Shortest path problems

Lecture 2
How to compute the intrinsic metric?

- So far, we represented $X$ itself.
- Our model of non-rigid shapes as metric spaces $(X, d_X)$ involves the intrinsic metric

$$d_X(x, x') = \min_{\Gamma(x, x')} \int_{\Gamma} d\ell$$

- **Sampling** procedure requires $d_X$ as well.
- We need a tool to compute geodesic distances on $X$. 
Shortest path problem
Shapes as graphs

- **Sample** the shape at $N$ vertices $X = \{x_1, ..., x_N\}$.
- Represent shape as an **undirected graph**

$$G = (X, E)$$

- $E \subseteq X \times X$ set of **edges** representing adjacent vertices.
- Define **length function** $L : E \to \mathbb{R}$ measuring local distances as Euclidean ones,

$$L(x_i, x_j) = \|x_i - x_j\|_2$$
Shapes as graphs

- **Path** between $x_i, x_j \in X$ is an ordered set of connected edges

$$
\Gamma(x_i, x_j) = \{e_1, e_2, ..., e_k\} \subset E
= \{(x_{i_1}, x_{i_2}), (x_{i_2}, x_{i_3}), ..., (x_{i_{k-1}}, x_{i_k}), (x_{i_k}, x_{i_{k+1}})\}
$$

where $x_{i_1} = x_i$ and $x_{i_{k+1}} = x_j$.

- **Path length** = sum of edge lengths

$$
L(\Gamma(x_i, x_j)) = \sum_{n=1}^{k} L(e_n) = \sum_{n=1}^{k} L(x_{i_n}, x_{i_{n+1}})
$$
Geodesic distance

- **Shortest path** between $x_i, x_j \in X$

$$\Gamma^*(x_i, x_j) = \arg \min_{\Gamma(x_i,x_j)} L(\Gamma(x_i, x_j))$$

- **Length metric** in graph

$$d_L(x_i, x_j) = \min_{\Gamma(x_i,x_j)} L(\Gamma(x_i, x_j))$$

- Approximates the **geodesic distance** $d_X \approx d_L$ on the shape.

- **Shortest path problem**: compute $\Gamma^*(x_i, x_j)$ and $d_L(x_i, x_j)$ between any $x_i, x_j \in X$.

- *Alternatively*: given a **source point** $x_0 \in X$, compute the distance map $d(x_i) = d_L(x_0, x_i)$. 
Bellman’s principle of optimality

Let $\Gamma^*(x_i, x_j)$ be the shortest path between $x_i, x_j \in X$ and $x_k \in \Gamma^*(x_i, x_j)$ a point on the path.

Then, $\Gamma(x_i, x_k)$ and $\Gamma(x_k, x_j)$ are shortest sub-paths between $x_i, x_k$, and $x_k, x_j$.

Suppose there exists a shorter path $\Gamma'(x_i, x_k)$.

$$L(\Gamma'(x_i, x_j)) = L(\Gamma'(x_i, x_k)) + L(\Gamma(x_k, x_j))$$
$$< L(\Gamma(x_i, x_k)) + L(\Gamma(x_k, x_j)) = L(\Gamma^*(x_i, x_j))$$

Contradiction to $\Gamma^*(x_i, x_j)$ being the shortest path.
Dynamic programming

- How to compute the **shortest path** between source \( x_0 \) and \( x_i \) on \( X \)?
- **Bellman principle**: there exists \( x_j \in \mathcal{N}(x_i) \) such that
  \[
  d_L(x_0, x_i) = L(x_0, x_j) + d_L(x_j, x_i)
  \]
- \( x_j \) has to **minimize path length**
  \[
  d_L(x_0, x_i) = \min_{x_j \in \mathcal{N}(x_i)} \{ L(x_0, x_j) + d_L(x_j, x_i) \}
  \]
- Recursive **dynamic programming equation**.
Numerical geometry of non-rigid shapes  Shortest path problems

Edsger Wybe Dijkstra (1930–2002)

[ˈɛtsəkəˌʃiəˈwbəˈdiəkstra]
Dijkstra’s algorithm

- Initialize $d(x_0) = 0$ and $d(x_i) = \infty$ for the rest of the graph;
- Initialize queue of unprocessed vertices $Q = X$.

- While $Q \neq \emptyset$
  - Find vertex with smallest value of $d$,
    $$x = \arg \min_{x \in Q} d(x)$$
  - For each unprocessed adjacent vertex $x' \in \mathcal{N}(x) \cap Q$,
    $$d(x') = \min\{d(x'), d(x) + L(x, x')\}$$
  - Remove $x$ from $Q$.
- Return distance map $d(x_i) = d_L(x_0, x_i)$.
Dijkstra’s algorithm
Dijkstra’s algorithm – complexity

- While there are still unprocessed vertices
  - Find and remove minimum
  - For each unprocessed adjacent vertex
    - Perform update

- Every vertex is processed exactly once: $N$ outer iterations.
- Minimum extraction straightforward complexity: $O(N)$
- Can be reduced to $O(\log N)$ using binary or Fibonacci heap.
- Updating adjacent vertices is in general $O(|V|) = O(|E|)$.
- In our case, graph is sparsely connected, update in $O(1)$.
- Total complexity: $O(N \log N)$.
Troubles with the metric

- Grid with 4-neighbor connectivity.
- True Euclidean distance
  \[ d_{\mathbb{R}^3} = \sqrt{2} \]
- Shortest path in graph (not unique)
  \[ d_L = 2 \]
- Increasing sampling density does not help.
Metrization error

4-neighbor topology  
Manhattan distance

\[ d_{L_1} = \sum_i |x^i_1 - x^i_2| \]

8-neighbor topology  
Continuous \( \mathbb{R}^2 \)

Euclidean distance

\[ d_{L_2} = \sqrt{\sum_i (x^i_1 - x^i_2)^2} \]

- **Graph representation** induces an **inconsistent metric**.
- Increasing **sampling size** does not make it consistent.
- Neither does increasing **connectivity**.
Metrification error

How to approximate the metric consistently?

\[ \lim_{r \to 0} d_L = d_{\mathbb{R}^2} \]

**Solution 1**

- Stick to **graph** representation.
- Change **connectivity** and **sampling**.
- Under certain conditions consistency is guaranteed.

**Solution 2**

- Stick to given **sampling** (and **connectivity**).
- Compute distance map on a **surface** in some representation (e.g., **mesh**).
- Requires a new algorithm.