Matching Image Patches: Notes

Michael Guerzhoy

January 8, 2015

1 Introduction

We are trying to solve the problem of *matching patches*. For example, suppose we have a grayscale image of an eye (p_1), and we would like to try to match it to every possible patch on Einstein's photo (refer to slides).

2 Linear Algebra Background

- The (Euclidean) **norm** of a vector v: $|v| = \sqrt{(\sum_i v[i]^2)}$. The Euclidean norm of a vector v is the same as the *length* of the vector.
- The dot product of the vectors v_1 and v_2 is $v_1 \cdot v_2 = \sum_i v_1[i]v_2[i]$.
- The dot product can also be written as $v_1 \cdot v_2 = |v_1||v_2|\cos\theta_{v_1,v_2}$. The angle θ_{v_1,v_2} is the angle between v_1 and v_2. That means we can compute the angle between v_1 and v_2 using $\frac{v_1 \cdot v_2}{|v_1||v_2|}$.
- We can compute the *difference* between v_1 and v_2 by computing $v_1 - v_2$. The difference can be written by coordinate using $(v_1[0] - v_2[0], v_1[1] - v_2[1], ...)$. The norm of the difference is just $\sqrt{\sum_i (v_1[i] - v_2[i])^2}$.
- Note: the length of the difference between v_1 and v_2 is the same as the distances between the endpoints of vectors v_1 and v_2.

3 Matching Grayscale Patches

Grayscale images are just rectangular tables of numbers, with 0's representing completely black pixels, 255's representing completely white pixels, and number between 0 and 255 representing various gray pixels. We can rearrange the numbers so that any patches is a *vector* – a sequence of numbers.

So the problem of matching patch p_1 to patch p_2 (i.e., the problem of determining how similar they are) is just (in this view of things) the problem of determining the similarity between vector p_1 and vector p_2.

4 Patch Distance Metrics

- The Sum of Squared Differences (SSD): this is just the square of the Euclidean distance between points p_1 and p_2 ($|p_1 - p_2|$, or $\sum_i (p_1[i] - p_2[i])^2$). If SSD is 0, the patches are identical. If SSD is large, the patches are different. Problem: Consider $.5p_1$ – it represents a darker version of p_1, but it's still the same image, in a way! However, the SSD between p_1 and $.5p_1$ can be quite large

- Dot product: we can take the length of the *projection* of p_1 on p_2. That's $p_1 \cdot p_2 = \sum_i p_1[i]p_2[i]$. The dot product is large when p_1 and p_2 are similar. Problem: if $p_2 = [255, 255, 255, 255, ...]$, the dot product will be large regardless of what p_1 is, pretty much

- The cosine of the angle between p_1 and p_2. Can be computed using the formula $\cos\theta_{p_1,p_2} = \frac{p_1 \cdot p_2}{|p_1||p_2|}$. This is great, since it will be the largest for both $p_1 = p_2$ and $p_1 = ap_2$ for any constant $a > 0$: so the overall brightness won't matter anymore. The images are similar if the cosine of the angle is close to 1 (so the angle is close to 0), and different if the cosine of the angle is close 0 (so the angle is close to $\pi/2$.

- We still have the problem of the dot product with bright and dark images being different. To solve that, we make it so that the average intensity of the patches we are comparing is 0. To achieve, that, we subtract the mean intensity from the image:

  $$NCC(p_1, p_2) = \frac{(p_1 - \mu_{p_1}) \cdot (p_2 - \mu_{p_2})}{|p_1 - \mu_{p_1}||p_2 - \mu_{p_2}|}$$

  .

  This is called the Zero-Mean Normalized Cross-Correlation (also known as Pearson's r, or plain *correlation* in statistics).