Re: 2D height map to 3D model?

Source: http://coding.derkeiler.com/Archive/Python/comp.lang.python/2004−10/1819.html

---

From: Mike C. Fletcher (*mcfletch_at_rogers.com*)
Date: 10/11/04

```
Date: Mon, 11 Oct 2004 10:37:24 −0400
To: python−list@python.org
```

Didn't see the original question, but going from the subject, you want to generate a 3D mesh from a height field (such as seen in an image). The algorithm for doing that is fairly straightforward:

```
rectangles = []
for m in range(dim1−1):
   for n in range(dim2−1):
      vertices = []
      vertices.append( (m,n,heights[m,n]))
      vertices.append( (m+1,n,heights[m+1,n]))
      vertices.append( (m+1,n+1,heights[m+1,n+1]))
      vertices.append( (m,n+1,heights[m,n+1]))
      rectangles.append( vertices )
```

However, that dramatically increases the size of your geometry in memory (you're storing 12 doubles for almost every data−point). It's easier to use a format where you define vertices and a separate topology (via indices into the vertices). Same basic approach works there, you just have to add m+(n*dim1) to get the index for a corner of the quad. There you're storing only 3 doubles for each vertex.

If you have a format that allows for triangle/quadrilateral strips, you can make the rendering far more efficient using them. There you render (m,n), (m+1,n), (m+1,n+1), (m,n+1), (m+1,n+2), (m,n+2), (m+1,n+3),... That reduces the size of your index−set as well, but most of the speedup is going to come from having fewer primitive operations.

Good luck,
Mike

PhilC wrote:

>Thanks Richard,
>
>I was actually thinking of faces but an aerial photograph would be
>similar. I'll check through those links and see if they help.
>

>*Again my appreciation for your reply.*
>
>*PhilC*
>
>
>

--

Mike C. Fletcher
Designer, VR Plumber, Coder
http://www.vrplumber.com
http://blog.vrplumber.com