

CS 331

Computer Vision

09 / 29 / 2026

Instructor: Michael Eckmann

Today's Topics

- Questions / Comments?
- Histograms
- Distance measures to determine closeness of vectors
- In-class exercises on histograms, normalized histograms and distance
- Computing bin numbers fast under certain conditions

Determining similarity

- We can represent objects with those region properties (e.g. a vector of values of those properties) or as a histogram or in countless other ways or combinations of them.
- What we'll have is a vector of values as the representation of an image and we will need a way to compare the vectors to determine how close they are to each other (or how far apart they are).

Distance measures

- There are many ways to compute distance between two multidimensional vectors
- the L1 distance (aka Manhattan or Taxicab) is the sum of the absolute values of the differences of each dimension
- the L2 distance (aka Euclidean) is the square root of the sum of squares of the differences of each dimension
- These are easy to see in 2 dimensions.
- Say we have $p = (p_1, p_2)$ and $q = (q_1, q_2)$ then the L1 distance between them is: $\text{abs}(p_1 - q_1) + \text{abs}(p_2 - q_2)$
- The L2 distance between them is: $\text{sqrt}((p_1 - q_1)^2 + (p_2 - q_2)^2)$

Distance measures

- In general for n dimensions,
- if $p=(p_1, p_2, \dots, p_n)$ and $q = (q_1, q_2, \dots, q_n)$
- the L1 distance between them is:

$$\text{abs}(p_1-q_1) + \text{abs}(p_2-q_2) + \dots + \text{abs}(p_n-q_n)$$

- The L2 distance between them is:

$$\text{sqrt}((p_1-q_1)^2 + (p_2-q_2)^2 + \dots + (p_n-q_n)^2)$$

- These distance measures (as well as many others) could be used to compute the distance between any n-dimensional sets of values, not only histograms. For example the region measures (area, perimeter, etc.) can each be dimensions of a vector and then one can take the L1 or L2 distance between vectors to determine how similar or dissimilar the regions are.

Distance measures

- Let's do an example where the vectors have 5 values (5 dimensions)
- Example: $p = (5,3,2,6,9)$ and $q=(4,4,4,7,6)$ and $r = (10,6,4,5,0)$
- Let's see whether p is closer to q or r using L1
- Example calculation:
- L1 distance between p and $q = |5-4| + |3-4| + |2-4| + |6-7| + |9-6| = 1+1+2+1+3 = 8$
- L1 distance between p and $r = |5-10| + |3-6| + |2-4| + |6-5| + |9-0| = 5+3+2+1+9 = 20$
- So p is closer to q than r when considering L1 distance
- In other words we determined that p is more similar to q than it is to r
- Let's do the same with L2 distance and see how they compare

histograms

- The p , q and r of last slide can each be histograms of multidimensions and the L1 or L2 distances can be used as a way to compute the distance between them.
- The idea being that similar images will have similar histograms and the distance between their histograms will be small, whereas dissimilar images will have very dissimilar histograms and the distance between their histograms will be large.
- But remember a histogram just counts up the number of pixels of certain ranges of grey values and disregards the position of the pixel values, so the histogram representation ignores the image spatially.
- If the images are of different size (that is different number of pixels) then we should normalize the histograms before computing distance.
- Normalized histograms are those that have each bin value from the original histogram divided by the total number of pixels in the image. Therefore the sum of all the values of a normalized histogram will equal 1. Each bin stores the fraction of the total pixels that have that bin grey value.

In-class exercises

- Exercises on computing histograms, normalizing histograms and computing L1 distance between histograms

histograms

- Let's look back at these bins and then efficient operations to compute bin# based on R,G,B value.
- Bin0 those with R in 0-127, G in 0-63, B in 0-127
- Bin1 those with R in 0-127, G in 0-63, B in 128-255
- Bin2 those with R in 0-127, G in 64-127, B in 0-127
- Bin3 those with R in 0-127, G in 64-127, B in 128-255
- Bin4 those with R in 0-127, G in 128-191, B in 0-127
- Bin5 those with R in 0-127, G in 128-191, B in 128-255
- Bin6 those with R in 0-127, G in 192-255, B in 0-127
- Bin7 those with R in 0-127, G in 192-255, B in 128-255
- Bin8 those with R in 128-255, G in 0-63, B in 0-127
- Bin9 those with R in 128-255, G in 0-63, B in 128-255
- Bin10 those with R in 128-255, G in 64-127, B in 0-127
- Bin11 those with R in 128-255, G in 64-127, B in 128-255
- Bin12 those with R in 128-255, G in 128-191, B in 0-127
- Bin13 those with R in 128-255, G in 128-191, B in 128-255
- Bin14 those with R in 128-255, G in 192-255, B in 0-127
- Bin15 those with R in 128-255, G in 192-255, B in 128-255

Histograms

- How to determine bin # for an RGB (assumes each channel is divided into a power-of-2 range)
- Since R and B are each divided into 2 groups of values (which means divide by $256/2 = 128$)
- G is divided into 4 groups of values (divide by $256/4 = 64$)
- $R_{\text{shifted}} = R \gg 7$ # same as $R // 128$ note: 2 to the 7th is 128
- $G_{\text{shifted}} = G \gg 6$ # same as $G // 64$ note: 2 to the 6th is 64
- $B_{\text{shifted}} = B \gg 7$
- Then to compute bin number:
- $R_{\text{shifted}} \ll 3 + G_{\text{shifted}} \ll 1 + B_{\text{shifted}}$
- Same as $R_{\text{shifted}} * 8$ (2 to the 3rd) + $G_{\text{shifted}} * 2 + B_{\text{shifted}}$