Разработка алгоритмов для распознавания черенковских колец в детекторе ФАРИЧ

Студент: Носорев Константин, ИШ НГУ 19137
Научный руководитель: Марченко Михаил Александрович, д.ф.-м.н, и.о. зав. Каф. ВС ММФ, директор ИВМиМГ СО РАН, профессор РАН
Научный соруководитель: Городничев Максим Александрович, ст. преп. Каф ВС ММФ, н.с. ИВМиМГ СО РАН

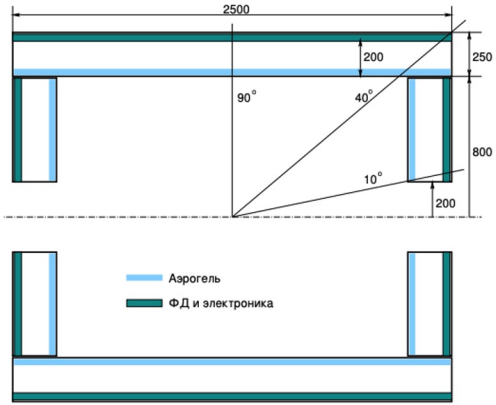
04.04.2023

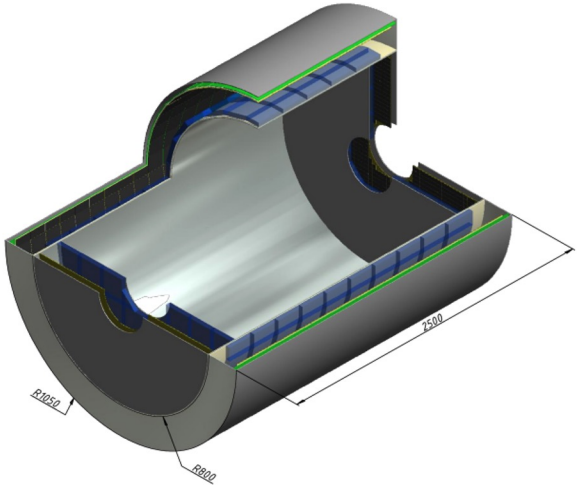
План доклада

1. Контекст исследования
2. Постановка задачи
3. Описание входных данных
4. Формальная постановка задачи
5. Моделирование данных
6. Обзор существующих подходов
7. Предлагаемое решение
8. Возможные улучшения
9. Вывод

Супер – Чарм – Тау фабрика

Проект электрон-позитронного коллайдера в Институте ядерной физики им. Г. И. Будкера СО РАН.
Одной из важных частей проекта является разработка системы детекторов, один из которых – FARICH.

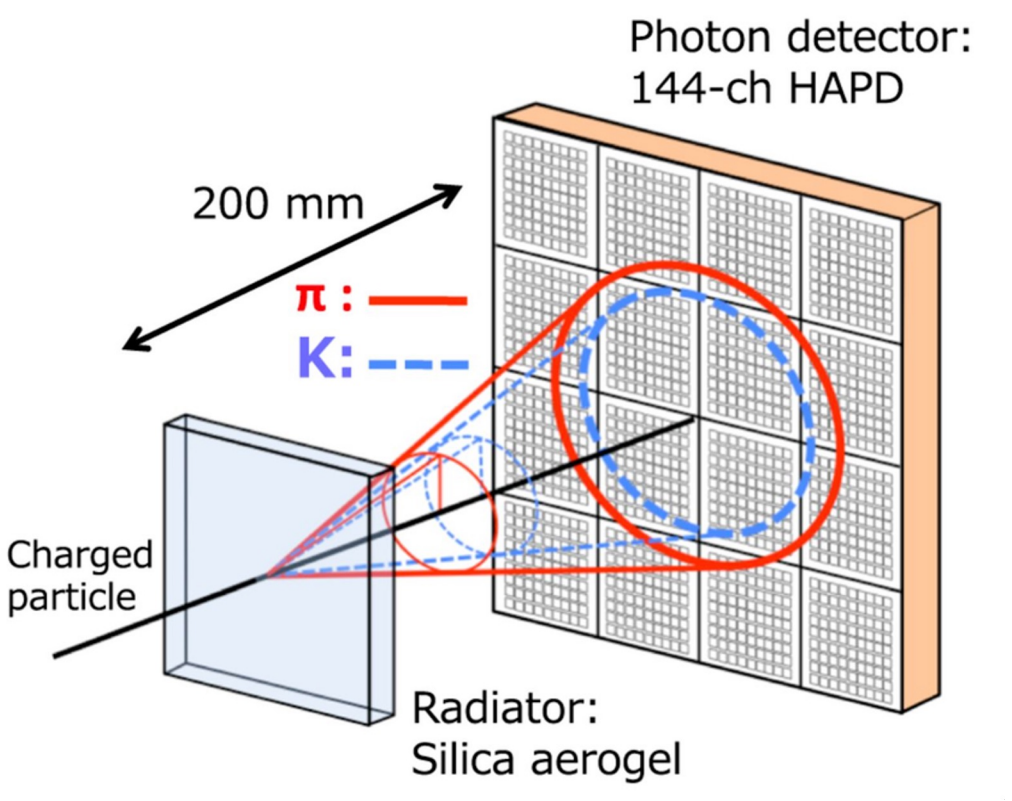




RICH-детектор частиц

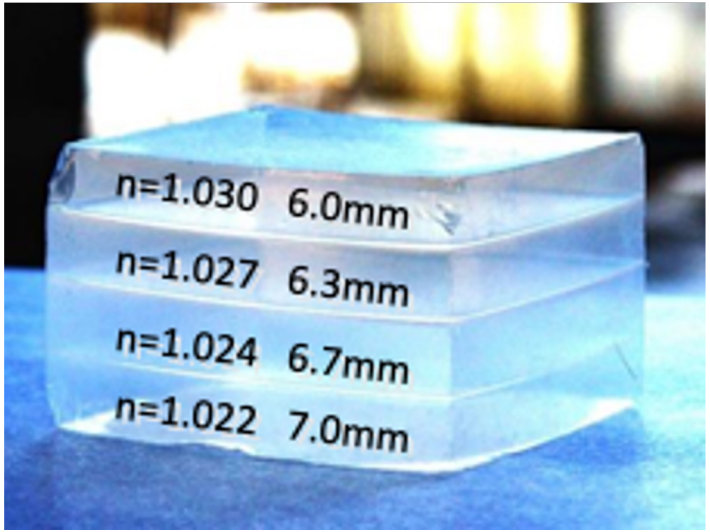
Rich **I**maging **CH**erenkov

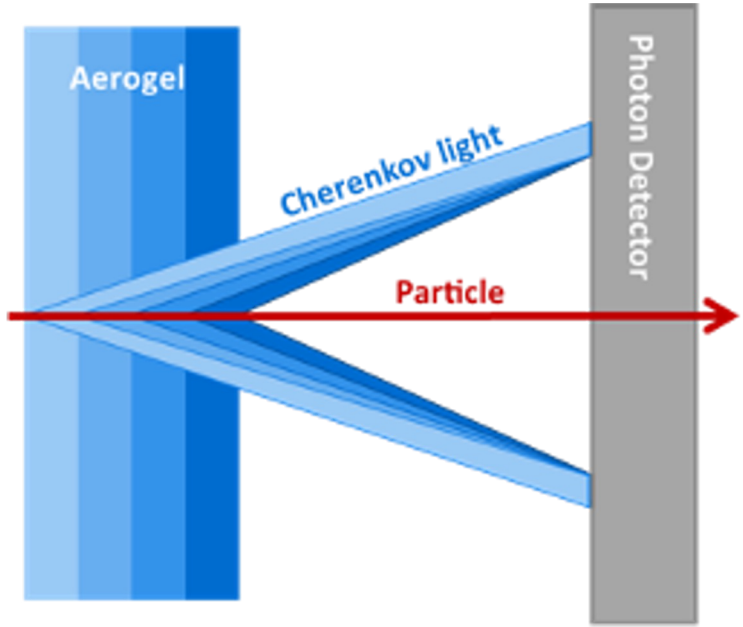
Принцип работы RICH-детекторов, основывается на эффекте Вавилова — Черенкова.



FARICH-детектор частиц

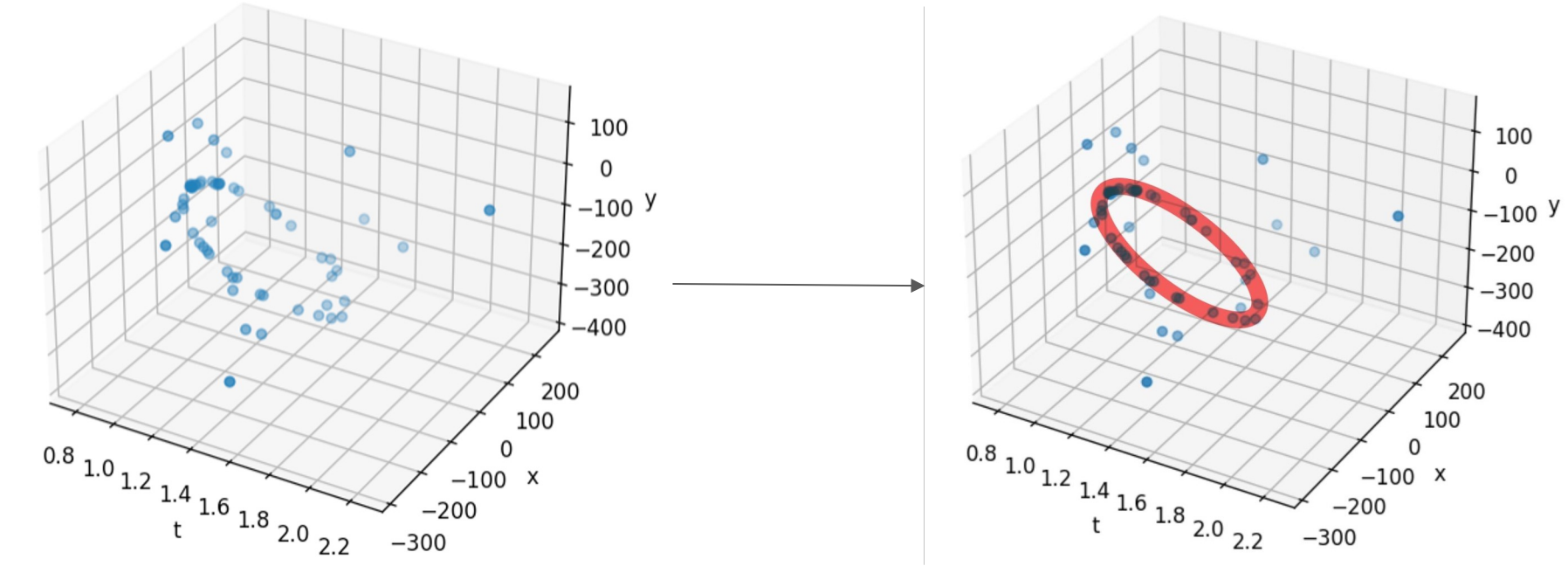
Focusing **A**erogel **RICH**





Постановка задачи

Разработать алгоритм, позволяющий распознавать черенковские кольца в потоке событий с заданной точностью.



Поток событий

Пусть N_x, N_y – число пикселей в детекторе по осям X и Y соответственно

$P_x = \{ x \in Z^+ | \, 0 \leq x \leq N_x \}$ – множество индексов пикселей детектора по оси X

$P_y = \{ y \in Z^+ | \, 0 \leq y \leq N_y \}$ – множество индексов пикселей детектора по оси Y

$$events = \{(x_t, y_t, t) \mid t_0 \leq t \leq T, x_t \in P_x, y_t \in P_y\}$$

entry	subentry	x_c	y_c	t_c
5	0	19	236	4.639383
	1	96	97	3.122469
	2	90	112	3.018600
	3	90	113	3.019780
	4	89	118	3.019531
	5	92	125	3.006703
	6	91	122	3.008479
	7	90	122	3.013435
	8	92	126	3.005071
	9	95	129	2.997678

Пример входных данных

Формальная постановка задачи

Пусть $events_{t_0,T} = \{(x_t, y_t, t) \mid t_0 \leq t \leq T, x_t \in P_x, y_t \in P_y\}$

Хотим построить алгоритм, который находит

$ellipse^* \subset events_{t_0,T}$, т.ч $|ellipse^*| < MinPoints$ и $\exists\pi$ – плоскость, в которой $ellipse^*$ образует эллипс, где $MinPoints$ – параметр алгоритма.

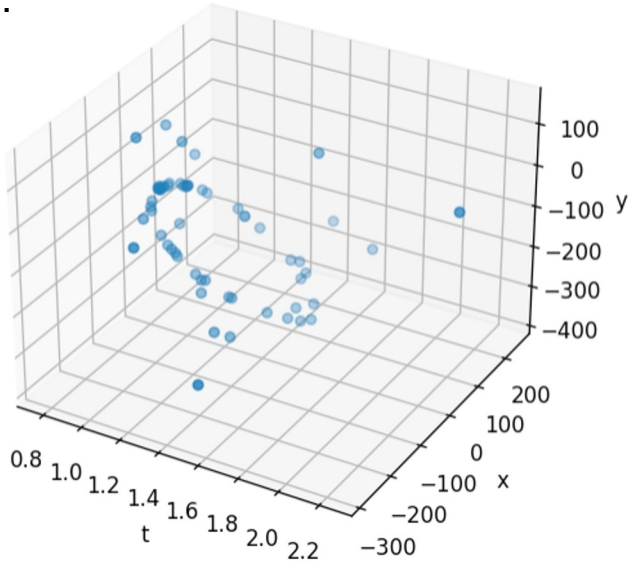
Причем, $\forall ellipse^{**}$, удовлетворяющий условию выше, то $|ellipse^*| \geq |ellipse^{**}|$.

Цель исследования

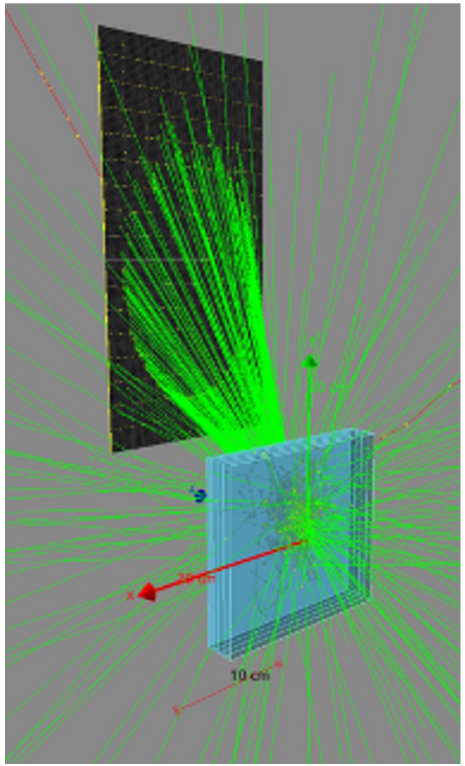
Разработать алгоритм детекции черенковских колец в детекторе ФАРИЧ с точностью не менее 80% и временем обработки одного события не более 1 секунды.

Моделирование входных данных

Сотрудниками института ядерной физики СО РАН была построена модель детектора в программном пакете Geant4.



События из модели



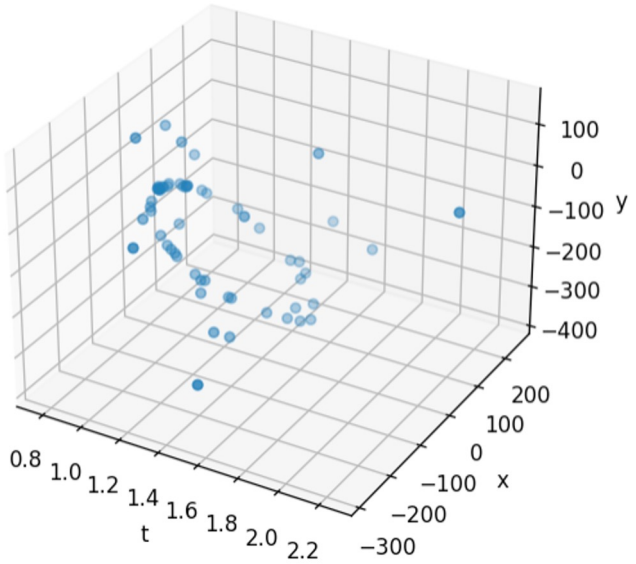
Модель в Geant4

Моделирование входных данных

Объем данных: 1,2 млн экспериментов.

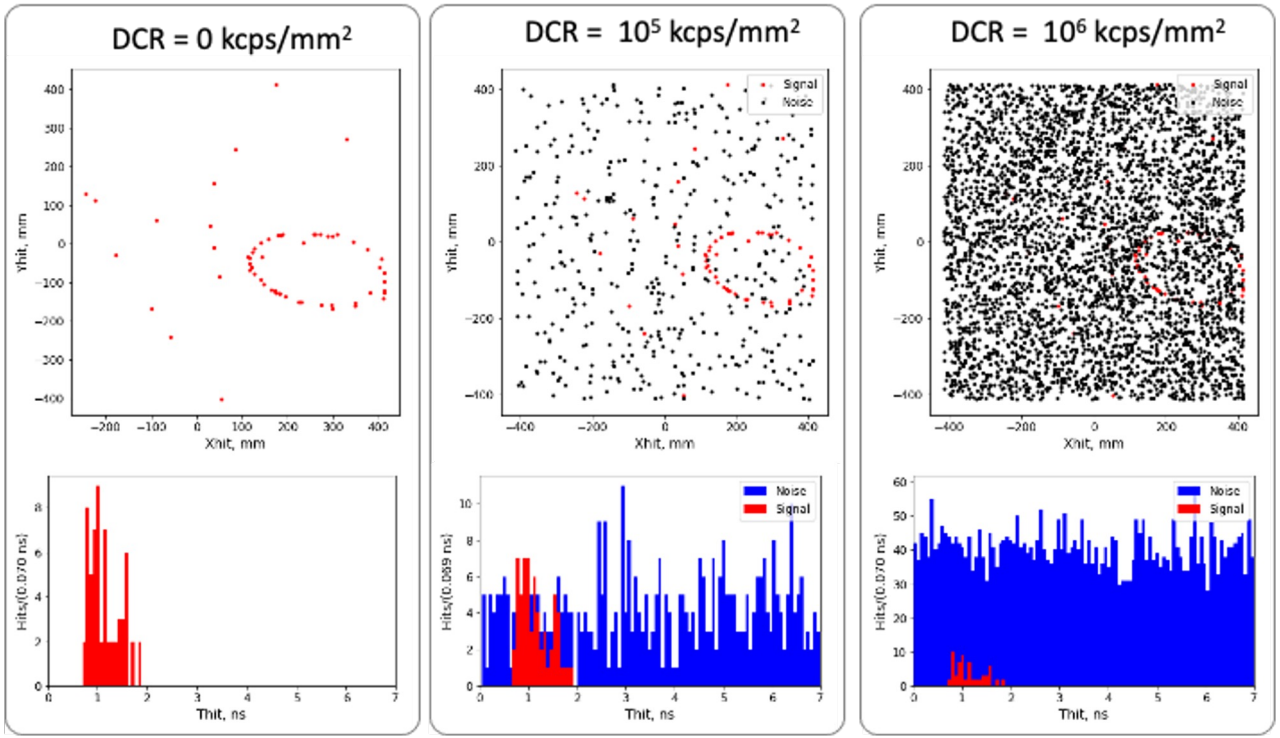
Этапы предобработки данных:

1. Генерация шума с заданным распределением (распределение Пуассона с параметром λ) с сохранением метки сигнал/шум
2. Дискретизация точек по оси X и Y
3. Сортировка данных по оси времени



События из модели

Моделирование входных данных



Проекция событий на плоскость XY
с различными параметрами λ

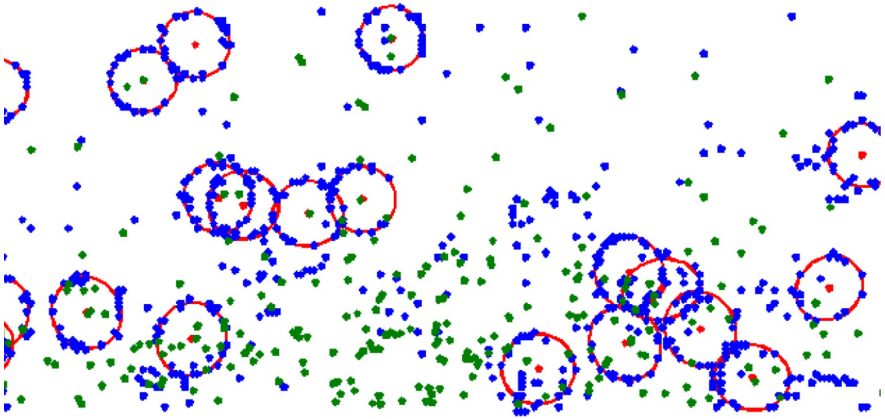
Существующие подходы к решению задачи

1. Метод для RICH детекции
    1. Преобразования Хафа
    2. Нейронные сети

Использование преобразования Хафа

Подход:

1. Взять область вокруг некоторый точки (радиусом R_{max})
2. С помощью преобразования Хафа строится 2D гистограмма центров и 1D гистограмма радиусов
3. Выбор ответов с помощью нейронных сетей



Результаты:

- Для крупных колец (N2-RICH) точность достигает 92%
- Для меньших колец (CO2-RICH) точность достигает 90%

Недостатки для данных FARICH:

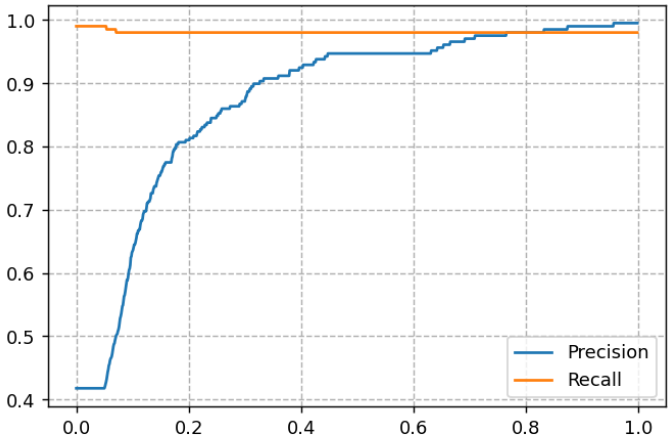
- В данных из статьи уровень шума, много ниже модельных данных
- Не используют информацию о времени прилета

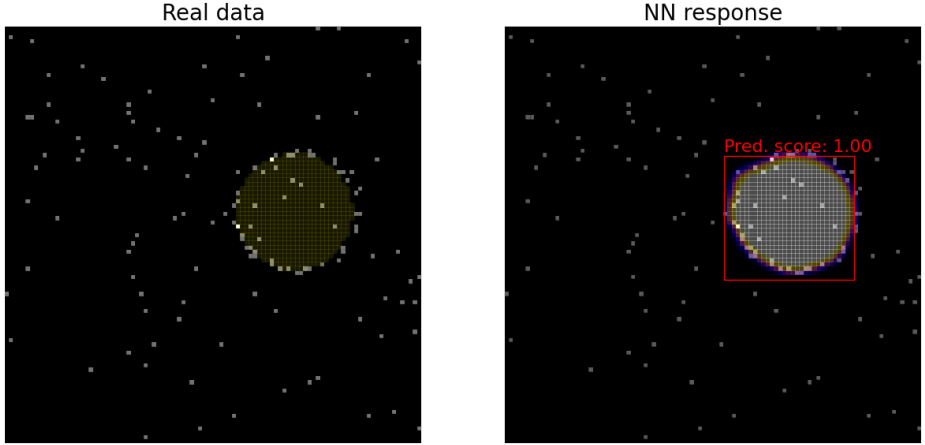
Lebedev S., H¨ohne C., Ososkov G. Fast ring recognition algorithm for the rich detector of the CBM experiment at fair // Discrete and Continuous Models and Applied Computational Science. 2010. №3-2.

Нейронные сети

Подход:

1. Проецируем данные на плоскость XY
2. С помощью НС сегментируем изображение





Недостатки для данных FARICH:

- Подход разобран для задачи без шума
- Распознает только круги, т.е. в случае когда частица прилетает перпендикулярно детектору
- Не используют информацию о времени прилета

Помулева С.И. Реконструкция черенковских колец в детекторе ФАРИЧ с использованием методов машинного обучения // ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА ФФ НГУ 2021

Предлагаемое решение

Подход:

1. Выбираем 3 произвольные точки.
2. Строим плоскость по 3 точкам
3. Проецируем все точки, находящиеся на расстоянии не больше, чем ε_p от плоскости
4. Если точек в плоскости меньше, чем $MinPoints$, то переходим к шагу 1
5. Ищем эллипс в плоскости, с помощью метода наименьших квадратов
6. Сохраняем параметры эллипса и набор точек, на которых он был получен.
7. Если остались точки, которые не обработаны, то идем на шаг 1
8. Выбираем итоговый эллипс*
9. Возвращаем набор точек, которые попали в итоговый эллипс

Параметры:

1. $\mathrm{MinPoints}$ – минимальное число точек в эллипсе
2. ε_{p}- допустимое расстояние проецируемой точки от плоскости

* Было рассмотрено несколько способов выбирать итоговый эллипс:

1. Нахождение средней точки центра
2. Выбор эллипса с максимальным числом точек

Предлагаемое решение. Нахождение средней точки центра

Подход:

1. Считаем медиану по все осям
2. Получаем точку «потенциального центра»
3. Выбираем все эллипсы, центр которых находится на расстоянии не больше, чем заданный параметр
4. Находим объединение множеств всех точек выбранных эллипсов
5. Возвращаем полученное множество

Недостатки:

- Метод слишком чувствителен к шумам
- Метод может не найти полный эллипс, а лишь какую-то часть от эллипса
- Метрика точности ~60% (среднее значение по тестовой выборке)

Предлагаемое решение. Выбор эллипса с максимальным числом точек

Подход:

1. Считаем медиану по всем осям
2. Получаем точку «потенциального центра»
3. Выбираем эллипс с максимальным числом точек
4. Возвращаем точки, которые покрыл полученный эллипс

Преимущества:

- Устойчив к шуму
- Требует меньше вычислений
- Метрика точности ~90% (среднее значение по тестовой выборке)

Недостатки:

- Метод выбирает лишние точки в решение
- Возможна маловероятная ситуация, когда шум образует эллипс из большего числа точек*

* Поскольку разрабатываемое решение, будет использоваться, как первая линия обработки данных, то выбор лишних событий не так критичен.

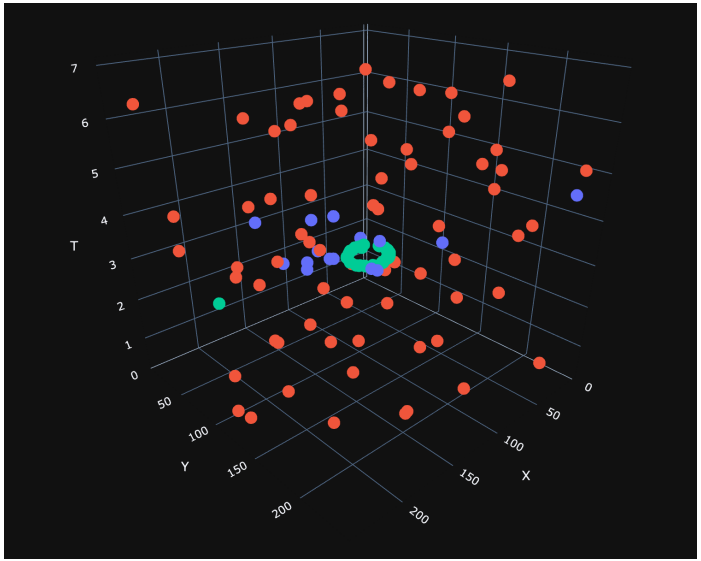
Предлагаемое решение

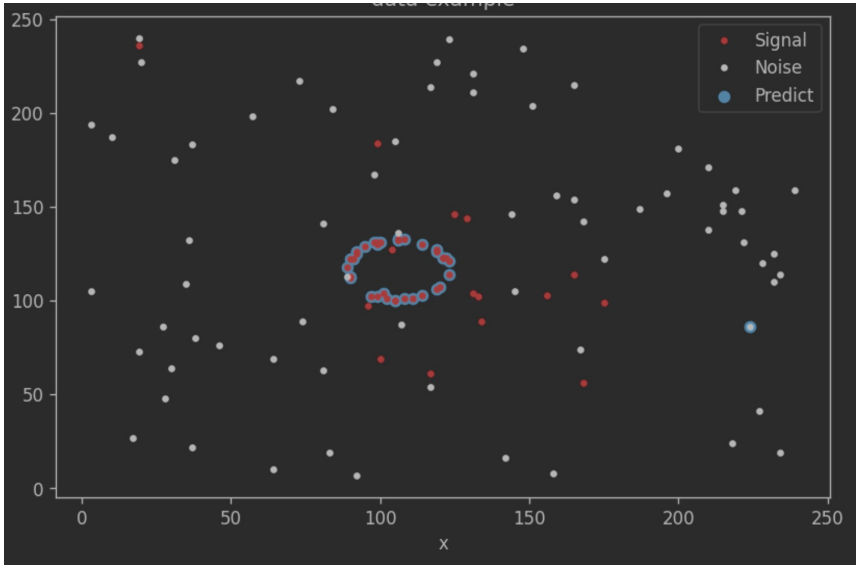
Преимущества:

- Метрика точности ~90% (среднее значение по тестовой выборке)
- Возможность параллельного запуска

Недостатки:

- Время работы алгоритма (асимптотическая сложность порядка $O(n^4)$)
- Необходим подбор параметров для настройки алгоритма





Возможные улучшения алгоритма

Основной проблемой алгоритма является полный перебор точек. Поэтому возникают оптимизации:

- Кэшировать подсчеты для одинаковых выбранных точек
- Уменьшить количество рассматриваемых точек

Способы уменьшение количества рассматриваемых точек, основываются на простой гипотезе: плотность событий в отрезке времени с эллипсом выше, чем когда его нет. Получаем следующие возможности для фильтрации данных:

- Статистическая фильтрация событий по времени.
- Сегментация эллипса в определенном интервале по времени. Обойти проверяемую область 3-х мерным кубом и посчитать плотность точек внутри. Если плотность точек выше среднего, то запускаем алгоритм поиска эллипса внутри данной области.

Выводы

Разработанный алгоритм позволяет решить поставленную задачу с точностью более 80%, но не удовлетворяет требованию на время обработки (не более 1 секунды).

Планы до защиты

- Добавить первичную фильтрацию данных
- Переписать решение с использованием технологий параллельного вычисления. В качестве языка программирования использовать C++
- Провести эксперименты с разным уровнями шума

Спасибо за внимание!