

# Computational Vision

## U. Minn. Psy 5036

### Daniel Kersten

#### Lecture 16: Shape from X

## Initialize

### ■ Spell check off

```
In[1]:= Off[General::spell1];
```

```
In[2]:= SetOptions[ArrayPlot, ColorFunction -> "GrayTones", DataReversed -> True,
  Frame -> False, AspectRatio -> Automatic, Mesh -> False,
  PixelConstrained -> True, ImageSize -> Small];
SetOptions[ListPlot, ImageSize -> Small];
SetOptions[Plot, ImageSize -> Small];
SetOptions[DensityPlot, ImageSize -> Small, ColorFunction -> GrayLevel];
nbinfo = NotebookInformation[EvaluationNotebook[]];
dir =
  ("FileName" /. nbinfo /. FrontEnd`FileName[d_List, nam_, ___] :->
    ToFileName[d]);
```

---

## Outline

### Last time

- Geometry, shape and depth: Representation issues & generative models
- Lambertian model & surface normal representations

Surface normal representation: Local, dense, metric, viewer-dependent coordinate system (e.g. slant from observer).

## Today

- Inference of shape from shading--set in context of other cues to shape,
- Perception of shape from shading
- Formal ambiguities in the generative model, the bas-relief transform
- Task analysis: "Shape for X" problem

---

## Perception of "Shape from X"

### What is shape?

One mathematical definition: "Geometrical properties that are invariant over translation and scale".

A computational vision definition: "Whatever is left over after discounting material, illumination and viewpoint variations in the images of an object"

We will take a more general view here:

"geometrical relationships within a surface that are useful for visual functions". More precise definitions depend on the function.

Later we will return to the "discounting" issues.

There are a variety of cues to shape. In natural images, these cues typically co-vary. Human vision can infer shape from any of them or in combination, hence "shape from X".

Later on we will talk about cue integration.

In a psychophysics lab, they can be manipulated independently. Let's look at several categories of shape cues.

### Texture

Later on we will talk in more detail about how to model and infer surface material properties. Texture is one property of surfaces that is particularly informative for shape and object identity. Texture by definition has some degree of regularity. Texture can be highly regular, or only regular in the statistical sense. For the moment, suppose a surface has a deterministic regular homogeneous pattern of "texture elements", where each texture element is the same 3D size, and they are distributed homogeneously.

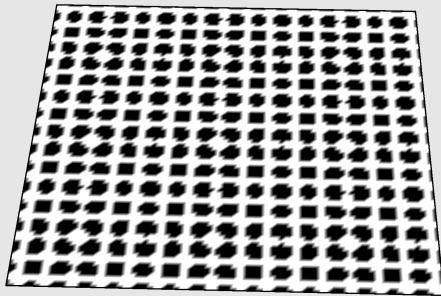
## ■ Slant & texture

Define a function to place circular disks on a regular grid

```
In[11]:= checker[x_,y_,space_,radius_]:= If[(Mod[x,space]^2+Mod[y,space]^2<radius^2)
pic = Table[{checker[x, y, 32, 12]}], {x, 1, 512}, {y, 1, 512}];
```

```
ListPlot3D[Table[1, {x, -5, 5, .1}, {y, -5, 5, .1}], Mesh -> None,
VertexColors -> {pic[[5 ;; -5 ;; 5, 5 ;; -5 ;; 5]]}, Lighting -> "Neutral",
Axes -> False, Boxed -> False, ImageSize -> Small]
```

Out[24]=



```
In[19]:= pic = Reverse[ExampleData[{"TestImage", "Mandrill"}, "Data"] / 255.];
```

### Try changing the orientation of the above texture

The generative assumptions above lead to regularities in the image that can be used to estimate surface slant--if the spatial scale of the texture is small in comparison to the surface curvature, so that the surface can be approximated locally by a plane.

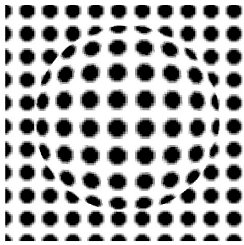
Then, cues to surface slant:

- 1) spatial gradient of the density of texture elements: the image density increases with distance;
- 2) an individual element (such as a circular disk) gets smaller with distance, and its image height to width ratio gets smaller ("compressed") with increasing slant;
- 3) the ratio of the back width to the front width of an individual element gets smaller with slant (imagine a small square, linear perspective on small scale) . (See Knill).

Tilt is the direction that the surface slants away most rapidly. Here the density and sizes of the elements change the most rapidly.

**Shape and texture: Adapt the above function to map a texture onto a smooth shape**

---



**Project idea: Map a small single textural element onto an "invisible" bump. Allow the user to move the element around. Can you infer the underlying shape?**

---

## Stereo disparity

### ■ Random dot stereograms

A simple planar surface in depth.

<http://demonstrations.wolfram.com/FloatingDiskStereogram/>

Cross your eyes so that you see three dots (rather than two) at the bottom. The above surface should appear like the one below, but viewed from above.

But smooth shape from stereo can also be conveyed through disparity.

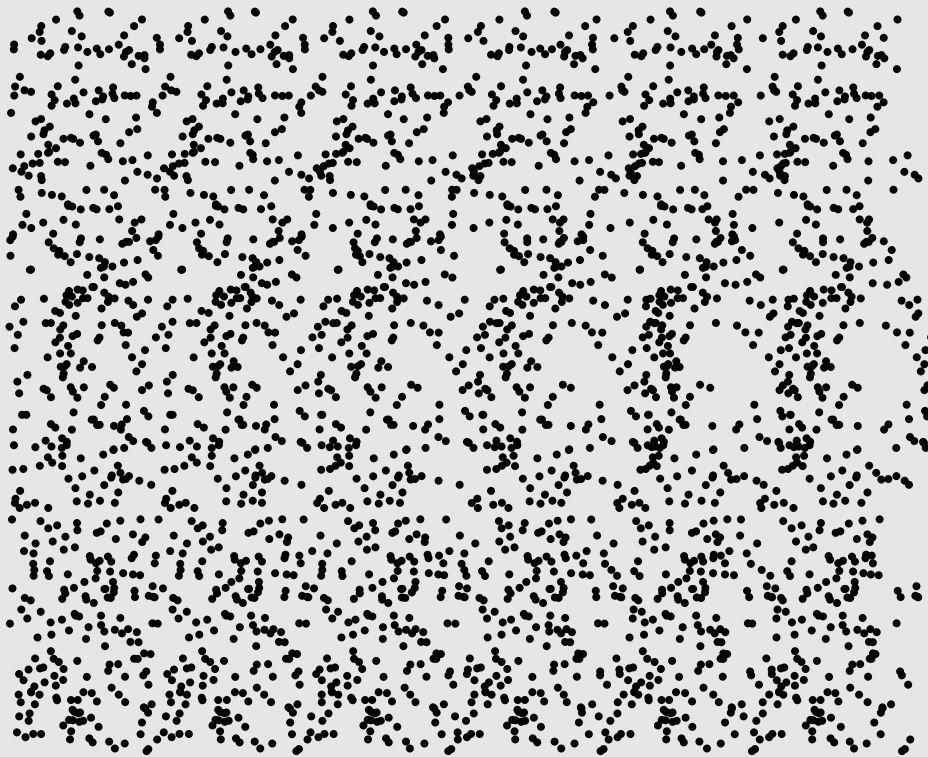
## ■ Random dot stereograms, "Magic eye" style

```
In[110]:= (*Copyright (c) Dror Bar-Natan 1994*)
(*Random Dot Stereograms in The Mathematica Journal 1-
  3 (1991) 69-75).*)

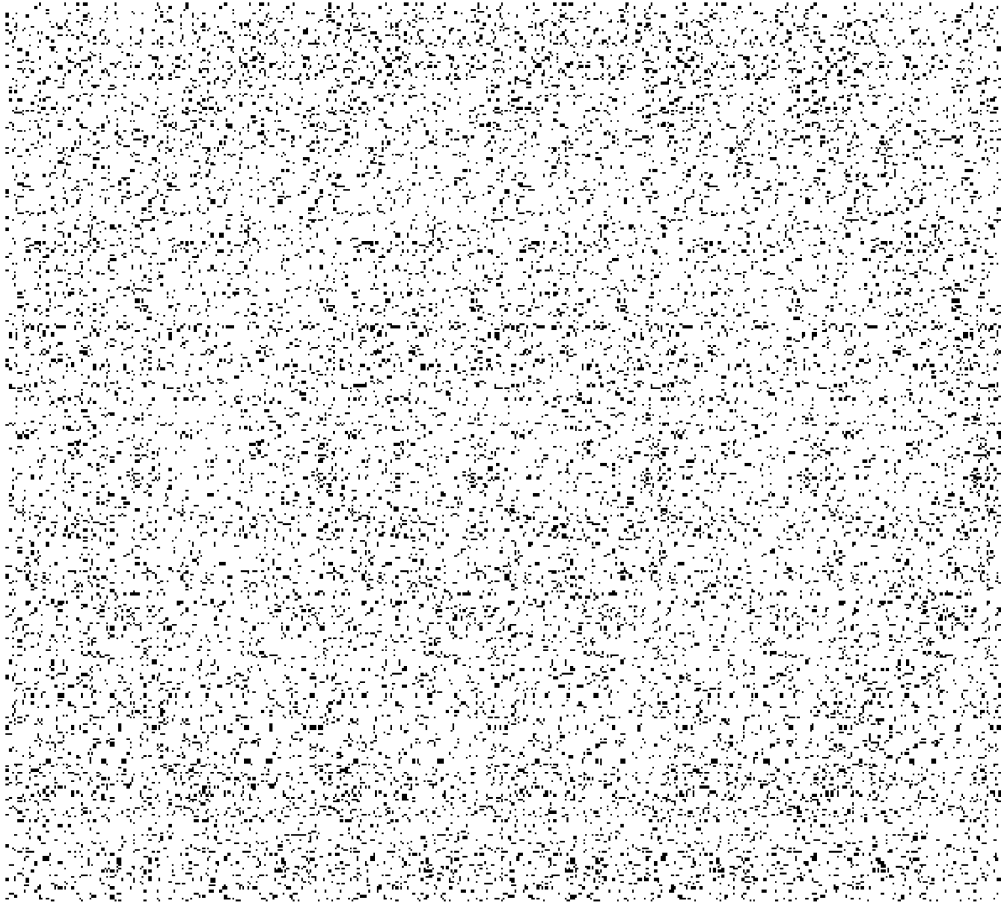
f[x_, y_] := Sin[20 (x^2 + y^2)] / (1 + 5 (x^2 + y^2))
(*the function drawn*)

Show[
  Graphics[Table[(*loop 500 times*)
    x = Random[Real, {-1, -0.6}];
    (*generate a random number between -1 and -0.6*)
    y = Random[Real, {-1, 1}];
    (*generate a random number between -1 and 1*)
    Table[(*loop 6 times*) x0 = x; x = x + 0.4 + f[x, y] / 15;
      (*compute coordinates of next pixel*) Point[{x0, y}],
      (*put a point*) {6}], (*end inner loop*)
    {500}]]]
(*end outer loop*)
```

Out[111]=



Here's one at a higher dot density:



See: Christopher Tyler. The wiki page provides nice summary: <http://en.wikipedia.org/wiki/Autostereogram>.

### ■ Stereo + shading + surface contours

```
In[61]:= pL=Plot3D[Cos[Sqrt[x^2+y^2]], {x,-10,10}, {y, -10, 10},  
            AspectRatio -> Automatic, PlotPoints-> {25,25}, ViewPoint->{1.3,-2.4,2.}
```

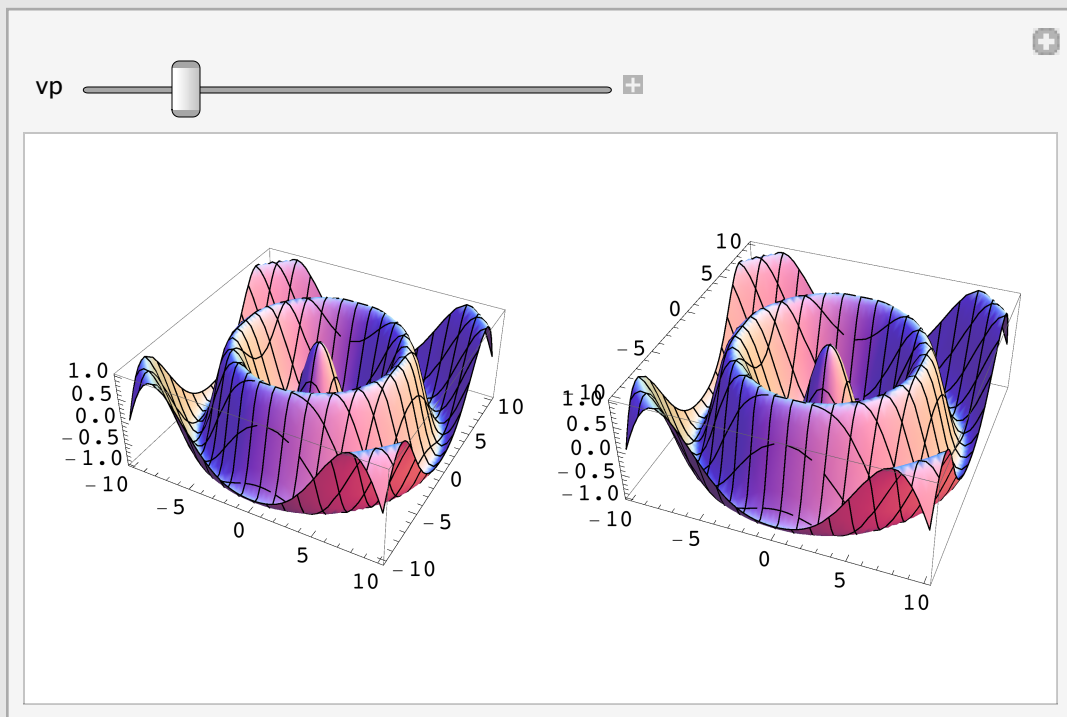
```

In[62]:= Manipulate[
  pR = Plot3D[Cos[Sqrt[x^2 + y^2]], {x, -10, 10}, {y, -10, 10},
    AspectRatio -> Automatic, PlotPoints -> {25, 25},
    ViewPoint -> {vp, -2.4, 2.}];

  Show[GraphicsArray[{pL, pR}]], {{vp, 1.1}, 0.8, 1.5}]

```

Out[62]=



## Motion

We'll look at structure & shape from motion in detail later. With some care, one can put up an animation in which any single frame shows no apparent structure, but when moving, the shape becomes clear. E.g. rotating "glass" cylinder with dots on it. Or, "biological motion".

random.mov

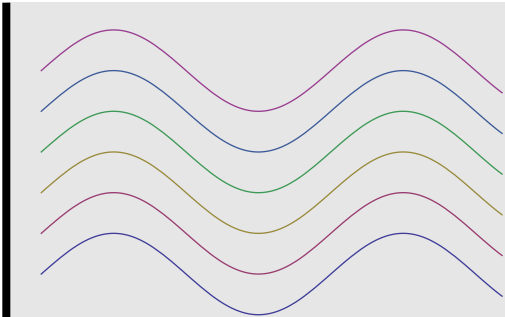
cube.mov

## Contours

### ■ Surface contour (markings)

In[63]:= `Plot[{Sin[x], Sin[x]+1, Sin[x]+2, Sin[x]+3, Sin[x]+4, Sin[x]+5}, {x, 0, 10}, Axes`

Out[63]=



### ■ Shape from moving contours

Here's an example taken from the *Mathematica* documentation. Apparent 3D shape emerges only with movement.

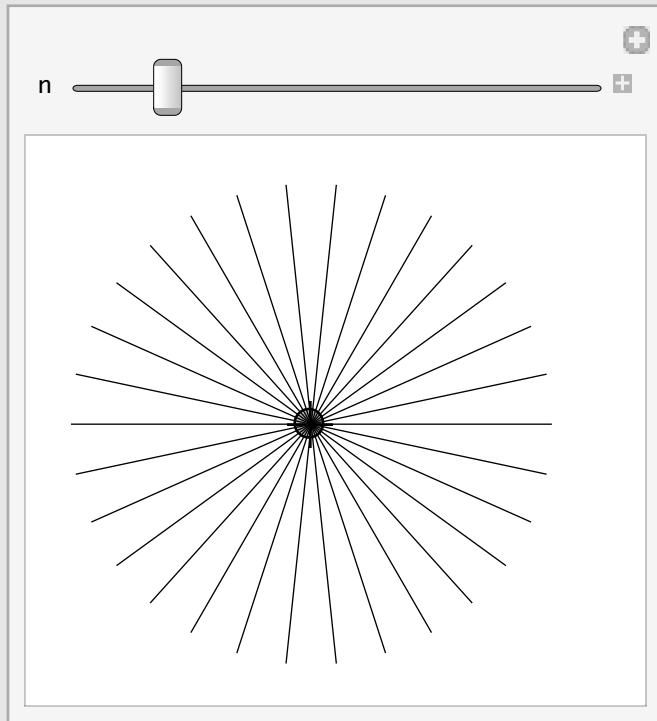
```
In[112]:= 
nsize = 128;
subnsize = 32;
noise = Table[RandomReal[], {nsize}, {nsize}];
subnoise = Table[RandomReal[], {subnsize}, {subnsize}];
gn = ArrayPlot[noise, PlotRange -> {0, 1}, PixelConstrained -> {1, 1}];
gsn = ArrayPlot[subnoise, PlotRange -> {0, 1}, PixelConstrained -> {1, 1}];
```



**Manipulate[**

```
Graphics[{{Line[Table[{{Cos[t], Sin[t]}, pt], {t, 2. Pi / n, 2. Pi, 2. Pi / n}}]},  
PlotRange -> 1, ImageSize -> Small], {{n, 30}, 1, 200, 1},  
{{pt, {0, 0}}, Locator}]
```

Out[119]=



### ■ Surface contours (creases or orientation discontinuities)

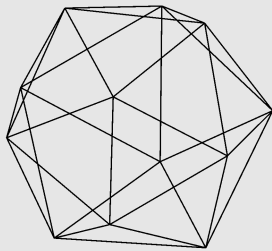
The lines mark surface orientation discontinuities.

In[64]:=

```
<< PolyhedronOperations`
```

```
In[65]:= Graphics3D[{Yellow, Opacity[.0],
  PolyhedronData["GreatDodecahedron", "Faces"]}, Boxed → False,
  ImageSize → Small]
```

Out[65]=

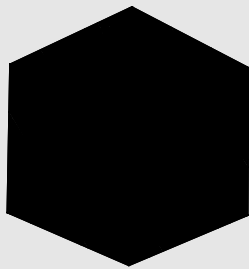


#### ■ Bounding contours, (depth discontinuities), e.g. silhouette

Orientation discontinuities can coincide with depth discontinuities.

```
In[66]:= Graphics3D[
  {Black, Opacity[1], PolyhedronData["GreatDodecahedron", "Faces"]},
  Boxed → False, Lighting → {"Ambient", Orange}, ImageSize → Small]
```

Out[66]=

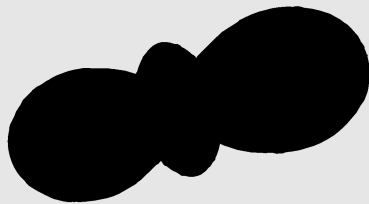


#### ■ ...and smooth occluding contours

In this smooth shape, orientation of a unit normal vector changes smoothly at the depth discontinuities of the boundary.

```
In[67]:= SphericalPlot3D[1 + 2 Cos[2  $\theta$ ], { $\theta$ , 0, Pi}, { $\phi$ , 0, 2 Pi}, Lighting  $\rightarrow$  None,  
Boxed  $\rightarrow$  False, Axes  $\rightarrow$  False, AspectRatio  $\rightarrow$  1, ImageSize  $\rightarrow$  Small]
```

Out[67]=



Shading...

---

## Computing shape from shading

### Introduction

The ambiguity of shading has had a newsworthy impact for well over a century, beginning with the "canals of Mars". From 1877, these "canals" attracted world-wide attention when the Italian astronomer, Giovanni Virginio Schiaparelli reported about a hundred of them. The American astronomer Percival Lowell thought the markings were vegetation, several kilometres wide, bordering irrigation canals dug by intelligent life to carry water from the poles. However, most astronomers couldn't see the canals. Photography through the Earth's atmosphere offered no solution because the lines were near the limit of resolution of the human eye and the physical optics. The controversy was settled in 1969 when photographs were taken from several hundred kilometres above the surface of Mars by the Mariner 6 and 7 spacecraft. These showed many craters and other formations but no canals.

However, the phenomenon of seeing intriguing shapes from shading hadn't gone away even in the late 20th century. With the Viking orbiter of 1976, NASA photographs have more interesting pictures from Mars:



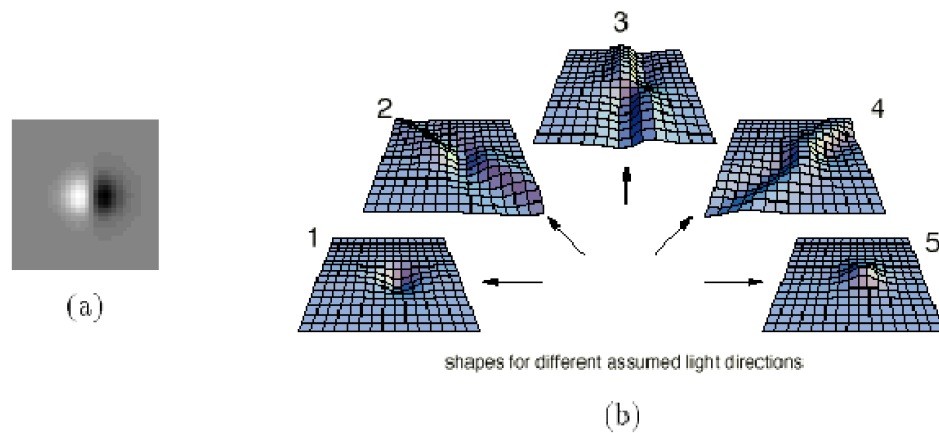
Are these shapes we see really there? If not, why do we see the particular shapes that we do?

## The ambiguities

Shape from shading is fascinating from a computational perspective--there is considerable local ambiguity, yet human vision is so sure of what it sees. To appreciate vision's solution, let's take a closer look at the ambiguities.

### ■ Overview

The following figure from William Freeman at MIT illustrates the light-direction/shape ambiguity problem of shape from shading:



**Figure 6:** (a) Perceptually, this image has two possible interpretations. It could be a bump, lit from the left, or a dimple, lit from the right. (b) Mathematically, there are many possible interpretations. For a sufficiently shallow incident light angle, if we assume different light directions, we find different shapes, each of which could account for the observed image.

(Figure from: <http://www.merl.com/reports/docs/TR93-11a.pdf>. See too: W. T. Freeman, Exploiting the generic viewpoint assumption, *International Journal Computer Vision*, 20 (3), 243-261, 1996.)

One solution to resolve the ambiguity is to use a surface smoothness constraint (e.g. as a Bayesian prior). Another is to approximate the shading as linear. A third is to learn a shape estimator. These methods include adopting or learning prior constraints, "active vision" in which looks for new information to resolve ambiguity, and the combination of information from other modules (e.g. illumination direction estimator).

Marginalization over illumination direction also provides a means to reduce shading ambiguity (Freeman, 1994). This is

equivalent to the principle that:

*Vision should pick the shape interpretation that is most robust to variations in illumination direction.*

The principle is analogous to the general viewpoint constraint which we will encounter later, in which viewpoint is substituted for illumination direction. This principle

doesn't specify a method, but suggests that a method to estimate shape given unknown illumination should have this property.

### Classic regularization approach to shape from shading (Ikeuchi and Horn, 1981)

See chapter 4 in Horn's book "Robot Vision": [http://people.csail.mit.edu/bkph/articles/Shape\\_from\\_Shading.pdf](http://people.csail.mit.edu/bkph/articles/Shape_from_Shading.pdf)

The theoretical approach to inverse problems can be formalized in terms of Bayesian estimation. Recall that scene-from-image problems, we start off with the generative model, and then seek an inverse solution.

Let's look at a classical solution to shape from shading due to Ikeuchi and Horn. Although it was not originally formulated as a Bayesian inference problem, it is essentially equivalent to MAP estimation.

Write down the generative model for image luminance  $L$  in terms of geometrical gradient space parameters  $p, q$ :

$$L_R = re \hat{N} \cdot \hat{E}$$

where  $r$  is the reflectance, and  $e$  the strength of illumination.  $N$  and  $E$  are as in the last lecture.  $E$  is a point light source (we'll assume  $e=1$ ), and  $E$ 's direction can be expressed in gradient space:

$$\hat{E} = \frac{(p_e, q_e, -1)}{\sqrt{p_e^2 + q_e^2 + 1}}$$

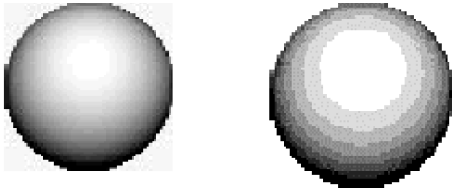
$$\hat{N} = \frac{(p, q, -1)}{\sqrt{p^2 + q^2 + 1}}$$

Taking the dot product, we have:

$$L_R(p, q) = \frac{r(pp_e + qq_e + 1)}{\sqrt{(p^2 + q^2 + 1)(p_e^2 + q_e^2 + 1)}}$$

Our data is  $L(x,y)$ . We require that  $L(x,y) = L_R(p,q)$ . One problem is that we do not know the reflectance  $r$  or the light source direction  $(p_e, q_e)$ . (Note that in general,  $r$  is not going to be a constant, but rather some pattern of pigmentation,  $r(x,y)$ ). Suppose we assume  $r$  and  $(p_e, q_e)$  are known and fixed constants, we still have many  $p$ 's and  $q$ 's which will satisfy  $L - L_R=0$ , in fact two unknowns for every equation. This makes clear that shape from shading is under-constrained or "ill-posed" mathematical problem.

Isophotes (lines of constant luminance in the image) do not necessarily imply constant (p,q):



A solution to ambiguity (that we'll explore in detail later in optic flow motion field estimation) is to impose a smoothness constraint (a local constraint) and boundary conditions (global constraint). Smoothness constraints can be formulated as "Bayesian priors", but more on this later. Let's see how Ikeuchi and Horn got around the problem of ill-posedness.

### Implementing a smoothness constraint.

The intuition is that many objects tend to be smooth. What does this mean in terms of how surface normals change across a surface? We would expect smooth surfaces to have small spatial derivatives. If we made a guess as to what the surface normals were, we could measure the degree of smoothness using first or second derivative magnitudes, and if too big, try to change the shape to make the surface smoother. The idea is to write an algorithm that searches for a surface that satisfies  $L - LR=0$  AND has low spatial derivatives.

Let us follow the approach of Ikeuchi and Horn (1981), and enforce a surface smoothness assumption (or constraint) by requiring that the squared values of the Laplacians of p and q be small. The Laplacians are:

$$\nabla^2 p = \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2}$$

$$\nabla^2 q = \frac{\partial^2 q}{\partial x^2} + \frac{\partial^2 q}{\partial y^2}$$

We then construct a local cost function function:

$$e(x, y) = (L(x, y) - L_R(p, q))^2 + \lambda((\nabla^2 p)^2 + (\nabla^2 q)^2)$$

We want  $e(x,y)$  to be small. This will be the case if (p,q) is chosen so that the first term tends toward zero, and if the spatial derivatives are small (i.e. because we've chosen p,q to change slowly). But we want this local cost function to be small over the whole image.

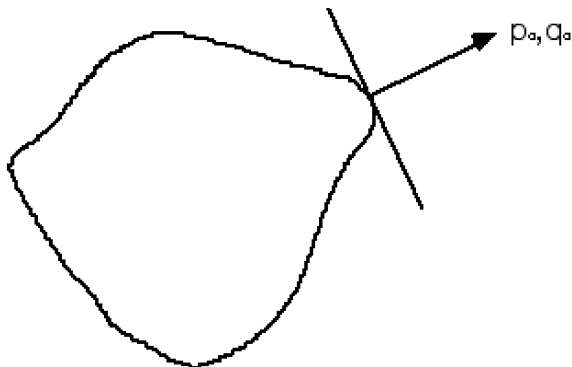
So we construct a global cost "functional" by integrating over the whole image:

$$e[p(x, y), q(x, y)] = \iint (L(x, y) - L_R(p, q))^2 + \lambda((\nabla^2 p)^2 + (\nabla^2 q)^2) dx dy$$

The goal is to find (p,q) at each point (x,y) which makes the global error or cost,  $e[p,q]$  as small as small as possible.  $\lambda$  is a weighting function on the smoothness. For example, it should be big if we have reason to believe our data are noisy--i.e.

we want to trust our prior smoothness constraint over the unreliable data.

The boundary values can also be used to constrain the solution, and may be necessary at times to produce a unique solution. They are also important to prevent smoothing over discontinuities. What are the boundary conditions?



The surface normal can be read directly off the image if the boundary contours are known. But we have a problem, at smooth occluding bounding contours,  $p_o$  and  $q_o$  (the spatial rates of change of the surface depth away from the view-point) are infinite. And if the bounding contours are not smooth, but sharp, the surface normal is not defined there.

One solution is to change coordinates to ones that don't blow up at boundaries. Stereographic coordinates are one solution to avoiding infinities at the boundaries (Ikeuchi and Horn, 1982).

$$f = \frac{2p \sqrt{1 + p^2 + q^2} - 1}{p^2 + q^2}$$

$$g = \frac{2q \sqrt{1 + p^2 + q^2} - 1}{p^2 + q^2}$$

Another representation that will avoid the boundary problem is slant and tilt introduced earlier (e.g, Mamassian and Kersten (1995)).

It can be shown that minimizing the above error function is equivalent to maximizing the posterior probability of the map of surface  $p(x)$ 's and  $q(x)$ 's conditional on  $L(x,y)$ . Using Bayes, the generative model determines the likelihood, and the laplacians of the gradient parameters determine the smoothness prior.

## Side notes

### ■ Linear approximation to the Lambertian generative model for shading

The image formation constraint that we have used is non-linear. It is worth asking to what degree a linear approximation would be adequate-- a question which has been addressed by Knill and Kersten (1990) and by Pentland (1990). If a linear approximation works, then the inverse problem is linear.

Let's derive a linear approximation to:

$$L_R(p, q) = \frac{r(pp_e + qq_e + 1)}{\sqrt{(p^2 + q^2 + 1)(p_e^2 + q_e^2 + 1)}}$$

```
In[68]:= Lmodel[pn_,qn_,pe_,qe_] := ({pn,qn,-1}/Sqrt[pn^2+qn^2 + 1]).({pe,qe,-1}/Sqrt
```

where we use the notation  $p \rightarrow pn, q \rightarrow qn$ .

### Taylor's series

We can expand the image luminance in a Taylor series about  $\{pn, qn\} = \{0, 0\}$ . Recall that this is done by calculating successive derivatives at  $\{pn, qn\} = \{0, 0\}$ , and then substituting  $\{pn, qn\} \rightarrow \{0, 0\}$ . The first derivative with respect to  $pn$  is:

```
In[69]:= D[Lmodel[pn, qn, pe, qe], pn] /. {pn -> 0, qn -> 0}
```

```
Out[69]= 
$$\frac{pe}{\sqrt{1 + pe^2 + qe^2}}$$

```

Note that this is the cosine of the slant of the light source (see previous lecture).

*Mathematica's Series[]* function puts the Taylor series together for us. Here is the expansion up to linear terms:

```
In[70]:= Series[Lmodel[pn, qn, pe, qe], {pn, 0, 1}, {qn, 0, 1}]
```

```
Out[70]= 
$$\left( \frac{1}{\sqrt{1 + pe^2 + qe^2}} + \frac{qe qn}{\sqrt{1 + pe^2 + qe^2}} + O[qn]^2 \right) + \left( \frac{pe}{\sqrt{1 + pe^2 + qe^2}} + O[qn]^2 \right) pn + O[pn]^2$$

```

You can improve the approximation by including quadratic terms: `Series[Lmodel[pn, qn, pe, qe], {pn, 0, 2}, {qn, 0, 2}]`.

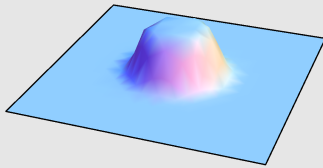
Try it out on `bump[x_, y_] := (1/4) (1 - 1/(1 + Exp[-10 (Sqrt[x^2 + y^2] - 1)]))`;

```
In[74]:= Clear[bump, x, y];
bump[x_, y_] := (1/4) (1 - 1/(1 + Exp[-10 (Sqrt[x^2 + y^2] - 1)]));
pn[x_, y_] := Evaluate[D[bump[x, y], y]];
qn[x_, y_] := Evaluate[D[bump[x, y], x]];
```



```
In[78]:= Plot3D[bump[x, y], {x, -3, 3}, {y, -3, 3}, Mesh -> False,
  PlotRange -> {0, .5}, ImageSize -> Small, Axes -> False, Boxed -> False]
```

Out[78]=



Light source:

```
In[79]:= {pe, qe} = {1, 1};
```

$$\text{Lapprox}[x_, y_] := \frac{1}{\sqrt{1 + pe^2 + qe^2}} + \frac{qe * qn[x, y]}{\sqrt{1 + pe^2 + qe^2}} + \frac{pe * pn[x, y]}{\sqrt{1 + pe^2 + qe^2}};$$

```
In[80]:= Lapprxom[x_, y_] := (1 + qe * qn[x, y] + pe * pn[x, y]) / Sqrt[1 + pe^2 + qe^2];
```

```
In[81]:= DensityPlot[Lappprox[x, y], {x, -3, 3}, {y, -3, 3}, Mesh → False,
  Frame → False, PlotPoints → 64, PlotRange → {0, 1},
  ColorFunction → "GrayTones", ImageSize → Small]
```

Out[81]=



**How could you use the above linear generative model for "shading from shape" to devise a solution to the inverse problem of "shape from shading"?**

In the next lecture, we'll look at a simple algorithm for shape from shading that assumes the linear generative model.

■ **Is shape from shading really ill-posed? Formal (exact) solution to the classic problem: lambertian, point light source**

Given a Lambertian reflectance, and a point light from the camera, the shape from shading problem is well-posed and a solution can be found (Dupuis and Oliensis, 1994). Oliensis showed that 1) this problem is generally **well-posed**, contrary to previous belief--and that when the light comes from the camera direction a shaded image determines the depicted surface **uniquely**; 2) that the image of a surface's **occluding boundary** (the edge where it rounds away from the viewer) is **not useful** for determining the surface--again contrary to previous belief; 3) that for an arbitrary shaded image there is generally no physical (differentiable) surface corresponding to it ( **nonexistence** ).

These are theoretical results--it is quite another matter to discover the built-in knowledge that the human visual system has regarding shape from shading, for example the boundary information is quite important (see below).

■ **Shape from shading on a "cloudy day"--diffuse lighting**

Most theoretical work on shape from shading started assuming point light sources, but diffuse lighting is the more realistic case. The generative physical problem becomes complicated. Ray-tracing and radiosity methods provide forward generative models (eg. Greenberg, 1989; Foley et al., 1990). But the straightforward inverse problem is impractical--can't check out all the ray-tracing bounces!

For an elegant solution to the problem of shape-from-shading on a cloudy day, see: Langer, M. S. & Zucker, S. W. (1994)

---

## Perception of shape from shading

### Perception of shape from shading & lighting direction

Some studies seem to indicate that human observers are not very accurate at estimating the local orientation of surface normals. We have a bias towards the fronto-parallel plane.

We also seem to have difficulty in estimating the slant of a light source (Todd, J. T., & Mingolla, E., 1983; Mingolla, E., & Todd, J. T. 1986). We are better at the tilt.

One problem with our above shape from shading analysis is that we assumed light source direction is known.

In fact, *human observers do seem to assume that the light source is from above*. This can be seen in the age-old "crater illusion" in which we have vertical luminance gradients of intensity giving an impression of a convex object on the left, and a concavity on the right (Ramachandran, V. S., 1988). (See previous lecture and code below that makes a bump lit from above and one that is upside-down). Interestingly, there also seems to be a slight bias towards assuming the light source is coming from the left (Sun & Perona, 1998).

### Some cool demonstrations

Let's look at some perceptual demonstrations of shape from shading due to V. S. Ramachandran. These will give us some clues as to how the human visual system might be resolving ambiguity.

Take a look at the shape from shading video clips at: <http://cbc.ucsd.edu/ramailusions.html>

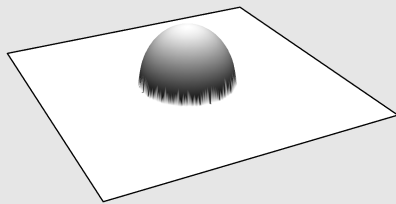
Ramachandran, VS (1990). Perceiving shape from shading. *The perceptual world: Readings from Scientific American magazine*. I. Rock. New York, NY, US, W. H. Freeman & Co, Publishers: 127-138.

## **Mathematica demonstration of: Illumination direction & shape ambiguity using the lambertian model**

### ■ Lambertian hemisphere

```
In[84]:= bump[x_, y_] := If[x^2 + y^2 > 1, 0, Sqrt[1 - x^2 - y^2]];
g1 = Plot3D[bump[x, y], {x, -3, 3}, {y, -3, 3}, PlotPoints -> 64,
  PlotRange -> {0, 2}, Mesh -> False,
  Lighting -> {"Directional", RGBColor[1, 1, 1], {{5, 5, 4}, {5, 5, 0}}},
  ViewPoint -> {1, 2, 1}, AxesLabel -> {"x", "y", "z"}, Ticks -> False,
  AspectRatio -> 1, ImageSize -> Small, Boxed -> False, Axes -> False]
```

Out[85]=



### ■ Calculate surface normals of surface

```
In[86]:= Clear[nx, ny]; little = 0.001; big = 1000;

nx[x_, y_] := Evaluate[D[bump[x, y], x]];
ny[x_, y_] := Evaluate[D[bump[x, y], y]];
nx[x_, y_] := big /; x^2 + y^2 == 1
ny[x_, y_] := big /; x^2 + y^2 == 1
nx[x_, y_] := 0. /; {x^2 + y^2 > 1}
ny[x_, y_] := 0. /; x^2 + y^2 > 1
```

$nx[x,y]$  is  $\frac{\partial z}{\partial x}$ , and similarly for  $ny$ . The rate of change of depth range is greatest as the face slopes away from the viewpoint:

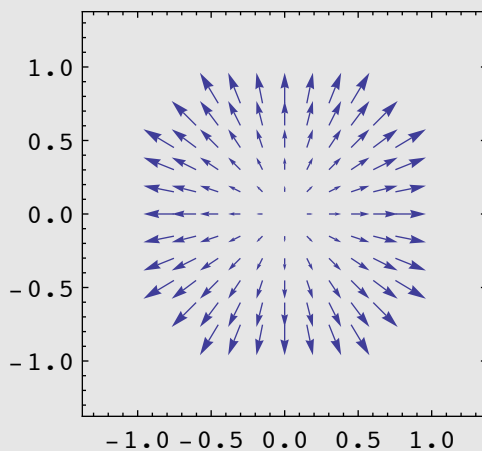
## Lambertian rendering: specification for normals, light, reflectance

### ■ Unit surface normals

Normalize the surface normal vectors to unit length:

```
In[93]:= normface[x_, y_] := -{nx[x, y], ny[x, y], 1} /
  (Sqrt[nx[x, y]^2 + ny[x, y]^2 + 1]);
VectorPlot[{normface[x, y][[1]], normface[x, y][[2]]}, {x, -1.2, 1.2},
  {y, -1.2, 1.2}, ImageSize -> Small]
```

Out[94]=



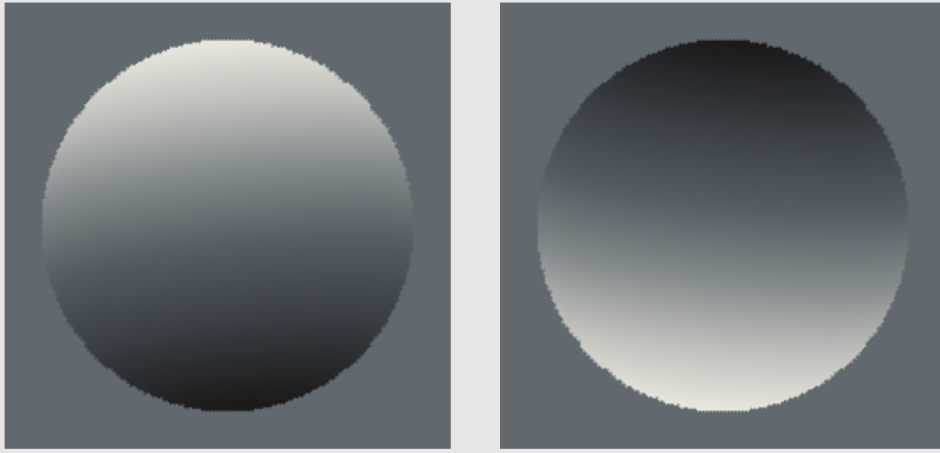
### ■ Render bump surface

Use the lambertian equation to render the surface illuminated from above and from below.

```
In[95]:= a[x_, y_] := 1;
s = {-10, 100, 10}; s = N[s / Sqrt[s.s]];
b1 = DensityPlot[a[x, y] * normface[x, y].s, {x, -1.2, 1.2},
  {y, -1.2, 1.2}, Mesh -> False, PlotPoints -> 64, Frame -> False,
  PlotRange -> {-1, 1}, ColorFunction -> "GrayTones"];
s = {-10, -100, 10}; s = N[s / Sqrt[s.s]];
b2 = DensityPlot[a[x, y] * normface[x, y].s, {x, -1.2, 1.2},
  {y, -1.2, 1.2}, Mesh -> False, PlotPoints -> 64, Frame -> False,
  PlotRange -> {-1, 1}, ColorFunction -> "GrayTones"];
```

In[100]:= `GraphicsArray[{b1,b2}]`

Out[100]=

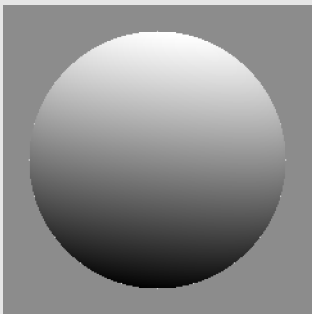


### Shading and contour interaction

A challenging problem is understanding how shape-from-contour interacts with shape from shading. Contour can override shading information. Let's cut out sections of part of an apparent sphere:

```
In[101]:= s={-10,100,10}; s=N[s/Sqrt[s.s]];
temp=Transpose[Table[normface[x,y].s,{x,-1.2,1.2,0.01},{y,-1.2,1.2,0.01}]];
b1=ArrayPlot[temp,Mesh->False,Frame->False,PlotRange->{-1,1}]
```

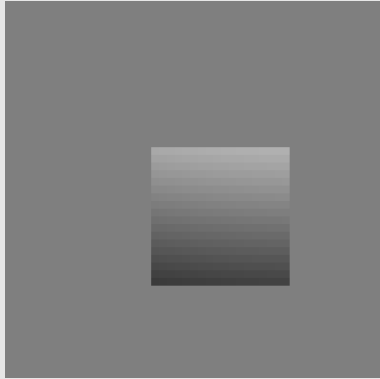
Out[103]=



..this patch is an apparent cylinder:

```
In[104]:= s={-10,100,10}; s=N[s/Sqrt[s.s]];
temp=Transpose[Table[If[(x<-1/4 || y<-1/4) || (x>3/5 || y>3/5),0,normface[x
ArrayPlot[temp,Mesh->False,Frame->False, PlotRange->{-1,1}]
```

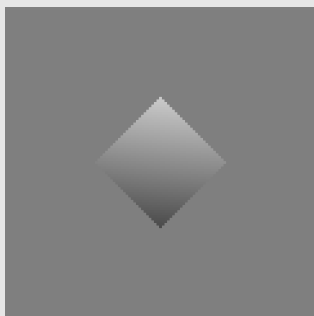
Out[106]=



This patch is an apparent diamond:

```
In[107]:= s={-10,100,10}; s=N[s/Sqrt[s.s]];
temp=Transpose[Table[If[Abs[x]+Abs[y]>.5,0,normface[x,y].s],{x,-1.2,1.2,0.0
ArrayPlot[temp,Mesh->False,Frame->False, PlotRange->{-1,1}]
```

Out[109]=



Somehow the visual system incorporates boundary conditions. This could be done either through low-level "propagation" analogous to the Ikeuchi & Horn algorithm, or it could be done by using the contour to "index" shape classes (polyhedra, cylinders, spheres, pills, chiclets, etc..) with which to infer the interior shape.

## Demo

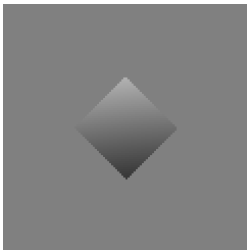
Another demo (due to David Knill) you can try to make yourself is generate a vertical sine-wave grating. Make the contours at the top and bottom have either 1) the same frequency as the grating or 2) twice the frequency.

One research problem is how do specularities affect the convex vs. concave perception in the above figures? It has been found that human observers can use the stereoscopic position of a specularity (which is in front of a concave surface, and behind a convex surface) to disambiguate the shape (Blake and Bülhoff, 1990; 1991).

## Project idea

Here is a simple project idea. One would expect that the influence of contour on shape-from-shading would depend on whether the contour is "attached" to the shaded region or not. A simple way of manipulating this is to use stereo. For example, one could make the boundary of a sphere to appear either at the correct boundary (i.e. towards the back of the bulge), or in front of the bulge, as if it is a hole. Perceived shape should be affected. One might even be able to design a shading pattern that looks very different depending on what the boundary gets attached to. For a discussion of "intrinsic" vs. "extrinsic" contours, see: Nakayama, K., & Shimojo, S. (1992).

Suppose that the diamond boundary below appeared as a diamond-shaped whole. Would one still interpret the shading as a pill, or would the shape of the internal region now appear more spherical?



## ■ Psychophysics

There have been a number of studies of the kinds of psychophysical errors made in slant and tilt judgements with a shape-from-shading algorithm that assumes smoothness in slant and tilt and knows the boundary values (e.g. Mamassian and Kersten. 1995). Mamassian and Kersten investigated the perception of local surface orientation for a simple object with regions of elliptic (egg-shaped) and hyperbolic (saddle-shaped) points. The local surface orientation was measured by the slant and tilt of the tangent plane at different points of the surface under several different illumination conditions. They found an underestimation of the perceived slant, and a larger variance for the perceived tilt. They also found that subjects were better at estimating the surface orientation when the shape was locally egg-shaped rather than saddle-shaped or cylindrical. From converging evidence based on (i) the light direction most consistent with the observer's settings, (ii) a supplementary experiment where the object is displayed as a silhouette, and (iii) the computer simulations of the shape from shading algorithm, they concluded that the occluding contour was the dominant source of information used by the observers.

At this point, very little is known about the neurophysiology of shape from shading/contour, although there has been speculation about the role of V1 spatial filters (e.g. simple cells) in the local estimate of shape; see some early ideas by Lehky and Sejnowski (1988), and by Pentland (1982). Knill and Kersten showed that orthogonal oriented linear filters could be used to estimate surface normals for Lambertian surfaces; however, we have very little idea of how neural systems may represent parameters of shape. Even basic issues such as viewpoint dependent (via extrinsic geometry) vs. viewpoint dependent (via intrinsic geometry) representations are unclear.

## ■ Neuroimaging

There have been a number of studies of the neural basis of human object and shape perception using fMRI. For example, see: Kourtzi and Kanwisher (2000) and Moore and Engel, 2001).



### **How to integrate? Shape from X1 and X2 and X3**

Later in the course, we will look at the problem of cue integration for depth and shape.

#### **■ How to pick? Shape from X1 or X2 or X3**

How should a visual system decide what kind of shape cue it has (shading from lambertian material, shading from shiny material, shading due to texture), and thus what kind of algorithm to use? One approach is to run a set of experts in parallel and pick the solution from the one that seems to "know what it is doing". Another approach is to use robust algorithms that are not sensitive to the details of the cues.

---

## Next time

### Bayes formulation of shape from shading & bas-relief Introduction to motion analysis

---

## Appendices

### References

- Belhumeur, P. N., Kriegman, D. J., & Yuille, A. (1997). *The Bas-Relief Ambiguity*. Paper presented at the IEEE Conf. on Computer Vision and Pattern Recognition.
- Brady, M., & Yuille, A. (1983). An Extremum Principle for Shape from Contour. *Proceedings IJCAI*, 969-972.
- Bülthoff, H. H., & Yuille, A. (1990). Shape-from-X: Psychophysics and computation. *Proceedings of the SPIE*, 1383, 165-172.
- Dupuis, P., & Oliensis, J. (1994). An optimal control formulation and related numerical methods for a problem in shape reconstruction. *The Annals of Applied Probability*, 4(No. 2), 287-346.
- Freeman, W. T. (1994). The generic viewpoint assumption in a framework for visual perception. *Nature*, 368(7 April 1994), 542-545.
- Freeman, W. T., Pasztor, E. C., & Carmichael, O. T. (2000). Learning low-level vision. *Intl. J. Computer Vision*, 40(1), 25-47.
- Gray, Alfred. *Modern Differential Geometry of Curves and Surfaces with Mathematica, Second Edition*. (CRC Press, 1997) (<http://store.wolfram.com/view/ISBN0849371643/?39FED89E-38F0>)
- Horn, B., & Brooks, M. J. (1989). *Shape from shading*. Cambridge, Mass.: MIT Press.
- Humphrey, G. K., Goodale, M. A., Bowen, C. V., Gati, J. S., Vilis, T., Rutt, B. K., et al. (1997). Differences in perceived shape from shading correlate with activity in early visual areas. *Curr Biol*, 7(2), 144-147.
- Humphrey, G. K., Symons, L. A., Herbert, A. M., & Goodale, M. A. (1996). A neurological dissociation between shape from shading and shape from edges. *Behavioural Brain Research*, 76(1-2), 117-125.
- Ikeuchi, K., & Horn, B. K. P. (1981). Numerical shape from shading and occluding boundaries. *Art. Intel.*, 17, 141-184.
- Kersten, D., O'Toole, A. J., Sereno, M. E., Knill, D. C., & Anderson, J. A. (1987). Associative learning of scene parameters from images. *Appl. Opt.*, 26, 4999-5006.
- Knill, D. C., & Kersten, D. (1990). Learning a near-optimal estimator for surface shape from shading. *CVGIP*, 50(1),

75-100.

Knill, D. C. (1992). Perception of surface contours and surface shape: from computation to psychophysics. *J Opt Soc Am [A]*, 9(9), 1449-1464.

Koenderink, J. J. (1998). Pictorial relief. *Philosophical Transactions of the Royal Society of London Series a-Mathematical Physical and Engineering Sciences*, 356(1740), 1071-1085.

Koenderink, J. J. (1990). *Solid Shape*. Cambridge, MA: MIT Press.

Koenderink, J. J., van Doorn, A. J., & Kappers, A. M. (1992). Surface perception in pictures. *Perception & Psychophysics*, 5, 487-496.

Koenderink, J.-J., Kappers, A., & van Doorn, A. (1992). Local Operations: The Embodiment of Geometry. *Artificial and Biological Vision Systems*.

Kourtzi, Z., & Kanwisher, N. (2000). Cortical regions involved in perceiving object shape. *J Neurosci*, 20(9), 3310-3318.

Langer, M. S., & Zucker, S. W. (1994). Shape from Shading on a Cloudy Day. *Journal of the Optical Society of America A*, 11(2), 467-478.

Langer M.S. and Bülthoff H. H. (2000), Depth discrimination from shading under diffuse lighting. *Perception*. 29 (6) 649-660.

Langer, M. S., & Bulthoff, H. H. (2001). A prior for global convexity in local shape-from-shading. *Perception*, 30(4), 403-410.

Mamassian, P., Kersten, D., & Knill, D. C. (1996). Categorical local-shape perception. *Perception*, 25(1), 95-107.

Mamassian P & Kersten D. (1996). Illumination, shading and the perception of local orientation. *Vision Research*., 36(15), 2351-2367.

Mingolla, E., & Todd, J. T. (1986). Perception of Solid Shape from Shading. *Biological Cybernetics*, 53, 137-151.

Moore, C., & Engel, S. A. (2001). Neural response to perception of volume in the lateral occipital complex. *Neuron*, 29(1), 277-286.

Norman, J. F., Todd, J. T., & Orban, G. A. (2004). Perception of three-dimensional shape from specular highlights, deformations of shading, and other types of visual information. *Psychol Sci*, 15(8), 565-570.

Oliensis, J. (1991). Uniqueness in Shape From Shading. *The International Journal of Computer Vision*, 6(2), 75-104.

Pentland, A. P. (1990). Linear shape from shading. *International Journal of Computer Vision*, 4, 153-162.

Pentland, A. P. (1988). *Shape Information from Shading: A Theory About Human Perception*.

TR-103. Paper presented at the M.I.T. Media Lab Vision Sciences.

Reichel, F. D., Todd, J. T., & Yilmaz, E. (1995). Visual-Discrimination of Local Surface Depth and Orientation. *Perception & Psychophysics*, 57(8), 1233-1240.

Sun, J., & Perona, P. (1998). Where is the sun? *Nature Neuroscience*, 1(3), 183-184.

Todd, J. T., & Mingolla, E. (1983). Perception of Surface Curvature and Direction of Illumination from Patterns of Shading. *Journal of Experimental Psychology: Human Perception & Performance*, 9(4), 583-595.

Todd, J. T., Norman, J. F., Koenderink, J. J., & Kappers, A. M. L. (1997). Effects of texture, illumination, and surface reflectance on stereoscopic shape perception. *Perception*, 26(7), 807-822.

Yuille, A. L. (1987). *Shape from Shading, Occlusion and Texture*. (No. A.I. Memo No. 885): M.I.T.

Yuille, A. L., & Bülthoff, H. H. (1996). Bayesian decision theory and psychophysics. In D. C. Knill & W. Richards (Eds.), *Perception as Bayesian Inference* (pp. 123-161). Cambridge, U.K.: Cambridge University Press.

© 2004, 2006, 2008, 2010 Daniel Kersten, Computational Vision Lab, Department of Psychology, University of Minnesota.  
kersten.org