

Fundamentals of Computer Graphics

Mesh processing I

Emanuele Rodolà
rodola@di.uniroma1.it



SAPIENZA
UNIVERSITÀ DI ROMA

Adjacency matrices

The mesh 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

The mesh 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

The mesh 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

The mesh 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

The mesh 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

The mesh 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

The mesh 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

The mesh 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: Vertex-vertex co-occurrence

Consider the product:

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

Adjacency matrices: Vertex-vertex co-occurrence

Consider the product:

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

\mathbf{PP}^\top is a $n \times n$ matrix with the same **zero pattern** as \mathbf{A}

Adjacency matrices: Vertex-vertex co-occurrence

Consider the product:

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

\mathbf{PP}^\top is a $n \times n$ matrix with the same **zero pattern** as \mathbf{A}

Cell (i, j) counts the number of **triangles** where v_i and v_j **co-occur**

Adjacency matrices: Vertex-vertex co-occurrence

Consider the product:

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

\mathbf{PP}^\top is a $n \times n$ matrix with the same **zero pattern** as \mathbf{A}

Cell (i, j) counts the number of **triangles** where v_i and v_j **co-occur**

- $\text{diag}(\mathbf{PP}^\top)$ contains the number of triangles for each vertex

Adjacency matrices: Vertex-vertex co-occurrence

Consider the product:

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

\mathbf{PP}^\top is a $n \times n$ matrix with the same **zero pattern** as \mathbf{A}

Cell (i, j) counts the number of **triangles** where v_i and v_j **co-occur**

- $\text{diag}(\mathbf{PP}^\top)$ contains the number of triangles for each vertex
- All other values are 0 ($e_{ij} \notin E$)

Adjacency matrices: Vertex-vertex co-occurrence

Consider the product:

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

\mathbf{PP}^\top is a $n \times n$ matrix with the same **zero pattern** as \mathbf{A}

Cell (i, j) counts the number of **triangles** where v_i and v_j **co-occur**

- $\text{diag}(\mathbf{PP}^\top)$ contains the number of triangles for each vertex
- All other values are 0 ($e_{ij} \notin E$), 1 (boundary)

Adjacency matrices: Vertex-vertex co-occurrence

Consider the product:

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

\mathbf{PP}^\top is a $n \times n$ matrix with the same **zero pattern** as \mathbf{A}

Cell (i, j) counts the number of **triangles** where v_i and v_j **co-occur**

- $\text{diag}(\mathbf{PP}^\top)$ contains the number of triangles for each vertex
- All other values are 0 ($e_{ij} \notin E$), 1 (boundary), or 2 (interior)

Adjacency matrices: Triangle-to-triangle

Consider the product:

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

which is a $m \times m$ matrix

Adjacency matrices: Triangle-to-triangle

Consider the product:

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

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:

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

which is a $m \times m$ matrix

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

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

Adjacency matrices: Triangle-to-triangle

Consider the product:

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

which is a $m \times m$ matrix

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

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

Adjacency matrices: Triangle-to-triangle

Consider the product:

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

which is a $m \times m$ matrix

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

- $\text{diag}(\mathbf{P}\mathbf{P}^\top)$ 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:

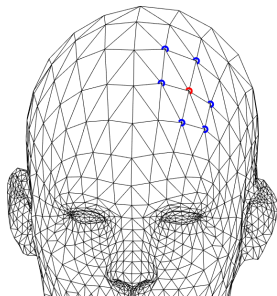
$$\mathbf{P}^\top \mathbf{P} = \begin{pmatrix} 1 & \dots & \dots & 0 \\ 0 & \dots & \dots & 1 \\ 0 & \dots & \dots & 0 \\ \vdots & \dots & \dots & \vdots \\ 0 & \dots & \dots & 1 \\ 1 & \dots & \dots & 0 \\ 1 & \dots & \dots & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 1 & 1 \\ \vdots &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

which is a $m \times m$ matrix

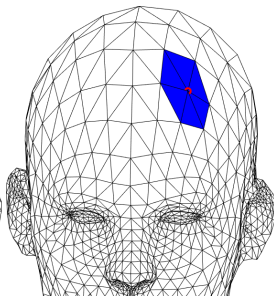
Cell (i, j) counts the number of **vertices** that t_i and t_j **share**

- $\text{diag}(\mathbf{P}\mathbf{P}^\top)$ 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)

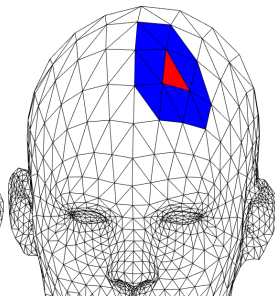
Examples: Adjacency



vertex-to-vertex

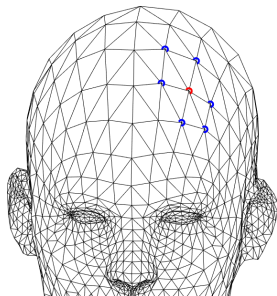


vertex-to-triangle

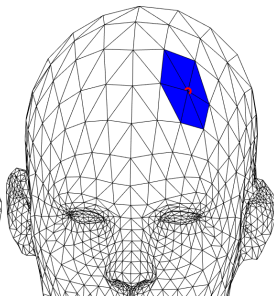


triangle-to-triangle

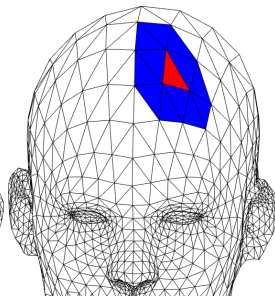
Examples: Adjacency



vertex-to-vertex



vertex-to-triangle



triangle-to-triangle

In general we have $m \approx 2n$, and with n in the order of several thousands these adjacency matrices can be **very large** (quadratic in n)

It is advisable to use **sparse** data structures to store them

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

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

- The same applies to triangle-to-triangle adjacency
- The k -th power of vertex-to-triangle is given by $\mathbf{A}^k \mathbf{P}$

Adjacency matrices: Powers

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

For $k > 1$:

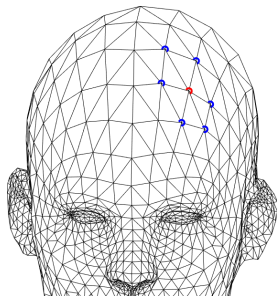
$$\mathbf{A}^k = \mathbf{A} \cdots \mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 1 \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ 1 & 0 & 1 & \cdots & 0 \end{pmatrix} \cdots \begin{pmatrix} 0 & 1 & 0 & \cdots & 1 \\ \vdots & \cdots & \cdots & \cdots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

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

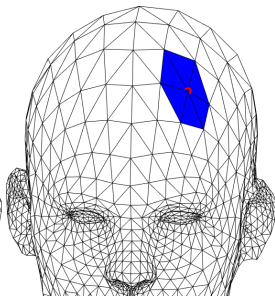
- The same applies to triangle-to-triangle adjacency
- The k -th power of vertex-to-triangle is given by $\mathbf{A}^k \mathbf{P}$

Manipulating adjacency is useful in many tasks relying upon local context

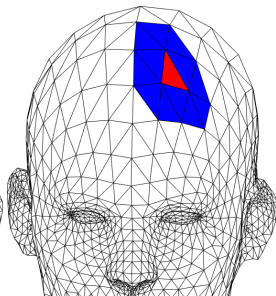
Examples: Powers



vertex-to-vertex
 $k = 1$

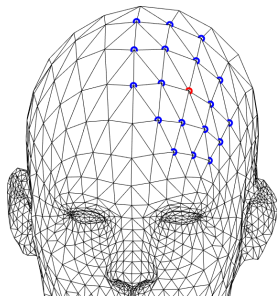


vertex-to-triangle
 $k = 1$

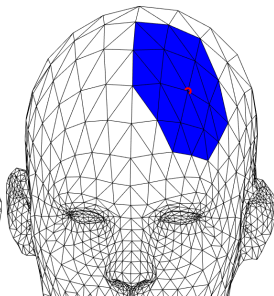


triangle-to-triangle
 $k = 1$

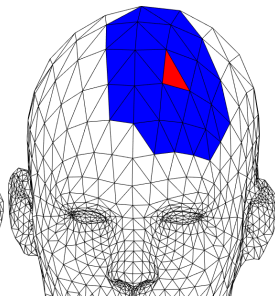
Examples: Powers



vertex-to-vertex
 $k = 2$

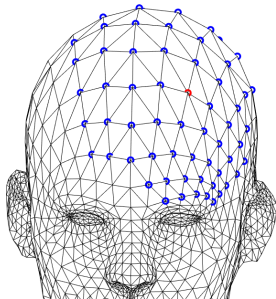


vertex-to-triangle
 $k = 2$

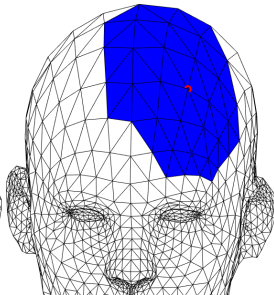


triangle-to-triangle
 $k = 2$

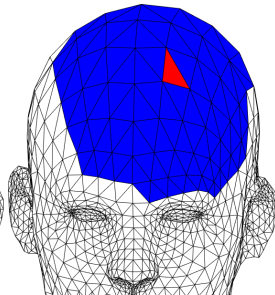
Examples: Powers



vertex-to-vertex
 $k = 3$



vertex-to-triangle
 $k = 3$



triangle-to-triangle
 $k = 3$

Adjacency matrices as operators

We can see adjacency matrices as **operators** when applied to functions

For example, $\mathbf{g} = \mathbf{A}\mathbf{f}$ yields a **vertex-based** function g defined as:

$$g(v_i) = \sum_{e_{ij} \in E} f(v_j)$$

Adjacency matrices as operators

We can see adjacency matrices as **operators** when applied to functions

For example, $\mathbf{g} = \mathbf{A}\mathbf{f}$ yields a **vertex-based** function g defined as:

$$g(v_i) = \sum_{e_{ij} \in E} f(v_j)$$

On this observation, one can construct new operators such as $\mathbf{I} - \mathbf{A}$:

$$g(v_i) = f(v_i) - \sum_{e_{ij} \in E} f(v_j)$$

Adjacency matrices as operators

We can see adjacency matrices as **operators** when applied to functions

For example, $\mathbf{g} = \mathbf{A}\mathbf{f}$ yields a **vertex-based** function g defined as:

$$g(v_i) = \sum_{e_{ij} \in E} f(v_j)$$

On this observation, one can construct new operators such as $\mathbf{I} - \mathbf{A}$:

$$g(v_i) = f(v_i) - \sum_{e_{ij} \in E} f(v_j)$$

And similarly for **triangle-based** functions

Adjacency matrices: Point clouds

Adjacency is a general notion that can be extended to **point clouds**

(Such notions of adjacency can of course be used on meshes as well)

Adjacency matrices: Point clouds

Adjacency is a general notion that can be extended to **point clouds**

For example, use **Euclidean distance** within a threshold τ :

$$a_{ij} = \begin{cases} 1 & \text{if } \|\mathbf{v}_i - \mathbf{v}_j\|_2 \leq \tau \\ 0 & \text{otherwise} \end{cases}$$



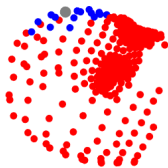
(Such notions of adjacency can of course be used on meshes as well)

Adjacency matrices: Point clouds

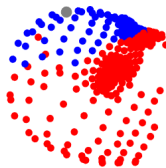
Adjacency is a general notion that can be extended to **point clouds**

For example, use **Euclidean distance** within a threshold τ :

$$a_{ij} = \begin{cases} 1 & \text{if } \|\mathbf{v}_i - \mathbf{v}_j\|_2 \leq \tau \\ 0 & \text{otherwise} \end{cases}$$



$k = 1$

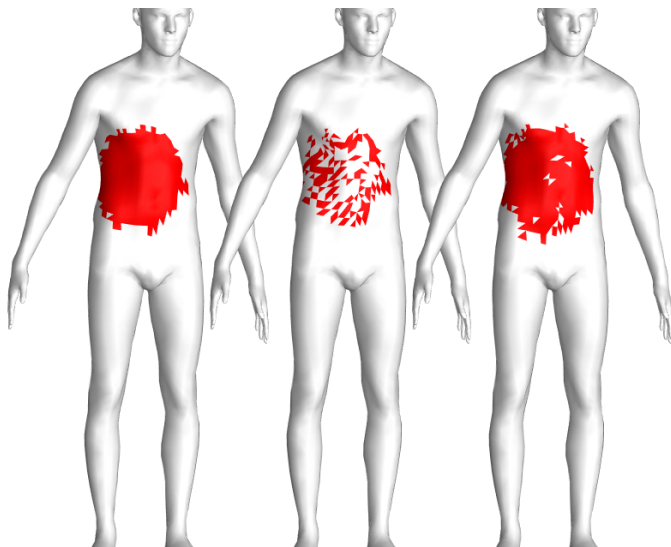


$k = 2$

Similarly to before, \mathbf{A}^k encodes k -th order adjacency

(Such notions of adjacency can of course be used on meshes as well)

Example: Hole filling

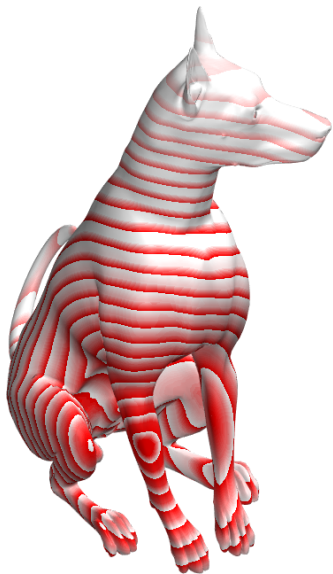


original
 \mathbf{f}

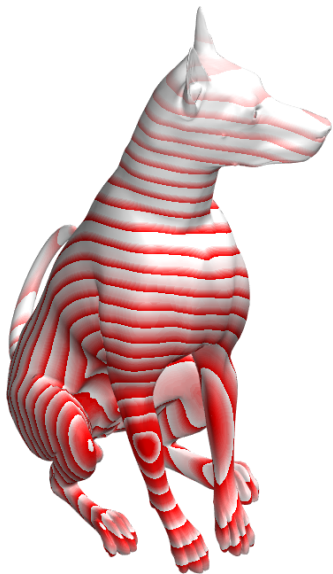
noisy
 $\tilde{\mathbf{f}}$

denoised
 $\mathbf{A}\tilde{\mathbf{f}}$

Shortest paths



Shortest paths

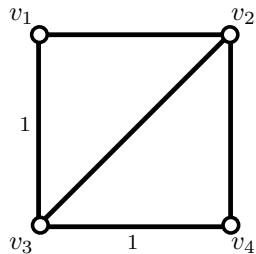


Euclidean

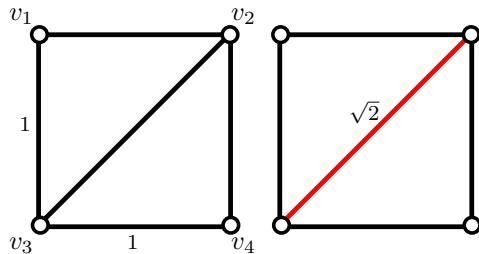


Geodesic

Shortest paths on a graph

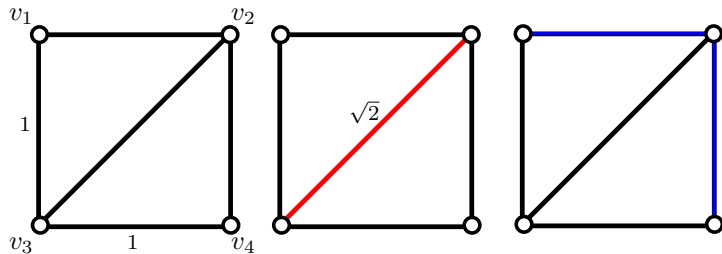


Shortest paths on a graph



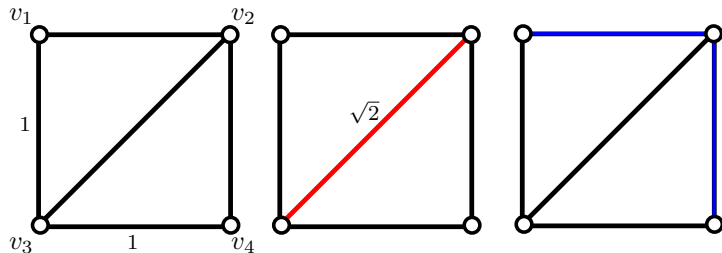
$$\sqrt{2} = d(v_2, v_3)$$

Shortest paths on a graph



$$\sqrt{2} = d(v_2, v_3) \neq d(v_1, v_4) = 2$$

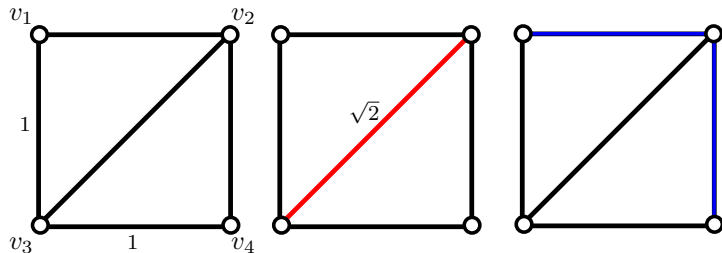
Shortest paths on a graph



$$\sqrt{2} = d(v_2, v_3) \neq d(v_1, v_4) = 2$$

Shortest paths along edges provide **upper bounds** to exact geodesics

Shortest paths on a graph

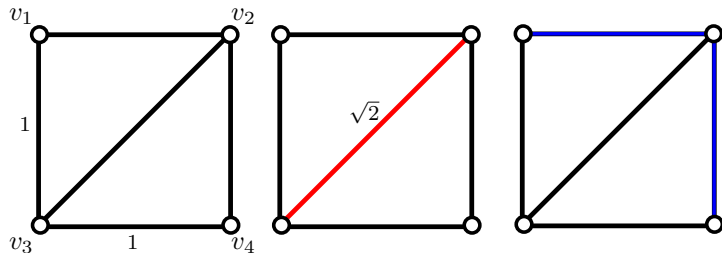


$$\sqrt{2} = d(v_2, v_3) \neq d(v_1, v_4) = 2$$

Shortest paths along edges provide **upper bounds** to exact geodesics

- Still useful with **high resolution** meshes or for **local** distances

Shortest paths on a graph



$$\sqrt{2} = d(v_2, v_3) \neq d(v_1, v_4) = 2$$

Shortest paths along edges provide **upper bounds** to exact geodesics

- Still useful with **high resolution** meshes or for **local** distances
- Solved by **Dijkstra's algorithm** on the mesh graph

Graph Laplacian

Given a mesh graph $G = (V, E)$, consider this condition on vertex v_i :

$$\mathbf{v}_i - \frac{1}{d_i} \sum_{j:(i,j) \in E} \mathbf{v}_j = 0$$

where d_i is the valence of v_i

Graph Laplacian

Given a **mesh graph** $G = (V, E)$, consider this condition on vertex v_i :

$$\mathbf{v}_i - \frac{1}{d_i} \sum_{j:(i,j) \in E} \mathbf{v}_j = 0$$

where d_i is the valence of v_i

The equation is satisfied if v_i lies in the **barycenter** of its neighbors

Graph Laplacian

Given a **mesh graph** $G = (V, E)$, consider this condition on vertex v_i :

$$\mathbf{v}_i - \frac{1}{d_i} \sum_{j:(i,j) \in E} \mathbf{v}_j = 0$$

where d_i is the valence of v_i

The equation is satisfied if v_i lies in the **barycenter** of its neighbors

In matrix notation, we define the $n \times n$ matrix \mathbf{L} as:

$$L_{ij} = \begin{cases} 1 & \text{if } i = j \\ -\frac{1}{d_i} & \text{if } e_{ij} \in E \\ 0 & \text{otherwise} \end{cases}$$

also known as the **graph Laplacian** of G

Graph Laplacian

Given a **mesh graph** $G = (V, E)$, consider this condition on vertex v_i :

$$\mathbf{v}_i - \frac{1}{d_i} \sum_{j:(i,j) \in E} \mathbf{v}_j = 0$$

where d_i is the valence of v_i

The equation is satisfied if v_i lies in the **barycenter** of its neighbors

In matrix notation, we define the $n \times n$ matrix \mathbf{L} as:

$$L_{ij} = \begin{cases} 1 & \text{if } i = j \\ -\frac{1}{d_i} & \text{if } e_{ij} \in E \\ 0 & \text{otherwise} \end{cases}$$

also known as the **graph Laplacian** of G ; we get to:

$$\mathbf{L}\mathbf{V} = \mathbf{0}$$

Least squares meshes

Is there an embedding satisfying $\mathbf{L}\mathbf{V} = \mathbf{0}$?

- Trivial embedding $\mathbf{V} = \mathbf{0}$

Least squares meshes

Is there an embedding satisfying $\mathbf{L}\mathbf{V} = \mathbf{0}$?

- Trivial embedding $\mathbf{V} = \mathbf{0}$
- One-dimensional subspace $\text{span}(\mathbf{1})$

Least squares meshes

Is there an **interesting** embedding satisfying $\mathbf{L}\mathbf{V} = \mathbf{0}$?

- Trivial embedding $\mathbf{V} = \mathbf{0}$
- One-dimensional subspace $\text{span}(\mathbf{1})$

Least squares meshes

Is there an **interesting** embedding satisfying $\mathbf{L}\mathbf{V} = \mathbf{0}$?

- Trivial embedding $\mathbf{V} = \mathbf{0}$
- One-dimensional subspace $\text{span}(\mathbf{1})$

Assume $m \geq 1$ **anchor vertices** $v_s \in \mathcal{A}$ with known 3D position

Least squares meshes

Is there an **interesting** embedding satisfying $\mathbf{L}\mathbf{V} = \mathbf{0}$?

- Trivial embedding $\mathbf{V} = \mathbf{0}$
- One-dimensional subspace $\text{span}(\mathbf{1})$

Assume $m \geq 1$ **anchor vertices** $v_s \in \mathcal{A}$ with known 3D position

Then, consider the linear system:

$$\begin{pmatrix} \mathbf{L} \\ \mathbf{A} \end{pmatrix} \mathbf{V} = \mathbf{b}$$

where

$$a_{ij} = \begin{cases} 1 & \text{if } v_j \in \mathcal{A} \\ 0 & \text{otherwise} \end{cases}, \quad b_k = \begin{cases} (0, 0, 0) & k \leq n \\ \mathbf{v}_{s_{k-n}} & n < k \leq n + m \end{cases}$$

Least squares meshes

$$\begin{pmatrix} \mathbf{L} \\ \mathbf{A} \end{pmatrix} \mathbf{v} \approx \mathbf{b}$$

Least squares meshes

$$\min_{\mathbf{v} \in \mathbb{R}^{n \times 3}} \left\| \begin{pmatrix} \mathbf{L} \\ \mathbf{A} \end{pmatrix} \mathbf{v} - \mathbf{b} \right\|_2^2$$

Least squares meshes

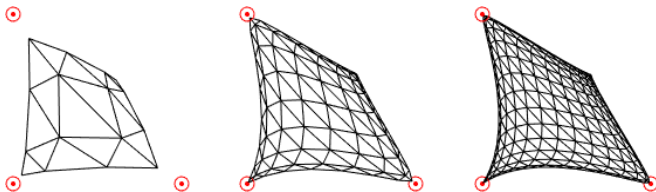
$$\min_{\mathbf{V} \in \mathbb{R}^{n \times 3}} \|\mathbf{LV}\|_2^2 + \sum_{i=1}^{n+m} \|\mathbf{v}_i - \mathbf{b}_i\|_2^2$$

Least squares meshes

$$\min_{\mathbf{V} \in \mathbb{R}^{n \times 3}} \|\mathbf{L}\mathbf{V}\|_2^2 + \sum_{v_i \notin \mathcal{A}} \|\mathbf{v}_i\|_2^2 + \sum_{v_i \in \mathcal{A}} \|\mathbf{v}_i - \mathbf{b}_i\|_2^2$$

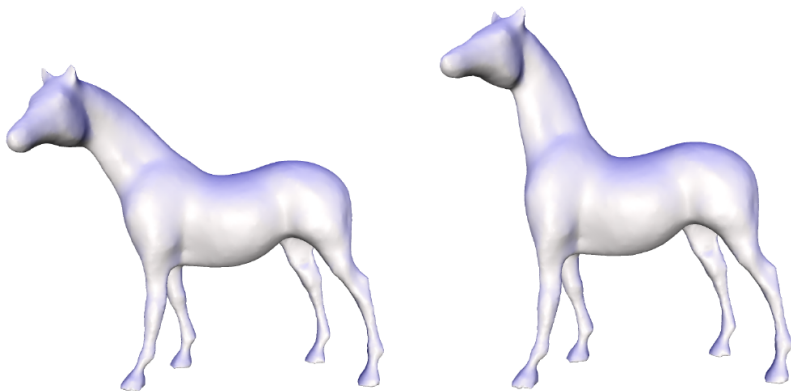
Least squares meshes

$$\min_{\mathbf{V} \in \mathbb{R}^{n \times 3}} \|\mathbf{L}\mathbf{V}\|_2^2 + \sum_{v_i \notin \mathcal{A}} \|\mathbf{v}_i\|_2^2 + \sum_{v_i \in \mathcal{A}} \|\mathbf{v}_i - \mathbf{b}_i\|_2^2$$



- Anchor constraints are not satisfied exactly
- At higher resolution, error distributes better among the constraints

Least squares meshes



Move the anchor positions to do **shape modeling**

Sorkine and Cohen-Or, "Least-squares meshes". Proc. SMI, 2004

Exercise: Least squares meshes

Implement the example in Figure 4 from:

Sorkine and Cohen-Or, “Least-squares meshes”. Proc. SMI, 2004