Prof. Raghuveer Parthasarathy
University of Oregon; Fall 2024

Physics 410/510 Image Analysis: Homework 5

Due date: Wednesday October 30 by 11:59 pm, submitted through Canvas. You'll see in "Assignments" a place to submit a **PDF**.

Reading: Please read / watch one of these before Thursday (Nov. 1) – you might find them useful beforehand as well, though they might spoil the surprise of Question #3!

- Sections 1-3 of Parthasarathy and Small, "Superresolution Localization Methods," *Annu. Rev. Phys. Chem.* **65:**107-125 (2013). (*http://www.annualreviews.org/doi/abs/10.1146/annurev-physchem-040513-103735*)
- R. J. Ober, S. Ram, E. S. Ward, "Localization Accuracy in Single-Molecule Microscopy," *Biophys. J.* **86**, 1185–1200 (2004). *https://www.cell.com/biophysj/fulltext/S0006-3495(04)74193-4*
- Regarding super-resolution and the challenge of finding objects, you might like this video, *http://www.ibiology.org/ibioseminars/cell-biology/xiaowei-zhuang-part-1.html*, which gets at the methods described in the first section or two of the Parthasarathy and Small paper.

Suggestion: Start with #2 and #3, before Tuesday!

1 The Signal-to-Noise Ratio in Images. (3 pts.) In Homework 4, we simulated images corresponding to some number of photons, N_{photon}. We often want to use such images to assess how accurate our image analysis is. However, in real life we usually don't know how many photons we're receiving; the camera gives us some number that's *proportional to* the voltage at each pixel, and the voltage is *proportional to* the number of photons. In other words, each pixel value $I = A\, N_{photon}$, and we don't know what A is. Does this matter? You can answer this in ***either*** of the following ways:

(i) (Math) Consider a random variable x. Its average value is $\langle x \rangle$, and the standard deviation is $\sigma = \sqrt{\langle (x - \langle x \rangle)^2 \rangle}$. (That's just the usual definition of the standard deviation.) Noting that the signal-to-noise ratio is SNR = $\langle x \rangle$ / σ, show that multiplying all the x values by a constant A doesn't change the SNR.

(ii) (Code) Make 500 arrays each of 1000 of random variables ("x"), with numbers drawn from distribution of your choice (Gaussian, Poisson, ...[1]). For 500 values of some constant "A", logarithmically spaced between 10^{-6} and 10^{6} (Note[2]), calculate the mean of the x values divided by the standard deviation of the x values for each of the 500 arrays; let's call this "r". Then calculate the mean divided by standard deviation for each of the 500 "A times x" arrays; call

[1] Numpy's list of distributions is here: https://numpy.org/doc/stable/reference/random/generator.html, under "Distributions"

[2] `numpy.logspace`, or MATLAB's `logspace`, are useful, e.g. np.logspace(3, 6, num=40) for 40 log-spaced numbers from 1000 to 1,000,000

this r_A. Show that on average, r_A/r is 1.0 ± something very small (set by numerical precision). Recommended: Do this without loops. (Python's `np.tile` or MATLAB's `repmat` may be useful.) [**510 students:** do this for two different distributions.] Show your code and the output.

Because of this, simulating an image corresponding to some number of photons that give an SNR that's the same as the measured SNR (see the last homework assignment) is valid, even though we're usually not measuring the number of photons directly. (Assuming, of course, that the relevant noise is photon noise.)

2 Centroid warmup (2 pts.). The centroid, or **center of mass,** of an object is its weighted average position, where the "weight" each point gets is the point's mass (or density). For an image, the weight is the intensity. If we have a 1D array of pixels with intensities ($\boldsymbol{I_i}$) at pixel positions ($\boldsymbol{x_i}$) we calculate the centroid as:

$$x_c = \frac{\sum_i x_i I_i}{\sum_i I_i}$$

For example, if $\boldsymbol{I}$ =

15	6	13	2	1	3

at the pixel positions ($\boldsymbol{x}$):

0	1	2	3	4	5

the centroid is

$$[(0 \times 15) + (1 \times 6) + (2 \times 13) + (3 \times 2) + (4 \times 1) + (5 \times 3)] \;/\; [15 + 6 + 13 + 2 + 1 + 3]$$

which is $\boldsymbol{x}_c = 57/40 = 1.42$. The center of mass is at position 1.42, which makes sense given the numbers – there's more weight left of center.

Here are two short exercises to practice calculating centroids:

(i) Consider intensities $I_i = 0.2\, i^{0.5}$, where the position index i goes from 0 to 999. (In other words, the intensity increases as the square root of position. Where is the centroid?

(ii) Consider intensities $I_i = 30 / (i + 20) + 0.05\, w$, where w is a Poisson-distributed random number of mean 10 and i again goes from 0 to 999. Where is the centroid?

INTRODUCTION TO #3. We now understand images of point sources, and we know that if we can "localize" the object in the image – i.e. determine its location, hopefully with much greater precision than λ/2 – we can do amazing things. How do we actually perform the localization? We come up with an algorithm! Just like Alice's and Bob's algorithms in class, some algorithms may be better than others, in terms of accuracy, computational speed, or other criteria. We can (and should!) use simulated images to test our algorithms. There's a "true" point source location **r** , (which in real life we'll never know, but in our simulated images we do), and there's our estimator $\hat{r}$. We'll consider a few different estimators that are practically useful and that illustrate general principles. For this assignment we'll just introduce one, a simple and intuitive estimator: the **centroid.**

Please see the "additional notes" for suggestions!

3 Centroid localization. (18 pts.) Write a function to calculate the center-of-mass, or centroid, of a 2D image. Note that the **x** position of the centroid for intensities I_i at positions x_i is

$$x_c = \frac{\sum_i x_i I_i}{\sum_i I_i}$$

and we can similarly calculate the **y** position of the centroid.

(a) (3 pts.) Apply your centroid estimator to at least $M = 100$ simulated images of a point source using your simulation code from HW4 [3], with 7 × 7 px images, scale 100 nm/px (i.e. 0.1 μm/px), λ = 510 nm, NA = 0.9, $N_{photon} = 500$, **and** a Poisson distributed background with mean = 10. For your "high resolution images," i.e. before pixelation, use a very fine grid size, 2.5 nm or smaller. (This will be important in later problems.) Make the true (simulated) center position $(x_0, y_0) = (0,0)$, i.e. the center of the image. For each image, calculate the centroid position, (x_c, y_c). You can leave x_c and y_c in pixels, or convert to real units. For the set of M images, calculate RMS error of the centroid localization. (See the Note below.) Submit (i) a histogram of all the output x_c values, and (ii) the RMS error you calculated.

(b) (5 pts.) Now consider a range of N_{photon} from 40 to 40,000, i.e. SNR from about 6 to about 200; make a logarithmically spaced set of at least 10 N_{photon} values in this range. For each N_{photon} value, make $M = 100$ images. For each image, calculate the centroid; for each set of M images, calculate RMS error of the centroid localization. Submit a plot of RMS error vs. N_{photon}, with logarithmic axes, and comment on its shape. Does it have the shape you expected? See the "Additional Notes" document for comments.

(c) (10 pts. For 410, 5 pts. For 510) Let's consider the error in x_c, which is simply $\Delta x = x_c - x_0$. (Note that this can be positive or negative.) For $N_{photons}$ = **1000,** simulate at least 100 images each with $(x_0, y_0) = (0.0, 0.0)$ px, $(x_0, y_0) = (0.3, 0.0)$ px, and $(x_0, y_0) = (-0.3, 0.0)$ px. (In physical units, 0.3 px = 0.03 microns = 30 nm, given the pixel scale.) For each set, use the centroid estimator and make a histogram of the Δx values you find. Use the same axis range for each histogram. Comment on the results. Is the centroid an unbiased estimator? (As we'll discuss in class, "unbiased" means that the magnitude of the error is independent of the true value.) Also do this for $N_{photons}$ = **100,000**; does more signal help?

(d) [510 students] (5 pts.) For $N_{photons}$ = **1000,** plot a graph of the mean Δx as a function of position p, for objects whose true position is (p, p), for p going from -0.5 to 0.5 px in steps of 0.1. Describe what you find: how does the error depend on p? If there is a bias, which way is it? (Towards the image center, image corner, somewhere else...) Also do this for a Poisson distributed background with mean = 100.

4 Reading (6 pts.) Start reading for the project! Look into **two** of the suggested project topics, even if you're already sure what topic you'll choose. Read the introduction of a paper, or find a good video, or find some substantive online description. Spend at least an hour on each topic. For each topic, write a few sentences on what resource(s) you used, what the topic is, something interesting you learned, and something that is confusing or unanswered by what you read. (Since this is new to you, probably much of it is confusing!) Do this individually, not in groups.

[3] See the "additional notes" for tips on testing your Homework 4 image generation function!