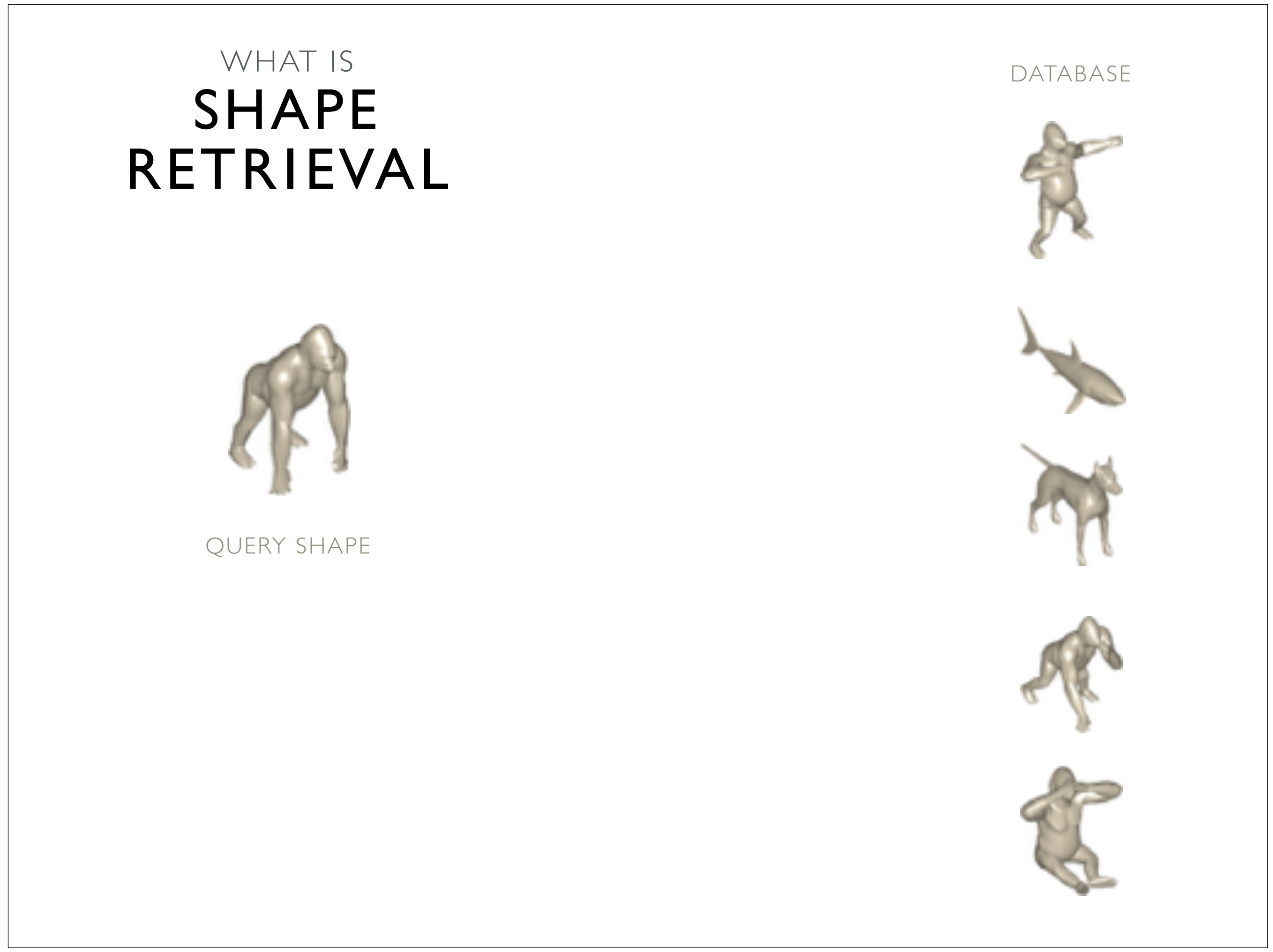
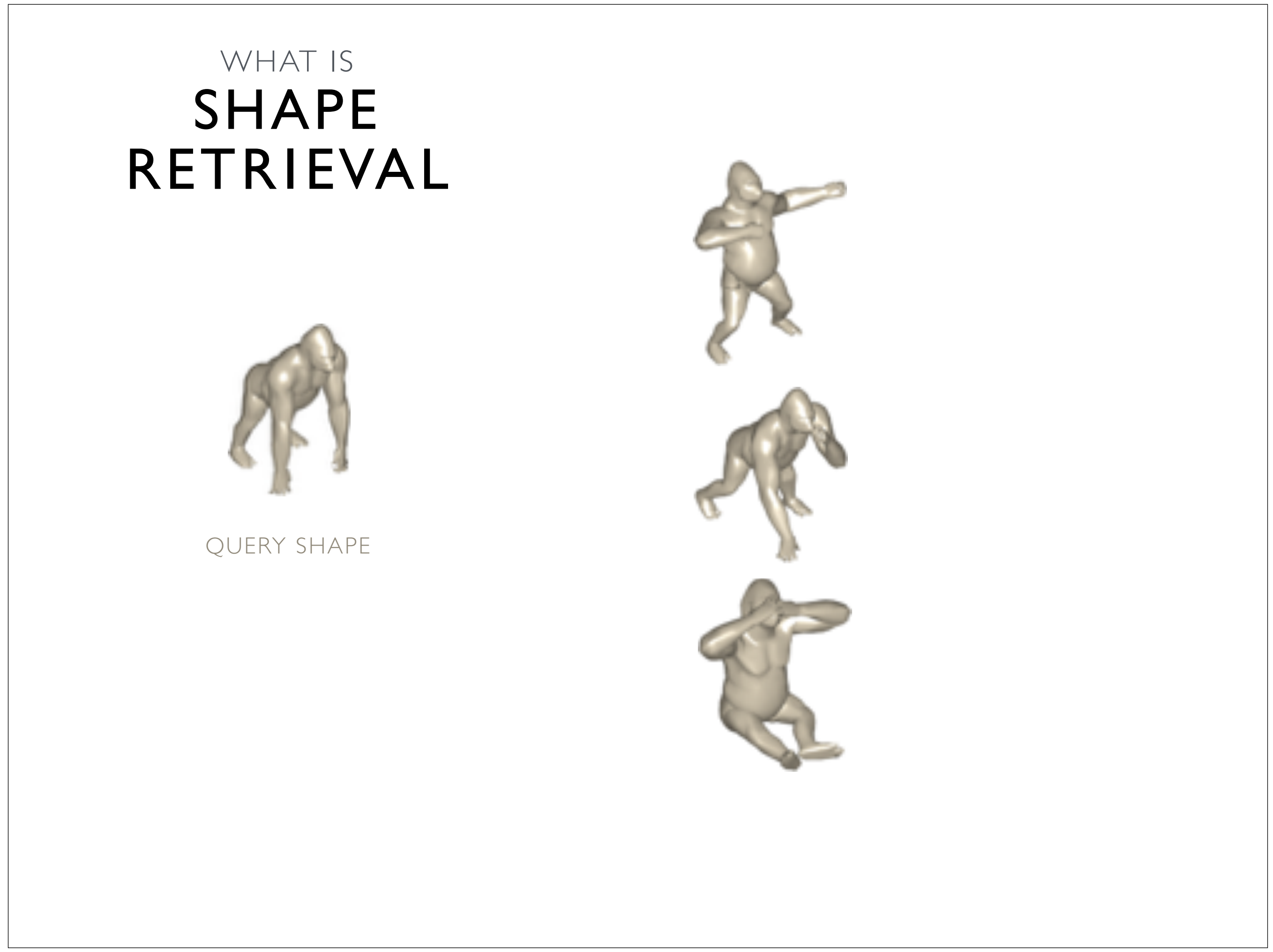
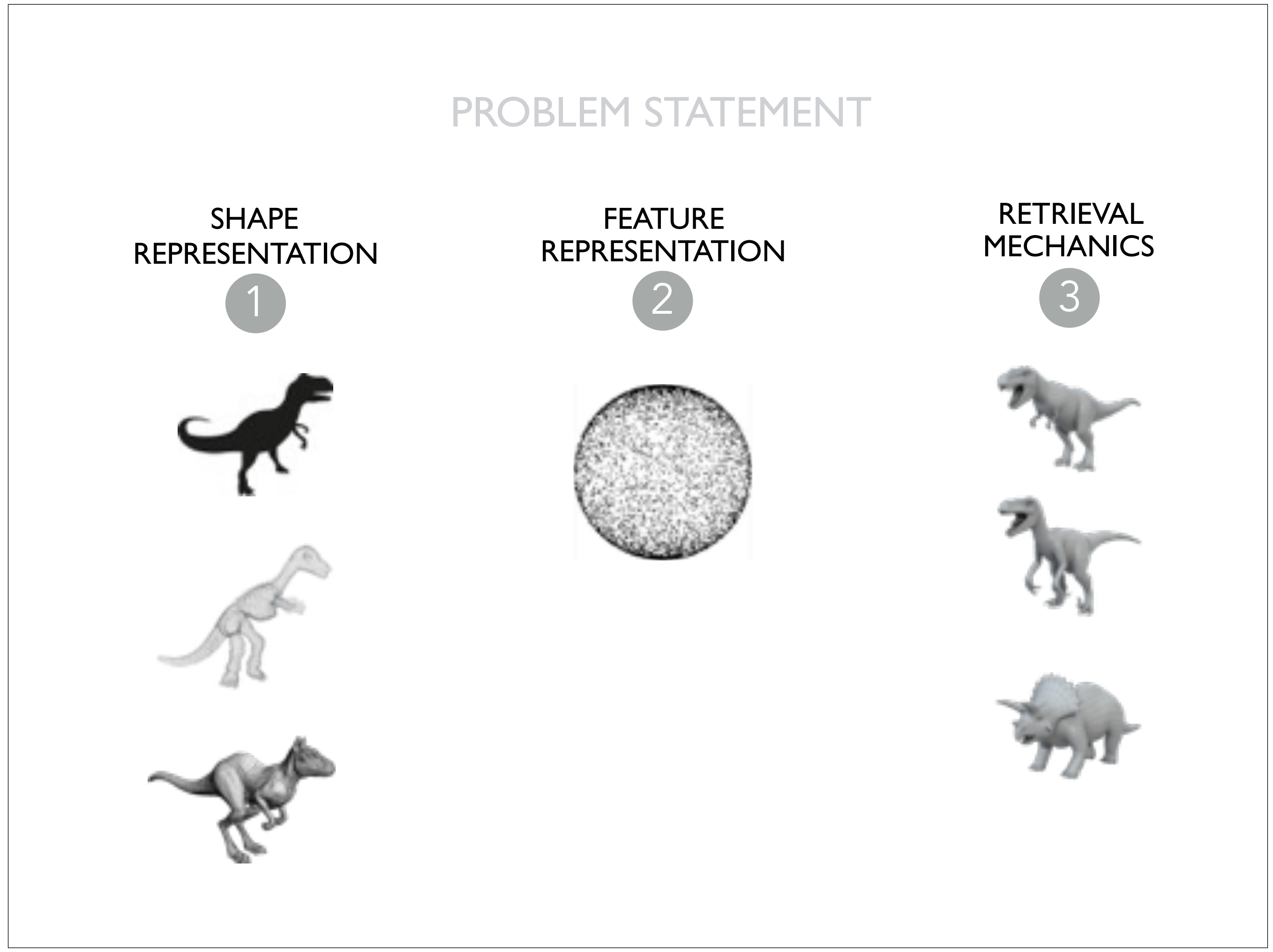
# Density Feature Representation



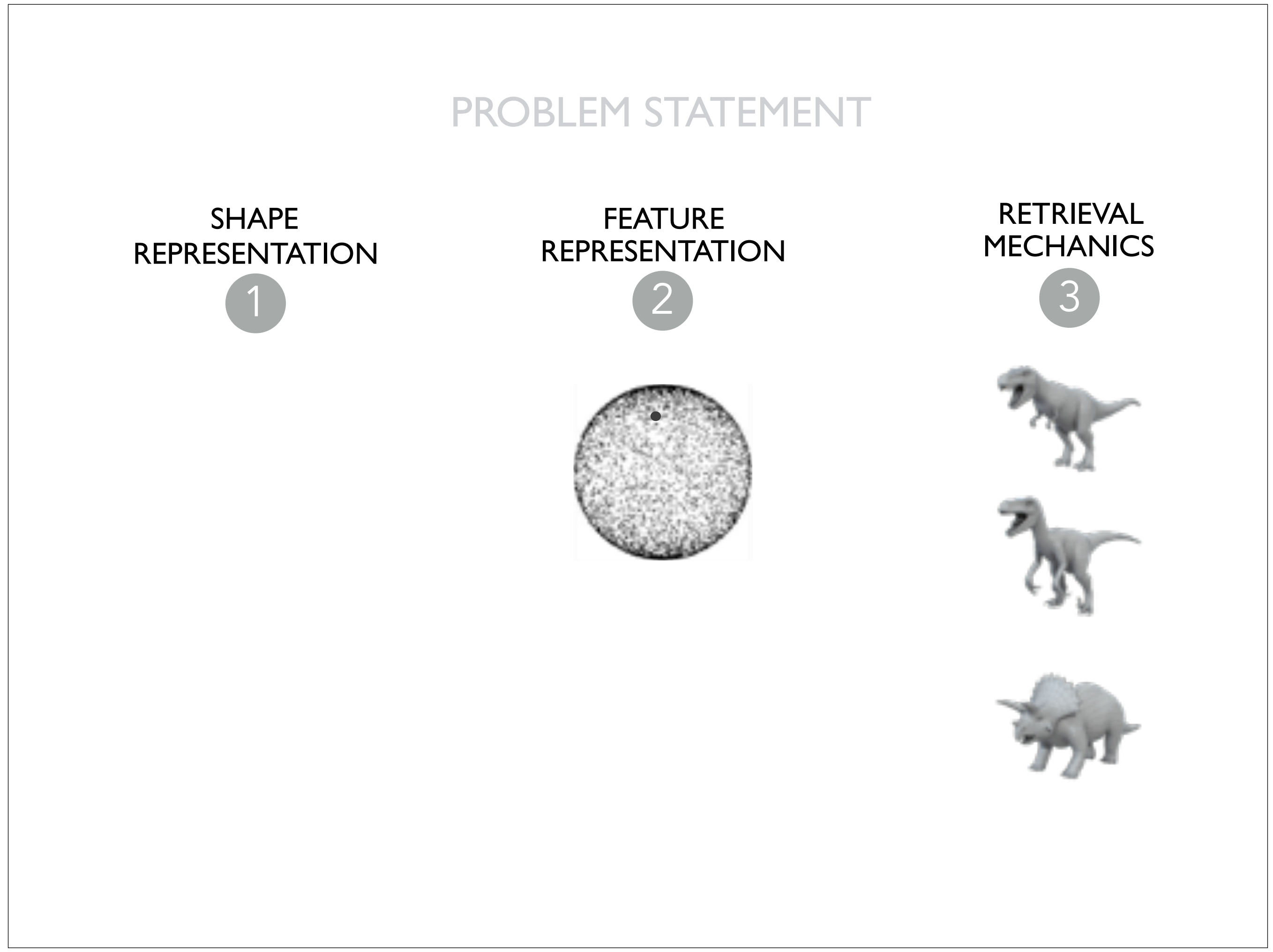

We have the same problem we mentioned in the last all hands meeting. When given a query shape, retrieve the most relevant objects based on the shapes geometry.




We have the same problem we mentioned in the last all hands meeting. When given a query shape, retrieve the most relevant objects based on the shapes geometry.




This is the general process of shape retrieval. We are given a shape representation, extract the feature representation for it, and retrieve related objects when given a query. Previously we focused on retrieval mechanics with hierarchical clustering. Now we will delve deeper into how we are representing objects as feature representations.




This is the general process of shape retrieval. We are given a shape representation, extract the feature representation for it, and retrieve related objects when given a query. Previously we focused on retrieval mechanics with hierarchical clustering. Now we will delve deeper into how we are representing objects as feature representations.

## LAPLACE-BELTRAMI OPERATOR

ISOMETRY INVARIANCE

Yixin

First, we apply the Laplace-Beltrami  Operator to shapes. This creates a shape signature: something that captures the intrinsic properties of a shape.

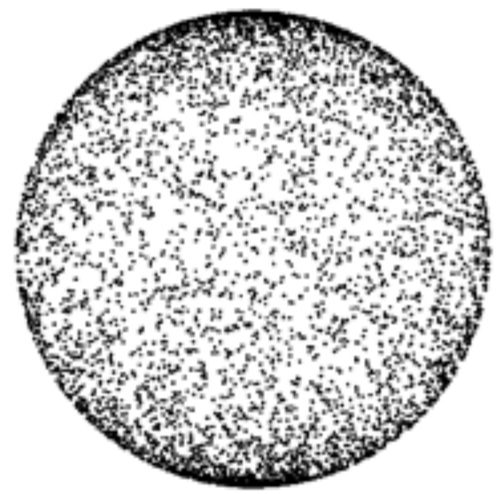
What is the LBO?
This is a generalization of the second derivative on manifolds.
Defined as the divergence of the gradient.

Why is this useful?
Isometry invariance: isometries map to the same point

What is an isometry? A distance-preserving map between spaces
Here’s an example of an isometry: even though they’re distorted, the points between points on the surface stay constant.

## LAPLACE-BELTRAMI OPERATOR

ISOMETRY INVARIANCE



Yixin

First, we apply the Laplace-Beltrami  Operator to shapes. This creates a shape signature: something that captures the intrinsic properties of a shape.

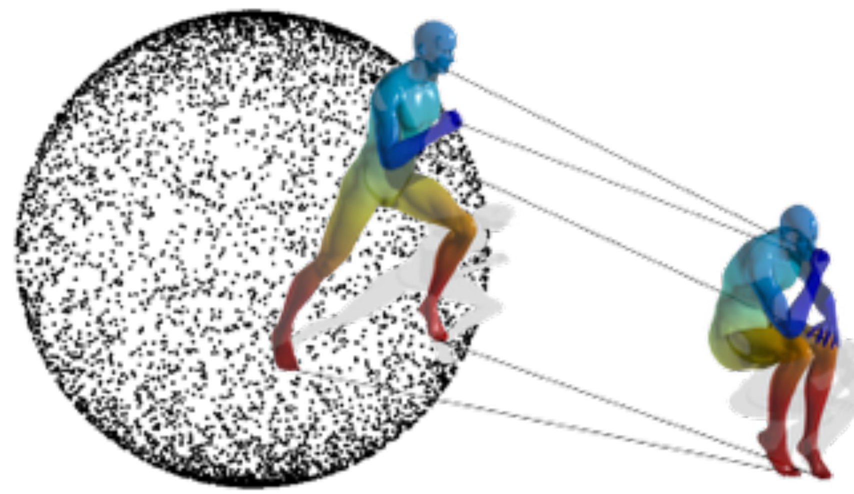
What is the LBO?
This is a generalization of the second derivative on manifolds.
Defined as the divergence of the gradient.

Why is this useful?
Isometry invariance: isometries map to the same point

What is an isometry? A distance-preserving map between spaces
Here’s an example of an isometry: even though they’re distorted, the points between points on the surface stay constant.

# LAPLACE-BELTRAMI OPERATOR

ISOMETRY INVARIANCE



Yixin

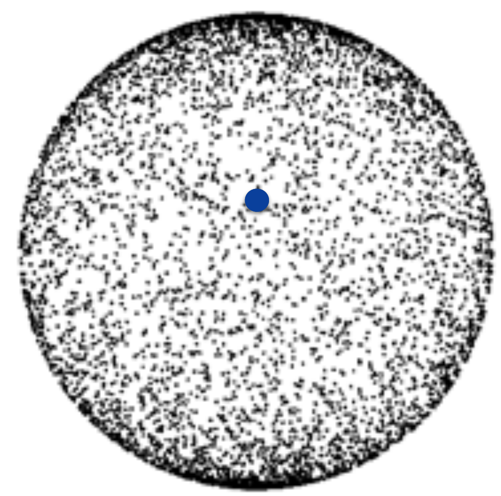
First, we apply the Laplace-Beltrami  Operator to shapes. This creates a shape signature: something that captures the intrinsic properties of a shape.

What is the LBO?
This is a generalization of the second derivative on manifolds.
Defined as the divergence of the gradient.

Why is this useful?
Isometry invariance: isometries map to the same point

What is an isometry? A distance-preserving map between spaces
Here’s an example of an isometry: even though they’re distorted, the points between points on the surface stay constant.

# LAPLACE-BELTRAMI OPERATOR



Yixin

First, we apply the Laplace-Beltrami  Operator to shapes. This creates a shape signature: something that captures the intrinsic properties of a shape.

What is the LBO?
This is a generalization of the second derivative on manifolds.
Defined as the divergence of the gradient.

Why is this useful?
Isometry invariance: isometries map to the same point

What is an isometry? A distance-preserving map between spaces
Here’s an example of an isometry: even though they’re distorted, the points between points on the surface stay constant.

# WAVELETS

- A wavelet is a wave-like oscillation with an amplitude that begins at zero, increases, and then decreases back to zero.
- Wavelets are commonly used in signal processing due to specific mathematical properties
- We use them for density estimation to construct probability densities that represent each shape
- Why do we use them for density estimation?

VISUAL: we have a graph of a true function with points under it signifying densities in 1D. We don’t know the equation of true function but we do know the points that make up that function. Our goal is to fit this unknown function using wavelets. Show another line trying to match the original line.

# Wavelet Math

$$p(x) = \sum_{j_0,k} \alpha_{j_0,k} \phi_{j_0,k}(x) + \sum_{j \geq j_0,k}^{\infty} \beta_{j,k} \psi_{j,k}(x)$$

- How do we estimate the density function?
- We are all familiar with bases from linear algebra. The vectors [1 0] and [0 1] are the basis of R2.
    - We can reach any point on a plane through a linear combination of the basis vectors.
        - The coefficients of this linear combination uniquely determine a point in R2
- Wavelets work in the same way, except over function spaces.
    - The father (phi) and mother (psi) wavelet are basis functions for a function space called L2
    - We can estimate any function in L2 by a linear combination of these basis functions
        - The coefficients of this linear combination uniquely determine a function (and therefore pdf)
        - So all we have to do is find the coefficients
    - You need multiple copies of the wavelet functions summed together in order to form a basis
        - The same functions phi and psi, translated and dilated
        - j and k are indexes into the dilation and translation (how much and how far)
        - Different dilations mean different resolution levels

# Wavelet Math

- What mathematical advantages do wavelets give us?
  - Wavelets have a compact support
    - Just a fancy way of saying they're nonzero on a small domain
    - Here's an example of a wavelet, called Mexican Hat
    - Here's an example of another wavelet, called Daubechies
      - Haar is just Daubechies 1
  - Most are differentiable
  - Orthonormal basis for the function space L2
    - This is the space of all functions whose squares integrate to a finite number
    - For example, because probability density functions all integrate to 1, they are a subset of L2
    - This proves useful for density estimation

- TODO: add animation show they are orthonormal.

# Wavelet Math

- Let's take a brief refresher of what density estimation is
    - The goal of density estimation is to find the best parameters for a probability distribution function that matches with our data
    - For example, (TODO: add example for density estimation)
    - Because we use wavelets which are a basis for L2, we can match approximate any pdf
- Cool visualization: histogram blocks

# Wavelet Math

$$\sqrt{p(x)} = \sum_{j_0,k} \alpha_{j_0,k}\phi_{j_0,k}(x) + \sum_{j \geq j_0,k}^{\infty} \beta_{j,k}\psi_{j,k}(x)$$

$$\sum_{j_0,k} \alpha_{j_0,k}^2 + \sum_{j \geq j_0,k}^{j_1} \beta_{j,k}^2 = 1.$$

Important equations

- Instead of measuring the pdf itself, we measure the square root of the density and square it to get the actual pdf
    - This guarantees non-negativity
    - Ambiguity with signs: just as both (-2)^2 and 2^2 equal 4
    - It also conveniently makes the coefficients sum square to 1 so it's on a hypersphere

# Wavelet Math

$$
\begin{aligned}
\phi_{j_0,\mathbf{k}}(\mathbf{x}) &= 2^{j_0}\phi\left(2^{j_0}x_1 - k_1\right)\phi\left(2^{j_0}x_2 - k_2\right) \\
\psi^1_{j,\mathbf{k}}(\mathbf{x}) &= 2^{j}\phi\left(2^{j}x_1 - k_1\right)\psi\left(2^{j}x_2 - k_2\right) \\
\psi^2_{j,\mathbf{k}}(\mathbf{x}) &= 2^{j}\psi\left(2^{j}x_1 - k_1\right)\phi\left(2^{j}x_2 - k_2\right) \\
\psi^3_{j,\mathbf{k}}(\mathbf{x}) &= 2^{j}\psi\left(2^{j}x_1 - k_1\right)\psi\left(2^{j}x_2 - k_2\right).
\end{aligned}
$$

Important equations

- That was the single resolution version
- To get a wavelet basis for 2D, we need to take the tensor product of our bases
- Given we have our bases, how can we find the coefficients?

# Wavelet Math

$$
\begin{aligned}
\alpha_{j_0,k} &= \int \sqrt{p(x)}\phi_{j_0,k}(x)dx \\
&= \int \frac{p(x)}{\sqrt{p(x)}}\phi_{j_0,k}(x)dx \\
&= \mathcal{E}\left[\frac{\phi_{j_0,k}(x)}{\sqrt{p(x)}}\right].
\end{aligned}
$$

$$(1/N)\sum_{i=1}^{N} \phi_{j_0,k}(x)/\sqrt{p(x)},$$

- How do we find coefficients in R2?
  - You just take the inner product between the vectors and the basis functions
- Same in L2
  - You just take the inner product between the points and the basis functions
  - Inner product in function spaces is the integral
  - But that integral is just the expected value
  - Because we take the square root, we divide by sqrt(p(x))

# Wavelet Math

$$
\begin{aligned}
-\log p(X;\{\alpha_{j_0,k},\beta_{j,k}\}) &= -\frac{1}{N}\log\prod_{i=1}^{N}\left[\sqrt{p(x_i)}\right]^2 \\
&= -\frac{1}{N}\sum_{i=1}^{N}\log\left[\sum_{j_0,k}\alpha_{j_0,k}\phi_{j_0,k}(x_i)+\sum_{j\geq j_0,k}^{j_1}\beta_{j,k}\psi_{j,k}(x_i)\right]^2
\end{aligned}
$$

Important equations

- One common loss function is the negative log likelihood function
    - A measure of how wrong we are with our density estimation
    - Literally just negative log of the likelihood (multiply all the probabilities together to get the likelihood)
- We want to minimize this
    - We use common optimization technique of gradient descent
        - We take the gradient of NLL and go the opposite direction

# Wavelet Math

Coefficients

- These coefficient represent a point in function space
  - They determine the probability density that we estimated
- We use these coefficients as our feature representation
- Conveniently on the unit hypersphere
  - So we get nice geometric properties for shape retrieval

# Optimization

Coefficient code
We have code that estimates the wavelet density.
So far we have been working closely with 2D shapes and we use this to find the scaling coefficients (single resolution).
We would process very large databases like MPEG7 that has 1400 shapes or Brown that has 99 shapes.
The problem is processing these databases are extremely slow.

# Optimization

For the database MPEG7 of 1400 shapes with domain [-1,1], resolution 3, and 24 translates
The runtime is roughly 4.3 hours.
And this is a relatively low resolution.
If we raise the resolution (j) we should expect a much slower runtime.
Our goal is to optimize this code for speed.

TODO:
Get different runtime results with different resolutions or domains and show different resolution images.

# Optimization

$$(1/N)\sum_{i=1}^{N}\phi_{j_0,k}(x)/\sqrt{p(x)},$$

- So there are two main functions that needed optimizing.
- The functions do the same operations, but gathers the data for processing in a more efficient way
- First, we optimized initializeCoefficients
  - This estimates the coefficients as a starting point
  - Two reasons it's slow
    - Calculates the tensor product of all scaling bases for a single point (TODO: show visual of how kron works)
      - But many of those bases are irrelevant because the current point falls outside of its compact domain
    - Second, it loops over each point instead of batching them
  - If we have 4007 points and 576 translations
    - Each data point requires 331,776 operations
    - 1.3e9 operations total
    - 6 seconds

VISUAL, have a grid with a shape made up of points, and take one point and show that it is being compared to all translations to find which ones it falls under.

# Optimization

Solution

- Instead of looping through each point and finding which wavelet functions it falls under,
- We looped through wavelet functions and found which points contribute to it
- We then evaluate the wavelet at all of those points simultaneously
    - 2^j*father(x)*father(y)/sqrt(p)
- Because we store this, we don't have to recompute it for negative log likelihood

VISUAL would be to have that grid with the points that make up a shape and show a single translation comparing to each point.

# Optimization

Show result timestamp of optimization of initializeCoefficients function.

# Optimization

$$
\begin{aligned}
-\log p(X;\{\alpha_{j_0,k},\beta_{j,k}\}) &= -\frac{1}{N}\log\prod_{i=1}^{N}\left[\sqrt{p(x_i)}\right]^2 \\
&= -\frac{1}{N}\sum_{i=1}^{N}\log\left[\sum_{j_0,k}\alpha_{j_0,k}\phi_{j_0,k}(x_i)+\sum_{j\geq j_0,k}^{j_1}\beta_{j,k}\psi_{j,k}(x_i)\right]^2
\end{aligned}
$$

So there are two main functions that needed optimizing
negativeLogLikelihood (show the timestamp of original function).
The problem is the same as initialize coefficients. But performed slightly different operations. Our solution to initializeCoefficients allows us to pass in the appropriate variables needed and simply perform operations. We complete got rid of loops. Made # lines of code into #.

VISUAL: just resulting optimized time stamp should be fine since it only does operations. But maybe we could show how it used to loop over samples (same visual) but then just show it doesn't need to loop, it just does math.

# Optimization

Time Results

Show shitty time stamp again and then put next to optimized time stamp. But into perspective. Like make the time it takes related to how it could take up your entire life but now we made it better. Show how it performs with various parameters.

# Shape L'Ane Rouge

- High level: warp densities to match each other
- Intuition: it's easier to warp similar shapes close to each other, harder to warp different shapes
- Formally: reduce non-rigid deformations between shapes
    - If we have two shapes that are the same except for a distortion, we want them to be the same in our feature representation
- L'Ane Rouge is the sliding block game in French, great analogy because each block slides around
- Use a combinatorial approach called linear assignment

# Linear Assignment

High Level example intro: assigning workers to specific tasks

- The linear assignment problem originally arose in transportation theory
- Question: Suppose you have n mines and n factories which require the ore from the mines
    - Match a mine to a factory in the least cost way
- Generally, the idea is to assign n workers n tasks while minimizing cost

Visual: Match grad mentors with grad students

# Formal assignment problem

Formally:

- Let X and Y be two sets of equal cardinality
- Let C be a cost function that returns a real number for every pair x,y (x in X, y in Y)
- Find the bijection X->Y that minimizes the cost
- Linear if cost function is the sum of individual costs

# Linear Assignment

How do we solve this?
Formulation as a linear program
Use the structure of the program to get faster than straight simplex
We use a speedy algorithm called Jonker-Volgenant
Input is a cost matrix that tells you the cost between every pair of points

# Shape L'Ane Rouge

Slide block thing

- We have these coefficient vectors we measured on the grid
- Each element represents the amplitude of a cell in the grid
- How can we fit the two shapes together as much as possible?

# Shape L’Ane Rouge

Implementation

- Shape L’Ane Rouge happens when you’re trying to find the distance between two shapes
- Let’s try to fit two shapes together first using linear assignment, and then take the hypersphere distance
    - To do linear assignment, we need to find the cost matrix between each pair of points
    - First, we reconstruct the distance matrix between the each pair of grid points
        - Optimization: only do this once for the whole dataset!
    - Second, we find which pairs of points in the two shapes match each other the best
        - Outer product between the coefficient vectors we’re matching up
    - We weight the difference distance matrix with a constant called lambda (we pick this)
    - Then we subtract the outer product from the weighted distance matrix and try to minimize this cost matrix

# Shape L'Ane Rouge

Implementation details:

- Jonker-Voltenant is an iterative algorithm
- Coefficients usually very close to 0 because sum squared = 1 in hundreds/thousands of dimensions
- Consistently converges when integers, so need to scale to certain large ranges, then round

# Linear Assignment

Results

Results 1

- This is an example of two of the same shapes warped together at various lambda values
- Higher lambda means distance matters more, decreases amount of warping
- Different weights for distance matrix gives us different degrees of warping

# Linear Assignment

Results

Results 2

- This is an example of two different shapes warped together at various lambda values
- Different weights for distance matrix gives us different degrees of warping
- As you can see, not much real change until very low lambda values

# Linear Assignment

Results

Results 3

- Here are the results for matching
- As you can see, for optimized lambda you get better than usual matching

## FUTURE RESEARCH

**RETRIEVAL MECHANICS**

- Run on standard retrieval datasets
- Investigate alternative retrieval structures

**FEATURE REPRESENTATION**

- Delve deeper into the wavelet density estimation
- Optimize wavelet density estimation code
- Linear Assignment

Optimization we will expand to multi resolution, 1D, 3D.
Linear assignment we will find the best lambda to get the best results and test on multi resolution
Future we will try to find better feature representations instead of LBO signatures.
Investigate ways to visualize these high dimensional feature representations.

## FUTURE RESEARCH

**RETRIEVAL MECHANICS**

- Run on standard retrieval datasets
- Investigate alternative retrieval structures

**FEATURE REPRESENTATION**

- Delve deeper into the wavelet density estimation
- Optimize wavelet density estimation code
- Linear Assignment

Optimization we will expand to multi resolution, 1D, 3D.
Linear assignment we will find the best lambda to get the best results and test on multi resolution
Future we will try to find better feature representations instead of LBO signatures.
Investigate ways to visualize these high dimensional feature representations.