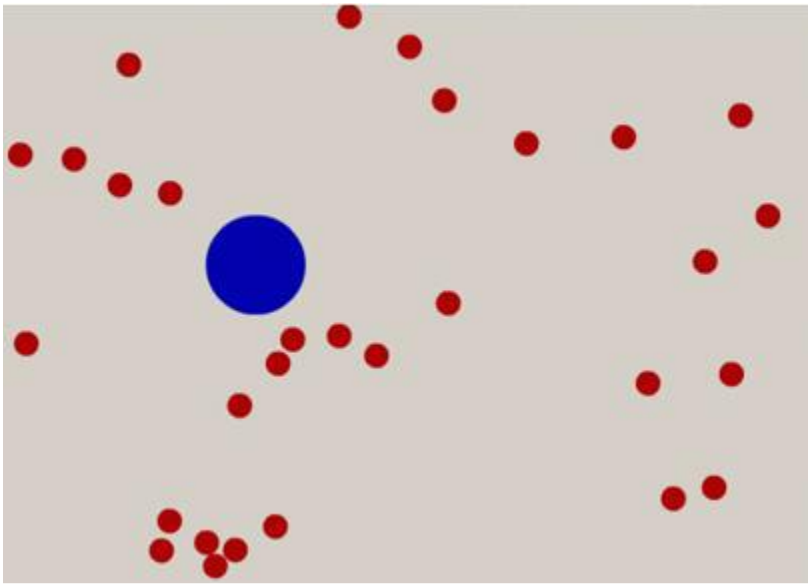
ماهیت اتمی ماده

پروژه درس برنامه‌نویسی پیشرفته با پایتون

تأکید دوباره بر ماهیت اتمی مواد با ردیابی حرکات ذرات تحت حرکت براونی، مطابقت دادن داده‌ها با مدل انیشتین و تخمین عدد آووگادرو.

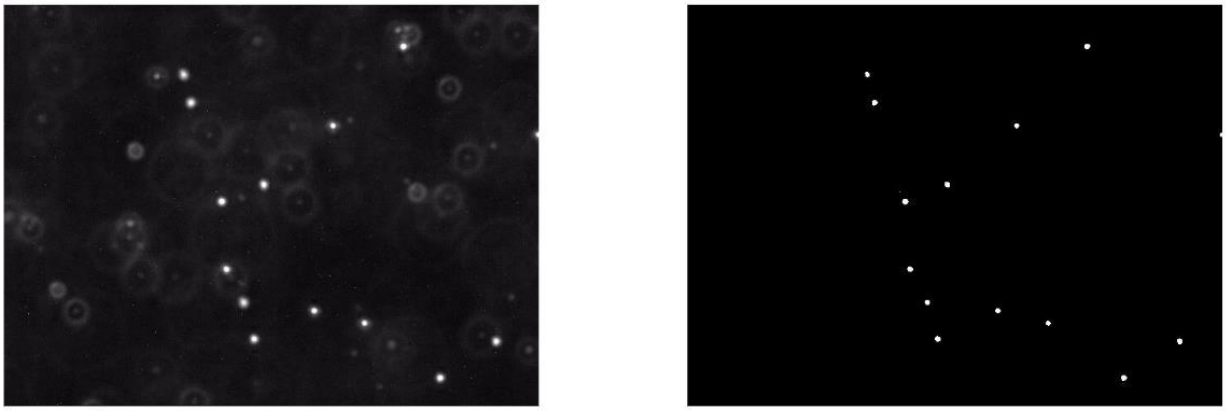
دیدگاه تاریخی. اتم نقش مهمی در فیزیک و شیمی قرن بیستم داشته است، اما وجود اتم‌ها و مولکول‌ها تا سال ۱۹۰۸ به طور عمومی پذیرفته نشده بود. در سال ۱۸۲۷ گیاه‌شناسی به نام رابرت براون از طریق مشاهده حرکت گرده‌های بید شناور در آب، به حرکت نامنظم تصادفی ذرات میکروسکوپی پی برد. این حرکت بعدها به عنوان حرکت براونی شناخته شد. اینشتین فرض کرد که این حرکت نتیجه برخورد میلیون‌ها ذره کوچک‌تر (اتم و مولکول) با ذرات بزرگ‌تر است.



انیشتین در یکی از مقالاتش (سال معجزه‌آسا) یک نظریه کمی از حرکت براونی را در تلاش برای توجیه وجود اتم‌ها با اندازه محدود مشخص تدوین کرد. نظریه او امکان ایجاد آزمایش‌هایی را با روشی برای شمارش مولکول‌ها توسط یک میکروسکوپ معمولی و به وسیله مشاهده اثر جمعی آن بر یک ذره غوطه‌ور بزرگ‌تر، فراهم کرد. در ۱۹۰۸ ژان باپتیست پرن، از اولترا میکروسکوپ که به‌تازگی اختراع شده بود، استفاده کرد تا به صورت تجربی نظریه جنبشی حرکت براونی انیشتین را تأیید کند. در نتیجه، وی اولین مدرک رسمی را که از ماهیت اتمی ماده حمایت می‌کرد فراهم آورد. آزمایش او همچنین یکی از اولین تخمین‌های عدد آگوادرو را ارائه داد. پرین به خاطر انجام این کار، برنده جایزه نوبل سال ۱۹۲۶ در فیزیک شد.

مسئله. در این پروژه، شما یک نسخه از آزمایش پرین را تکرار خواهید کرد. کار شما در اینجا بسیار ساده شده است؛ زیرا با فناوری مدرن ویدئو و کامپیوتر (و در کنار آن، مهارت‌های برنامه‌نویسی شما)، امکان اندازه‌گیری دقیق و ردیابی حرکت یک ذره غوطه‌ور در حرکت براونی وجود دارد. ما نمونه‌های میکروسکوپی تصویری را از کره‌ای پلی‌استایرن (بید) که در آب و با حرکت براونی به حالت شناور درآمده است، در اختیارتان قرار می‌دهیم. وظیفه شما نوشتن یک برنامه برای تجزیه و تحلیل این داده‌ها، تعیین میزان حرکت بیدها در بین مشاهدات، مطابقت این نمونه‌ها با مدل انیشتین و تخمین عدد آووگادرو است.

نمونه‌ها. ما ۱۰ مجموعه‌داده داریم که توسط ویلیام ریو با استفاده از تصویربرداری فلورسنت جمع‌آوری شده است. هر مجموعه‌داده در یک پوشه به نام run-1 تا run-10 ذخیره شده و شامل یک دنباله از دویست تصویر رنگی ۴۸۰ × ۶۴۰ با نام frame00000.jpg تا frame00199.jpg است. در داده‌هایی که در اختیار شما قرار داده شده است، یک فیلم از چند بید که در حال حرکت براونی هستند، آورده شده است. در زیر، یک تصویر خام معمولی (چپ) و یک نسخه تمیز شده (راست) با استفاده از حد آستانه که در زیر توضیح داده شده، دیده می‌شود.



هر تصویر یک مقطع دوبعدی از یک اسلاید میکروسکوپ را نشان می‌دهد. بیدها به سمت داخل و خارج از میدان دید (جهت x و y) حرکت می‌کنند. همچنین بیدها در جهت z نیز حرکت می‌کنند، بنابراین می‌توانند در عمق کانون عدسی نیز قرار بگیرند و یا خارج شوند. این امر می‌تواند منجر به ایجاد هاله و حتی محو شدن کامل بیدها از تصویر شود.

چالش ۱) شناسایی ذرات. اولین چالش، شناسایی بیدها در میان نمونه‌های نویزی است. هر تصویر ۶۴۰ در ۴۸۰ پیکسل است و هر پیکسل با یک شی Color نشان داده می‌شود که باید به مقدار درخشندگی بین صفر (سیاه) و ۲۵۵ (سفید) تبدیل شود. پیکسل‌های سفیدتر به بیدها (پیش‌زمینه) و پیکسل‌های سیاه‌تر به آب (پس‌زمینه) مربوط می‌شوند. ما مسئله را به سه بخش تقسیم

می‌کنیم: I) خواندن تصویر، II) دسته‌بندی پیکسل‌ها به عنوان پیش‌زمینه یا پس‌زمینه؛ III) یافتن دسته‌هایی از پیکسل‌ها به شکل دیسک که تشکیل دهنده بیدها هستند.

- **خواندن تصویر:** برای خواندن تصویر می‌توانید از پکیج matplotlib یا opencv استفاده کنید.
- **دسته‌بندی پیکسل‌ها به عنوان پیش‌زمینه یا پس‌زمینه:** ما از یک روش ساده اما مؤثر استفاده می‌کنیم که به عنوان آستانه‌گذاری برای جداکردن پیکسل‌ها در پیش‌زمینه و پس‌زمینه شناخته می‌شود: تمام پیکسل‌هایی که دارای مقدار درخشندگی تک‌رنگ کوچک‌تر از مقدار آستانه تاو هستند، پس‌زمینه و بقیه به عنوان پیش‌زمینه در نظر گرفته می شوند. دو تصویر بالا نشانگر تصویر اصلی (بالا سمت چپ) و همان تصویر پس از آستانه‌گذاری (بالا سمت راست)، با استفاده از مقدار تاو برابر با ۱۸۰ هستند.
- **پیدا کردن بلاب‌ها:** بید پلی‌استایرن توسط یک شکل دیسک مانند که حداقل با کمترین میزان min_pixels (معمولاً ۲۵) پیکسل پیش‌زمینه متصل به هم است، نشان داده می‌شود. یک مؤلفه متصل یا بلاب، یک مجموعه بیشینه از پیکسل‌های متصل به هم در پیش‌زمینه بدون در نظر گرفتن شکل و اندازه آن است. ما هر بلابی که حداقل تعداد پیکسل را داشته باشد (مثلاً ۲۵ پیکسل)، به عنوان بید در نظر خواهیم گرفت. مرکز بلاب (یا بید) میانگین مختصات x و y پیکسل‌های تشکیل دهنده آن است.

یک نوع دادهای کمکی Blob ایجاد کنید که دارای API زیر است:

Operation	description
()Blob	ایجاد یک بلاب خالی
b.add(x, y)	اضافه‌کردن پیکسل (x, y) به بلاب b
b.mass()	جرم بلاب (تعداد پیکسل‌های بلاب)
b.distanceTo(c)	فاصله اقلیدسی میان مرکز دو بلاب b و c
str(b)	ایجاد یک بازنمایی رشته‌ای از بلاب (bبه توضیحات زیر مراجعه کنید)

همچنین یک تابع main به منظور آزمایش تمامی متدهای تعریف شده در این API بنویسید.

بازنمایی رشته‌ای: متد str(b) یک رشته شامل جرم بلاب (یک عدد صحیح بیانگر تعداد پیکسل‌های بلاب)، سپس یک فاصله، سپس مختصات x و y مرکز بلاب (محصور در پرانتز و جدا شده با یک ویرگول و فاصله، با دقت چهار رقم اعشار) برمی‌گرداند. به مثال زیر که بیانگر یک بلاب شامل ۳۱ پیکسل و به مرکز (370.9355, 365.4194) است توجه کنید:

31 (370.9355, 365.4194)

توجه. سازنده و متدها باید پیچیدگی زمانی ثابتی داشته باشند.

سپس، یک نوع دادهای با نام BeadFinder بنویسید که API زیر را دارد. از یک روش جستجوی کارا برای شناسایی بلاب‌ها و بیدها به شیوه مؤثر استفاده کنید (به پروژه تراوش مراجعه نمایید. در آن پروژه شبیه همین کار را برای یافتن یک مسیر باز از خانه‌های سطر اول گرید به خانه‌های سطر آخر آن انجام دادیم).

Operation	description
BeadFinder(picture, tau)	**یافتن تمامی بلاب‌ها در تصویر مشخص شده با استفاده از مقدار آستانه تاو**
bf.getBeads(min_pixels)	**برگرداندن یک لیست شامل تمامی بیدها (بلاب‌هایی که تعداد پیکسل‌های آنها بزرگ‌تر یا مساوی min_pixels است)**

همچنین یک تابع main به منظور آزمایش تمامی متدهای تعریف شده در این API بنویسید. تابع main() عدد صحیح min_pixels ، عدد اعشاری tau و نام تصویر ورودی را به عنوان آرگومان‌های خط فرمان دریافت می‌کند و سپس یک شی BeadFinder را با مقدار آستانه tau ایجاد می‌کند؛ سپس تمام بیدها را (که شامل حداقل min_pixels پیکسل هستند) چاپ می‌کند. در ادامه، یک نمونه از اجرای ماژول beadfinder.py به منظور راهنمایی شما آمده است:

```
% python beadfinder.py 0 180.0 run_1/frame00001.jpg
29 (214.7241,  82.8276)
36 (223.6111, 116.6667)
 1 (254.0000, 223.0000)
42 (260.2381, 234.8571)
35 (266.0286, 315.7143)
31 (286.5806, 355.4516)
37 (299.0541, 399.1351)
35 (310.5143, 214.6000)
31 (370.9355, 365.4194)
28 (393.5000, 144.2143)
27 (431.2593, 380.4074)
36 (477.8611,  49.3889)
38 (521.7105, 445.8421)
35 (588.5714, 402.1143)
13 (638.1538, 155.0000)

% python beadfinder.py 25 180.0 run_1/frame00001.jpg
29 (214.7241,  82.8276)
36 (223.6111, 116.6667)
42 (260.2381, 234.8571)
35 (266.0286, 315.7143)
31 (286.5806, 355.4516)
37 (299.0541, 399.1351)
35 (310.5143, 214.6000)
31 (370.9355, 365.4194)
28 (393.5000, 144.2143)
27 (431.2593, 380.4074)
36 (477.8611,  49.3889)
38 (521.7105, 445.8421)
35 (588.5714, 402.1143)
```

در تصویر نمونه run_1/frame00001.jpg ، ۱۵ بلاب وجود دارد که ۱۳ تای آنها بید هستند (شامل حداقل ۲۵ پیکسل هستند.)

چالش ۲) ردیابی ذرات: گام بعدی تعیین میزان حرکت بید در بازه زمانی t تا $t + \Delta t$ است. برای نمونه‌های ما $\Delta t = 0.5$ ثانیه بین فریم‌ها وجود دارد. فرض کنید که نمونه‌ها به گونه‌ای هستند که هر بید به مقدار نسبتاً کوچکی حرکت می‌کند و این بیدها هرگز با یکدیگر برخورد نمی‌کنند. (با این حال باید توجه داشته باشید که احتمال می‌رود که بید از فریم یا با خروج میدان دید میکروسکوپ در جهت x یا y و یا حرکت در عمق کانون عدسی میکروسکوپ در جهت z خارج شود) بنابراین، برای هر بید در زمان $t + \Delta t$ ، نزدیک‌ترین بید را در زمان t (با فاصله اقلیدسی) پیدا کنید و این دو را به عنوان یک بید مشخص کنید. با این حال، اگر فاصله خیلی بزرگ‌تر از دلتا پیکسل باشد، فرض کنید که یکی از بیدها، سفر خود را آغاز کرده یا به پایان رسانیده است.

تابع main() را در BeadTracker به صورتی بنویسید که عدد صحیح min_pixels ، عدد اعشاری tau ، عدد اعشاری delta و دنباله‌ای از نام‌های تصاویر را به عنوان آرگومان‌های خط فرمان دریافت کند؛ بیدها را برای هر تصویر (توسط BeadFinder شناسایی کند (با استفاده از مقادیر مشخص min_pixels و tau)؛ و فاصله‌ای را که هر بید از یک فریم به فریم بعدی طی می‌کند را (با فرض

اینکه فاصله نمی‌تواند بیشتر از delta باشد) چاپ کند. شما این کار را برای بیدهای هر جفت فریم متوالی با چاپ هر فاصله‌ای که می‌یابید انجام خواهید داد. در ادامه یک نمونه از اجرای ماژول beadtracker.py به منظور راهنمایی شما آمده است:

```
% python beadtracker.py 25 180.0 25.0 run_1/*.jpg
 7.1833
 4.7932
 2.1693
 5.5287
 5.4292
 4.3962

...
```

توجه: با این روش، نیازی به ساختن یک ساختار دادهای نیست که یک بید مشخص را از طریق یک توالی از فریم‌ها دنبال کند.

چالش ۳) تحلیل داده‌ها: نظریه حرکت براونی انیشتین خواص میکروسکوپی (به عنوان مثال، شعاع، نفوذ) بیدها را به خواص ماکروسکوپی (به عنوان مثال، دما، گرانروی) مایعی که بیدها در آن غوطه‌ور هستند مرتبط می‌کند. این نظریه شگفت‌انگیز به ما این امکان را می‌دهد که عدد آووگادرو را با مشاهده اثر جمعی میلیون‌ها مولکول آب روی بیدها توسط یک میکروسکوپ معمولی تخمین بزنیم.

- **تخمین ثابت خود انتشاری**. ثابت خود انتشاری D ، حرکت تصادفی یک مولکول (بید) را از طریق یک سیال همگن (مولکول‌های آب) به عنوان نتیجه انرژی گرمایی تصادفی توصیف می‌کند. معادله اینشتین - اسمولاچوفسکی بیان می‌کند که جابه‌جایی تصادفی بید در یک بعد، یک توزیع گاوسی با میانگین صفر و واریانس $\sigma^2 = 2D\Delta t$ دارد، که در آن Δt فاصله زمانی بین اندازه‌گیری موقعیت است. به این معنا که میانگین جابهجایی یک مولکول صفر است و میانگین مربع جابهجایی آن با زمانی که بین اندازهگیریها است (با ضریب ثابت$2D$) متناسب است. ما σ^2 را با محاسبه واریانس تمامی جابهجایی بیدها در جهتهای x و y تخمین میزنیم. اگر $(\Delta x_1 . \Delta y_1) . \dots . (\Delta x_n . \Delta y_n)$ بیانگر جابهجایی n بید و $r_1 . \dots . r_n$ بیانگر جابهجایی شعاعی باشد. آنگاه:

$$\hat{\sigma}^2 = \frac{(\Delta x_1^2 + \cdots + \Delta x_n^2) + (\Delta y_1^2 + \cdots + \Delta y_n^2)}{2n} = \frac{r_1^2 + \cdots + r_1^2}{2n}$$

برای دادههای ما $\Delta t = 0.5$ است؛ بنابراین $\hat{\sigma}^2$ یک تخمین برای D است.

توجه داشته باشید که جابه‌جایی شعاعی در رابطه بالا برحسب متر اندازه‌گیری می‌شود. تغییرات شعاعی خروجی توسط برنامه beadtracker.py شما برحسب پیکسل است. برای تبدیل از پیکسل به متر، در $0.175 * 10^{-6}$ ضرب کنید (متر در هر پیکسل).

- **تخمین ثابت بولتزمن.** رابطه استوکس - انیشتین ادعا می‌کند که ثابت خود انتشاری یک ذره کروی که در سیال غوطه‌ور شده است، توسط رابطه زیر محاسبه می‌شود.

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

که برای نمونه ما:

- T: دمای مطلق برابر ۲۹۷ کلوین (دمای اتاق)
- η: ویسکوزیته آب در دمای اتاق $9.135 \times 10^{-4} N.s.m^{-2}$
- ρ: شعاع بید برابر 0.5×10^{-6} متر

و k ثابت بولتزمن است. تمامی پارامترها در واحدهای SI ارائه می‌شوند. ثابت بولتزمن یک ثابت فیزیکی بنیادی است که متوسط انرژی جنبشی یک مولکول را به دمای آن مرتبط می‌سازد. ما k را با محاسبه تمام پارامترها توسط معادله استوکس - انیشتین و حل آن برای k تخمین می‌زنیم.

- **تخمین عدد آووگادرو.** عدد آووگادرو N_A به عنوان تعداد ذرات در مول تعریف میشود. با تعریف $k = \frac{R}{N_A}$ که در آن ثابت جهانی گاز R برابر است با ۸,۳۱۴۴۶ ؛ از R / k به عنوان تخمینی از عدد آووگادرو استفاده می‌کند.

به عنوان بخش آخر، تابع main() را در avogadro.py به شکلی بنویسید که جابه‌جایی شعاعی $(r_1, r_2, r_3, \dots)$را از ورودی استاندارد خوانده و ثابت بولتزمن و عدد آووگادرو را با استفاده از روابطی که در بالا شرح داده شد تخمین بزند.

```
% more displacements-run_1.txt
 7.1833
 4.7932
 2.1693
 5.5287
 5.4292
 4.3962
...

% python Avogadro.py < displacements-run_1.txt
Boltzmann = 1.2535e-23
Avogadro  = 6.6329e+23

% python beadtracker.py 25 180.0 25.0 run_1/*.jpg | python Avogadro.py
Boltzmann = 1.2535e-23
Avogadro  = 6.6329e+23
```

قالب خروجی. همه‌جا از چهار رقم دقت بعد از اعشار استفاده کنید.

نمونه‌ها. شما می‌توانید مجموعه‌داده‌ها و فایل‌های کمکی دیگر را از atomic.zip دانلود کنید.

آنچه باید تحویل دهید. شما باید فایل‌های blob.py ، beadfinder.py ، beadtracker.py و avogrado.py را در یک پوشه با شماره دانشجویی خود ارسال کنید.