Computer Graphics

Proximity

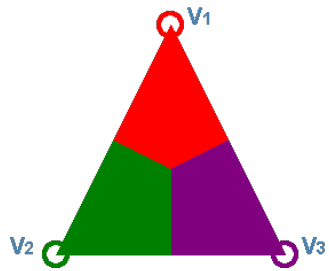
Emanuele Rodolà
rodola@di.uniroma1.it

SAPIENZA
UNIVERSITÀ DI ROMA

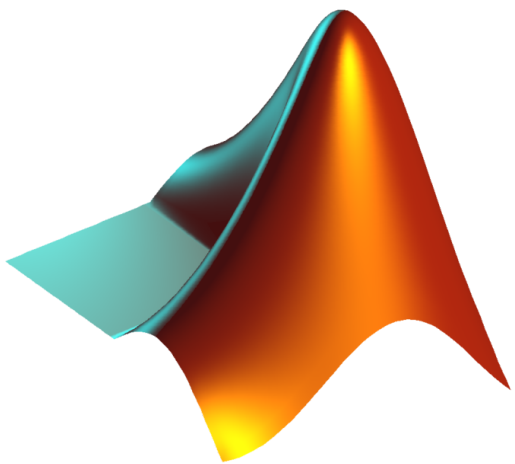
2nd semester a.y. 2018/2019 · March 20, 2019

Exercises

- Visualize function $f(x_i) = \sum_j \|x_i - x_j\|$ with the blue-white-red colormap



- Nearest-neighbor colors for arbitrary triangles

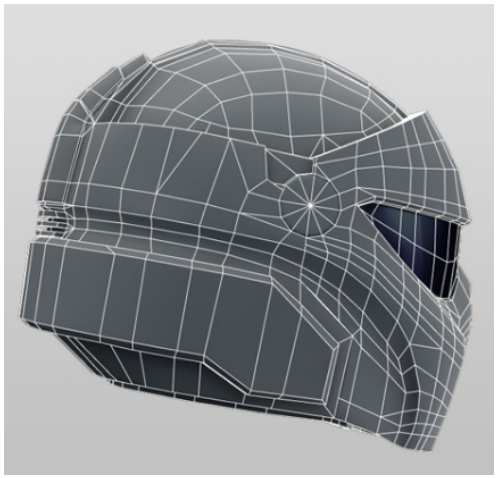


Proximity

The general notion of proximity or neighborhood arises everywhere

Proximity

The general notion of proximity or neighborhood arises everywhere

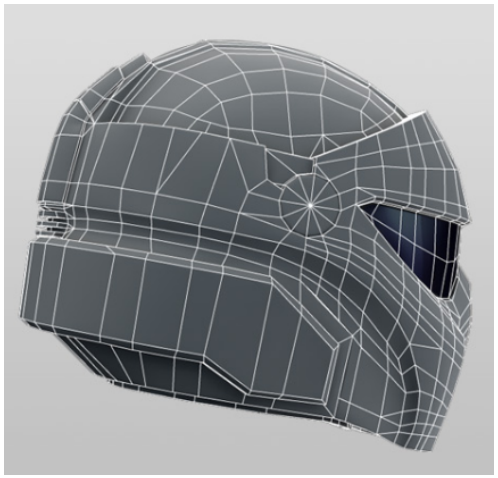


connectivity of nodes
in a mesh/graph

distance between pixels
in color space

Proximity

The general notion of proximity or neighborhood arises everywhere



connectivity of nodes
in a mesh/graph

distance between pixels
in color space

Proximity is not necessarily a spatial notion in the classical sense

Adjacency matrices

Graph connectivity can be encoded in adjacency matrices

Let $|V| = n$, $|E| = e$, $|F| = m$ for a mesh $M = (V, E, F)$

Adjacency matrices: Vertex-to-vertex

Graph connectivity can be encoded in adjacency matrices

Let $|V| = n$, $|E| = e$, $|F| = m$ for a mesh $M = (V, E, F)$

The vertex-to-vertex adjacency is defined as the $n \times n$ binary matrix:

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 1 \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ 1 & 0 & 1 & \cdots & 0 \end{pmatrix}$$

where $a_{ij} = 1$ if vertex v_i is connected to v_j (that is, $e_{ij} \in E$)

Adjacency matrices: Vertex-to-vertex

Graph connectivity can be encoded in adjacency matrices

Let $|V| = n$, $|E| = e$, $|F| = m$ for a mesh $M = (V, E, F)$

The vertex-to-vertex adjacency is defined as the $n \times n$ binary matrix:

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 1 \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ 1 & 0 & 1 & \cdots & 0 \end{pmatrix}$$

where $a_{ij} = 1$ if vertex v_i is connected to v_j (that is, $e_{ij} \in E$)

- The diagonal is always 0

Adjacency matrices: Vertex-to-vertex

Graph connectivity can be encoded in adjacency matrices

Let $|V| = n$, $|E| = e$, $|F| = m$ for a mesh $M = (V, E, F)$

The vertex-to-vertex adjacency is defined as the $n \times n$ binary matrix:

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 1 \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ 1 & 0 & 1 & \cdots & 0 \end{pmatrix}$$

where $a_{ij} = 1$ if vertex v_i is connected to v_j (that is, $e_{ij} \in E$)

- The diagonal is always 0
- $\mathbf{A}$ is symmetric

Adjacency matrices: Vertex-to-vertex

Graph connectivity can be encoded in adjacency matrices

Let $|V| = n$, $|E| = e$, $|F| = m$ for a mesh $M = (V, E, F)$

The vertex-to-vertex adjacency is defined as the $n \times n$ binary matrix:

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 1 \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ 1 & 0 & 1 & \cdots & 0 \end{pmatrix}$$

where $a_{ij} = 1$ if vertex v_i is connected to v_j (that is, $e_{ij} \in E$)

- The diagonal is always 0
- $\mathbf{A}$ is symmetric
- Each row and column has at least one 1 (that is, $\sum_{ij} a_{ij} = e$)

Adjacency matrices: Vertex-to-triangle

Graph connectivity can be encoded in adjacency matrices

Let $|V| = n$, $|E| = e$, $|F| = m$ for a mesh $M = (V, E, F)$

The vertex-to-triangle adjacency is defined as the $n \times m$ binary matrix:

$$\mathbf{P} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 1 & 1 \\ \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ 0 & 1 & 0 & \cdots & 1 & 0 & 1 \end{pmatrix}$$

where $p_{ij} = 1$ if vertex v_i belongs to triangle t_j

Adjacency matrices: Vertex-to-triangle

Graph connectivity can be encoded in adjacency matrices

Let $|V| = n$, $|E| = e$, $|F| = m$ for a mesh $M = (V, E, F)$

The vertex-to-triangle adjacency is defined as the $n \times m$ binary matrix:

$$\mathbf{P} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 1 & 1 \\ \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ 0 & 1 & 0 & \cdots & 1 & 0 & 1 \end{pmatrix}$$

where $p_{ij} = 1$ if vertex v_i belongs to triangle t_j

- Each row has at least one 1 (each vertex belongs to some triangle)

Adjacency matrices: Vertex-to-triangle

Graph connectivity can be encoded in adjacency matrices

Let $|V| = n$, $|E| = e$, $|F| = m$ for a mesh $M = (V, E, F)$

The vertex-to-triangle adjacency is defined as the $n \times m$ binary matrix:

$$\mathbf{P} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 1 & 1 \\ \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ 0 & 1 & 0 & \cdots & 1 & 0 & 1 \end{pmatrix}$$

where $p_{ij} = 1$ if vertex v_i belongs to triangle t_j

- Each row has at least one 1 (each vertex belongs to some triangle)
- Each column sums up to 3 (each triangle has exactly 3 vertices)

Adjacency matrices: Triangle-to-triangle

Consider the product:

$$\mathbf{P}^\top \mathbf{P} = \begin{pmatrix} 1 & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & 1 \\ 0 & \cdots & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & \cdots & \cdots & 1 \\ 1 & \cdots & \cdots & 0 \\ 1 & \cdots & \cdots & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 1 & 1 \\ \vdots & & & & & & \vdots \\ \vdots & & & & & & \vdots \\ 0 & 1 & 0 & \cdots & 1 & 0 & 1 \end{pmatrix}$$

which is a $m \times m$ matrix

Adjacency matrices: Triangle-to-triangle

Consider the product:

$$\mathbf{P}^\top \mathbf{P} = \begin{pmatrix} 1 & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & 1 \\ 0 & \cdots & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & \cdots & \cdots & 1 \\ 1 & \cdots & \cdots & 0 \\ 1 & \cdots & \cdots & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 1 & 1 \\ \vdots & & & & & & \vdots \\ \vdots & & & & & & \vdots \\ 0 & 1 & 0 & \cdots & 1 & 0 & 1 \end{pmatrix}$$

which is a $m \times m$ matrix

Cell (i, j) counts the number of vertices that t_i and t_j share

Adjacency matrices: Triangle-to-triangle

Consider the product:

$$\mathbf{P}^\top\mathbf{P} = \begin{pmatrix} 1 & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & 1 \\ 0 & \cdots & \cdots & 0 \\ \vdots & \cdots & \cdots & \vdots \\ 0 & \cdots & \cdots & 1 \\ 1 & \cdots & \cdots & 0 \\ 1 & \cdots & \cdots & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 1 & 1 \\ \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ 0 & 1 & 0 & \cdots & 1 & 0 & 1 \end{pmatrix}$$

which is a $m \times m$ matrix

Cell (i, j) counts the number of vertices that t_i and t_j share

- $\mathrm{diag}(\mathbf{P}^\top\mathbf{P})$ is always 3 (=number of vertices per triangle)

Adjacency matrices: Triangle-to-triangle

Consider the product:

$$\mathbf{P}^\top\mathbf{P} = \begin{pmatrix} 1 & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & 1 \\ 0 & \cdots & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & \cdots & \cdots & 1 \\ 1 & \cdots & \cdots & 0 \\ 1 & \cdots & \cdots & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 1 & 1 \\ \vdots & & & & & & \vdots \\ \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & & & & & & \vdots \\ 0 & 1 & 0 & \cdots & 1 & 0 & 1 \end{pmatrix}$$

which is a $m \times m$ matrix

Cell (i, j) counts the number of vertices that t_i and t_j share

- $\mathrm{diag}(\mathbf{P}^\top\mathbf{P})$ is always 3 (=number of vertices per triangle)
- All other values are 0 (non-adjacent triangles)

Adjacency matrices: Triangle-to-triangle

Consider the product:

$$\mathbf{P}^\top\mathbf{P} = \begin{pmatrix} 1 & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & 1 \\ 0 & \cdots & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & \cdots & \cdots & 1 \\ 1 & \cdots & \cdots & 0 \\ 1 & \cdots & \cdots & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 1 & 1 \\ \vdots & & & & & & \vdots \\ \vdots & & & & & & \vdots \\ 0 & 1 & 0 & \cdots & 1 & 0 & 1 \end{pmatrix}$$

which is a $m \times m$ matrix

Cell (i, j) counts the number of vertices that t_i and t_j share

- $\mathrm{diag}(\mathbf{P}^\top\mathbf{P})$ is always 3 (=number of vertices per triangle)
- All other values are 0 (non-adjacent triangles), 1 (triangles share one vertex)

Adjacency matrices: Triangle-to-triangle

Consider the product:

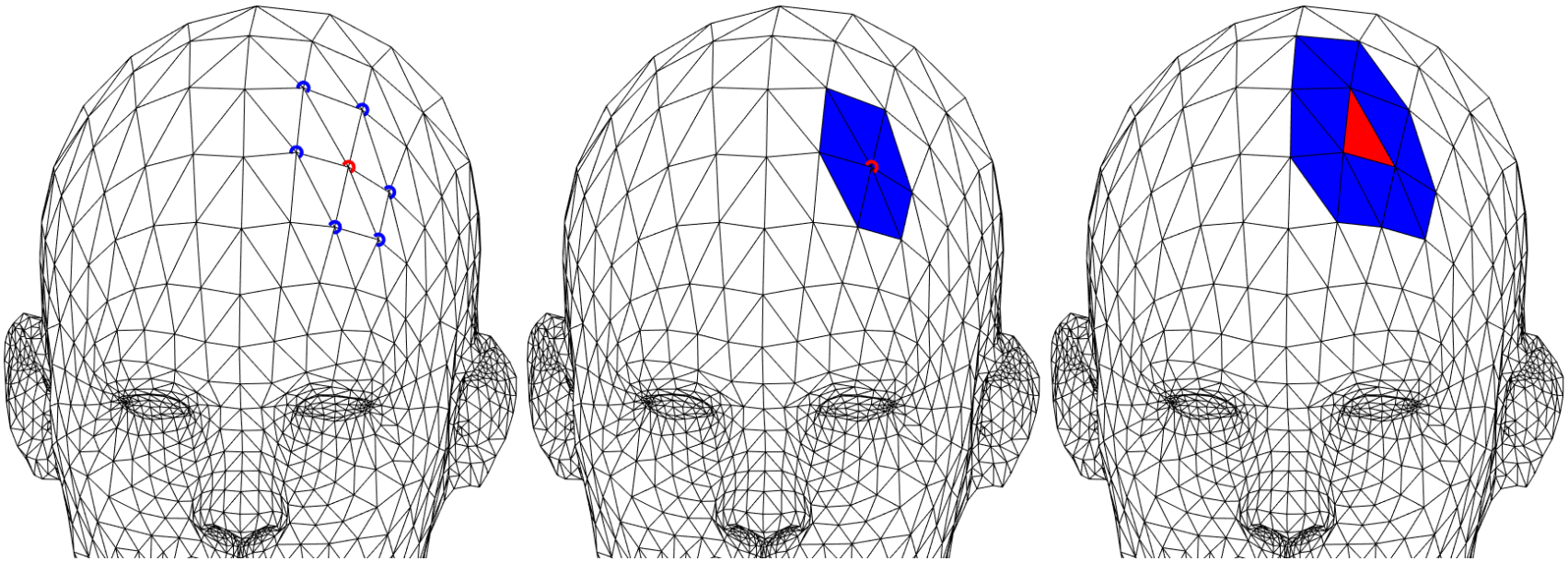
$$\mathbf{P}^\top\mathbf{P} = \begin{pmatrix} 1 & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & 1 \\ 0 & \cdots & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & \cdots & \cdots & 1 \\ 1 & \cdots & \cdots & 0 \\ 1 & \cdots & \cdots & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 1 & 1 \\ \vdots & & & & & & \vdots \\ \vdots & & & & & & \vdots \\ 0 & 1 & 0 & \cdots & 1 & 0 & 1 \end{pmatrix}$$

which is a $m \times m$ matrix

Cell (i, j) counts the number of vertices that t_i and t_j share

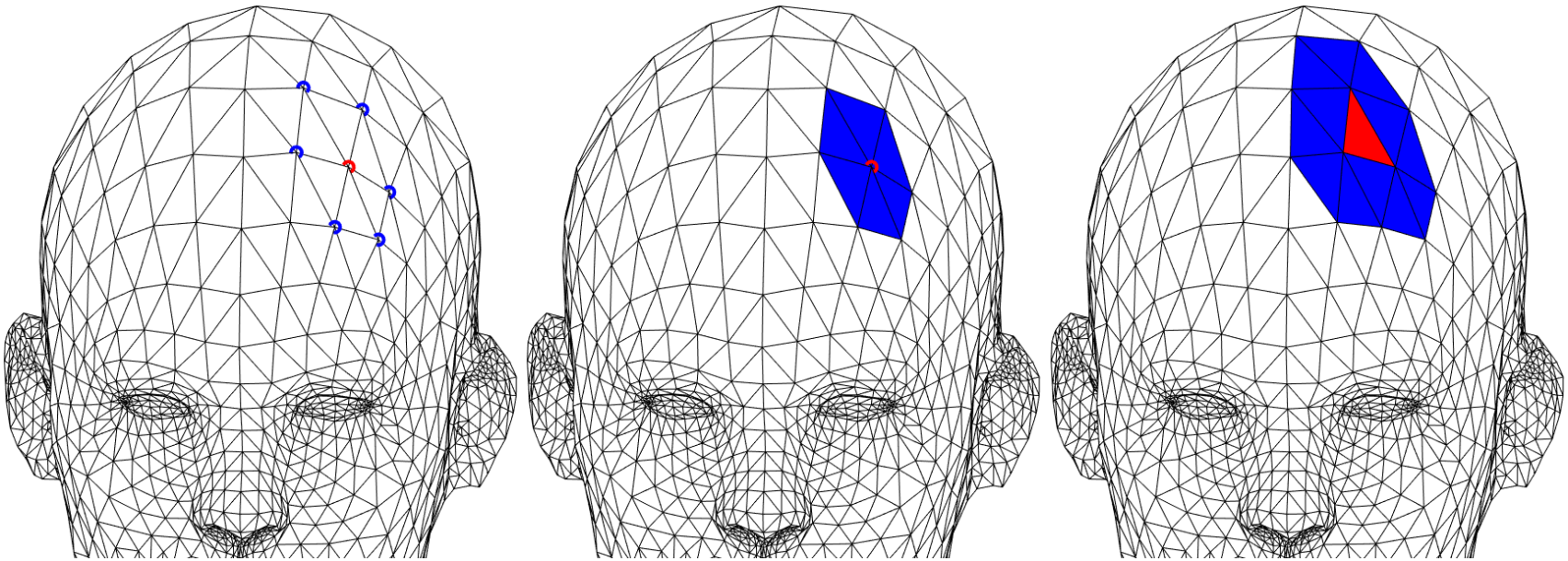
- $\mathrm{diag}(\mathbf{P}^\top\mathbf{P})$ is always 3 (=number of vertices per triangle)
- All other values are 0 (non-adjacent triangles), 1 (triangles share one vertex), or 2 (triangles share an edge)

Examples: Adjacency



vertex-to-vertex | vertex-to-triangle | triangle-to-triangle

Examples: Adjacency



vertex-to-vertex | vertex-to-triangle | triangle-to-triangle

Adjacency matrices can be very large (quadratic in n)

Better use sparse data structures to store them

Adjacency in color space

We can define a distance between colors, for example:

$$d_{ij} = \|\mathbf{c}_i - \mathbf{c}_j\|$$

This can be seen as a soft notion of adjacency

Adjacency in color space

We can define a distance between colors, for example:

$$d_{ij} = \|\mathbf{c}_i - \mathbf{c}_j\|$$

This can be seen as a soft notion of adjacency

Classical "hard" (or binary) adjacency may be defined as:

$$a_{ij} = \begin{cases} 1 & \text{if } \|\mathbf{c}_i - \mathbf{c}_j\| \leq \tau \\ 0 & \text{otherwise} \end{cases}$$

Adjacency in color space

We can define a distance between colors, for example:

$$d_{ij} = \|\mathbf{c}_i - \mathbf{c}_j\|$$

This can be seen as a soft notion of adjacency

Classical "hard" (or binary) adjacency may be defined as:

$$a_{ij} = \begin{cases} 1 & \text{if } \|\mathbf{c}_i - \mathbf{c}_j\| \leq \tau \\ 0 & \text{otherwise} \end{cases}$$

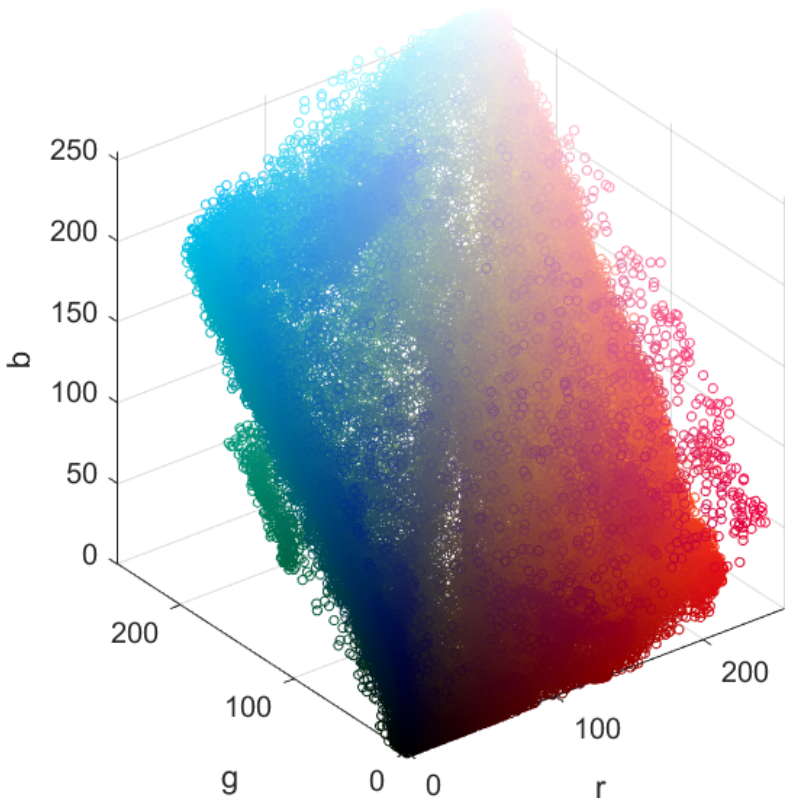
Images as point clouds

This suggests a new representation of images as point clouds in a color space (3D for RGB, 4D for CMYK, etc.)

Images as point clouds

This suggests a new representation of images as point clouds in a color space (3D for RGB, 4D for CMYK, etc.)

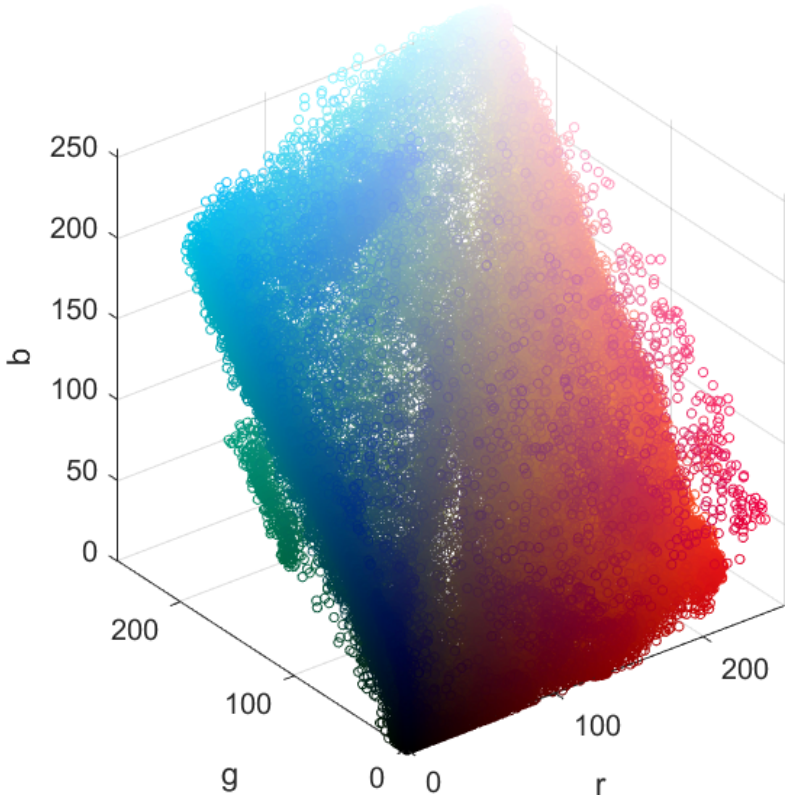
$\Rightarrow$

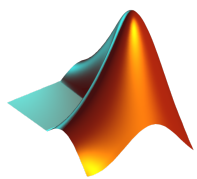


Images as point clouds

This suggests a new representation of images as point clouds in a color space (3D for RGB, 4D for CMYK, etc.)

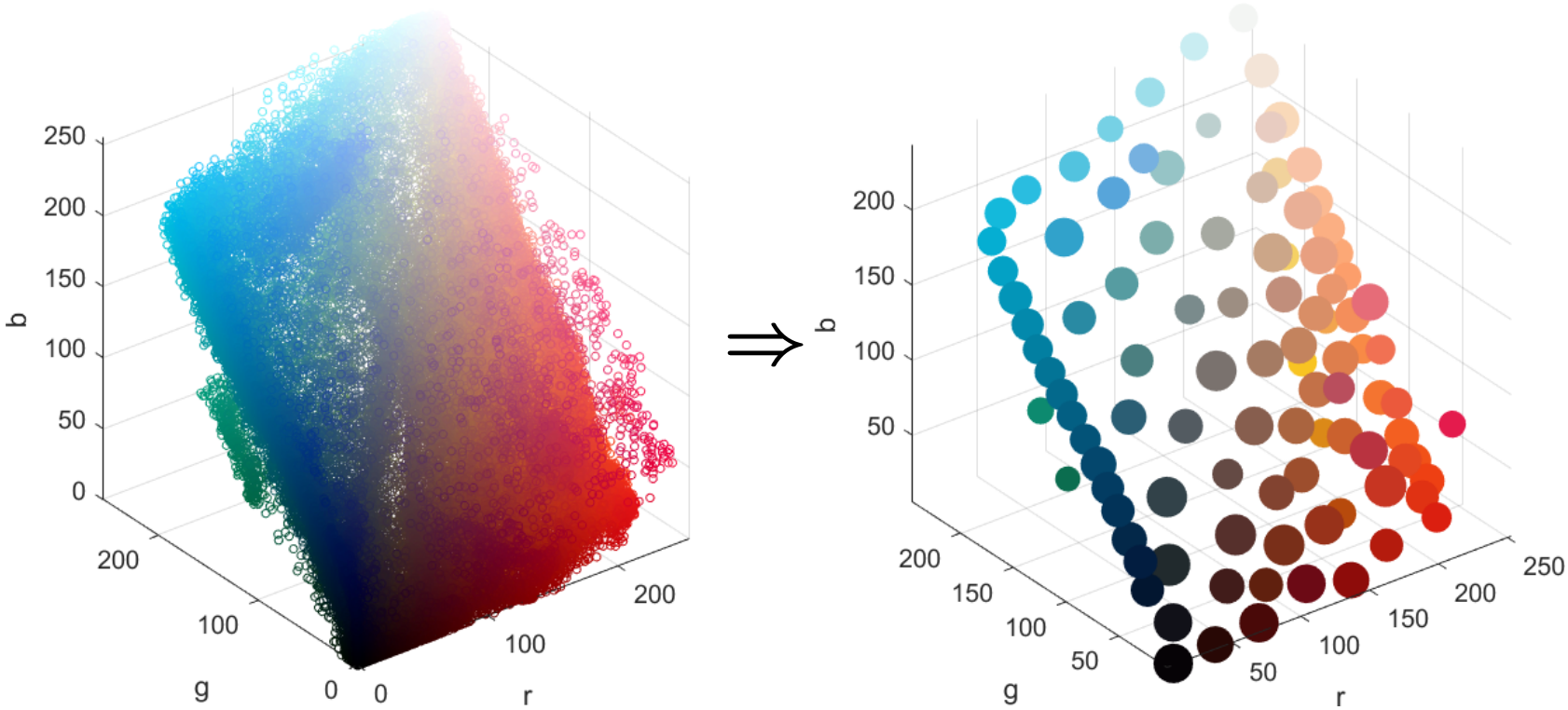
$\Rightarrow$





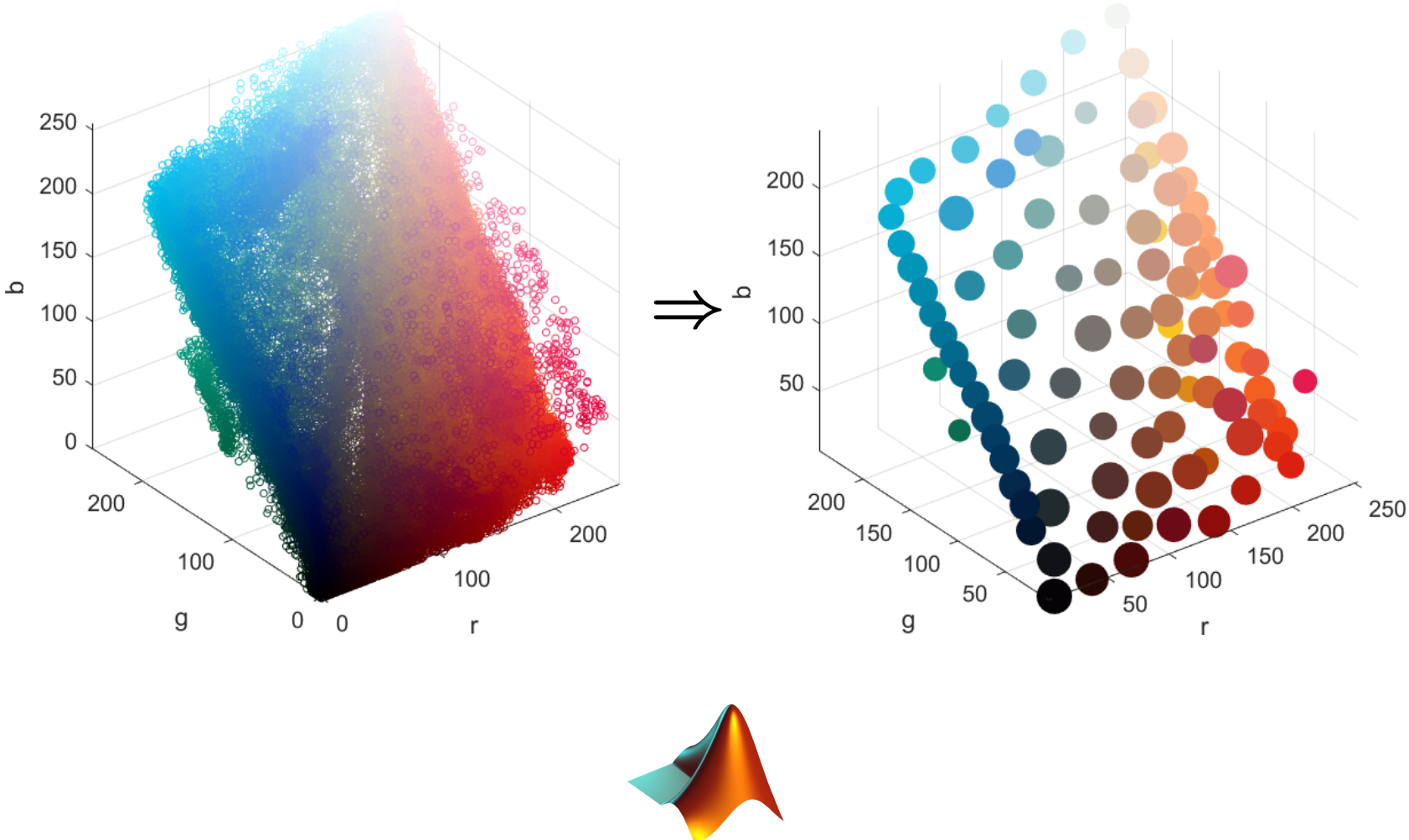
Images as k-D histograms

A more efficient and noise-reducing (although not equivalent) representation uses histograms:



Images as k-D histograms

A more efficient and noise-reducing (although not equivalent) representation uses histograms:



Adjacency matrices: Powers

The k-th power of $\mathbf{A}$ corresponds to composing $\mathbf{A}$ with itself $k \geq 1$ times

For example, for $k = 2$:

$$\mathbf{A}^2 = \mathbf{A}\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 1 \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ 1 & 0 & 1 & \cdots & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & \cdots & 1 \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ 1 & 0 & 1 & \cdots & 0 \end{pmatrix}$$

Adjacency matrices: Powers

The k-th power of $\mathbf{A}$ corresponds to composing $\mathbf{A}$ with itself $k \geq 1$ times

For example, for $k = 2$:

$$\mathbf{A}^2 = \mathbf{A}\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 1 \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ 1 & 0 & 1 & \cdots & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & \cdots & 1 \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ 1 & 0 & 1 & \cdots & 0 \end{pmatrix}$$

The result is a $n \times n$ matrix encoding 2nd order adjacency

Adjacency matrices: Powers

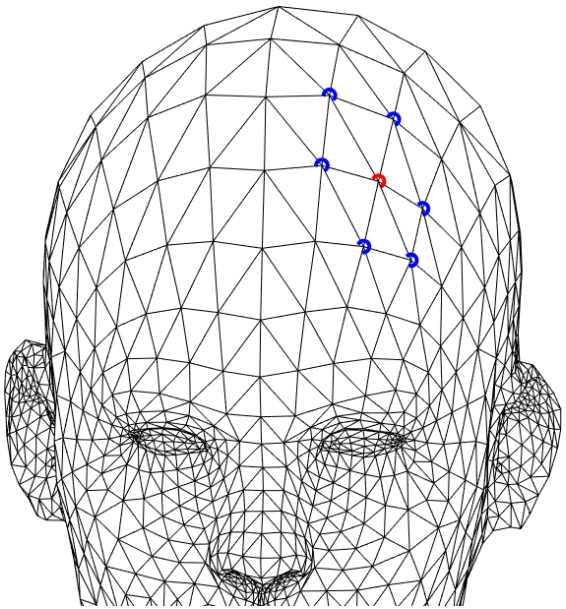
The k-th power of $\mathbf{A}$ corresponds to composing $\mathbf{A}$ with itself $k \geq 1$ times

For $k > 1$:

$$\mathbf{A}^k = \mathbf{A} \cdots \mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 1 \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ 1 & 0 & 1 & \cdots & 0 \end{pmatrix} \cdots \begin{pmatrix} 0 & 1 & 0 & \cdots & 1 \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ 1 & 0 & 1 & \cdots & 0 \end{pmatrix}$$

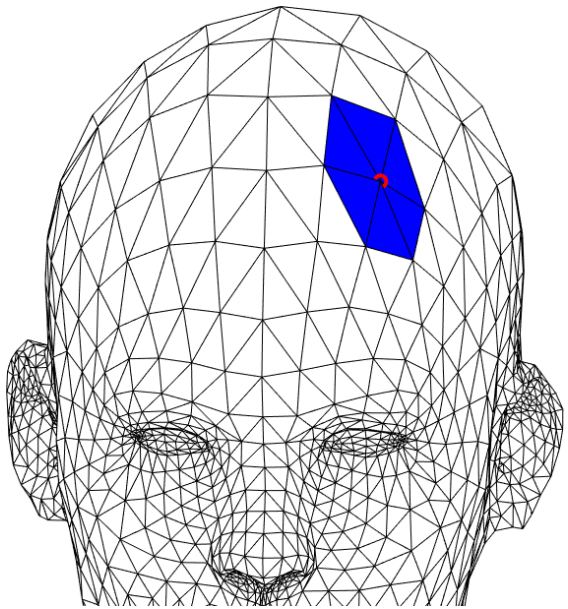
The result is a $n \times n$ matrix encoding k-th order adjacency

Examples: Powers



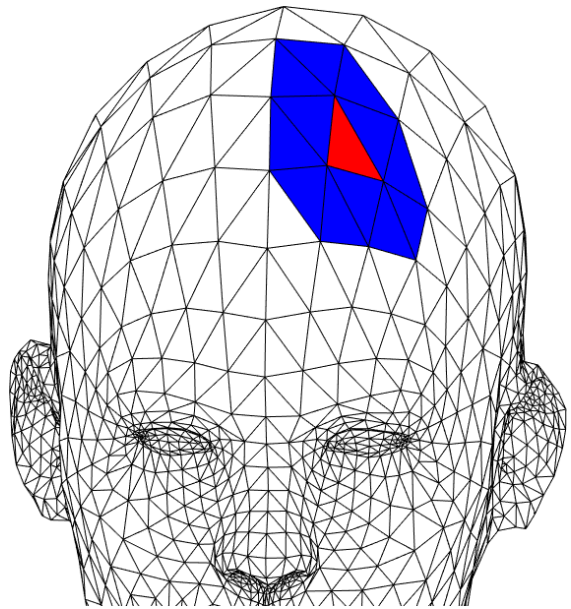
vertex-to-vertex

$k = 1$



vertex-to-triangle

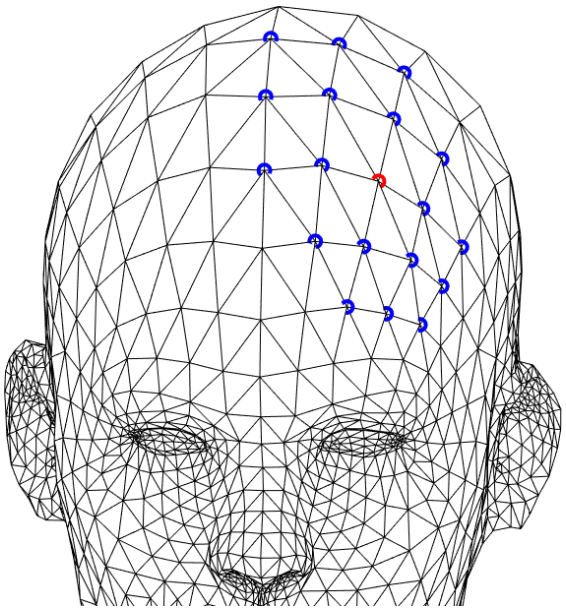
$k = 1$



triangle-to-triangle

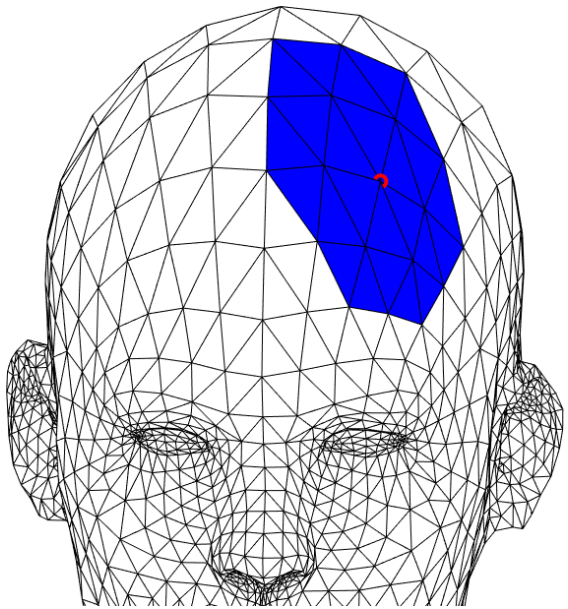
$k = 1$

Examples: Powers



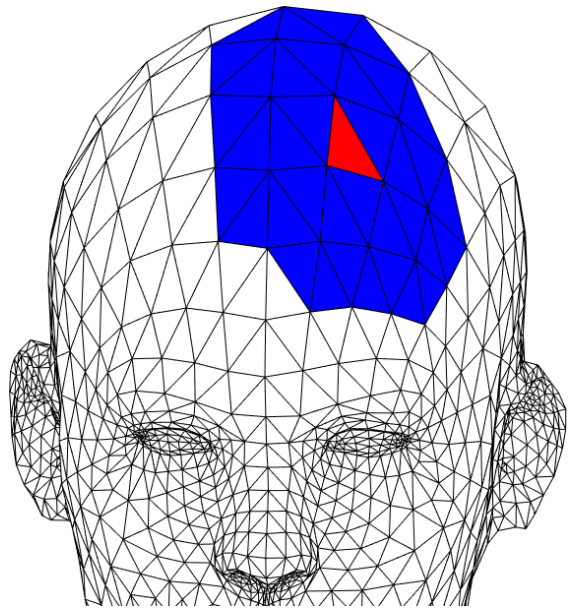
vertex-to-vertex

$k = 2$



vertex-to-triangle

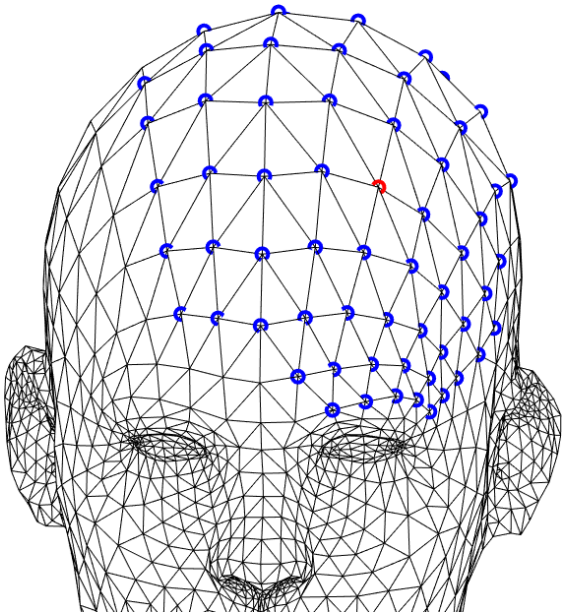
$k = 2$



triangle-to-triangle

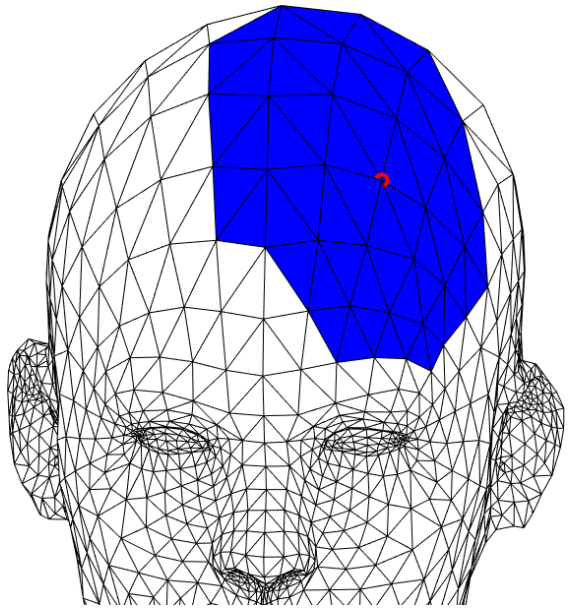
$k = 2$

Examples: Powers



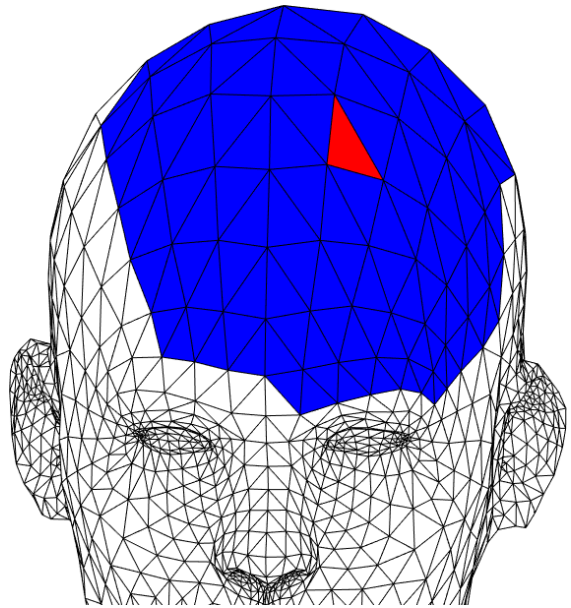
vertex-to-vertex

$k = 3$



vertex-to-triangle

$k = 3$



triangle-to-triangle

$k = 3$

Exercise: K-means clustering

Given a set of n points $X \subset \mathbb{R}^3$, implement the following algorithm:

1. Select a set of $k < n$ points $S \subset \mathbb{R}^3$ ("seed").
   Note: the selected points are not necessarily from X.
2. For each point in X, find its nearest neighbor in S.
   The result is k clusters of points.
3. For each cluster K, compute its centroid as $c = \frac{1}{|K|} \sum_i x_i$.
4. Replace S with the k centroids.
5. Repeat from 2 until converge.

Test your code by computing 3D color histograms as done during class.

Exercise: Palette reduction

Use k-means clustering to obtain p5_projected.png from p5.png
(download the two images from the course website)