Shortest path to a segment and quickest visibility queries

Topi Talvitie


SoCG 2015, Eindhoven
Shortest path queries
Shortest path queries

How should one move from $s$ in order to reach $q$ as soon as possible?
Shortest path queries

How should one move from $s$ in order to reach $q$ as soon as possible?
Shortest path queries

How should one move from $s$ in order to reach $q$ as soon as possible?
Shortest path queries

How should one move from $s$ in order to reach $q$ as soon as possible?
Shortest path queries

Fixed $s$:

Preprocessing: $O(n \log n)$
Query: $O(\log n)$  
[Hershberger & Suri]
Shortest path queries

Preprocessing: $O(n \log n)$
Query: $O(\log n)$

[Hershberger & Suri]
Shortest path queries

Preprocessing: $O(n \log n)$

Query: $O(\log n)$

[Hershberger & Suri]
Shortest path queries

Preprocessing: $O(n \log n)$

Query: $O(\log n)$

[Hershberger & Suri]
Shortest path queries

Preprocessing: $O(n \log n)$
Query: $O(\log n)$  

[Hershberger & Suri]
Shortest path queries

Preprocessing: $O(n \log n)$
Query: $O(\log n)$

[Hershberger & Suri]
Shortest path queries

Preprocessing: $O(n \log n)$

Query: $O(\log n)$

[Hershberger & Suri]
Shortest path queries

Preprocessing: $O(n \log n)$
Query: $O(\log n)$

[Hershberger & Suri]
Shortest path queries

Preprocessing: $O(n \log n)$

Query: $O(\log n)$

[Hershberger & Suri]
Shortest path queries

Preprocessing: $O(n \log n)$

Query: $O(\log n)$

[Hershberger & Suri]
Shortest path queries

Preprocessing: $O(n \log n)$
Query: $O(\log n)$

[Hershberger & Suri]
Shortest path queries

Preprocessing: $O(n \log n)$

Query: $O(\log n)$

[Hershberger & Suri]
Shortest path queries

Preprocessing: $O(n \log n)$

Query: $O(\log n)$

[Hershberger & Suri]
Shortest path queries

Preprocessing: $O(n \log n)$

Query: $O(\log n)$

[Hershberger & Suri]
Shortest path queries

Fixed $s$:

Preprocessing: $O(n \log n)$
Query: $O(\log n)$

[Hershberger & Suri]
Shortest path queries

How should one move from $s$ in order to reach $q$ as soon as possible?
Quickest visibility paths

How should one move from $s$ in order to see $q$ as soon as possible?
Quickest visibility paths

Fixed $s$:

Preprocessing: $P = ?$

Query: $Q = ?$
Quickest visibility paths

Fixed $s$:

Preprocessing: $P = ?$

Query: $Q = ?$
Quickest visibility paths

Fixed $s$:

Preprocessing: $P = ?$

Query: $Q = ?$
Quickest visibility paths

Fixed $s$:

Preprocessing: $P = ?$
Query: $Q = ?$
Quickest visibility paths

Fixed $s$:

Preprocessing: $P = ?$

Query: $Q = ?$
Quickest visibility map

QVM = subdivision of the domain into cells such that QVP from $s$ to $q$ is the same for all $q$ in that cell.
Quickest visibility map

$\text{QVM} =$ subdivision of the domain into cells such that QVP from $s$ to $q$ is the same for all $q$ in that cell.

Lower bound on worst-case complexity: $\Omega(n^4)$
Quickest visibility map

QVM = subdivision of the domain into cells such that QVP from \( s \) to \( q \) is the same for all \( q \) in that cell.

Lower bound on worst-case complexity: \( \Omega(n^4) \)
QVM = subdivision of the domain into cells such that QVP from $s$ to $q$ is the same for all $q$ in that cell.

Quickest visibility map
QVM = subdivision of the domain into cells such that QVP from s to q is the same for all q in that cell.
Quickest visibility map

$QVM = \text{subdivision of the domain into cells such that QVP from } s \text{ to } q \text{ is the same for all } q \text{ in that cell.}$

Lower bound on worst-case complexity: $\Omega(n^4)$
Quickest visibility map

QVM = subdivision of the domain into cells such that QVP from \( s \) to \( q \) is the same for all \( q \) in that cell.

Lower bound on worst-case complexity: \( \Omega(n^4) \)

Fixed \( s \):

Preprocessing: \( P = O(n^8 \log n) \)
Query: \( Q = O(\log n) \)
Size: \( S = O(n^7) \)
Find $a \in L_1$, $b \in L_2$, $c \in L_3$ such that $(a, b, c)$ is an arithmetic sequence.
Conjecture \[ P + nQ = \Omega(n^2). \]

Find \( a \in L_1, b \in L_2, c \in L_3 \) such that \((a, b, c)\) is an arithmetic sequence.

Using quickest visibility paths: \( O(P + nQ) \) time.

Conjecture \( \Rightarrow P + nQ = \Omega(n^2). \)
Curse of visibility: 3SUM hardness

\[ |L_1| = |L_2| = |L_3| = n. \]

Find \( a \in L_1, b \in L_2, c \in L_3 \) such that \((a, b, c)\) is an arithmetic sequence.

Using quickest visibility paths: \( O(P + nQ) \) time.

Conjecture \( \Rightarrow P + nQ = \Omega(n^2) \).
Curse of visibility: 3SUM hardness

\[ |L_1| = |L_2| = |L_3| = n. \]

Find \( a \in L_1, \ b \in L_2, \ c \in L_3 \) such that \((a, b, c)\) is an arithmetic sequence.

Using quickest visibility paths: \( O(P + nQ) \) time.

Conjecture \( \Rightarrow P + nQ = \Omega(n^2) \).
|\text{Curse of visibility: 3SUM hardness}|

$$|L_1| = |L_2| = |L_3| = n.$$  
Find $a \in L_1$, $b \in L_2$, $c \in L_3$ such that $(a, b, c)$ is an arithmetic sequence. (3SUM hard)

Using quickest visibility paths: $O(P + nQ)$ time.

Conjecture $\Rightarrow P + nQ = \Omega(n^{2-\epsilon})$ for $\epsilon > 0$.

[Grønlund, Pettie 2014]: $O(n^2 \left(\frac{\log \log n}{\log n}\right)^2)$
Visibility polygon
Visibility polygon
Visibility polygon
Visibility polygon
Visibility polygon

Find the visibility polygon of any \( q \):

Simple polygons:

<table>
<thead>
<tr>
<th></th>
<th>Preprocessing</th>
<th>Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Joe, Simpson 1987]</td>
<td>–</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>[Guibas, Motwani, Raghavan 1997]</td>
<td>( O(n^3 \log n) )</td>
<td>( O(\log n + K) )</td>
</tr>
<tr>
<td>[Bose, Lubiw, Munro 2002]</td>
<td>( O(n^3 \log n) )</td>
<td>( O(\log n + K) )</td>
</tr>
<tr>
<td>[Aronov, Guibas, Teichman, Zhang 2002]</td>
<td>( O(n^2 \log n) )</td>
<td>( O(\log^2 n + K) )</td>
</tr>
</tbody>
</table>

Polygons with holes:

<table>
<thead>
<tr>
<th></th>
<th>Preprocessing</th>
<th>Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Heffernan, Mitchell 1995]</td>
<td>–</td>
<td>( O(n + h \log h) )</td>
</tr>
<tr>
<td>[Zarei, Ghodsi 2008]</td>
<td>( O(n^3 \log n) )</td>
<td>( O(K + \min(h, K) \log n) )</td>
</tr>
<tr>
<td>[Inkulu, Kapoor 2009]</td>
<td>( O(n^2 \log n) )</td>
<td>( O(K \log^2 n) )</td>
</tr>
<tr>
<td>[Lu, Yang, Wang 2011]</td>
<td>( O(n + h^2 \log h) )</td>
<td>( O(K + \log^2 n + h \log(n/h)) )</td>
</tr>
<tr>
<td>[Chen, Wang 2013]</td>
<td>( O(n^2 \log n) )</td>
<td>( O(K + \log^2 n + \min(h, K) \log n) )</td>
</tr>
<tr>
<td>[Chen, Wang 2013]</td>
<td>( O(n + h^2 \log h) )</td>
<td>( O(K \log n) )</td>
</tr>
</tbody>
</table>

\( K = \) the size of the visibility polygon

\( h = \) the number of holes
# Visibility polygon

Find the visibility polygon of any \( q \):

## Simple polygons:

<table>
<thead>
<tr>
<th></th>
<th>Preprocessing</th>
<th>Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Joe, Simpson 1987]</td>
<td>–</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>[Guibas, Motwani, Raghavan 1997]</td>
<td>( O(n^3 \log n) )</td>
<td>( O(\log n + K) )</td>
</tr>
<tr>
<td>[Bose, Lubiw, Munro 2002]</td>
<td>( O(n^3 \log n) )</td>
<td>( O(\log n + K) )</td>
</tr>
<tr>
<td>[Aronov, Guibas, Teichman, Zhang 2002]</td>
<td>( O(n^2 \log n) )</td>
<td>( O(\log^2 n + K) )</td>
</tr>
</tbody>
</table>

## Polygons with holes:

<table>
<thead>
<tr>
<th></th>
<th>Preprocessing</th>
<th>Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Heffernan, Mitchell 1995]</td>
<td>–</td>
<td>( O(n + h \log h) )</td>
</tr>
<tr>
<td>[Zarei, Ghodsi 2008]</td>
<td>( O(n^3 \log n) )</td>
<td>( O(K + \min(h, K) \log n) )</td>
</tr>
<tr>
<td>[Inkulu, Kapoor 2009]</td>
<td>( O(n^2 \log n) )</td>
<td>( O(K \log^2 n) )</td>
</tr>
<tr>
<td>[Lu, Yang, Wang 2011]</td>
<td>( O(n + h^2 \log h) )</td>
<td>( O(K + \log^2 n + h \log(n/h)) )</td>
</tr>
<tr>
<td>[Chen, Wang 2013]</td>
<td>( O(n^2 \log n) )</td>
<td>( O(K + \log^2 n + \min(h, K) \log n) )</td>
</tr>
<tr>
<td>[Chen, Wang 2013]</td>
<td>( O(n + h^2 \log h) )</td>
<td>( O(K \log n) )</td>
</tr>
</tbody>
</table>

\( K \) = the size of the visibility polygon  
\( h \) = the number of holes
Visibility polygon
Visibility polygon
Visibility polygon

Find quickest visibility path from $s$ to any $q$:

Preprocessing: $P = P_v + P_s = O(n^2 \log n + P_s)$

Query: $Q = Q_v + KQ_s = O(K(\log^2 n + Q_s))$
Shortest path to segment

$q$

$s$
Shortest path to segment

$\text{q}$

$\text{s}$
Shortest path to segment
Shortest path to segment

q

s
Shortest path to segment
Types of shortest paths to segments
Types of shortest paths to segments
Types of shortest paths to segments
Types of shortest paths to segments
Types of shortest paths to segments
Types of shortest paths to segments
Types of shortest paths to segments
Types of shortest paths to segments
Types of shortest paths to segments

Shortest path to the segment $q$ is the shortest of:

- Shortest path to endpoint $q_1$,
- Shortest path to endpoint $q_2$,
- Shortest orthogonal path to the interior of $t$. 
Types of shortest paths to segments

Shortest path to the segment $q$ is the shortest of:

- Shortest path to endpoint $q_1$,
- Shortest path to endpoint $q_2$,
- Shortest *orthogonal* path to the interior of $t$. 
Wavefront propagation algorithm
Wavefront propagation algorithm
Wavefront propagation algorithm
Wavefront propagation algorithm
Wavefront propagation algorithm
Wavefront propagation algorithm
Wavefront propagation algorithm
Wavefront propagation algorithm
Wavefront propagation algorithm
Wavefront propagation algorithm
Wavefront propagation algorithm
Wavefront propagation algorithm
Wavefront propagation algorithm
Wavefront propagation algorithm
Wavefront propagation algorithm
Wavefront propagation algorithm
Wavefront propagation algorithm
Wavefront propagation algorithm
Wavefront propagation algorithm
Wavefront propagation algorithm
Wavefront propagation algorithm
Wavefront propagation algorithm
Wavefront propagation algorithm
Wavefront propagation algorithm
Critical times
Critical times
Critical times
Critical times
Critical times
Critical times
Critical times
Critical times
Critical times
Critical times
Critical times
Critical times
Critical times
Critical times
Critical times
Critical times
Critical times
Critical times
Critical times
Critical times
Critical times
Critical times
Critical times
Critical times
Critical times
Critical times
Critical times
Critical times
Critical times
Critical times
Critical times
Critical times
Critical times
Critical times
Critical times
Critical times
Finding the previous critical time
Finding the previous critical time
Finding the previous critical time
Finding the previous critical time
Finding the previous critical time

Find out whether segment $q$ lies in the geodesic disk of the critical time using ray shooting queries:

Preprocessing: $O(n \log n)$  [Chazelle, Edelsbrunner, Grigni, Guibas, Hershberger, Sharir, Snoeyink 1994]
Query: $O(\log n)$
Find out whether segment $q$ lies in the geodesic disk of the critical time using ray shooting queries:

Preprocessing: \( O(n \log n) \)  
Query: \( O(\log n) \)

[Chazelle, Edelsbrunner, Grigni, Guibas, Hershberger, Sharir, Snoeyink 1994]
Finding the colliding wavelet
Finding the colliding wavelet
Finding the colliding wavelet
Finding the colliding wavelet
Finding the collision point
Finding the collision point
Finding the collision point
Finding the collision point

$q$
Finding the collision point
Finding the collision point
Finding the collision point
Finding the collision point
Finding the collision point
Finding the collision point
Finding the collision point

$q$
Finding the collision point
Complexities

<table>
<thead>
<tr>
<th>Shortest path to segment query</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preprocessing: $P_s = O(n^2 \log n \ 2^{\alpha(n)})$</td>
</tr>
<tr>
<td>Query: $Q_s = O(\log^2 n)$</td>
</tr>
</tbody>
</table>
Complexities

**Shortest path to segment query**

Preprocessing: \( P_s = O(n^2 \log n \ 2^{\alpha(n)}) \)

Query: \( Q_s = O(\log^2 n) \)

**Quickest visibility path query:**

Preprocessing: \( P = O(n^2 \log n + P_s) \)

\[ = O(n^2 \log n \ 2^{\alpha(n)}) \]

Query: \( Q = O(K(\log^2 n + Q_s)) \)

\[ = O(K \log^2 n) \]

\((K = \text{the size of the visibility polygon})\)
Simple polygons

**Shortest path to segment query**

Preprocessing: \( P'_s = O(n) \)

Query: \( Q'_s = O(\log n) \)

\[\Downarrow\]

**Quickest visibility path query:**

Preprocessing: \( P' = O(n) \)

Query: \( Q' = O(\log n) \)
### Simple polygons

<table>
<thead>
<tr>
<th>Shortest path to segment query</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Preprocessing:</strong> $P'_s = O(n)$</td>
</tr>
<tr>
<td><strong>Query:</strong> $Q'_s = O(\log n)$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quickest visibility path query:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Preprocessing:</strong> $P' = O(n)$</td>
</tr>
<tr>
<td><strong>Query:</strong> $Q' = O(\log n)$</td>
</tr>
</tbody>
</table>

Previously $P' = O(n^2)$ [Khosravi, Ghodsi 2005]
Thank you

Check out the visualization applet:

Visualizing Quickest Visibility Maps
SoCG Multimedia 2015
http://dy.fi/xwj