CS 376A
Digital Image Processing

03 / 06 / 2023

Instructor: Michael Eckmann

Today's Topics

- Questions? Comments?
- Ideas for colorizing algorithms
- Discussion of "Transferring Color to Greyscale Images" by Welsh, Ashikhmin and Mueller
- K-means clustering start code based on design ideas from last time

Colorizing a Grayscale image

- Let's hear some thoughts from you about what you think could be ways to colorize a grayscale image
- Is there one right answer?

Discussion

- Your homework was to read Welsh paper
  - outline of steps for the technique
    - how does each step work and what's its purpose
  - which steps have we learned? which haven't we?
  - reasoning behind each decision
  - assumptions (stated and unstated)
  - limitations of the technique

Correlation of RGB channels

- RGB is not used in favor of a different color space

Correlation of RGB channels

- Let me summarize the main part of a paper cited by the Welsh et. al. paper. In the cited paper (Reinhard et. al "Color Transfer Between Images" IEEE Computer Graphics & Applications, 2001) is a description of a color space that has its channels minimally correlated (for natural scenes).

paper discussion

- Given: a grayscale image to be colorized and a color image (w/ some similar content) to get the colors from.
- 1. convert the grayscale and color images to l-alpha-beta color space
  - a. do luminance remapping on the grayscale (save original luminance of grayscale and make an additional remapped luminance value per pixel)
- 2. select a set of sample pixels from the color image (via random jittered sampling)
- 3. for all pixels in the grayscale image find the best match of the remapped luminance value to the set of sample pixels from color image
  - Do this by distance between luminance channels and neighborhood stats
  - Copy the alpha and beta channels of the best match and the original luminance of the grayscale image to an output image for each pixel
  - Convert the output image l-alpha-beta to RGB

paper discussion

- Swatches
  - What was the purpose of the swatches idea?

paper discussion

- Luminance remapping

  (stdDevBLum / stdDevALum)*(PixelALum – meanALum) + meanBLum

  - Purpose?
    - Consider a fairly bright grayscale image and fairly dark color image --- the luminance values in the bright image will all be closest to the brightest pixels in the color image --- not good
  - Let's see example on the board of
    - B image mean 128, stddev 20 and
    - A image mean 200, stddev 5
    - and see what a pixel with value of 210 in A image maps to in B

paper discussion

- Why only consider a sample of the color image pixels?
  - Positives?
  - Negatives?

Sampling

- Sampling pixels
  - Uniform Grid sampling
  - Random sampling
  - Random Jittered sampling (this is the choice made in the paper)

paper discussion

- Neighborhood statistics
  - What is the standard deviation of the luminances in a neighborhood telling us about that pixel's neighborhood?

paper discussion

- Neighborhood statistics
  - What is the standard deviation of the luminances in a neighborhood telling us about that pixel's neighborhood?
  - Consider neighborhoods of fairly constant luminance
    - What's the std dev of luminance in those neighborhoods?
  - What about a highly textured area (such as a pixel on leaf of a tree)?

paper discussion

- Neighborhood statistics
  - Precompute the standard deviation of the luminances of a neighborhood around the samples
  - For each pixel in gray scale image, compute the standard deviation of the remapped luminances of the same neighborhood size
  - To compute stddev of a set of numbers we have to first compute the mean
  - Then sum up the difference between each number and the mean squared
  - Stddev = sqrt of that sum

paper discussion

- Distance used is L2 (Euclidean distance) between the luminance and stddev pairs
- Best match is smallest distance
- Equal weight is given to luminance difference and stddev difference

Back to K-means design

- Let me write some code based on the design ideas from last time which were:
- n is the dimensions of the feature vector (e.g. 3 for R,G,B, 2 for Intensity,Texture)
- K is # of classes
- class FeatureInfo

  double [] features length n

  int classNum
- array of size K means of type FeatureInfo
- 2d array of FeatureInfo same size as the image to hold the features of each pixel and the class number of each pixel

Correlation of RGB channels

- Let's code the transformation between RGB and l-alpha-beta and back.
  - some notes:
    - let's create another class that stores an image in l-alpha-beta space
    - let's get RGB mapped from 0-255 to 0-1 (as floating point) and work on those
    - convert to LMS
    - take log base 10 of LMS channels
    - compute l-alpha-beta space
  - how can we test our code to check if implemented correctly?