САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Математико-Механический факультет
Кафедра Системного Программирования

Бакаев Евгений Владимирович

Алгоритм детекции пятен на стереокамере

Отчёт по учебной практике

Научный руководитель:
ст. преп. Смирнов М. Н.

Консультант:
Руководитель проекта, Системы Компьютерного Зрения,
Богданюк И. А.

Санкт-Петербург
2022 г.

Содержание

1 Введение

Компьютерное зрение [1] направлено на создание автономных систем, способных выполнять задачи зрительной системы человека. Алгоритмы компьютерного зрения в качестве основного источника данных используют набор изображений. Артефакты на изображениях могут приводить к тому, что получаемые данные будут неполные или неточные. Таким образом, данных может быть недостаточно для принятия решения или принятое решение может быть неверным.

Полнота и точность данных для автономных систем компьютерного зрения реального времени является необходимым условием работы системы. Автономные системы должны уметь анализировать качество получаемых данных с камер и, в случае их недостоверности, уведомлять, что система в таких условиях работать не может, так как работа может нанести вред окружающему миру или самой себе.

Одной из причин появления артефактов на изображениях могут быть неблагоприятные погодные условия. Капли дождя, снег и грязь могут попадать на линзы камеры и приводить к появлению пятен на изображениях.

Данная работа содержит описание алгоритма детекции пятен на стереопаре, который может быть использован в системах компьютерного зрения реального времени, например, при создании автономных транспортных систем. Предложенный алгоритм был реализован и встроен в систему машинного зрения, работающую с частотой 10 кадров в секунду, являющуюся частью автономной транспортной системы, выполняющей работу на гольф-поле.

2 Постановка задачи

Автономная транспортная система для корректной работы при любых условиях должна содержать алгоритм, который занимается самодиагностикой и оценкой качества данных, получаемых с устройств. Детекция пятен может быть его частью.

Цель данной работы – разработать алгоритм детекции пятен на линзах стереокамеры, позволяющий в реальном времени уведомлять о загрязнении линзы при работе на Jetson AGX Xavier, причем на каждую эпоху алгоритм должен занимать не более 5 миллисекунд. Для достижения цели были поставлены следующие задачи:

- Выполнить обзор предметной области
- Ознакомиться с существующими алгоритмами детекции
- Отснять датасет с пятнами на линзе камеры
- Реализовать алгоритм детекции пятен
- Оценить скорость и качество работы алгоритма

3 Обзор предметной области

3.1 Стереокамера

Стереокамера – камера с несколькими линзами, симулирующая бинокулярное зрение. Синхронные кадры с левой и правой камеры позволяют вычислить карту глубины, построить облако точек изображенной сцены.

Рис. 1: FRAMOS Depth Camera D455e

3.2 Интенсивность пикселя (в оттенках серого)

Интенсивность пикселя – это значение пикселя. Один пиксель интенсивнее другого, если значение первого больше значения второго.

3.3 Градиент изображения

Градиент изображения – это направленное изменение интенсивности изображения. В каждом пикселе входного изображения градиент измеряет изменение интенсивности пикселя в заданном направлении. Оценивая направление и то, как быстро происходит изменения интенсивности, можно находить края объектов [2] (оператор Кэнни, оператор Собеля).

3.4 Размытие

Размытие – это способ снижение шума и детализации области изображения путем усреднения быстрых изменений интенсивности в ней.

3.5 Библиотека OpenCV

OpenCV – кроссплатформенная библиотека компьютерного зрения с открытым исходным кодом, доступная на C++, Python и Java.

OpenCV позволяет работать с изображениями, содержит реализации известных алгоритмов компьютерного зрения и их оптимизации на GPU, используя CUDA.

4 Обзор аналогов

На данный момент существует множество работ по детекции и сегментации капель. Многие из них находят капли на лобовом стекле автомобиля [3, 4], а не на самой камере.

В работах, использующих нейронные сети для детекции и удаления капель, основным недостатком является время принятия решения, не позволяющее работать в режиме реального времени на Jetson AGX Xavier [5, 6].

Исследования, основанные на предположении, что область пятна на изображении размыта, используют последовательность изображений для анализа [7, 8], вычисляют усредненную градиентную карту. Это приводит к тому, что решение о наличии артефактов выдается не каждый кадр, а каждые N кадров, но позволяет не нагружать CPU/GPU на каждой эпохе. При оценки точности детекции в этих работах используют датасеты в чертах города, но в ходе данной работы выяснилось, что при движении в чистом поле точность детекции падает.

Таким образом, было принято решение реализовать новый алгоритм на основе градиентных карт, как и в решениях [7, 8], так как существующие решения детекции пятен на основе нейронных сетей не позволяют работать со скоростью 5 миллисекунд на эпоху или решают другую задачу (капли на лобовом стекле и пятна на линзе камеры имеют разные формы, плотность и размер).

5 Ход работы

5.1 Датасет

В открытом доступе нет датасетов, использующих стереокамеру в неблагоприятных погодных условиях, поэтому датасет нужно было создать. Первый способ – это искусственно сгенерировать пятна на датасете без пятен [8], а второй – отснять на улице.

Во время снегопада был отснят датасет на FRAMOS Depth Camera D455e. Датасет состоит из 3964 стереопар и карт глубины с разрешением 1280x720.

Рис. 2: Левое и правое изображение. 1986 эпоха.

Рис. 3: Появление нового пятна на левом изображении. 1987 эпоха.

5.2 Реализация алгоритма детекции пятен

5.2.1 Вычисление усредненной карты градиента последовательности изображений для каждой камеры

Пусть N длина рассматриваемой последовательности изображений, то есть каждые N кадров алгоритм будет выдавать результат о наличии или об отсутствии пятен.

Тогда $G_{avg} = \frac{\sqrt{\sum_{i=1}^{N}((G_x^i)^2+(G_y^i)^2)}}{N}$, где (G_x^i) и (G_y^i) это градиентные карты изображения с номером i по горизонтали и вертикали соответственно (применение оператора Собеля).

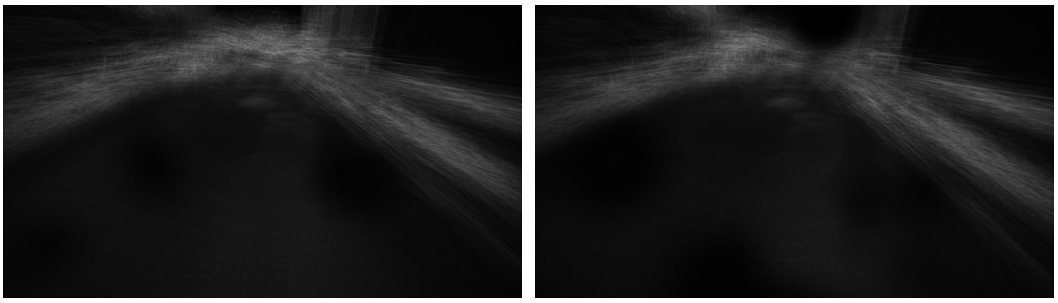

Рис. 4: Левая и правая усредненные карты градиента для последовательности изображения 2300-2399.

Основным предположением является то, что пятно остается в том же месте изображения в течение последовательности кадров. Выбор значения N зависит от качества изображений, скорости движения автономной системы, и особенностей ландшафта (движение в чертах города или в полях). В работе [8] рассматривается $N = 10$, но для детекции пятен в нижней половине изображения (на асфальте) размера 100x100 пикселей, этого значения было недостаточно, так как асфальт имеет близкий к нулю градиент, то есть не содержит четких границ.

5.2.2 Бинаризация градиентной карты, создание масок

Отсутствие границ внутри пятен приводит к тому, что область внутри них является наименее интенсивной. Наименее интенсивными также являются объекты, не содержащие четких границ, например, белая или черная стена.

Перед бинаризацией необходимо применить фильтр Гаусса с ядром K для удаления шума на градиентной карте. При бинаризации градиентной карты с пороговым значением t получается бинарная маска, у которой пиксель с координатами (x, y) имеет значение 255 только в том случае, если интенсивность пикселя (x, y) градиентной карты меньше t.

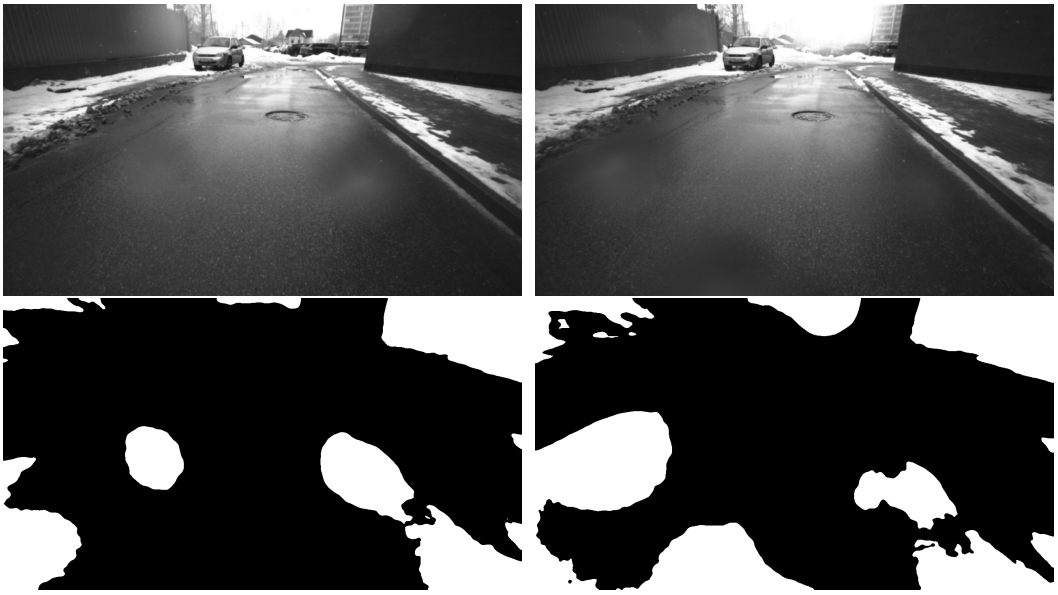

Рис. 5: Левое и правое изображения 2350 эпохи. Маски для последовательности изображений 2300-2399.

Правый верхний угол каждой маски на данном этапе определяется как пятно, но это стена здания. На следующем этапе алгоритма будет показано, как исключать такие случаи.

5.2.3 Фильтрация маски

Стоит отметить, что до этого момента алгоритм распараллеливается очевидным образом: для каждой камеры создается свой поток, который выполняет нужные вычисления. Так же вычисление градиентной карты на каждой эпохе: по горизонтали и вертикали можно считать независимо друг от друга.

Алгоритм фильтрации основан на предположении о том, что пятна от снега или дождя по площади сравнимы с площадью эллипса.

Для каждой белой области на бинарной маске находится ее контур P_i [9]. Каждый контур вписывается в прямоугольник со сторонами W, H.

Тогда площадь вписанного в прямоугольник еллипса $S_{ell} = \frac{1}{4}\pi WH$. Далее, в случае, когда контур не лежит на границе изображения будет использоваться два критерия t_{ecc} и t_s для оценки эксцентриситета эллипса и его площади соответственно, иначе только t_s.

Эксцентриситет равен нулю, если эллипс является окружностью, тогда чем ближе друг к другу значения W и H, тем ближе эллипс к окружности, то есть порог t_{ecc} говорит о том, что пятном будут называться только те контуры, у которых $\frac{min(W,H)}{max(W,H)} > t_{ecc}$. Проще говоря, пятно от капли не может быть непропорционально узким или широким.

При оценки площади идея та же. Площадь контура S_P должна быть сравнима с площадью вписанного эллипса, то есть $\frac{min(S_P,S_{ell})}{max(S_P,S_{ell})} > t_s$. При попадании пятна на границу изображения значение t_s увеличивается на некоторый ε и не используется t_{ecc}, потому что в кадр попадает только видимая часть пятна и она может иметь любую форму.

5.3 Результат алгоритма

Преимущество стереокамеры перед монокамерой заключается в том, что левая и правая камера видят одну и ту же сцену, но попадание грязи хотя бы на одну камеру может приводить к построению некорректной карты глубины, что понижает точность алгоритмов, использующих бинокулярное зрение, например, построение облака точек, детекция препятствий и определение расстояния до них.

Рис. 6: Облако по стереопаре и правое изображение (номера автомобиля закрашены для данного отчёта). Пятна приводят к появлению несуществующих точек в воздухе

Воспользуемся этим преимуществом следующим образом: применим XOR левой и правой масок, полученную маску будем называть XOR-маской. Делается это с той целью, чтобы затемненные области изображения не выделялись как пятна, например, при движении в тени дерева. Таким образом, в XOR-маске белыми областями будут только те пиксели, которые были белыми ровно на одной маске. XOR-маска для каждого объекта на изображении будет содержать полоску шириной в несколько пикселей, так как одна и та же точка сцены на левом и правом изображении имеет разную координату (диспаритет), но даная полоска будет отбрасываться фильтрацией на эллипсы. $dirtArea$ – это количество белых пикселей на XOR-маске.

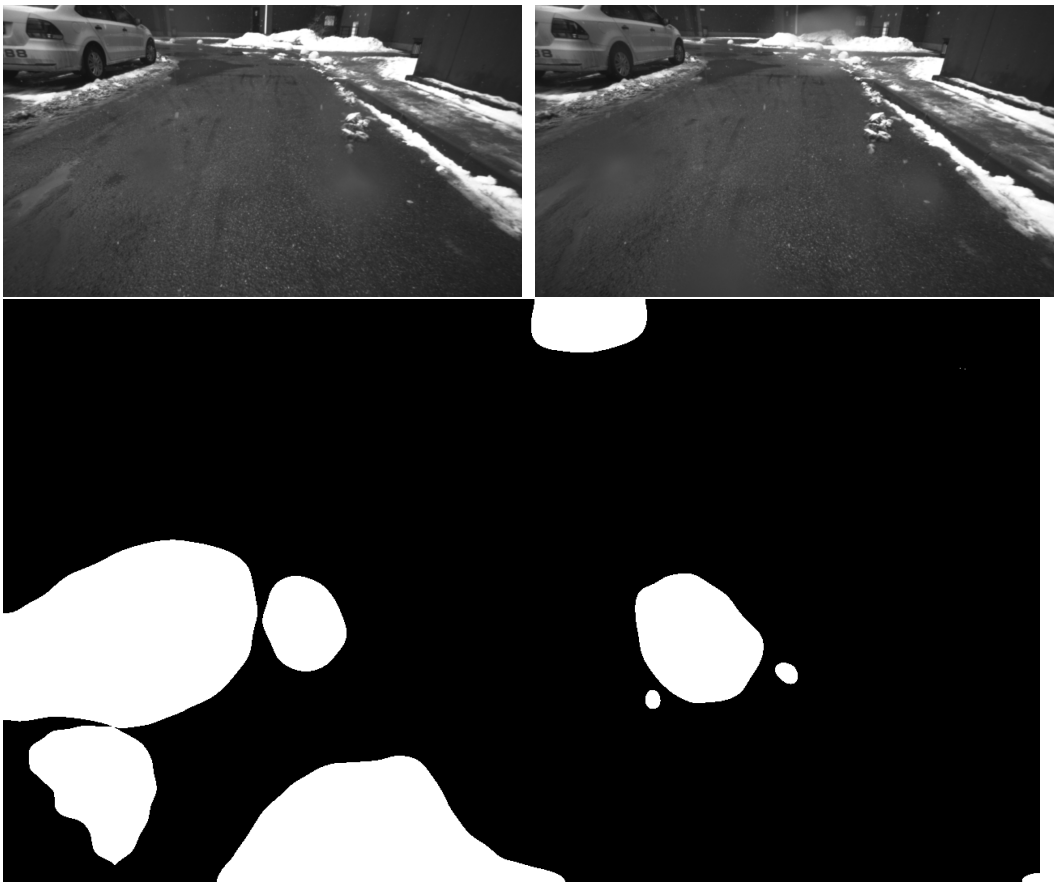

Рис. 7: Левое и правое изображение 2582 эпохи. XOR-маска для последовательности изображений 2500-2599.

Результатом алгоритма является число от 0 до 1, определяющее чистоту изображения. Для этого устанавливается два порога: $minDirtArea$ и $maxDirtArea$, а результат вычисляется как $\frac{maxDirtArea-dirtArea}{maxDirtArea-minDirtArea}$ для $dirtArea$ такой, что $maxDirtArea > dirtArea > minDirtArea$. Если $dirtArea < minDirtArea$, то результат 1, если $dirtArea > maxDirtArea$, то результат 0.

На рисунках 5, 7 видно, что не только объекты могут быть в одной и той же части на левом и правом изображении, но и пятна, что может приводить к тому, что некоторые пятна будут необнаружены. Это сделка между количеством ложно-положительных и ложно-отрицательных срабатываний.

При движении в чертах города, с постоянно меняющейся окружающий обстановкой, находить пятна значительно проще, чем, например, при движении на гольф-поле, так как градиентная карта чаще будет иметь во всех частях изображения отличные от нуля значения (проезжающие машины, появляющиеся здания, дорожные знаки и люди имеют четкие границы), в то же время трава (не имеет четких границ) имеет

близкий к нулю градиент и незначительно меняется от кадра к кадру (рисунок 8).

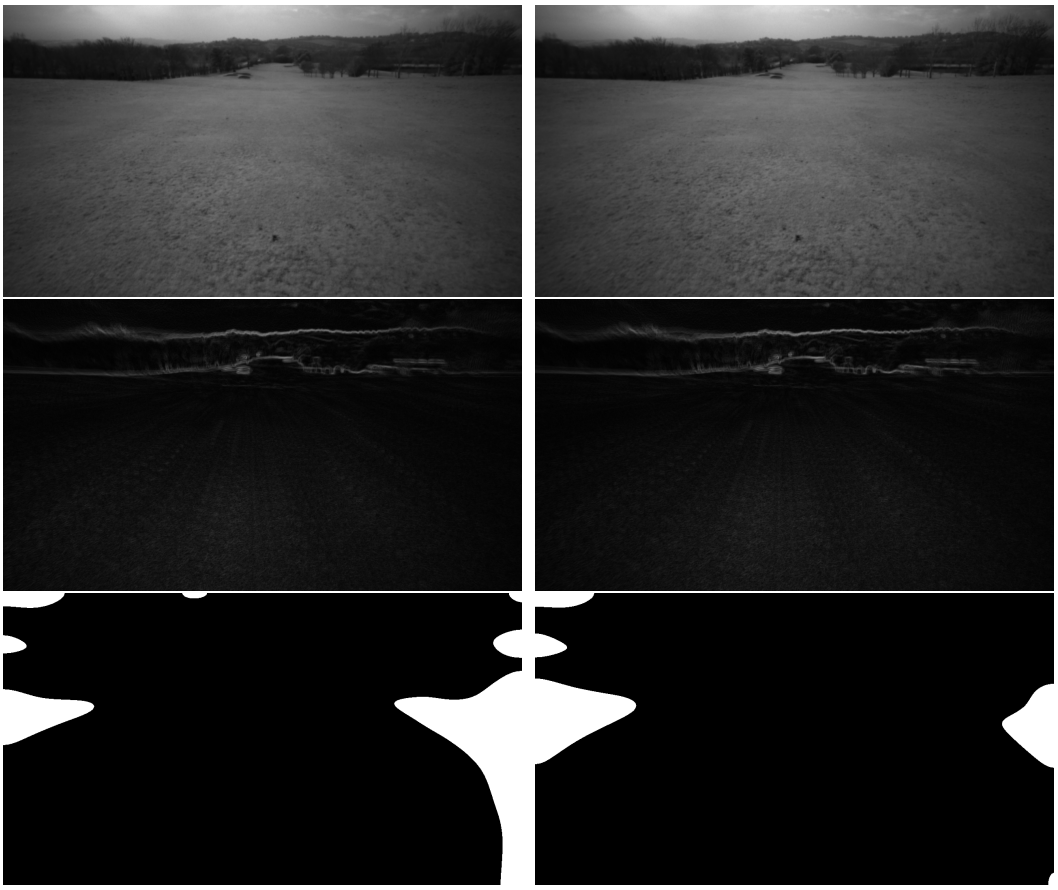

Рис. 8: Стереопара, усредненные градиентные карты и маски при движении с чистой камерой на гольф-поле. Ложные детекции по краям изображения возникают на каждой маске из-за близкого к нулю градиента у травы в затемненной части изображения (XOR-маска помогает и здесь). $N = 10$

5.4 Время работы алгоритма

Алгоритм реализован на C++ с использованием библиотеки OpenCV, запускался на Jetson AGX Xavier с использованием всех ядер процессора на максимальной мощности (MAXN mode) на полученном в ходе работы датасете.

Всего было совершено 20 запусков подряд, изображения уменьшались в 16 раз (в 4 по ширине, в 4 по высоте).

Из оптимизаций использовалось распараллеливание, описаное выше, а так же принятие решение выделено в отдельную эпоху, например, при $N = 100$ (результат алгоритма выдается каждый сотый кадр) первые 98 кадров вычисляется усредненная градиентная карта, на следующем вычисляются маски для левой и правой камеры, а на последнем фильтрация маски (самая долгая часть алгоритма) и вычисление результата. Вторая оптимизация была сделана, чтобы не выходить по времени работы за 5 миллисекунд (за 20 запусков нет ни одной эпохи, на которой время работы превышает 5 миллисекунд).

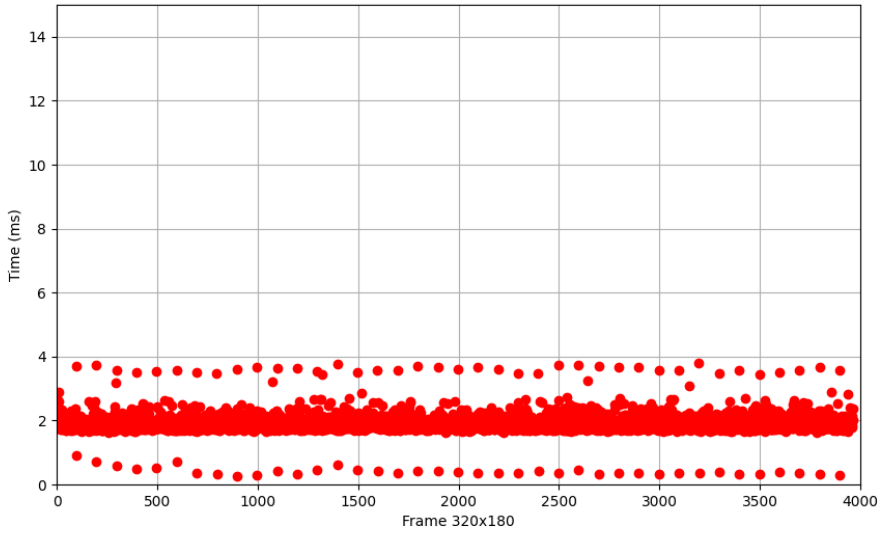

Рис. 9: Среднее время работы алгоритма при $N = 100$

5.5 Качество работы алгоритма

Для проверки точности алгоритма так же был сгенерирован датасет с пятнами на основе чистого датасета с гольф-поля, всего в нём 4667 изображений для каждой камеры. При работе на исходном изображении минимальное пятно вмещается в прямоугольник 120x110, максимальное 230x340 пикселей. Минимальным пятном на реальном датасете является область размера 65x55 пикселей, а максимальным 360x180.

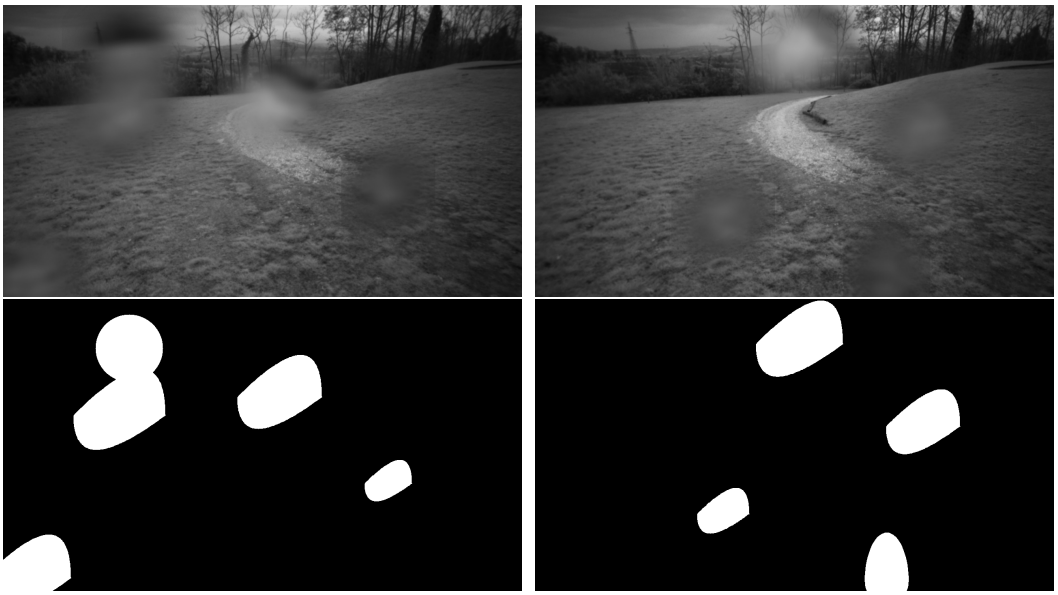

Рис. 10: Искусственно сгенерированные пятна на изображении с гольф-поля.

Качество алгоритма измеряется количеством пропущенных пятен. Количество объектов, ложно отмеченных пятнами, одно из измерений которого больше 50 пикселей, выявлено не было. Причем на датасете, полученном в реальных условиях, человеку почти невозможно определить – является ли область такого размера пятном.

Качество зависит от длины N последовательности изображений, после которой алгоритм выдаёт результат. На полном изображении при $N = 50$ 12% всех пятен было пропущено, при $N = 100$ количество пропущенных пятен уменьшается и достигает 7%.

При уменьшении исходного изображения в 16 раз качество детекции падает, потому что наименьшие пятна на левом и правом изображении перестают обнаруживаться. При условии, что пятна на изначальном изображении имеют площадь хотя бы 15000 пикселей и уменьшение происходит не более чем в 16 раз, качество будет оставаться прежним.

Качество алгоритма считается приемлемым, потому что аналоги либо занимают больше времени на эпохе, либо их методов недостаточно

для работы на гольф-поле. Так же количество пропущенных пятен можно уменьшить подбором параметров, описанных выше. Например, порог t для бинаризации зависит от качества изображения и может иметь разные значения при работе на изображениях с разных камер.

6 Заключение

В ходе работы были выполнены следующие задачи:

- Сделан обзор предметной области
- Сделан обзор известных алгоритмов детекции пятен
- Подготовлен датасет в неблагоприятных погодных условиях
- Реализован алгоритм детекции пятен на стереокамере
- Проведена оценка скорости и качества работы алгоритма

В будущем планируется перенести выполнение алгоритма с CPU на GPU.

Список литературы

[1] T. S. Huang. “Computer Vision: Evolution and Promise”. In: *19th CERN School of Computing* (1996), pp. 21–25.

[2] Anchal Kalra and Roshan Lal Chhokar. “A Hybrid Approach using Sobel and Canny operator for Digital Image Edge Detection”. In: *International Conference on Micro-Electronics and Telecommunication Engineering* (2016).

[3] Fadi Al Machot et al. “Real-time raindrop detection based on cellular neural networks for ADAS”. In: *Journal of Real-Time Image Processing* 16 (2019), pp. 931–943.

[4] Martin Roser and Andreas Geiger. “Video-based raindrop detection for improved image registration”. In: *IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops* (2009).

[5] Zhixiang Hao et al. “Learning From Synthetic Photorealistic Raindrop for Single Image Raindrop Removal”. In: *IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)* (2019).

[6] Florin Bogdan Marin and Mihaela Marin. “Real-Time Raindrop Detection Based on Deep Learning Algorithm”. In: (2020). DOI: https://doi.org/10.35219/mms.2020.4.09.

[7] Fawzi Nashashibi, Raoul de Charette, and Alexandre Lia. “Detection of Unfocused Raindrops on a Windscreen using Low Level Image Processing”. In: *11th International Conference on Control, Automation, Robotics and Vision, ICARCV 2010* (2010).

[8] Vera Soboleva and Oleg Shipitko. “Raindrops on Windshield: Dataset and Lightweight Gradient-Based Detection Algorithm”. In: (2021).

[9] Satoshi Suzuki and Keiichi Abe. “Topological structural analysis of digitized binary images by border following”. In: *Computer Vision, Graphics, and Image Processing* 30 (1985), pp. 32–46.