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:

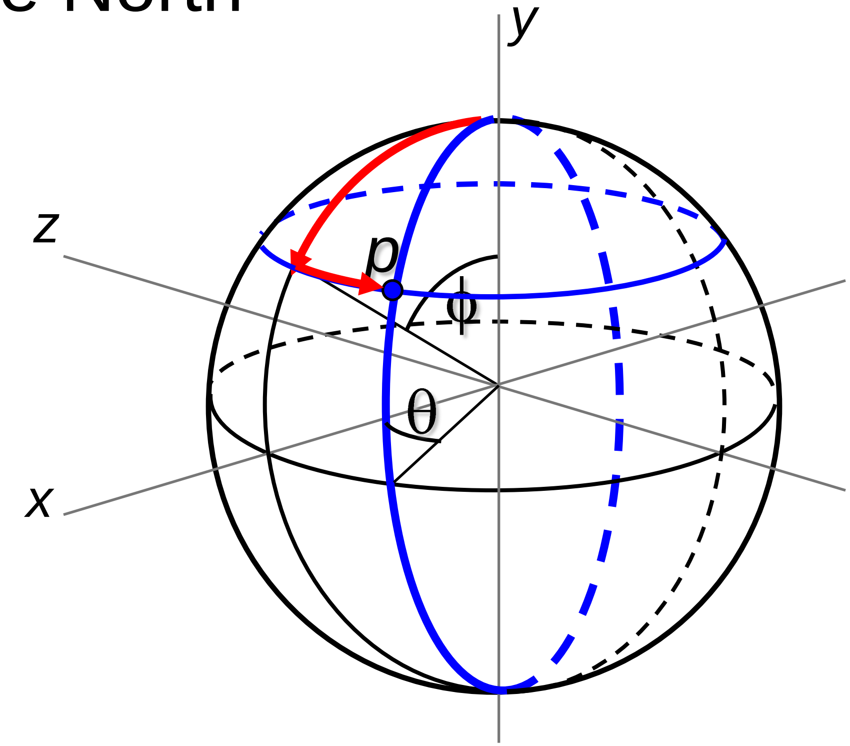
$$R = R_y(\theta) \cdot R_z(\phi) \cdot R_y(\psi)$$

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>:

If we express a rotation in terms of its Euler angles (θ,ϕ,ψ), then the angles (θ,ϕ) correspond to the rotation that takes the North pole to the point $p=\Phi(\theta,\phi)$.



Review

Recall 3:

If we represent a rotation *R* in terms of the Euler angles (θ,ϕ,ψ), then the inverse of *R* can be represented by the Euler angles $(-\psi,-\phi,-\theta)$:

Review

Recall 3:

If we represent a rotation *R* in terms of the Euler angles (θ,ϕ,ψ), then the inverse of *R* can be represented by the Euler angles (-ψ,-ϕ,-θ):

$$R^{-1} = \left(R_y(\theta)\cdot R_z(\phi)\cdot R_y(\psi)\right)^{-1}$$

Review

Recall 3:

If we represent a rotation *R* in terms of the Euler angles (θ,ϕ,ψ), then the inverse of *R* can be represented by the Euler angles (-ψ,-ϕ,-θ):

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

Review

<u>Recall 3</u>:

If we represent a rotation *R* in terms of the Euler angles (θ,φ,ψ), then the inverse of *R* can be represented by the Euler angles (-ψ,-φ,-θ):

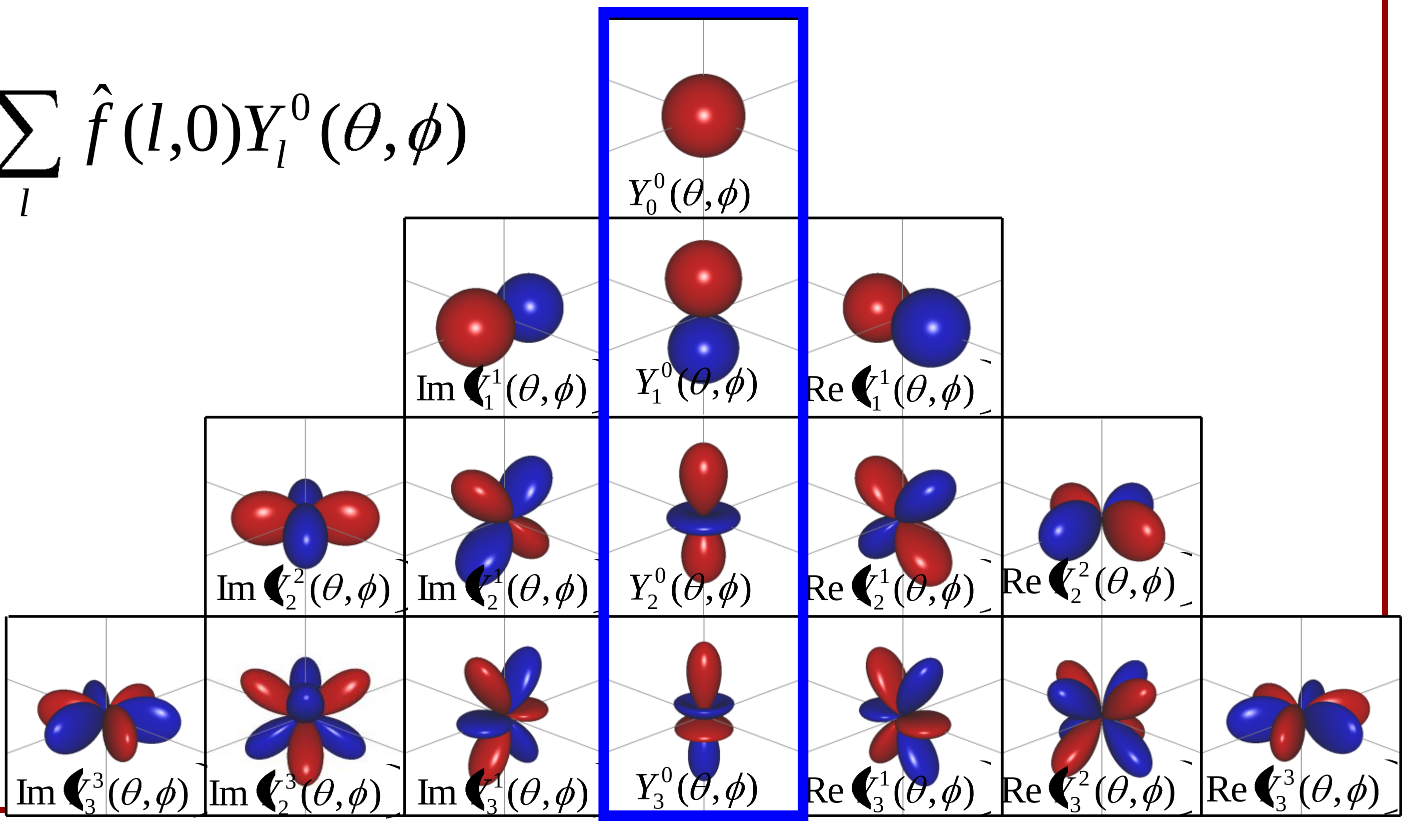
$$\begin{aligned} R^{-1} &= \left(R_y(\theta)\cdot R_z(\phi)\cdot R_y(\psi)\right)^{-1} \\ &= \left(R_y(\psi)\right)^{-1}\cdot\left(R_z(\phi)\right)^{-1}\cdot\left(R_y(\theta)\right)^{-1} \\ &= R_y(-\psi)\cdot R_z(-\phi)\cdot R_y(-\theta) \end{aligned}$$

Review

Recall 4:

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

$$f(\theta,\phi)=\sum_l \hat{f}(l,0)Y_l^0(\theta,\phi)$$



Review

Recall 5:

Rotating the spherical harmonic Y_l^m about the *y*-axis by an angle of α is equivalent to multiplying it by $e^{-im\alpha}$.

Review

Recall 5:

Rotating the spherical harmonic Y_l^m about the *y*-axis by an angle of α is equivalent to multiplying it by $e^{-im\alpha}$:

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

$$Y_l^m(\theta,\phi) = P_l^m(\cos\phi)e^{im\theta}$$

Review

Recall 5:

Rotating the spherical harmonic Y_l^m about the y-axis by an angle of α is equivalent to multiplying it by $e^{-im\alpha}$:

So rotating by α about the y-axis gives:

$$\rho_{R_y(\alpha)} Y_l^m(\theta,\phi) = Y_l^m(\theta-\alpha,\phi)$$

Review

Recall 5:

Rotating the spherical harmonic Y_l^m about the *y*-axis by an angle of α is equivalent to multiplying it by $e^{-im\alpha}$:

So rotating by α about the *y*-axis gives:

$$\rho_{R_y(\alpha)} Y_l^m(\theta,\phi) = Y_l^m(\theta-\alpha,\phi)$$

$$= P_l^m(\cos\phi)e^{im(\theta-\alpha)}$$

Review

Recall 5:

Rotating the spherical harmonic Y_l^m about the *y*-axis by an angle of α is equivalent to multiplying it by $e^{-im\alpha}$:

So rotating by α about the *y*-axis gives:

$$\rho_{R_y(\alpha)} Y_l^m(\theta,\phi) = Y_l^m(\theta-\alpha,\phi)$$

$$= P_l^m(\cos\phi) e^{im(\theta-\alpha)}$$

$$= e^{-im\alpha} P_l^m(\cos\phi) e^{im\theta}$$

Review

<u>Recall 5</u>:

Rotating the spherical harmonic Y_l^m about the *y*-axis by an angle of α is equivalent to multiplying it by $e^{-im\alpha}$:

So rotating by α about the *y*-axis gives:

$$\begin{aligned}\rho_{R_y(\alpha)} Y_l^m(\theta,\phi) &= Y_l^m(\theta-\alpha,\phi)\\ &= P_l^m(\cos\phi)e^{im(\theta-\alpha)}\\ &= e^{-im\alpha}P_l^m(\cos\phi)e^{im\theta}\\ &= e^{-im\alpha}Y_l^m(\theta,\phi)\end{aligned}$$

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:

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:

$$\rho_{R(\theta,\phi,\psi)} f = \rho_{R_y(\theta)R_z(\phi)R_y(\psi)} f$$

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:

$$\rho_{R(\theta,\phi,\psi)} f = \rho_{R_y(\theta) R_z(\phi) R_y(\psi)} f$$
$$= \rho_{R_y(\theta)} \rho_{R_z(\phi)} \rho_{R_y(\psi)} f$$

Review

<u>Recall 6</u>:

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:

$$\rho_{R(\theta,\phi,\psi)} f = \rho_{R_y(\theta) R_z(\phi) R_y(\psi)} f$$

$$= \rho_{R_y(\theta)} \rho_{R_z(\phi)} \rho_{R_y(\psi)} f$$

$$= \rho_{R_y(\theta)} \rho_{R_z(\phi)} f$$

Review

Recall 7:

Given a spherical function *f* of frequency *l*:

$$f=\sum_{m=-l}^{l}\hat{f}(l,m)Y_l^m$$

correlating *f* with a zonal harmonic of frequency *l* is equivalent to multiplying *f* by a scalar:

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

Review

Recall 8:

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

$$\mathrm{Dot}_{f,g}(R) = \langle f, \rho_R g \rangle$$

Review

Recall 8:

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

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

$$\mathrm{Dot}_{f,g}(R) = \langle f, \rho_R g \rangle$$

$$\mathrm{Dot}_{f,g}(\theta,\phi,\psi) = \left\langle \sum_l \sum_{m=-l}^{l} \hat{f}(l,m) Y_l^m, \rho_{R(\theta,\phi,\psi)} \left(\sum_l \hat{g}(l,0) Y_l^0 \right) \right\rangle$$

Review

Recall 8:

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

By the conjugate linearity of the Hermitian inner product, this becomes:

$$\mathrm{Dot}_{f,g}(\theta,\phi,\psi) = \left\langle \sum_l \sum_{m=-l}^{l} \hat{f}(l,m) Y_l^m, \rho_{R(\theta,\phi,\psi)} \left(\sum_l \hat{g}(l,0) Y_l^0 \right) \right\rangle$$

$$\mathrm{Dot}_{f,g}(\theta,\phi,\psi) = \sum_l \sum_{m=-l}^{l} \hat{f}(l,m) \overline{\hat{g}(l,0)} \left\langle Y_l^m, \rho_{R(\theta,\phi,\psi)} Y_l^0 \right\rangle$$

Review

Recall 8:

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

Which can be simplified to:

$$\mathrm{Dot}_{f,g}(\theta,\phi,\psi)=\sum_l\sum_{m=-l}^{l}\hat{f}(l,m)\overline{\hat{g}(l,0)}\left\langle Y_l^m,\rho_{R(\theta,\phi,\psi)}Y_l^0\right\rangle$$

$$\mathrm{Dot}_{f,g}(\theta,\phi,\psi)=\sum_l\sum_{m=-l}^{l}\hat{f}(l,m)\overline{\hat{g}(l,0)}\sqrt{\frac{4\pi}{2l+1}}Y_l^m(\theta,\phi)$$

Review

Recall 8:

$$\text{Dot}_{f,g}(\theta,\phi,\psi)=\sum_{l}\sum_{m=-l}^{l}\hat{f}(l,m)\overline{\hat{g}(l,0)}\sqrt{\frac{4\pi}{2l+1}}Y_l^m(\theta,\phi)$$

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 f by the $(l,0)$-th coefficient of g times sqrt$(4\pi/(2l+1))$
  $O(N^2)$
- Computing the inverse transform
  $O(N^2\log^2 N)$

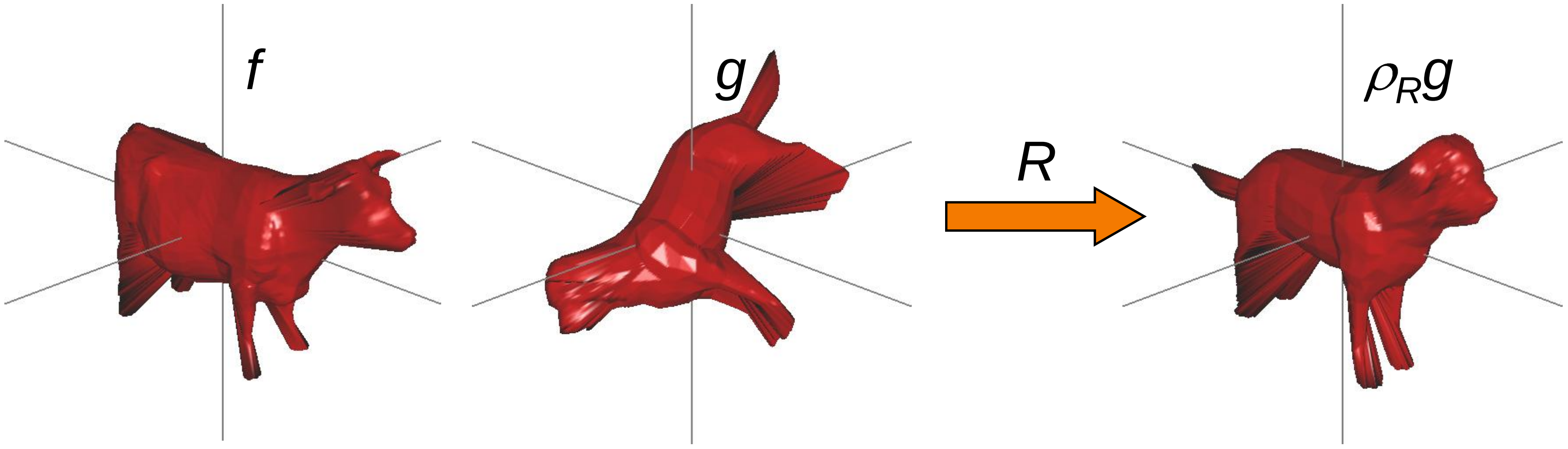
Outline

- Math Review
- Fast Rotational Alignment

Goal

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

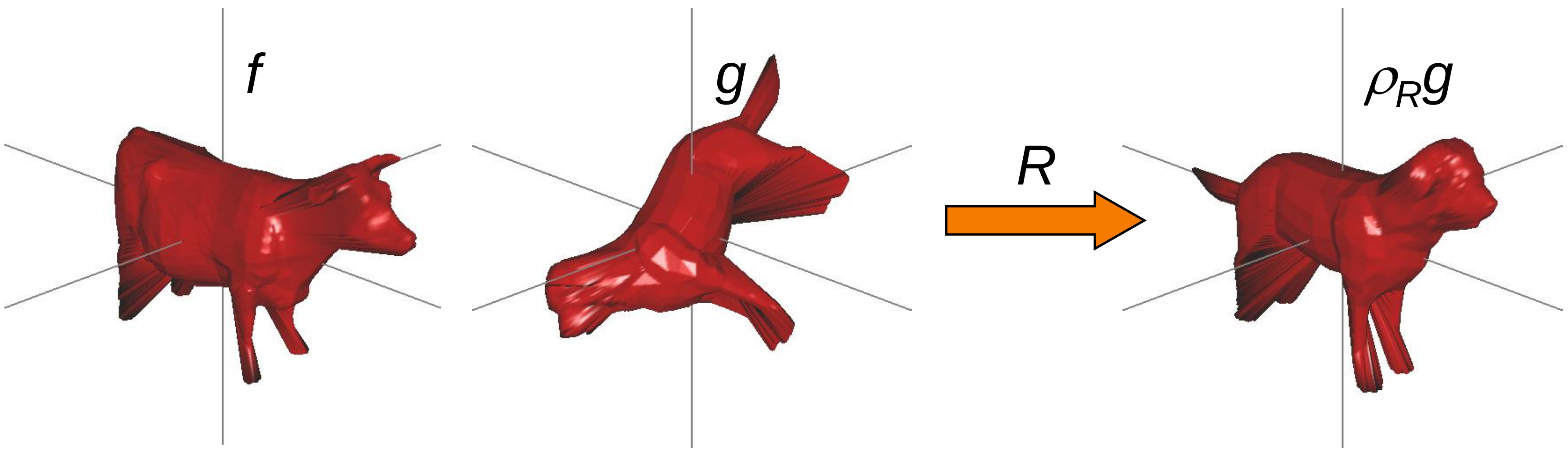
$$R = \underset{R \in \text{Rotations}}{\arg\min} \left\| f - \rho_R g \right\|^2$$



Approach

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

$$R = \underset{R \in \text{Rotations}}{\arg\max} \langle f, \rho_R g \rangle$$



Approach

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

$$\text{Dot}_{f,g}(R) = \langle f, \rho_R g \rangle$$

and finding the rotation *R* that maximizes 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
  - Computing the appropriate $O(N^2)$ dot-product

Approach

<u>Fast Spherical Correlation</u>:

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

- Computing the spherical harmonic coefficients of *f* and *g*:
  $O(N^2\log N)$
- Cross multiplying the spherical harmonic coefficients within each frequency to get the Wigner D-coefficients:
  $O(N^3)$
- Performing 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 markedly faster than the brute force, for many applications, a cubic algorithm may still be too slow.

Efficiency

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

What we would like is 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 a lot of 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.

This turns out to be more information than we actually need since all we want is the single, optimal rotation.

Parameter Optimization

Given a function $F(x,y)$, we would like to find the parameters (x_0,y_0) at which F is maximal:

$$(x_0, y_0) = \arg\max_{x,y} \left(F(x, y) \right)$$

Parameter Optimization

Given a function $F(x,y)$, we would like to find the parameters (x_0,y_0) at which F is maximal:

$$(x_0, y_0) = \arg\max_{x,y} \left\{ F(x, y) \right\}$$

We can find the parameters (x_0,y_0) by searching over the entirety of the parameter domain to find the parameters at which F is maximal.

Parameter Optimization

Given a function $F(x,y)$, we would like to find the parameters (x_0,y_0) at which F is maximal:

$$(x_0, y_0) = \arg\max_{x,y} \left(F(x, y) \right)$$

We can find the parameters (x_0,y_0) by searching over the entirety of the parameter domain to find the parameters at which F is maximal.

The down side is that 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_0,y_0) at which F is maximal:

$$(x_0, y_0) = \arg\max_{x,y} \left(F(x,y) \right)$$

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

- First, find the optimal value for x_0, and then
- Holding x_0 fixed, find the optimal value for y_0.

Parameter Optimization

Parameter Splitting:

Given a function $F(x,y)$, we would like to find the parameters (x_0,y_0) at which F is maximal:

$$(x_0, y_0) = \arg\max_{x,y} \left(F(x, y) \right)$$

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

- First, find the optimal value for x_0, and then
- Holding x_0 fixed, find the optimal value for y_0.

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_0,y_0) maximizes $F(x,y)$ then x_0 maximizes $G(x)$.

Parameter Optimization

Parameter Splitting:

To do this, we need to define a 1D function $G(x)$ with the property that if (x_0,y_0) maximizes $F(x,y)$ then x_0 maximizes $G(x)$.

Then the problem of optimizing $F(x,y)$:

$$(x_0, y_0) = \arg\max_{x,y} \left(F(x, y) \right)$$

Parameter Optimization

Parameter Splitting:

To do this, we need to define a 1D function $G(x)$ with the property that if (x_0,y_0) maximizes $F(x,y)$ then x_0 maximizes $G(x)$.

Then the problem of optimizing $F(x,y)$:

$$(x_0, y_0) = \arg\max_{x,y} \left(F(x,y) \right)$$

turns into the sequence of problems:

$$x_0 = \arg\max_{x} \left(G(x) \right)$$

$$y_0 = \arg\max_{y} \left(F(x_0, y) \right)$$

Parameter Optimization

Application to Rotational Alignment:

To find the optimal alignment, we would like to find the Euler angles (θ_0,ϕ_0,ψ_0) that maximize the correlation:

$$(\theta_0,\phi_0,\psi_0) = \arg\max_{\theta,\phi,\psi} \left\langle f, \rho_{R_y(\theta)R_z(\phi)R_y(\psi)} g \right\rangle$$

Parameter Optimization

Application to Rotational Alignment:

Instead of trying to optimize over all three parameters simultaneously, we can optimize over two of the parameters, and then fixing the two optimal parameters, optimize over the third:

$$(\theta_0,\phi_0) = \arg\max_{\theta,\phi} \left(G(\theta,\phi) \right)$$

$$\psi_0 = \arg\max_{\psi} \left\langle f, \rho_{R_y(\theta_0)R_z(\phi_0)R_y(\psi)} g \right\rangle$$

Parameter Optimization

Application to Rotational Alignment:

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

Parameter Optimization

Application to Rotational Alignment:

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 *g* that is axially symmetric about the *y*-axis:

$$h(\theta,\phi) = \sum_l \hat{g}(l,0) Y_l^0(\theta,\phi)$$

Parameter Optimization

Application to Rotational Alignment:

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 *g* that is axially symmetric about the *y*-axis:

$$h(\theta,\phi)=\sum_l \hat{g}(l,0)Y_l^0(\theta,\phi)$$

we can define:

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

Parameter Optimization

Application to Rotational Alignment:

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

The function *G* has two important properties:

Parameter Optimization

Application to Rotational Alignment:

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

The function *G* has two important properties:

- In the case that *g* is already axially symmetric about the *y*-axis (i.e. *h*=*g*), the optimizing angles (θ_0,ϕ_0) are guaranteed to define the optimal transformation.

Parameter Optimization

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

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

The function *G* has two important properties:

- In the case that *g* is already axially symmetric about the *y*-axis (i.e. *h*=*g*), the optimizing angles (θ_0,ϕ_0) are guaranteed to define the optimal transformation.
- Since the function *h* is axially symmetric about the *y*-axis, we can find the optimizing angles (θ_0,ϕ_0) in $O(N^2\log^2 N)$ time using the fast spherical harmonic transform.

Parameter Optimization

Application to Rotational Alignment:

Having solved for the optimal angles (θ_0,ϕ_0), we can solve for the optimal ψ_0 by solving:

$$\psi_0 = \arg\max_{\psi} \left\langle f, \rho_{R_y(\theta_0)R_z(\phi_0)R_y(\psi)} g \right\rangle$$

Parameter Optimization

Application to Rotational Alignment:

Having solved for the optimal angles (θ_0,ϕ_0), we can solve for the optimal ψ_0 by solving:

$$\psi_0 = \arg\max_{\psi} \left\langle f, \rho_{R_y(\theta_0)R_z(\phi_0)R_y(\psi)} g \right\rangle$$

Since the representation is unitary, this becomes:

$$\psi_0 = \arg\max_{\psi} \left\langle \rho_{R_z(-\phi_0)R_y(-\theta_0)} f, \rho_{R_y(\psi)} g \right\rangle$$

Parameter Optimization

Application to Rotational Alignment:

Having solved for the optimal angles (θ_0,ϕ_0), we can solve for the optimal ψ_0 by solving:

$$\psi_0 = \arg\max_{\psi} \left\langle f, \rho_{R_y(\theta_0)R_z(\phi_0)R_y(\psi)} g \right\rangle$$

Since the representation is unitary, this becomes:

$$\psi_0 = \arg\max_{\psi} \left\langle \rho_{R_z(-\phi_0)R_y(-\theta_0)} f, \rho_{R_y(\psi)} g \right\rangle$$

And in terms of the spherical harmonics coefficients of *g*, we get:

$$\psi_0 = \arg\max_{\psi} \left\langle \rho_{R_z(-\phi_0)R_y(-\theta_0)} f, \sum_{l} \sum_{m=-l}^{l} \hat{g}(l,m) \rho_{R_y(\psi)} Y_l^m \right\rangle$$

Parameter Optimization

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

But now we can use the fact that a rotation of the spherical harmonic Y_l^m about the y-axis by an angle of α corresponds to multiplication by $e^{-im\alpha}$:

$$\psi_0 = \arg\max_{\psi} \left\langle \rho_{R_z(-\phi_0)R_y(-\theta_0)} f, \sum_l \sum_{m=-l}^{l} \hat{g}(l,m)\rho_{R_y(\psi)} Y_l^m \right\rangle$$

$$\Downarrow$$

$$\psi_0 = \arg\max_{\psi} \left\langle \rho_{R_z(-\phi_0)R_y(-\theta_0)} f, \sum_l \sum_{m=-l}^{l} \hat{g}(l,m) e^{-im\psi} Y_l^m \right\rangle$$

Parameter Optimization

Application to Rotational Alignment:

Thus, to find ψ_0, we need to find the maximum of the function:

$$\sum_l \sum_{m=-l}^{l} \left\langle \rho_{R_z(-\phi_0)R_y(-\theta_0)} f, \hat{g}(l,m) Y_l^m \right\rangle e^{im\psi}$$

Parameter Optimization

Application to Rotational Alignment:

Thus, to find ψ_0, we need to find the maximum of the function:

$$\sum_{l} \sum_{m=-l}^{l} \left\langle \rho_{R_z(-\phi_0) R_y(-\theta_0)} f, \hat{g}(l,m) Y_l^m \right\rangle e^{im\psi}$$

But this is just an expression for a function of ψ as a sum of complex exponentials.

Parameter Optimization

Application to Rotational Alignment:

Thus, to find ψ_0, we need to find the maximum of the function:

$$\sum_{l}\sum_{m=-l}^{l}\left\langle \rho_{R_z(-\phi_0)R_y(-\theta_0)} f, \hat{g}(l,m)Y_l^m \right\rangle e^{im\psi}$$

But this is just 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

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

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

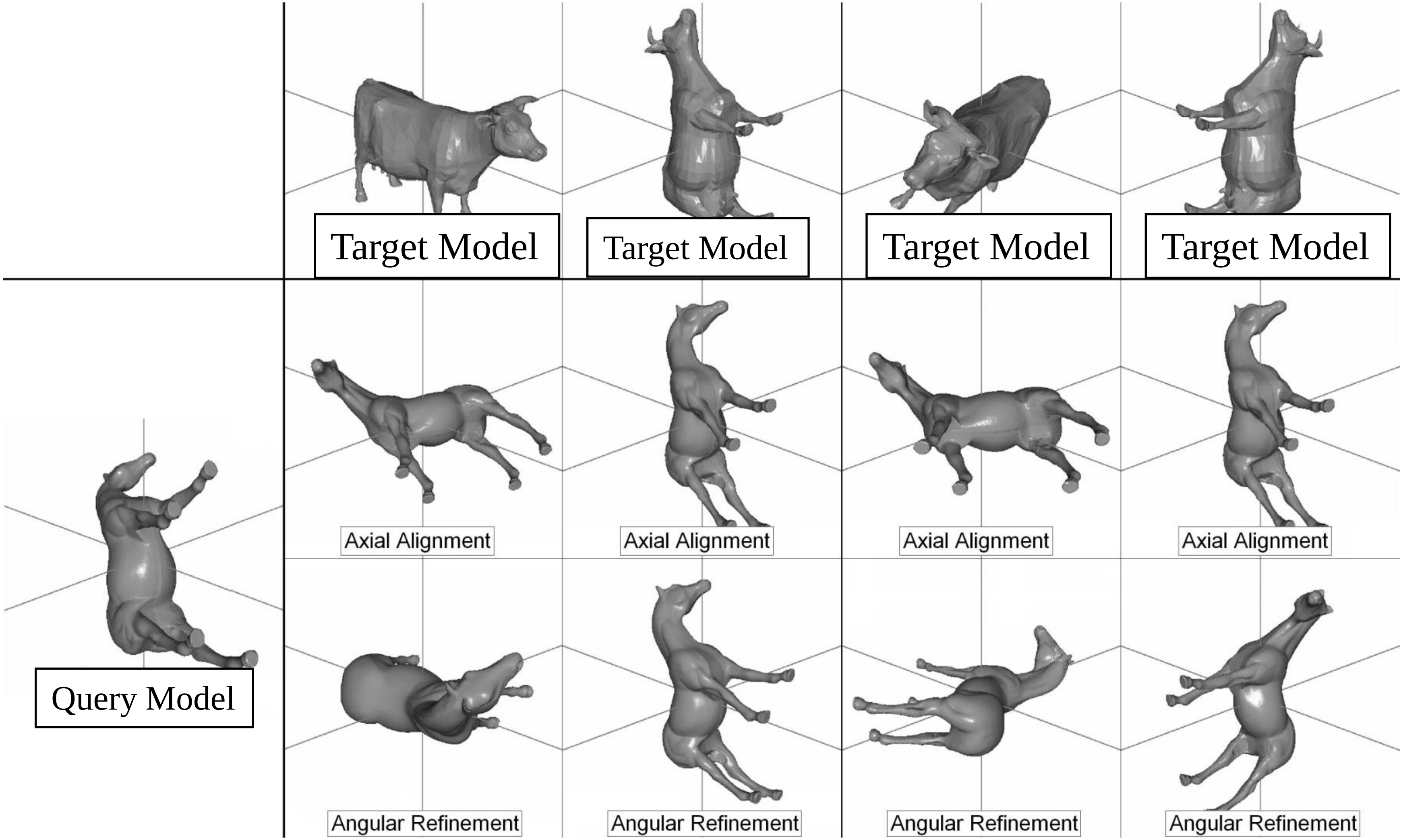
- Correlating *f* with the component of *g* that is axially symmetric about the *y*-axis
  $O(N^2 \log^2 N)$
- Getting the Fourier coefficients of the function in ψ
  $O(N^2)$
- Computing the inverse Fourier transform
  $O(N \log N)$
- Finding the ψ maximizing the function
  $O(N)$

Parameter Optimization

How well does this work in practice?

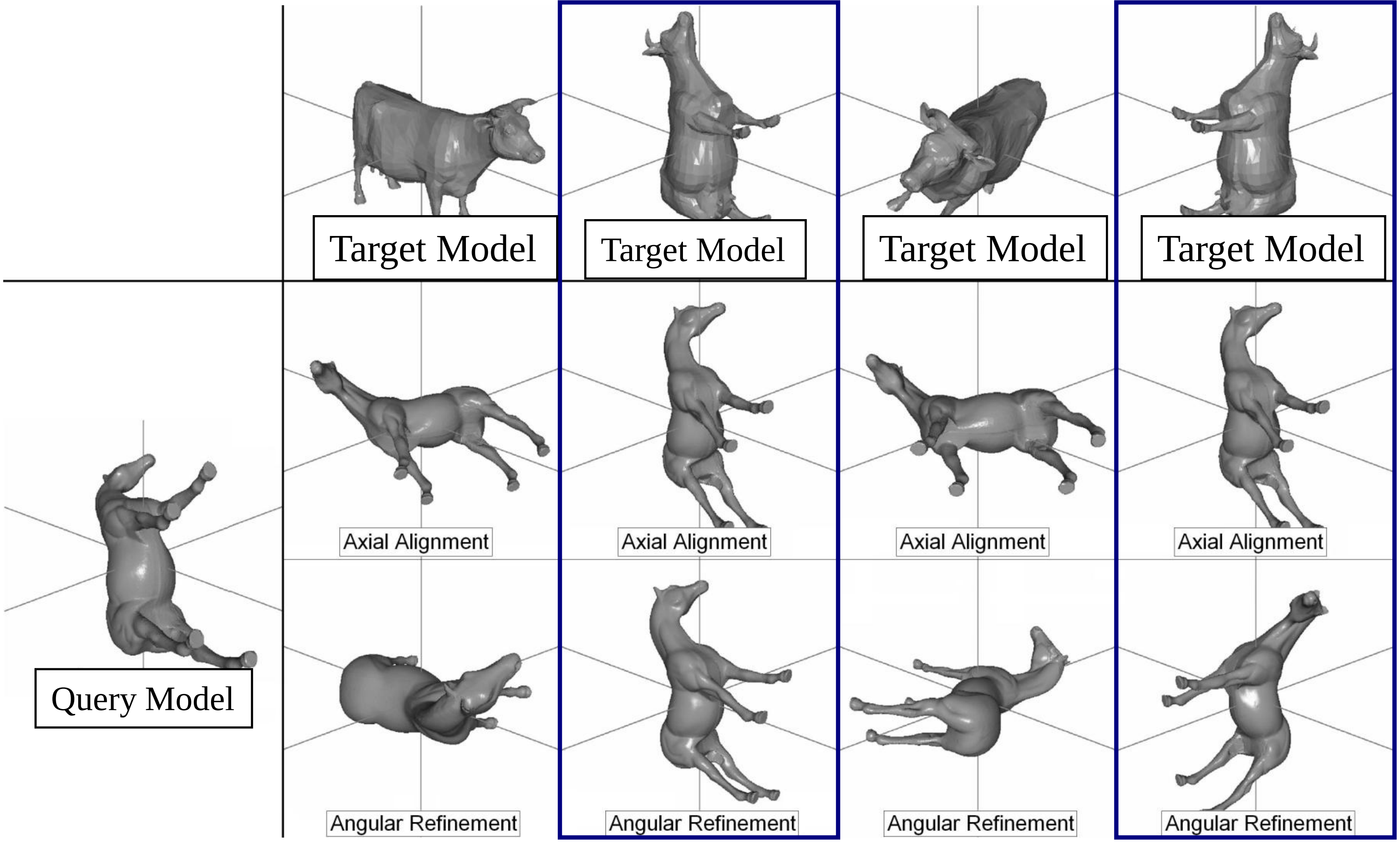
Parameter Optimization

How well does this work in practice?



Parameter Optimization

How well does this work in practice?



Parameter Optimization

How well does this work in practice?

The quality of this method depends on the initial optimization of the function $G(\theta,\phi)$:

- If we get a good guess for the optimal values (θ_0,ϕ_0), the method will perform well.
- Otherwise, the results are less robust.

Parameter Optimization

How well does this work in practice?

The quality of this method depends on the initial optimization of the function $G(\theta,\phi)$.

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

Parameter Optimization

How well does this work in practice?

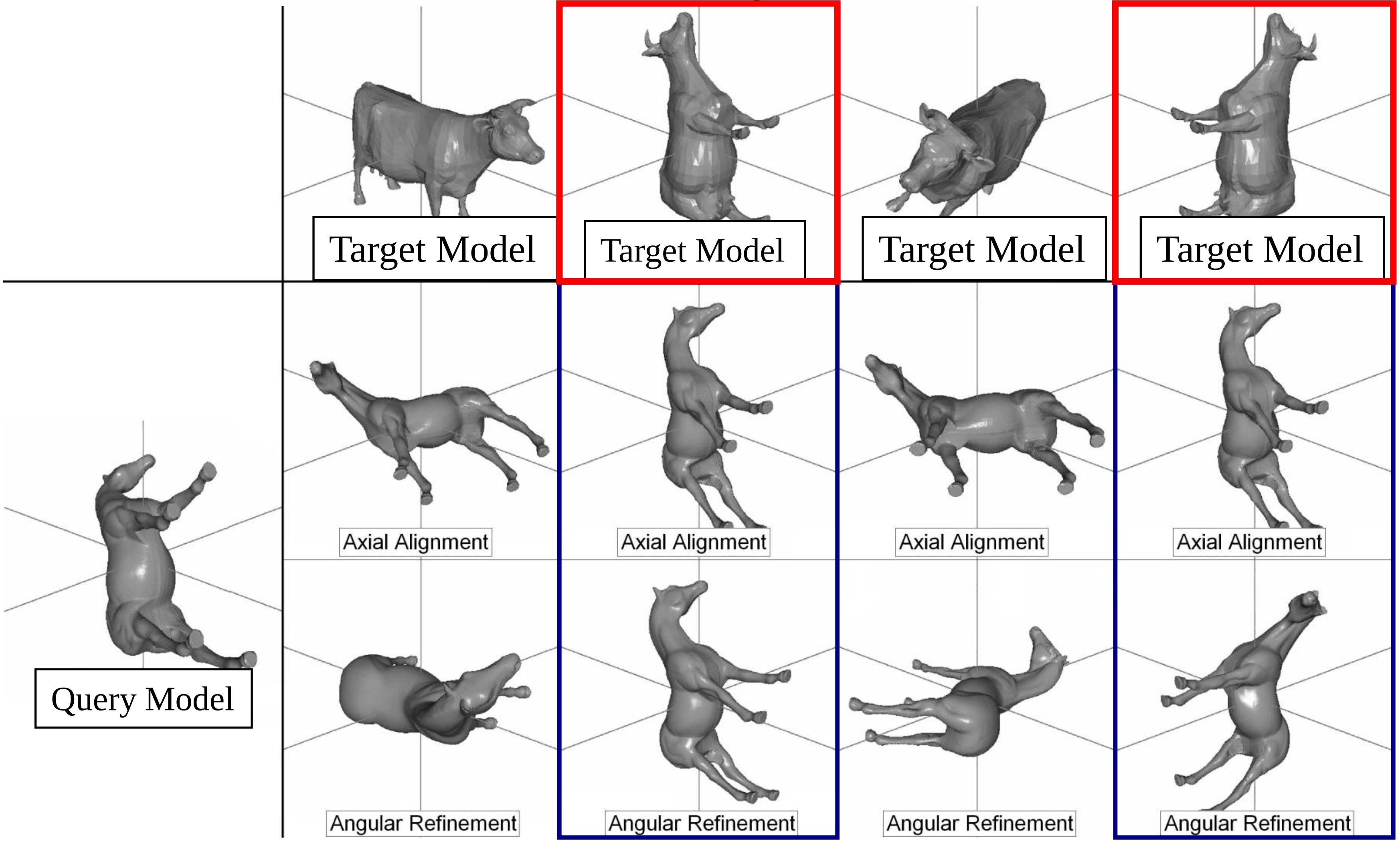
The quality of this method depends on the initial optimization of the function $G(\theta,\phi)$.

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

This means that if the function *g* 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?

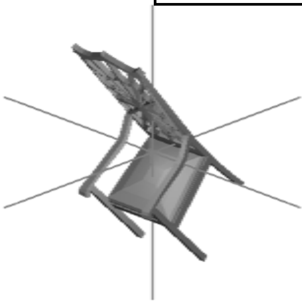
We can leverage this observation by performing a pre-processing step in which we align the function *g* so that the axis with maximal axial symmetry gets mapped to the *y*-axis.

Parameter Optimization

How well does this work in practice?

Pre-Processing

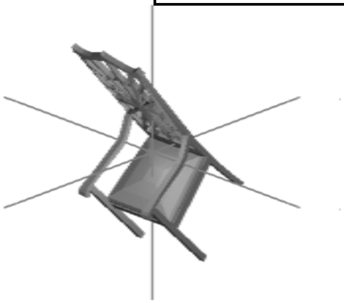
Run-Time

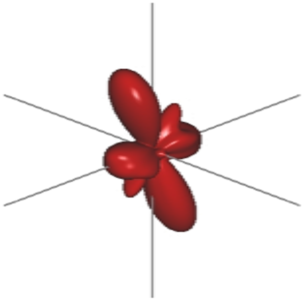


Parameter Optimization

How well does this work in practice?

Pre-Processing



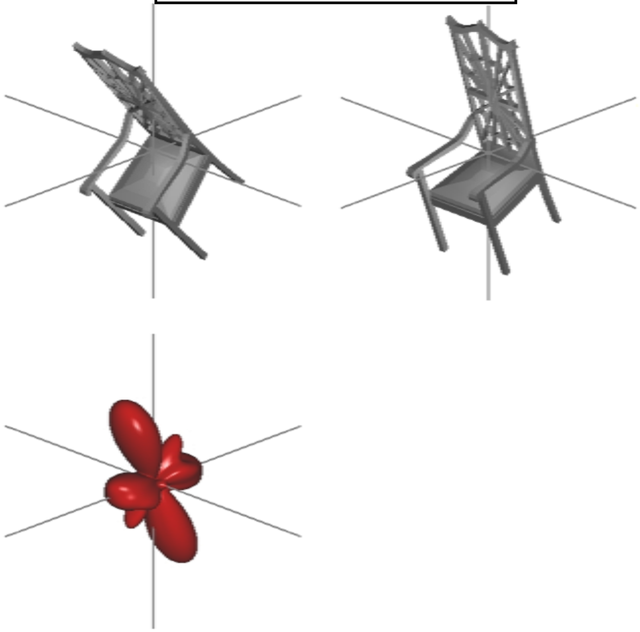


Run-Time

Parameter Optimization

How well does this work in practice?

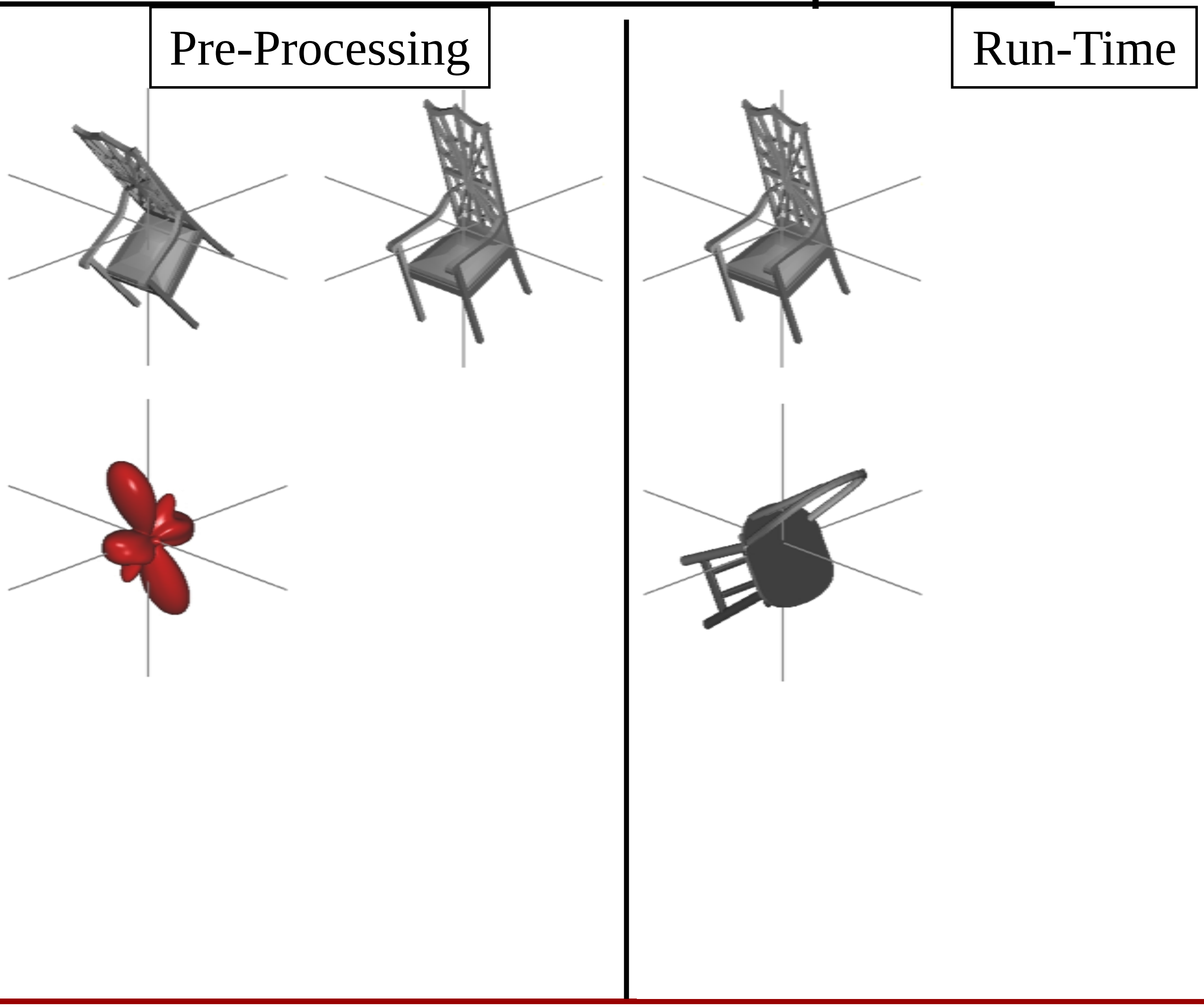
Pre-Processing



Run-Time

Parameter Optimization

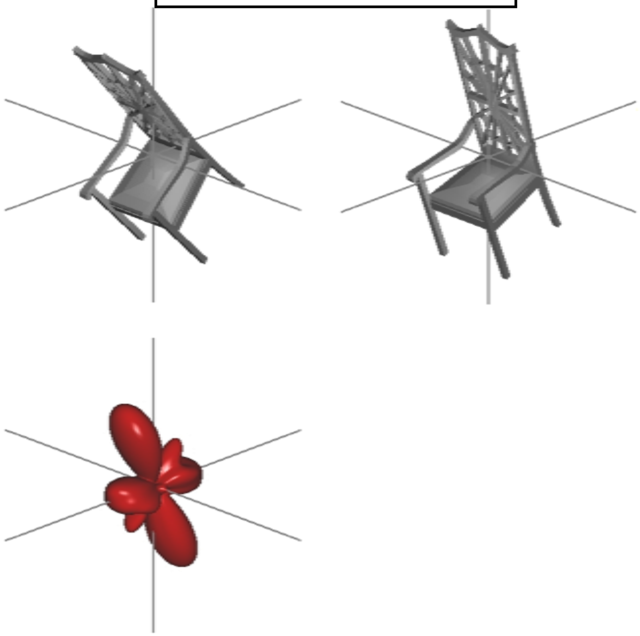
How well does this work in practice?



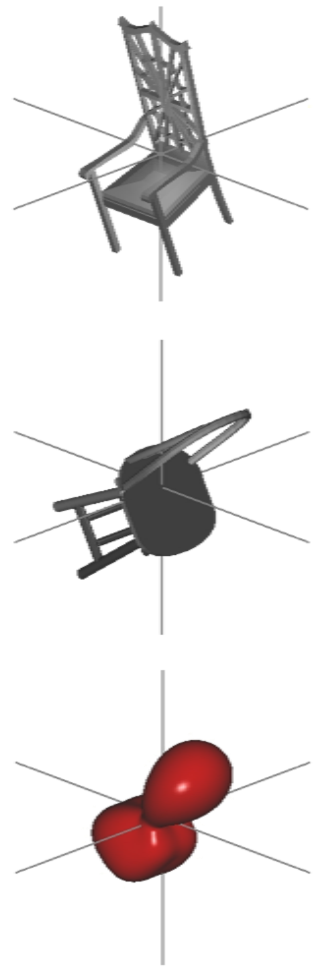
Parameter Optimization

How well does this work in practice?

Pre-Processing

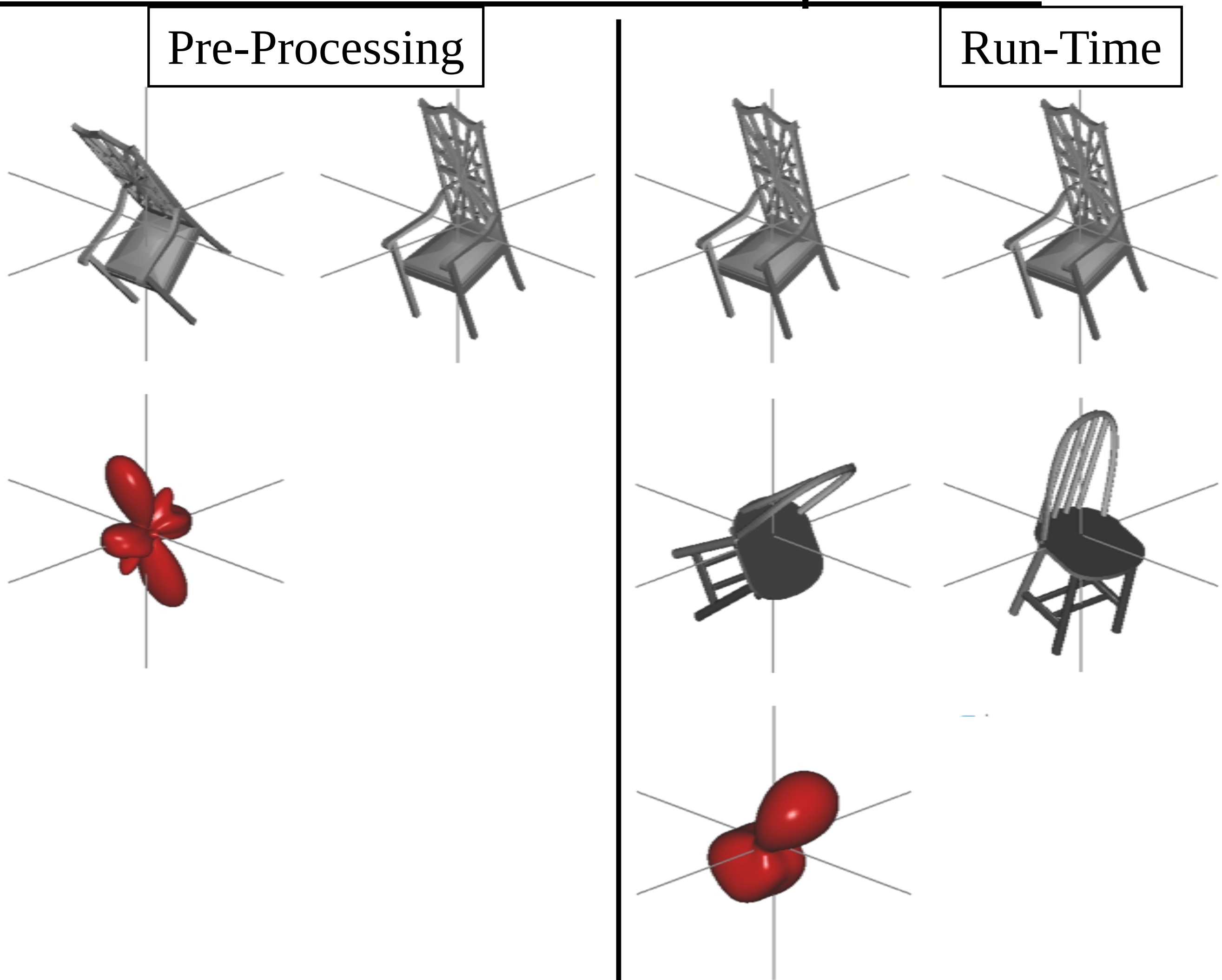


Run-Time



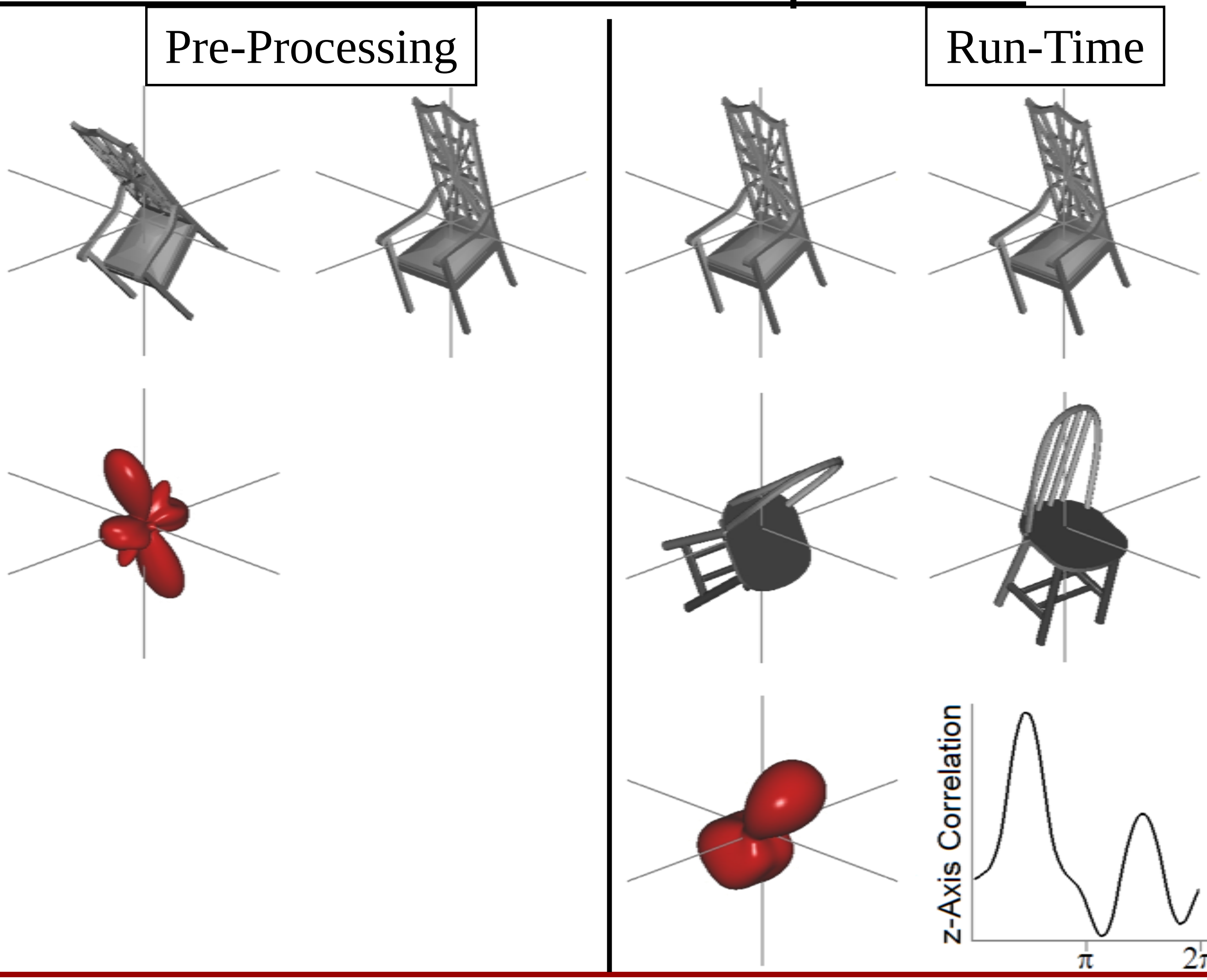
Parameter Optimization

How well does this work in practice?



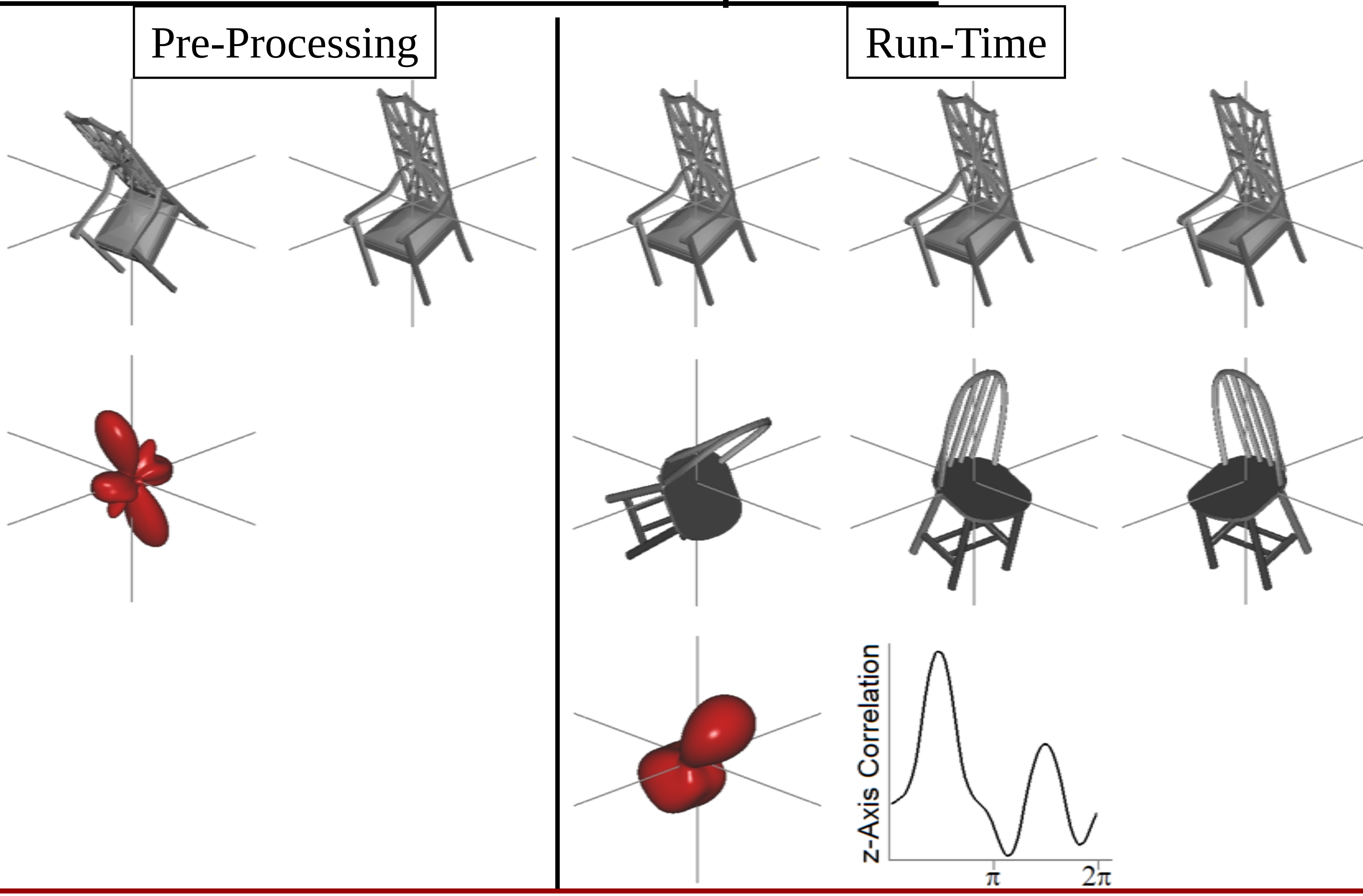
Parameter Optimization

How well does this work in practice?



Parameter Optimization

How well does this work in practice?



Parameter Optimization

Performance:

In order to perform the alignment we need to pre-align the models so that there major axis of axial symmetry is aligned with the *y*-axis.

Parameter Optimization

Performance:

In order to perform the alignment we need to pre-align the models so that there major axis of axial symmetry is aligned with the *y*-axis.

- This requires computing the axial symmetry descriptor which takes $O(N^3 \log^2 N)$

Parameter Optimization

Performance:

In order to perform the alignment we need to pre-align the models so that there 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.

Parameter Optimization

<u>Performance</u>:

In order to perform the alignment we need to pre-align the models so that there 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 $O(N^2 \log^2 N)$.