

Санкт-Петербургский Политехнический Университет Петра Великого
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Название предмета

Отчет по лабораторной работе №1

Тема работы

Работу выполнил:

Петров В.Д.

Группа: 43501/4

Преподаватель:

Ицыксон В.М.

Санкт-Петербург
2021

Содержание

1	Введение	2
1.1	Принцип стереозрения	2
2	Восстановление	3
3	Поиск соответствий	3
3.1	Block matching	3
3.2	Feature matching	4
3.3	Градиенты	4
3.4	Методы динамического программирования	4
3.5	Graph cut	5
4	Получение изображений	5
4.1	Стерео одной камерой	5
4.2	Fisheye-stereo	5
5	Выводы	5

1 Введение

Получение трёхмерной структуры пространства по стереоснимкам - это задача, первые решения которой были получены десятилетия назад. Ранние работы фокусировались в основном на способах поиска соответствий и геометрических основах, лежащих в основе процесса. Существенный объём научной работы продолжает производиться в области стереозрения и по сей день. Был достигнут заметный прогресс в повышении точности результатов и понижении вычислительных мощностей, требуемых для их достижения, однако эти области остаются фокусом исследований.

Улучшение точности и производительности алгоритмов является нетривиальной задачей. На точность полученных результатов оказывает влияние нехватка информации, вызванная заслонением объектов, наличием наклонных плоскостей и другими факторами, влияющими на сложность восстановления трёхмерных объектов. Разрешение сенсоров также растёт с каждым годом, увеличивая вычислительную сложность поиска соответствий на кадрах с каждой камеры. Таким образом, исследователи в области стереозрения пытаются найти компромисс между этими двумя характеристиками. Однако для каждого конкретного алгоритма этот компромисс может быть смещён в ту или иную сторону.

Первые работы (1970-1980гг) выполнялись преимущественно в ...

Остальная часть статьи организована следующим образом: ...

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. Существуют способы [9] решения этой задачи с использованием лишь одной камеры совместно с системой линз и зеркал, но принцип их функционирования по своей сути симулирует двухкамерную реализацию.

Как правило, система стереозрения состоит из двух камер, наблюдающих сцену с разных точек, как изображено на рисунке 1 [4]. 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.

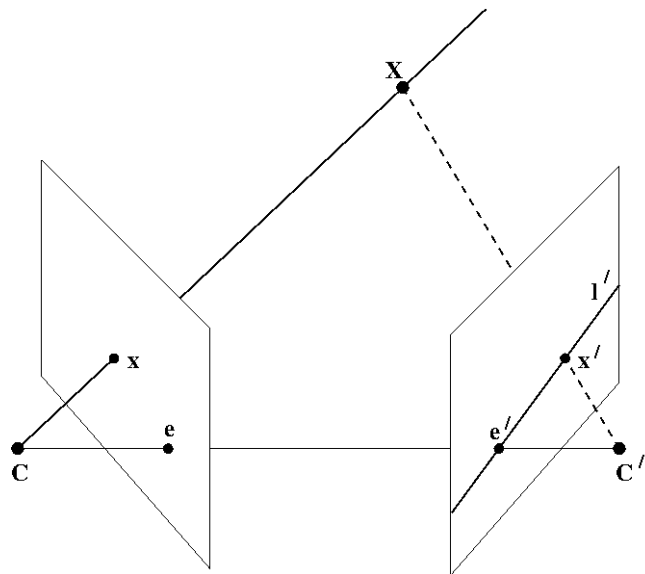


Рис. 1: Эпиполярная геометрия

Таким образом, соблюдение указанных выше требований позволяет использовать следующий геометрический принцип. Пусть имеются две камеры, как изображено на рисунке 1. 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 эпиполярной линией. Образ точки X на плоскости изображения второй камеры обязательно лежит на эпиполярной линии l' . Таким образом, каждой точке x на изображении левой камеры соответствует эпиполярная линия l' на изображении правой камеры. При этом пара для x на изображении правой камеры может лежать только на соответствующей эпиполярной линии. Аналогично, каждой точке x' на правом изображении соответствует эпиполярная 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 либо к облаку точек, либо к карте глубины. Пример такой карты представлен на рисунке 2 [6].



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.

2 Восстановление

Вообще восстановление из диспарити карты просто считается через фокус. Мб тут рассказать про методы повышения качества диспарити карт или борьбы с перекрытиями. Или про методы, не основанные на расхождениях.

3 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 ограничений на зону поиска соответствий. При подсчёте расхождений лишь в небольшой окрестности или окне вокруг интересующего нас пикселя мы говорим о локальных методах, эти методы используют небольшое количество информации и относительно быстродействены. Локальные методы в свою очередь делятся на три категории: block matching (сопоставление блоков?), градиентные методы и сопоставление особенностей (feature matching). //TODO: вставить правильный перевод

Глобальные методы же проводят поиск в целом ряде пикселей или во всём изображении. Они не так чувствительны к локальным дефектам, мешающим процессу поиска соответствий (например, заслонение), но при этом имеют куда большую вычислительную сложность. Одним из самых популярных подходов в этой группе методов является динамическое программирование, которое позволяет разбить вычислительно сложную задачу на множество простых подзадач [8].

3.1 Block matching

БМ - 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 их интенсивности:

$$AD(x, y, d) = |I_l(x, y) - I_r(x + d, y)|, \quad (3.1)$$

где (x, y) - координаты первого пикселя; d - сдвиг по оси x между двумя пикселями или диспаратет; 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 [1].

Для квадратичной разности

$$SD(x, y, d) = |I_l(x, y) - I_r(x + d, y)|^2. \quad (3.2)$$

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. Благодаря всё ещё достаточно высокой скорости работы эти метрики позволяют повышать качество итоговой карты смещений за счёт варьирования размера окна и дополнительных проходов по изображениям [3].

Также применяется нормированная кросс-корреляция (NCC) - стандартный статистический метод для поиска соответствий с шаблоном, который записывается для случая поиска соответствий пикселей согласно уравнению 3.3.

$$NCC(x, y) = \frac{\sum_{x,y} (I_l(x, y) - \bar{I}_l)^2 * (I_r(x + d, y) - \bar{I}_r)^2}{\sqrt{\sum_{x,y} (I_l(x, y) - \bar{I}_l)^2 * (I_r(x + d, y) - \bar{I}_r)^2}}, \quad (3.3)$$

где \bar{I}_l и \bar{I}_r - средние интенсивности соответствующих изображений. Эта метрика устойчива к перепадам яркости и контрастности изображения благодаря нормализации [5], но требует выполнения значительно большего числа арифметических операций.

3.2 Feature matching

Довольно распространёнными являются методы, основанные на использовании характерных точек - методы SIFT [1], SURF [2] и методы, основанные на детекторах Харриса и других. Различным способам поиска этих 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.

Новые работы в этой области продолжают публиковаться. Так был предложен метод на основе SIFT, использующий самоорганизующиеся карты для достижения более высокой производительности [7]. Работа Liu и других [2] 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.

Метод менее чувствителен к перекрытиям и слаботекстурированным областям, но, к сожалению, плотность точек, для которых возможно подсчитать глубину, получается относительно низкой, что привело к снижению интереса к этой группе методов.

3.3 Градиенты

3.4 Методы динамического программирования

Методы, основанные на динамическом программировании, являются одними из самых часто используемых среди глобальных подходов. В случае с поиском совпадений на изображениях это означает введение новой 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 разность между совпадающими пикселями записывается в DSI, который в итоге содержит в себе разницы каждого пикселя с каждым для всех попарных рядов исходных изображений. Строки DSI, соответствующие заведомо невозможным значениям (например, расхождения больше максимально допустимых, пиксель на левом изображении находится левее соответствующего ему на правом) отбрасываются. Рисунок 3 [1] иллюстрирует этот принцип.

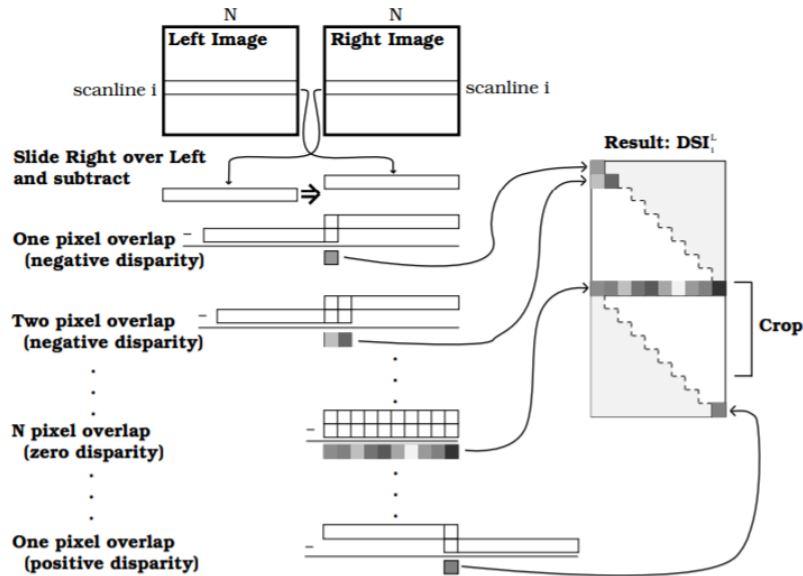


Рис. 3: Процесс построения DSI. Строка левого изображения считается неподвижной, пока строка правого изображения смещается вдоль неё. Результат вычитания перекрывающихся пикселей складывается в очередную строку DSI. Сам DSI обрезается с учётом ограничений.

Далее задача состоит в поиске оптимального пути на полученной матрице. Для каждого элемента матрицы возможны три типа "движения": горизонтальный (означающий совпадение), вертикальный и диагональный (оба соответствуют заслонённым областям). Тем не менее, количество путей, проходящих через каждую точку даже с этими ограничениями остаётся довольно большим. Как уже упоминалось, динамическое программирование уменьшает вычислительную сложность задачи автоматизации за счёт разбиения её на множество более простых и мелких задач. При применении к этой задаче это означает присвоение каждому элементу DSI стоимости, зависящей от доступного типа движения, и поиск оптимального пути от левого верхнего угла матрицы до правого нижнего. Поиск при этом опирается на расставленные ранее стоимости для выбора следующего шага пути.

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 $O(n^2)$) и возможность распространения локальной ошибки на последующие пиксели, что приводит к характерным горизонтальным полосам на картах глубины.

Высокий интерес к этому методу привёл к появлению большого числа работ, пытающихся устранить недостатки метода и повысить его эффективность. Так была предложена модель [10], которая опирается на ограничение видимости - предположение, что каждый заслонённый пиксель не имеет соответствия на втором изображении, а незаслонённый - обязательно имеет.

3.5 Graph cut

4 Получение изображений

4.1 Стерео одной камерой

4.2 Fisheye-stereo

5 Выводы

Список литературы

- [1] Александр Вахитов, Лев Гуревич, and Дмитрий Павленко. Обзор алгоритмов стереозрения. *Стохастическая оптимизация в информатике*, 4(1), 2008.
- [2] Fredrik Ekstrand, Carl Ahlberg, Mikael Ekström, and Giacomo Spampinato. High-speed segmentation-driven high-resolution matching. In *Seventh International Conference on Machine Vision (ICMV 2014)*, pages 944–51Y, 2015.
- [3] Raj Kumar Gupta and Siu-Yeung Cho. Window-based approach for fast stereo correspondence. *IET Computer Vision*, 7(2):123–134, 2013.
- [4] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
- [5] Heiko Hirschmuller and Daniel Scharstein. Evaluation of cost functions for stereo matching. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.
- [6] Lahav Lipson, Zachary Teed, and Jia Deng. RAFT-Stereo: Multilevel Recurrent Field Transforms for Stereo Matching. *arXiv preprint arXiv:2109.07547*, 2021.
- [7] Jing Liu, Xinzhu Sang, Changxin Jia, Nan Guo, Yangdong Liu, and Guozhong Shi. Efficient stereo matching algorithm with edge-detecting. In *Optoelectronic Imaging and Multimedia Technology III*, pages 927–335, 2014.
- [8] Yuichi Ohta and Takeo Kanade. Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(2):139–154, 1985.
- [9] Bing Pan, Liping Yu, and Qianbing Zhang. 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.
- [10] Jian Sun, Yin Li, S.B. Kang, and Heung-Yeung Shum. Symmetric stereo matching for occlusion handling. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pages 399–406, 2005.