

Scientific Visualization

Spring 2018

Center for Data Science

New York University

Claudio Silva

Gustavo Nonato

Lecture 2: Data Types and Grids

Data Model vs Conceptual Model

Data: information that can be represented in computer

Data model: mathematical abstraction (data abstraction)

Math: Sets with operations on them

Example: integers with + and \times operators

Conceptual models: are mental/semantical constructions

Temperature, Image

Examples of data vs. conceptual model

- *Float numbers vs. Temperature*

- *3D vectors vs. Space*

Data Types and Structures

Data Types: fundamental units

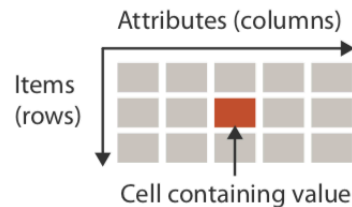
Item, Link, Attribute, Position, Grid

Data Structures: combinations of data types

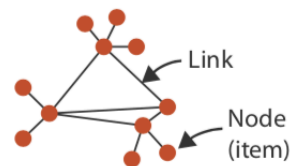
tables, networks, grids, etc

Dataset Types

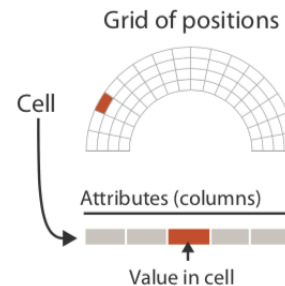
→ Tables



→ Networks



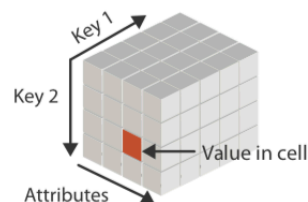
→ Fields (Continuous)



→ Geometry (Spatial)



→ Multidimensional Table



→ Trees

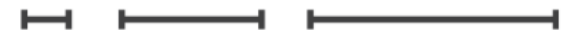


Attribute Types

Quantitative (Q)

- numbers, range of values, etc.

→ Quantitative



Ordinal (ordered) (O)

- small, medium, large

→ Ordered

→ Ordinal



Nominal (categorical) (N)

- apples, oranges, bananas

→ Categorical



Data Classification

Two aspects to consider:

- dimension of the data
- dimension of the domain

Data Classification

Two aspects to consider:

- dimension of the data
- dimension of the domain

Examples:

Set of points with scalar values

- which are the data and the domain?

Data Classification

Two aspects to consider:

- dimension of the data
- dimension of the domain

Examples:

Set of points with scalar values

- which are the data and the domain?

domain: points (dimension 0)

data: scalar (dimension 1)

Data Classification

Two aspects to consider:

- dimension of the data
- dimension of the domain

Examples:

Set of points with scalar values

- which are the data and the domain?

domain: points (dimension 0)

data: scalar (dimension 1)

3D arrays in 2D regular grid

- which are the data and the domain?

Data Classification

Two aspects to consider:

- dimension of the data
- dimension of the domain

Examples:

Set of points with scalar values

- which are the data and the domain?

domain: points (dimension 0)

data: scalar (dimension 1)

3D arrays in 2D regular grid

- which are the data and the domain?

domain: 2D grid (dimension 2)

data: 3D arrays (dimension 3)

Data Classification

Two aspects to consider:

- dimension of the data
- dimension of the domain

Examples:

Set of points with scalar values

- which are the data and the domain?
domain: points (dimension 0)
data: scalar (dimension 1)

3D arrays in 2D regular grid

- which are the data and the domain?
domain: 2D grid (dimension 2)
data: 3D arrays (dimension 3)

3D volume with 2D arrays

- which are the data and the domain?

Data Classification

Two aspects to consider:

- dimension of the data
- dimension of the domain

Examples:

Set of points with scalar values

- which are the data and the domain?

domain: points (dimension 0)

data: scalar (dimension 1)

3D arrays in 2D regular grid

- which are the data and the domain?

domain: 2D grid (dimension 2)

data: 3D arrays (dimension 3)

3D volume with 2D arrays

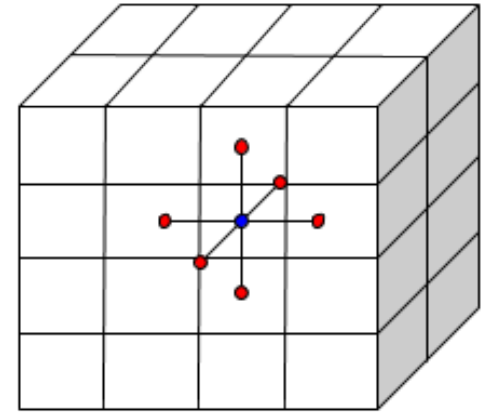
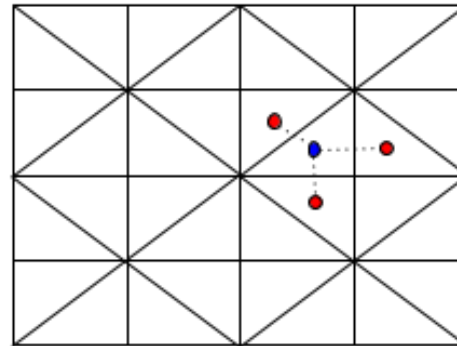
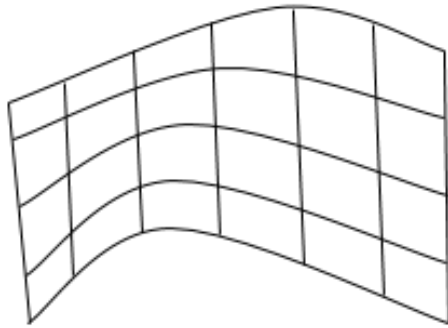
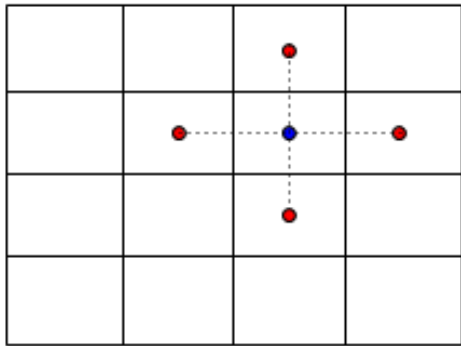
- which are the data and the domain?

domain: 3D volume (dimension 3)

data: 2D arrays (dimension 2)

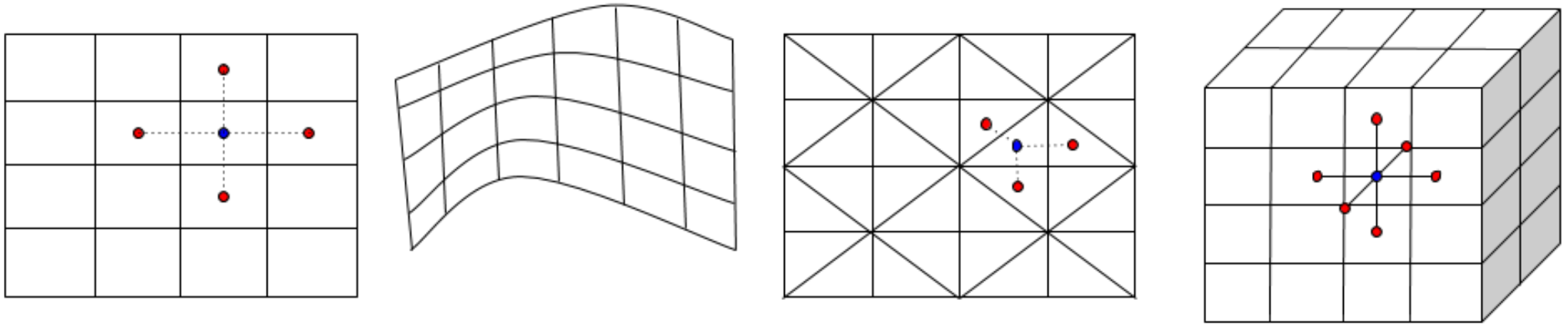
Grids

Structured Grid

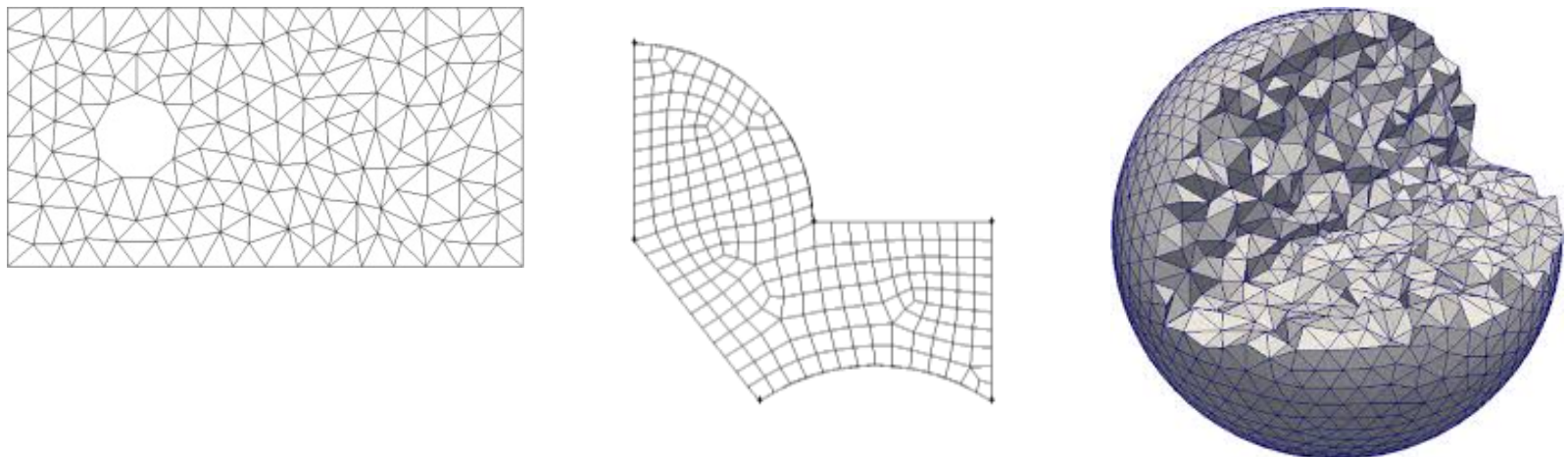


Grids

Structured Grid

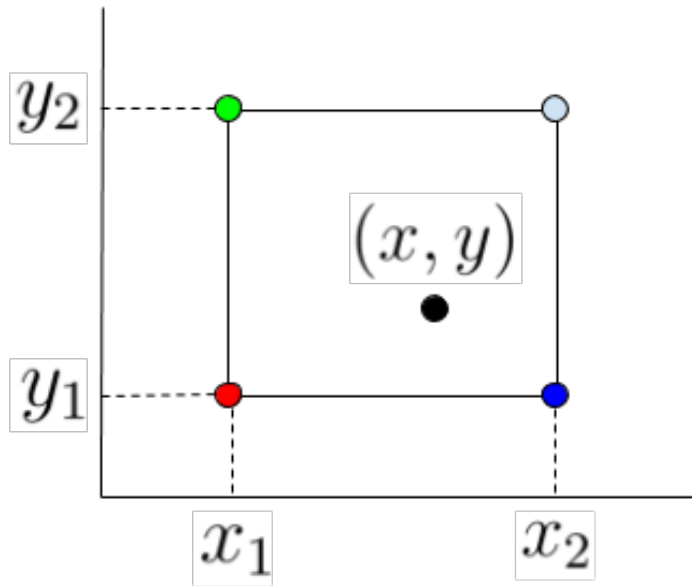


Unstructured Grid



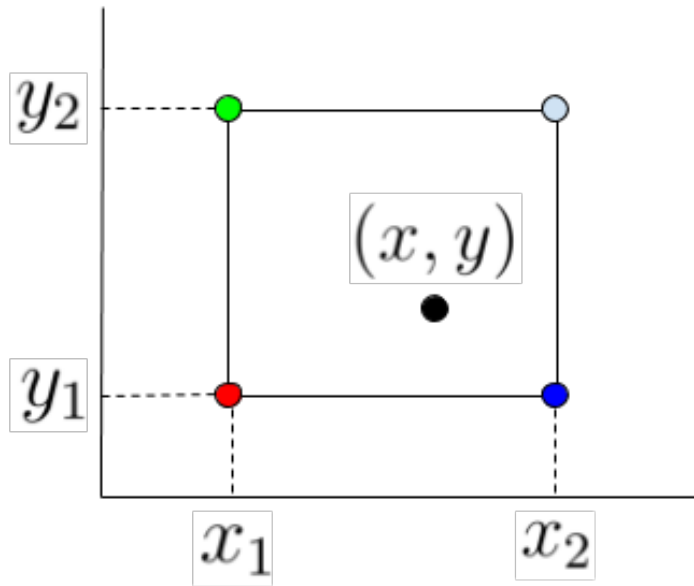
Interpolation

Bilinear Interpolation



Interpolation

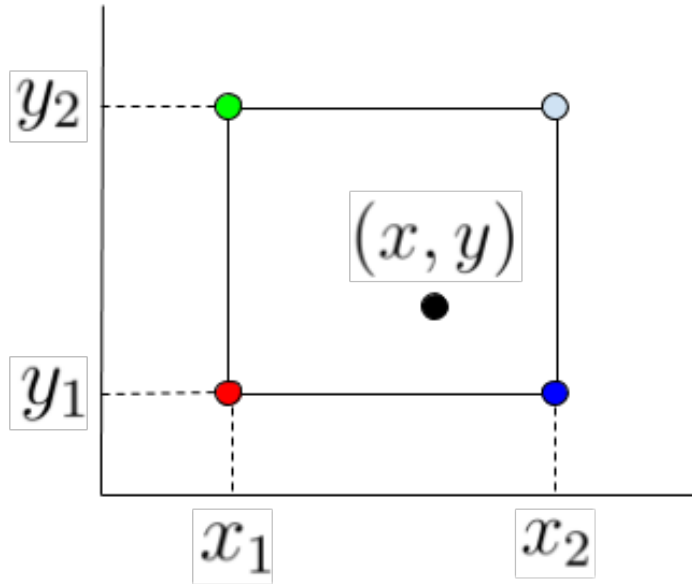
Bilinear Interpolation



$$f(x, y) = \frac{1}{(x_2 - x_1)(y_2 - y_1)} \begin{bmatrix} x_2 - x & x - x_1 \end{bmatrix} \begin{bmatrix} f(x_1, y_1) & f(x_1, y_2) \\ f(x_2, y_1) & f(x_2, y_2) \end{bmatrix} \begin{bmatrix} y_2 - y \\ y - y_1 \end{bmatrix}$$

Interpolation

Bilinear Interpolation



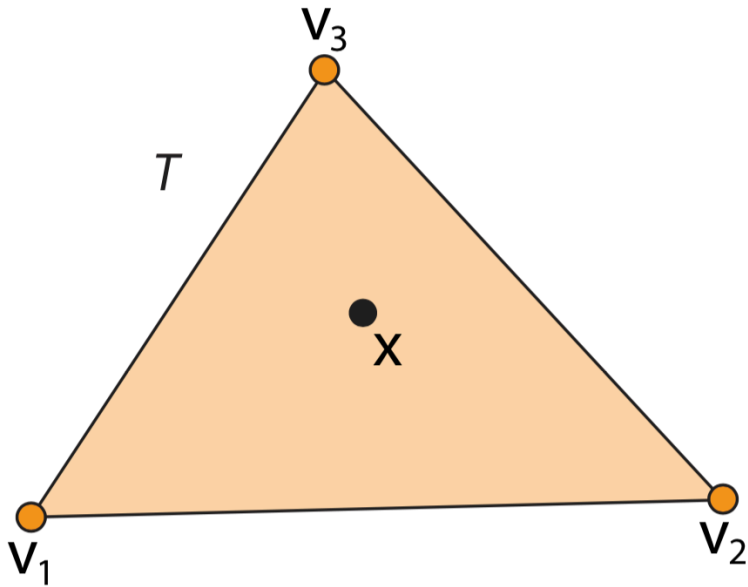
$$\begin{bmatrix} 1 & x_1 & y_1 & x_1 y_1 \\ 1 & x_1 & y_2 & x_1 y_2 \\ 1 & x_2 & y_1 & x_2 y_1 \\ 1 & x_2 & y_2 & x_2 y_2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} f(x_1, y_1) \\ f(x_1, y_2) \\ f(x_2, y_1) \\ f(x_2, y_2) \end{bmatrix}$$

$$f(x, y) = a_0 + a_1 x + a_2 y + a_3 xy$$

$$f(x, y) = \frac{1}{(x_2 - x_1)(y_2 - y_1)} \begin{bmatrix} x_2 - x & x - x_1 \end{bmatrix} \begin{bmatrix} f(x_1, y_1) & f(x_1, y_2) \\ f(x_2, y_1) & f(x_2, y_2) \end{bmatrix} \begin{bmatrix} y_2 - y \\ y - y_1 \end{bmatrix}$$

Interpolation

Barycentric Coordinates

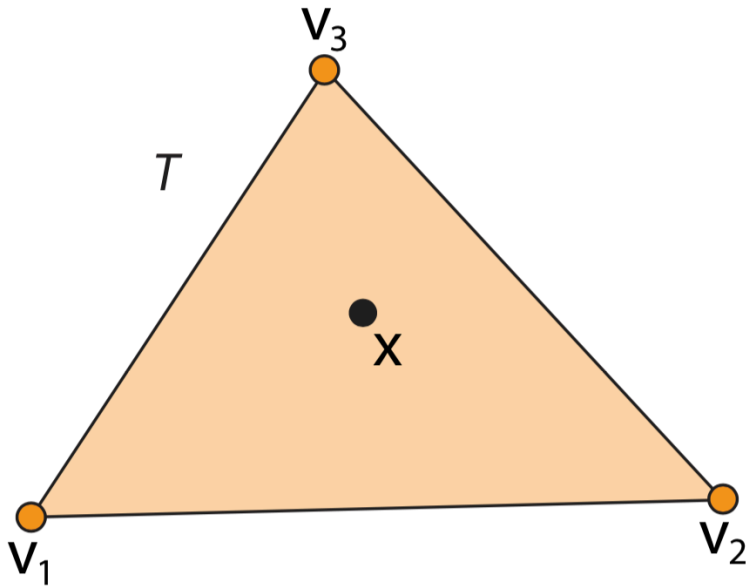


$$\mathbf{x} = \lambda_1 \mathbf{v}_1 + \lambda_2 \mathbf{v}_2 + \lambda_3 \mathbf{v}_3$$

$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$

Interpolation

Barycentric Coordinates



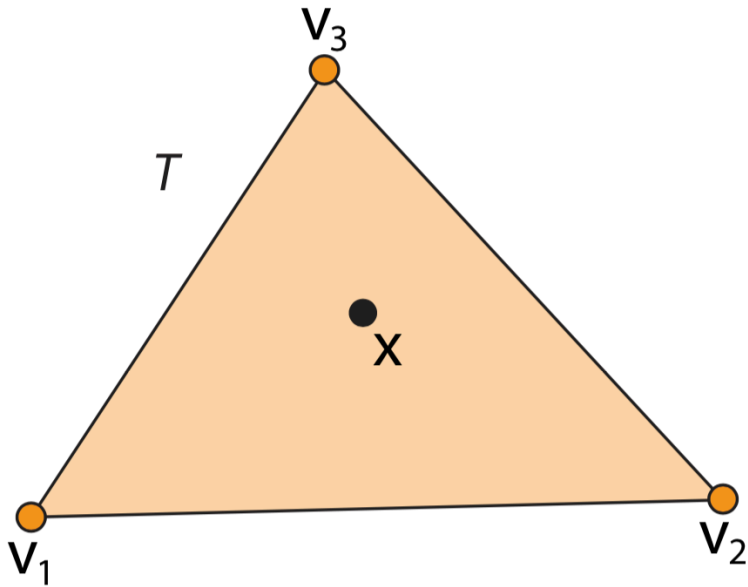
$$\mathbf{x} = \lambda_1 \mathbf{v}_1 + \lambda_2 \mathbf{v}_2 + \lambda_3 \mathbf{v}_3$$

$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$

How to compute the "lambdas"?

Interpolation

Barycentric Coordinates



$$\mathbf{x} = \lambda_1 \mathbf{v}_1 + \lambda_2 \mathbf{v}_2 + \lambda_3 \mathbf{v}_3$$

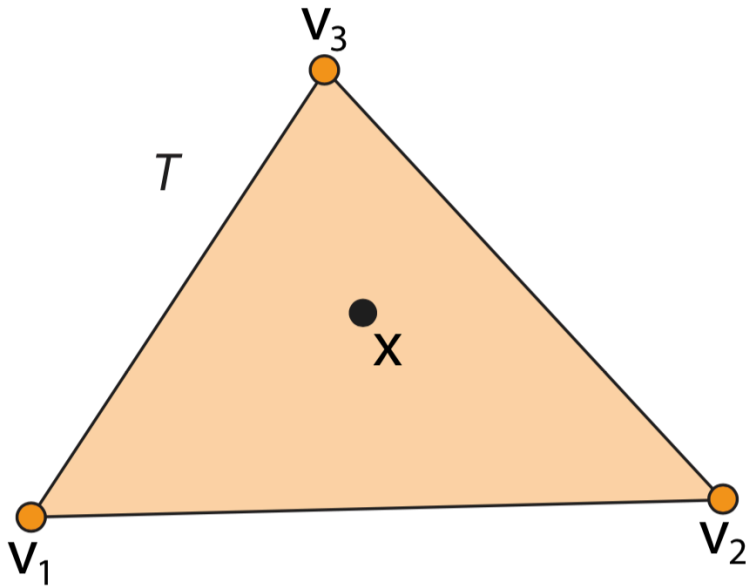
$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$

How to compute the "lambdas"?

$$\begin{bmatrix} 1 & 1 & 1 \\ v_1^1 & v_1^2 & v_1^3 \\ v_2^1 & v_2^2 & v_2^3 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}$$

Interpolation

Barycentric Coordinates

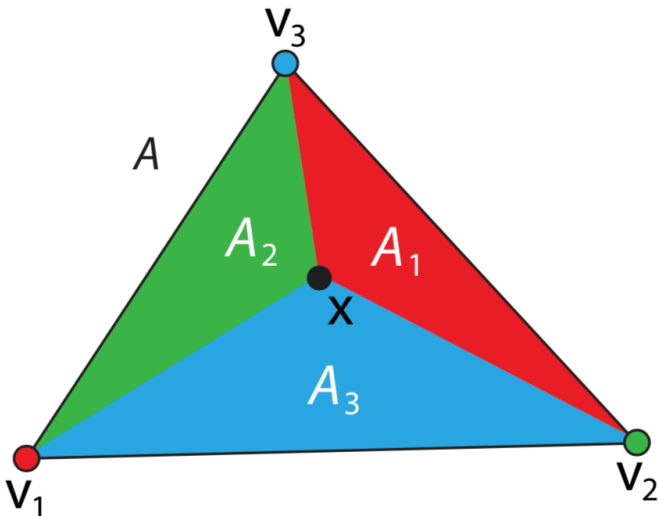


$$\mathbf{x} = \lambda_1 \mathbf{v}_1 + \lambda_2 \mathbf{v}_2 + \lambda_3 \mathbf{v}_3$$

$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$

How to compute the "lambdas"?

$$\begin{bmatrix} 1 & 1 & 1 \\ v_1^1 & v_1^2 & v_1^3 \\ v_2^1 & v_2^2 & v_2^3 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}$$



$$\lambda_1 = \frac{A_1}{A}, \quad \lambda_2 = \frac{A_2}{A}, \quad \lambda_3 = \frac{A_3}{A}$$

Interpolation

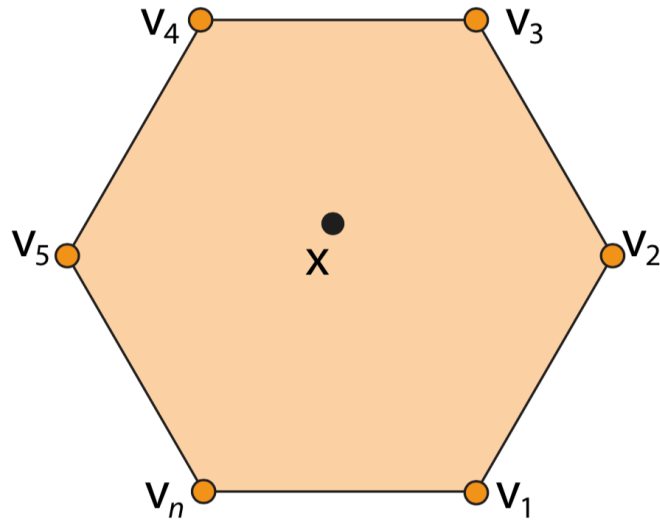
Barycentric Coordinates

If values are known in the vertices of a triangle, those values can be interpolated in the interior of the triangle using barycentric coordinates.

The same construction can be defined for higher dimensional simplices as tetrahedra in 3D.

Interpolation

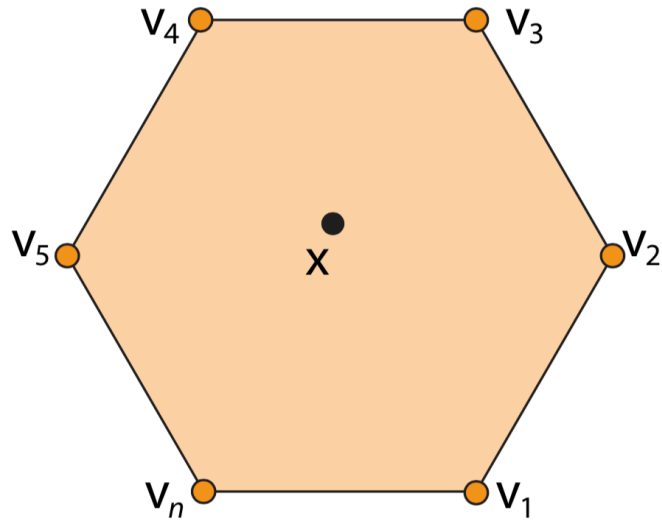
Mean Value Coordinates



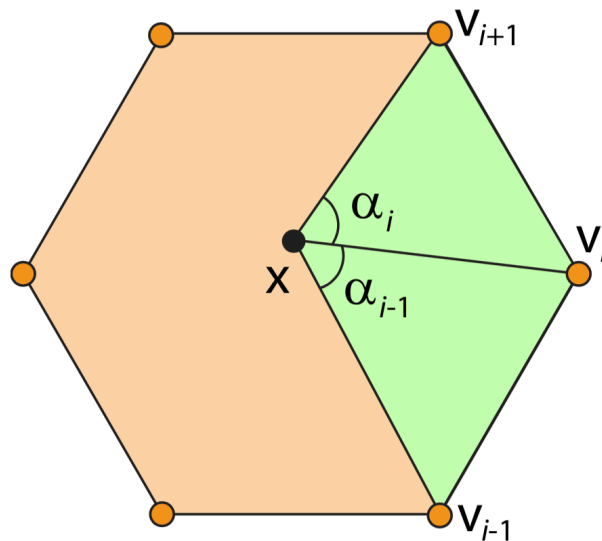
$$\mathbf{x} = \sum_{i=1}^n \lambda_i(\mathbf{x}) \mathbf{v}_i \quad \sum_{i=1}^n \lambda_i(\mathbf{x}) = 1$$

Interpolation

Mean Value Coordinates



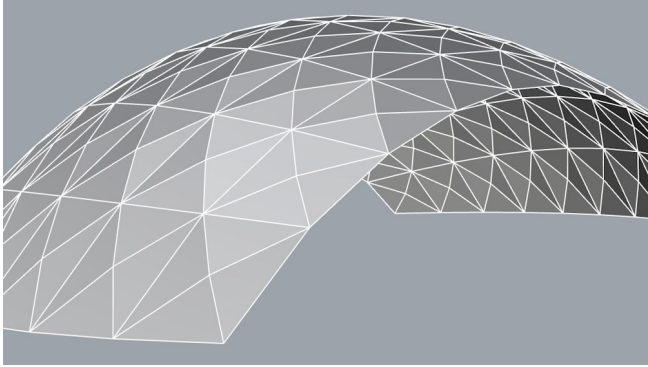
$$\mathbf{x} = \sum_{i=1}^n \lambda_i(\mathbf{x}) \mathbf{v}_i \quad \sum_{i=1}^n \lambda_i(\mathbf{x}) = 1$$



$$w_i(\mathbf{x}) = \frac{1}{\|\mathbf{v}_i - \mathbf{x}\|} \left[\tan \left(\frac{\alpha_{i-1}(\mathbf{x})}{2} \right) + \tan \left(\frac{\alpha_i(\mathbf{x})}{2} \right) \right]$$

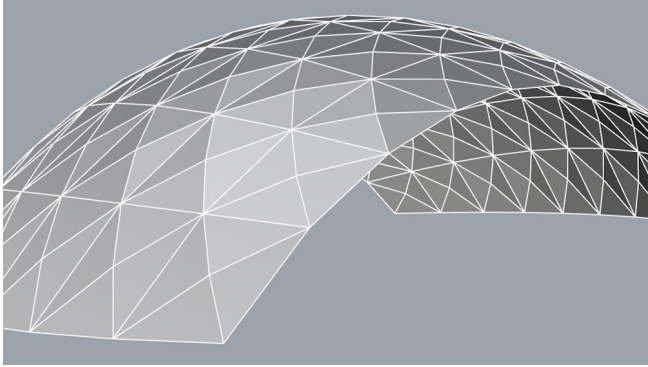
$$\lambda_i(\mathbf{x}) = w_i(\mathbf{x}) / \sum_{j=1}^n w_j(\mathbf{x})$$

Triangle Mesh Data Structure



What to store in the data structure?

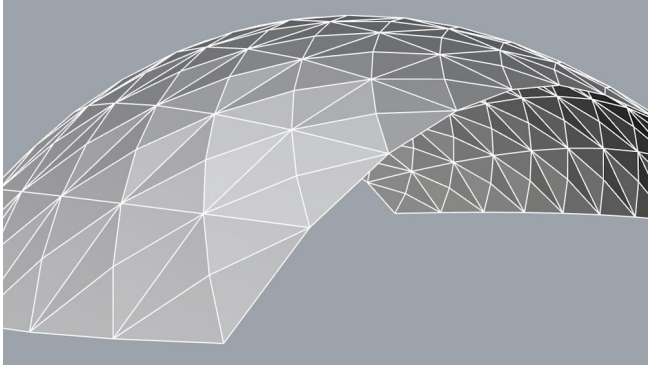
Triangle Mesh Data Structure



What to store in the data structure?

Geometry:

Triangle Mesh Data Structure

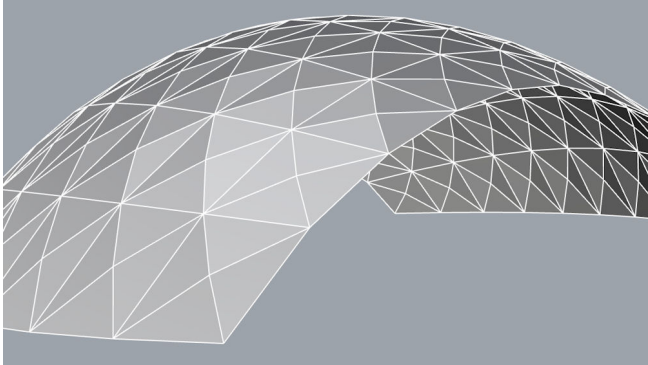


What to store in the data structure?

Geometry:

- Vertex coordinates
- Triangle normals
- Vertex and triangle attributes

Triangle Mesh Data Structure



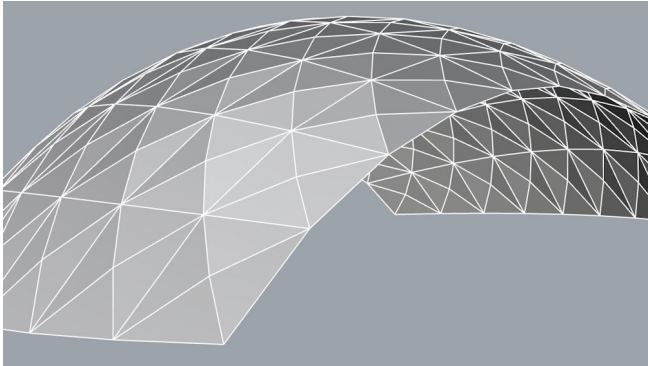
What to store in the data structure?

Geometry:

- Vertex coordinates
- Triangle normals
- Vertex and triangle attributes

Topology:

Triangle Mesh Data Structure



What to store in the data structure?

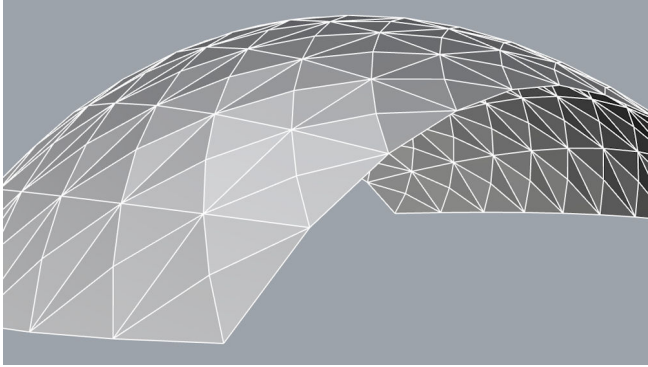
Geometry:

- Vertex coordinates
- Triangle normals
- Vertex and triangle attributes

Topology:

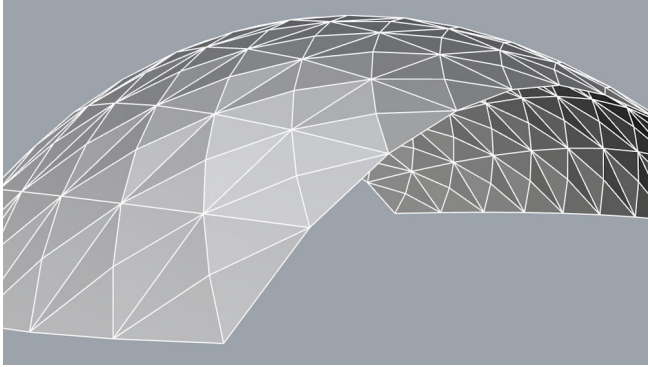
- adjacency relation among elements

Triangle Mesh Data Structure



Which operations should it support?

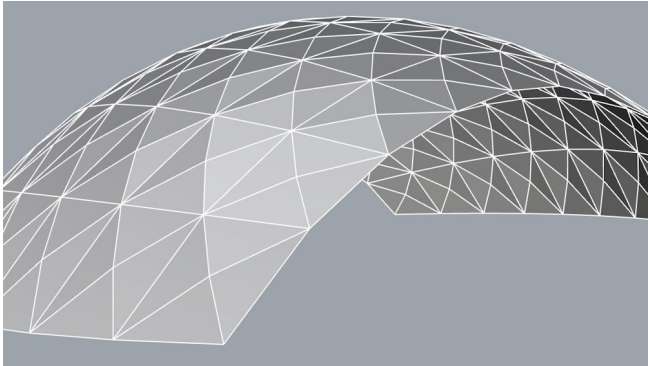
Triangle Mesh Data Structure



Which operations should it support?

Queries:

Triangle Mesh Data Structure



Which operations should it support?

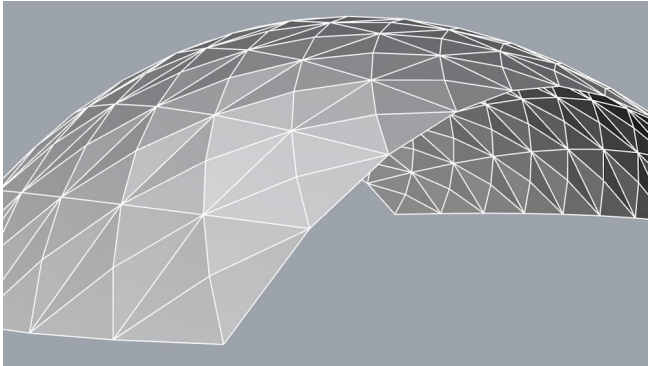
Queries:

- Which are the vertices of a given face?
- Which are the vertices adjacent to a given vertex?
- Which are the faces a vertex belong to?

Structural changes:

- edge flip
- mesh refinement

Triangle Mesh Data Structure

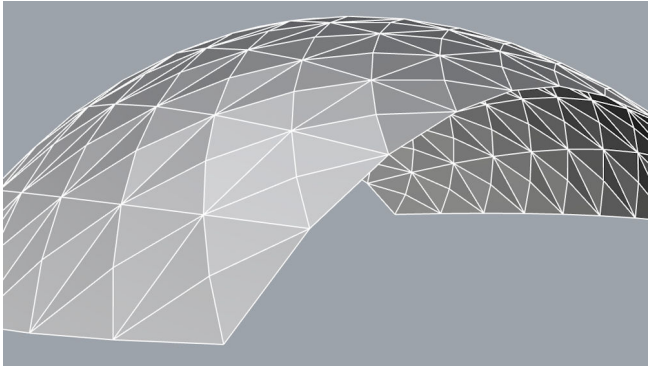


Which operations should it support?

Queries:

- Which are the vertices of a given face?
- Which are the vertices adjacent to a given vertex?
- Which are the faces a vertex belong to?

Triangle Mesh Data Structure



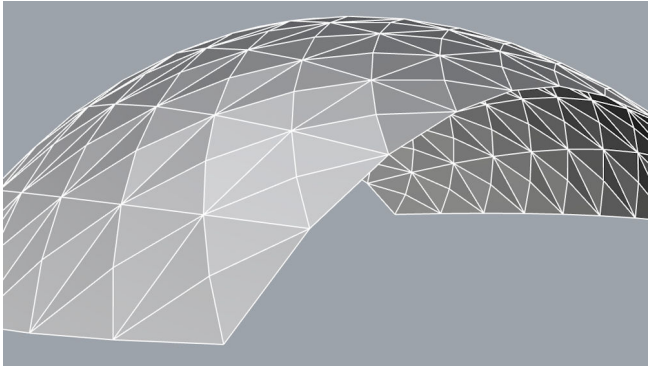
Which operations should it support?

Queries:

- Which are the vertices of a given face?
- Which are the vertices adjacent to a given vertex?
- Which are the faces a vertex belong to?

Structural changes:

Triangle Mesh Data Structure



Which operations should it support?

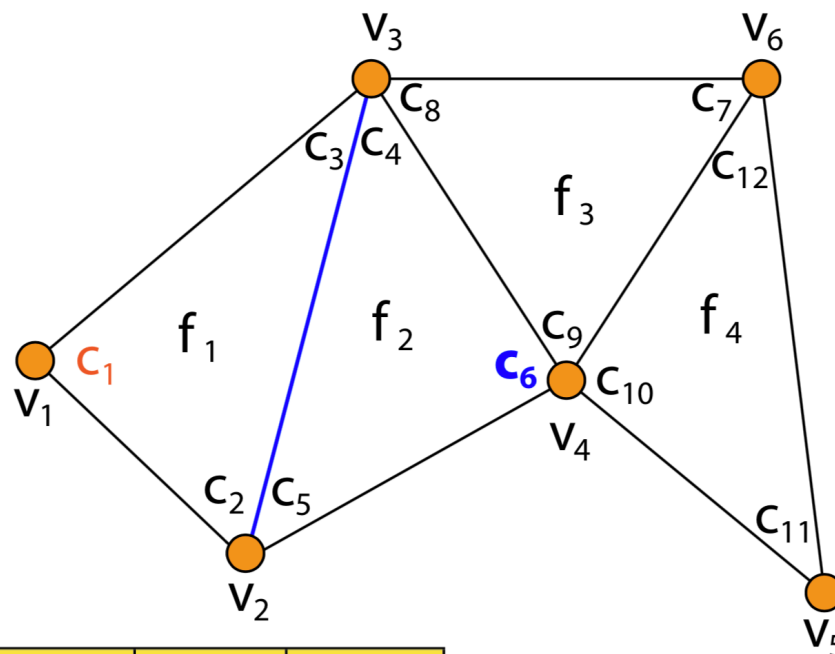
Queries:

- Which are the vertices of a given face?
- Which are the vertices adjacent to a given vertex?
- Which are the faces a vertex belong to?

Structural changes:

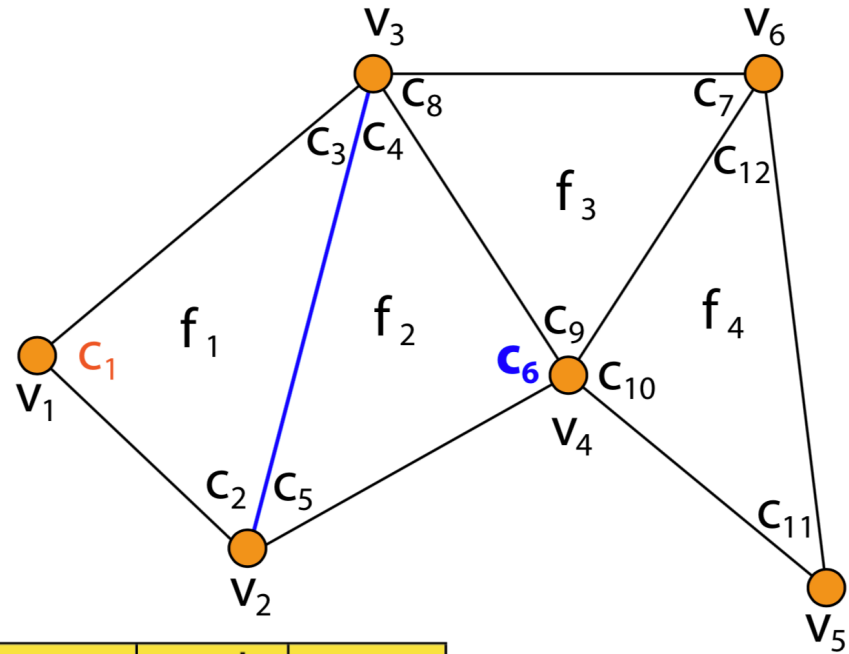
- edge flip
- mesh refinement

Corner Table Data Structure

[illegible]

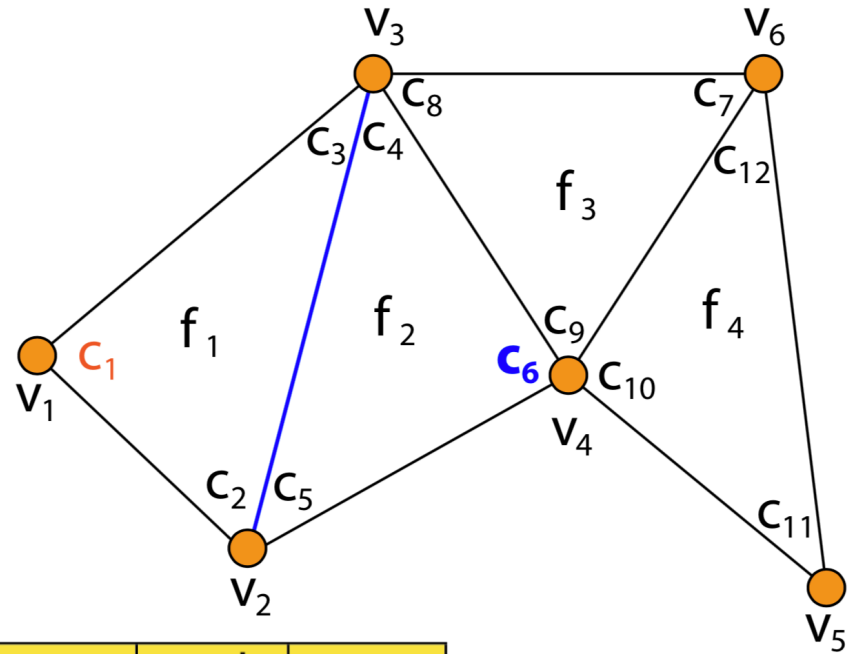
Corner Table Data Structure

- Which are the faces that contains v4?

[illegible]

Corner Table Data Structure

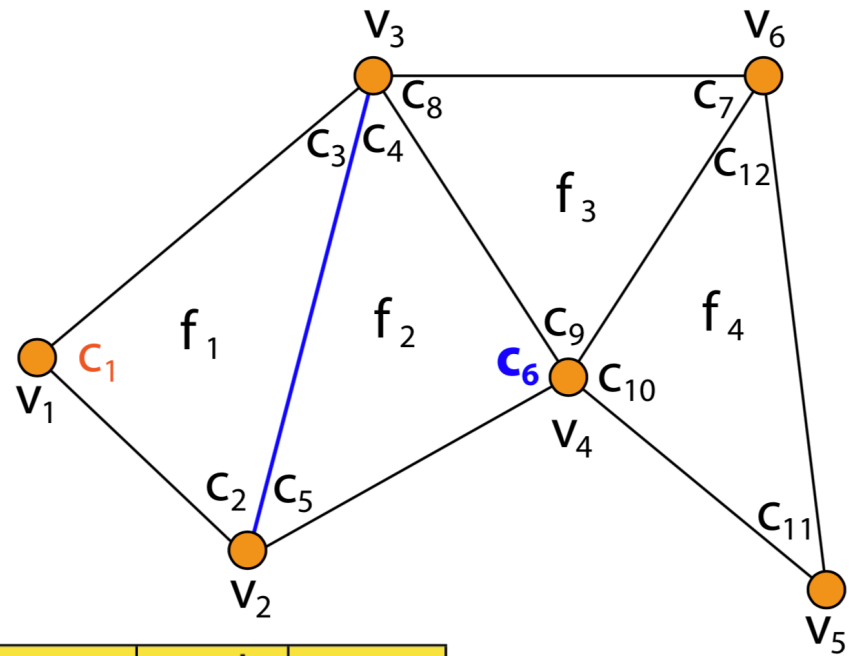
- Which are the faces that contains v4?
- Which are the vertices adjacent to v3?



corner	c.v	c.t	c.n	c.p	c.o	c.l	c.r
C ₁	V ₁	f ₁	C ₂	C ₃	C ₆	∅	∅
C ₂	V ₂	f ₁	C ₃	C ₁	∅	∅	C ₆
C ₃	V ₃	f ₁	C ₁	C ₂	∅	C ₆	∅
C ₄	V ₃	f ₂	C ₅	C ₆	∅	C ₇	C ₁
C ₅	V ₂	f ₂	C ₆	C ₄	C ₇	C ₁	∅
C ₆	V ₄	f ₂	C ₄	C ₅	C ₁	∅	C ₇
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

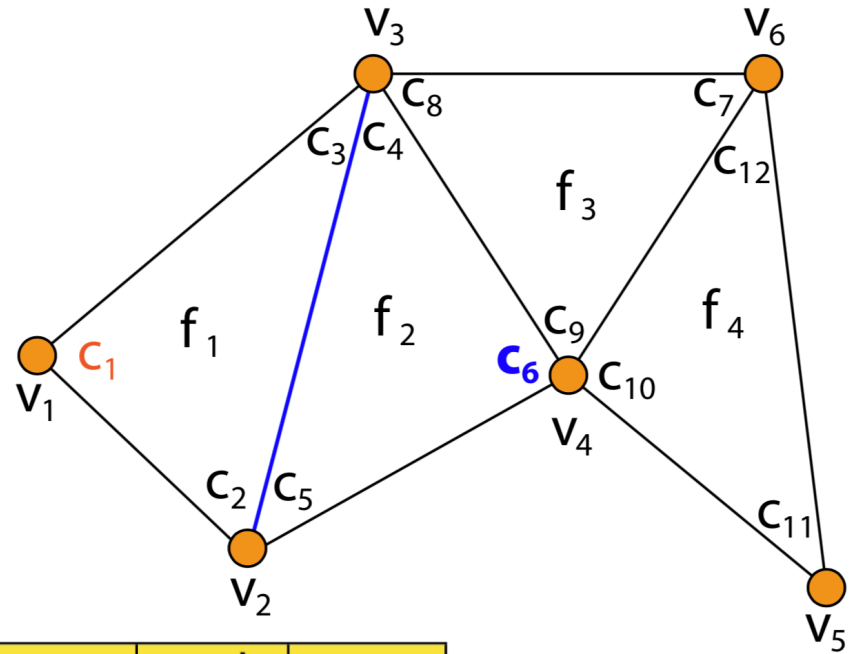
Corner Table Data Structure

- Which are the faces that contains v4?
- Which are the vertices adjacent to v3?
- Which are the vertices of f2?

[illegible]

Corner Table Data Structure

- Which are the faces that contains v4?
 - Which are the vertices adjacent to v3?
 - Which are the vertices of f2?
- (tables with faces+corner and vertices+corners are also built to speedup queries)



corner	c.v	c.t	c.n	c.p	c.o	c.l	c.r
c ₁	v ₁	f ₁	c ₂	c ₃	c ₆	∅	∅
c ₂	v ₂	f ₁	c ₃	c ₁	∅	∅	c ₆
c ₃	v ₃	f ₁	c ₁	c ₂	∅	c ₆	∅
c ₄	v ₃	f ₂	c ₅	c ₆	∅	c ₇	c ₁
c ₅	v ₂	f ₂	c ₆	c ₄	c ₇	c ₁	∅
c ₆	v ₄	f ₂	c ₄	c ₅	c ₁	∅	c ₇
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮