DOI: 10.14714/CP101.1789

Improving Detail in Shaded Relief

Gene Trantham
gene.trantham@pm.me

The standard "hillshade" tool included in most GIS software suites implements a simple model of lighting with a set of assumptions that make the tool fast and easy to use. This simplified lighting model can visually degrade steep terrains, producing over-dark areas and removing important terrain detail. The underlying model can, however, be manipulated to output displays without these drawbacks. This mimics the effect of ambient light without complicating the lighting model by introducing additional light sources. This article will briefly describe the underpinnings of Lambertian shaders, then demonstrate how the traditions and assumptions built into most GIS tools can be removed to give more flexibility and control over results. Finally, shadows will be discussed as a separate addition to shaded relief.

INTRODUCTION

The stock "hillshade" tool in a typical GIS toolkit operates on a simplified model of light behavior, and generalizes its output to 256 shades of gray. It gives useful results at low computational expense, with an appealingly simple interface. However, critical assumptions built into this tool have the unfortunate side effect of removing detail in steep areas.

Consider the sample scene in Figure 1. Both panels stretch their grayscale ramps across the same range, with no adjustments to brightness or contrast. The top panel is the output of the standard hillshade tool. Note the fully saturated black areas in the steep caldera rim, and also the blackened southeast slope of the island. The bottom panel, with the same lighting conditions and underlying lighting model, includes much more detail in these steep areas. The effect is as if we've added an overhead fill light to help illuminate the scene, so that all parts of the terrain get a little ambient light.

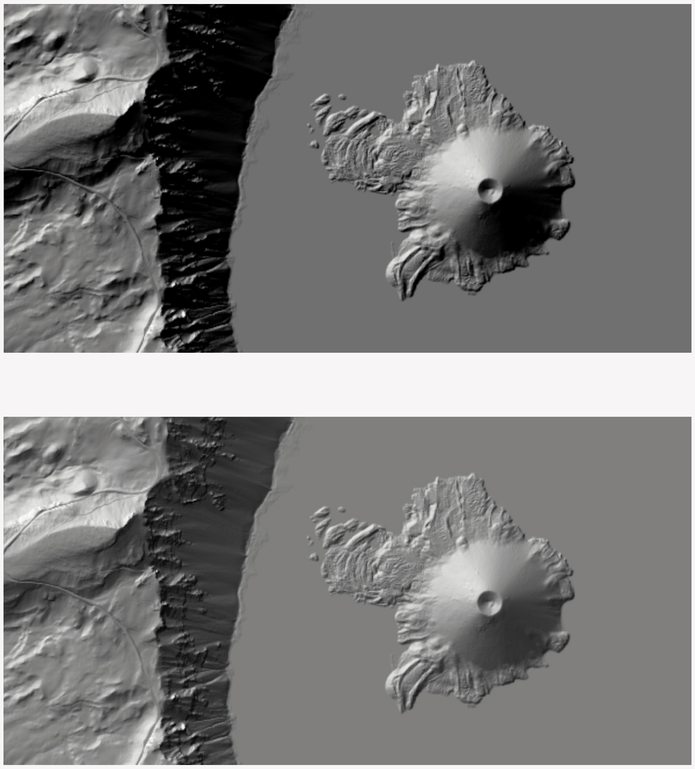


Figure 1. Traditional/standard vs "Half-Lambert" hillshade methods. Light from the northwest at 45° altitude. Note the heavy shadow-mass on the top compared with those same areas on the bottom.

This is an illusion, but a useful one. We can't easily make the standard hillshade algorithm properly model ambient light, but we can manipulate the underlying reflectance model to mimic the effects of ambient light without adding computational load. We're going to look at a few ways to preserve and accentuate detail in areas which are typically shaded as black using the standard hillshade. These methods will not make your shaded relief look like the output from Blender, but they will give you a little more control over the detail available in the darkest corners of the terrain.

To do that, we'll need to take apart the hillshade tool, breaking its function into the separate concepts of shade and shadow. We're going to need to get a little theory under our belts before we start making shaded scenes.

SURFACE REFLECTANCE

The brightness levels of a shaded relief symbology are a way of representing the degree to which a surface reflects light back to a viewer. This *surface reflectance* is modeled based on some set of assumptions about light behavior and the properties of the surface itself, which varies according to the relief method. Light that is scattered or refracted by the atmosphere or light reflected from other objects in the environment is not considered when determining surface reflectance. See Horn (1979; 1981) for a thorough discussion of reflectance in the context of cartography.

The traditional GIS hillshade tool implements a *Lambertian* shader as its foundation to compute surface reflectance. The Lambertian shader models idealized reflectance based on how well aligned the surface is with the light source—and it assumes that all light is reflected, with none absorbed by the surface. A point on the terrain surface will be brightly shaded if it "faces" the light. A point which faces "away" from the light source will have low reflectance and thus a low brightness value (BV). Lambert quantifies this idea of alignment using the *cosine emission law*: The cosine of the angle between a light vector and the surface normal vector is proportional to the BV.[1] The surface normal vector is a unit-length vector emitting from a plane tangent to the surface at a given point, which completely encodes the orientation of the surface at that point. It is roughly synonymous with a combination of the more GIS-familiar measures of slope and aspect.

THEORY VS. PRACTICE

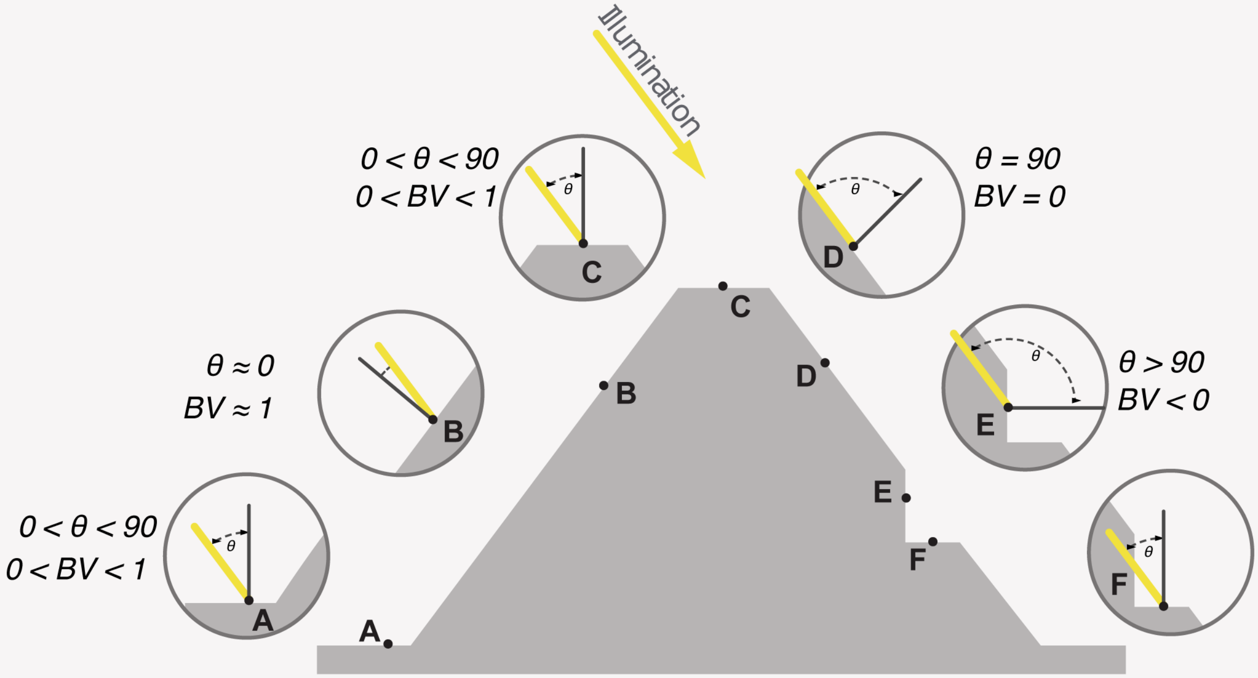
Lambert's cosine emission law provides the theoretical BV for a point on the surface. We will refer to this as the "pure" or "raw" Lambert BV. Because Lambert's brightness value is tied to the cosine function, its BVs fall between [-1, 1] as a floating-point number. However, the GIS hillshading tool you are probably familiar with produces brightness values between 0 and 255 as an 8-bit unsigned integer. The transformation between the raw Lambertian brightness value to the traditional 8-bit hillshade value is important to understand, because we are going to re-wire it to achieve different results without changing the underlying Lambertian shader.

In an unrestricted 3D space, the light and surface normal vectors could be separated by as much as 180 degrees, although this would be unusual in a cartographic context. We are often using DEMs which cannot encode overhanging terrain, and the light is always modeled to come from above the horizon. Let's look at these vectors in the context of a demonstration terrain. Figure 2 is a profile view of a cross section through a roughly pyramidal terrain to illustrate subtle differences in BVs for key points.

High BVs (typically symbolized as bright) are associated with small angles between the surface normal and light vectors. A small angle implies close alignment; the closer θ is to zero, the brighter the BV. In Figure 2, point B is an example of a brightly lit point. Point C has a slightly larger angle, so will be a fraction darker, but still relatively bright.

It is tempting, but misleading, to think of areas with high BV as being *illuminated*. Consider point F, for example. Should that point be bright or dark? Lambert examines only the angle between surface normal and light vectors, so this point has the same surface reflectance as points A and C. The urge to darken point F comes from considering shadows, which is a *viewshed* property, completely different from surface reflectance.

[1] This angle and its cosine can be computed a few different ways, which we're not going to get into. The result, cos(θ), is the BV.



Point	θ	BV = cos(θ)	Notes
A	0 < θ < 90	0 < BV < 1	The closer θ is to zero, the brighter the BV
B	0 < θ < 90	0 < BV < 1	Brighter than point A, as θ is smaller
C	0 < θ < 90	0 < BV < 1	Same brightness as point A
D	θ = 90	BV = 0	θ is exactly 90; cos(θ) = 0
E	θ > 90	-1 < BV < 0	BV is negative
F	0 < θ < 90	0 < BV < 1	Same brightness as points A and C

Figure 2. Profile view of pyramid.

The tendency to conflate shade with shadow is misleading in another important respect—the "self-shaded" points on the surface. Consider points D and E. At D, the angle between light and surface normal vector is exactly 90 degrees, leading to a BV of zero (the cosine of 90° is zero). At E, that angle is larger than 90, leading to a BV below zero.

Negative BVs from Lambert are sometimes described as "self-shaded" areas, where the surface seems to block the light which might fall on that point. However, using terms like "block" and "fall" for light are misleading in this context, because we are not considering the light's viewshed. The point of transition between positive and negative Lambert shade values occurs when the surface normal and light vectors are separated by 90 degrees (e.g., point D).

This transition is sometimes called a *terminator*.[2] It is clear that point E in Figure 2 should be dark, but should it be darker than point D, or should it be shaded the same as point D? Lambert says they should be different, with E being darker than D. The standard hillshade tool you probably use treats these as having the same BV, namely

[2] This terminology is borrowed from astronomy. The line where the dark side transitions to the lit side of the moon, for example, is the terminator. This is a slight misuse of the term, given that it includes both shade and shadow, but it is very useful to describe this transition point, so we will keep it with the understanding of its limitations here.

zero. Specifically, the tool *clamps* Lambert's raw output to discard all negative values (setting those values to zero). The main problem to address with clamping is that it is a *surjective* function, meaning that it is **not reversible**. Many input values (all of the negative pure Lambert outputs) are mapped to zero. There is no way to take all of the hillshade's zero values and recover any information to differentiate points on the dark side of the terminator. Points D and E from Figure 2 are assigned the same BV after clamping—there is no way to recover any difference between these points by "un-clamping." No amount of fiddling with contrast or brightness will reveal any detail in these areas, because they all have the same BV.

If we could keep Lambert's full output range, and manipulate it without discarding values through clamping, we can give a more nuanced representation of these parts of the scene without having to add extra lights and the extra computations they will require. This is the main concept behind what Preppernau (2020) calls a "soft hillshade."

MANIPULATING LAMBERT OUTPUT

Our strategy is going to be to take the Lambert output directly, before the stock hillshade tool clamps and otherwise manipulates the shader values. Unfortunately, the way this tool is built doesn't permit us to intercept shade values at the correct point. We will instead use a simple home-grown Lambert shader to experiment with its results.

The math-curious and persistent reader could certainly build a pure Lambertian shader using existing GIS tools (see Preppernau [2020] for guidance on how that might be done in Python). To keep our focus on what to do with Lambert instead of how to calculate it, I have developed Lambertian shaders for ArcGIS Pro and QGIS which are available for download alongside this article, and which will form the basis of the following examples.

All of the examples here will examine the area around Wizard Island in Crater Lake National Park in Oregon, USA. The data source for this study area is extracted from the Crater Lake sample DEM available at shadedrelief.com/SampleElevationModels (Kennelly et al. 2021).

PURE LAMBERT

Lambert's canonical output, spanning the range [-1, 1], is what we have been calling pure or raw Lambertian brightness values. It is the cosine of the angle θ between the illumination and surface normal vectors. If we apply a grayscale ramp to this output, with our colors spanning the entire theoretical range, the result (in Figure 3) shows maximum detail across the terrain. No areas are flattened into darkness via clamping. Note from the histogram that this shader includes BVs below zero.

This study area and the lighting conditions have been selected to draw attention to the terminator and the terrain behind it. These examples will use an azimuth of 315°, and an elevation of 30°. The result in Figure 3 and the top panel of Figure 1 were both made with the same light conditions, yet they look quite different. The cause of this difference is the effect that clamping (removing detail) has on the range used on the final color ramp.

The histogram of BVs in the pure Lambert output in Figure 3 shows a bimodal distribution, with an obvious spike indicating the large flat area of the lake surface. This spike shows the BV for generally flat terrain under these lighting conditions. The much smaller peak centered on zero in Figure 3 draws attention to where this example terrain begins to self-shade with the chosen lighting

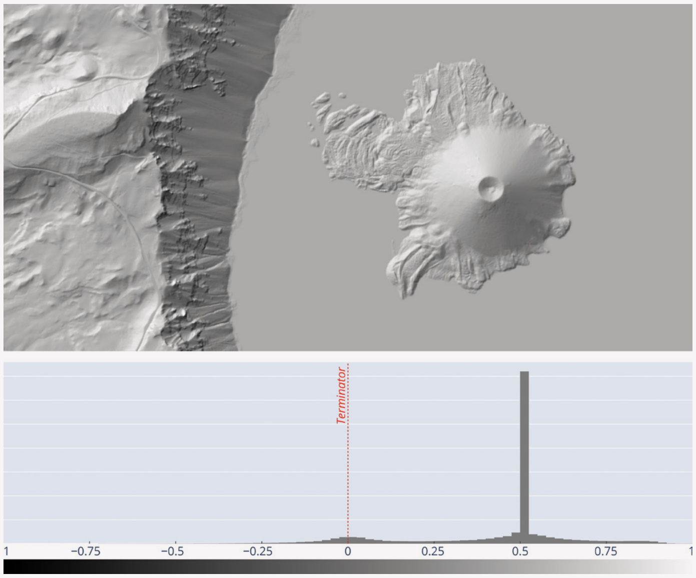


Figure 3. "Pure" Lambertian output.

angle. This demonstration/sample configuration makes it easy to see where the terminator BV lies, just by looking at the histogram. The terminator is always at 90° between surface normal and light vectors. For pure Lambert, that angle is mapped to BV=0.

You can use the tools available for download alongside this article to apply a pure Lambertian shader to your own terrain. If your terrain is not especially steep, or if you define a light high in the virtual sky relative to the steepest parts of your terrain, then you may not get many negative BVs. Try with a relatively low altitude of 30° above the horizon to start.

Use the pure Lambert tool (or your own version) to shade a terrain, saving the output as a new raster which we'll call `PureLambert`. Update the symbology for this layer to ensure that the color ramp stretches from -1 (black) to 1 (white). That ramp is useful for these examples, but in practice you might choose to restrict this color ramp using a percent clip or a standard deviation pattern to focus the ramp on where most of the action is. With careful color ramp control, you can precisely define just how dark the terrain behind the terminator should be. We'll get deeper into color ramps shortly.

This `PureLambert` output can be used as-is in contemporary GIS software. Its negative values are not inherently problematic, nor are floating point data values. There is no need to clamp or to re-range to [0, 255] as the standard hillshade tool does.[3] It would be useful, however, to re-range our [-1, 1] data into a more broadly accepted range of [0, 1].

There are real advantages to putting output in this range, the most important of which is compatibility with general-purpose image editors outside of GIS. Conceptually, there is also an advantage to thinking of brightness values as a *percentage*. If we manipulate the BV into a [0, 1] range, it becomes exactly that: percent brightness. There are a few approaches to fitting the raw output range into this more universal [0, 1] range. The options we will look at are: clamped, soft, and Half-Lambert.

CLAMPED LAMBERT

Clamping the pure output to [0, 1] will remove negative values, promoting them to zero. We've already examined at length why this is destructive and undesirable. Unfortunately, this approach and the associated data loss is the behavior of most GIS hillshade tools, as previously described. The raw Lambert data is clamped and then stretched across the color ramp spanning this narrower range (0–1) as seen in Figure 4.

For these manipulated outputs, I've included an output curve to illustrate how the full theoretical range of input angles maps to BV output. The x axis is the angle between

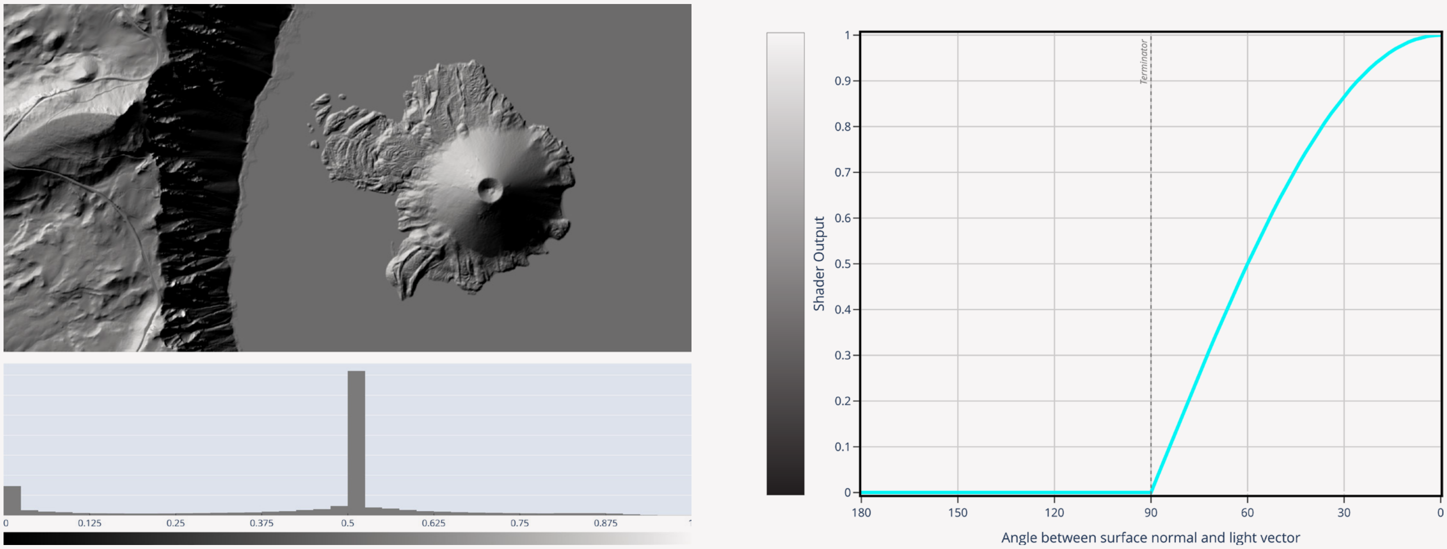


Figure 4. Clamped Lambert output.

[3] This practice has more to do with the history of image file formats than it does with any intrinsic need to limit the range of the output.

the light vector and surface normal vector. Note that all possible angles are given, even those behind the terminator (90–180°). The y axis is the output BV. The curve indicates how this clamping scheme relates the two.

You can create a clamped version of your Lambertian output using your preferred GIS's raster calculator/map algebra tool. The **con** operator in ArcGIS Pro is one easy way to do this (see Figure 5) with this expression, which checks whether `PureLambert` is less than zero and outputs 0 if that condition is met, or else it outputs `PureLambert`:

```
Con(“PureLambert” < 0, 0, “PureLambert”)
```

In QGIS, the raster calculator uses a slightly different command. Conditionals are specified with an **IF** statement:

```
IF(“PureLambert@1” < 0, 0, “PureLambert@1”)
```

Again, be sure to set the color ramp on the result to cover the entire output range, which in this case is [0, 1].

The effect of clamping is to make the output very dark, with many areas set to fully saturated black. The nuance/detail is scrubbed from areas behind the terminator, flattening the dark areas to a uniform BV.

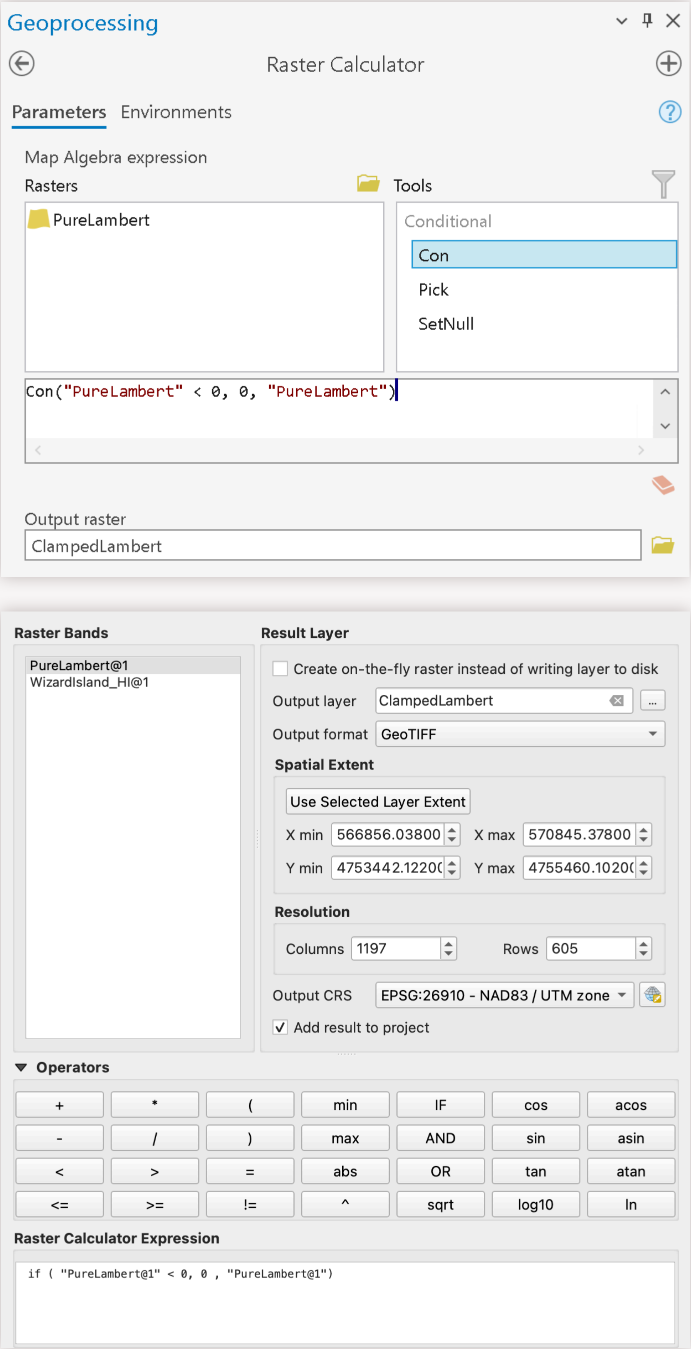
Note in the histogram that there is now a spike at zero, representing all the cells which were negative (or zero) in the `PureLambert` output. Contrast and brightness can be manipulated to lighten the dark cast, but the detail that was originally below zero in the `PureLambert` output is gone with this approach.

SOFT HILLSHADE

The detail available by leaving the output unclamped allows a “soft” hillshade (as described by Preppernau 2020). Preppernau's soft shade is easily achieved using the `PureLambert` output directly. But we're modeling maximum compatibility here, so we'll rescale the values with some light algebra:

$$BV_{soft} = (\cos(\theta) + 1) / 2$$

Note that this is not a clamping operation as above. This is an *affine transformation* in which the raw output is scaled (multiplied by ½) and translated (increased by ½) to fall between zero and one. A GIS raster calculator is again an easy way to do this, given that we already have pure Lambert values:



Rasters: PureLambert

```
Con("PureLambert" < 0, 0, "PureLambert")
```

Output raster: ClampedLambert

Raster Bands: PureLambert@1, WizardIsland_HI@1

□ Create on-the-fly raster instead of writing layer to disk

Output layer: ClampedLambert

Output format: GeoTIFF

Spatial Extent: Use Selected Layer Extent

X min 566856.03800 X max 570845.37800

Y min 4753442.1220(Y max 4755460.1020(

Resolution: Columns 1197 Rows 605

Output CRS: EPSG:26910 - NAD83 / UTM zone

☑ Add result to project

Raster Calculator Expression

```
if ( "PureLambert@1" < 0, 0 , "PureLambert@1")
```

Figure 5. Clamping with the Raster Calculator in ArcGIS Pro (top) & QGIS (bottom).

```
(“PureLambert” + 1.0) / 2.0
```

Let's save this output as `SoftLambert`. The entire output detail from `PureLambert` is preserved, but it is squeezed into a smaller range. So long as the output shade values are stored as floating-point numbers, the precision of that data type will retain all detail and nuance among shade values.

Notice the location of the terminator at 90° on the x axis of the output curve in Figure 6. This `SoftLambert` output, if on a linear color ramp, will map to 50% gray at that point, which is unrealistically bright for an area which is supposed to be just slipping into darkness behind the terminator.

One way to improve this display is to apply a non-linear color ramp to the output, where BVs [0, 0.5] (those areas behind the terminator) occupy very little of the color space—say, from black to 80% gray—with the remaining shade values (BVs [0.5, 1]) taking the rest of the ramp between 80% gray to white. The results for this 80/20 split are shown below. Feel free to adjust the grayscale ramp until you get a shading effect that brings out details in the darkest areas of your map while still displaying an overall pattern of gray shading that you find representative of your terrain.

The same `SoftLambert` output with this segmented color ramp is shown in Figure 8. Note that the steep areas of the caldera rim are now quite dark but are not completely black. By changing the color value of that mid-point color stop, you can adjust how dark the areas behind the terminator will be.

Because the Lambertian output contains all detail for all areas, the visual effect can be manipulated entirely with color ramp adjustments.

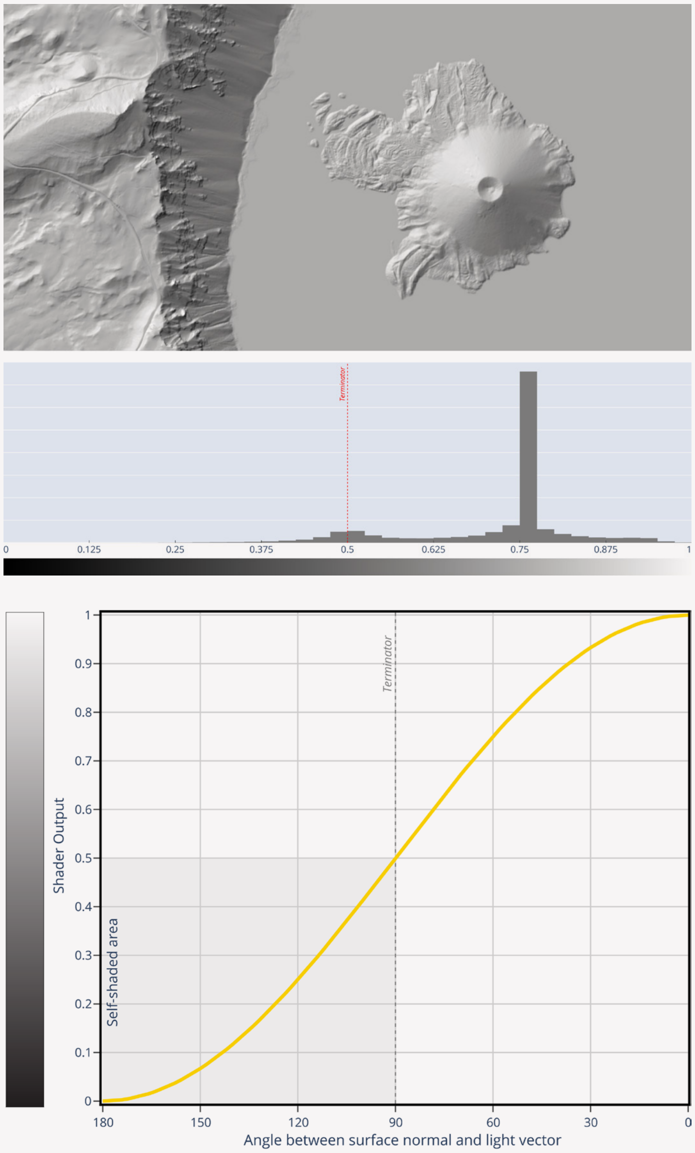
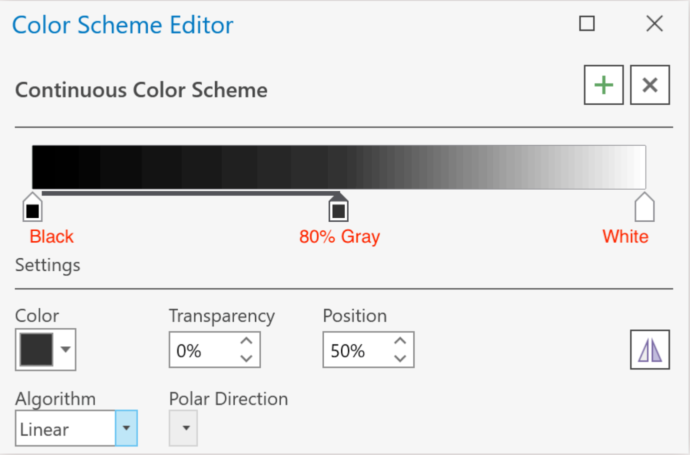


Figure 6. Soft hillshade output.


Color Scheme Editor

Continuous Color Scheme

Black 80% Gray White

Settings

Color: ■

Transparency: 0%

Position: 50%

Algorithm: Linear

Polar Direction

Figure 7. 80/20 color ramp.

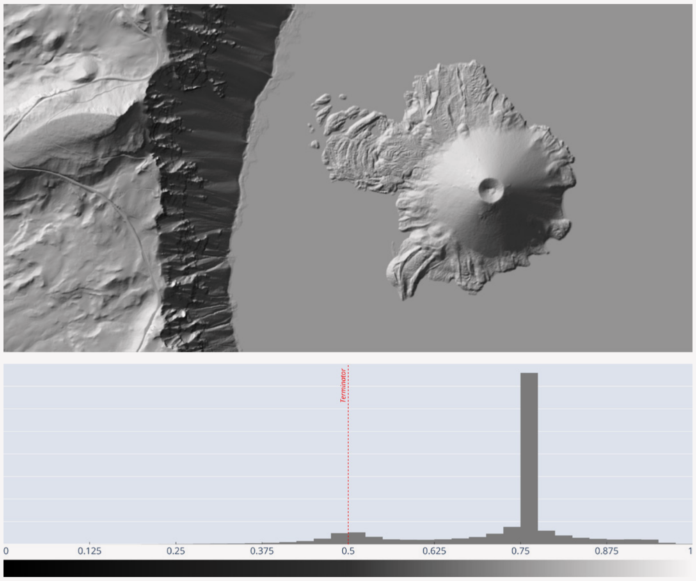


Figure 8. Soft hillshade with custom color ramp.

HALF-LAMBERT

The soft hillshade above is a little finicky, in that the color ramp needs to be tweaked for best results. One way to avoid this limitation is to use the Half-Lambert shader, which performs quite well with a linear color ramp applied to its full output range of [0, 1]. It can be thought of as a hybrid between the soft hillshade and the clamped hillshade, taking the best characteristics of each.

The Half-Lambert squares the output values from the `SoftLambert`. Squaring this value changes the shape of the output curve, notably flattening it on the dark side of the terminator:

Within the ArcGIS Raster Calculator, use the **power** operator to do this:

```
Power(“SoftLambert”, 2)
```

In QGIS, the Raster calculator implements the exponent with a separate operator:

```
“SoftLambert@0” ^ 2
```

Set a linear color ramp to span the full theoretical range of [0, 1]. The effect here is to keep output quite dark, but not uniformly black (Figure 9).

A few details are worth noting here:

- The terminator at 90° maps to a quite dark (but not fully black) value.
- The shape of the output curve on the dark side of the terminator gives most of the BV output range to those values nearest to the terminator. Meaning: areas within the dark side, but close to the terminator, get the most variation of the output color ramp behind the terminator.
- The two peaks in the histogram are worth mentioning. The lake surface is quite noticeable. The second, smaller peak (indicating the terminator in this example) lies at BV=0.25. This matches the output curve, which shows the terminator (θ of 90°) at BV=0.25.

The default linear color ramp gives reasonable results, but it can also be fine-tuned, much like we did above with the soft Lambert. In that case, however, the terminator lies at 0.25 rather than at 0.5, so your terminator color stop should be placed 25% along the ramp rather than halfway.

The three options for coercing Lambert into [0,1] are plotted together in Figure 10, so that their output curves might be directly compared. Stacked on the same graph, it is easy to see why the clamped Lambert is generally darker than the other options. For most inputs, its output BV is the same or lower than the other two options.

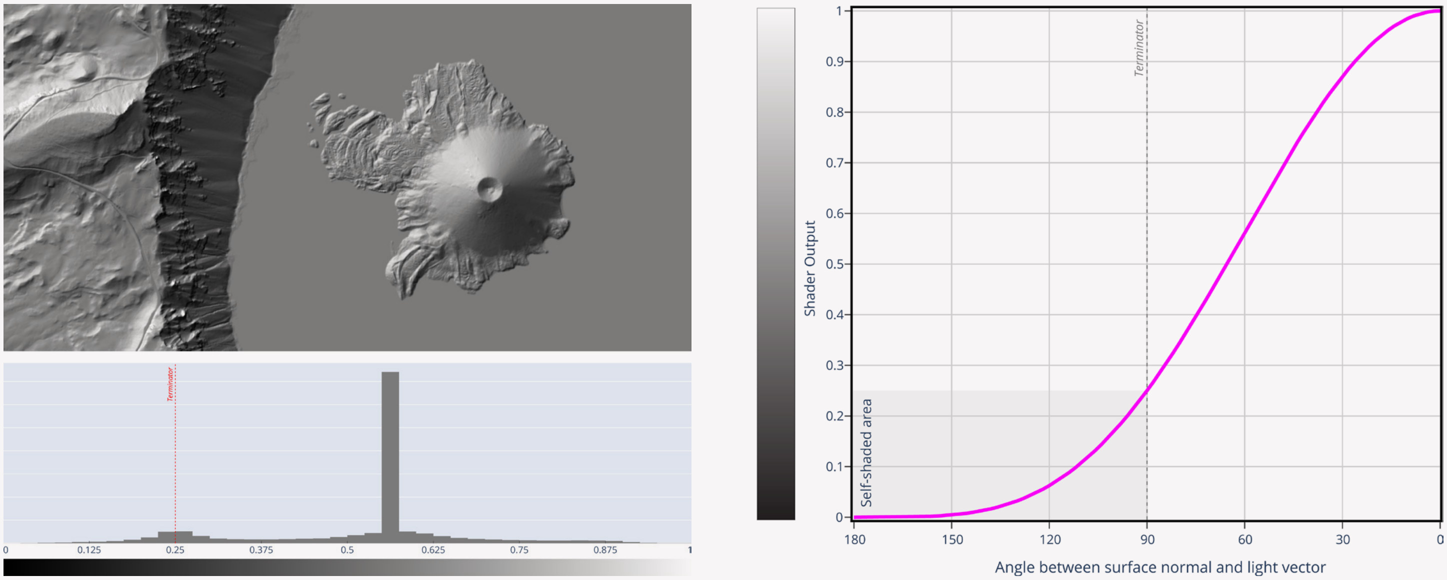


Figure 9. Half-Lambert hillshade.

SHADOWS

In the discussion of shaded relief so far, we've intentionally set aside shadows. The Lambertian shader and its variations described above symbolize a *surface* property only. One could easily argue that this is unrealistic, if not unfair, because the light we are simulating would not necessarily reach every pixel in the DEM. So, let's now look at shadows—a viewshed analysis—to take into account where the light actually lands on the surface.

The GIS tool you have probably used for shadows is limited in its nuance or subtlety: shadows are symbolized as a binary operator: a pixel is either in shadow (and thus fully blackened) or not (where the shade value is unmodified). Doing so will erase a lot of the work we just did above to keep that detail in the darker corners of the terrain. A more flexible workflow is to treat the shadow layer as an influence that darkens, rather than blackens.

To give the most flexibility and control over how strong this darkening influence is, we will keep shadows as a separate layer and use it as a semi-transparent layer on top of the scene. Even better, it can be effectively used with one of the darkening blend modes now available in most GIS software.

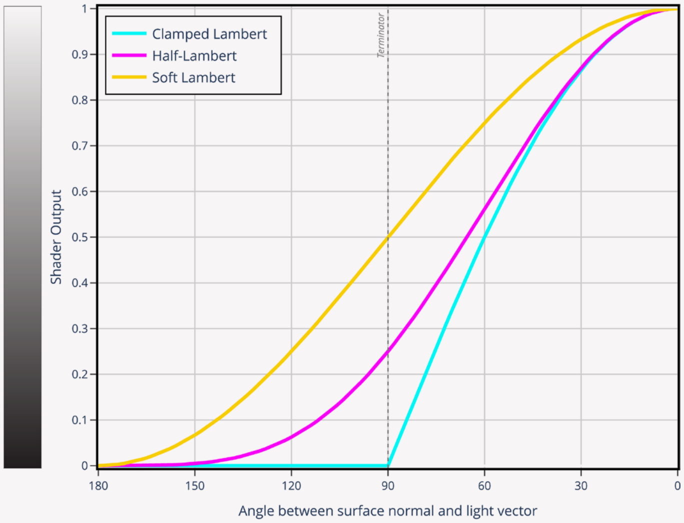


Figure 10. Comparison of output values for different Lambertian shader transforms.

The trick, then, is to generate a layer which represents shadows only. The typical method recommended for doing this is to compute a conventional hillshade with shadows modeled (using **the stock hillshade tool**), then using the raster calculator or a reclassify tool to discard all non-zero values. While that does work, I find it a little long-way-around-the-barn, when shadows can be computed directly in one go. The shadow algorithm I'll be illustrating here, adapted from Ware (1989), produces a binary raster image indicating yes/no for enshadowed pixels. Yes is represented as 1, no is represented as NoData. This tool for calculating simple shadows is included in the downloadable tools previously mentioned.

Figure 11 shows how a shadows-only layer might be applied to a plain Half-Lambert scene, using the "multiply" blend mode. The exact method for darkening under the shadow is now under the complete control of the cartographer: the color of the shadow, its transparency, the exact blend mode, and more. Compare this with the more

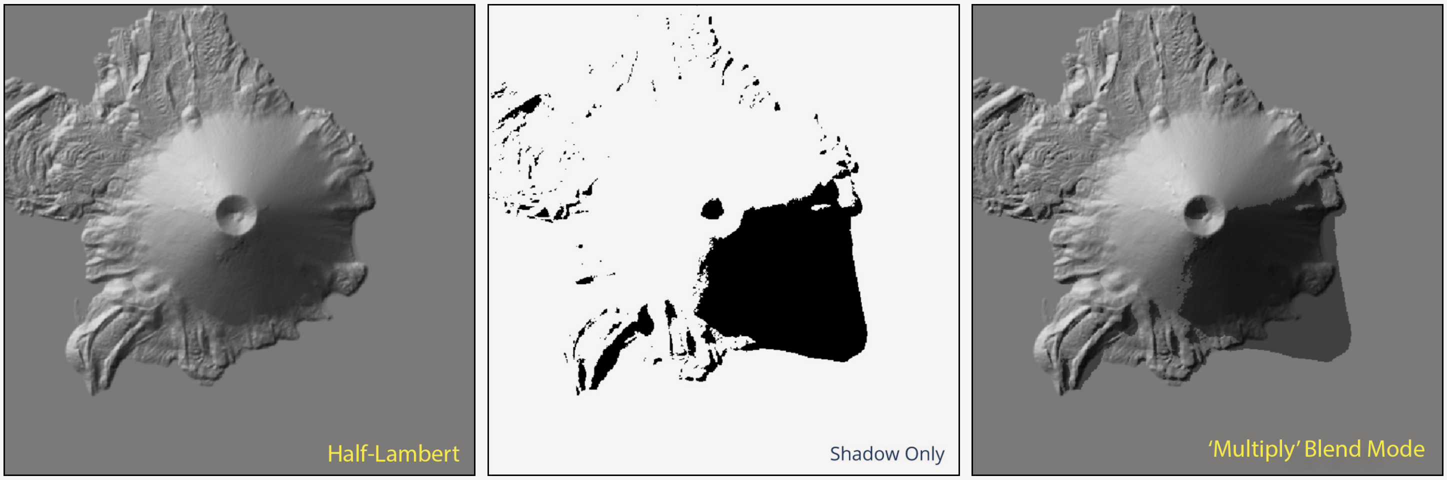


Figure 11. Shadows-only and shade layers.

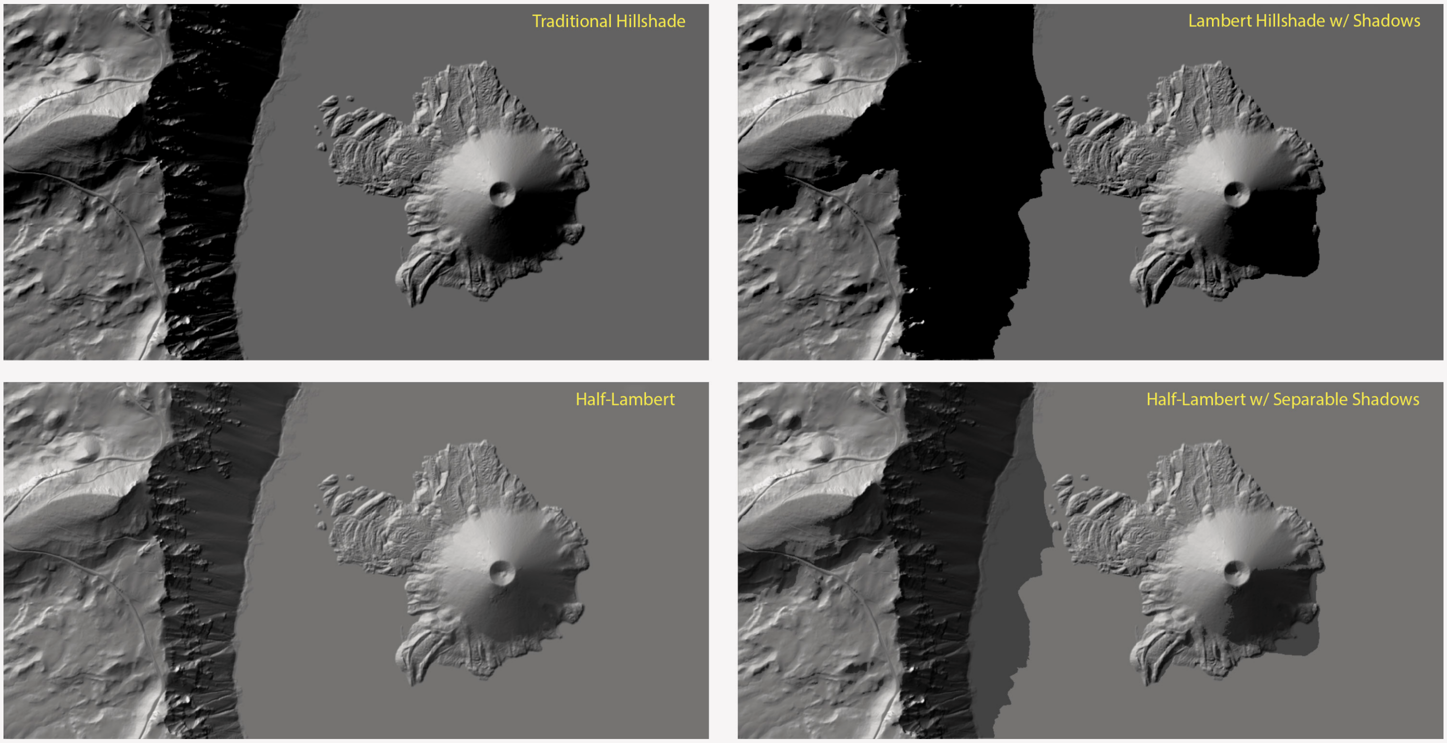

Traditional Hillshade
Lambert Hillshade w/ Shadows
Half-Lambert
Half-Lambert w/ Separable Shadows

Figure 12. Traditional vs. "separable" shadows. In this figure, I've lowered the light source to lengthen shadows and show their impact.

conventional shadow mechanism available as the default option with the standard tool in Figure 12.

Note the difference between the flat, dark shadow mass in the traditional method, compared with the very dark, but not fully blackened, areas in the Half-Lambert with separate shadows. The effect mimics ambient light behavior. It is important to understand that this is not actually accounting for other light sources, reflections in the environment, or other real-world light behaviors. We are merely lending that impression by manipulating a simplified lighting model.

Because it is not casting additional light rays from other light sources, the tool used to produce these scenes runs at the same speed as the traditional tool. This subtlety and control can be added to scenes without incurring any additional computational cost.

SHADOW EFFECTS

A shadow-only layer can be manipulated in other interesting ways before being blended with the shade layer underneath. Suppose you find the shadow line is too crisp: Blurring the shadow-only layer with a standard low-pass filter will give a softer edge between enshadowed and illuminated areas.

Adding multiple shadow-only layers from slightly different azimuth directions will allow you to build up a penumbra-like effect. Consider changing the color of each layer to construct an atmospheric effect (see Nelson 2019).

CONCLUSION

Each of the above shader options is derived from a Lambertian shader—the same basic shading algorithm used in the standard hillshade tool—yet the soft Lambert and Half-Lambert options offer different results with greater control. The main source of those differences is a consequence of "clamping," a destructive, irreversible manipulation of the Lambertian shade value. The loss of detail is most noticeable in very steep terrain or with low lighting angles.

With non-destructive transforms, Lambert can be squeezed into a standardized range, expressible as a percentage, without loss of detail in areas behind the terminator. For both soft hillshade and Half-Lambert, the key to showing detail in the display, yet keeping areas behind the terminator appropriately dark, is the color ramp. With a carefully partitioned color ramp, the cartographer has full control over how dark these areas will render.

Maintaining shadows as a separate layer offers even more control when composing the scene. A shadow-only layer allows for variation in color, intensity, edge crispness, and more. The blending of this layer with a detailed surface shading can yield an effect very similar to ambient lighting models, without the computational complexity required to render ambient light realistically.

ACKNOWLEDGEMENTS

Pat Kennelly has been a key source of support and guidance as I explored the ideas behind this article. His many helpful suggestions and insights shaped this article and helped to clarify its message.

REFERENCES

Kennelly, Patrick J., Tom Patterson, Bernhard Jenny, Daniel P. Huffman, Brooke E. Marston, Sarah Bell, and Alexander M. Tait. 2021. "Elevation models for Reproducible Evaluation of Terrain Representation." *Cartography and Geographic Information Science* 48 (1): 63–77. **https://doi.org/10.1080/15230406.2020.1830856**.

Horn, Berthold K. P., and Robert W. Sjoberg. 1979. "Calculating the Reflectance Map." *Applied Optics 18:* 1770–1779. **https://doi.org/10.1364/ao.18.001770**.

———. 1981. "Hill Shading and the Reflectance Map." *Proceedings of the IEEE* 69 (1): 14–47. **https://doi.org/10.1109/PROC.1981.11918**.

Nelson, John. 2019. "Throwing Shade." Accessed April 1, 2023. **https://www.esri.com/arcgis-blog/products/arcgis-pro/mapping/throwing-shade**.

Preppernau, Charles. 2020. "Normalizing the Normal Map." *Cartographic Perspectives* 96: 61–74. **https://doi.org/10.14714/CP96.1669**.

Ware, Colin. 1989. "Fast Hill Shading with Cast Shadows." *Computers and Geosciences* 15 (8): 1327–1334. **https://doi.org/10.1016/0098-3004(89)90096-4**.