Computational Vision
U. Minn. Psy 5036
Daniel Kersten
Lecture 16: Shape from X

Initialize

Spell check off

In[88]:=
```
Off[General::spell1];
```

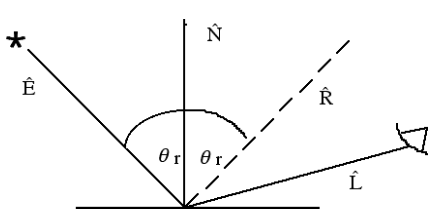
In[89]:=
```
SetOptions[ArrayPlot, ColorFunction → "GrayTones",
   DataReversed → True, Frame → False, AspectRatio → Automatic,
   Mesh → False, PixelConstrained → True, ImageSize → Small];
SetOptions[ListPlot, ImageSize → Small];
SetOptions[Plot, ImageSize → Small];
SetOptions[DensityPlot, ImageSize → Small, ColorFunction → GrayLevel];
SetOptions[ListPlot3D, SphericalRegion -> True,
   Axes -> False, Boxed → False, ImageSize → Small];
SetOptions[Plot3D, SphericalRegion -> True, Axes -> False,
   Boxed → False, Mesh → False, ImageSize → Small];
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



$$L(x, y) = r(x, y)\,\hat{E} \cdot \hat{N}(x, y)$$

For a constant reflectance surface:

$$L(x, y) = \hat{E} \cdot \hat{N}(x, y)$$

Note that the luminance doesn't depend on the viewpoint, $\hat{L}$. Further, the cosine fall-off exactly compensates for the loss of light intensity flux as the surface is inclined towards or away from the eye, giving "brightness constancy" for free.

The above formula is only valid for points that aren't blocked by the light source. And doesn't taken into account how light bounces around between surfaces. Specularities depend on how much light gets reflected in the direction of $\hat{R}$. Adding two "fudge terms", one for ambient light and one for specularities, provides a useful, although physically incorrect, model ("Phong") for a world of plastic surfaces with a dominant point light source, and bathed in an "ambient" background:

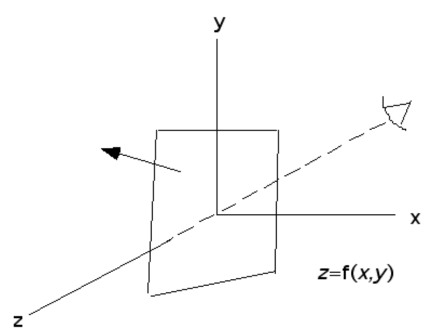
$$L = r_a E_a + r \hat{E}_p \cdot \hat{N} + r_s \left(\hat{R} \cdot \hat{L}\right)^n$$

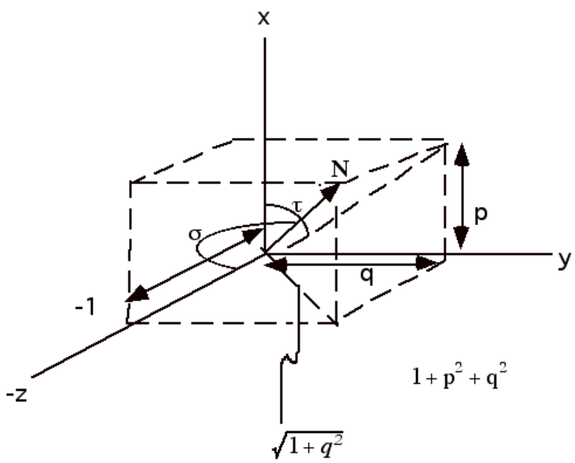
One can include other fudge factors, for example to capture the "proximity luminance " bias:

$$L = r_a E_a + \frac{E_p}{R + K} \left[r(\hat{E} \cdot \hat{N}) + r_s (\hat{R} \cdot \hat{L})^n\right]$$

Local surface orientation representations

Local, dense, metric, viewer-dependent coordinate system (e.g. slant from observer).
Gradient space (p,q), surface normal direction, $\frac{(p,q)}{|(p,q)|}$, and slant and tilt, (σ,τ). Pros and cons.





Let $\phi(x,y,z)$=f(x,y)-z .

$$\nabla\phi = \frac{\partial f}{\partial x}\hat{\mathrm{i}} + \frac{\partial f}{\partial y}\hat{\mathrm{j}} - \hat{\mathrm{k}} = p\hat{\mathrm{i}} + q\hat{\mathrm{j}} - \hat{\mathrm{k}} \longrightarrow (p, q, -1)$$

is a normal vector to the surface at *(x,y,f(x,y))*. Gradient space is defined by the mapping of (p,q,-1)-->(p,q), i.e. the orthographic projection. If we normalize to unit length, we get an expression for the surface normal at location (x,y):

$$\mathbf{N}(x,y) = \left(\frac{p}{\sqrt{p^2+q^2+1}}, \frac{q}{\sqrt{p^2+q^2+1}}\right).$$

Slant and tilt provide an alternative representation, given by:

$$\sigma = \tan^{-1}\left(\sqrt{q^2+p^2}\right)$$

$$\tau = \tan^{-1}\left(\frac{q}{p}\right)$$

Today

Perceptual demonstrations of shape from X

Computation of local, view-dependent shape from shading

Perception of "Shape from X"

How can we define shape more precisely?

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, i.e. the task requirements.

Later we will return to the "discounting" issues.

We’ve seen that there is 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, cues can be manipulated independently to study how humans combine and weight cues.

Last time we had an overview of cues to distance and to shape.

Let's look more closely at several categories of shape cues, including some for the last lecture.

Texture

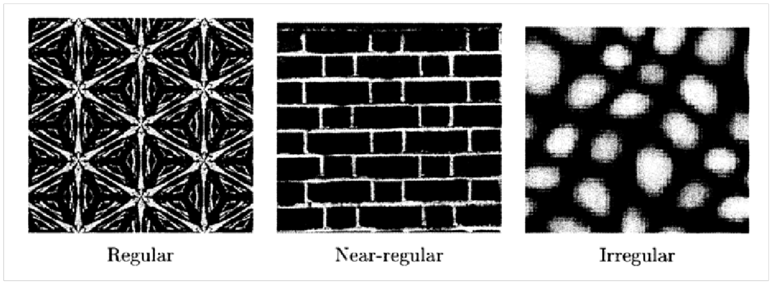
The term “texture” can be used in several different contexts. It can refer to properties perceived in the image, but it can also refer to object textural properties (e.g. that correlate with physical material causes such as roughness). But texture can also be informative (an image-based “cue’’) as to an object’s shape.

(Later on we will talk in more detail about how to model and infer surface material properties which may be characterized by texture patterns.)

Texture by definition has a degree of regularity represented by a small scale pattern that gets repeated over a larger spatial scale. Texture can be highly regular, or only regular in a statistical sense. Textures can be precisely defined in terms of symmetries. In particular, translational symmetry says that a pattern remains unchanged if shifted by a fixed discrete amount.

► 1. Given an image with translational texture symmetry, how could you detect and measure the amount of translation that defines the symmetry? For a function tool, see: FindTransientRepeat[].

For a stochastic or statistical texture, there is a statistic or collection of statistics (e.g. intensity variance, or local correlation) that is unchanged with spatial translation. Textures are sometimes described as spatially "homogeneous" with respect to some statistical measure. And sometimes the adjective "uniform" is added to make sure that one isn't referring to texture that is deviating from homogeneity, i.e. becoming a spatial "flow".



From: http://repository.cmu.edu/robotics/687, Deformable texture : the irregular-regular-irregular cycle. Yanxi Liu and Wen-Chieh Lin.

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. The general idea behind “shape from texture” is to assume that this underlying uniform texture that has gotten deformed by a change in the viewpoint relative to the surface patch being viewed.

Let’s first look at the simplest case where we have a planar surface and we want to estimate the orientation of the surface relative to the viewpoint. When we think about “shape”, a small planar surface can be considered a local approximation to a curved surface, and then shape from texture amounts to computing surface normal vectors over each patch. But the same texture cues can also be used to estimate the orientation of a large flat surface (such as a table top).

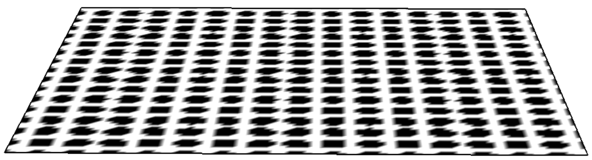
Texture and slant/tilt

Define a function to place circular disks on a regular grid

In[134]:=
```
checker[x_,y_,space_,radius_]:= If[(Mod[x,space]^2+Mod[y,space]^2<radius^2) || (Mod[-x,spa
piccheck = Table[{checker[x, y, 32, 12]}, {x, 1, 512}, {y, 1, 512}];
```

In[136]:=
```
ListPlot3D[Table[1, {x, -5, 5, .1}, {y, -5, 5, .1}], Mesh → None,
 VertexColors → {piccheck[[5 ;; -5 ;; 5, 5 ;; -5 ;; 5]]}, Lighting → "Neutral"]
```

Out[136]=



► 2. How important is the contour information? Judge the local orientation through an aperture (use a straw or a rolled-up piece of paper)

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:

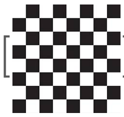
1) spatial gradient of the density of texture elements (“tokens”): the image token density increases with distance;

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;

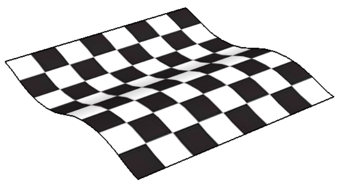
3) the ratio of the back width to the front width of an individual element gets smaller with slant (imagine a small square, linear perspective on small scale). (See Knill).

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

Texture and shape

In[99]:=
```
Plot3D[Sin[0 + Sin[y / 2]], {x, -8, 8}, {y, -8, 8},
 PlotStyle → Texture[
```

```
], BoxRatios → {1, 1, 0.08}, Lighting -> "Neutral"]
```

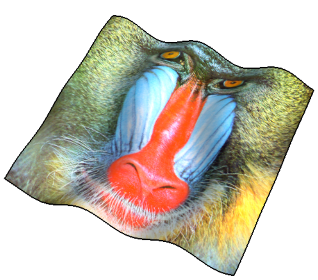
Out[99]=



Try mapping an inhomogeneous pattern. In this demo, you can experience the perceptual tension between different sources of shape information. There are texture changes due to the 3D surface onto which the Mandrill has gotten mapped, texture and material changes due to the original actual shape of the Mandrill's head. And as you rotate, you experience motion flow cues from the 3D surface. We talk about cue integration, and cue conflict below.

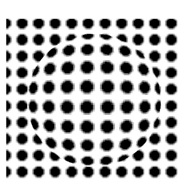
In[100]:=
```
pic = Reverse[ExampleData[{"TestImage", "Mandrill"}, "Data"] / 255.];
Plot3D[Sin[x + Sin[y / 2]], {x, -8, 8}, {y, -8, 8},
 PlotStyle → Texture[pic], BoxRatios → {1, 1, 0.08}, Lighting -> "Neutral"]
```

Out[101]=



► 3. Try a web search for "folded dollar motion illusion".

► 4. Adapt the above function to map a regular dot texture onto a hemisphere



- Project idea: Map a small single textural element (e.g. a disk) onto an "invisible" bump. Allow the user to move the element around. Does your perceptual system infer the underyling shape? (Look up Joseph Lappin).

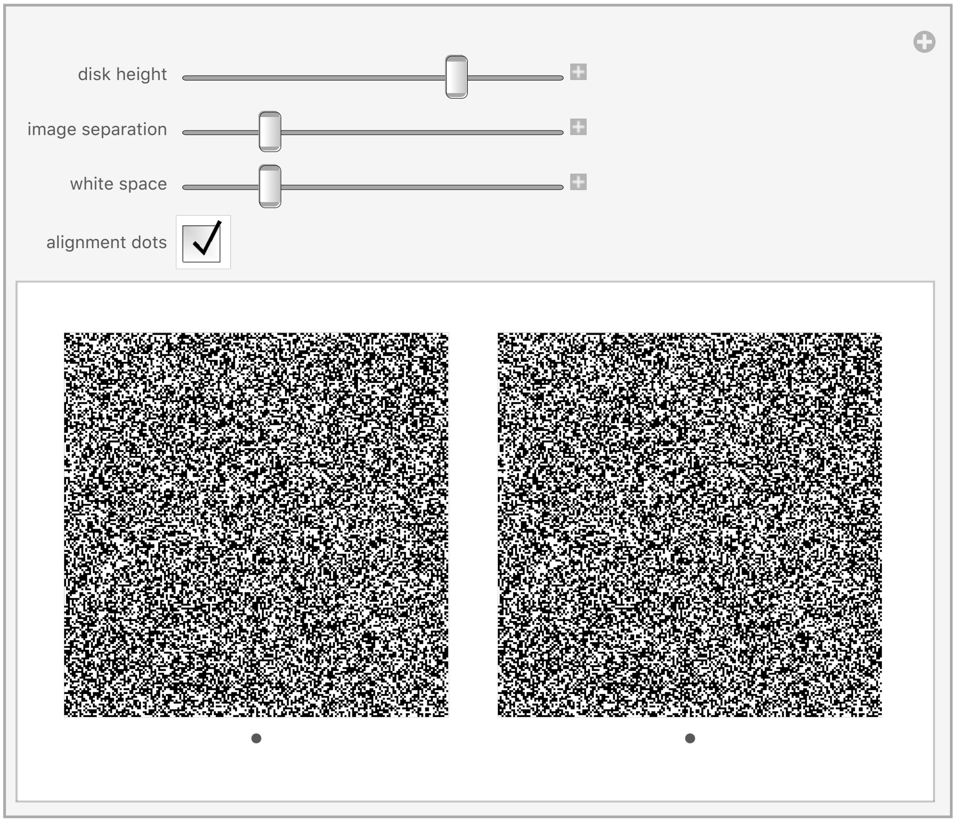
Stereo disparity

Random dot stereograms

Bela Julesz produced the first random dot stereogram in the late 1950s. Here is a demo producing simple planar surface in depth. Take from: http://demonstrations.wolfram.com/FloatingDiskStereogram/

In[155]:=
```
Manipulate[DynamicModule[{nb = 200, nf = 100, background, foreground},
  SeedRandom[1];
  background = Table[RandomInteger[{0, 1}], {nb}, {nb}];
  foreground = Table[{RandomInteger[{0, 1}],
     If[(i - nf / 2)^2 + (j - nf / 2)^2 < nf^2 / 4, 1, 0]}, {i, 1, nf}, {j, 1, nf}];
  Pane[Grid[{{Graphics[{
         Raster[background],
         Raster[foreground,
          Dynamic[{{1, 1} (nb - nf) / 2 + {shift, 0},
            {nf, nf} + {1, 1} (nb - nf) / 2 + {shift, 0}}]]
        }, PlotRange → {{0, nb}, {0, nb}}, ImageSize → {nb, nb}],
       Graphics[{
         Raster[background],
         Raster[foreground,
          Dynamic[{{1, 1} (nb - nf) / 2 - {shift, 0},
            {nf, nf} + {1, 1} (nb - nf) / 2 - {shift, 0}}]]
        }, PlotRange → {{0, nb}, {0, nb}}, ImageSize → {nb, nb}]
      }, Table[Dynamic[If[dots, "•", ""]], {2}]},
    Spacings → Dynamic[{columnSpacing, Automatic}],
    Alignment → Center], ImageMargins → Dynamic[margins]]
 ],
 {{shift, 4, "disk height"}, -8, 8, 1},
 {{columnSpacing, 2, "image separation"}, 0, 10},
 {{margins, 10, "white space"}, 0, 50},
 {{dots, True, "alignment dots"}, {False, True}}
]
```

Out[155]=



Cross your eyes so that you see three dots (rather than two) at the bottom. Smooth shape can also be conveyed through disparity -- “shape from stereo”. A random dot version can be made with a pair of images as above, but let’s make one using the autostereogram technique invented by Christopher Tyler.

Autostereograms -- "Magic eye" style

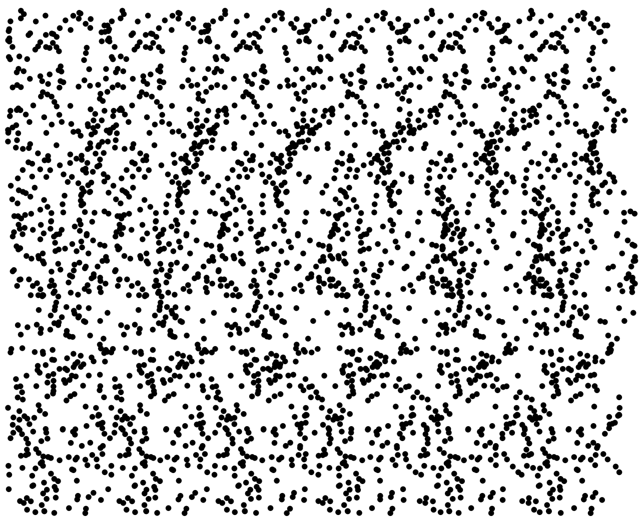
Christopher Tyler produced the first autosterogram in 1979. The wiki page provides nice overview. http://en.wikipedia.org/wiki/Autostereogram.

In[185]:=
```
(*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))

Show[Graphics[Table[x = Random[Real, {-1, -0.6}];
   y = Random[Real, {-1, 1}];
   Table[x0 = x;
    x = x + 0.4 + f[x, y] / 15;
    {AbsolutePointSize[3], Point[{x0, y}]}, {6}], {500}]]]
```

Out[186]=



- Project idea: Does human vision show orientation-selective adaptation to stereoscopically defined shape?

Stereo + shading + surface contours

In general, there are multiple cues to shape. With the advent of 3D graphics, it became possible to parametrically manipulate multiple natural cues. For some of the first uses of 3D graphics to study perception of shape using naturalistic rendering, see: Todd, J. T., & Mingolla, E. (1983) and Bülthoff, H. H., & Mallot, H. A. (1988). Bülthoff & Mallot showed how shading, texture, and specular information contributed to the perception of curvature of an ellipse. These parameters can be explored using this next expression:

In[143]:=
```
ParametricPlot3D[{2 Cos[u] Sin[v], Sin[u] Sin[v], Cos[v]},
 {v, 0, Pi}, {u, 0, 2 Pi}, PlotStyle → Directive[Specularity[White, 30],
   Texture[ExampleData[{"ColorTexture", "WhiteMarble"}]]],
 TextureCoordinateFunction → ({#4, 2 #5} &), Lighting → "Neutral", Mesh → None,
 PlotRange → {{-2, 2}, {-1, 1}, {-1, 1}}, Axes → False, Boxed → False]
```

Out[143]=

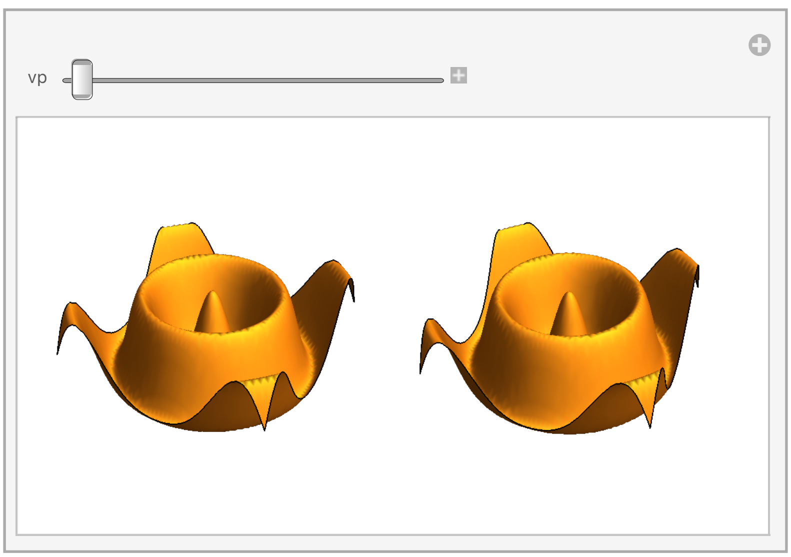
Here is an example of shading, contours and stereo. You can manipulate the disparity between the two views using the slider.

In[190]:=
```
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[191]:=
```
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[GraphicsRow[{pL, pR}]], {{vp, .8}, 0.8, 1.5}]
```

Out[191]=



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".

See: random.mov, cube.mov demos on the class web page.

Contours

Surface contour (markings)

Surface contours are particular form of texture which can provide strong cues to shape. Human perception treats the contours as if they are 2D projections of geodesics on a curved surface in 3D. See Knill, D. C. (1992).

In[107]:=
```
Plot[{Sin[x],Sin[x]+1,Sin[x]+2, Sin[x]+3, Sin[x]+4, Sin[x]+5},{x,0,10}, Axes->False,ImageS
```

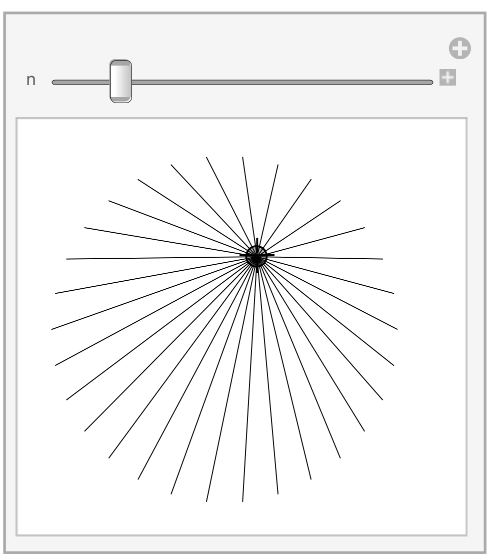
Out[107]=

Shape from moving contours

Here's an example taken from the *Mathematica* documentation. Apparent 3D shape emerges only with movement. In the demonstration below, use the computer mouse to drag the point where the lines converge.

In[108]:=
```
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[108]=

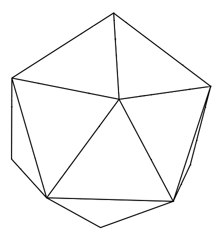


Surface contours (markings of orientation discontinuities--“creases”)

The lines mark surface orientation discontinuities.

In[197]:=
```
Graphics3D[{EdgeForm[{Opacity[1.0]}], FaceForm[{White, Opacity[1.]}],
   PolyhedronData["GreatDodecahedron", "Faces", "Polygon"]},
  Lighting → {"Ambient", White}, Boxed → False, ImageSize → Small,
  PlotRange → {{-1, 1}, {-1, 1}, {-1, 1}}]
```

Out[197]=

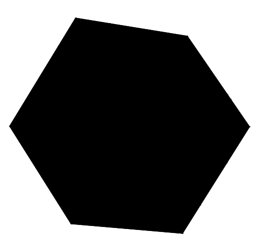


Bounding contours, (depth discontinuities), e.g. silhouette

Orientation discontinuities can coincide with depth discontinuities. Use the mouse to rotate the silhouetted object below.

In[110]:=
```
Graphics3D[{EdgeForm[{Opacity[1.0]}], FaceForm[{Black, Opacity[1.]}],
   PolyhedronData["GreatDodecahedron", "Faces", "Polygon"]},
  Lighting → {"Ambient", White}, Boxed → False, ImageSize → Small]
```

Out[110]=



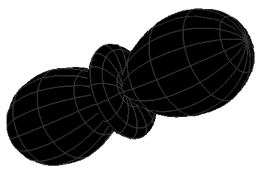
▶ 5. Try rotating the above silhouette. Is it possible to generate internal illusory contours given rotation motion?

...and smooth occluding contours

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

In[111]:=
```
SphericalPlot3D[1 + 2 Cos[2 θ], {θ, 0, Pi}, {ϕ, 0, 2 Pi}, Lighting → None,
 Boxed → False, Axes → False, AspectRatio → 1, ImageSize → Small]
```

Out[111]=



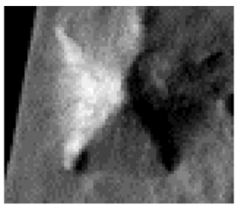
For a concrete example of computing shape, we next focus on the problem of shape from shading...

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". Beginning in 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 of the time. The controversy was firmly 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 produced a number of 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.

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

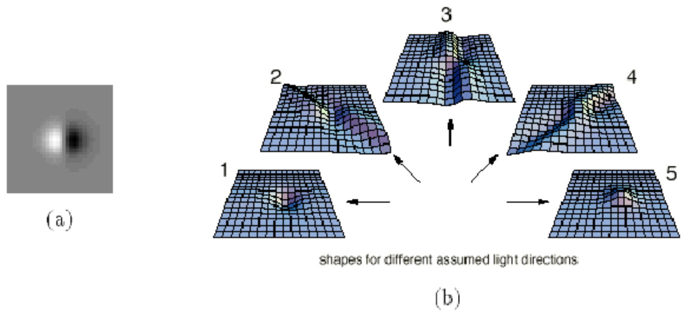


Figure 6: (a) Perceptually, this image has two possible interpretations. It could be a bump, lit from the left, or a dimple, lit from the right. (b) Mathematically, there are many possible interpretations. For a sufficiently shallow incident light angle, if we assume different light directions, we find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

► 6. Look up the ideas of "accidental" and "generic" views. How might they apply to "accidental illumination direction"?

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

Shape from shading can be treated as an inverse problem. Earlier in the course, we saw that inverse problems can be formalized in terms of Bayesian estimation. Recall that for 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 Lambertian 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 unit vectors, as in the last lecture. E is a point light source (we'll assume e=1), and E and N's directions can be expressed in gradient space:

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

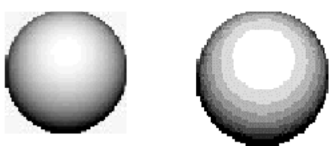
$$\hat{N} = \frac{(p, q, -1)}{\sqrt{p^2 + q^2 + 1}}$$

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 our model satisfy the data, i.e. 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)). To keep it simple, we assume r and (p_e, q_e) are known and fixed constants. But 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.

Just in case you are thinking that we might be able to use regions of constant intensity, note that lines of constant luminance in the image (called isophotes) 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 1) a region-based smoothness constraint (a local constraint) and 2) boundary conditions (global constraint based on the shape of the bounding contour). 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 deriviatives. 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 current estimate of the shape to make the surface smoother. The idea is to write an algorithm that searches for a surface that satisfies L - L_R = 0 AND has low magnitudes of the 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:

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

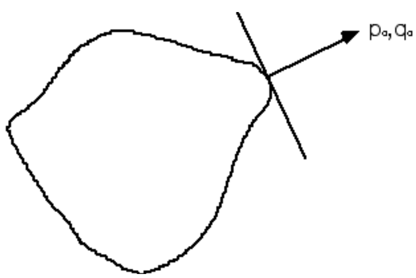
We want e(x,y) to be small. This will be the case if (p,q) is chosen so that the first term tends toward zero, and if the spatial derivatives are small (i.e. because we've chosen p,q to change slowly). But we want this local cost function to be small over the whole image.

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. λ is a weighting function on the smoothness. For example, it should be big if we have reason to believe our image 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 viewpoint) 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,y)'s and q(x,y)'s conditional on L(x,y). In Bayesian terms, the generative model determines the likelihood, and the laplacians of the gradient parameters determine the smoothness prior.

Human 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. It has also been shown that human visual judgments show 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.

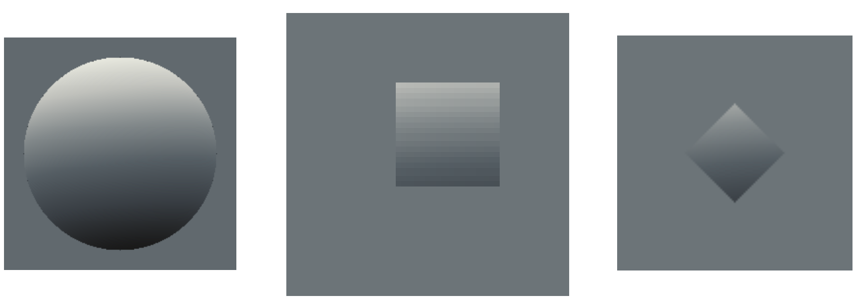
Human observers do assume that the light source is from above. This can be seen in the age-old "crater illusion" in which we have vertical luminance gradients of intensity giving an impression of a convex object on the left, and a concavity on the right (Ramachandran, V. S., 1988). (See previous lecture and code below that makes a bump lit from above and one that is upside-down). However, the light from above prior is relatively weak, which can be seen when vision is given a richer shape structure (Morgenstern, Y., Murray, R. F., & Harris, L. R. (2011).

Interestingly, there also seems to be a slight bias towards assuming the light source is coming from the left (Sun & Perona, 1998).

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:

```
(*Make a 3D bump, where bump=depth*)
Clear[x, y, nx, ny];
bump[x_, y_] := If[x^2 + y^2 > 1, 0, Sqrt[1 - x^2 - y^2]];
(*Calculate surface normals to define local shape. *)
(*nx[x,y] is ∂z/∂x,
and similarly for ny. We need to allow for infinities at the boundary,
where the rate of change of depth range is
 greatest as the face slopes away from the viewpoint.*)
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
normface[x_, y_] := -{nx[x, y], ny[x, y], 1} / (Sqrt[nx[x, y]^2 + ny[x, y]^2 + 1]);
(*Render images using Lambertian model*)
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}];
temp = Transpose[Table[If[(x < -1 / 4  ||  y < -1 / 4)  ||  (x > 3 / 5  ||  y > 3 / 5),
     0, normface[x, y].s], {x, -1.2, 1.2, 0.05}, {y, -1.2, 1.2, 0.05}]];
b2 = ArrayPlot[temp, Mesh → False, Frame → False,  PlotRange → {-1, 1}];
temp = Transpose[Table[If[Abs[x] + Abs[y] > .5, 0, normface[x, y].s],
    {x, -1.2, 1.2, 0.02}, {y, -1.2, 1.2, 0.02}]];
b3 = ArrayPlot[temp, Mesh → False, Frame → False,  PlotRange → {-1, 1}];
GraphicsRow[{b1, b2, b3}]
```



The same patch taken from the image of the sphere can appear as an apparent cylinder, or diamond depending on the bounding contours.

How the visual system incorporates boundary conditions is still not well-understood. 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.

7. Demo idea

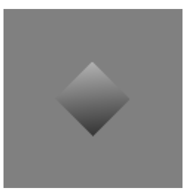
Another demo (originally due to David Knill) you can try to make yourself is generate a vertical sine-wave grating. Make the contours at the top and bottom have either 1) the same frequency as the grating or 2) twice the frequency.

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 Bülthoff, 1990; 1991).

- Project idea

Here is a 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 of shape-from-shading and contours

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.

▶ 8. Think about it: Does the human visual system compute pure "shape from shading" in the absence of other cues?

Neural basis of shape-from-X?

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 (1990) showed that orthogonal oriented linear filters could be used to estimate surface normals for Lambertian surfaces; however, we have 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.

There have been a number of studies of the neural basis of human object shape perception using fMRI. For example, see: Kourtzi and Kanwisher (2000), Moore and Engel (2001). See also Mamassian et al. (2003). Like most fMRI studies, these are informative regarding the general nature of representations in different cortical regions, but tell us little about the underlying computations.

Next time

Python

Bayes formulation of shape from shading & bas-relief

Task-dependency

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.

How to combine cues? Shape from X1 and X2 and X3

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

Appendices

Mathematica demonstration of: Illumination direction & shape ambiguity using

the lambertian model

Lambertian hemisphere from above

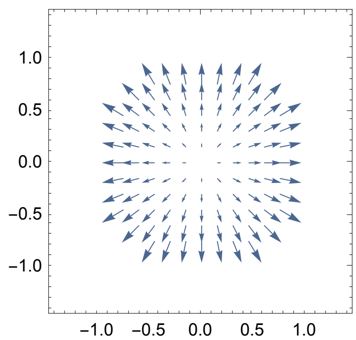
```
In[132]:= g1 = Plot3D[bump[x, y], {x, -3, 3}, {y, -3, 3},
   PlotPoints → 64, PlotRange → {0, 2}, Mesh → False,
   Lighting → {{"Directional", RGBColor[1, 1, 1], {{5, 5, 4}, {5, 5, 0}}}},
   ViewPoint → {1, 2, 1}, AxesLabel → {"x", "y", "z"}, Ticks → False,
   AspectRatio → 1, ImageSize → Small, Boxed → False, Axes → False]

Out[132]= $Aborted
```

Lambertian rendering: specification for normals, light, reflectance

Visualize the surface normals

```
VectorPlot[{normface[x, y][[1]], normface[x, y][[2]]},
 {x, -1.2, 1.2}, {y, -1.2, 1.2}, ImageSize → Small]
```

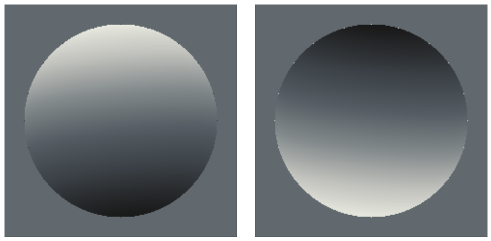


Render bump surface

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

```
a[x_, y_] := 1;
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}];
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}]];
b2 = ArrayPlot[temp, Mesh → False, Frame → False, PlotRange → {-1, 1}];
```

```
GraphicsRow[{b1, b2}]
```



Linear approximation

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:

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

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

where we use the notation p->pn, q->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}->{0,0}. The first derivative with respect to pn is:

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

$$\frac{\text{pe}}{\sqrt{1 + \text{pe}^2 + \text{qe}^2}}$$

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:

```
Series[Lmodel[pn,qn,pe,qe], {pn, 0, 1}, {qn, 0, 1}]
```

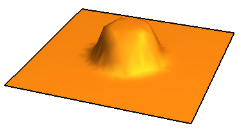
$$\left(\frac{1}{\sqrt{1 + \text{pe}^2 + \text{qe}^2}} + \frac{\text{qe qn}}{\sqrt{1 + \text{pe}^2 + \text{qe}^2}} + O[\text{qn}]^2\right) + \left(\frac{\text{pe}}{\sqrt{1 + \text{pe}^2 + \text{qe}^2}} + O[\text{qn}]^2\right) \text{pn} + O[\text{pn}]^2$$

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)]));

```
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]];
```

```
Plot3D[bump[x, y], {x, -3, 3}, {y, -3, 3}, Mesh → False,
 PlotRange → {0, .5}, ImageSize → Small, Axes → False, Boxed → False]
```



Light source:

```
{pe,qe}={1,1};
```

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

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

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

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"?

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).

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 straightfoward 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)

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., & Mallot, H. A. (1988). Integration of depth modules: stereo and shading. Journal of the Optical Society of America A, 5(10), 1749–1758.
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 Buelthoff 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.
Mamassian, P., Jentzsch, I., Bacon, B. A., & Schweinberger, S. R. (2003). Neural correlates of shape from shading. *NeuroReport*, *14*(7), 971–975. doi:10.1097/01.wnr.0000069061.85441.f2
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.
Morgenstern, Y., Murray, R. F., & Harris, L. R. (2011). The human visual system's assumption that light comes from above is weak. *Proceedings of the National Academy of Sciences of the United States of America*, *108*(30), 12551–12553. doi:10.1073/pnas.1100794108/-/DCSupplemental/pnas.201100794SI.pdf
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.