q-v)dq$$

D2 Shape Distribution

Thus the binning function is just the cross-correlation of the rasterization with itself:

$$\text{DBin}(v) = \int \text{Raster}(q) \cdot \text{Raster}(q - v)\,dq$$

$$= \left(\text{Raster} * \text{Raster}\right)(v)$$

D2 Shape Distribution

But the Fourier decomposition of the cross-correlation of *f* with *g* can be obtained by multiplying the Fourier coefficients of *g* by the conjugates of the Fourier coefficients of *f*:

$$(f * g)(\theta) = \sum_{l=-\infty}^{\infty} \left(\overline{\hat{f}(l)} \cdot \hat{g}(l) \right) e^{il\theta}$$

D2 Shape Distribution

But the Fourier decomposition of the cross-correlation of *f* with *g* can be obtained by multiplying the Fourier coefficients of *g* by the conjugates of the Fourier coefficients of *f*:

$$(f * g)(\theta) = \sum_{l=-\infty}^{\infty} \left(\overline{\hat{f}(l)} \cdot \hat{g}(l) \right) e^{il\theta}$$

In the case that *f* and *g* are the same function, this gives:

$$(f * f)(\theta) = \sum_{l=-\infty}^{\infty} \left\| \hat{f}(l) \right\|^{2} e^{il\theta}$$

D2 Shape Distribution

Thus, the binning function implicitly converts the rasterization function into a function whose Fourier coefficients are the square norms of the Fourier coefficients of the rasterization.

D2 Shape Distribution

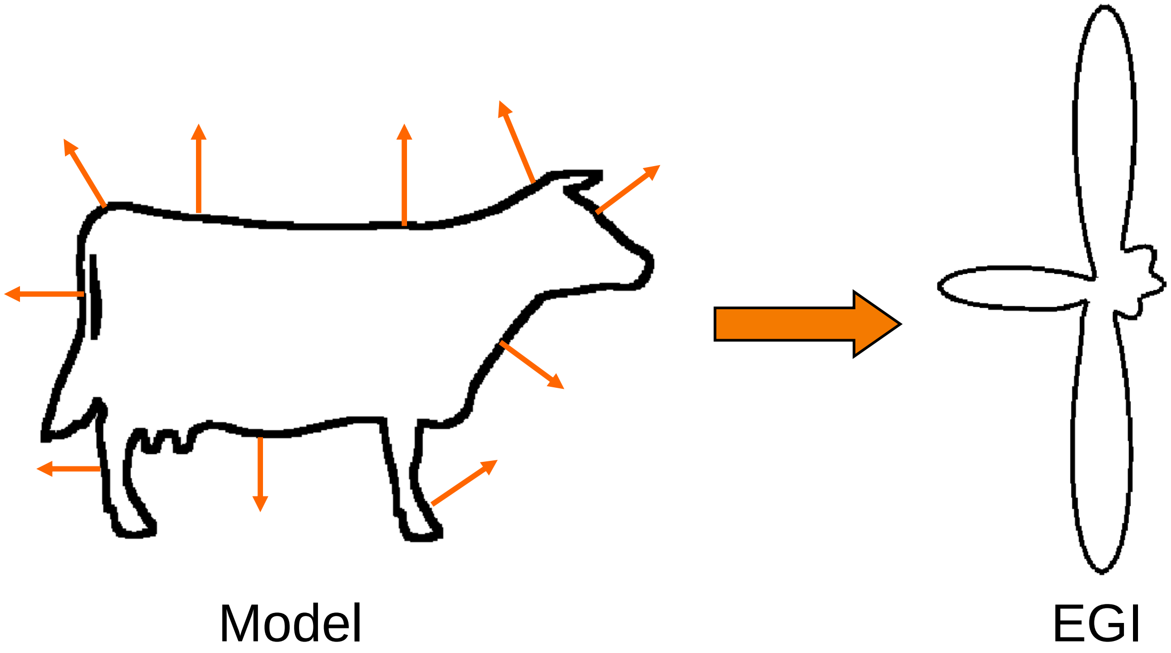
Thus, the binning function implicitly converts the rasterization function into a function whose Fourier coefficients are the square norms of the Fourier coefficients of the rasterization.

But this is what we do to make a function translation invariant.

Extended Gaussian Image

This spherical shape descriptor represents a 3D shape by a histogram on the sphere.

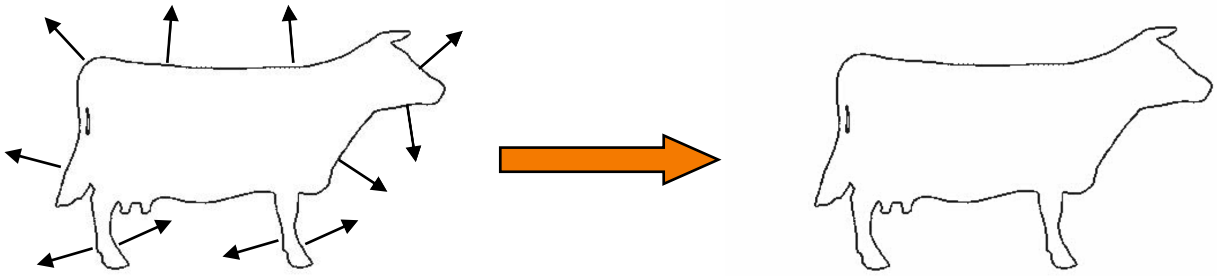
It is obtained by binning points by their normal direction, and is translation invariant.



Extended Gaussian Image

To obtain the EGI representation, we can think of points on the model as living in a 5D space:

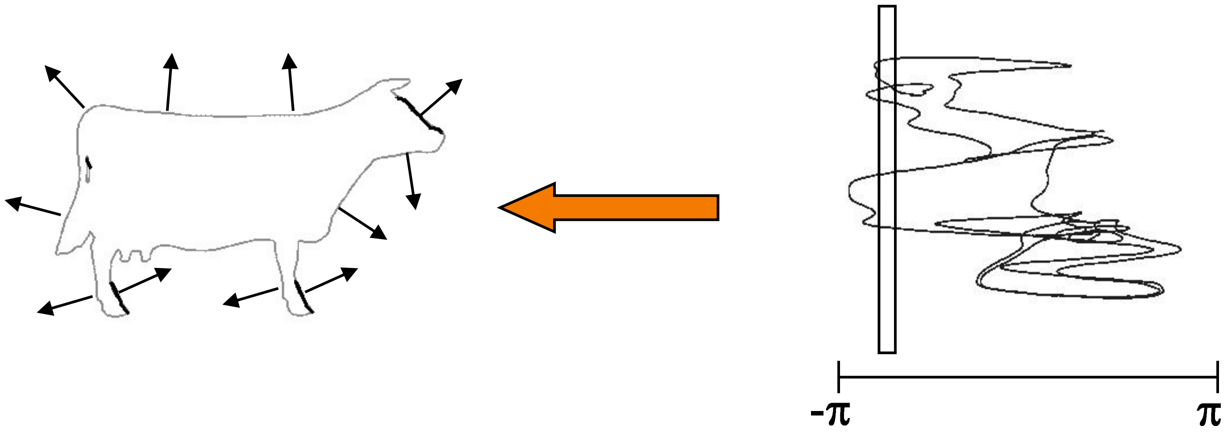
- The first three dimensions are indexed by the position.
- The last two dimensions are indexed by the normal direction.



Extended Gaussian Image

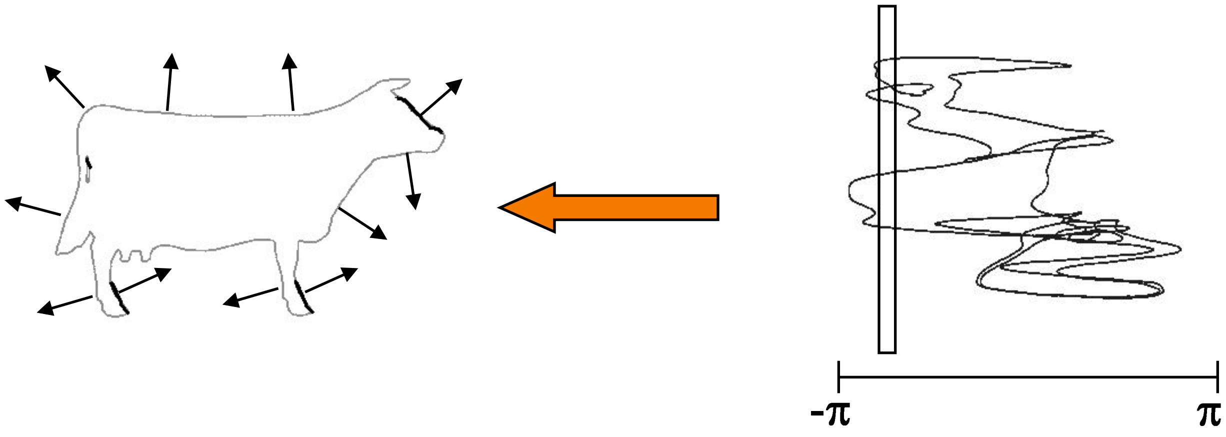
To obtain the EGI representation, we can think of points on the model as living in a 5D space.

If we fix the normal angle, we get a 3D slice of the 5D space, corresponding to all the points on the surface with the same normal:



Extended Gaussian Image

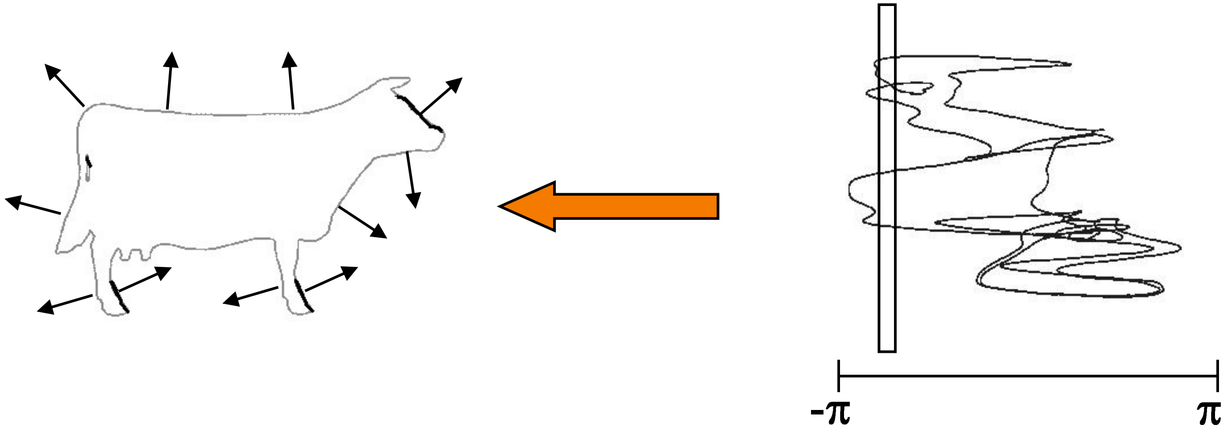
For each normal *n*, the EGI stores the “size” of the points in the normal slice corresponding to *n*.



Extended Gaussian Image

For each normal *n*, the EGI stores the “size” of the points in the normal slice corresponding to *n*.

This is just the 0th order frequency component of the rasterization of the points on the model with normal *n*.



Extended Gaussian Image

For each normal *n*, the EGI stores the “size” of the points in the normal slice corresponding to *n*.

This is just the 0th order frequency component of the rasterization of the points on the model with normal *n*.

Extension:

We can get a more discriminating descriptor, without giving up translation invariance, by storing the size of every frequency component.