Announcements

- Big mistake on hint in problem 1 (I'm very sorry).

$$\int_{-\infty}^{\infty} e^{-\frac{1}{2}Ax^2+Zx}\,dx = \sqrt{\frac{2\pi}{A}}e^{\frac{Z^2}{2A}} \qquad \int_{-\infty}^{\infty} e^{-\frac{1}{2}Ax^2+Zx}\,dx = \sqrt{\frac{2\pi}{A}}e^{\frac{Z^2}{2A}}$$

Announcements

- On 2e, use

I = [zeros(1,50), 9*ones(1,10), zeros(1,10), 3*ones(1,40), zeros(1,50)].

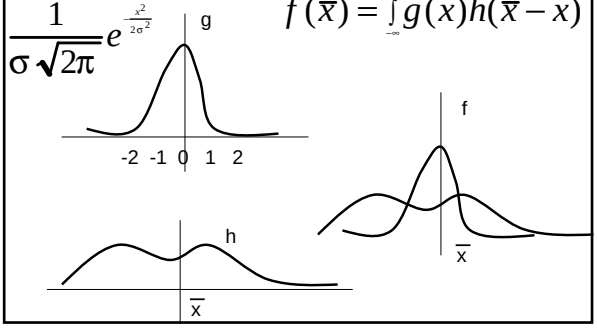
Result will be more interesting. (If you used I in 2c already, that's ok).

Announcements

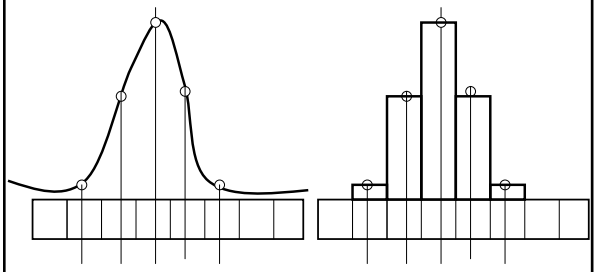
- Best not to use built in code **conv** or **fspecial**. These don't give you easy control needed for assignment.

Problem Set 2: Convolution

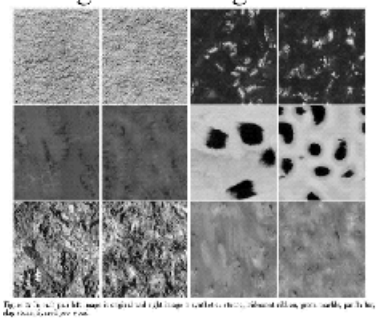
$$\frac{1}{\sigma\sqrt{2\pi}}e^{\frac{x^2}{2\sigma^2}} \qquad f(\bar{x}) = \int_{-\infty}^{\infty} g(x)h(\bar{x}-x)$$



PS 2: Discrete Filter

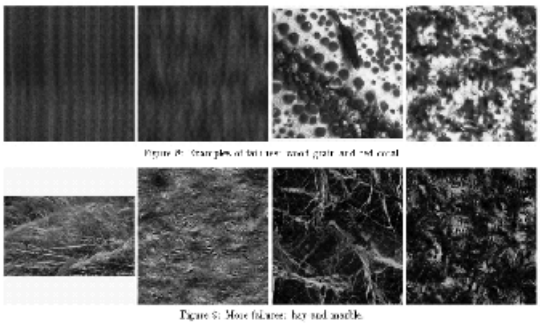


Bergen and Heeger results



From Tuesday

Bergen and Heeger failures



Markov Model

- Captures local dependencies.
  - Each pixel depends on neighborhood.
- Example, 1D first order model

P(p1, p2, …pn) =
P(p1)*P(p2|p1)*P(p3|p2,p1)*…
= P(p1)*P(p2|p1)*P(p3|p2)*P(p4|p3)*…

Example 1st Order Markov Model

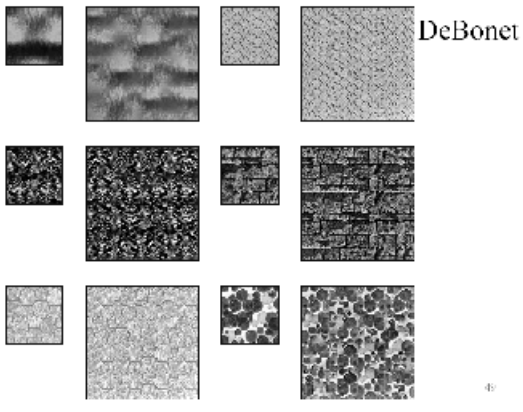
- Each pixel is like neighbor to left + noise with some probability.

Matlab

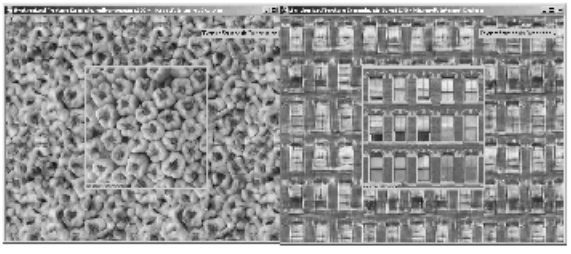
- These capture a much wider range of phenomena.

There are dependencies in Filter Outputs

- Edge
  - Filter responds at one scale, often does at other scales.
  - Filter responds at one orientation, often doesn't at orthogonal orientation.
- Synthesis using wavelets and Markov model for dependencies:
  - DeBonet and Viola
  - Portilla and Simoncelli



Portilla and Simoncelli

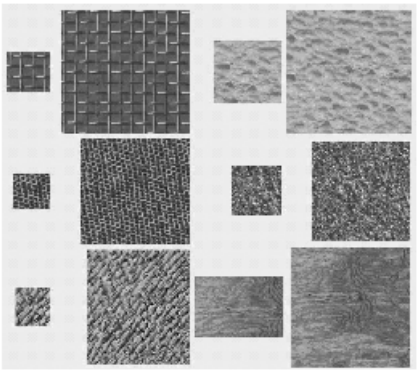


We can do this without filters

- Each pixel depends on neighbors.

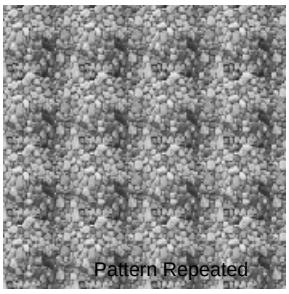
1. As you synthesize, look at neighbors.
2. Look for similar neighborhood in sample texture.
3. Copy pixel from that neighborhood.
4. Continue.

Efros and Leung

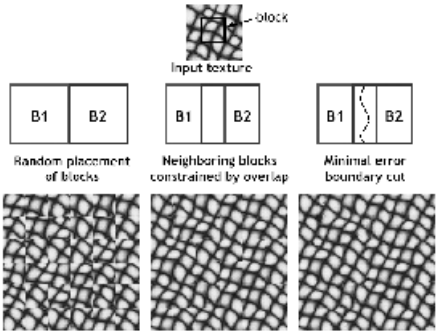


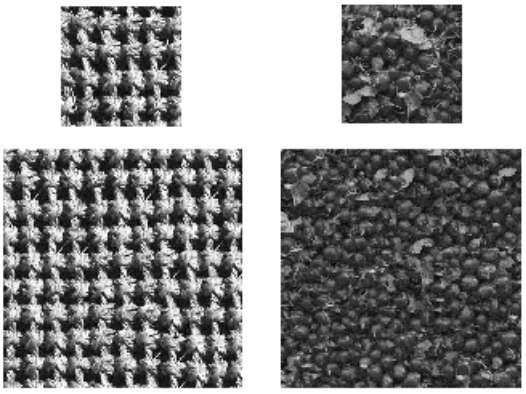
This is like copying, but not just repetition

Photo

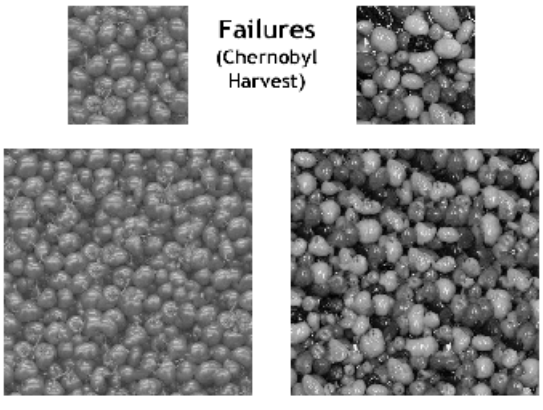


With Blocks





Failures (Chernobyl Harvest)



Conclusions

- Model texture as generated from random process.
- Discriminate by seeing whether statistics of two processes seem the same.
- Synthesize by generating image with same statistics.

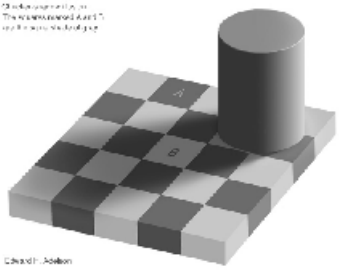
To Think About

- 3D effects
  - Shape: Tiger's appearance depends on its shape.
  - Lighting: Bark looks different with light angle
- Given pictures of many chairs, can we generate a new chair?

Lightness

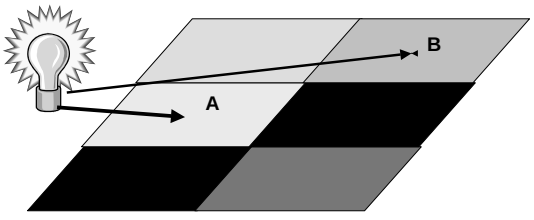
- Digression from boundary detection
- Vision is about recovery of properties of scenes: lightness is about recovering material properties.
  - Simplest is how light or dark material is (ie., its *reflectance*).
  - We'll see how boundaries are critical in solving other vision problems.

Basic problem of lightness



Luminance (amount of light striking the eye) depends on *illuminance* (amount of light striking the surface) as well as reflectance.

Basic problem of lightness



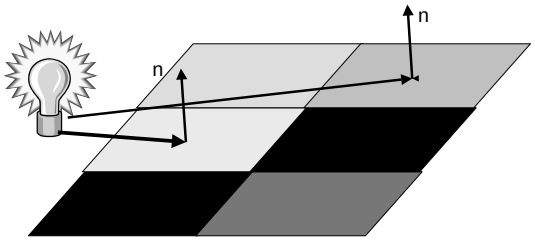
Is B darker than A because it reflects a smaller proportion of light, or because it's further from the light?

Planar, Lambertian material.

$L = r*\cos(\theta)e$ where r is reflectance (aka albedo)

θ is angle between light and n

e is illuminance (strength of light)



If we combine θ and e at a point into E(x,y) then:

$L(x,y) = R(x,y)*E(x,y)$

L(x,y) = R(x,y)*E(x,y)

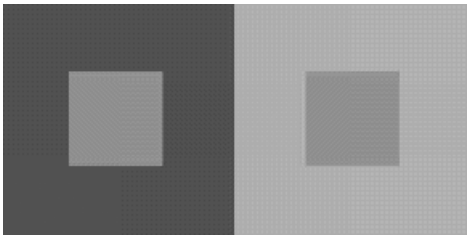
Can think of E as appearance of white paper with given illuminance.

R is appearance of planar object under constant lighting.

L is what we see.

Problem: We measure L, we want to recover R. How is this possible?

Answer: We must make additional assumptions.



Simultaneous contrast effect

Illusions

- Seems like visual system is making a mistake.
- But, perhaps visual system is making assumptions to solve underconstrained problem; illusions are artificial stimuli that reveal these assumptions.

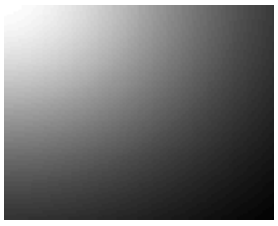
Assumptions

- Light is slowly varying
  - This is reasonable for planar world: nearby image points come from nearby scene points with same surface normal.
- Within an object reflectance is constant or slowly varying.
- Between objects, reflectance varies suddenly.

This is sometimes called the Mondrian world.

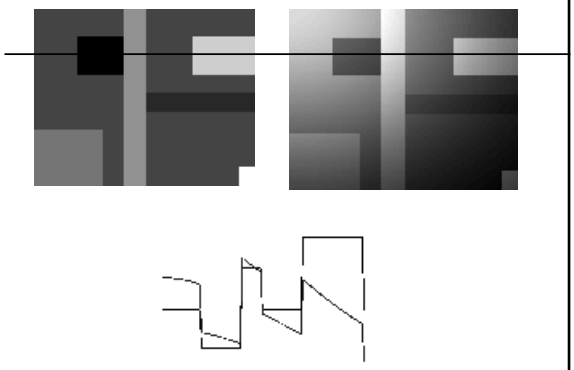
L(x,y) = R(x,y)*E(x,y)

• Formally, we assume that illuminance, E, is low frequency.



L(x,y) = R(x,y)*E(x,y)

Smooth variations in image due to lighting, sharp ones due to reflectance.



So, we remove slow variations from image. Many approaches to this. One is:

- Log(L(x,y)) = log(R(x,y)) + log(E(x,y))
- Hi-pass filter this, (say with derivative).
  - •Why is derivative hi-pass filter?

  d sin(nx)/dx = ncos(nx). Frequency n is amplified by a factor of n.
- Threshold to remove small low-frequencies.
- Then invert process; take integral, exponentiate.

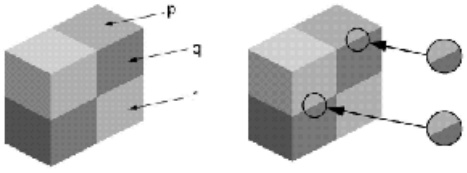
Reflectances | Reflectances* Lighting | Restored Reflectances

(Note that the overall scale of the reflectances is lost because we take derivative then integrate)

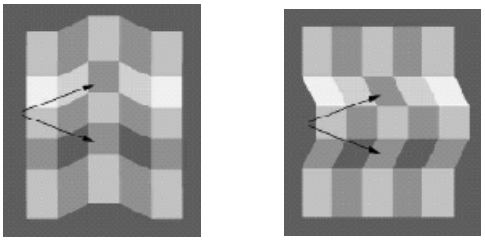
These operations are easy in 1D, tricky in 2D.

- For example, in which direction do you integrate?

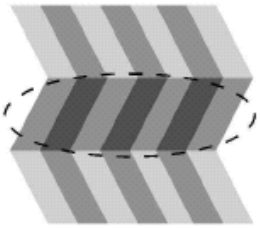
Many techniques exist.



These approaches fail on 3D objects, where illuminance can change quickly as well.



Our perceptions are influenced by 3D cues.



To solve this, we need to compute reflectance in the *right* region. This means that lightness depends on surface perception, ie., a different kind of boundary detection.