Санкт-Петербургский Государственный Университет

Математико-Механический факультет

Кафедра системного программирования

Реализация алгоритма Semi-Global Matching

Выполнил студент 445 гр. Руслан Мокаев

научный руководитель: Александр Пименов

Санкт-Петербург
2012

Оглавление

1. Введение

Задача восстановления глубины по нескольким изображениям, сделанным одновременно, является крайне важной и необходимой для многих приложений, например для тех, задачей которых является восстановление 3D-модели объекта по нескольким его фотографиям, снятым с разных ракурсов. Эта фундаментальная проблема была и остается предметом многолетних исследований. Для решения этой задачи существует целый ряд алгоритмов, которые можно разделить на 2 категории:

- **Локальные походы**. Вычислительная сложность невысока, но результат работы может содержать ошибки, которые могут оказаться неприемлемыми для некоторых приложений
- **Глобальные подходы**. Результаты, как правило, очень хорошие, однако скорость выполнения невысока (использовать их в реальном времени практически невозможно)

Алгоритм Semi-Global Matching дает практически такие же результаты, как и глобальные алгоритмы, однако скорость его работы гораздо выше.

2. Формальное задание проблемы и предобработка данных

Дадим основные определения и сформулируем понятия, возникающие при решении задачи восстановления глубины. Рассмотрим весь процесс: от получения входных данных (изображения с фото- или видео-камер) до получения конечного результата.

Для простоты будем считать, что у нас есть две камеры (C1 и C2), снимающие одну и ту же сцену в один и тот же момент времени; съемку они производят с разных точек обзора. В результате съемки мы получаем два изображения одной и той же сцены (рис.1):

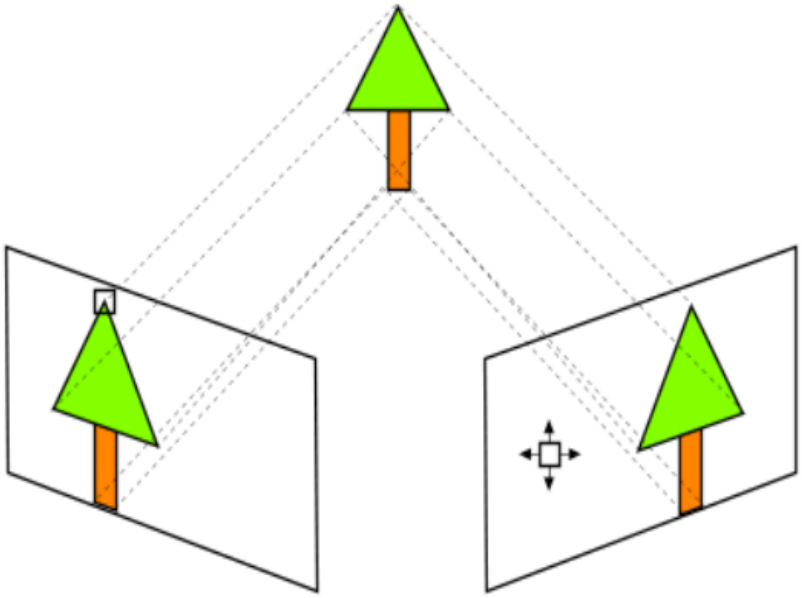

Рис.1. Изображения, получаемые в результате фото-видео съемки

Далее для каждой точки левого изображения необходимо найти соответствующую точку на правом изображении. Эта проблема называется проблемой соответствия. Как видно из рис.1 пространством поиска соответствующей точки левого изображения является всё правое изображение; таким образом поиск производится в двух измерениях. Хотелось бы неким образом выровнять оба изображения так, чтобы пространство поиска сузилось до одного измерения (до одной прямой) и таким образом затраты на решение проблемы соответствия были бы сокращены. Процесс коррекции нескольких изображений, после которого они имеют общую поверхность изображения (image surface), называется ректификацией. После ректификации, изображения на рис.1 станут выглядеть так :

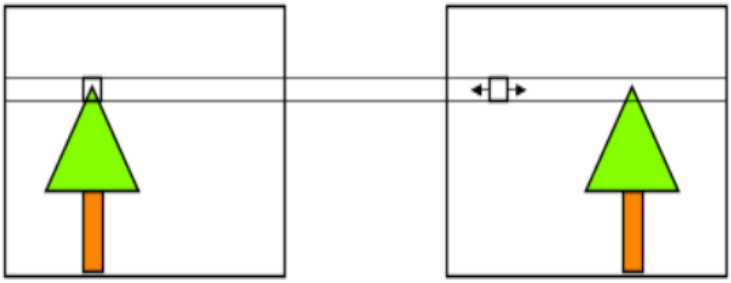

Рис.2. Изображения после ректификации. Пространство поиско сужено до одной прямой

Ректификация позволяет сузить пространство поиска решений для проблемы соответствия до одного измерения. Другими словами: после проведения ректификации, все пиксели правого изображения, которые могут соответствовать данной точке левого изображения, будут находиться в одной сроке. Пара ректифицированных изображений и является входными параметрами в задаче восстановления глубины.

Следующим шагом является решение проблемы соответствия, а именно, построение карты диспаратности (disparity map) по двум ректифицированным изображениям. Рассмотрим следующий пример для того, чтобы понять, что такое карта диспаратности:

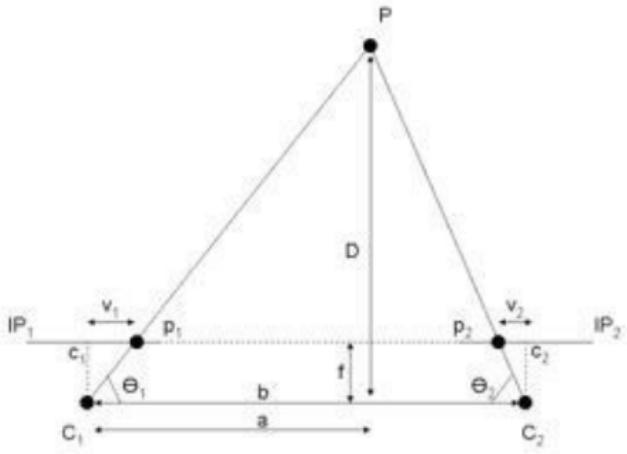

Рис.3. Геометрическое представление наблюдаемой сцены

C1 и C2 – это центры камер. IP1 и IP2 – это плоскости изображения, то есть по сути фотографии. P – наблюдаемая точка, а P1 и P2 – это перспективные проекции на соответствующую плоскость изображения. Расстояние между камерами – b. Наша задача – найти D, то есть расстояние до точки в реальном мире. Для этого мы посчитаем диспаратность для точки P, которая равна разнице между V1 и V2, где V1(2) - это смещение точки P1(2) относительно центра первой (второй) камеры, то есть точки c1(2). Диспаратность можно понимать по-другому, как смещение в пикселях точки P2 относительно P1. После

того, как в каждой точке посчитана диспаратность, нахождение реальной глубины не является сложной задачей. Пусть f – фокусное расстояние камеры, тогда $D = b \cdot f / d$. Причем, поскольку фокусное расстояние и расстояние между камерами остаются постоянными для всех точек на фотографии, то диспаратность можно использовать как относительную глубину точек. Зная диспаратность в каждой точке, мы можем построить карту диспаратности, например :

Рис.4. Фотографии Пентагона и карта диспаратности (крайняя правая)

Сформулируем нашу проблему более строго. Будем считать, что на вход нам подается два ректифицированных изображения; I_{bp} = {I(p), где p — пиксель в базовом изображении (base image)}, I_{mq} = {I(q), где q — пиксель в парном изображении (match image)}, где функция I — функция, сопоставляющая каждому пикселю изображения его интенсивность. Наша задача состоит в построении карты диспаратности f = {D_p | p — пиксель базового изображения}, где каждое D_p представляет соответствие между пикселем p = (p_x, p_y) из базового изображения и пикселем (p_x - D_p , p_y) парного изображения. Как видно, в силу ректифицированности базового и парного изображения, D_p можно рассматривать как скаляр, а не как двумерный вектор.

Построение карты диспаратности – сложная вычислительная задача. Кроме того, проблема соответствия была и остается нерешенной полностью проблемой. Это связано со следующими причинами, в соответствии с [3]

- **Шум**. Если мы говорим о съемке в реальных, а не лабораторных условиях, то на получаемых фотографиях практически всегда будут присутствовать такие неизбежные факторы: световые блики, размытие изображения и просто шум, полученный из-за несовершенных сенсоров. Из этого следует, что алгоритмы, решающие проблему соответствия, должны быть устойчивыми.

- **Однотонные области**. Эта проблема также получила название проблема апертуры. Информация, получаемая с крайне текстурированных областей, должна быть распространена на однотонные области
- **Разрывность глубины**. Многие алгоритмы требуют для своей работы того, чтобы на некоторой области глубина была непрерывной. Это создает проблемы, когда в нужную область попадают границы объектов, то есть места, где глубина терпит разрыв.
- **Заслоненные области**. Те пиксели, которые являются заслоненными на одном изображении, не должны быть поставлены в соответствие пикселям из другого изображения.

Несмотря на все перечисленные проблемы, существуют различные методы, которые позволяют с ними бороться (этим методам посвящена следующая часть).

3. Обзор существующих решений

Принято различать два достаточно широких класса алгоритмов для решения проблемы соответствия: локальные и глобальные алгоритмы. В сумме эти два подхода объединяют большинство из существующих решений. В соответствии с [4]: большинство решений проблемы соответствия выполняют, частично или полностью, следующие шаги

1. Вычисление стоимостей соответствий.
2. Суммирование стоимостей.
3. Вычисление диспаратностей на основе вычисленных стоимостей.
4. Улучшение карты диспаратности.

Реальная последовательность шагов зависит от конкретного алгоритма.

3.1. Локальные алгоритмы.

Этот класс алгоритмов подсчитывает диспаратность каждого пикселя в отдельности, используя окна фиксированного или адаптируемого размера для корреляции. Выбор формы окна и его размеров является непростой задачей. Это связано с тем, что, с одной стороны, корреляция предполагает, что глубина всех пикселей внутри окна не терпит разрывов. Это значит, что увеличение размеров окна ведет к нарушению этого условия и, как следствие, к некорректной работе алгоритма. С другой стороны, уменьшение размеров окна ведет к увеличению влияния шума, что приводит к уменьшению корректных совпадений. Для локальных алгоритмов стоимость соответствия (т.е. первый шаг работы) определяется как схожесть между двумя областями, одна из которых находится в базовом изображении, а другая в парном. Форма этих областей зависит от конкретного алгоритма. Стандартные подходы, появившиеся на заре исследований проблемы соответствия, используют прямоугольное окно фиксированного размера. Размеры этого окна определяются опытным путем. Объем вычислений, в этом случае, сильно сокращается по сравнению с современными методами (о них позже), но существует ряд проблем, избежать которых, используя фиксированный размер окна, практически невозможно. Для того чтобы понять, почему фиксированный размер окна порождает проблемы, мы рассмотрим, как, в общем, работают локальные алгоритмы.

Рассмотрим произвольный пиксель базового изображения, вокруг него строится наше окно. Для данного пикселя базового изображения существует ряд подозрительных на соответствие пикселей парного изображения. Вокруг каждого такого пикселя строится такое

же окно. Подобная ситуация показана на рис.5

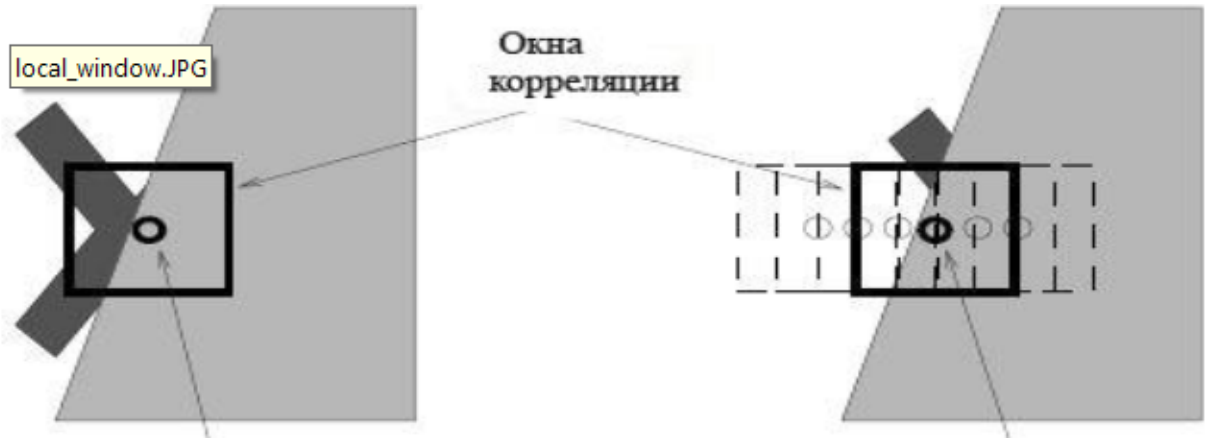

Рис.5. Рассматриваемый пиксель (слева) и соответствующий пиксель (справа)

Возможные позиции пикселей в парном изображении определены минимально допустимым расстоянием между камерами и объектом, что дает нам максимальную диспаратность. Для каждой возможной пары окон подсчитывается стоимость соответствия. Наиболее распространенными стоимостями являются *квадрат разности интенсивностей* (squared intensity differences , SD; Hannah, 1974; Anandan, 1989; Matthies et al., 1989; Simoncelli et al., 1991) и *абсолютная разность интенсивностей* (absolute intensity differences , AD; Kanade, 1994). Пара окон с наибольшей или наименьшей, в зависимости от алгоритма вычисления, стоимостью и определяют пиксель соответствующий данному. Тем не менее, если в окне нарушается условие на непрерывность глубины, то его часть будет влиять на результирующую стоимость неопределенным образом, в соответствии с [5]. Вернемся к рисунку 5. Левая часть корреляционного окна, как в базовом изображении, так и в парном, содержит фон. Но этот фон разный для обоих окон. Более того, стоит отметить, что левая часть окна в левом изображении, по-крайней мере частично, заслонена на правом битмапе. Размеры заслоненной части зависят от разности диспаратностей и размеров окна. Все это приводит нас к тому, что та часть окна, которая содержит фон, вносит ошибки в подсчет стоимостей. Возможным решением проблемы будет уменьшение размеров окна, но, как уже отмечалось, это приведет к возрастанию влияния шума при подсчете стоимости.

Как было показано выше, фиксированный размер окна не дает необходимой точности при решении проблемы соответствия. Kanade в [6] предложил решение этой проблемы путем изменения размеров и формы окна в зависимости от локальных характеристик диспаратности. Этот алгоритм позволил уменьшить число ошибок, возникающих на границах объектов. Тем не менее, данный алгоритм является слишком медленным для

выполнения в реальном времени на неспециализированном оборудовании, в соответствии с [4].

Адаптацией подхода, предложенного Kanade в [6], является многооконный алгоритм, разработанный Fusiello в [7]. Стандартная конфигурация состоит из 9 окон одного размера, но расположенных в разных местах относительно рассматриваемого пикселя. Корреляция производится для каждого из 9 окон в отдельности, но только результат, показанный «лучшим» окном, используется. Данный алгоритм показывает лучшие результаты в сравнении со стандартной корреляцией и его эффективность достаточна для использования в реальном времени. Тем не менее, по сравнению с глобальными подходами, данное решение дает неточные результаты.

Boykov в [8] разработал алгоритм, работающий следующим образом: для каждого пикселя метод выбирает окно произвольной формы. Эта форма меняется от пикселя к пикселю. Несмотря на возможность использовать метод в реальном времени, он дает довольно большой процент ошибок, что было отмечено самими авторами алгоритма.

Объединенный стерео-алгоритм, предложенный Kanade и Zitnick в [4] демонстрирует, что количество ошибок может быть уменьшено значительно, если время исполнения не является важным фактором. Данный алгоритм является слишком медленным для реализации его в реальном времени.

Hirschmuller в [5] создал метод коррекции ошибок на границе объектов. Данная коррекция применяется на этапе постпроцесса и позволяет уменьшить количество ошибок на границах на 50%. Идея алгоритма состоит в использовании нескольких (а именно 5) окон для уменьшения количества ошибок, а также в предоставлении функции, которая определяет неверные совпадения.

Наряду с исследованиями, связанными с формой и размерами окна, существуют различными подходы к измерению стоимости соответствия. Уже были упомянуты два стандартных подхода SD (разность интенсивностей в квадрате)

$$SDC(\mathrm{p},\mathrm{d}) = \sum_{\mathrm{p}\in\mathrm{W}} \left| I_{bp} - I_{mq} \right|^2, q = p - d.$$

И AD (абсолютная разность интенсивностей)

$$ADC(p,d) = \sum_{p\in W} \left| I_{bp} - I_{mq} \right|, q = p - d.$$

В обоих случаях, подсчет стоимости можно сделать очень эффективным. Однако, в случае выбора этих формул для вычисления, неявно делается предположение, что для данного пикселя базового изображения, соответствующий пиксель парного имеет примерно такую же интенсивность. Это значит, что если у двух камер была выставлена разная экспозиция или на фотографии, сделанной одной из камер, появился блик, то эти формулы будут давать неверный результат. Также, оба этих подхода довольно сильно подвержены влиянию шума и случайных выбросов в данных.

Zabih в [9] представил измерение, основанное не на значениях интенсивностей, а на их числовом порядке:

$$\xi(I,P,P') = \begin{cases} 1 & \textit{если } I(P) < I(P') \\ 0 & \textit{в другом случае} \end{cases},$$

$$Err(P,P') = \xi(I_b,P,P') - \xi(I_m,P,P'),$$

$$C(W) = \sum_{P \in W_1} \sum_{P' \in W_2} \left| Err(P,P') \right|,$$

$$R(W) = \sum_{P \in W_1} \left(\sum_{P \in W_2} Err(P,P') \right)^2.$$

Здесь C(W) – это трансформация данных корреляции, а R(W) - упорядочивание данных корреляции. На их основе вычисляется итоговая стоимость соответствия. Такой подход позволяет уменьшить воздействие шума и случайных выбросов данных по сравнению со стандартным подходом. Тем не менее, различия в настройке камер по-прежнему могут сильно влиять на итоговый результат.

Градиентное измерение, представленное Scharstein в [10] является нечувствительным к различиям между настройками в цветопередаче камер.

После того, как для каждого пикселя p в базовом изображении и для каждой возможной диспаратности d, от 0 до максимальной диспаратности, подсчитана стоимость соответствия, локальный алгоритм, обычно, переходит к шагу суммирования стоимостей. Стандартный подход состоит в суммировании или усреднении стоимостей соответствия в некоторой области. Эта область может быть или двумерной, в этом случае мы считаем, что диспаратность фиксирована, или трехмерной, когда диспаратность изменяется наравне с координатами пикселей. Усреднение может производиться с помощью свертки с каким-

нибудь ядром, чаще всего Гауссовым.

Наконец, финальным шагом является вычисление диспаратности. На основе просуммированных или усредненных стоимостей нахождение соответствия становится тривиальной задачей. Диспаратность для данного пикселя p базового изображения – это такое d, на котором достигается минимальное значение стоимости. Такой подход получил название «победитель получает все» (WTA). Проблема, возникающая при таком подходе, состоит в том, что каждой точке выбирается уникальное соответствие в парном изображении, но на самом же деле, одной точке может соответствовать сразу несколько. К сожалению, эта проблема возникает не только у локальных алгоритмов, но и у подавляющего большинства методов, целью которых является построение карты диспаратности.

Просуммировав все вышесказанное про локальные алгоритмы, можно сказать, что существует огромное число локальных методов, решающих проблему соответствия. Тем не менее, если методу удается решить задачу быстро, то это означает большое количество ошибок в вычислениях, что, конечно, не всегда приемлемо. И наоборот, локальные методы решающие проблему с малым числом ошибок, такие как метод Zitnick в [11], не представляется возможным реализовать в реальном времени на современном оборудовании.

3.2. Глобальные алгоритмы

В отличие от локальных алгоритмов, где нахождение диспаратности происходит для каждого пикселя отдельно, целью глобального подхода является поиск наилучшей карты диспаратности для всего изображения сразу. Глобальные методы практически всю работу выполняют на шаге вычисления диспаратностей, часто пропуская шаг суммирования стоимостей соответствия. Чаще всего глобальные алгоритмы решают проблему соответствия путем минимизации функционала энергии.

Допустим, мы хотим найти наилучшую конфигурацию f . Подходящая конфигурация будет минимизировать функционал E – функционал глобальной энергии

$$E(f) = E_{data}(f) + \lambda \cdot E_{smooth}(f)$$

Первое слагаемое показывает: насколько хорошо данная конфигурация согласуется с парой входных изображений. Для того чтобы подсчитать первое слагаемое, используются попиксельные стоимости :

$$E_{data}(f) = \sum C(x, y, f(x, y))$$

Второе слагаемое отвечает за штраф, налагаемый на данную конфигурацию, в случае, когда она нарушает непрерывность диспаратностей. Чаще всего рассматриваются соседние пиксели:

$$E_{smooth}(f) = \sum_{(x,y)} \rho(f(x, y) - f(x+1, y)) + \rho(f(x, y) - f(x, y+1))$$

Здесь, в качестве функции ρ выступает некоторая монотонно- возрастающая функция штрафа. В [12] был представлен другой подход в вычислении E_{smooth}:

$$E_{smooth}(f) = \sum_{(x,y)} \rho_f(f(x, y) - f(x+1, y)) \cdot \rho_I(\|I(x, y) - I(x+1, y)\|) +$$
$$\sum_{(x,y)} \rho_f(f(x, y) - f(x, y+1)) \cdot \rho_I(\|I(x, y) - I(x, y+1)\|)$$

В данном случае, ρ_I - это некоторая монотонно-убывающая функция, которая снижает стоимость в случае областей, где интенсивность меняется значительно от пикселя к пикселю.

После того, как функционал энергии определен, существует целый ряд алгоритмов, позволяющих найти его минимум. Классическими подходами являются: Марковские сети, максимальный поток, graph-cuts. К сожалению, поиск минимума это очень сложная вычислительная задача (поиск минимума $E(f)$ является NP - полной задачей). Одним из методов, призванных решить эту проблему, является динамическое программирование. Вместо поиска минимума для всего изображения сразу, поиск ведется построчно, причем строки обрабатываются независимо друг от друга. Такой подход применяется, например, в [13]. Нахождение минимума для одной строки возможно, в этом случае, за полиномиальное время. Проблемы данного подхода очевидны, при вычислении карты диспаратностей мы совершенно не учитываем связей между пикселями в разных строках, кроме того, этот подход требует, чтобы относительный порядок пикселей в строке был одинаковым для левого и правого изображения, что невозможно выполнить в том случае, когда на сцене находится узкий объект на переднем плане.

Глобальные алгоритмы дают отличные результаты в нахождении карты диспаратностей. Количество ошибок, допускаемых ими, относительно невелико. Тем не

менее, в настоящее время, ни один из предложенных алгоритмов не может быть реализован в реальном времени на современном оборудовании, в соответствии с [4].

4. Semi-Global Matching

Алгоритм SGM (Semi-Global Matching) был предложен Heiko Hirschmuller в 2005 году в статье [1]. Преимущества этого подхода, по сравнению с другими решениями, состоит в том, что, с одной стороны, он дает очень неплохие результаты с точки зрения качества, так как не является уже локальным алгоритмом; с другой стороны, результат достигается за приемлемое время.

Получение карты диспаратности алгоритмом Semi-Global Matching состоит из 3х шагов:

1. Вычисление попиксельной стоимости.
2. Суммирование попиксельных стоимостей.
3. Вычисление карт диспаратности.

Перед выполнением всех этих шагов необходимо вычислить интенсивности пикселей обоих изображений. Далее все вычисления производятся с интенсивностями пикселей.

Рассмотрим каждый из шагов более подробно:

4.1. Вычисление попиксельной стоимости.

В данной работе попиксельная стоимость вычислялась по формуле

$$Cost(p, d) = |I_L(p_x, p_y) — I_R(p_x - d, p_y)|$$

Другой способ вычисления стоимости — способ , предложенной S.Berchfield и C.Tomasi в [14]. Стоимость $C_{BT}(p, d)$ считается следующим образом: для пикселей x_i и y_i базового и парного изображений соответственно, определим следующие значения:

$$I_R^- \equiv \hat{I}_R(y_i - \tfrac{1}{2}) = \tfrac{1}{2}(I_R(y_i) + I_R(y_i - 1))$$
$$I_R^+ \equiv \hat{I}_R(y_i + \tfrac{1}{2}) = \tfrac{1}{2}(I_R(y_i) + I_R(y_i + 1)).$$

где I_L и I_R — какие-то функции интенсивности. Получается, что I_R^- - линейно интерполированная интенсивность между y_i и пикселем слева. Аналогично образом определяется и I_R^+

В итоге получаем функцию:

$$\bar{d}(x_i, y_i, I_L, I_R) = \min_{y_i - \frac{1}{2} \le y \le y_i + \frac{1}{2}} |I_L(x_i) - \hat{I}_R(y)|.$$

Итого, $C_{BT}(p, disp) = d(p, p — disp, I_L, I_R)$. Другой способ, представленный в [14], немного модифицирует первый: определяются значения

$$I_{min} = \min(I_R^-, I_R^+, I_R(y_i))$$

и

$$I_{max} = \max(I_R^-, I_R^+, I_R(y_i)).$$

А d определяется теперь следующим образом:

$$\bar{d}(x_i, y_i, I_L, I_R) = \max\{0, I_L(x_i) - I_{max}, I_{min} - I_L(x_i)\}.$$

4.2. Суммирование попиксельных стоимостей и вычисление карты диспаратности

После того как вычислены попиксельные стоимости, необходимо совершить последний шаг работы алгоритма SGM, а именно, суммирование стоимостей. Как уже упоминалось в 3.2, стандартным подходом для глобальных алгоритмов является нахождение такой конфигурации, которая минимизирует некий функционал энергии вида:

$$E(f) = E_{data}(f) + \lambda \cdot E_{smooth}(f)$$

И как уже упоминалось, такая задача является NP-полной. Классическим оптимизационным подходом является динамическое программирование. В этом случае суммирование происходит вдоль одного направления, однако в этом случае связи между пикселями в разных рядах практически, а часто и вообще, не учитываются. В [1] был предложен новый подход. Суть его заключается в том, что суммирование будет происходить одновременно по всем направлениям. Рассмотрим этот метод более подробно.

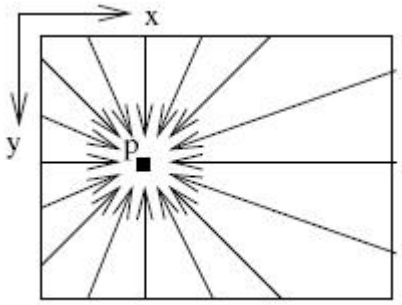

Рис.6. 16 путей для всех направлений

Пусть мы хотим просуммировать стоимости в направлении r для данного пикселя p и диспаратности d. Суммирование происходит рекурсивно:

$$L'_r(p,d) = C(p,d) + \min(L'_r(p-r,d), L'_r(p-r,d-1) + P_1, L'_r(p-r,d+1) + P_1, \min_i(p-r,i) + P_2).$$

Здесь первое слагаемое – это попиксельная стоимость, а второе – это минимум из четырех чисел, которые зависят от суммированной стоимости для предыдущего пикселя в данном направлении. P_1 - это константный штраф, который налагается в том случае, если диспаратности у двух соседних пикселей отличаются на один. P_2 - это константный штраф, который налагается в том случае, когда диспаратности у двух соседних пикселей отличаются больше, чем на один.

Значения L' постоянно растут вдоль пути, что может в итоге привести к большим значениям. Тем не менее может быть модифицированно:

$$L_{\mathbf{r}}(\mathbf{p},d) = C(\mathbf{p},d) + \min(L_{\mathbf{r}}(\mathbf{p}-\mathbf{r},d), L_{\mathbf{r}}(\mathbf{p}-\mathbf{r},d-1) + P_1, L_{\mathbf{r}}(\mathbf{p}-\mathbf{r},d+1) + P_1, \min_i L_{\mathbf{r}}(\mathbf{p}-\mathbf{r},i) + P_2) - \min_k L_{\mathbf{r}}(\mathbf{p}-\mathbf{r},k)$$

Вычитаемое слагаемое константно для всех диспаритетов пикселя p. Таким образом, минимум никак не изменился, а любое значение L теперь не будет превышать $C_{max} + P_2$.

В [1] предлагается производить подсчеты по 16 разным направлениям. В предложенной реализации алгоритма подсчет производится только по восьми направлениям, результат такой оптимизации не сильно ухудшает итоговую карту диспаратностей, скорость же увеличивается в два раза. Так же существует подход, при котором подсчет производится только по пяти направлениям, который также был реализован.

После того, как подсчитаны суммарные стоимости для каждого из направлений, вычисляется общая сумма всех стоимостей для каждого из пикселей:

$$S(p,d) = \sum_r L_r(p,d).$$

Несложно заметить, что если k — количество направлений, то $S \le k \cdot (C_{max} + P_2)$.

Для каждого пикселя базового изображения p, соответствующая ему диспаратность находится как минимум по всем диспаратностям в общем массиве стоимостей:

$$f(p) = \min_d S(p,d).$$

В итоге получаем на выходе карту диспаратности.

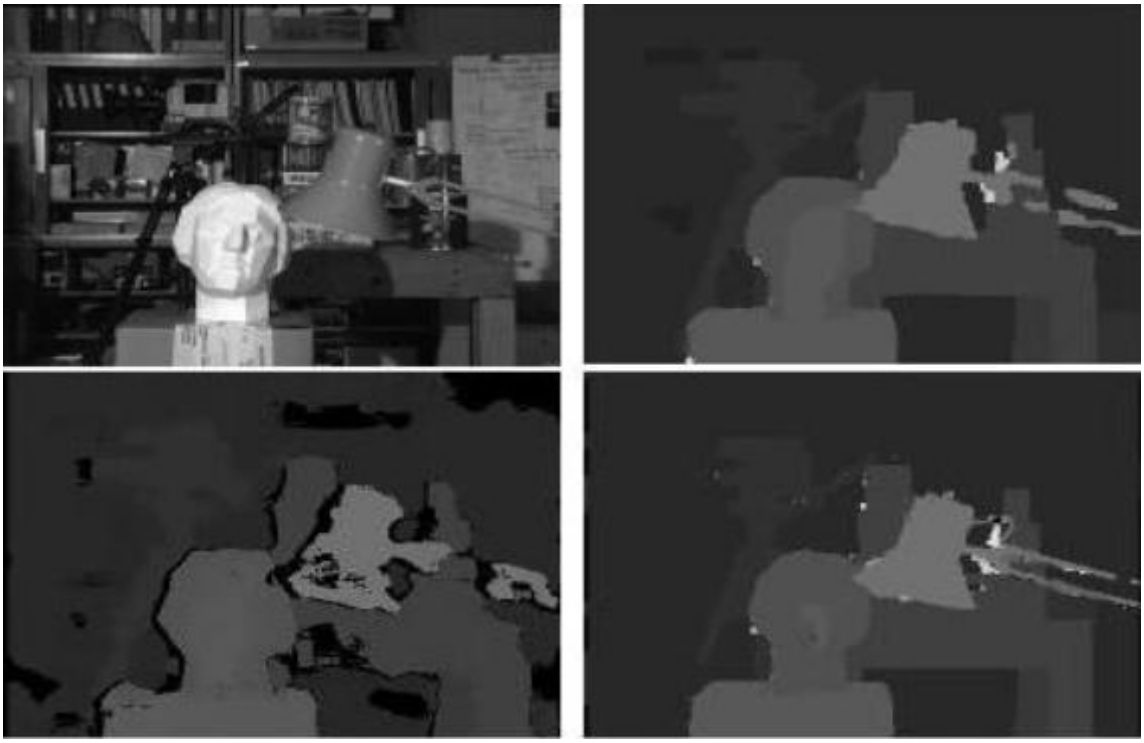

Рис.7. Результаты работы различных алгоритмов

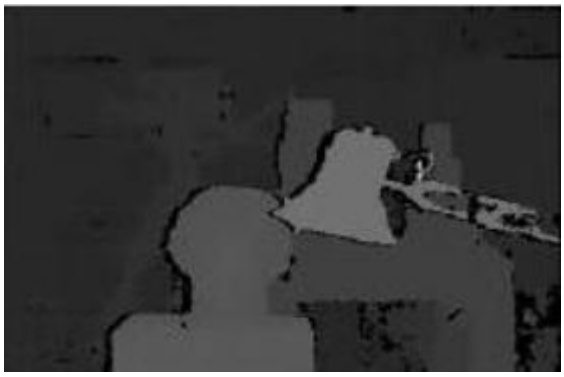

Рис.8. Результат работы алгоритма SGM

На рисунке 7 изображены результаты работы различных алгоритмов: в первом ряду находятся базовое изображение и результат работы алгоритма Belief Propagation из [3]. Во втором ряду — результаты работы локального алгоритма из [6] и алгоритма Graph Cuts. На рисунке 8 изображен результат работы алгоритма SGM.

5. Заключение

В рамках данной работы были достигнуты следующие результаты:

- Изучены различные подходы и алгоритмы для решения проблемы восстановления глубины изображения
- Реализован алгоритм Semi-Global Matching

Список использованной литературы

[1] Heiko Hirschmuller. Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information , 2005

[2] Heiko Hirschmuller. Stereo Vision in Structured Environments by Consistent Semi-Global Matching , 2006

[3] J.Sun, N.Zheng, H.Shum. Stereo Matching Using Belief Propagation. IEEE Pattern Analysis and Machine Intelligence, страницы 787–800, 2003

[4] DANIEL SCHARSTEIN , RICHARD SZELISKI . A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms , 2002.

[5] H.Hirschmuller, P.R.Innocent, J.M.Garibaldi. Real-time correlation-based vision with reduced border errors. International Journal of Computer Vision

[6] T.Kanade, M.Okutomi. A stereo matching algorithm with an adaptive window: theory and experiment. IEEE Transaction on Pattern Analysis and Machine Intelligence, страница 920, 1994

[7] A.Fusiello, V.Roberto, E.Trucco. Efficient stereo with multiple windowing. Proceedings of the Conference on Computer Vision and Pattern Recognition, страницы 858-863, 1997

[8] Y.Boykov, O.Veksler, R.Zabih. A variable window approach to early vision. IEEE Transaction on Pattern Analysis and Machine Intelligence, 1998

[9] R.Zabih, J.Woodfill. Non-parametric local transforms for computing visual correspondence. Proceedings of the Conference on Computer Vision, страницы 151-158, 1994

[10] D.Scharstein. Matching images by comparing their gradient fields. B ICPR, страницы 572-575, 1994

[11] C.Zitnick, T.Kanade. A cooperative algorithm for stereo matching and occlusion detection. Tech.Rep. CMU-RI-TR-99-35, Carnegie Mellon University, 1999

[12] E.Gamble, T.Poggio. Visual integration and detection of discontinuities: the key role of intensity edges. A. I. Memo 970, MIT, 1987

[13] J.J.Cox, S.L.Hingorani, S.B.Rao, B.M.Maggs. A maximum likelihood stereo algorithm. CVIU, страницы 542-567, 1996

[14] Stan Birchfield and Carlo Tomasi . Depth Discontinuities by Pixel-to-Pixel Stereo , 1998