

Санкт-Петербургский государственный университет

Кафедра системного программирования

Группа 21.Б10-мм

Патов Артемий Сергеевич

Анализ дальности видимости по видео в ADAS

Отчёт по учебной практике
в форме «Решение»

Научный руководитель:
доцент кафедры СП, к.т.н., Ю. В. Литвинов

Консультант:
программист АО «Кама» М.С. Осечкина

Санкт-Петербург
2023

Оглавление

1. Введение	3
2. Постановка задачи	5
3. Обзор	6
3.1. Обзор используемых алгоритмов	6
3.1.1. Составление изображения с минимальным значе- нием между несколькими кадрами для каждого пикселя	6
3.1.2. Размытие по Гауссу	6
3.1.3. Оператор Лапласа	6
3.1.4. Детектор границ Кэнни	7
3.1.5. Векторизация изображения	7
3.2. Обзор аналогов	8
4. Реализация	10
5. Заключение	13
Список литературы	14

1. Введение

ADAS — это комплекс систем дополненной реальности для автомобилей, ориентированных на то, чтобы помочь водителю проще и безопаснее вести машину. ADAS включает в себя системы предупреждения о покидании полосы, предупреждения о столкновении и автоматического экстренного торможения, системы обнаружения пешеходов, системы дополненной навигации и прочие. Согласно исследовательскому отчету Canalys за 2021 год [2], примерно 33% новых автомобилей, проданных в США, Китае и Японии, имеют функции ADAS, а по данным Главного управления дорожного движения Испании [7], система помощи водителю может снизить риск ДТП на 57%.

Туман, плохое освещение, сильные осадки в большинстве случаев являются причинами серьезных аварий: в такую погоду значительно ухудшается дальность видимости. Чтобы избежать ДТП в подобных условиях, водитель должен очень быстро ориентироваться, однако в опасных ситуациях скорости реакции человека может не хватить, и именно в таких случаях выручает ADAS: во-первых, система помощи водителю корректирует поведение автомобиля на дороге в зависимости от дорожных условий; во-вторых, в экстренных ситуациях ADAS способна отреагировать за десятые доли секунды и предотвратить аварию. Метод, позволяющий точно анализировать дальность видимости при любых погодных условиях, сыграет значительную роль в уменьшении количества аварий, так как он позволит установить верхнюю границу скорости автомобиля,отрегулирует дальность света фар. Помимо этого он сможет использоваться при принятии решений об отключении автопилота, при установке области для вывода дополненных объектов и области на кадре для распознавания объектов.

В результате поиска существующих решений удалось найти статьи, в которых описаны варианты решения проблемы оценивания дальности видимости, однако ни один из представленных способов не решает задачу в полной мере: например, в статье [5] данные поступают не с видеорегистратора, а с придорожных камер, которыми могут быть не

оборудованы некоторые трассы, также описанный метод не предполагает высокой точности оценок; в статьях [8, 1] описаны методы, способные определить дальность видимости в туман, однако не умеющие делать это при плохом освещении.

В ходе данной работы предлагается реализовать алгоритм, позволяющий определять дальность видимости в условиях тумана, осадков, темноты и прочего. Метод можно применять в тех случаях, когда обзор ничем не перекрыт и машина движется по прямой, прочие случаи не рассматриваются.

2. Постановка задачи

Цель данной работы — предложить метод, позволяющий оценить дальность видимости, и реализовать его. Метод должен работать в любых погодных условиях и при любом освещении, однако в данной работе действует предположение о том, что машина движется по прямой, а обзор камеры ничем не перекрыт.

Для достижения цели были поставлены следующие задачи.

1. Изучить существующие решения, позволяющие оценивать дальность видимости.
2. Предложить метод, позволяющий измерять дальность в любых погодных условиях.
3. Реализовать предложенный алгоритм.
4. Оценить скорость и качество работы алгоритма.

3. Обзор

3.1. Обзор используемых алгоритмов

3.1.1. Составление изображения с минимальным значением между несколькими кадрами для каждого пикселя

При наличии последовательности из n кадров $X_1, X_2, \dots, X_n$, изображением с минимальным значением между несколькими кадрами для каждого пикселя называется изображение I , где значение пикселя на позиции (i, j) в матрице этого изображения вычисляется следующим образом:

$$I[i, j] = \min(X_1[i, j], X_2[i, j], \dots, X_n[i, j]).$$

3.1.2. Размытие по Гауссу

Чтобы избавиться от осадков и других различных шумов, можно применить размытие к изображению. В данной работе будет использоваться размытие по Гауссу [4], которое работает следующим образом: выбирается размер и отклонение σ , строится матрица с заданным размером, где элементы — это значения гауссовой функции

$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$ с данным отклонением. Матрица нового изображения получается путем конволюции между ядром Гаусса и подматрицами исходного изображения.

3.1.3. Оператор Лапласа

На границах объектов значения первой производной функции интенсивности изображения, как правило, становятся локальными экстремумами. Соответственно, вторые производные функции интенсивности изображения в таких точках будут обращаться в ноль. На этой идее построен детектор границ с использованием оператора Лапласа [6]. Его проблема состоит в том, что обращение второй производной в ноль не является критерием границы в этой точке, поэтому могут быть ложные срабатывания. Помимо этого, границы, которые получаются в резуль-

тате применения маски Лапласа, очень нечеткие и разрывные, что не подходит под описанный в этой работе метод.

3.1.4. Детектор границ Кэнни

В данной работе для обнаружения границ дороги используется оператор Кэнни [3]. Его преимущество над методом Лапласа состоит в том, что он не создает нового шума и очень точно определяет границы дороги. Детектор границ Кэнни состоит из следующих этапов:

1. Применение размытия по Гауссу.
2. Поиск градиентов изображения, осуществляющийся посредством применения оператора Собеля.
3. Подавление немаксимумов — проверка на то, что каждый пиксель является локальным максимумом в своей окрестности в направлении градиента. Если это не так, то значение пикселя меняется на 0.
4. Двойная пороговая фильтрация. Считается, что точки, градиент которых больше верхнего фиксированного порога, принадлежат контуру. В случае, когда градиент точки меньше нижнего порога, значение пикселя устанавливается в 0.
5. Трассировка области неоднозначности — проверка на наличие в окрестности каждого пикселя по меньшей мере одного пикселя, значение которого больше верхнего порога.

3.1.5. Векторизация изображения

Под векторизацией изображения понимается выделение всех контуров изображения. Выделение контуров является конечным этапом обработки изображения для обнаружения границ дороги, он осуществляется при помощи метода `findContours` из библиотеки OpenCV, реализующего алгоритм Сузуки [9]. Алгоритм Сузуки — это алгоритм поиска границ изображения, основанный на алгоритме следования границ.

Он работает следующим образом: изображение полностью сканируется, начиная с верхнего левого пикселя, заканчивая правым нижним. Сканирование прерывается при нахождении пикселя, удовлетворяющего условиям принадлежности границе. В таком случае запускается алгоритм следования границ, в итоге все точки границы, к которой относится данный пиксель, помечаются одним и тем же номером i , либо $-i$. После этого сканирование продолжается. Стоит отметить, что номера, которыми помечаются пиксели, могут меняться при определенных условиях в ходе алгоритма.

При помощи алгоритма Сузуки можно не только сгенерировать массив контуров, но и установить иерархические отношения между контурами. Однако в рамках данной работы требуется лишь определение самых длинных контуров, а также нахождение их самых дальних точек, поэтому иерархия игнорируется.

3.2. Обзор аналогов

В статье [5] приводится метод, основанный на машинном обучении, который позволяет оценивать дальность видимости в любых погодных условиях. Однако, во-первых, в этой статье система обучается по кадрам со статичных камер, установленных по краям трассы, следовательно, подавать изображения, полученные с видеорегистратора не валидно, а в рамках данной работы действует предположение о том, что данные поступают именно с камеры, установленной на борту автомобиля. Во-вторых, алгоритм, приведенный в статье, использует метод опорных векторов с шестью категориями, дальность видимости оценивается по шестибальной шкале, точная оценка не предполагается. В рамках данной работы планируется измерять дальность видимости в метрах, как расстояние от камеры до границы видимой части горизонта. В-третьих, классификация нечеткая, соответственно, есть вероятность внесения человеческой ошибки.

В статьях «Image based fog detection and visibility estimation for driving assistance systems» [8] 2006 года и «Automatic fog detection and

estimation of visibility distance through use of an onboard camera» [1] 2013 года предложены алгоритмы, основанные на законах Кошмидера и Дантли. Первый устанавливает соотношение между яркостью объекта на расстоянии и вблизи с поправкой на плотность тумана. При этом Второй закон выводится из первого и может применяться только в условиях дневного освещения. Соответственно, данный метод может определять дальность видимости в условиях тумана, но не работает при плохом освещении. В рамках данной работы планируется измерить дальность видимости еще и в условиях плохого освещения.

4. Реализация

Инструменты

Для совместимости с другими проектами заказчика в качестве языка программирования для данной работы был выбран C++. Для обработки видео и изображений используется библиотека OpenCV, написанная на языке C++.

Удаление шума

Осадки, шумы различного рода — всё это может помешать качественному определению границ дороги. При наличии видеоряда от многих подобных проблем можно избавиться, особым образом составляя изображение из нескольких кадров. Для этого была реализована функция, которая, при наличии n кадров, составляет новое изображение, где для каждого пикселя выбирается минимальное значение между несколькими кадрами.

Для удаления оставшихся шумов к полученному изображению применяется размытие по Гауссу, реализованное в методе `GaussianBlur` из библиотеки OpenCV.

Поиск контуров

После удаления шума к изображению применяется оператор Кэнни с подобранными опытным путем верхним и нижним порогами.

Чтобы определить самую дальнюю видимую точку, было принято решение векторизовать изображение. Для этого после бинаризации изображения и выделения контуров применяется алгоритм Сузуки, что позволяет получить массив контуров. Однако возникает проблема: из-за разного уровня освещенности длинные контуры в некоторых местах могут разорваться, поэтому был реализован алгоритм, который соединяет края контуров в некоторой окрестности. Метод `findContours` [10] возвращает массив контуров, каждый контур представляется в виде

массива точек. Алгоритм соединения контуров работает следующим образом: на вход подается контур, находится его самая высокая точка (с наименьшим значением y). Далее рассматривается квадратная окрестность этой точки, $n \times n$ пикселей; найденные в этой окрестности другие контуры присоединяются к начальному контуру. Затем алгоритм рекурсивно запускается от нового полученного контура. Этот метод применяется к первым N самым длинным контурам. После применения данного алгоритма получаются длинные, непрерывные контуры, по которым можно определять горизонт видимости.

Определение границы видимой части

Из множества всех контуров необходимо выделить те, по которым будет оцениваться дальность видимости. Для этого нужно знать точку схода: рассматриваются только контуры, полностью лежащие ниже точки схода на кадре. В идеале оценивать дальность видимости нужно только по контурам разметки, однако в рассмотрение могут попасть еще и контуры барьерных ограждений или сигнальных столбиков, стоящих по бокам трассы. Их влияние на итоговую оценку необходимо снизить, ведь на кадре их крайние точки находятся выше точек дороги, соответственно, реальная оценка может быть значительно хуже оценки, полученной при рассмотрении контуров ограждений. Чтобы отличать контуры ограждений от границ дороги, можно установить порог угла между вертикалью, на которой лежит точка схода, и отрезком от точки схода до высшей точки контура. При выходе за этот порог, координата высшей точки контура будет умножаться на коэффициент, подобранный опытным путем.

Алгоритм рассматривает только N первых самых длинных контуров, удовлетворяющих условию пребывания ниже точки схода.

Измерение расстояния

Для нахождения расстояния применяется формула, основанная на пересечении плоскости дороги и плоскости, проходящей через точку

камеры и прямой, до которой измеряется расстояние.

$$dist = \frac{focal * height}{D - horizon}$$

focal — фокальное расстояние в пикселях

height — высота камеры в метрах

D — горизонталь на кадре, до которой измеряется расстояние

horizon — горизонт на кадре

5. Заключение

В результате работы были выполнены следующие задачи.

- Проведен обзор существующих решений, позволяющих оценивать дальность видимости.
- Предложен и реализован метод, позволяющий выделять границу видимой части горизонта.
- Реализовано измерение расстояние до границы видимой части горизонта.

Дальнейшие планы:

- Проведение экспериментов.
- Улучшение алгоритма.

Код работы доступен на сервисе GitHub.

Список литературы

- [1] Automatic fog detection and estimation of visibility distance through use of an onboard camera / Nicolas Hautière, Jean-Philippe Tarel, Jean Lavenant, Didier Aubert // *Mach. Vis. Appl.* — 2006. — 04. — Vol. 17. — P. 8–20.
- [2] Canalys. Huge opportunity as only 10% of the 1 billion cars in use have ADAS features. — URL: <https://www.canalys.com/newsroom/huge-opportunity-as-only-10-of-the-1-billion-cars-in-use-have-a> (online; accessed: 2022-12-06).
- [3] Canny edge detector. — URL: https://en.wikipedia.org/wiki/Canny_edge_detector (online; accessed: 2022-12-06).
- [4] Gaussian blur. — URL: https://en.wikipedia.org/wiki/Gaussian_blur (online; accessed: 2022-12-06).
- [5] Image-Based Visibility Estimation Algorithm for Intelligent Transportation Systems / Li Yang, Radu Muresan, Arafat Al-Dweik, Leontios J. Hadjileontiadis // *IEEE Access*. — 2018. — Vol. 6. — P. 76728–76740.
- [6] Laplace operator. — URL: https://en.wikipedia.org/wiki/Laplace_operator (online; accessed: 2022-12-06).
- [7] Mapfre Fundacion. What are ADAS systems? — URL: <https://www.fundacionmapfre.org/en/education-outreach/road-safety/adas-systems/what-are-adas-systems/> (online; accessed: 2022-12-06).
- [8] Negru Mihai, Nedevschi Sergiu. [Image based fog detection and visibility estimation for driving assistance systems](#) // 2013 IEEE 9th International Conference on Intelligent Computer Communication and Processing (ICCP). — 2013. — P. 163–168.

- [9] Suzuki Satoshi, be KeiichiA. Topological structural analysis of digitized binary images by border following // [Computer Vision, Graphics, and Image Processing](#). — 1985. — Vol. 30, no. 1. — P. 32–46. — URL: <https://www.sciencedirect.com/science/article/pii/0734189X85900167>.
- [10] findContours. — URL: https://docs.opencv.org/3.4/d3/dc0/group__imgproc__shape.html#ga17ed9f5d79ae97bd4c7cf18403e1689a (online; accessed: 2022-12-06).