SHAPE REPRESENTATIONS

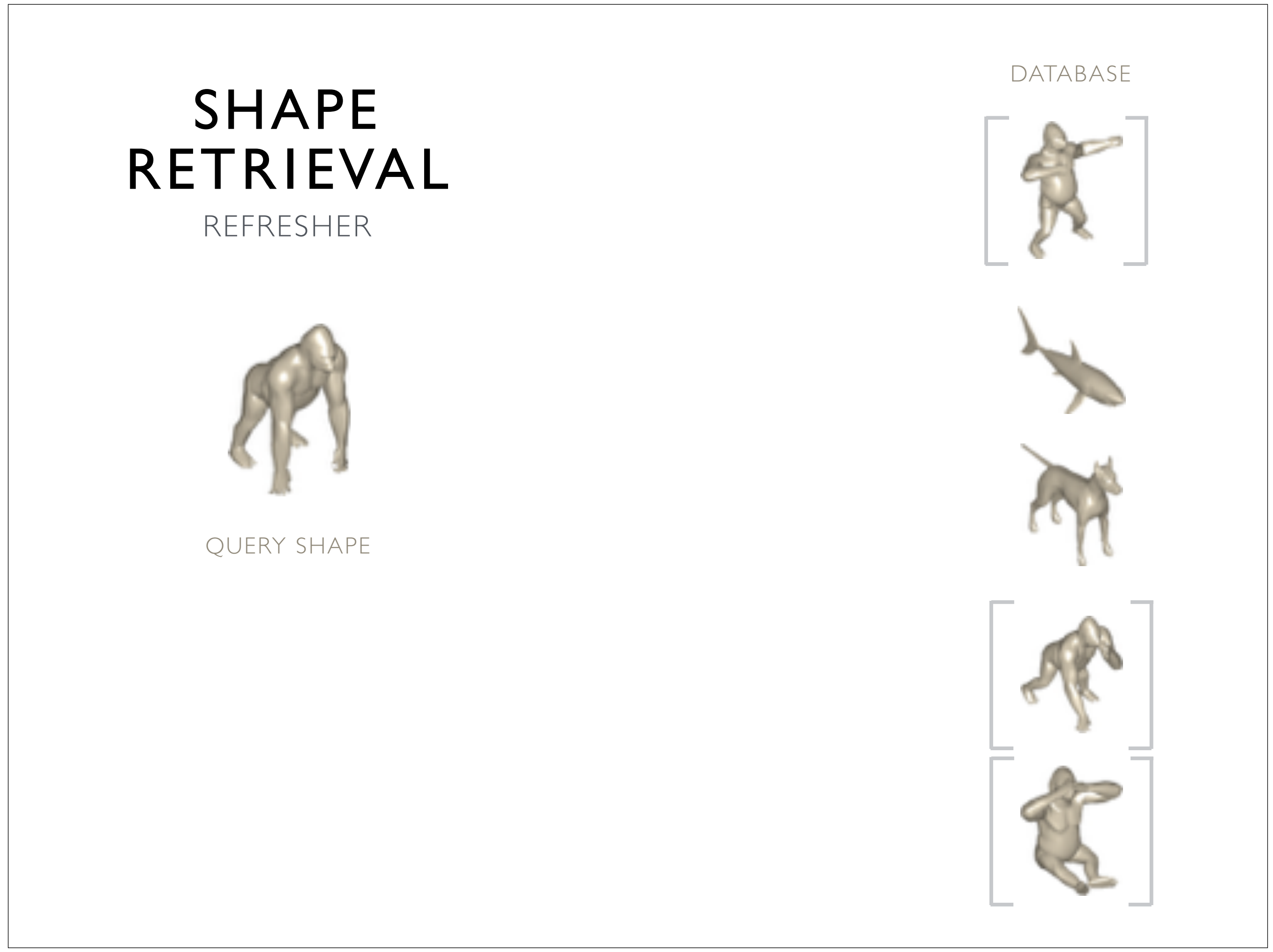
glizela taino

yixin lin

mark moyou

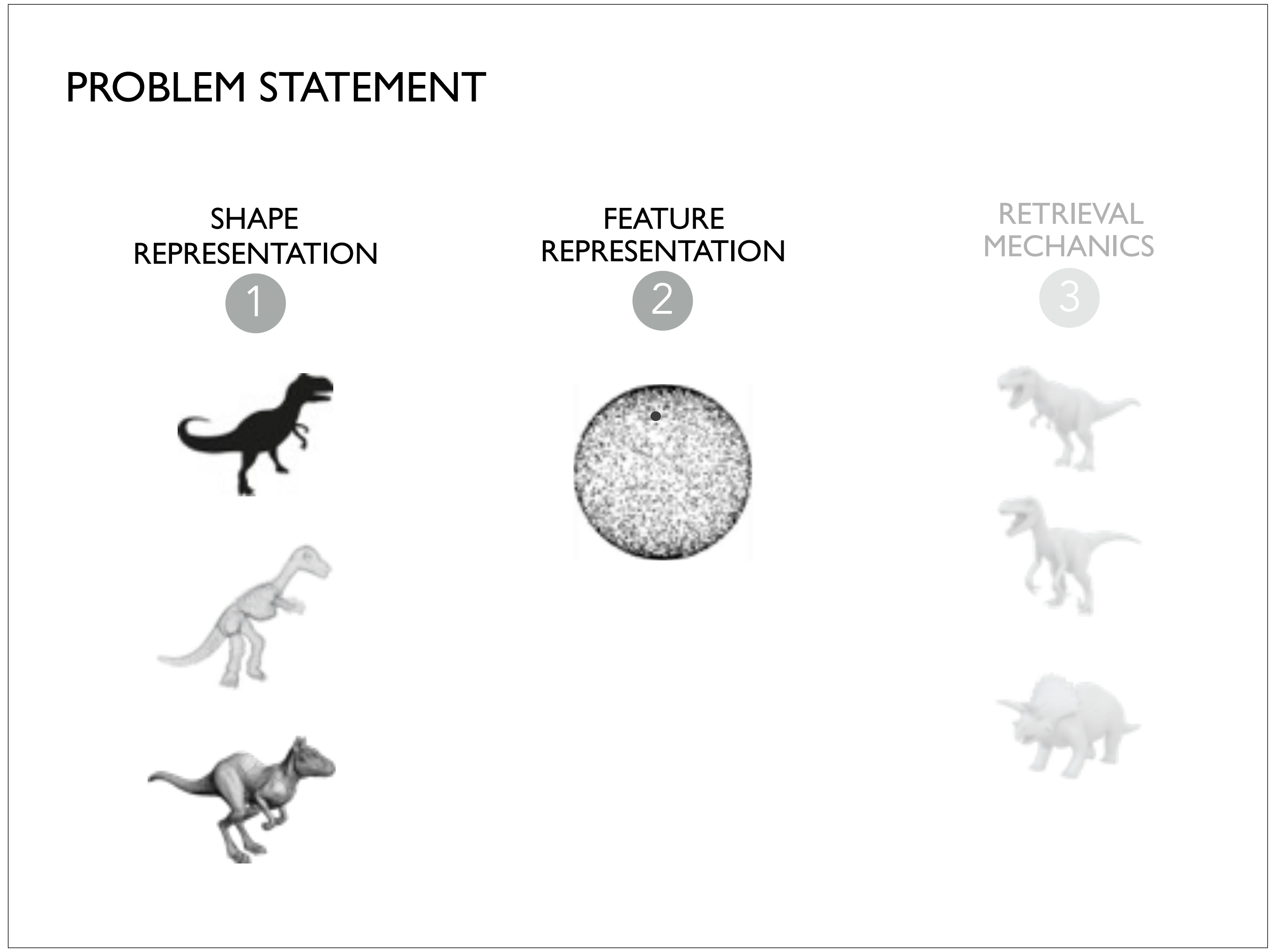
adrian peter

ZIXIN



ZELA

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.



YIXIN

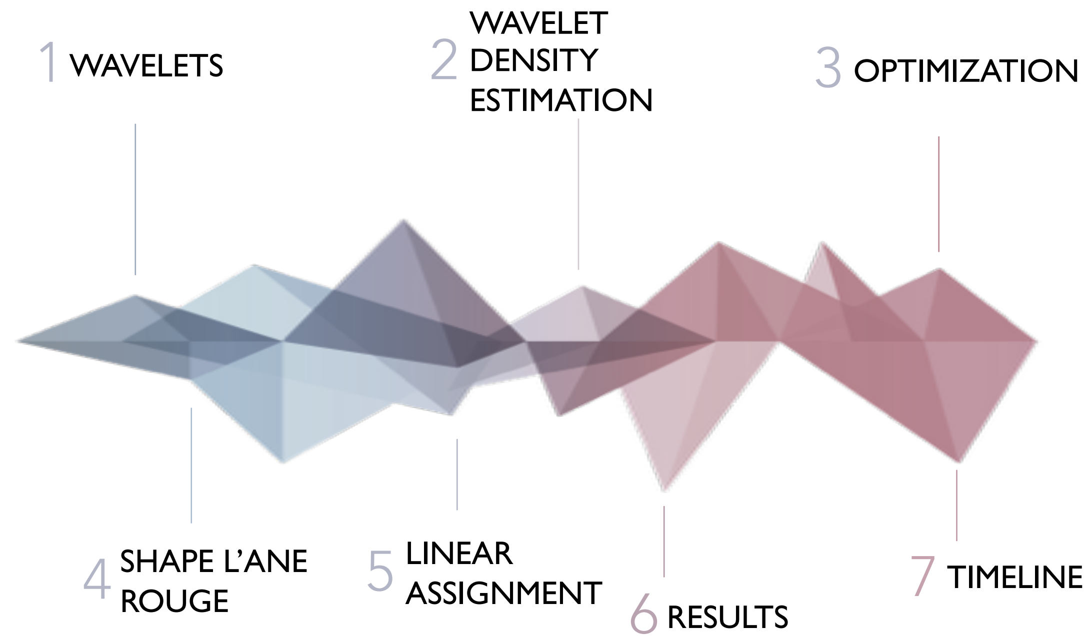
This is the general process of shape retrieval.

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.

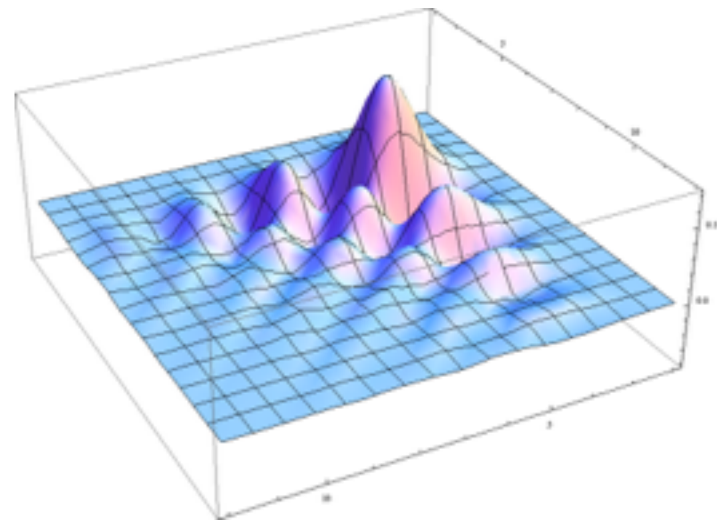
ROADMAP FOR FEATURE REPRESENTATION



ZELA

Shape lane rouge, way to get better distances between points

WAVELETS



ZELA

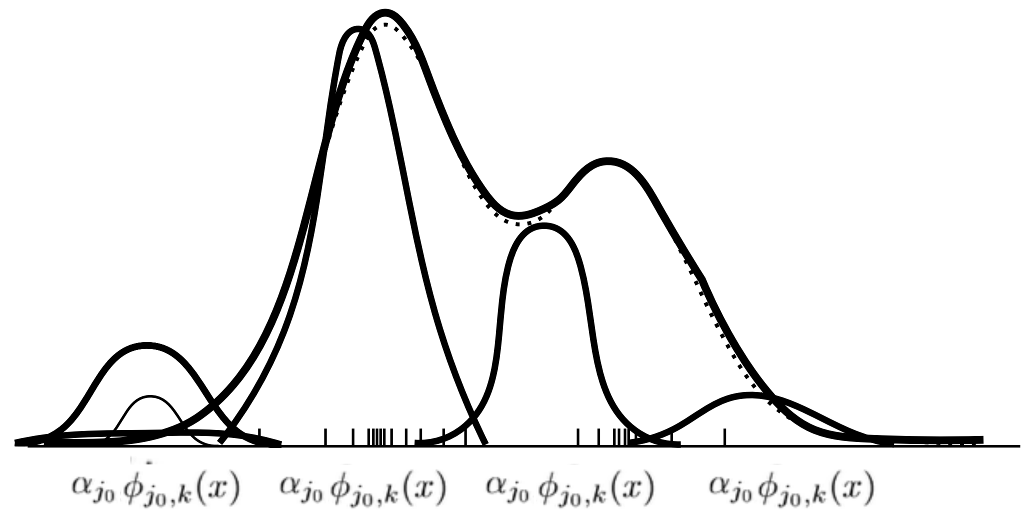
- A wavelet is a wave-like oscillation with an amplitude that begins at zero, increases, and then decreases back to zero.
- We use them for density estimation to construct probability densities that represent each shape and use as feature representation
- Why do we use them for density estimation?

WHY WAVELETS

YIXIN

- 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
    - 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

WAVELET DENSITY ESTIMATION



ZELA

- We have a set of points in 1D
- We want to estimate the the PDF
- We use wavelets which are a basis for L2, we can approximate the pdf
- You can choose resolution level which determine dilation
- In order to form a basis you need to make multiple copies over different translates
- Wavelets are scaled with a coefficients to estimate a PDF

WAVELETS, FORMALLY

$$\sqrt{p(x)} = \sum_{j_0,k} \underbrace{\alpha_{j_0,k}}_{\text{Scaling Coefficient}} \underbrace{\phi_{j_0,k}(x)}_{\text{Scaling Basis Function}} + \sum_{j \geq j_0,k}^{\infty} \underbrace{\beta_{j,k}}_{\text{Wavelet Coefficient}} \underbrace{\psi_{j,k}(x)}_{\text{Wavelet Basis Function}}$$

Father Mother

$$\alpha_{j_o,\mathbf{k}} = \frac{1}{N} \frac{\sum_{i=1}^{N} \phi_{j_o,\mathbf{k}}(\mathbf{x})}{\sqrt{p(\mathbf{x})}}$$

$$\psi^2_{j,\mathbf{k}}(\mathbf{x}) = 2^j \psi \int \frac{}{\sqrt{p(x)}} \phi_{j_0,k}(x)dx$$

$$\psi^3_{j,\mathbf{k}}(\mathbf{x}) = 2^j \psi\, \mathcal{E}\left[\frac{\phi_{j_0,k}(x)}{\sqrt{p(x)}}\right].$$

YIXIN

- 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 act as scalars of this linear combination to 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 scale the basis functions to uniquely determine a function (and therefore pdf)

- 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

- That was the single resolution version
  - To get a wavelet basis for 2D, we need to take the tensor product of our bases

- 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

WAVELET DENSITY ESTIMATION

NEGATIVE LOG LIKELIHOOD

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

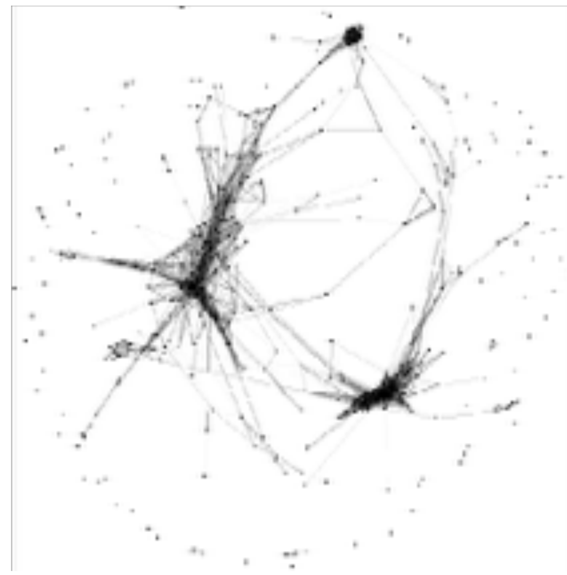
$$
\sum_{j_0,k} \alpha_{j_0,k}^2 + \sum_{j \geq j_0,k}^{j_1} \beta_{j,k}^2 = 1.
$$

YIXIN

Important equations

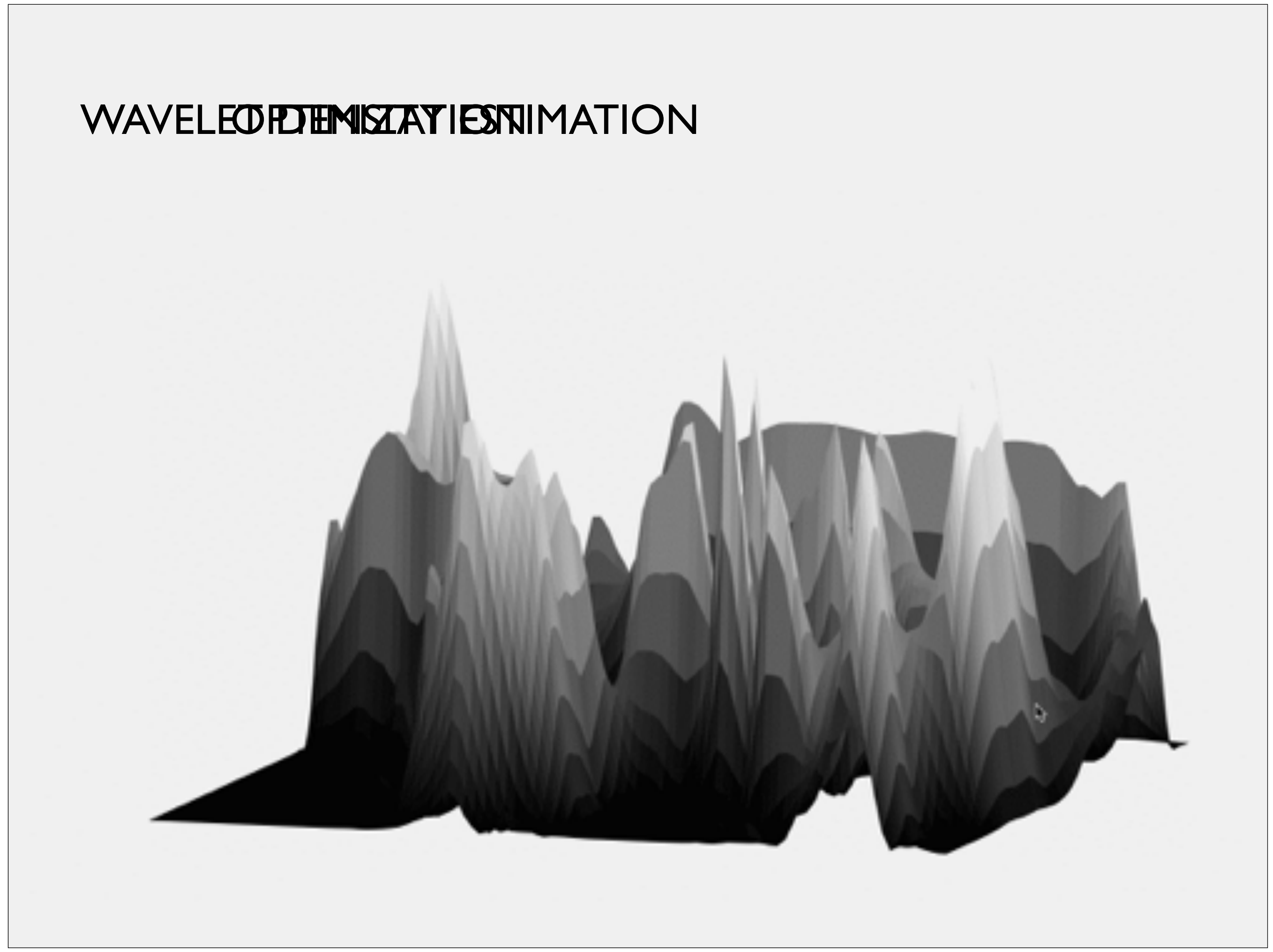
- One common loss function is the negative log likelihood function
  - A measure of how wrong we are with our density estimation
- We want to minimize this
- It turns out that we have a constraint: coefficients sum to one

WAVELET DENSITY ESTIMATION



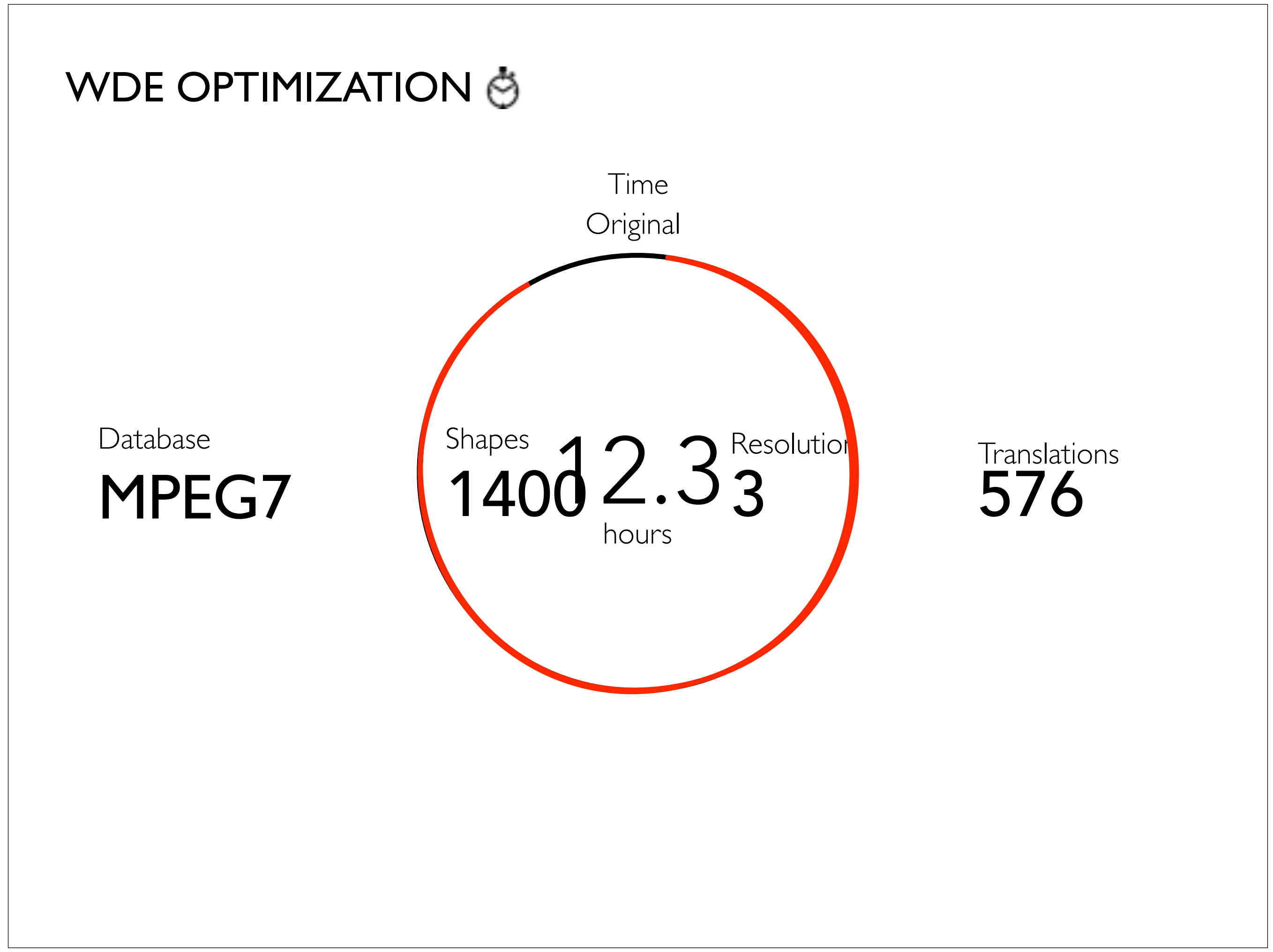
ZELA

- Since we use wavelets to estimate the probability density and found the coefficients
- Use as our feature representation
- Conveniently mapped on the unit hypersphere
    - So we get nice geometric properties for shape retrieval



YIXIN

- We have code that estimates the wavelet density.
- Our goal is to estimate a 2d density function on these shape points.
- The density function being estimated on the points that make up the cow.
- And notice the density looks like the point set.
- We can use this as a feature representation of our 2d shapes.



ZIXIN
For the database MPEG7 of 1400 shapes with domain [-1,1], resolution 3, and 24 translates

We would process very large databases like MPEG7 that has 1400 shapes or Brown that has 99 shapes.
The problem is that processing these databases are extremely slow.

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.

How many PHD students are going to be using this code? A million? No, more than that. In a few years, I bet five million PHD students will be running their Wavelet Density Estimators at least once a day.

Well, let's say you can shave 11 hours off the run time. Multiply that by five million users and thats 55 million hours, every single day. Over a year, that's probably 30,000 lifetimes. So if you make it run 11 hours faster, you've saved a 30,000 lives. That's really worth it, don't you think?

WDE OPTIMIZATION

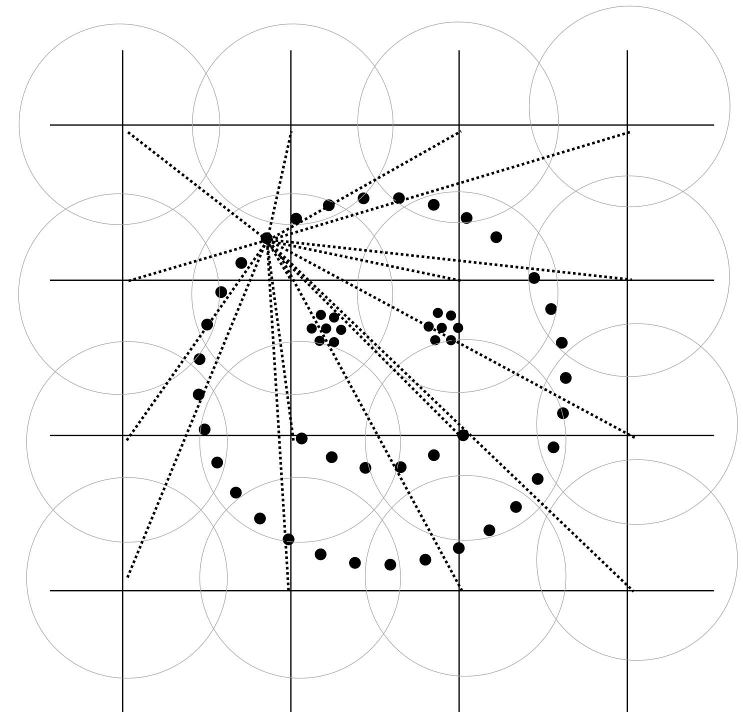
INITIALIZE COEFFICIENTS PROBLEM

$$\alpha_{j_o,\mathbf{k}} = \frac{1}{N}\frac{\sum_{i=1}^{N}\phi_{j_o,\mathbf{k}}(\mathbf{x})}{\sqrt{p(\mathbf{x})}}$$

576 wavelet functions

$\times$ 4007 samples

2,308,032 operations



ZELA

Problem

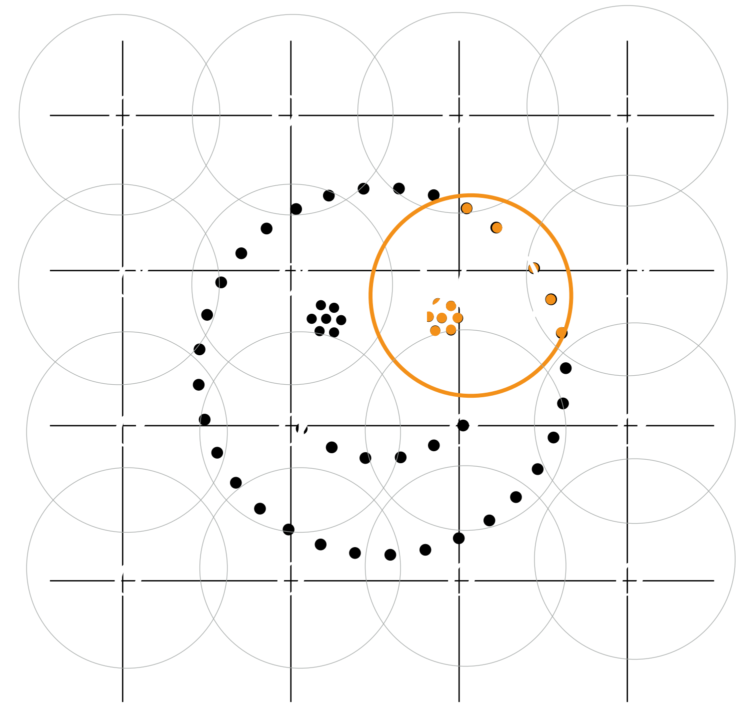
- Function that is extremely slow
- Given a point set, we cover the shape with wavelet functions
- each function has a coefficient we want to calculate.
- Find how each point contributes to each basis function
- PROBLEM: The code performs the tensor product of a single point over all translations.
- IF 576 translations. And does this 4007 more times for each sample performing a total of 331,776 operations.

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.

WDE OPTIMIZATION

INITIALIZE COEFFICIENTS SOLUTION

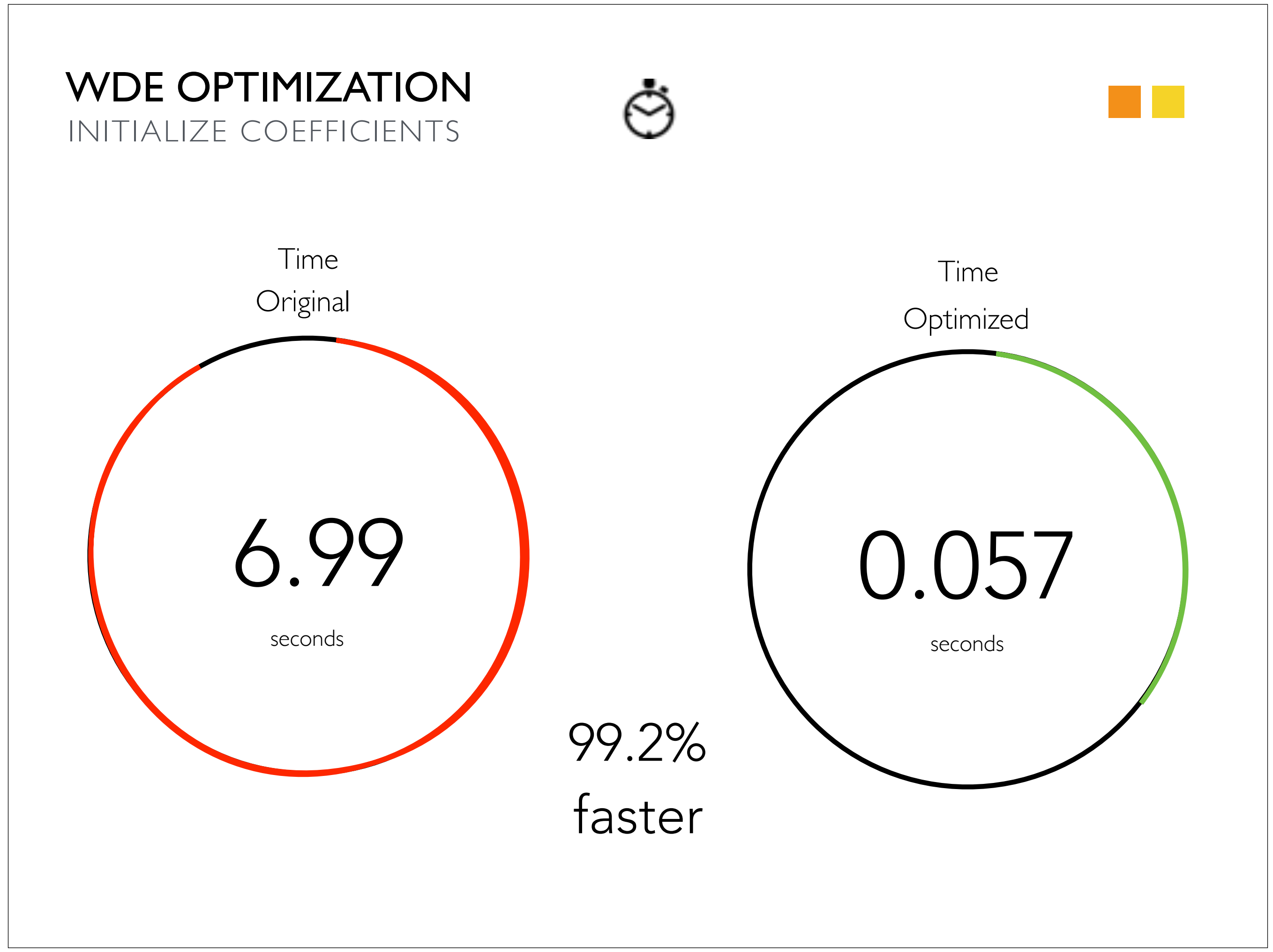
$$\alpha_{j_o,\mathbf{k}} = \frac{1}{N}\frac{\sum_{i=1}^{N}\phi_{j_o,\mathbf{k}}(\mathbf{x})}{\sqrt{p(\mathbf{x})}}$$



ZELA

Solution

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



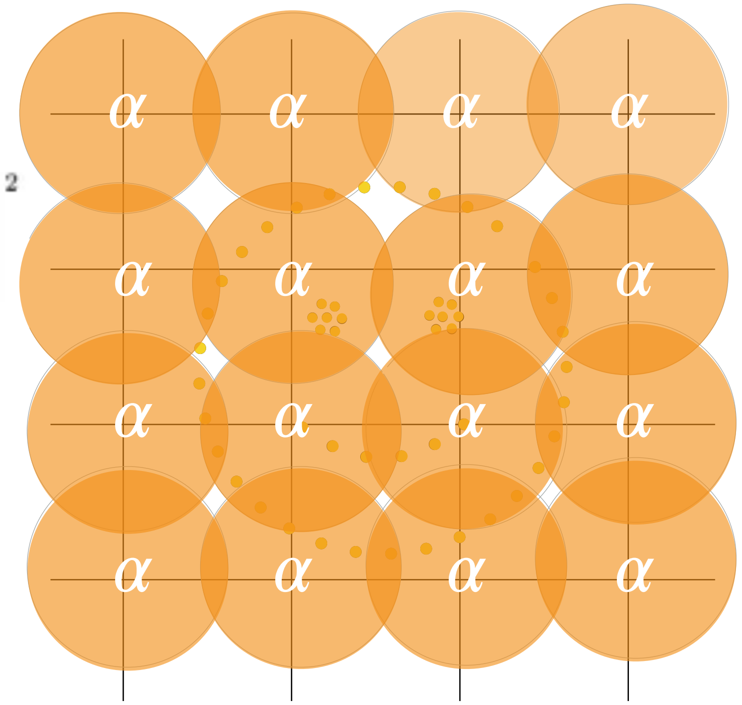
YIXIN
Show result timestamp of optimization of initializeCoefficients function.

WDE OPTIMIZATION

NEGATIVE LOG LIKELIHOOD SOLUTION

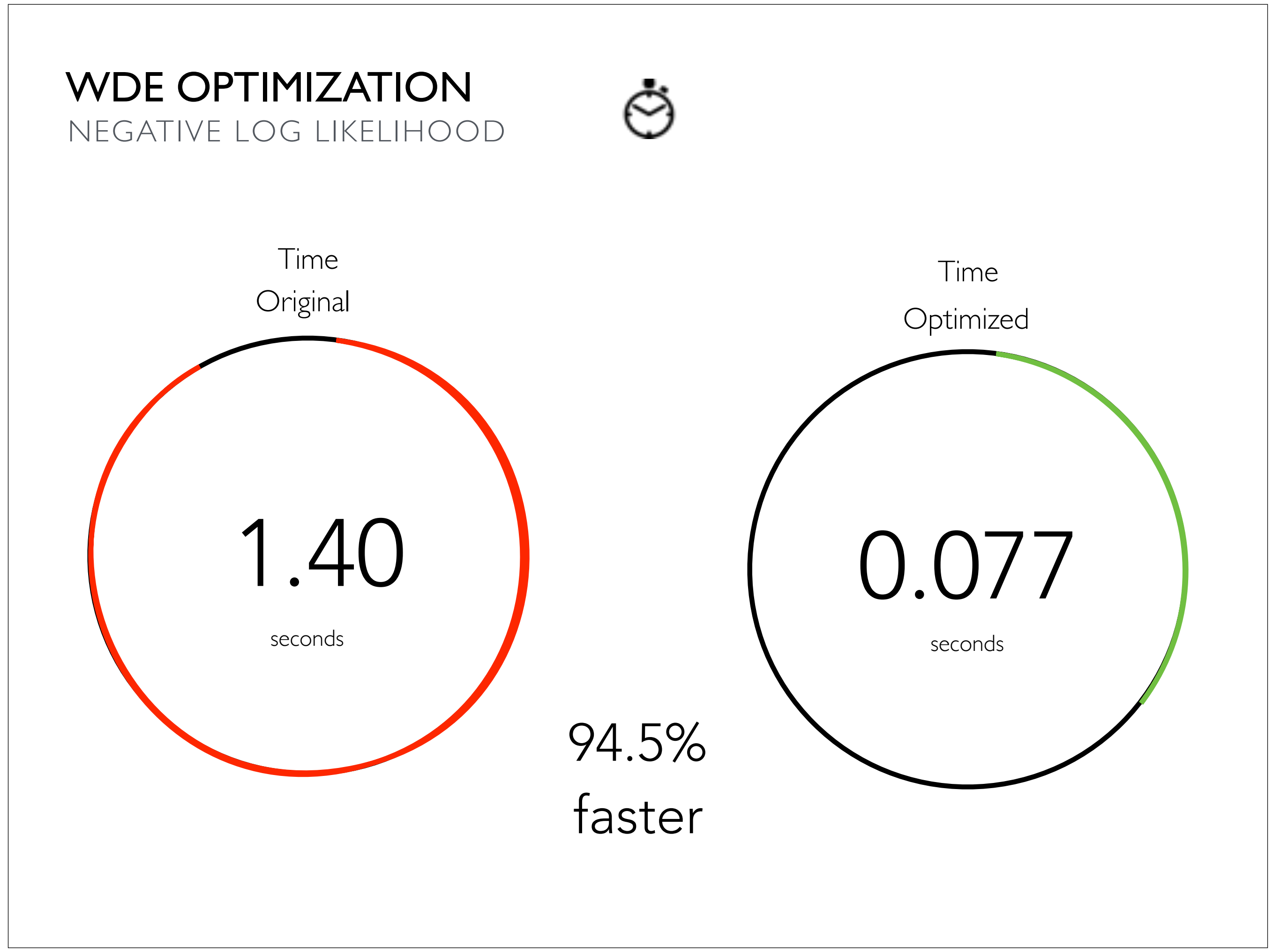
$$-\log p(X;\{\alpha_{j_0,k},\beta_{j,k}\})$$

$$= -\frac{1}{N}\sum_{i=1}^{N}\log\left[\sum_{j_0,k}\alpha_{j_0,k}\phi_{j_0,k}(x_i)\right]^2$$

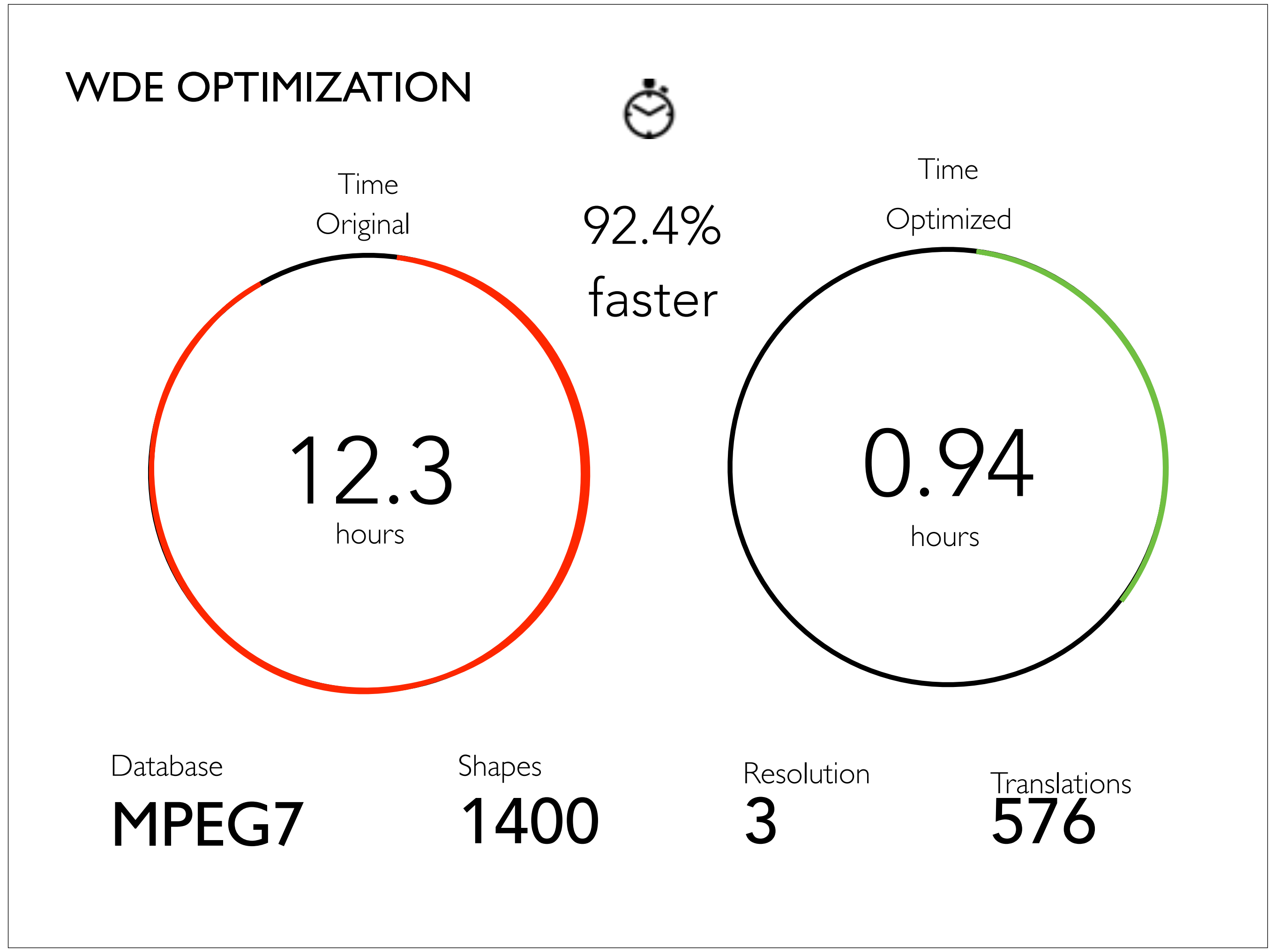


ZELA 16 MINS

- negativeLogLikelihood is also extremely slow
- The problem is the same as initialize coefficients. Takes a single point and calculates wavelet function over all translations
- Our solution to initializeCoefficients allows us to pass in the appropriate variables needed and simply perform operations.
- We complete got rid of loops.



YIXIN
Show result timestamp of optimization of initializeCoefficients function.



ZELA

- Before optimization it took half a day to estimate densities
- Now it take about an hour
- Saving 30,000 lives a year

YIXIN

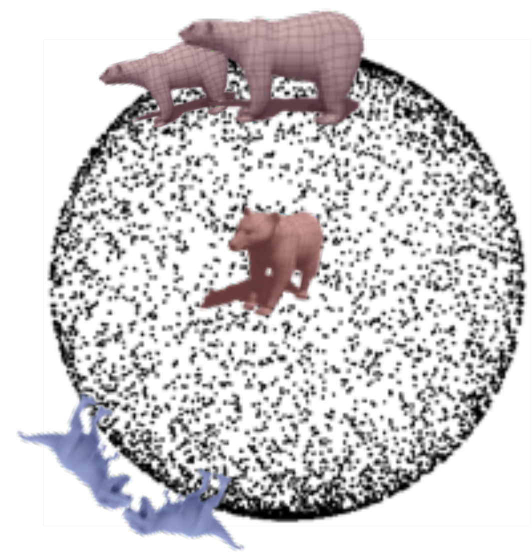
- Now we found feature representations we will investigate better ways to measure distances between them.

SHAPE L'ANE ROUGE

ZELA

- High level: our goal is to get a more accurate distance metric between shapes
- We warp shapes to try to match each other

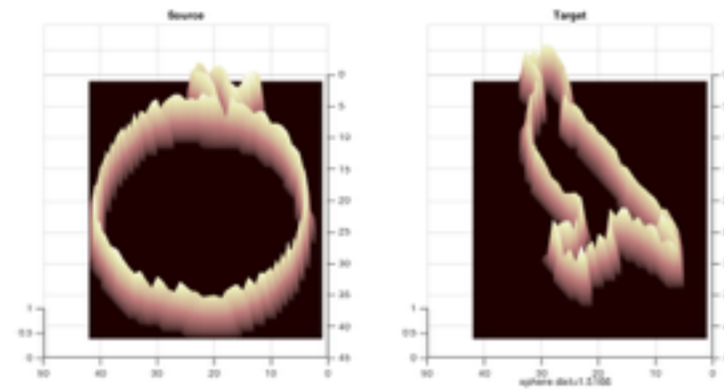
SHAPE L’ANE ROUGE



YIXIN

- Intuition: it’s easier to warp similar shapes close to each other, harder to warp different shapes
    - Distances get smaller between similar shapes
    - Not much smaller between dissimilar shapes
- Formally: reduce 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

SHAPE L'ANE ROUGE



YIXIN

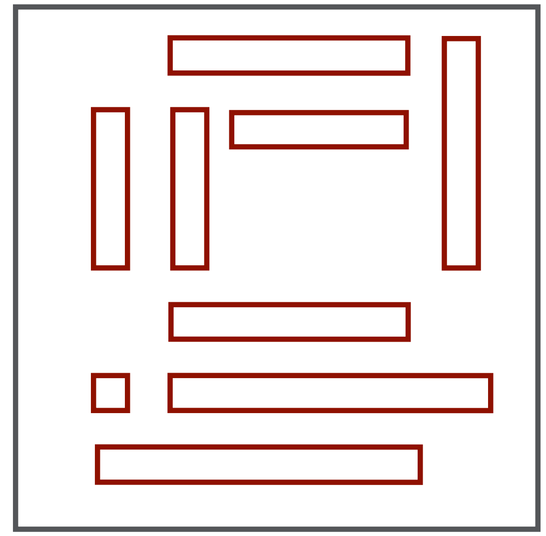
SHAPE L'ANE ROUGE

$$\begin{bmatrix} 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0361 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0472 \end{bmatrix}$$

ZELA

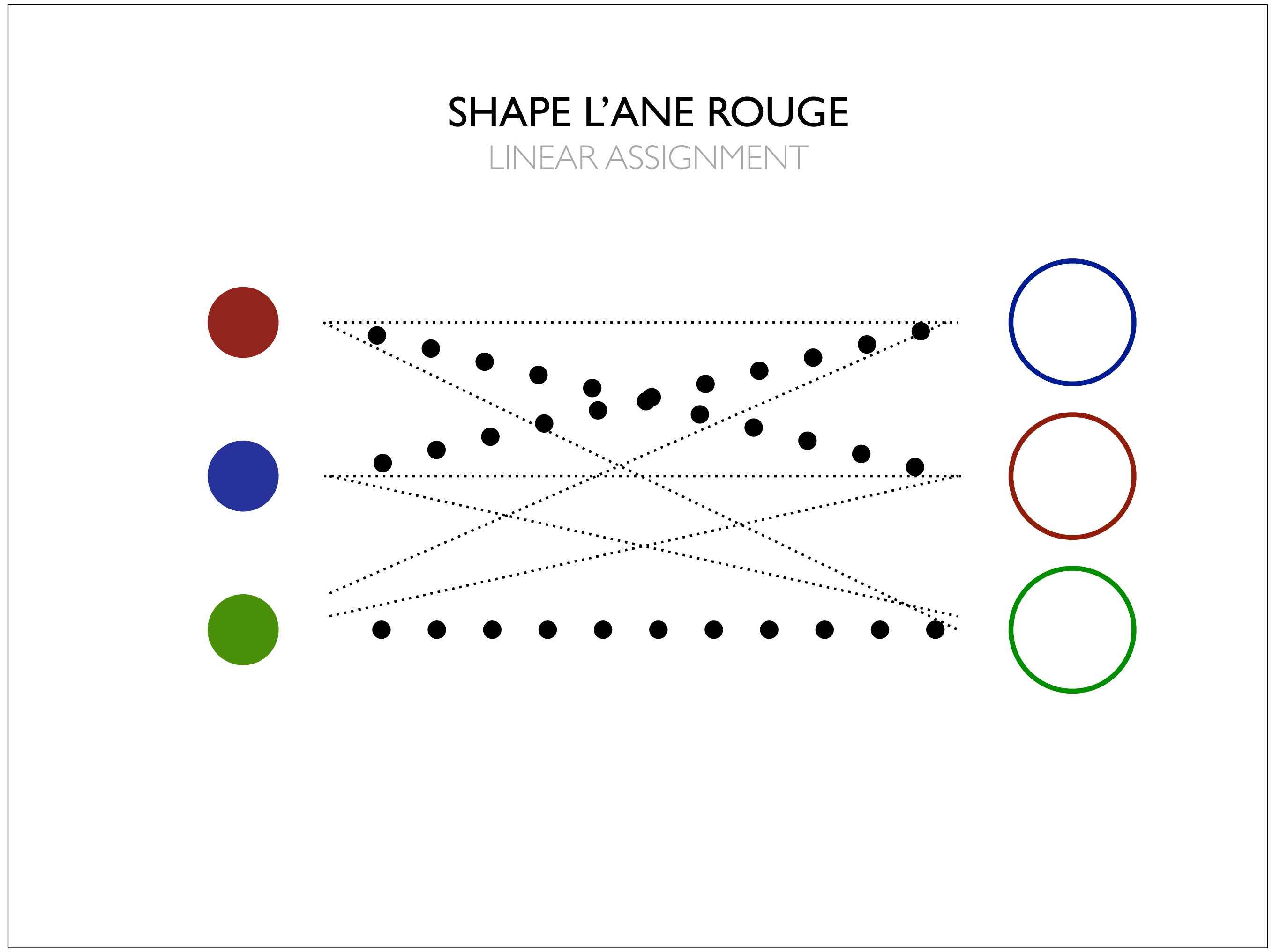
- Heres a grid
- Each point in the grid represent a coefficient
  - Got this from the code that we optimized
- So what that looks like in MATLAB is this matrix
- We need it as a vector
- We have these coefficient vectors constructed on the grid
- When comparing two shapes, we want to fit the two vectors together

SHAPE L'ANE ROUGE



YIXIN

- Why is it called L'Ane Rouge?
- Is a French children's game involving sliding blocks
- Literally means red donkey
- Good analogy because each block slides around
    - We slide the coefficients around
- For Shape L'Ane Rouge, use a combinatorial approach called linear assignment



YIXIN

High Level example: assigning workers to specific tasks

- Originally arose in transportation theory
- Question: You have n students and n grad mentors
    - Disclaimer: not based on real people
    - Suppose each student prefers to work with a grad mentor a different amount
        - Represent this by a “cost”: lower cost means they like the mentor better
    - The question is to match grad mentors to students in the least cost way
- Generally, the idea is to assign n workers n tasks while minimizing cost

SHAPE L'ANE ROUGE

LINEAR ASSIGNMENT, FORMALLY

$$\mathrm{s}\,X, Y\,| = |Y|$$

$$C : X \times Y \to \mathbb{R}$$

$$X \to Y$$

YIXIN

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
- Our goal is to find a permutation

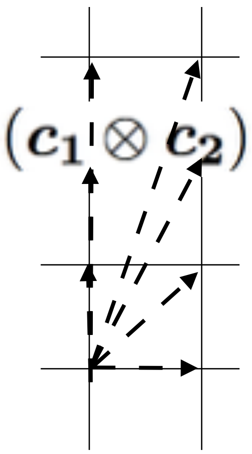
SHAPE L'ANE ROUGE
JONKER-VOLGENANT

ZELA
- How do we solve this?
- We use a speedy algorithm called Jonker-Volgenant
- Input is a cost matrix that tells you the cost between every pair of points
- We need to construct this cost matrix that tells you how hard it is to move coefficients around

SHAPE L'ANE ROUGE

COST MATRIX

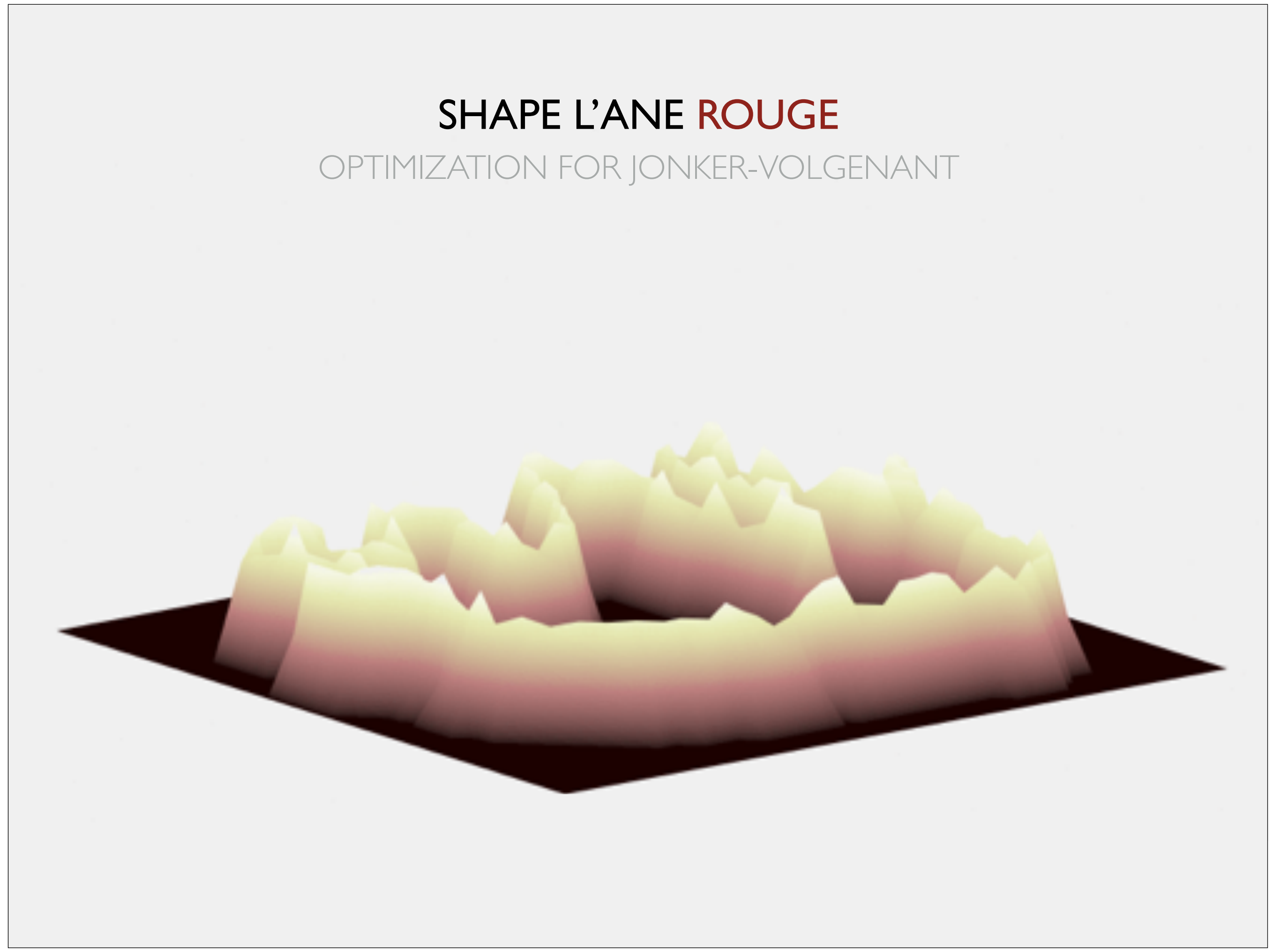
$$C = -(\boldsymbol{c_1} \otimes \boldsymbol{c_2}) + D$$



YIXIN

Implementation

- When you're trying to find the distance between two shapes
- Shape L'Ane Rouge makes it better
- Let's try to fit two shapes together first using linear assignment, and then take the 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
        - Why do we have distance in our cost matrix?
            - Interpoint distances: how freely can we move the coefficients?
    - 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 distance matrix with a constant called lambda (we pick this)
        - Interpoint matters less with small lambda
        - Large lambda means harder to shift points farther
        - Small lambda means easier to shift points farther
    - Then we subtract the outer product from the weighted distance matrix and try to minimize this cost matrix



YIXIN

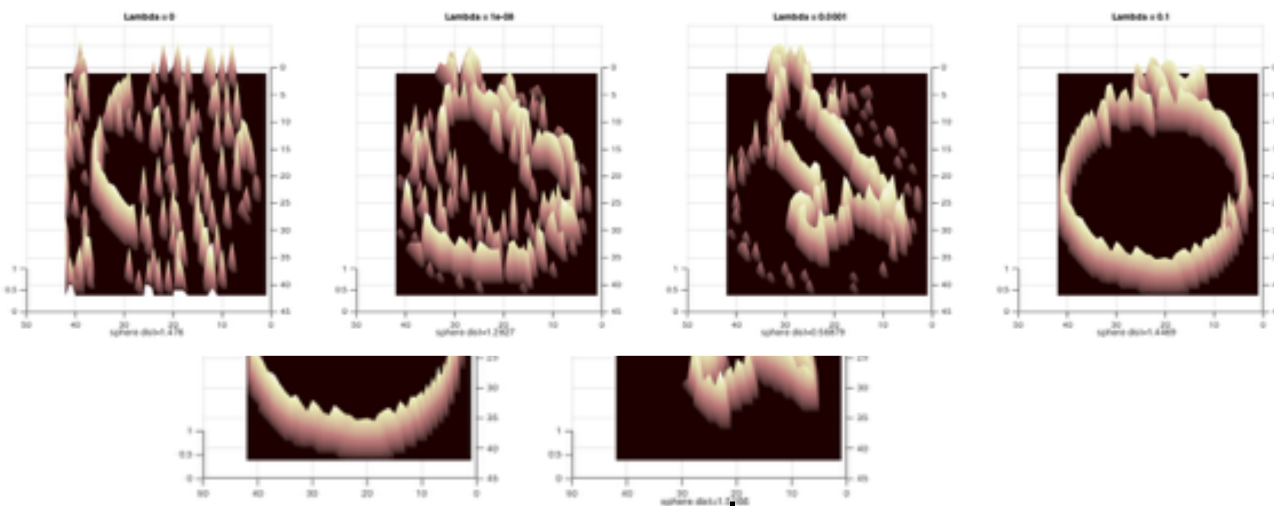
PROBLEM

- Jonker-Volgenant
- Coefficients usually very close to 0
- Consistently converges when integers, so need to scale to certain large ranges, then round

VISUAL: two histograms that show the processing of the data.

SHAPE L'ANE ROUGE

DIFFERENT SHAPE WARP



$\frac{1}{\lambda}$ ⋯⋯▸ MOVEMENT

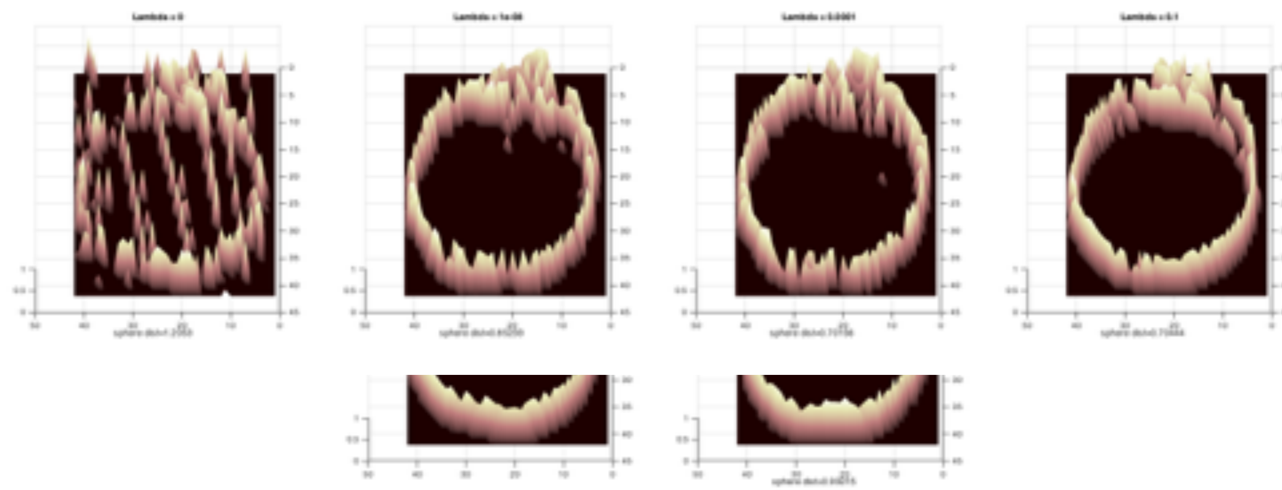
YIXIN

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

SHAPE L'ANE ROUGE

SAME SHAPE WARP



$\frac{1}{\lambda} \dashrightarrow$ MOVEMENT

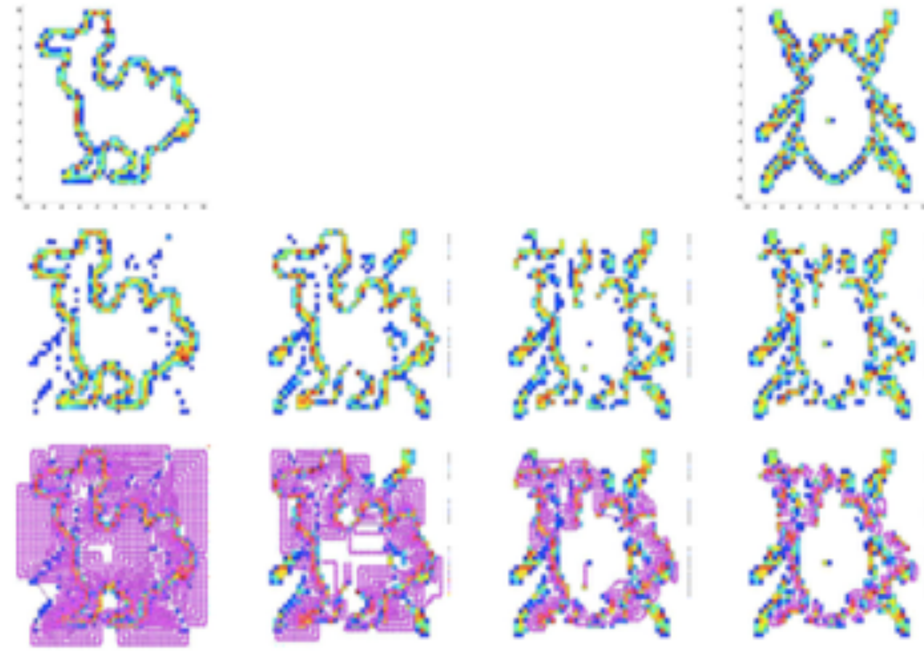
YIXIN

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
    - Ignore interpoint distance relationships
- Different weights for distance matrix gives us different degrees of warping

SHAPE L'ANE ROUGE

RESULTS ON DATASETS



YIXIN

Results 3

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

FUTURE RESEARCH

- Optimization
  - Optimize multi-resolution
  - Extend to different dimensions
- Shape L'Ane Rouge
  - Optimize λ
  - Test on datasets
- Find better feature representations
- Investigate high-dimension visualization

FEATURE REPRESENTATION

ZIXIN
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.

THANK YOU

ZIXIN