Hierarchical path planning for walking (almost) anywhere

FOSS4G 2018, Dar es Salaam
Knut Landmark, Espen Messel
Introduction and overview

- Problem and goal
- Prospective applications
- Open data (vs. restricted)
- Graph partitions, hierarchical method
- Accuracy
- Cost of walking

Detailed National Elevation Model, http://kartverket.no/ (Norwegian Mapping Authority)
Problem and goal:

• Find route between two arbitrary land area positions.
• Take into account
  • transport networks
  • topography
  • terrain type, land cover/use
  • infrastructure.
• Exploit open-access LiDAR data.
• Arbitrary distance and resolution.
Open-access LiDAR data

- National initiatives
- All-covering aerial LiDAR survey projects
- Open access policy
- "Transparency, efficiency, innovation"
- UK:
  - UK Environmental Agency
  - Open Government Licence
  - https://data.gov.uk/
- Norway:
  - Norwegian Mapping Authority
  - National Detailed Elevation Model
  - Creative Commons 4.0
  - https://hoydedata.no/LaserInnsyn/
LiDAR-based DSMs and DEM enable realistic path planning
Applications and prospects

- Archaeology, ancient networks
- Search and rescue
- Public transit planning
- Forestry
- Military operations
- Hiking, exercise, recreation
- Robotics, autonomous systems
- Animal migration patterns
- etc.
Two challenges in cross-country path planning

• Realism, how to adapt data
  – Accurate weight (cost) function
  – Is a stream traversible; if so, at what cost?
  – Answer not always found in data.

• Computational:
  – Manage very large graphs
  – Find optimum solutions efficiently
Two challenges in cross-country path planning

• Realism, how to adapt data
  – Accurate weight (cost) function
  – Is a stream traversible; if so, at what cost?
  – Answer not always found in data.

• Computational:
  – Manage very large graphs
  – Find optimum solutions efficiently

Image: www.flickr.com
Road network routing and graphs

• Natural hierarchical structure
• Overlays, multiple levels
• Transit nodes
• Table look-ups
• Contractions
• Orders of magnitude faster than Dijkstra

Delling and Werneck (2013)
Matrix-graph duality

\[ G = \begin{bmatrix} 0 & w_{12} & 0 & w_{14} \\ w_{21} & 0 & w_{23} & w_{24} \\ 0 & w_{32} & 0 & 0 \\ 0 & 0 & w_{43} & 0 \end{bmatrix} \]

\[ v_i = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \]

Node index \( i \) row \( i \)
Matrix-graph duality (no parallel arcs)

- Breadth-first search ↔ matrix multiplication
- Sparse matrix representation
- Adjacency matrix
- Vectorization
- Array-based software
Assumptions

- Graph $G = (V, E, w)$, $V = \{ v_1, v_2, \ldots, v_N \}$
- Directed
- Positive weight
- No self-loops
- Connected (consider each component by itself)
- (Simple)
- Spatial position $x(v)$
Generalized path planning with graphs

- No natural hierarchical structure
- Additional degree(s) of freedom
- High node density (almost) everywhere
- Finite position accuracy
What is accuracy?

1. Node position close to true position.
2. Length of graph solution $\approx$ length of real (actual) optimum path.
   - Accuracy is an issue in all graph-based approaches!
   - Discrete model allows finite set of positions, movements
Simple strategy

- Very detailed basal graph (eliminates position uncertainty)
- Restrict search to small part of basal graph
• Must reduce search space.
• Subgraphs may become disconnected.
• Consequently, no solution is found.
Aggregate nodes
Modified strategy

- For a graph $G_0$ with $N$ nodes, partition $G$ into $K < N$ connected components $U_1, U_2, \ldots, U_K$.
- Form a simplified graph $G$ with nodes $V(G_1) = \{ U_1, U_2, \ldots, U_K \}$.
- Control the “size” (graph-theoretical radius) of components (accuracy!).
- Expand a solution in $G_1$ to obtain a reduced search space (subgraph) in $G_0$.
- Can be iterated several times to form hierarchy of simplified graphs, $\mathcal{G} = \{ G_0, G_1, \ldots, G_h \}$.
- A solution in $G_0$ is found (if it exists).
Complexity-accuracy trade-off

- As small radius \((R)\) as possible
- As few components \((K)\) as possible
**k-center problems**

- Find partition $V(G) = \bigcup_{k=1}^{K} U_k$, each $G|_{U_k}$ is (strongly) connected, and
  - C1: Minimum $K$ such that $\max_{1 \leq k \leq K} r(U_k) \leq R$ for fixed radius $R$.
  - C2: Smallest radius $R$ such that $\max_{1 \leq k \leq K} r(U_k) \leq R$ for a fixed $K$.
- NP-hard optimization: $k$-center problems
**k-centre algorithm**

Find $K$ cluster representatives (CRs) $H = \{ h_1, \ldots, h_K \}$ and assign nodes to their nearest CR.

- Start with one random CR, $H = \{ h_1 \}$.
- Compute distance from $h_1$ to all nodes.
- Add the node farthest from present clusters.
- Recompute cluster assignments for all nodes.
**k-center algorithm**

- Requires at most $K - \max$ single-source all-shortest-path computations.
- In each iteration, can store distance from new CR to its vicinity (in $K - \max \times N$ sparse matrix).
- From this we obtain edge weights in simplified graph.
The Dijkstra wavefront
1 Modified Dijkstra loop

\[ s \leftarrow \{ \text{source node} \} \]

\[ N \leftarrow \text{number of nodes} \]

\[ \Lambda \leftarrow \text{cut-off distance} \]

\[ c \leftarrow 0 \]

\[ U \leftarrow \{ \text{all nodes} \} \]

\[ \textbf{while} \ U \ \text{not empty} \ \textbf{do} \]

\[ u \leftarrow \text{node } v \in U \text{ with minimum distance } d(v) \]

\[ U \leftarrow U \setminus \{ u \} \]

\[ // \text{ Additional work:} \]

\[ c \leftarrow \frac{N}{N-1} c + \frac{1}{N} d(u)/d_{\text{EUCL}}(u) \]

\[ \text{if } d(u) > \Lambda \text{ then} \]

\[ \text{break} \]

\[ // \text{ Update distances:} \]

\[ \text{for all } v \in \{ \text{active neighbors of } u \} \ \textbf{do} \]

\[ d(v) = \min(d(u) + \text{weight}(u,v), d(v)) \]

\[ \text{for all } v \in U \ \textbf{do} \]

\[ d(v) = \max(\Lambda, c^{-1} d_{\text{EUCL}}(v)) \]
Two extreme cases

- For a hierarchical graph $\mathcal{G} = \{ G_0, G_1, \ldots, G_h \}$, accuracy of solution depends on choice of search levels $0 \leq h_1 \leq \ldots \leq h$.

- If $h = 0$, exact solution is found on complete basal graph (at high cost).

- If start and end nodes are adjacent in $G_h$, the distance uncertainty at level $h$ has the same magnitude as the true distance.
Accuracy revisited

What is accuracy?

1. Length of hierarchical solution $\approx$ length of basal graph solution
2. Length of basal graph solution $\approx$ length of real (actual) optimum path

Accuracy is an issue in all graph-based approaches!

- Discrete model implies finite set of positions, movements
- Hierarchical approach allows arbitrarily fine basal mesh
- Requires careful selection of levels
- Start node $s$, target node $t$
- Basal graph distance $d_G$
- Euclidean distance $d_{EUCL}$
- Scaling factor
- Number of sets in path
- Partition set diameter

$$d_{EUCL}(s, t) \leq d_G(s, t) \leq M \cdot \text{diam}U$$
Walking patterns

- Uphill: extrinsic
- Downhill: concentric
- Level and moderate slopes: inverted pendulum

Lobet, Detrembleur & Hermans (2013)
Energetics: Power balance

- Kinetic energy $E_{\text{kin}}$
- Mass $m$
- Tangential speed $v$
- Inclination (slope) $\alpha$
- Locomotive power $P_{\text{loc}}$
- Friction coefficient $\mu$
- Normal force $F_N$
- Gravitational $g = 9.81 \, m/s^2$
- Cross sectional area $A$
- Air density $\rho$

\[
\frac{dE_{\text{kin}}}{dt} = m \frac{dv}{dt} = P_{\text{loc}} - \mu F_N v - mg v \sin \alpha - \frac{1}{2} \rho C_d A v^3
\]

- Friction power
- Gravitational power
- Air resistance (here: negligible)
Metabolic power and Pandolf equation (rewritten)

- Metabolic energy per unit mass $E$
- Terrain coefficient $\eta$
- Correction term (C.T.) for downhill slopes (Santee et al., 2003)
- Metabolic power per unit mass

$$\frac{dE}{dt}(v; \eta, \alpha) = 1.5 \frac{W}{kg} + \eta \frac{1.5 v}{1 s} + 3.6 g \sin \alpha + \text{C.T.},$$

- "Energy cost of standing still"
- "Energy cost of level walking"
- "Gravitational power"
- Gross energy cost per unit distance $C_g = \left[ \frac{dE}{dt} \right] / v$. 

FFI
Weight function (example)

- Metabolic energy per unit mass and distance
- Any path is traversible in both directions
- Connected components are strongly connected
- Symmetric with respect to ground condition
- Asymmetric with respect to slope
Data

Open data:
- LiDAR: Creative Commons 4.0 (No), Open Government (UK), etc.
- OpenStreetMap (ODbL)
- ElVeg (roads, free data)

Restricted data:
- Land cover/land use from national competent authorities (attribution)
- 1:1000 – 1:5000 vector data (detailed infrastructure)
Open vs restricted data

- OSM + open remote sensing data yield detailed land cover in some areas (Schultz et al., 2017).
- Dedicated public land cover datasets (e.g. CORINE) generally good but not sufficiently detailed.
- Currently detailed land cover from national competent authorities required.
- However, information from (open) LiDAR data may reduce need for hi-res vector data.
- OSM infrastructure sufficient in rural areas
Going the right way:

- LiDAR processing
- Raster/Vector pre-processing
- Data inspection
- Graph representation
Going the right way:

- Optimum paths
- Processing service (WPS)
- Web application
- Graph construction, partitioning
Conclusion

- Open LiDAR data enable realistic cross-country path planning.
- LiDAR reduces the need for detailed restricted data.
- The hierarchical method can eliminate position uncertainty at low computational cost.
- Requires careful selection of search levels.
- Efficient preprocessing is accomplished with $k$-center algorithm and heuristic Dijkstra.
- Current work to find cost function constrained by biomechanical principles, tuned to empirical data on walking.
References

- Har-Peled (2008), Geometric approximation algorithms, University of Illinois.