



Обнаружение метеоров с использованием алгоритма детектирования сегментов линий.

Д.С. Дроздов¹ , М.Ю. Ховричев² 

¹ СПбГУ, ²ИПА РАН

Поступила в редакцию 25 марта 2025 / Принята к публикации 10 апреля 2025

Аннотация

Программное обеспечение, разрабатываемое для оптического мониторинга метеорных явлений, должно позволять вести анализ потока кадров в реальном времени ввиду отсутствия ресурсов для хранения такого большого объема данных. В настоящее время общей частью большинства алгоритмов поиска метеоров, используемых для обработки данных современных метеорных сетей, является преобразование Хафа для детектирования линий на изображении. Данный метод применялся для задачи поиска метеоров с момента появления первых сетей как имеющий оптимальную скорость работы и эффективность детектирования. Обнаружение линий на изображениях и в видеопотоке не является исключительно астрономической задачей и относится к широкому классу проблем детектирования аномалий. В последние десятилетия в данной области произошел скачок, предоставив большое количество новых методов. Одним из таких является алгоритм детектирования сегментов линий, имеющий преимущество перед преобразованием Хафа в виде меньшей алгоритмической сложности. Алгоритм позволяет находить области с одинаковым направлением градиента, которые обрамляют линии на изображении.

В данной работе исследуются возможности детектирования метеоров с использованием метода детектирования сегментов линий. Для этого алгоритм был видоизменен с целью проведения параллельных вычислений на GPU. Так же предложен способ анализа областей, полученных алгоритмом и определения наличия на них метеора. Программа показывает высокий отклик на метеорные события с отношением сигнал/шум выше 3 и позволяет детектировать подобные события на данных с секундной экспозицией с камеры ZWO ASI290MM обсерватории Ассы-Тургень. Код программы представлен на [GitHub](#).

ключевые слова: метеоры, компьютерное зрение, детектирование и мониторинг метеорных явлений

Введение

Оптический мониторинг метеорных событий является значимой задачей в современной астрономии. В настоящий момент в метеорных сетях по всему миру используются различные подходы: от сети небольших камер (Vida и др., 2016), детектирующих яркие события, до поиска слабых метеоров на телескопах с большим полем зрения (Izmailov и др., 2024). При этом каждая такая установка за ночь наблюдений способна накопить гигабайты наблюдательных данных с несколькими десятками секунд пролета метеоров. Ограниченность ресурсов для хранения данных требует завершения обработки, хотя бы, к следующей ночи. Поэтому эволюция метеорных сетей происходила параллельно с развитием вычислительных мощностей. Так, с повсеместным распространением в конце XX века персональных компьютеров появилась возможность обрабатывать данные в реальном времени, что привело к появлению таких программ, как Meteorscan (P. Gural, 1999). В 2010-х годах большое распространение получили мини компьютеры типа Raspberry Pi,

*e-mail:paganelle7@mail.ru

по мощности сравнимые с машинами начала века, но позволяющие размещать камеры в любых местах и обрабатывать данные на месте, что так же расширило возможности метеорных сетей.

Теперь обратимся к методам детектирования метеоров. В работе Peter Gural, 2008 перечисляются пять основных алгоритмов, из которых реально применяются три. Наибольшую точность определения даёт применение согласованного фильтра, но его применение к двухсекундному видео с разрешением в четверть мегапикселя требует часов работы, поэтому данный способ используется только для подтверждения наличия метеорного трека, уже найденного другим методом (P. Gural и др., 2022). Высокую скорость показывает алгоритм кластеризации пикселей, но он плохо отслеживает линейные структуры. Основным же методом становится преобразование Хафа (Duda и Hart, 1972), которое хорошо выделяет линии на изображении и имеет высокую скорость работы. Данный список методов не претерпел изменений за последнее время и в работе P. Gural и др., 2022 встречается в том же виде.

Рассмотрим подробнее методы детектирования в недавно запущенных метеорных сетях. Croatian Meteor Network (Vida и др., 2016) реализована на серии небольших камер, в качестве алгоритма обработки используется преобразование Хафа. Программа, обрабатывающая данные с камер GWAC (Chen и др., 2023) так же основана на использовании этой операции. Для применения на Raspberry Pi (P. Gural, 2016) была предложена вариация метода использованного для сети AIM-IT в виде кластеризации пикселей выше фона, но данный метод имеет низкую эффективность при поиске тонких и длинных метеоров и хорошо показывает себя только в связке с преобразованием Хафа (P. Gural и др., 2022). Таким образом, современные алгоритмы обычно включают в себя две части — детектирование метеоров и их подтверждение, причем преобразование Хафа используется в разных алгоритмах для разных целей.

Необходимо упомянуть так же нейронные сети, расцвет которых произошел в последние годы. В частности, массово начали применяться сверточные нейросети в задаче поиска аномалий на изображениях. Их использование так же требует больших вычислительных мощностей и не рассматривается как быстрый алгоритм. Например, сверточная нейросеть на базе ResNet-34 (Peña-Asensio и др., 2023) применяется именно для отбрасывания ложных срабатываний, то есть выполнения работы, ранее делавшейся человеком вручную. В данном случае обрабатываются изображения с 30-секундной экспозицией, сжатые до размера 400x400 пикселе. Авторы прямо указывают, что применение нейросетей пригодно только для изображений низкого разрешения.

Если обратиться к современным методам нахождения линий на изображении, то сейчас для этого, наравне с преобразованием Хафа, используются алгоритмы детектирования границ и обнаружения сегментов линий (Grompone von Gioi и др., 2012). Так как метеор образует, зачастую очень тусклый, отрезок линии, второй метод является подходящим для его обнаружения. Данный метод является развитием идеи алгоритма детектирования прямых отрезков (Burns, Hanson и Riseman, 1986). Основным преимуществом является его линейная, в отличие от преобразования Хафа, алгоритмическая сложность, что в ряде случаев повышает время работы на порядки (Grompone von Gioi и др., 2012). В данной работе предлагается использование этого алгоритма в качестве основного средства детектирования, поставляющего данные для дальнейшей проверки.

Предложенная программа поиска метеоров, как и большинство существующих средств детектирования состоит из трех основных частей: 1) Очистка изображения от шума (Предпроцессинг), 2) Детектирование (здесь вместо преобразования Хафа используется алгоритм детектирования сегментов линий), 3) Постпроцессинг — соединение найденных сегментов в метеоры и отбрасывание лишних сегментов. Все эти шаги будут подробнее разобраны далее в разделе 2.

1 Алгоритм детектирования сегментов линий

1.1 Оригинальный алгоритм

Для поиска прямых отрезков данный метод использует градиенты, а точнее, линии уровня на изображении, так как вдоль прямой градиент интенсивности направлен перпендикулярно, а ли-

нии уровня параллельно прямой. Алгоритм определяет линии уровня на изображении, а затем группирует пиксели со схожими значениями, получая группу, протянувшуюся вдоль линии на изображении (Рис. 1).

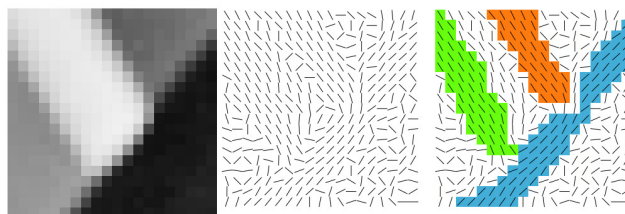


Рис. 1: Пример определения линий уровня на изображении и их группировки

Первый шаг алгоритма — определение направлений линий уровня на изображении. Для этого вычисляется градиент с использованием маски 2×2 (Grompone von Gioi и др., 2012).

$$g_x(x, y) = \frac{1}{2}(i(x+1, y) + i(x+1, y+1) - i(x, y) - i(x, y+1))$$

$$g_y(x, y) = \frac{1}{2}(i(x, y+1) + i(x+1, y+1) - i(x, y) - i(x+1, y))$$

Направление линии уровня далее может быть вычислено как

$$\arctan \frac{g_x(x, y)}{-g_y(x, y)}$$

После этого решается классическая задача программирования — кластеризация. Базовая часть решения данной задачи требует одного прохода по всем пикселям изображения. Реализуется итеративный подсчет, на каждой итерации которого выбирается случайный пиксель, становящийся “лидером”. Затем проверяется совместимость со всеми соседями, в случае совпадения направлений линий уровня, сосед присоединяется к группе. Каждый присоединенный пиксель добавляет новых соседей. Просмотр продолжается до тех пор, пока все соседи группы не будут иметь сильно отличающееся направление линий уровня. Направление группы определяется как среднее от направлений всех ее членов и пересчитывается после каждого дополнения. При присоединении проводится проверка значения скалярного произведения линии уровня соседа и группы на превышение некоторого порога α .

Алгоритм детектирования сегментов линий обнаруживает границы яркости на изображении. Для обнаружения метеоров необходимо из найденных границ “сформировать” метеоры. Базовым условием наличия изображения метеора, является обнаружение двух, близко расположенных групп со схожим количеством элементов и противоположно направленными градиентами. Сопоставление групп производится на этапе постпроцессинга.

Линейное время работы алгоритма делает его подходящим для задачи обработки видеопотока высокого разрешения в реальном времени. Тем не менее, для этого есть серьезное препятствие: так как группы разрастаются непредсказуемым образом, параллельная обработка разных пикселей (разрастание групп из разных точек) невозможна, что ограничивает использование данного алгоритма на GPU, то есть на основном средстве обработки. Это становится серьезным препятствием для его использования. Далее мы модернизируем поиск групп для возможности применения на GPU.

1.2 Адаптация алгоритма

Основная цель адаптации — сделать алгоритм доступным для параллельных вычислений на GPU. Для этого кластеризация проходит в два прохода вдоль каждой из осей при которой появляется два массива групп с различными направлениями. Алгоритм реализован на языке Python с использованием библиотеки PyTorch, встроенные функции которой оптимизированы для работы с

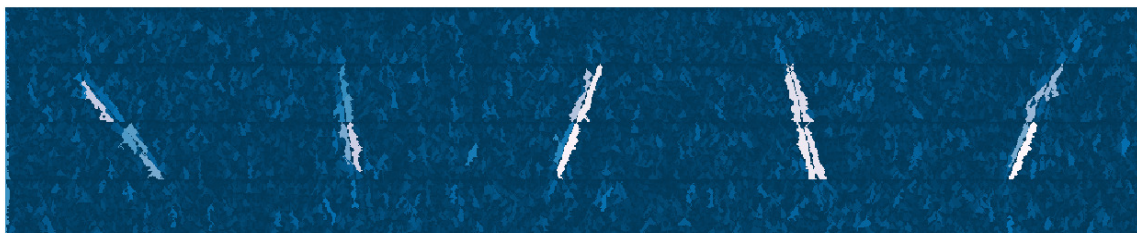


Рис. 2: Пример работы алгоритма на тестовых изображениях. Видны найденные алгоритмом группы. Яркость группы пропорциональна количеству элементов в ней с учетом порога в 100 элементов.

тензорами, что сильно повышает быстродействие. Итерирование производится следующим образом: осуществляется проход по строкам массива пикселей. В начале каждого шага уже имеется строка предыдущего шага, чьи пиксели уже являются частями групп. На первом шаге каждый пиксель становится группой из одного элемента. После чего производится сопоставление с соседними пикселями из следующей строки. Каждый элемент из новой строки проверит схожесть своего градиента с тремя, или меньше, группами, представленными тремя соседними пикселями, из предыдущей строки. Пиксель выбирает группу с наибольшей схожестью, и если угол отклонения меньше некоторого порога α , производится присоединение. После этого происходит пересчет количества элементов групп и значений направлений. Таким образом группа может расширяться от лидера в форме треугольника и иметь угол отклонения от оси, вдоль которой производится проход $(-\pi/4, \pi/4)$. Так как в линии оказывается несколько элементов одной группы, необходимо определить уникальные группы, но, чтобы не усложнять алгоритм, поиск производится окном некоторого размера n . Этот размер определяет максимальную ширину группы вдоль строк. Увеличение прекращается, когда в новой строке ни один близлежащий градиент не совпал с градиентом группы. Для увеличения быстродействия изображение переформатируется в форму с меньшим числом строк большей длины, это делается для уменьшения количества итераций, так как функции PyTorch дают сильный выигрыш по скорости.

Итогом работы алгоритма является массив лидеров групп, массив, хранящий координаты лидера группы для каждого пикселя и массив с координатами конца группы. Сами группы так же сохраняются, но для дальнейшей обработки не используются. На получившейся карте групп каждый метеор оказывается “обрамлен” с разных сторон группами с противоположными градиентами (Рис. 2).

При данном способе кластеризации групп работа алгоритма управляется следующими параметрами: максимальным углом отклонения от направления группы для присоединения к ней и максимальной шириной группы по строкам. Варьирование обоих параметров может давать различные отклики на изображении метеоров. Так, при маленькой ширине линий и “широкой” функции рассеяния точки (относительно большой величине FWHM — полной ширине на половине максимума изображения) яркий метеор будет создавать вокруг себя множество параллельно идущих групп. На реальных, зашумленных, данных наилучшие результаты показывают значения $\alpha = 30^\circ$ и $n = 15$.

2 Поиск метеоров

Алгоритм детектирования сегментов линий позволяет получить расположение всех протяженных границ яркости на изображении. Для обнаружения метеоров необходимо найденные границы сопоставить. Ниже представляется последовательность шагов для обнаружения метеора на изображении:

1. Алгоритм детектирования сегментов линий применяется к двумерному изображению. Поэтому сначала необходимо из ряда кадров это двумерное изображение получить. Здесь все

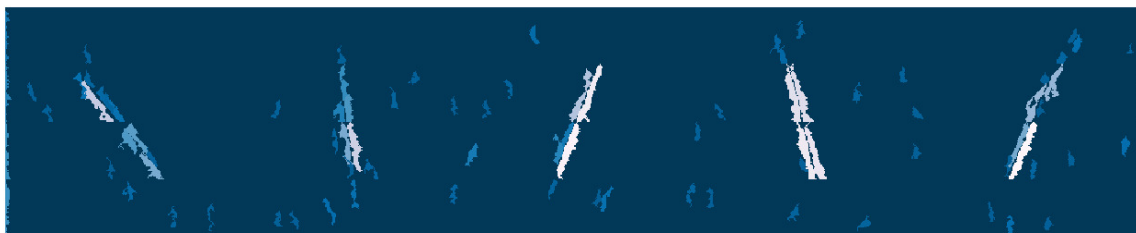


Рис. 3: Группы с количеством членов больше 60

зависит от экспозиции (времени накопления) кадров. Так как характерное время пролета метеора составляет от сотых долей до секунды и более, необходимо просуммировать подряд идущие кадры для получения сравнимого временного промежутка. Затем производится предобработка полученного изображения. Для данного алгоритма она будет отличаться от привычных в астрономии, так как необходимо не только очистить источники от шума, сколько увеличить их влияние на окружающее пространство. Это позволяет сильно “сгладить” изображение. Для этого первым шагом обработки является применение максимального объединения. Пикселю присваивается максимальное значение в некоторой окрестности. Затем к изображению применяется медианный фильтр. Данный фильтр каждый пиксель заменяет на значение медианы пикселей в некоторой окрестности, то есть показывает уровни интенсивности, которые анализируются в дальнейшем. После появления ярко выраженных границ линий уровня производится сглаживание с помощью свертки небольшим ядром. Применение последнего этапа усиливает “связность” групп.

- Вторым этапом является, непосредственно, использование алгоритма детектирования сегментов линий. Производится два прохода вдоль осей изображения, которые обнаруживают сегменты линий с разными наклонами. Этот алгоритм описан во второй части данной работы. Для каждой группы на следующий шаг программы передаётся значение градиента, положения начала и конца группы, количество элементов.
- Последний этап — это выделение из сегментов линий метеоров. Для каждой части групп он проводится отдельно и представляет из себя ряд последовательных фильтраций. Первым делом отбрасываются группы с количеством членов меньше определенного порога (пример такого отсева для тестовых изображений представлен на Рис 3.), то есть группы не являющиеся границей значительной области с интенсивностью выше фона.

Затем все оставшиеся группы разбиваются на “правые” и “левые”. Данное разбиение производится согласно их градиенту, так как для одного прохода все границы имеют границы от $\pi/4$ до $3\pi/4$, а градиент перпендикулярен границе, то он может иметь значения углов либо $(-\pi/4, \pi/4)$, либо $(3\pi/4, 5\pi/4)$. После чего проводится попарная проверка левых и правых групп. Группы, ограничивающие метеор, должны иметь скалярное произведение своих градиентов, близкое к -1, быть параллельными, находиться на близком расстоянии друг к другу и по нужную сторону (так, чтобы градиенты смотрели друг на друга, иначе будет выделяться наиболее “темная” линия на изображении).

Итогом работы алгоритма является список левых и правых “палочек”, нашедших себе пару. Причем их количество может различаться, например, когда под одну сторону метеора образовалась одна длинная группа, а по другую две меньшей длины.

3 Тестирование

Для проверки работы алгоритма был написан python-скрипт, имитирующий пролет метеора. Как и у Р. Gural и др., 2022, генерируется линия не проходящая через разные кадры, затем она

размывается посредством свертки с двумерной гауссовой функцией рассеяния точки. В качестве кривой блеска взята парабола (Campbell, Hawkes и Babcock, 1999). Пример генерации метеоров представлен на Рис. 4.

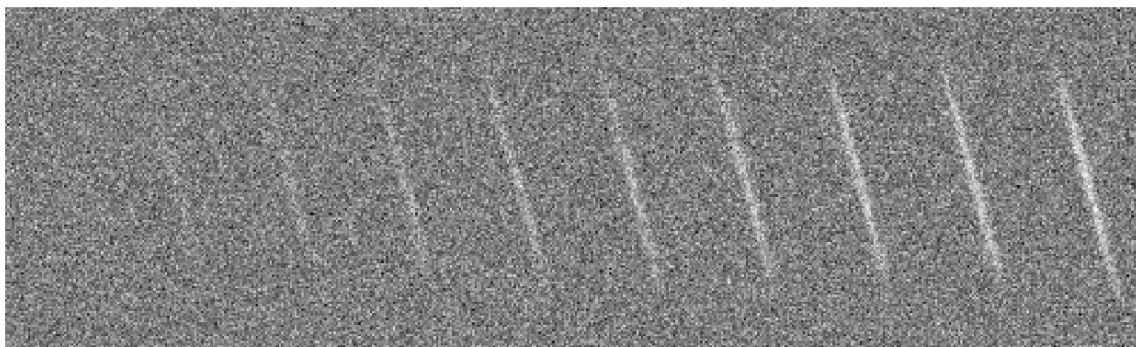


Рис. 4: Сгенерированные метеоры с отношением сигнал/шум от 1 до 10

Тестирование проводилось на изображении с набором метеоров, которое подавалось на вход алгоритма. Выводом программы детектирования является набор парных сегментов. Во время проверки метеор считался обнаруженным, если в некотором, окружающем его, прямоугольнике присутствует обнаруженный парный сегмент, как показано на рис. 5. Так же отслеживалось об-

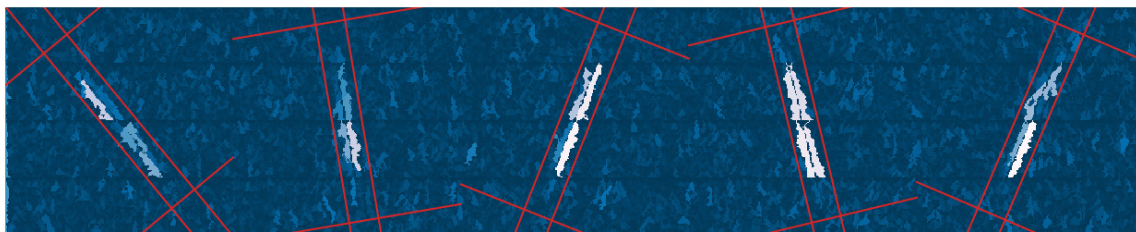


Рис. 5: Критерий проверки присутствия метеора на изображении

щее количество откликов алгоритма и количество ложных откликов. Тестирование проводилось для произвольных направлений и для отношения сигнал/шум от 1 до 10, для каждого значения отношения сигнал/шум генерировалось 200 метеоров. Результаты представлены на Рис. 6. Тестирование на искусственных метеорах показало почти 100% вероятность отклика на метеор при значениях отношения сигнал/шум выше 3-х. Количество ложно положительных срабатываний оставалось на одном уровне независимо от наличия метеоров. Количество найденных метеоров выросло быстрее общего количества срабатываний, так как у более слабых метеоров сегменты могут обнаруживаться не по всей длине. Так же с возрастанием значения сигнал/шум (при значении больше 9) наблюдается падение числа подходящих сегментов за счет увеличения их протяженности: с уменьшением доли шума метеор сегменты становятся более цельными.

4 Результаты

Работа программы была проверена на данных, полученных на стационарной камере пулковского 40-см телескопа (Khovritchev и др., 2022) в обсерватории Ассы-Тургень, принадлежащей астрофизическому институту им. В. Г. Фесенкова (АФИФ) Национального Центра Космических Исследований и Технологий при Аэрокосмическом Комитете Республики Казахстан. Данная обсерватория расположена на высокогорном плато и характеризуется высокими астроклиматическими показателями. Для наблюдений использовалась миниатюрная планетная камера ZWO ASI 290MM с объективом HD 3mm 2.8-12 mm (1/2. 7"1:1.4). Диаметр объектива составляет 12 мм, рабочее поле 27 на 15 градусов.

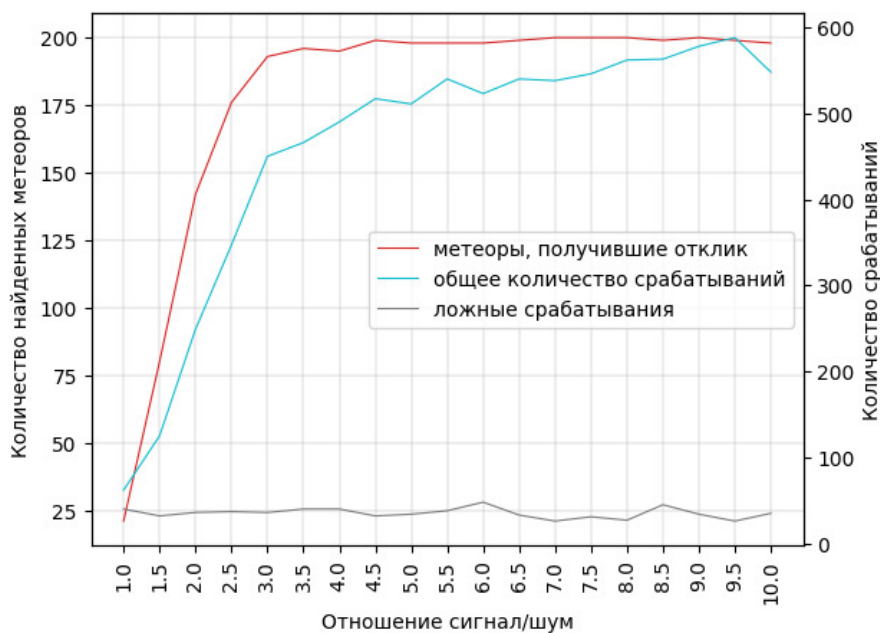


Рис. 6: Доля метеоров, обнаруженных при разных значениях отношения сигнал/шум

В данном случае обрабатывалась последовательность кадров с экспозицией в 1 секунду. Так как поиск имеет малое, но постоянное число ложных срабатываний, для поиска метеоров были ужесточены пороги фильтрации. Это позволило определить метеоры с показателем сигнал/шум вплоть до 3 (Рис. 7). Примеры обнаруженных метеоров представлены на Рис. 8.

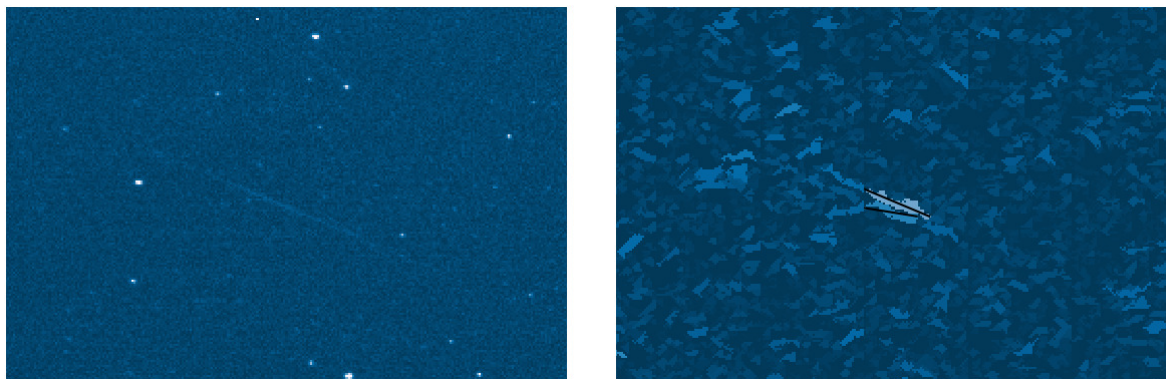


Рис. 7: Метеор с отношением сигнал/шум=3

Метод детектирования сегментов линий даёт хороший отклик на метеорные события, но сложность представляет их дальнейшая обработка (сопоставление парных сегментов). Так, одним из критериев наличия метеора является близость друг к другу сегментов с противоположными градиентами. Но отбрасывание далеких границ ухудшает обнаружение ярких метеоров, имеющих большую ширину трека. Так, на Рис. 4 представлены два последовательных кадра с ярким метеором, который дал слишком широкие группы градиентов, оказавшиеся на большом расстоянии друг от друга и принятые алгоритмом за слишком далекие, тем не менее, метеор все же был обнаружен по тонкому следу на следующем кадре.

Алгоритм детектирования сегментов линий может быть использован для детектирования, в том числе слабых, метеоров в качестве основного способа детектирования, но требует улучшения способа классификации найденных сегментов. Спецификой данного алгоритма является определение метеоров независимо от их интенсивности, что с одной стороны, позволяет обнаруживать

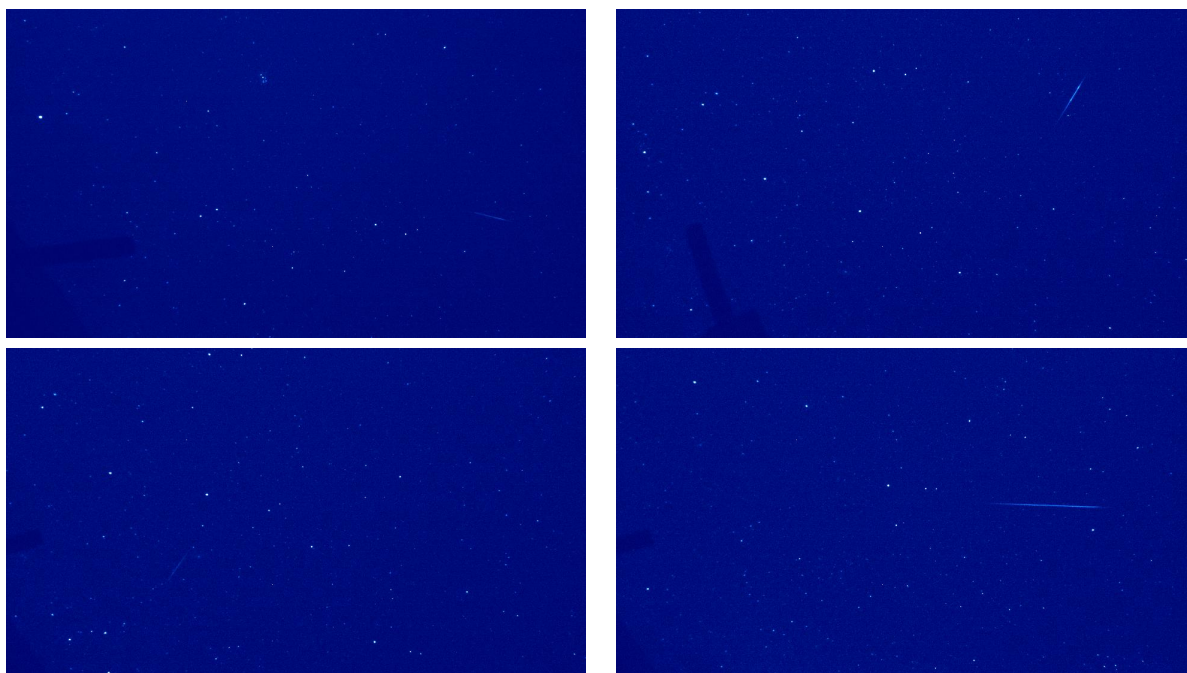


Рис. 8: Примеры найденных метеоров

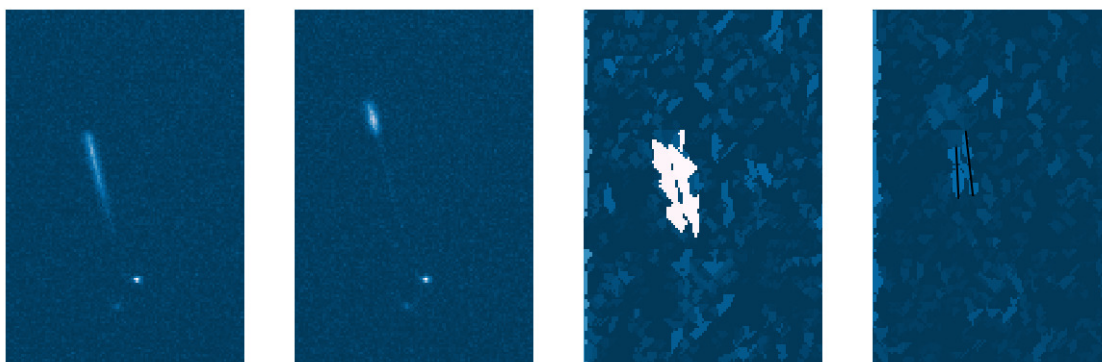


Рис. 9: Яркий метеор, найденный по угасшему треку.

слабые события, но с другой, даёт реакции на небольшие линии в шуме, требующие отдельной фильтрации.

Как отмечалось ранее, в работе Izmailov и др., 2024 представлен метод решения задачи детектирования метеорных явлений на основе обнаружения линий из локальных максимумов посредством преобразования Хафа. Этот метод особенно эффективен для сравнительно небольших рабочих полей телескопов с фокусным расстоянием порядка 0.5 метра, когда речь идет о наблюдении слабых и коротких метеоров до 12 звездной величины и слабее. Алгоритм, представленный в нашей работе, лучше подходит для короткофокусных камер с рабочими полями в сотни квадратных градусов. Мы ожидаем, что использование разных методов детектирования на одном наборе данных должно повысить эффективность выявления метеоров. Это является темой для дальнейших исследований и разработок в данном направлении.

Финансирование

Настоящее исследование выполнено за счет гранта Российского научного фонда и Санкт-Петербургского научного фонда № 24-22-20044, <https://rscf.ru/project/24-22-20044/>.

Благодарности

Авторы выражают свою признательность коллегам из Астрофизического института им. В. Г. Фесенкова (АФИФ) Национального Центра Космических Исследований и Технологий при Аэро-космическом Комитете Республики Казахстан за помощь в ходе проведения наблюдений.

Список литературы

- Vida, D., D. Zubović, D. Šegon, P. Gural и R. Cupec (2016). [Open-source meteor detection software for low-cost single-board computers](#). В: *International Meteor Conference Egmond, the Netherlands, 2-5 June 2016*. Под ред. A. Roggemans и P. Roggemans, с. 307.
- Izmailov, I. и др. (2024). [Detection of meteor event images using the SSA400 telescope at the Assy-Turgen Observatory](#). *Publications of the Pulkovo Observatory* 234, с. 22–35.
- Gural, P. (1999). [MeteorScan Documentation and User's Guide](#).
- Gural, Peter (2008). [Algorithms and Software for Meteor Detection](#). *Earth, Moon, and Planets* 102, с. 269–275.
- Gural, P., T. Mills, M. Mazur и P. Brown (2022). [Development of a very faint meteor detection system based on an EMCCD sensor and matched filter processing](#). *Experimental Astronomy* 53.3, с. 1085–1126.
- Duda, Richard O и Peter E Hart (1972). [Use of the Hough transformation to detect lines and curves in pictures](#). *Communications of the ACM* 15.1, с. 11–15.
- Chen, Yicong и др. (2023). [A Meteor Detection Algorithm for GWAC System](#). *Universe* 9.11, с. 468.
- Gural, P. (2016). [A fast meteor detection algorithm](#). В: *International Meteor Conference Egmond, the Netherlands, 2-5 June 2016*. Под ред. A. Roggemans и P. Roggemans, с. 96.
- Peña-Asensio, Eloy, Josep M. Trigo-Rodríguez, Pau Grèbol-Tomás, David Regordosa-Avellana и Albert Rimola (2023). [Deep machine learning for meteor monitoring: Advances with transfer learning and gradient-weighted class activation mapping](#). *Planet. Space Sci.* 238, с. 105802.
- Grompone von Gioi, Rafael, Jérémie Jakubowicz, Jean-Michel Morel и Gregory Randall (2012). [LSD: a Line Segment Detector](#). *Image Processing On Line* 2. <https://doi.org/10.5201/ipol.2012.gjmr-lsd>, с. 35–55.
- Burns, J. Brian, Allen R. Hanson и Edward M. Riseman (1986). [Extracting Straight Lines](#). *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8.4, с. 425–455.
- Campbell, M., R. Hawkes и O. Babcock (1999). [Light curves of shower meteors: implications for physical structure](#). В: *Meteoroids 1998*. Под ред. W. J. Baggaley и V. Porubcan, с. 363.
- Khovritchev, M. Yu., A. A. Arkharov, V. Yu. Kim и I. S. Izmailov (2022). [The prospects for the implementation of the Pulkovo Observatory observational program at the Assy-Turgen observatory](#). *Publications of the Pulkovo Observatory* 227, с. 58–76.

Meteor detection using the line segment detection algorithm.

D.S. Drozdov¹, M.Yu. Khovrichiev²

¹ Saint Petersburg State University , ²IAA RAS

Received 25 March 2025 / Accepted 10 April 2025

Abstract

Currently, a common part of most meteor search algorithms used to process data from modern meteor networks is the Hough transform to detect lines in an image. This method has been used to search for meteors since the first networks appeared, as it provides optimal detection speed and efficiency. Detecting lines in images and in a video stream is not an exclusively astronomical task and belongs to a wide class of anomaly detection tasks. In recent decades, there has been a significant leap in this field, which has led to the emergence of a large number of new methods. One of them is the line segment detection method which has an advantage over the Hough transform in the form of lower algorithmic complexity. The algorithm allows you to find areas with the same gradient direction that frame the lines in the image.

In this paper, the possibilities of detecting meteors using the line segment detection method are investigated. To do this, the algorithm was modified in order to perform parallel calculations on the gpu. A method is also proposed for analyzing the areas obtained by the algorithm and determining the presence of a meteor on them. The program shows a high response to meteor events with a signal-to-noise ratio above 3 and allows you to detect such events on data with a second exposure from the ZWO ASI290MM camera of the Assi-Turgen Observatory. The program code is presented on [GitHub](#).

key words: meteor event; image detection; computer vision