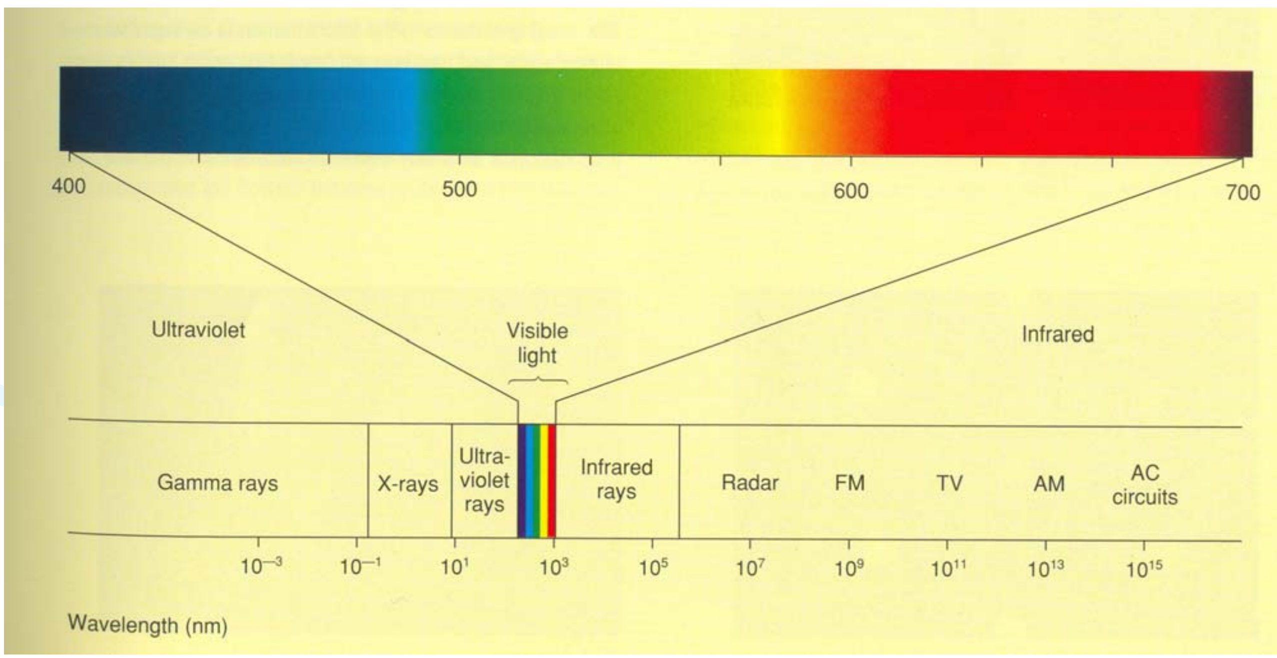
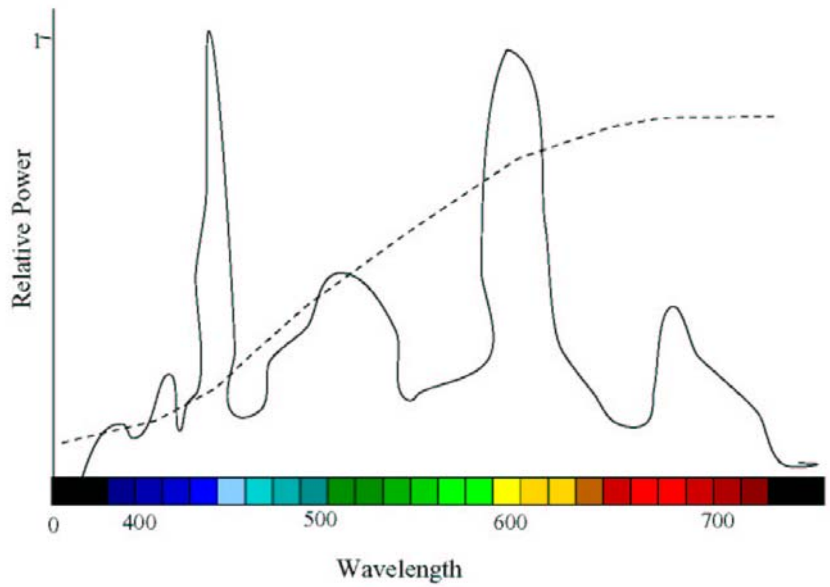
Color Image Processing

What is color?

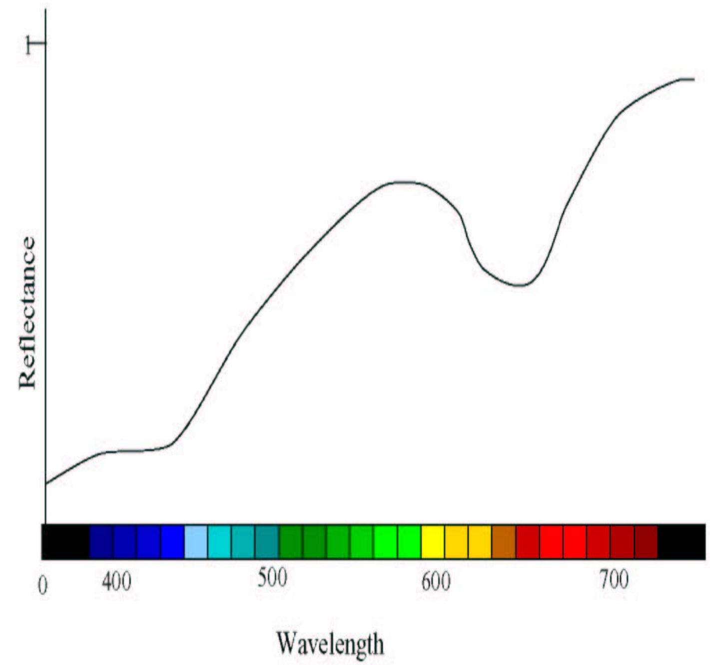
- Selective emission/reflectance of different wavelengths



What is color?

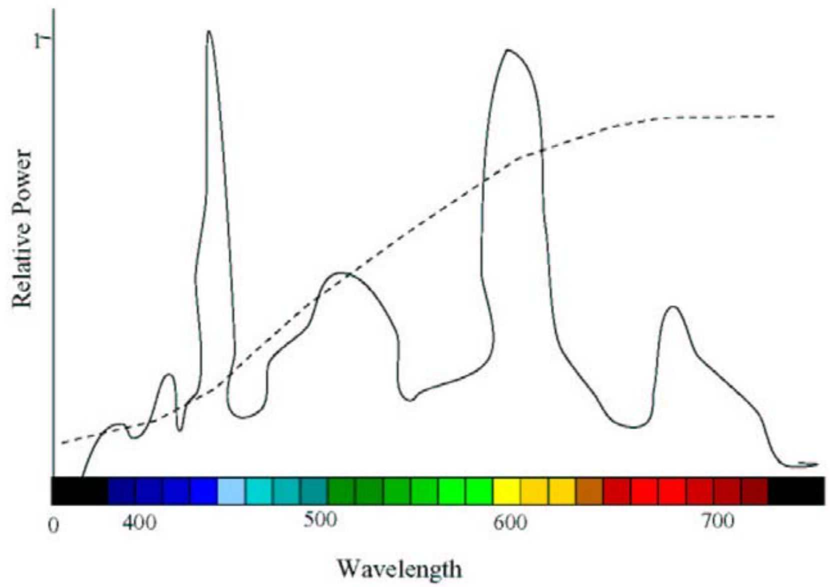


Illumination

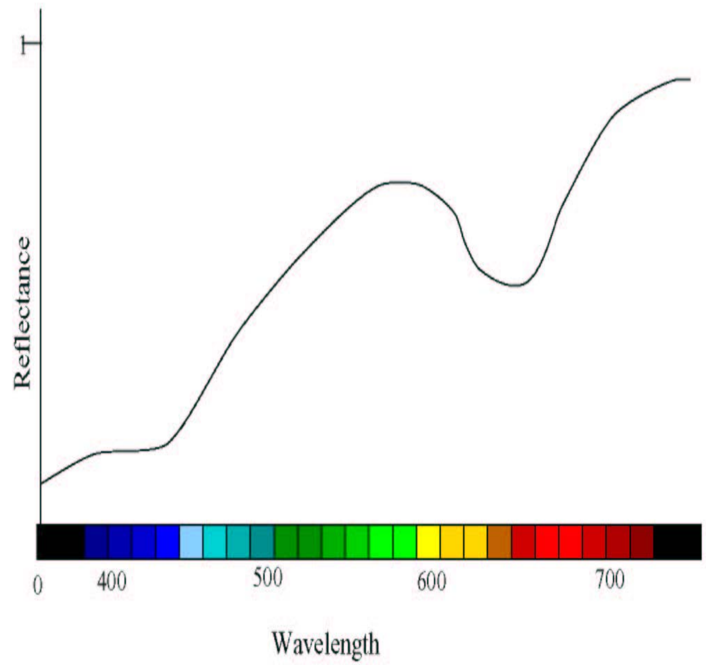


Reflectance

What is color stimuli?



X

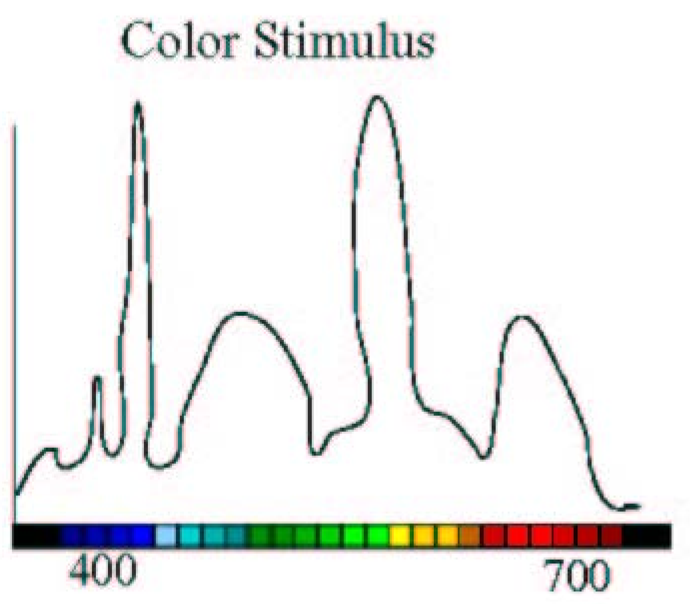


Illumination

Reflectance

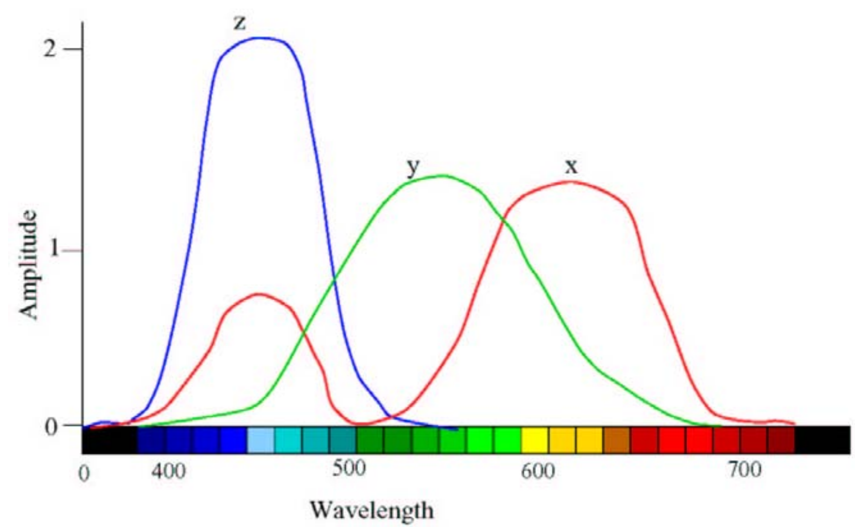
What is perceived color?

- The response generated by a stimulus in the cones gives the perceived color
- Three responses



X

Response of three human cones



Computations on Color

- Very difficult using spectrums
- Can we have some sort of coordinate space to define color?

Tristimulus Values

- Integration over wavelength

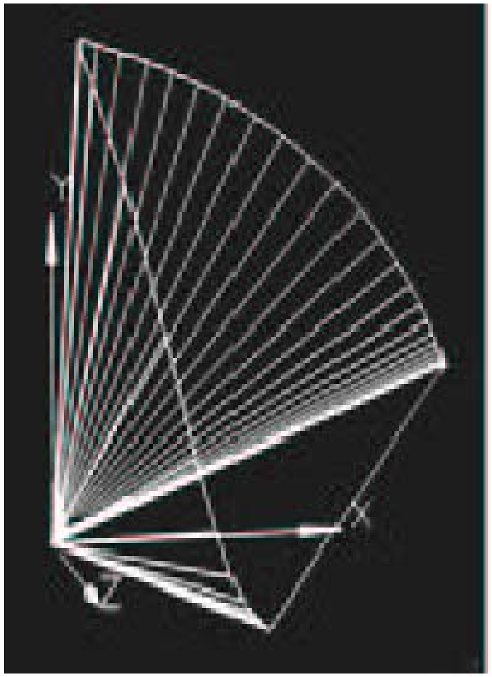
$$X = \int_{\lambda} C(\lambda)x(\lambda) = \sum_{\lambda=400}^{\lambda=700} C(\lambda)x(\lambda)$$

$$Y = \int_{\lambda} C(\lambda)y(\lambda) = \sum_{\lambda=400}^{\lambda=700} C(\lambda)y(\lambda)$$

$$Z = \int_{\lambda} C(\lambda)z(\lambda) = \sum_{\lambda=400}^{\lambda=700} C(\lambda)z(\lambda)$$

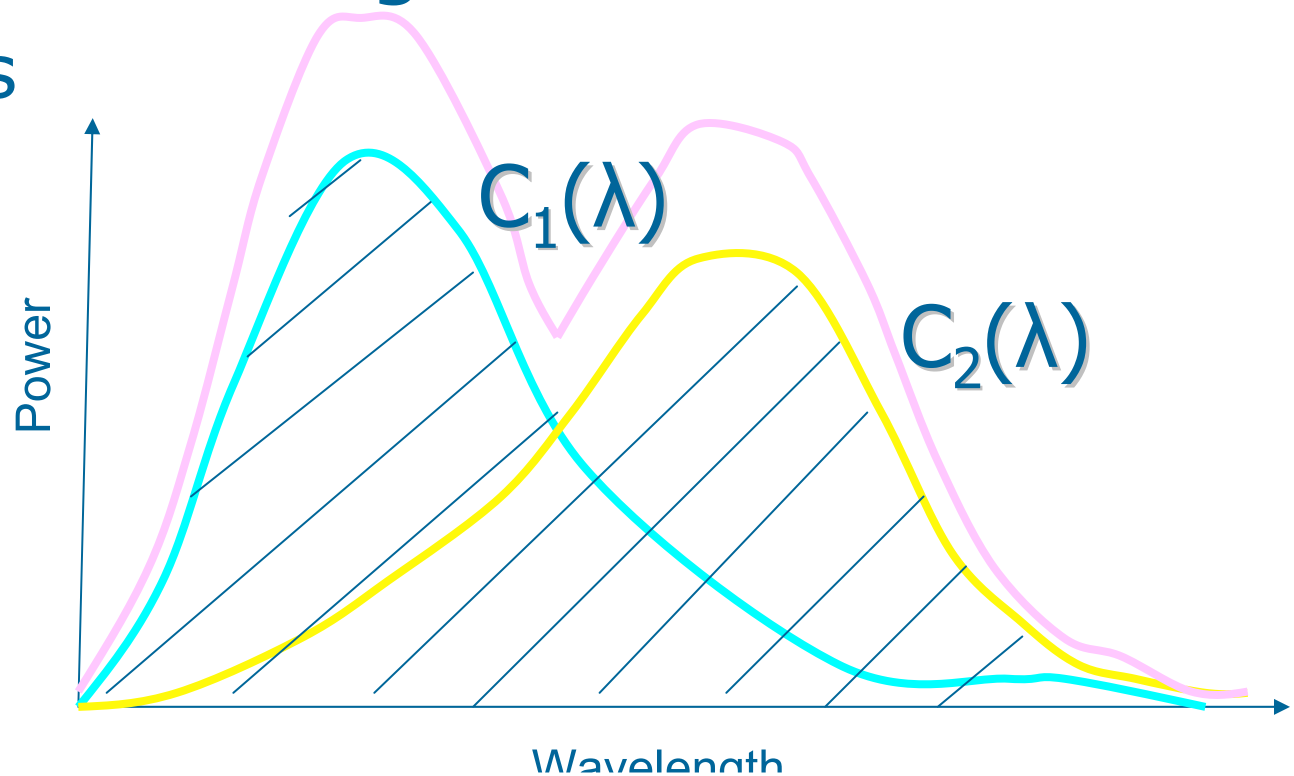
CIE XYZ Space

- Real colors span a sub-set of the XYZ space
- Two different stimuli can have same XYZ values
  - Metamerism



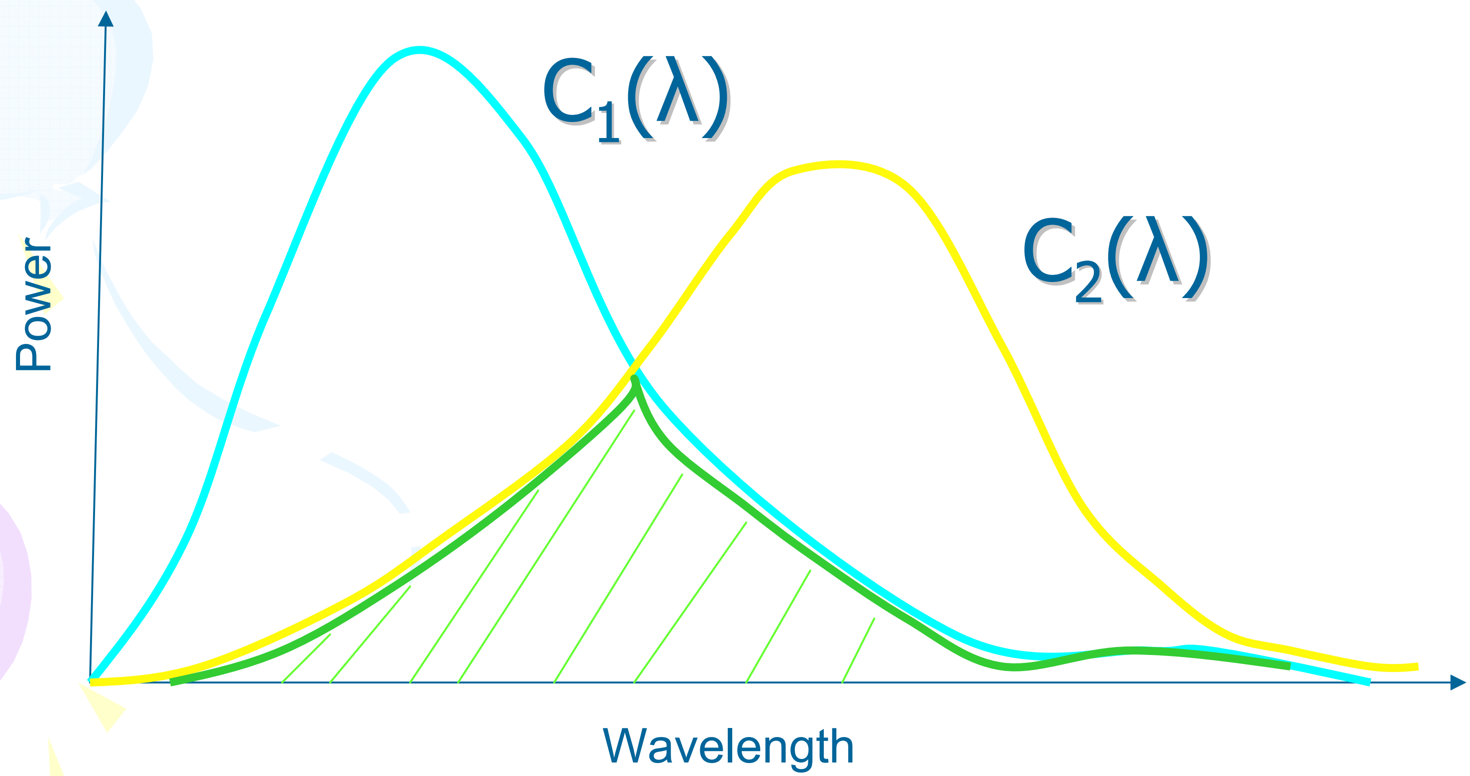
How does this help?

- Additive color mixtures modeled by addition in XYZ space
- When spectrums get added
  - Displays



Can there be any other mixture?

- Subtractive like paint
- Cannot be modeled by CIE XYZ space



What does it not offer?

- No physical feel as to how colors are arranged
- How do brightness change?
- How does hue change?

What are the perceived properties?

- Intensity
  - Sum of the spectrum
  - Energy under the spectrum
- Hue
  - Mean wavelength of the spectrum
  - What wavelength sensation is dominant?
- Saturation
  - Standard deviation of the spectrum
  - How much achromatic/gray component?
- Chrominance – Hue and saturation

CIE XYZ space

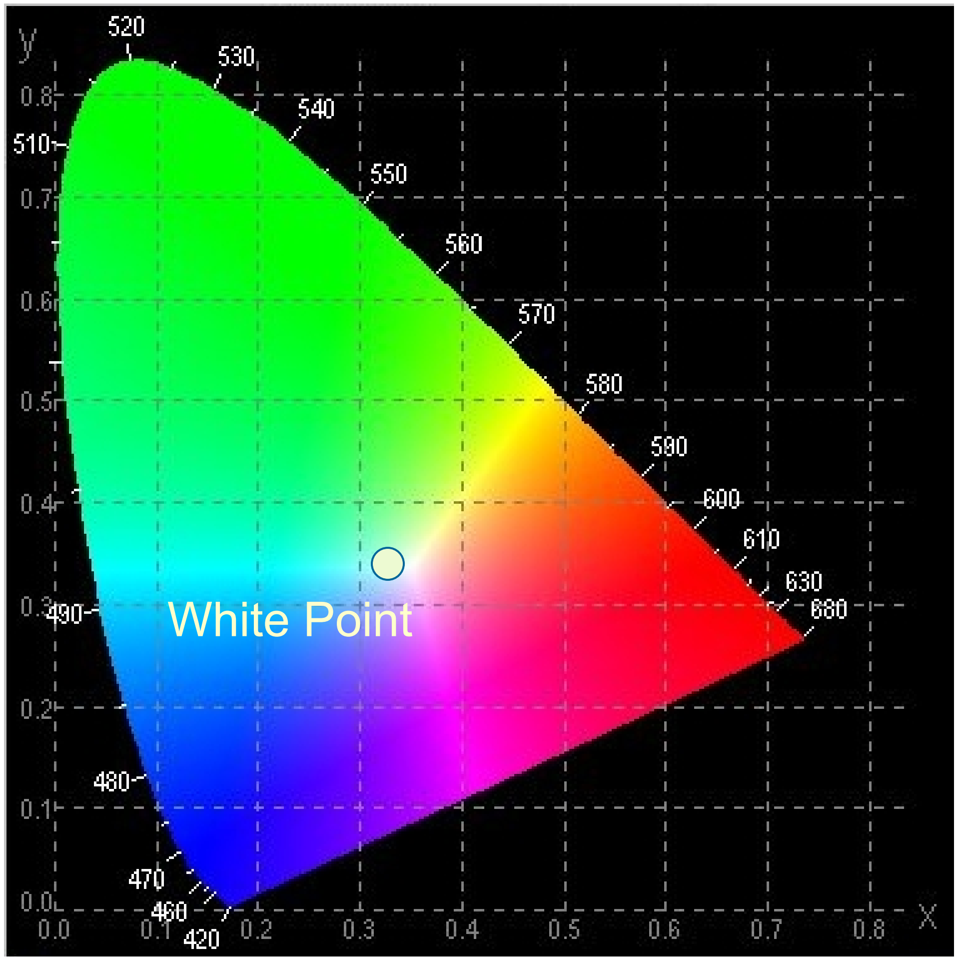
- Intensity (I) – X+Y+Z
- Chrominance (x,y) - (X/I, Y/I)
  - Chromaticity chart
  - Projection on a plane with normal (1,1,1)
  - Reduction of dimension
  - Similar to 3D to 2D in geometry

What does this mean?

- Scaling a vector (kX,kY,kZ)
  - (x,y) does not change
  - Each vector from (0,0,0) is an iso-chrominance line
  - Each vector map to a point in the chromaticity chart

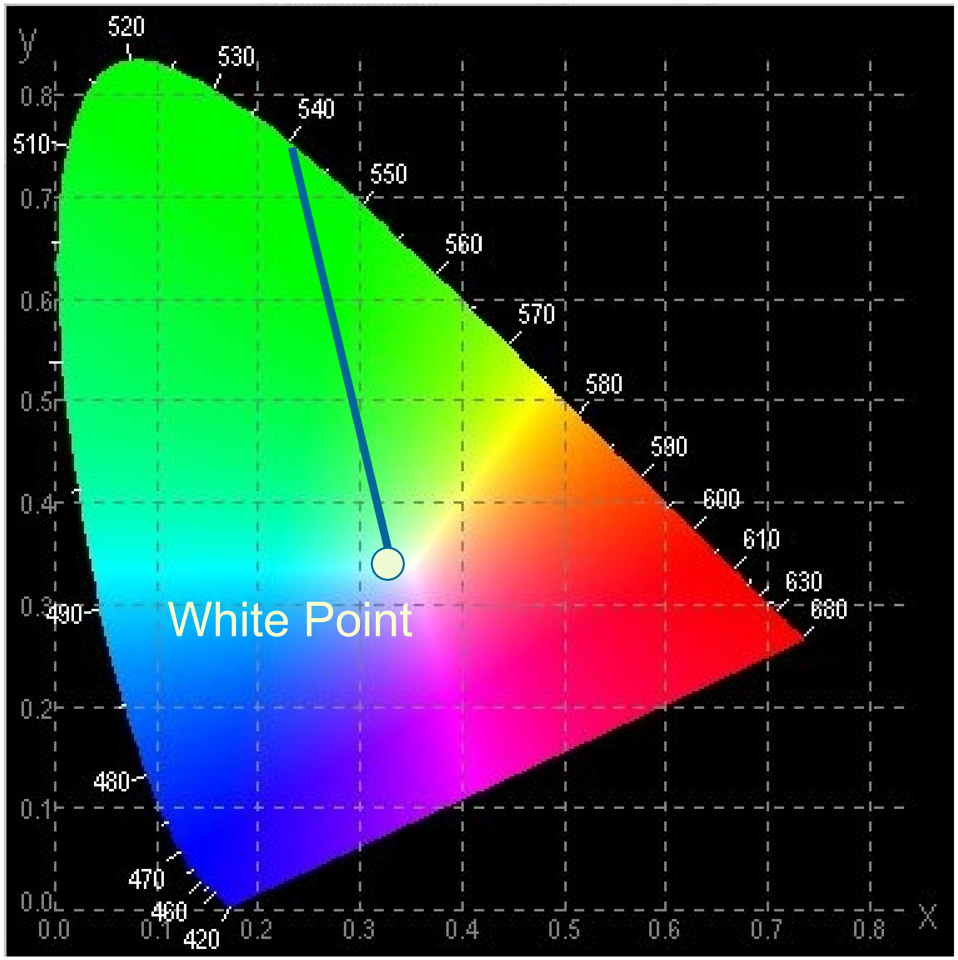
Chromaticity Coordinates

- Shows all the visible colors
- Achromatic Colors are at (0.33,0.33)
  - Why?
  - Called white point
- The saturated colors at the boundary
  - Spectral Colors



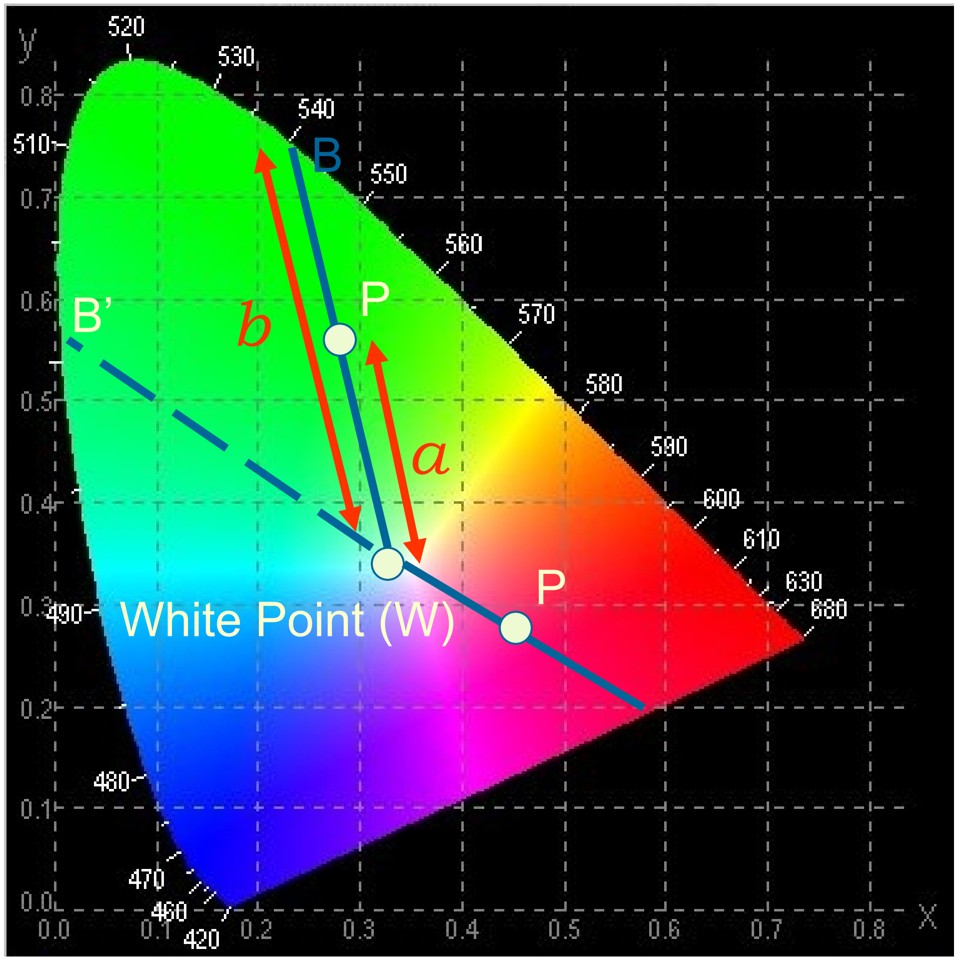
Chromaticity Chart

- Exception is purples
  - Non-spectral region in the boundary
- All colors on straight line from white point to a boundary has the same spectral hue
  - Dominant wavelength



Chromaticity Chart

- What happens here?
  - Complimentary wavelength
  - When mixed generate achromatic color
- Purity (Saturation)
  - How far shifted towards the spectral color
  - Ratio of a/b
  - Purity =1 implies spectral color with maximum saturation



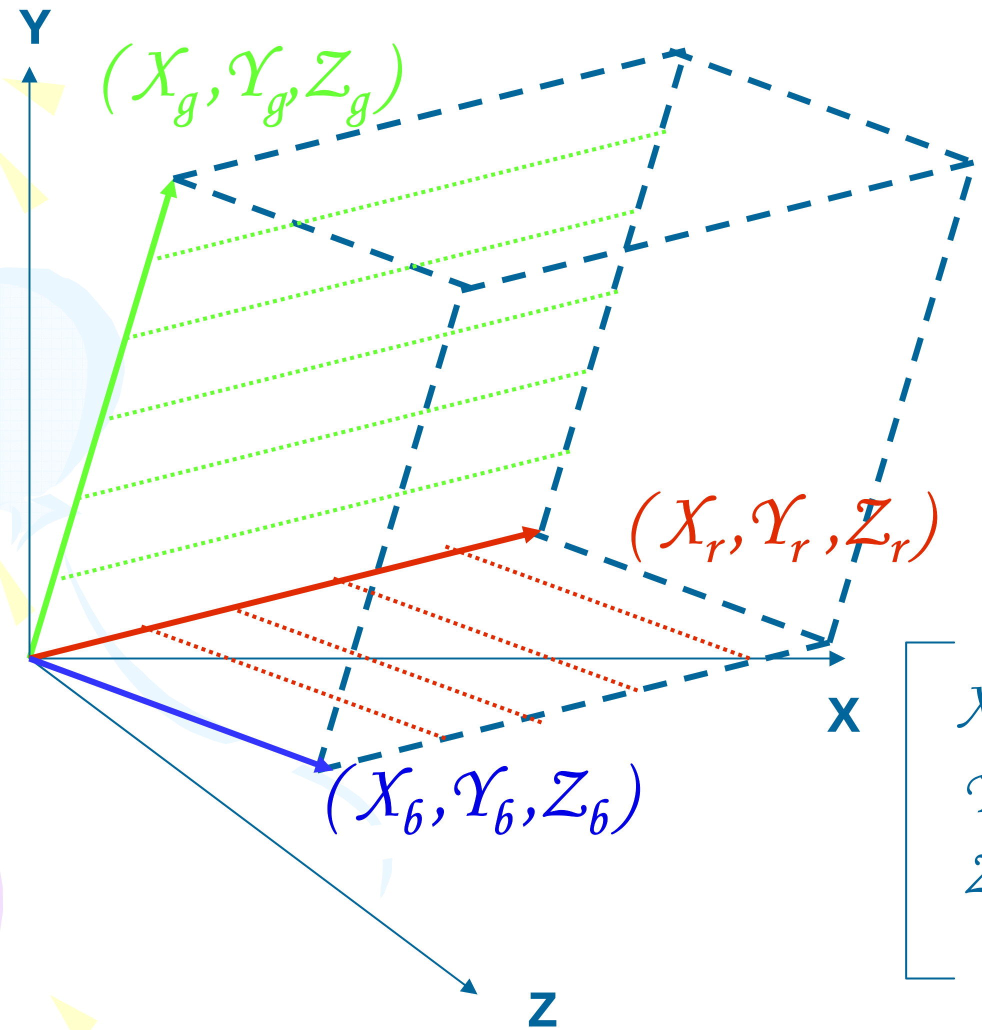
Luminance

- Perceived brightness
  - Based on eye’s response
- Same brightness green looks brighter than blue or red
- This is proportional to Y

How to add colors?

- Add (X,Y,Z) coordinates
- What does this mean in terms of brightness and chrominance?
  - Add brightness
  - Linear combination of chrominance in proportion of the brightness
    - Look for errors in literature (I and not Y)

What is the RGB color?



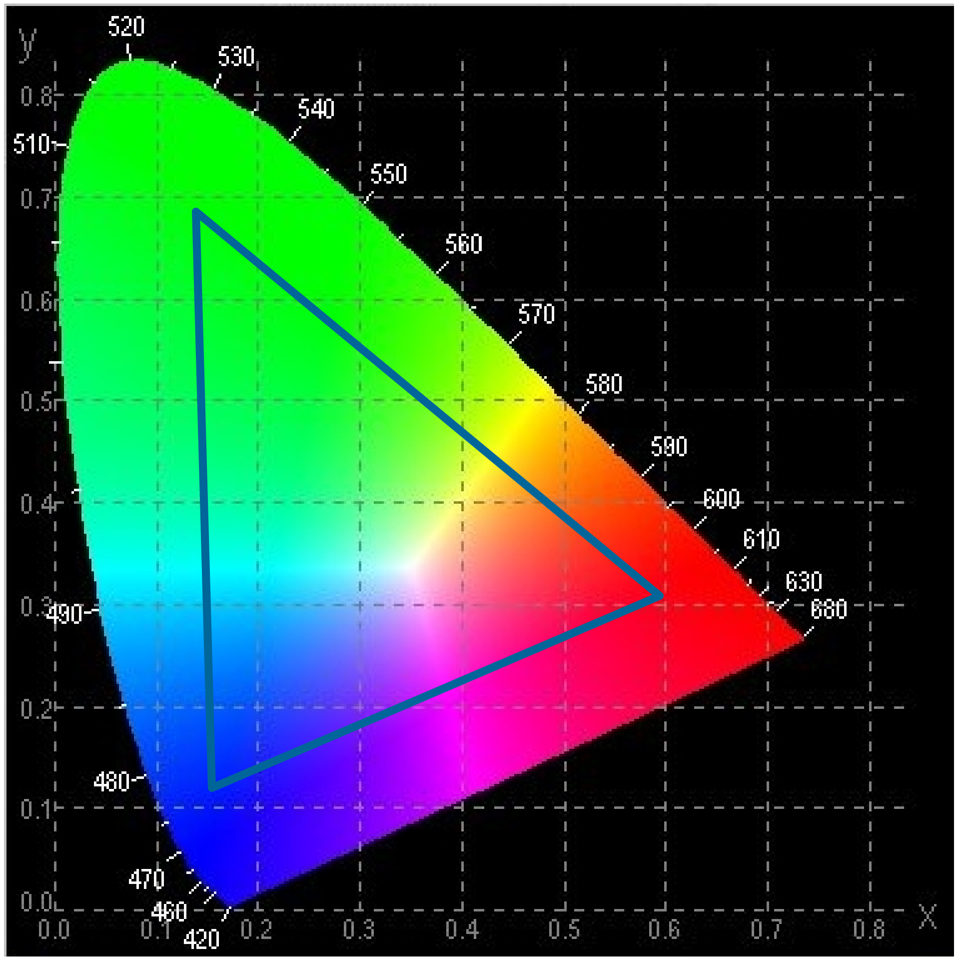
$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix} \begin{bmatrix} i_r \\ i_g \\ i_b \end{bmatrix}$$

Color reproducibility

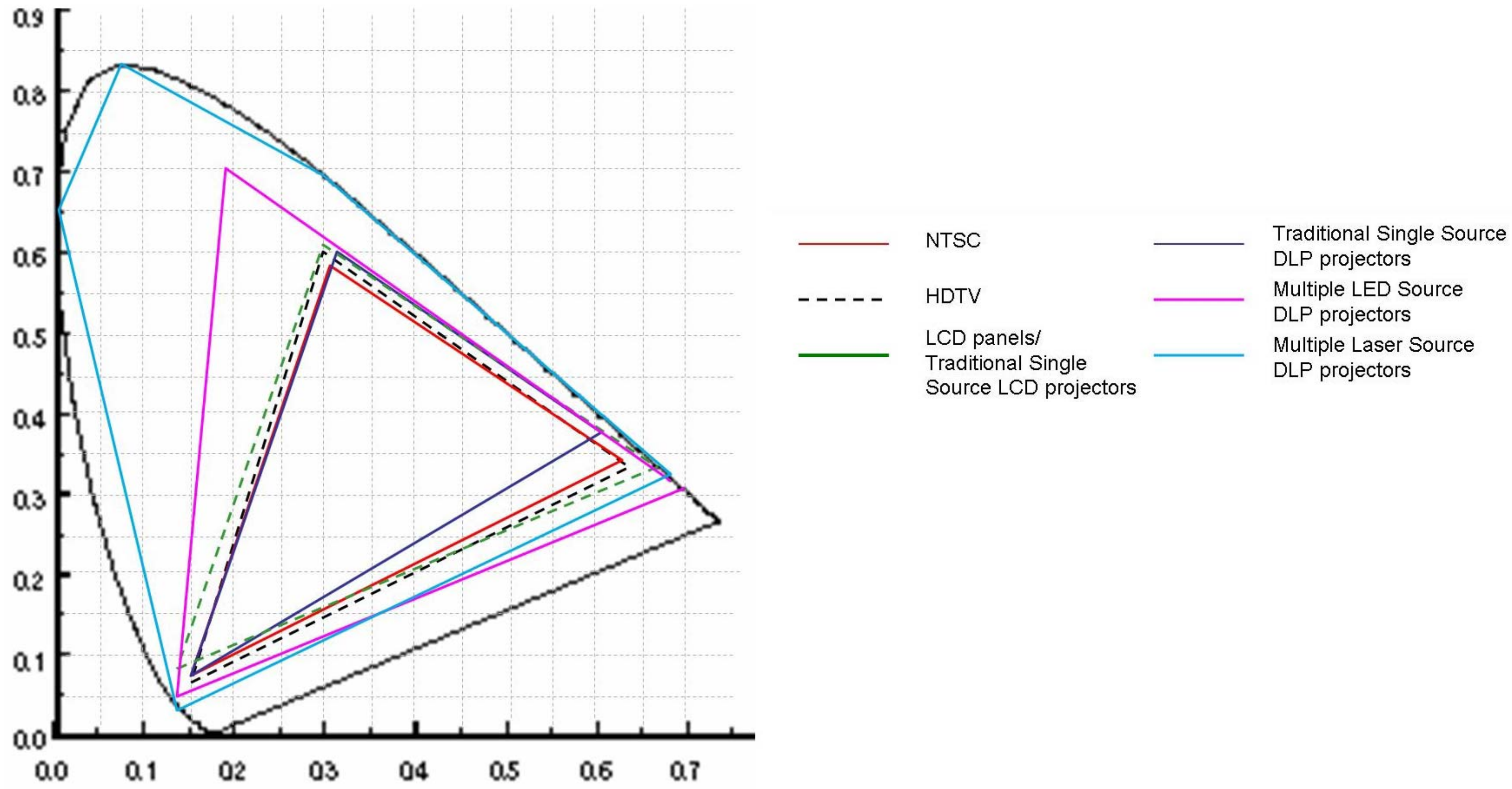
- Only a subset of the 3D CIE XYZ space called 3D color gamut
- Projection of the 3D color gamut on the same plane with normal (1,1,1)
  - Triangle
  - 2D color gamut
    - Cannot describe brightness range reproducibility

Specification Protocols

- Brightness or Luminance
- 2D gamut
  - Large if using more saturated primaries



Current standards and devices



Gamut Transformation

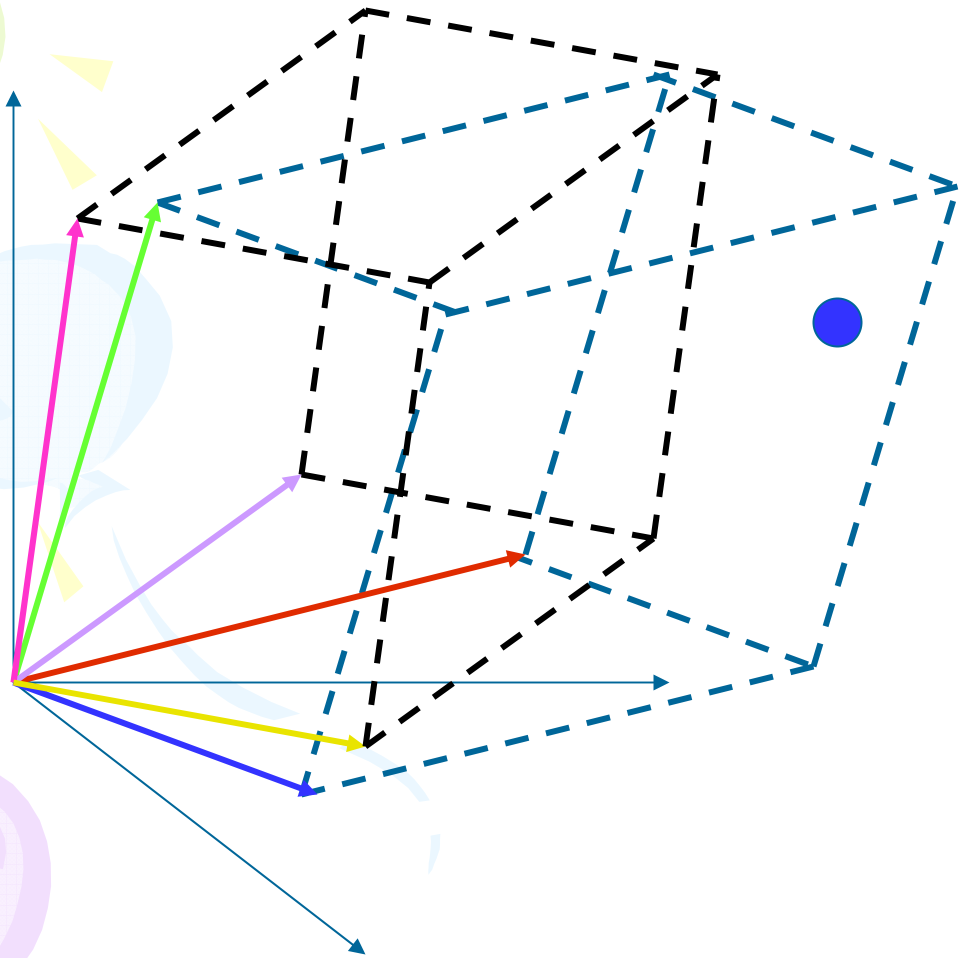
- Assume linear gamma
- $[X\ Y\ Z\ 1]^T = M\ [R\ G\ B\ 1]^T$
- Two devices
  - $[X\ Y\ Z\ 1]^T = M_1\ [R_1\ G_1\ B_1\ 1]^T$
  - $[X\ Y\ Z\ 1]^T = M_2\ [R_2\ G_2\ B_2\ 1]^T$
- $[R_2\ G_2\ B_2\ 1]^T = M_2^{-1}[X\ Y\ Z\ 1]$

$$= M_2^{-1}M_1[R_1\ G_1\ B_1\ 1]^T$$

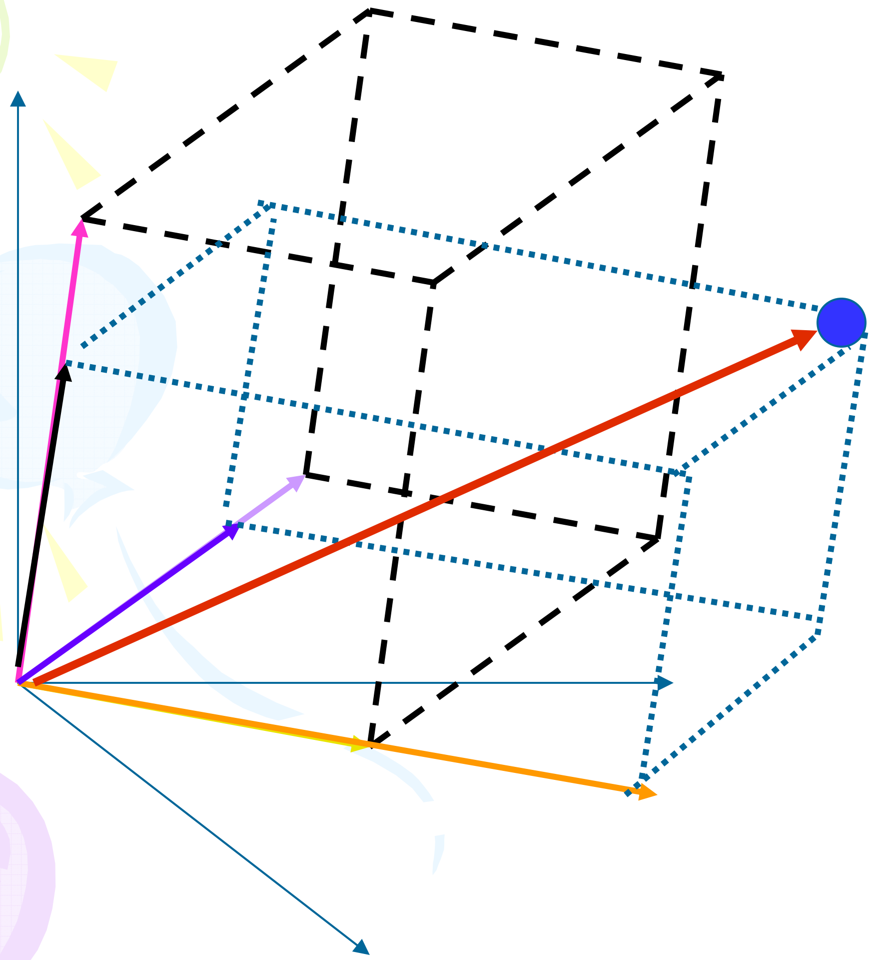
Gamut Transformation

- How to get the matrix from the standard spec?
- Given (Y,x,y) or (I,x,y) for the three vectors, you can compute (X,Y,Z)
  - (x. Y/y, Y, (1-x-y). Y/y)
  - (x.I, y.I, (1-x-y).I)
- **Does not change the color**, finds the new coordinates when using the new basis

Problem



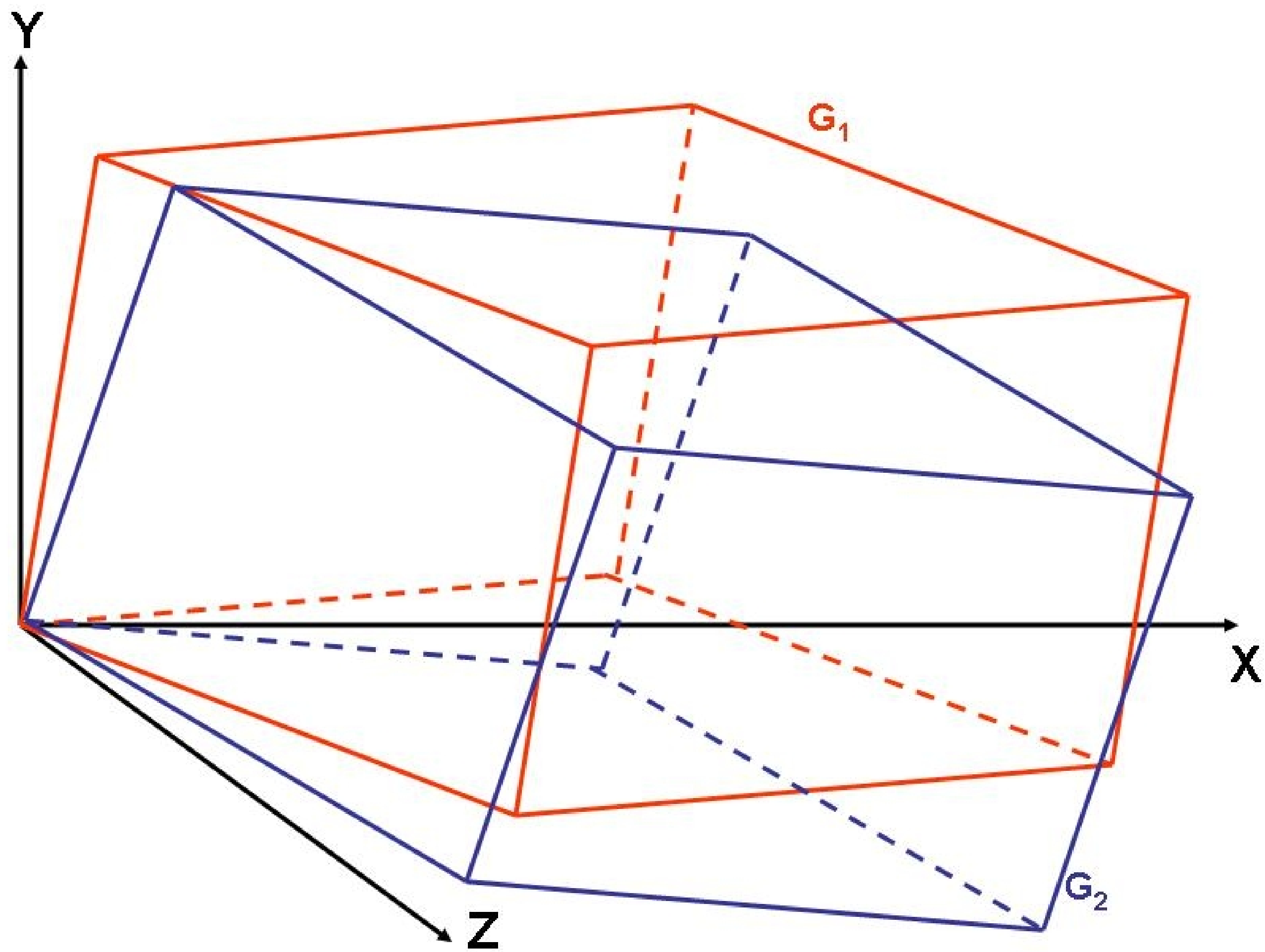
Problem: Out of Gamut colors



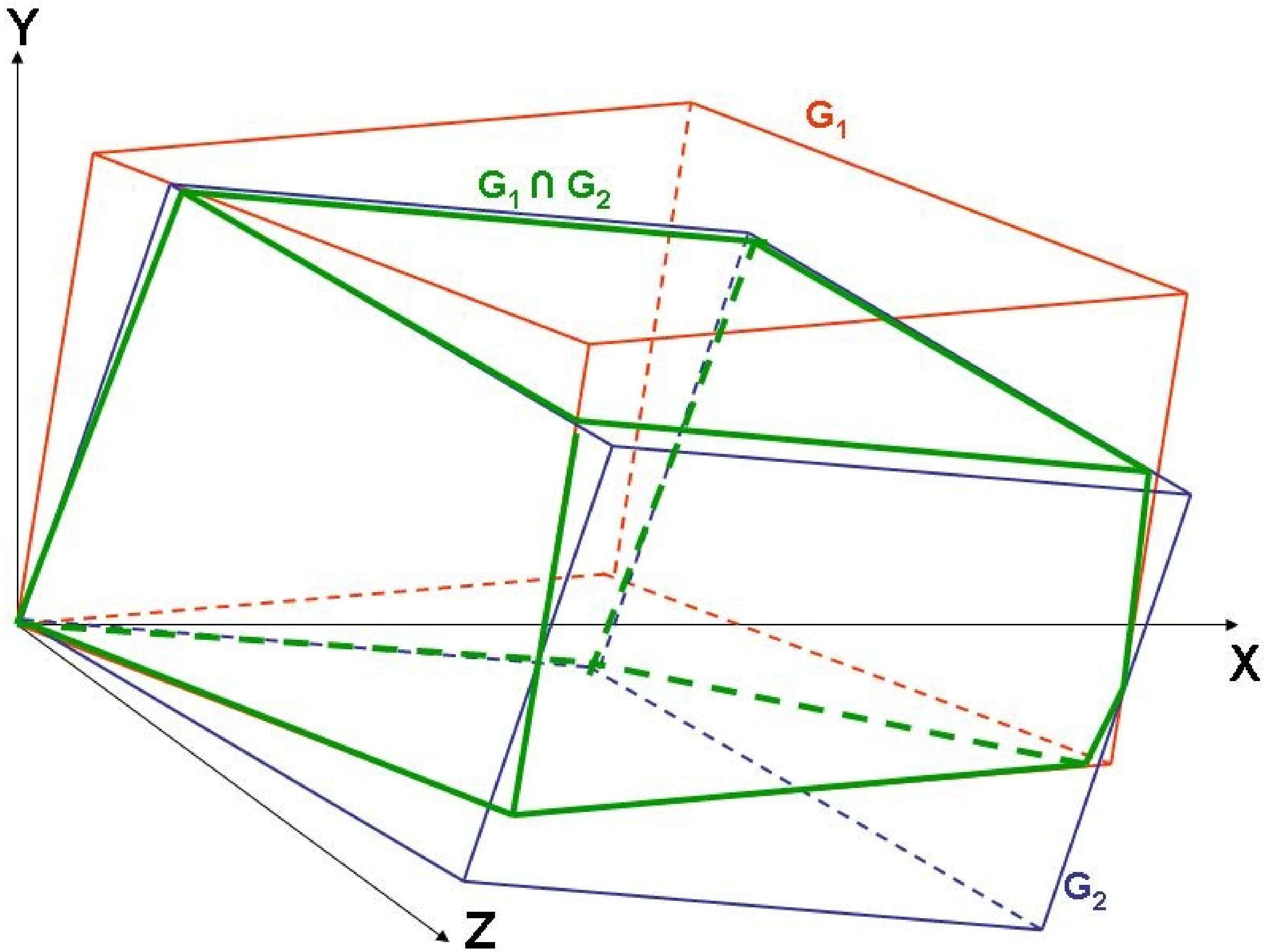
Gamut Matching

- Find a common color gamut defined by R_c, G_c, B_c
- Find the common function M_c
  - $[X\ Y\ Z\ 1]^T = M_c\ [R_c\ G_c\ B_c\ 1]^T$
- For any device i
  - $[R_i\ G_i\ B_i\ 1]^T = M_i^{-1} M_c\ [R_c\ G_c\ B_c\ 1]^T$

Two gamut

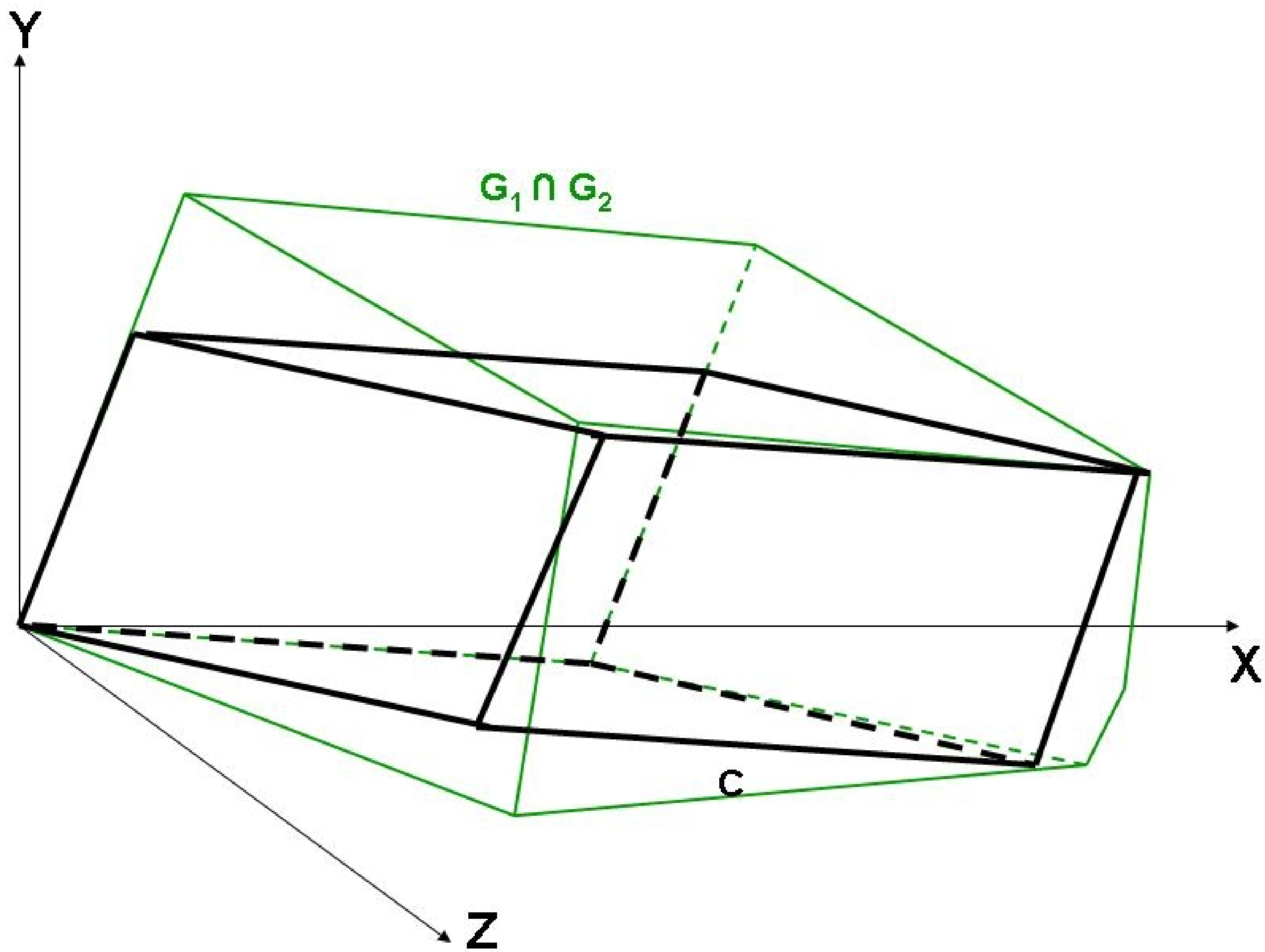


Find their intersection

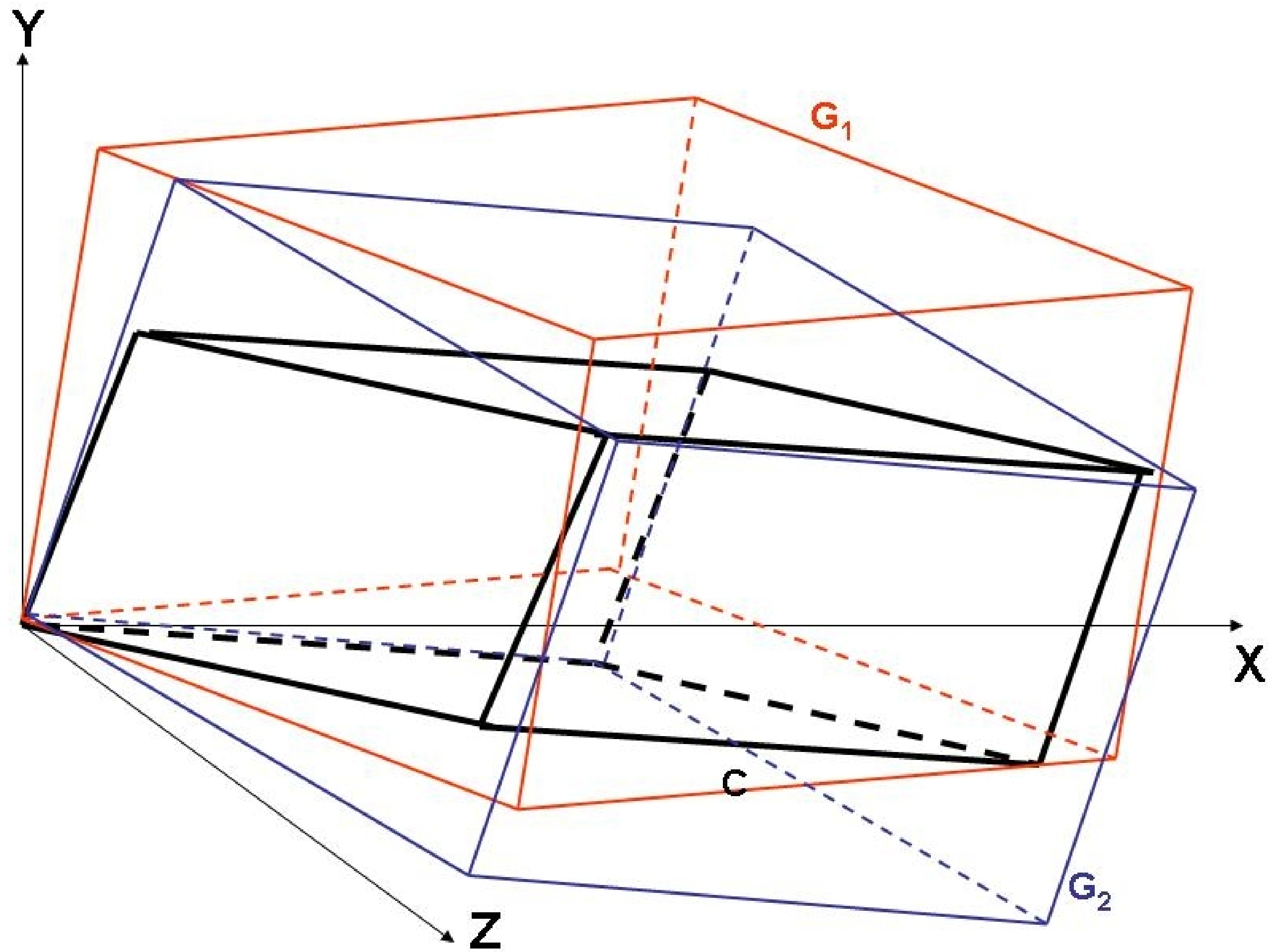


Need not be a parallelopipped

Find the common gamut



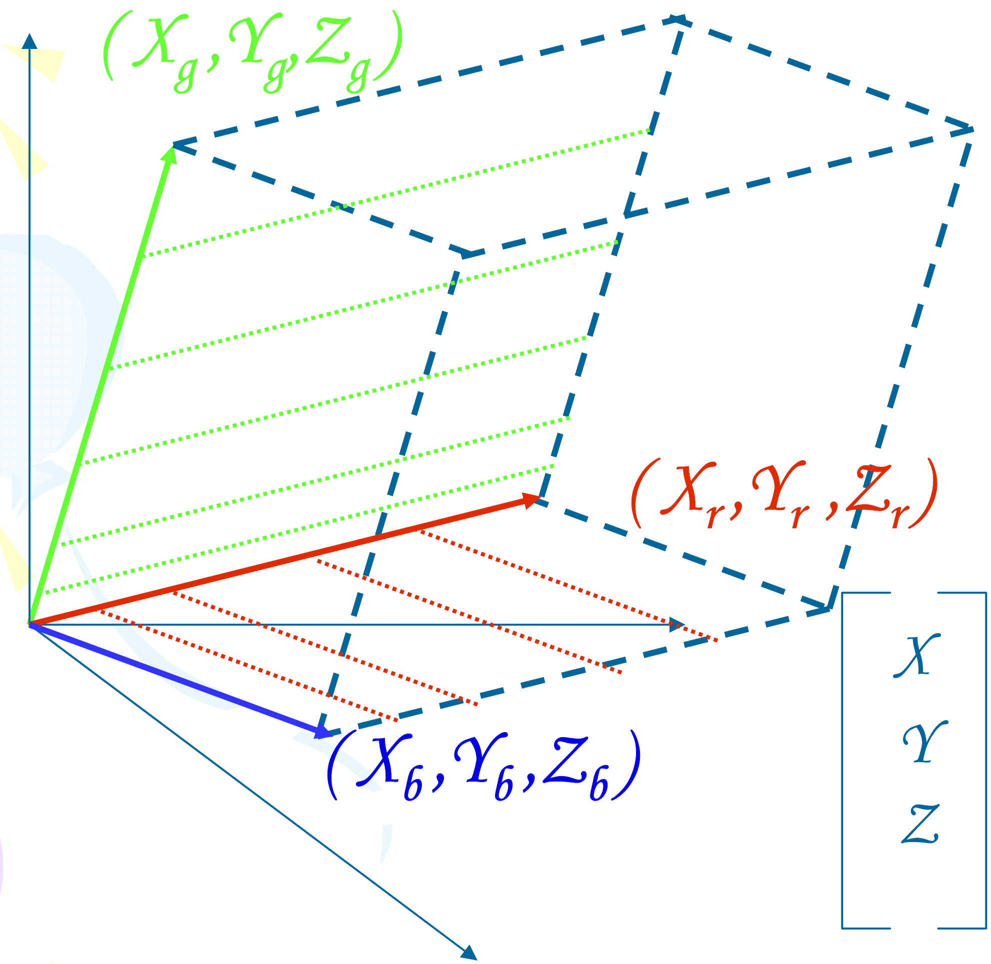
Find the mapping function



Gamut Mapping

- Changing the actual colors
  - Mapping color in one gamut to another in the new gamut
- If just dealing with two devices, may choose to move colors from one gamut to another

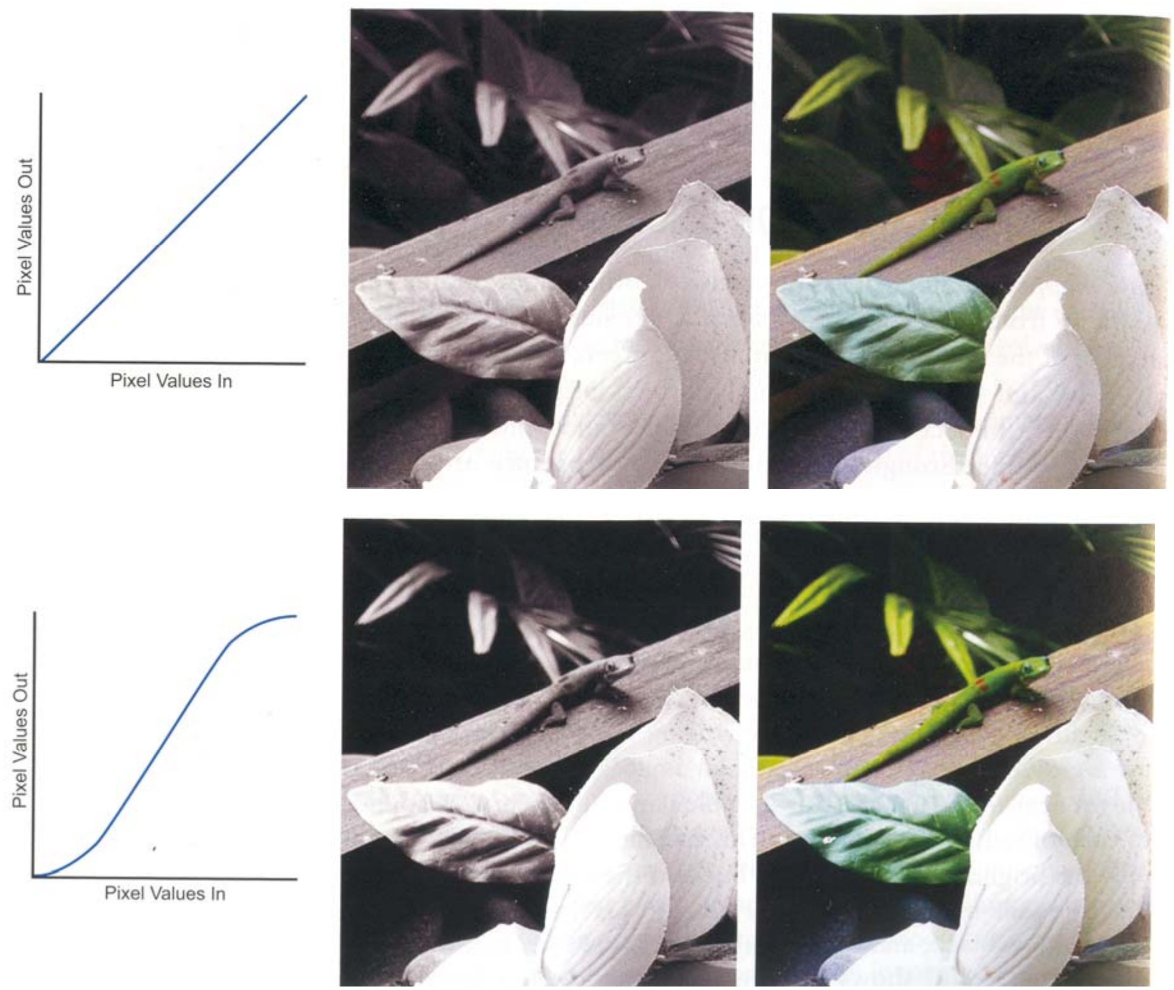
What is gamma function?



$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix} \begin{bmatrix} h_r(i_r) \\ h_g(i_g) \\ h_b(i_b) \end{bmatrix}$$

Tone Mapping Operator

- How the input value maps to output intensity
- Affects brightness, contrast and saturation



Tone Mapping Operator

- How the input value maps to output intensity
- Affects brightness, contrast and saturation

Transfer Function

- Monotonic, smooth with no flat regions
- Brightness and contrast controls

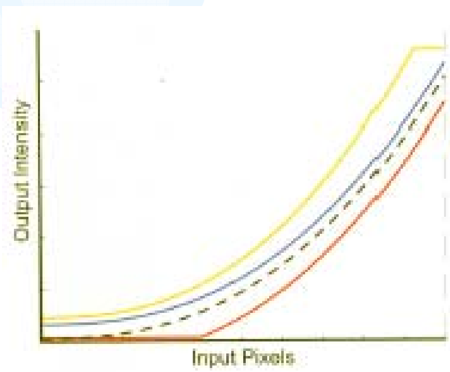
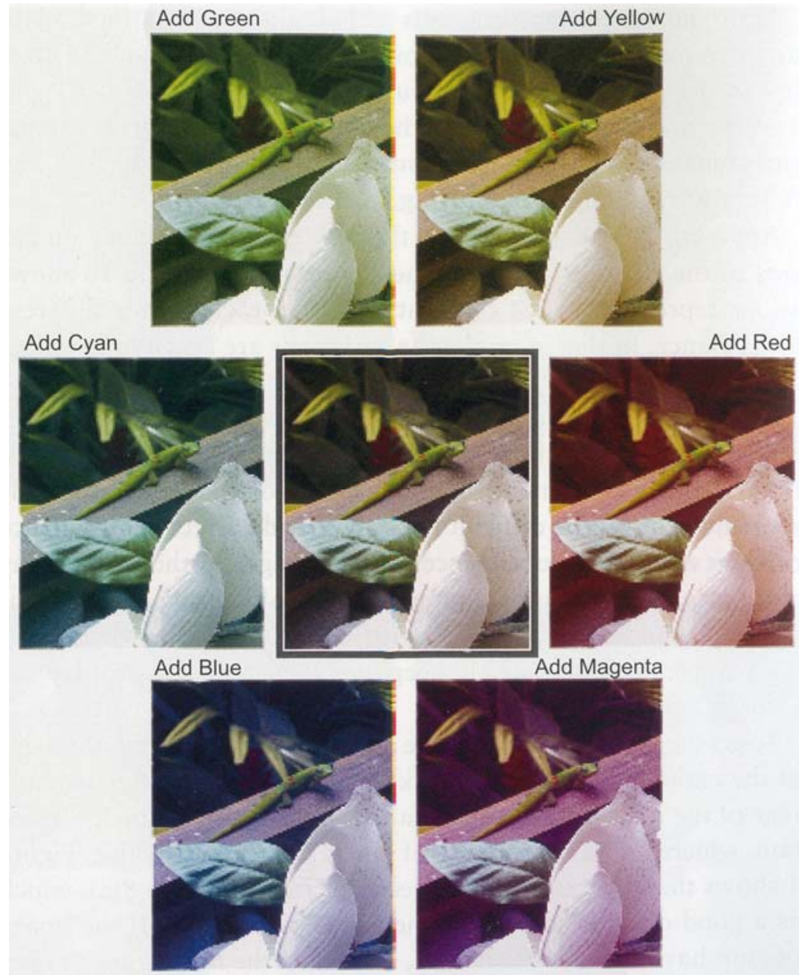


Image Correction

Color Balance

- Relative proportions of primaries while forming a color
- Affects hue, saturation and brightness
- Can be changed by changing the transfer function

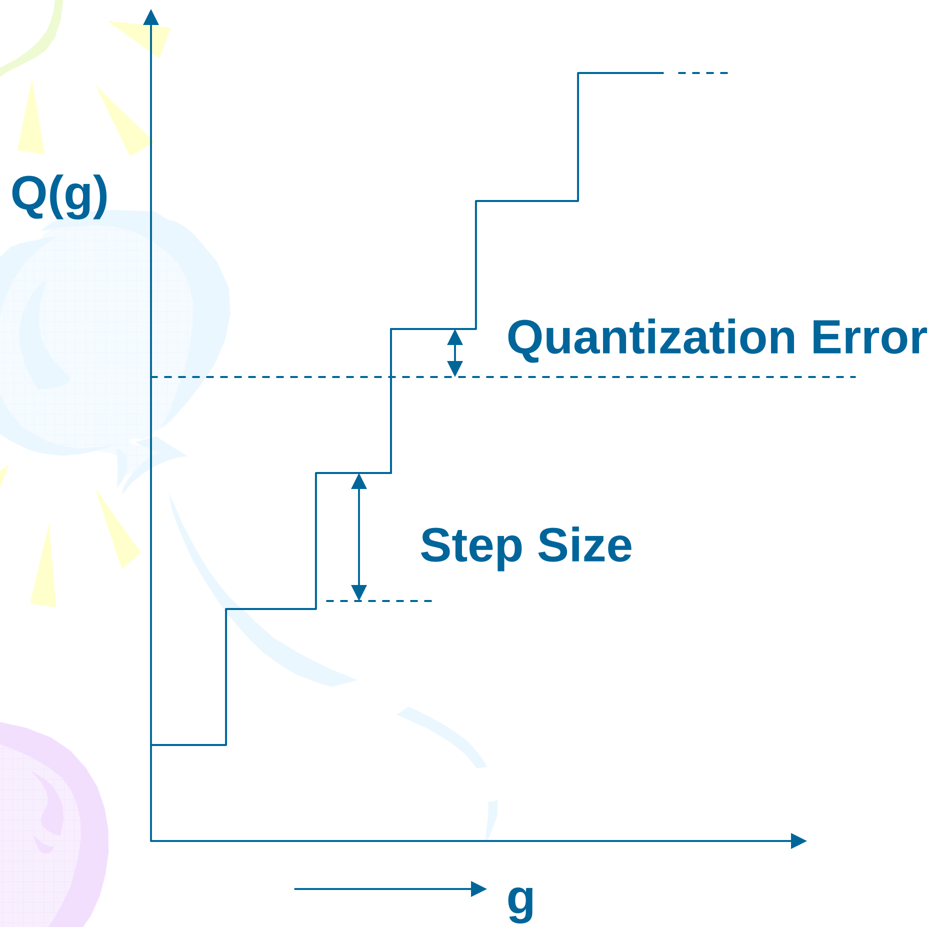


Color Balancing

Quantization

- Digitization of color
- Gray scale – infinite grays between 0 and 1
  - 8 bit representation – 256 levels
  - A range of grays represented by a single value
- Any value is assigned to one of k values
- Choose number of levels and range of each level

Quantization Error



Uniform Quantization

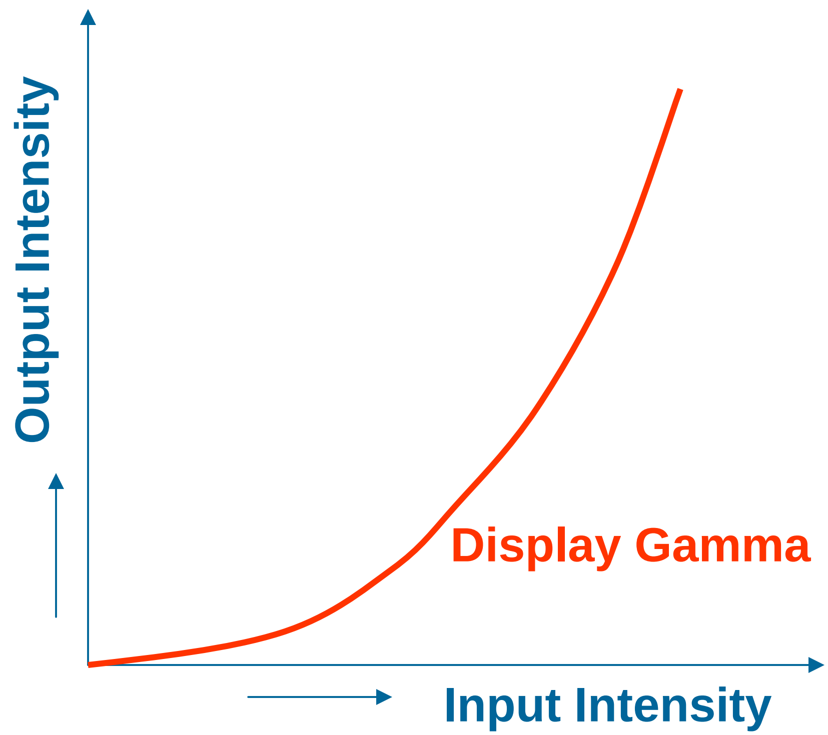
Maximum Error = ½ Step Size

Is it constant across all levels?

- Only when linear transfer function
- Usually non-linear transfer function
- Quantization error changes with input

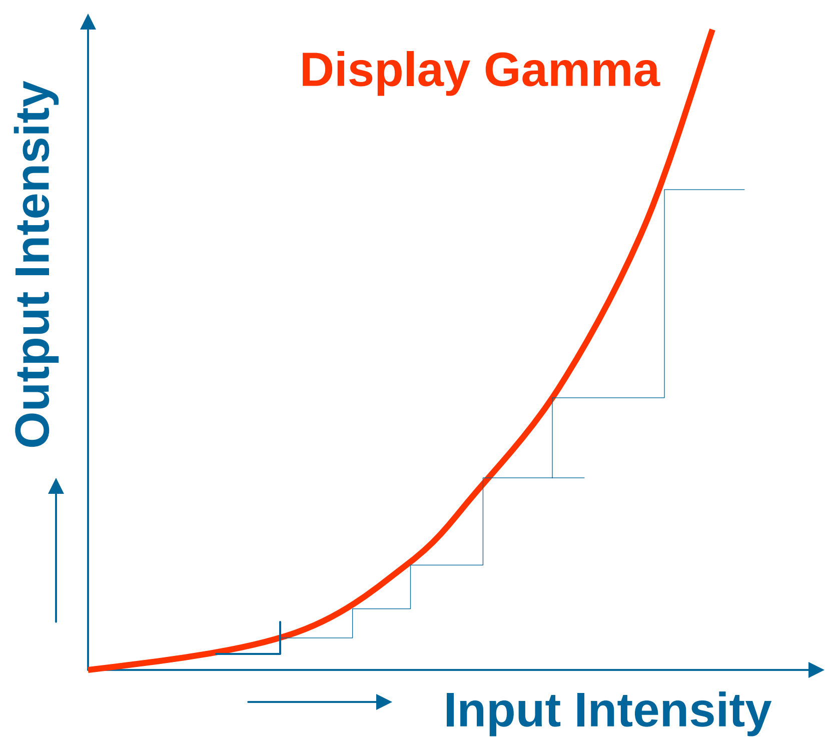
Gamma Function

- Usually a gamma transfer function
  - $O = I^{\gamma}$



Non-Uniform Quantization

- Note how quantization changes
- Non-uniform step size
- Maximum Error
  - ½ of maximum step size
- # of levels is the color resolution
  - # of bits



Color Resolution

Analog Image

4 Steps

8 Steps

16 Steps

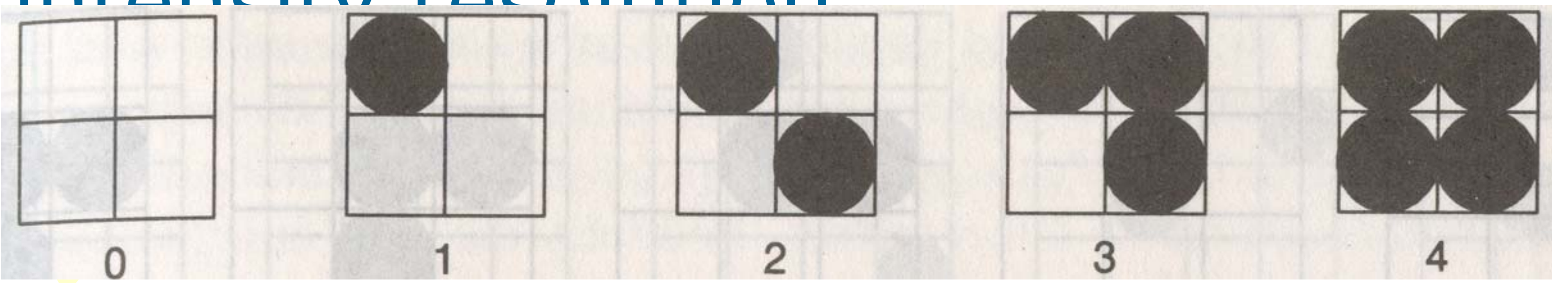
Quantization Artifacts

64 Steps

32 Steps

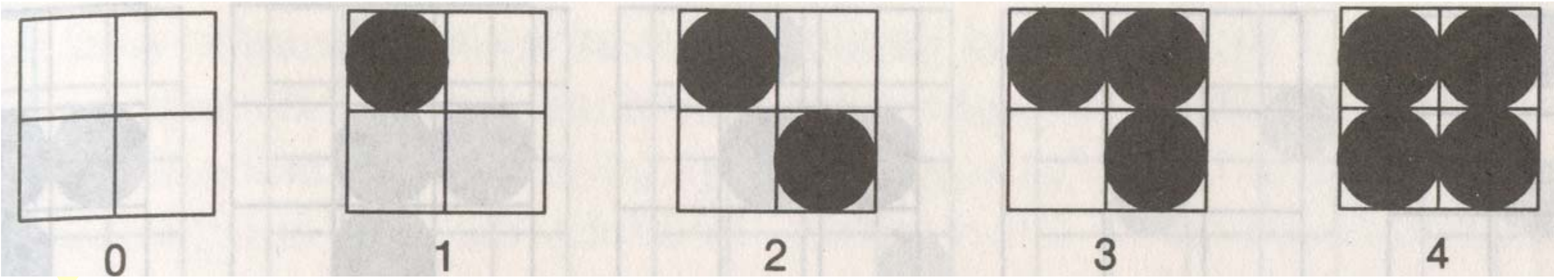
Dithering

- What if the color resolution is low?
  - Newsprint – Bi-level, only black and white
- Can we expand the # of colors?
  - Spatial integration of eye
- Trading off spatial resolution for intensity resolution



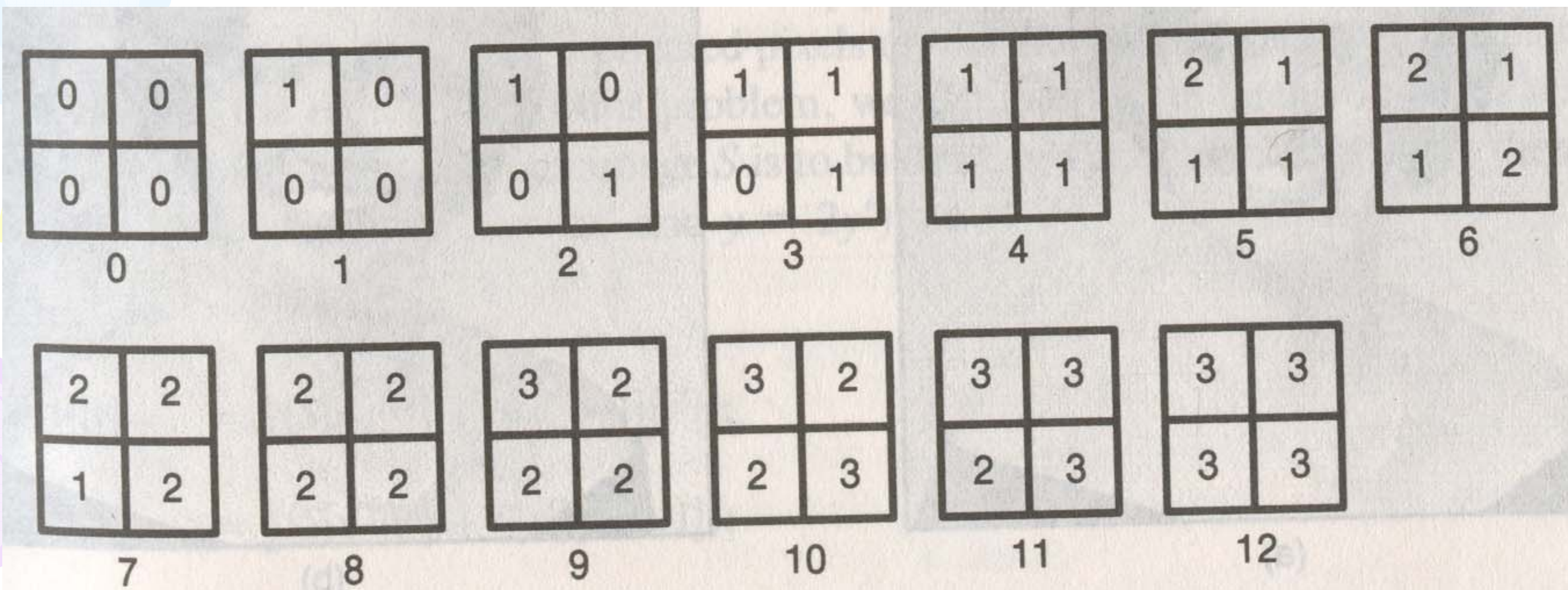
Dithering

- Represented by a dither matrix $\begin{bmatrix} 0 & 2 \\ 1 & 3 \end{bmatrix}$
- nxn pixels, bi-level intensity, can produce n^2+1 intensities
- If more than two levels – k levels
  - $n^2 . (k-1) + 1$
  - Used for increasing the color resolution

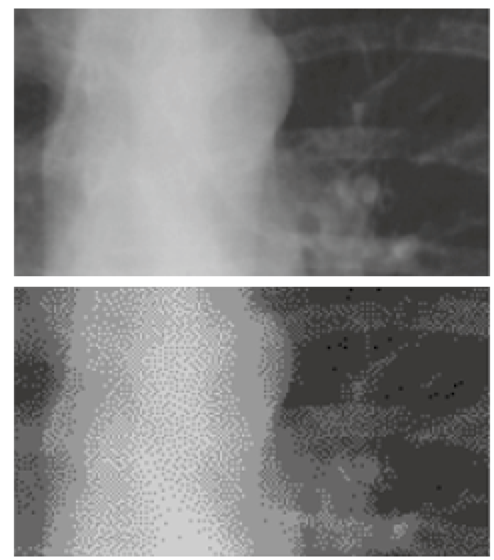


Dithering

- If more than two levels – k levels
  - $n^2 . (k-1) + 1$
  - For k = 4 (0,1,2,3) and n=2



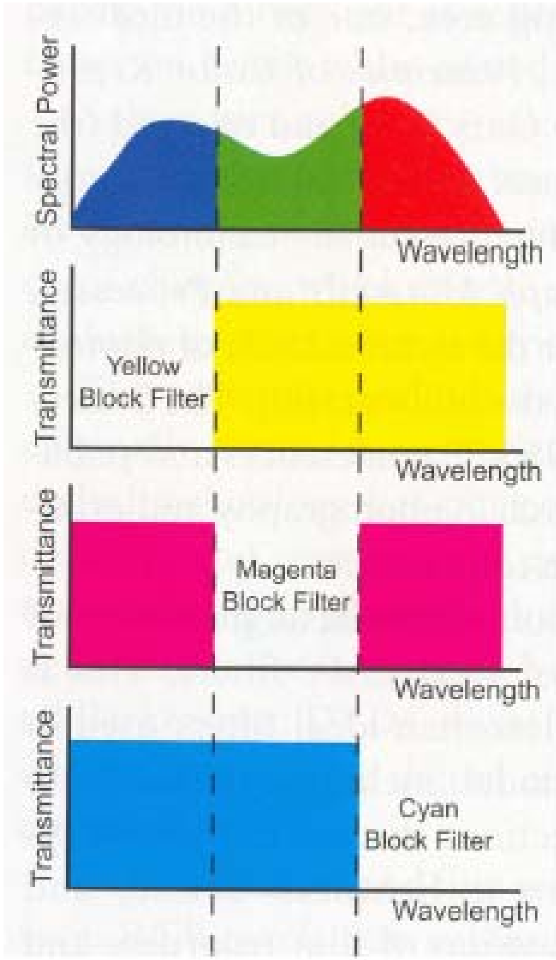
Examples



Loss of tone and details
(Intensity and Spatial Resolution)

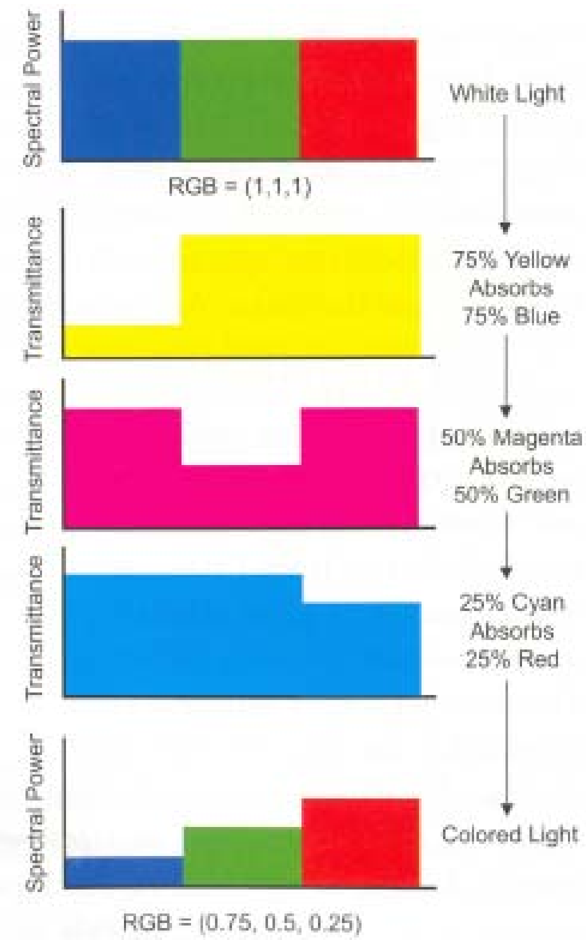
Subtractive Color System

- Layers of cyan, yellow and magenta dyes
  - Absorb red, blue and green light
- Depends on the illuminant
- Act as absorption filter
  - Ideally block filters
- Overlaying all the three dyes absorbs all wavelengths creating black



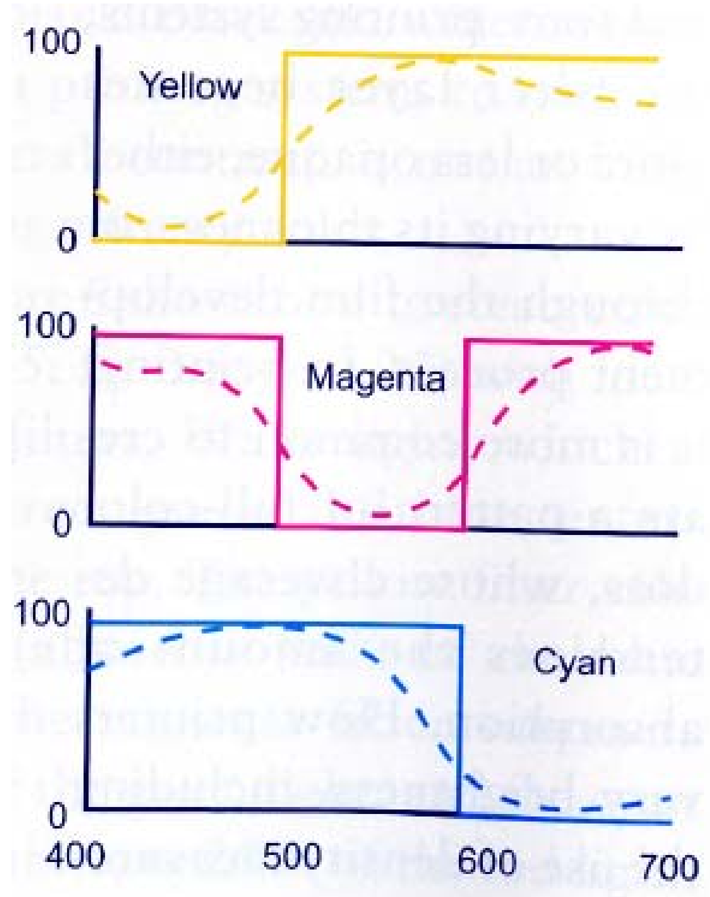
Creation of a color

- CMY = (1, 1, 1) – RGB
- (0.25, 0.5, 0.75) = (1, 1, 1) – (0.75, 0.5, 0.25)
- This works only for block filters



Real Fiilters

- Are not block filters
- Cross talk across different filters
- Due to ink impurities
- Grays should be formed by equal amount of three primaries
  - Seldom happens



Why use black?

- Better contrast
- Use of inexpensive black in place of expensive colored dyes
- Superimposing multiple dyes cause tearing of wet paper
- K for key
- Not an independent primary
  - Hence makes dark colors darker

How to use black?

- Initially only for neutral colors
  - Called *undercolor removal (UCR)*
- Colors with three components
  - Minimum of the three is the gray component
- Full gray component replacement
  - Only in inkjets where registration is a problem
- Partial gray component replacement
  - To achieve the best contrast

Gray Balancing

- The first step in printing is to decide how much of GCR to be used for the neutral grays
- However, every gray needs to be decided separately
- Called gray balancing
- Usually done by iteration
- No simple tristimulus model to decide components

Dependency on Content

- Discussed content independent
- Can also be done by understanding the color distribution of the particular content
- Usually non-linear

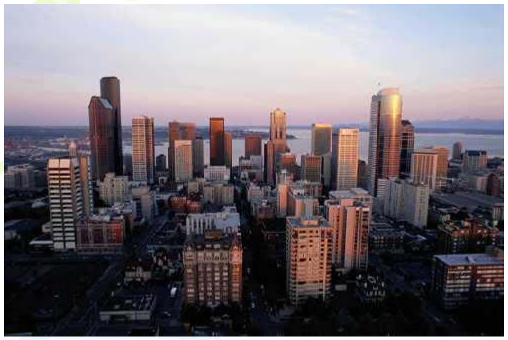
Image Compositing

Mosaic Blending

Image Compositing

Compositing Procedure

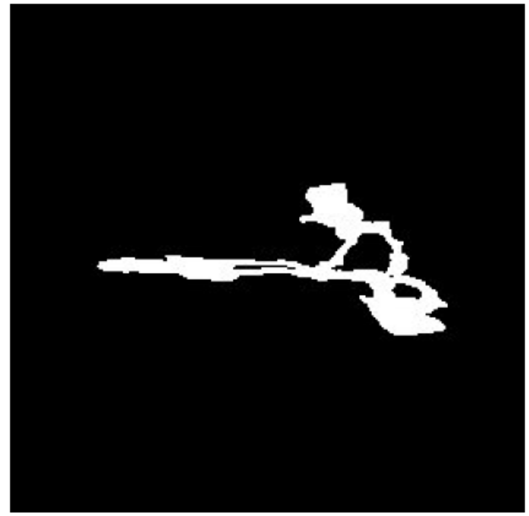
1. Extract Sprites (e.g using *Intelligent Scissors* in Photoshop)



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

Composite by
David Dewey

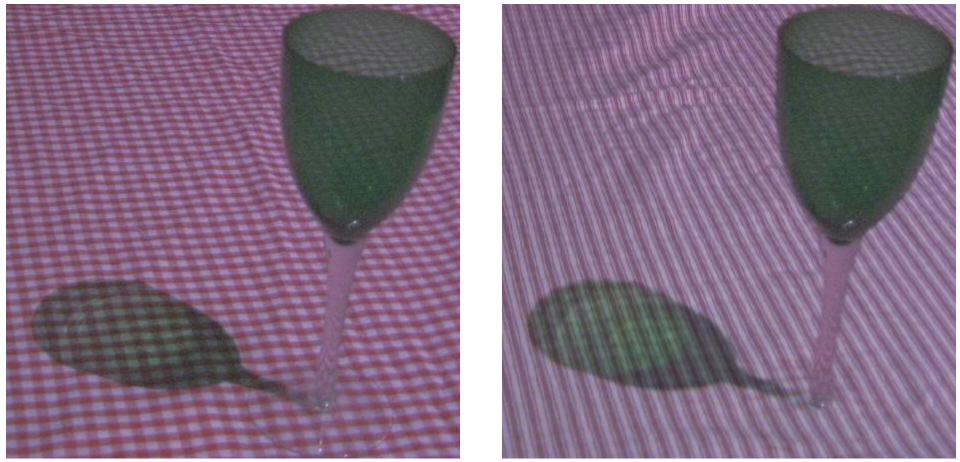
Replacing pixels rarely works



Binary mask

Problems: boundries & transparency (shadows)

Two Problems:



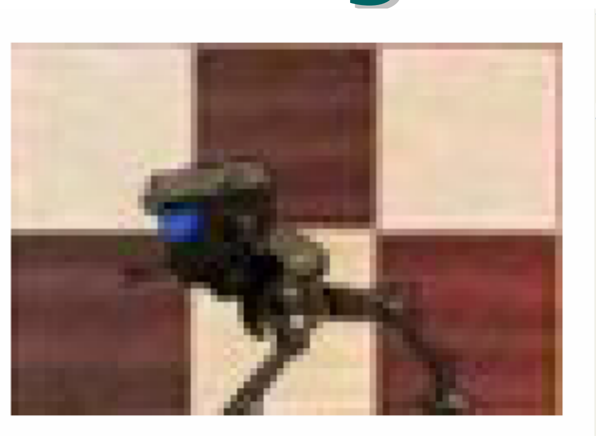
Semi-transparent objects

Pixels too large

Alpha Channel

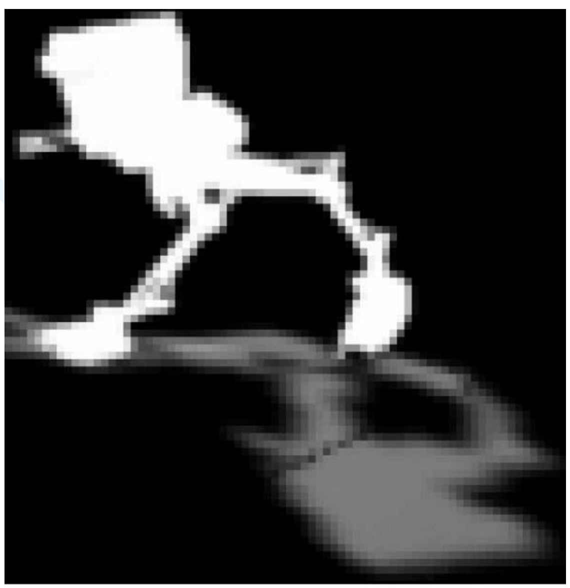
- Add one more channel:
  - Image(R,G,B,alpha)
- Encodes transparency (or pixel coverage):
  - Alpha = 1: opaque object (complete coverage)
  - Alpha = 0: transparent object (no coverage)
  - 0<Alpha<1: semi-transparent (partial coverage)
- Example: alpha = 0.3

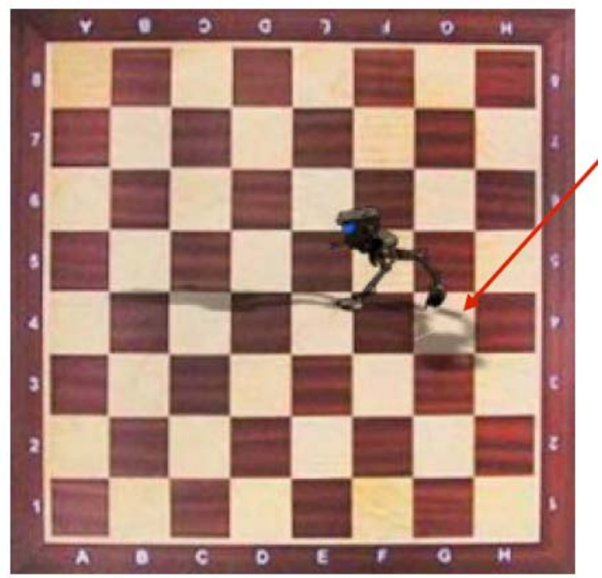
Alpha Blending



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

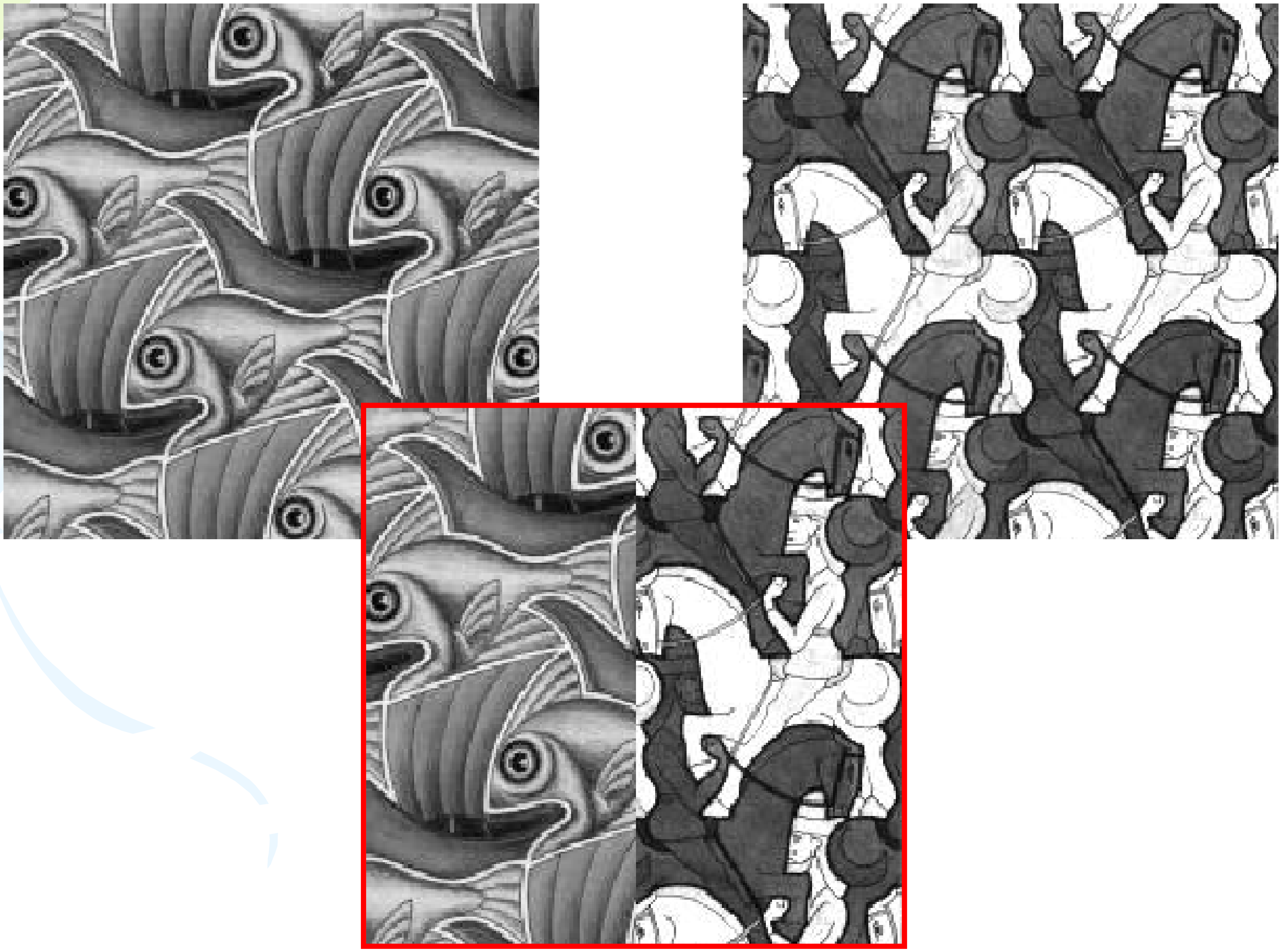
alpha mask





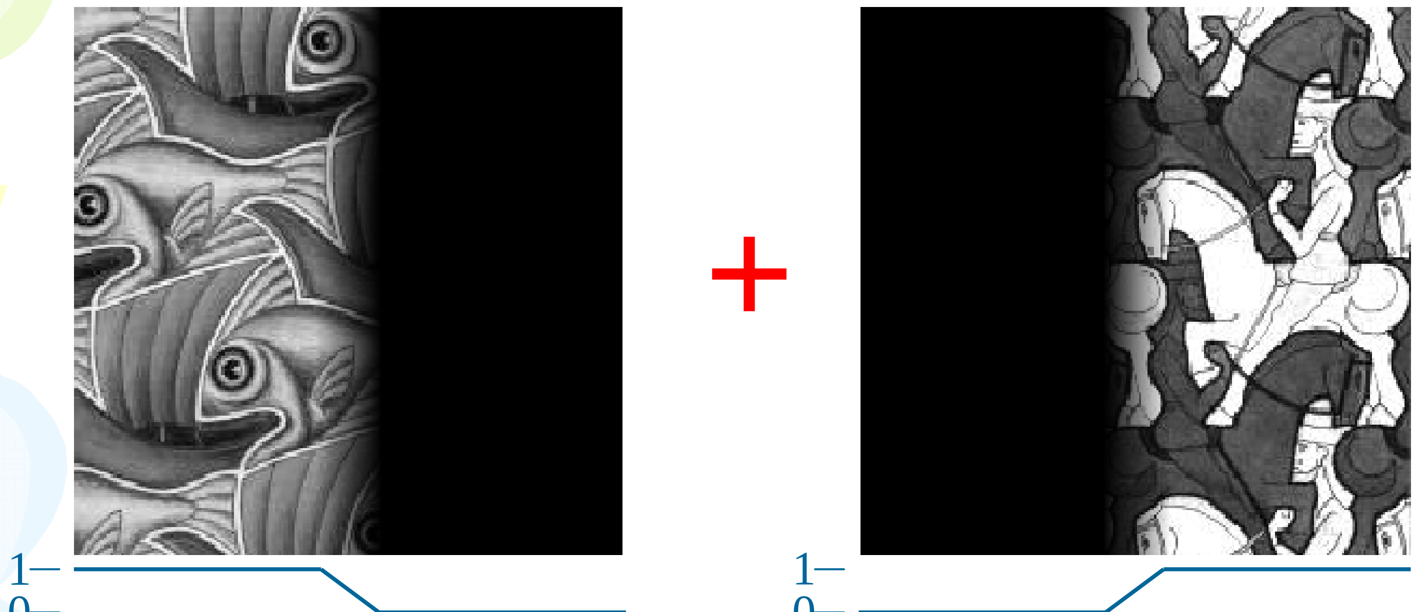
shadow

Alpha Hacking…



No physical interpretation, but it smoothes the seams

Feathering



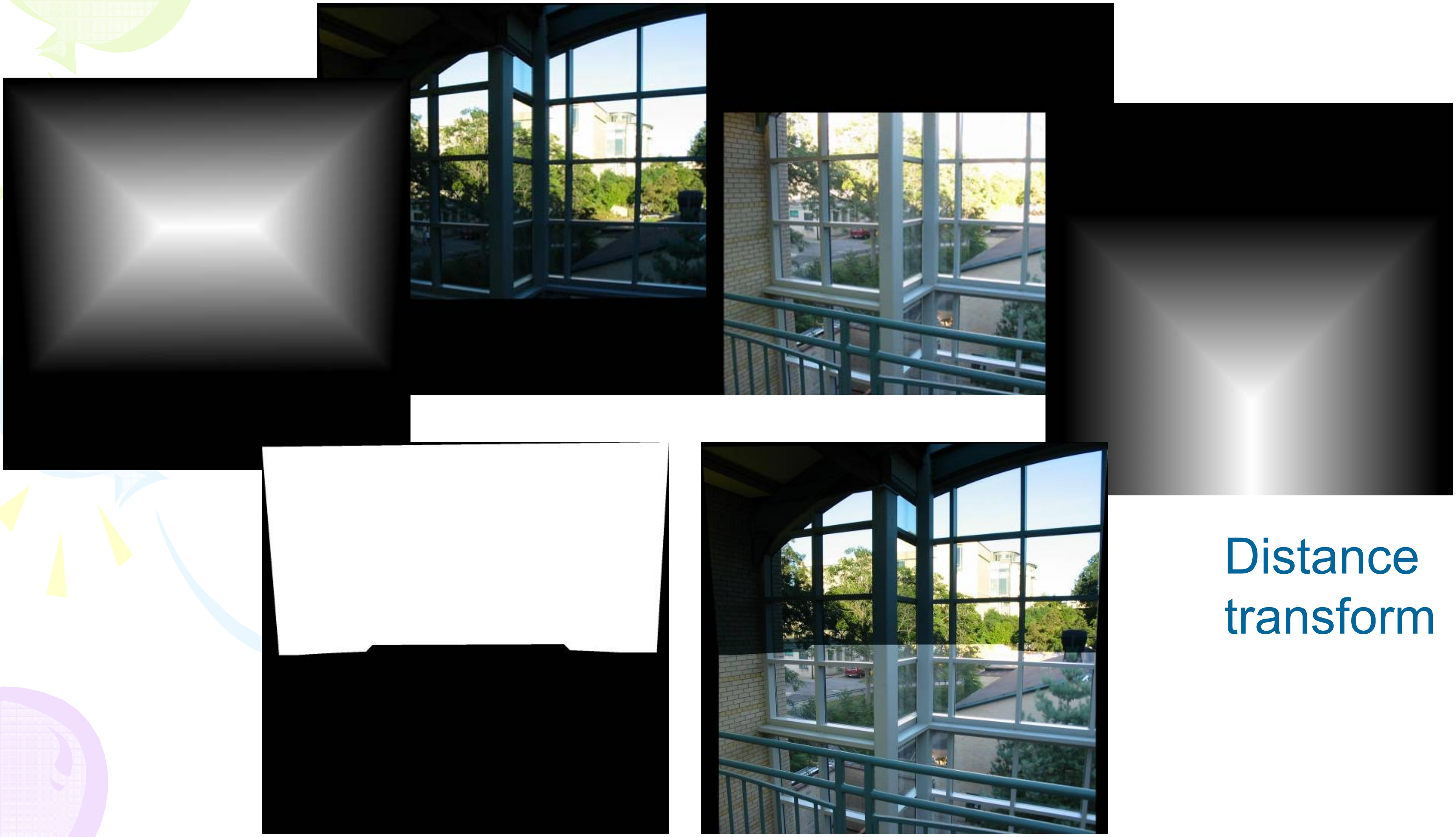
Encoding as transparency

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

Setting alpha: simple average

Alpha = .5 in overlap region

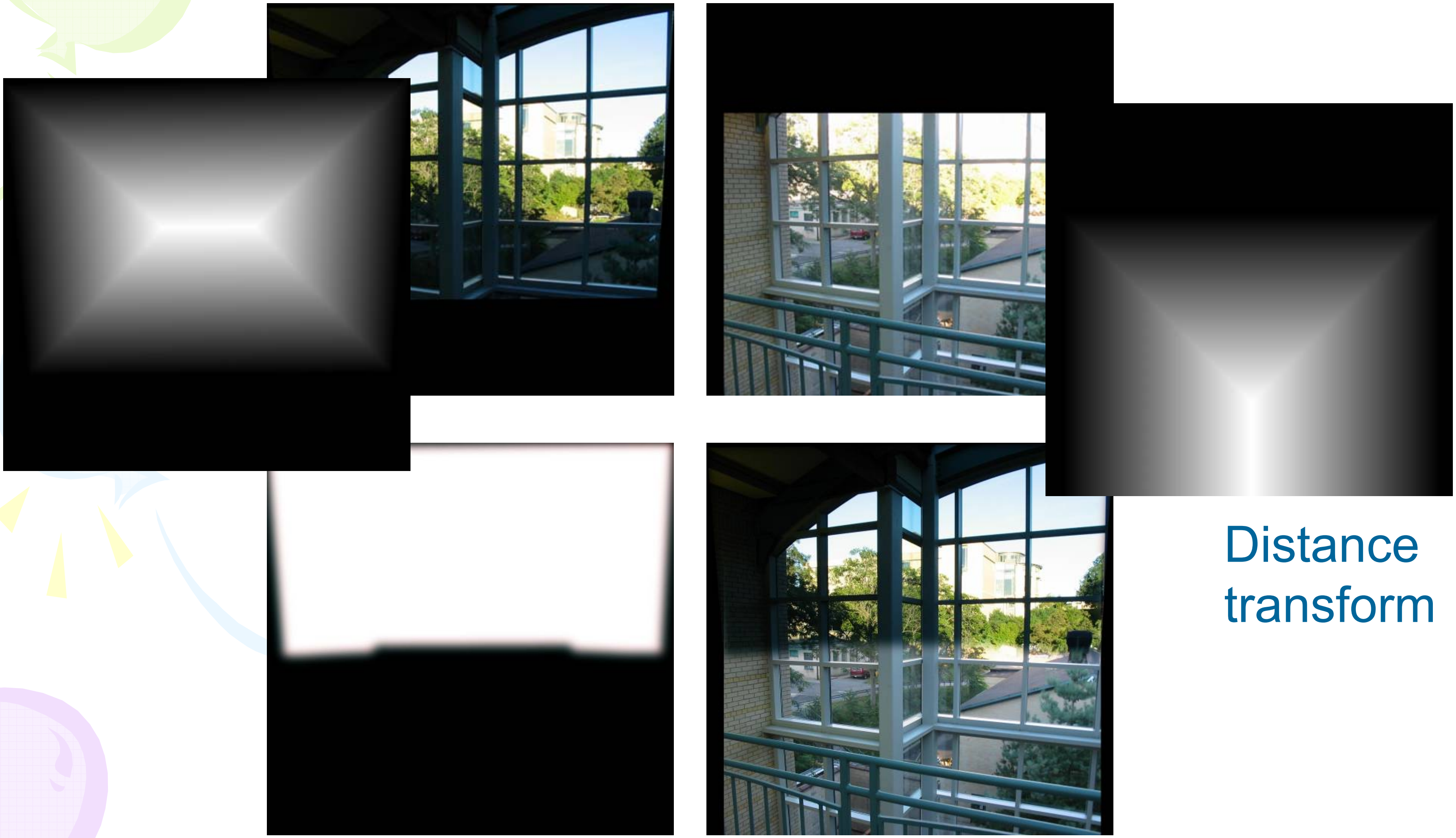
Setting alpha: center seam



Distance transform

Alpha = logical(dtrans1>dtrans2)

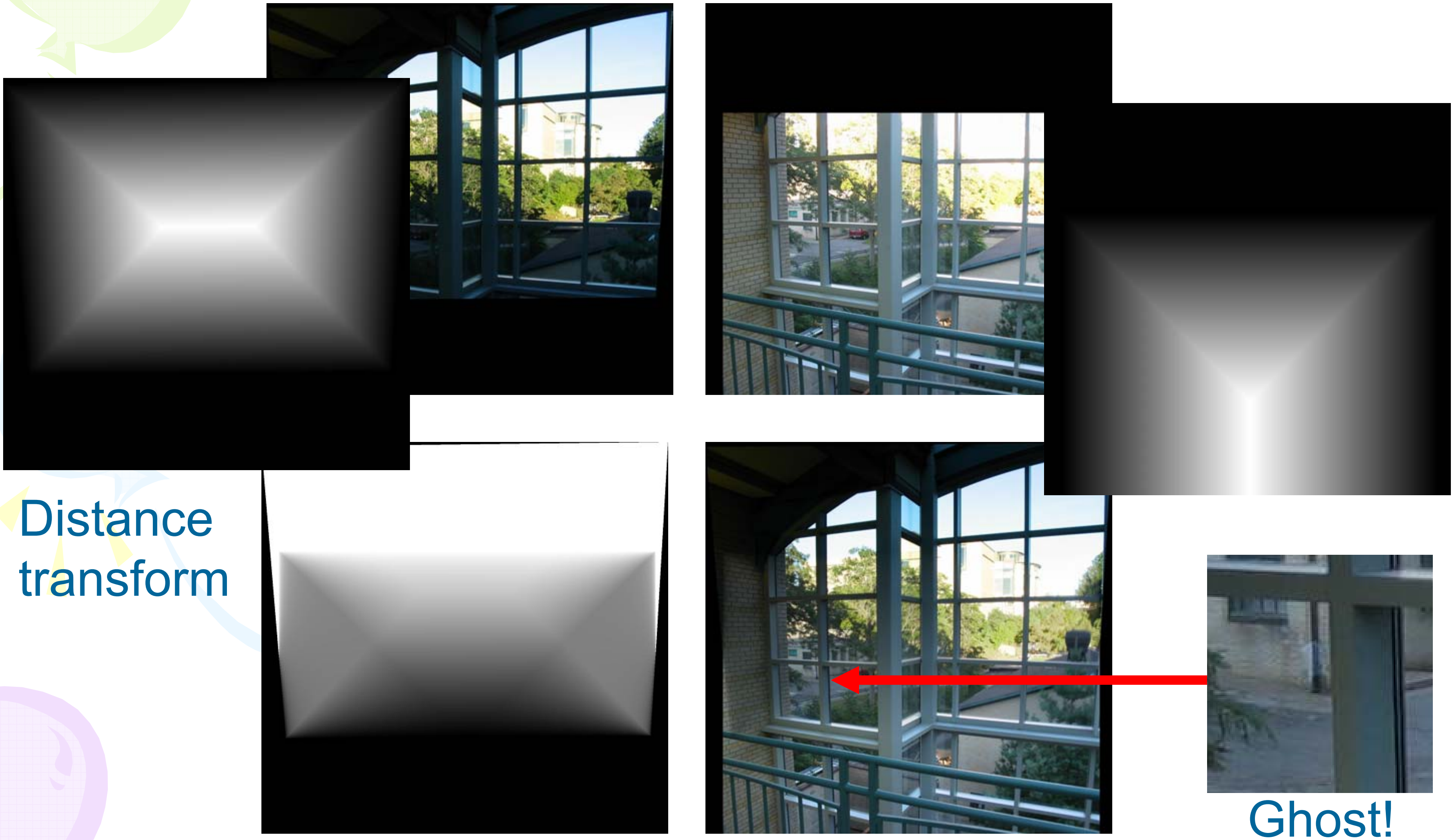
Setting alpha: blurred seam



Distance transform

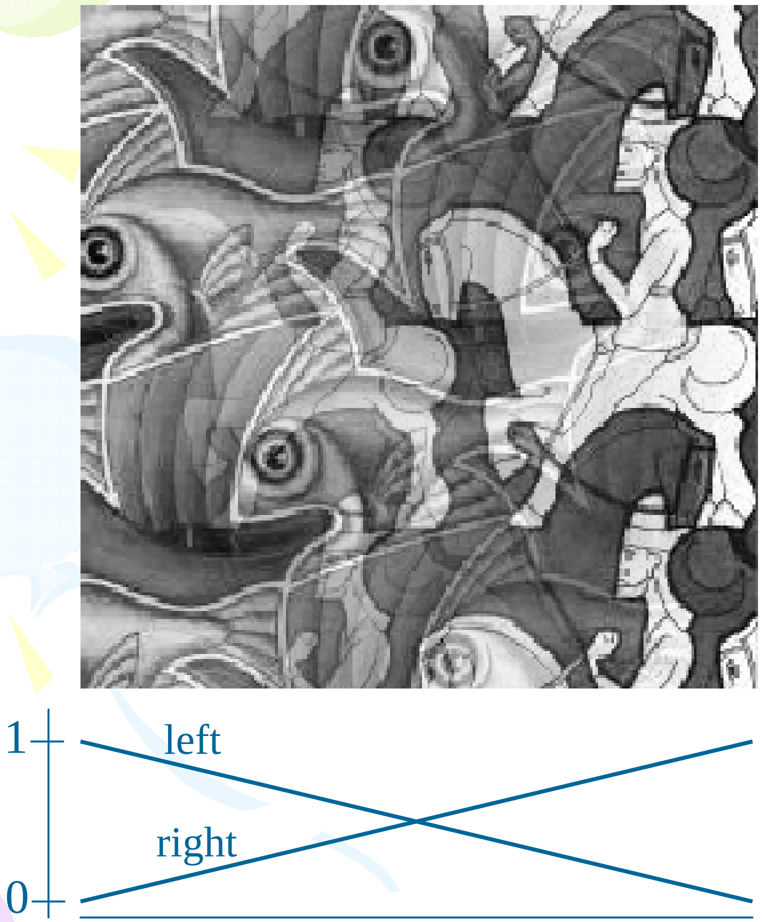
Alpha = blurred

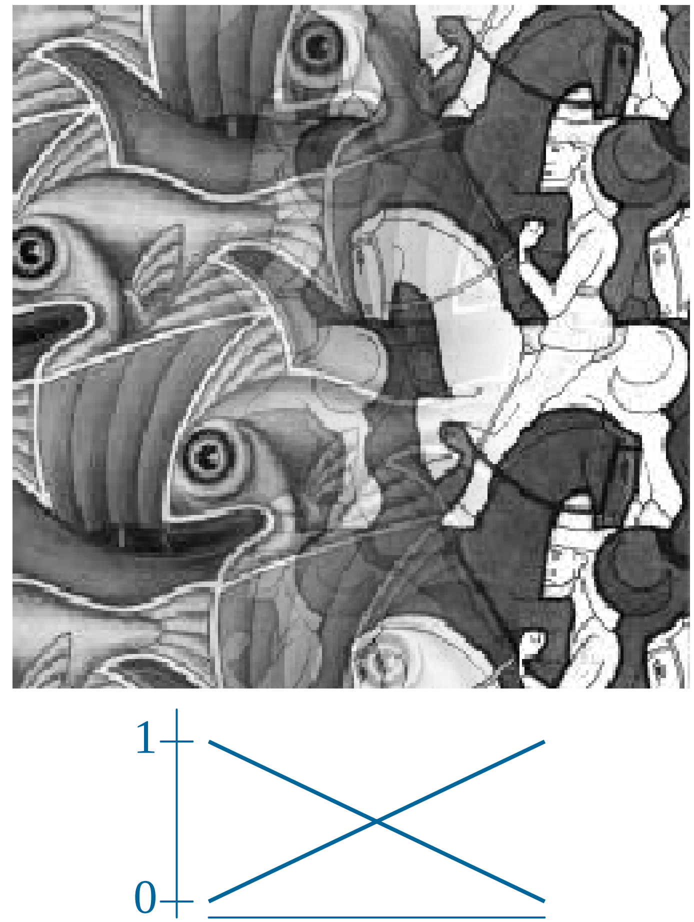
Setting alpha: center weighting



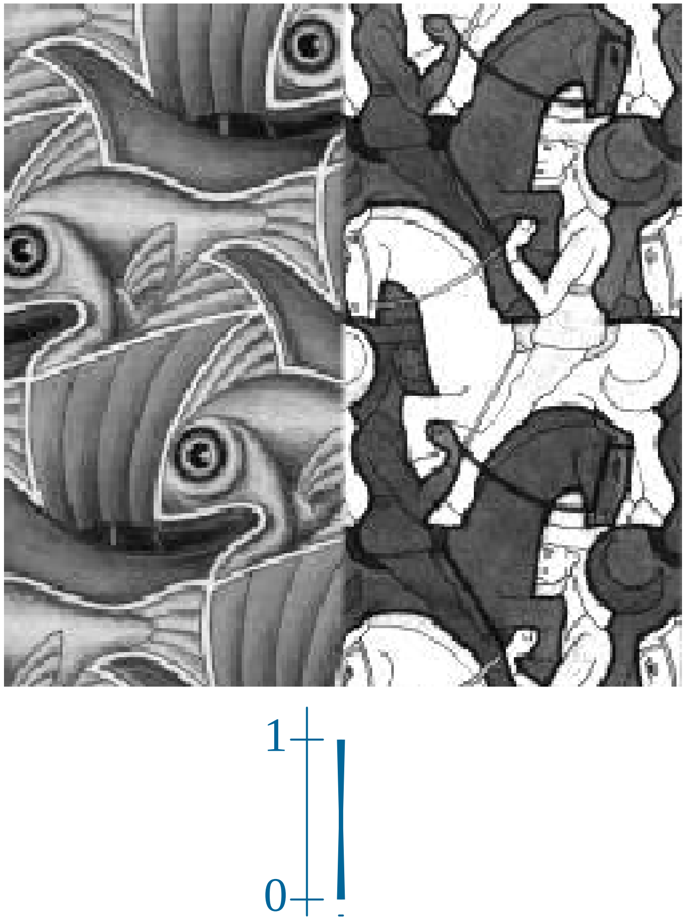
Alpha = dtrans1 / (dtrans1+dtrans2)

Affect of Window Size

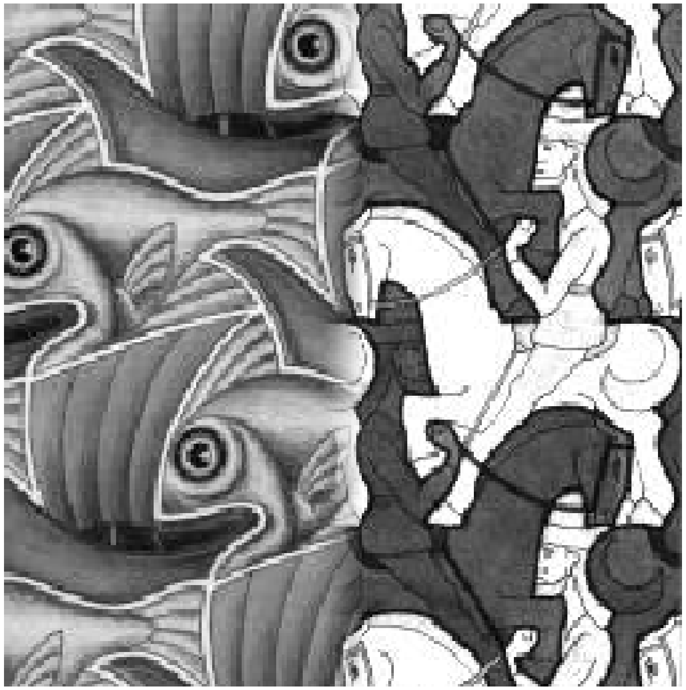


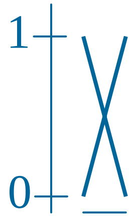


Affect of Window Size



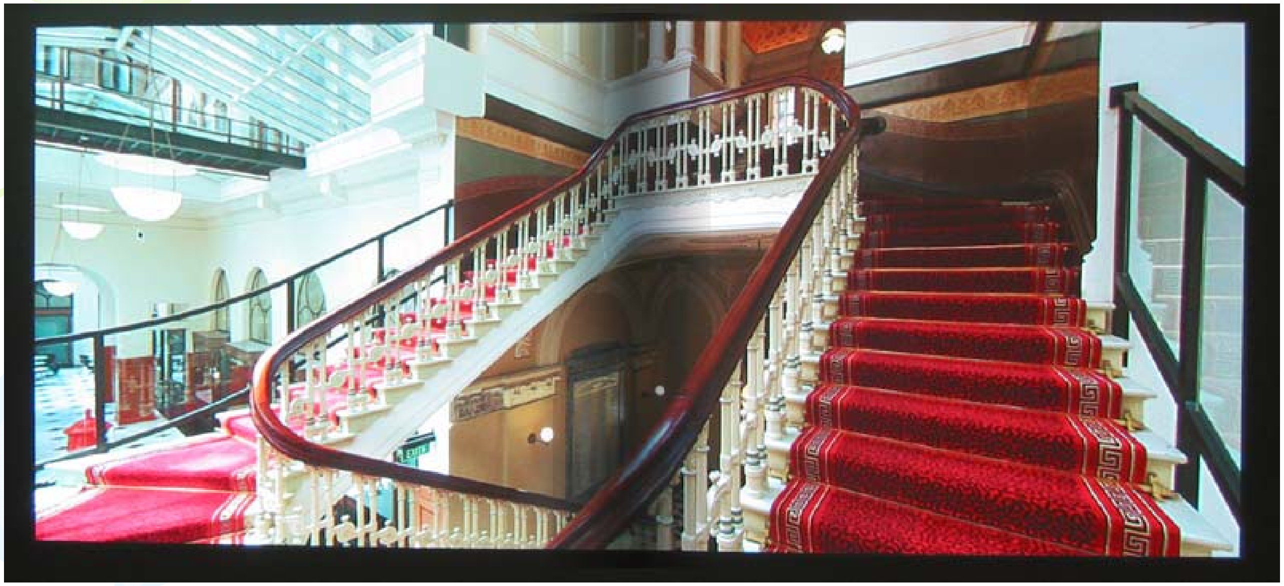
Good Window Size



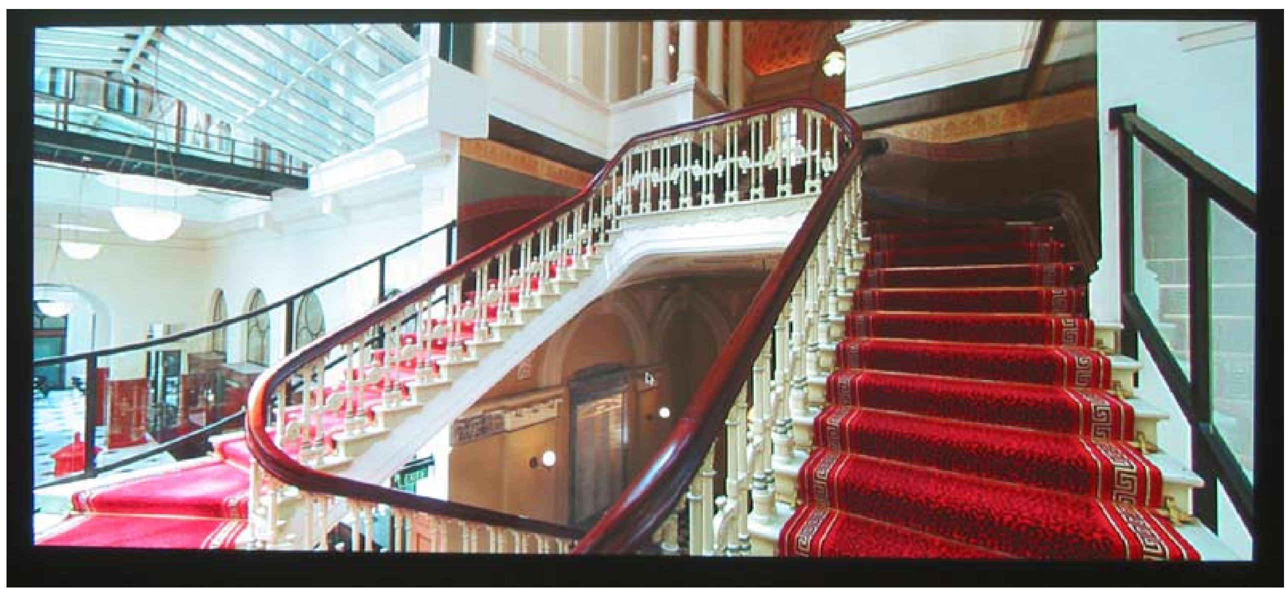


"Optimal" Window: smooth but not ghosted

Type of Blending function



Linear
(Only function continuity)



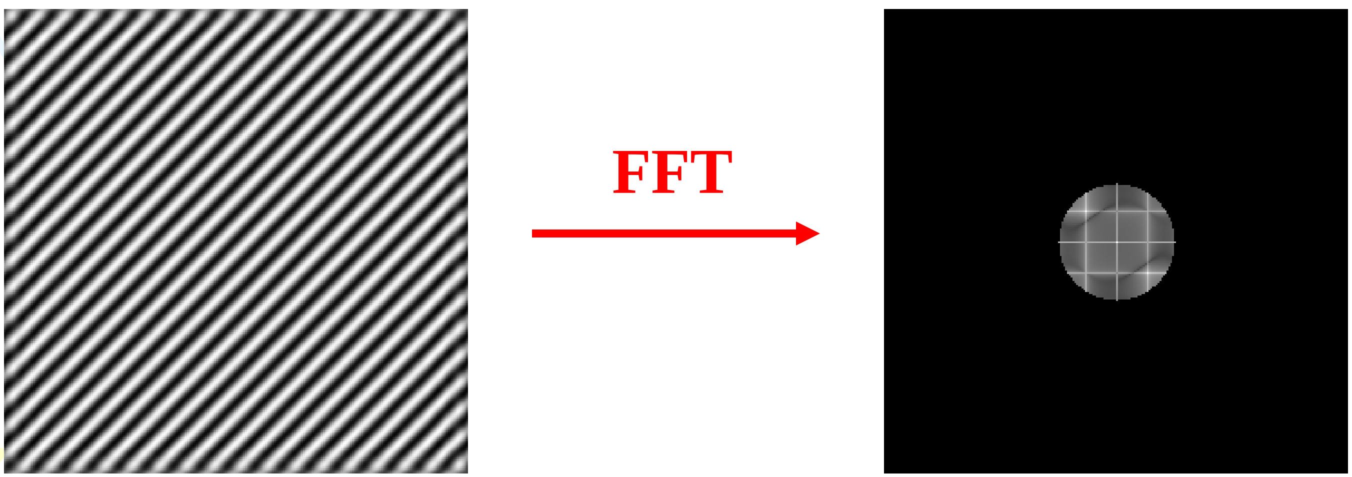
Spline or Cosine
(Gradient continuity also)

What is the Optimal Window?

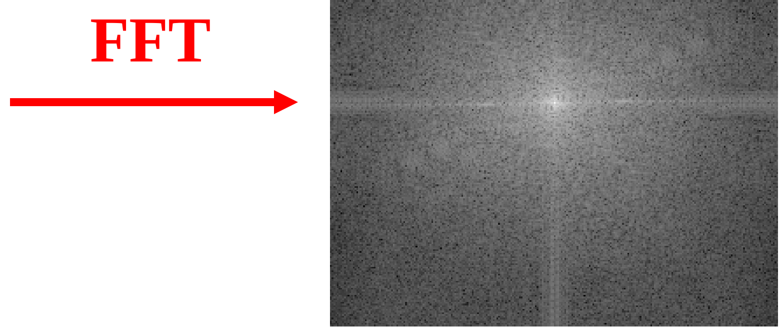
- To avoid seams
  - window = size of largest prominent feature
- To avoid ghosting
  - window <= 2*size of smallest prominent feature

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)



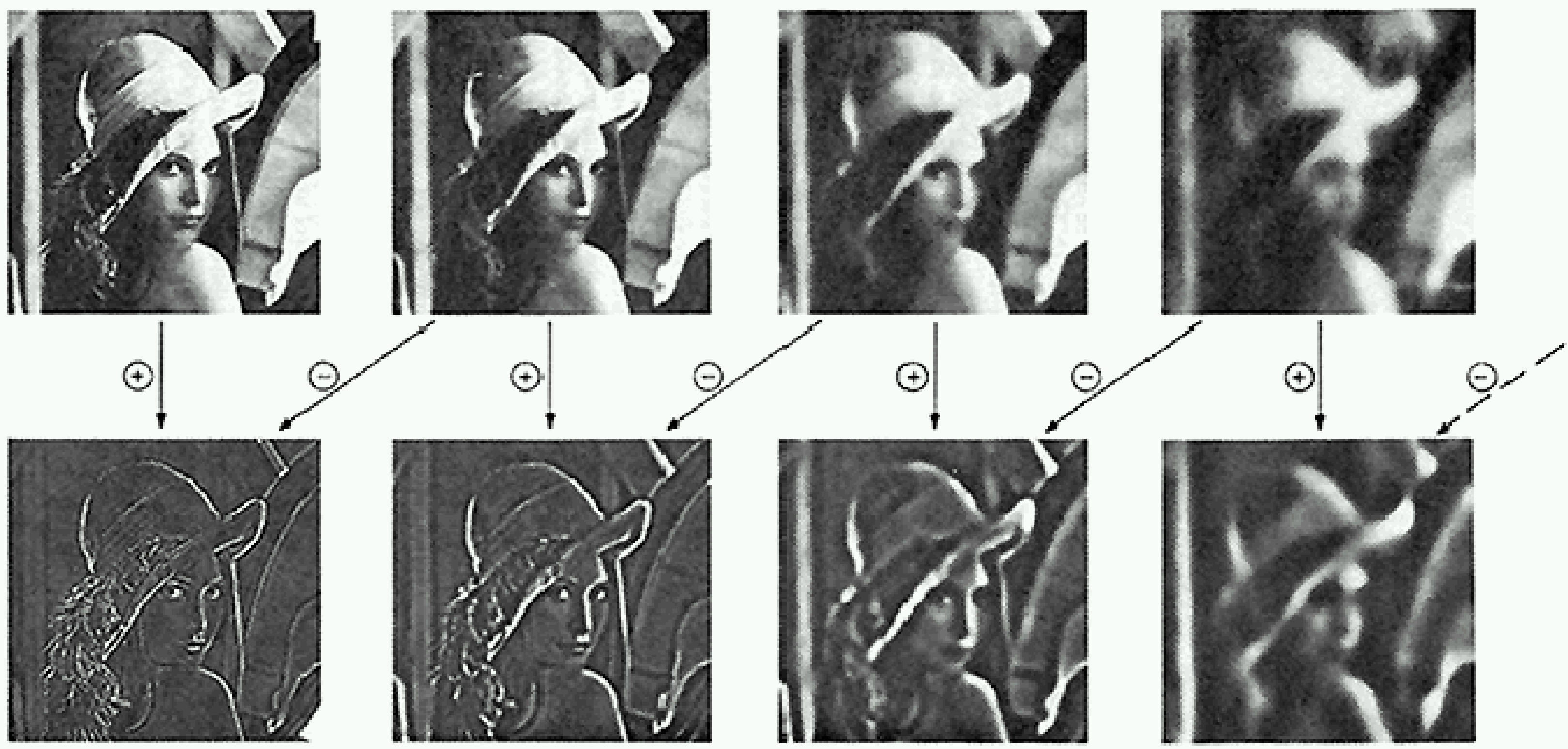
Frequency Spread is Wide



- 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

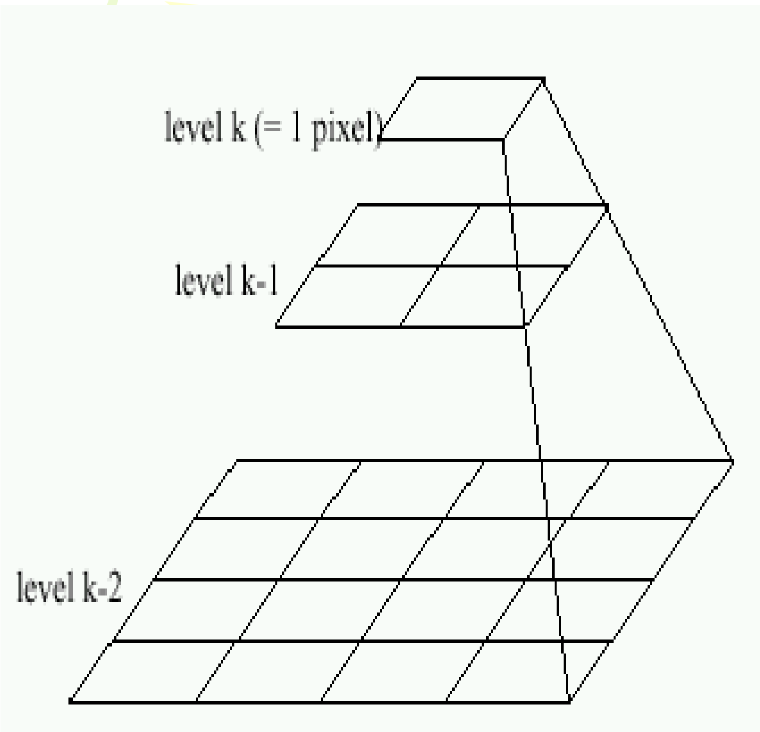
Octaves in the Spatial Domain

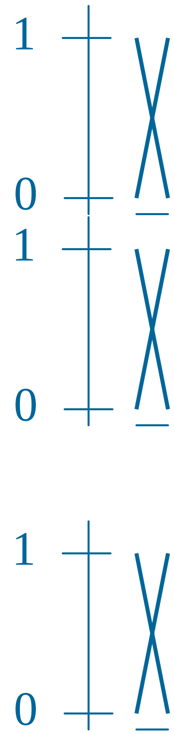
Lowpass Images

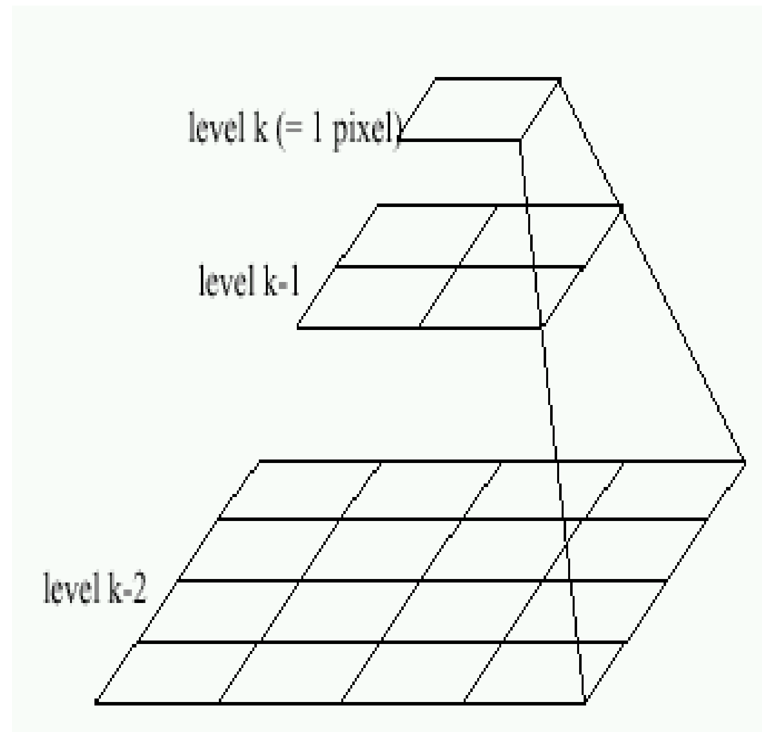


- Bandpass Images

Pyramid Blending





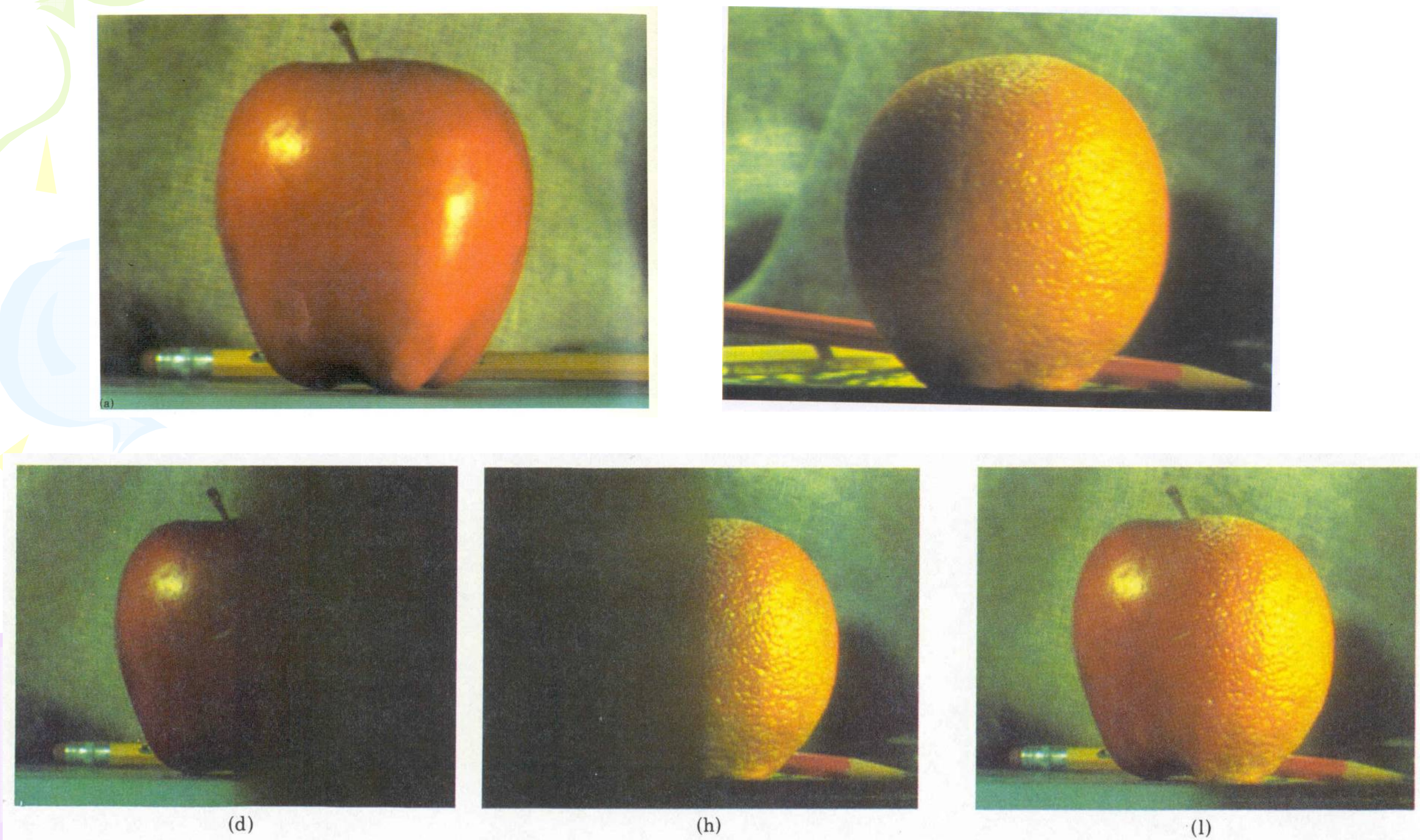


Left pyramid

blend

Right pyramid

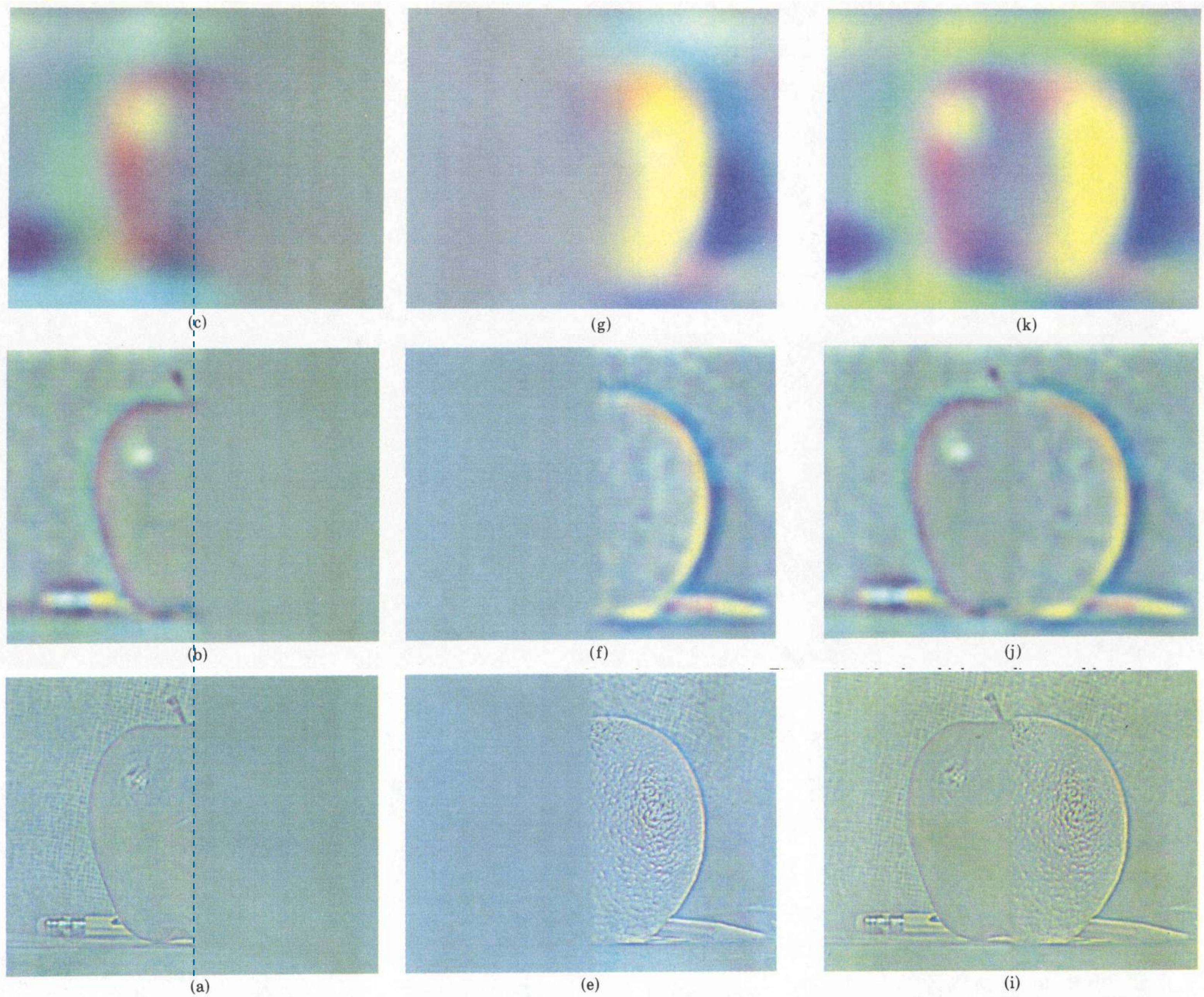
Pyramid Blending



laplacian level 4

laplacian level 2

laplacian level 0



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