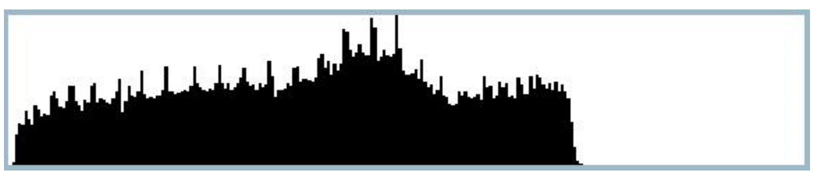
Photometric Processing

Histogram

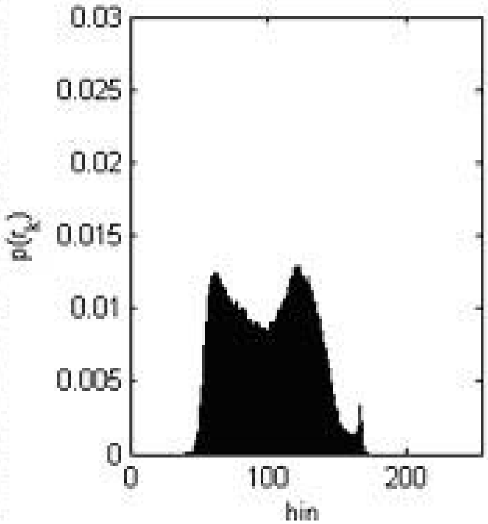
- Probability distribution of the different grays in an image

$$p(x_i) = \frac{n_i}{n}$$



Contrast Enhancement

- Limited gray levels are used
- Hence, low contrast
- Enhance contrast

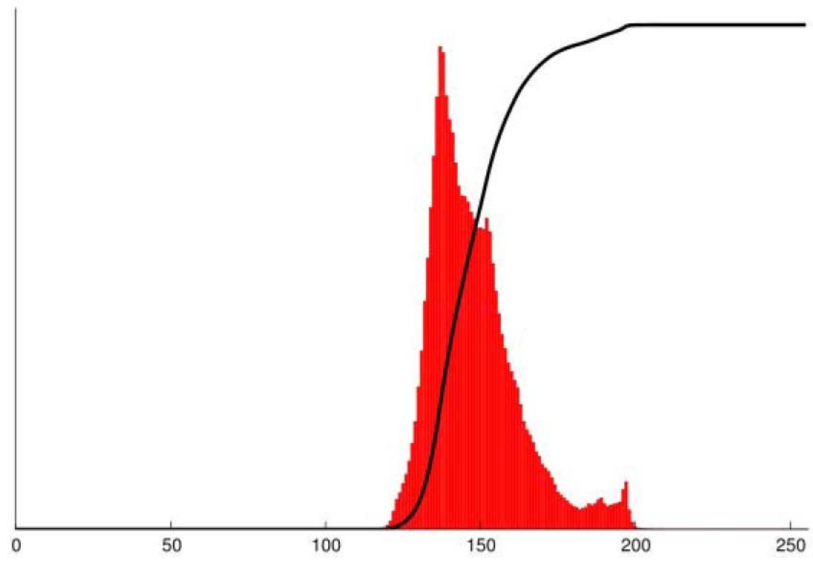


Histogram Stretching

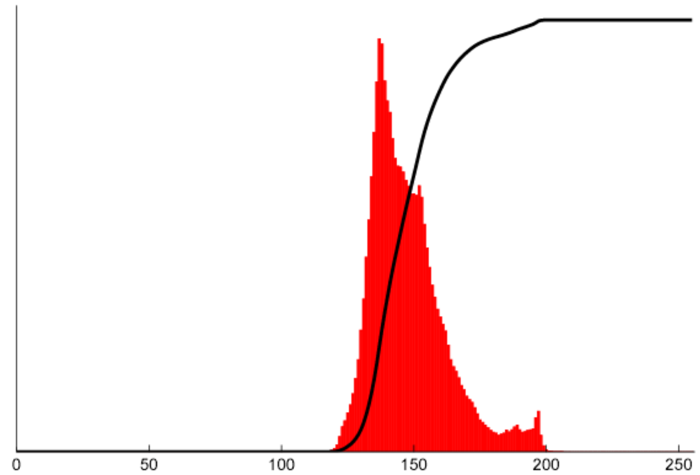
$$c(i) = \sum_{j=0}^{i} p(x_j)$$

- Monotonically increasing function between 0 and 1
- c(0) = 0
- c(1) = 1

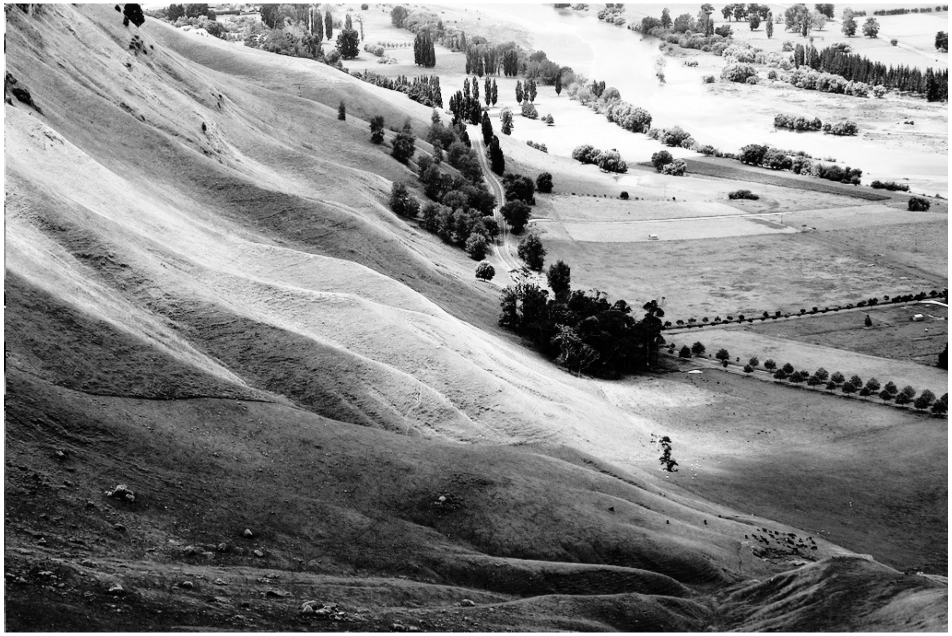
$$y_i = T(x_i) = c(i)$$

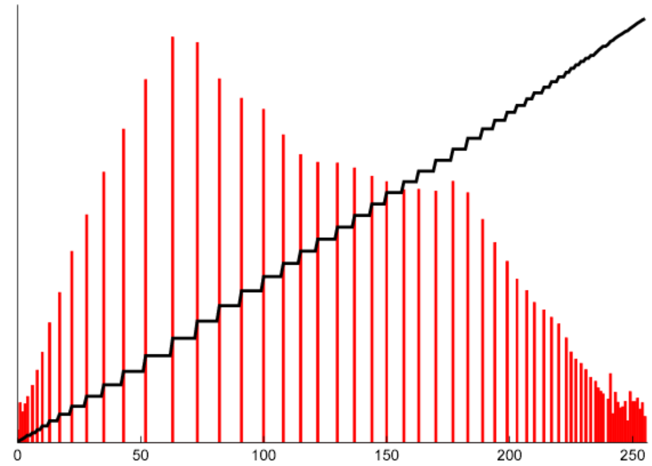


Results



Results



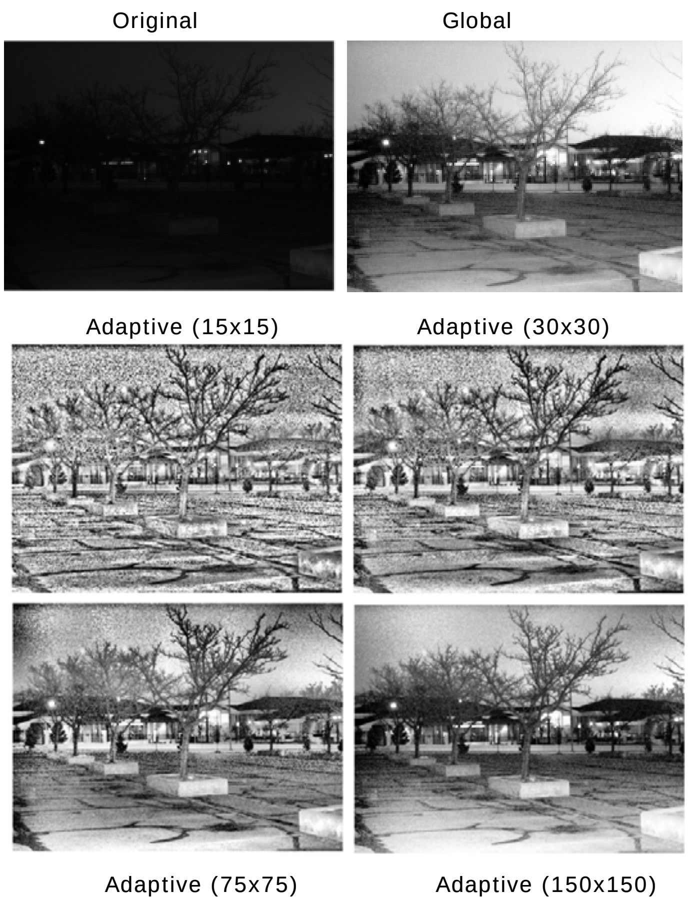


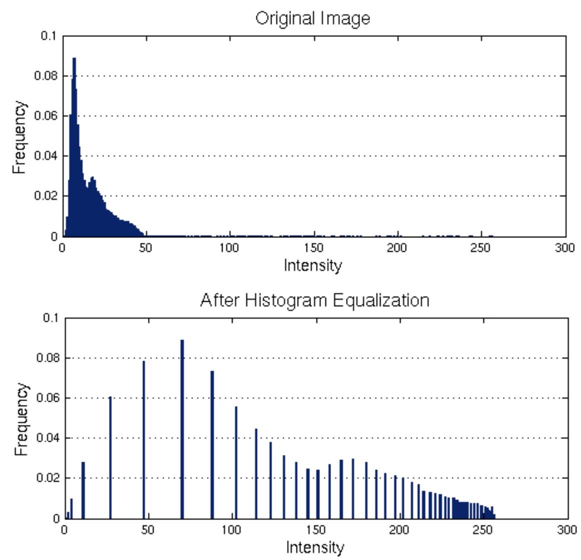
Burn out effects

Adaptive Histogram Stretching

- Choose a neighborhood
- Apply histogram equalization to the pixels in that window
- Replace the center pixel with the histogram equalized value
- Do this for all pixels
- Compute intensive
- Leads to noise

Results


Original
Global
Adaptive (15x15)
Adaptive (30x30)
Adaptive (75x75)
Adaptive (150x150)

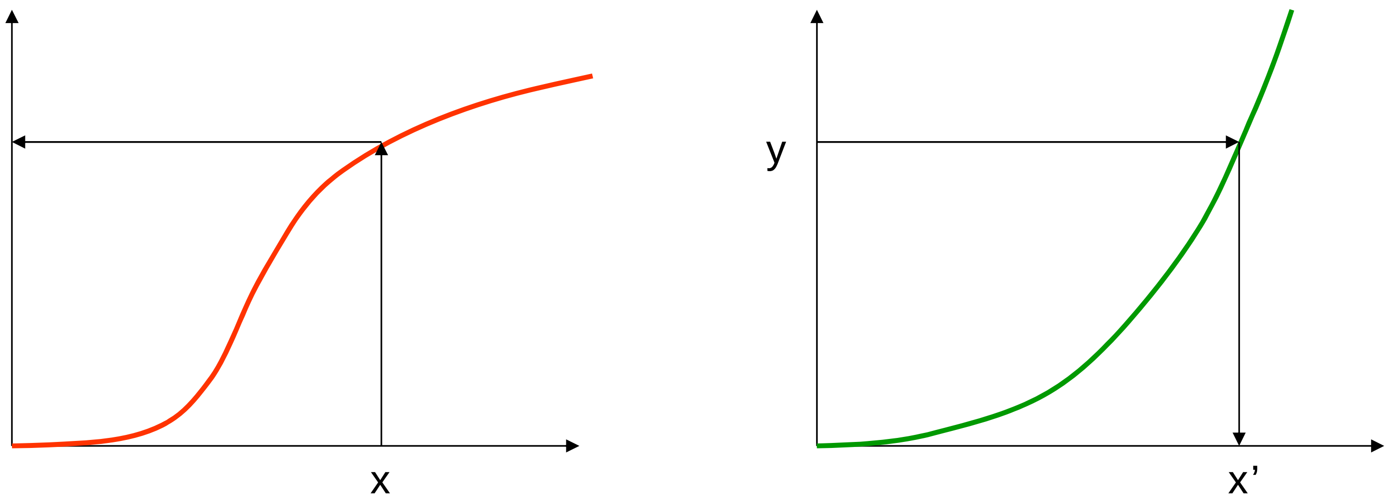

Original Image
Frequency
Intensity
After Histogram Equalization
Frequency
Intensity

Histogram Matching

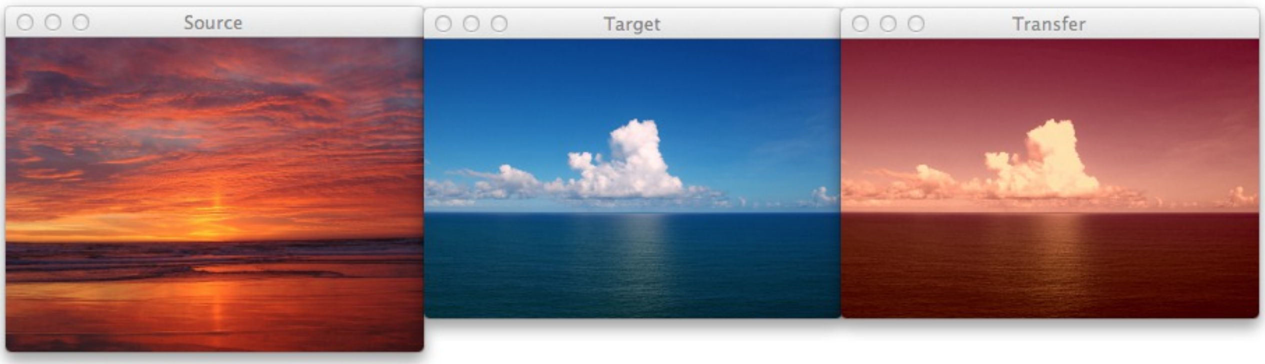
Histogram 1

Histogram 2

$$c(i) = \sum_{j=0}^{i} p(x_j)$$



Appearance Transfer



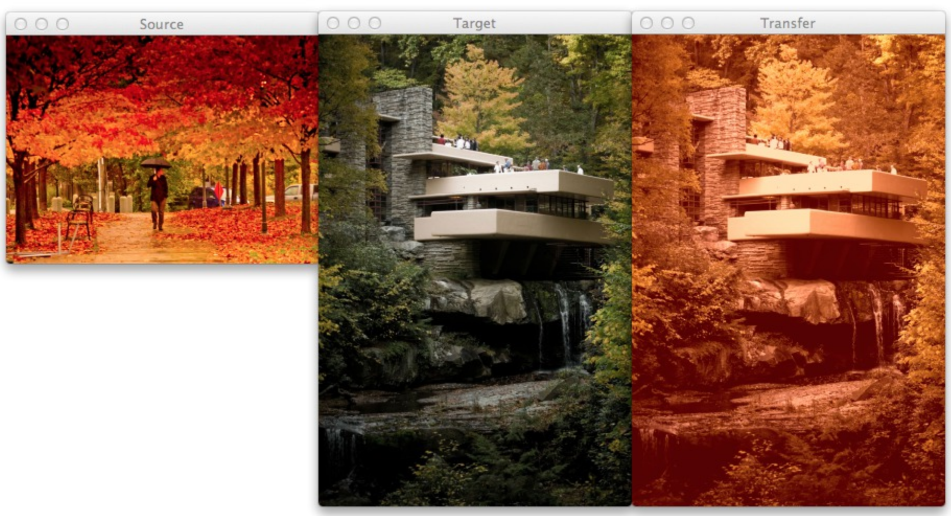
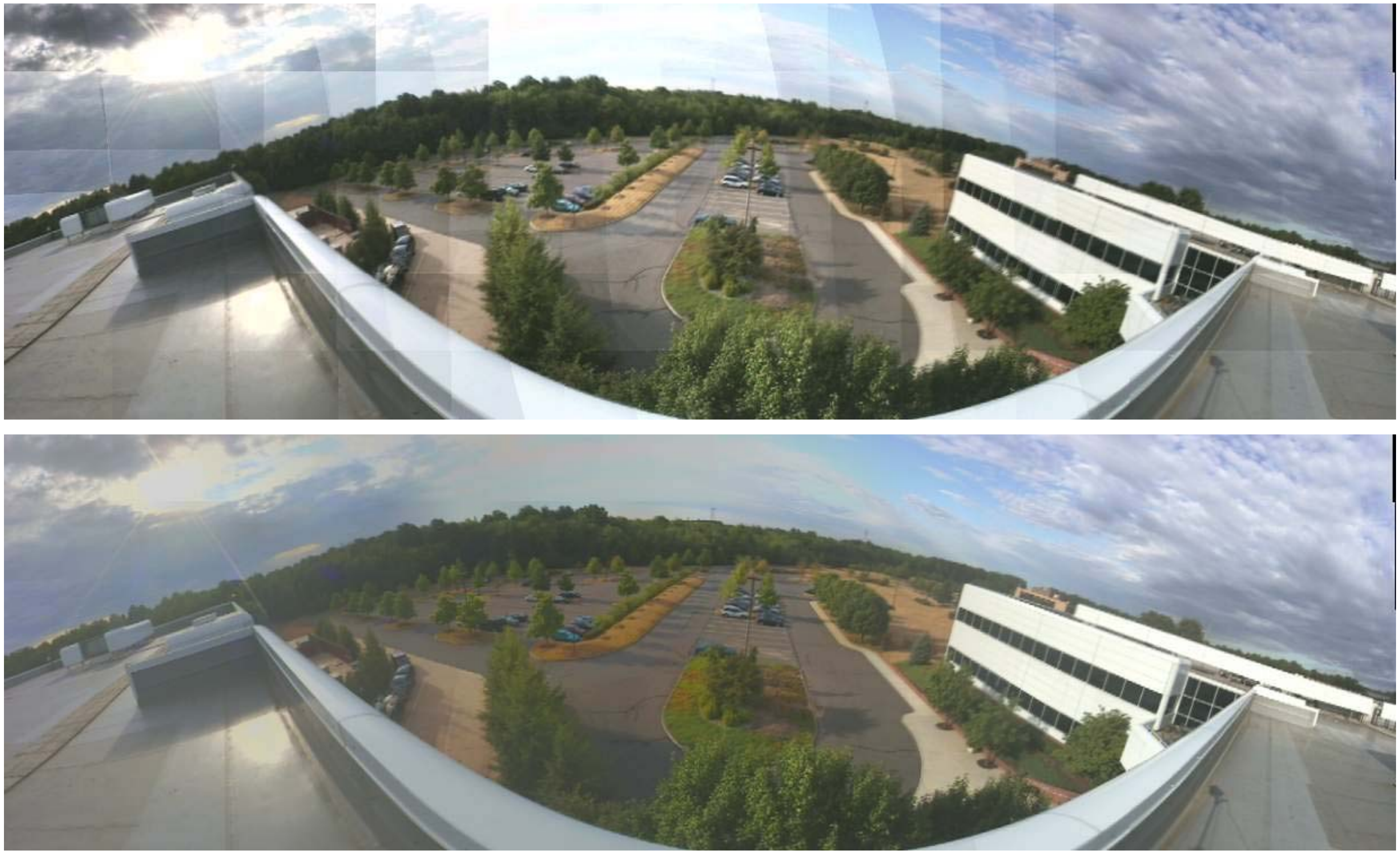


Image Compositing



Mosaic Blending

Image Compositing

Compositing Procedure

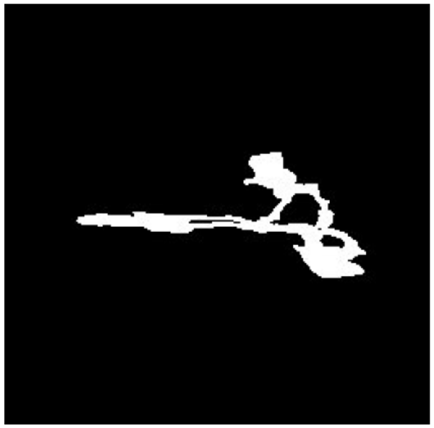
1. Extract Sprites (e.g using Intelligent Scissors in Photoshop)




2. Blend them into the composite (in the right order)

Composite by
David Dewey

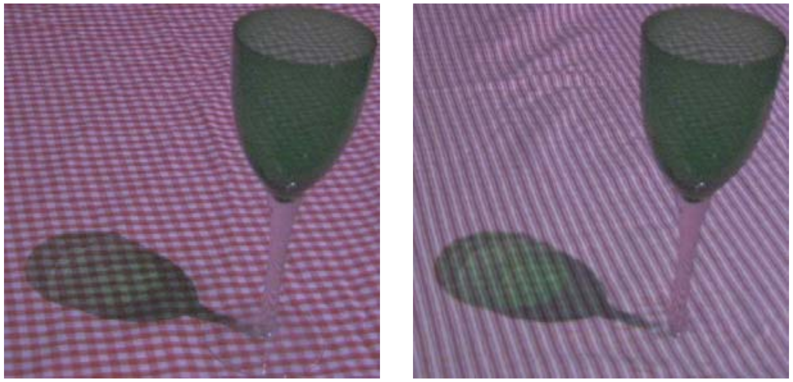
Replacing pixels rarely works



Binary mask

Problems: boundries & transparency (shadows)

Two Problems:



Semi-transparent objects

Pixels too large

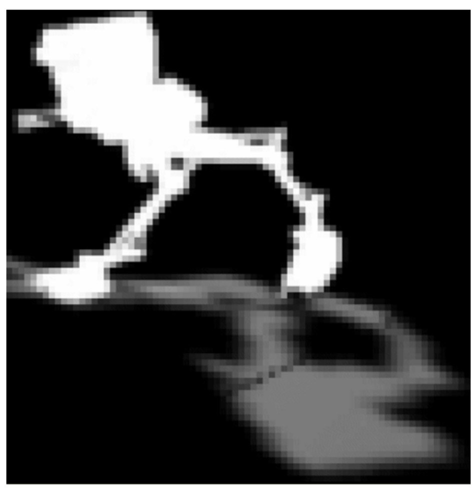
Alpha Channel

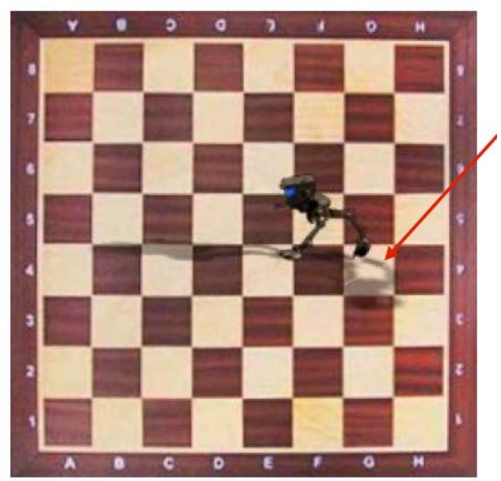
- Add one more channel:
  - Image(R,G,B,alpha)
- Encodes transparency (or pixel coverage):
  - Alpha = 1: opaque object (complete coverage)
  - Alpha = 0: transparent object (no coverage)
  - 0<Alpha<1: semi-transparent (partial coverage)
- Example: alpha = 0.3

Alpha Blending

$$I_{comp} = \alpha I_{fg} + (1-\alpha) I_{bg}$$

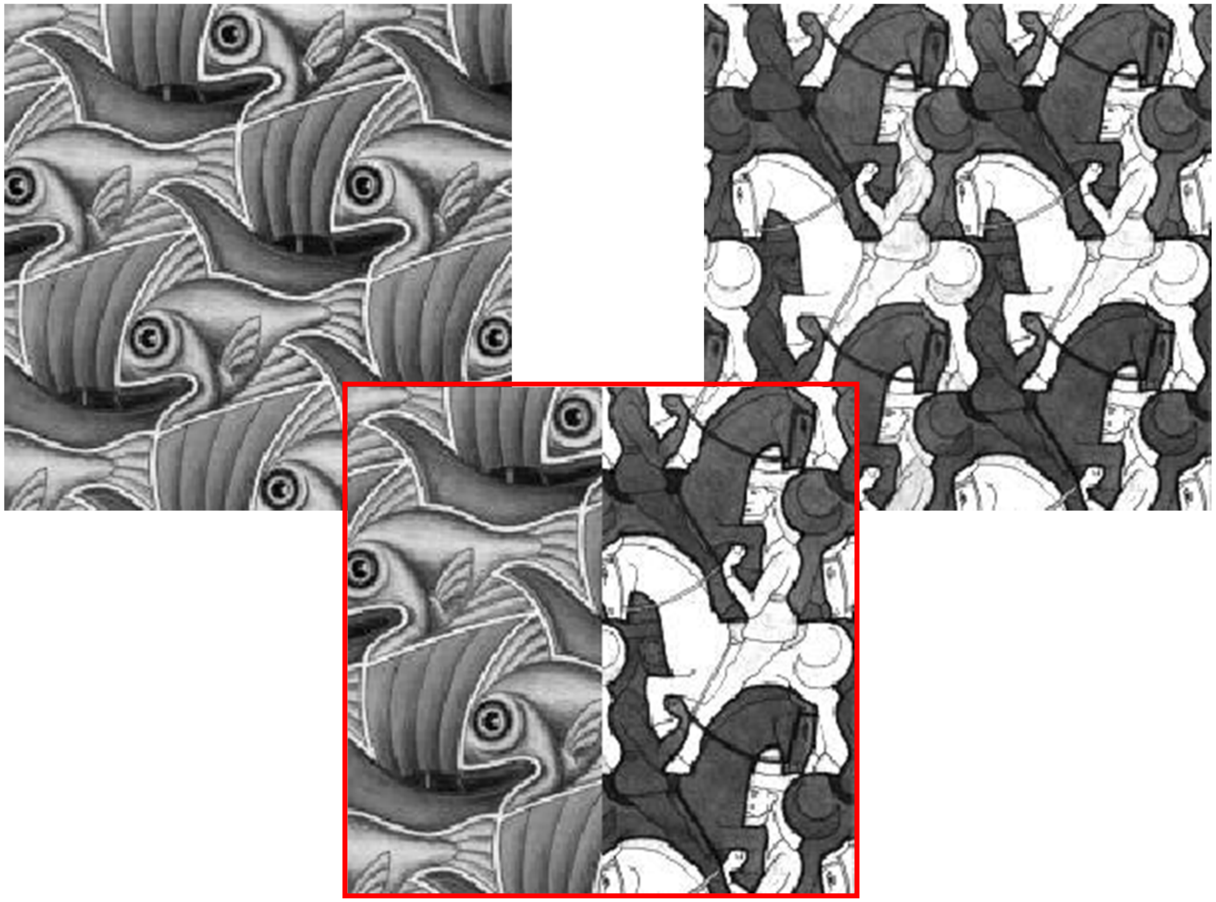
alpha
mask





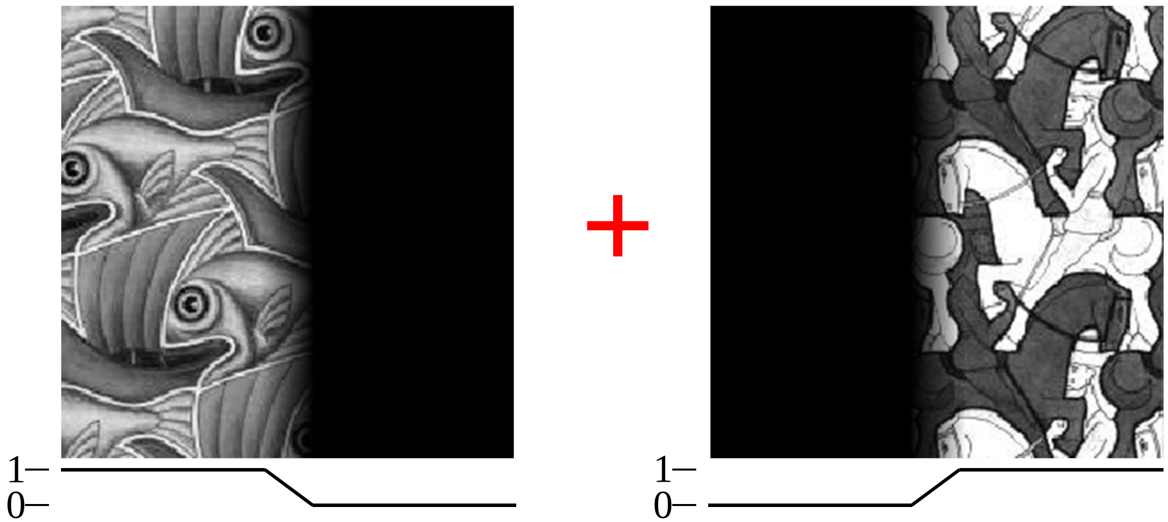
shadow

Alpha Hacking…



No physical interpretation, but it smoothes the seams

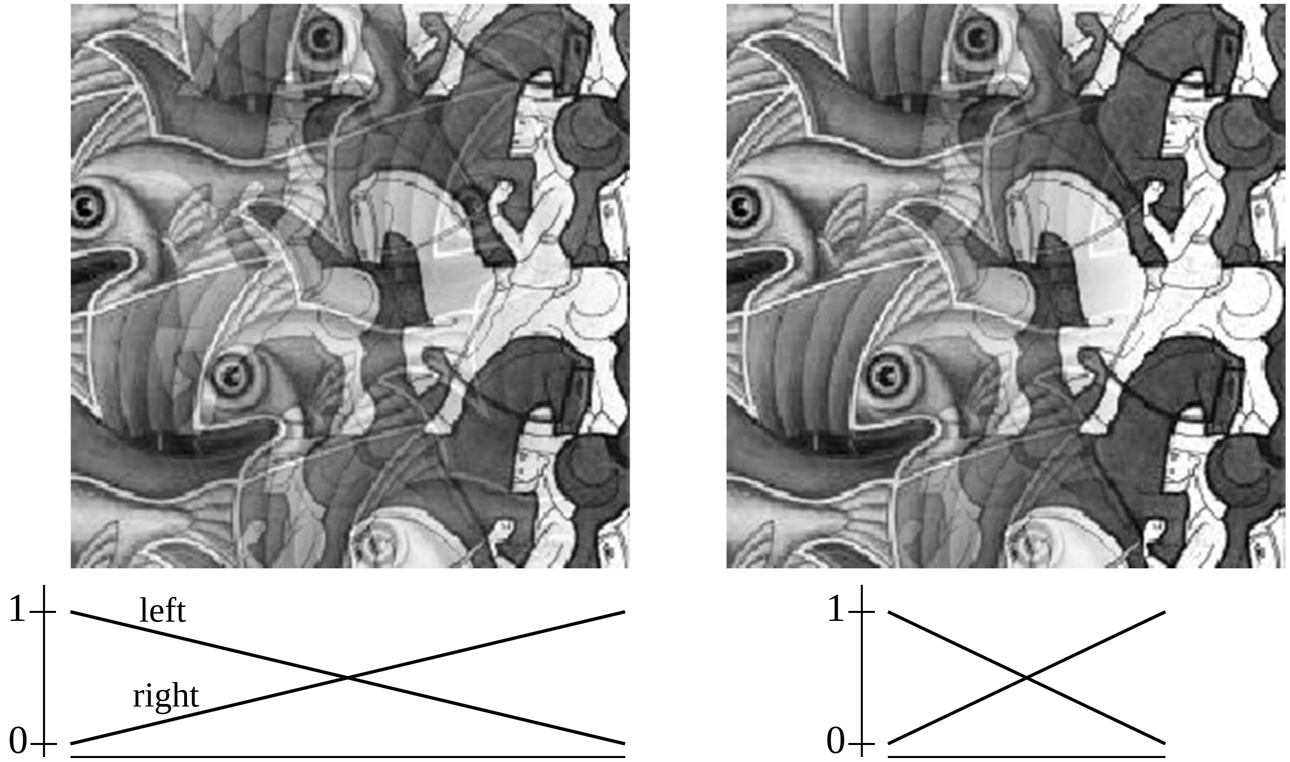
Feathering



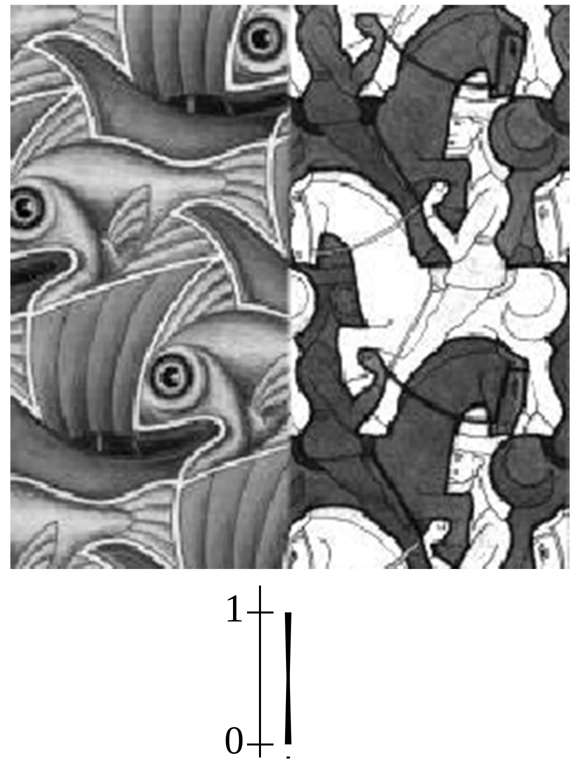
Encoding as transparency

$$I_{blend} = \alpha I_{left} + (1-\alpha) I_{right}$$

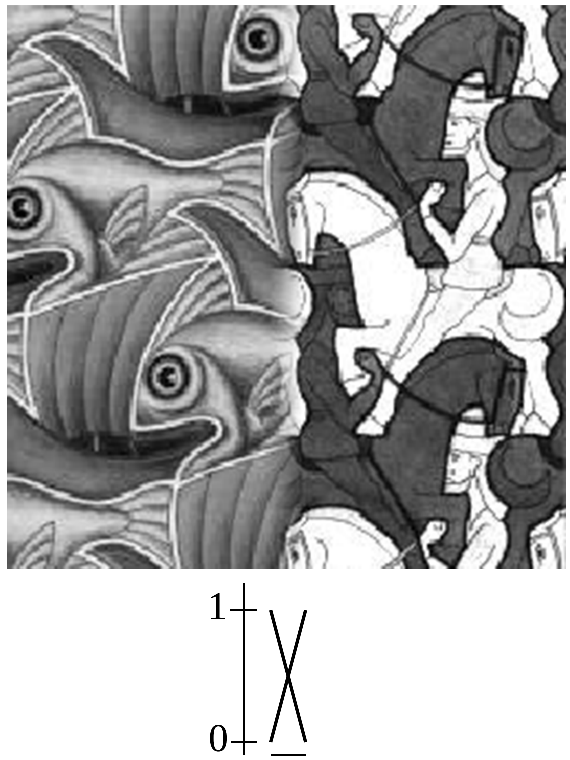
Affect of Window Size



Affect of Window Size

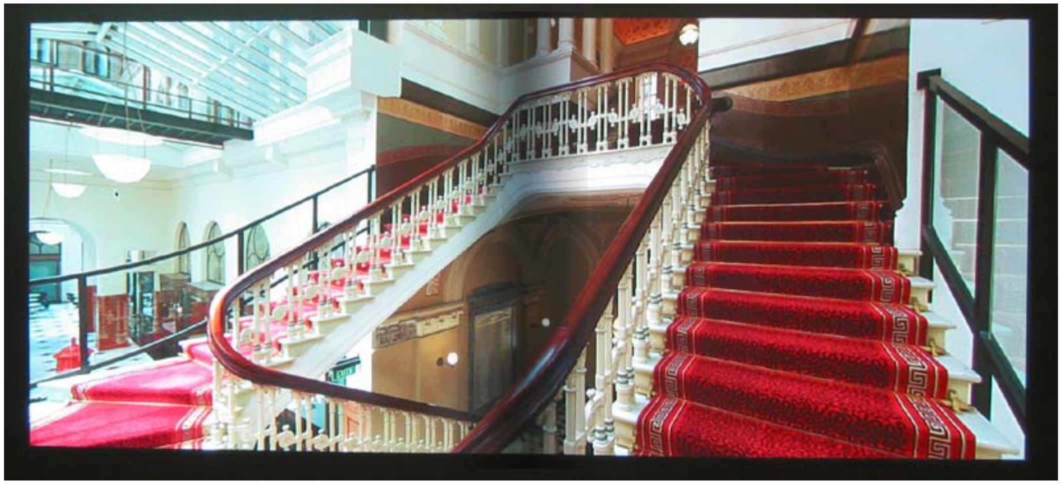


Good Window Size



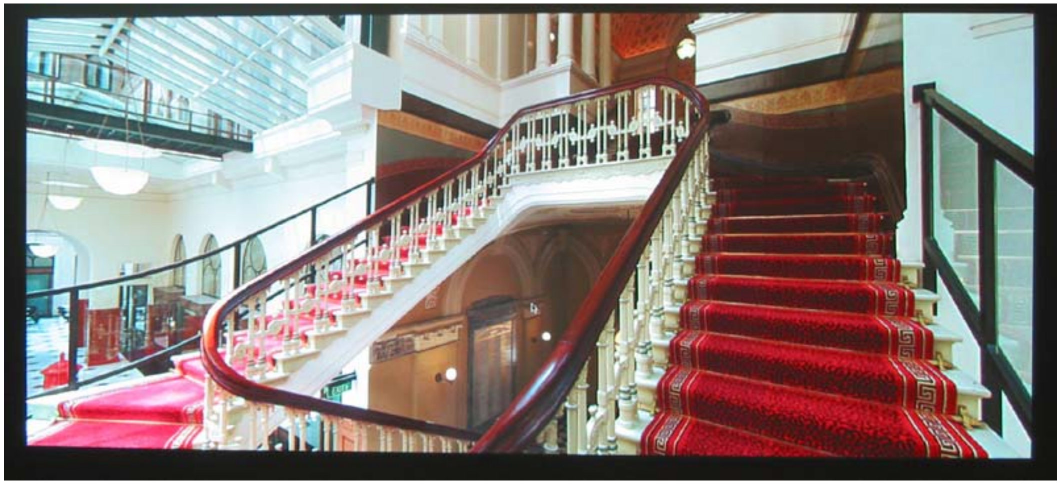
“Optimal” Window: smooth but not ghosted

Type of Blending function



Linear
(Only function continuity)

Spline or Cosine
(Gradient continuity also)

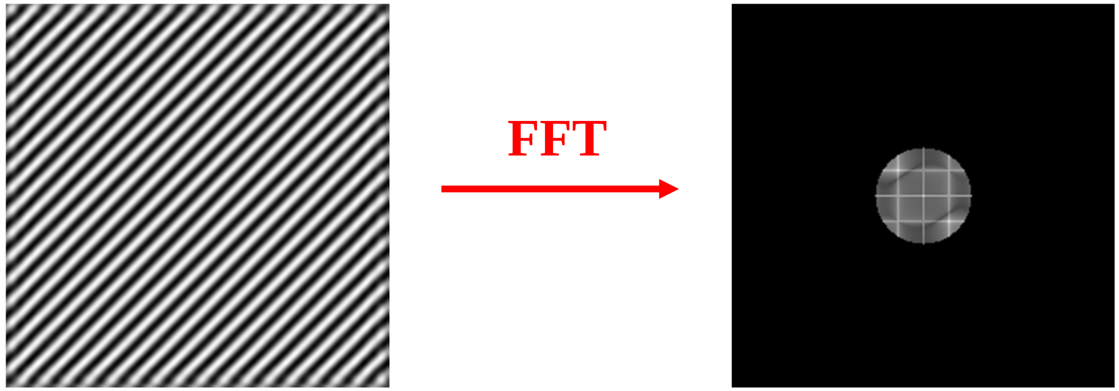


What is the Optimal Window?

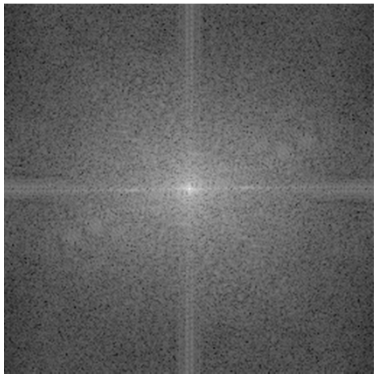
- To avoid seams
  - window = size of largest prominent feature
- To avoid ghosting
  - window <= 2*size of smallest prominent feature

Natural to cast this in the *Fourier domain*

- largest frequency <= 2*size of smallest frequency
- image frequency content should occupy one “octave” (power of two)



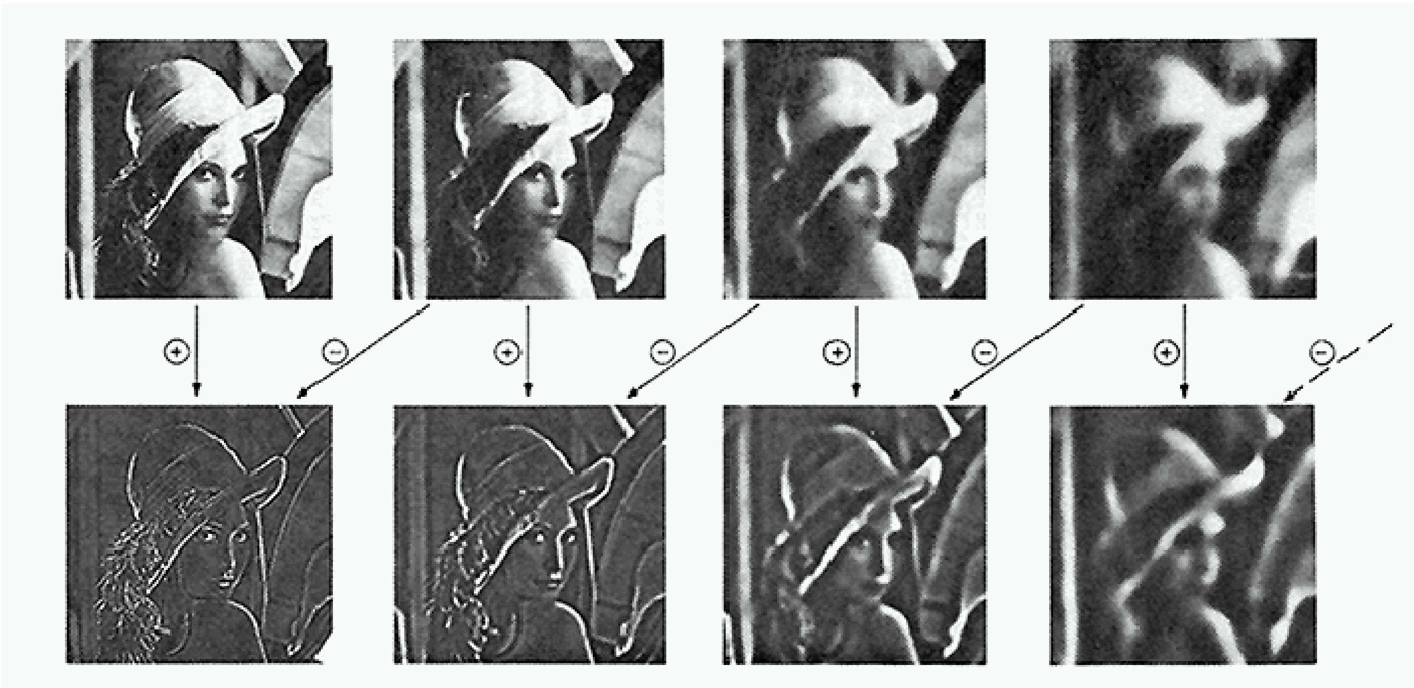
Frequency Spread is Wide



- Idea (Burt and Adelson)
  - Compute Band pass images for L and R
    - Decomposes Fourier image into octaves (bands)
  - Feather corresponding octaves L^i with R^i
    - Splines matched with the image frequency content
    - Multi-resolution splines
    - If resolution is changed, the width can be the same
  - Sum feathered octave images

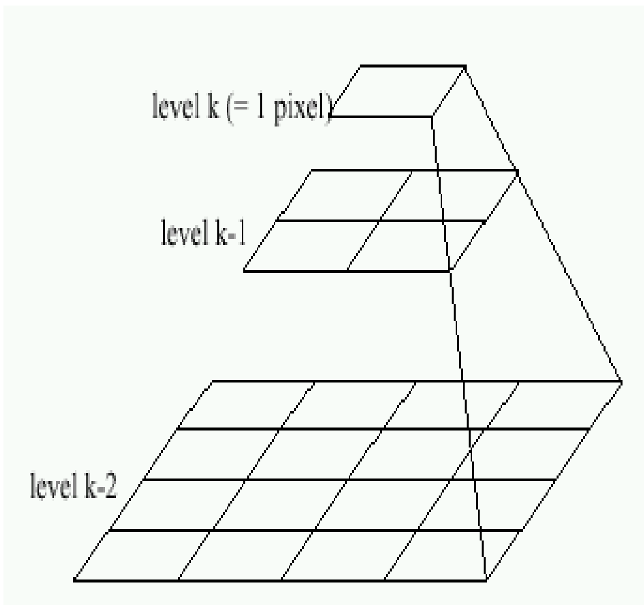
Octaves in the Spatial Domain

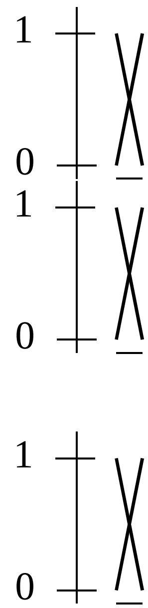
Lowpass Images

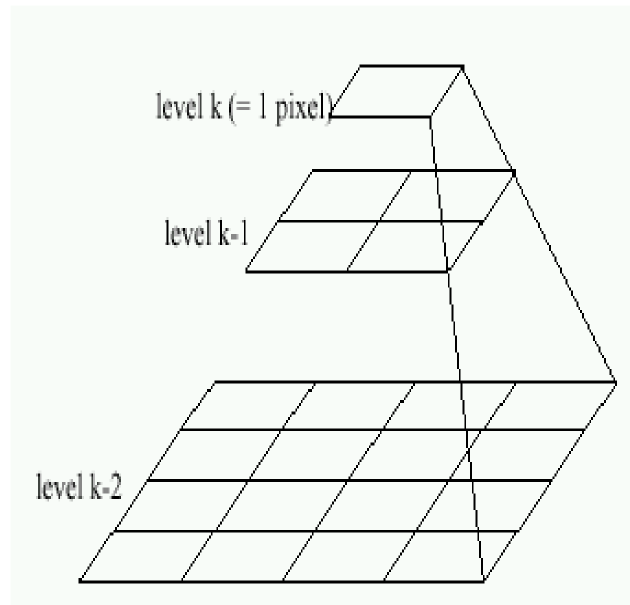


- Bandpass Images

Pyramid Blending





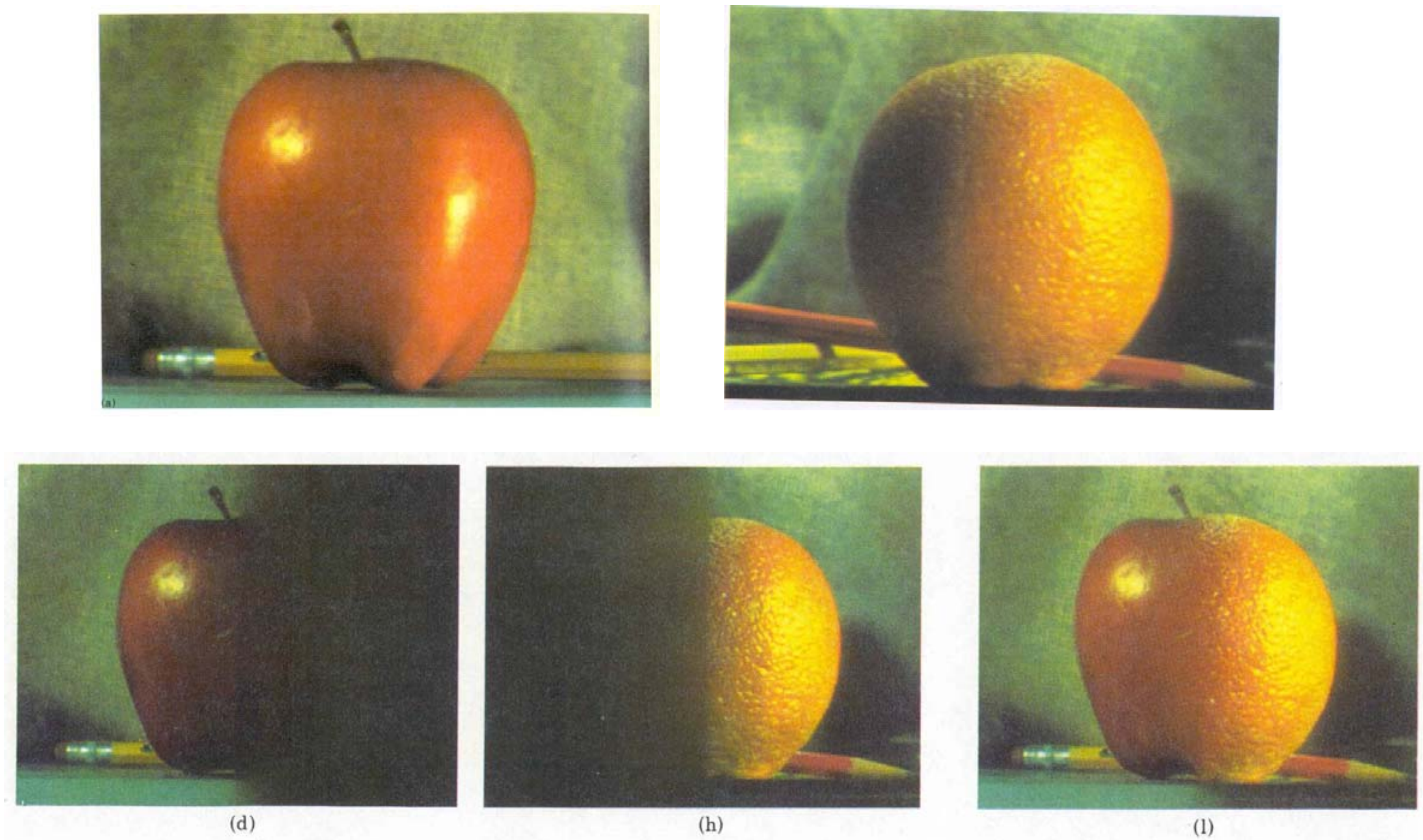


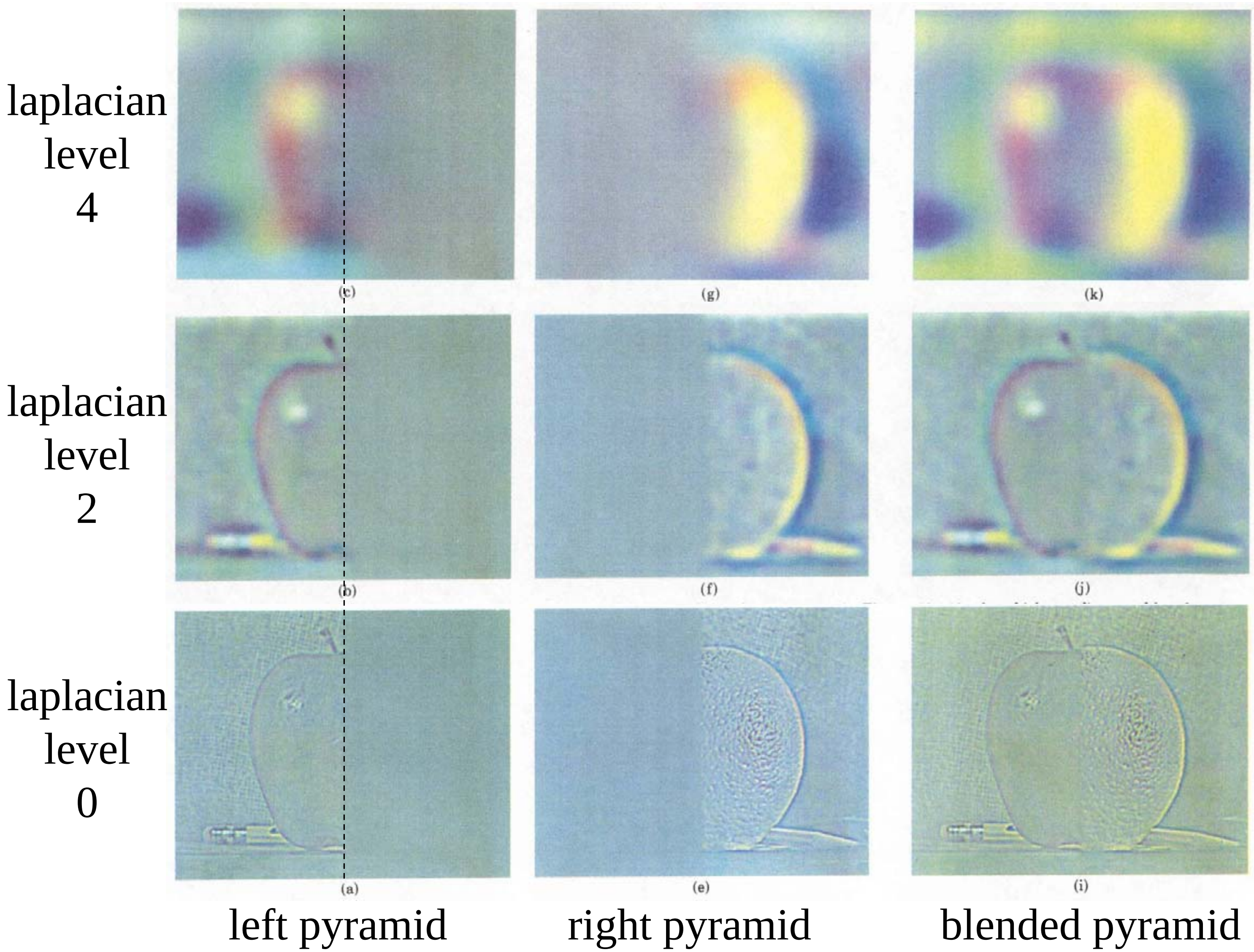
Left pyramid

blend

Right pyramid

Pyramid Blending




laplacian
level
4
laplacian
level
2
laplacian
level
0
(c)
(g)
(k)
(b)
(f)
(j)
(a)
(e)
(i)
left pyramid
right pyramid
blended pyramid

Laplacian Pyramid: Blending

- General Approach:
  1. Build Laplacian pyramids LA and LB from images A and B
  2. Build a Gaussian pyramid GR from selected region R
  3. Form a combined pyramid LS from LA and LB using nodes of GR as weights:
     - LS(i,j) = GR(i,j,)*LA(I,j) + (1-GR(i,j))*LB(I,j)
  4. Collapse the LS pyramid to get the final blended image

Don’t Blend, CUT!

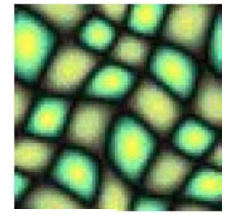
Moving objects become ghosts

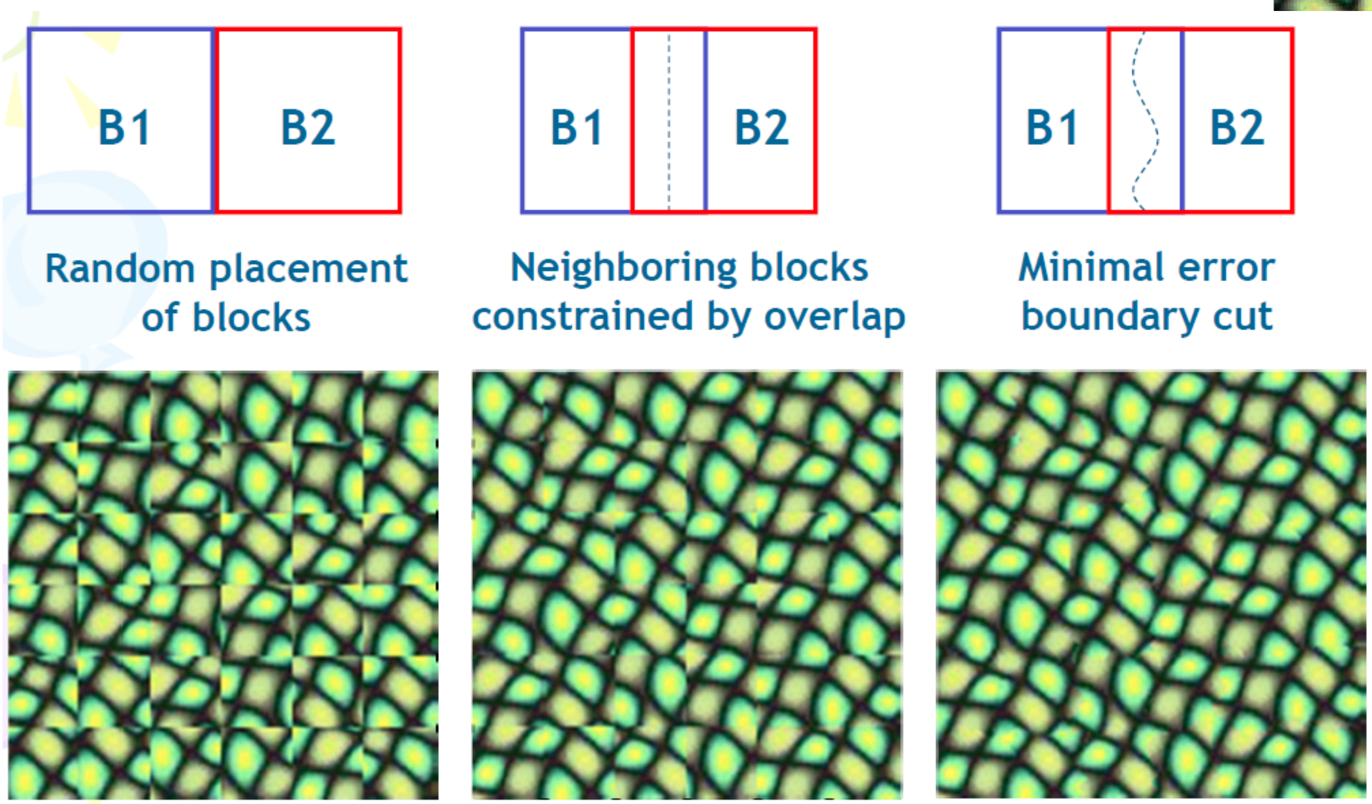
- So far we only tried to blend between two images. What about finding an optimal seam?

Davis 1998

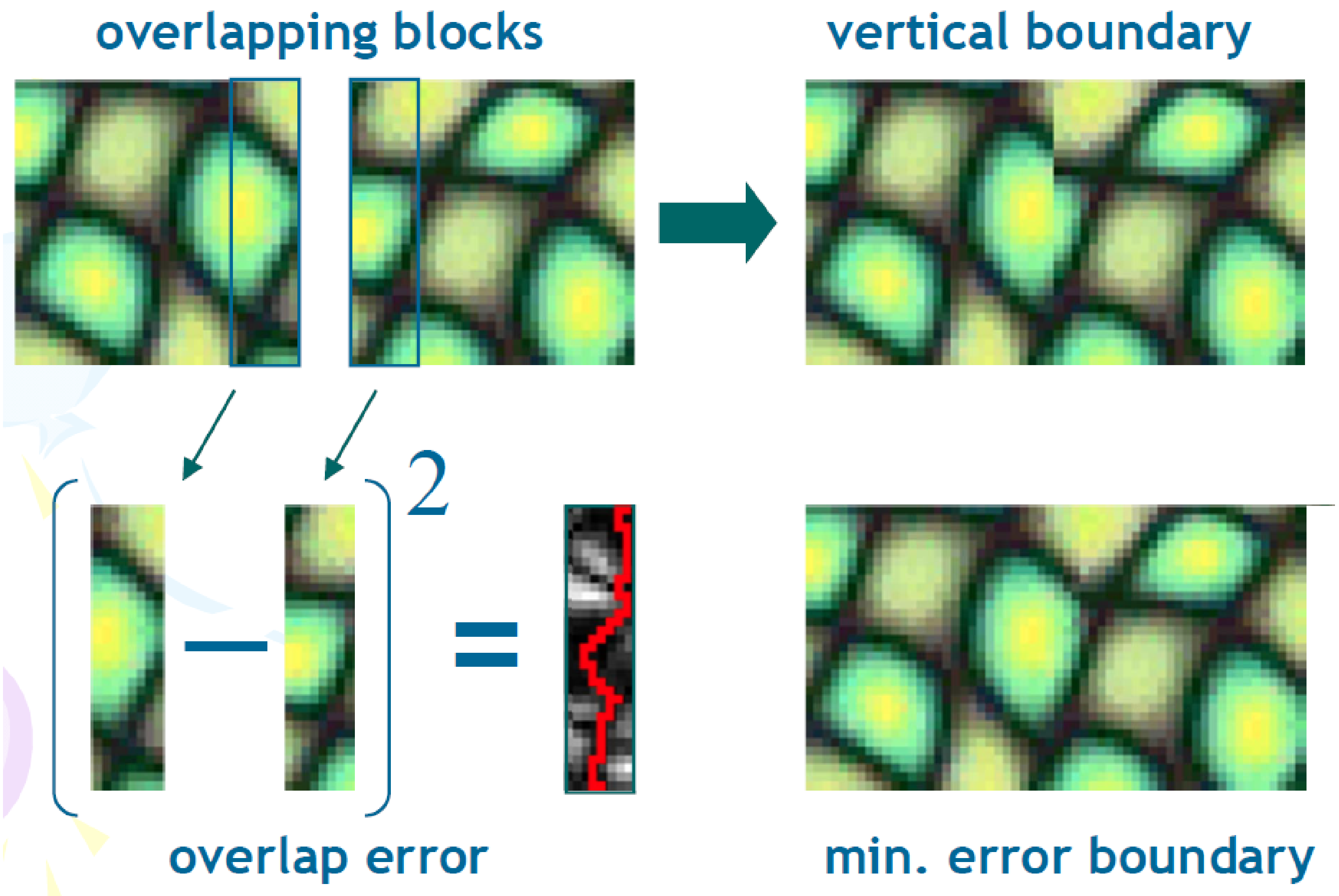
- Segment into regions
  - Single source per region
  - Avoid artifacts along the boundary
    - Dijkstra's shortest path method

Eros and Freeman 2001





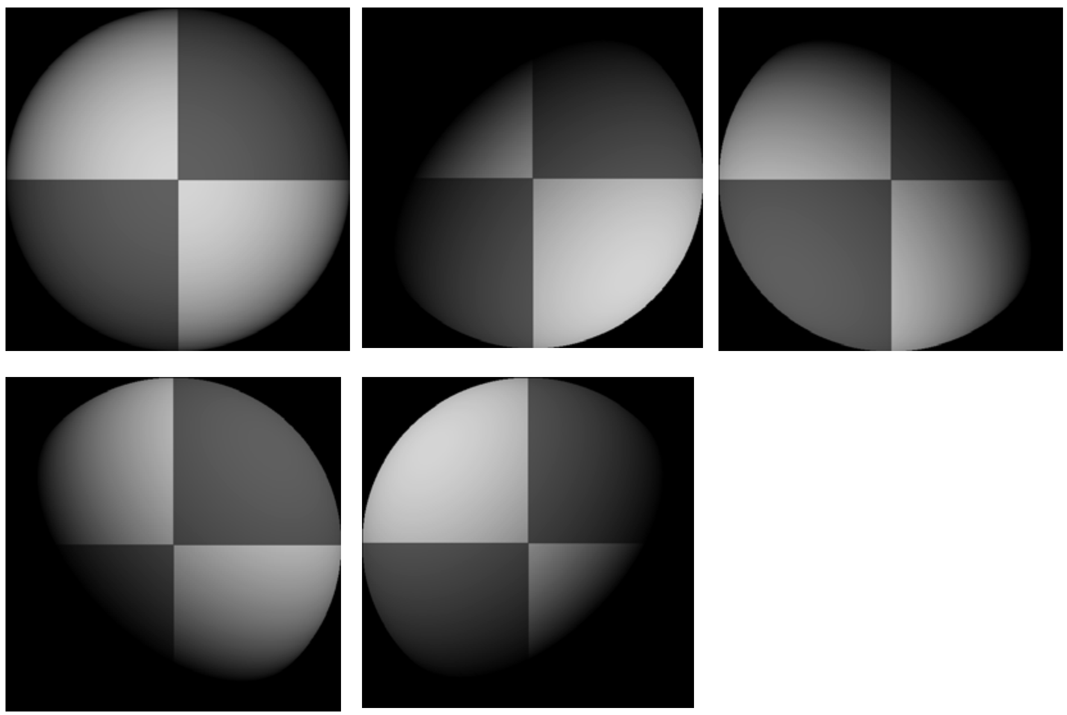
Minimum Error Boundary



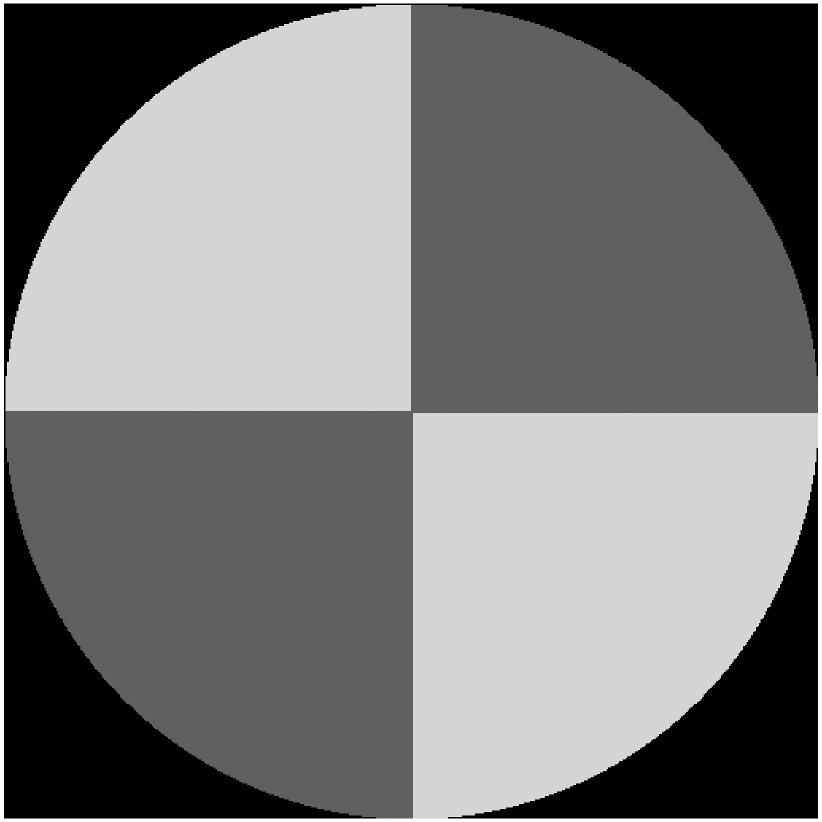
Photometric Stereo

Example figures

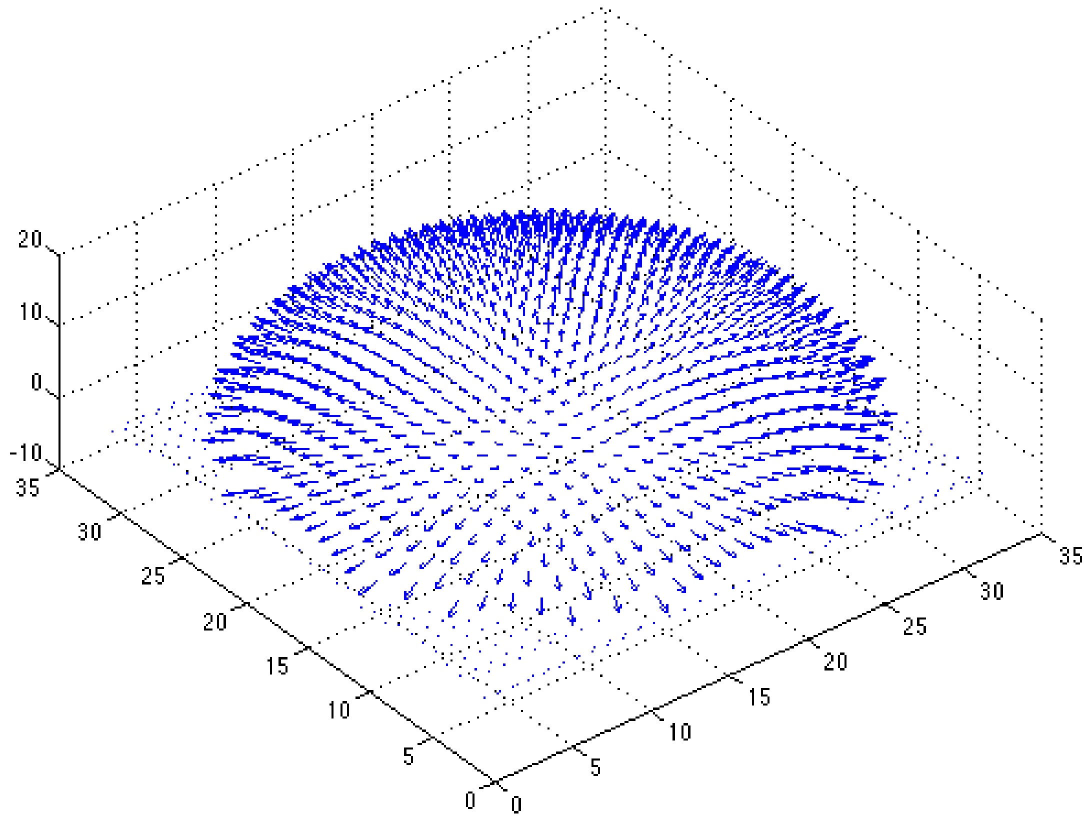
- five input images taken by changing only the light position



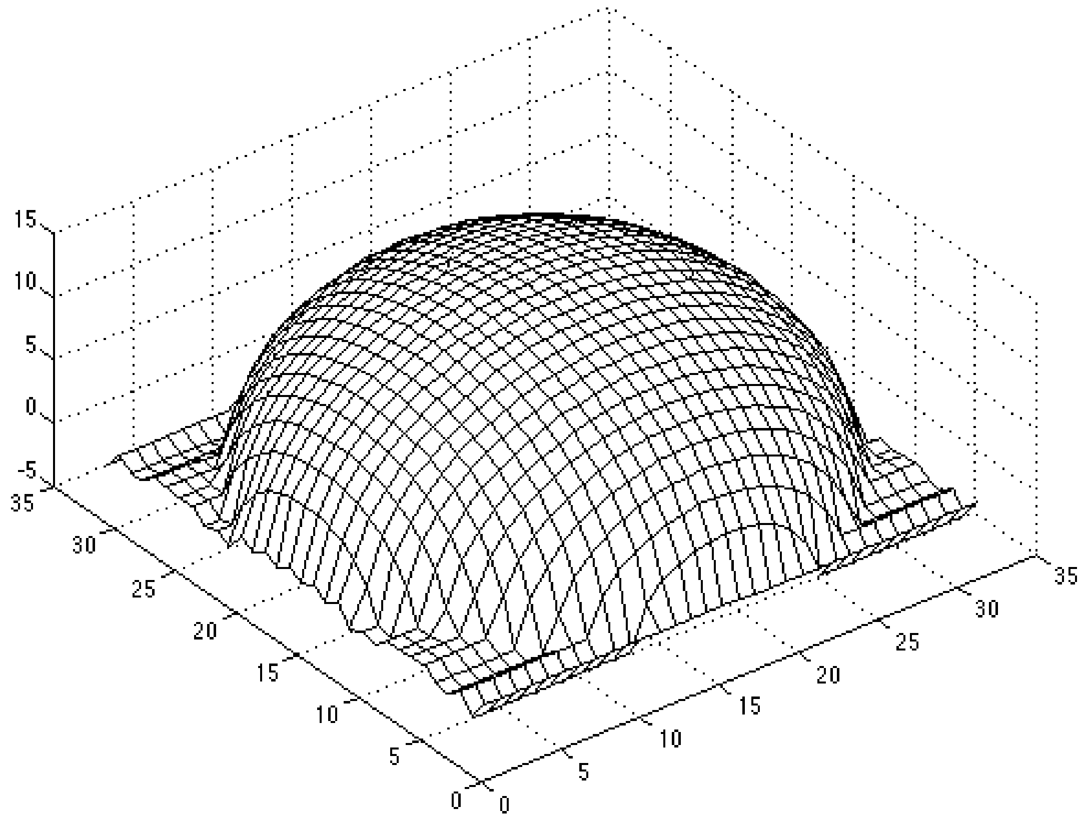
Recovered reflectance



Recovered normal field

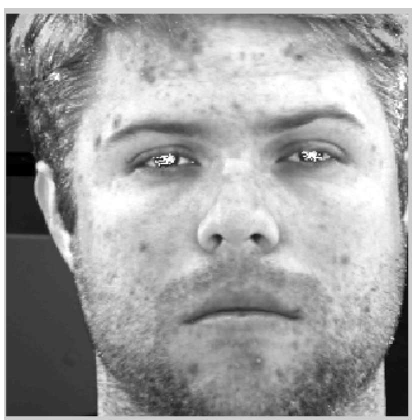
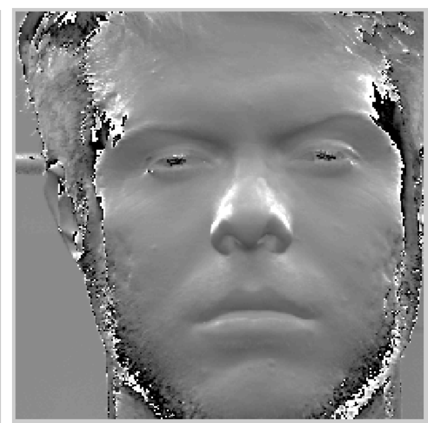


Surface recovered by integration



Photometric stereo example

data from: http://www1.cs.columbia.edu/~belhumeur/pub/images/yalefacesB/readme