

# 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 not only recognized this as a map, you also instantly discerned many features about this region. You probably recognized 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, or maybe that this face is pretty rugged while this slope is not quite as technical.

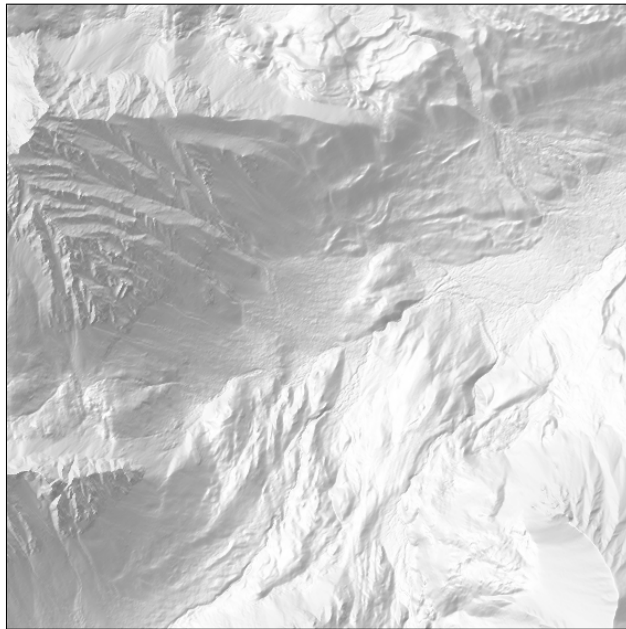
You are able to get all that complex information from a pretty simple grayscale image – despite the fact that there is no precise information like contour lines. In fact, I would be willing to bet that your understanding of this map is better than if I had given you the contour lines even though that 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.

That is all by design. The map is intentionally set up so that your brain can intuit all that information. 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 thing I'd like to discuss today is called hillshading. We're going to walk through the math behind it and talk about why it's such an effective method of lighting terrain.



## 1.2 The Puzzle

### 1.2.1 The Problem

Say you are some cartographer. Your goal is to produce maps that help people understand a particular area. Essentially, your goal is to convey information. You have done research on some area, you know what it looks like. And you want to record that information in such a way that anyone can read it and understand what is going on. The medium that you happen to use is a map. So you want to put all your craft into producing a map that is easy for people to understand.

But, you have a problem. Your map is inherently 2D and the terrain is 3D.

### 1.2.2 The spark

Well, you think, everything people perceive with their eyes is essentially in 2-dimensions anyways. Your brain is really just receiving signals from your eyes and you reconstruct the 3-dimensional object inside your head. What convinces us that things are 3D is the difference in colors as the object moves.

That's really all you need to replicate. And, do you really need to replicate it? Maybe we can just evoke the essence of that lighting and rely on people's brain to fill in the gaps?

## 2 Directional Lighting

### 2.1 Light Direction

### 2.2 Surface Normal

### 2.3 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.

### 2.4 Law of Cosines => Dot Product

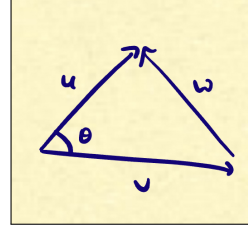
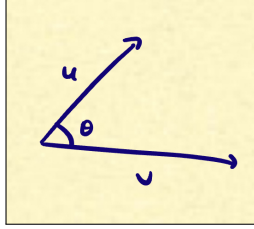
We now know that we want to vary the strength of our lighting according to  $\cos \theta$ . But that doesn't really help us much because we don't know what  $\theta$  is. All we know are the vectors  $l$  and  $n$ . At the end of the day, those are just lists of numbers. Sure, they have some constraints (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  $\theta$  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.

TODO: the wording in this paragraph could use some work: Vectors add by being placed tip to tail. This means that the vector  $w$ , which starts at  $l$  and ends at  $n$ , has the following equality  $l = n + w \Rightarrow w = l - n$ .



The full Law of Cosines says:  $c^2 = a^2 + b^2 - 2ab \cos C$ . In our case,  $c = |l - n|$ ,  $a = |l|$ ,  $b = |n|$ , and  $C = \theta$ . So we have:

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

TODO: introduce the dot product (but don't call it that)

$$\begin{aligned} (l - n) \cdot (l - n) &= 2 - 2 \cos \theta && \text{by definition of dot product} \\ l \cdot l - l \cdot n - n \cdot l + n \cdot n &= 2 - 2 \cos \theta && \text{by distributive property of dot product} \\ 1 - 2l \cdot n + 1 &= 2 - 2 \cos \theta && \text{simplyfying} \\ l \cdot n &= \cos \theta && \text{simplifying} \end{aligned}$$

TODO: mention how the LoC is true for even degenerate triangles

TODO: discuss how we didn't even assume dimension

TODO: discuss the generalized form (non-unit vectors)

TODO: discuss how surprising the dot product identity is

## 3 Hillshade

Hillshading isn't just one thing. It is actually a term for a family of effects that can be applied to a map. We now have the mathematical tools to build towards a very version of hillshading, but there are lots of additional techniques out there.

## 4 Endnotes

### 4.1 Pseudoscopic Illusion

### 4.2 Modified techniques

Modifications: ambient, multiple lights, colored lights

TODO: Possibly mention Eduard Imhof.

### 4.3 2D vs 3D

Something that I find really fascinating is the fact that this isn't even 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

ANIMATION: 3D flythrough?