

Обзор алгоритмов стереозрения и их применимости для получения результатов в реальном времени

Пантелеев М.

Санкт-Петербургский Политехнический Университет Петра Великого
`panteleev.md@edu.spbstu.ru`

Аннотация В этой статье приводится обзор основных алгоритмов, связанных с получением глубины по снимкам одной сцены с нескольких камер. Улучшение существующих и создание новых методов происходит постоянно, так как проблема точного и быстрого стерео до сих пор не решена. Разные подходы склоняются либо в сторону высокой, производительности, либо в сторону повышенной точности. Алгоритмы, способные выполнять сопоставление в реальном времени представляют особый интерес для исследователей, поэтому выбранные методы рассматриваются также на возможность такого применения.

1 Введение

Получение трёхмерной структуры пространства по стереоснимкам - это задача, подходы к которой начали искать очень давно. И хотя множество решений было найдено, исследователи продолжают работу над более эффективными алгоритмами. В силу дешевизны (относительно других сенсоров, позволяющих получить объёмную информацию о пространстве) и универсальности камер, на подобные методики есть большой спрос со стороны таких областей как автономный транспорт и мобильная робототехника. Эти области весьма требовательны к точности результатов и скорости работы алгоритмов, так как устройства должны принимать оперативные решения на основе данных о внешнем мире, а мощности бортовых вычислителей весьма ограничены. Именно эти качества находятся в фокусе современных исследований, поэтому в этой работе также будет уделено внимание возможности того или иного алгоритма работать в реальном времени. Эта возможность характеризуется частотой, с которой алгоритм просчитывает карты глубины. Очевидно, что фактическое значение частоты кадров сильно зависит от используемого вычислителя [36], разрешения камер и диапазона глубины, который способен определить алгоритм, поэтому для теоретического сравнения больше подходит вычислительная сложность алгоритма. К сожалению, не все авторы приводят оценку вычислительной сложности собственных алгоритмов, в этих случаях указана реальная производительность, заявленная авторами или одним из популярных бенчмарков.

При этом в стереозрении есть ряд проблем и ограничений, которые осложняют получение надёжной и полной информации по снимкам. Часть этих проблем связана с недостатками самого способа получения изображений (заслонение), другая часть - с особенностями отображения некоторых поверхностей на двумерных снимках (наклонные плоскости или слаботекстурированные области). Снижение влияния этих особенностей на результат путём создания новых алгоритмов или модификации существующих является второй важной целью исследователей.

Остальная часть статьи организована следующим образом: в продолжении этой секции рассмотрен базовый принцип стереозрения, секция 2 описывает самые популярные подходы к построению карт расхождения и их применимость для получения результатов в реальном времени, в секции 3 представлены выводы.

1.1 Принцип стереозрения

Несмотря на существование разных алгоритмов стереозрения, все они реализуют общий принцип. Задача стереозрения состоит в использовании двух или более камер для получения данных о дальности до объектов в кадре.

Как правило, система стереозрения состоит из двух камер, наблюдающих сцену с разных точек, как изображено на рисунке 1 [17]. Фундаментальная основа принципа заключается в предположении, что каждой точке в пространстве соответствует уникальная пара пикселей на снимках с двух камер.

При этом к камерам предъявляются некоторые требования [1]:

- Камеры откалиброваны. Это значит, что известны внутренние (оптические) и внешние (расположение камер в пространстве) параметры камер.
- Ректификация. Подразумевает выравнивание изображения с обеих камер по строкам.
- Ламбертовость поверхностей. Означает независимость освещения поверхности от угла зрения.

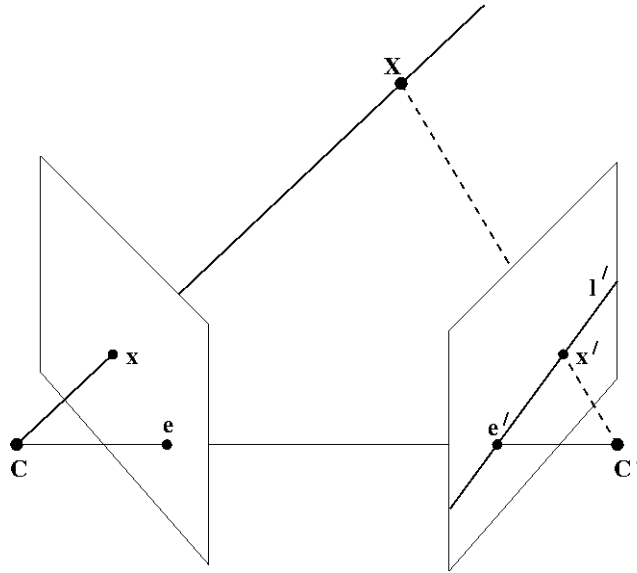


Рис. 1: Эпикулярная геометрия

Таким образом, соблюдение указанных выше требований позволяет использовать следующий геометрический принцип. При наличии двух камер, как изображено на рисунке 1, где C — центр первой камеры, C' — центр второй камеры, точка пространства X проецируется в x на плоскость изображения левой камеры и в x' на плоскость изображения правой камеры. Прообразом точки x на изображении левой камеры является луч xX . Этот луч проецируется на плоскость второй камеры в прямую l' , называемую эпикулярной линией. Образ точки X на плоскости изображения второй камеры обязательно лежит на эпикулярной линии l' .

В результате каждой точке x на изображении левой камеры соответствует эпикулярная линия l' на изображении правой камеры. При этом соответствие для x на изображении правой камеры может лежать только на соответствующей эпикулярной линии. Аналогично, каждой точке x' на правом изображении соответствует эпикулярная линия l на левом.

Далее с помощью точек x и x' возможно посчитать смещения каждого пикселя одного изображения относительно другого, что даёт карту смещений (disparity map). Очевидно, что смещения будут подсчитаны только для точек, видимых обеими камерами. Карта смещений же приводится далее либо к облаку точек, либо к карте глубины. Пример такой карты представлен на рисунке 2 [23].



Рис. 2: Примеры результата работы

На практике работу большинства алгоритмов можно разделить на 3 этапа: получение изображений, поиск соответствий и восстановление информации о глубине. Это позволяет организовать классификацию алгоритмов на основе подходов к каждому из этих этапов. В этой статье будут рассмотрены в основном лишь методы поиска соответствий на изображениях.

Восстановление заключается в расчёте глубины по известным внешним параметрам камер. Глубина каждой точки X , воспринимаемой двумя камерами с оптическими центрами C и C' , определяется проекциями x, x' этой точки на каждый кадр. Тогда значение глубины можно найти по уравнению 1.1, указанному ниже.

$$Z = f \frac{T}{d}, \quad (1.1)$$

где T - расстояние между C и C' ; d - смещение, $d = x - x'$; f - фокусное расстояние камеры.

Получение изображений обычно осуществляется с использованием двух синхронизированных камер. Однако есть работы [29,7], которые решают эту задачу с использованием лишь одной камеры совместно с системой линз и зеркал, но принцип их функционирования по своей сути симулирует двухкамерную реализацию.

Другой вектор работ направлен на получение информации о глубине по снимкам, сделанным с помощью камер со сверхшироким углом обзора, так как они часто встречаются в мобильных роботах [12,30], но существующие работы пока не могут обеспечить результаты, сравнимые с традиционными камерами без существенных вычислительных затрат.

2 Поиск соответствий

Как было ранее сказано, поиск соответствий можно выполнять широким набором методов. Цель каждого метода - постараться найти для каждого пикселя одной картинке соответствующий ему пиксель на второй. Достичь этого можно двумя подходами, в зависимости от накладываемых на зону поиска соответствий ограничений. При подсчёте расхождений лишь в

небольшой окрестности (окне) вокруг интересующего нас пикселя мы говорим о локальных методах - эти методы используют небольшое количество информации и относительно быстродейственны. Их принято делить на три категории: сопоставление блоков, градиентные методы и сопоставление признаков. Глобальные методы же обрабатывают целый ряд пикселей или всё изображение целиком. Они не так чувствительны к локальным дефектам, мешающим процессу поиска соответствий (например, заслонение), но при этом имеют куда большую вычислительную сложность. Одними из самых популярных подходов в этой группе методов являются динамическое программирование [28], и алгоритм разреза графа.

2.1 Сопоставление блоков

Block Matching - это локальный метод, который заключается в оценке расхождения в точке на одной картинке с помощью сравнения небольшой области вокруг этой точки с такими же областями на другой []. Благодаря выпрямлению (ректификации) поиск соответствующей области на другой картинке ограничивается одним измерением. По этому измерению (как правило, строке) считается одна из доступных метрик, и регион с наименьшим её значением считается искомым. Существует целый набор метрик, часто используемых в этом методе.

Для отдельных пар пикселей можно рассчитывать абсолютную разность (AD) или квадратичную разность (SD), выраженную через интенсивности:

$$AD(x, y, d) = |I_l(x, y) - I_r(x + d, y)|, \quad (2.1)$$

где (x, y) - координаты первого пикселя; d - сдвиг по оси x между двумя пикселями (смещение); I - интенсивность данного пикселя. Обычно I_l обозначают опорное изображение, а I_r - целевое. Абсолютная разность - простейшая метрика, за счёт чего до сих пор используется в многих алгоритмах, от которых требуется производительность в реальном времени.

Для квадратичной разности:

$$SD(x, y, d) = |I_l(x, y) - I_r(x + d, y)|^2. \quad (2.2)$$

На основании этих метрик были выведены более точные и сложные, такие как сумма квадратичных разностей (SSD) и сумма абсолютных разностей (SAD). Они применяют описанный выше принцип для целого окна пикселей вокруг исследуемого.

Также применяется нормированная кросс-корреляция (NCC) - стандартный статистический метод для поиска соответствий с шаблоном, который записывается для случая поиска соответствий пикселей согласно уравнению 2.3.

$$NCC(x, y) = \frac{\sum_{x,y} (I_l(x, y) - \bar{I}_l)^2 * (I_r(x + d, y) - \bar{I}_r)^2}{\sqrt{\sum_{x,y} (I_l(x, y) - \bar{I}_l)^2 * (I_r(x + d, y) - \bar{I}_r)^2}}, \quad (2.3)$$

где $\overline{I_l}$ и $\overline{I_r}$ - средние интенсивности соответствующих изображений. Эта метрика устойчива к перепадам яркости и контрастности изображения благодаря нормализации [18], но требует выполнения значительно большего числа арифметических операций для своего подсчёта.

Таким образом, почти любой алгоритм сопоставления блоков позволяет получать относительно быстрые результаты сам по себе, но точность остаётся маленькой и чувствительной к различным условиям освещения. Поэтому многие работы в последние годы пытались сгладить недостатки этого метода, используя вместе с тем возможность независимого подсчёта для параллелизации. Например, эти метрики позволяют повышать качество итоговой карты смещений за счёт варьирования размера окна и дополнительных проходов по изображениям [14]. Шан и другие [34] реализовали эффективный метод стереосопоставления на основе программируемых логических интегральных схем (ПЛИС) и оптимизированного сопоставления по сумме абсолютных разностей. Их алгоритм позволяет обрабатывать изображения со скоростью до 46 кадров в секунду при разрешении в $1280 * 1024$ пикселей. Ейнике и Эггерт [8] предложили алгоритм многоблочного сопоставления (Multi-Block-Matching), который комбинирует окна разного размера и формы вероятностным образом, что позволило значительно повысить качество выходных карт глубины. Позже их алгоритм был реализован на GPU, что позволило добиться производительности в 41 кадр в секунду при разрешении $2888 * 1920$ пикселей и большим диапазоне глубины [5].

2.2 Сопоставление признаков

Довольно распространёнными являются методы, основанные на поиске и сопоставлении характерных точек. За поиск этих точек отвечают методы SIFT [25], SURF [2], детектор Харриса и других. Различным способам поиска характерных точек посвящено немало статей.

Суть методов, относящихся к этой группе, заключается в выделении на снимках характерных точек: углов и точек смены контраста. Далее для найденных точек считается дескриптор - вектор, являющийся численной характеристикой окрестности характерной точки. Таким образом, установление соответствий сводится к сравнению численных величин этих векторов.

Метод менее чувствителен к перекрытиям и слаботекстурированным областям, но, к сожалению, плотность точек, для которых возможно подсчитать глубину, получается относительно низкой. Результаты получаются быстро, но поиск соответствий сильно усложняется при отличающихся ракурсах камер. Всё это привело к снижению интереса к группе методов.

Однако новые работы в этой области всё-таки продолжают публиковаться. Так был предложен метод на основе SIFT, использующий нейронные сети для достижения более высокой производительности [24] сопоставления признаков. Работа Экстранда и других [9] же опирается на использование комбинации из сегментации изображения и обнаружения границ в качестве метрики поиска совпадений. Такая реализация работает быстро, но точность

результатов по-прежнему остаётся невысокой, особенно в областях прерывистости.

2.3 Методы динамического программирования

Методы, основанные на динамическом программировании, являются одними из самых часто используемых среди глобальных подходов. В случае с поиском совпадений на изображениях их применение означает использование конструкции DSI (disparity space image). Строится она следующим образом: выбираются i -ые строки левого и правого изображения (они должны быть ректифицированы), далее одна строка постепенно "сдвигается" относительно другой и на каждом этапе одна из метрик из пункта 2.1 подсчитывается для совпадающей пары пикселей и записывается в DSI, который в итоге содержит в себе расхождения каждого пикселя с каждым для всех попарных рядов исходных изображений. Строки DSI, соответствующие заведомо невозможным значениям (например, расхождения больше максимально допустимых или пиксель на левом изображении находится левее соответствующего ему на правом) отбрасываются. Рисунок 3 [3] иллюстрирует этот принцип.

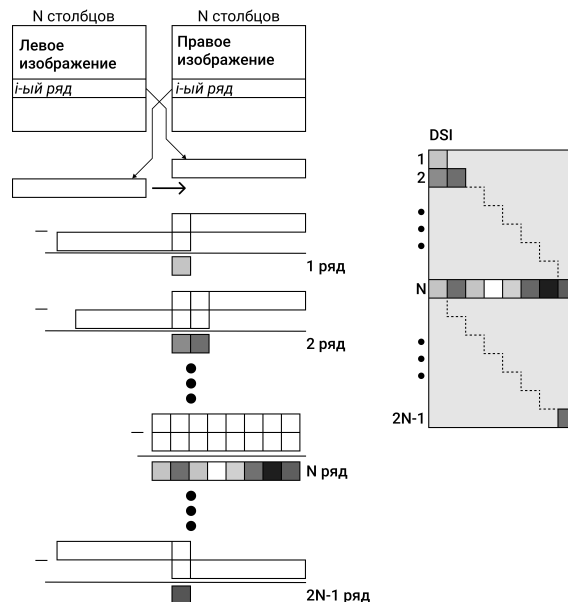


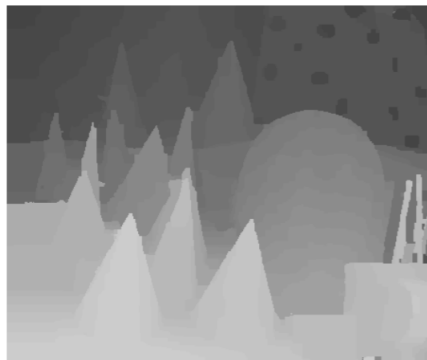
Рис. 3: Процесс построения DSI. Строка левого изображения считается неподвижной, пока строка правого изображения смещается вдоль неё. Результат вычитания перекрывающихся пикселей складывается в очередную строку DSI. Сам DSI обрезается с учётом ограничений.

Далее задача состоит в поиске оптимального пути от левого верхнего угла до правого нижнего на полученной матрице. Как уже упоминалось, динамическое программирование уменьшает вычислительную сложность задачи оптимизации за счёт разбиения её на множество более простых и мелких задач. В данном случае это происходит за счёт того, что глобальная "стоимость" пути складывается рекурсивно из локальных, которые в свою очередь сводятся к стоимости движения по отдельным пикселям. Эта стоимость зависит от типа движения, который определяется величиной расхождения в точке. Для каждого элемента матрицы возможны три типа "движения": горизонтальный (означающий совпадение), вертикальный и диагональный (оба соответствуют заслонённым областям). Это позволяет внести дополнительные ограничения и уменьшить число доступных путей. Тем не менее, количество путей, которые приходится подсчитывать, даже с этими ограничениями остаётся довольно большим.

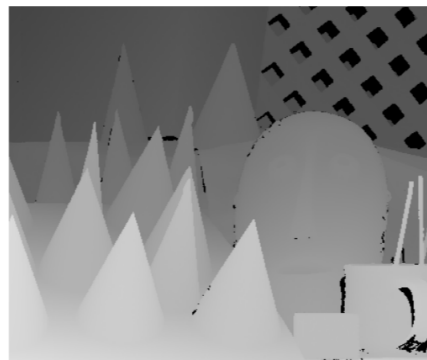
Данный алгоритм позволяет обрабатывать каждую строку изображения независимо и выполнять вычисления параллельно. Другими преимуществами является лучшее распознавание в слаботекстурированных областях по сравнению с локальными методами и возможность обработки заслонённых областей. К недостаткам метода относят высокую вычислительную сложность (до $O(n^4)$, где n - количество пикселей в строке изображения) и возможность распространения локальной ошибки на последующие пиксели, что приводит к характерным горизонтальным полосам на картах глубины.

Дополнительное ограничение, позволяющее уменьшить количество доступных путей, может дать использование заранее известных GCP (Ground Control Points). Эти точки, положение и соответствие которых можно определить заранее с большой точностью. Их использование позволяет снизить сложность поиска пути до 25% от оригинальной задачи [3].

Высокий интерес к этому методу привёл к появлению большого числа работ, пытающихся устранить недостатки метода и повысить его эффективность. Так была предложена модель [35], которая опирается на ограничение видимости - предположение, что каждый заслонённый пиксель не имеет соответствия на втором изображении, а незаслонённый - обязательно имеет. В результате карты глубины не теряют информацию в местах, которые видит только одна камера, но при этом ухудшается детализация (рис. 4).



Результаты алгоритма



Реальная глубина

Рис. 4: Сравнение карты глубины, полученной алгоритмом, с эталонным изображением.

С точки зрения производительности в реальном времени глобальные методы, основанные на динамическом программировании, до сих пор показывают себя не лучшим образом из-за своей вычислительной сложности. Однако всё-таки имеются успешные примеры подобных систем, реализованных на специальных вычислителях, способных воспользоваться возможностью параллелизации данных алгоритмов. Лараби и другие [22] смогли реализовать построение DSI с последующей его обработкой на ПЛИС и добиться производительности в 138 кадров в секунду для изображений 450×375 пикселей. Реализацию на GPU осуществили Халлек и другие [15]. Они ввели новую метрику поиска соответствий, называемую ZSAD (сумма абсолютных разностей с нулевым средним), и построили с помощью неё DSI. Алгоритм достиг производит 29 кадров в секунду в сравнительных тестах и показал большую точность в сравнении с другими предложенными авторами алгоритмами, работающими в реальном времени.

2.4 Алгоритм разреза графа

Однако у предыдущего метода есть недостаток, выраженный в неспособности учитывать горизонтальную и вертикальную непрерывность поверхностей на снимке, так как столбцы/строки рассматриваются независимо. Эффективно исправить это в рамках динамического программирования пока не удалось. Альтернативный подход предлагает привести задачу к поиску максимального потока в графе. Для этого часто используется случайное поле Маркова (MRF) - неориентированный граф, состоящий из независимых переменных, в качестве которых принято использовать пиксели левого изображения. Рёбра в таком случае отражают зависимости между связываемыми ими пикселями. Таким образом, граф выглядит как сетка, в которой

переменные (пиксели) зависят только от своих соседей (локальное свойство Маркова), и задачу стереозрения можно переформулировать и решать в терминах MRF. Тогда каждая переменная может принимать значения, соответствующие диапазону расхождений, а каждое рёбро - пропускную способность, зависящую от значений соединяемых им пикселей. Для решения задачи стереозрения могут использоваться известные алгоритмы минимизации энергии на случайных полях Маркова. Например, карта расхождений по этому методу находится как плоскость, формирующая разрез представленного графа на две части с минимальной суммарной стоимостью входящих в него рёбер. Особенности построения и математическая формализация этих структур приведены в работах [31,21].

Очевидно, что классические методы, основанные на алгоритме разреза графа, требуют даже больше вычислительных операций, чем динамическое программирование. Были предприняты попытки ускорить метод [4,31], которые приблизили его по эффективности к динамическому программированию. Помимо вычислительных ресурсов алгоритм так же довольно требователен к памяти (до 150 раз больше объёма одного изображения), что осложняет его использование с большими изображениями, но и эта проблема решается применением более эффективных структур данных [11]. Однако такая модификация алгоритма показывает результаты в среднем лишь на 1.6% лучше, чем ДП. Он также был реализован на ПЛИС, что позволило добиться существенного увеличения скорости относительно классического метода на ЦПУ, но пока с сильным дисбалансом по использованию ресурсов вычислителя. [19]. В 2018 году была предложена версия алгоритма, достигшая производительности в 6 кадров в секунду [10]. При этом использовался вычислитель NVIDIA Jetson TX2, часто применяемый в мобильной робототехнике. Тем не менее, современные алгоритмы разреза графа являются одними из самых медленных, и самых точных, алгоритмов стереозрения.

2.5 Алгоритмы на нейронных сетях

Рост доступного объёма памяти в графических ускорителях и популярности алгоритмов, построенных с использованием нейросетей, а также возникновение качественных датасетов позволили реализовать надёжные и быстрые способы получения карт глубины, опирающиеся на глубокое обучение [16]. В настоящий момент существенная доля самых точных алгоритмов в популярном бенчмарке Middlebury Stereo Evaluation [32] представлена именно подобными решениями.

При этом большинство алгоритмов реализуют один из двух основных подходов. Первый заключается в том, что свёрточные нейронные сети используются для выполнения лишь одного этапа из более комплексного алгоритма стереозрения, например, только поиск соответствий в эпиполярной линии [37]. Более современные работы экспериментируют с архитектурой сети, добиваясь, например, подсчёта стоимости сопоставления для каждого возможного расхождения аналогично глобальным методам [33]. К этому же подходу можно отнести использование нейронных сетей для постобработки

карт глубины [13]. Второй подход подразумевает решение всей задачи получения карты глубины с помощью глубокого обучения без необходимости в пост-обработке. Обучение и дизайн сети при таком подходе обычно проще, а алгоритмы получаются более быстрыми (относительно первого подхода). DispNetC [26], например, использует подход, который ранее применялся для решения задачи вычисления оптического потока (тоже заключается в поиске соответствий, но обычно по двум последовательным снимкам одной камеры), которую можно считать более общей, так как расхождения могут быть найдены по любому направлению, а не только в рамках одного ряда. Наложение дополнительных ограничений позволило получить точный и относительно быстрый метод подсчёта карт глубины, способный на производительность в реальном времени при использовании мощного графического ускорителя. GC-Net [20] впервые использовала модуль 3D-свёртки, который позже стал основой для целого класса объёмных методов. Текущий лидер бенчмарка и датасета KITTI 2015 [27] LEAStereo [6] тоже является объёмным.

Таким образом, современные алгоритмы на глубоком обучении позволяют оценивать глубину по паре изображений с высокой точностью и производительностью в реальном времени для некоторых алгоритмов. Тем не менее их использование затруднено необходимостью использования современных графических ускорителей с большим энергопотреблением.

3 Выводы

Задача 3D-реконструкции остаётся активным полем работы для исследователей, занимающихся техническим зрением. В этой работе были представлены основные алгоритмы, применяемые на данный момент. По изученной литературе видно, что за последние два десятилетия фокус сместился с фундаментальных основ стереозрения к более частным исследованиям, направленным на повышение качества и скорости получения результатов, а также проблемным областям вроде обработки заслонений и слаботекстурированных регионов. Использование нейросетей для данной задачи также привело к появлению нового весьма перспективного класса алгоритмов, который, вероятно, будет становиться только популярнее. В настоящий момент присутствует запрос на эффективные методы стереозрения, способные работать в реальном времени и на мобильных платформах, удовлетворению этого запроса, вероятно, и будут посвящены дальнейшие исследования на эту тему.

Список литературы

1. Вахитов, ., Гуревич, ., Павленко, .: Обзор алгоритмов стереозрения. Стохастическая оптимизация в информатике **4**(1) (2008)
2. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf). *Computer Vision and Image Understanding* **110**(3), 346–359 (2008). <https://doi.org/https://doi.org/10.1016/j.cviu.2007.09.014>, similarity Matching in Computer Vision and Multimedia
3. Bobick, A., Intille, S.: Large occlusion stereo. *International Journal of Computer Vision* **33**, 181–200 (09 1999). <https://doi.org/10.1023/A:1008150329890>
4. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence* **23**(11), 1222–1239 (2001)
5. Chang, Q., Maruyama, T.: Real-time stereo vision system: A multi-block matching on gpu. *IEEE Access* **6**, 42030–42046 (2018). <https://doi.org/10.1109/ACCESS.2018.2859445>
6. Cheng, X., Zhong, Y., Harandi, M., Dai, Y., Chang, X., Drummond, T., Li, H., Ge, Z.: Hierarchical neural architecture search for deep stereo matching (2020)
7. Dong, B., Zeng, F., Pan, B.: A simple and practical single-camera stereo-digital image correlation using a color camera and x-cube prism. *Sensors(Basel)* **19**(21) (2019)
8. Einecke, N., Eggert, J.: A multi-block-matching approach for stereo. In: 2015 IEEE Intelligent Vehicles Symposium (IV). pp. 585–592. IEEE (2015)
9. Ekstrand, F., Ahlberg, C., Ekström, M., Spampinato, G.: High-speed segmentation-driven high-resolution matching. In: Seventh International Conference on Machine Vision (ICMV 2014). pp. 944–51Y (2015)
10. Fan, R., Jiao, J., Pan, J., Huang, H., Shen, S., Liu, M.: Real-time dense stereo embedded in a uav for road inspection. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 535–543 (2019). <https://doi.org/10.1109/CVPRW.2019.00079>
11. Fusiello, A., Roberto, V., Trucco, E.: Efficient stereo with multiple windowing. In: Proceedings of IEEE Computer Society conference on computer vision and pattern recognition. pp. 858–863. IEEE (1997)
12. Gao, W., Shen, S.: Dual-fisheye omnidirectional stereo. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 6715–6722 (2017). <https://doi.org/10.1109/IROS.2017.8206587>
13. Gidaris, S., Komodakis, N.: Detect, replace, refine: Deep structured prediction for pixel wise labeling. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5248–5257 (2017)
14. Gupta, R.K., Cho, S.Y.: Window-based approach for fast stereo correspondence. *IET Computer Vision* **7**(2), 123–134 (2013)
15. Hallek, M., Boukamcha, H., Smach, F., Atri, M.: Real time stereo matching using two step zero-mean sad and dynamic programing. In: 2018 15th International Multi-Conference on Systems, Signals Devices (SSD). pp. 1234–1240 (2018). <https://doi.org/10.1109/SSD.2018.8570471>
16. Hamid, M.S., Manap, N.A., Hamzah, R.A., Kadmin, A.F.: Stereo matching algorithm based on deep learning: A survey. *Journal of King Saud University - Computer and Information Sciences* (2020). <https://doi.org/https://doi.org/10.1016/j.jksuci.2020.08.011>, <https://www.sciencedirect.com/science/article/pii/S1319157820304493>

17. Hartley, R.I., Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edn. (2004)
18. Hirschmuller, H., Scharstein, D.: Evaluation of cost functions for stereo matching. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8 (2007)
19. Kamasaka, R., Shibata, Y., Oguri, K.: An fpga-oriented graph cut algorithm for accelerating stereo vision. In: 2018 International Conference on ReConFigurable Computing and FPGAs (ReConFig). pp. 1–6 (2018). <https://doi.org/10.1109/RECONFIG.2018.8641737>
20. Kendall, A., Martirosyan, H., Dasgupta, S., Henry, P., Kennedy, R., Bachrach, A., Bry, A.: End-to-end learning of geometry and context for deep stereo regression. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 66–75 (2017)
21. Kolmogorov, V., Monasse, P., Tan, P.: Kolmogorov and Zabih's Graph Cuts Stereo Matching Algorithm. *Image Processing On Line* **4**, 220–251 (2014), <https://doi.org/10.5201/ipol.2014.97>
22. Larabi, S., Baha, N., Touzene, H.: Fpga implementation for stereo matching algorithm. In: 2013 Science and Information Conference. pp. 448–454 (2013)
23. Lipson, L., Teed, Z., Deng, J.: RAFT-Stereo: Multilevel Recurrent Field Transforms for Stereo Matching. arXiv preprint arXiv:2109.07547 (2021)
24. Liu, J., Sang, X., Jia, C., Guo, N., Liu, Y., Shi, G.: Efficient stereo matching algorithm with edge-detecting. In: Optoelectronic Imaging and Multimedia Technology III. pp. 927–335 (2014)
25. Lowe, D.G.: Distinctive image features from scale-invariant keypoints (2003)
26. Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4040–4048 (2016)
27. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
28. Ohta, Y., Kanade, T.: Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-7**(2), 139–154 (1985). <https://doi.org/10.1109/TPAMI.1985.4767639>
29. Pan, B., Yu, L., Zhang, Q.: Review of single-camera stereo-digital image correlation techniques for full-field 3d shape and deformation measurement. *Science China Technological Sciences* **61**, 1–19 (09 2017). <https://doi.org/10.1007/s11431-017-9090-x>
30. Roxas, M., Oishi, T.: Real-time variational fisheye stereo without rectification and undistortion (2019)
31. Roy, S., Cox, I.: A maximum-flow formulation of the n-camera stereo correspondence problem. In: Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271). pp. 492–499 (1998). <https://doi.org/10.1109/ICCV.1998.710763>
32. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision* **47**, 7–42 (04 2002). <https://doi.org/10.1023/A:1014573219977>
33. Shaked, A., Wolf, L.: Improved stereo matching with constant highway networks and reflective confidence learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4641–4650 (2017)

34. Shan, Y., Wang, Z., Wang, W., Hao, Y., Wang, Y., Tsoi, K., Luk, W., Yang, H.: Fpga based memory efficient high resolution stereo vision system for video tolling. In: 2012 International Conference on Field-Programmable Technology. pp. 29–32 (2012). <https://doi.org/10.1109/FPT.2012.6412106>
35. Sun, J., Li, Y., Kang, S., Shum, H.Y.: Symmetric stereo matching for occlusion handling. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). pp. 399–406 (2005)
36. Ustukov, D.I., Muratov, Y.R., Nikiforov, M.B., Melnik, O.V.: Analysis of the efficiency of dense stereovision algorithm implementation on different computer architectures. In: 2017 2nd International Conference on Computer and Communication Systems (ICCCS). pp. 78–81 (2017). <https://doi.org/10.1109/CCOMS.2017.8075271>
37. Zbontar, J., LeCun, Y.: Computing the stereo matching cost with a convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1592–1599 (2015)