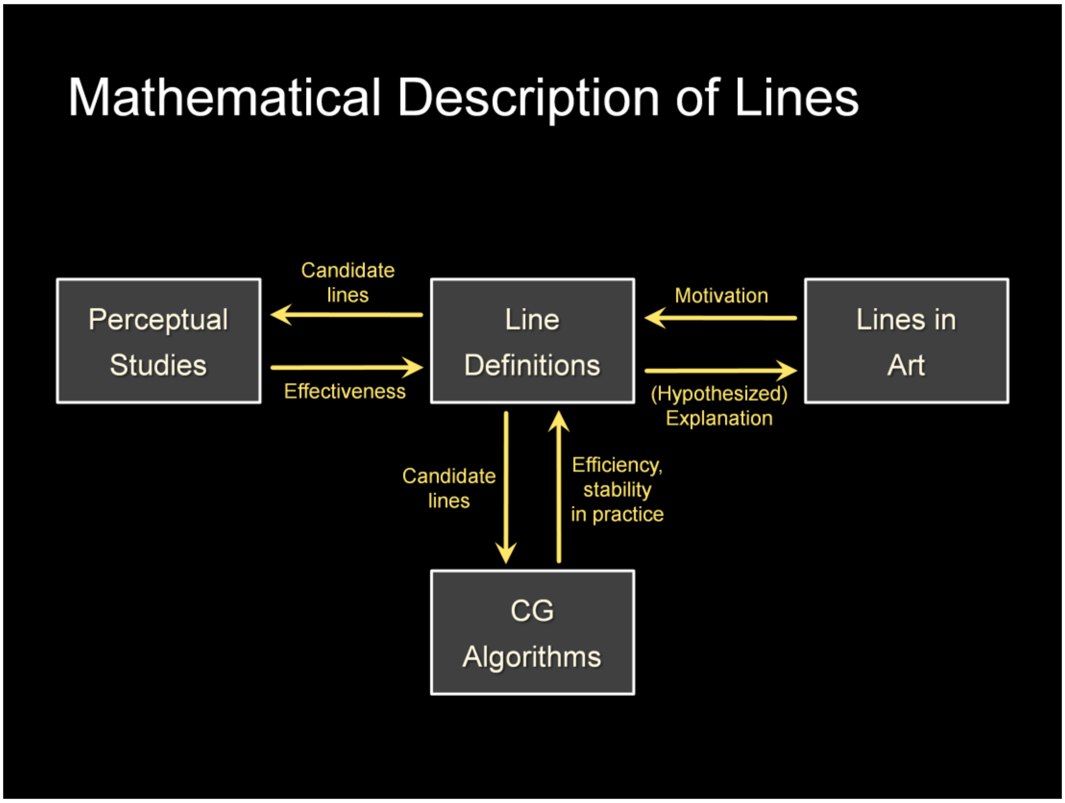
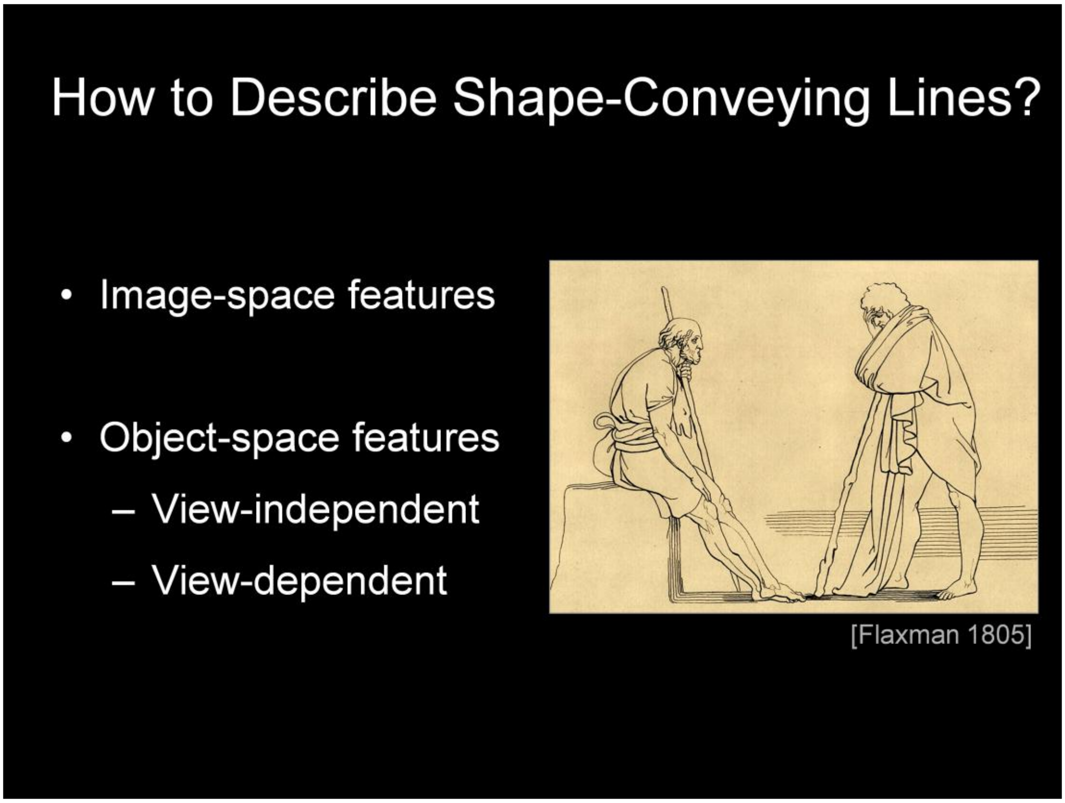
Part III: Mathematical Description of Lines

Szymon Rusinkiewicz

Line Drawings from 3D Models
SIGGRAPH 2008



In this section of the class, we will look at mathematical definitions of linear features on surfaces. This is an important component in our study of line drawings, since it serves to formalize the intuitions we get from looking at art. Moreover, the mathematical definitions can be turned into algorithms for producing candidate line locations on surfaces or images; this serves as the backbone of NPR line-drawing systems. Finally, as we will see later, we can ask questions about what shaped is perceived, given each different line definition.



Here’s a hand-drawn illustration by John Flaxman that illustrated a 19th century translation of the Odyssey. Notice how there are a variety of lines illustrating various effects, including things like shading, but in particular there are many lines that convey shape. When a viewer looks at these lines, these lines are naturally interpreted as indicating shape: they are not perceived as lines drawn on the surface!

In studying these shape-conveying lines, people have proposed three categories of mathematical definitions. First, there are “image-space” lines corresponding to features extracted from an image. In the case of lines from 3D models, the images are usually renderings with Lambertian reflection, in most cases using a headlight as the only light source. Next, there are “object-space” features computed directly on the 3D surface. Finally, there are object-space features whose definition depends on the view position.

Image-Space Lines

+ Intuitive motivation; well-suited for GPU
– Difficult to stylize

Examples:

- Isophotes (toon-shading boundaries)
- Edges (e.g., [Canny 1986])
- Ridges, valleys of illumination [Pearson 1985, Rieger 1997, DeCarlo 2003, Lee 2007, …]

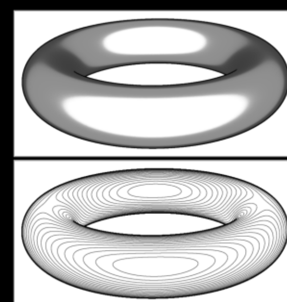
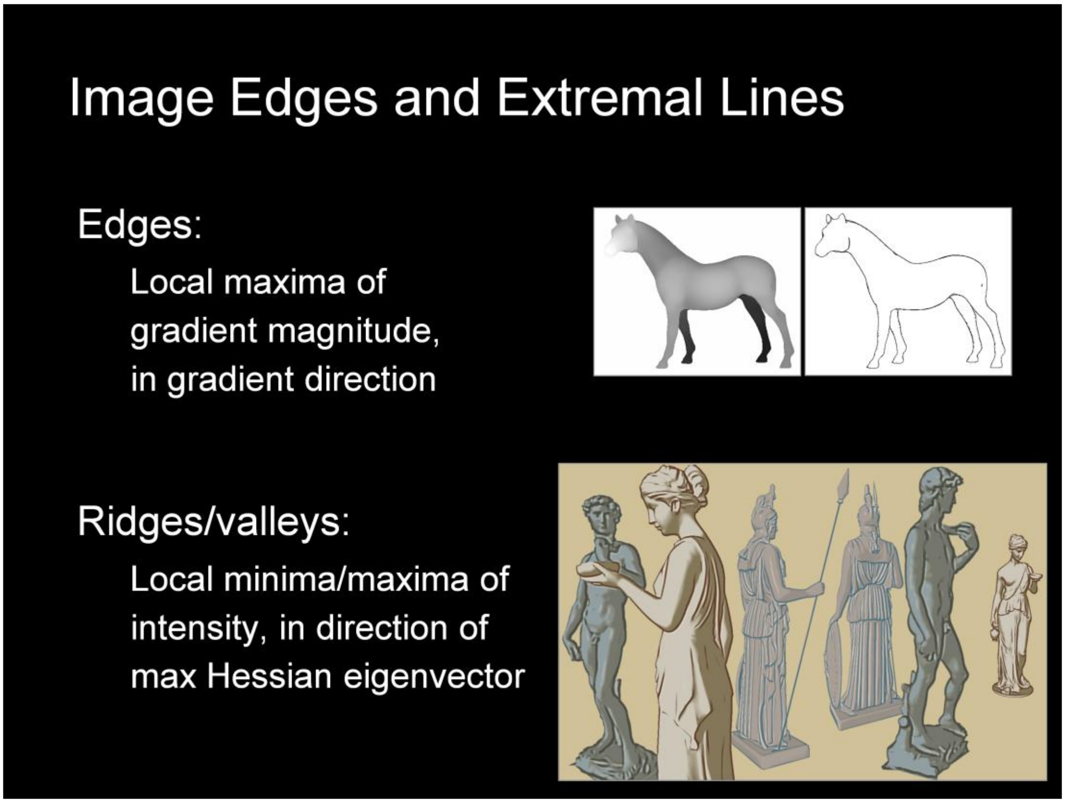


Image-space lines have a very intuitive motivation and a strong connection to art: artists are frequently taught to "draw what they see". Another big advantage is that they translate into efficient CG algorithms, because they can often be implemented in graphics hardware. On the other hand, such GPU-computed lines can be difficult to stylize, since they are often extracted as collections of pixels rather than as complete curves.

As an example of image-space lines, isophotes are curves of constant illumination, which also correspond to toon shading boundaries.



Another definition for image-space lines is edges. Using the definition of the popular Canny edge detector, these are locations where the image gradient magnitude has a local maximum, when looking in the gradient's direction. (In practice, the gradient direction is sometimes quantized to 45-degree increments, to simplify finding the local maxima.) Canny's algorithm also includes a sophisticated system of hysteresis thresholding, to keep the most important lines.

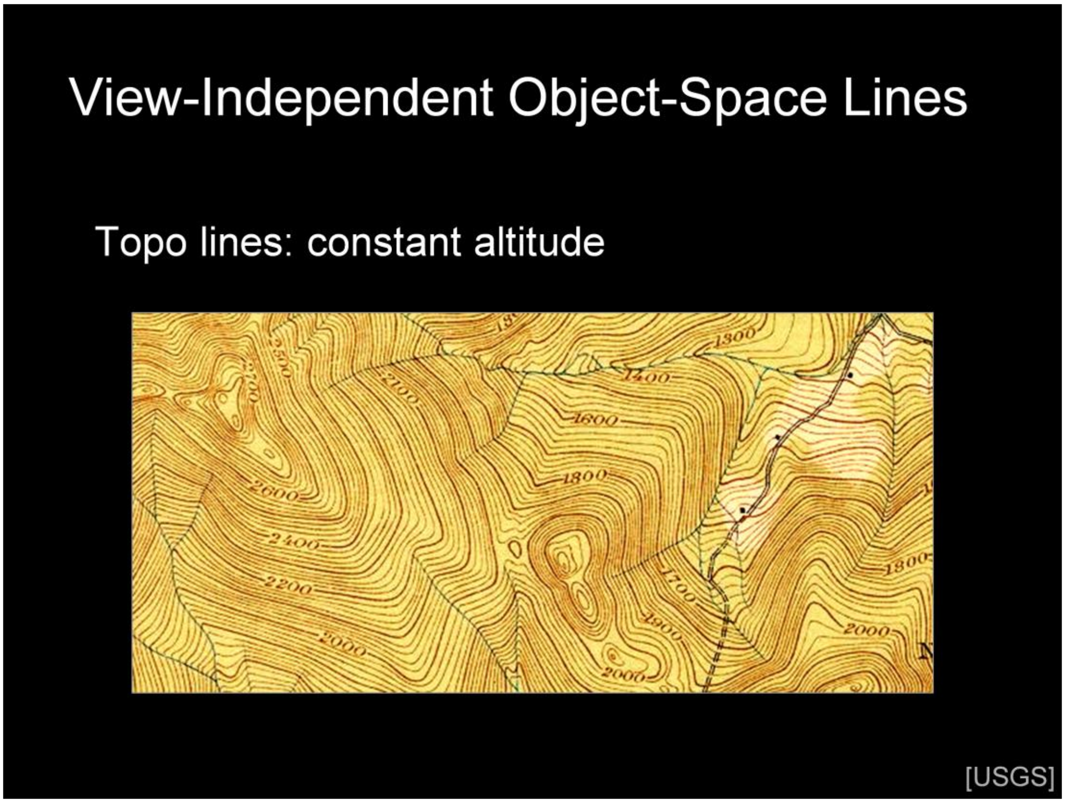
A final very important class of image-space lines are the image intensity "ridges" and "valleys". These are curves on the surface of locally maximal/minimal intensity, and are very naturally drawn in white/black on a mid-tone color, as in this example from Lee et al. There are several different definitions of image ridges and valleys, many of which were originally developed in the course of analyzing watercourses on terrain (see, for example, Rieger's paper). The definition used by Lee et al. looks for local maxima/minima of intensity, in a direction determined by fitting a paraboloid locally and looking for the direction of highest curvature. In other words, the direction is the eigenvector corresponding to the largest-magnitude eigenvalue of the image Hessian (matrix of second partial derivatives).

View-Independent Object-Space Lines

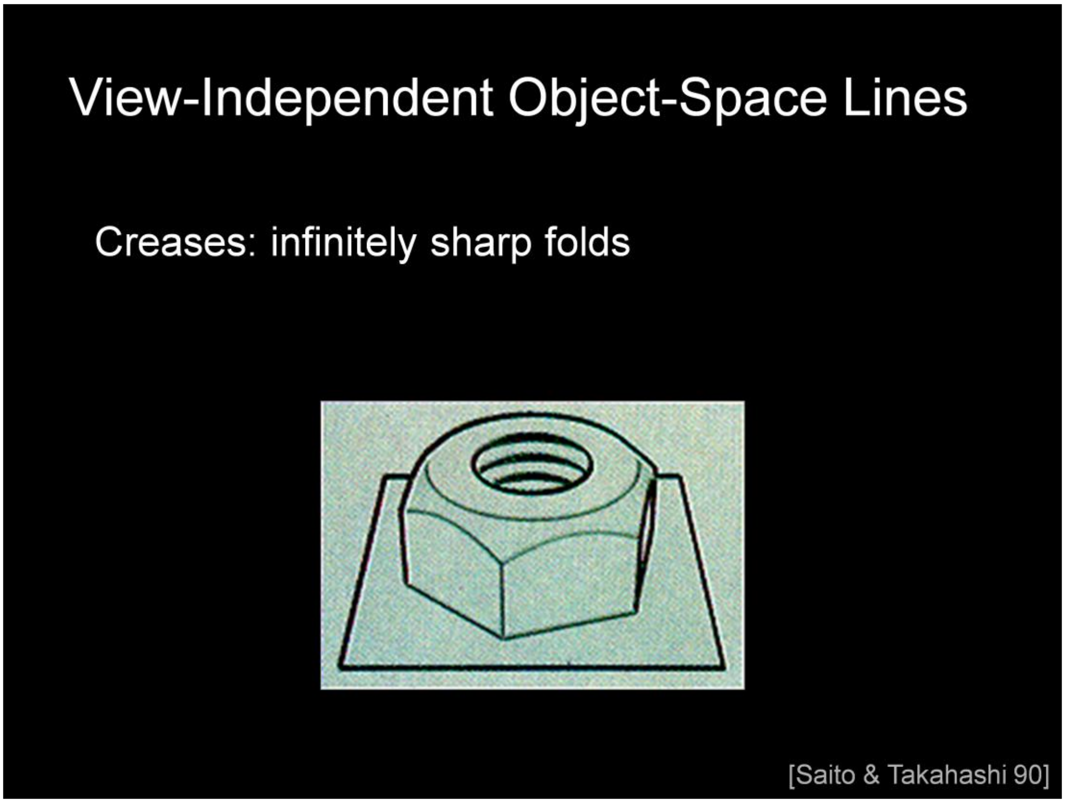
+ Intrinsic properties of shape; can be precomputed
– Under changing view, can be misinterpreted as surface markings

For the next class of lines, we consider “object-space” definitions that look directly at the 3D shape of a surface. View-independent lines are those that are defined only based on the shape, without considering the viewer’s position. This is somehow intuitively satisfying, since these lines depend only on the surface shape, and a view-independent definition means that these lines can be precomputed and then re-used for multiple viewer positions.

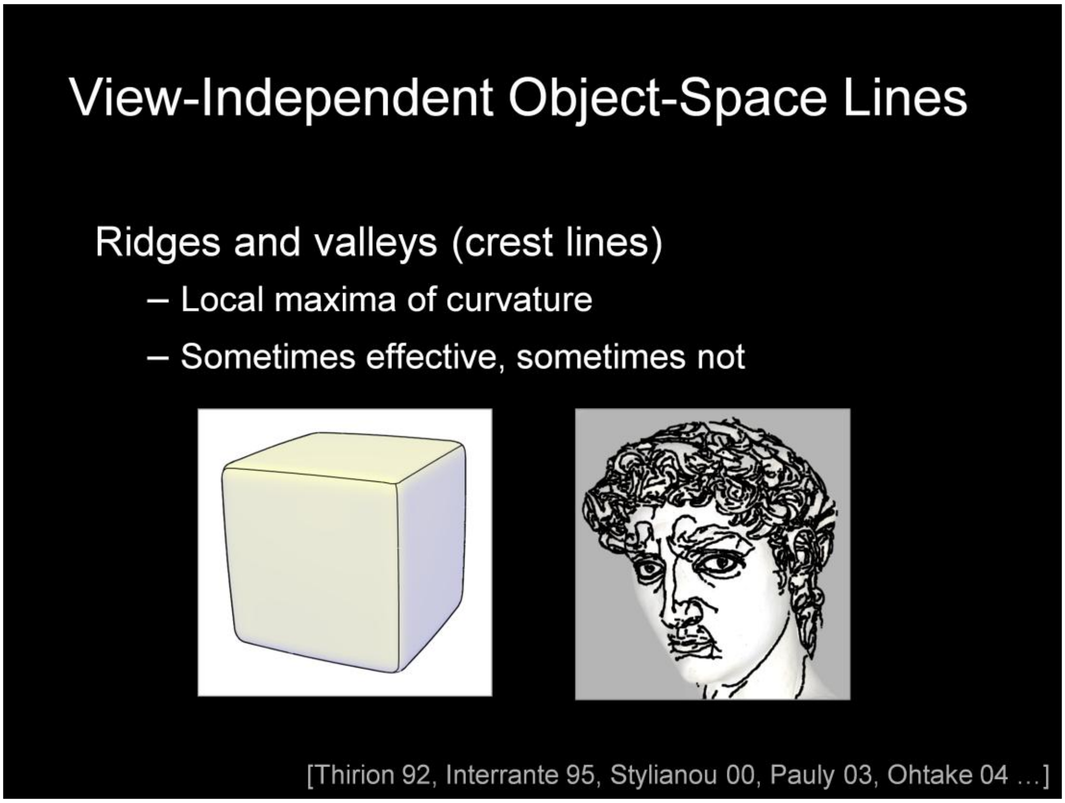
However, it has been observed that, under changing view, these lines are more naturally interpreted as markings on the surface, rather than as lines depicting the shape of the surface. We believe that further perceptual studies will certainly be necessary to investigate this.



One example of view-independent lines are the constant-altitude lines found on topographic maps. They certainly are effective at conveying shape, both through the shape of the lines and their density.



Another example, appropriate for objects with sharp folds (i.e., discontinuities in the normal), is lines at those creases of the surface. This works well for polyhedral objects, and is a frequent ingredient in technical drawings. The definition is very simple: just look for a dihedral angle (that is, the angle between two faces connected along an edge) smaller than a threshold.



Unfortunately, the natural generalization for smooth surfaces, ridge and valley lines, leads to mixed results. For example, the ridge lines on the rounded cube at left successfully convey its shape. If you look at the picture on the right, you can see that some of the ridge and valley lines, such as those around the eyes, do a good job of marking features. However, other lines look like surface markings and no good artist would include them in a hand-made drawing.

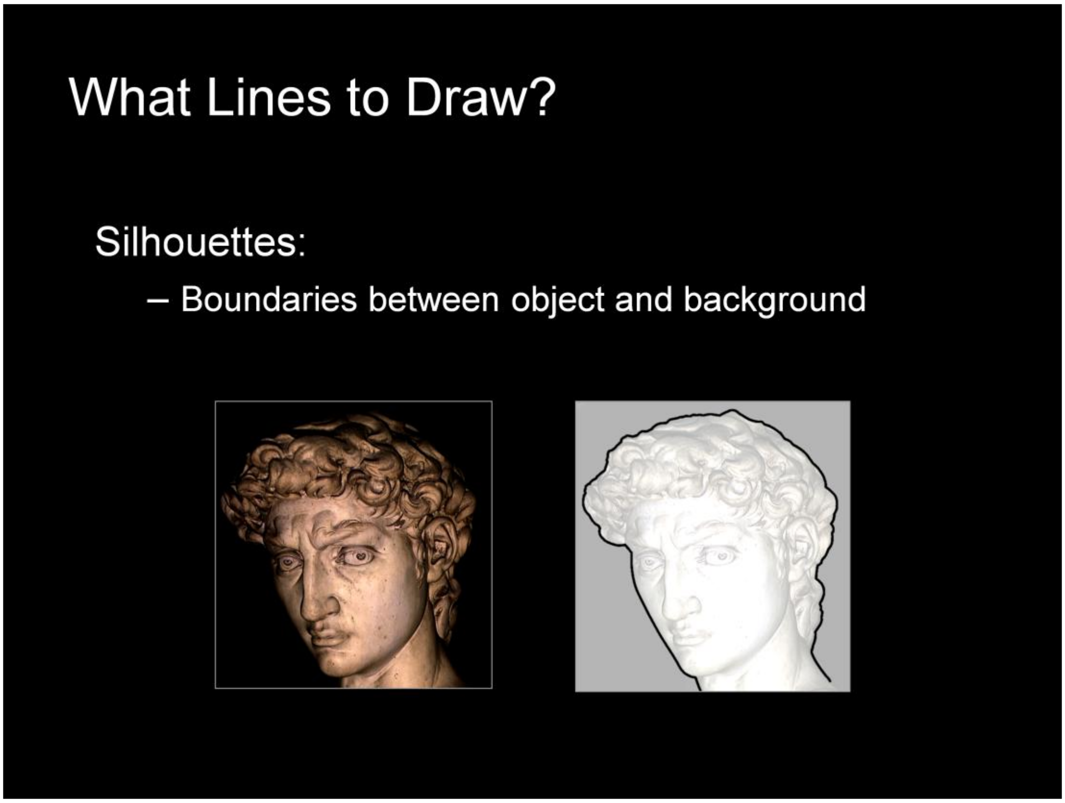
The definition of ridge and valley lines is slightly complex: they are local maxima of principal curvature, in the corresponding principal direction. (There will be a description of differential-geometry principles later, which should clarify what this means!)

View-Dependent Object-Space Lines

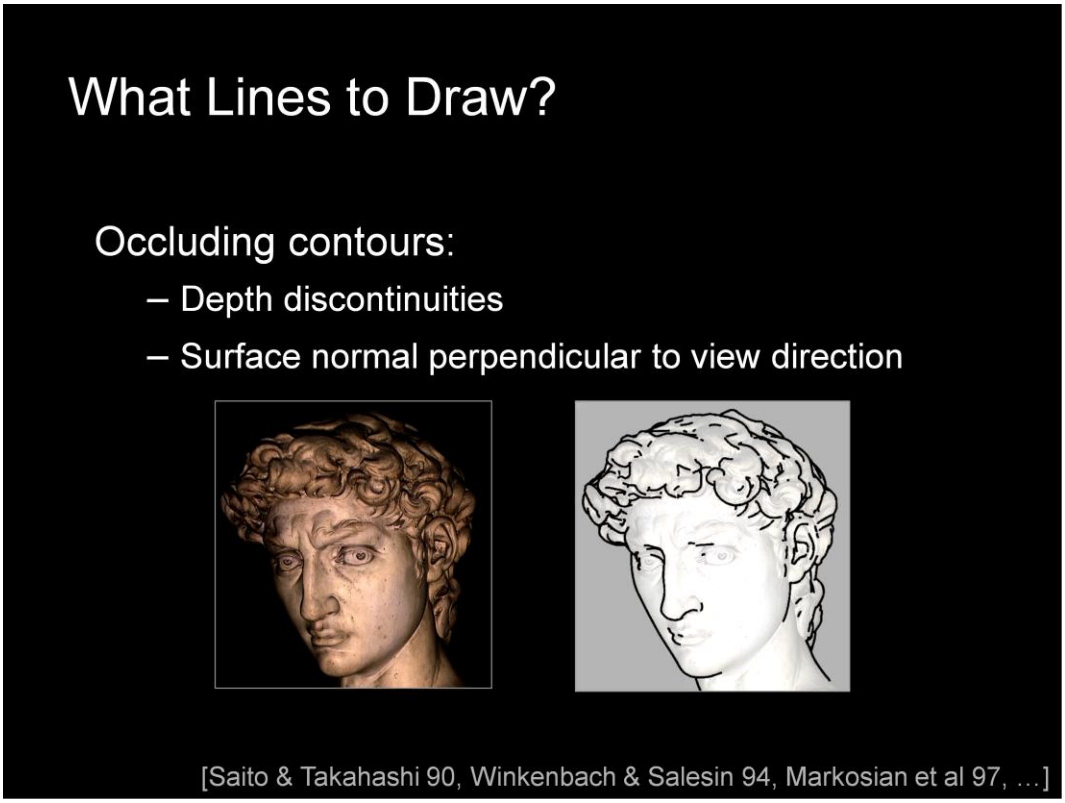
+ Seem to be perceived as conveying shape
– Must be recomputed per frame

This brings us to the third class of mathematical line definitions, and the one on which we will spend the most time. These are lines defined on the 3D surface, but taking the view position into account. (To be precise, sometimes we think of an orthographic viewer, and so talk about a view direction rather than a viewer position.)

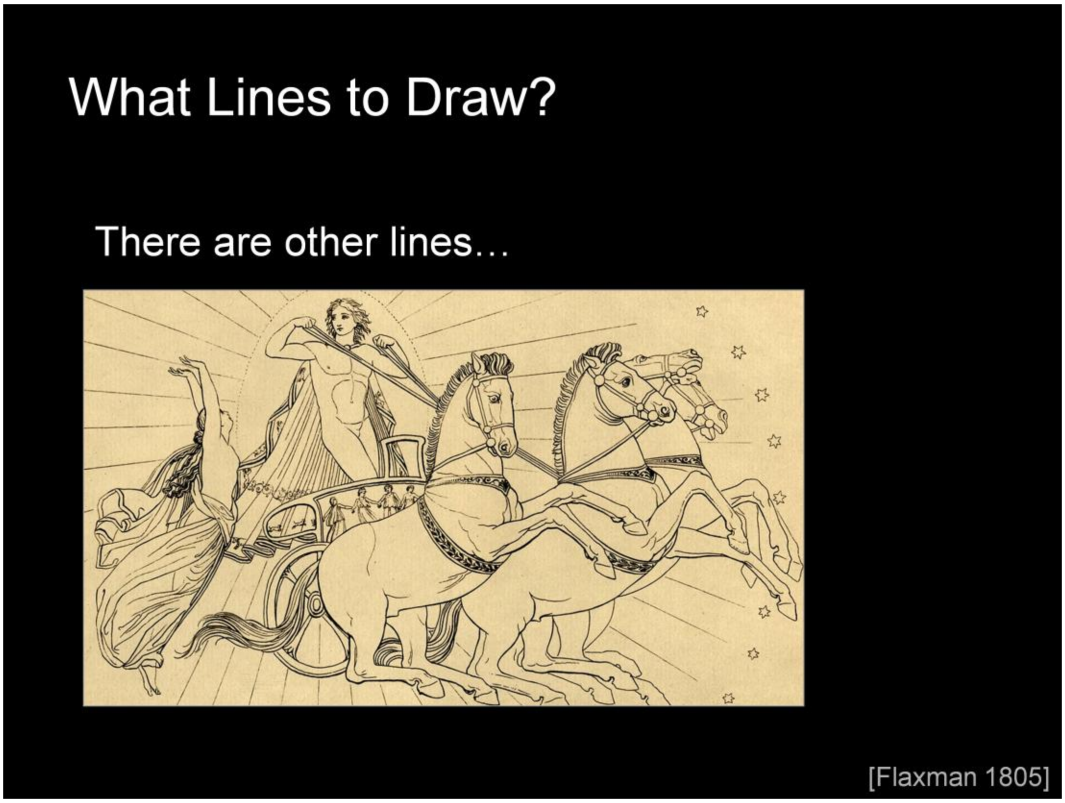
These are the lines that appear to be most effective at conveying shape, and lend themselves naturally to many stylization techniques. On the other hand, they do have to be recomputed whenever view changes, so they aren't necessarily the most efficient (though there has been some progress in GPU-based implementations).



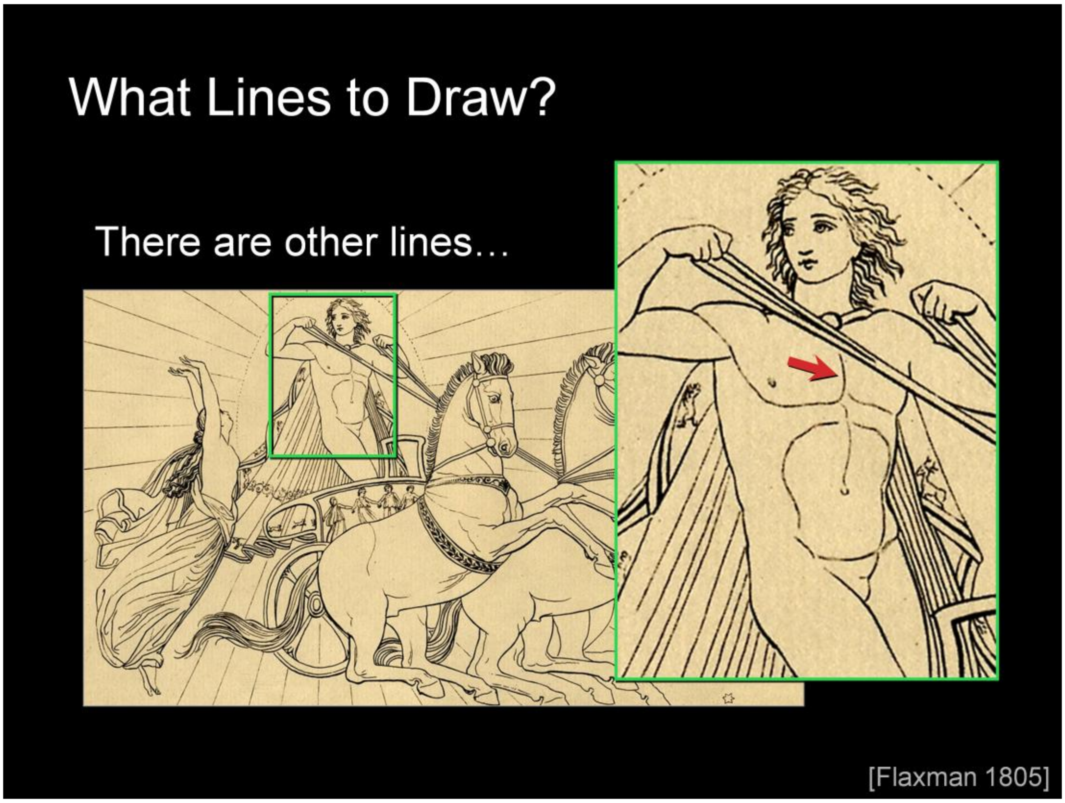
First, we start with the silhouette: the boundary between the object and the background. Here, we draw the silhouette on the right, superimposed over a contrast-reduced version of the photograph on the left (just so we can see what's going on). Silhouettes are obviously very important, and are an essential ingredient in any line drawing. However, as you can see here, they're clearly not enough.



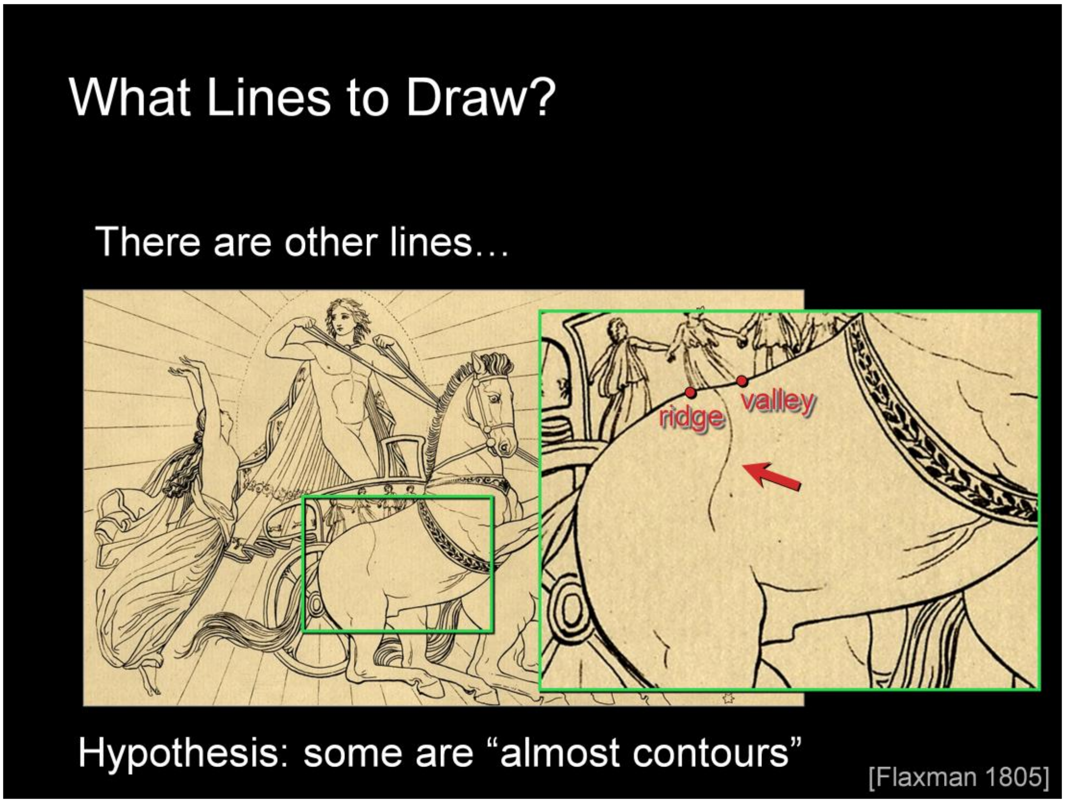
Another common definition is a generalization of silhouettes called occluding contours (or sometimes “interior silhouettes”). These mark any depth discontinuities, not just those against the background. As seen on the right, this adds a lot of important detail to the drawing, but still does not convey shallow features, particularly those viewed head-on. Still, these are a common component in NPR systems.



So, to complement silhouettes and contours we need another line definition. If we go back and look at a real line drawing, we see that there are, in fact, more lines that pretty clearly are not contours (and not ridges or valleys).

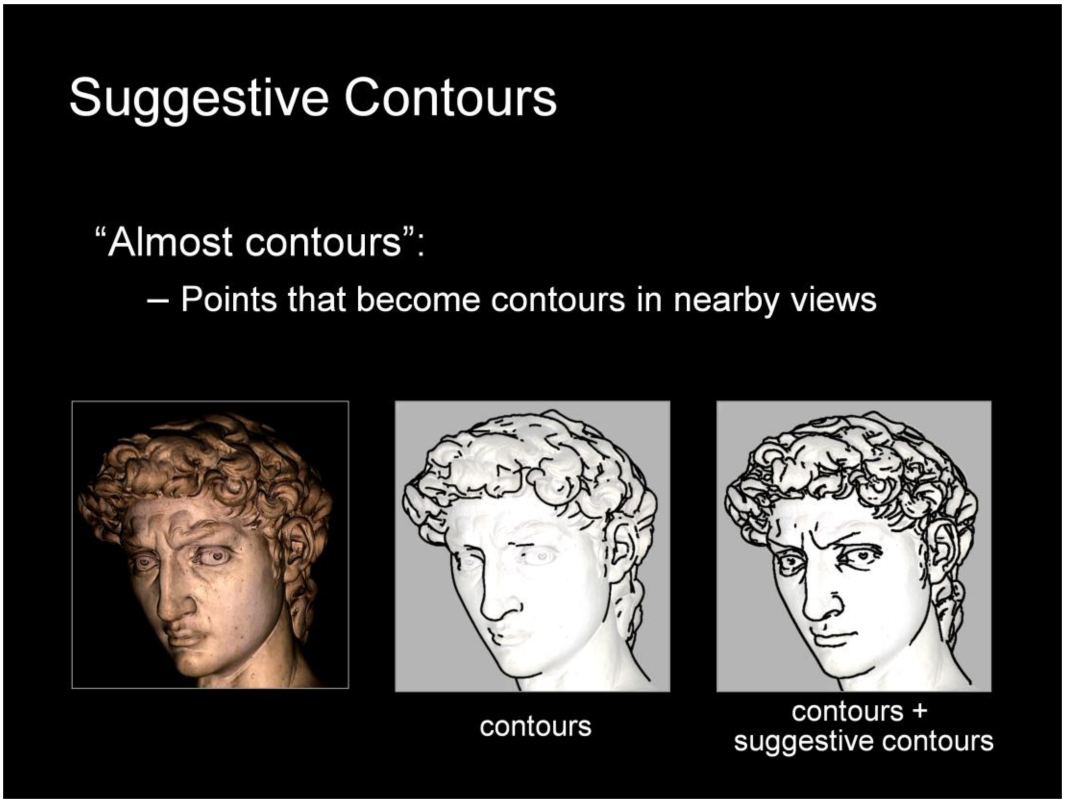


For example, there's this line on the chest that's clearly on the right side of the torso, not in the middle of the "valley".

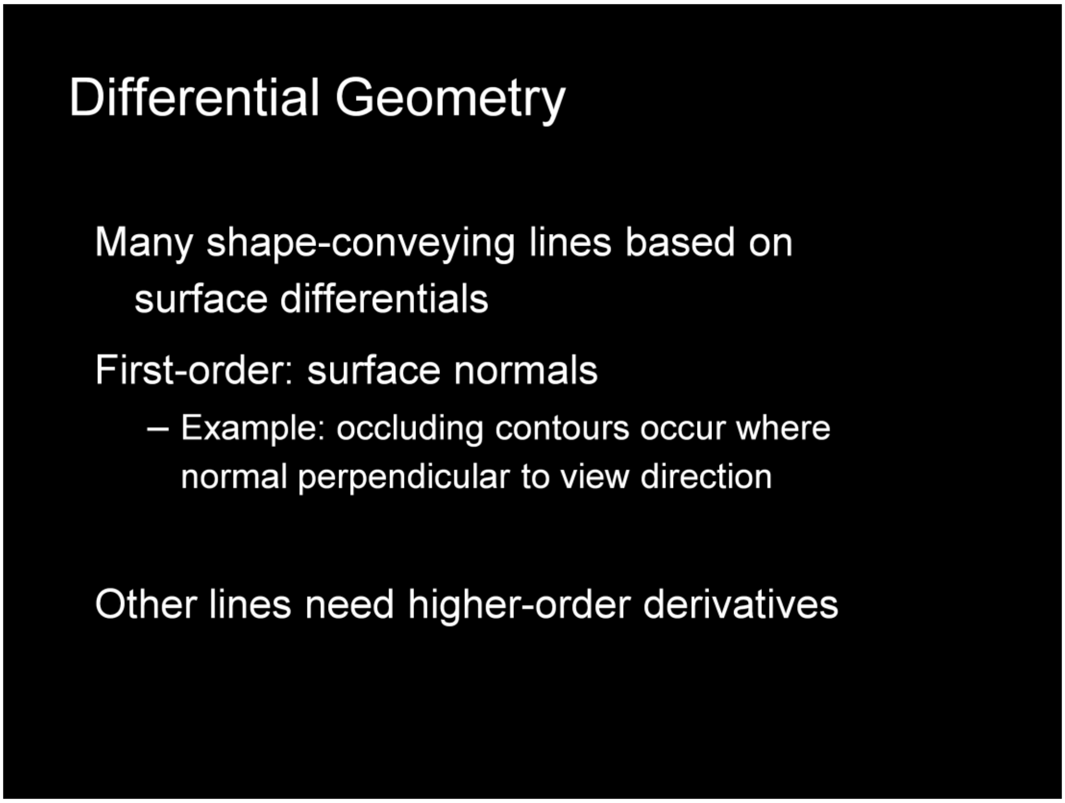


Another example is the line on the back of the horse. See how it misses the ridges and valleys on the back of the horse. Also, it’s certainly not a contour.

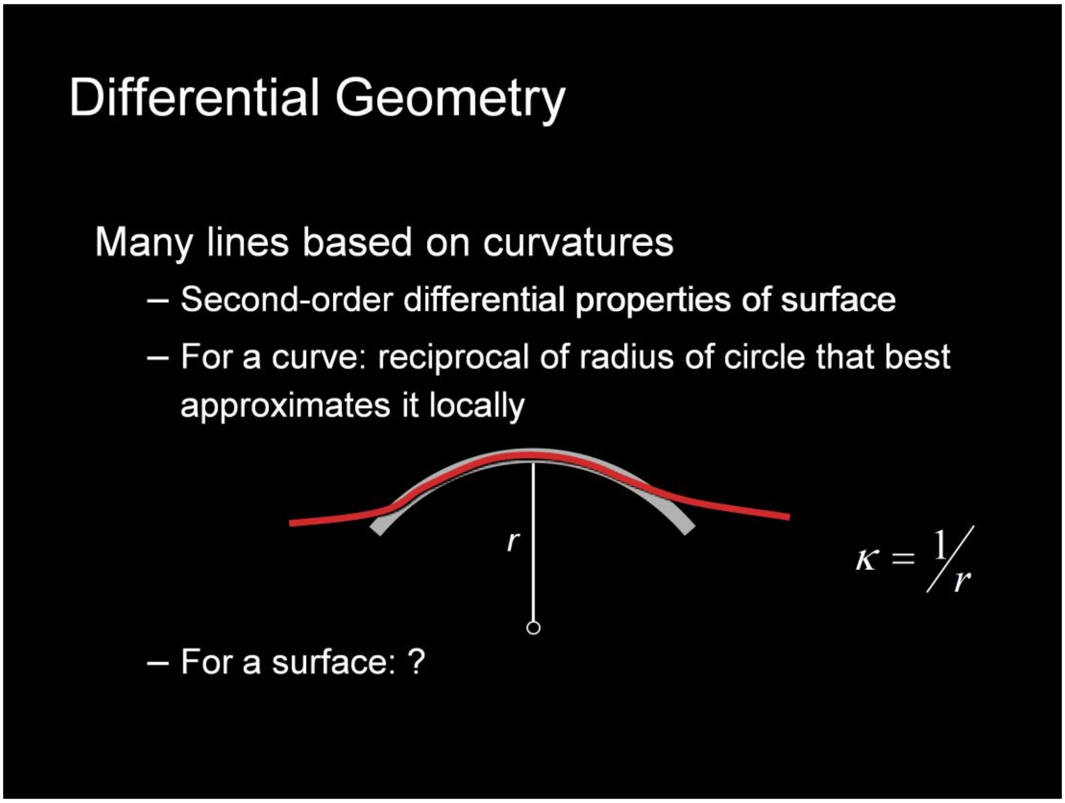
We hypothesize that these lines are “almost contours”: if you moved your head a bit to the left, this line would in fact become a contour. We call these lines “suggestive contours”, and we’ll later see how to formalize what they are and how to find them on a surface.



So, here's what suggestive contours look like on the David. You can see that they complement the contours nicely (in fact, you can prove that they line up with the contours in the image). Also, they include a lot of the detail that's missing in the contours-only drawing.



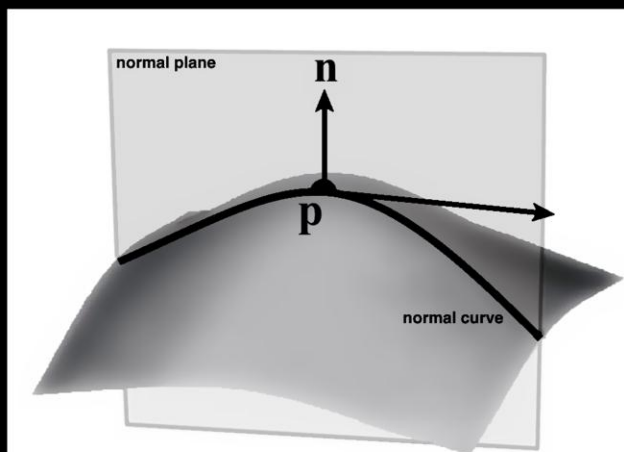
It turns out that in order to formalize different families of 3D lines on a surface, we'll need some math from a field called differential geometry. This is the field that concerns itself with what it means to take "derivatives" of curves and surfaces. You are already familiar with a first-order differential quantity of surfaces: the normal. In fact, as has already been mentioned, occluding contours critically depend on the normal: they are zeros of the dot product between the normal and the view direction. In a very similar way, different kinds of lines, like suggestive contours, will have definitions that depend on higher-order derivatives.



So, let's move on to exploring second-order derivatives, or curvatures. To start with, let's recall the familiar definition of the curvature of a curve: at each point, it is the reciprocal of the radius of a circle that best approximates the curve locally. The sharper the bend in the curve, the higher the curvature. Curvature has units of one-over-length: if you scale an entire curve up by a factor of two, all the curvatures are halved.

Normal Curvature

Curvature of a normal curve



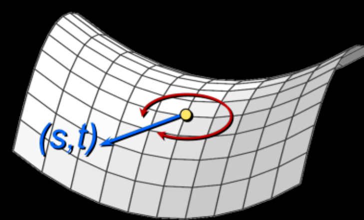
For a surface, we can talk about the curves formed by intersecting the surface with any plane containing the normal. These are called "normal curves", and their curvature is "normal curvature". So, for each point on the surface, there are many different curvatures, corresponding to all the different normal planes passing through that point.

Curvature on a Surface

Normal curvature varies with direction, but for a smooth surface satisfies

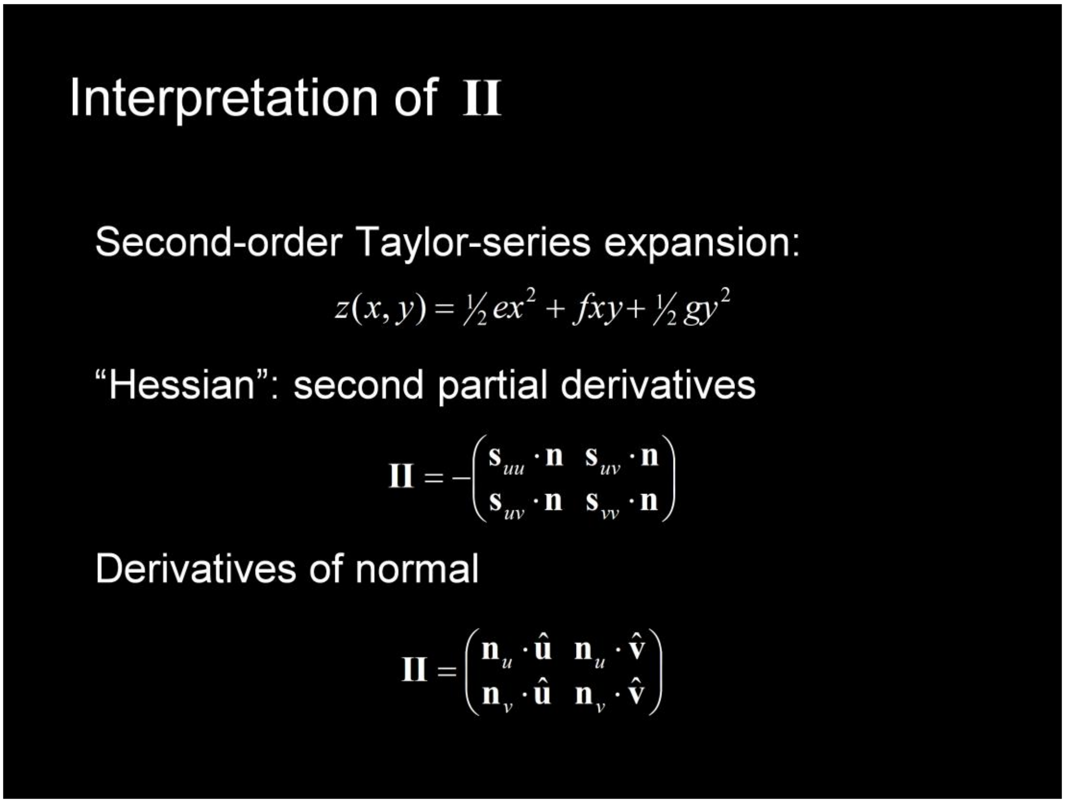
$$\kappa_n = \begin{pmatrix} s & t \end{pmatrix} \begin{pmatrix} e & f \\ f & g \end{pmatrix} \begin{pmatrix} s \\ t \end{pmatrix}$$

$$= \begin{pmatrix} s & t \end{pmatrix} \mathbf{II} \begin{pmatrix} s \\ t \end{pmatrix}$$



for a direction (s,t) in the tangent plane and a symmetric matrix $\mathbf{II}$

There is something interesting that happens, though. For a smooth surface, the variation of normal curvature with direction can't be arbitrary – it has a very specific form. Imagine setting up a local orthonormal coordinate system in the tangent plane at a point on a surface. For any direction (s,t), expressed in terms of that coordinate system, we can find the normal curvature in that direction in terms of a simple formula involving a symmetric matrix II. This matrix is known as the "second fundamental tensor", and as we'll see is related to how much the surface is bent. Note that if you were to expand this formula you'd get terms quadratic in s and t: this whole expression is therefore just a fancy way of writing a quadratic form.



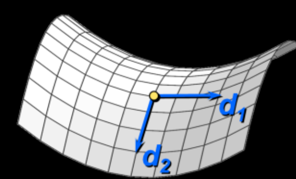
What exactly is II? What are its elements? It turns out that there are many formulas for them, all involving some notion of the second derivative of the surface. For example, they are precisely the second-order terms in a Taylor series expansion of the surface (assuming that z is oriented along the surface normal). Equivalently, they are the derivatives of the normal as you move along the surface.

Incidentally, if you look for formulas like this in various textbooks, there's a good chance you may see them written with the opposite sign. This is because when writing the formulas you need to establish the conventions of whether normals are considered to point into or out of the surface, and in addition whether convex surfaces are taken to have positive or negative curvature. We assume that convex surfaces have positive curvature, and we use the usual graphics convention of outward-pointing normals, leading to the signs used here.

Principal Curvatures and Directions

Can always rotate coordinate system so that $\mathbf{II}$ is diagonal:

$$\mathbf{II} = \mathbf{R}^{\mathrm{T}} \begin{pmatrix} \kappa_1 & 0 \\ 0 & \kappa_2 \end{pmatrix} \mathbf{R}$$



κ_1 and κ_2 are *principal curvatures*, and are minimum and maximum of normal curvature

Associated directions are *principal directions*

Eigenvalues and eigenvectors of $\mathbf{II}$

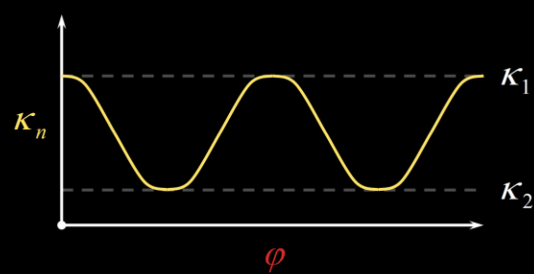
In many situations it is convenient to rotate the local coordinate system to make the matrix II diagonal. It turns out you can always do this: the new coordinate axes are the eigenvectors of II. (You might recall a neat theorem from linear algebra that the eigenvalues of symmetric matrices are guaranteed to be real: here's a real-life application that relies on this fact.)

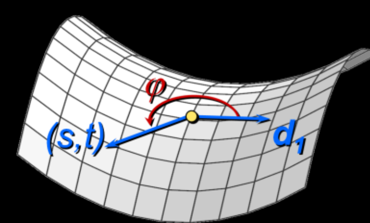
Once you've done this change of coordinates, the new axes are known as the principal directions, and the corresponding curvatures are the principal curvatures. If we plug in the new form of II into the formula for normal curvature, we see that all normal curvatures have to lie between the principal curvatures. So, the principal curvatures are the minimum and maximum curvatures for any direction (at that point on the surface).

Euler's Formula

This lets us write normal curvature differently:

$$\kappa_n = \kappa_1 \cos^2 \varphi + \kappa_2 \sin^2 \varphi$$





This leads to something called Euler's formula for normal curvature, which expresses the curvature in any direction as a function of the principal curvatures.

Gaussian and Mean Curvature

The Gaussian curvature $K = \kappa_1 \kappa_2$

The mean curvature $H = ½ (\kappa_1 + \kappa_2)$

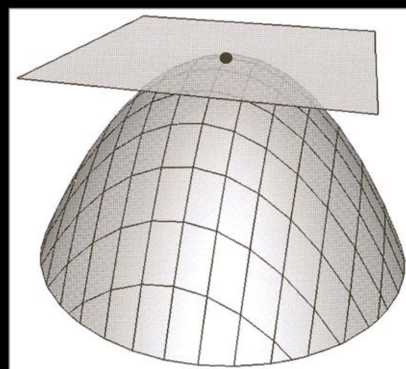
Equal to the determinant and half the trace, respectively, of the curvature matrix

Enable qualitative classification of surfaces

When talking about curvatures, there are a couple more terms that often crop up: Gaussian curvature and mean curvature. These are equal to the product and average of the principal curvatures, and can also be computed directly from II (expressed in terms of any coordinate system), as the determinant and trace. Notice one interesting feature about Gaussian curvature: it has units of "curvature squared", which is different from all the other flavors of curvature we've talked about.

Gaussian and mean curvature are very useful for qualitatively talking about the shape of a surface.

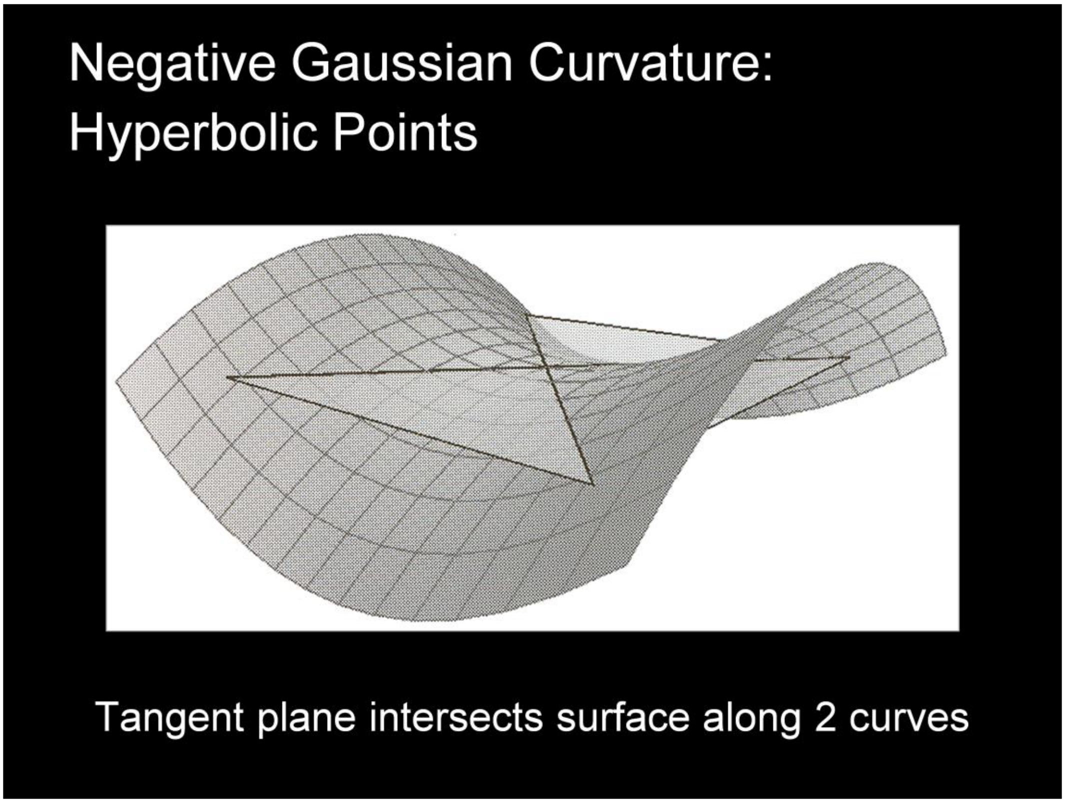
Positive Gaussian Curvature: Elliptic Points



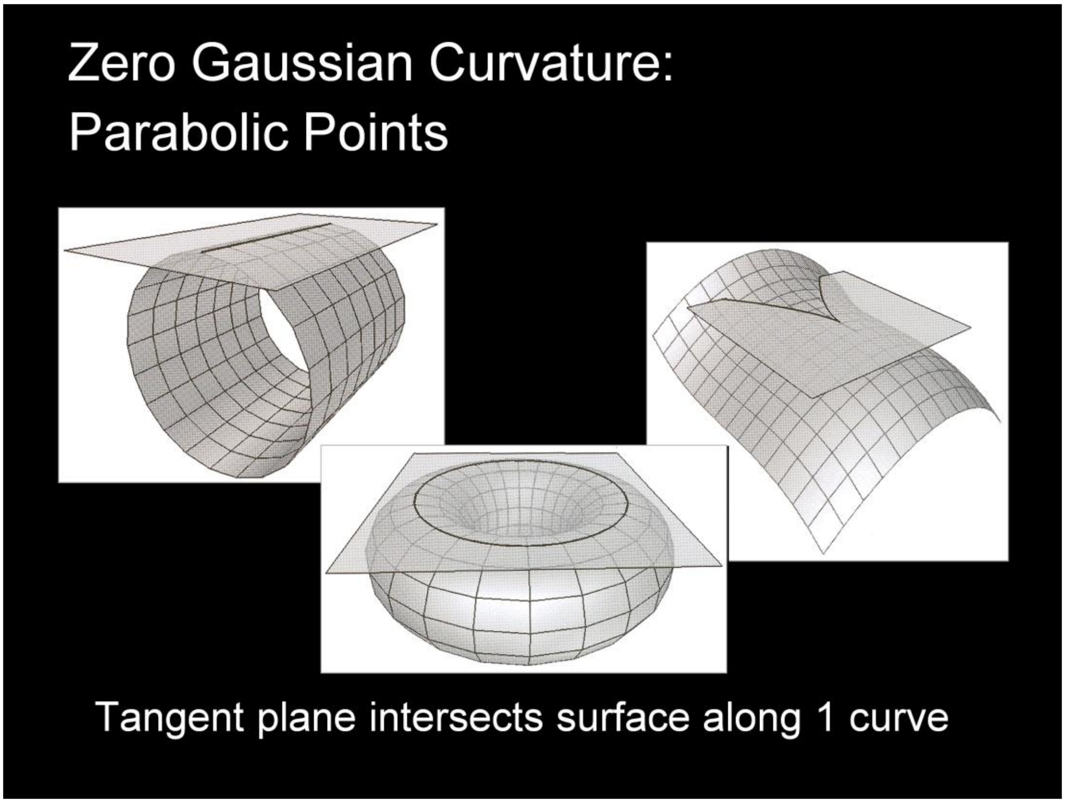
Convex/concave depending on sign of H

Tangent plane intersects surface at 1 point

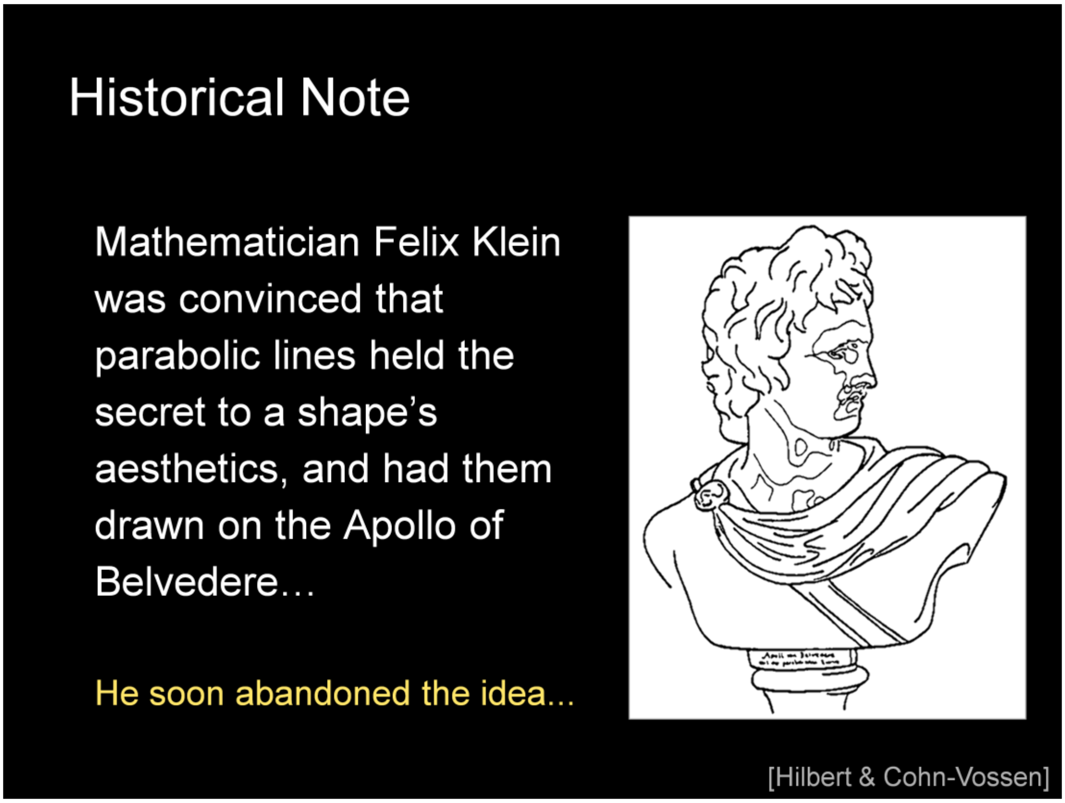
The most basic classification looks at the sign of Gaussian curvature. If it is positive, then the principal curvatures are either both positive or both negative (and you can tell which one by looking at the sign of the mean curvature). Points of positive Gaussian curvature are known as elliptic points, and are either convex or concave regions.



If the Gaussian curvature is negative, we have what are known as hyperbolic points, at which the surface is saddle-shaped. So, if you look in one direction the surface is convex, while in the perpendicular direction the surface is concave.



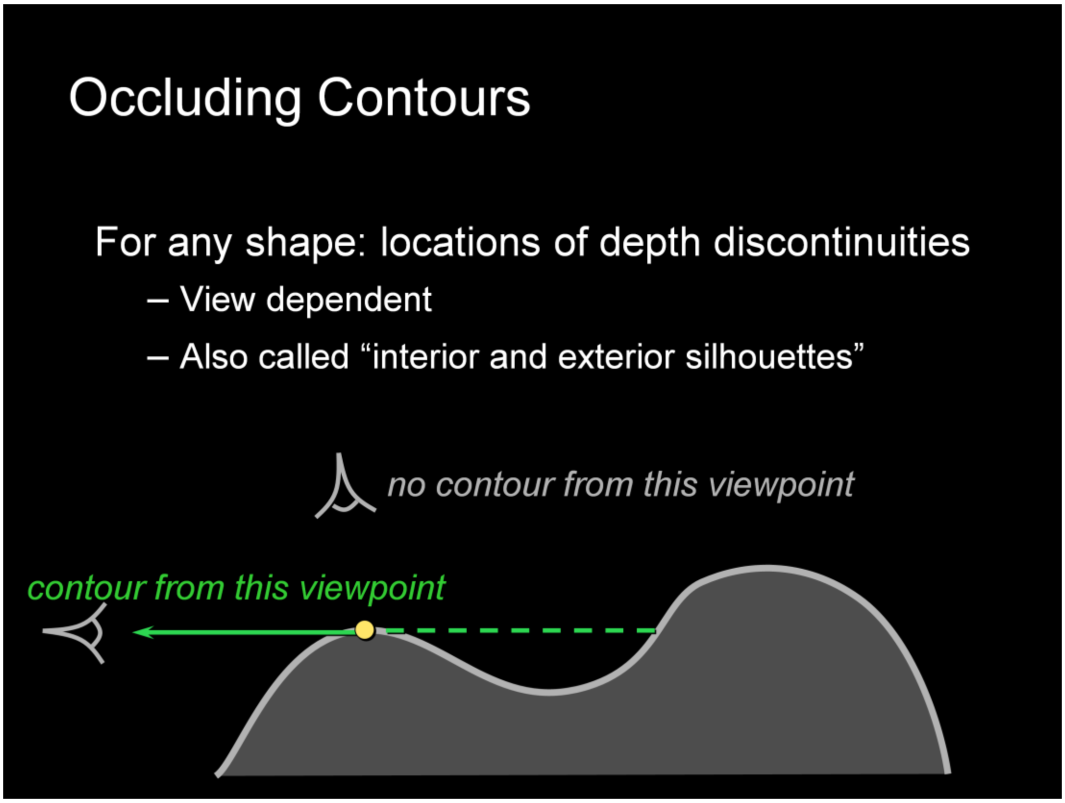
Finally, we have parabolic points, at which one of the principal curvatures is zero. The most basic shape with zero Gaussian curvature is a cylinder, but there are many more complex surfaces at which K=0 as well. In general, except for degenerate cases like cylinders, the parabolic points will form curves on the surface (known as parabolic lines), separating regions of positive and negative Gaussian curvature.



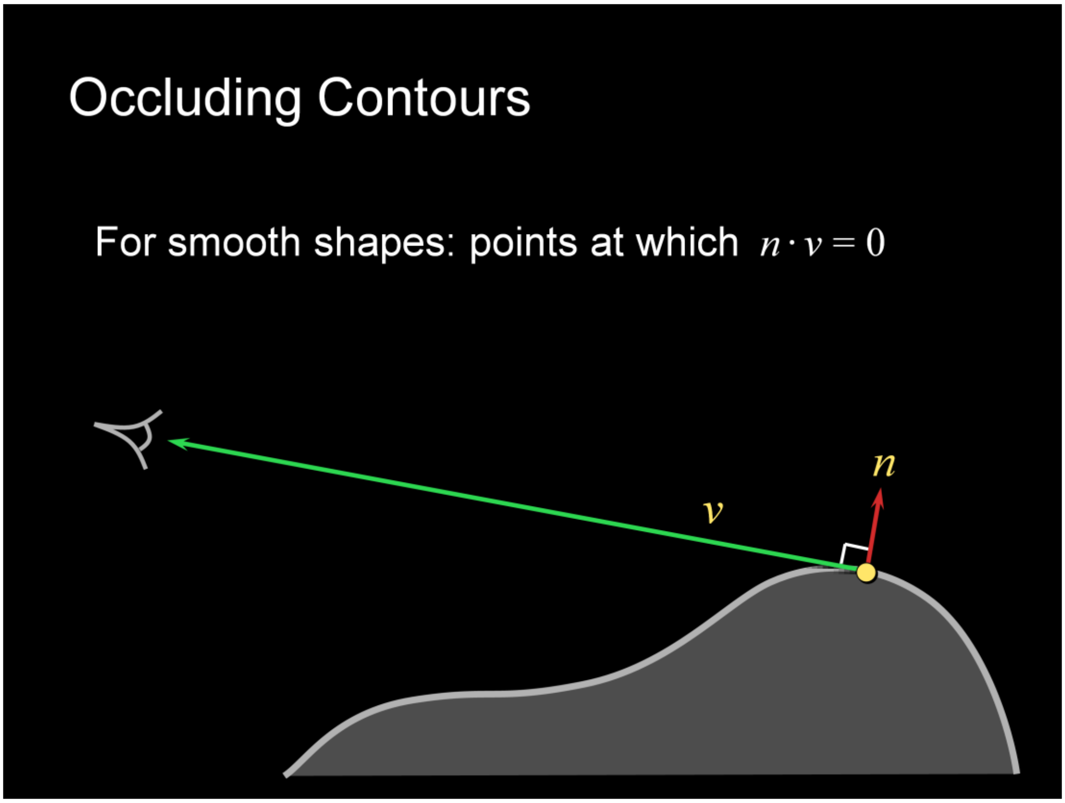
As long as we’re talking about parabolic lines in a course about line drawings, it would be remiss not to relate an anecdote about the mathematician Felix Klein, who thought that parabolic lines might, in fact, be interesting lines to draw on a surface. He had them drawn (probably by some poor grad student) on the Apollo of Belvedere.

Unfortunately, the experiment didn’t turn out that great.

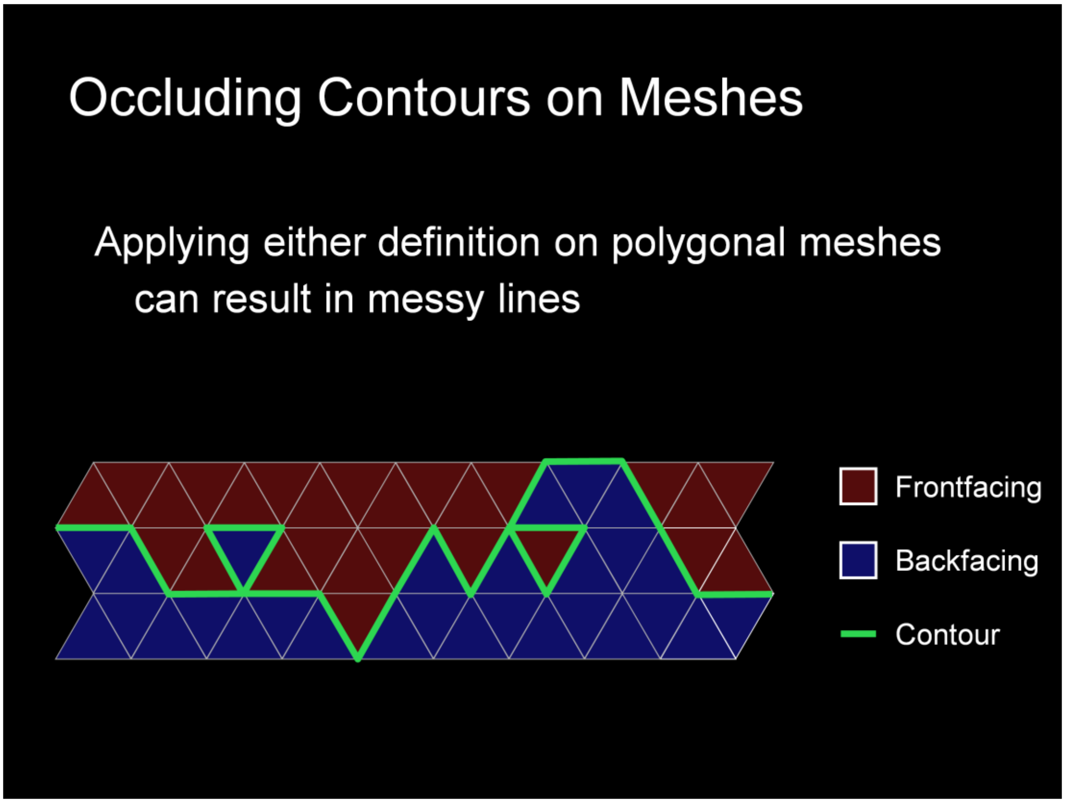
A little later, we’ll see that Klein wasn’t entirely wrong: it is possible to select a subset of the parabolic lines that look OK…



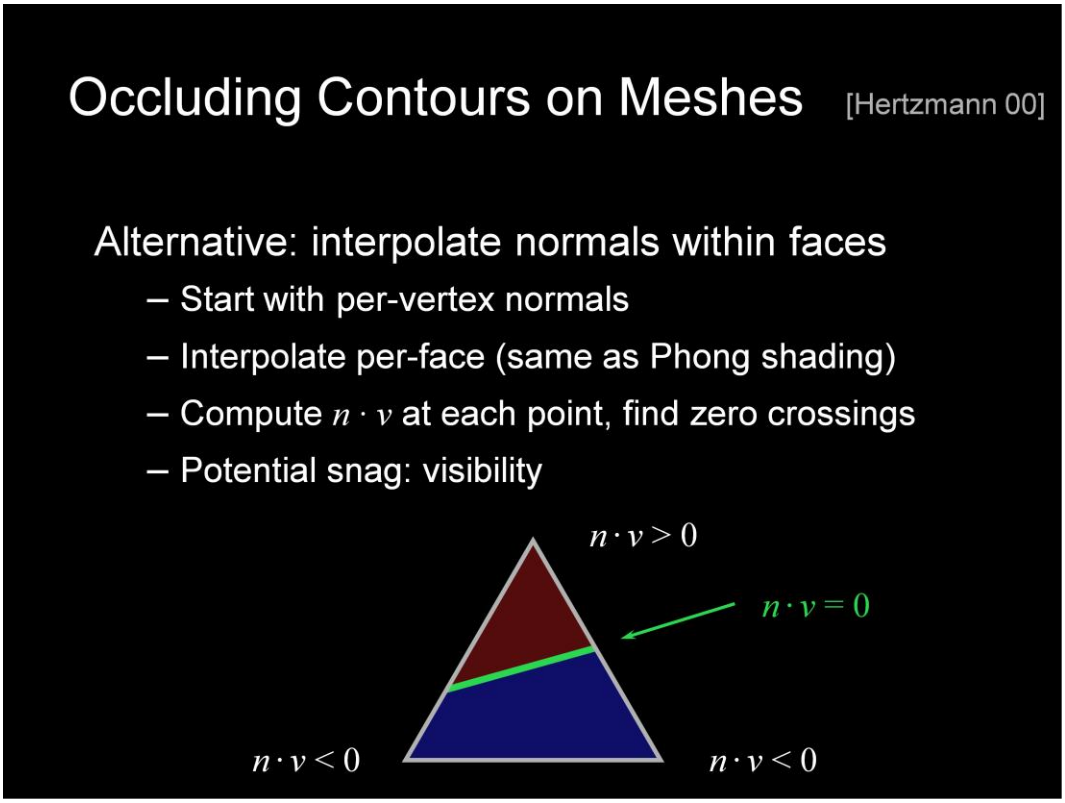
Now that we have some math under our belts, let us look in more detail at the different types of lines. We begin with occluding contours, sometimes also called “interior and exterior silhouettes.” There are a few different ways of defining these, of which a very straightforward definition is simply those locations at which, from the current viewpoint, there is a depth discontinuity. Note that these are view-dependent lines, which implies both benefits and drawbacks. On the plus side, the view dependence makes it much more likely that these lines are interpreted as conveying shape, rather than as surface markings. On the other hand, this means that the lines will have to be recomputed for each frame, and potentially makes it harder to do things like line drawings in stereo.



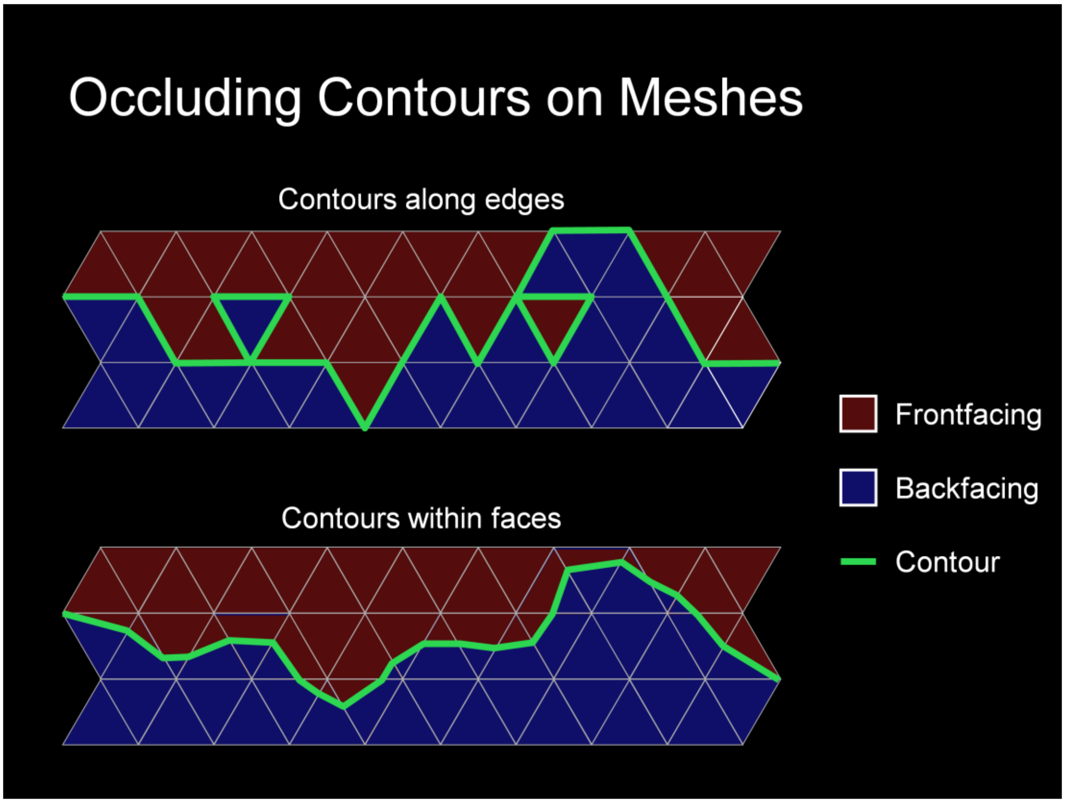
On smooth surfaces, there is another definition of contours that is useful: contours are those surface locations where the surface normal n is perpendicular to the viewing direction v. That is, places where n dot v is equal to zero.



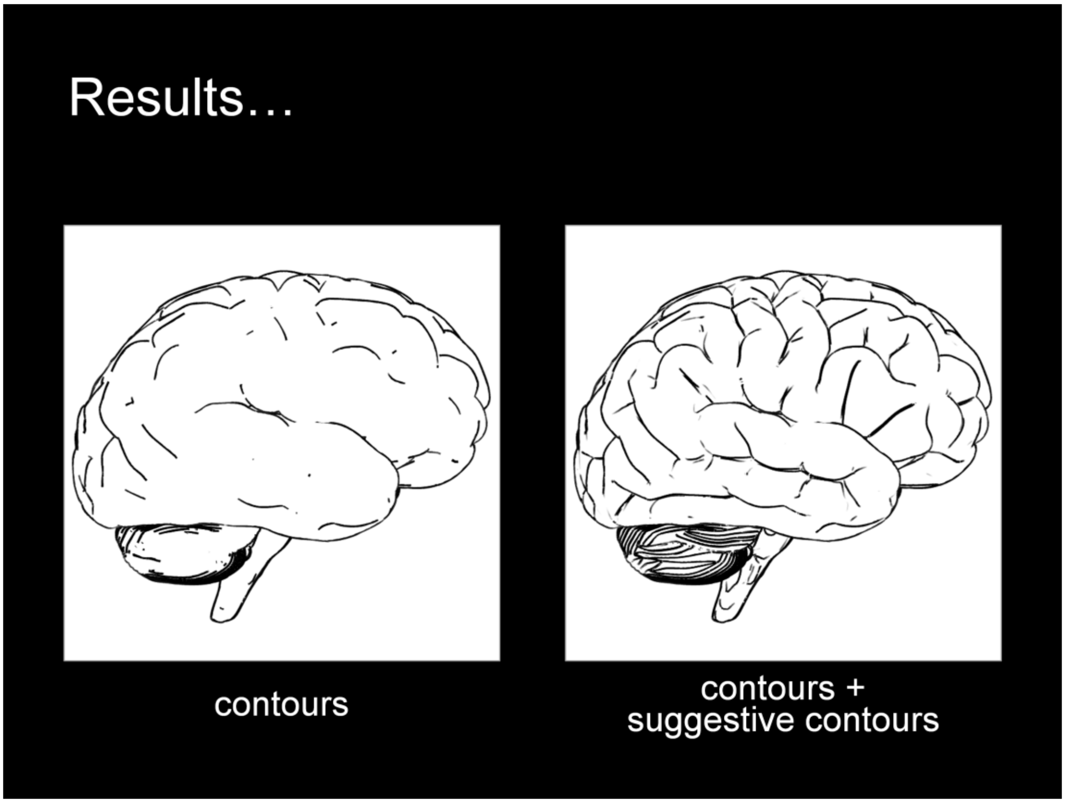
We can apply the same principles to polyhedra (i.e., polygonal meshes), but there's a problem. With only a tiny bit of noise, we can run into a situation where polygons along the boundary are "just barely" front- or back-facing, and the boundary between them is not a simple curve: it can entirely surround certain faces. This isn't necessarily a problem if the only thing you're doing is drawing the curve, since it will be viewed edge-on. However, if you are doing any further processing on the curve, such as trying to draw them with stylization, this can lead to big problems.



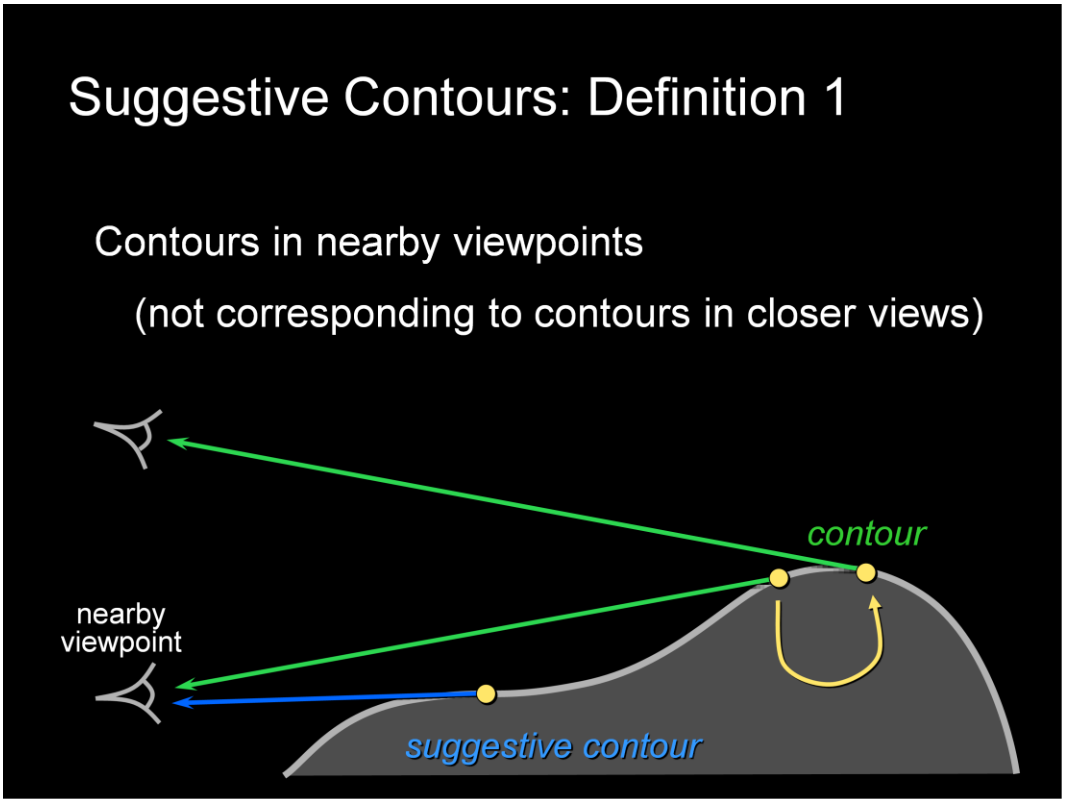
So, there's a frequently-used alternative for polygonal meshes that tends to produce much nicer curves. It is a bit similar to Phong shading, in that it involves starting with per-vertex normals and interpolating them across a face. Once you know n at each point, you can find n dot v, and locate the curve on the face that corresponds to $n \cdot v = 0$. A slightly simpler variant of this is to just compute n dot v at the vertices, interpolate across the face, and figure out where the zero crossing is.



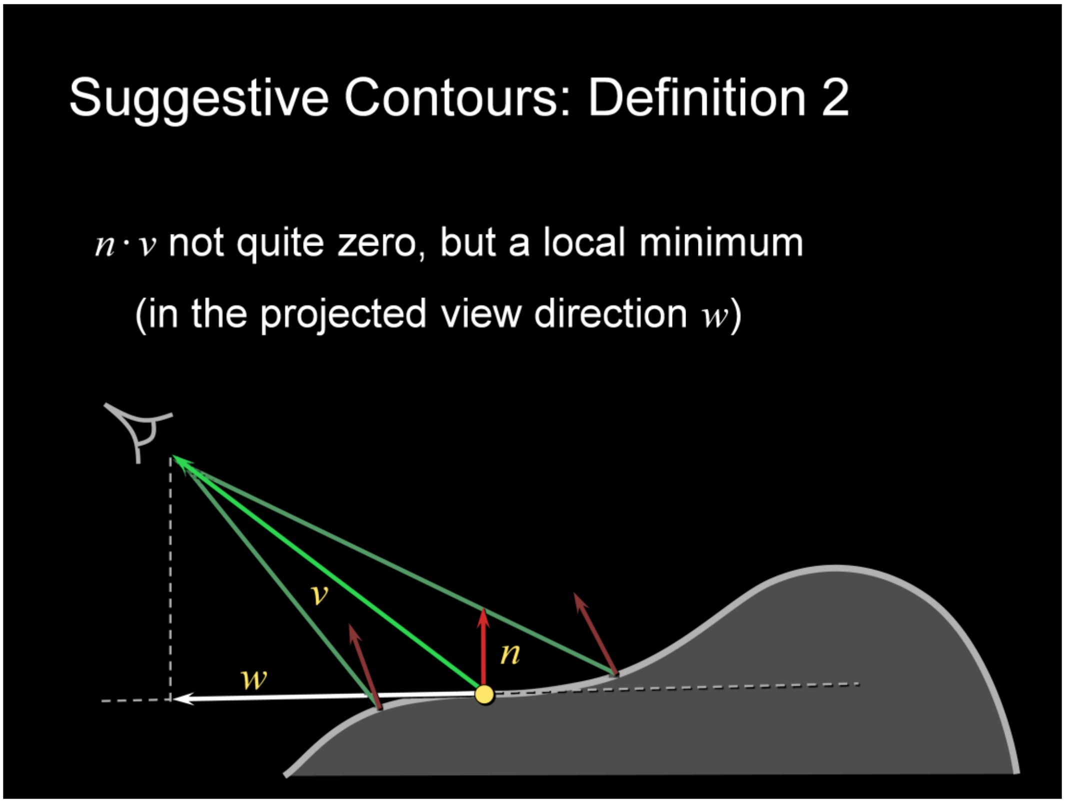
We can apply the same principles to polyhedra (i.e., polygonal meshes), but there's a problem. With only a tiny bit of noise, we can run into a situation where polygons along the boundary are "just barely" front- or back-facing, and the boundary between them is not a simple curve: it can entirely surround certain faces. This isn't necessarily a problem if the only thing you're doing is drawing the curve, since it will be viewed edge-on. However, if you are doing any further processing on the curve, such as trying to draw them with stylization, this can lead to big problems.



Here is your brain on contours. Here is your brain on suggestive contours. Any questions?



Now that we've seen contours, let's move on to defining suggestive contours. Here's the first definition: contours in nearby views. What happens if we start with the viewpoint at top (which produces a contour), then move the viewpoint down a little bit? First, the contour slides along a surface to a new location where its surface normal is perpendicular to the new view direction. But something else happens here too. A new contour appears that does not correspond to any in a closer viewpoint. This is a suggestive contour from the original viewpoint. The other contour corresponds to that in the original viewpoint, and is not a suggestive contour. Adding this qualification to our definition completes it.



While intuitive, the first definition doesn't really lead to efficient algorithms for computing suggestive contours. So, let's look at a second definition (which can be proven equivalent to the first one). The idea is that suggestive contours are places where n dot v doesn't quite make it to zero (at which point we'd have a contour), but is a local minimum on the surface. That is, the location of a suggestive contour from this viewpoint (assumed to be distant) is where the normal is locally closest to perpendicular to the view direction, as you consider points along this normal slice of the surface. This involves moving in the direction "w", which we define to be the projection of v, the view direction, into the local tangent plane of the surface.

Minima vs. Zero Crossings

Definition 2: Minima of $n \cdot v$

Finding minima is equivalent to:

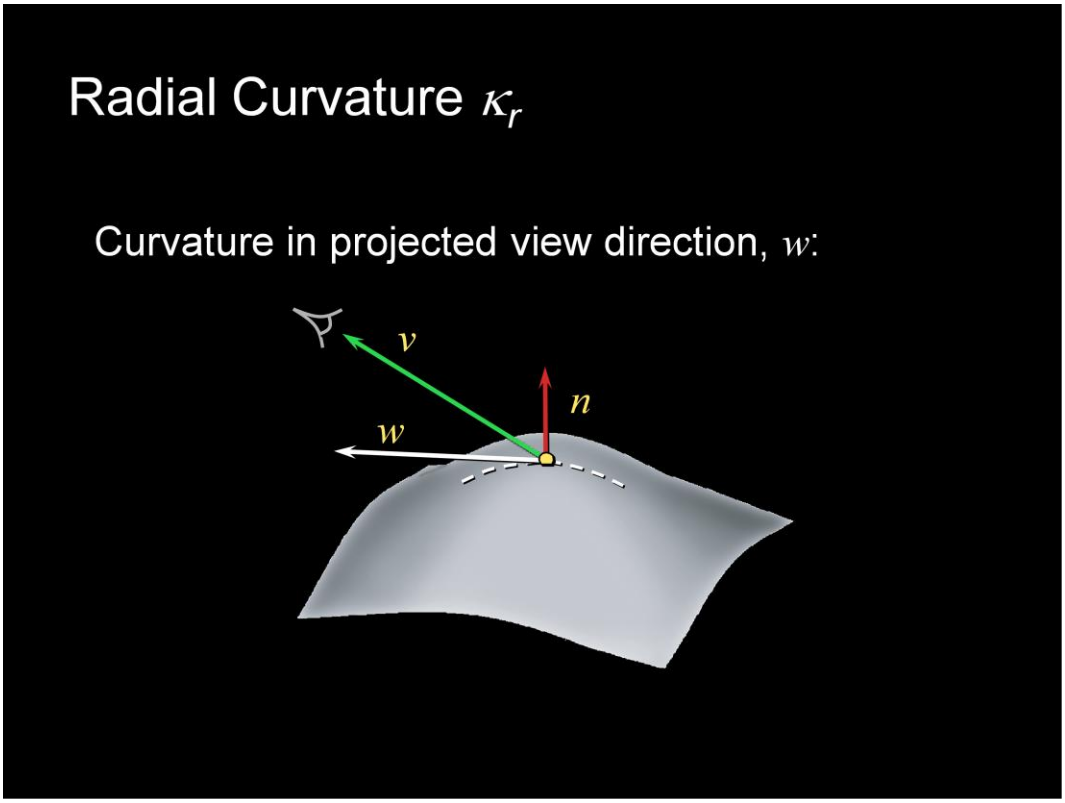
finding zeros of the derivative

checking that 2nd derivative is positive

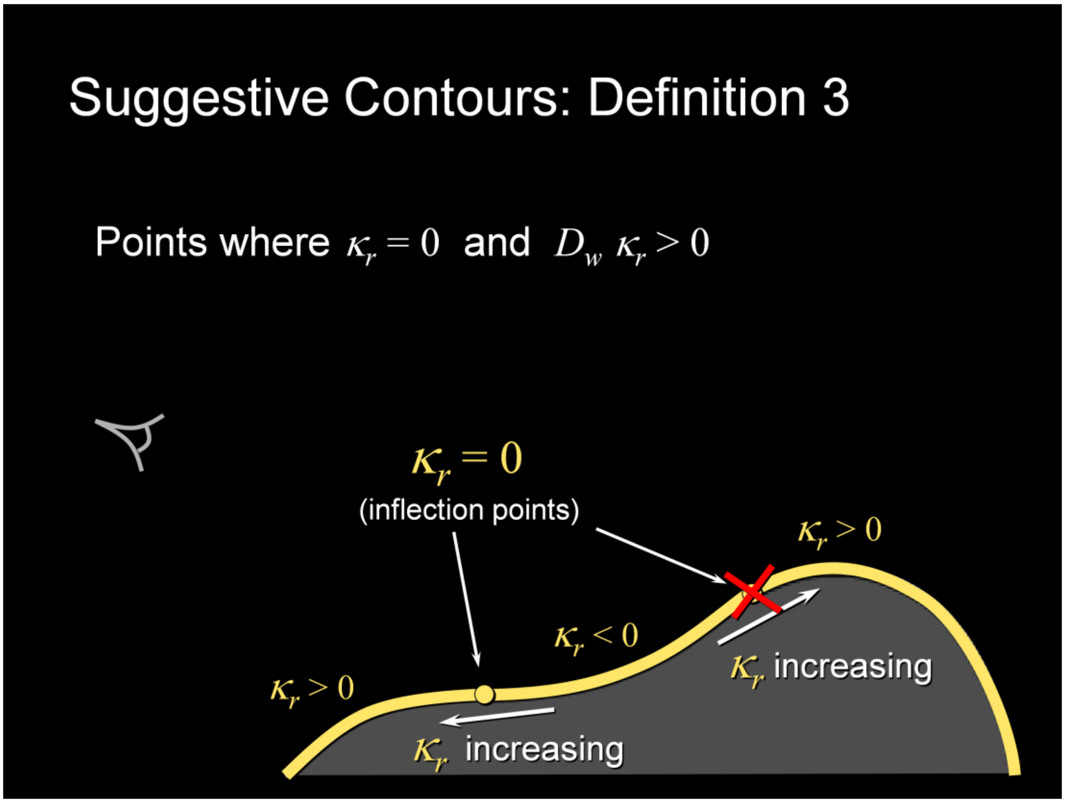
This leads to definition 3.

Derivative of $n \cdot v$ is a form of curvature…

While definition 2 is better from the point of view of computation, we can transform it into yet another form that is still more convenient. The basic idea is that we're looking for local minima, so we use the usual definition that minima are places where the derivative is zero, and the next higher-order derivative is positive (which distinguishes them from maxima). Now, it turns out that derivatives of n dot v are related to curvature.



In particular, the derivative of n dot v has the same zeros as a quantity called "radial curvature", which is just the curvature in the direction w (which, you'll recall, is the projection of the view direction).



So, our third definition of suggestive contours is that they are zeros of radial curvature, subject to a derivative test. This test needs to enforce that the "directional derivative" of radial curvature, in the direction w, is positive. To figure out what that is, we'll need to go back and look at the next higher order of surface differentials.

Finding Suggestive Contours

Finding κ_r:

$$\kappa_r = \mathbf{II}(\hat{w}, \hat{w})$$

Finding $D_w \kappa_r$:

?

So, to recap, the most computationally convenient definition of suggestive contours involves finding the zeros of radial curvature, which you compute by multiplying II by w twice, then checking the sign of the directional derivative of radial curvature, which you get by multiplying the C tensor by w three times (there’s also an extra term due to the chain rule, which accounts for the change of w itself as you move in the w direction).

Derivative of Curvature

Just as $\mathbf{II} = \begin{pmatrix} \frac{\partial n}{\partial u} & \frac{\partial n}{\partial v} \end{pmatrix}$, can define $\mathbf{C} = \begin{pmatrix} \frac{\partial \mathbf{II}}{\partial u} & \frac{\partial \mathbf{II}}{\partial v} \end{pmatrix}$

$\mathbf{C}$ is a rank-3 tensor or “cube of numbers”

Symmetric, so 4 unique entries: $\mathbf{C} = \begin{pmatrix} P^{Q} & Q^{S} \\ Q^{S} & S^{T} \end{pmatrix}$

Multiplying by a direction three times gives (scalar) derivative of curvature

Once we know about curvatures, derivatives of curvature are really nothing special. The only really interesting thing is that, as opposed to the normal (which was a vector) and the second fundamental matrix II, the derivative of curvature is now a three-dimensional tensor, which can be thought of as a vector of matrices or as a “cube of numbers”. In order to get the derivative of curvature in a particular direction, you multiply this tensor by that direction three times.

Finding Suggestive Contours

Finding κ_r:

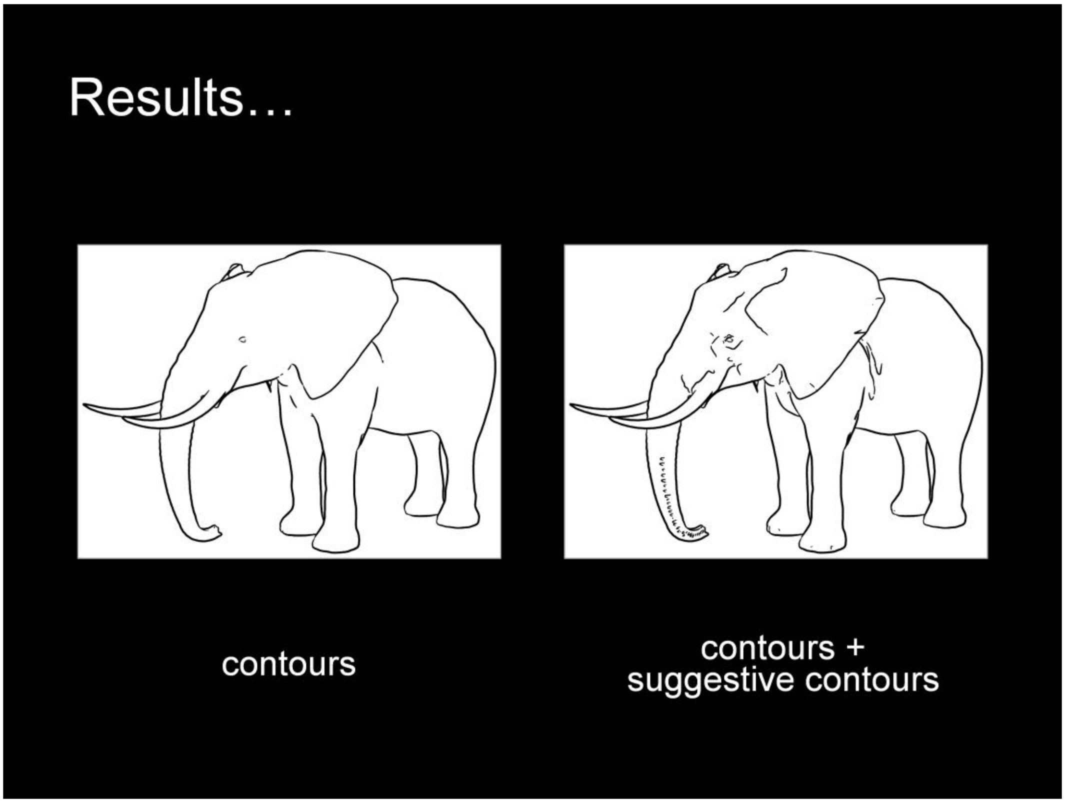
$$\kappa_r = \mathbf{II}(\hat{w}, \hat{w})$$

Finding $D_w \kappa_r$: (extra term due to chain rule)

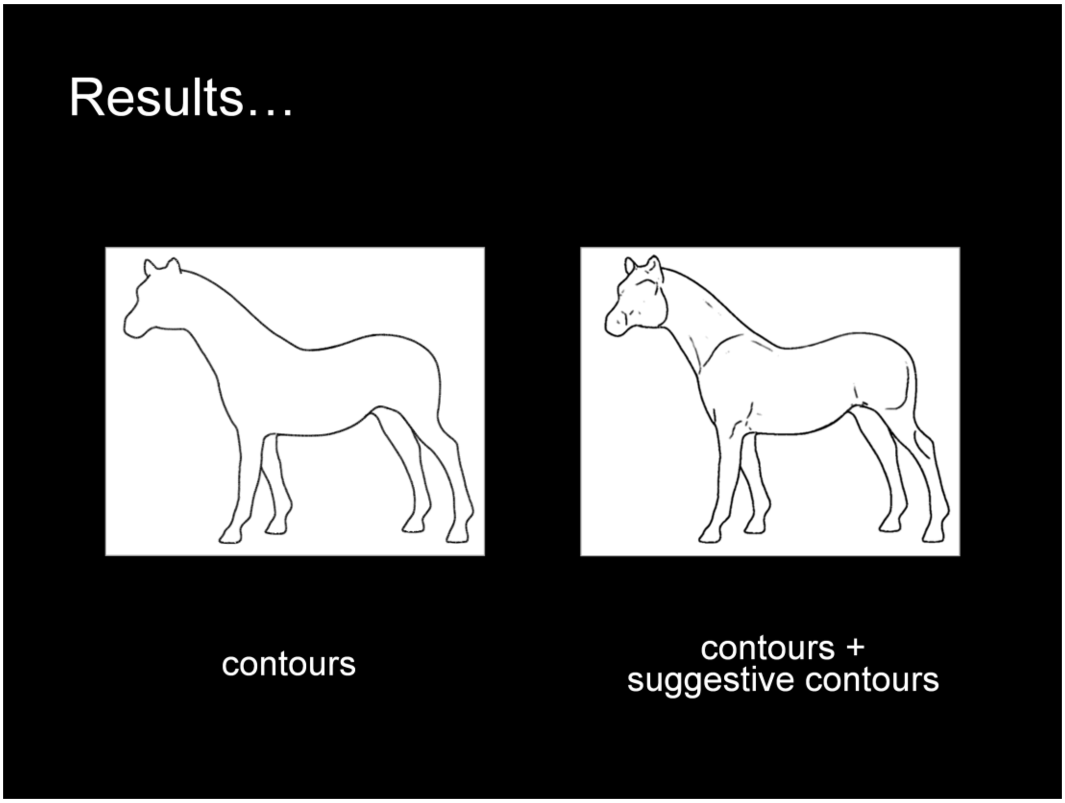
$$D_{\hat{w}} \kappa_r = \mathbf{C}(\hat{w}, \hat{w}, \hat{w}) + 2K \cot\theta,$$

$$\text{where } \kappa_r = 0$$

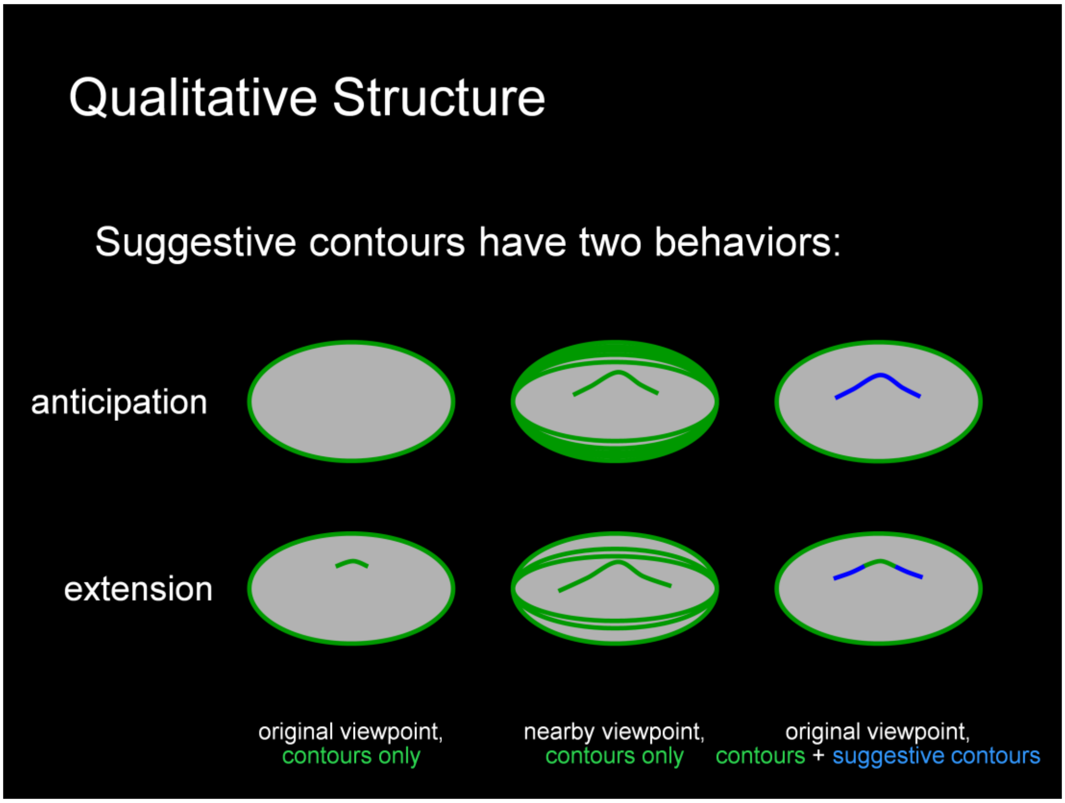
So, to recap, the most computationally convenient definition of suggestive contours involves finding the zeros of radial curvature, which you compute by multiplying II by w twice, then checking the sign of the directional derivative of radial curvature, which you get by multiplying the C tensor by w three times (there’s also an extra term due to the chain rule, which accounts for the change of w itself as you move in the w direction).



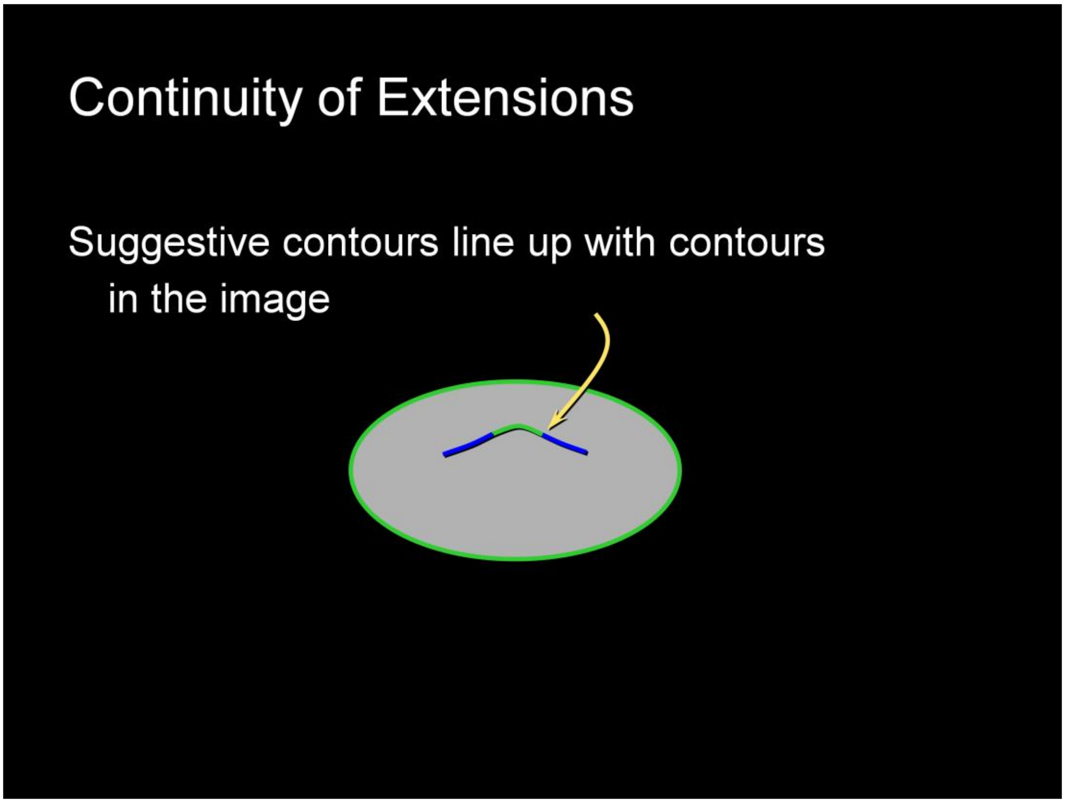
Here is your elephant on contours… (rest of not-a-joke omitted in the interests of good taste)



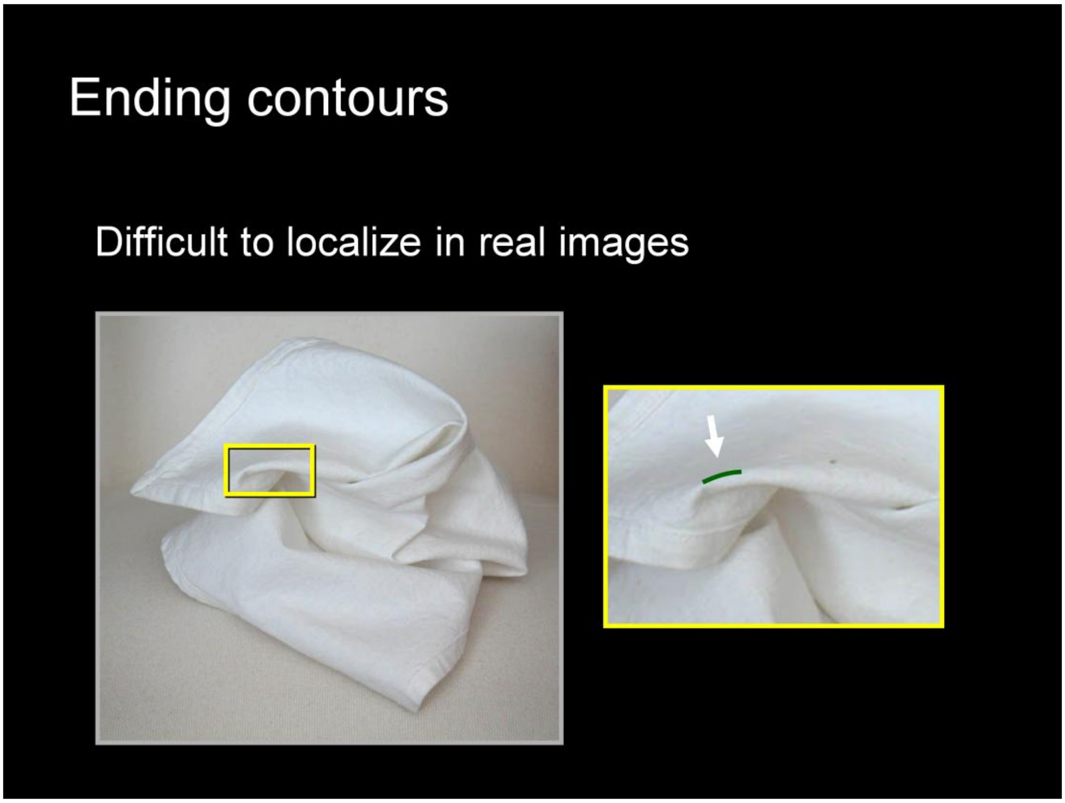
Here's another example…



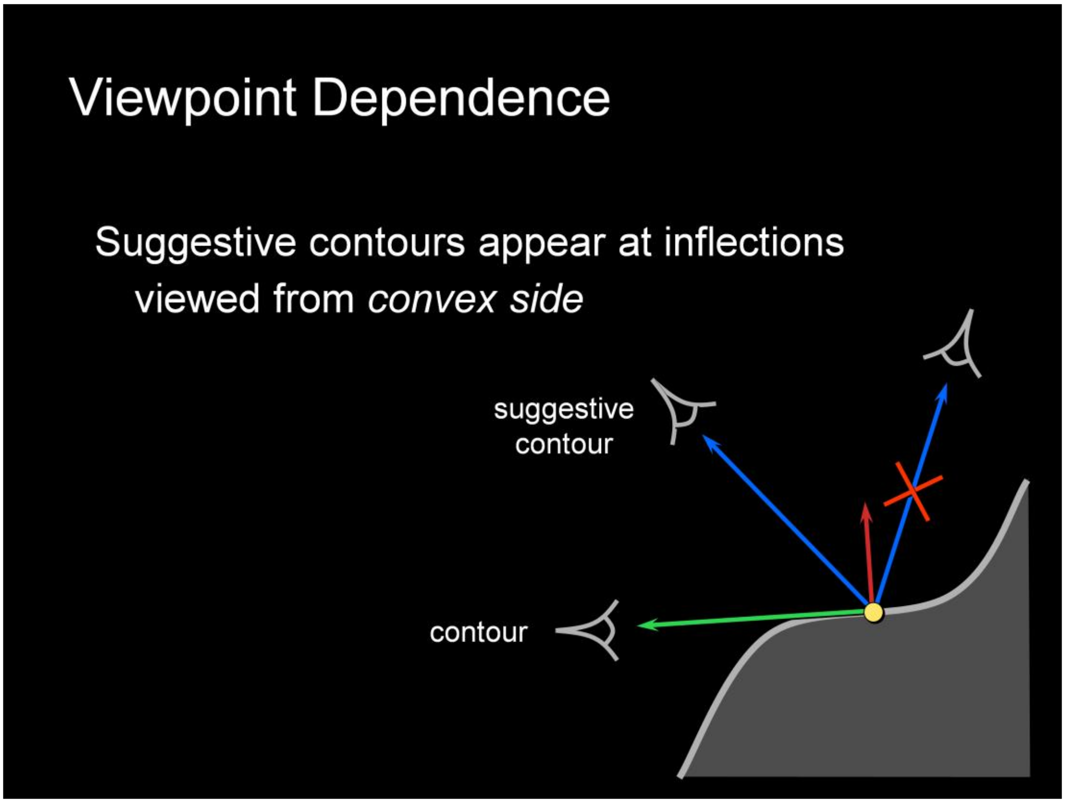
It turns out that suggestive contours have some nice properties that let them complement contours very nicely. First, they can either "anticipate" contours by showing up in nearby viewpoints (i.e., definition 1), or "extend" contours in one view. Here we use the color convention that contours are green while suggestive contours are drawn in blue.



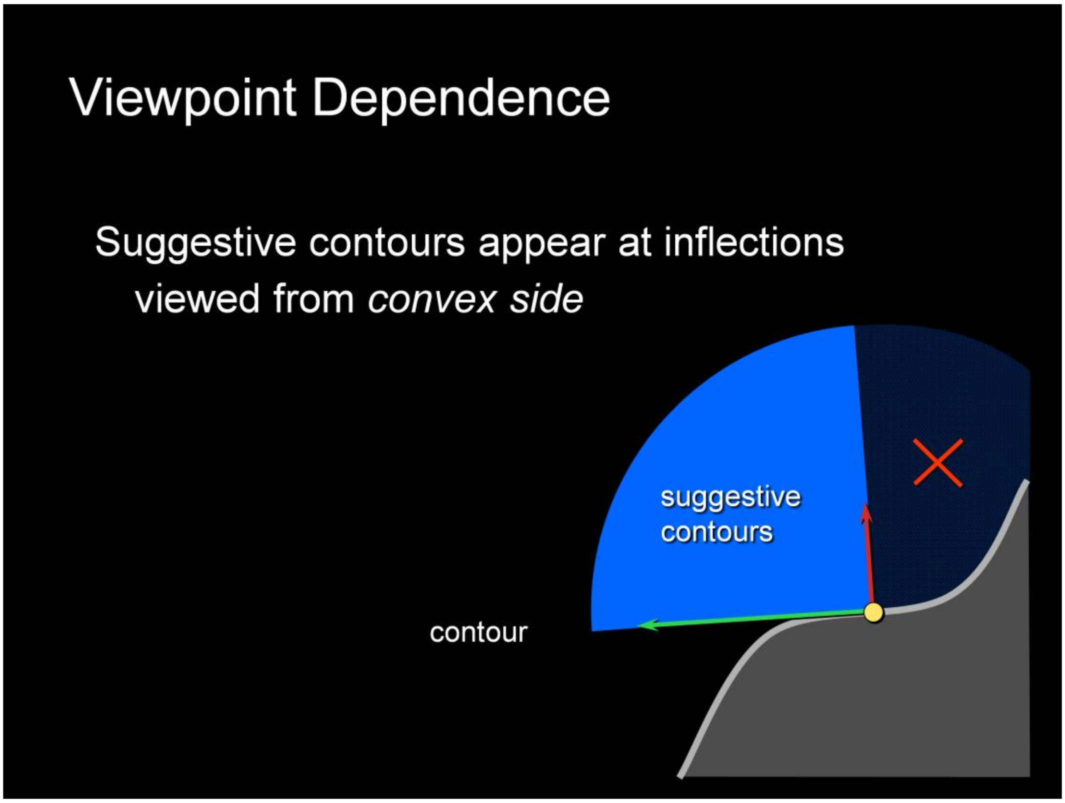
Moreover, in the case of extension, the suggestive contours line up (with G^1 continuity) with contours in the image.



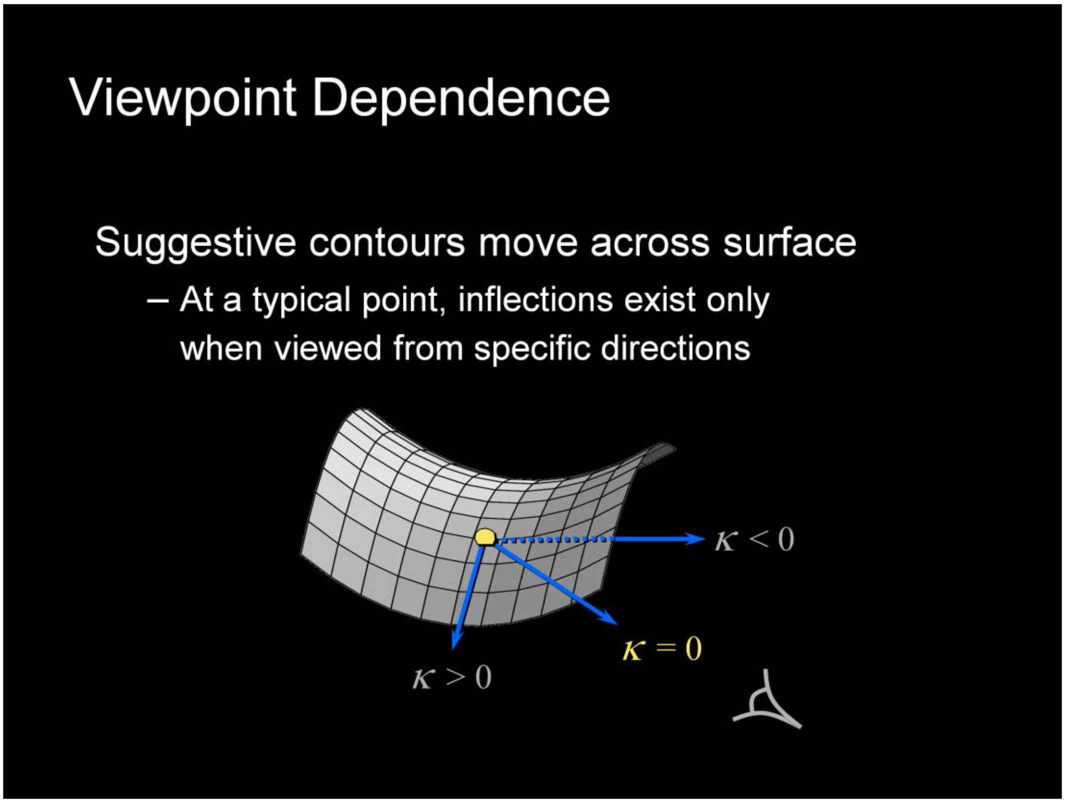
The above property is useful, because in real-world images it can in fact be difficult to tell exactly where a contour ends. The fact that suggestive contours extend contours smoothly means that there is a single line that corresponds well to the behavior visible in such cases.



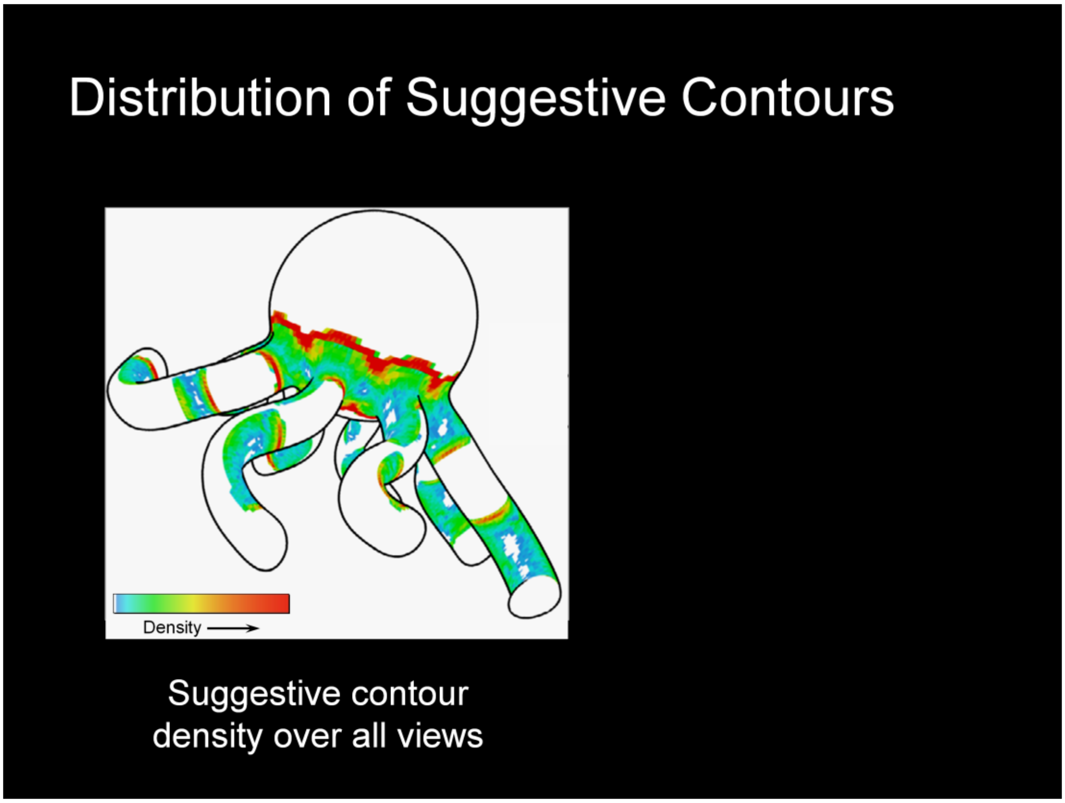
Another property that becomes apparent from definition 3 is that contours show up at inflections (of normal curves) on the surface, but only when viewed from the convex side. In this case, the derivative test eliminates the suggestive contour at the rightmost viewpoint.



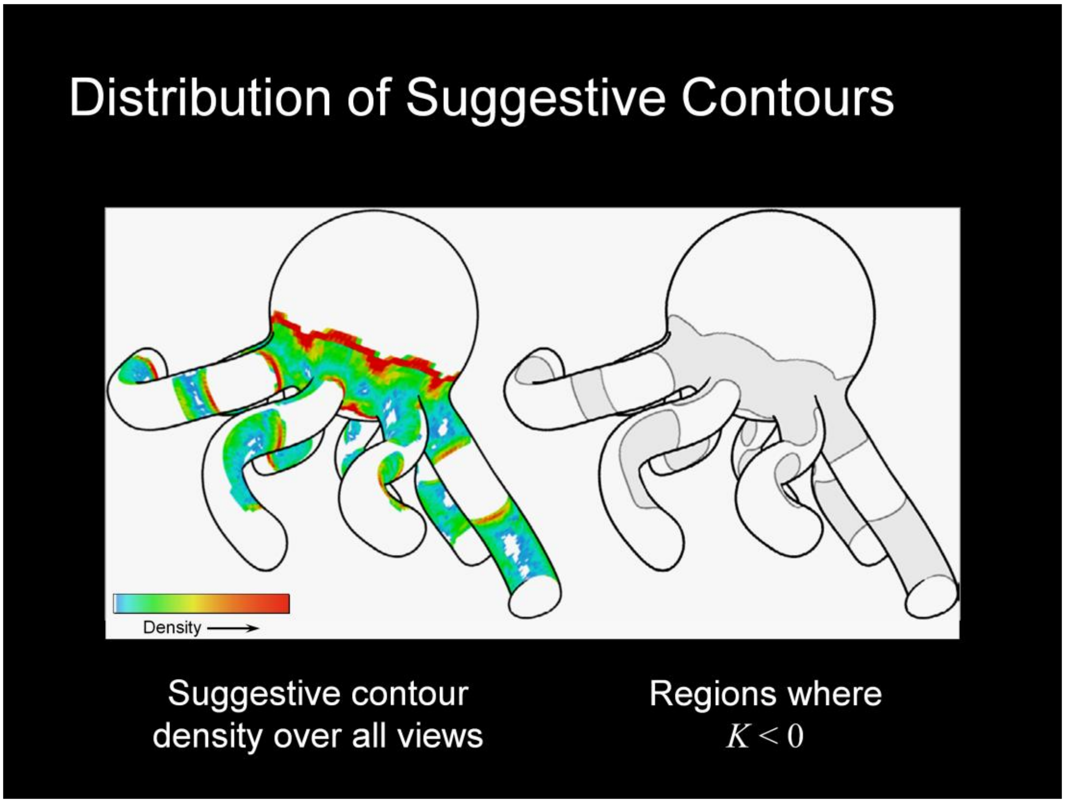
So, considering an inflection on a surface, there is some region of viewpoints from which suggestive contours get drawn, a region where they don't, and a threshold direction at which you start getting contours.



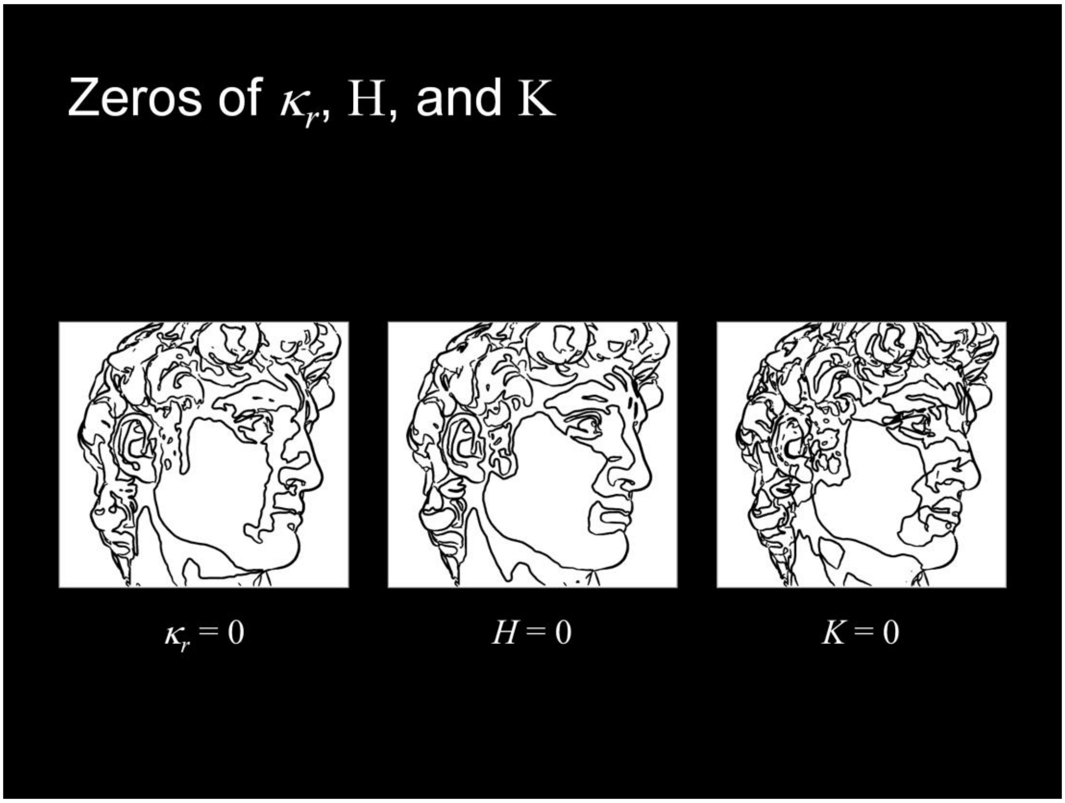
Moving the viewpoint out of the plane, we see that suggestive contours can only happen when a surface is viewed from a very particular direction such that the curvature is zero. If you rotated the viewpoint one way, you'd get positive curvatures, and negative curvatures if you rotated the other way. Note also that having a direction for which the curvature is zero implies that the principal curvatures (which are the minimum and maximum limits for normal curvature) can't be both positive or both negative. This, in turn, implies that in order to get a suggestive contour, the Gaussian curvature must be negative (or at worst zero).



This property can be illustrated empirically as well. Here we've taken a model and plotted a histogram of how many views have suggestive contours at each point on the surface.

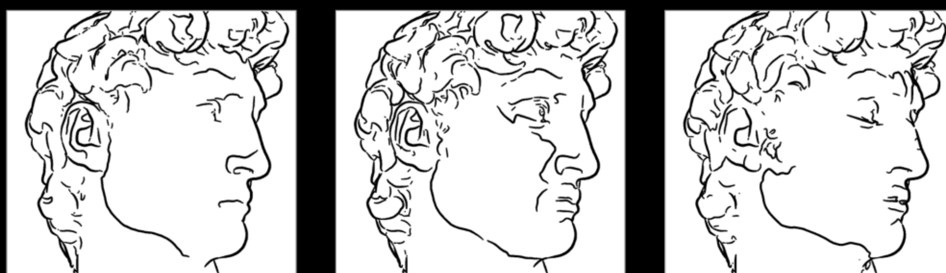


Comparing this to regions of negative Gaussian curvature, we see complete agreement. We also see the surprising fact that suggestive contours tend to hug the lines of zero Gaussian curvature (i.e., our friends the parabolic lines). We'll see later how to show this mathematically, but meanwhile let's think back to Klein's experiment. Even if the suggestive contours were always close to the parabolic lines, there's still a big difference between drawing them and our definition of suggestive contours: the derivative test.



In fact, if we didn't apply the derivative test, the lines of zero radial curvature would look pretty bad: just as bad as drawing all the parabolic lines. (Here we also show zeros of mean curvature for the sake of completeness.)

Zeros of κ_r, H, and K (with derivative tests)

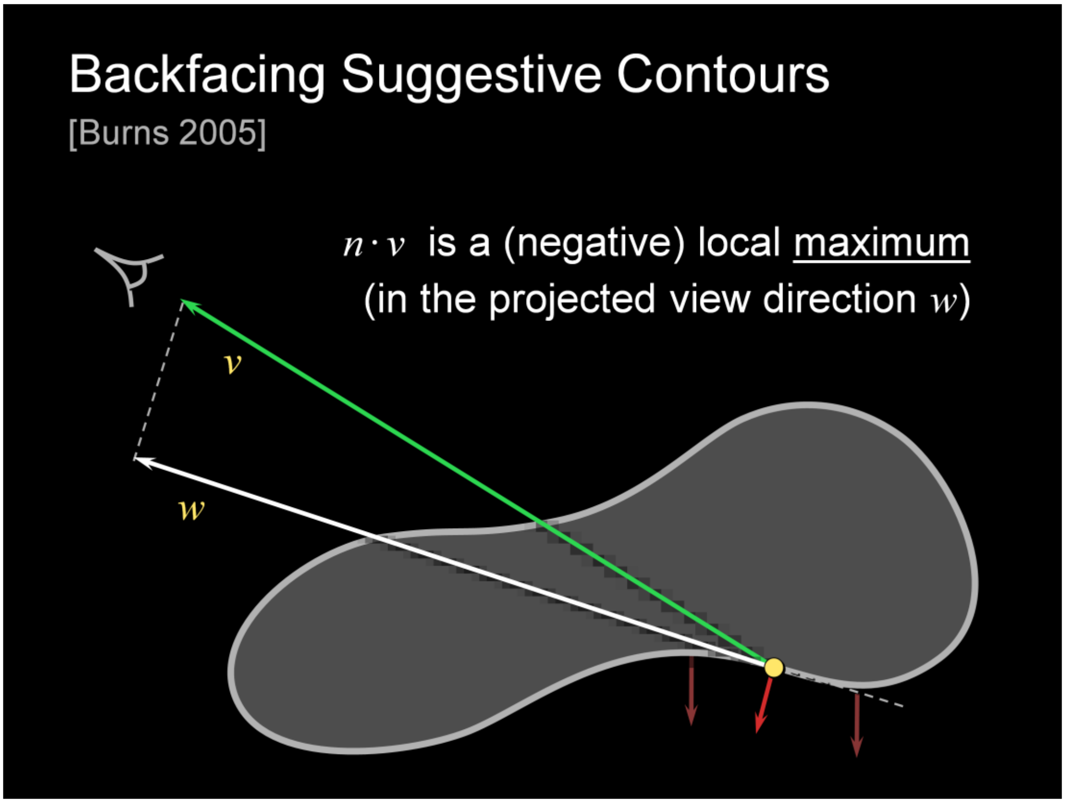


$\kappa_r = 0$, $D_w \kappa_r > 0$

$H = 0$, $D_w H > 0$

$K = 0$, $D_w K > 0$

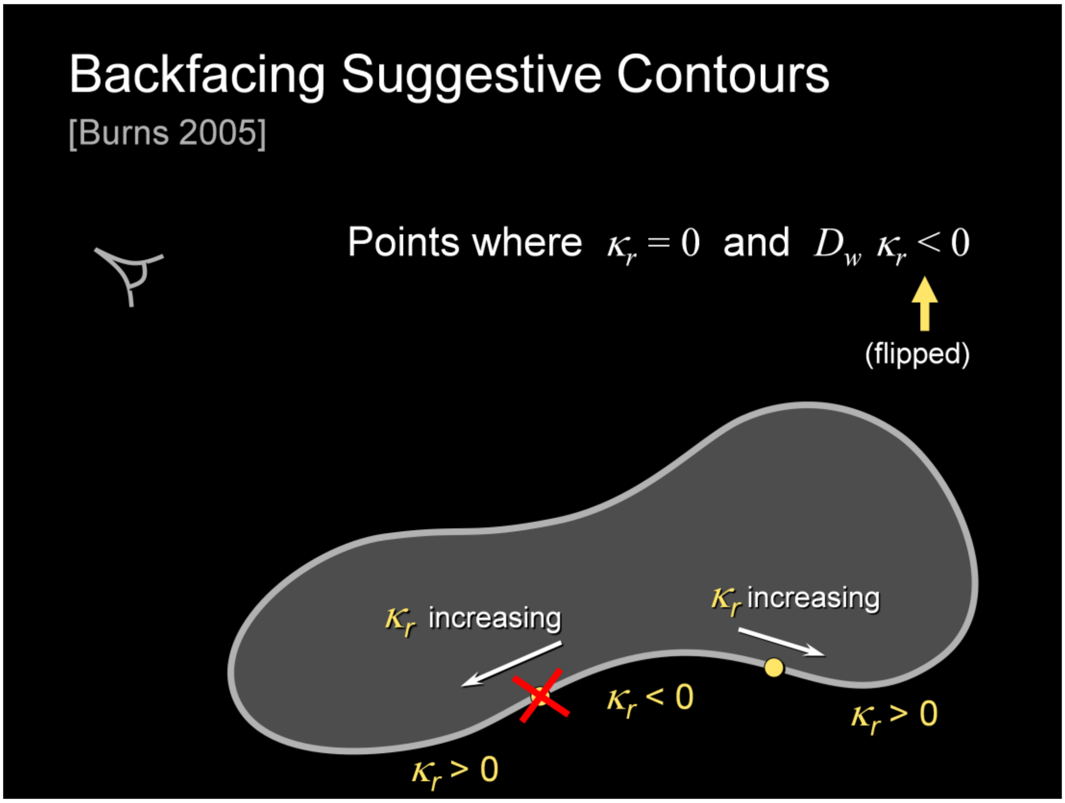
If we add a derivative test, you can see that parabolic lines suddenly don’t look so bad, though in general the suggestive contours still look better (and have the other properties of lining up with contours, etc.)



We also need to consider what happens with suggestive contours on backfacing parts of the object; such lines are an effective ingredient in transparent renderings. In fact, this was worked out in a paper on volumetric line drawings at Siggraph 2005.

The first definition of suggestive contours still applies: contours in nearby viewpoints.

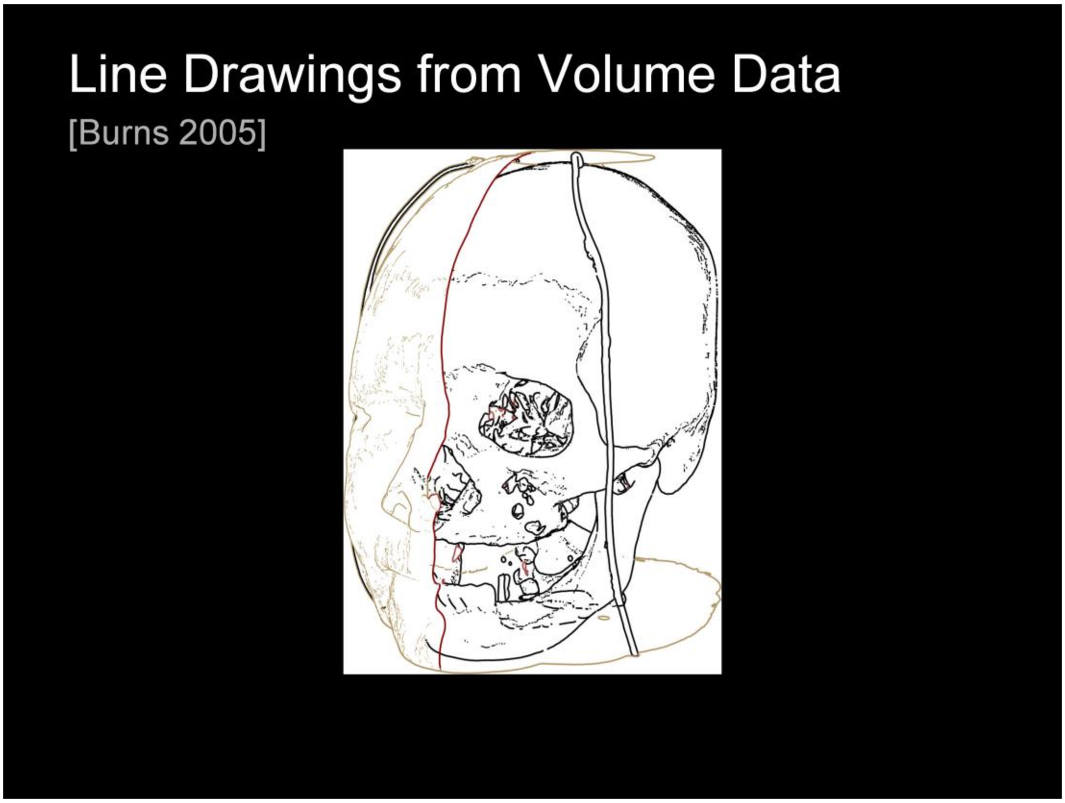
We need to change definition 2 a little bit. When looking at how values of n dot v change across the surface, we are still looking for places where n dot v is almost but doesn’t quite reach zero. The difference is that we’re now looking for *maxima* of n dot v: negative maxima.



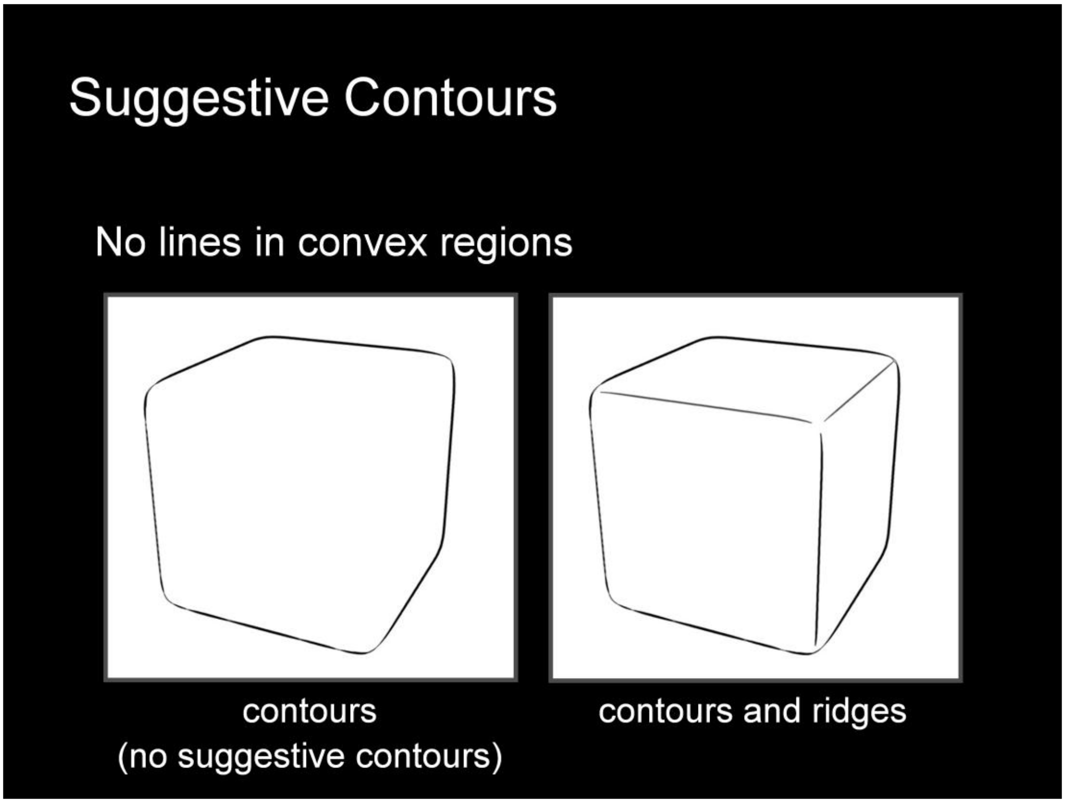
For the third definition, we are now looking for places where the radial curvature is zero where the radial curvature is increasing AWAY from the camera. So the sign of our derivative test gets flipped.

Perhaps a good way of thinking about this is that for backfaces, we're just considering what the inside-out version of the surface looks like (where the normals and curvatures are negated).

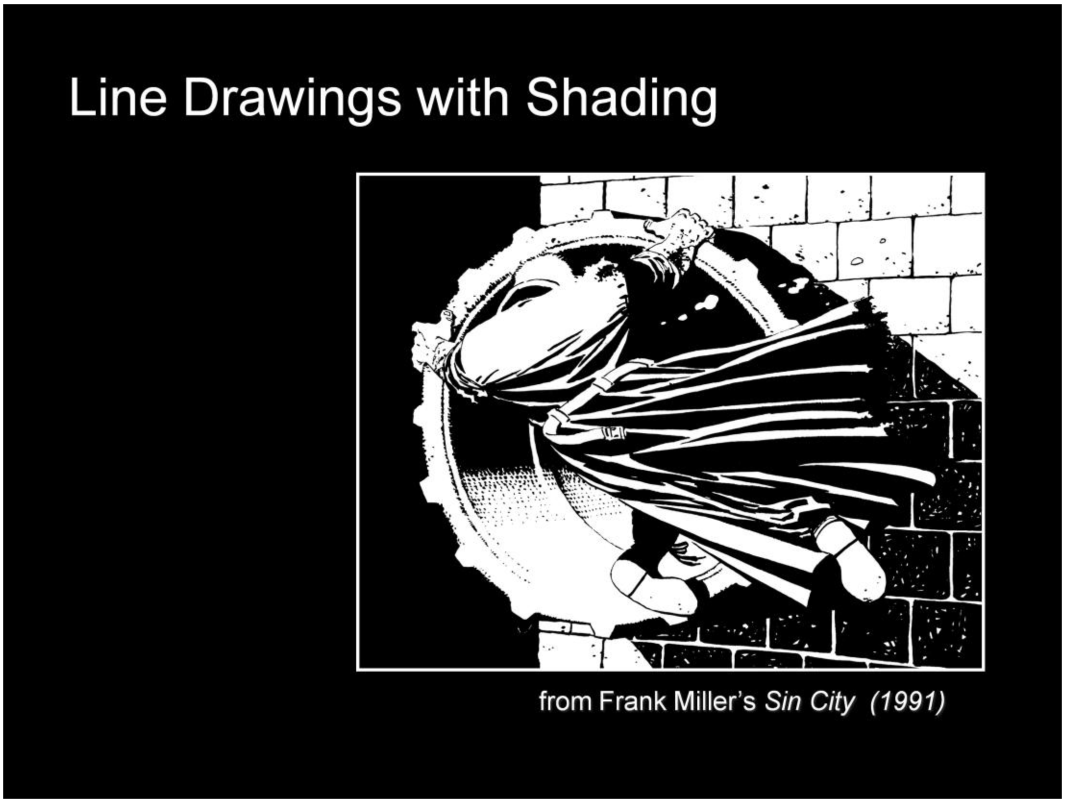
Of course, these suggestive contours still smoothly extend transparently rendered contours.



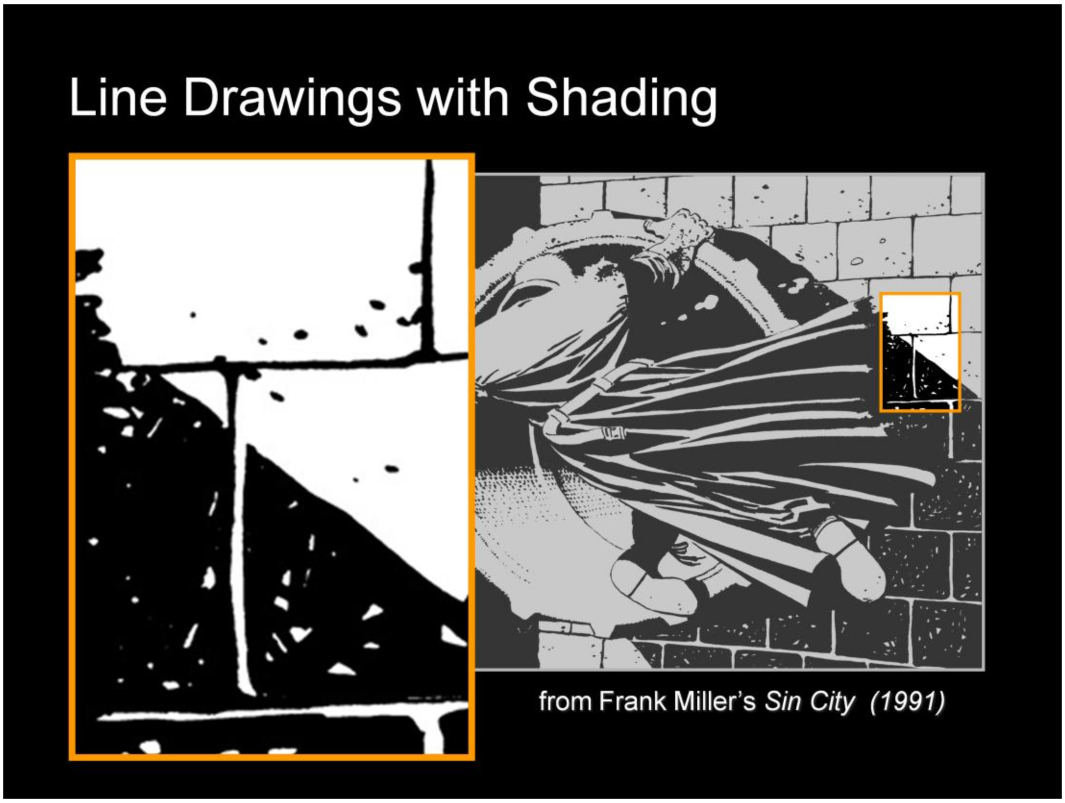
So, here's an example of contours and suggestive contours (together with cutting-plane intersections) produced from volume data.



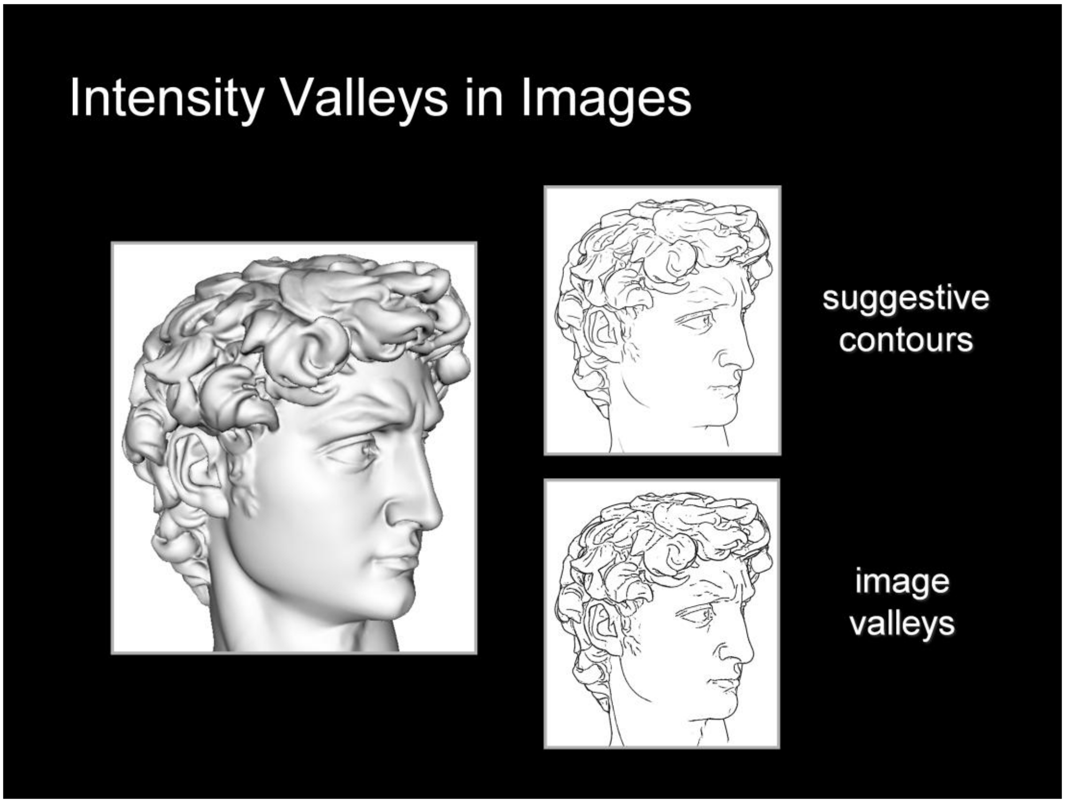
Although (it is our belief that) suggestive contours are useful, there are shapes for which even this is not enough. For example, convex surfaces such as this rounded cube have no suggestive contours, yet probably need some more lines to be conveyed clearly.



Another situation in which it is not clear that suggestive contours provide the right answer are two-tone comics, such as this example by Frank Miller.

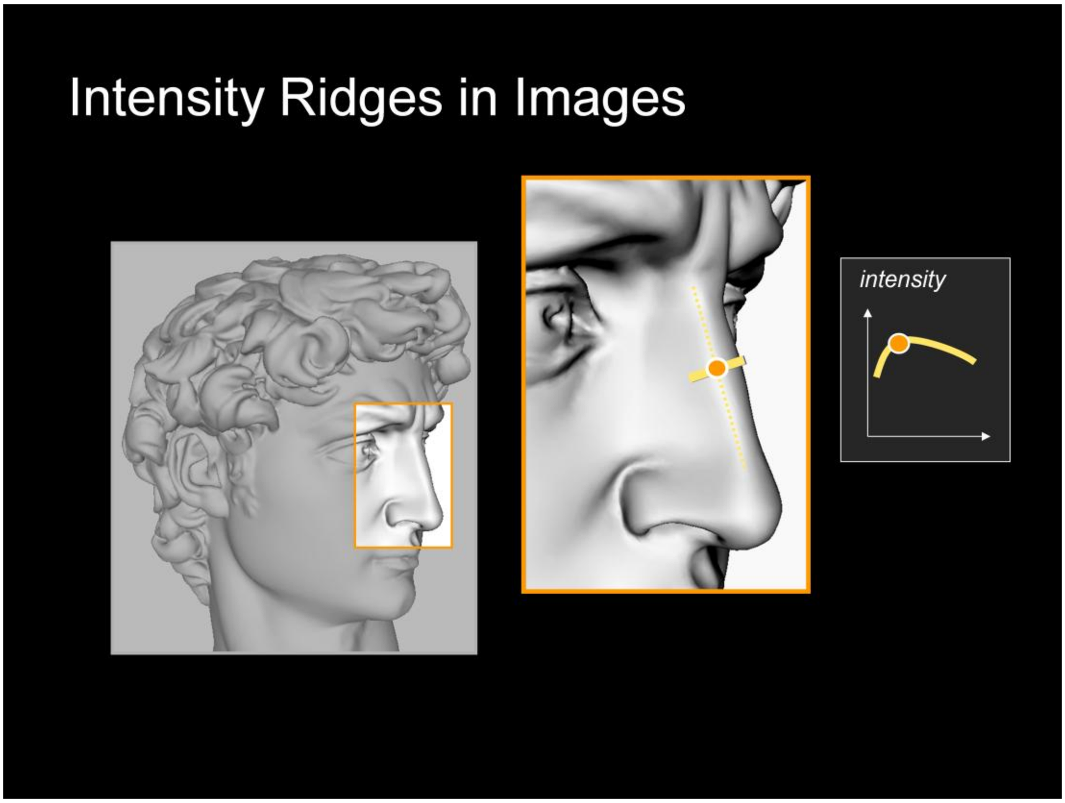


Zooming in, we see that the black line in the white region, which might be a suggestive contour, does not line up with the white line in the black region. Therefore, we hypothesize that there is something else going on with those white lines.



To see how to add more lines, it is useful to step back a moment and compare suggestive contours with some image-space lines. For example, starting with the head-lit diffuse-shaded view at left, we see that there is a remarkable match between the valleys of illumination and the drawing containing contours and suggestive contours.

(Incidentally, there isn't a perfect match, especially where the surface is twisting in weird ways, and there is ongoing research to characterize the exact conditions under which these two families of lines match.)



So, in looking for more families of lines to complement contours and suggestive contours, it makes sense to look at the other flavor of image intensity extrema: ridges of illumination.

Intensity Ridges in Images

Assume:

- Light at camera
- Lambertian material

What lines on the surface correspond to intensity ridges?

- Depends on how a ridge is defined (Saint-Venant, principal curvature extrema, …)
- Exact answer very messy

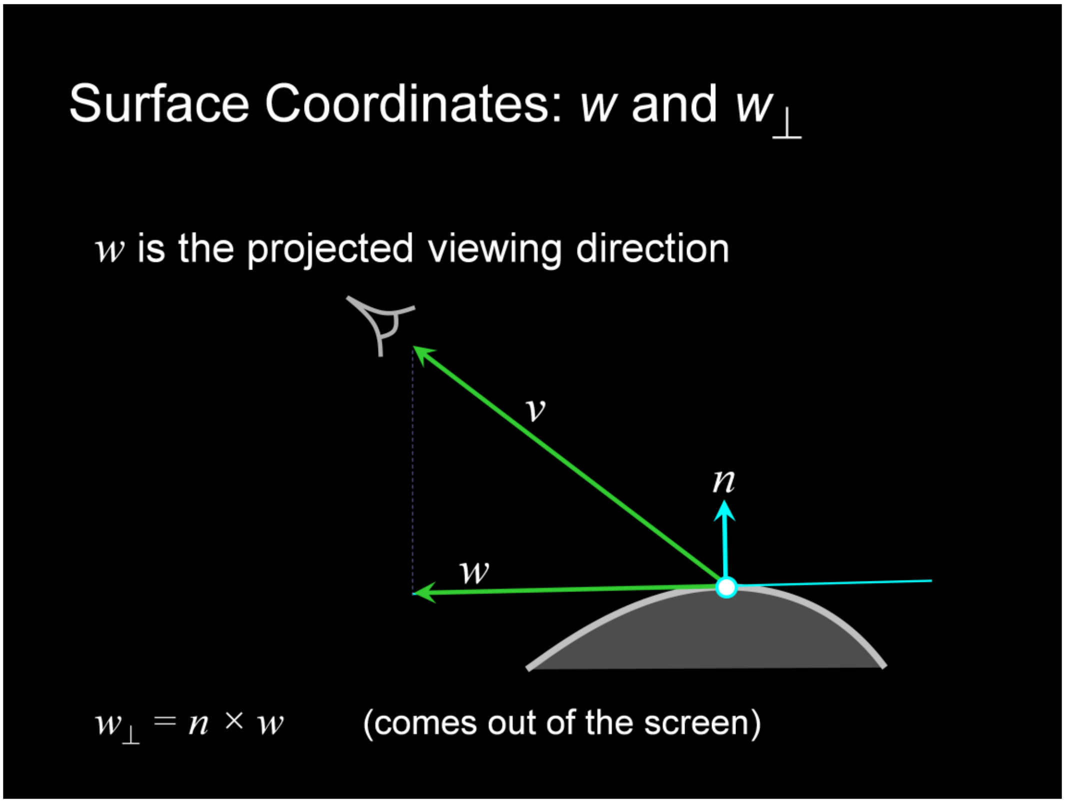
Unfortunately, even if we set up the problem in a restricted way (headlight, diffuse shading), the answer turns out to be very “messy” mathematically.

Intensity Ridges in Images

- Instead look for maxima of $n \cdot v$ along view-dependent directions w and $w_\perp$
  - Analog of intensity valleys in images as suggestive contours (minima of $n \cdot v$ along w)
- Appears to be a good approximation

Instead, we will look for simpler line definitions, which nevertheless qualitatively match intensity ridges well (just as suggestive contours qualitatively match intensity valleys). To do this, we will look at local maxima of n dot v, in the projected view direction w and its perpendicular. This corresponds to the definition of suggestive contours as local minima of n dot v in the direction w.

(Incidentally, we have also examined local *minima* of n · v in the direction perpendicular to w – they don't appear to be especially interesting…)



To review, w is just the projection of the view direction into the tangent plane of the surface. $w_\perp$ is also in the tangent plane, and perpendicular to w. In the above drawing, $w_\perp$ would be pointing towards you.

Highlight Lines

Suggestive highlights

– Maxima of $n \cdot v$ along w

Principal highlights

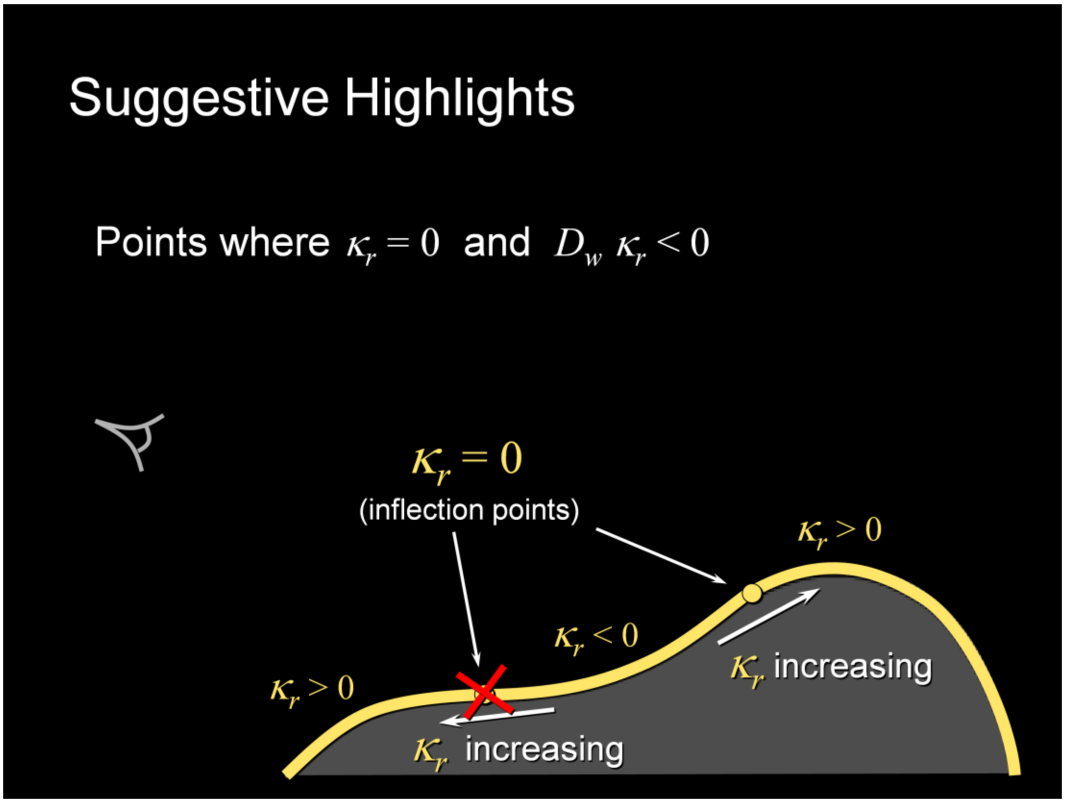
– Maxima of $n \cdot v$ along $w_\perp$

Lines are drawn in white

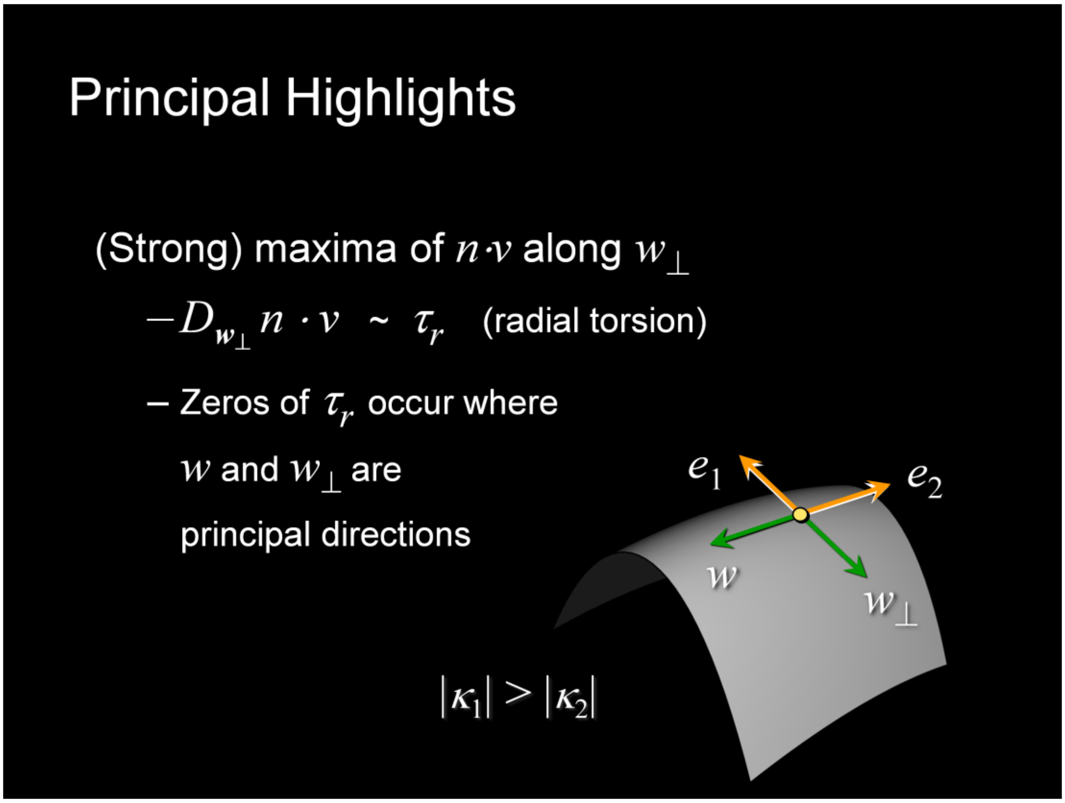
– In practice only draw strong maxima

Now we can define the two families of “highlight” lines corresponding to the above definitions. Suggestive highlights are local maxima of n dot v (corresponding to image intensity under a headlight) in the w direction, while principal highlights are the local maxima in the $w_\perp$ direction.

The styles that we will examine draw principal highlights in white (as opposed to the contours and suggestive contours, which are drawn in black). We find that this makes it easier for the visual system to interpret them.

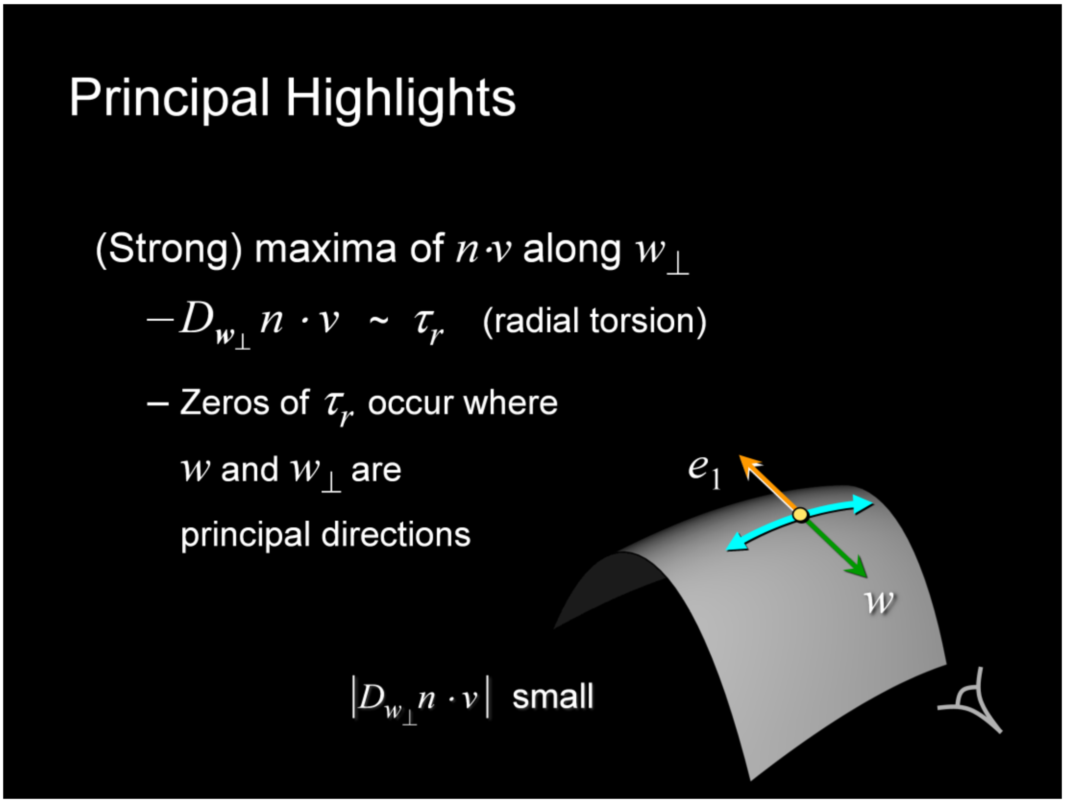


Let's look at suggestive highlights. These are just the inflections we threw away (with the derivative test) when finding suggestive contours.



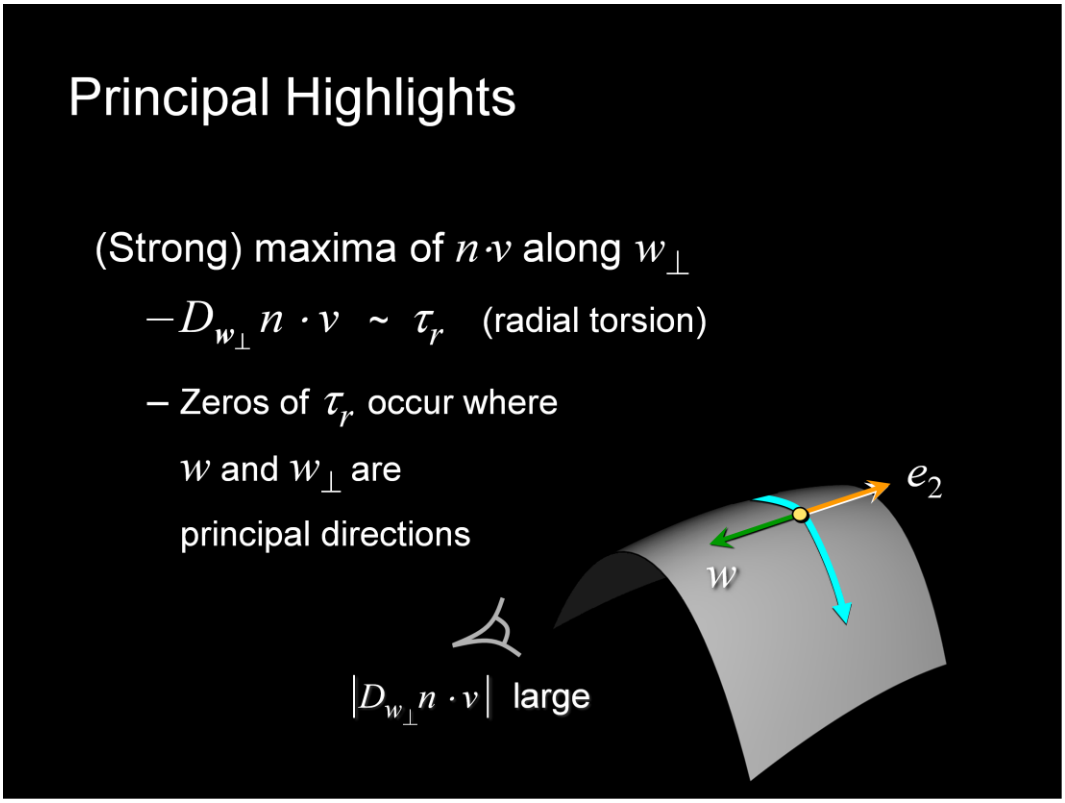
Principal highlights are another matter entirely. If we go through the math, we find that just as the derivatives of n · v in the w direction were related to the radial curvature, the derivatives in the $w_\perp$ direction are related to another quantity called "radial torsion". Intuitively, torsion represents the "twisting" of the normal direction as we move along the surface. Surfaces of zero radial torsion (corresponding to the maxima of n · v) are the ones that don't exhibit this twist, in the view direction.

This turns out to happen precisely when the view direction is aligned with one of the principal directions.

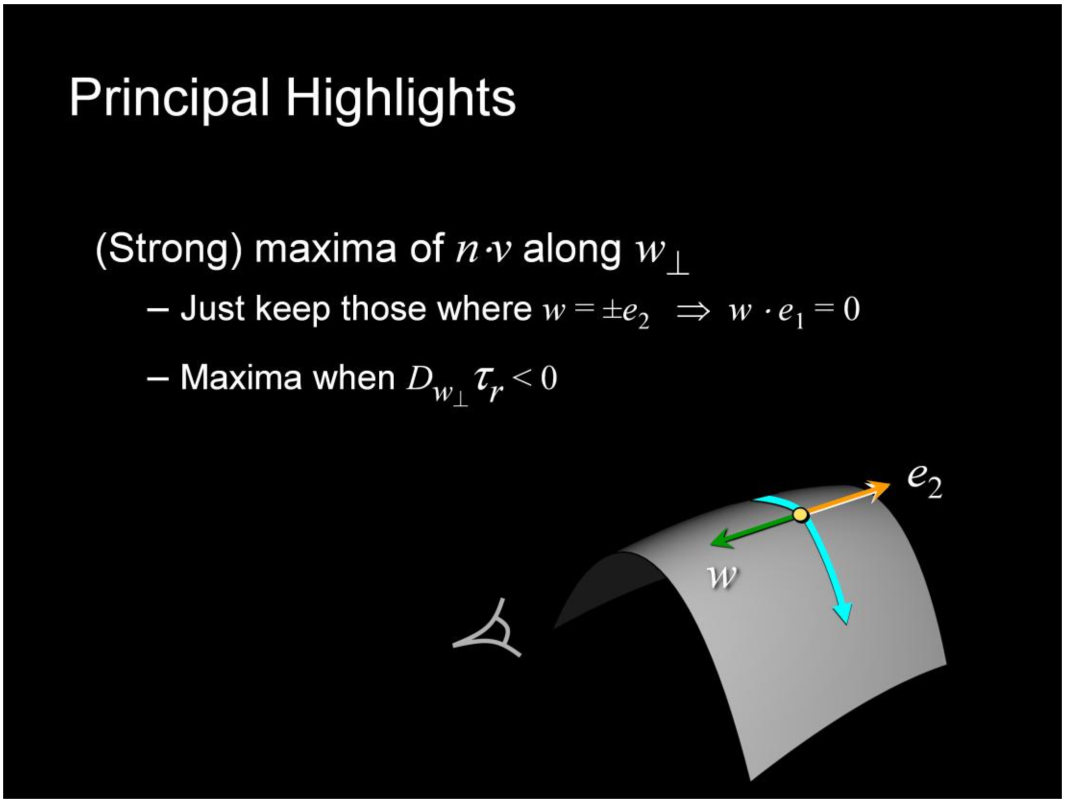


We now move to a test in the principal highlight definition designed to keep only the strong intensity maxima in the $w_\perp$ direction.

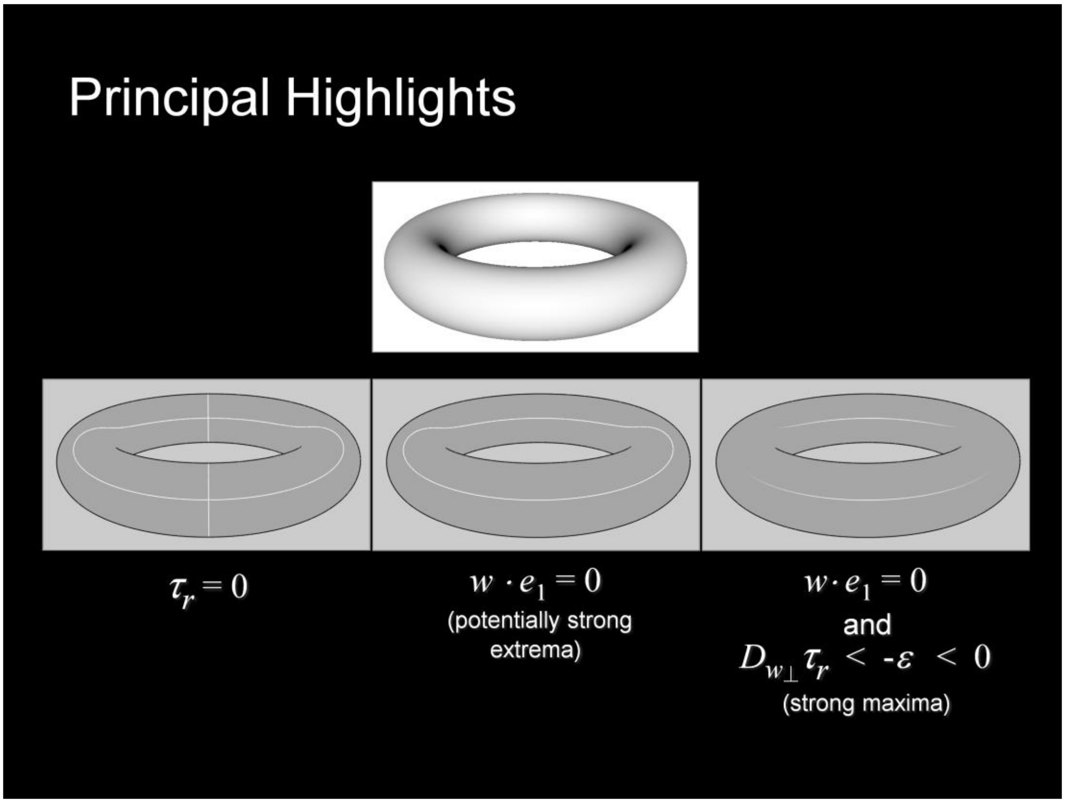
The intuition is that when the view direction is aligned with the higher-curvature principal direction e_1, the surface is not curving very quickly in the perpendicular direction.



The opposite case, when we are looking along the ridge (and w is aligned with e_2, the weaker principal direction), leads to strong intensity maxima in the perpendicular direction, because that's the direction in which the normal is changing quickly.

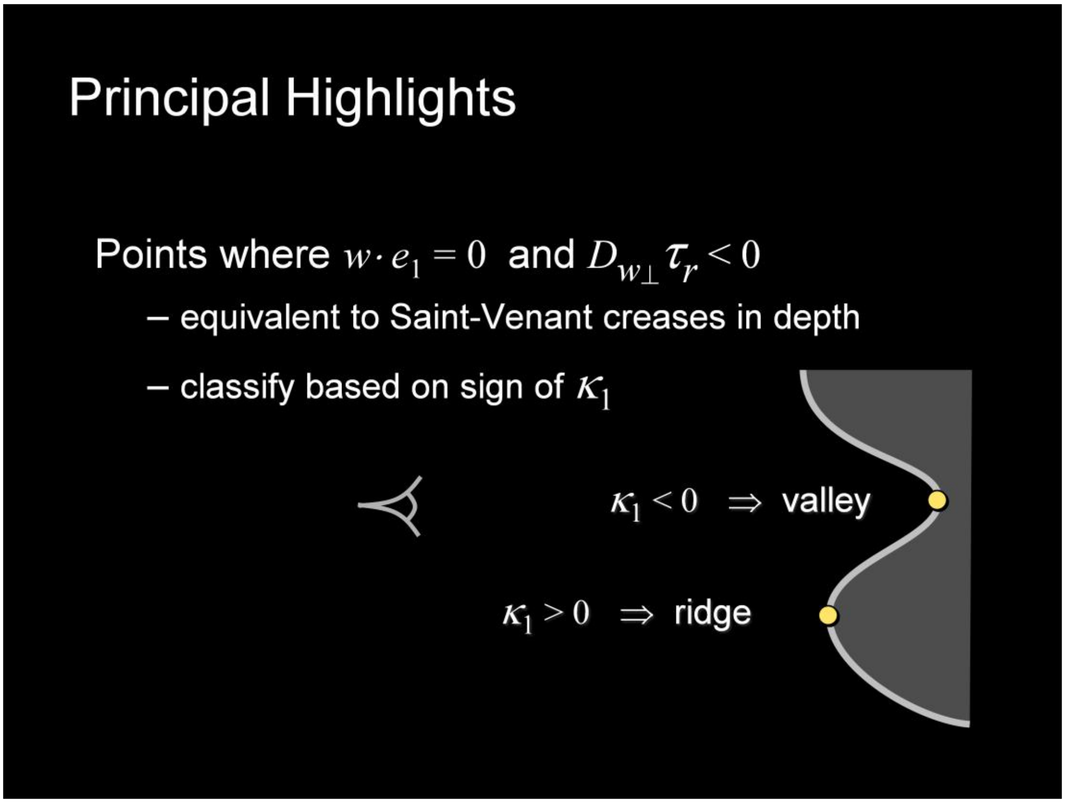


So, we keep the locations where w is along e_2, which means that it is perpendicular to e_1. So, instead of basing our definition for principal highlights on torsion, we adopt $w \cdot e_1 = 0$ as the primary definition of principal highlights. As with suggestive highlights, there is a derivative test necessary to keep only local maxima, not minima.



Here’s what the different definitions look like for a simple model.

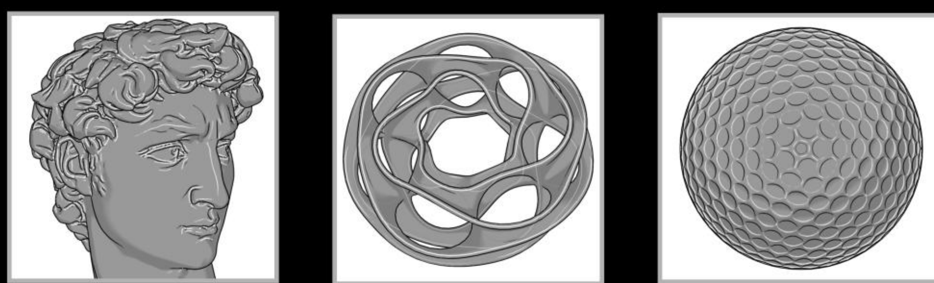
At left, we see all zeros of radial torsion. At center, we keep only those locations where w lines up with e_2, and at right we apply a derivative test (with a small but non-zero threshold).



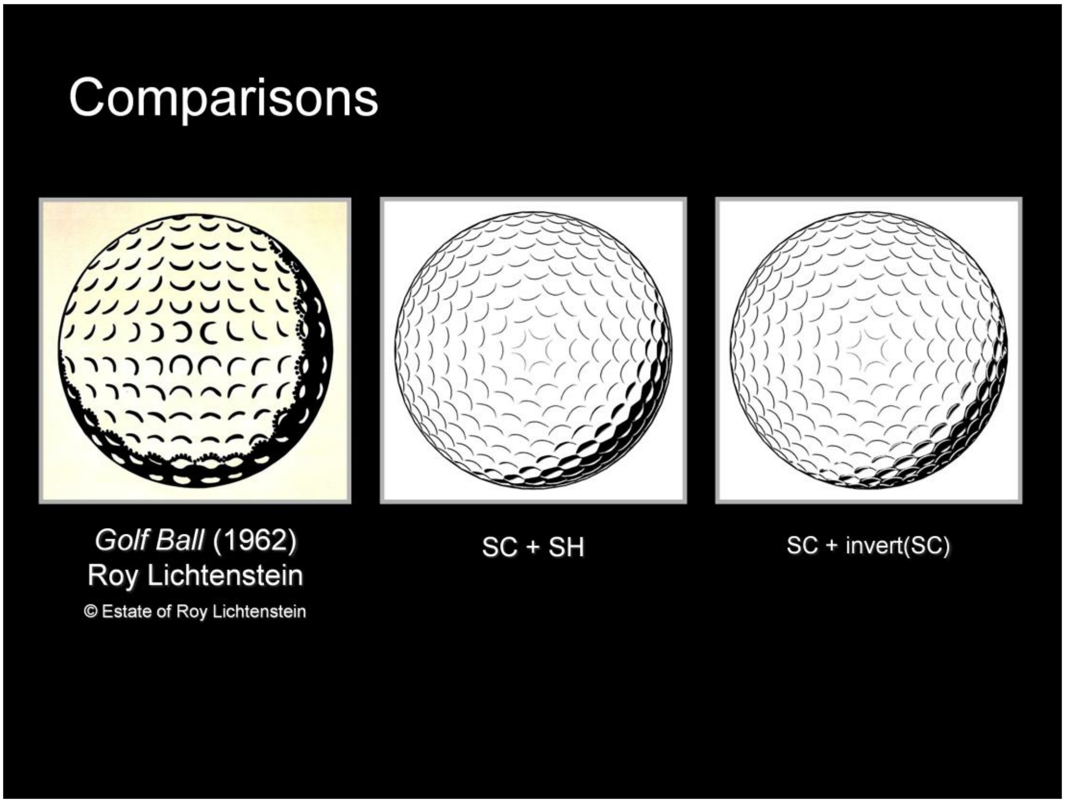
It is possible to show that principal highlights correspond exactly to converting the depth map to a range image, then looking for extrema (illumination ridge and valley lines) according to a particular definition due to Saint-Venant. In other words, they are one possible view-dependent analog to the view-independent crest lines. (Another possible view-dependent analog is apparent ridges, which we will see later.)

By looking at the sign of the first principal curvature, it is possible to classify these lines as ridge-like or valley-like, and use this additional information to stylize them differently or omit one or the other family.

Results



Here are some examples with both suggestive contours and suggestive and principal highlights. (Drawn together with a gray background and subtle toon shading.)



Here is a slightly different style, where we still draw the lines in black and white, but draw the shape with a black/white toon shader (so that only suggestive contours are visible in the white regions, and only suggestive highlights are visible in the dark regions). This corresponds to the style of the Frank Miller comic we saw earlier, as well as this painting of a golf ball by Roy Lichtenstein.

Looking at this painting, we note that the direction of the half-round strokes in the dark region corresponds well with our suggestive-highlight rendering. In contrast, simply inverting the suggestive contours in the black toon region (as shown at right) gives lines that face the wrong way, and don't match the Lichtenstein painting any more.

Apparent Ridges [Judd et al. 2007]

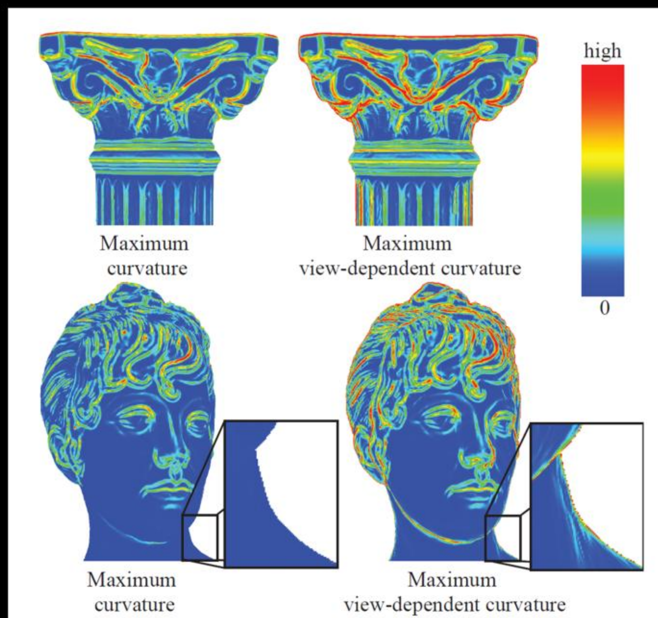
- View-dependent variant of ridge and valley lines
- Motivation: look for rapid *screen-space* normal variation
  - Different from ridge and valley lines because of (view-dependent) foreshortening
  - Includes occluding contours, and other lines that smoothly join to them
  - Approach ridges and valleys when looking head-on

Matisse

There is one final recently-introduced family of lines that we will look at, namely apparent ridges. These were motivated by some drawings, such as this one by Matisse, that seem to combine ridge-like features with contours.

The approach of apparent ridges is to apply the standard ridge and valley line definition (local maxima of principal curvature, in the corresponding principal direction), but replace the use of standard surface curvature with a view-dependent quantity that takes foreshortening into account. Specifically, the curvatures in the projected view direction are divided by $n \cdot v$, to account for the fact that normals vary more rapidly with respect to screen-space location where the surface is tilted away from the viewer.

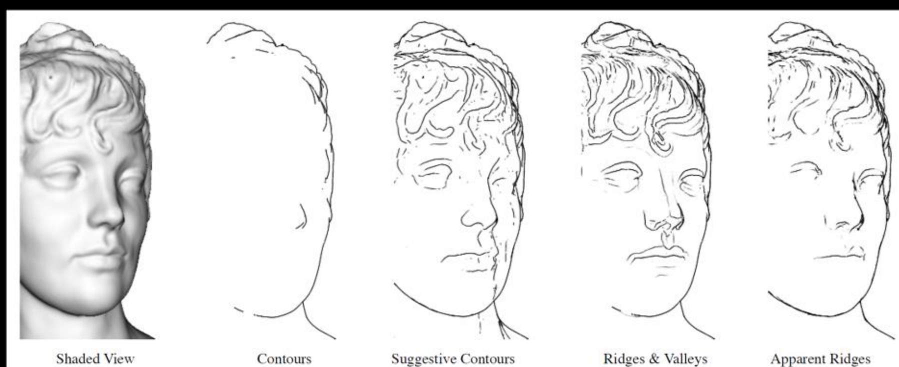
View-Dependent Curvature



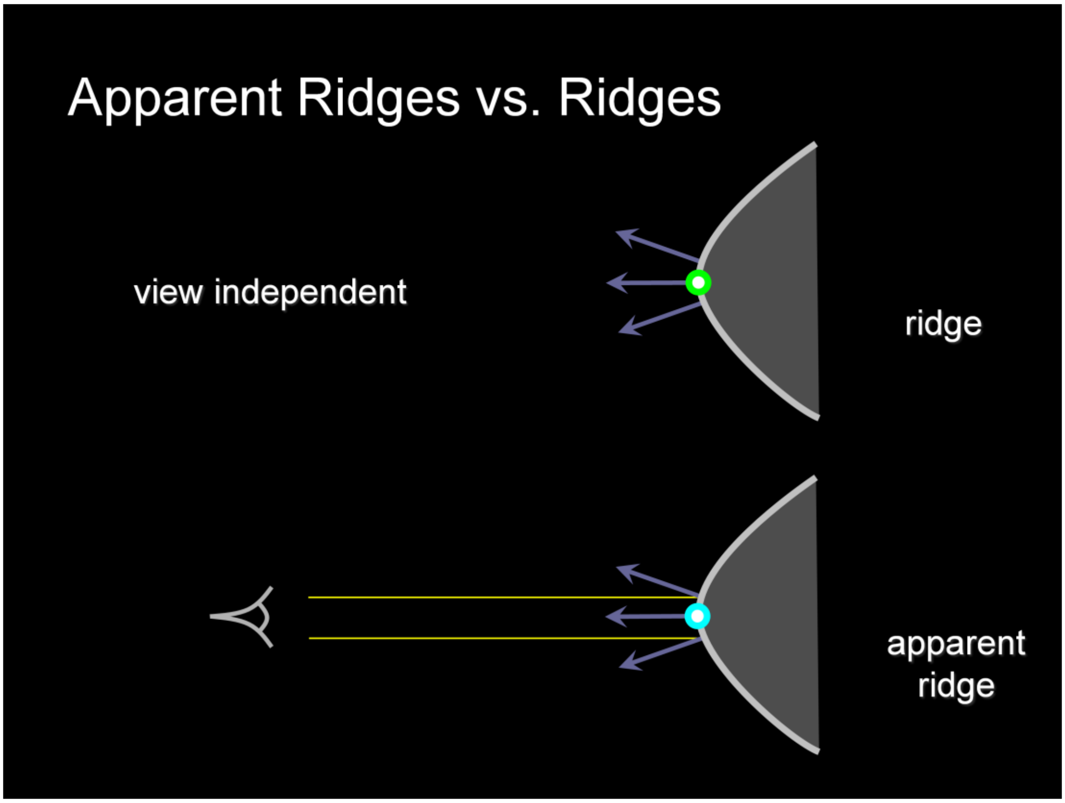
Here is a visualization of view-dependent curvature, as compared to standard curvature. It is obvious that view-dependent curvature grows quickly near occluding contours, leading to a strong tendency for the technique to place lines near those locations.

Apparent Ridges

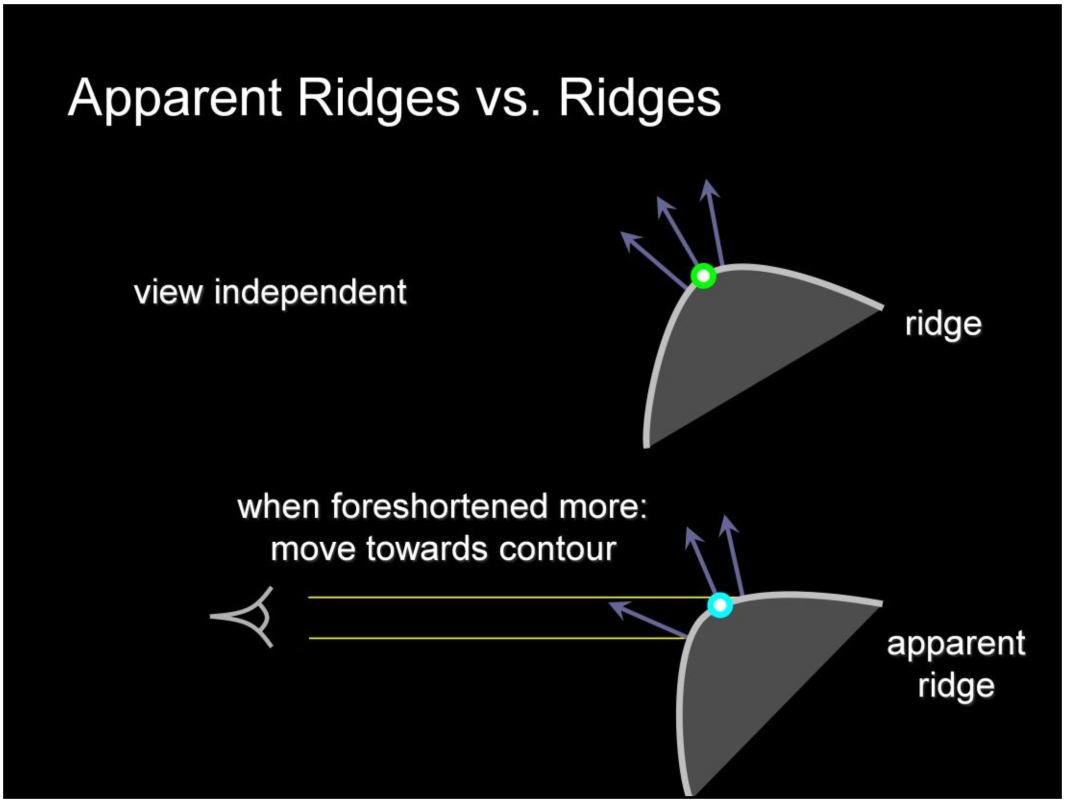
Local maxima of view-dependent curvature
(in view-dependent principal direction)



Here is a comparison of apparent ridges (right) to renderings with suggestive contours and view-independent ridge/valley lines.



Looking at the qualitative behavior of apparent ridges, we see that they match standard ridges and valleys when viewed head-on.



As the view becomes more oblique, they smoothly slide along the surface until they reach the contours. In fact, they connect up to the contours smoothly (just like suggestive contours).

Lines Summary

Derivative Order	Image-Space	View-Independent Object-Space	View-Dependent Object-Space
0^{th}	Isophotes	Topo-lines	Cutting planes
1^{st}		Isophotes	Occluding contours
2^{nd}	Edges, extremal lines	Parabolic lines	Suggestive contours, suggestive highlights, principal highlights
3^{rd}		Crest lines (ridges and valleys)	Apparent ridges

In summary, we have seen three classes of mathematical line definitions: image-space, view-independent object-space, and view-dependent object-space. Although the jury's still out, it appears that the latter category is the most interesting because it yields shape-conveying lines, while being amenable to sophisticated NPR stylization algorithms.

This table classifies the lines we've seen according to the order of derivatives used in their definition. (Incidentally, we don't *think* there are especially interesting line definitions in the empty boxes, but we certainly could be wrong…)