Media and Features

Text vs. images

- Text
  - Symbolic
  - Artificial
  - Single meaning (reader independent ?)
  - Small storage requirements
- Images
  - Visual
  - Natural, artificial
  - Multiple meanings (viewer dependent)
  - Large storage requirements

Processing vs. analysis..

- **Image processing**

- **Image analyzing**

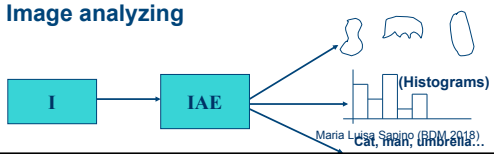


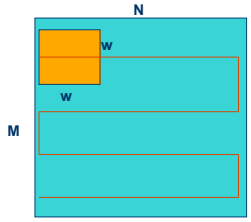
Image processing

- Operators
  - Sharpening
  - Bluring
  - Rotating
  - Translating
  - Brightening
  - Cut/paste/resize
  - Warping

Feature extraction

- Image analysis gives the features
  - Color histogram
  - Texture
  - Edges
  - Shapes
  - Objects
  - Semantics
  - Depth (stereoscopic images)
- Many of these operators require a filter to be convoluted over the original image

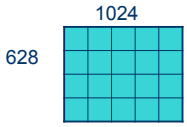
Processing requirements...



Cost of the operation: M × N × w × w

Storage size

- # of pixel: 628 × 1024
- # of bytes: 628 × 1024 × 3



What to do???

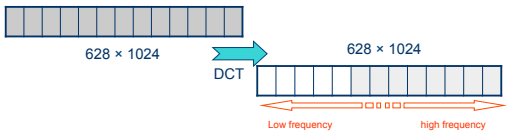
- Change coding
  - Data have coding redundancies
    - Get rid of them
  - Loseless!
- Examples:
  - Static/dynamic Huffman coding
    1. Count the character frequencies
    2. Find patterns that are repeated (less information!!)
    3. Replace them with shorter codewords
  - Aritmetic coding
  - LZW (Lempel-Ziv-Welch) coding

What to do???

- Lossy compression
  - Image may contain details that human eye can not recognize
  - Use domain transformation
    - Convert images from spatial domain to frequency domain
      - DCT (discrete cosine transform)
      - DFT (discrete fourier transform)
      - DWT (discrete wavelet transform)
    - Get rid of the frequencies which do not contain information

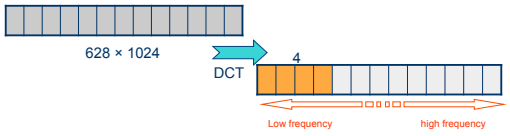
Domain transformations

- Given a signal of length “n” you get a sequence of “n” frequencies



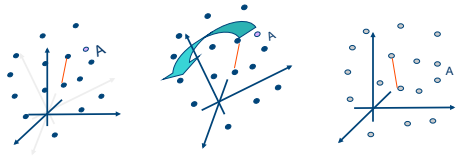
Domain transformations

- Given a signal of length “n” you get a sequence of “n” frequencies



Domain transformations

- Given a signal of length “n” you get a sequence of “n” frequencies
  - preserves euclidean distance

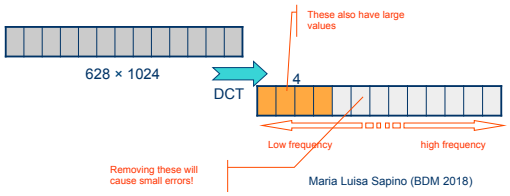


Domain transformations

- Given a signal of length “n” you get a sequence of “n” frequencies
  - not sensitive to shifts!
    - take an image and shift it…both images will be mapped to the same point

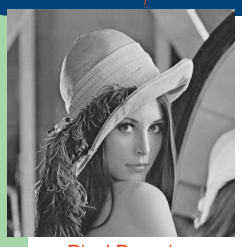
Domain transformations

- Given a signal of length “n” you get a sequence of “n” frequencies
  - concentrates energy better



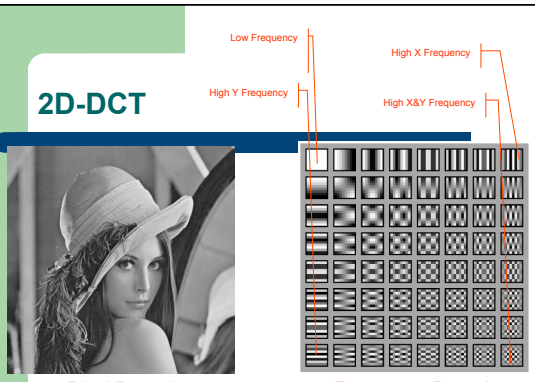
2D-DCT

Meet “Lena”: this picture is the most commonly used benchmark in image processing

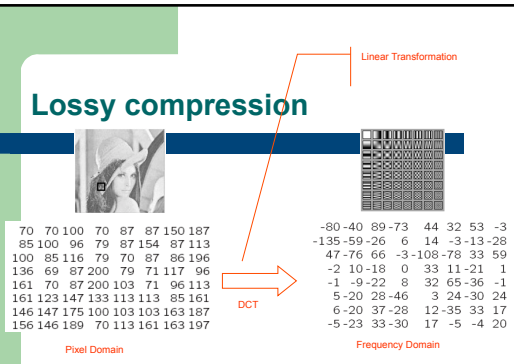


Pixel Domain

2D-DCT



Lossy compression



$$F(u,v) = \frac{C(u)C(v)}{4}\sum_{i=0}^{7}\sum_{j=0}^{7}\cos\frac{(2i+1)u\pi}{16}\cos\frac{(2j+1)v\pi}{16}f(i,j)$$

Figure from Li&Drew 2003

Lossy compression

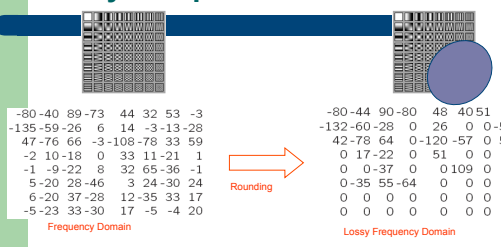
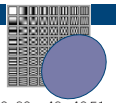


Figure from Li&Drew 2003

Lossy compression



Linear Transformation

-80	-44	90	-80	48	40	51	0
-132	-60	-28	0	26	0	0	-55
42	-78	64	0	-120	-57	0	56
0	17	-22	0	51	0	0	0
0	0	-37	0	0	109	0	0
0	-35	55	-64	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Lossy Frequency Domain

Inverse DCT

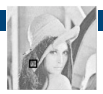
70	60	106	94	62	103	146	176
85	101	85	75	102	127	93	144
98	99	92	102	74	98	89	167
132	53	111	180	55	70	106	145
173	57	114	207	111	89	84	90
164	123	131	135	133	92	85	162
141	159	169	73	106	101	149	224
150	141	195	79	107	147	210	153

Lossy Pixel Domain

$$\hat{f}(i,j) = \sum_{u=0}^{7}\sum_{v=0}^{7} \frac{C(u)\,C(v)}{4} \cos\frac{(2i+1)u\pi}{16} \cos\frac{(2j+1)v\pi}{16} F(u,v)$$

Figure from Li&Drew 2003

Lossy compression



70	70	100	70	87	87	150	187
85	100	96	79	87	154	87	113
100	85	116	79	70	87	86	196
136	69	87	200	79	71	117	96
161	70	87	200	103	71	96	113
161	123	147	133	113	113	85	161
146	147	175	100	103	103	163	187
156	146	189	70	113	161	163	197

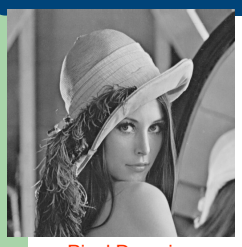
0	10	-6	-24	25	-16	4	11
0	-1	11	4	-15	27	-6	-31
2	-14	24	-23	-4	-11	-3	29
4	16	-24	20	24	1	11	-49
-12	13	-27	-7	-8	-18	12	23
-3	0	16	-2	-20	21	0	-1
5	-12	6	27	-3	2	14	-37
6	5	-6	-9	6	14	-47	44

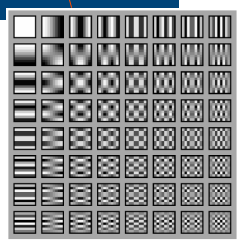
70	60	106	94	62	103	146	176
85	101	85	75	102	127	93	144
98	99	92	102	74	98	89	167
132	53	111	180	55	70	106	145
173	57	114	207	111	89	84	90
164	123	131	135	133	92	85	162
141	159	169	73	106	101	149	224
150	141	195	79	107	147	210	153

Figure from Li&Drew 2003

Meet "DCT bases": these are used in JPEG compression

2D-DCT



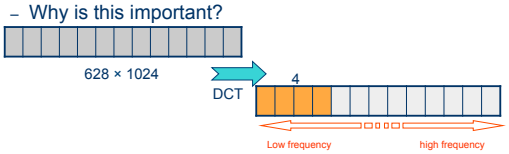


Pixel Domain

Frequency Domain

Under-estimation of distances

- If we remove some frequencies, this will cause underestimation of distances.
  - Why?
  - Why is this important?

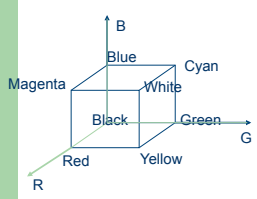


$\Delta > \Delta'$

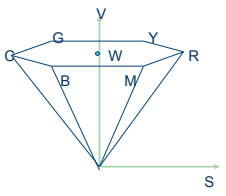
What to do???

- Lossy compression
  - Image may contain details that human eye can not recognize
  - Color table:
    1. reduce the number of colors to 256 (1 byte per pixel)
    2. Cluster similar colors into a single bucket and assign a single color to the bucket
    3. the set of buckets is called color table

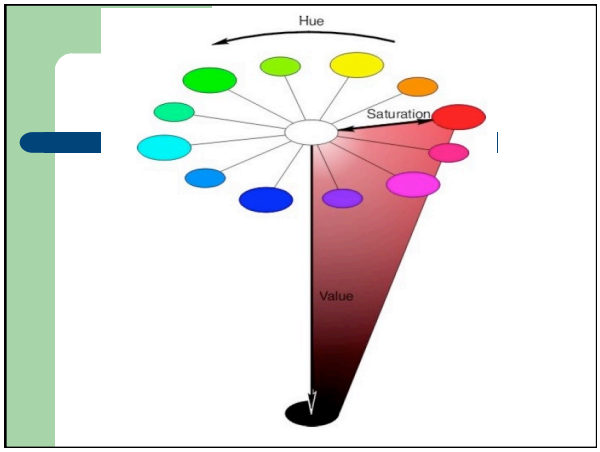
Alternative color spaces...



RGB (Red, Green, Blue)



HSV (Hue, Saturation, Value)

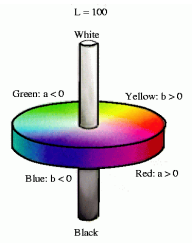


Color models

- RGB: describes colors in terms of the combinations of the intensities of Red, Green and Blue colors
- HSV
  - Hue: main color
  - Saturation: Amount of white
  - Value: Amount of energy

Why different color models?

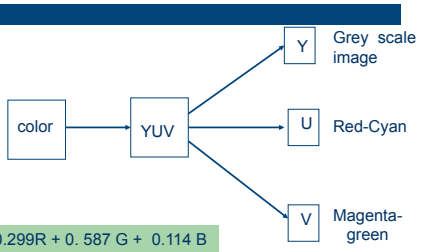
- Some color models reflect human perception better.
- Ex: CIELAB models the perceived differences in color and brightness



YUV model

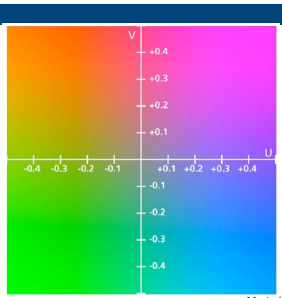
- YUV, a linear transformation from RGB
  - Y: luminance (amount of light) – grey scale
  - U: red - cyan
  - V: magenta-green

YUV (ex. PAL television system)



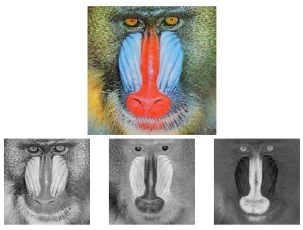
Y=0.299R + 0. 587 G + 0.114 B
U= 0.492 (B-Y)
V =0.877 (R-Y)

YUV transforms the space



http://upload.wikimedia.org/wikipedia/en/thumb/8/8a/YUV_UV_plane.jpg/600px-YUV_UV_plane.jpg

Y is more important than U and V

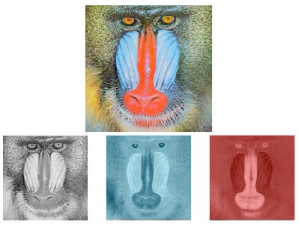


Y U V

Li & Drew ©Prentice Hall 2003

Maria Luisa Sapino (BDM 2018)

Y is more important than U and V

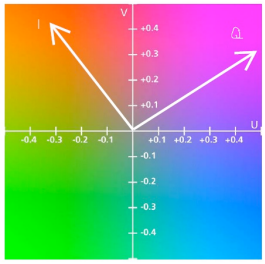


Y U V

Li & Drew ©Prentice Hall 2003

Maria Luisa Sapino (BDM 2018)

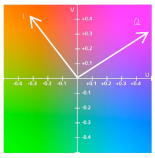
..not enough (YIQ)..



Maria Luisa Sapino (BDM 2018)

..not enough (YIQ)..

Li & Drew ©Prentice Hall 2003



Human eye is more sensitive to te Orange-Blue range than the Purple-Green range

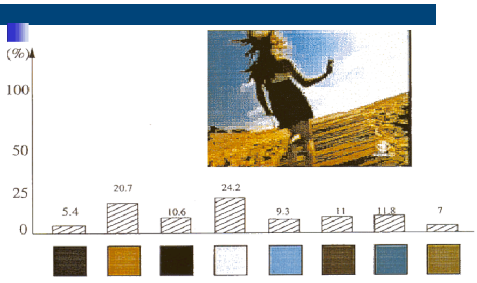
Maria Luisa Sapino (BDM 2018)

What to do???

- Lossy compression
  - Image may contain details that human eye can not recognize
  - Change the color representation (don't use RGB!)
    - human eye is not sensitive to color anyway
    - Human eye is more sensitive to contrast!
  - Use a representation which increases the contrast and compress the color element
    - Use YRB (Luminance, Red, Blue)
    - Quantization:
      - 1 byte for Luminance
      - ¼ byte for Red
      - ¼ byte for Blue

Maria Luisa Sapino (BDM 2018)

Color histograms



Maria Luisa Sapino (BDM 2018)
Courtesy of Misha Pavel, OGI

Problems with histograms

Histogram: {green:4, purple:2, red:3}

- Are these similar???

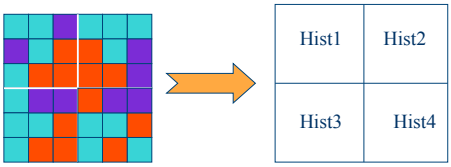
Problems with histograms

Histogram: {green:4, purple:2, red:3}

- Are these similar???

- Color associations????:
  - blue is similar to purple
  - yellow is similar to orange

Color locality??



Comparison of color histograms

- Euclidean distance

$$\sqrt{(b_1-b_2)^2 + (g_1-g_2)^2 + (p_1-p_2)^2 + (r_1-r_2)^2 + \ldots}$$

- Intersection similarity

$$\frac{\min(b1,b2) + \min(g1,g2) + \min(p1,p2) + \min(r1,r2) + \ldots}{b2 + g2 + p2 + r2 + \ldots}$$

Complete Euclidean Distance

- Let x and y be two histogram vectors, each of length n

$$d^2 = \sum_{i=1..n} \sum_{j=1..n} a_{ij} (x_i-y_i) (x_j-y_j)$$

  - a_{ij} = cross talk factor between i-th and j-th color

No cross-talk→ $a_{ij} = 1$ if $i = j$
$a_{ij} = 0$ otherwise

Quadratic distance bounding

- Use average color of an image

$$R_{avg} = (1/N) \sum_{i=1..N} R(p_i) \qquad G_{avg} = (1/N) \sum_{i=1..N} G(p_i)$$
$$B_{avg} = (1/N) \sum_{i=1..N} B(p_i)$$

$$\underline{x} = (R_{avg}, G_{avg}, B_{avg})^T$$

$$d_{avg}^2 (\underline{x},\underline{y}) = (\underline{x}-\underline{y})^T (\underline{x}-\underline{y})$$
$$= (R_{avgx} - R_{avgy})^2 + (G_{avgx} - G_{avgy})^2 + (B_{avgx} - B_{avgy})^2$$

Quadratic distance bounding

$d_{avg}^2(\underline{x},\underline{y}) <= c \cdot d_{hist}^2(\underline{x},\underline{y})$

Quadratic distance bounding

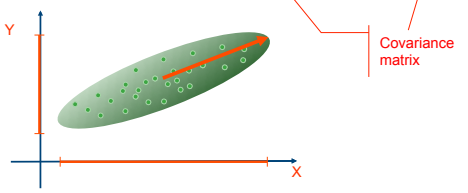
$d_{avg}^2(\underline{x},\underline{y}) <= c \cdot d_{hist}^2(\underline{x},\underline{y})$

Why is this good??

$S[x,y] = Cov(x,y) = E\left((x-\mu_x)(y-\mu_y)\right)$

Mahalonobis Distance

$$\Delta_{Mah}(\vec{a},\vec{b}) = \sqrt{(\vec{a}-\vec{b})^T S^{-1}(\vec{a}-\vec{b})}$$



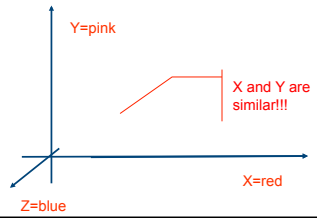
Covariance matrix

Covariance matrix

$$S[i,j] = Cov(i,j) = E\left((i-\mu_i)(j-\mu_j)\right)$$

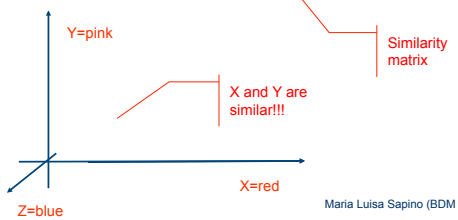
Quadratic Distance

$$\Delta_{Euc}\left(\vec{a},\vec{b}\right) = \sqrt{\left(\vec{a}-\vec{b}\right)^T\left(\vec{a}-\vec{b}\right)}$$

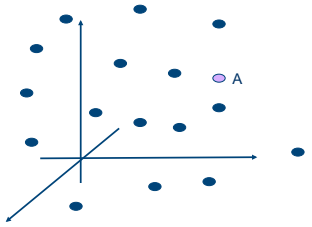


Quadratic Distance

$$\Delta_{quad}\left(\vec{a},\vec{b}\right) = \sqrt{\left(\vec{a}-\vec{b}\right)^T A\left(\vec{a}-\vec{b}\right)}$$



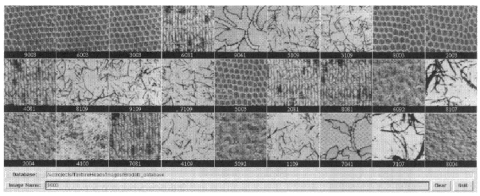
Are there other similarity measures?



Intersection similarity

$$\mathrm{sim}_{\mathrm{int}}(\vec{a},\vec{b}) = \frac{\sum_{i=1..n} \min(a_i, b_i)}{\sum_{i=1..n} \max(a_i, b_i)}$$

Texture

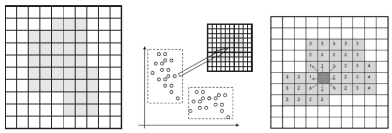


Courtesy of Misha Pavel, OGI

Texture

- Primitives:
  - Grey level
  - Shape
  - Homogeneity
  - Smoothness
  - Finess
  - Coarseness
  - Granularity
  - Regularity
  - Linearity
  - Directionality

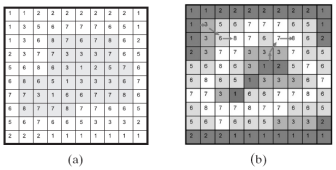
Image segmentation



(a) (b) (c)

Figure 2.17. (a) An image with a single region. (b) Clustering-based segmentation uses a clustering algorithm that identifies which pixels of the image are similar to each other first, and then finds the boundary on the image between different clusters of pixels. (c) Regiongrowing techniques start from a seed and grow the region until a region boundary with pixels with different characteristics is found (the numbers in the figure correspond to the distance from the seed).

Watershed transformation



(a) (b)

Figure 2.18. (a) Gradient values for the example in Figure 2.17 and (b) the topographical surface view (darker pixels correspond to the highest points of the surface and the lightest pixels correspond to the watershed) – the figure also shows the quickest descent (or water drainage) paths for two flood starting points.

Chaining

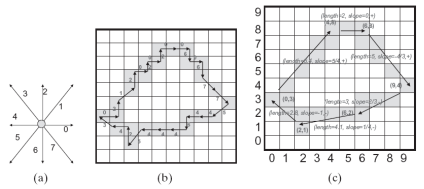


Figure 2.19. (a) The eight direction codes. (b) (If we start from the leftmost pixel) the 8-connected chain code for the given boundary is "0212020222626775446444524 3." (c) Piecewise linear approximation of the shape boundary.

Hough Transform

- a technique for isolating features of a particular shape within an image.
- computes a global description of a feature given (possibly noisy) local measurements.
- Most commonly used for the detection of regular curves such as lines, circles, ellipses
- Generalized Hough transform in applications where simple analytic description of features is not possible