Детковский Алексей, 10 класс

АНОО “Физтех-лицей”имени П.Л.Капицы

Система детектирования космических тел

Руководитель Капранов В.В.

Начнем с краткого введения в проект: основной задачей, стоящей передо мной, было сконструировать и запрограммировать прибор, детектирующий движения тел, а так же определяющий их химический состав.

Ну и первая вещь, с которой необходимо было разобраться – как будет выглядеть прибор. Данный экземпляр представляет собой два окуляра, соединенных вместе и зафиксированных на оргстекле. К левому подсоединена камера, а так же микрокомпьютер Rasberry Pi ,отвечающий за корректность ее работы; правый же используется спектрометром. Также на всей конструкции по периметру располагаются винты для корректирования углов обзоров(тут важен каждый миллиметр)

Одна из сложностей конструкции была в том, чтобы правильно настроить изображения на окулярах, то есть то, что фиксирует камера , должен видеть и спектрометр(а так же то, что всвязи с размерами прототипа на расстояние большем 12 метров уже наблюдаются проблемы с корректностью картинки с камеры) .

Несколько слов о камере. Главная сложность была в том, чтобы обучить камеру распознавать вспышки. Из модулей питона(а именно на этом языке написана вся аппаратная часть проекта) были взяты Numpy и OpenCv(так же Picamera, но лишь для основных настроек камеры). Первое, что следовало сделать – перевести изображение из трехканального цвета RGB в одноканальный Gray , затем размытие. В программе использовалось Гауссово, как одно из самых простых в исполнении и результативных .

С точки зрения математики размытие по Гауссу выглядит следующим образом:

$x_{med} = \frac{\sum_{i=1}^{n} x_i}{n}$ медиана выборки

$\sigma(x) = \frac{\sum_{i=1}^{n} (x_i - x_{med})^2}{n-1}$ отклонение данной переменной от выборки массива

$Ga(x, y) = \frac{-(x - x_{\max}\ )^2}{2\sigma_x^2} + \frac{-(y - y_{\max}\ )^2}{2\sigma_y^2}$ Гауссово размытие

Далее следовало увеличить контрастность(темные места сделать черными, светлые – светлыми, если просто, то наложение ограничение на значения переменных в массиве), а за ним проверка на то, что у нас действительно движущийся 'яркий' объект, а не какая-то часть фона – вычитание картинок с разностью в кадр(param = framerate) , поверх этого изображения накладывается маска(все то,что в нее попадает – белое, не попадает – черное), после чего опять обрезаем и накладываем еще один круг, ну а теперь финальная проверка – проверка эллипса на площадь, а если быть точнее проверка фигуры на подобность эллипсу и счет площади.

Другая интересная задача – автомотизированность процесса (подключив сервисы Google можно настроить сбор изображения данных со спектрометра, речь о котором пойдет дальше) , что несомненно облегчает отбор изображений и ,в принципе, очень удобно с практической точки зрения.

Для спектрального анализа в первую очередь необходимо было собрать базу данных с химическими элементами , в помощь опять пришел питон и Jupyter Labs.

Далее , основанная задача спектрального анализа – определение пиков на графике, так как пик означает присутствие данного химического элемента. Кроме того, сам процесс счета так же был автоматизирован(камера , заметившая изображение, отправляет сигнал на сервер, который стоит на ноутбуке, в свою очередь он запускает скрипт спектрометра, строящий график и выводящий результаты с поиском химических элементов)

И , пожалуй , последняя и самая интересная вещь, которая находится на данном прототипе – ‘сглаживание’ графика, а если быть точнее метод наименьших квадратов, а если быть еще точнее – метод Ньютона-Гаусса . Сам метод был выбран как один из самых продуктивных (с точки зрения работы программы), а так же как один из самых интересных, кроме того обычный метод Ньютона не подходит для нахождения 'сглаженных графиков' для кривых второго порядка, как и метод Наискорейшего спуска, который не только может, а скорее всего будет очень долго сходиться(конечно, метеоритные тела могут двигаться и по прямой относительно камеры, если находятся на достаточно большом расстоянии от нее, но приближение окуляра камеры(а если быть точнее, то отдаление), которое гораздо полезнее нежели зеркало, о котором расскажется ниже, поможет исправить эту небольшую неточность)

Если кратко , то используя матрицу Якобиана(сейчас будет много сложных математических терминов прямиком из линейной алгебры), первых производных функции F как раз для нахождения вектора x_k значений параметра, который минимизирует сумму квадратных отклонений (которые мы предсказали, от тех , что присутствуют в опыте).

По шагам:

1. Сделаем начальное предположение x_0 для x
2. Сделаем предположение при k = 1
3. Создадим вектор f_k с элементами $f_i(x_k)$
4. Создадим матрицу Якоби
5. Решить следующее матричное уравнение $J^T_k J_k p_k = - J^T_k f_k$, где p – коэффициент ‘предсказания'(вероятности), решением будут все вероятности p для всех k

6. Решить F(xk + spk) относительно s, где F должно удовлетворять условиям Вольфе
7. Пусть xk+1 = xk + spk
8. Повторять шаги 1-7 до схождения

Краткое пояснения:

Матрицей Якоби называется матрица по частным производным функции f

$$
\begin{matrix}
\frac{\delta f1}{\delta x_1} & \frac{\delta f1}{\delta x_2} & **** & \frac{\delta f1}{\delta x_m} \\
 & & & * \\
 & & & * \\
 & & & * \\
\frac{\delta fn}{\delta x_1} & \frac{\delta f_n}{\delta x_2} & **** & \frac{\delta f_n}{\delta x_m}
\end{matrix}
$$

В случае m = n(а это именно наш случай, определитель данной матрицы Jf(x) :

$$Jf(x) = \frac{\partial(f_1, f_2, \dots \dots f_n)}{\partial(x_1, x_2, \dots \dots x_n)} = \begin{matrix} \frac{\delta f1}{\delta x_1} & \frac{\delta f1}{\delta x_2} & ****** \frac{\delta f1}{\delta n} \\ * & * & * \\ \frac{\delta f_n}{\delta x_1} & \frac{\delta f_n}{\delta x_2} & ****** \frac{\delta f_n}{\delta x_n} \end{matrix}$$

Условия Вольфе , связанные с теорией оптимизации , впервые были опубликованы в 1969г.

Если кратко, то если есть некоторое x_k решение уравнения (достаточно приближенного) и был найден вектор направления p_k, то тогда $x_{k+1} = x_k + \alpha\ p_k$, , где α удовлетворяет условиям:

$$f(x_k + \alpha_k * p_k) \leq f(x_k) + c_1 \alpha_k \nabla f_k^T \quad p_k$$

$$\nabla f(x_k + \ \alpha_k p_k)^T p_k \geq c_2 \nabla f_k^T p_k$$

С точки зрения всех формул мне повезло так как в python уже есть готовые решения(кроме Ньтона-Гаусса)

В принципе,под конец хочется сказать о том самом зеркале: в данной установке с точки зрения конструкции мало, что интересного; была попытка поставить зеркало на электроприводе(для увеличения угла обзора), которая уже на теоретической части показала свою неэффективность

Воспользуемся школьным курсом физики:

Расчет будем вести в граммах так как в документации у электроприводов момент в *г/см*
$m_{\text{ср}} = \ \ \rho_{\{\text{ср}\}} * V \leftrightarrow \frac{2.2+7.5}{2} * (0.5 * 4 * 20) \ \ = 0.243\ гр$

Кроме того к самому стеклу должна крепится металлическая пластинка(если быть точнее система из двух, соединенных шарниром) данная модель(в собранном виде, так как не было трудностей в ее сборке) имеет массу ≈ 350 гр

Конечно, рассматривать среднюю плотность стекла ,которое встречается, таким образом некорректно, но в данном случае мы делаем оценку, чтобы примерно прикинуть ,что нам нужно

И отсюда находим момент:

$M = (m_{cp} + m_{cm}) * g * l \ \leftrightarrow (0.243 + 0.35) * 10 * 20 * 10^{-3} = 1186 * 10^{-4}$ *кг*см*

Для такого куска зеркала с лихвой хватит FL20STH30-0604A, сам прибор потребляет 0.6A и имеет длину 30мм, тогда за 1 час непрерывной работы будет затрачена $P = I^2 * R = 2.34\ Вт$ (все также рассматриваем максимальные величины(то есть с учетом того, что модель будет работать весь час без остановки) . Фактически это немного, даже очень мало, по меркам потребляемы мощностей камеры и компьютером, но что же за это получаем?

Почти все время зеркало будет находится в прямом положении(по касательной), чтобы не закрывать основной обзор камеры, изредка двигаясь, в связи с чем данное дополнение было признано нецелесообразным.

Источники:

https://ib.mazurok.com/2019/06/14/jacobian-matrix-and-determinant/

https://google-info.org/4809561/1/usloviya-volfe.html

http://www.wiki-wiki.ru/wp/index.php/%D0%A3%D1%81%D0%BB%D0%BE%D0%B2%D0%B8%D1%8F_%D0%92%D0%BE%D0%BB%D1%8C%D1%84%D0%B5

https://habr.com/ru/post/308626/

http://www.machinelearning.ru/wiki/index.php?title=%D0%92%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BC%D0%B0%D1%82%D1%80%D0%B8%D1%86_%D0%AF%D0%BA%D0%BE%D0%B1%D0%B8_%D0%B8_%D0%93%D0%B5%D1%81%D1%81%D0%B5

http://vmath.ru/vf5/algebra2/dets/jacobian

http://www.machinelearning.ru/wiki/index.php?title=%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%9D%D1%8C%D1%8E%D1%82%D0%BE%D0%BD%D0%B0-%D0%93%D0%B0%D1%83%D1%81%D1%81%D0%B0

https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%93%D0%B0%D1%83%D1%81%D1%81%D0%B0_%E2%80%94_%D0%9D%D1%8C%D1%8E%D1%82%D0%BE%D0%BD%D0%B0

https://ru.stackoverflow.com/questions/1150851/%D0%BC%D0%B5%D1%82%D0%BE%D0%B4-%D0%B3%D0%B0%D1%83%D1%81%D1%81%D0%B0-%D0%BD%D0%B0-python-3

https://stackoverflow.com/questions/49553006/compute-the-jacobian-matrix-in-python

https://wiki2.org/ru/%D0%AF%D0%BA%D0%BE%D0%B1%D0%B8%D0%B0%D0%BD

https://pyprog.pro/linear_algebra_functions/linalg_solve.html

http://espressocode.top/

https://electroprivod.ru/fl28sth.htm

https://habr.com/ru/post/277421/

https://wikichi.ru/wiki/Matrix_analysis

https://zaochnik.ru/blog/matricy-i-osnovnye-dejstviya-nad-nimi/

https://habr.com/ru/post/151157/

https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%93%D0%B0%D1%83%D1%81%D1%81%D0%B0_(%D0%BE%D0%BF%D1%82%D0%B8%D0%BC%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F)

https://ru.qaz.wiki/wiki/Gaussian_blur

https://vuzlit.ru/968115/teoreticheskaya_osobennosti_optimizatsii_metoda_gaussa_zeydelya

Зорич учебник по математическому анализу

Кудрявцев лекции по математическому анализу

https://ru.wikipedia.org/wiki/%D0%A2%D0%B0%D0%B1%D0%BB%D0%B8%D1%86%D0%B0_%D1%85%D0%B0%D1%80%D0%B0%D0%BA%D1%82%D0%B5%D1%80%D0%B8%D1%81%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D1%85_%D1%87%D0%B0%D1%81%D1%82%D0%BE%D1%82_%D0%B2_%D0%B8%D0%BD%D1%84%D1%80%D0%B0%D0%BA%D1%80%D0%B0%D1%81%D0%BD%D0%BE%D0%B9_%D1%81%D0%BF%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D1%81%D0%BA%D0%BE%D0%BF%D0%B8%D0%B8

http://chem.spbu.ru/files/Vladimir/Vasiliev/Mass.pdf

http://chem.spbu.ru/files/Vladimir/Vasiliev/Introduction.pdf