

## Алгоритм поиска и идентификации малых тел в больших системах

В. С. Осадченко  
Научный руководитель  
А. С. Байгашов

<sup>1</sup>Балтийский федеральный университет им. И. Канта, Калининград, Россия

<sup>2</sup>Название организации, Город, Страна

E-mail: [manevrik39@yandex.ru](mailto:manevrik39@yandex.ru)

Актуальность моей работы в том, что детекция малых тел востребована в разных отраслях: в астрономии, военном деле, биологии и медицине. С этой задачей столкнулась мой научный руководитель, изучая движение заряженных частиц в однородном магнитном поле: возникла проблема получения данных о траектории каждой частицы. Существующие инструменты поиска и идентификации малых тел узкоспециализированы, а где-то — отсутствуют. Цель проекта — создать универсальный программный инструмент, способный обрабатывать фото- и видеоматериалы, фиксирующие движение малых тел, и обеспечивать определение их координат. Задачи: изучить методы компьютерного зрения и машинного обучения, создать программу детектирования частиц, вычисления координат и их идентификации по ID, представить целостный продукт, решить проблему проекта на примере обнаружения потока движущихся светящихся частиц.

После постановки цели и задач члены нашей команды разделили их. Моё направление в проекте — это создание программы поиска и регистрации малых тел, а также визуализация траекторий движущихся малых тел. Свою программу я реализовал на языке Python с использованием библиотек OpenCV и NumPy. Программа работает в несколько этапов: предобработка кадра, поиск контуров частиц и их центров по яркости и контрасту с фоном, идентификация частиц и запись данных в таблицу.

Определение центра частицы вызвало сложности. Частицы имеют размеры, поэтому за координату тела мы принимаем координаты его центра масс. Его нахождение осуществляется специальными методами. Кроме того, свечение от частиц искажает их истинные размеры.

Следующий шаг — присвоение ID и их регистрация. Система «помнит» каждую частицу. Программа анализирует её скорость и направление, предсказывая, где объект окажется в следующую секунду. Поэтому, даже если частицы на миг перекрывают друг друга, уникальный номер надёжно «приклеивается» к объекту и ведёт его до самого конца пути. На основе полученных координат пользователь может попросить визуализировать траекторию каждой частицы, вписав её ID.

Но качество материала может быть не идеальным: съёмка часто идёт не в абсолютной темноте, и в кадр попадают яркие объекты, не относящиеся к частицам. Для решения этой проблемы разработан алгоритм поиска области интереса (ROI). Пользователю предлагается выбор: либо настроить область вручную, либо нейросеть (CNN) с архитектурой YOLO выделит её автоматически.

Объединив части проекта, мы получили программу, которая находит малые объекты на видео и фото, определяет координаты этих объектов. Пользователь получает обработанное видео с выделенными объектами и областью интереса, файл с их координатами и наглядную визуализацию траекторий каждого малого тела в виде анимированного графика matplotlib.

Выводы: в ходе работы создан программный инструмент для детекции, идентификации и трассировки малых тел. Программа вычисляет центры частиц весовым методом, присваивает каждой уникальный ID, отслеживает траекторию даже при перекрытиях и выделяет область интереса. Итоговая программа выдаёт обработанное видео, таблицу координат и анимированные траектории. Уникальность разработанного решения — в его универсальности: мой алгоритм успешно справился с детекцией частиц в магнитном поле и был опробован в области звёздного и микробиологического детектирования, где также показывал отличные результаты. В перспективе — создание интерфейса и дальнейшая оптимизация программы.

### **Возьмите на заметку:**

1) Алгоритм нахождения центра: разбиение на пиксели, присвоение каждому «веса» (**аналог силы тяжести**), вычисление координат. Реализовано два метода: вес = 1 для пикселей в контуре, либо вес = яркости (0–255).

2) От второго метода отказались из-за медленной работы на функциях Python вместо низкоуровневой OpenCV, что давало плохие результаты. OpenCV критически важен для быстрой обработки кадров и поиска контуров.