FFTs in Graphics and Vision

Fast Alignment of Spherical Functions

Outline

- Math Review
- Fast Rotational Alignment

Review

Recall 1:

We can represent any rotation R in terms of the triplet of Euler angles (θ, ϕ, ψ), with the correspondence defined by:

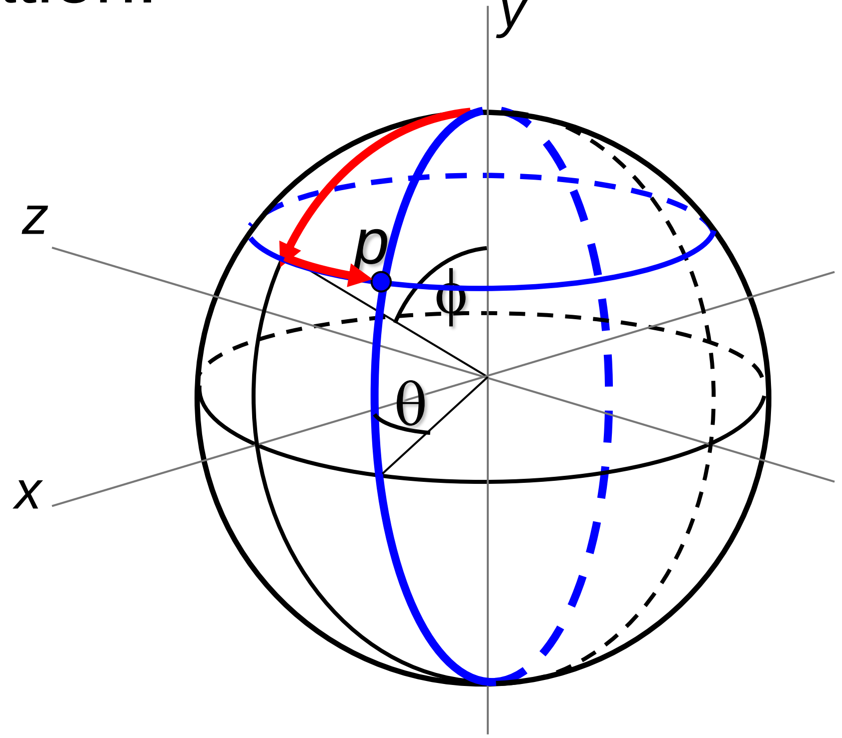
$$R(\theta, \phi, \psi) = R_y(\theta) \cdot R_z(\phi) \cdot R_y(\psi)$$

where $R_y(\alpha)$ is the rotation about the y-axis by an angle of α, and $R_z(\beta)$ is the rotation about the z-axis by an angle of β.

Review

<u>Recall 2</u>:

Expressing a rotation in terms of its Euler angles (θ, ϕ, ψ), the angles (θ, ϕ) describe the image of the North pole after the rotation.



Review

Recall 3:

Expressing a rotation in terms of its Euler angles (θ, ϕ, ψ), the inverse rotation is given by the Euler angles $(-\psi, -\phi, -\theta)$:

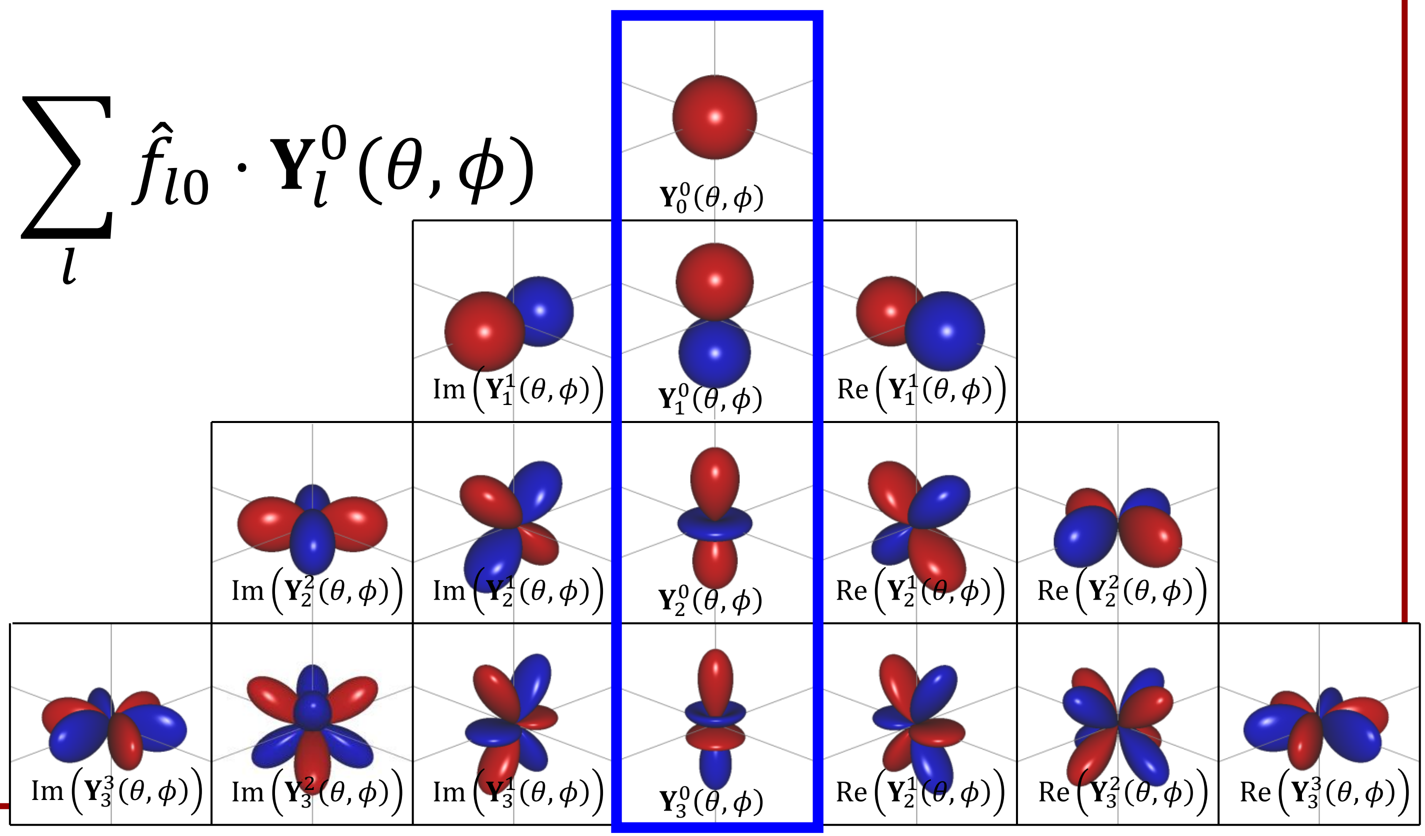
$$\begin{aligned} R^{-1}(\theta, \phi, \psi) &= \left(R_y(\theta) \cdot R_z(\phi) \cdot R_y(\psi)\right)^{-1} \\ &= R_y^{-1}(\psi) \cdot R_z^{-1}(\phi) \cdot R_y^{-1}(\theta) \\ &= R_y(-\psi) \cdot R_z(-\phi) \cdot R_y(-\theta) \end{aligned}$$

Review

<u>Recall 4</u>:

A function f is axially symmetric about the y-axis if and only if it is composed entirely of the zonal harmonics:

$$f(\theta,\phi)=\sum_{l}\hat{f}_{l0}\cdot\mathbf{Y}_{l}^{0}(\theta,\phi)$$



Review

Recall 5:

Rotating the spherical harmonic $\mathbf{Y}_l^m$ about the y-axis by angle α is the same as multiplying by $e^{-im\alpha}$.

Expressing the spherical harmonic in terms of the associated Legendre polynomials, we get:

$$\mathbf{Y}_l^m(\theta,\phi) = \mathbf{P}_l^m(\cos\phi)\cdot e^{im\theta}$$

Review

Recall 5:

Rotating the spherical harmonic $\mathbf{Y}_l^m$ about the y-axis by angle α is the same as multiplying by $e^{-im\alpha}$.

Rotating by α about the y-axis gives:

$$\begin{aligned}\rho_{R_y(\alpha)}(\mathbf{Y}_l^m)(\theta,\phi) &= \mathbf{Y}_l^m(\theta-\alpha,\phi)\\ &= \mathbf{P}_l^m(\cos\phi)\cdot e^{im(\theta-\alpha)}\\ &= e^{-im\alpha}\cdot\mathbf{P}_l^m(\cos\phi)\cdot e^{im\theta}\\ &= e^{-im\alpha}\cdot\mathbf{Y}_l^m(\theta,\phi)\end{aligned}$$

Review

Recall 6:

If f is axially symmetric about the y-axis, then a rotation of f by a rotation with Euler angles (θ, ϕ, ψ) is independent of the value of ψ:

$$\begin{aligned}\rho_{R(\theta,\phi,\psi)}(f) &= \rho_{R_y(\theta)\cdot R_z(\phi)\cdot R_y(\psi)}(f)\\ &= \rho_{R_{y(\theta)}}\left(\rho_{R_z(\phi)}\left(\rho_{R_y(\psi)}(f)\right)\right)\\ &= \rho_{R_{y(\theta)}}\left(\rho_{R_z(\phi)}(f)\right)\end{aligned}$$

Review

Recall 7:

Given a spherical function f of frequency l:

$$f = \sum_{m=-l}^{l} \hat{f}_{lm} \cdot \mathbf{Y}_l^m$$

correlating the l-th zonal harmonic with f is the same as scaling the conjugate of f:

$$\left\langle \rho_{R(\theta,\phi,\psi)}\left(\mathbf{Y}_l^0\right), \bar{f} \right\rangle = \sqrt{\frac{4\pi}{2l+1}} f(\theta,\phi)$$

Review

Recall 8:

Given spherical functions f and g, if f is axially symmetric about the y-axis, we can compute the correlation of f with g in $O(N^2 \log^2 N)$.

In terms of the spherical harmonic decomposition, this equation becomes:

$$D_{f,g}(R) = \langle \rho_R(f), g \rangle$$

$$\Downarrow$$

$$D_{f,g}(\theta,\phi,\psi) = \left\langle \rho_{R(\theta,\phi,\psi)}\left(\sum_l \hat{f}_{l0} \cdot \mathbf{Y}_l^0\right), \sum_l \sum_{m=-l}^{l} \hat{g}_{lm} \cdot \mathbf{Y}_l^m \right\rangle$$

Review

<u>Recall 8</u>:

Given spherical functions f and g, if f is axially symmetric about the y-axis, we can compute the correlation of f with g in $O(N^2 \log^2 N)$.

By the conjugate linearity and the fact that harmonics of degree l form a sub-representation:

$$D_{f,g}(\theta,\phi,\psi) = \left\langle \rho_{R(\theta,\phi,\psi)}\left(\sum_l \hat{f}_{l0} \cdot \mathbf{Y}_l^0\right), \sum_l \sum_{m=-l}^{l} \hat{g}_{lm} \cdot \mathbf{Y}_l^m \right\rangle$$

$$\Downarrow$$

$$D_{f,g}(\theta,\phi,\psi) = \sum_l \sum_{m=-l}^{l} \hat{f}_{l0} \cdot \overline{\hat{g}_{lm}} \left\langle \rho_{R(\theta,\phi,\psi)}\left(\mathbf{Y}_l^0\right), \mathbf{Y}_l^m \right\rangle$$

Review

Recall 8:

Given spherical functions f and g, if f is axially symmetric about the y-axis, we can compute the correlation of f with g in $O(N^2 \log^2 N)$.

Which simplifies to:

$$D_{f,g}(\theta,\phi,\psi) = \sum_l \sum_{m=-l}^{l} \hat{f}_{l0} \cdot \overline{\hat{g}_{lm}} \left\langle \rho_{R(\theta,\phi,\psi)}\left(\mathbf{Y}_l^0\right), \mathbf{Y}_l^m \right\rangle$$

$$\Downarrow$$

$$D_{f,g}(\theta,\phi,\psi) = \sum_l \sum_{m=-l}^{l} \hat{f}_{l0} \cdot \overline{\hat{g}_{lm}} \sqrt{\frac{4\pi}{2l+1}} \cdot \overline{\mathbf{Y}_l^m(\theta,\phi)}$$

Review

Recall 8:

$$D_{f,g}(\theta,\phi,\psi) = \sum_{l} \sum_{m=-l}^{l} \hat{f}_{l0} \cdot \overline{\hat{g}_{lm}} \sqrt{\frac{4\pi}{2l+1}} \cdot \overline{\mathbf{Y}_l^m(\theta,\phi)}$$

So we can compute the correlation by:

- Computing the spherical harmonic transforms.
  $O(N^2 \log^2 N)$
- Scaling the (l,m)-th harmonic coefficient of g by the $(l,0)$-th coefficient of f times $\sqrt{4\pi/2l+1}$.
  $O(N^2)$
- Computing the conjugate of the inverse transform.
  $O(N^2 \log^2 N)$

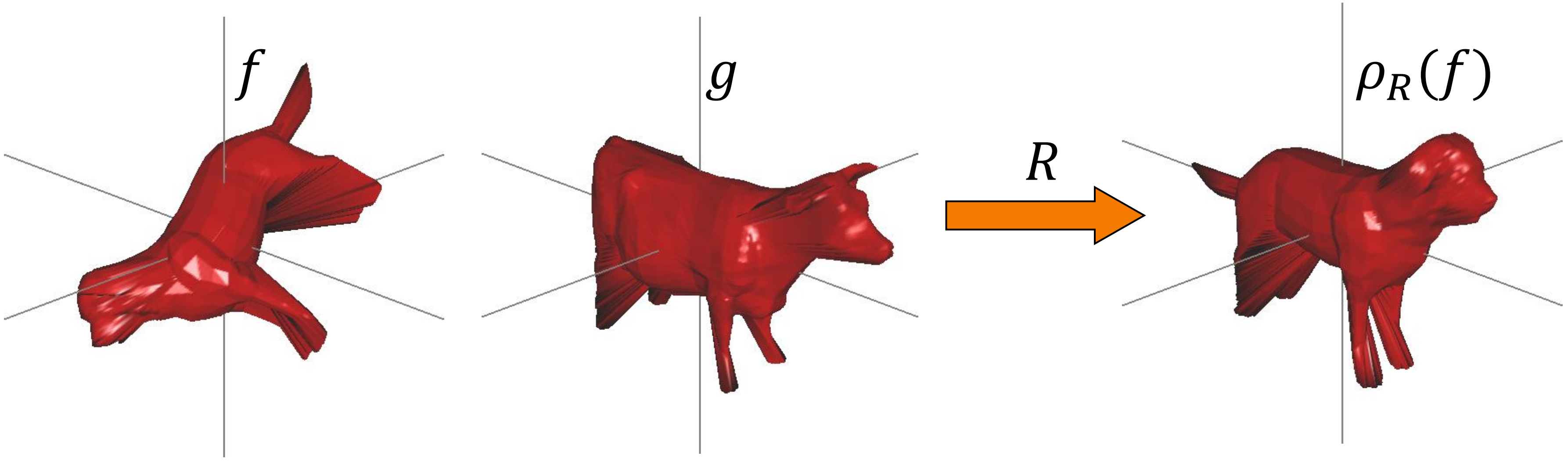
Outline

- Math Review
- Fast Rotational Alignment

Goal

Given two spherical functions f and g, we would like to find the rotation R that aligns f to g:

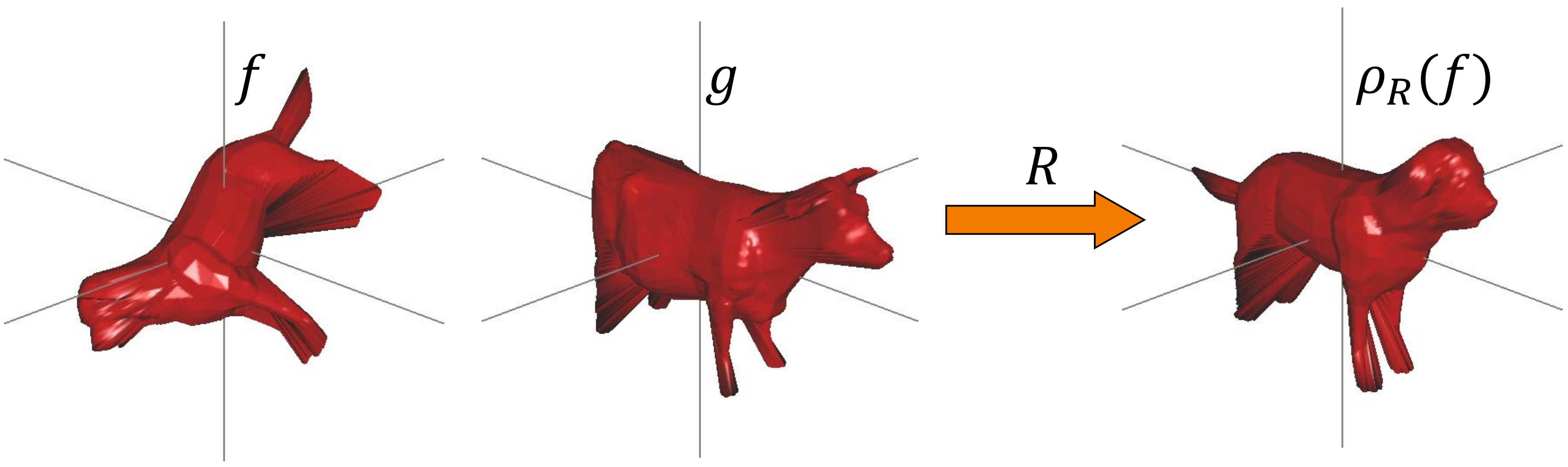
$$R = \underset{R \in SO(3)}{\arg\min} \|\rho_R(f) - g\|^2$$



Approach

We had shown that the rotation minimizing the difference is the rotation maximizing the correlation:

$$R = \underset{R \in SO(3)}{\arg\max} \langle \rho_R(f), g \rangle$$



Approach

Solving for the aligning rotation can be done by computing the function on the space of rotations:

$$D_{f,g}(R) = \langle \rho_R(f), g \rangle$$

and finding the rotation maximizing this function.

Approach

Brute Force:

If the resolution of the spherical grid is N, then we can find the optimal rotation in $O(N^5)$ time by:

- For each of $O(N^3)$ rotations
  - Compute the appropriate $O(N^2)$ dot-product

Approach

Fast Spherical Correlation:

Using the Wigner D-Transform, we can implement this in $O(N^3 \log^2 N)$ time by:

- Get the spherical harmonic coefficients of f and g.
  $O(N^2 \log^2 N)$
- Cross multiply the coefficients within each frequency to get the Wigner D-coefficients.
  $O(N^3)$
- Perform the inverse Wigner D-Transform to get the value of the correlation at every rotation.
  $O(N^3 \log^2 N)$

Efficiency

Although the Wigner D-Transform provides an algorithm that is faster than brute force, for many applications, a cubic algorithm is still too slow.

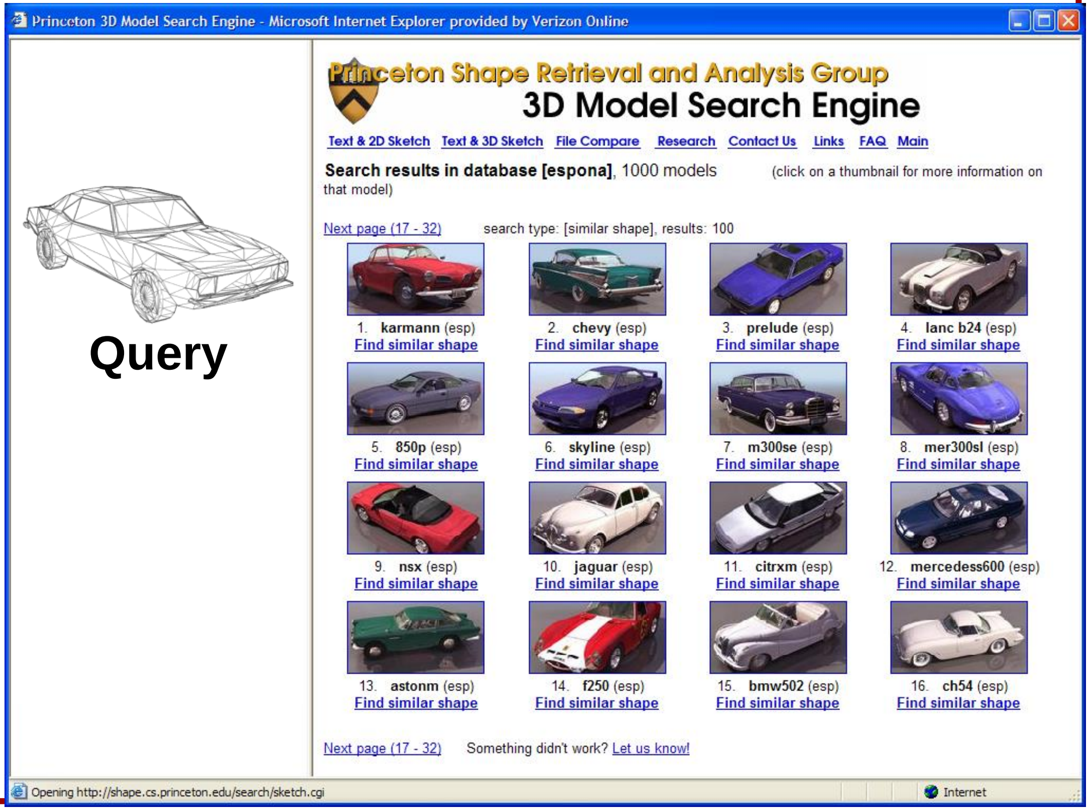
We would like an algorithm for aligning two functions that is on the order of the size of the spherical functions (i.e. quadratic in N).

Efficiency

Example:

For database retrieval, we would like to minimize the amount of work that needs to be done online.

We can afford to do more work on a per-model basis in pre-processing, but we can't spend too much time aligning pairs of models for matching.



Efficiency

Observation:

In using the Wigner D-Transform, we obtain the alignment error at every rotation.

All we want is the single, optimal rotation.

Parameter Optimization

Parameter Splitting:

Given a function $F(x,y)$, we would like to find the parameters (x^*,y^*) at which F is maximal:

$$(x^*,y^*) = \underset{(x,y)\in\mathbb{R}^2}{\arg\max}\big(F(x,y)\big)$$

We can find the parameters (x^*,y^*) by searching over the entirety of the parameter domain to find the parameters at which F is maximal.

This would require a search over a large space of parameters.

Parameter Optimization

Parameter Splitting:

Given a function $F(x,y)$, we would like to find the parameters (x^*,y^*) at which F is maximal:

$$(x^*,y^*) = \underset{(x,y)\in\mathbb{R}^2}{\arg\max}\big(F(x,y)\big)$$

Instead, we can try to decompose the optimization problem into two parts:

- First, find the optimal value for x^*, and then
- Holding x^* fixed, find the optimal value for y^*.

This way, we trade one search over a large space, for two searches over smaller spaces.

Parameter Optimization

<u>Parameter Splitting</u>:

To do this, we need to define a 1D function $G(x)$ with the property that if (x^*, y^*) maximizes $F(x, y)$ then x^* maximizes $G(x)$.

$$(x^*, y^*) = \underset{(x,y)\in\mathbb{R}^2}{\arg\max}\big(F(x, y)\big)$$

$$\Downarrow$$

$$x^* = \underset{x\in\mathbb{R}}{\arg\max}(G(x))$$

$$y^* = \underset{y\in\mathbb{R}}{\arg\max}\big(F(x^*, y)\big)$$

Parameter Optimization

Application to Rotational Alignment:

To find the optimal alignment, we would like to find the Euler angles $(\theta^*, \phi^*, \psi^*)$ that maximize the correlation:

$$(\theta^*, \phi^*, \psi^*) = \arg\max_{(\theta,\phi,\psi)} \left\langle \rho_{R_y(\theta)\cdot R_z(\phi)\cdot R_y(\psi)}(f), g \right\rangle$$

Parameter Optimization

<u>Application to Rotational Alignment</u>:

Instead of optimizing over the three parameters simultaneously, we can first optimize over two of the parameters.

$$(\theta^*, \phi^*) = \underset{(\theta,\phi)}{\arg\max}\big(G(\theta,\phi)\big)$$

Then fixing the two optimal parameters, optimize over the third:

$$\psi^* = \underset{\psi}{\arg\max}\left\langle \rho_{R_y(\theta^*)\cdot R_z(\phi^*)\cdot R_y(\psi)}(f), g\right\rangle$$

Parameter Optimization

<u>Application to Rotational Alignment</u>:

To define the function $G(\theta,\phi)$ we choose a function that represents correlation information related to rotations defined by θ and ϕ.

Specifically, if we let h be the component of f that is axially symmetric about the y-axis:

$$h(\theta,\phi) = \sum_l \hat{f}_{l0} \cdot \mathbf{Y}_l^0(\theta,\phi)$$

we can define:

$$G(\theta,\phi) = \left\langle \rho_{R(\theta,\phi,0)}(h), g \right\rangle$$

Parameter Optimization

Application to Rotational Alignment:

$$G(\theta,\phi) = \langle \rho_{R(\theta,\phi,0)}(h), g \rangle$$

The function G has two properties:

- If f is already axially symmetric about the y-axis (i.e. $h = f$), the optimizing angles (θ^*, ϕ^*) give the optimal transformation.
- Since h is axially symmetric, we can find the optimizing angles (θ^*, ϕ^*) in $O(N^2 \log^2 N)$ using the fast spherical harmonic transform.

Parameter Optimization

Application to Rotational Alignment:

Having solved for the optimal angles (θ^*, ϕ^*), we can solve for the optimal ψ_0 by solving:

$$\psi^* = \arg\max_{\psi} \left\langle \rho_{R_y(\theta^*)\cdot R_z(\phi^*)\cdot R_y(\psi)}(f), g \right\rangle$$

Since the representation is unitary, this becomes:

$$\psi^* = \arg\max_{\psi} \left\langle \rho_{R_y(\psi)}(f), \rho_{R_z(-\phi^*)\cdot R_y(-\theta^*)}(g) \right\rangle$$

In terms of the spherical harmonics coefficients:

$$\psi^* = \arg\max_{\psi} \left\langle \sum_{l} \sum_{m=-l}^{l} \hat{f}_{lm} \cdot \rho_{R_y(\psi)}(\mathbf{Y}_l^m), \rho_{R_z(-\phi^*)\cdot R_y(-\theta^*)}(g) \right\rangle$$

Parameter Optimization

Application to Rotational Alignment:

Since rotation of the spherical harmonic $\mathbf{Y}_l^m$ about the y-axis by angle of α is multiplication by $e^{-im\alpha}$:

$$\psi^* = \arg\max_{\psi} \left\langle \sum_{l} \sum_{m=-l}^{l} \hat{f}_{lm} \cdot \rho_{R_y(\psi)}(\mathbf{Y}_l^m), \rho_{R_z(-\phi^*)\cdot R_y(-\theta^*)}(g) \right\rangle$$

$$\Downarrow$$

$$\psi^* = \arg\max_{\psi} \left\langle \sum_{l} \sum_{m=-l}^{l} \hat{f}_{lm} \cdot e^{-im\psi} \cdot \mathbf{Y}_l^m, \rho_{R_z(-\phi^*)\cdot R_y(-\theta^*)}(g) \right\rangle$$

Parameter Optimization

<u>Application to Rotational Alignment</u>:

Thus, to find ψ^*, we need to maximize:

$$\sum_l \sum_{m=-l}^{l} \left\langle \hat{f}_{lm} \cdot \mathbf{Y}_l^m, \rho_{R_z(-\phi^*)\cdot R_y(-\theta^*)}(g) \right\rangle e^{-im\psi}$$

This is an expression for a function of ψ as a sum of complex exponentials!

So we can get the values at every angle ψ by computing the inverse Fourier transform.

Parameter Optimization

Application to Rotational Alignment:

Thus, we can align two spherical function f and g in $O(N^2 \log^2 N)$ time by:

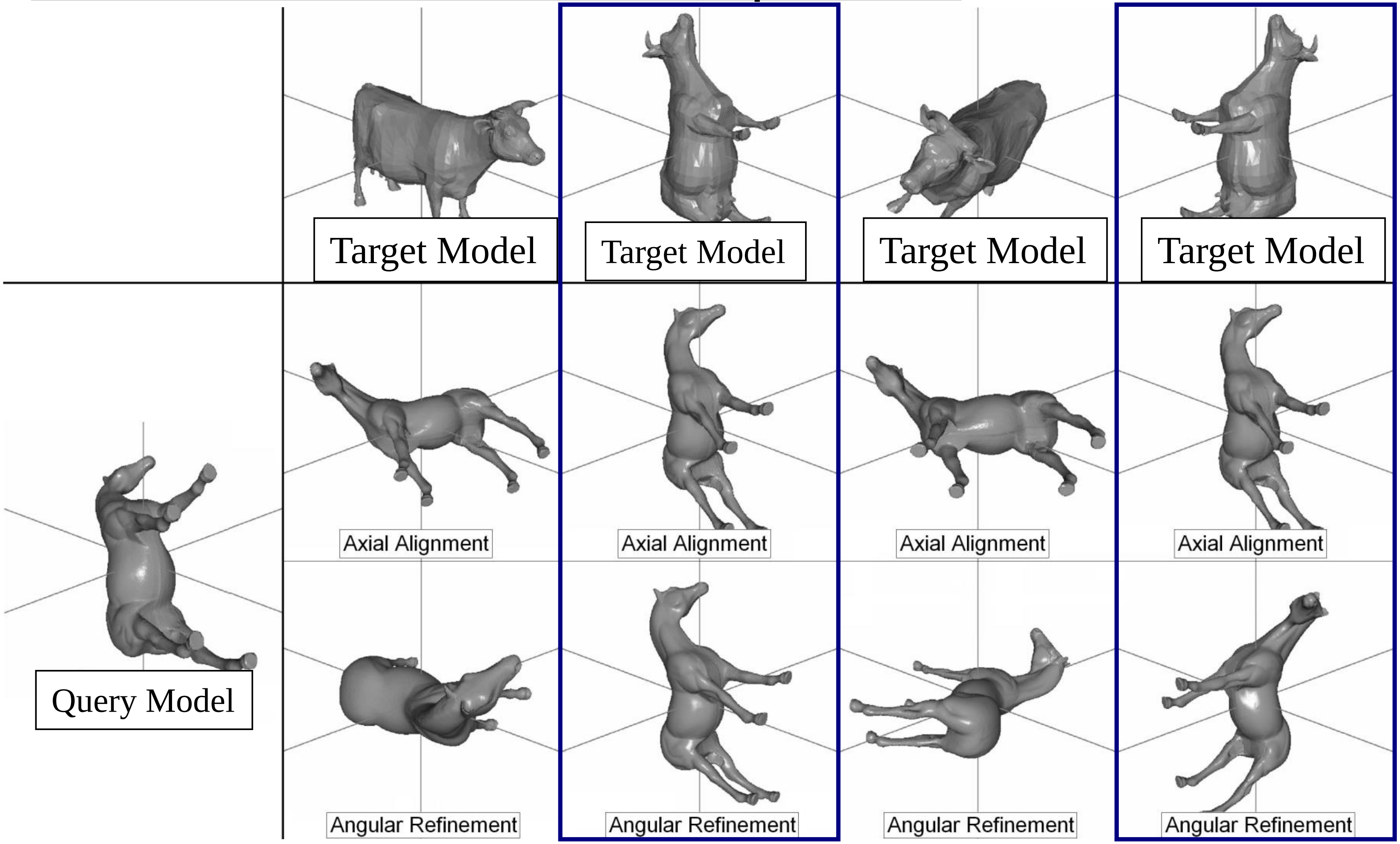
- Correlating f with the component of g that is axially symmetric about the y-axis
  $O(N^2 \log^2 N)$
- Getting the Fourier coefficients of the function in ψ
  $O(N^2)$
- Computing the inverse Fourier transform
  $O(N \log N)$
- Finding the ψ maximizing the function
  $O(N)$

Parameter Optimization

How well does this work in practice?

Parameter Optimization

How well does this work in practice?



Parameter Optimization

How well does this work in practice?

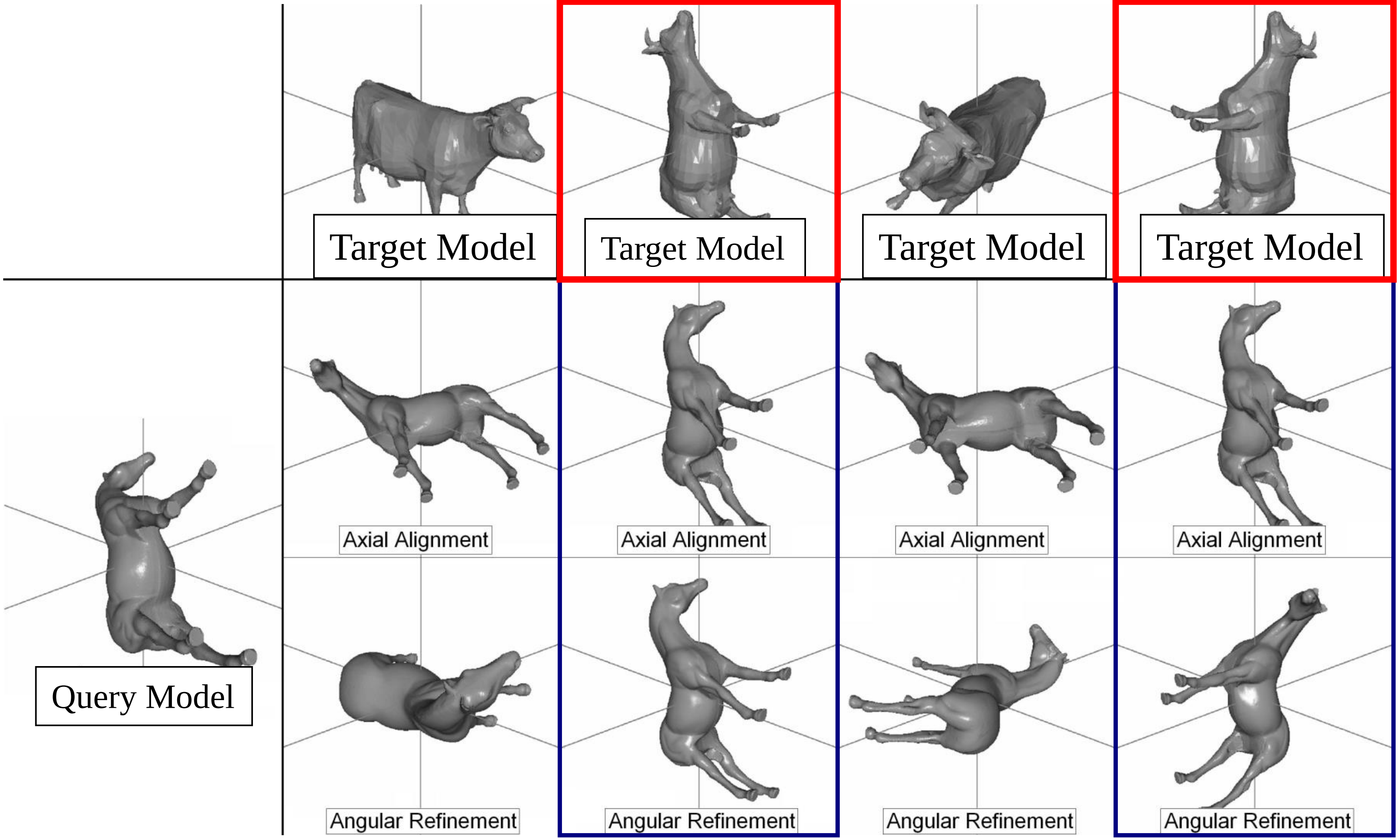
The quality of the alignment depends on how well $G(\theta, \phi)$ captures the behavior of the (θ, ϕ) components of the rotational alignment.

When we optimize $G(\theta, \phi)$, we are looking for the rotation that best aligns the y-axially symmetric component of f to the function g.

So if the function f is (nearly) axially symmetric about the y-axis, the method will perform well.

Parameter Optimization

How well does this work in practice?



Parameter Optimization

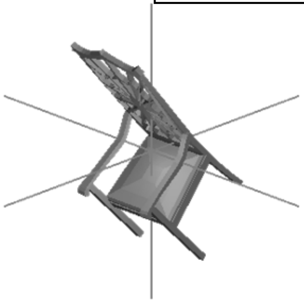
How well does this work in practice?

⇒ In a pre-processing step align the function f so that the axis with maximal axial symmetry gets mapped to the y-axis.

Parameter Optimization

How well does this work in practice?

Pre-Processing



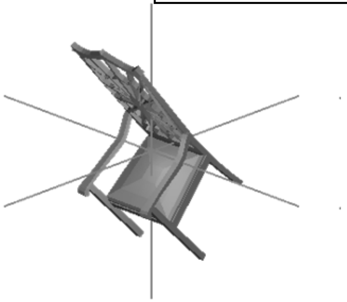
Run-Time

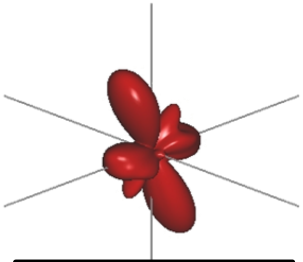
Parameter Optimization

How well does this work in practice?

Pre-Processing

Run-Time



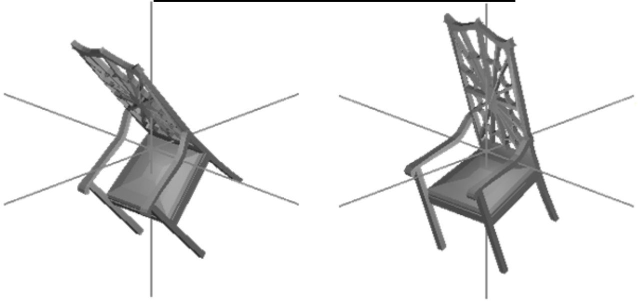


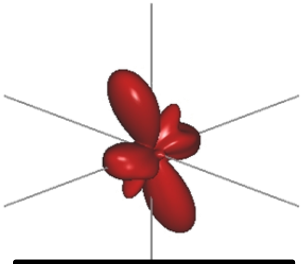
axial symmetry descriptor

Parameter Optimization

How well does this work in practice?

Pre-Processing





axial symmetry descriptor

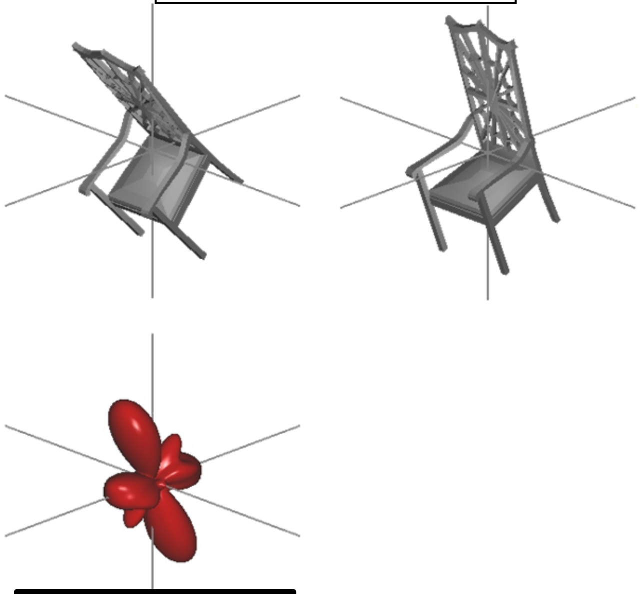
Run-Time

Parameter Optimization

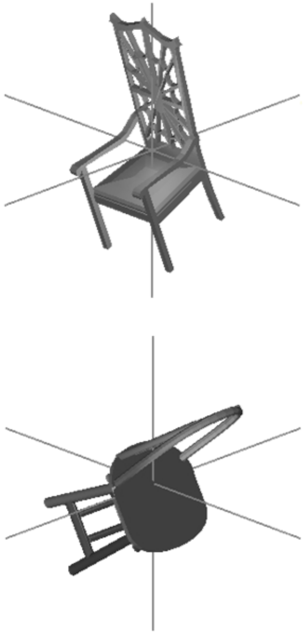
How well does this work in practice?

Pre-Processing

Run-Time



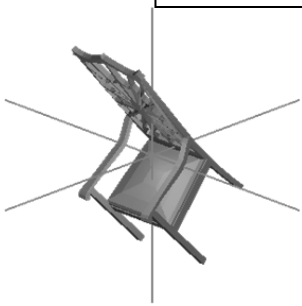
axial symmetry descriptor

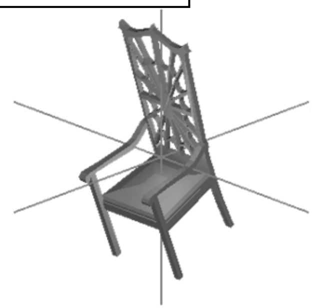


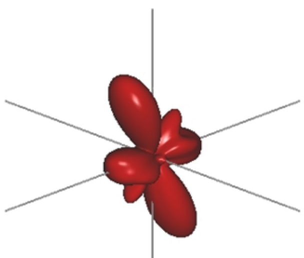
Parameter Optimization

How well does this work in practice?

Pre-Processing

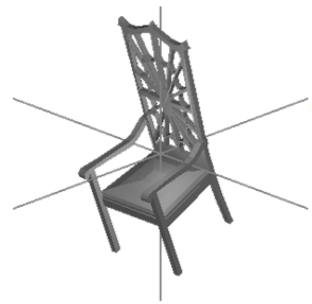


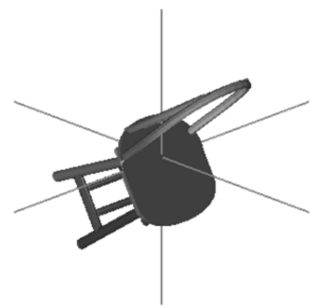


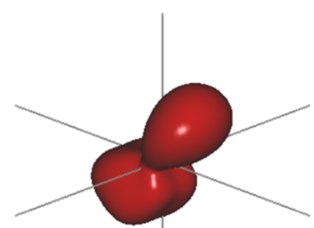


axial symmetry descriptor

Run-Time



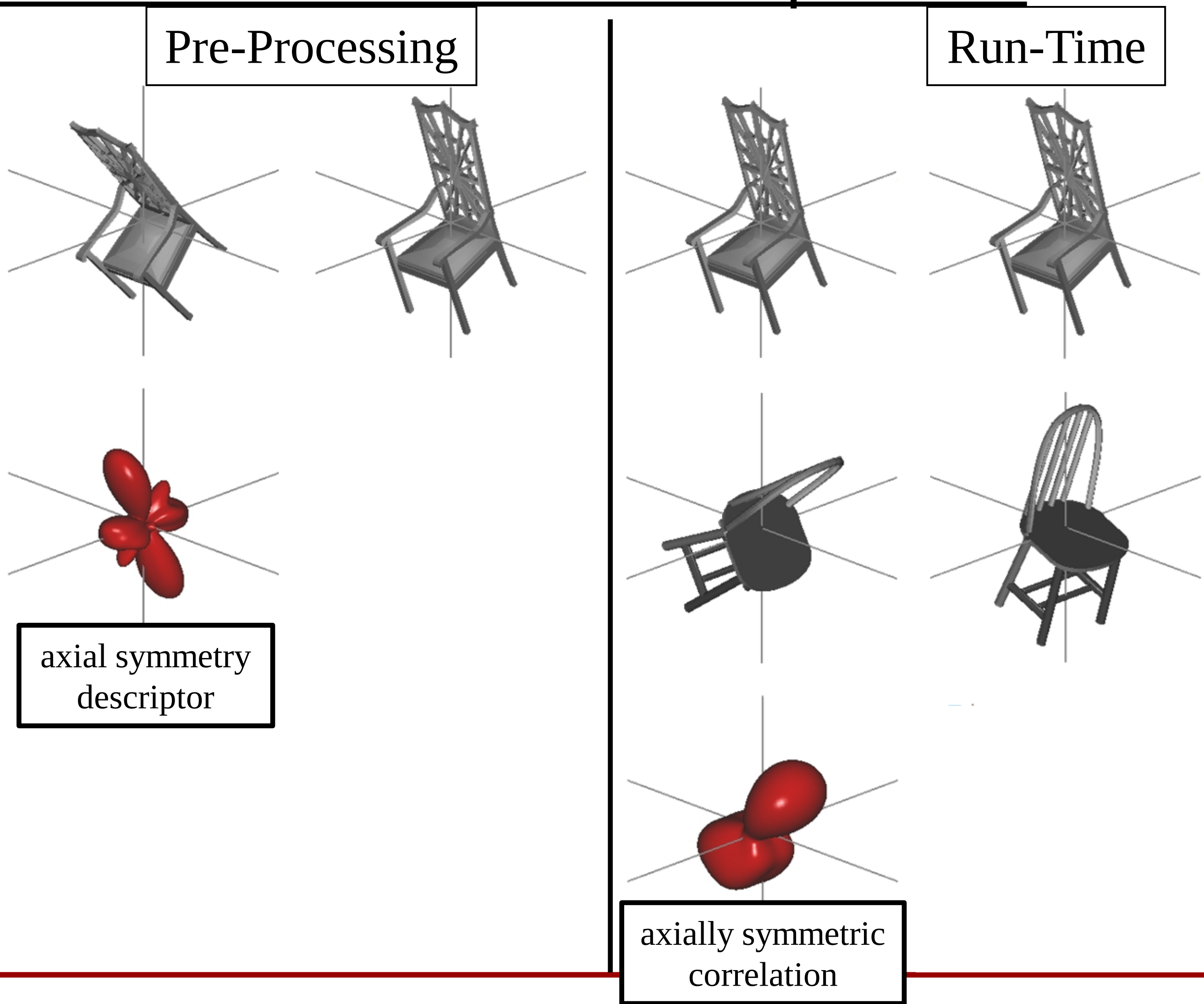




axially symmetric correlation

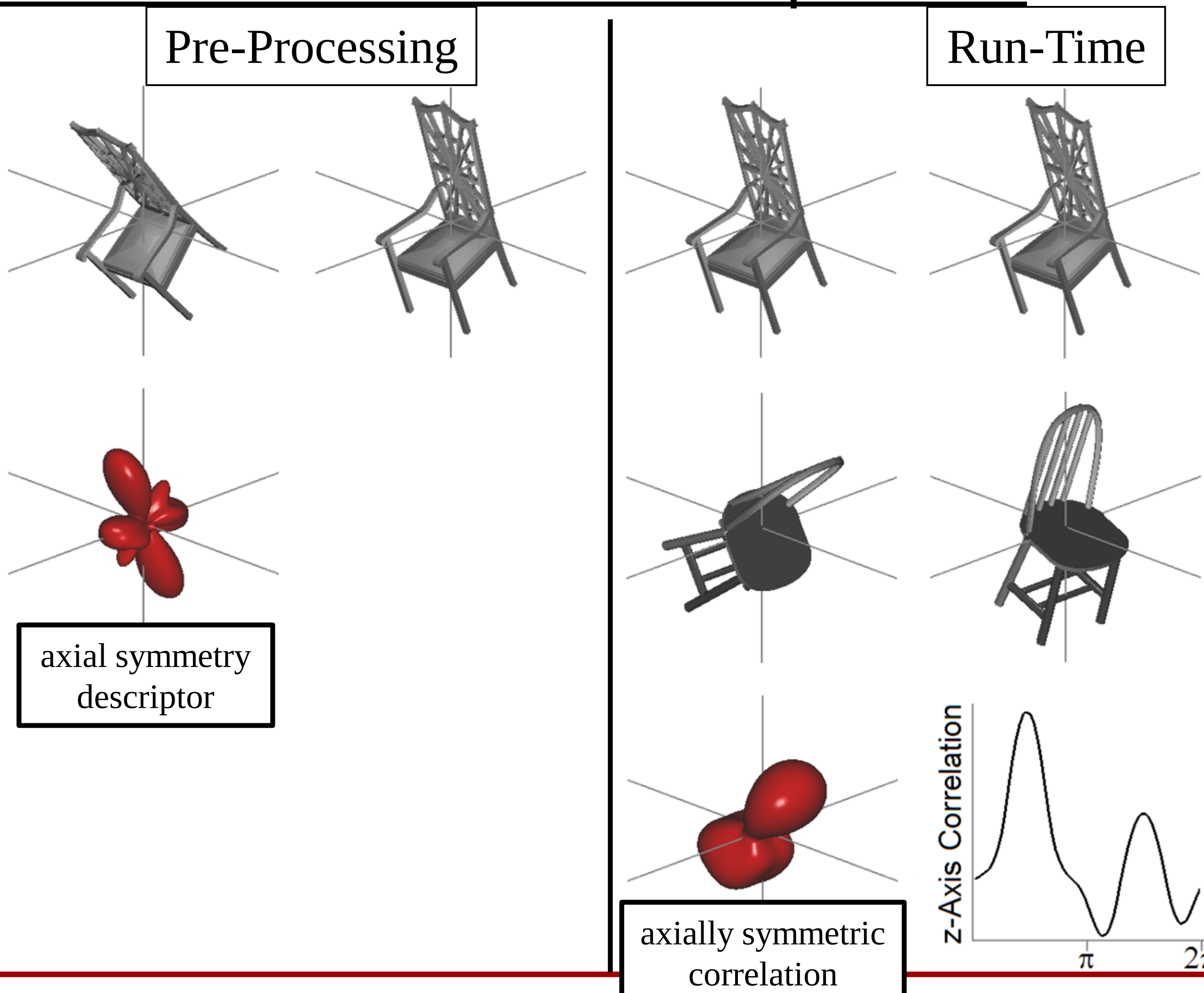
Parameter Optimization

How well does this work in practice?



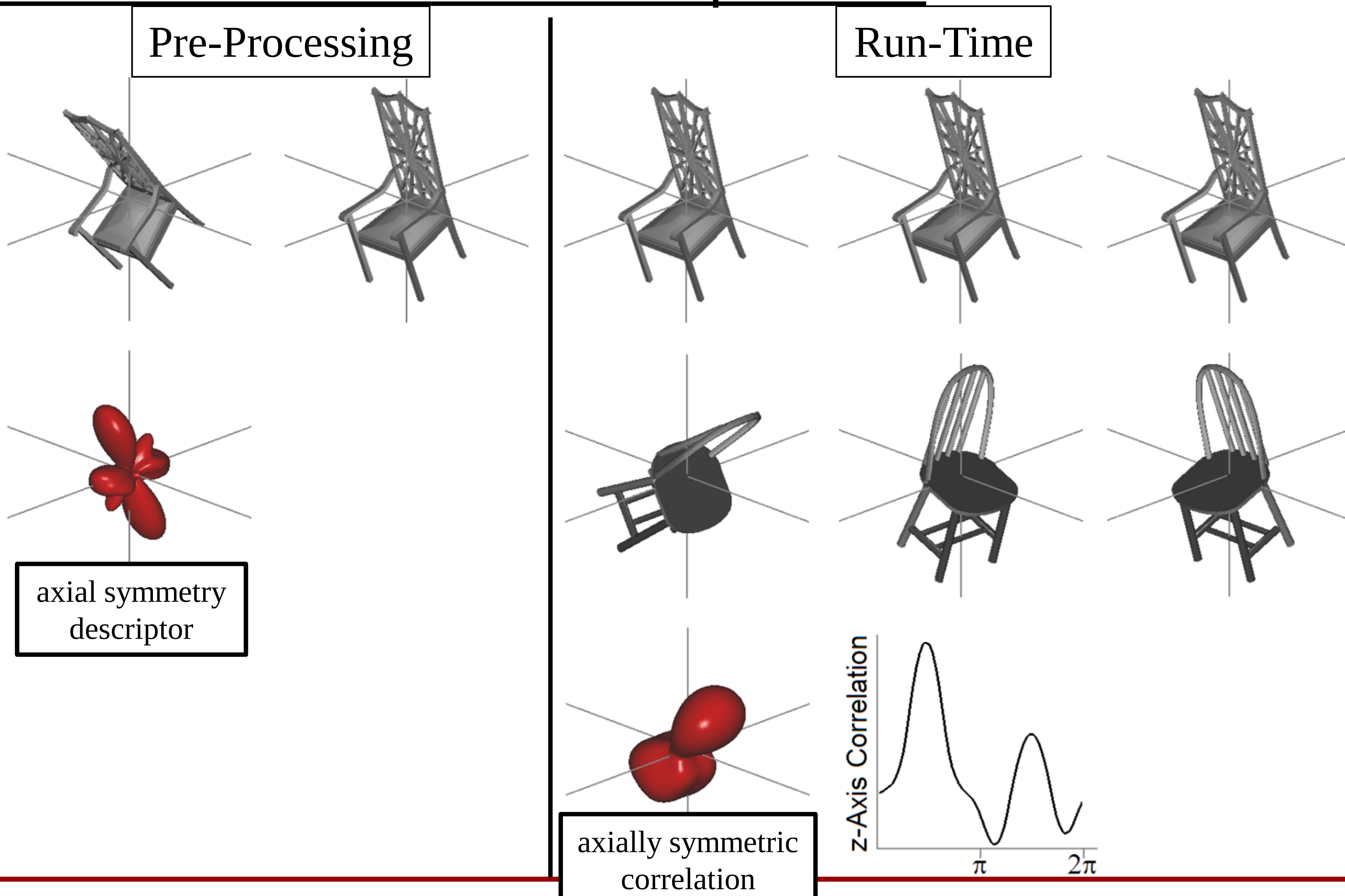
Parameter Optimization

How well does this work in practice?



Parameter Optimization

How well does this work in practice?



Parameter Optimization

Performance:

To perform the alignment we need to pre-align the models so that their major axis of axial symmetry is aligned with the y-axis.

- ☒ This requires computing the axial symmetry descriptor which takes $O(N^3 \log^2 N)$
- ☑ This needs to be done on a per-model basis so this can be done offline.

The online running time of the alignment algorithm remains $O(N^2 \log^2 N)$