Computer Vision, Lecture 7

Professor Hager
http://www.cs.jhu.edu/~hager

Hough Transform Idea

- Each edge point in an image is a constraint on line parameters:
  - constraint is a line
  - each unique point adds another constraint
- Algorithm:
  - Initialize a 2-D array of counters to zero.
  - For each edge point (x,y), increment any counter which contains a parameter point (b,m) satisfying $b = -x\,m + y$
  - Threshold counters
  - Group edgels that belong to above threshold counters into contours
  - Optional: Refit lines to get higher precision.

Hough Transform Variation

- Problem:
  - m and b are, in principle, unbounded
    - rewrite as sin(t) x + cos(t) y = d
    - For each discrete value of t from 0 to pi, increment counter with nominal d minimal distance from exact value
  - coarse grid leads to grouping of distinct lines
    - in post-fitting stage, re-hough at higher resolution
- Generalizations
  - Any linear in parameters model: e.g a x^2 + b y^2 = 1 can use the same algorithm
  - For f(x,a) = 0, choose any cell a_c s.t. f(x,a_c) < t for some threshold t.
- Limitations:
  - the curse of dimensionality

Grouping on Grouping

- Grouping lines (recall Gestalt discussion)
  - group by proximity
  - $\mu_{pro} = (l_s/g)^2$ where g is the shortest distance between two endpoints
  - under some assumptions you can show $1/\mu_{pro}$ is proportional to the probability the two endpoints are close by accident
  - $\mu_{par} = \{l_s\}^2 / \theta \, s \, l_l$
  - $\mu_{col} = \{l_s\}^2 / \theta \, s \, \{l_s + g\}$

- Algorithm:
  - compute μ_{pro} and either μ_{col} or μ_{par} for all pairs
  - group pairs for which these values exceed threshold

Image Segmentation RoadMap

- Segmentation
  - criteria
  - region group and counting
- Simple Color Segmentation
- Color Histograms and matching
- Selection of Color Regions
- Texture

Segmentation: Definitions

- A distance measure $d(R_1,R_2) \rightarrow \Re$ or a homogeneity measure $m(R)$
  - note possibly $d(R_1,R_2) = |m(R_1) – M(R_2)|$
- A similarity threshold τ (could operate on distance or homogeneity)
- A region definition (e.g. square tiles)
- A neighborhood definition
  - 4 neighbors

```
  x
x 0 x
  x
```

  - 8 neighbors

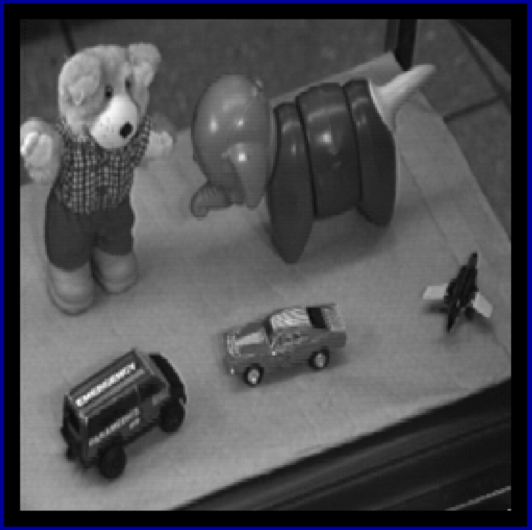
```
x x x
x 0 x
x x x
```

Simple Thresholding

- Choose an image criterion c
- Compute a binary image by b(i,j) = 1 if c(I(i,j)) > t; 0 otherwise
- Perform "cleanup operations" (image morphology)
- Perform grouping
  - Compute connected components and/or statistics thereof

An Example: Motion

Detecting motion:



Thresholded Motion

Detecting motion:

|

–

| > 50

Candidate areas for motion

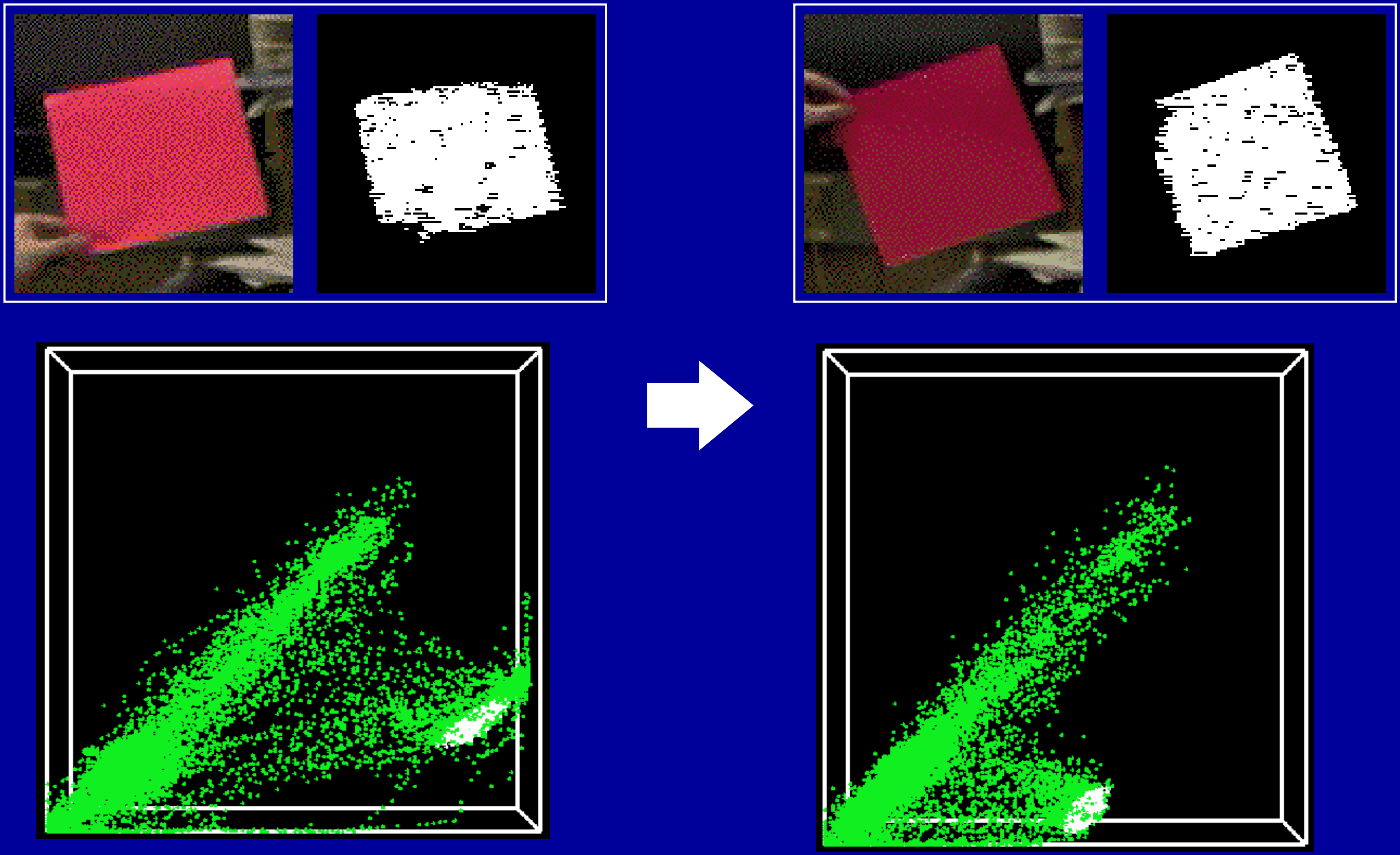
A Closer Look

Color: A Second Example

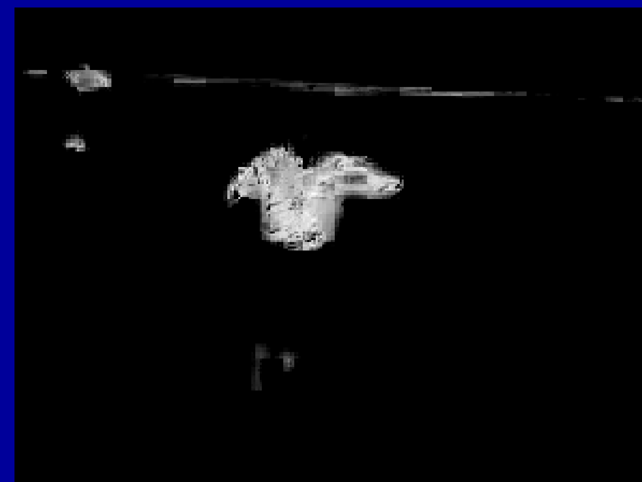
- Color representation
  - DRM [Klinker et al., 1990]: if P is Lambertian, has *matte* line and *highlight* line
  - User selects *matte* pixels in *R*
  - Compute first and second order statistics of cluster
  - Decompose ellipsoid $(\mathbf{S}, \mathbf{R}^T, \mathbf{T})$ of variance of matte cluster
  - Color similarity $\gamma(\mathbf{I}(x, y))$ is defined by Mahalanobis distance

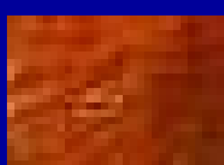
$$|\mathbf{S}^{-1}\mathbf{R}^T(\mathbf{I}(x, y) - \mathbf{T})|$$

Homogeneous Color Region: Photometry

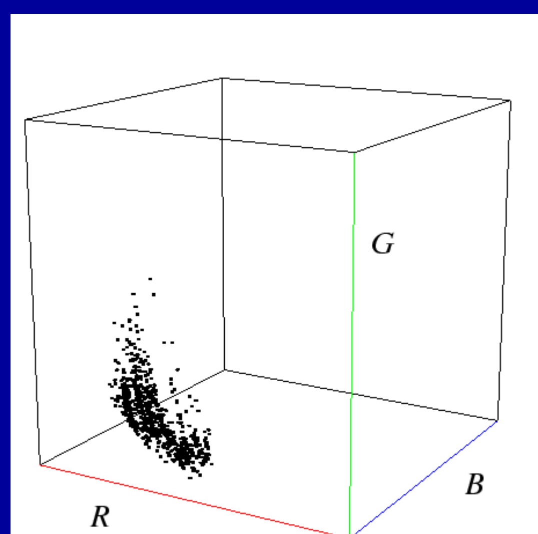


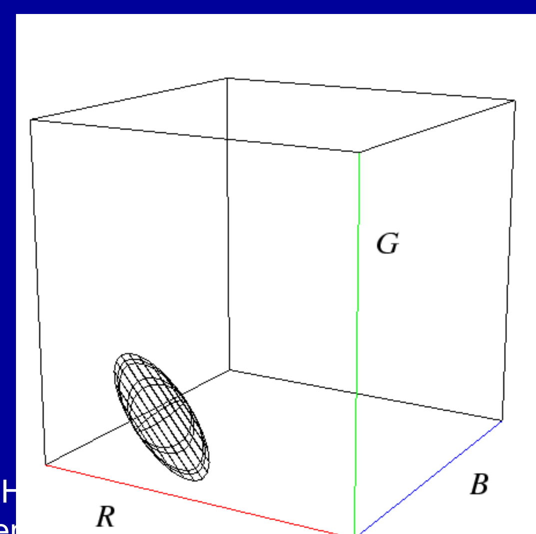
Homogeneous Region: Photometry





Sample





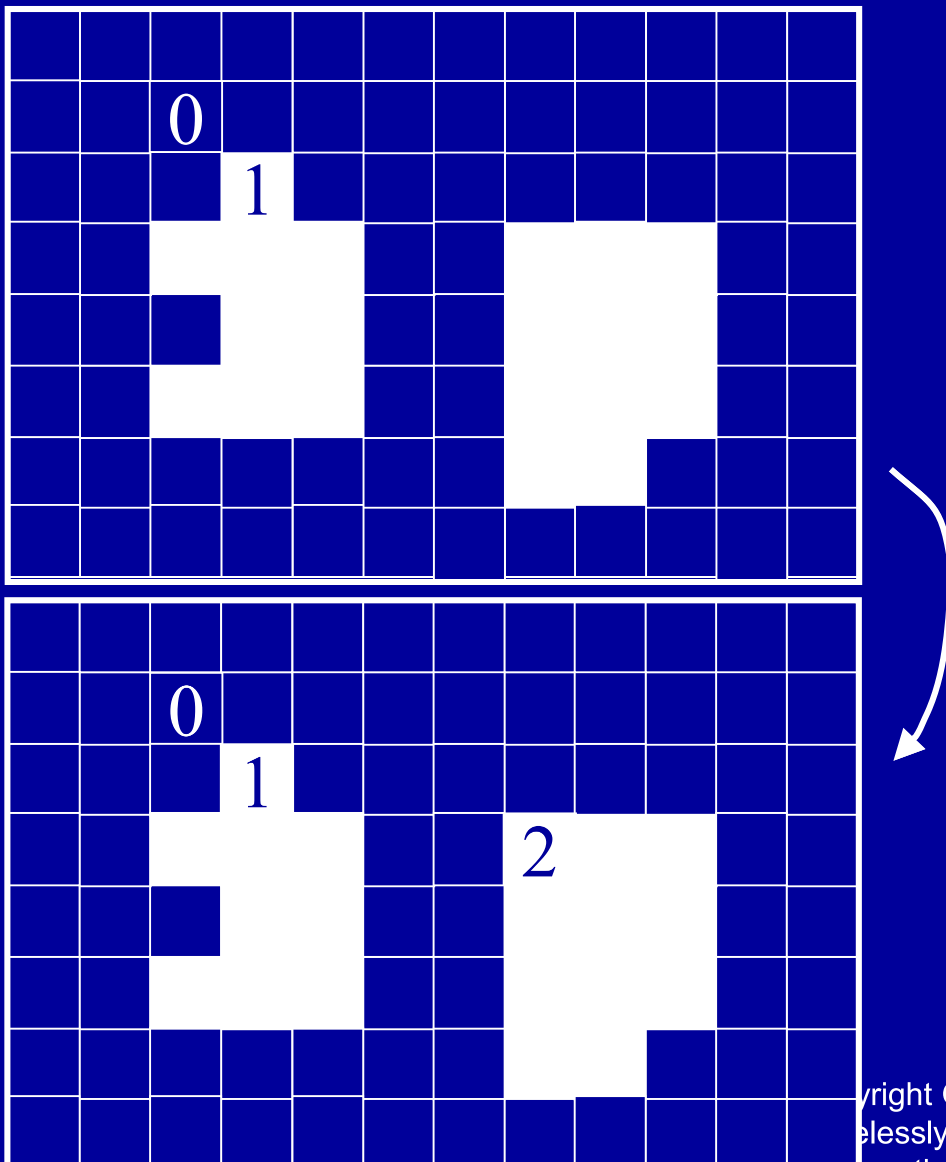
PCA-fitted ellipsoid

Binary Image Processing

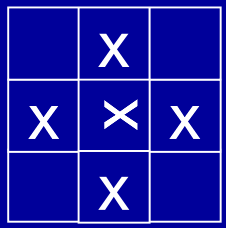
After thresholding an image, we want to know something about the regions found ...

- [x] How many objects are in the image?
- [ ] Where are the distinct "object" components?
- [ ] "Cleaning up" a binary image?

Connected Component Labeling



Goal: Label contiguous areas of a segmented image with unique labels



4 neighbors vs. 8 neighbors

Morphological Operators summarized

Let S_t be the *translation* of a set of pixels S by t.

$$S_t = \{ x + t \mid x \in S \}$$

structuring element

The *dilation* of a binary image A by a mask S is then

$$A \oplus S = \bigcup_{b \in S} A_b$$

The *erosion* of a binary image A by a mask S is

$$A \ominus S = \{ x \mid x + b \in A, \forall b \in S\}$$

The *closing* of a binary image A by S is

$$A \bullet S = (A \oplus S) \ominus S$$

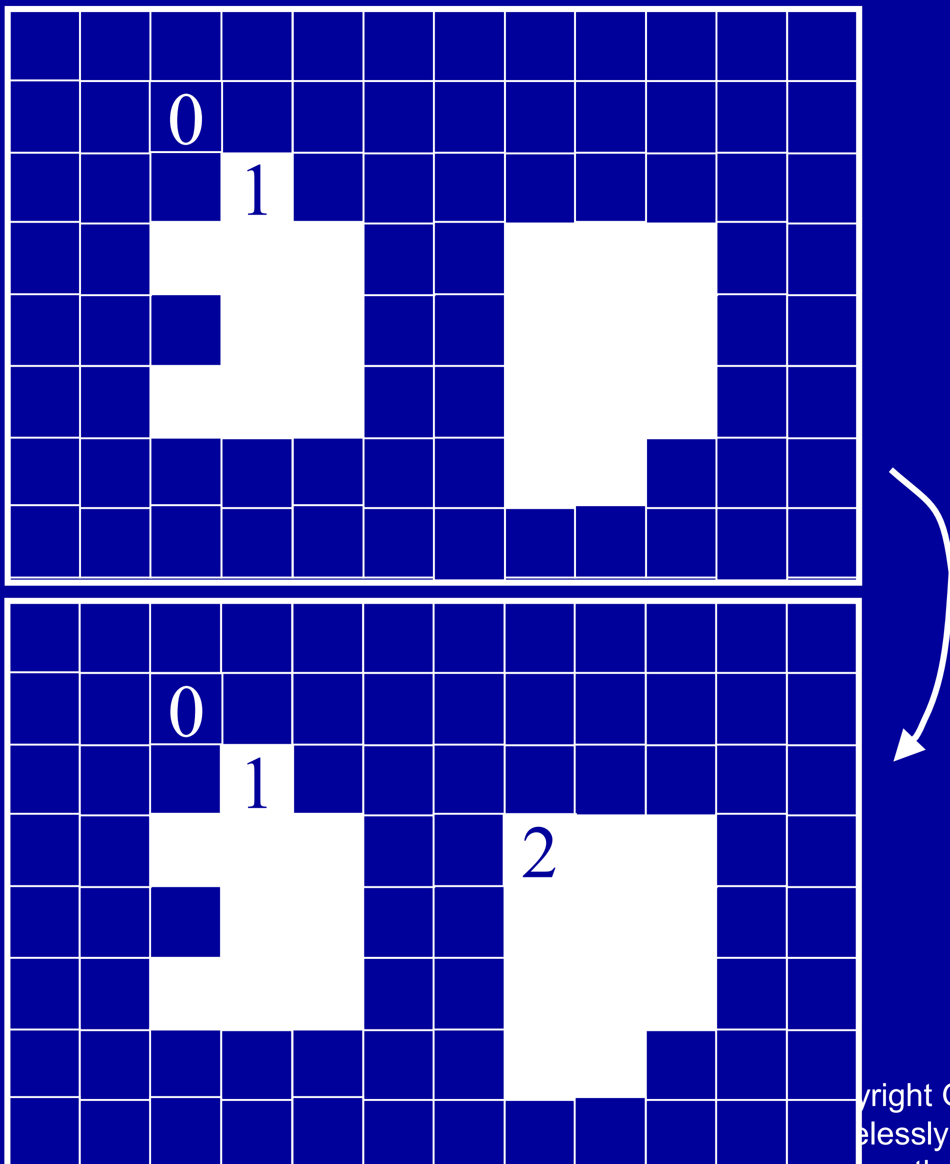
The *opening* of a binary image A by S is

$$A \circ S = (A \ominus S) \oplus S$$

The “Poor Man’s” Closing

- Note that median (or more generally any order statistic) filtering is one way of achieving similar effects. On binary images this can be also implemented using the averaging filter

Connected Component Labeling



Algorithm

1. Image is A. Let A = -A;

2. Start in upper left and work L to R, Top to Bottom, looking for an unprocessed (-1) pixel.

3. When one is found, change its label to the next unused integer. Relabel all of that pixel's unprocessed neighbors and their neighbors recursively.

4. When there are no more unprocessed neighbors, resume searching at step 2 -- but do so where you left off the last time.

Limitations of Thresholding

- A uniform threshold may not apply across the image
- It measures the uniformity of regions (in some sense), but doesn't examine the inter-relationship between regions.

More General Segmentation

- Region Growing:
  - Tile the image
  - Start a region with a seed tile
  - Merge similar neighboring tiles in the region body
  - When threshold exceeded, start a new region
- Region Splitting
  - Start with one large region
  - Recursively
    - check a region for tile neighbor similarity
    - If sufficiently similar, stop, otherwise split
    - repeat until no more splitting occurs

Segmentation: Bottom Up (Grouping)

- 1. Divide the image into the smallest region of interest
- 2. Compute the distance from each region to its neighbors and create a sorted list of distance/neighbor pairs: (measure/region pairs)

  $L = (d_1,p_1), (d_2,p_2), \ldots. (d_N,p_N)$

- 3. While the smallest distance $d_{min} < \tau$ (smallest measure $< \tau$)
  - a. merge the region pair (r_a,r_b) associated with d_{min} creating a new region r'
  - b. remove all pairs containing r_a or r_b from the sorted list
  - c. compute the distance between r' and its neighbors and add these new pairs to the list (add new measure to the list)

Example metrics:

- mean gray value
- gray value variance
- color distance
- edge direction

Note that we can replace "region" with "feature" and perform feature grouping with the same algorithm

Segmentation: Top Down (Partitioning)

- Start with one large region and compute homogeneity m. Call it R* and add it to a list L
- While m(first(L)) > τ
  - 1. consider splits of R* into R_1 and R_2 and choose that which minimizes $m(R_1) + m(R_2)$ (note this is effectively a distance!).
  - 2. remove R* and put R_1 and R_2 onto the list in decreasing order

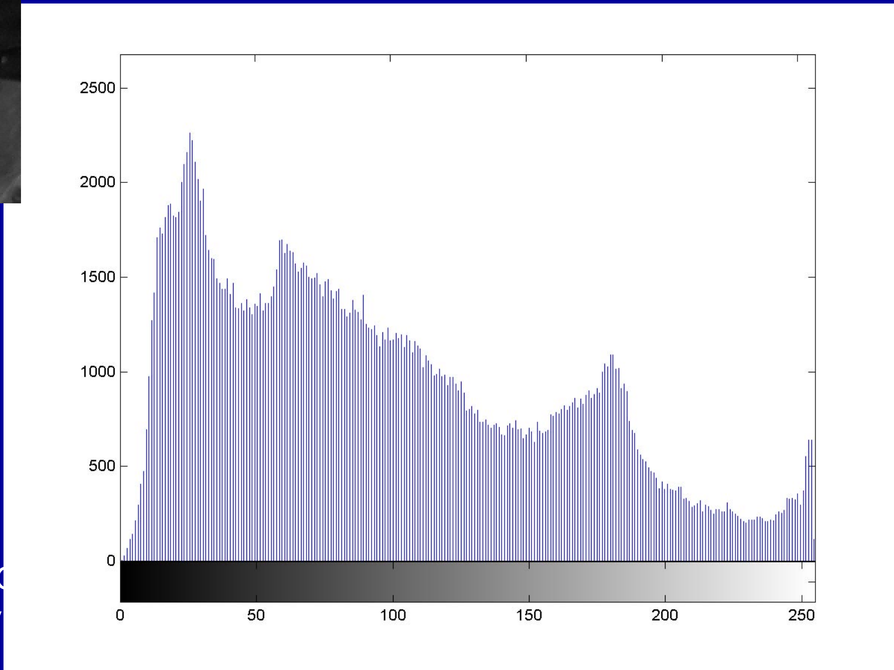
- Note that the resulting segmentation is not guaranteed to be optimal or even connected. It often makes sense to first do a top down segmentation, followed by a bottom-up merge.

An Example: Image Segmentation

- The goal: to choose regions of the image that have similar "statistics."
- Possible statistics:
  - mean
  - variance
  - co-occurance matrix
- Recall:
  - a histogram is a representation of the distribution of values in a range of values
- Idea: merge regions with similar histograms

An Image Histogram

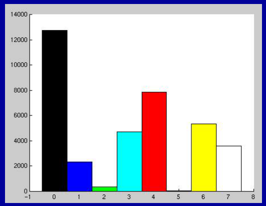
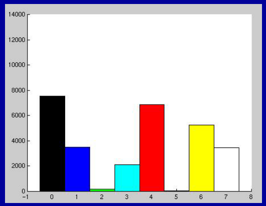
- For each pixel p_i, let h(p_i) = h(p_1) + 1



- EMD

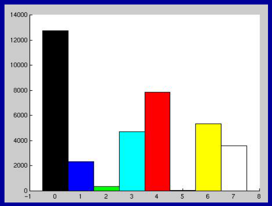
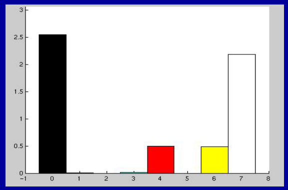
Histogram Metrics

Intersection

 $\cap$  $=$

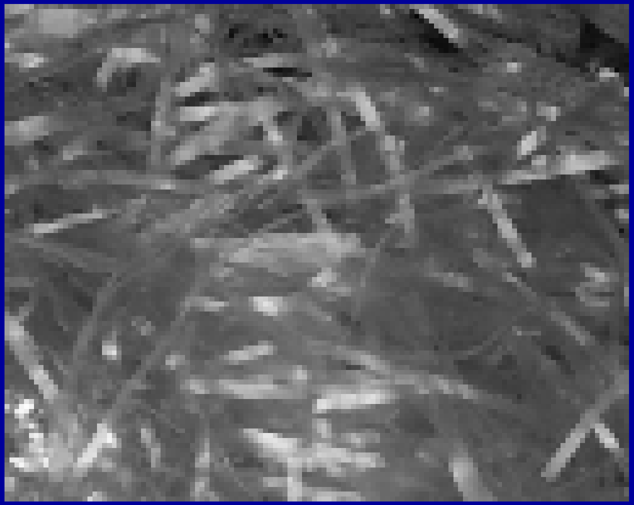
$$\frac{\sum_i \min(M_i\ ,\ T_i)}{\sum_i\ T_i}$$

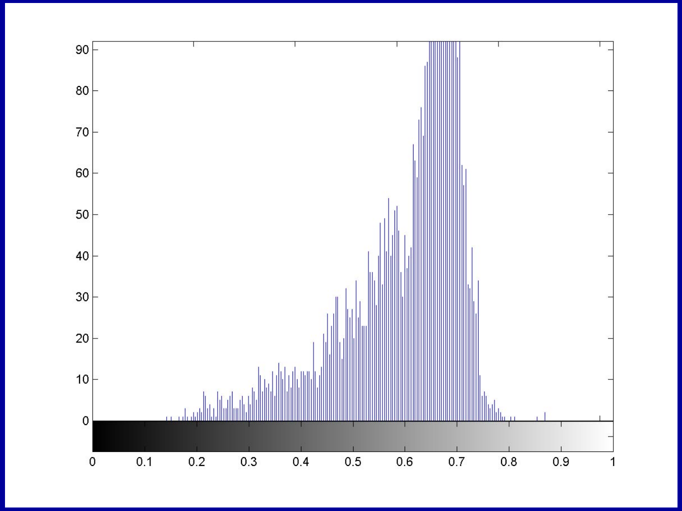
Inner Product

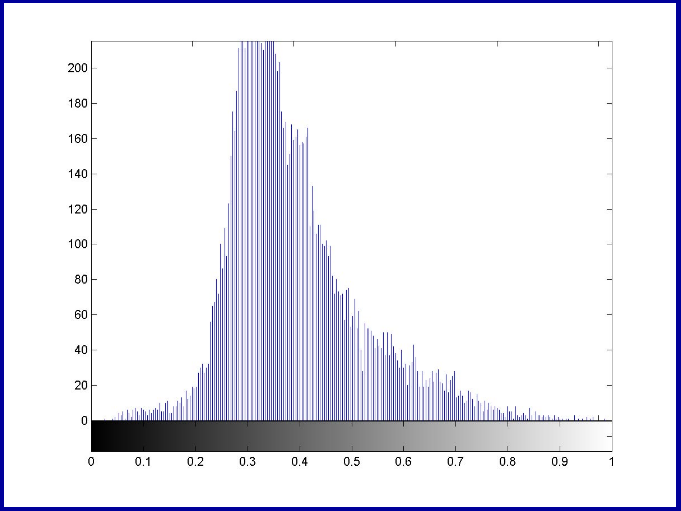
 $\cap$  $=$

$$M \cdot T = \frac{\sum M_i T_i}{\sqrt{\sum M_i^2}\sqrt{\sum T_i^2}}$$

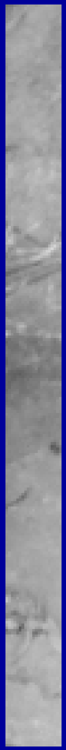
Comparing Histograms

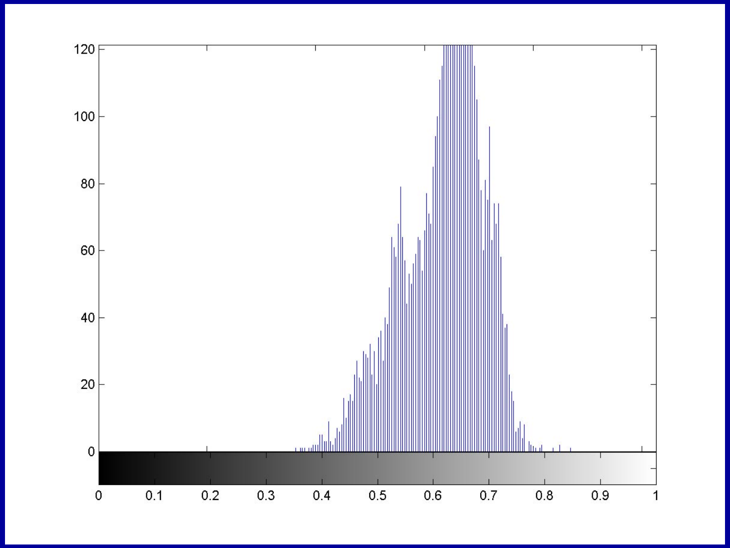




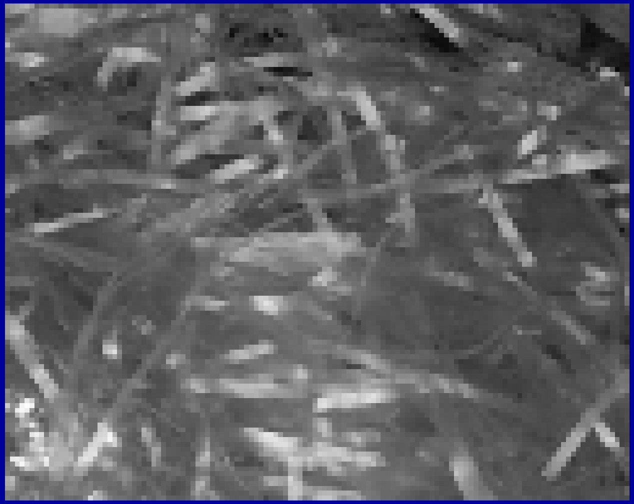


Which is More Similar?





.906

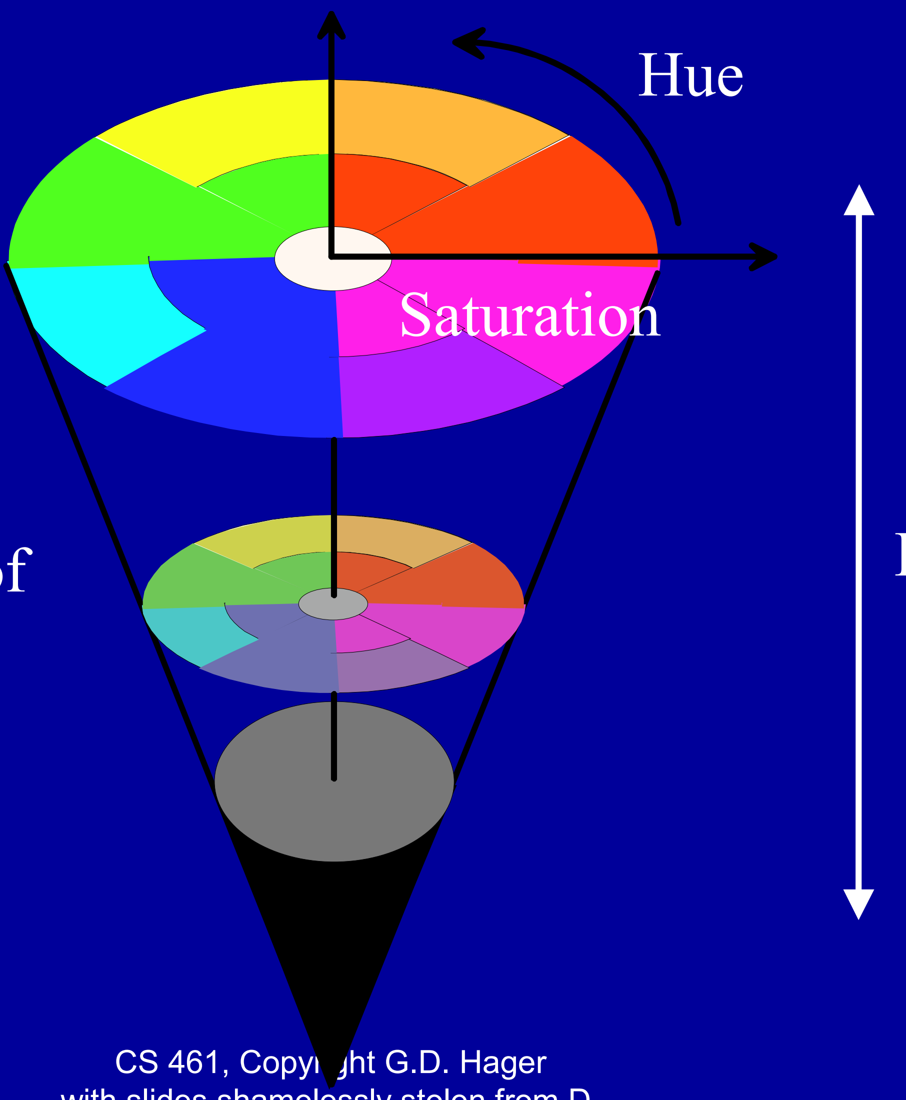


.266

An Example: Color Image Segmentation

- The goal: to choose “unique” regions of an image
- Useful for mobile robot navigation
- Provides a useful example of color histograms
- Recall:
  – a histogram is a representation of the distribution of values in a range of values:

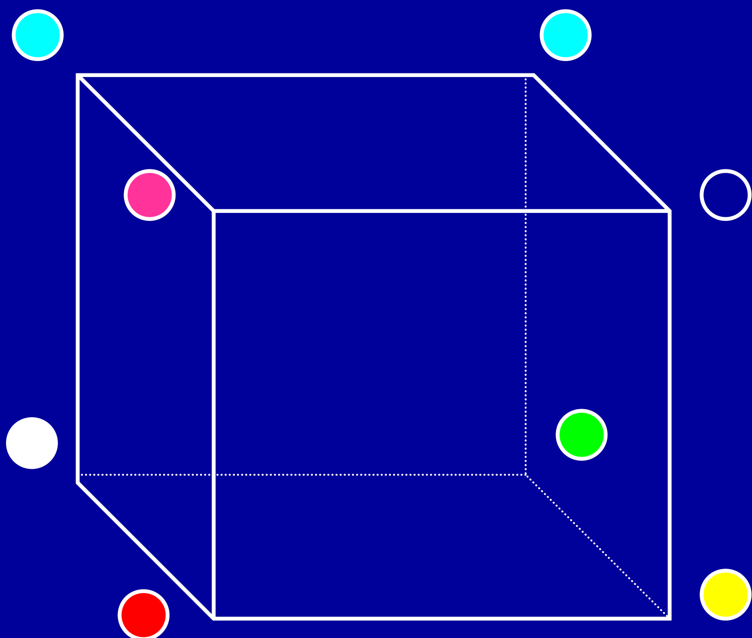
Dividing Up Color Space



- HSI representation of color space
- Variable resolution gridding of space

Color Histograms

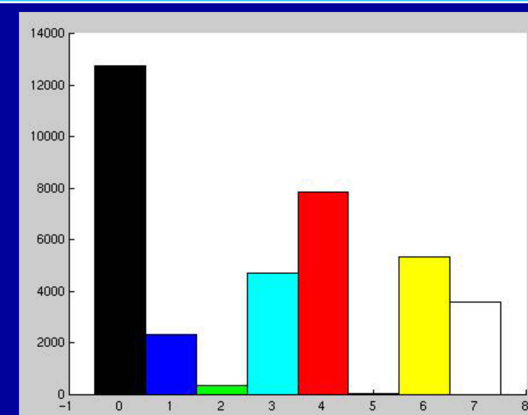
feature vectors



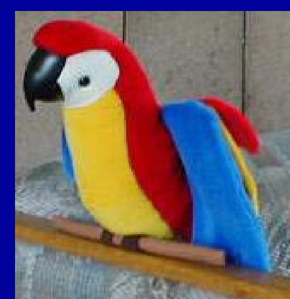
RGB

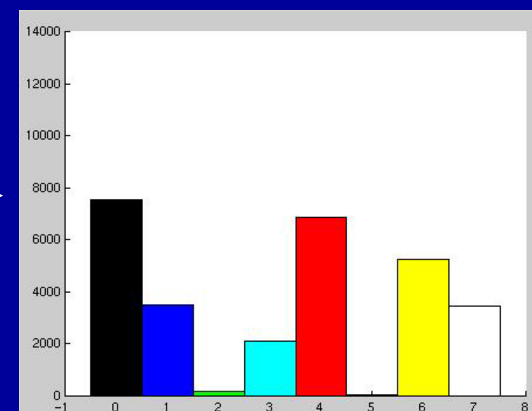
A bit of binning...

model

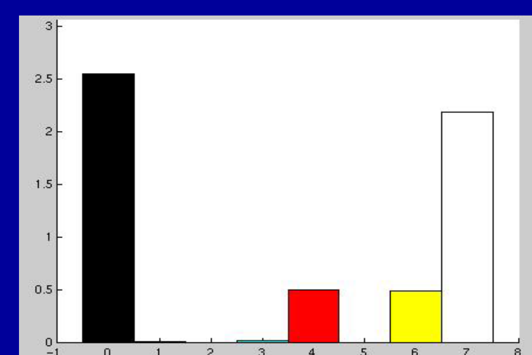


test

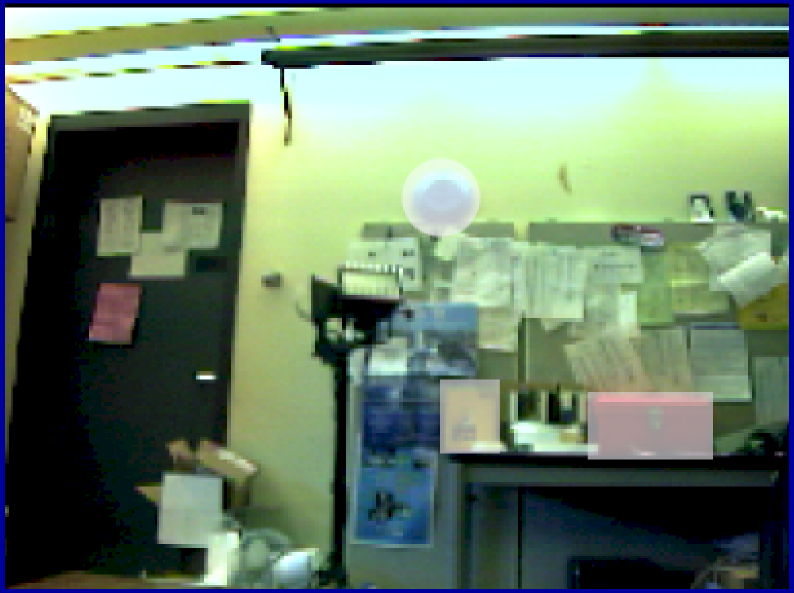


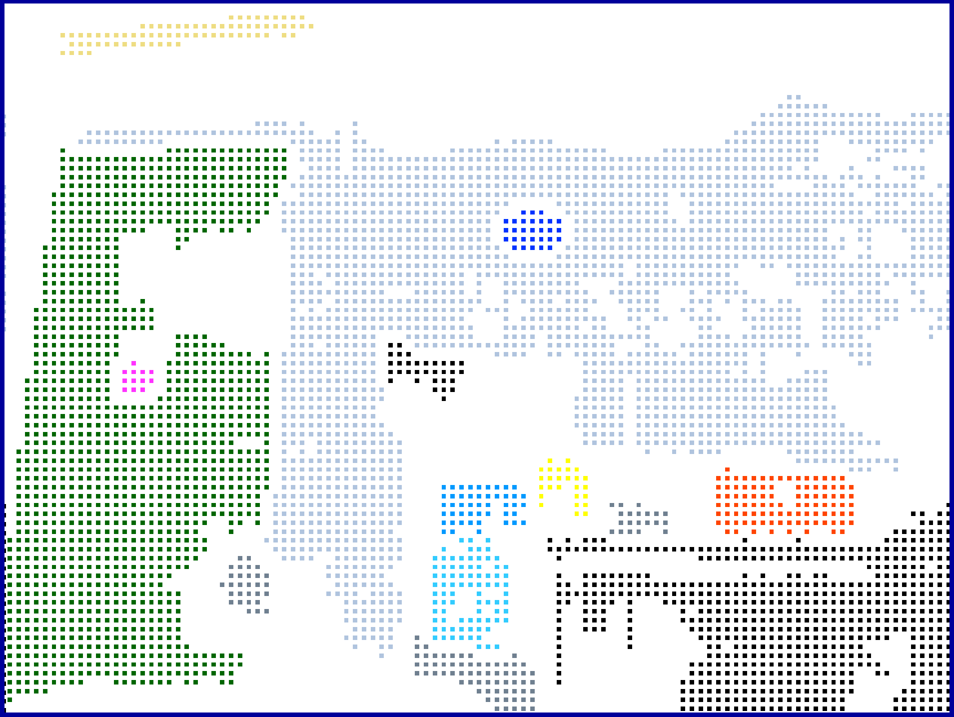


test



Results of a Merge Segmentation

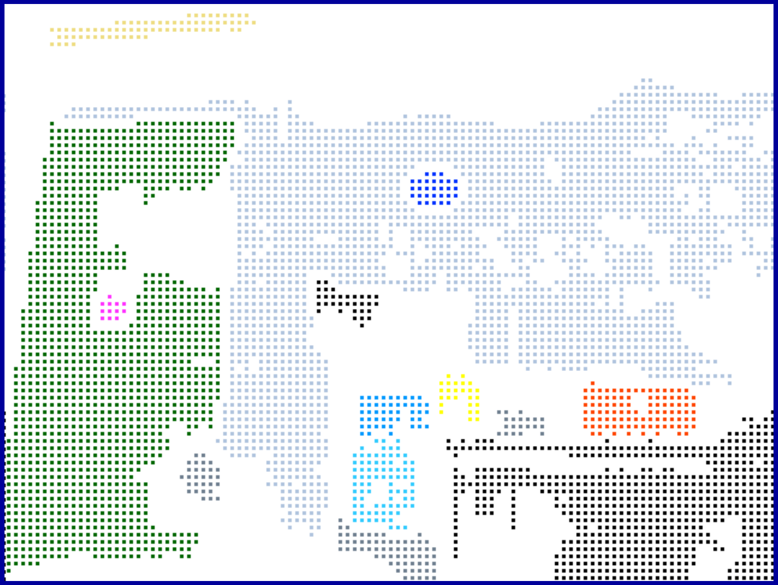




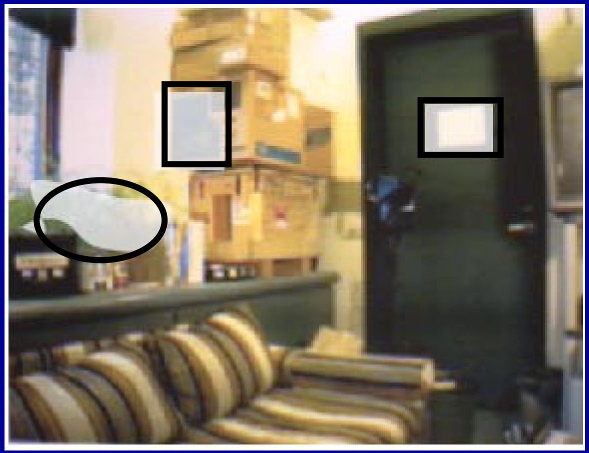
Characterizing Regions

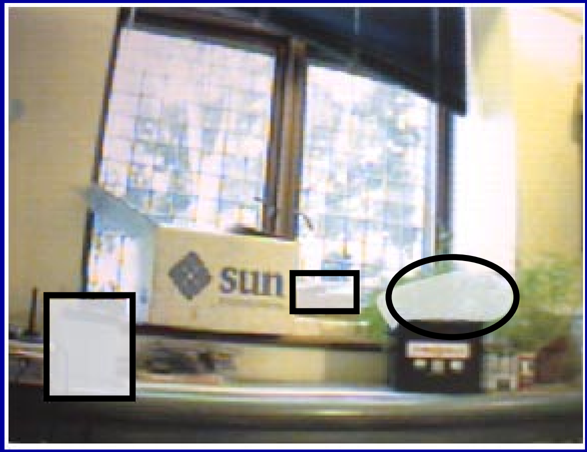
• Distinctiveness	
– distractors (image distance to nearest similar region)	3.0
– uniqueness (whole-image correlation)	1.0
• Scale changes	
– size weighting	0.5
• Overlapping objects	
– circularity (ratio of the area to the perimeter^2)	0.8
• Lighting effects	
– frame correlation	1.0
– average saturation	2.5

Top Three Features



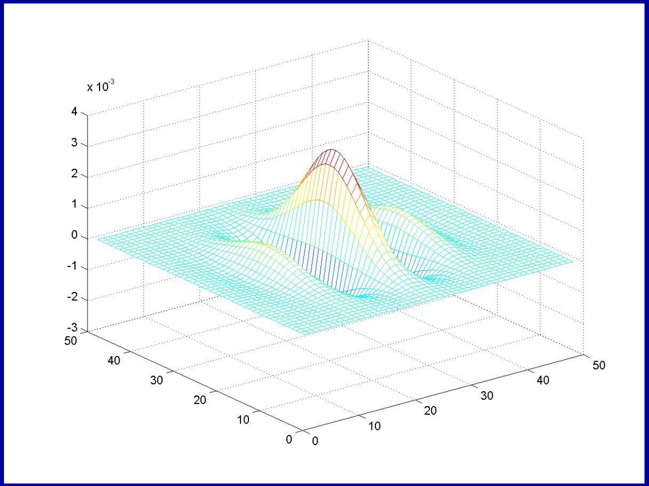
More Examples



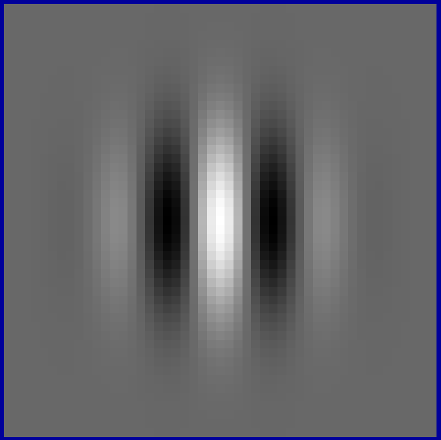


Even More Dimensions

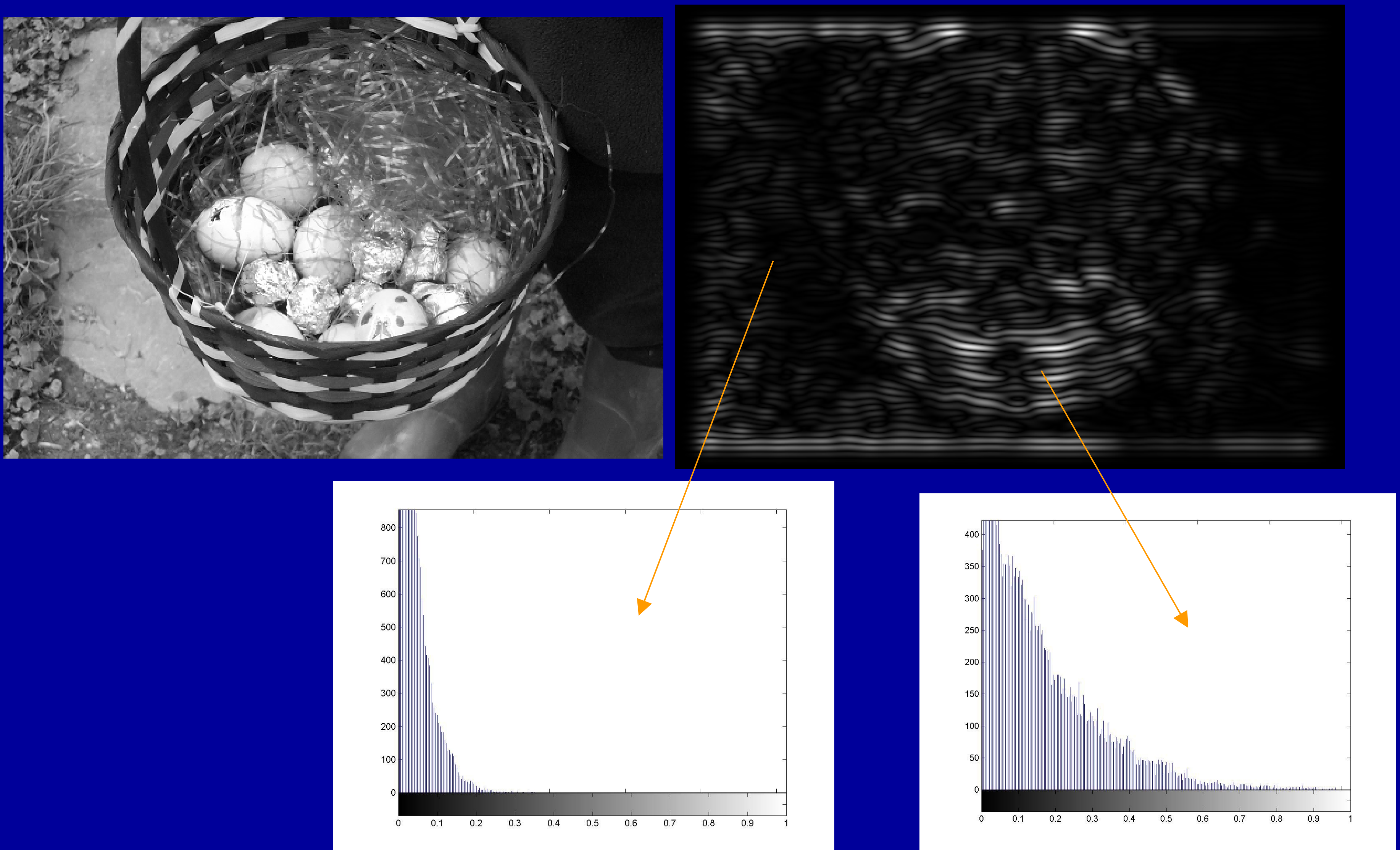
- Are gray value histograms all we need?
  - not brightness invariant (what is?)
  - don't capture orientation
  - don't capture other pattern regularity
- How about computing something "richer"
  - use derivative measures
  - oriented
  - multiple scales



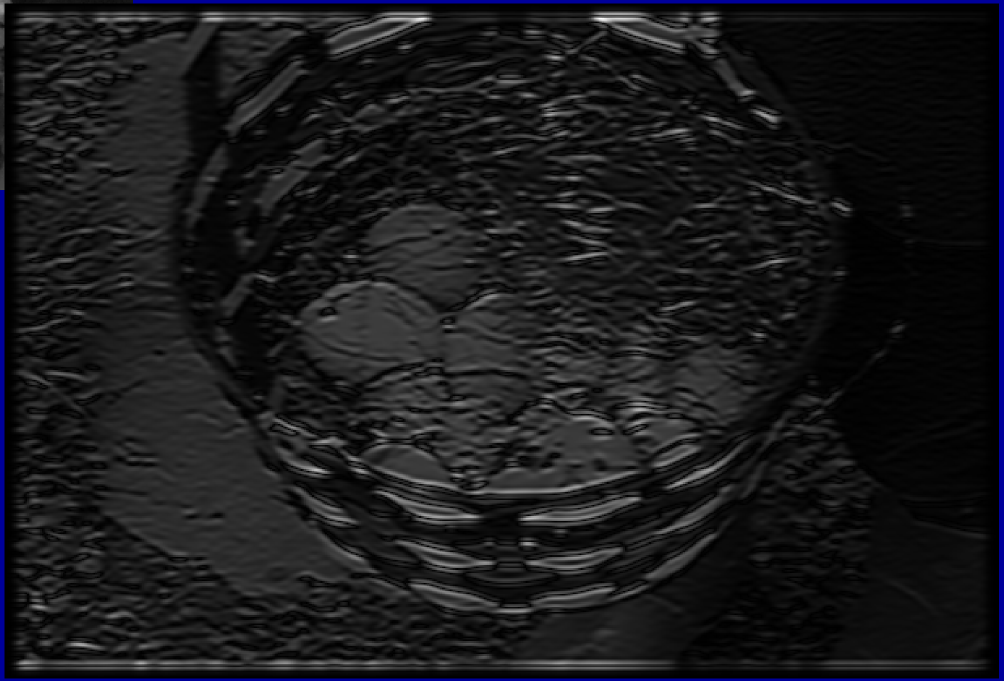
$$\cos(ax + by)\exp(-(x^2 + y^2)/2\,\sigma^2)$$



Gabor Response



Gabor Response



An (Example) Plan

- Image Filters:
  - gabor at multiple scales
  - gabor at multiple orientations
- Histograms of results
- Segmentation based on histograms
- Use of texture to
  - segment
  - compute surface properties
  - etc.

Summary

- Hough transform is a useful tool for grouping
- An example of feature grouping from edges.
- Basic thresholding and region labeling is a simple tool for segmentation
- Top-down and bottom up methods allows inter-region comparisons

Computer Vision, Lecture 8

Professor Hager
http://www.cs.jhu.edu/~hager

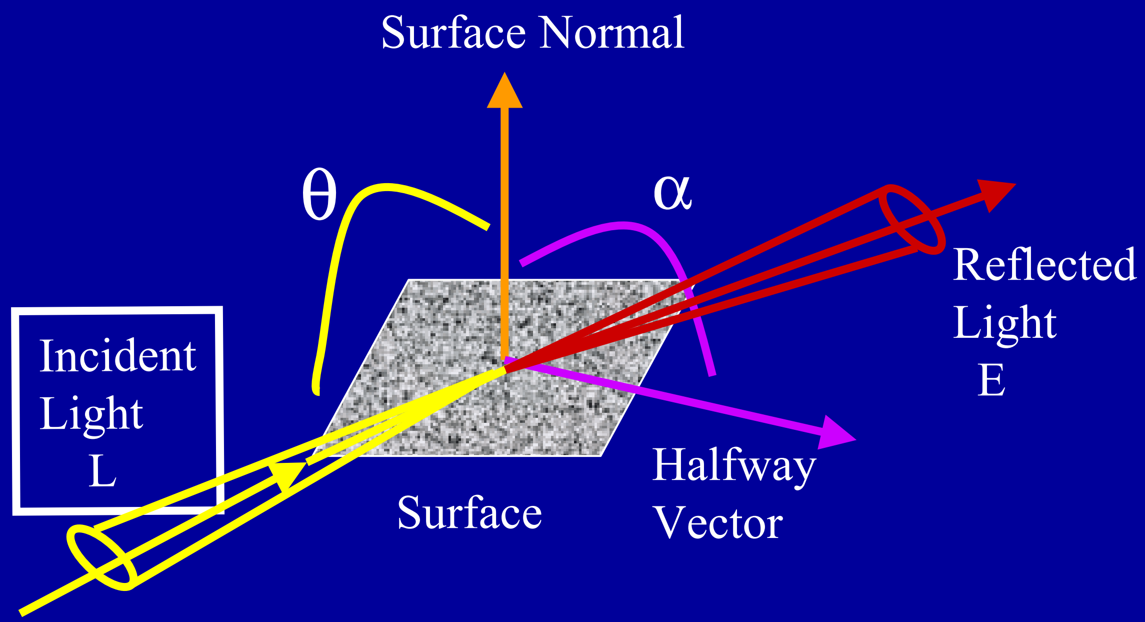
Outline for Today

- Reflectance

- Examples:
  - understanding surfaces
  - finding specularities
  - image-based rendering

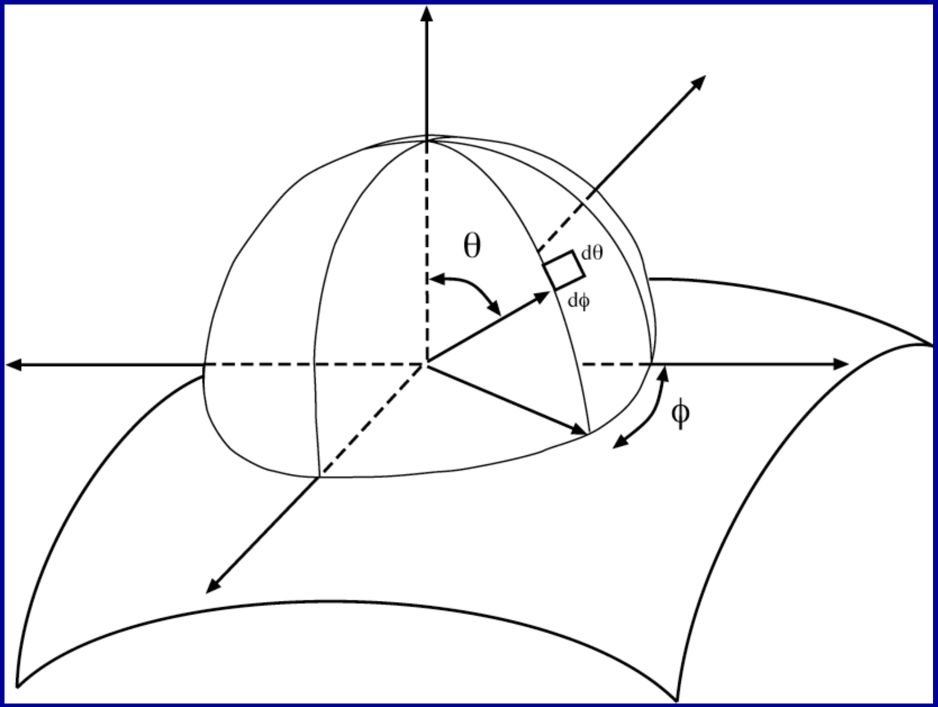
REFLECTANCE MODELS

- Description of how light energy incident on an object is transferred from the object to the camera sensor



Radiometry

- Core Questions:
  - how “bright” will surfaces be?
  - what is “brightness”?
    - measuring light
    - interactions between light and surfaces
- Core idea - think about light arriving at a surface
- around any point is a hemisphere of directions
- Simplest problems can be dealt with by reasoning about this hemisphere



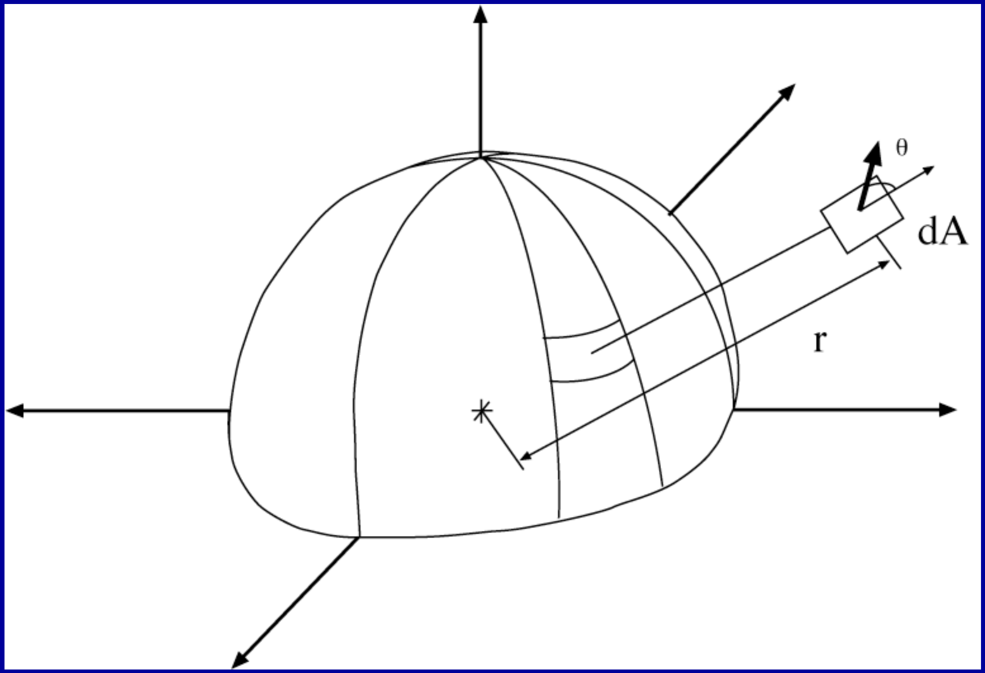
Solid Angle

- Defined by analogy with angle (in radians)
  - solid angle subtended by a patch is the area covered by the patch on a unit sphere
- The solid angle subtended by an infinitesimal patch area dA is given by

$$d\omega = \frac{dA\cos\vartheta}{r^2}$$

- Another useful expression:

$$d\omega = \sin\vartheta (d\vartheta)(d\phi)$$



Radiance

- Measure the "amount of light" at a point, in a direction
- Definition: *Radiant power per unit foreshortened area per unit solid angle*
- Units: watts per square meter per steradian ($wm^{-2}sr^{-1}$)
- Usually written as:

$$L(\underline{x}, \vartheta, \varphi)$$

- Crucial property: In a vacuum, radiance leaving p in the direction of q is the same as radiance arriving at q from p
  - hence the units

Irradiance

- How much light is arriving at a surface?
- Sensible unit is *Irradiance*
- Incident power per unit area *not foreshortened*
- This is a function of incoming angle.
- A surface experiencing radiance $L(x,\theta,\phi)$ coming in from $d\omega$ experiences irradiance

$$L(\underline{x},\vartheta,\varphi)\cos\vartheta d\omega$$

Irradiance

- Crucial property:
  - Total power arriving at the surface is given by adding irradiance over all incoming angles

- Total power is

$$\int_{\Omega} L(\underline{x}, \vartheta, \varphi) \cos\vartheta \sin\vartheta d\vartheta d\varphi$$

Surfaces and the BRDF

- Many effects when light strikes a surface -- could be:
  - absorbed
  - transmitted
  - reflected
  - scattered
- Assume that
  - surfaces don't fluoresce
  - surfaces don't emit light (i.e. are cool)
  - all the light leaving a point is due to that arriving at that point

The BRDF

- Given assumptions, we can model effects at a surface with a record of outgoing vs incoming illumination
  - the Bidirectional Reflectance Distribution Function (BRDF)
- Definition:
  - the ratio of the radiance in the outgoing direction to the incident irradiance
- Units: Inverse steradians

$$\rho_{bd}(\underline{x}, \vartheta_o, \varphi_o, \vartheta_i, \varphi_i,) = \frac{L_o(\underline{x}, \vartheta_o, \varphi_o)}{L_i(\underline{x}, \vartheta_i, \varphi_i)\cos\vartheta_i d\omega}$$

$$L_o(\underline{x}, \vartheta_o, \varphi_o) = \rho_{bd}(\underline{x}, \vartheta_o, \varphi_o, \vartheta_i, \varphi_i,) L_i(\underline{x}, \vartheta_i, \varphi_i)\cos\vartheta_i d\omega$$

Lambertian surfaces and albedo

- for a Lambertian surface, BRDF is independent of angle.
- Useful fact:

albedo

$$\rho_{brdf} = \frac{\rho_d}{\pi}$$

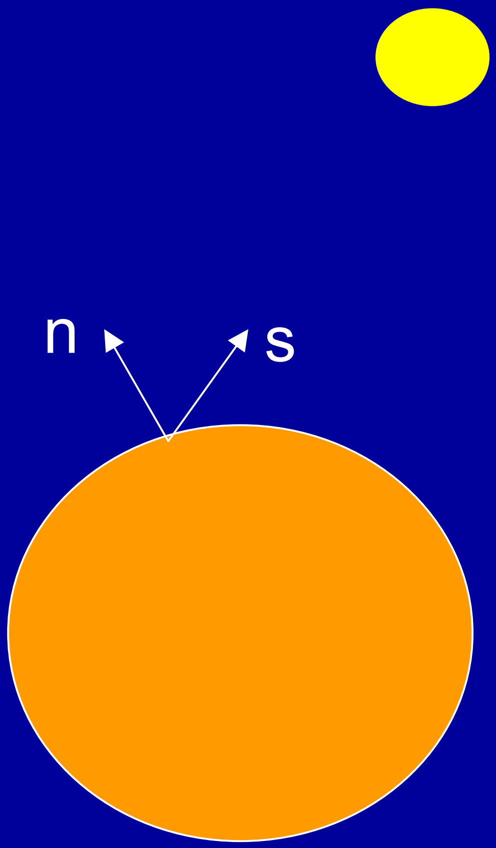
$$L_o(\underline{x}) = \rho_{brdf}(x) \int L_i(\underline{x}, \vartheta_i, \varphi_i) \cos \vartheta_i d\omega$$

Suppose now that we have an ideal distant pinhole source in a given direction. Then we if look at the entire power

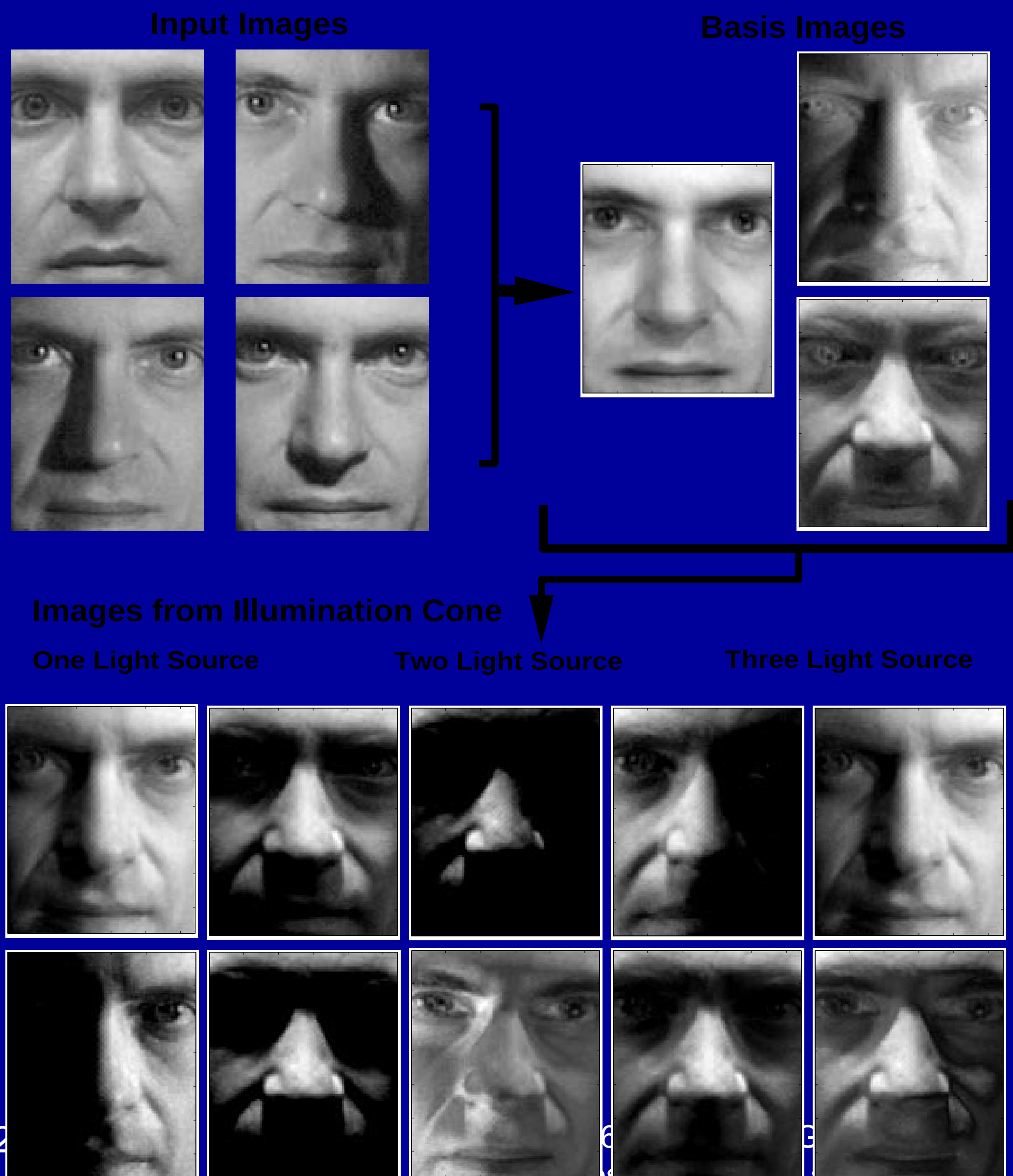
$$L_o(\underline{x}) = \rho_{brdf}(x) \int_{\Omega} L_i(\underline{x}, \vartheta_i, \varphi_i) \cos \vartheta_i d\omega = \rho(x)_{brdf} L_{source} \cos \vartheta_{source}$$

Another View

- Recall that $\cos(\theta) = n \cdot s$
  - n unit normal to surface
  - s unit vector directed toward source
- So, we can also write $L_0 = \rho_{brdf}\, n \cdot s$
- An interesting exercise (Example Use 1):
  - show that three pinhole light sources suffice to fully determine the photometric structure of a Lambertian surface
  - hint: combine ρ and n into a single (now not unit vector) quanitity and consider varying the direction s
- A second interesting execise: show that given this you can synthesize a view of the scene from a new direction



A Slightly More Elaborate Approach



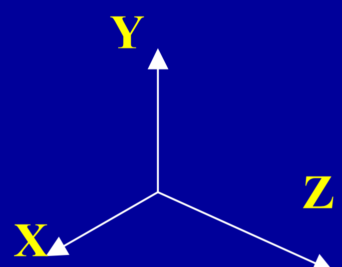
One issue is dealing with occlusion; there are methods that work around that.

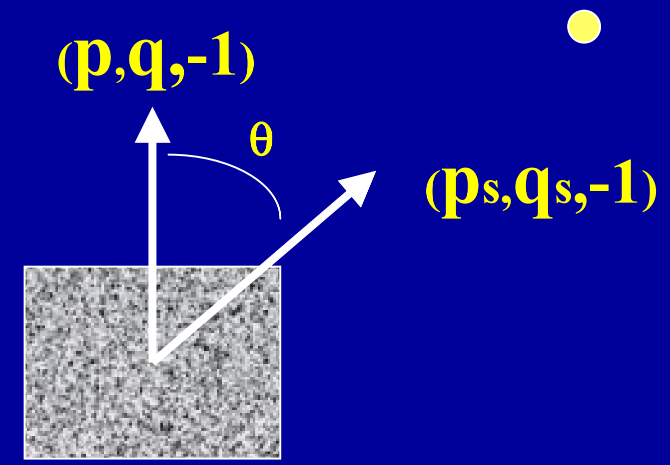
Recent work on *Lumigraphs* is an example of taking this general approach of rendering from image to the extreme.

LAMBERTIAN REFLECTANCE MAP

LAMBERTIAN MODEL

$$E = L \rho \, COS\,\theta$$





$$COS\theta = \frac{1 + pp_s + qq_s}{\sqrt{1 + p^2 + q^2}\sqrt{1 + p_s{}^2 + q_s{}^2}}$$

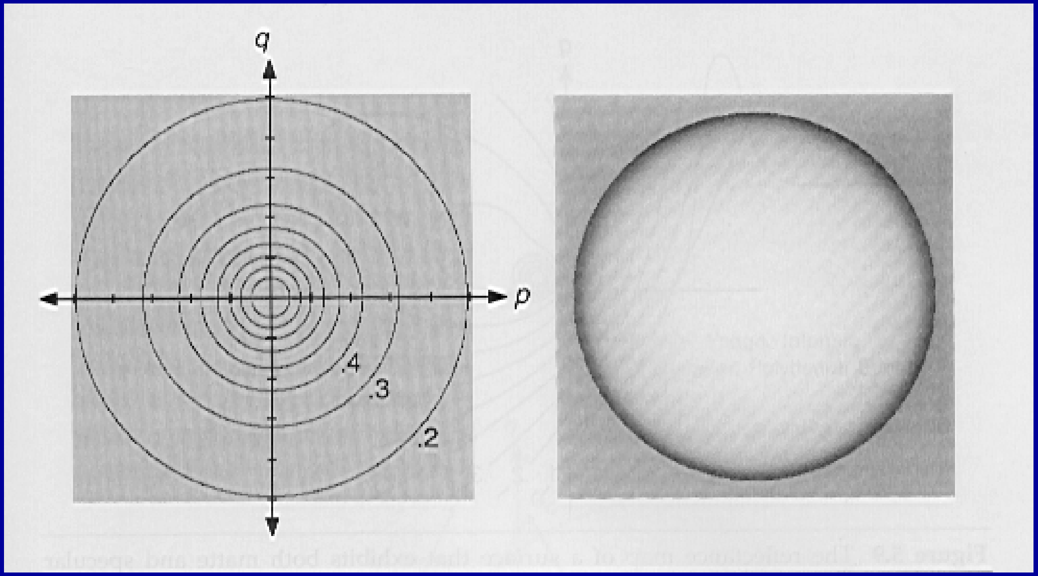
LAMBERTIAN REFLECTANCE MAP

$$E = L\rho \frac{1 + pp_s + qq_s}{\sqrt{1 + p^2 + q^2}\sqrt{1 + p_s{}^2 + q_s{}^2}}$$

Grouping L and ρ as a constant , local surface orientations that produce equivalent intensities under the Lambertian reflectance map are quadratic conic section contours in gradient space.

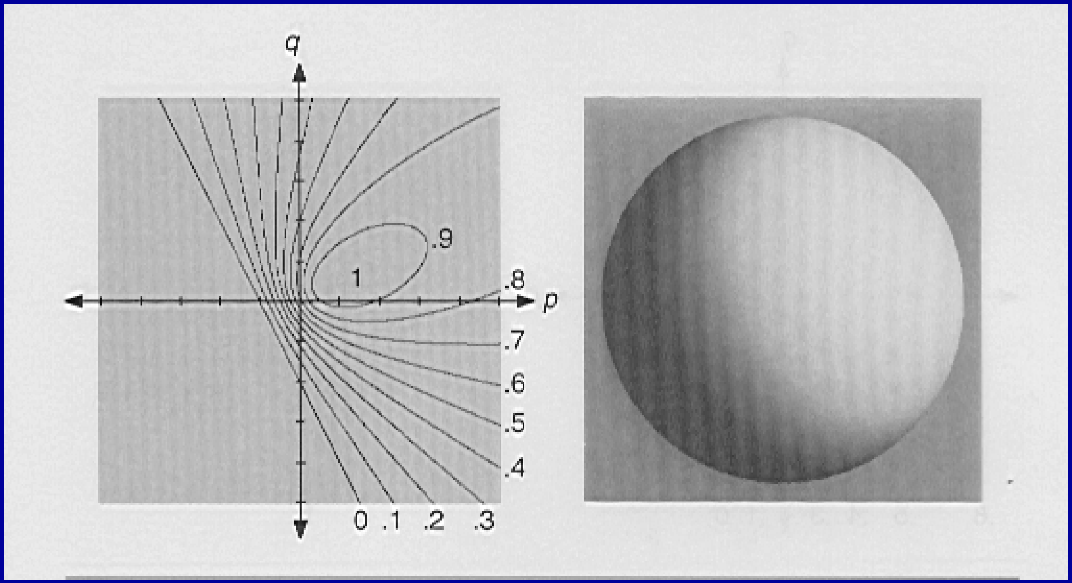
$$I = \frac{1 + pp_s + qq_s}{\sqrt{1 + p^2 + q^2}\sqrt{1 + p_s{}^2 + q_s{}^2}}$$

LAMBERTIAN REFLECTANCE MAP



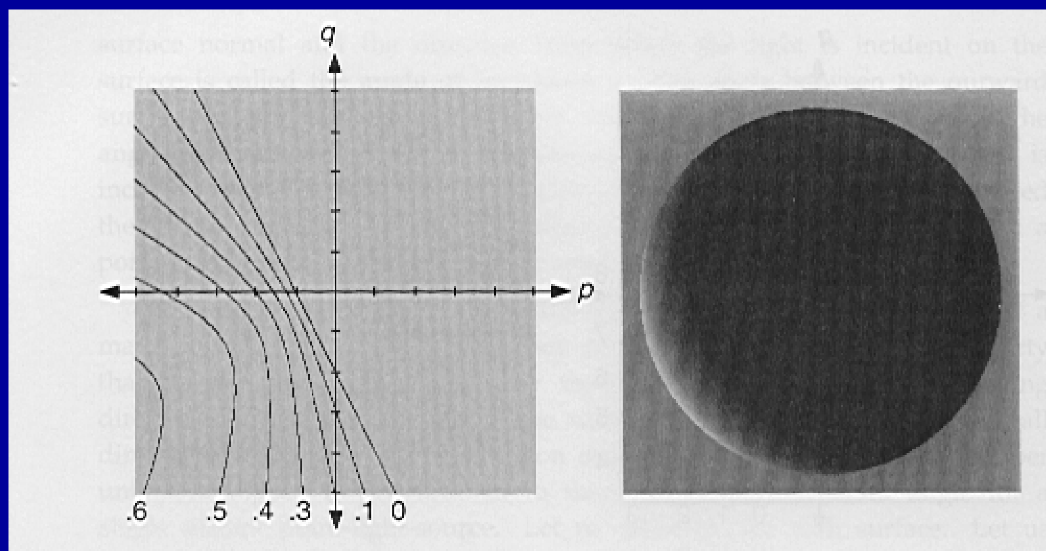
$p_s=0 \quad q_s=0$

LAMBERTIAN REFLECTANCE MAP



$p_s=0.7 \quad q_s=0.3$

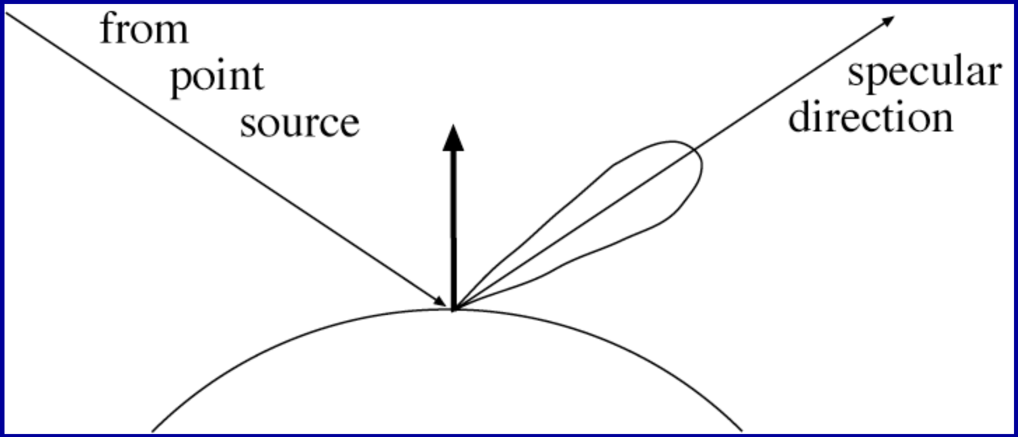
LAMBERTIAN REFLECTANCE MAP



$$p_s = -2 \quad q_s = -1$$

Specular surfaces

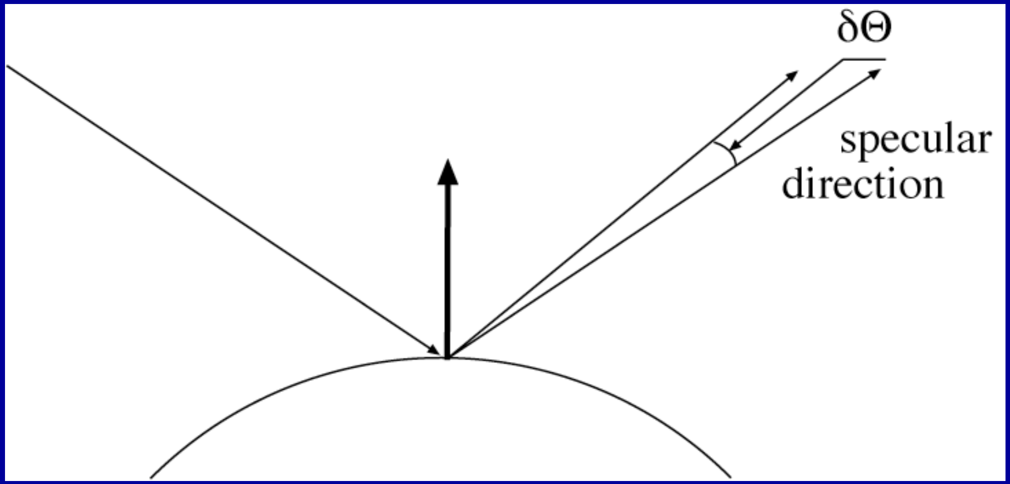
- Another important class of surfaces is specular, or mirror-like.
  - radiation arriving along a direction leaves along the specular direction
  - reflect about normal
  - some fraction is absorbed, some reflected
  - on real surfaces, energy usually goes into a lobe of directions
  - can write a BRDF, but requires the use of funny functions



Phong's model

- There are very few cases where the exact shape of the specular lobe matters.
- Typically:
  - very, very small --- mirror
  - small -- blurry mirror
  - bigger -- see only light sources as "specularities"
  - very big -- faint specularities
- Phong's model
  - reflected energy falls off with

$$\cos^n(\delta\vartheta)$$



Lambertian + specular model

- Widespread model
  - all surfaces are Lambertian plus specular component
- Advantages
  - easy to manipulate
  - very often quite close true
- Disadvantages
  - some surfaces are not
    - e.g. underside of CD's, feathers of many birds, blue spots on many marine crustaceans and fish, most rough surfaces, oil films (skin!), wet surfaces
  - Generally, very little advantage in modelling behaviour of light at a surface in more detail -- it is quite difficult to understand behaviour of L+S surfaces

$$L(x,\theta_0,\phi_0) = \rho_l(x)\cos\theta_i + \rho_s(x)L(x,\theta_s,\phi_s)\cos^n(\theta_s - \theta_o)$$

Finally --- Light Goes to the Camera

- Question: How much light reaches a camera sensor?
  - In particular, note that camera pixels are *foreshortened* as we move away from the optical center.

$$E(p) = L(P)\pi / 4(d / f)^2 \cos^4 \alpha$$

This implies that we expect objects away from the axis to be darker than those near the axis. E.g. for 25 degrees off center, a drop-off of approximately 33%.

Example Use 2

- Suppose I want to determine the brightness of a light illuminating a normal scene
  1. Assume a point light source and a point sensor

  2. Assume the normal light intensity reaching a detector is

     $L_0 = \int \rho_{bd}(x,\theta_0,\phi_0,\theta_i,\phi_i)L_i(x,\theta_i,\phi_i)\cos\theta_i d\omega$

  3. Now, adding our new point light source and noting that x, θ_0, ϕ_0, θ_i, and ϕ_i are fixed values for the source and detector, we have

  4. $L_{new} = \int \rho_{bd}(x,\theta_0,\phi_0,\theta_i,\phi_i)L_i(x,\theta_i,\phi_i)\cos\theta_i d\omega + \rho_{bd}(x,\theta_0,\phi_0,\theta_\lambda,\phi_\lambda)L_i(x,\theta_\lambda,\phi_\lambda)\cos\theta_l = L_0 + r_\lambda I$

  5. Thus, if we
     1. Take a background image (L_0)
     2. Illuminate the scene full blast ($L_{max}L_0 + r_\lambda I$)
     3. We can compute the light level for some new setting $L_{measured}$ as
        1. $I_{measured} = L_{measured}/(L_{max} - L_0)$

When Will This Fail?

1. $L_{new} = \int \rho_{bd}(x,\theta_0,\phi_0,\theta_i,\phi_i)L_i(x,\theta_i,\phi_i)\cos\theta_i d\omega + \rho_{bd}(x,\theta_0,\phi_0,\theta_\lambda,\phi_\lambda)L_i(x,\theta_\lambda,\phi_\lambda)\cos\theta_l$
   $= L_0 + r_\lambda\, I$

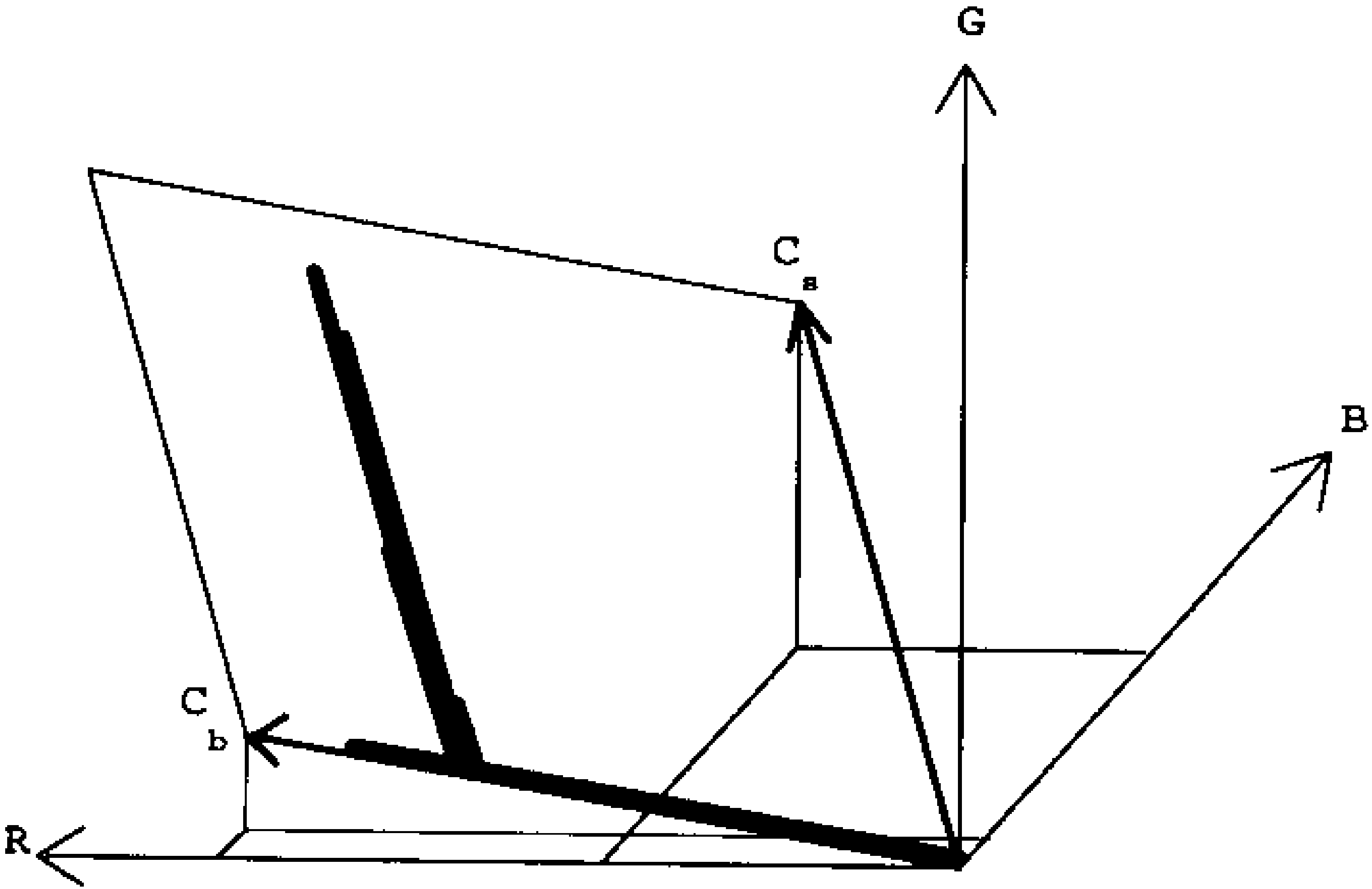
2. $I_{measured} = L_{measured}/(L_{max} - L_0)$

- When is the difference in the denominator small?
  - cos \theta is small → surface is turned away from light source
  - surface albedo is small
  - L_0 is large relative to projected intensity
    - consider sun shining into a specularity oriented toward the detector

  - Surface is highly specular and there is no direct path from source to detector

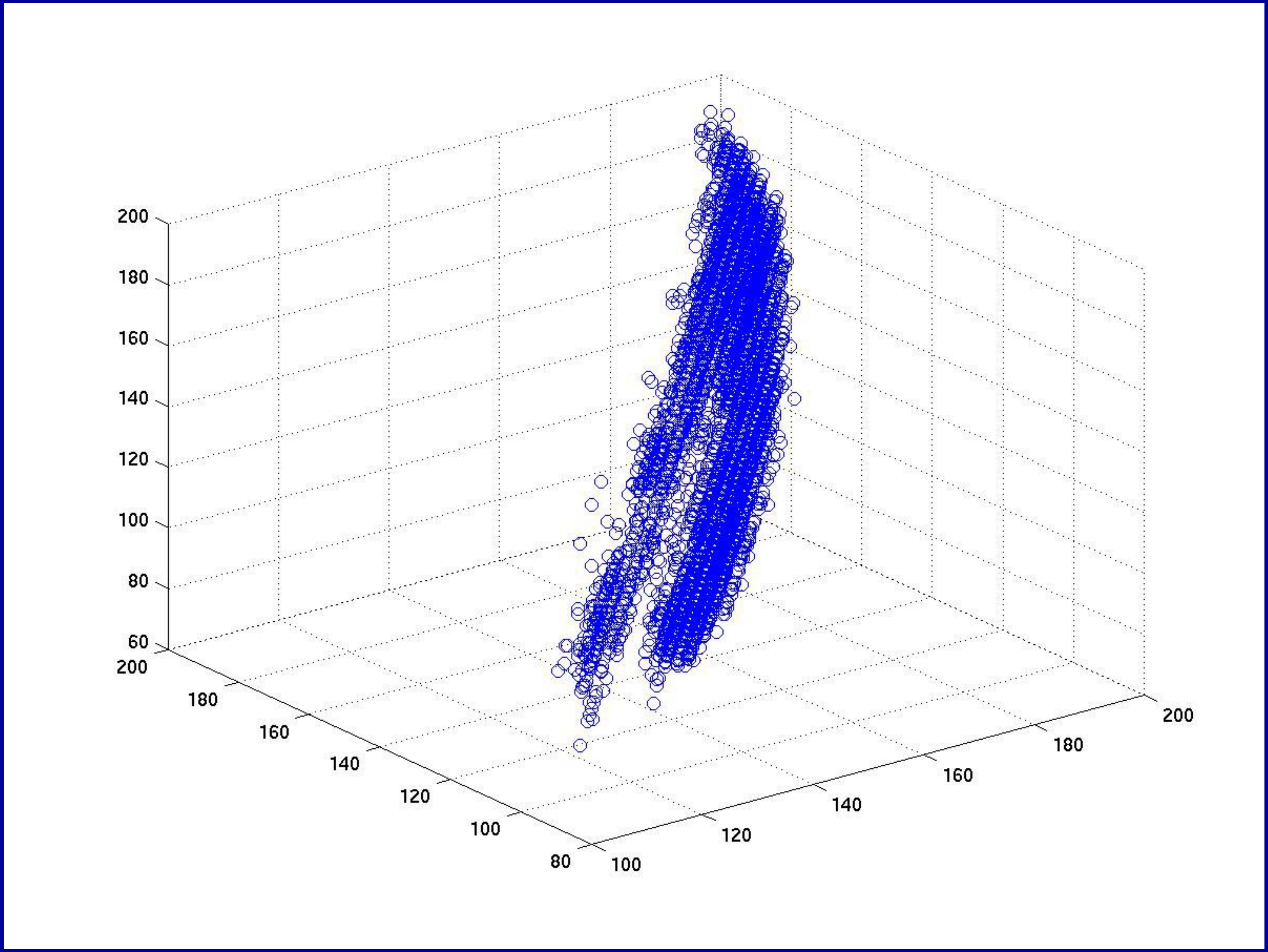
Example Use 3: Finding Specularities

- Assume we are dealing with dielectrics
  - specularly reflected light is the same colour as the source
- Reflected light has two components
  - diffuse
  - specular
  - and we see a weighted sum of these two


G
C
a
B
C
b
R

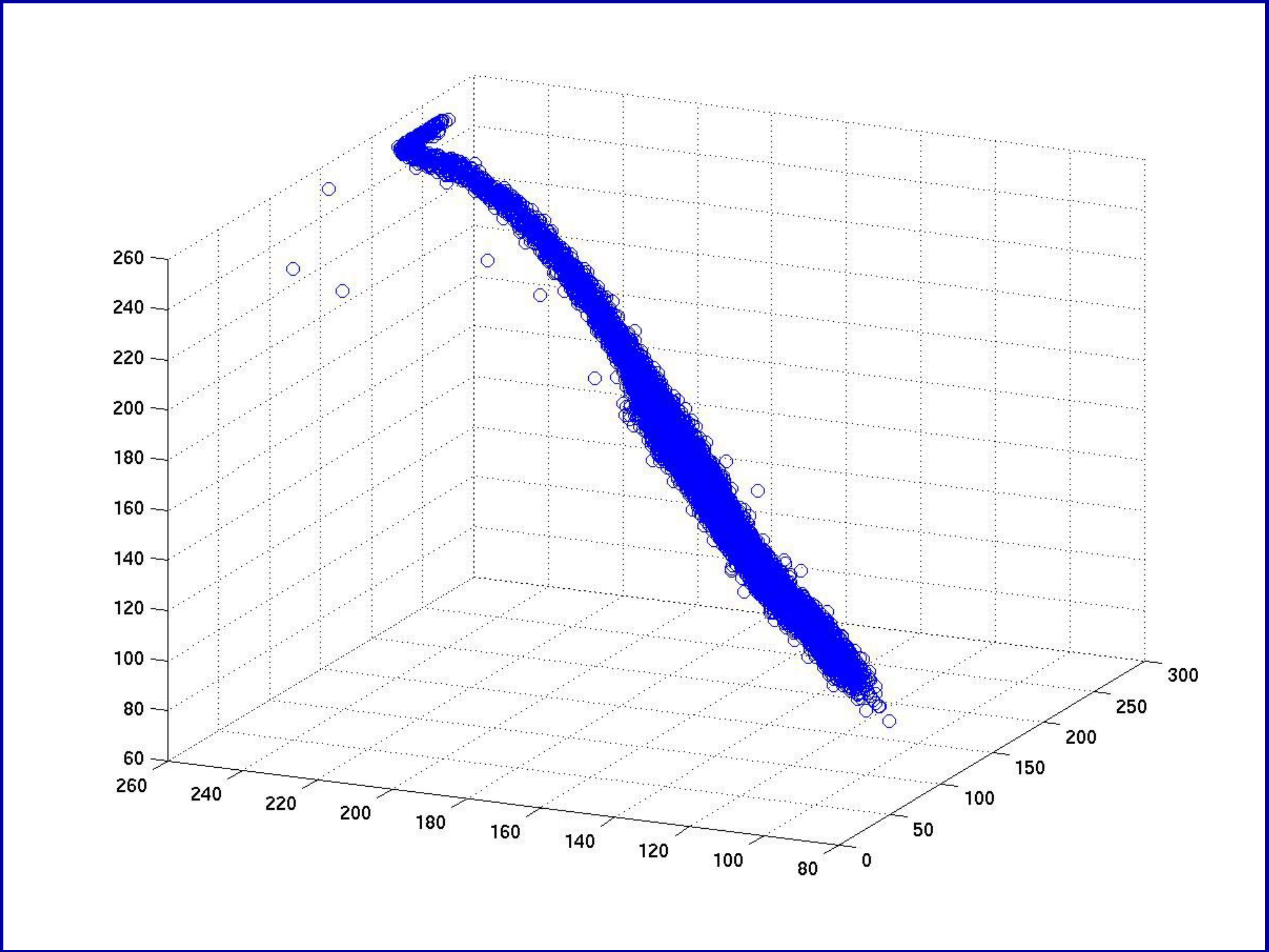
Initial Image

Desk Alone



200
180
160
140
120
100
80
60
200
180
160
140
120
100
80
100
120
140
160
180
200

Desk with Specularity



Complete Image