

Hillshade Video Script

Nathan Stouffer

1 Intro

1.1 Motivation

What I'm showing you right now is a map. But you probably didn't need to be told that. In fact, I would be willing to bet that you know a lot more than that. You can probably see that this particular spot is relatively flat and that this other area is pretty steep, that this is a small gully and this is a ridgeline, and that this face is pretty rugged while this slope is not quite as technical.

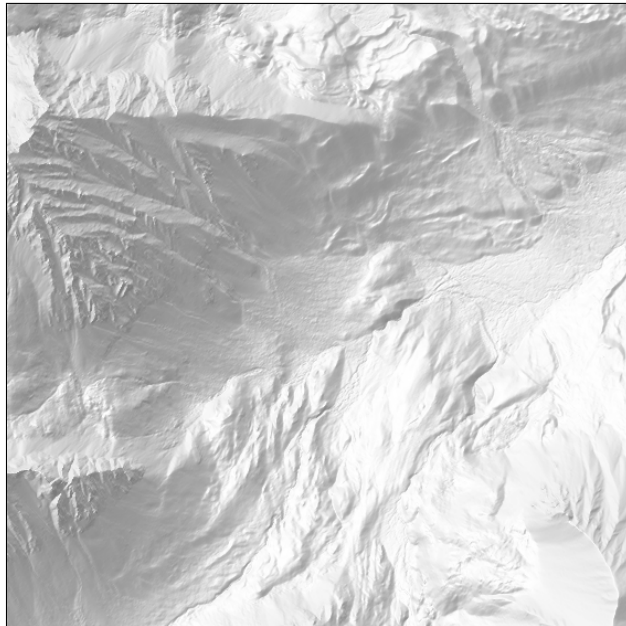
You are able to get all that complex information from a pretty simple grayscale image. In fact, I would be willing to bet that your understanding of this map is better than if I had given you the contour lines even though that is a more precise way to convey information.

[ANIMATION: Bring in more maps \(one with just contours and another with contours/hillshade\)](#)

Here is a side-by-side to compare.

This map is intentionally set up so that your brain can intuit the shape of the terrain. Modern computers do much of this work today, but cartographers have been using versions of this technique for hundreds of years. And many other applications use similar strategies to help convey geometric information to your brain.

The topic I'd like to discuss today is a shading technique called hillshade. We're going to talk about why it's such an effective method of terrain lighting and walk through the math that powers it.



1.2 Directional Lighting

Hillshading is an example of something called a directional light. Directional lights are imitations of

2 Directional Lighting

2.1 Light Direction

2.2 Surface Normal

3 Hillshading

3.1 Cosine

TODO: We can derive $\cos \theta$ by computing the percentage of directional light that hits a small square based on its surface normal. That follows pretty directly from the definition of \cos .

TODO: This is probably the spot to remap $[-1, 1]$ to $[0, 1]$. It is a small piece of the puzzle but this might be a good spot to introduce it.

3.2 Law of Cosines => Dot Product

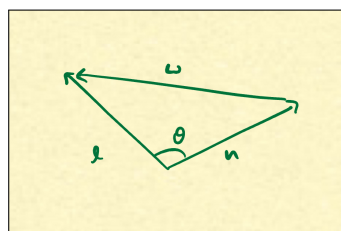
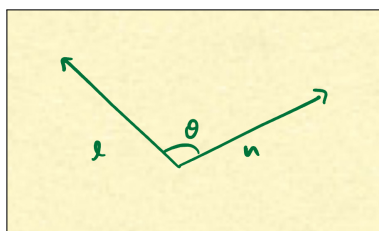
We now know that we want to vary the strength of our lighting according to $\cos \theta$. But that doesn't really help us much because we don't know what θ is. All we know are the vectors l and n . At the end of the day, those are just lists of numbers. Sure, they have some constraints (like the fact that their length is 1). And they also have some geometric meaning (they are directions in 3d space). But when it comes down to it, we need to turn two lists of numbers into something as complex as the cosine function. How are we supposed to do it? What is the link between these two vectors and cosine?

If you were some early mathematician, at this point in the proof, you would have to start experimenting with ideas and wait for inspiration to strike. This sort of thing takes practice and lots of dead ends to feel out. It's sort of one of those things where experience is the best teacher.

... But if I were to offer one piece of advice, I would say it's often worth constructing a triangle and using some of the many theorems about them to reason your way towards your goal. Maybe it's the types of problems that I tend to interact with, but I find triangles to be a surprisingly effective tool. In this case, we are going to draw the third leg of our triangle here and use the Law of Cosines.

We can relate l and n with $\cos \theta$ by using the Law of Cosines. We already have θ labeled and we know $|l| = |n| = 1$ because we have constructed them to be unit vectors. To do this, we need to figure out what the third leg of our triangle is.

Vectors add by being placed tip to tail. This means that the vector Δ , which starts at n and ends at l , is the thing that you can add to n to get l : $l = n + \Delta \Rightarrow \Delta = l - n$.



The full Law of Cosines says that for any triangle with angles/sides labeled like so, we have $c^2 = a^2 + b^2 - 2ab \cos C$. In our case, $c = |l - n|$, $a = |l|$, $b = |n|$, and $C = \theta$. So we have:

$$\begin{aligned} |l - n|^2 &= |l|^2 + |n|^2 - 2|l||n| \cos \theta && \text{substitute from generalized LoC} \\ |l - n|^2 &= 2 - 2 \cos \theta && \text{since we know } |l| = |n| = 1 \end{aligned}$$

To help with this next part, I'm going to introduce some shorthand notation for an operation that we are performing. The operation takes two vectors, performs a pairwise multiplication and sums the entries of the resulting vector. The notation I will use to signify this is a \cdot between two vectors: $a \cdot b$.

If we expand $|l - n|^2$ as $\sqrt{(l_x - n_x)^2 + (l_y - n_y)^2 + (l_z - n_z)^2}$, our new notation comes in handy right away because we can rewrite that big expression as $\sqrt{(l - n) \cdot (l - n)}^2 = (l - n) \cdot (l - n)$.

$$\begin{aligned}
 (l - n) \cdot (l - n) &= 2 - 2 \cos \theta && \text{by using our new notation} \\
 l \cdot l - l \cdot n - n \cdot l + n \cdot n &= 2 - 2 \cos \theta && \text{under the hood, this is happening each dimension and this is the result} \\
 |l|^2 - 2l \cdot n + |n|^2 &= 2 - 2 \cos \theta && \text{reversing our new notation} \\
 1 - 2l \cdot n + 1 &= 2 - 2 \cos \theta && \text{simplyfying} \\
 l \cdot n &= \cos \theta && \text{simplifying}
 \end{aligned}$$

Many of you have seen this result before. For those of you who haven't, that "dot" symbol that represents the operation of pairwise multiplying two vectors and summing the result is known as the dot product. It is an exceedingly useful operation in mathematics and computer science, in large part because of the property that we just proved: that the dot product of two unit vectors is equal to the cosine of the angle between them.

3.2.1 Takeaways

I want you to take away a few things from this proof.

First: The Law of Cosines is true for all triangles – even degenerate triangles (ones where the three points are co-linear or even identical)! So no matter what our two vectors are, we can compute the dot product $l \cdot n$ and it will always be $\cos \theta$!

Second: We didn't really assume that our vector had to be in 3-dimensions. We did only expand our vector for x, y, and z but the same argument will hold for any dimension.

Third: We did assume that our two vectors had unit length. But this relationship also has a more general form. All we need to do is scale $\cos \theta$ by the lengths of the relevant vectors. Given any two vectors $a, b \in \mathbf{R}^n$: $a \cdot b = |a||b| \cos \theta$. It might be a fun exercise for you to build on this proof and show why that is the case.

Finally: I'd like you to recognize how surprising and elegant this result is. I mean, it's crazy that combining two lists of numbers in this really simple way exactly matches $\cos \theta$. Sometimes simple operations are really powerful.

4 Final product

And there you have it.

5 Endnotes

5.1 Modified techniques

Hillshading isn't just one thing. It is actually a term for a family of effects that can be applied to a map. There are a lot of variations out there, and you now have the mathematical framework that sits behind all of them. Some examples include ambient lighting, exaggerating the normal vector, using multiple light sources, and playing around with colored lights.

TODO: Possibly mention Eduard Imhof.

5.2 Pseudoscopic Illusion

In mapping, the light source for hillshade is typically placed in the northwest. This is because many maps use a north-up convention, placing the light source in the top left. Broadly speaking, people recognize features better when this is the case.

However, this can some backfire when a map with static hillshade is oriented with a south-up convention. When that occurs, your brain might interpret everything backwards (valleys as ridges and ridges and valleys). This is called a pseudoscopic illusion.

ANIMATION: Show a south-up map light from the northwest and then fade in the same map (still south up) with a southeast (top left) light.

5.3 2D vs 3D

I'd like to end the video by making one last comment on the simplicity of this effect: this is not a 3-dimensional map. And I don't just mean that the screen you're viewing is 2-dimensional. I mean that the actual model that I am rendering is 2D.

ANIMATION: Pitch the camera

Despite that, it is an incredibly effective method of conveying terrain information. I find that fascinating.

ANIMATION: 3D flythrough?