Daniel Sage & Arne Seitz

Course

Introduction to Graph Theory

Motivation

Graph theory
- Study mathematical structure and relationship

Applications
- Linguistic, computer, chemistry, biology, network

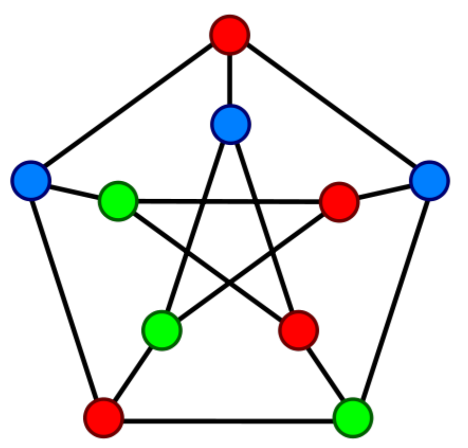
Identify the problem
Re-use existing algorithms

NP-hard problems	P problems
nondeterministic polynomial time	

NP-hard problems

nondeterministic polynomial time

- Traveling salesman
- Color labelling



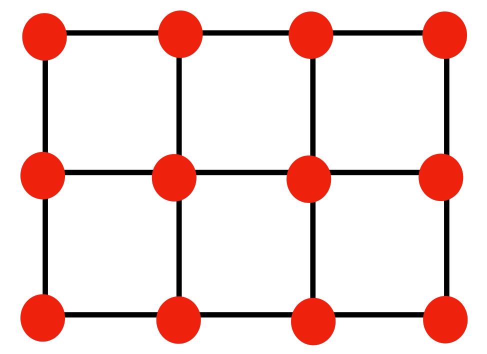
P problems

- Voronoi
- Minimum span tree
- Shortest path
- Assignment
- Max-flow / min-cut

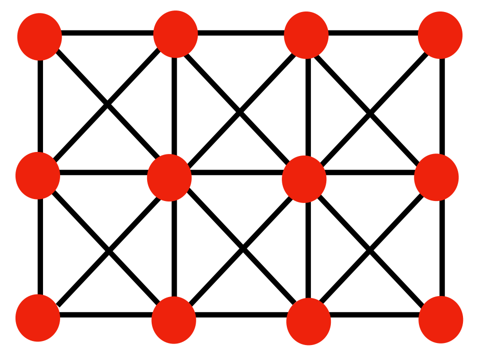
Graph Neural Network

Graphs in Bioimage Analysis

Data modeling
- Representation of image at pixel level

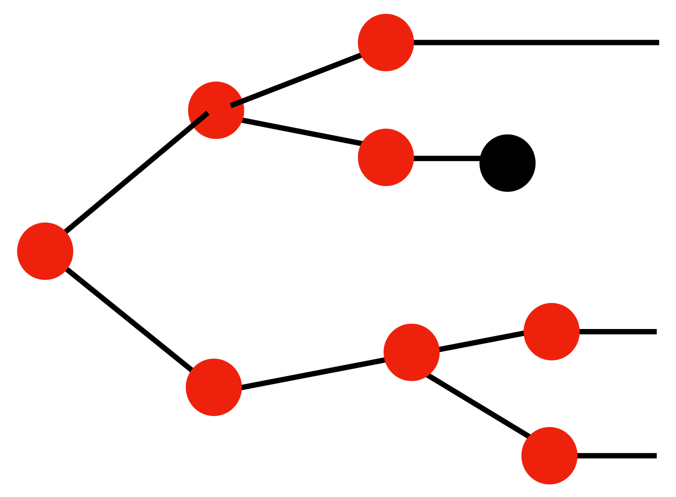


2D: 4-connected
3D: 6-connected

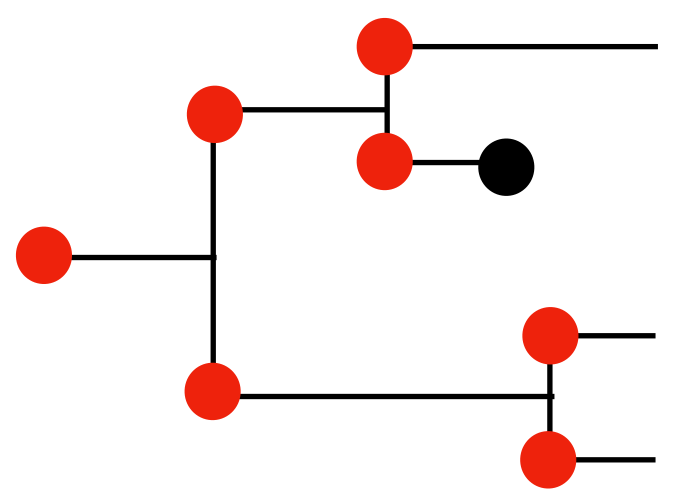


2D: 8-connected
3D: 26-connected

- Dimension: scale, space, time
- Object: particles, cell
- Spatial organization:
- Vascular structure: tree

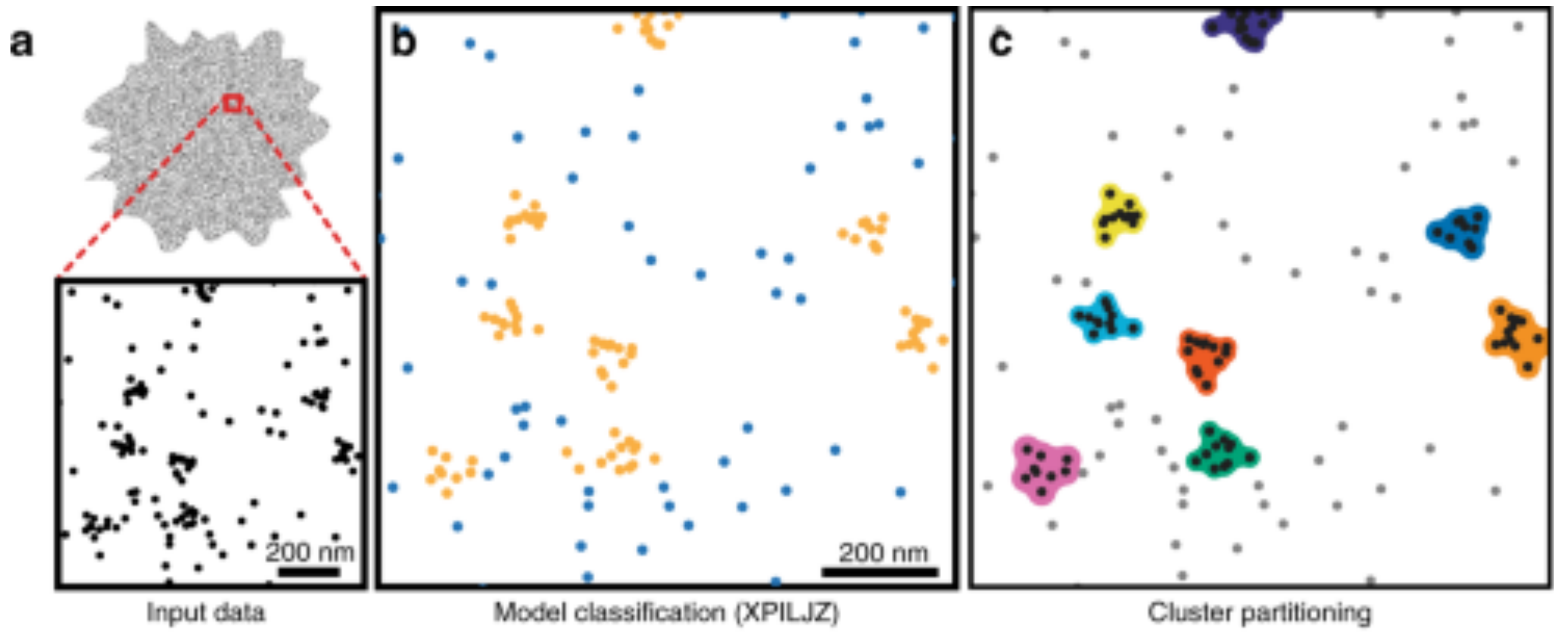


- Relation: cell lineage, tracking, ...

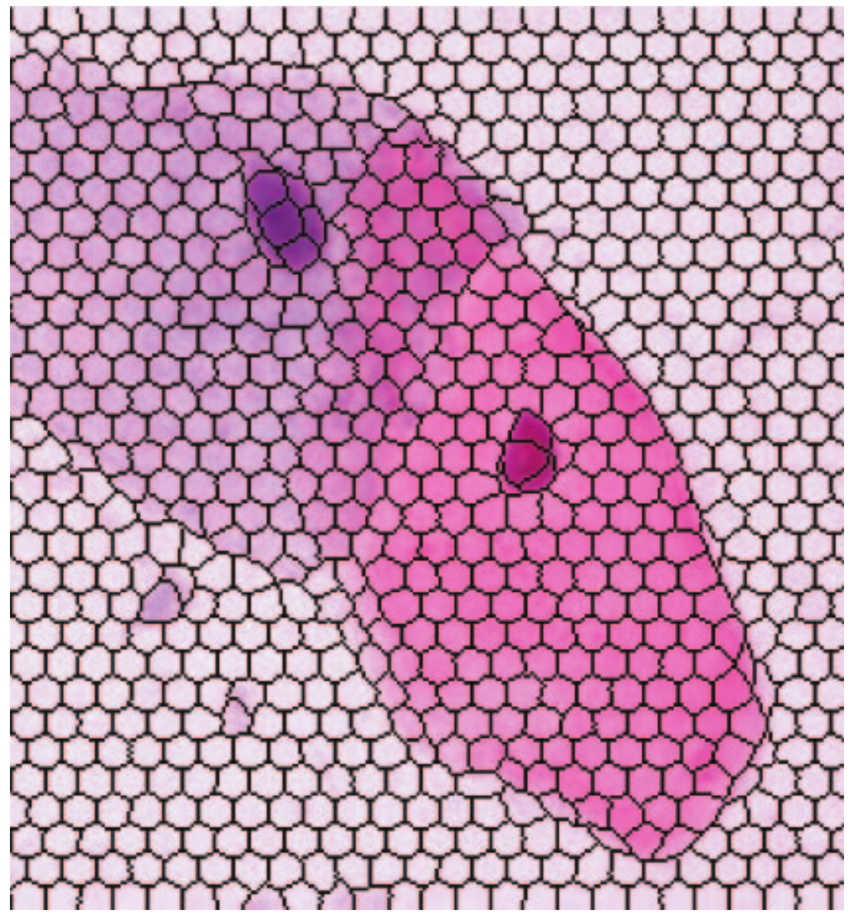


- Mosaic of images

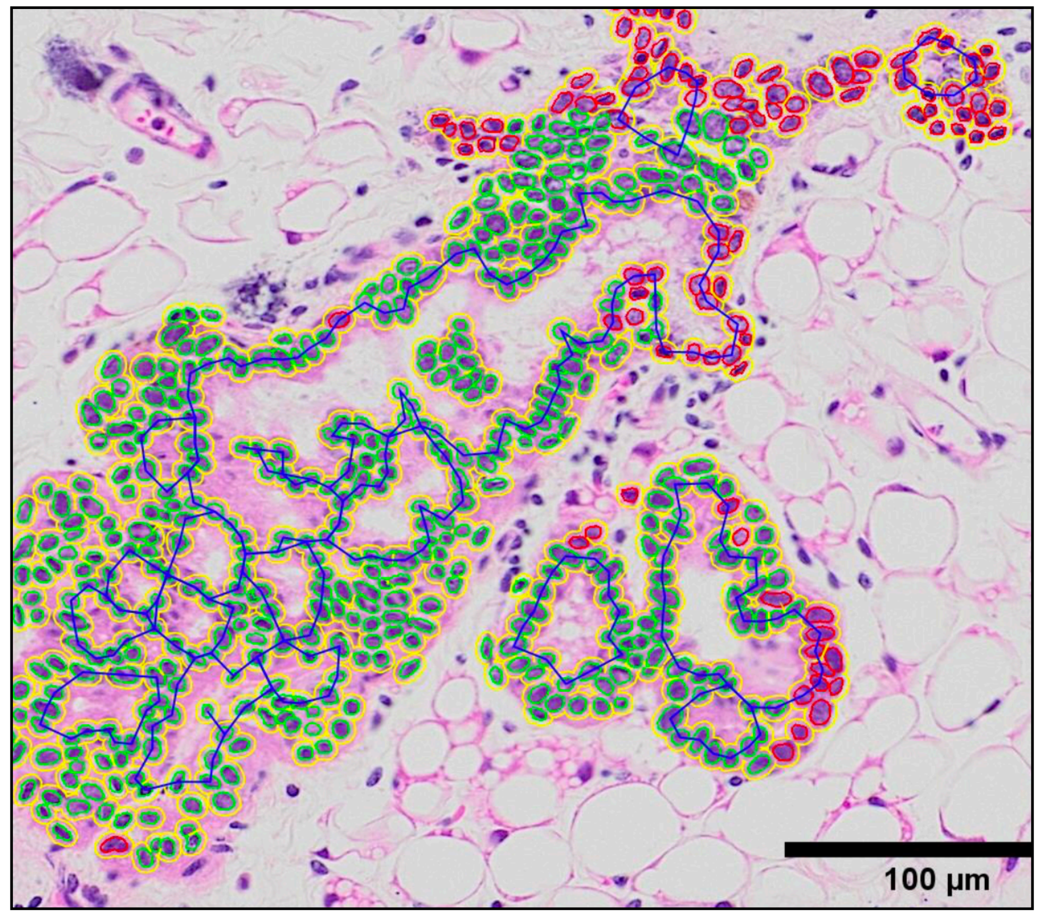
Graph Modelling



Molecules ➡ Objects

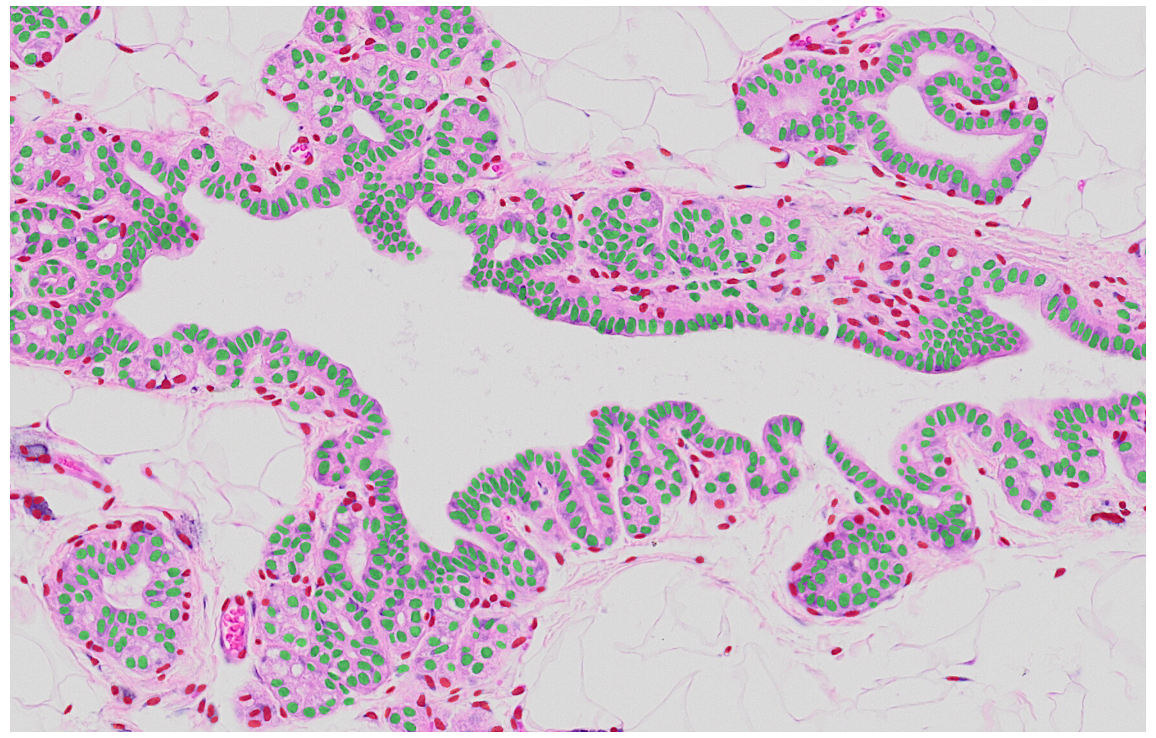


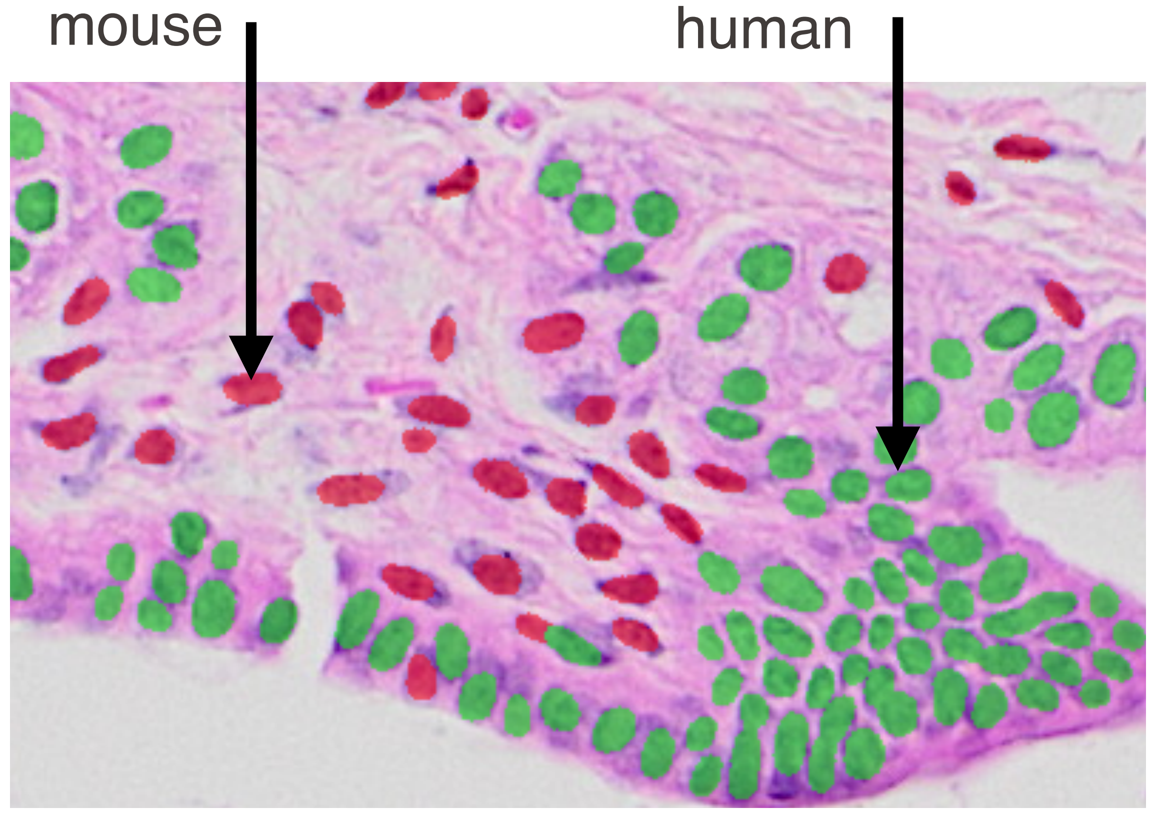
Superpixel ➡ Objects



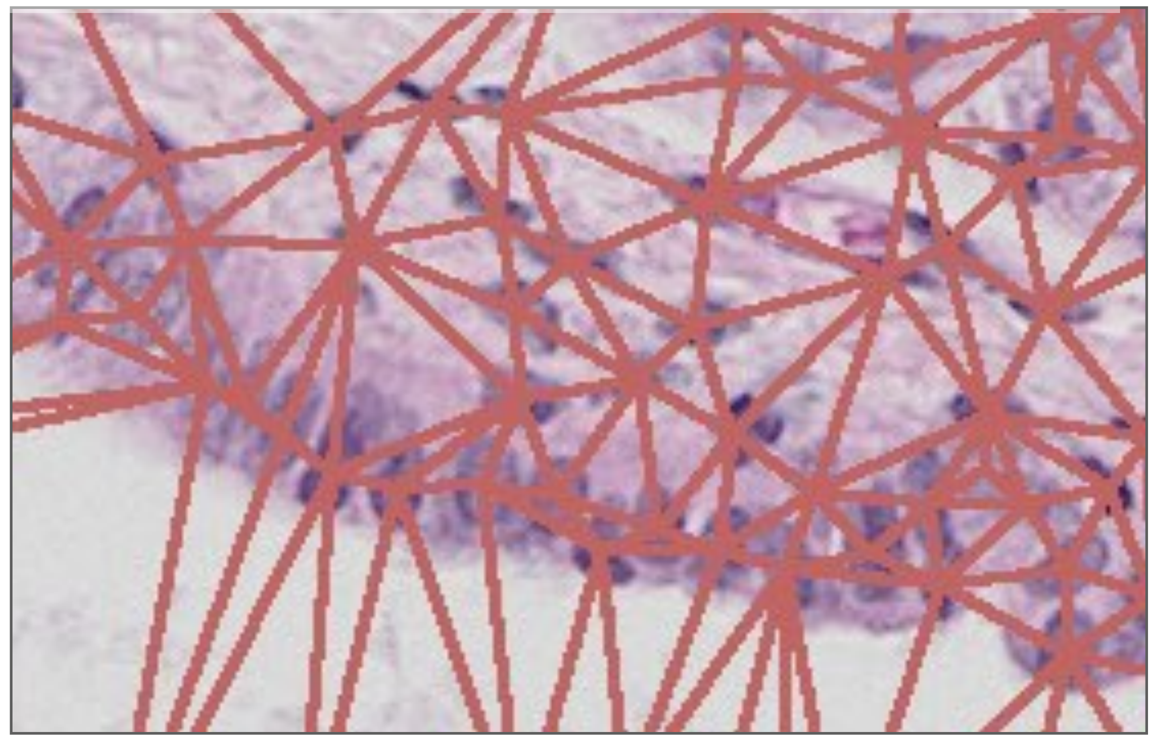
Cell ➡ Objects

Image ➡ Objects





Delaunay Graph



Morphometric features: 68.8 %

Textural information: 85.7%

Contextual, connectivity: 96.4%

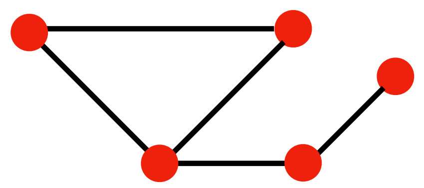
Q. Juppet, et al., Journal of Mammary Gland Biology and Neoplasia, 2021

Graph Structure

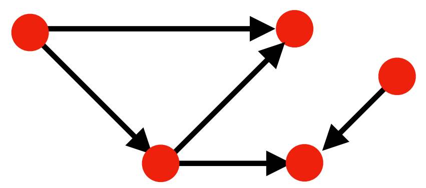
Graph
Vertex (nodes, points)
Edges (arc, line)

$$G = (V, E)$$

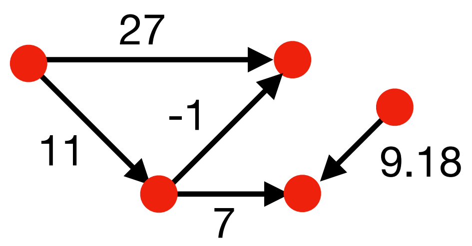
Type



undirected

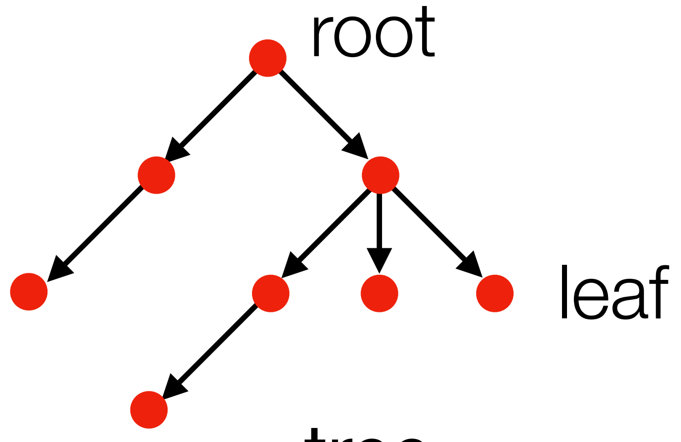


directed

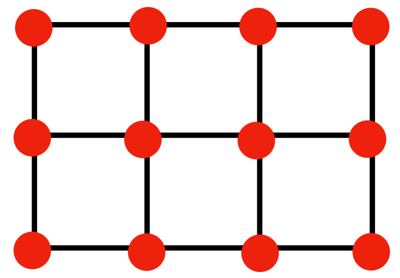


weighted

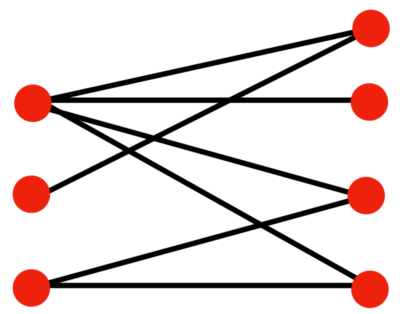
Structure



tree

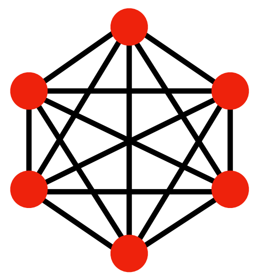


adjacency graph

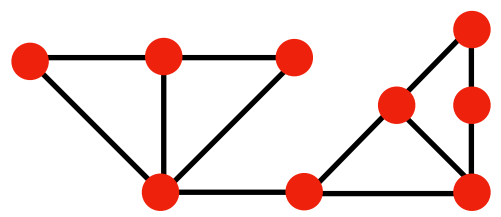


bipartite graph

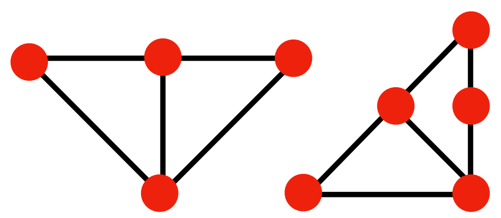
Connectivity



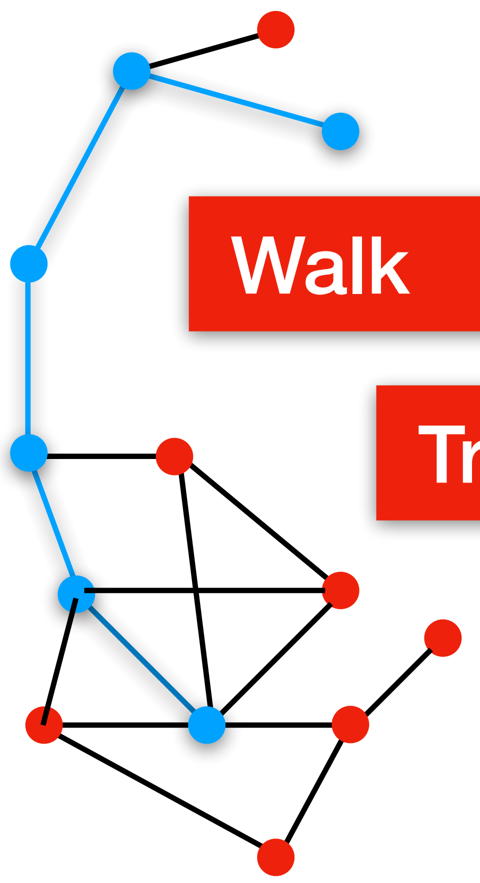
full



weak



disconnected

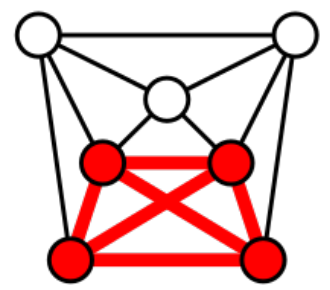


Walk sequence of alternating vertices and edges

Trial walk with no repeated edges

Path open trail with no repeated vertices

Cycle closed trail with no repeated vertices



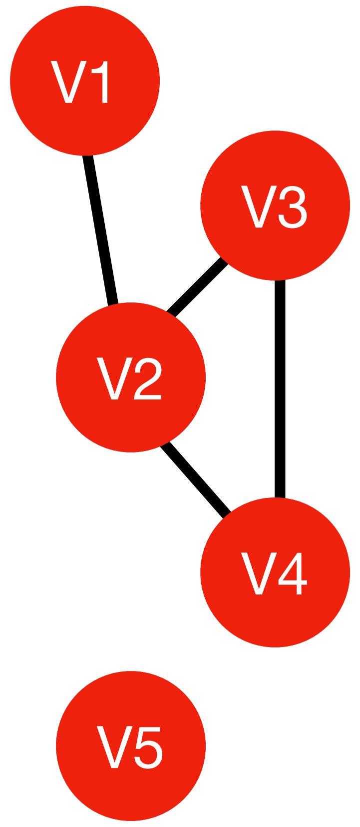
clique
subgraph fully connected
Find cliques is a
NP-hard problem

Graph Representation in Java

Data Structure

- e.g. Object, ArrayList, Linked List, Map

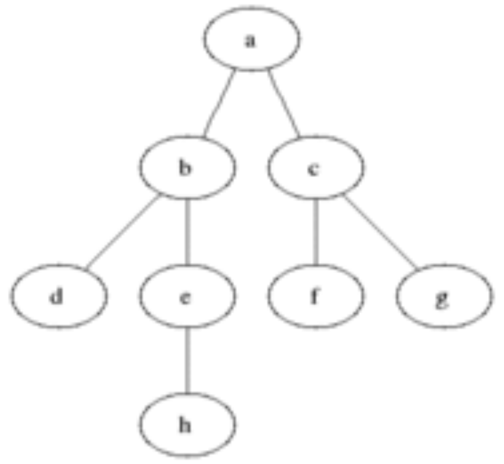
```
public class Vertex {
    public String name;
    public ArrayList<Vertex> vertices;
    public Vertex(String name) {
        vertices = new ArrayList<Vertex>();
        this.name = name;
    }
}

public GraphExample() {
   List<Vertex> graph =
          new ArrayList<Vertex>();
    for(int i=1; i<5; i++)
        graph.add(new Vertex("V" + i));
    addEdge(graph, 1, 2);
    addEdge(graph, 2, 3);
    addEdge(graph, 3, 4);
    addEdge(graph, 2, 4);
}

void addEdge(List<Vertex> g, int a, int b) {
    g(a).vertices.add(g.get(b));
    g(b).vertices.add(g.get(a));
}
```



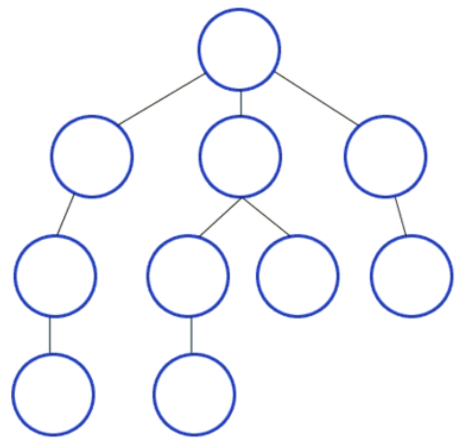
Recursive algorithm

- e.g. graph traversal

```
void dfs(int i, int[][] mat, boolean[] visited) {
    if(!visited[i]) {
        visited[i] = true;
        System.out.print( (i+1) + " ");
        for (int j = 0; j < mat[i].length; j++)
            if (mat[i][j] == 1 && !visited[j])
                dfs(j, mat, visited); // Visit
    }
}
```



Breath First Search



Depth First Search

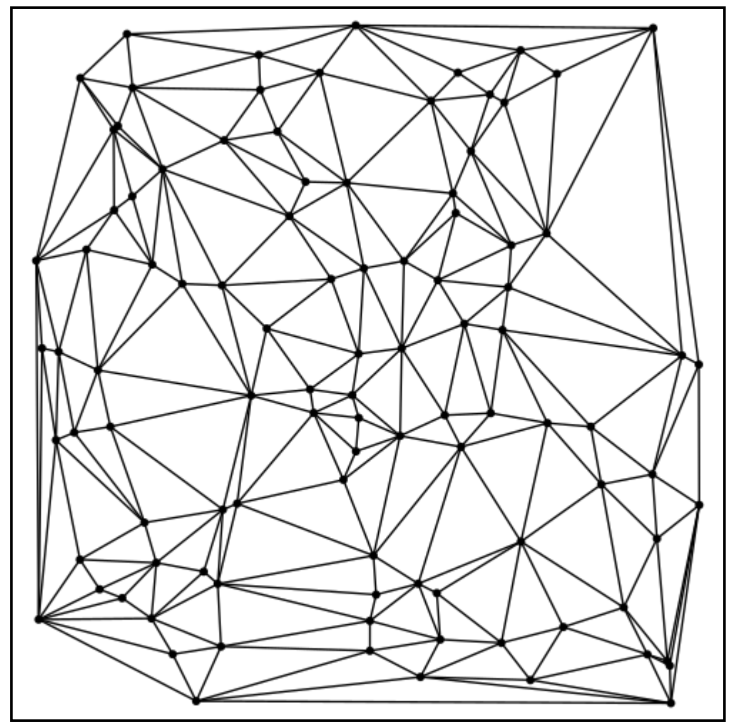
Third-party Java libraries: JGraphT

- e.g. Voronoi, Delaunay, Hungarian

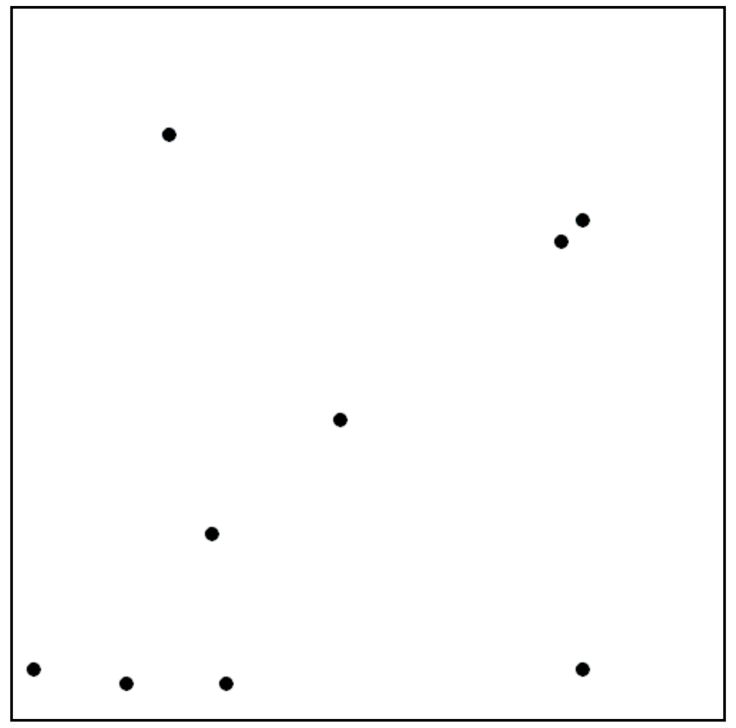
Planar Graph

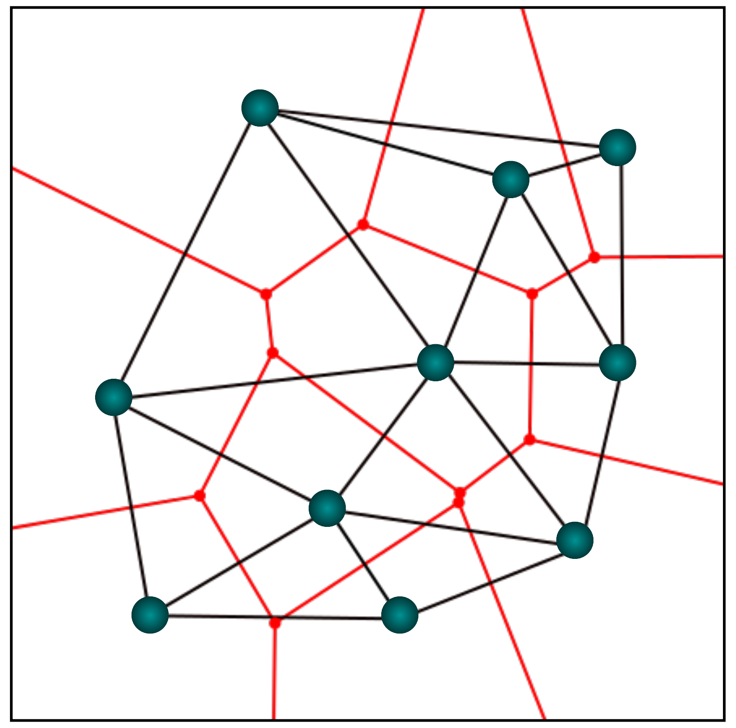
A planar graph is drawn without edges crossing ➔ the edges are defining regions (faces)

Delaunay triangulation built from points



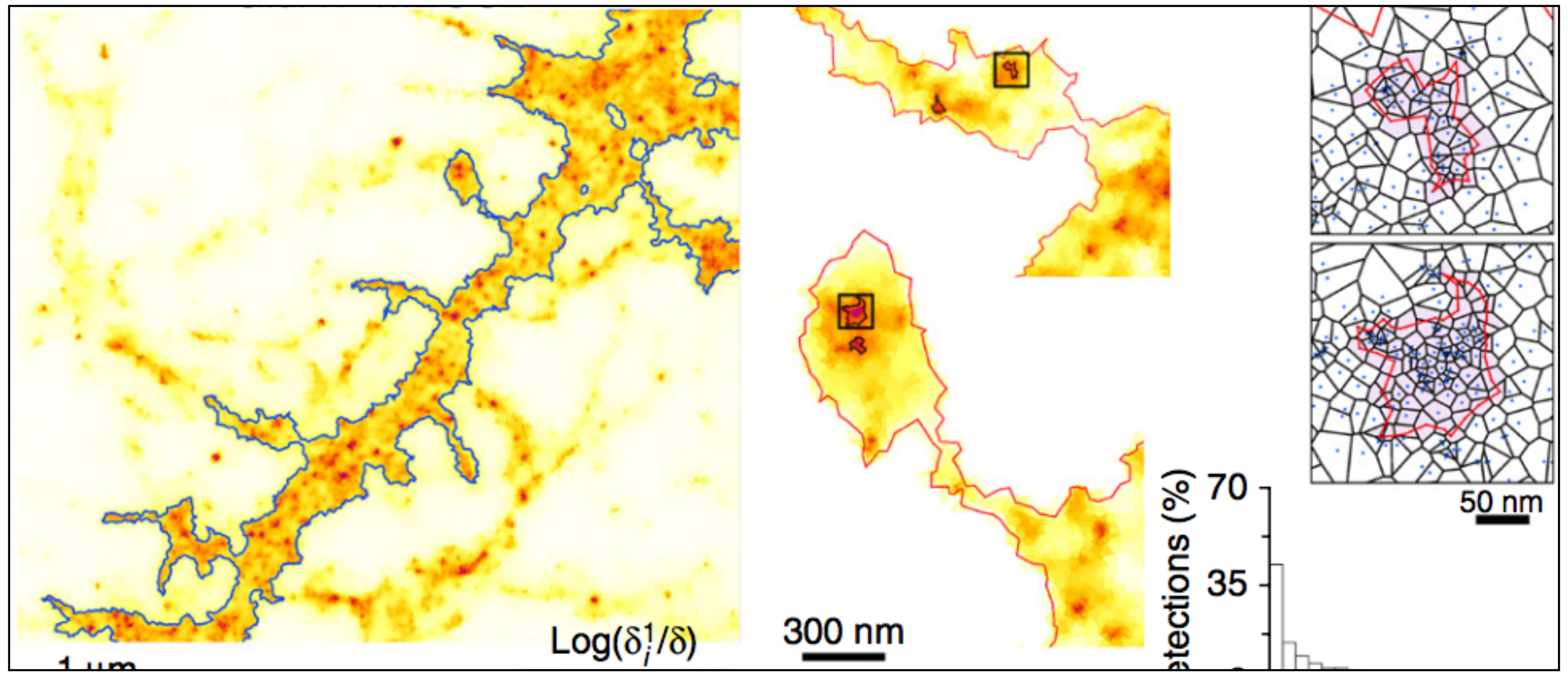
Voronoi tessellation from seed points



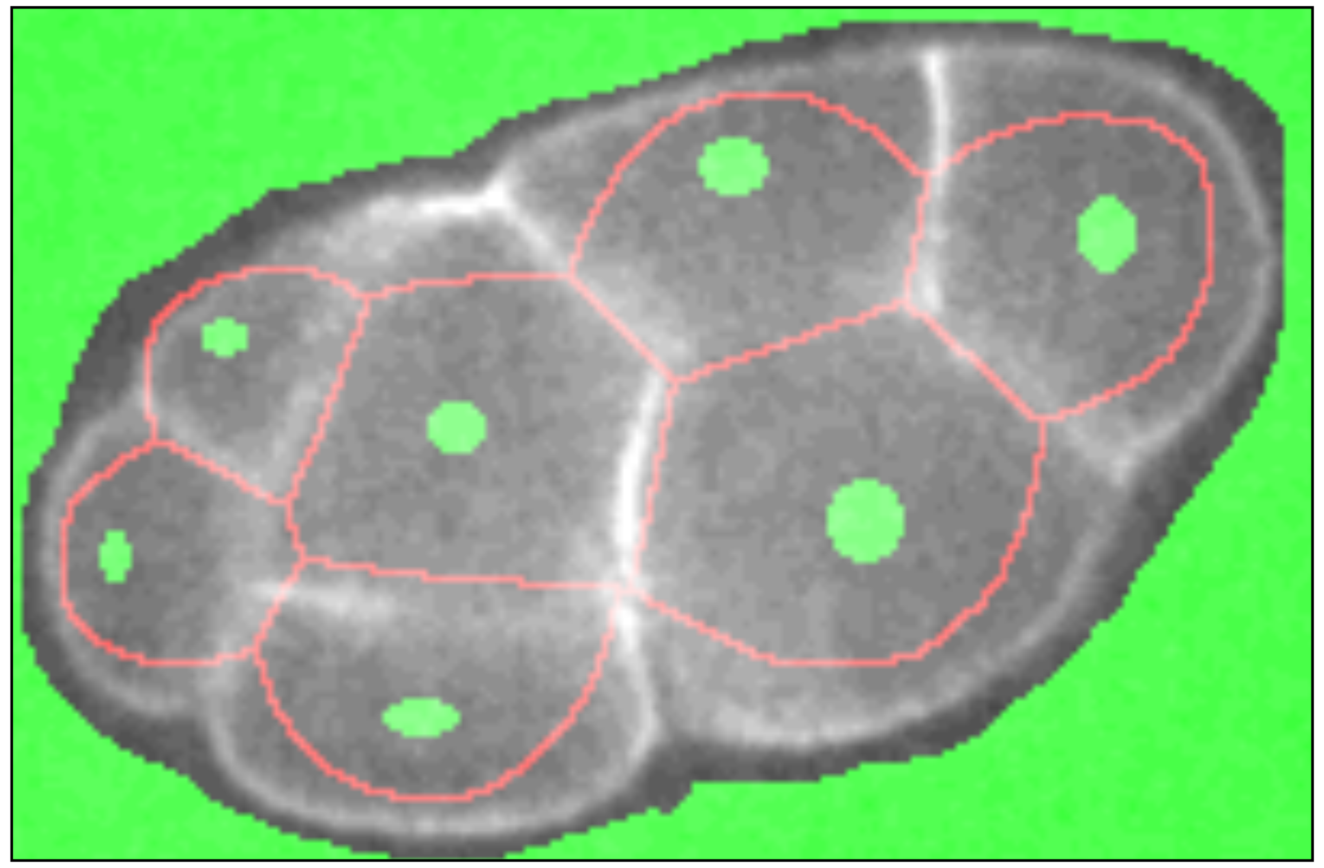


Dual graph

- Voronoi diagram
- Delaunay triangulation



Florian Levet. SR-Tesseler: segment and quantify localization-based super-resolution microscopy data, Nature Method 2015



C elegans embryo, cell membrane and nucleus. Courtesy of Radek Janeke, EPFL

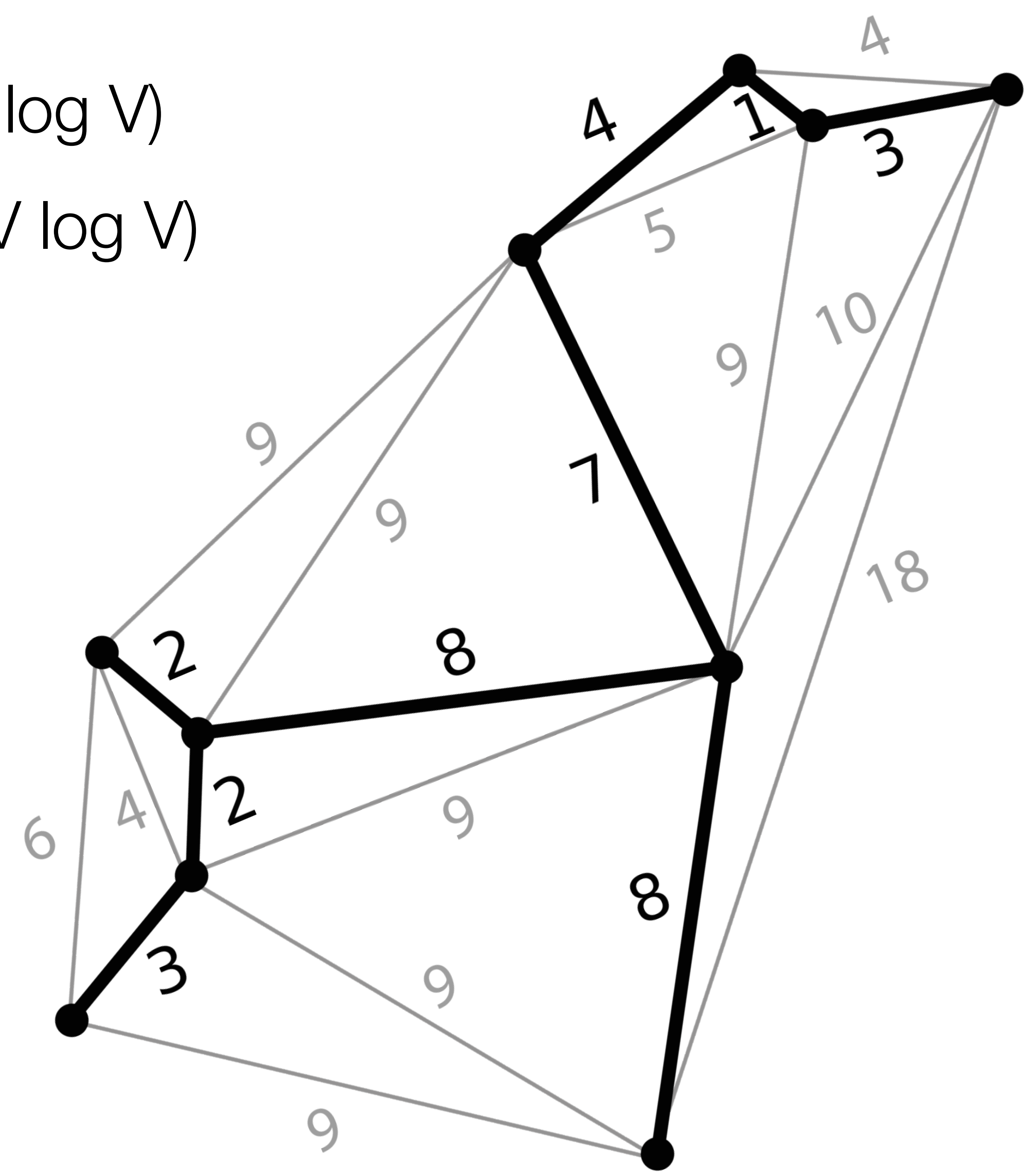
Minimum Spanning Tree

Tree-like structure

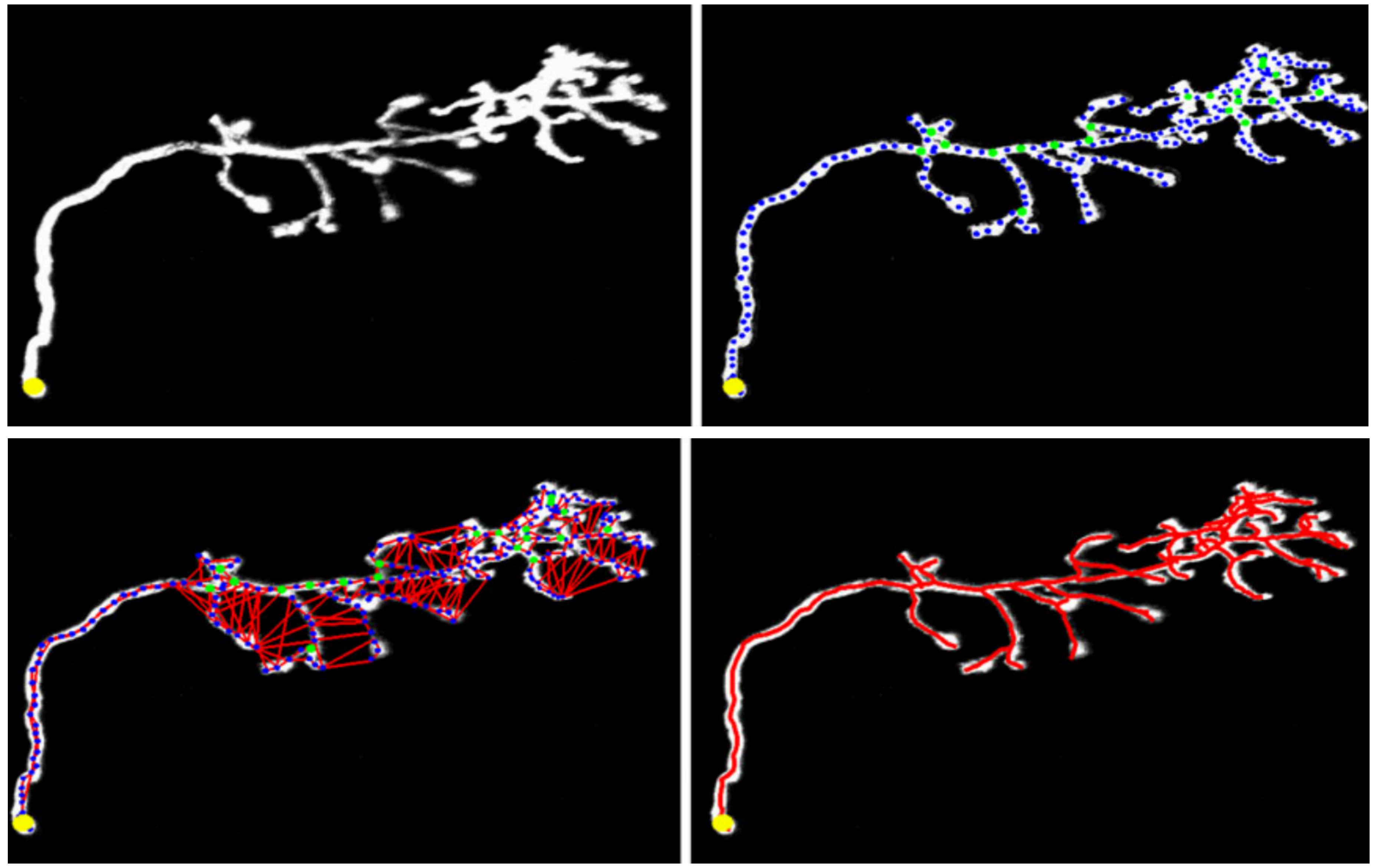
- Vascular network, dendrites, ...
- Space 2D or 3D
- Time: Cell lineage

Solver

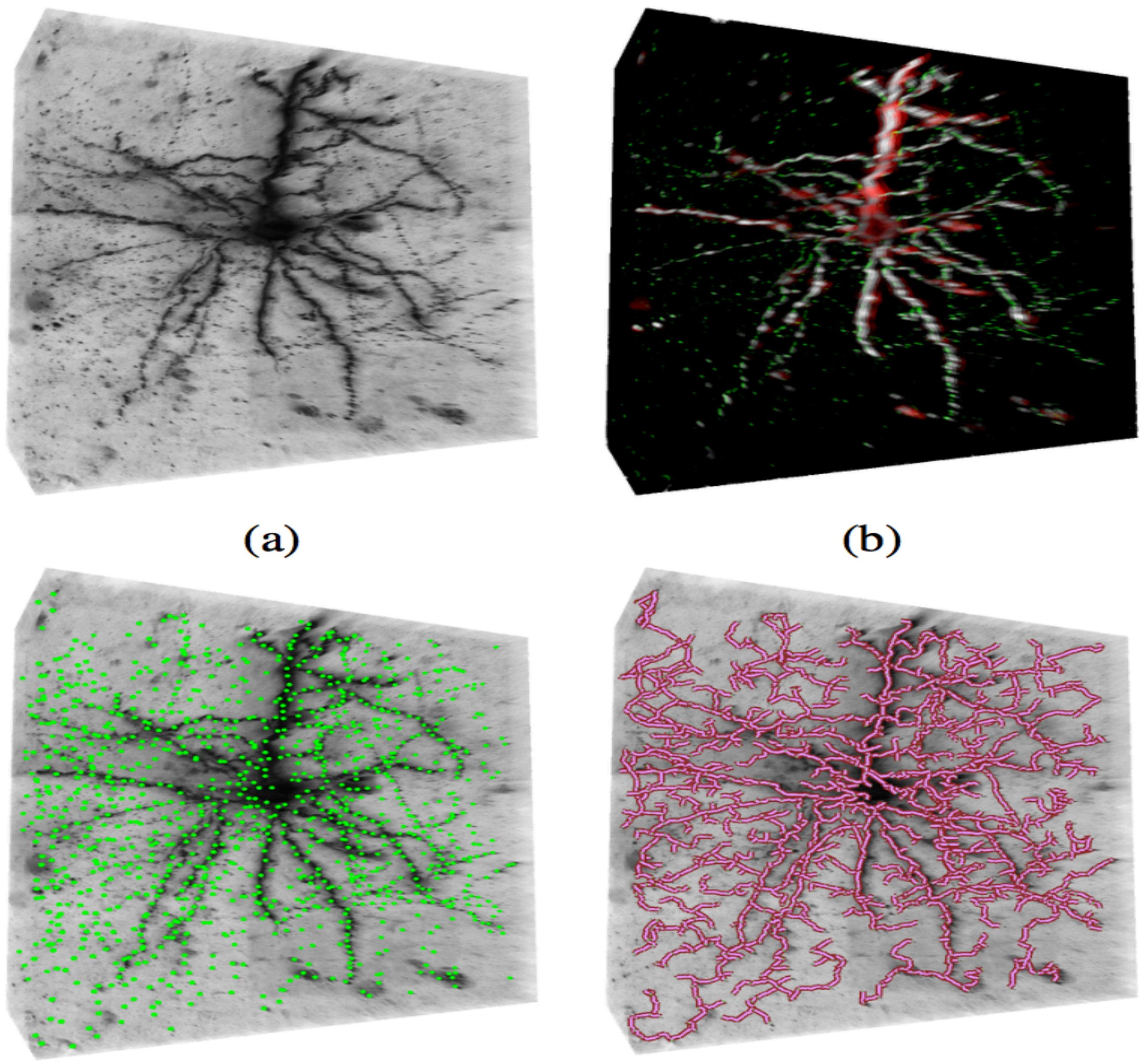
- Kruskal O(E log V)
- Prim O(E + V log V)



https://visualgo.net/en/mst



Olfactory projection fibers, E. Türetken, Neuroinform, 2011



Dendrites. German Gonzalez, ECCV 2008

Bipartite Graph Matching

$$G = (V, E)$$

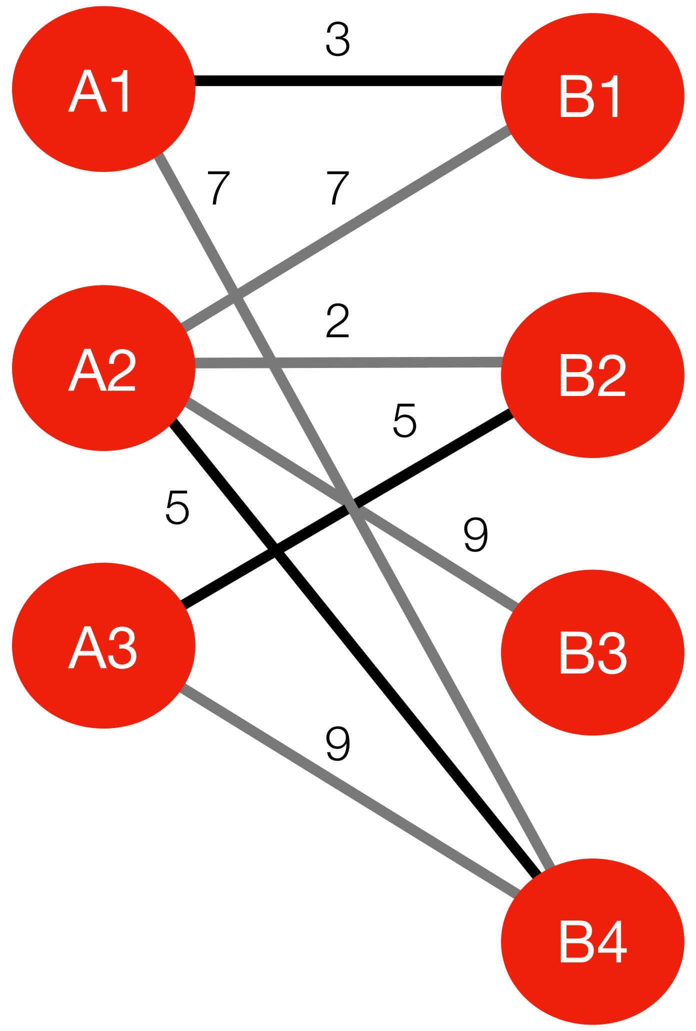
$$V = V_A \cup V_B$$

$$V_A \cap V_B = \varnothing$$

- Directed
- Weight (cost ξ_{ij})
- Balanced

Assignment Problem

- Minimum weight perfect matching
- Hungarian algorithm O(E V²)



	A1	A2	A3	A4	A5
B1	5	9			6
B2	4	6			
B3	8	5	1		
B4			9	3	2
B5			6	5	3

?

Is it the best match?

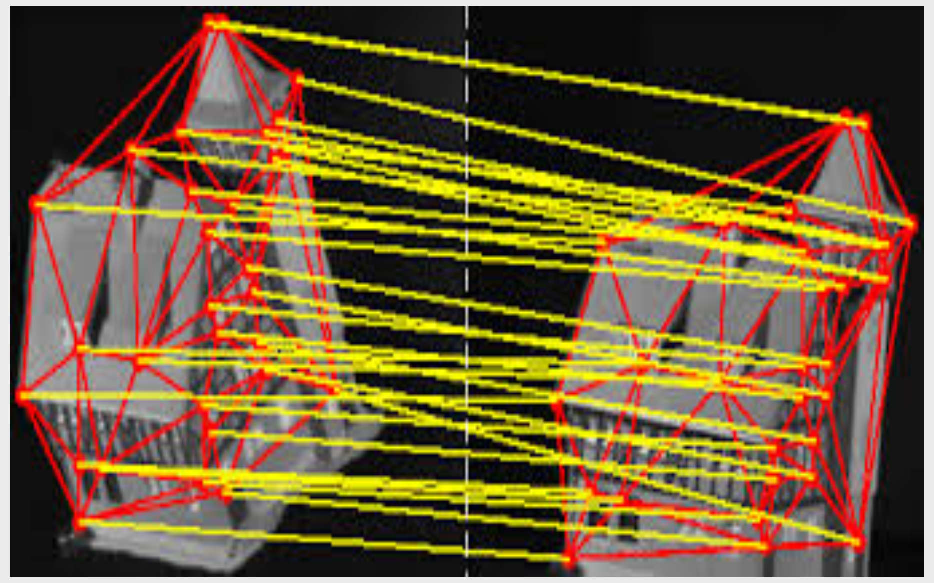
	A1	A2	A3	A4	A5
B1	5	9			6
B2	4	6			
B3	8	5	1		
B4			9	3	2
B5			6	5	3

First Minimum
4+5+6+3+6=24

	A1	A2	A3	A4	A5
B1	5	9			6
B2	4	6			
B3	8	5	1		
B4			9	3	2
B5			6	5	3

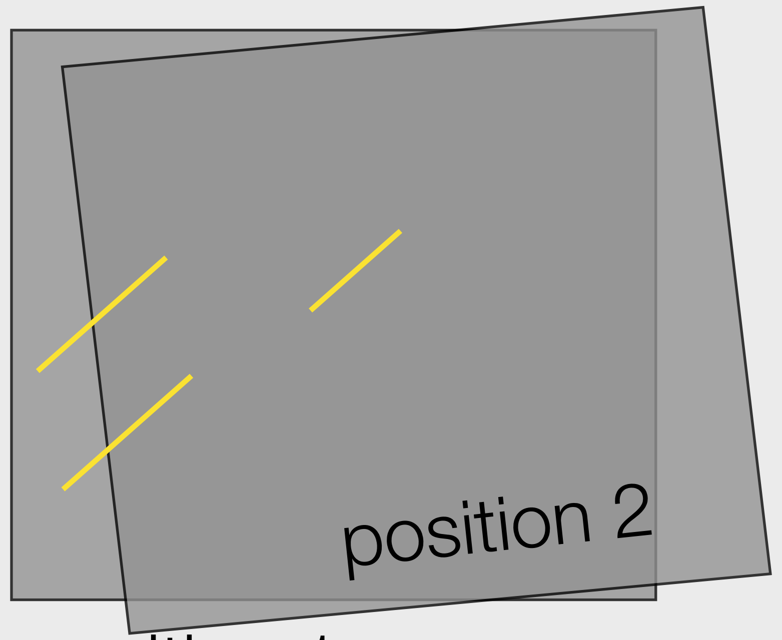
Greedy
1+2+4+5+9=21

Stereo matching



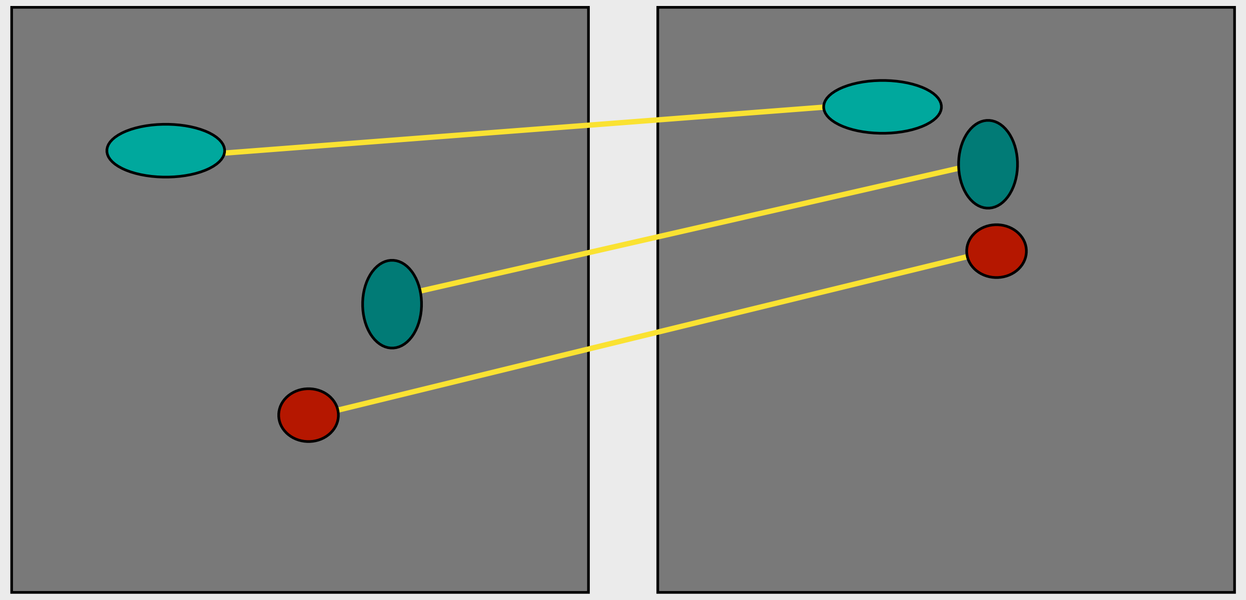
left right

Registration



position 1

Tracking

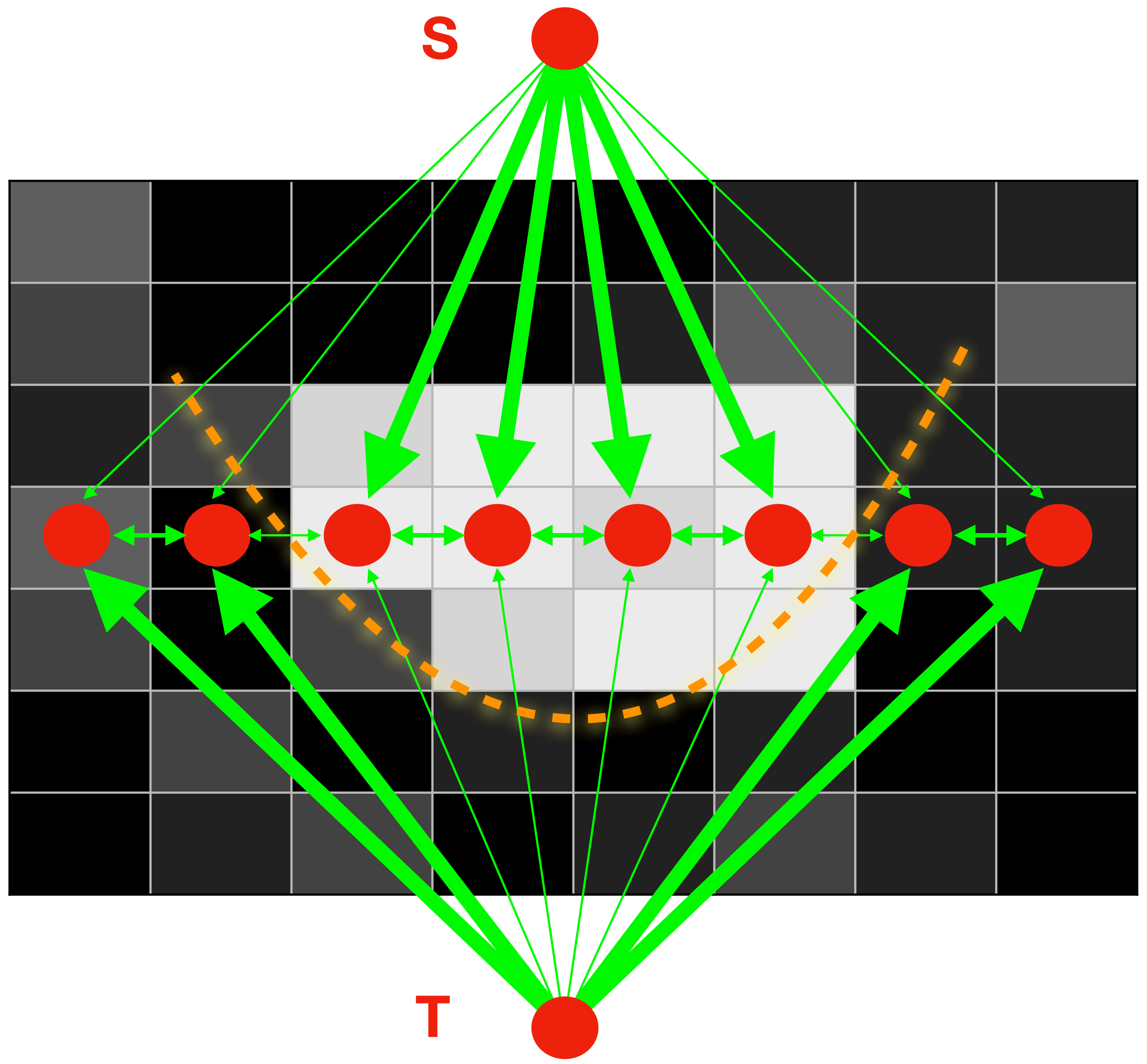


frame k frame k+1

Graph Cut

Minimal cut

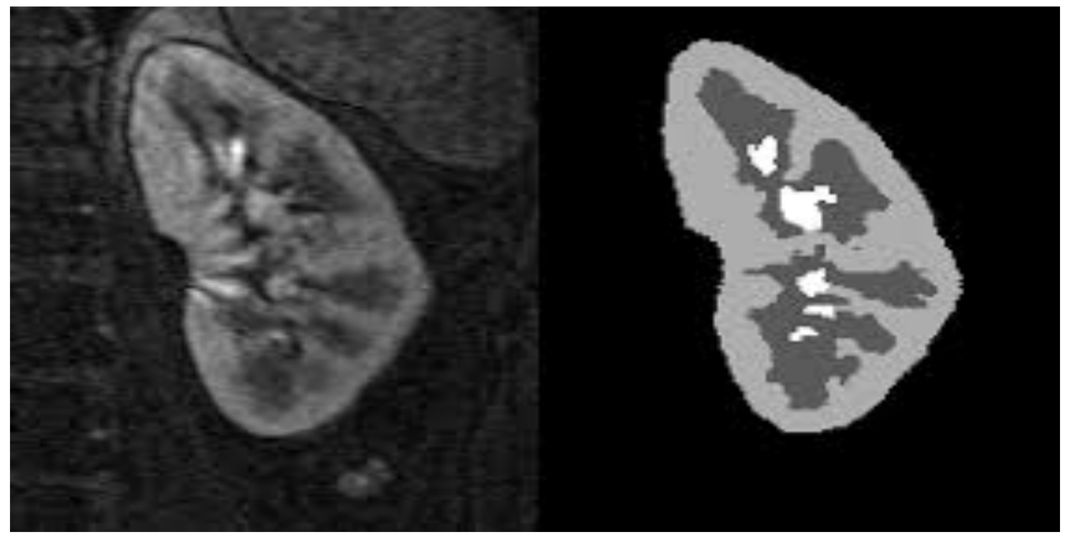
- Build a graph with a source **S** and a sink **T**
- Based on min-cut max-flow theorem
- Efficient way to compute (Boykov-Kolmogorov)
- Interactivity: user guide partition



Weights for image segmentation

- **Source** with probability to belong to the foreground
- **Sink** with probability to belong to the foreground
- **Neighborhood** with a measure of similarity of adjacent pixels

Application to image segmentation with complex background

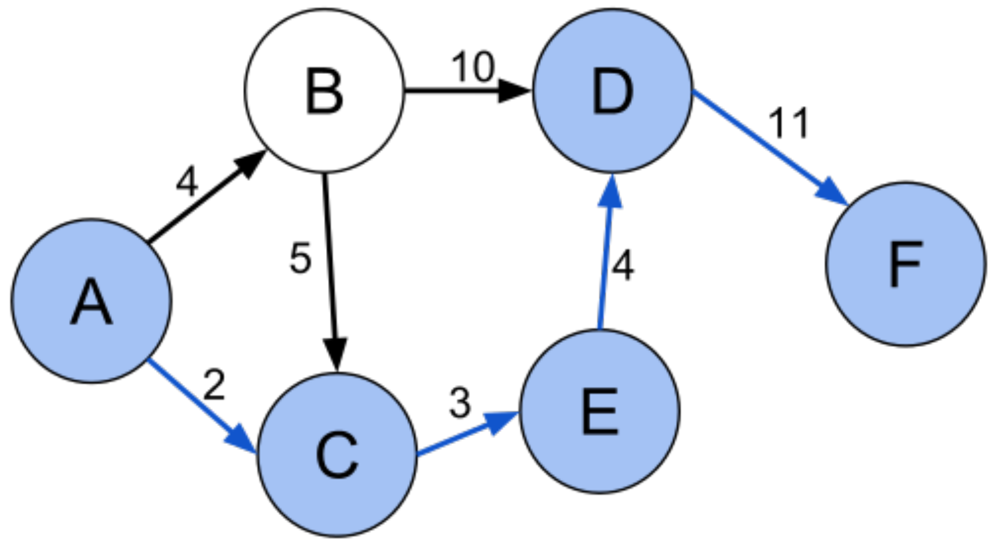


Interactive Graph Cuts www.csd.uwo.ca

Shortest Path

Shortest path problem in weighted graph

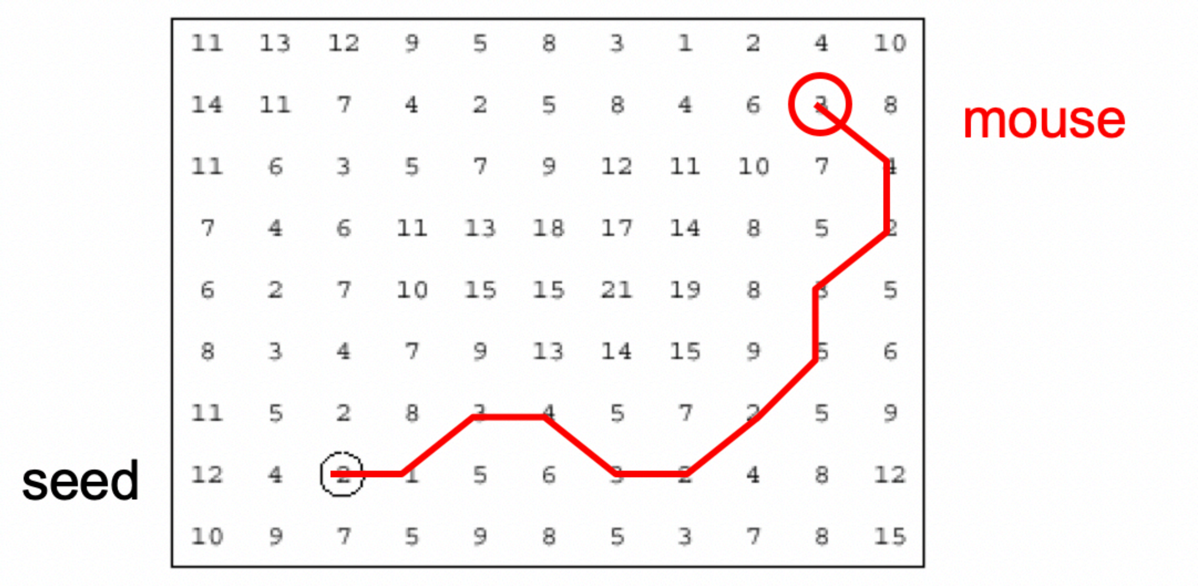
- Find a path between a starting vertex and and ending vertex
- Minimise the sum of weights of edges
- Applications: GPS route, intelligent scissors in imaging, clipping path in photo…

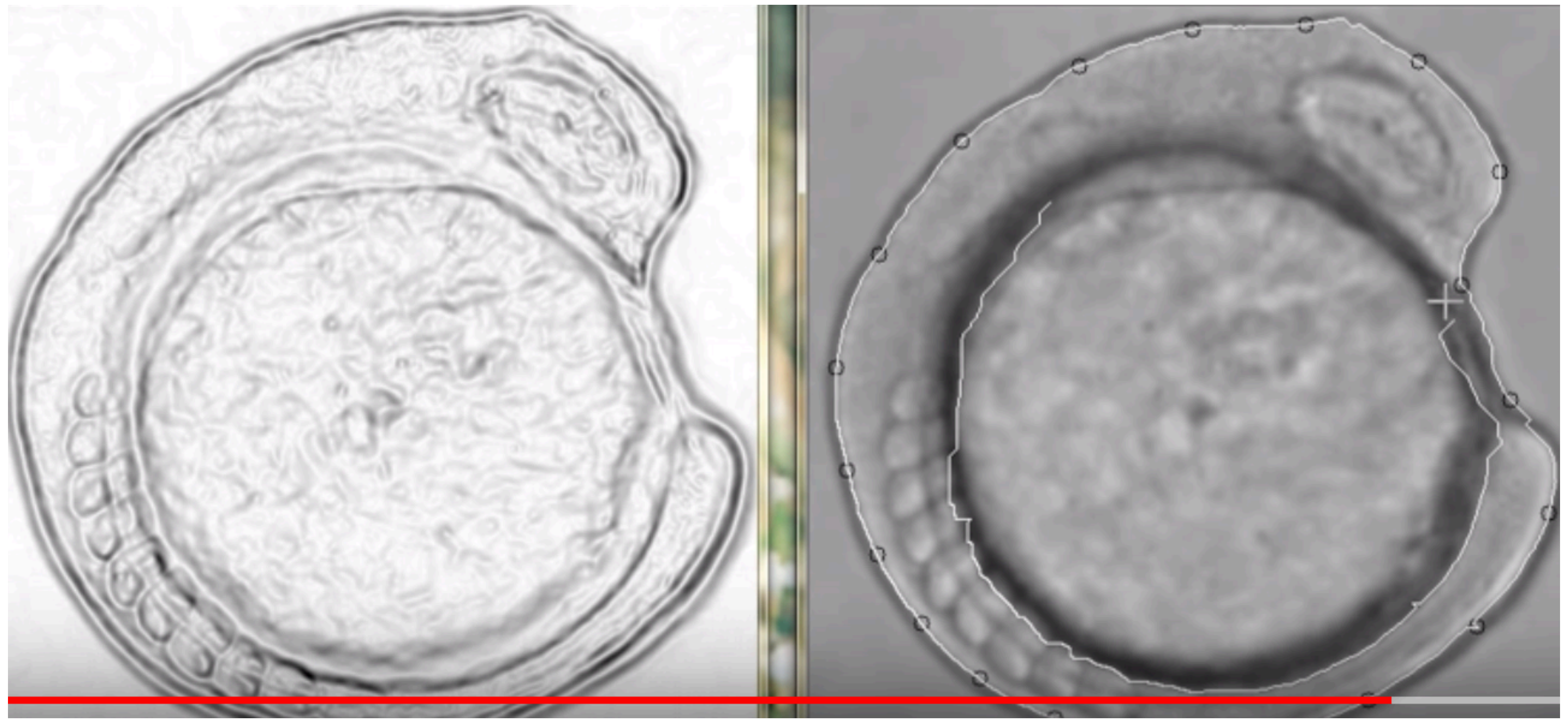


Djikstra Algorithm

Explore to the end point and backtrack to start point

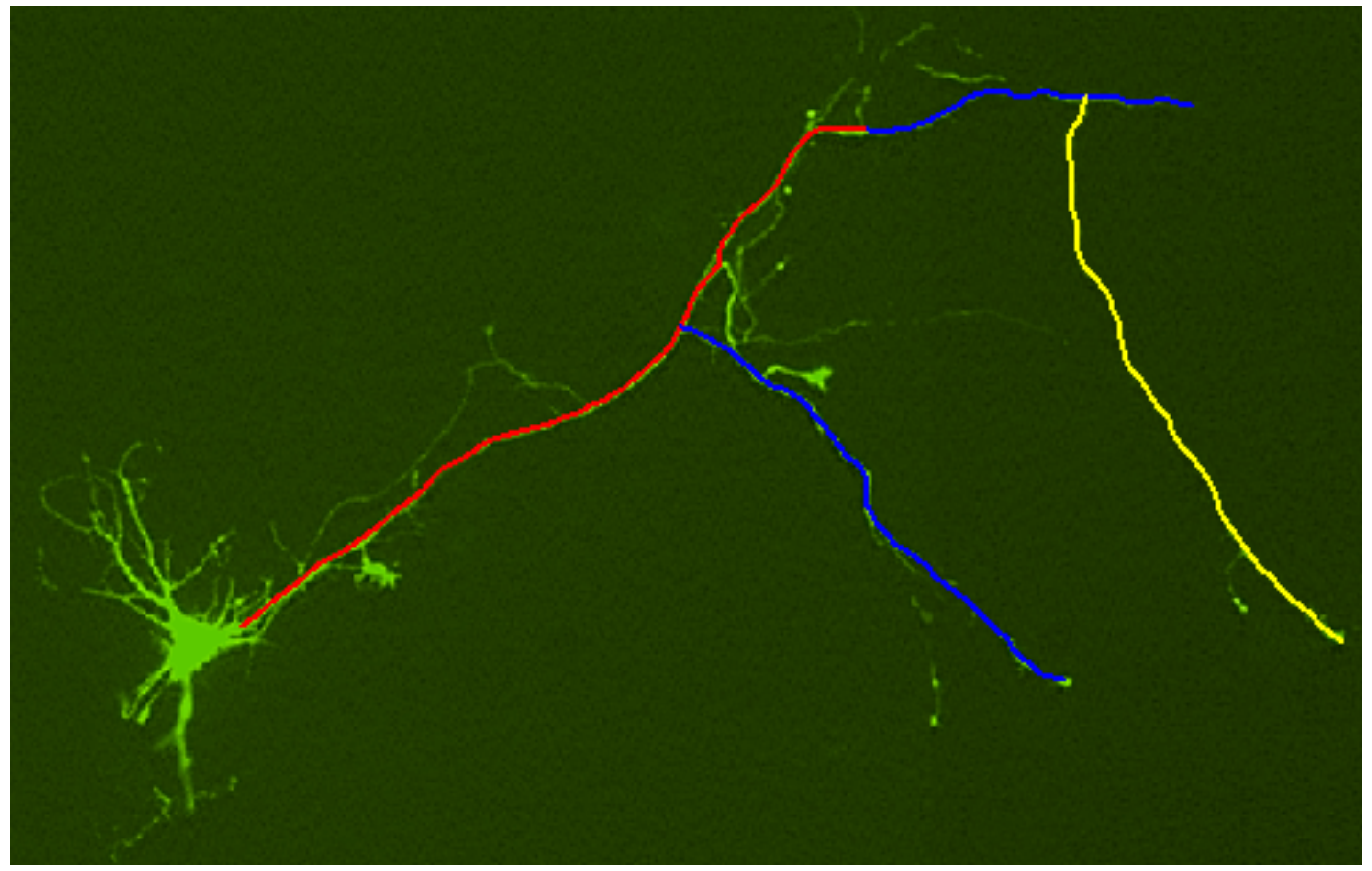
- Picks the unvisited vertex with the lowest distance
- Distance through it to each unvisited neighbor
- Direct implementation $O(V)$ ➜ Fast $O(E+V\log(V))$





https://www.youtube.com/watch?v=X_dZ_7xAcIM

Franklin Fang, Intelligent Scissor



Livewire

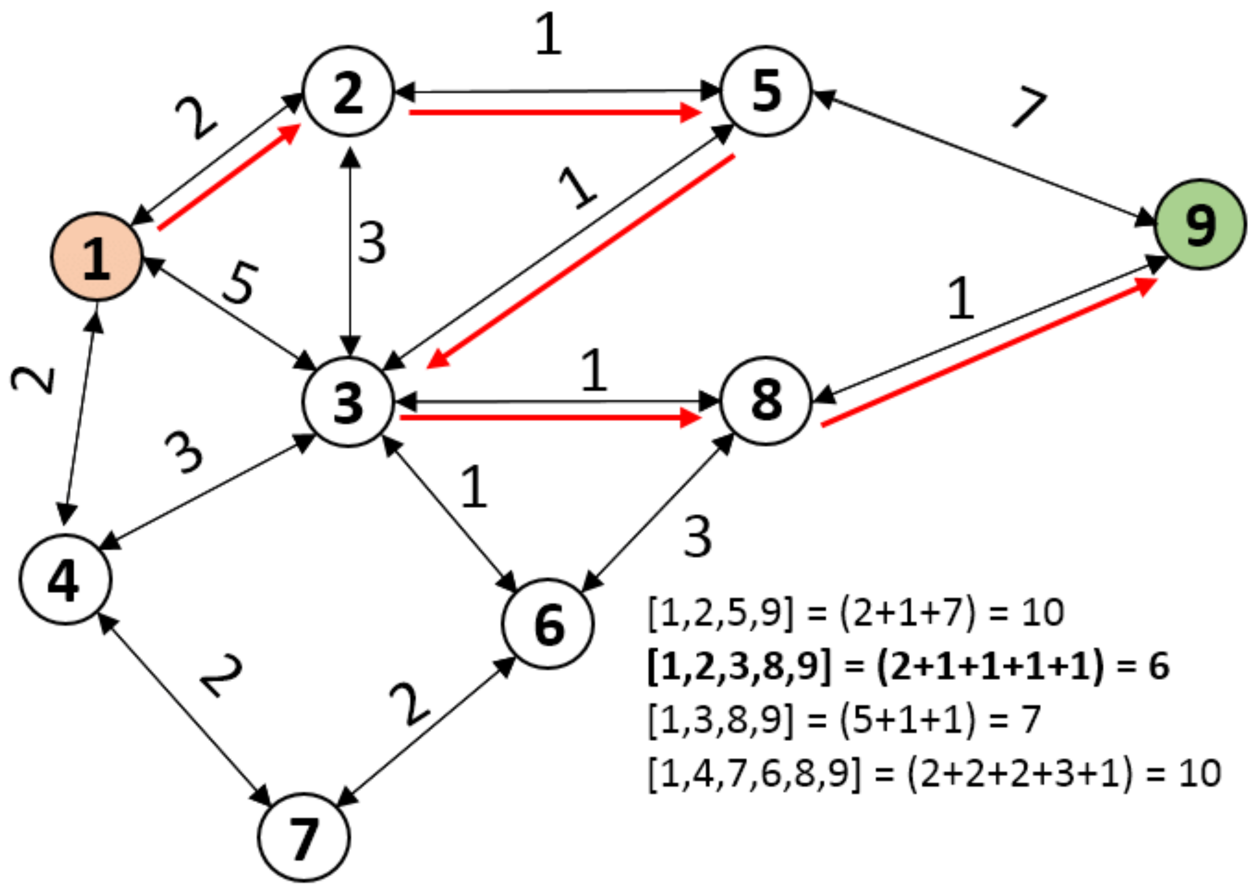
Erik Meijering, Neuronj

Shortest Path

Shortest path problem in weighted graph

- Find a path between a starting vertex and and ending vertex
- Minimise the sum of weights of edges
- Applications: GPS route, intelligent scissors in imaging, clipping path in photo…

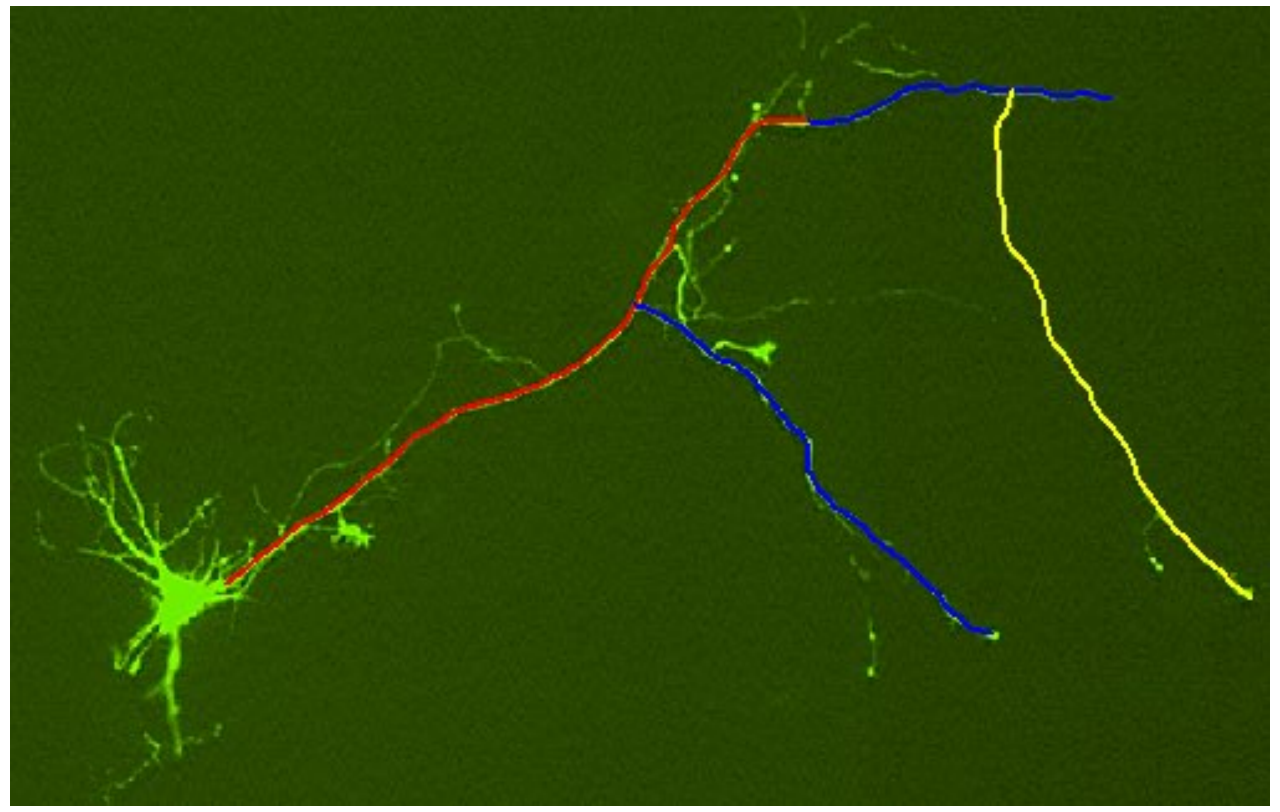
Shortest Path



Dijkstra Algorithm

Find the shortest route

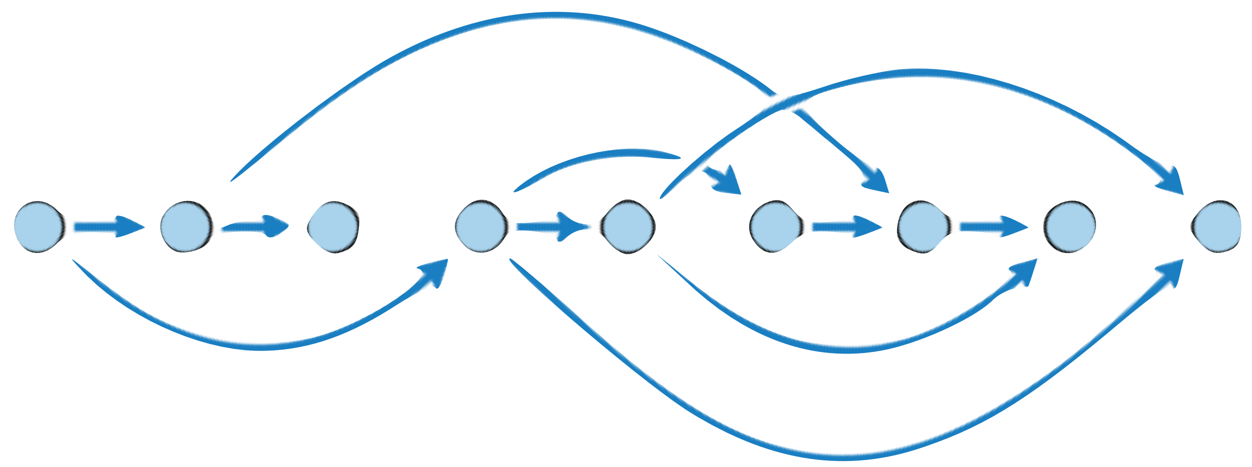
NeuronJ



E. Meijering et al. Cytometry 2004

Dynamic Programming

A directed acyclic graph (DAG) is is a directed graph with no directed cycles
➔ A DAG has a topological ordering of its vertices into a sequence



Dynamic Programming

Viterbi algorithm
Bellman's principle: break this decision problem into smaller subproblems

Building the graph on the pixel image (connectivity)

- Possible paths from start point to end point
- Smoothness aperture Δ controls the possible jump
- Confinement area limitation the search area

DP engine

- Explore the graph in sequence
- Backtrack
- Minimising the sum of costs

Cost a the edge

- from a vertex A to a vertex B
- customise to a specific problem
- cost = weight of edge

DP Optimal Detection of Curves

Powerful and fast optimiser for curve detector in images

- Discretization along the initial line
- Cost: weighted sum of penalisation / attraction
- Easy to dd constraint point

$$\xi_{\mathbf{uv}} = \lambda_1 \frac{f(\mathbf{u})}{N_{int}} + \lambda_2 \frac{|\mathbf{u} - \mathbf{v}|}{N_s}$$

↑ data term

↑ regularisation term