Lighting affects appearance

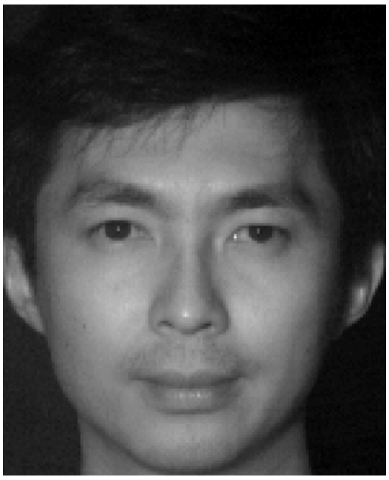


Image Normalization

- Global
  - Histogram Equalization. Make two images have same histogram. Or, pick a standard histogram, and make adjust each image to have that histogram.
    - Apply monotonic transform to intensities.
  - Additive and multiplicative normalization.
    - Subtract mean intensity, divide by total magnitude of result.
- Local
  - Normalized cross-correlation: Normalize windows and then compare with SSD.
  - Normalize intensity and first derivatives -> direction of gradient.
  - Normalize filter outputs: eg., Gabor Jets.

Histogram: H: I in R^2 -> f:Z->R. That is, computing a histogram takes in an image as input, and returns a function as output that maps integer intensity values to a frequency. In this case, f(i) = sum_x I(x)==i, where I(x)==i acts like an indicator function.

Histogram equalization: I in R^2,f: (R->R) -> J in R^2. That is, histogram equalization takes an image and a desired histogram as input, and produces a new image. We have J(x) = g(I(x)), where x is an index into the image. J(x) is a histogram equalized version of I(x) if H(J) = f (that is, the J has the desired histogram, f) and g is a montonically increasing function.

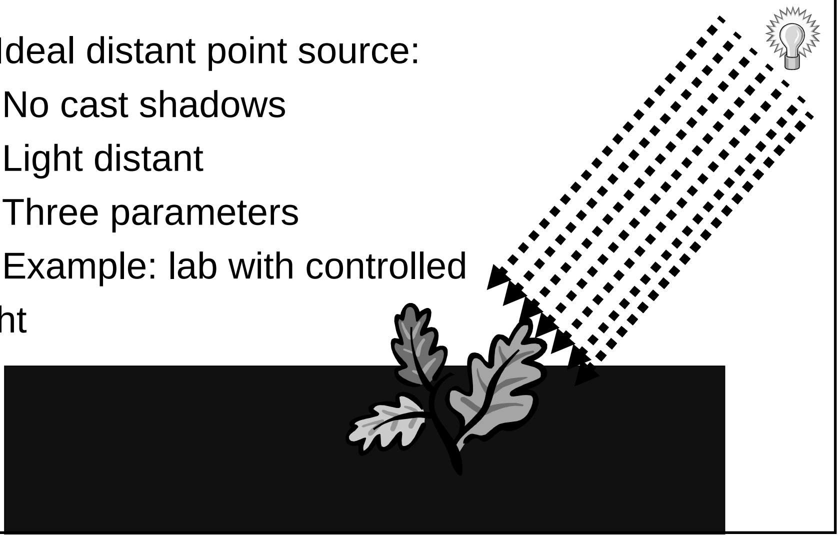
Histogram equalization undoes any montonic change to the intensities. That is, suppose h is a montonically increasing function. Suppose also that I and I' are images, such that I'(x) = h(I(x)). Then, for any target histogram, f, H(I',f) = H(I,f). That is, I and I' will be the same after histogram equalization.

Normalization. Suppose I is an image. Let mean(I) denote the mean intensity of I. Then I' = I - mean(I) is normalized to have zero mean. Let std(I') be the standard deviation of intensities in I. Then I'' = (I-mean(I))/std(I) is normalized to have zero mean and unit standard deviation. This removes additive and multiplicative changes to the image.

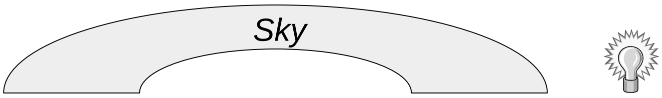
Direction of gradient. Let grad(I(x)) denote the image gradient of I at point x. Then we can represent the image with the direction of the gradient, D(x) = grad(I)/||grad(I)||. This is equivalent to normalizing the image very locally, since additive and multiplicative changes to the local image can map the intensity and magnitude of the image gradient to arbitrary values.

How do we represent light? (1)

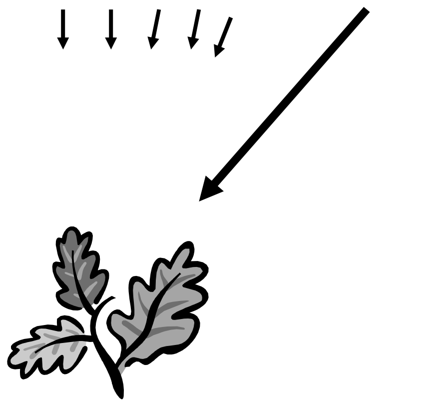
- Ideal distant point source:
  - No cast shadows
  - Light distant
  - Three parameters
  - Example: lab with controlled light

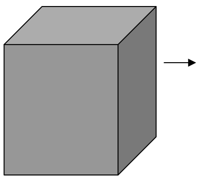


How do we represent light? (2)

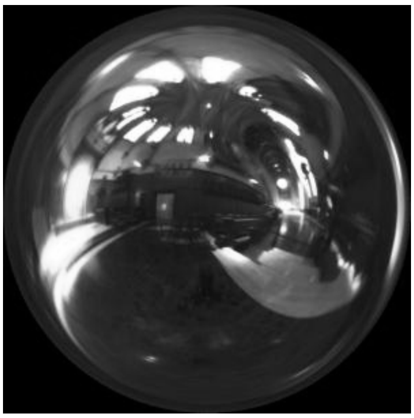


- Environment map: $I(\theta,\phi)$
  - Light from all directions
  - Diffuse or point sources
  - Still distant
  - Still no cast shadows.
  - Example: outdoors (sky and sun)





`



Lambertian + Point Source

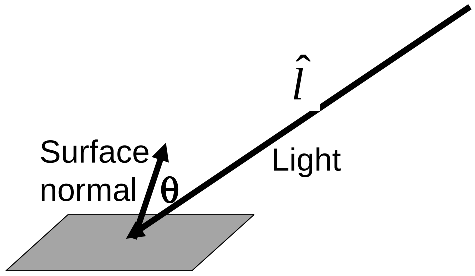
$$\vec{l} = l \bullet \vec{l} \quad \begin{cases} \vec{l} \text{ is direction of light} \\ l \text{ is intensity of light} \end{cases}$$

$$i = \max(0, \lambda(\vec{l} \bullet \hat{n})$$

i is radiance

λ is *albedo*

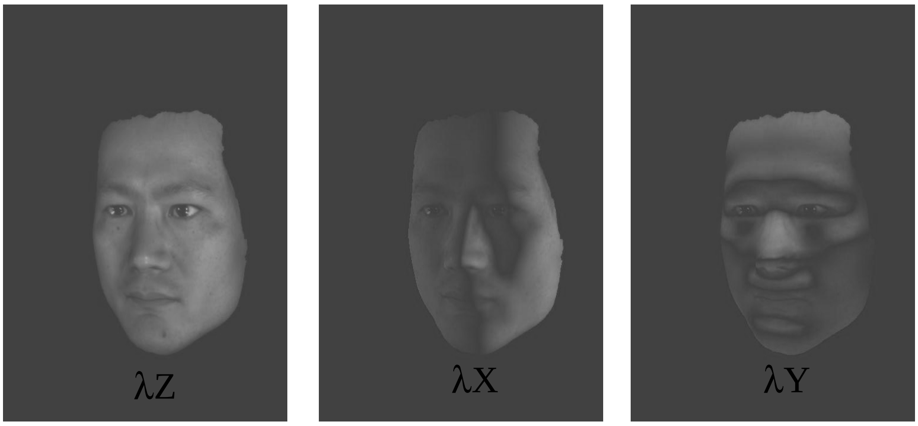
$\hat{n}$ is surface normal



Lambertian, point sources, no shadows. (Shashua, Moses)

- *Whiteboard*
- Solution linear
- Linear ambiguity in recovering scaled normals
- Lighting not known.
- Recognition by linear combinations.

Linear basis for lighting



A brief Detour: Fourier Transform, the other linear basis

- Analytic geometry gives a coordinate system for describing geometric objects.
- Fourier transform gives a coordinate system for functions.

Basis

- P=(x,y) means P = x(1,0)+y(0,1)
- Similarly:

$$f(\theta) = a_{11}\cos(\theta) + a_{12}\sin(\theta)$$
$$+ a_{21}\cos(2\theta) + a_{22}\sin(2\theta) + \ldots$$

Note, I’m showing non-standard basis, these are from basis using complex functions.

Example

$$\forall c, \exists a_1, a_2 \text{ such that :}$$
$$\cos(\theta + c) = a_1 \cos\theta + a_2 \sin\theta$$

Orthonormal Basis

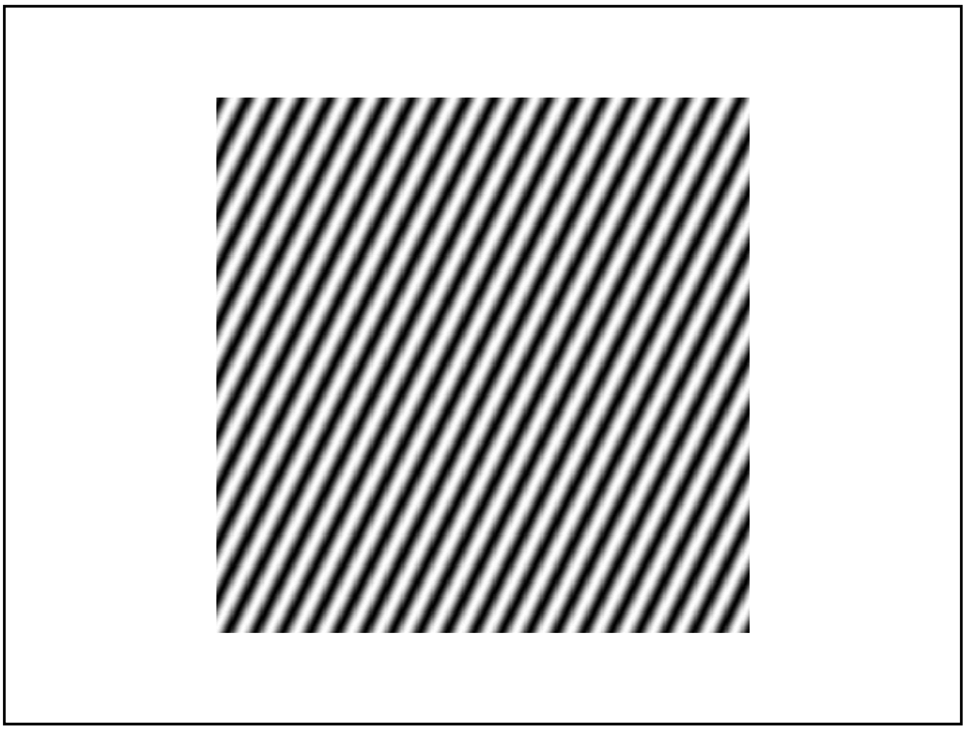
- ||(1,0)||=||(0,1)||=1
- (1,0).(0,1)=0
- Similarly we use normal basis elements eg:

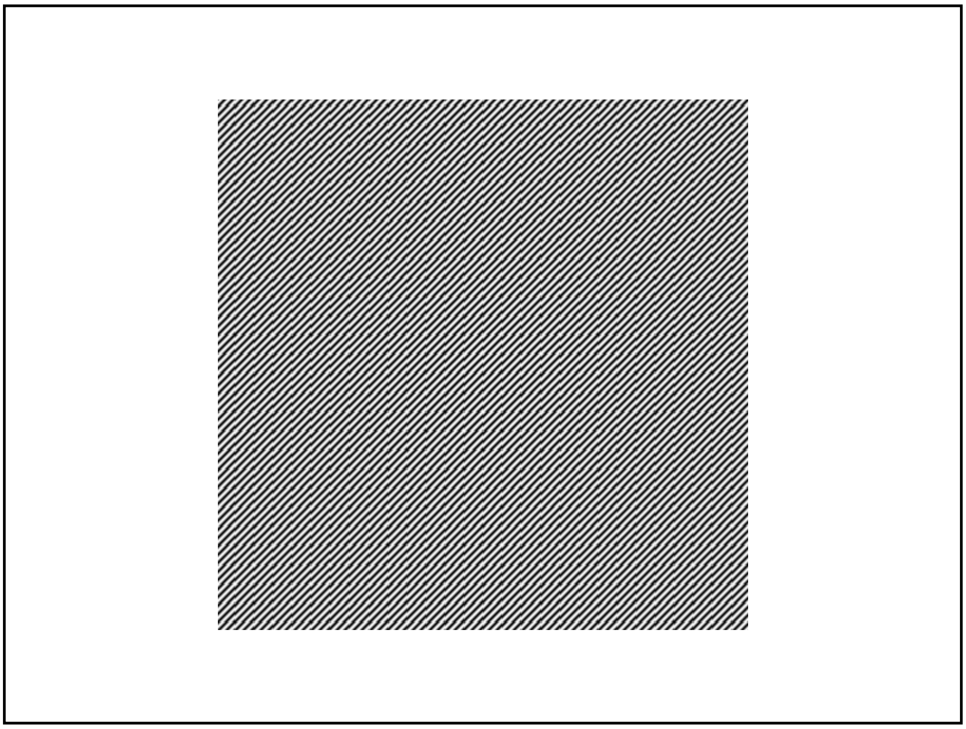
$$\frac{\cos(\theta)}{\left\|\cos(\theta)\right\|} \qquad \left\|\cos(\theta)\right\| = \sqrt{\int_0^{2\pi} \cos^2 \theta \, d\theta}$$

- While, eg:

$$\int_0^{2\pi} \cos\theta \sin\theta \; d\theta = 0$$

2D Example





Convolution

$$f(x) = g * h = \int g(x - x_0)h(x_0)dx_0$$

Imagine that we generate a point in *f* by centering *h* over the corresponding point in *g*, then multiplying *g* and *h* together, and integrating.

Convolution Theorem

$$f \otimes g = T^{-1} F * G$$

- *F,G* are transform of *f,g*

That is, *F* contains coefficients, when we write *f* as linear combinations of harmonic basis.

Examples

$$\cos\theta \otimes \cos\theta = ?$$

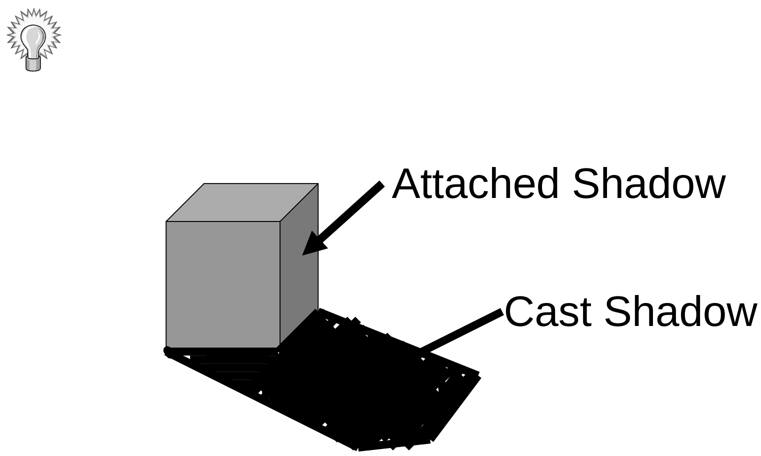
$$\cos\theta \otimes \cos 2\theta = ?$$

$$\cos\theta \otimes f = ?$$

$$(\cos\theta + .2\cos 2\theta + .1\cos 3\theta) \otimes f = ?$$

Low-pass filter removes low frequencies from signal. Hi-pass filter removes high frequencies. Examples?

Shadows



With Shadows: PCA

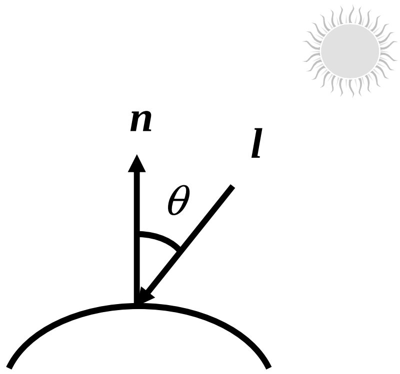
(Epstein, Hallinan and Yuille;
see also Hallinan; Belhumeur and Kriegman)

	Ball	Face	Phone	Parrot
#1	48.2	53.7	67.9	42.8
#3	94.4	90.2	88.2	76.3
#5	97.9	93.5	94.1	84.7
#7	99.1	95.3	96.3	88.5
#9	99.5	96.3	97.2	90.7

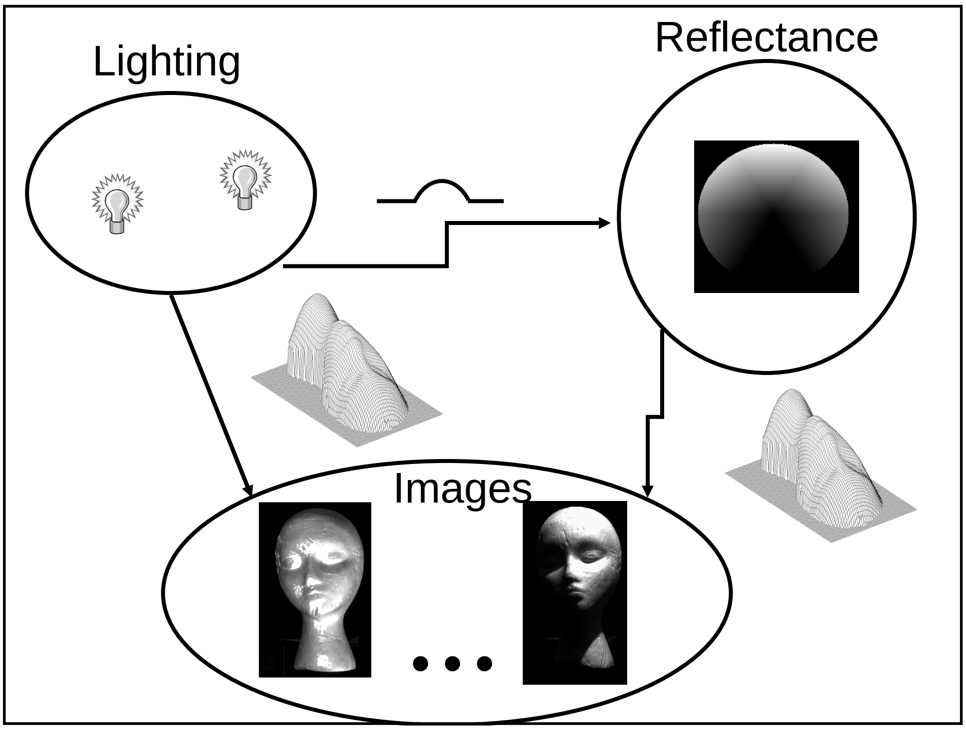
Dimension: 5 ± 2D

Domain

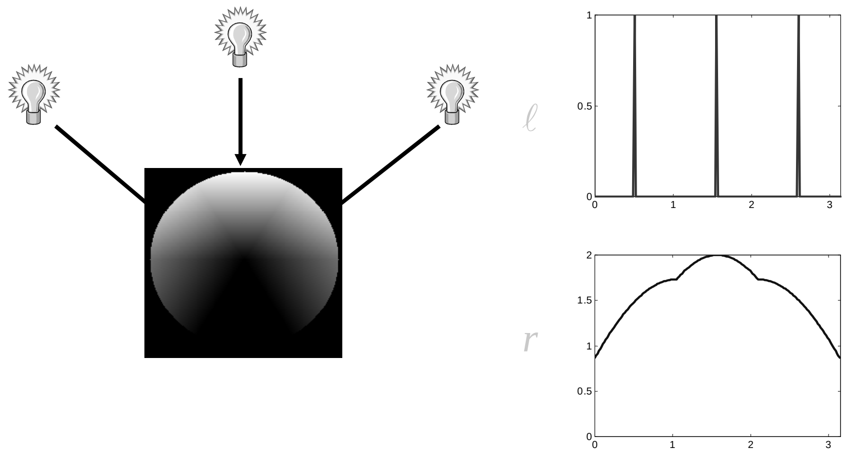
- Lambertian
- Environment map

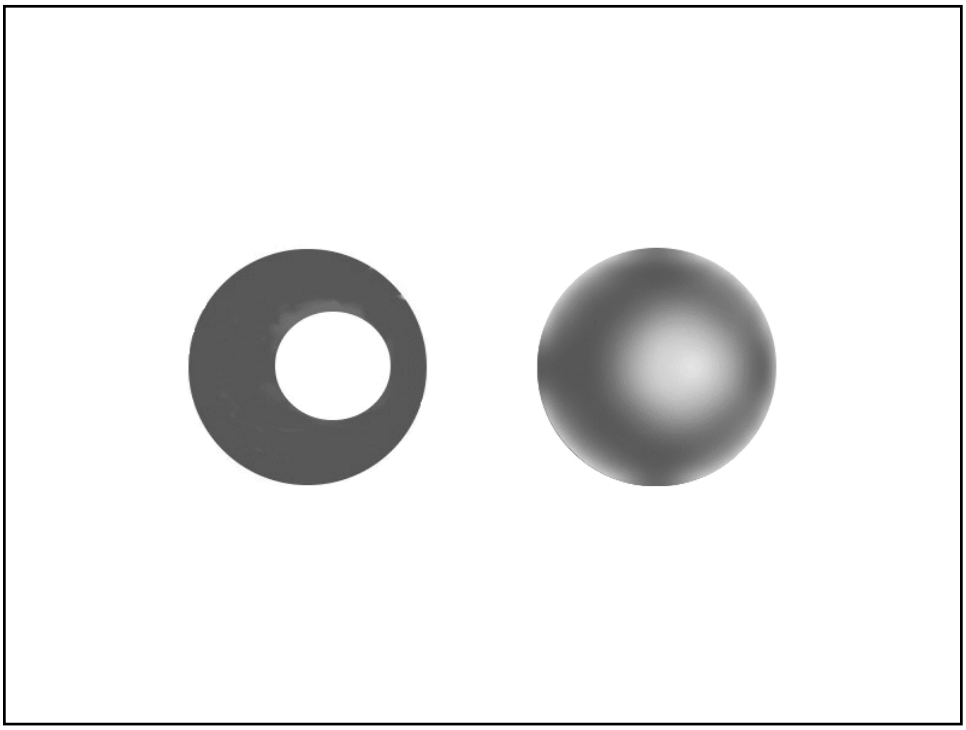


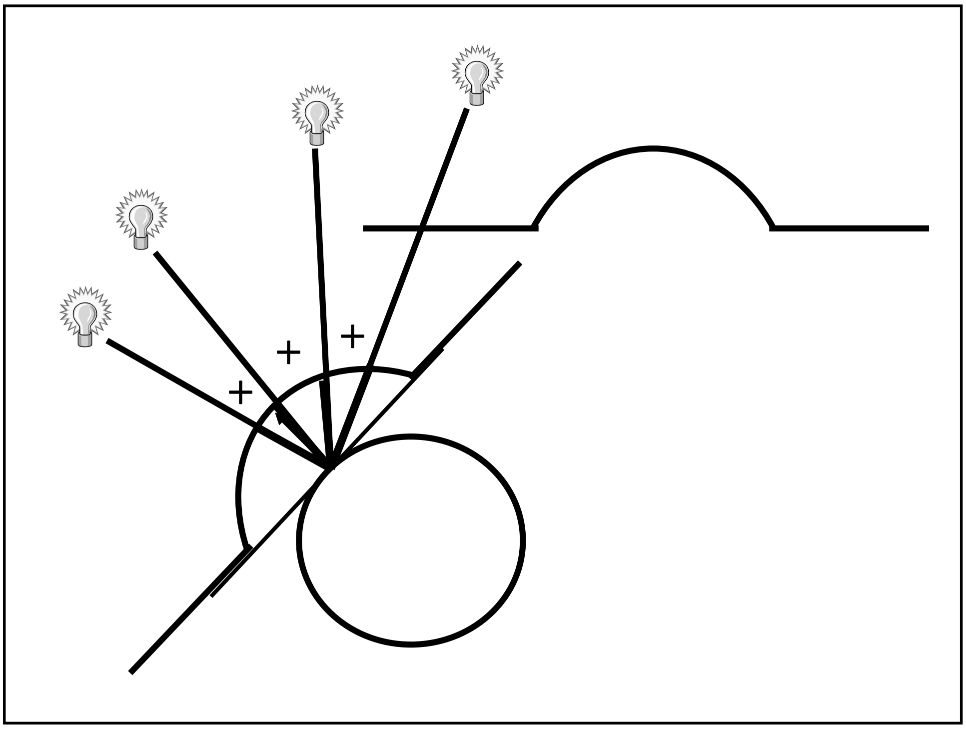
$l\lambda$max $(\cos\theta, 0)$



Lighting to Reflectance: Intuition







Spherical Harmonics

- Orthonormal basis, h_{nm}, for functions on the sphere.
- *n*'th order harmonics have 2*n*+1 components.
- Rotation = phase shift (same *n,* different *m*).
- In space coordinates: polynomials of degree *n.*
- S.H. used for BRDFs (Cabral et al.; Westin et al;). (See also Koenderink and van Doorn.)

$$h_{nm}(\theta,\phi) = \sqrt{\frac{(2n+1)}{4\pi}\frac{(n-m)!}{(n+m)!}}P_{nm}(\cos\theta)e^{im\phi}$$

$$P_{nm}(z) = \frac{(1-z^2)^{m/2}}{2^n n!}\frac{d^{n+m}}{dz^{n+m}}(z^2-1)^n$$

S.H. analog to convolution theorem

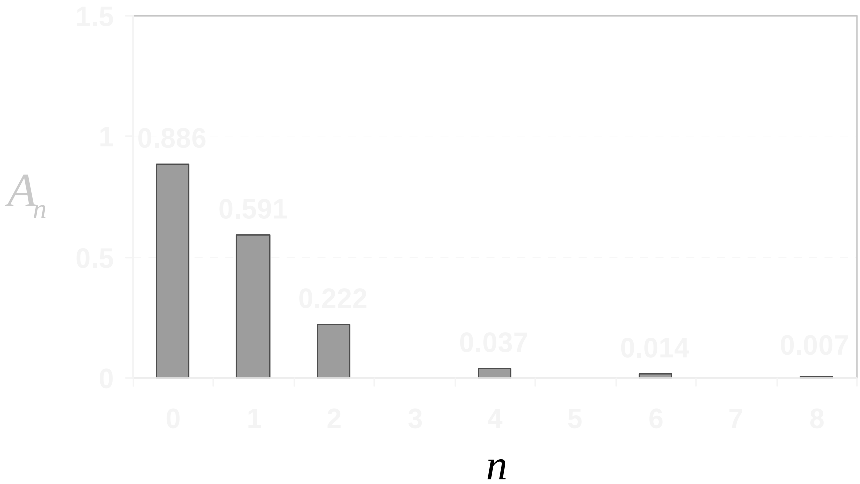
- Funk-Hecke theorem: "Convolution" in function domain is multiplication in spherical harmonic domain.
- *k* is low-pass filter.

Harmonic Transform of Kernel

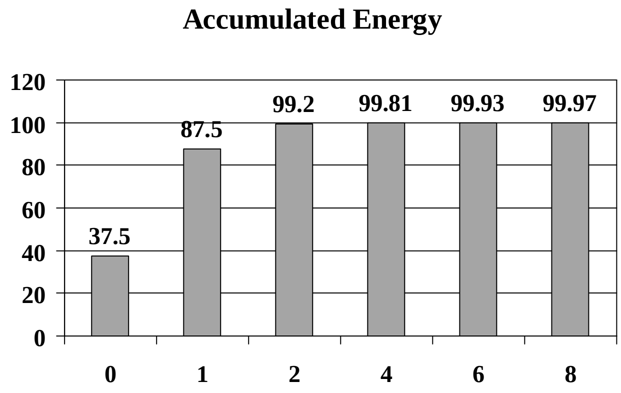
$$k(\theta) = \max(\cos\theta, 0) = \sum_{n=0}^{\infty} k_n h_{n0}$$

$$k_n = \begin{cases} \frac{\sqrt{\pi}}{2} & n = 0 \\ \sqrt{\frac{\pi}{3}} & n = 1 \\ (-1)^{\frac{n}{2}+1} \frac{(n-2)!\sqrt{(2n+1)\pi}}{2^n(\frac{n}{2}-1)!(\frac{n}{2}+1)!} & n \geq 2, \text{even} \\ 0 & n \geq 2, \text{odd} \end{cases}$$

Amplitudes of Kernel



Energy of Lambertian Kernel in low order harmonics



Reflectance Functions Near Low-dimensional Linear Subspace

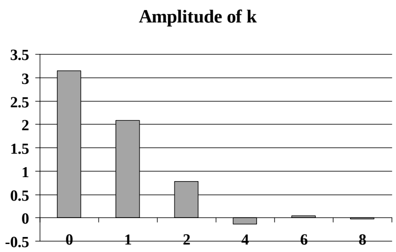
$$r = k * l = \sum_{n=0} \sum_{m=-n}^{n} (K_{nm} L_{nm}) h_{nm}$$

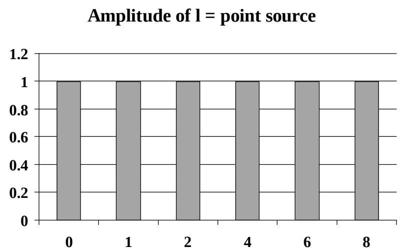
$$\approx \sum_{n=0}^{2} \sum_{m=-n}^{n} (K_{nm} L_{nm}) h_{nm}$$

Yields 9D linear subspace.

How accurate is approximation? Point light source

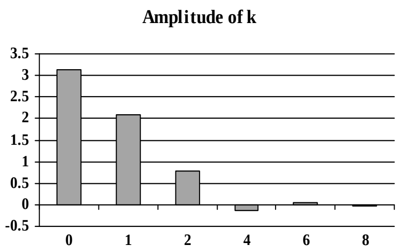
$$r = k * l = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} (K_{nm} L_{nm}) h_{nm} \approx \sum_{n=0}^{2} \sum_{m=-n}^{n} (K_{nm} L_{nm}) h_{nm}$$

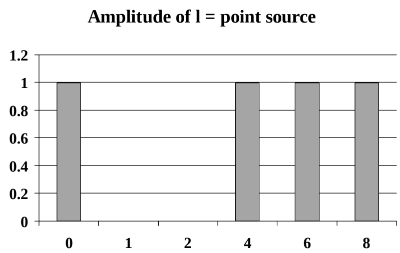




9D space captures 99.2% of energy

How accurate is approximation? (2) Worst case.



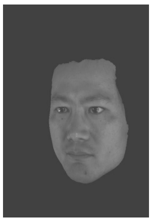


- DC component as big as any other.
- 1st and 2nd harmonics of light could have zero energy

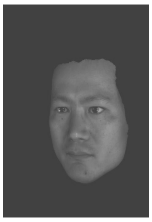
9D space captures 98% of energy

Forming Harmonic Images

$b_{nm}(p) = \lambda r_{nm}(X,Y,Z)$



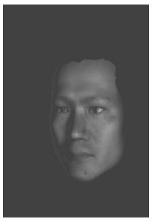
λ



λZ

λX

λY



$2\lambda(Z^2-X^2-Y^2)$

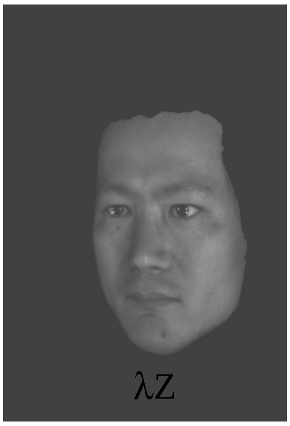
$\lambda(X^2-Y^2)$

λXY

λXZ

λYZ

Compare this to 3D Subspace



Accuracy of Approximation of Images

- Normals present to varying amounts.
- Albedo makes some pixels more important.
- *Worst case approximation arbitrarily bad.*
- *“Average” case approximation should be good.*