Algorithms for GIS:
Computing visibility on terrains
Visibility on terrains

- Are two points (on a terrain) visible to each other?
- What can one see from a given point (on a terrain)?
- How much does the visible area increase if we stand on a 10ft ladder?
- What is the point with largest visibility?
- What is the point with lowest visibility?
- How to place an ugly pipe in a scenic area?
- How to place a scenic highway?
- What is the cumulative visible area from these set of cell towers?
- Find a set of tower locations to cover the terrain
- ...
Visibility on terrains

Problem:

- Terrain T + viewpoint v
- Compute the **viewshed** of v: the set of points in T visible from v
Visibility on terrains

Input: terrain model (DEM = digital elevation model)
- grid
- TIN (triangulation)

Output: viewshed model
- grid elevation model ==> grid viewshed
- TIN elevation model ==> TIN viewshed
Visibility on grid terrains

Sierra Nevada, 30m resolution
(u,v) visible iff segment uv does not intersect T

uv is called line-of-sight (LOS)
Basic viewshed algorithm

Input: elevation grid
Output: visibility grid, each point marked visible/invisible

- For each p in grid
  - compute intersections between vp and grid lines
  - if all these points are below vp then p is visible
Basic viewshed algorithm

Input: elevation grid
Output: visibility grid, each point marked visible/invisible

- For each $p$ in grid
  - compute intersections between $vp$ and grid lines
  - if all these points are below $vp$ then $p$ is visible
Basic viewshed algorithm

Input: elevation grid
Output: visibility grid, each point marked visible/invisible

- For each p in grid
  - compute intersections between vp and grid lines
  - if all these points are below vp then p is visible

Assume grid of \( n \) points
(\( \sqrt{n} \times \sqrt{n} \))
Running time: \( O(n\sqrt{n}) \)
\[ y_1 = d_1 \tan a \]
\[ y_2 = d_2 \tan a \]
Viewshed on grids

• The straightforward $O(n \sqrt{n})$ algorithm
  • uses linear interpolation
  • “exact” as much as data allows

• Better?

• Van Kreveld, using different model
  • nearest neighbor interpolation
  • $O(n \log n)$

Grid of $n$ points: $\sqrt{n} \times \sqrt{n}$
Grids with linear interpolation
Grids with nearest neighbor interpolation

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>23</td>
<td>25</td>
<td>26</td>
<td>32</td>
<td>46</td>
</tr>
<tr>
<td>21</td>
<td>20</td>
<td>24</td>
<td>28</td>
<td>41</td>
<td>46</td>
</tr>
<tr>
<td>24</td>
<td>21</td>
<td>23</td>
<td>31</td>
<td>36</td>
<td>36</td>
</tr>
<tr>
<td>23</td>
<td>22</td>
<td>24</td>
<td>27</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>32</td>
<td>22</td>
<td>29</td>
<td>30</td>
<td>35</td>
<td>34</td>
</tr>
<tr>
<td>29</td>
<td>30</td>
<td>33</td>
<td>34</td>
<td>36</td>
<td>37</td>
</tr>
</tbody>
</table>
vertical slope(p,a) = (h_p - h_a) / d(a,p)
Van Kreveld’s radial sweep algorithm
Van Kreveld’s radial sweep algorithm
Van Kreveld’s radial sweep algorithm
Van Kreveld’s radial sweep algorithm
Accuracy!!

with ioradial from
Fishman et al 2009

with GRASS