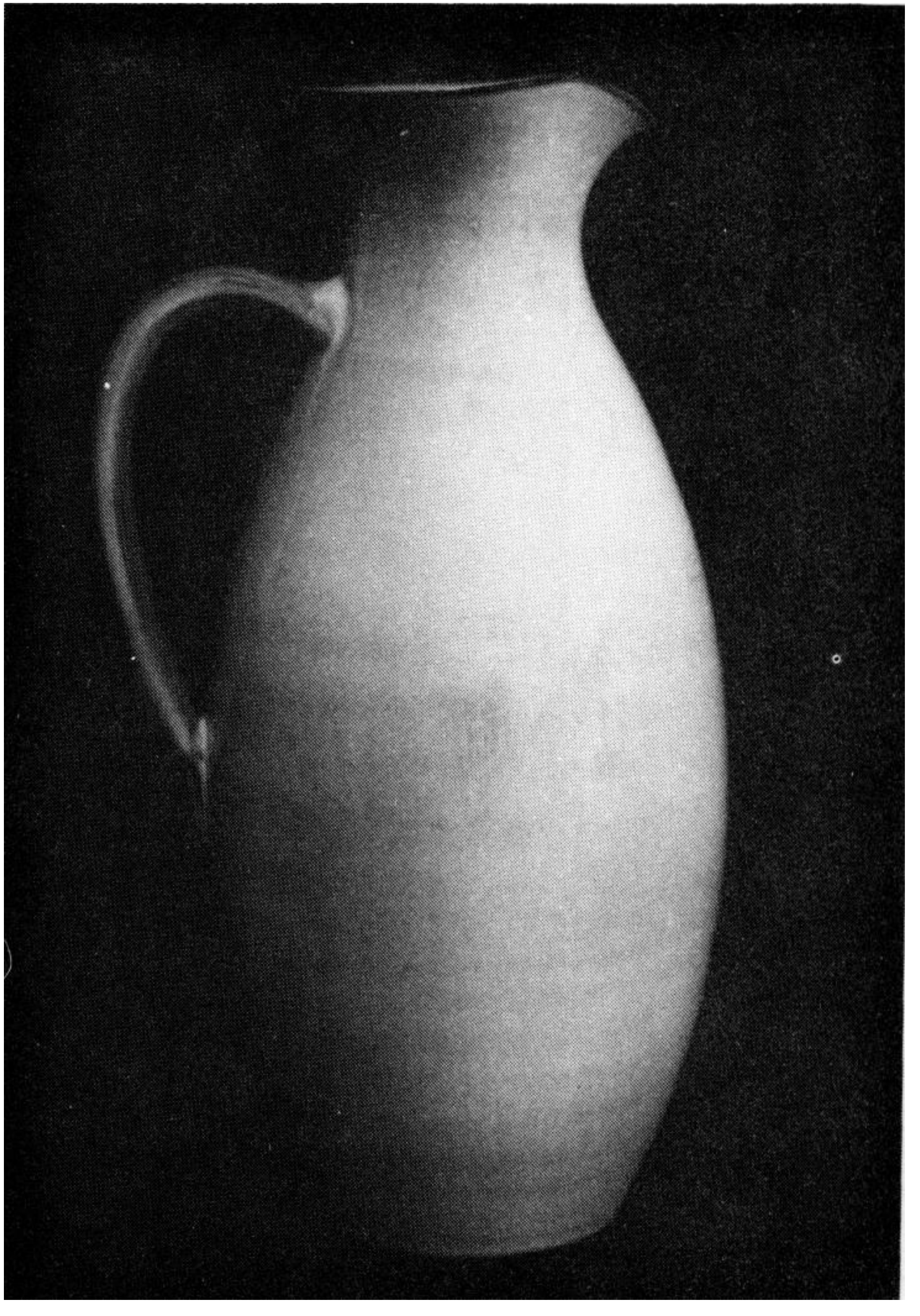
What is visual structure?

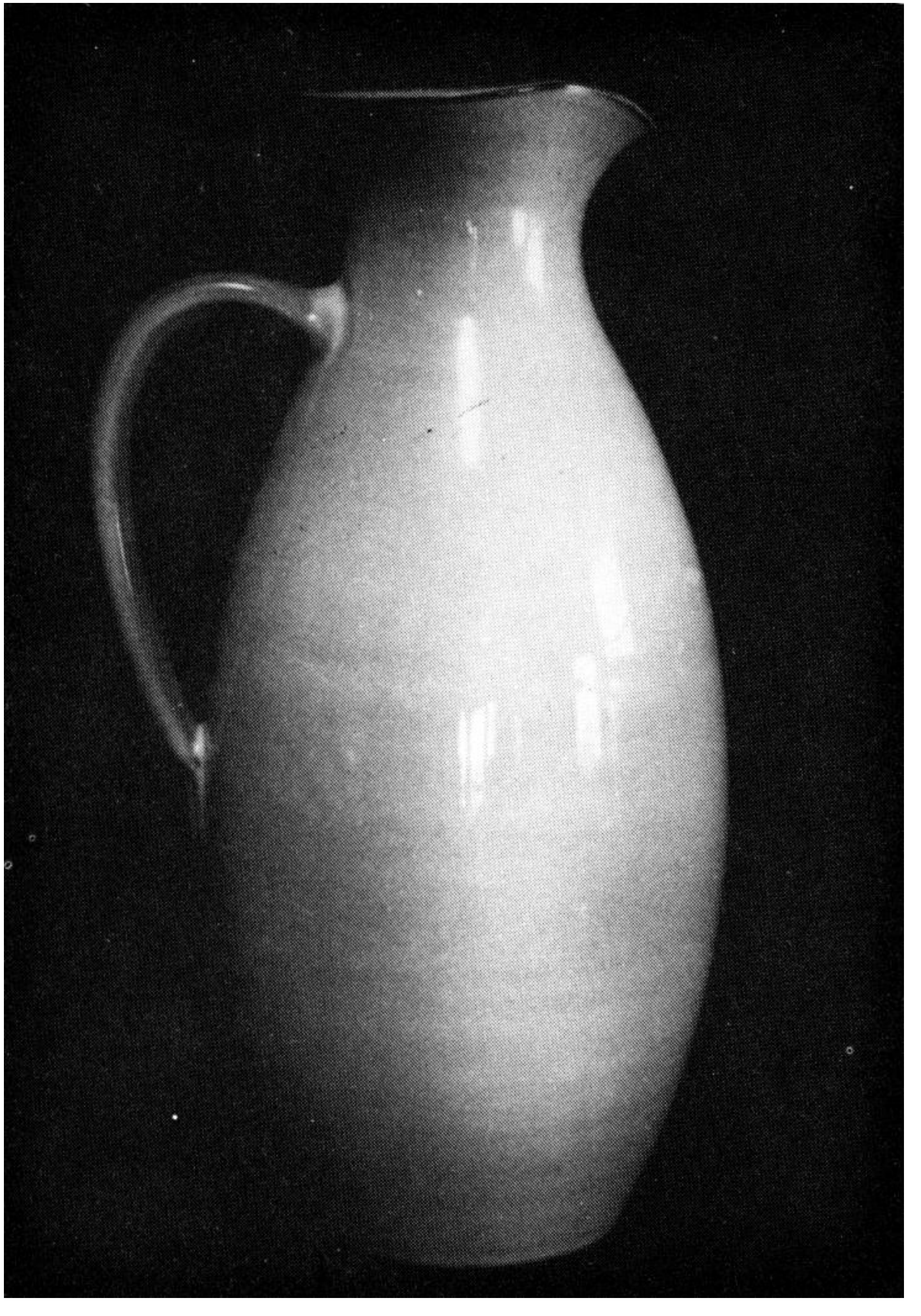
That which allows you to identify a object?

- surface structure: shape, form, roughness, etc.
- surface properties: color, texture, material, etc.
- others?

Non-locality of surface structure



Non-locality of surface structure

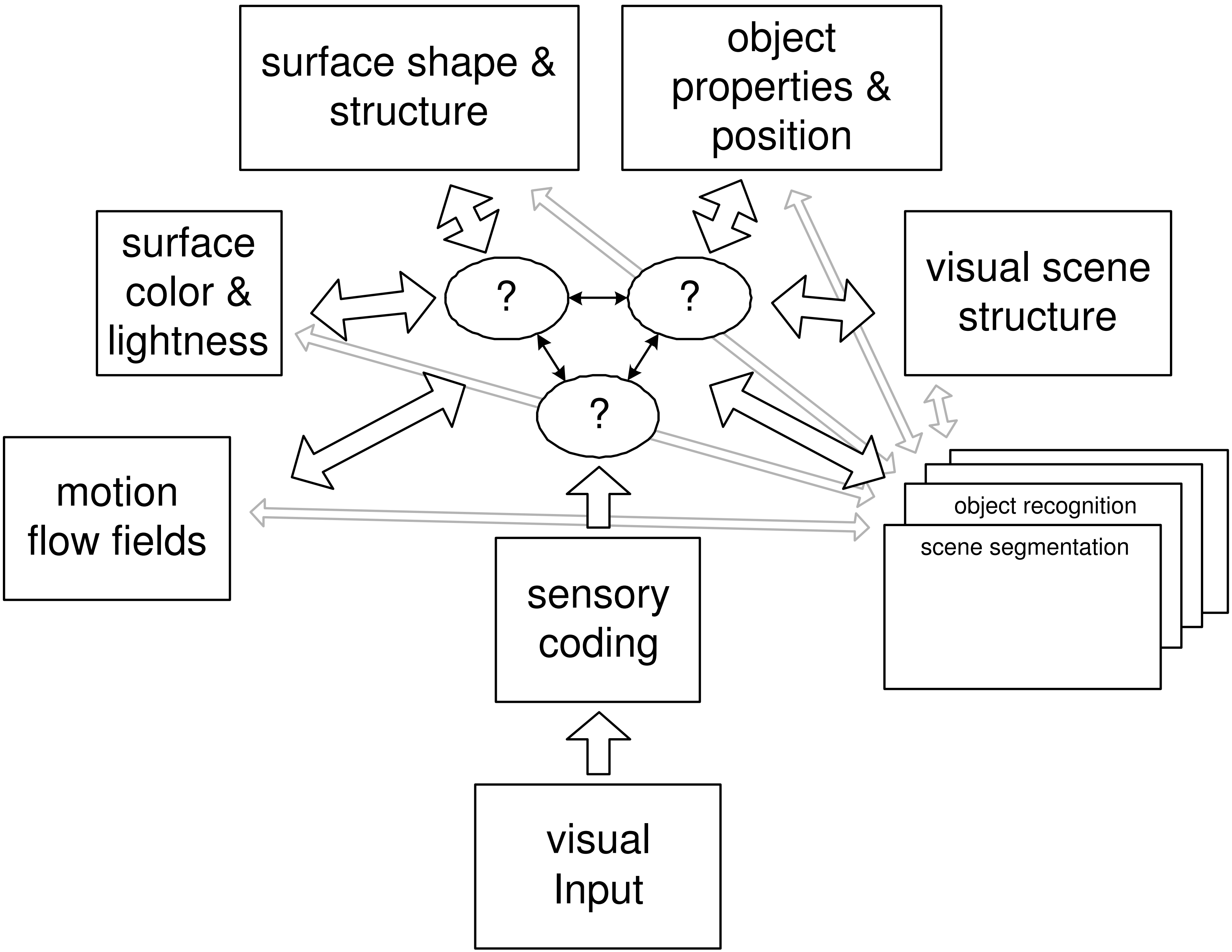


Artificial vision problems

Computation of

- features (edges, blobs, junctions, etc)
- feature classes (e.g. orientation, depth, illumination, and reflectance edges or L-, arrow-, Y-, and T- vertices)
- motion fields and optical flow
- depth maps
- lightness and color maps
- texture maps
- binocular correspondence
- image segmentation
- figure/ground organization

The problem of mid-level perception



Relevant structure depends on task

Could ask for each type of problem: When is it necessary to solve it?

You don't need to solve the lightness problem if all you want to do is avoid obstacles. How much vision does the fly do?

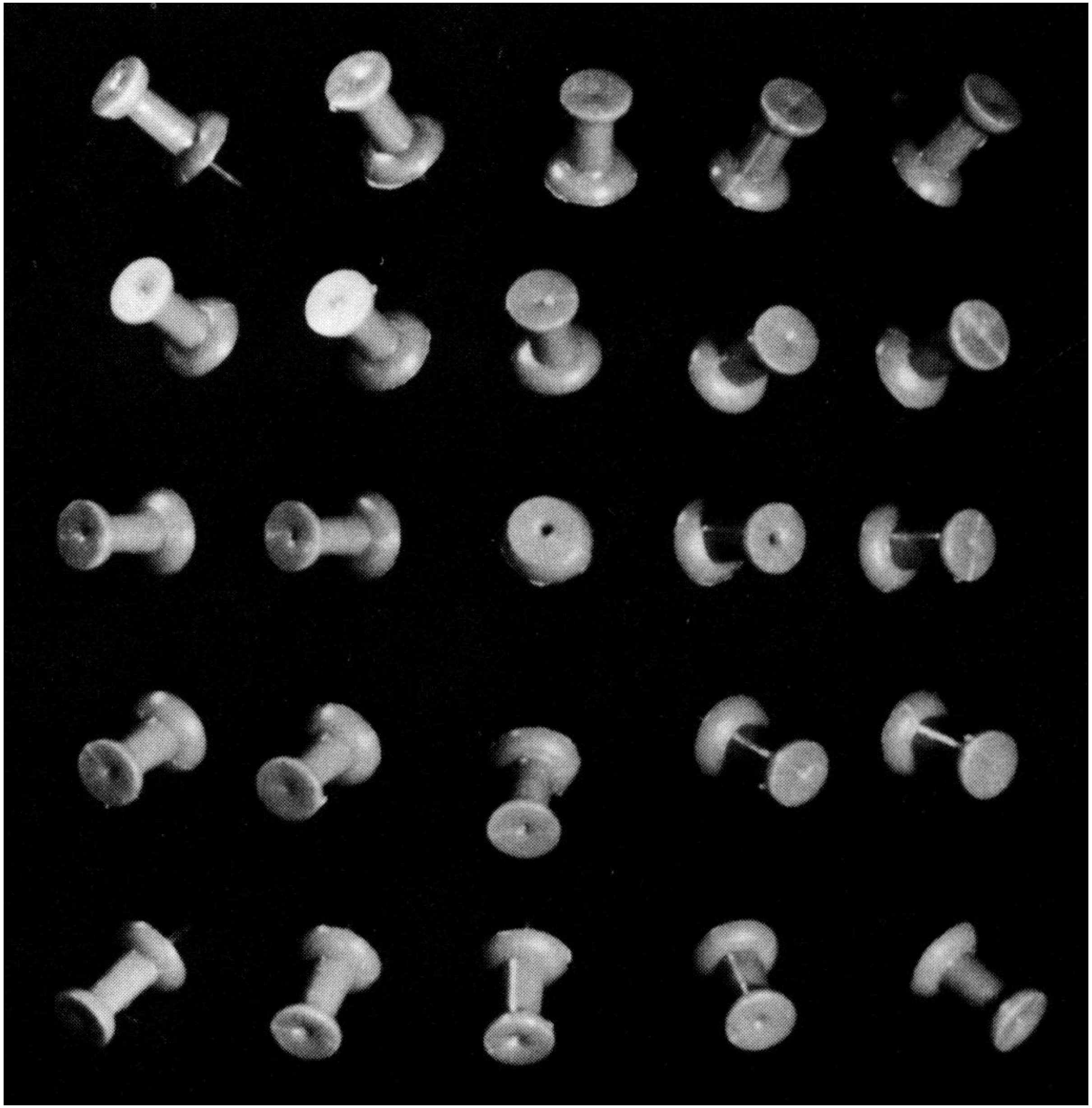
Is structure 2D or 3D?
Marr's answer was 2.5.

Computing shape

Many of the probems in computational vision involve the inference of 3D form:

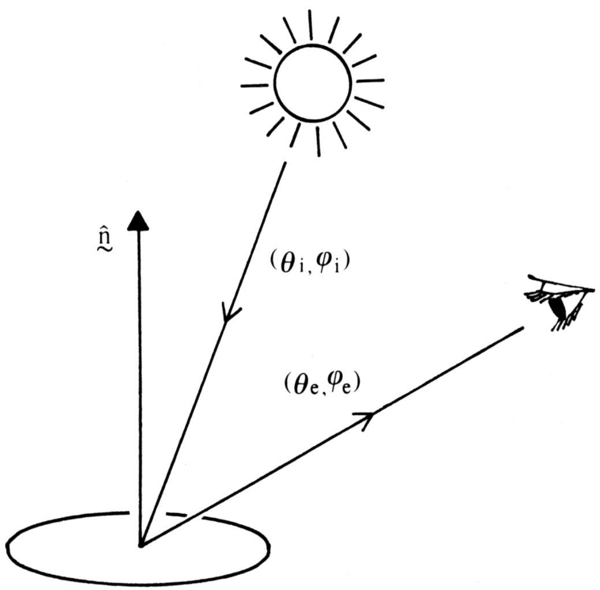
- shape from shading
- shape (structure) from motion
- shape from stereopsis

Shape from shading



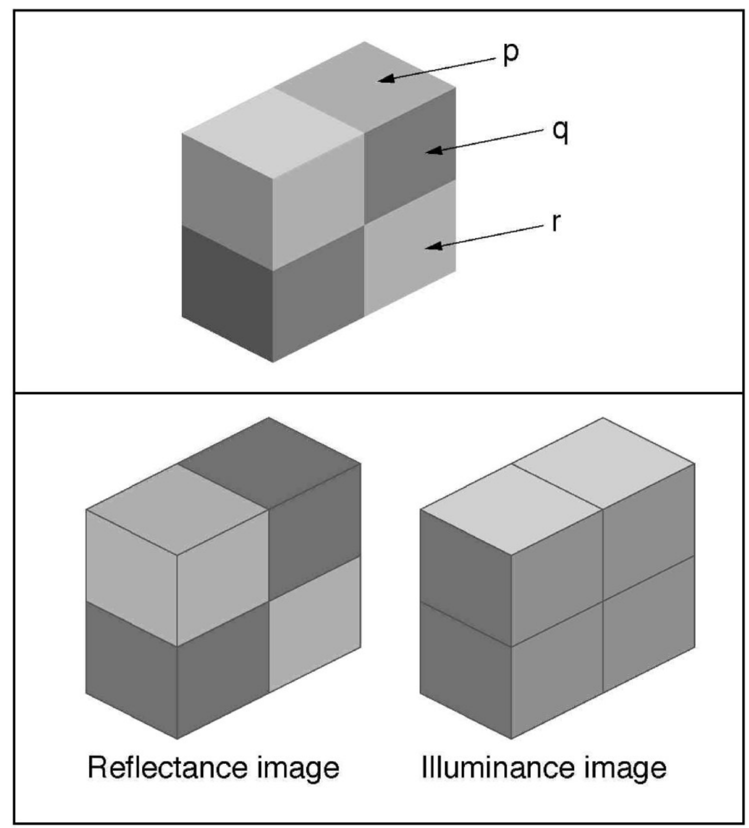
- We easily recognize these as the same shape
- But: retinal images vary greatly
- Shadows and brightness gradients provide most shape information
- How can we recover structure from intensity patterns?
- Harder problem: How do we determine similar 3D structures?

Luminance, irradiance, and reflectance



In general, the luminance image is the product of the illumance and the reflectance

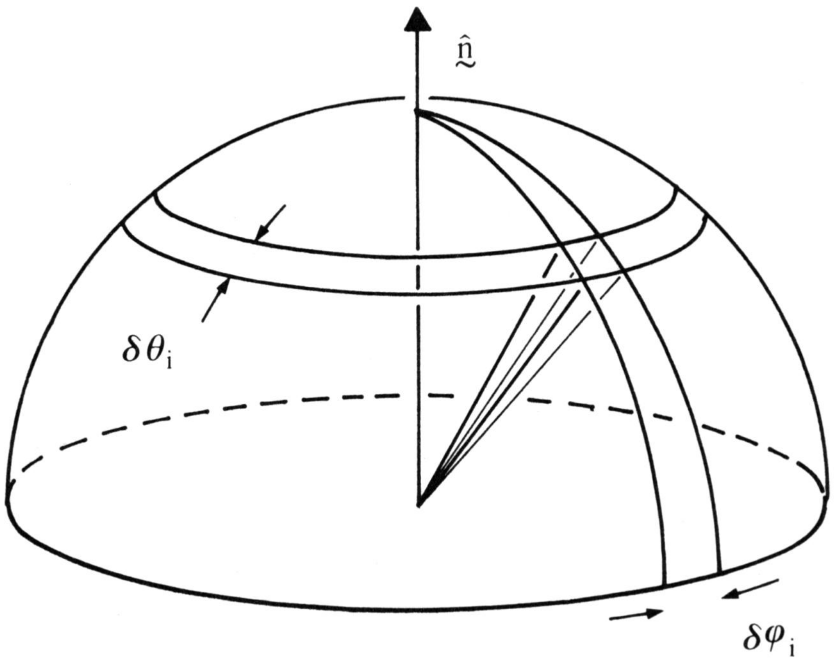
$$L(x, y, \phi_e) = E(x, y, \phi_i)R(x, y, \phi_i, \phi_e)$$



BRDF is ratio of radiance (reflected light) to irradiance (illuminant light). Using $\phi = (\theta, \varphi)$

$$R(\phi_i, \phi_e) = \frac{\delta L(\phi_e)}{\delta E(\phi_i)}$$

Non-point light sources



How to calculate radiance in the presence of extended light sources?

Intergrate product of source radiance and BRDF over all indicent directions:

$$L(\phi_e) = \int_\Omega E(\phi_i) R(\phi_i, \phi_e) \cos\theta_i d\phi_i$$

where $d\phi_i = \sin\theta_i d\theta_i d\varphi_i$.

- $\cos\theta_i$ accounts for *forshortening* of surface as seen from direction $\phi_i = (\theta_i, \varphi_i)$
- Some surfaces are complex, i.e. radiances changes with viewpoint (e.g. brushed metal).

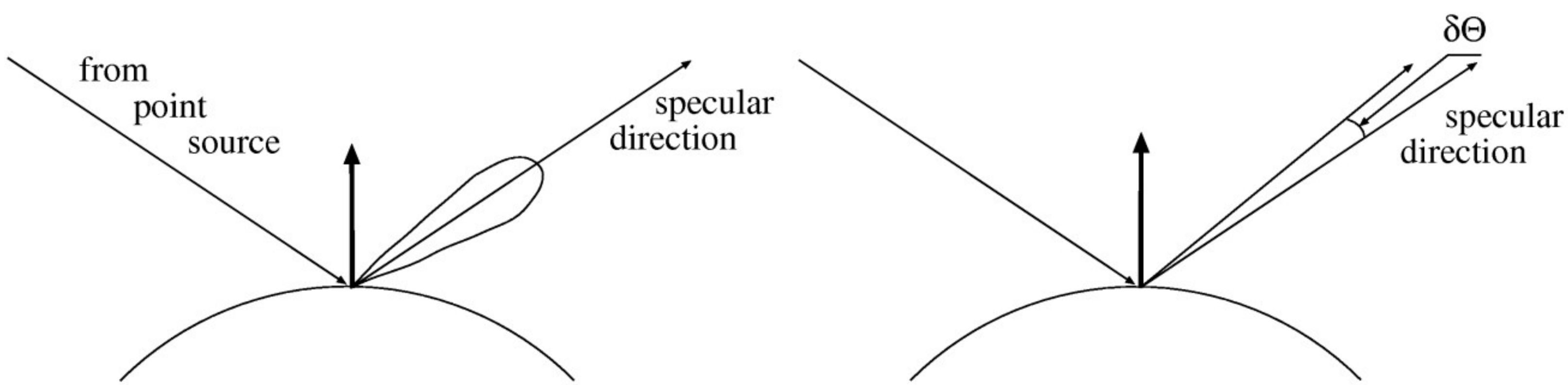
Idealized surfaces

- Lambertian: $R(\phi_i, \phi_e)$ is constant (matte surfaces).

$$L = \frac{\rho(\lambda)}{\pi} E_0 \cos\theta_i$$

$\rho(\lambda)$ is the *albedo* and describes the ratio of irradiance caused by illumination and total radiance scattered back. For colored objects, this depends on wavelength λ.

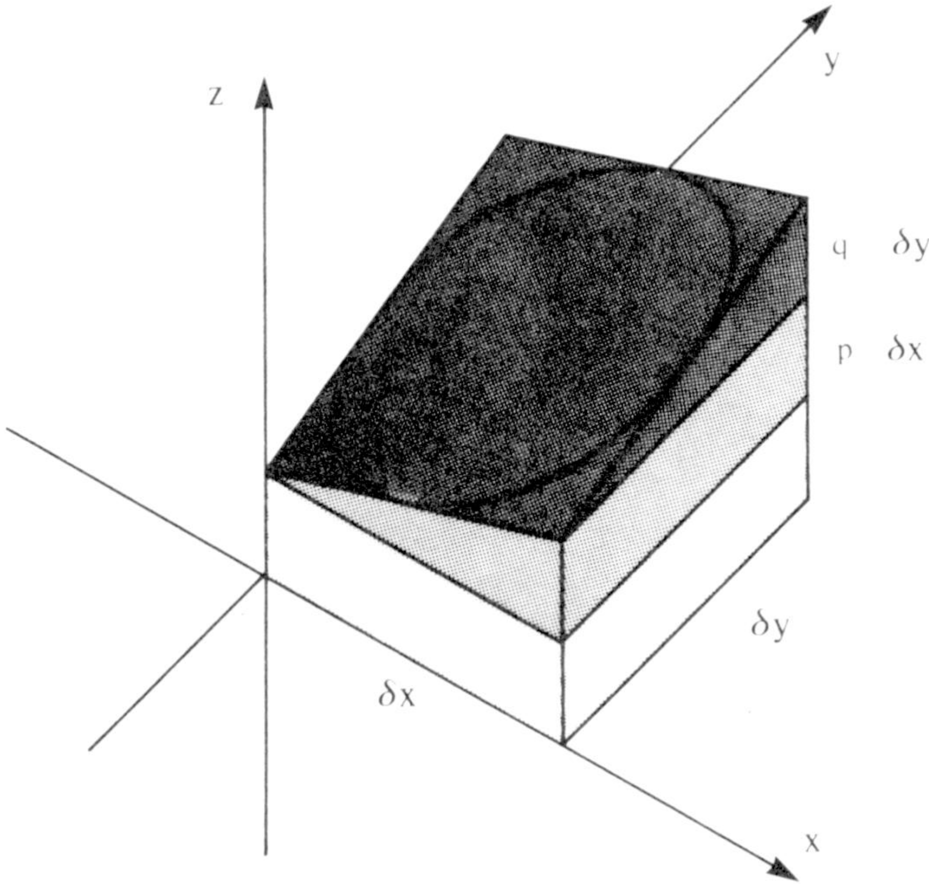
- Specular: light is reflected like a mirror.

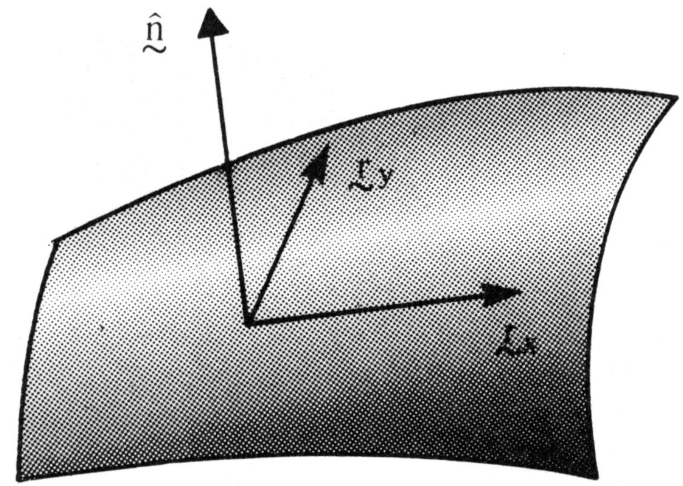


Specular direction $\delta\theta$ has finite width due to

– variable scatter from surface variations
– non-point light sources

Surface orientation





The surface normal n is the cross product of any two distinct tangents.
Take $r_x = (1, 0, p)^T$ and $r_y = (0, 1, q)^T$, then (if n points toward viewer)

$$n = r_x \times r_y = (-p, -q, 1)^T$$

Surface normals

The unit surface normal is

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

Now we can calculate angle θ_e between viewer and surface normal. Assuming viewer is at $(0, 0, 1)^T$, then taking dot product yields

$$\cos\theta_e = \frac{1}{\sqrt{1 + p^2 + q^2}}$$

This relates surface orientation to the image gradient.

Reflectance maps

We still need the light source. Assume $(-p_s, -q_s, 1)^T$ points to the light source. If we consider a Lambertian surface

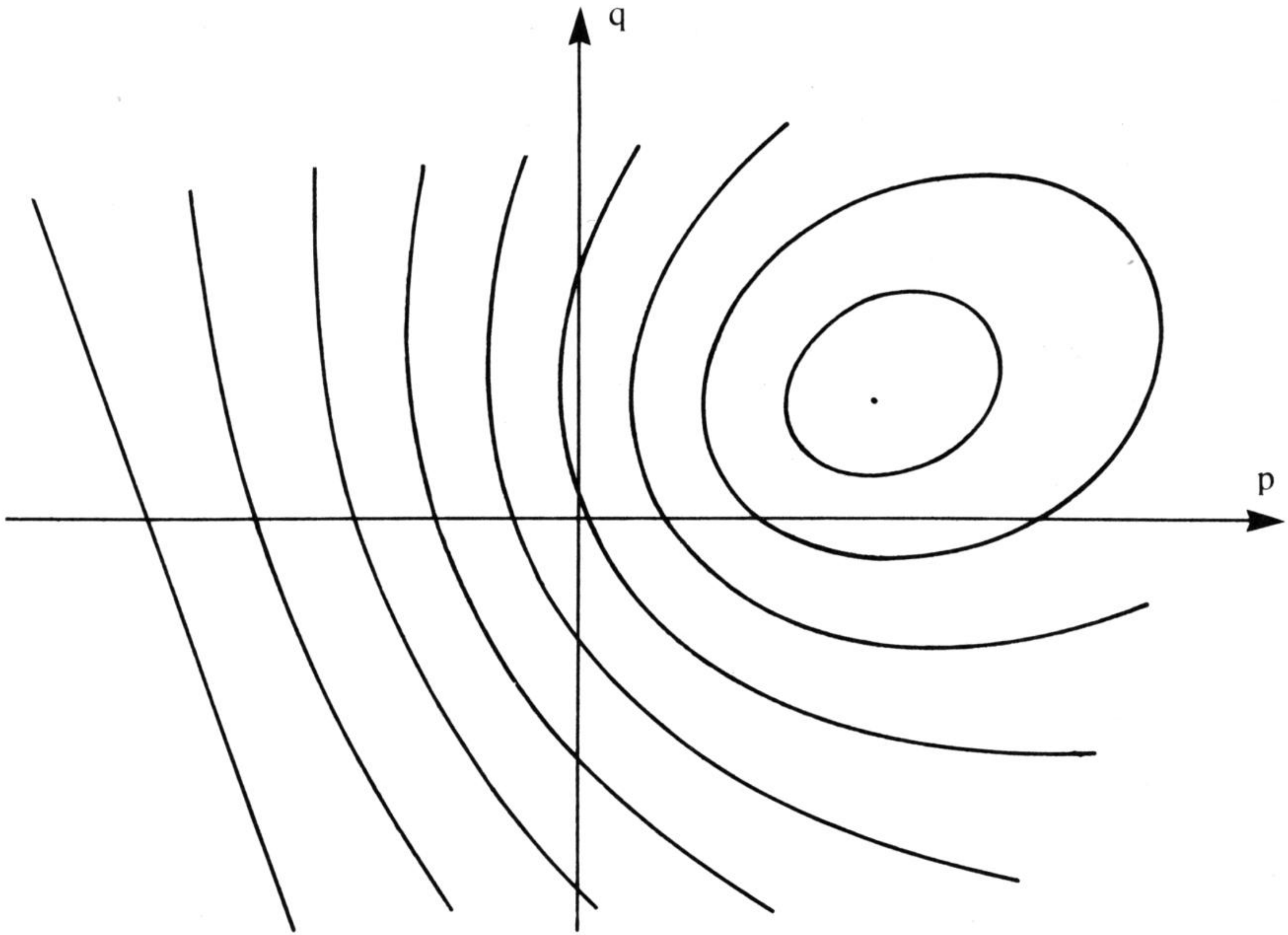
$$L = \frac{1}{\pi} E \cos\theta_i \quad \theta_i > 0$$

where θ_i is the angle between n and source, then we can compute a *reflectance map* for a known light source. Taking the dot product

$$R(p, q) = \cos\theta_i = \frac{1 + p_s p + q_s q}{\sqrt{1 + p^2 + q^2}\sqrt{1 + p_s^2 + q_s^2}}$$

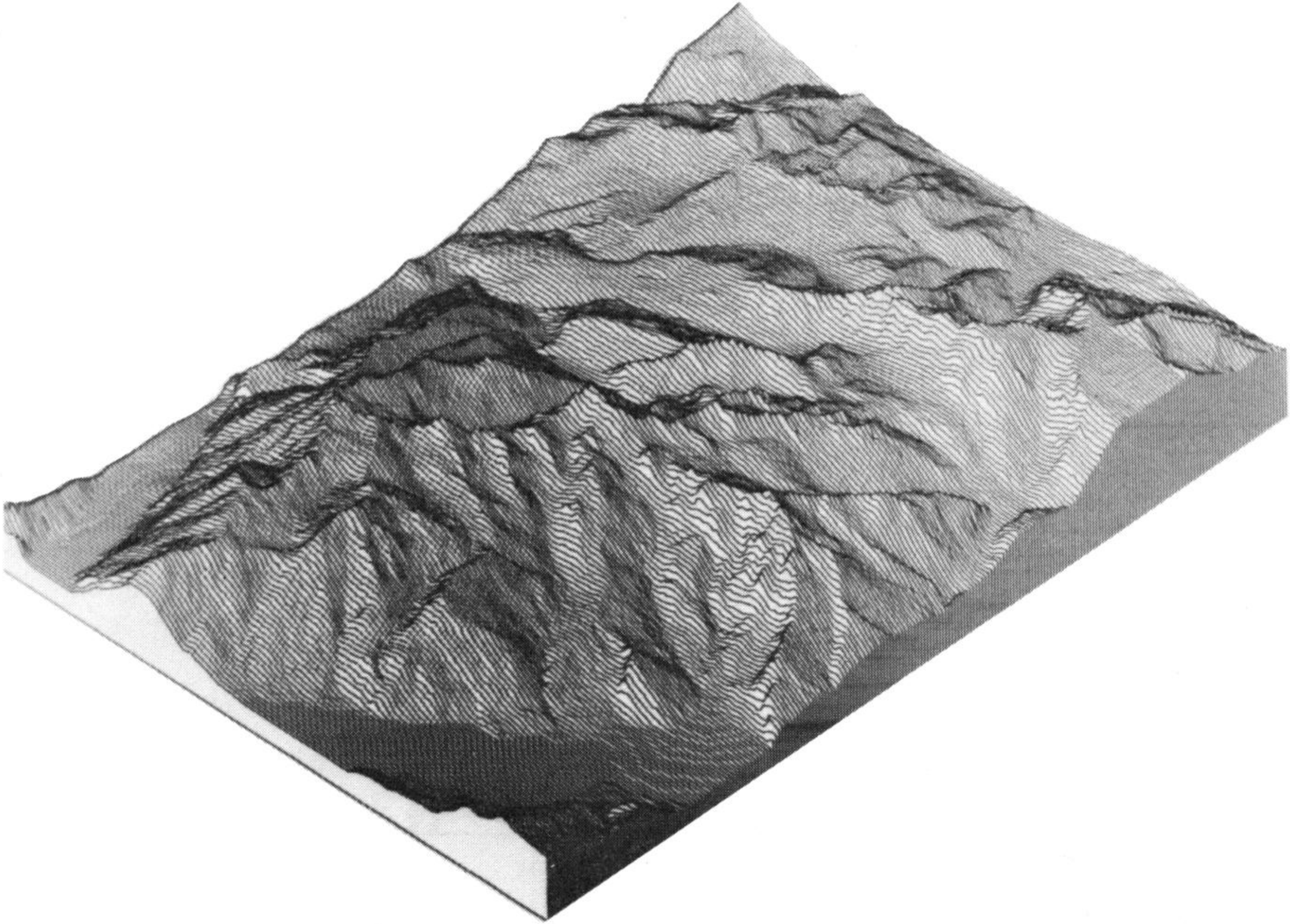
Different assumptions (e.g. specular instead of Lambertian) require different formulas.

An example reflectance map

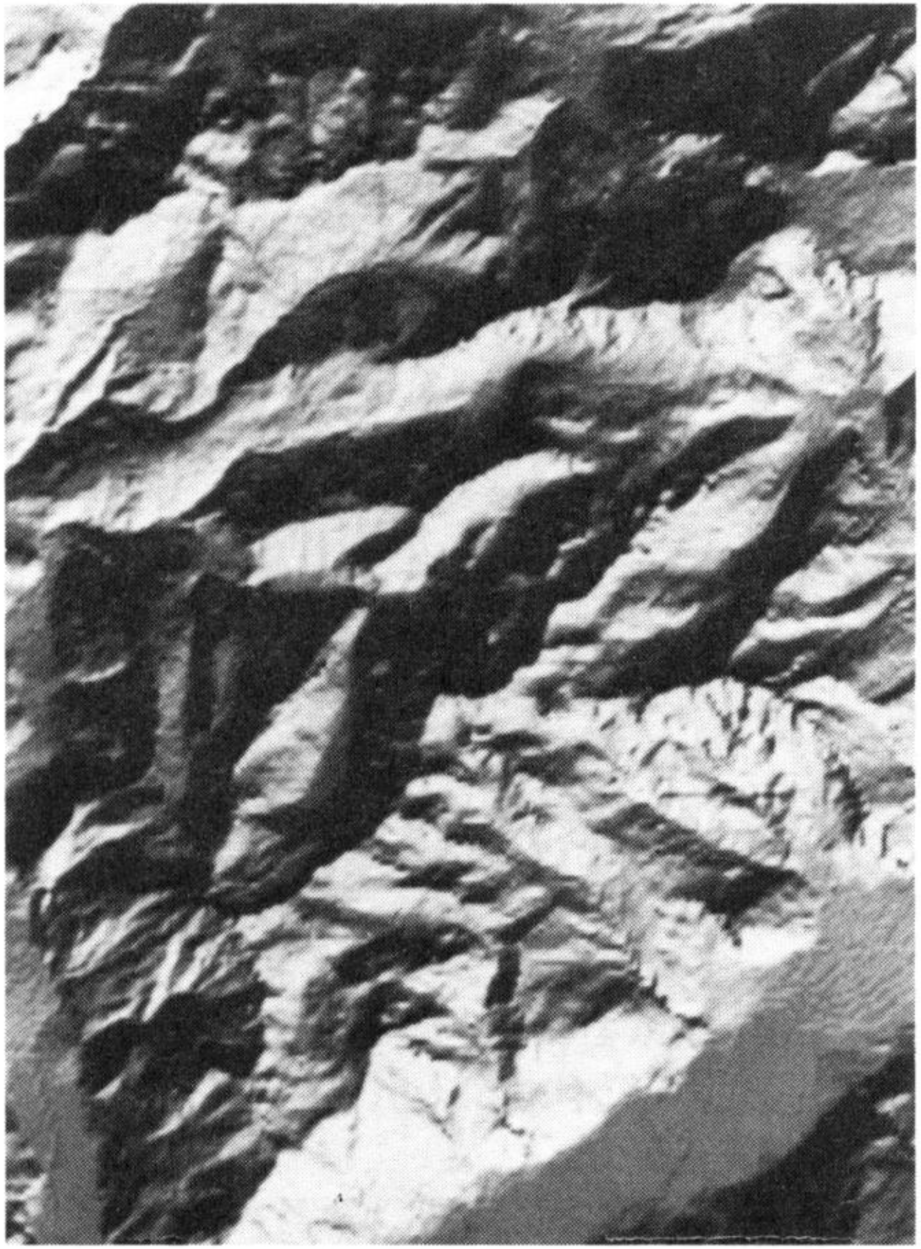


Makes explicit the relation between surface orientation and brightness.

Applying the reflectance map to a surface



Shaded reflectance map using lighting directions



View is looking down from above.

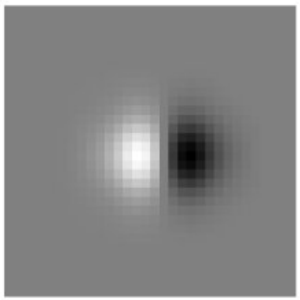
That's graphics, but what about vision?

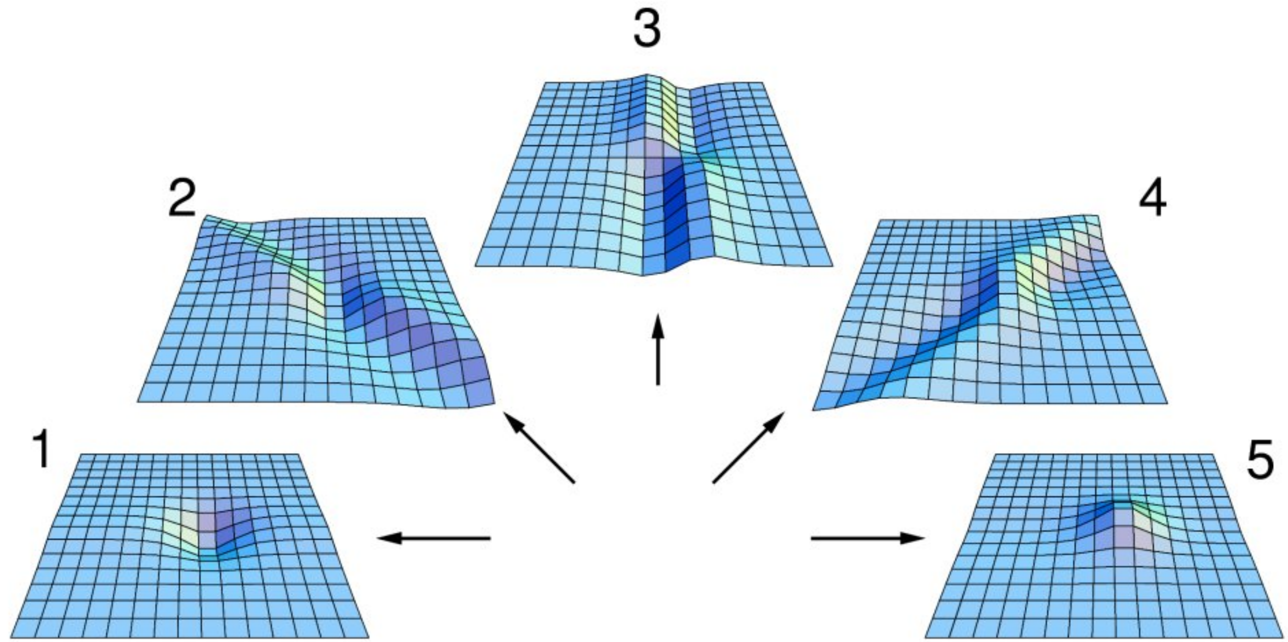
We go backwards. But there are many problems:

- How do we determine the lighting direction?
- What about multiple light sources, or ambient lighting?
- How do we separate shading from paint?
- How do we resolve ambiguity?

A simple shading pattern has many possible interpretations

Freeman: A Bayesian approach to the generic viewpoint assumption.

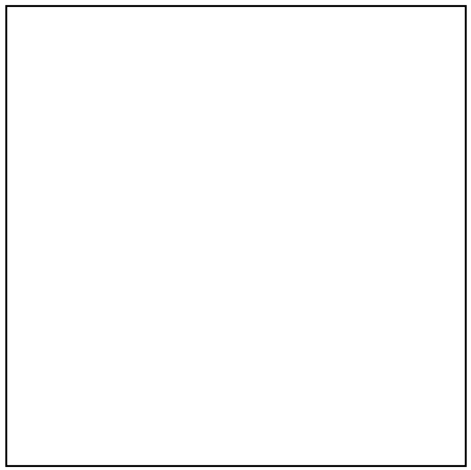




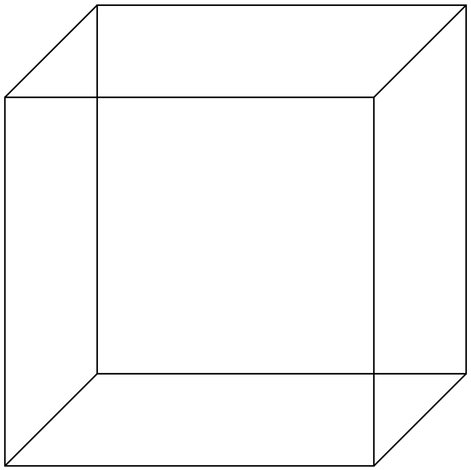
shapes for different assumed light directions

- Mathematically, many shapes and lighting directions could explain the pattern
- Perceptually, there only two.
- How does the visual system know which shape is correct?
- Shapes 2 - 4 require accidental alignment with the light source.

Generic viewpoint assumption

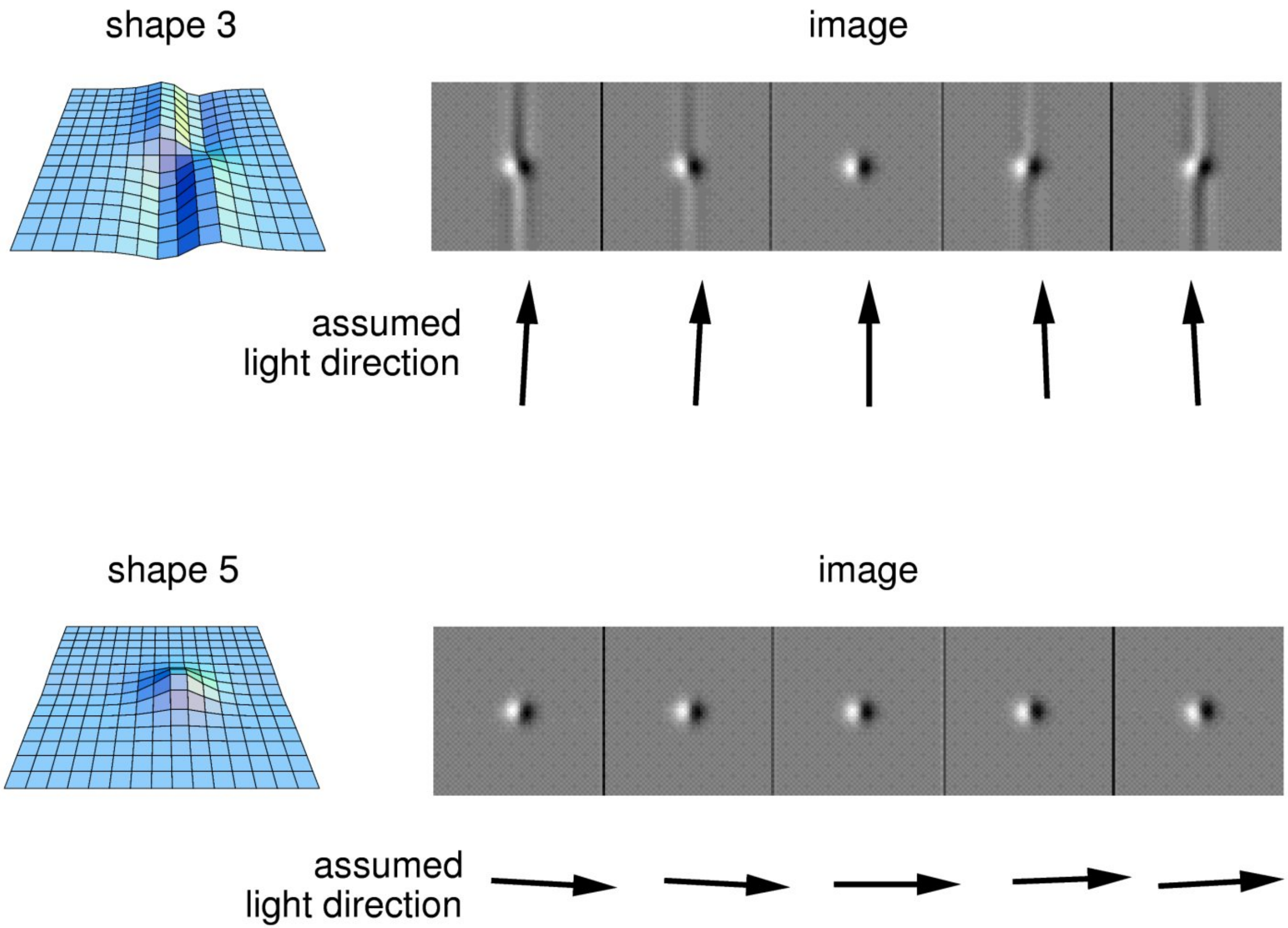


Interpreting this as cube requires assuming an accidental view.



The *generic viewpoint assumption* assumes a scene is not viewed from a special position.

Change in shading with respect to image



- shape 3: only a small range of directions are consistent with image
- shape 5: much larger range of directions give same image
- Equally likely light directions $\Rightarrow$ shape 5 more probable than shape 3

Freeman: Exploiting the Generic Viewpoint Assumption

Main points:

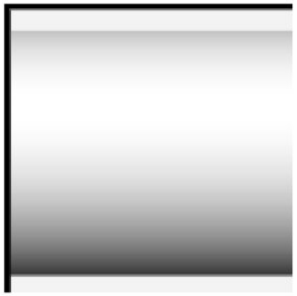
- quantify generic view probabilities in a general case
- generic variable can be general, e.g. object orientation or lighting position
- generic view assumption can strongly influence scene interpretations

Key is to quantify how visual data would change if the data were to change: large changes correspond to improbable scenes

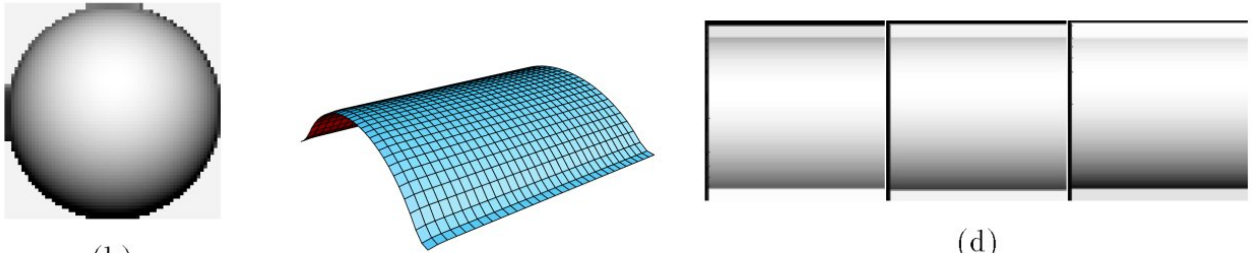
Scene probability equation

- Fidelity: how well rendering model can describe data
- Prior probability: how parameters fit prior expectations
- Genericity: favors interpretations for which the image is stable w.r.t changes in generic variables

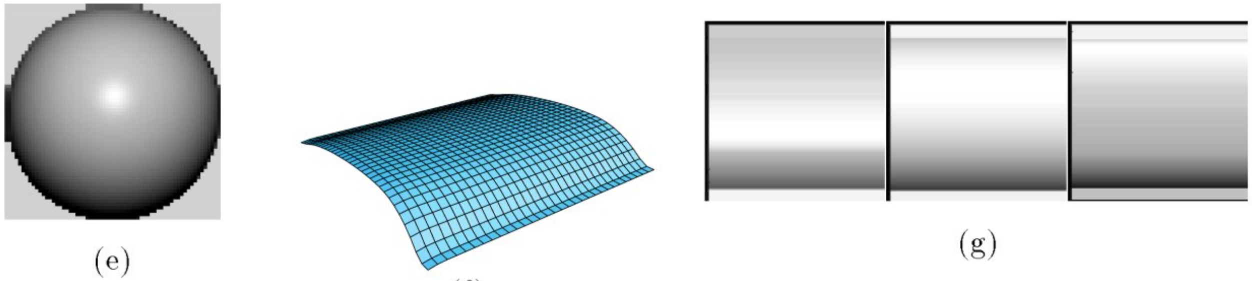
Different reflectances and shapes produce same image



(a)



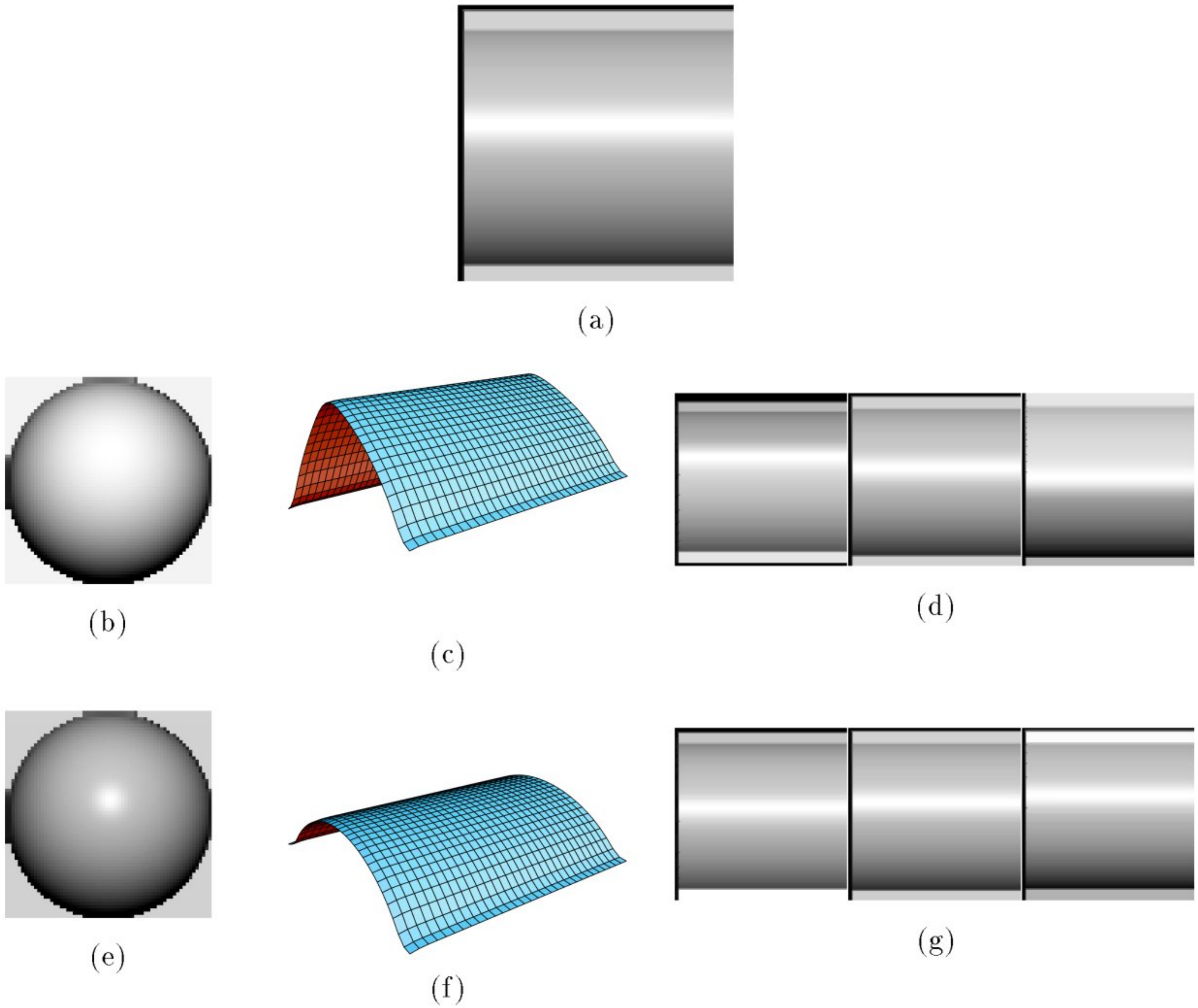
(b) (c) (d)



(e) (f) (g)

- Small Rotations can distinguish between possibilities.
- If all viewpoints, shapes, and reflectances are equally likely
  $\Rightarrow$ (b-c) more likely than (e-g)

Roles reversed: Lambertian surface changes more than shiny



If all object orientations are equally likely, then
probability of object $\propto$ range of angles that yield "similar" images

The scene probability equation

The image y is modeled by the "rendering function" plus noise:

$$y = f(\beta, x) + n$$

- β: vector of scene parameters we want to estimate, e.g. the object shape and reflectance function
- x: vector of generic variables we don't want to estimate, e.g. viewpoint, object orientation, lighting position
- $n \sim$ Gaussian $\Rightarrow$ image *likelihood* is Gaussian

$$P(y|\beta, x) \sim \mathcal{N}(y - f(\beta, x), \sigma)$$

Goal is to derive $P(\beta|y)$, i.e. the probability of the scene parameters given the image, independent of the generic variables.

Question:

Why do we want $P(\beta|y)$ and not $P(\beta|x, y)$ or $P(\beta, x|y)$?

Deriving $P(\beta|y)$

Use Bayes rule:

$$P(\beta, x|y) = \frac{P(y|\beta, x)P(\beta)P(x)}{P(y)}$$

This gives joint probability of β and x given y.

$P(\beta|y)$ is obtained by *marginalization*

$$P(a) = \int P(a, b)da$$

In the case of the image model

$$P(\beta|y) = \frac{P(\beta)}{P(y)} \int P(y|\beta, x)P(x)dx$$

How do we evaluate the integral?

Approximating the integral $\int P(y|\beta, x)P(x)dx$

The image likelihood, $P(y|\beta, x)$, has the form

$$P(y|\beta, x) = \frac{1}{Z_\sigma} \exp\left[\frac{-|y - f(\beta, x)|^2}{2\sigma^2}\right]$$

Cannot solve analytically because of the form of f, so approximate:

- Taylor expand around peak of $P(y|\beta, x)$,
  i.e. around value of x_0 that minimizes $|y - f(\beta, x)|^2$.
- This is called *Laplace's approximation*
- It approximates the posterior volume with a Gaussian.

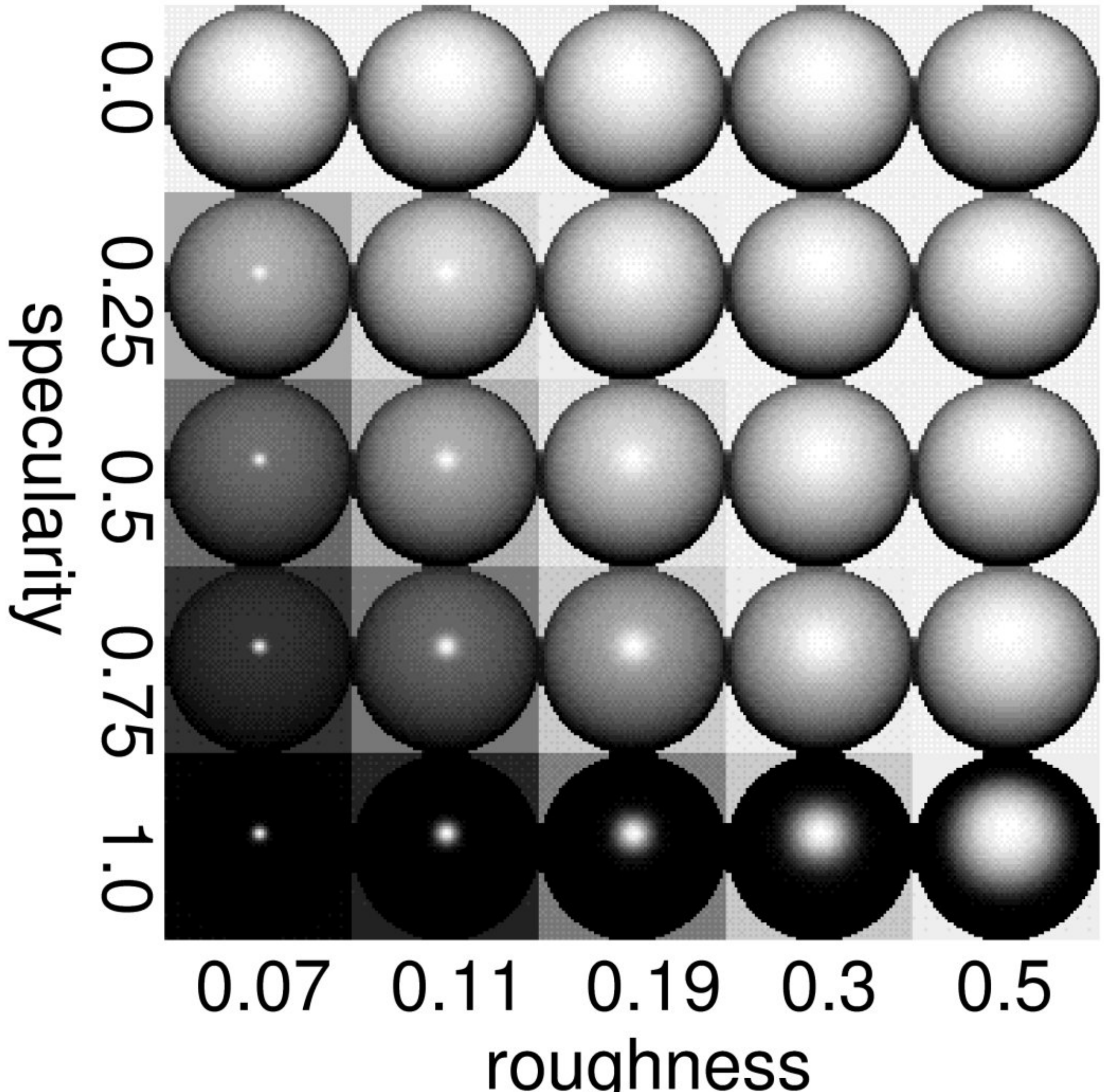
The posterior probability is then given by

$$\begin{aligned} P(\beta|y) &\propto \exp\left[\frac{-|y - f(\beta, x_0)|^2}{2\sigma^2}\right] P(\beta)P(x_0)\frac{1}{\sqrt{\det(A)}} \\ &\propto \quad \text{“fidelity”} \quad \text{prior} \quad \text{“genericity”} \end{aligned}$$

Specifying the model

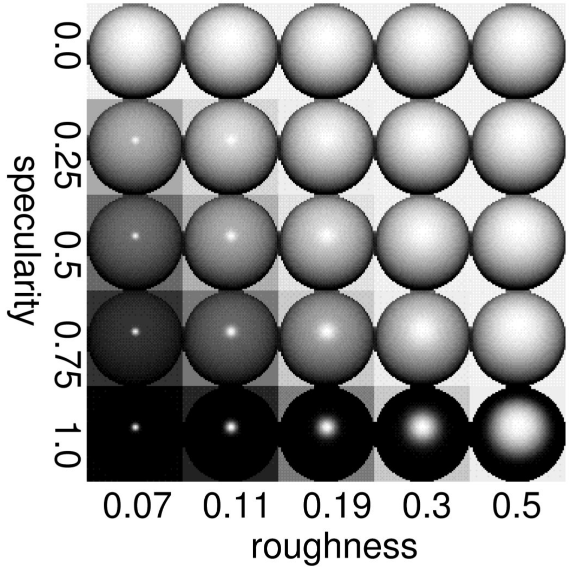
- Still need to specify $f(\beta, x)$
- Use simple models based on rendering equations

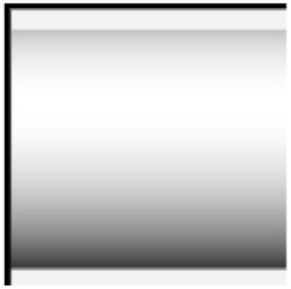
Parameterization of the reflectance function



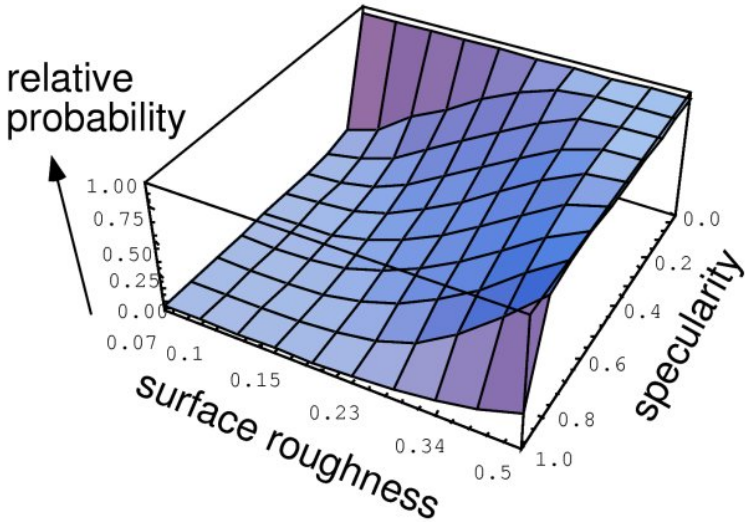
- *surface roughness* governs width of specular highlight
- *specularity* determins ratio of diffuse and specular reflections

Relative probabilities of surface properties

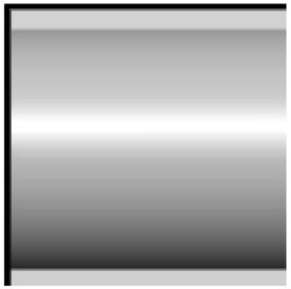




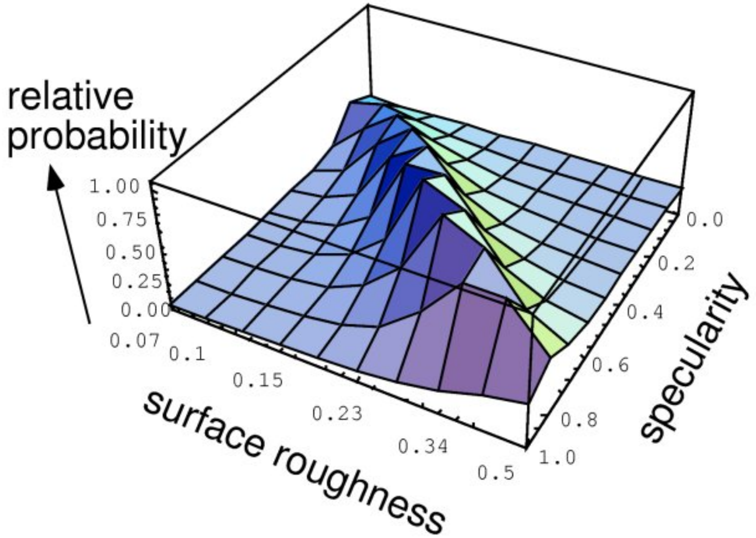
(a)



(b)



(c)



(d)

Generic light direction

Linear shading (Pentland, 1990): assume that image intensities are approximately proportional to surface slopes p and q

$$I = k_1 p + k_2 q$$

Related facts:

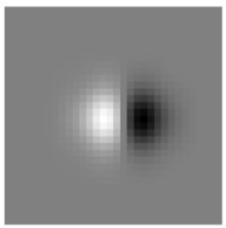
- direction of light is $\tan^{-1}(k_1, k_2)$
- product of lighting strength and surface reflectance is $\sqrt{k_1^2 + k_2^2}$
- surface slopes scale inversely with $\sqrt{k_1^2 + k_2^2}$

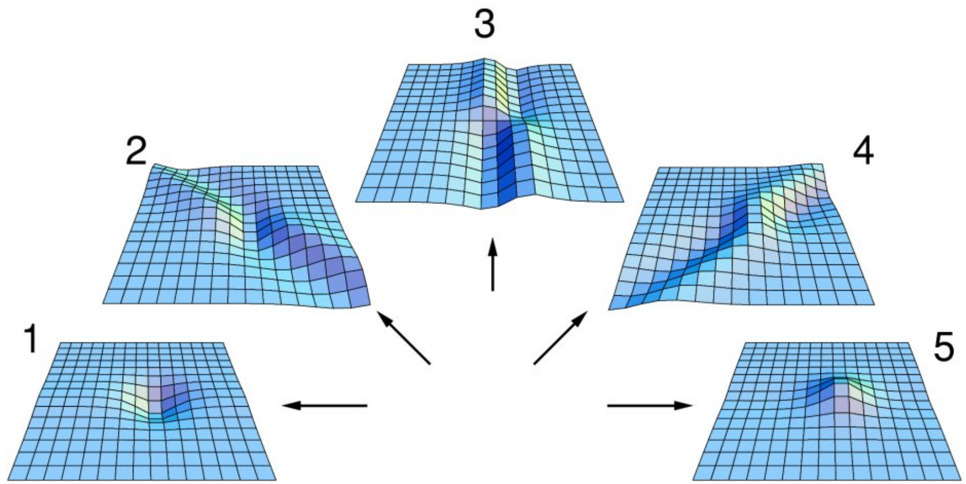
Any assumed lighting direction and strength can explain an image by assuming a different shape.

Which shape and lighting parameters are best?

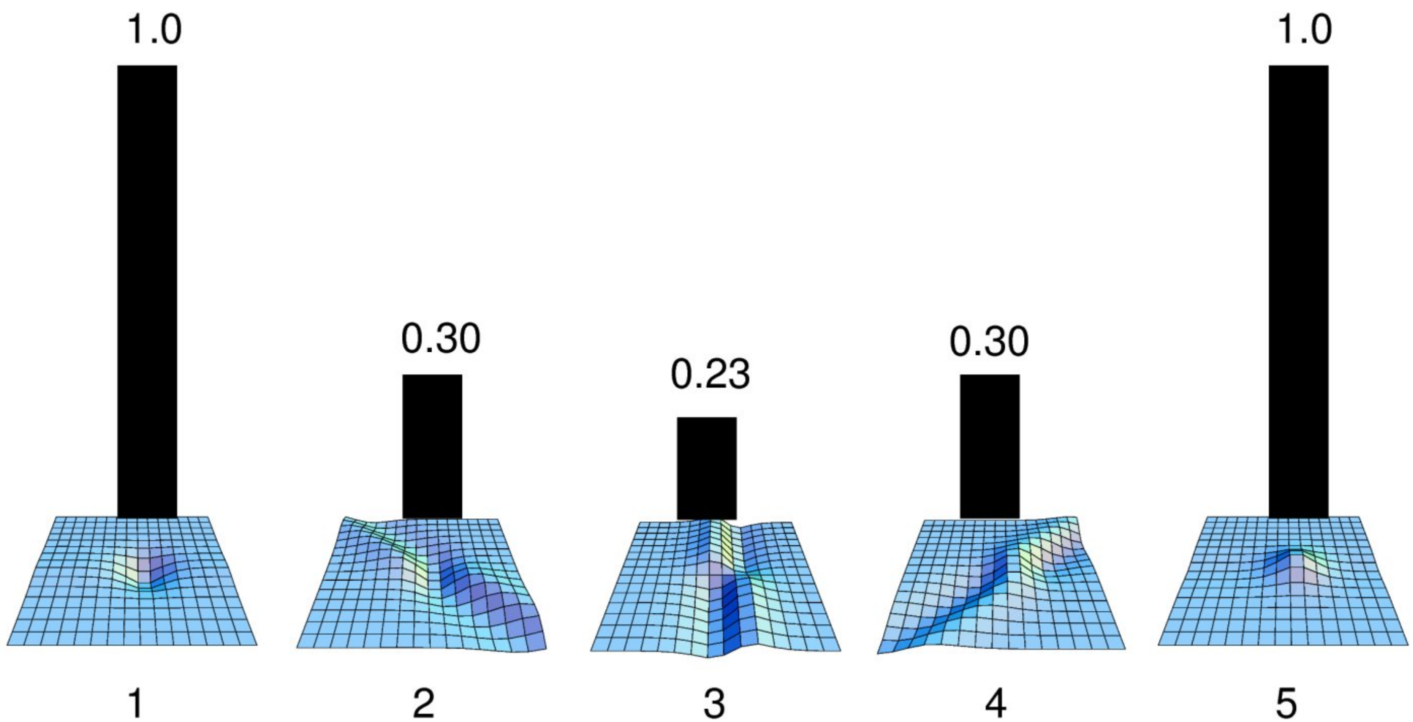
- Don't want to assume aribitrarily that some shapes are more likely.
  $\rightarrow$ Derive scene probability equation for linear shading model.

Relative probabilities of the original bump image



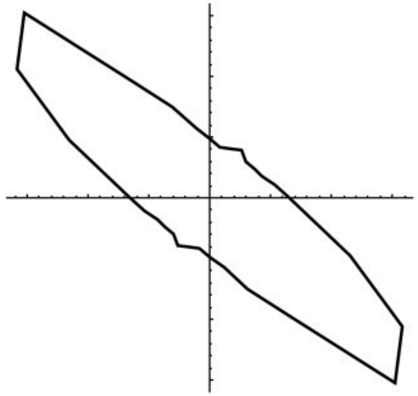


shapes for different assumed light directions



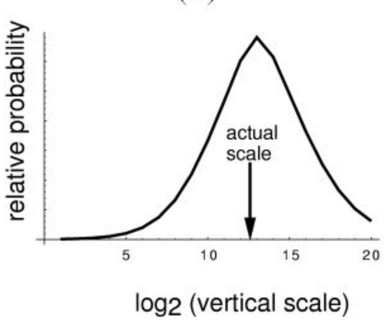
Proability of shape vs lighting direction

(a)



Polar plot of shape probability
as function of
assumed light direction

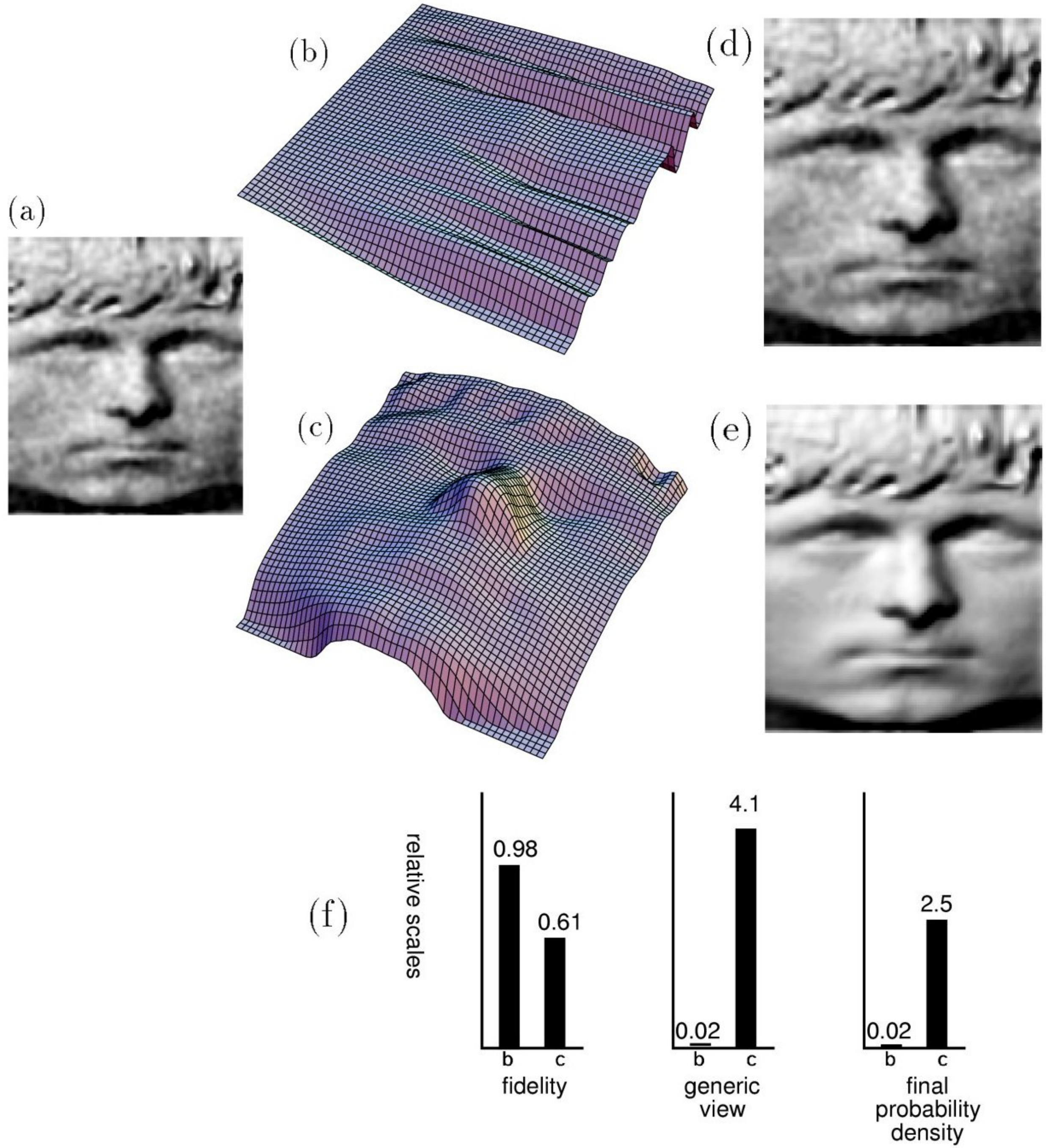
(b)



(c)

- All shapes are equally probable a priori
- Shapes reconstructed from correct lighting direction are more probable than other directions
- can also infer correct vertical scaling and light strength

The effect of the genericity term



- image with 7dB Gaussian noise added (a)
- image has two explanations: lighting from left (b) and lighting from top (c)
- side lighting better explains image, i.e. has higher fidelity (f)
- but light source must be very percisely positioned, so genericity of b is much lower than c (f)
- final probability favors c even though b is a better fit to image data

Performance of Shape from Shading Algorithms

Shape from Shading: A Survey

Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah

Abstract—Since the first shape-from-shading (SFS) technique was developed by Horn in the early 1970s, many different approaches have emerged. In this paper, six well-known SFS algorithms are implemented and compared. The performance of the algorithms was analyzed on synthetic images using mean and standard deviation of depth (Z) error, mean of surface gradient (p, q) error, and CPU timing. Each algorithm works well for certain images, but performs poorly for others. In general, minimization approaches are more robust, while the other approaches are faster. The implementation of these algorithms in C and images used in this paper are available by anonymous ftp under the pub/tech_paper/survey directory at eustis.cs.ucf.edu (132.170.108.42). These are also part of the electronic version of paper.

– Summary –

To analyze the accuracy, the output for the synthetic images was compared with the true surface shapes and the results of comparison were shown in the forms of the average depth error, the average gradient error, and the standard deviation of depth error. The output for real images was only analyzed and compared visually. The conclusions drawn from the analysis are as follows:

1. All the SFS algorithms produce generally poor results when given synthetic data,
2. Results are even worse on real images, and
3. Results on synthetic data are not generally predictive of results on real data.

Lambertian reflectance model

- brightness proportional to energy of incident light
- amount of light energy on surface $\propto$ surface area as seen from light source position (foreshortening)
- Brightness of Lambertian surface is product of strengh of light source, A, the albedo of surface, ρ, and foreshortened area, $\cos\theta_i$

$$I_L \equiv R = A\rho\cos\theta_i$$

where R is reflectance map, θ_i is angle between surface normal $\vec{N}$ and source direction $\vec{S}$. For unit normals, we can write

$$I_L = A\rho\vec{N} \cdot \vec{S}$$

Specular reflectance model

- Specularity only occurs when incident angle of light source equals reflected angle

- simplest model for the brightness of the specular reflection is

$$I_S = B\delta(\theta_s - 2\theta_r)$$

  where B is the strength of the specular component, θ_s is the angle between the light source and the viewing direction, and θ_r is the angle between the surface normal and the viewing direction.

- this assumes a point reflection, which is not realistic. There are many ways to make this more realistic, but a simple one is

$$I_S = K \exp^{-(\frac{\alpha}{m})^2}$$

  where K is a constant, α is the angle between the surface normal $\vec{N}$ and the bisector $\vec{H}$ of the viewing direction and source direction.

Non-ideal surfaces

- Most surfaces are not idea.

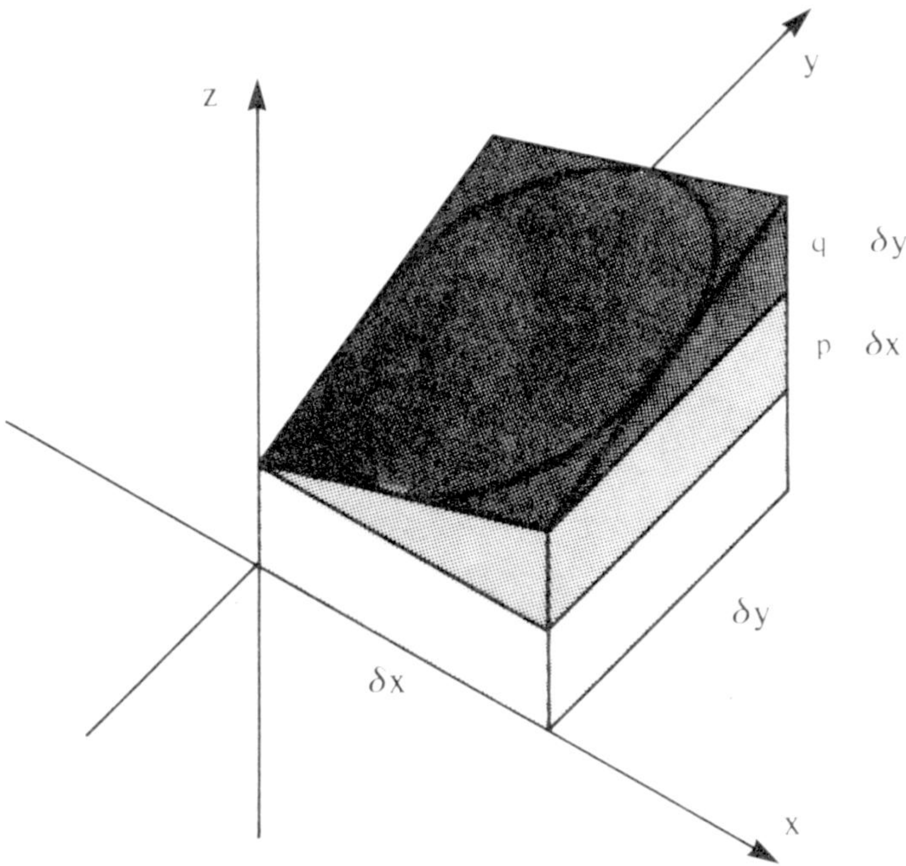
- can be approximated by hybrid surfaces:

$$I = (1 - \omega) I_L + \omega I_s$$

- many more complex models exist

- Lambertian model is usually a poor approximation due to surfaces roughness

Inferring surface orientation



- p and q are the surface gradients along the x and y direction.
- general problem is to infer p and q (or equivalently the surface normal $\vec{N} = (-p, -q, 1)^T$) from the observed image intensities
- some algorithms assume knowledge of lighting sources and viewing angles, more sophisticated algorithms estimate these from the image

Minimization approaches

- describe constraints that are consistent with reflection equations and prior knowledge

- find solution (i.e. depth and surface orientation) over whole image that minimizes constraint equation

brightness constraint: minimize total brightness error of reconstructed image

$$\int\int (I - R)^2 \, dx \, dy$$

Intensity gradient constraint: minize error between reconstructed intensity gradient and image intensity gradient in both x and y directions

$$\int\int (R_x - I_x)^2 + (R_y - I_y)^2 \, dx \, dy$$

Other constraints

Solution of brightness constraint is *ill-posed*, so assume smooth surface

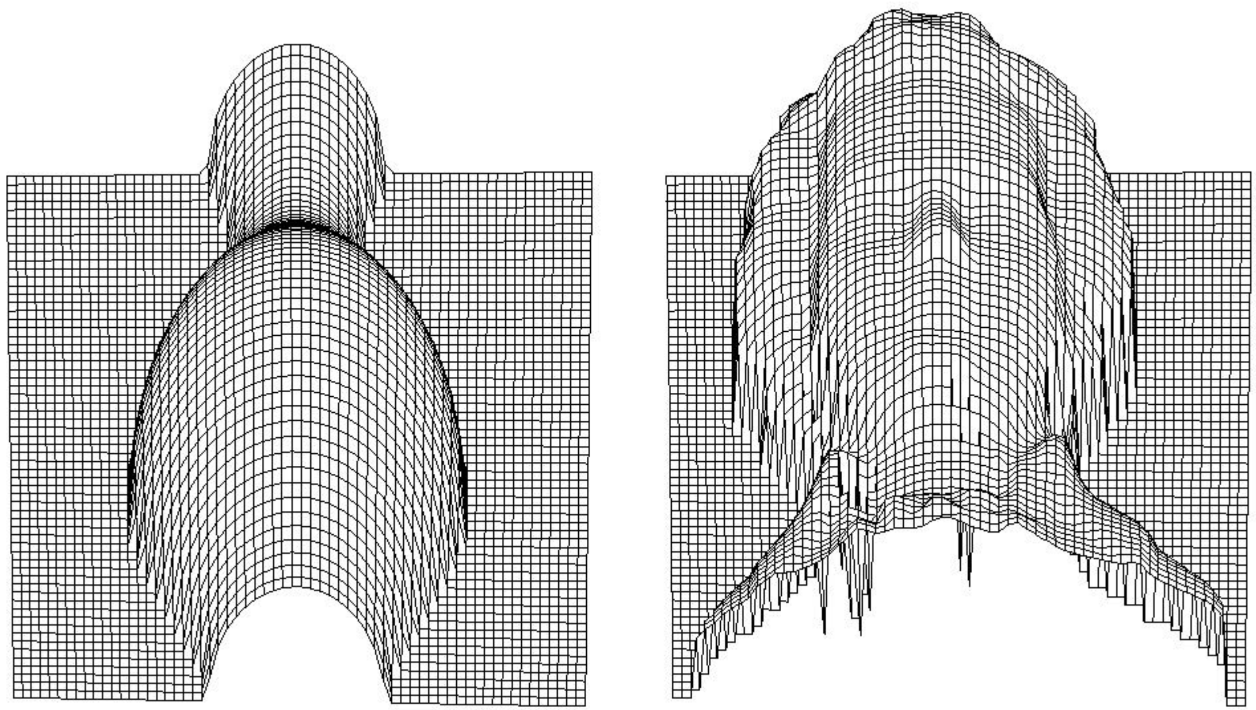
smoothness constraint: bias toward smooth surfaces to obtain unique solution

$$\int\int (p_x^2 + p_y^2 + q_x^2 + q_y^2)\, dx\, dy$$

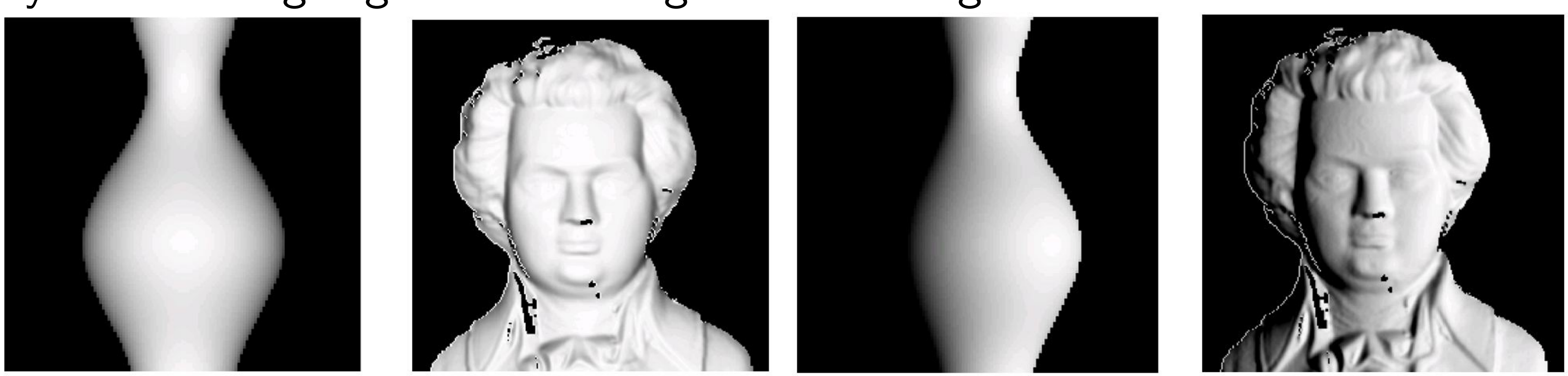
Integrability constraint: ensure that the surface height $Z(x, y)$ obtained by integrating over surface slopes is independent of the path of integration, i.e. $Z_{x,y} = Z_{y,x}$. This can be written

$$\int\int ((Z_x - p)^2 + (Z_y - q)^2\, dx\, dy$$

Depth maps of synthetic images



Synthetic images generated using 2 different light sources:



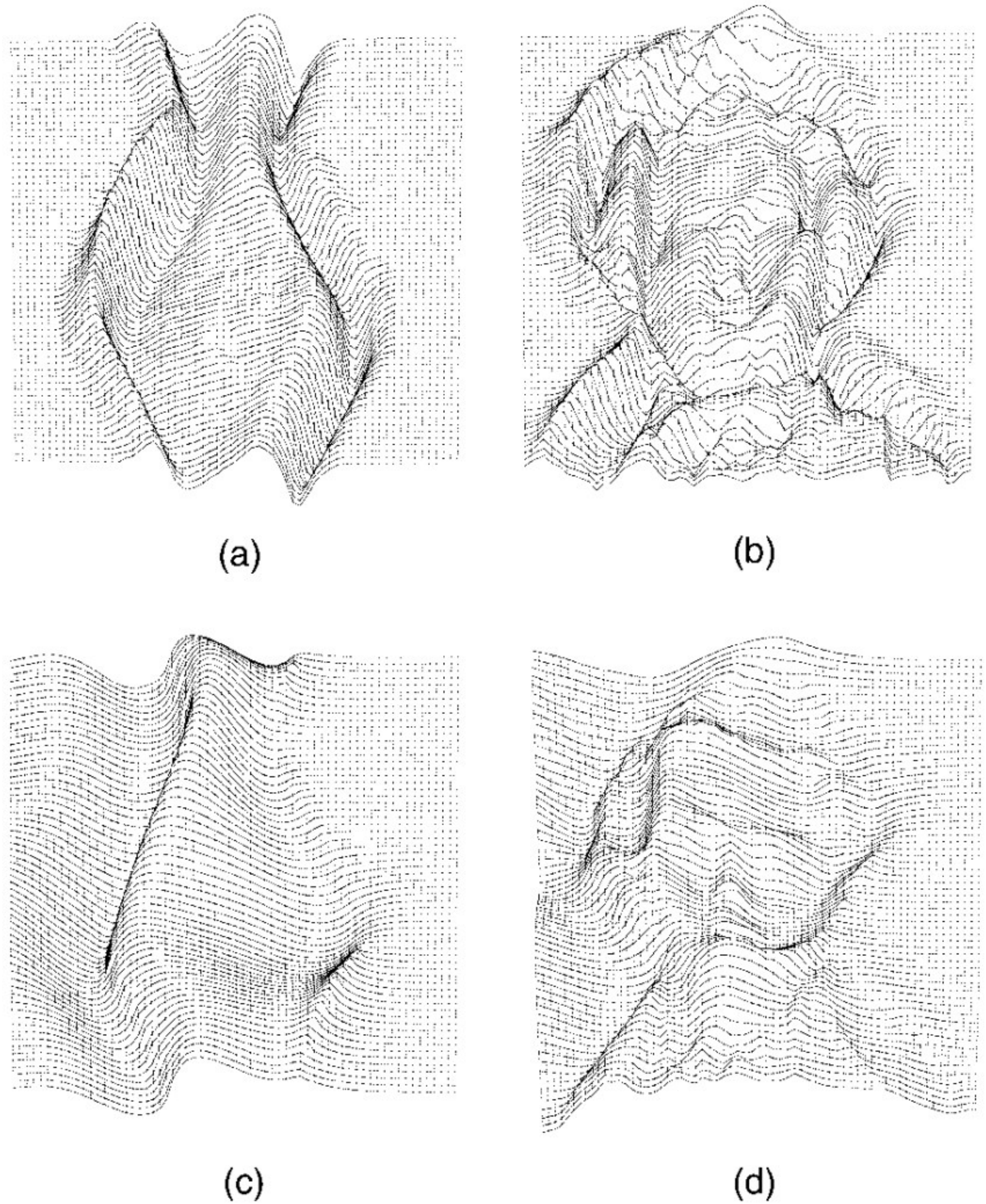
Example minimization algorithm: Zheng and Chellappa

Miminize the constraint:

$$\int\int (I - R)^2 + (R_x - I_x)^2 + (R_y - I_y)^2 + \mu((Z_x - p)^2 + (Z_y - q)^2)\, dx\, dy$$

- this combines brightness, intensity gradient, and integrability constraints
- minimized iteratively by optimizing depth and gradients
- use hierarchical (pyramid) structure to speed up computation
- $\mu = 1$

Results of minimization algorithm



(a,b) results with light source 1. (c,d) light source 2.