Assignment 3: The Atomic Nature of Matter

Science One CS 2015-2016
Instructor: Mike Gelbart
Last Updated Sunday, Jan 24, 2016 around 7:45pm
Part One due Saturday, Jan 30, 2016 at 5:00pm
Part Two due Saturday, Feb 6, 2016 at 5:00pm

Introduction

Goal. Re-affirm the atomic nature of matter by tracking the motion of particles undergoing Brownian motion, fitting this data to Einstein's model, and estimating Avogadro's number.

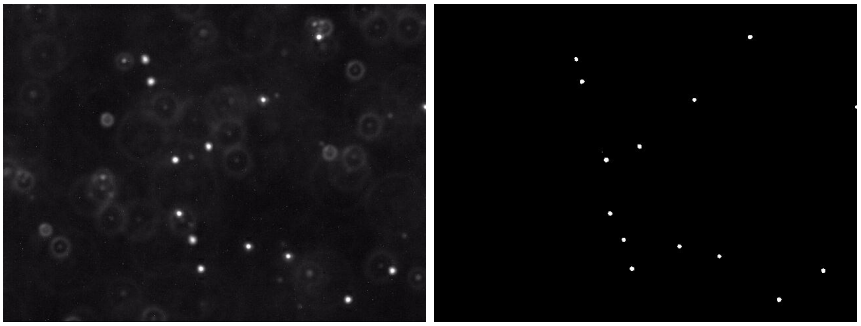
Historical perspective. The atom played a central role in 20th century physics and chemistry, but prior to 1908 the reality of atoms and molecules was not universally accepted. In 1827, the botanist Robert Brown observed the random erratic motion of wildflower pollen grains immersed in water using a microscope. This motion would later become known as Brownian motion. Einstein hypothesized that this Brownian motion was the result of millions of tiny water molecules colliding with the larger pollen grain particles.

In one of his "miraculous year" (1905) papers, Einstein formulated a quantitative theory of Brownian motion in an attempt to justify the "existence of atoms of definite finite size." His theory provided experimentalists with a method to count molecules with an ordinary microscope by observing their collective effect on a larger immersed particle. In 1908 Jean Baptiste Perrin used the recently invented ultramicroscope to experimentally validate Einstein's kinetic theory of Brownian motion, thereby providing the first direct evidence supporting the atomic nature of matter. His experiment also provided one of the earliest estimates of Avogadro's number. For this work, Perrin won the 1926 Nobel Prize in physics.

The problem. In this assignment, you will redo a version of Perrin's experiment. Your job is greatly simplified because with modern video and computer technology (in conjunction with your programming skills), it is possible to accurately measure and track the motion of an immersed particle undergoing Brownian motion. We supply video microscopy data of polystyrene spheres ("beads") suspended in water, undergoing Brownian motion. Your task is to write a program to analyze this data, determine how much each bead moves between observations, fit this data to Einstein's model, and estimate Avogadro's number.

The data. We provide a data set obtained by William Ryu using fluorescent imaging. The data contains a sequence of two hundred 640-by-480 color JPEG images, frame00000.jpg through frame00199.jpg. We also provide a movie of several beads undergoing Brownian motion.

Figure 1 shows a typical raw image (left) and a cleaned up version (right) using thresholding, as described below. Each image shows a two-dimensional cross section of a microscope slide. The beads move in and out of the microscope's field of view (the x- and y-directions). Beads also move in the z-direction, so they can move in and out of the microscope's depth of focus; this results in halos, and it can also result in beads completely disappearing from the image.



(a) Raw image (b) Thresholded

Figure 1: (a) A typical raw image frame of polystyrene spheres immersed in water and (b) a cleaned up version using thresholding.

Particle identification.

The first challenge is to identify the beads amidst the noisy data. Each image is 640-by-480 pixels. Whiter pixels correspond to beads (foreground) and blacker pixels to water (background). We break the problem into three pieces: (i) read in the picture, (ii) classify the pixels as foreground or background, and (iii) find the disc-shaped clumps of foreground pixels that constitute each bead.

1. *Read in the image.* Use `skimage.io.imread` with the keyword argument `as_grey=True`. This reads in the image as a 2D numpy array with pixel values between 0.0 (black) and 1.0 (white).[1]

2. *Classify the pixels as foreground or background.* We use a simple, but effective, technique known as thresholding to separate the pixels into foreground and background components: all pixels with monochrome luminance values strictly below some threshold τ are considered background, and all others are considered foreground. Figure 1 illustrates the original frame (above left) and the same frame after thresholding (above right), using $\tau = 0.7$. This value of τ results in an effective threshold for the supplied data.

3. *Find the blobs.* A polystyrene bead is typically represented by a disc-like shape of at least some minimum number P (typically 25) of connected foreground pixels. A blob or connected component is a maximal set of connected foreground pixels, regardless of its shape or size. We will refer to any blob containing at least P pixels as a bead. The center-of-mass of a blob (or bead) is the average of the x- and y-coordinates of its constituent pixels. Use the functions `skimage.measure.label` and `skimage.measure.regionprops` to find the connected components in each image. For example, if `x` is a 2D numpy array represents a thresholded image, then the following code automatically finds the connected components and prints out their areas and locations:

[1](Optional) If you would like to look at the images (or the thresholded images later on), use `plt.imshow(img)` where `img` is the 2D array containing the image. Then you'll need a `plt.show()` or `plt.savefig(filename)` as usual.

```
regions = skimage.measure.regionprops(skimage.measure.label(x))
for region in regions:
    print 'This region has area %f and' % region.area
    print 'is located at (%f,%f)' % (region.centroid[1], region.centroid[0])
```

Note that the centroids are stored as (y, x) pairs rather than (x, y) pairs.

Task. Fill in the `get_blobs` function in the provided code skeleton. To test it out, we have provided the `test_get_blobs` function. Here is the output you should get when you run `test_get_blobs()`:

```
Found 13 beads:
(477.86,  49.39)
(214.68,  82.74)
(223.62, 116.57)
(393.50, 144.21)
(310.55, 214.68)
(260.39, 234.77)
(266.03, 315.71)
(286.58, 355.45)
(370.94, 365.42)
(431.26, 380.41)
(299.05, 399.14)
(588.59, 402.16)
(521.71, 445.84)
```

*****You must submit your work up to this point by the first deadline, which is Jan 30 at 8:00pm.*****

Particle tracking.

The next step is to determine how far a bead moved from one time step t to the next $t + \Delta t$. For our data, $\Delta t = 0.5$ seconds per frame. We assume the data is such that each bead moves a relatively small amount, and that two beads do not collide. (However, we must account for the possibility that the bead disappears from the frame, either by departing the microscope's field of view in the x- or y- direction, or moving out of the microscope's depth of focus in the z-direction.) Thus, for each bead at time $t + \Delta t$, we calculate the closest bead at time t (in Euclidean distance) and identify these two as the same beads. However, if the distance is too large (greater than δ pixels) we assume that one of the beads has either just begun or ended its journey. We record the displacement that each bead travels in the Δt units of time. Note that it is not necessary to explicitly track a bead through a sequence of frames–you only need to worry about identifying the same bead in two consecutive frames.

Task. In the provided code skeleton, fill in the `closest` function and the first "`TODO`" in the `main` function (we are assuming you have already written `get_blobs`). To test your particle tracking code, we provide the following information: of the computed displacements, the minimum is 0.10, the maximum is 20.61, and the mean is 4.66.

Data analysis.

Einstein's theory of Brownian motion connects microscopic properties (e.g., radius, diffusivity) of the beads to macroscopic properties (e.g., temperature, viscosity) of the fluid in which the beads are immersed. This amazing theory enables us to estimate Avogadro's number with an ordinary microscope by observing the collective effect of millions of water molecules on the beads.

Estimating the self-diffusion constant. The self-diffusion constant D characterizes the stochastic movement of a molecule (bead) through a homogeneous medium (the water molecules) as a result of random thermal energy. The Einstein-Smoluchowski equation states that the random displacement of a bead in one dimension has a Gaussian distribution with mean zero and variance $\sigma^2 = 2D\Delta t$, where Δt is the time interval between position measurements. That is, a molecule's mean displacement is zero and its mean square displacement is proportional to the elapsed time between measurements, with the constant of proportionality $2D$. We estimate σ^2 by computing the variance of all observed bead displacements in the x and y directions. Let $(\Delta x_1, \Delta y_1), \ldots, (\Delta x_n, \Delta y_n)$ be the n bead displacements, and let $r_1, \ldots, r_n$ denote the radial displacements. Then

$$\begin{aligned}\hat{\sigma}^2 &= \frac{(\Delta x_1^2 + \ldots + \Delta x_n^2) + (\Delta y_1^2 + \ldots + \Delta y_n^2)}{2n} \\ &= \frac{r_1^2 + \ldots + r_n^2}{2n}\end{aligned}$$

For our data, $\Delta t = 0.5$ so our estimate for σ^2 is an estimate for D as well.

Note that the radial displacements in the formula above are measured in meters. The radial displacements output by your BeadTracker program are measured in pixels. To convert from pixels to meters, multiply by 0.175×10^{-6} (meters per pixel).

The value of n is the count of the total number of displacements read.

Estimating the Boltzmann constant. The Stokes-Einstein relation asserts that the self-diffusion constant of a spherical particle immersed in a fluid is given by

$$D = \frac{kT}{6\pi\eta\rho}$$

where, for our data,

- T = absolute temperature = 297 degrees Kelvin (room temperature)
- η = viscosity of water = $9.135 \times 10^{-4}\mathrm{N} \cdot \mathrm{s/m}^2$ (at room temperature)
- ρ = radius of bead = 0.5×10^{-6} meters

and k is the Boltzmann constant. All parameters are given in SI units. The Boltzmann constant is a fundamental physical constant that relates the average kinetic energy of a molecule to its temperature. We estimate k by measuring all of the parameters in the Stokes-Einstein equation, and solving for k.

Estimating Avogadro's number. Avogadro's number N_A is defined to be the number of particles in a mole. By definition, $k = \frac{R}{N_A}$, where the universal gas constant R is approximately 8.31457 J K^{-1} mol^{-1}. Use R/k as an estimate of Avogadro's number.

Task. Fill in the second "`TODO`" in the `main` function in the code skeleton. To test your whole program, we provide the estimates your code should produce:

```
>> python AtomicNature.py 25 0.7 25.0 data

Estimate of Boltzmann's constant: 1.236e-23 J/K
Estimate of Avogadro's number:    6.727e+23
```

Provided files. You can download the data and a partially completed program from the course website.

Assignment adapted with permission from Princeton COS 126, "The Atomic Nature of Matter" at `http://www.cs.princeton.edu/courses/archive/fall15/cos126/assignments/atomic.html`. Original assignment was created by David Botstein, Tamara Broderick, Ed Davisson, Daniel Marlow, William Ryu, and Kevin Wayne.