

Linear Correlation

CS 510

Lecture #5

February 4th, 2013

The logo for Colorado State University, featuring a green wavy line with yellow lines underneath, and the text "Colorado State University" in a gold serif font.

Colorado State University

What to do if you miss a lecture

- Catch up – quickly
 - Read through the lecture notes on-line
 - If you don't understand something:
 - Read the suggested readings (e.g. Szeliski)
 - Come talk to me
 - Check the assignments page
- This course moves quickly
- Don't fall behind!

Where are we?

Pearson's Correlation

$$\frac{\sum_x \sum_y ((A[x,y] - \bar{A})(B[x,y] - \bar{B}))}{\sqrt{\sum_x \sum_y (A[x,y] - \bar{A})^2} \sqrt{\sum_x \sum_y (B[x,y] - \bar{B})^2}}$$

Used to match two *registered* (i.e same size, overlaid) images

Computing Correlation

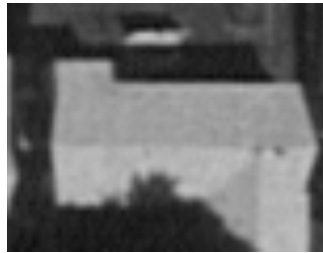
- To simplify the computation:
 - subtract average A from all $A(x,y)$
 - compute the length of $A = \sqrt{\sum_{i,j} A(i,j)^2}$
 - Multiply A by the inverse of its length
 - Repeat this process for B
- Correlation reduces to

$$A \cdot B$$

Limitations

- To match images this way, they must be
 - The same width & height
 - In correspondence : coordinates match
- More importantly, objects in the scene must
 - Be in the same location
 - Be at the same scale
 - Be at the same orientation
 - Be seen from the same viewpoint

New Goal: Find a small image within a larger one

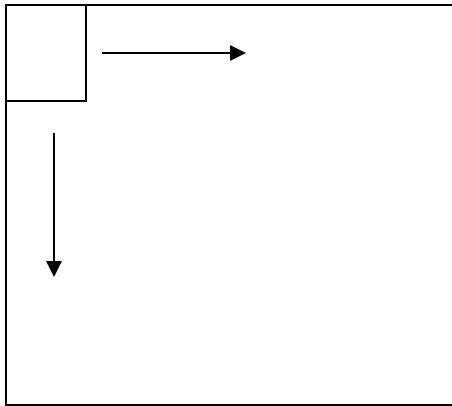


- The image above is a small piece of the image to the right. But from where?



Brute-Force Translation Invariance

To find a small image in a large one, “slide” the small one across the large, computing Pearson’s correlation at every possible position.



Cross-Correlation

- The process of “slide & correlate” is called cross-correlation
- Complexity is $O(nm)$
 - $N = \#$ of pixels in image ($w \times h$)
 - $M = \#$ of pixels in the template ($w \times h$)
- Highly parallel (every position can be computed independently)
- Still sensitive to
 - Rotation
 - in-plane
 - out-of-plane
 - Scale
 - Perspective

Computing Cross-Correlation

- In cross-correlation, a template is correlated repeatedly to image windows
 - zero-mean & unit length the template (once)
 - zero-mean & unit length the image window
 - compute the sliding dot product

In other words, *convolve* the image with the mask

Linear correlation

- Normalizing every image window is expensive
 - What if we skip this step?
 - Just slide the template over the image, computing dot products with each window?

This is common, and is often what is meant by “correlation” (e.g. in Seliski). We will call it *linear correlation*

Details are on the assignments page... you have one week

Programming Assignment #1



Source (Target) Images
Templates



Left Eye



Right Eye



Left Ear



Nose

Colorado State University

Application: Tracking

- Cut out a picture of a target from the first frame of a video
 - Use it as a template /mask
- Correlate the target in the following frames
 - Find the location with the highest correlation

Click here [for](#) video

- Improvement: update target with each new frame

Application: Mosaicing

- Take several, overlapping images from a translating camera
 - Camera cannot move along optical axis
- Correlate images to each other
 - Find windows that match
 - Compute geometric transformation that aligns them
 - Stitch them into a single, larger image

Mosaicing (II)



Image 1



Image 2



Image 3



Colorado State University

Where are we?

- We now have our first image matching technique
- On the good side...
 - It's insensitive to camera gain and offset
 - It's insensitive to translation
- But, on the down side..
 - It's slow
 - It's sensitive to every other geometric transformation
 - It's sensitive to changes in appearance

Motivation for Fourier Analysis

- We will now briefly detour into Fourier Analysis
- Why?
 - It will make template matching faster
 - It will help us build better templates
 - It will improve our intuitive understanding of image space
- Other benefits we will not address
 - Understand image aliasing, Nyquist rates, etc.
 - Understand JPG compression