Computer Vision | CS 670, Fall 2019

HW07: Matching and Alignment

Remember that only PDF submissions are accepted. We encourage using LaTeX to produce your write-ups. See `hw00.tex` for an example of how to do so. You can make a `.pdf` out of the `.tex` by running "`pdflatex hw00.tex`".

1. Consider a 1D Gaussian $g = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$. Show that the second derivative of this Gaussian w.r.t. x is proportional to its derivative w.r.t. scale parameter σ.

$$g = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-x^2/2\sigma^2}$$

$$* \quad \frac{dg}{dx} = -\frac{x}{\sigma^3\sqrt{2\pi}} e^{-x^2/2\sigma^2} = \left(\frac{x}{\sigma^2} g\right)$$

$$\xrightarrow{D} \Rightarrow \frac{d^2 g}{dx^2} = -\left[\frac{\sigma^2 - x^2}{\sigma^5\sqrt{2\pi}} e^{-x^2/2\sigma^2}\right]$$

$$* \quad \frac{dg}{d\sigma} = \frac{d}{d\sigma}\left[\frac{1}{\sqrt{2\pi}\,\sigma} e^{-x^2/2\sigma^2}\right]$$

$$= \frac{1}{\sqrt{2\pi}}\left[\frac{d}{d\sigma}\sigma^{-1} e^{-x^2/2\sigma^2}\right] = \frac{1}{\sqrt{2\pi}}\left[-1\sigma^{-2} e^{-x^2/2\sigma^2} + \frac{1}{\sigma} e^{-x^2/2\sigma^2}\left(+\frac{x^2 \cancel{2}}{\cancel{2}}\sigma^{-3}\right)\right]$$

$$= \frac{1}{\sqrt{2\pi}}\left[-\frac{1}{\sigma^2} e^{-x^2/2\sigma^2} + \frac{x^2}{\sigma^4} e^{-x^2/2\sigma^2}\right]$$

$$= -\left[\frac{\sigma^2 - x^2}{\sqrt{2\pi}\,\sigma^4} e^{-x^2/2\sigma^2}\right]$$

$$= \left[\frac{1}{\sigma}\frac{d^2 g}{dx^2}\right]$$

Hence, proved. $\boxed{\frac{dg}{d\sigma} \propto \frac{d^2 g}{dx^2}}$

2. Why is normalized correlation preferred to SSD when comparing vectors of raw pixel intensity values?

This is because normalized correlation have mean = 0 and length =1. Thus, during correlation the brighter pixels would not have any advantage over the darker ones. In other words, when comparing 2 different pixel vectors of 2 pictures with differing brightness, we should use normalized correlation rather than SSD, to make sure that intensity does not play any role in it even though the pixels may be the same (varying only at brightness). This is because in normalized correlation we divide the vector by its magnitude and hence the we get a vector of unit length irrespective of the intensity values.

3. Consider the line fitting example using RANSAC. Suppose the fraction of points that are outliers is p, and at each iteration of RANSAC algorithm you sample two points to fit a line.

(a) What is the probability that RANSAC will terminate without finding the correct solution after T iterations?

Number of outliers = p
Failure probablity in 1 iteration =
Pr(finding one inlier and one outlier) + Pr(finding two outliers) = $p * (1 - p) + p * (1 - p) + p*p$

Failure probability in 1 iteration = $1 - (1 - p)(1 - p)$

So the probabilty that RANSAC will terminate without finding the correct solution after T iterations is that if in every iteration it fails with the probability $1 - (1 - p)(1 - p)$

Then the probability for RANSAC to fail after T iteration is

$(1 - (1 - p)^2)^T$

(b) Suppose $p = 0.5$, i.e., 50% of the points are outliers. How many iterations (T) are needed to find the correct solution with probability > 0.99?

The above condition gives us a equation of the form mentioned below:

$=> (1 - (1 - p)^2)^T > 0.99$ *where p is the the fraction of outliers*
$=> (1 - (1 - 0.5)^2)^T > 0.99$
$=> T * \log(1 - (1 - 0.5)^2) > \log 0.01$
$=> T > \log(0.01)/\log(0.75)$
$T \geq 16.01$

The number of iterations required is 17.

(c) Now suppose, instead of fitting a line you are interested in fitting a transformation that requires k points. For example, in the image alignment example discussed in class you needed 3 matches to estimate an affine transformation between two point sets. How does the answer to the question 1(a) change?

If we need more points then the probability of failure will go up and hence the success probability will go down. For example if we need 3 points to estimate affine transformation: Failure probability will be

$= ppp + 3pp(1 - p) + 3p(1 - p)(1 - p) = p^3 + p^2 - p^3 + p(1 - 2p + p^2)\ p^3 - p^2 + p = p(p^2 - p + 1)$

Failure probability for 2 points is $1 - (1 - p)^2$.
Comparing various values of p we get:

p	Failure Probability for 3 points	Failure Probability for 3 points
0.25	0.578	0.4375
0.5	0.875	0.75
0.75	0.984375	0.9375

Figure 1. Failure probabilities for 1 iteration for *p*

Thus, the probability for failure in RANSAC where 3 points have to be matched is more than that where two points are needed to be calculated. This implies that when *k* increases, the failure of probability increases where *k* is the number of points required for fitting a transformation.