Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование информационных систем

Системное программирование

Овчинников Сергей Андреевич

Развитие метода полуглобального стереосопоставления и его применение к реконструкции поверхностей лиц

Бакалаврская работа

Научный руководитель:
к. ф.-м. н. Вахитов А. Т.

Рецензент:
Кривоконь Д. С.

Санкт-Петербург
2016

SAINT-PETERSBURG STATE UNIVERSITY

Software and Administration of Information Systems

Software Engineering

Sergei Ovchinnikov

Improvement of semiglobal stereomatching algorithm and its application to facial surface reconstruction

Bachelor's Thesis

Scientific supervisor:
Ph. D. Alexandr Vakhitov

Reviewer:
Dmitry Krivokon

Saint-Petersburg
2016

Оглавление

Введение

Нахождение расстояния до различных точек сцены относительно положения камеры - одна из важнейших задач компьютерного зрения. Самый распространенный метод для нахождения глубин точек - использовать две камеры, находящиеся друг от друга на известном расстоянии, и с помощью них получить пару изображений, левое из которых обычно называют источником, а правое - целью (типичная стереосистема приведена на рисунке 1). Описанная проблема важна во многих областях: например, автономное вождение, робототехника, спорт (генерирование промежуточных углов зрения с помощью 2 камер, см. [7]), а также реконструкция сцен и даже лиц людей, например, в целях безопасности, как в [1]. Первым этапом воссоздания полной 3D моде-

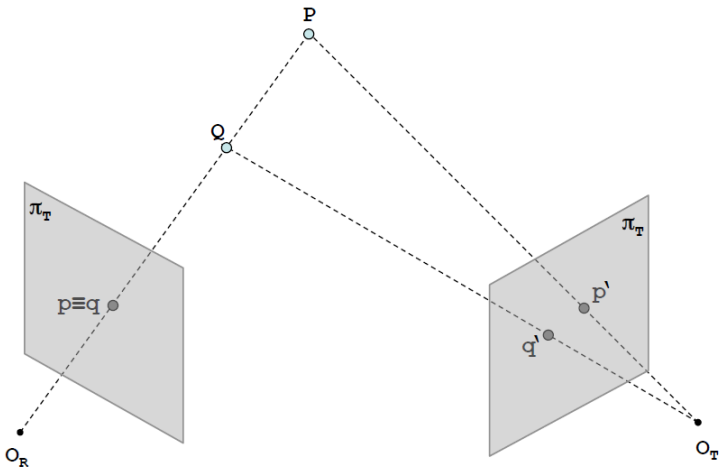

Рис. 1: Схема типичной стереоустановки

ли с помощью данного подхода является решение задачи соответствия. Для того, чтобы воспользоваться дополнительной информацией, которая имеется в виду наличия второй камеры, необходимо знать все соответствия между точками источника и цели. Следует также учитывать две основные проблемы, которые встают на этом пути - это перекрытия и перепады глубин на краях объектов. Объект, видимый с одной камеры, может быть недосягаем для другой в виду перекрытий.

Оказывается, при известных параметрах камер (обычно сами камеры идентичны) и расстоянии между ними задачу можно свести из

$2D \times 2D$ к $1D \times 1D$. Рисунок 1 показывает, как одной точке проекции на источник могут соответствовать две точки цели (на самом деле ей может соответствовать целая линия цели, называемая эпиполярной, $e2$ на рис. 2). Если же теперь спроецировать $e2$ обратно на плоскость источника - получим другую эпиполярную линию - $e1$. Две эти линии получены от пересечения двух проективных плоскостей камер и так называемой эпиполярной плоскости $O_R P O_T$.

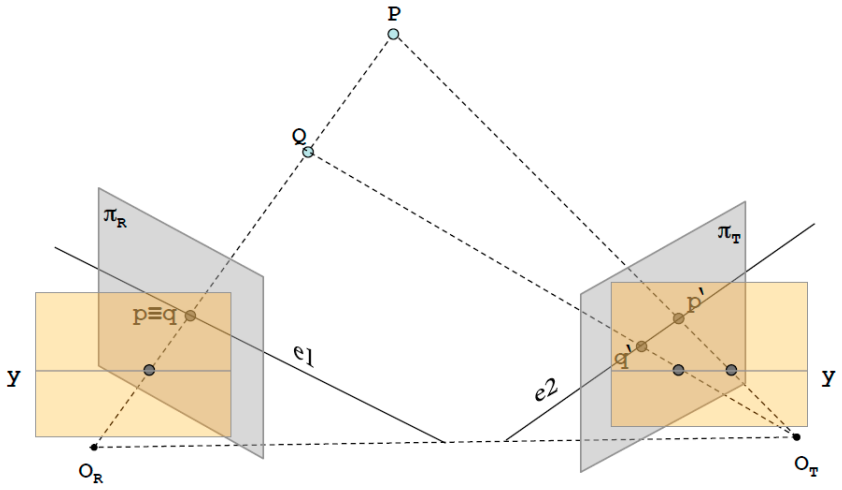

Рис. 2: Эпиполярное ограничение

Более эффективным является решение сначала "выпрямить" системы координат, так, чтобы эпиполярные линии были горизонтальны и соответствующие линии находились на одинаковой высоте (рис. 2). Для этого сначала необходимо "виртуально" повернуть камеры, чтобы они смотрели перпендикуляро линии, соединяющей их оптические центры - $O_R O_T$. Далее регулируется кручение вокруг оптических осей камер, чтобы соответствующие эпиполярные линии были горизонтальны и смещение для точек в бесконечности было нулевым. Последнее - перемасштабировать изображения, чтобы учесть, возможно, разные фокусы камер.

Смещение между расположениями двух соответственных точек на "выпрямленной" (также эту форму называют стандартной) паре изображений называют смещением ($d = X_R - X_T$ на рис. 3). Если рассмотреть подобные треугольники $PO_R O_T$ и Ppp', то окажется, что смещение

непосредственно связано с глубиной следующим образом:

$$d = \frac{B * f}{Z} \tag{1}$$

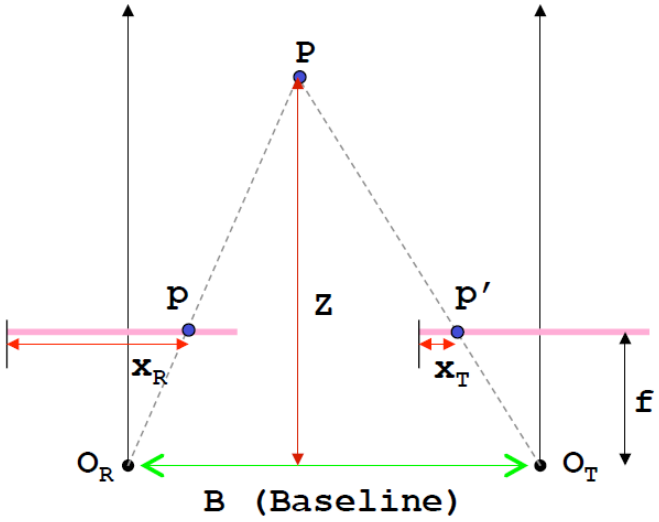

Рис. 3: Связь смещения и глубины

И далее, имея смещение, можно получить исходные в 3D координаты (опять же, имея внутренние и внешние параметры камер), таким образом произведя второй этап - реконструкцию (на самом деле, в общем случае формулы более сложны и принимают матричную форму):

$$Z = \frac{B * f}{d} \qquad X = \frac{Z * x_R}{f} \qquad Y = \frac{Z * y_R}{f} \tag{2}$$

Главной частью стереоалгоритма является поиск соответствий. Этот процесс можно разделить на следующие этапы, согласно [16]: вычисление стоимостей для каждого пикселя, вычисление их взвешенной суммы, минимизация общей стоимости для всего изображения, уточнение полученного решения (схема приведена на рис. 4). Эффективность каждой из предложенных частей непосредственно влияет на все последующие. В последнее время популярность получили так называемые глобальные алгоритмы, которые ищут гладкое решение (в том смысле, что "штрафуется" наличие "скачков" в глубинах). Однако, глобальные алгоритмы довольно медленны, поэтому ищутся всевозможные приближения к глобальному решению - это так называемые полуглобальные

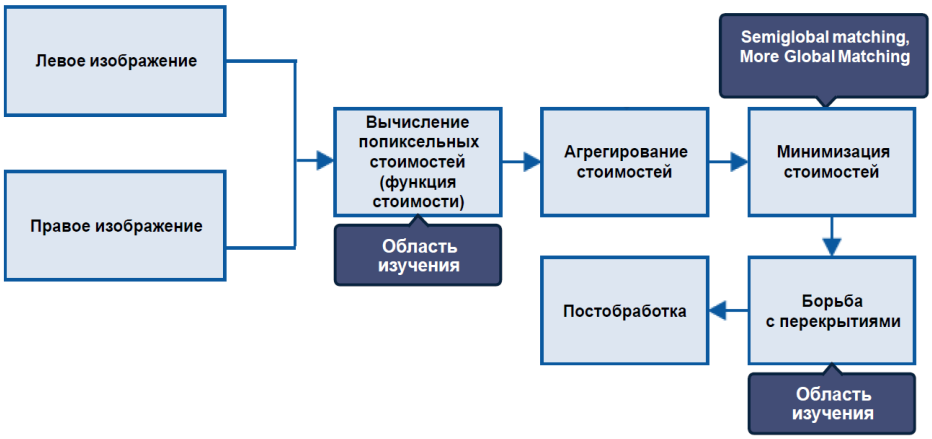

Рис. 4: Схема, демонстрирующая область изучения в данной работе, и место алгоритма More global Matching (MGM)

алгоритмы. Учитывая, что для многих аглоритмов соотношение скорости/точности для различных датасетов очень сильно варьируется, проблема поиска наилучшего алгоритма для данного набора изображений все еще пользуется интересом.

Алгоритм Semi-global matching (SGM, алгоритм полуглобального сопоставления) - один из ведущих стереоалгоритмов, реализующий этап минимизации. Этот алгоритм использует эффективную стратегию для приблизительной минимизации энергии, которая состоит из попиксельной стоимости и попарной (между соседними пикселями) гладкости. На основе предложенной интерпретации SGM как алгоритма распространения доверия, совсем недавно был преложен новый алгоритм - Significantly More Global Matching (MGM, значительно более глобальный поиск соответствий [4]) - который позволяет в пять раз минимизировать зазор энергии невязки решения по сравнению с SGM, при этом почти не имеет накладных расходов. Однако сам алгоритм реализует этап минимизации, и авторы алгоритма оставили за пределами своего исследования использование вместе с их алгоритмом многих других популярных решений, лежащих на других этапах, например, использование более точной функции стоимости или борьбу с перекрытиями.

Эта работа посвящена дальнейшему развитию алгоритма MGM - применение вместе с ним алгоритмов, часть из которых модифицирована нами, но ранее применялась с SGM, а часть - впервые предложена

нами, с целью выяснения, какие из них в паре с минимизационным алгоритмом MGM демонстрируют наилучший результат. В связи с тем, что MGM был разработан совсем недавно, дальнейшего развития на момент написания работы не получил.

1. Постановка задачи

Целью работы является развитие алгоритма минимизации более глобального поиска соответствий (MGM [4]) - поиск и реализация более точных алгоритмов (нежели используемые сейчас), которые предшествуют/следуют за этапом минимизации, чтобы добиться максимальных результатов от названного стереоалгоритма на одном из популярных наборов изображений. Также ставится цель исследовать применимость алгоритма MGM на наборах изображений с лицами людей, чтобы понять, насколько успешно он может использоваться в реконструкции лиц и определить с какими параметрами и предложенными алгоритмами достигается наилучший результат.

В ходе работы были поставлены следующие подзадачи:

1. Адаптация MGM [4] под библиотеку OpenCV
2. Разработка и реализация алгоритма оценки стоимостей на основе синтеза алгоритмов Census и Mutual Information [6]
3. Разработка и реализация функции стоимостей на основе сиамских сверточных нейронных сетей
4. Разработка и реализация алгоритма поиска перекрытий и устранение их проявлений (а вместе с тем и повышения точности самого алгоритма)
5. Использование адаптивных весов - одних из стандартных параметров MGM - на основе градиента изображения
6. Проверка полученного алгоритма на наборах изображений Middlebury [13], а также на наборах изображений лиц людей и анализ полученных результатов

2. Известные результаты

Методы поиска стересоответствий обычно подразделяют на локальные и глобальные методы. Локальные методы оценивают смещение независимо для каждого пикселя, сравнивая некоторое свойство (обычно в пределах какого-то окна вокруг пикселя) на левом и правом изображениях. Локальные методы вычислительно дешевы, однако им присущи проблемы в работе с нетекстурированными областями, повторяющимися шаблонами и резкими скачками в глубинах. Глобальные методы борются с этими недостатками путем введения специального члена V, отвечающего за гладкость решения:

$$E(D) = \sum_{p \in I} C_p(D_p) + \sum_{p,q \in \epsilon} V(D_p, D_q) \tag{3}$$

Здесь D_p - матрица смещений для пикселя p. Сумма берется по всем пикселям p, принадлежащим изображению. V - функция, штрафующая за разрыв в смещениях для соседних пикселях p и q. C (далее также будет записываться в форме $C(p, q)$) - функция, которая оценивает "штраф" (невязку) за назначение пикселю p смещения D_p (в другой форме записи - $q - p$). Для соответствующих пикселей на изображениях она должна быть как можно меньше, для различных - как можно больше.

Подобная задача поиска соответствий также часто формулируется в терминах случайных полей Маркова (MRF). MRF определяется на ненаправленном графе $G(V, \epsilon)$, где ϵ - возможные соединения с соседними вершинами. Каждая вершина $p \in V$ - случайная переменная, принимающая конечный набор возможных смещений. Лучший такой набор смещений D^*, минимизирующий 3 будет решением. Для задачи минимизации на MRF, имеющим представление в виде дерева, был предложен алгоритм распространения доверия [15], минимизирующего сумму 3. Он вычисляет для каждого узла доверие $B_p(d_p)$, отправляя сообщения вдоль связей между узлами. Как только узел q принял сооб-

щения от всех своих соседей, кроме p, он может отправить p сообщение:

$$m_{q\to p}(d) = min_{d'\in D}(C_q(d') + \sum_{(q,k)\in\epsilon, k\neq p} m_{k\to q}(d') + V(d,d')) \tag{4}$$

И доверие узла p к значению d вычисляется следующим образом:

$$B(p,d) = C_p(d) + \sum_{(q,p)\in\epsilon} m_{q\to p}(d) \tag{5}$$

Далее для данного узла p берется $d \in D$, такое, что для него доверие $B(p,d)$ минимально по всем d.

Также существует ряд подходов, которые используют ground truth данные, чтобы оценить параметры модели MRF. Zhang и Seitz в [20] оценили таким образом оптимальные параметры для случайных полей Маркова.

Hirschmuller предложил еще один алгоритм минимизации энергии на сетках - SGM (алгоритм полуглобального сопоставления) [6]. Ему удалось свести эту задачу к минимизации на однонаправленных линиях, взятых в разных направлениях, рис. 5. Задача решается на каждой линии (r - направление) отдельно, а затем все полученные решения суммируются:

$$L_r(p,d) = C_p(d) + min_{d'\in D}(L_r(p-r,d') + V(d,d')) \tag{6}$$

$$S_p(d)(p,d) = \sum_r L_r(p,d) \tag{7}$$

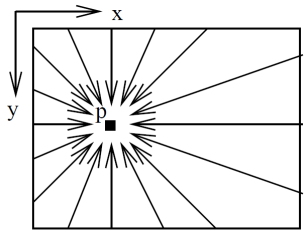

Рис. 5: Пример 16 линий-направлений, на которых ищется оптимальное решение в SGM

Чуть позже [14] была продемонстрирована эквивалентность этих двух подходов для 8 направлений, если использовать следующую формулу (рис. 6):

$$S_{oc}(p,d) = \sum_{r} L_r(p,d) - (N_{dir} - 1) * C_p(d) \tag{8}$$

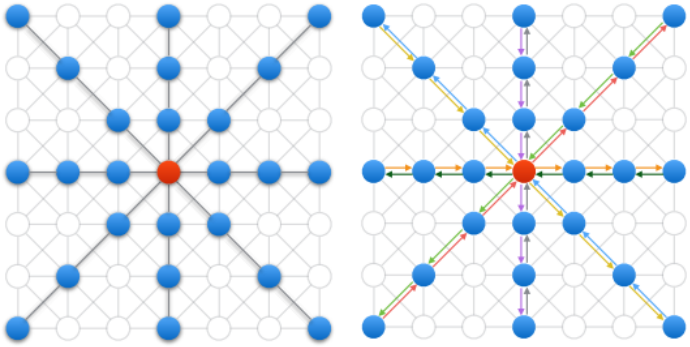

Рис. 6: Пример эквивалентных схем для BP(слева) и SGM (справа) для 8 направлений

В 2015 году группой французских ученых был предложен алгоритм - MGM [4] A Significantly More Global Matching - главным преимуществом которого являлось большая глобальность решения относительно SGM. Они видоизменили формулу 8 в духе BP, предложив использовать информацию более чем в одном направлении. В MGM сообщения также передаются с перпендикулярных(r') линий обхода:

$$L_r(p,d) = C_p(d) + \sum_{x \in r, r'} \frac{1}{2} min_{d' \in D}(L_r(p-x,d') + V(d,d')) \tag{9}$$

MGM на 20% медленее SGM, однако на 40% лучше минимизирует энергию функционала. Также MGM очень легко распараллеливается, достаточно быстр, а также обладает высокой точностью даже используя простые функции стоимостей - именно поэтому этот минимизационный алгоритм был выбран для дальнейшей работы.

Помимо выбора минимизационного алгоритма, большое влияние на точность работы всего алгоритма оказывает выбор функции стоимости. Census - непараметрическая функция, используемая в MGM по умолча-

нию. Она устойчива к локальным и некоторым глобальным радиометрическим изменениям (радиометрические изменения - различия в цветовых характеристиках на двух снятых изображениях (с двух разных камер) одних и тех же точек объекта ввиду различных характеристик съемочных приборов (например, чувствительность сенсоров) и различных ракурсов съемки) и отображает структуру изображения в пределах заданного окна. Каждому пикселю в пределах окна сопоставляется единица, если его значение меньше или равно значению центрального пикселя окна. Таким образом получается битовая строка, которая сравнивается с аналогично полученной строкой на участке целевого изображения путем вычисления расстояния Хэмминга ($C_{index}(p, q)$ - полученная функция стоимости с индексом $index$):

$$C_{census}(p, q) = \bigotimes(T_{census}(p), T_{census}(q)), \tag{10}$$

где

$$T_{census}(P) = \bigoplus_{[i,j]\in D} \xi(P, P + [i, j]), \tag{11}$$

где $\oplus$ обозначает конкатенацию, D - непараметрическое окно вокруг P, ξ - следующий индикатор:

$$\xi(P, P + [i, j]) = \begin{cases} 1 & \text{если P} > \text{P+[i, j]} \\ 0 & \text{иначе} \end{cases} \tag{12}$$

Для дальнейших сравнений был выбран размер окна D равный 5 $\times$ 5. Однако, как описано в [21], census порождает неточности на краях объектов, что отрицательно сказывается на общей точности.

Также существует ряд подходов, которые используют ground truth данные, чтобы оценить параметры модели функции стоимости. Например, Yunpeng Li и Daniel P. Huttenlocher в [10] предложили непараметрическую функцию стоимости, которая может автоматически обучаться с помощью метода опорных векторов. Jure Zbontar [19] первым применил нейронные сети для обучения функции стоимости, используя ее вместе с алгоритмом SGM (обучение происходило на базе библиотеки

нейронных сетей Torch).

Еще одной проблемой в стереосопоставлении является борьба с перекрытиями. Перекрытые пиксели обычно видны лишь на одном изображении, так что невозможно оценить информацию о смещении на основе перекрытой пары пикселей.

Heiko Hirschmuller в [6] предложил использовать экстраполяцию верно назначенных значений смещений вдоль восьми направлений, однако на многих датасетах такое большое количество источников экстраполяции порождает артефакты ввиде целых линий, вдоль которых значение было экстраполировано неверно. Bleyer в [3] предложил идею, согласно которой перекрытым пикселям не разрешается распространять на этапе минимизации функционала свое неверное значение.

Dongbo Min в [11] попробовал внедрить схему по борьбе с перекрытиями прямо на этап минимизации стоимостей, используя итеративный процесс. Vladimir Kolmogorov в [9] предложил использовать специальный член в формуле 3, который будет штрафовать ситуацию, при которой для пикселя нарушается ограничение уникальности (неоднозначное соответствие пикселей источника и цели).

3. Предлагаемые алгоритмы

3.1. Функции стоимости

На изображениях, снятых радиометрически откалиброванными установками, соответствующие пиксели имеют схожие значения интенсивности. Для таких изображений простая функция стоимости (например, абсолютная разность интенсивностей) не уменьшает точности стереоалгоритма. В реальных ситуациях, однако, цветовые значения могут быть подвергнуты радиометрическим вариациям, включая глобальные изменения интенсивности (например, различная гамма-коррекция, значение чувствительности (ISO) камер), а также локальные радиометрические изменения (различное освещение, шум, а также при наличии поверхностей, для которых не выполняется закон Ламберта). Эти вариации часто встречаются при съемке на обычные камеры, поэтому использование устойчивых функций стоимости является необходимым.

Функция census, используемая в MGM, устойчива к локальным радиометрическим изменениям, однако неустойчива к глобальным. Поэтому мы предложили использовать новую функцию стоимости MIC, а также (в случае наличия тренировочного датасета) функцию стоимости CNN.

3.1.1. Функция стоимости Mutual Information and Census (MIC)

Чтобы повысить точность работы алгоритма на краях объектов, а также повысить устойчивость к глобальным радиометрическим изменения, мы прибегли к использованию функции стоимости Mutual Information, которую впервые в стереосопоставлении применил Heiko Hirschmuller [6]. Она основывается на энтропии как каждого из двух отдельных изображений, так и на энтропии их совместного распределения, которое получается на основе априорного знания о соответствии. Функция стоимости в данном случае задается следующим образом (здесь и далее I_r - интенсивность изображения-источника, I_t -

целевого изображения):

$$C_{MI}(p,q) = -mi_{I_r,f_D(I_r)}(I_r(p), I_t(q)) \tag{13}$$

$$mi_{I_r,I_t}(i,k) = h_{I_r}(i) + h_{I_t}(k) - h_{I_r,I_t}(i,k) \tag{14}$$

Совместная энтропия $h_{I_r,I_t}(i,k)$ оценивается следующим образом:

$$h_{I_r,I_t}(i,k) = -\frac{1}{n} log(P_{I_r,I_t}(i,k) * g(i,k)) * g(i,k), \tag{15}$$

где $g(i,k)$ - ядро Гаусса (было взято 7×7), $P_{I_r,I_t}(i,k)$ - распределения вероятностей:

$$P_{I_r,I_t}(i,k) = \frac{1}{n} \sum_p T[(i,k) = (I_{r_p}, I_{t_{f(p)}})] \tag{16}$$

Аналогично, для индивидуальных энтропий, только лишь индивидуальное распределения вероятностей рассматривается как сумма колонок/рядов совместного распределения вероятностей:
$P_{I_r}(i) = \sum_k P_{I_r,I_t}(i,k)$. Эта поправка учитывает, что некоторые пиксели могут являться перекрытыми на одном из изображений.

Также стоит отметить важность проведения операции winsorising перед нормализацией стоимостей: был выбран порог отбрасывания 0.05 (значения меньше 0.05-квантиля и больше 0.95-квантиля заменяются 0.05-квантилем и 0.95-квантилем соответственно). Это заметно способствует "правильной" нормализации данных, когда шум на краях выборки занимает бóльший диапазон всех возможных значений.

В качестве интенсивности для распределения вероятностей использовалось значение интенсивности в оттенках серого (в диапазоне $[0, 255]$). Нормализация значений mi_{I_r,I_t} проходила приведением по параметрам

$$\min, \max_{(i,k)\in[0,255]\times[0,255]} mi_{I_r,I_t}(i,k)$$

к диапазону $[0., 255.]$ после операции winsorising.

В отличие от метода, предложенного Hirschmuller (т.н. hierarchical

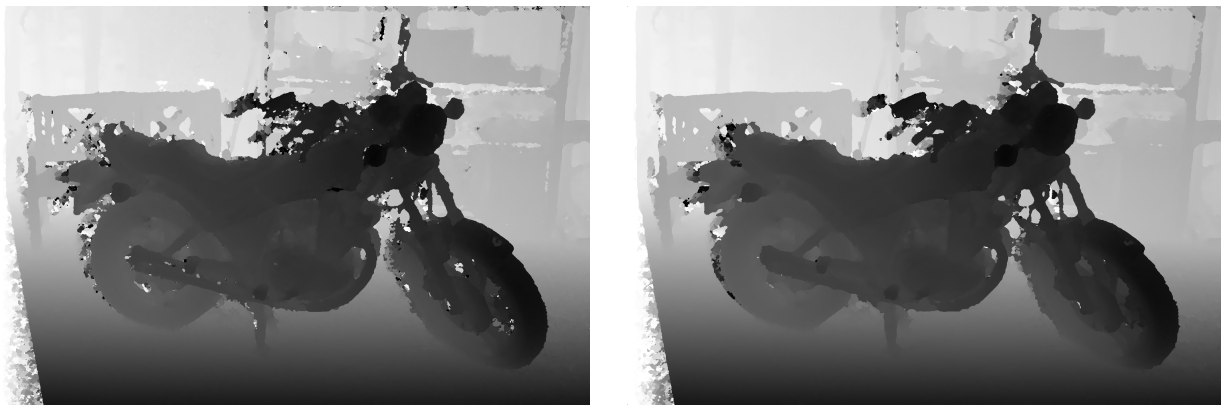

Рис. 7: Карта смещений, полученная с помощью функции стоимости census (слева) и MIC $w_{MI} = 0.4$ (справа)

MI), для первого априорного приближения распределения вероятностей на первой итерации была использована функция стоимости census. Затем, зная априорное распределение, вычислялось значение mi. Такой подход медленнее (требуется две итерации на полноразмерном изображении), но дает точное решение за меньшее число всех итераций и не требует постоянного перемасштабирования изображений.

Сам по себе Mutual Information на большинстве датасетов демонстрирует худший результат, нежели census (как видно из полученных нами результатов). Поэтому была введена новая функция Mutual Information and Census, которая является их линейной комбинацией с некоторыми весами:

$$C_{MIC}(p,q) = w_{MI} * C_{MI}(p,q) + (1 - w_{MI})C_{census}(p,q)$$

Таким образом, данная функция сочетает в себе преимущества обеих функций стоимости: устойчивость к локальным радиометрическим изменениям (как в census) и бóльшая точность на краях объектов (MI). Сравнение полученных карт смещений приведено на рисунке 7.

3.1.2. Функция стоимости на основе сиамской сверточной нейронной сети (CNN)

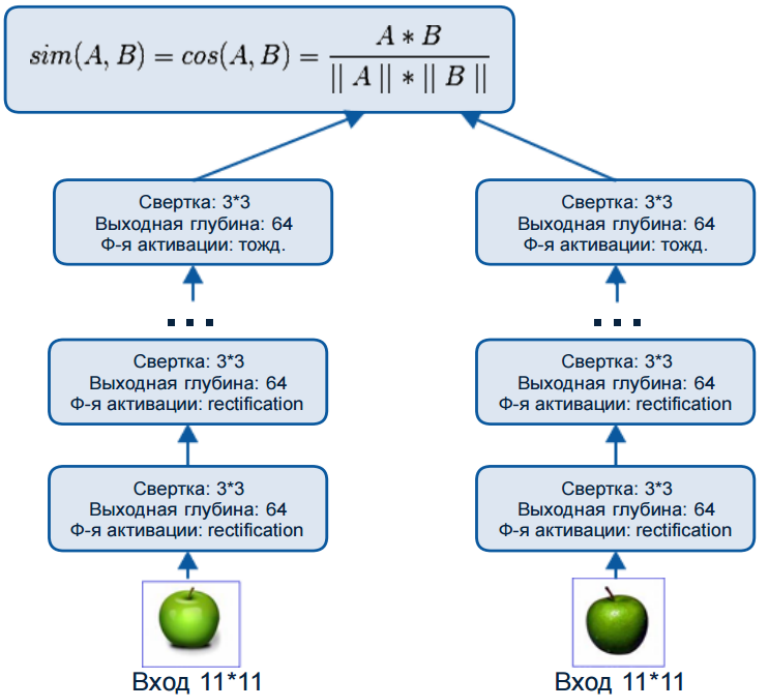

Рис. 8: Используемая архитектура сверточной нейронной сети

Архитектура используемой сиамской сверточной нейронной сети (ССНС) приведена на рисунке 8. На вход двум равнозначным веткам сверточной нейронной сети подаются участки изображений (все их возможные сопоставления) соответственно с источника и цели. На выходе нейронная сеть определяет степень похожести этих участков, то есть с какой вероятностью это сопоставление является истинным. Для обучения сети нужно было использовать уже три участка: один участок с источника и два участка с цели - истинное сопоставление участку с источника и ложное. Обучение состоит в том, чтобы значение выхода сети для пары положительного участка цели и участка с источника было больше значений выхода для всех пар ложных участков цели и участка с источника.

Все сверточные слои имеют выходную глубину 64 (64 карты признаков) и связаны со всеми картами признаков на предыдущем слое (таблица соединений содержит все единицы), а размеры всех сверточных матриц - 3×3. Использовались параметры-размеры сверточных

матриц, рекомендуемые в работе [19] для данного разрешения изображений (порядка 1800×1300 пикселей).

Нами была использована иная функция схожести двух выходов сети на участках изображения A и B:

$$sim(A, B) = cos(A, B) = \frac{A * B}{|| A || * || B ||}$$

Так как оригинальный MGM написан на C++, использовалась легковесная библиотека tiny-cnn в качестве фреймворка сверточных нейронных сетей, в первую очередь, в силу того, что она самодостаточна и не зависит от других библиотек. Однако так как данный фреймворк не предполагал использование именно сиамских НС, а также использование выбранной нами функции потерь, то эта часть была реализована вручную.

Первая из описанных проблем была решена созданием двух аналогичных НС, и усреднением дельт весов и смещений левой и правой части НС перед шагом обновления весов нейронной сети (чтобы параметры НС всегда были идентичны). Вторая проблема кроется в том, что выбранная функция потерь $L(\theta) = \sum_{(x_l, x_p, x_n)} max(0, m + sim(x_l, x_n) - sim(x_l, x_p))$, где x_l есть фрагмент с источника, x_p - соответствующий ему позитивный участок на целевом изображении, x_n - соответствующий ему негативный участок на целевом изображении, зависит не только от двух входных пар, но и от третьей пары. Выходом стало включать в каждый mini-batch тройку дважды: каждый раз мы рассматриваем нашу НС как функцию лишь от двух входных участков, второй их которых в одном случае - x_p, в другом - x_n.

Градиент функции потерь по выходному вектору НС, вычисленный нами (был выбран $m = 0.3$):

$$\frac{\partial E}{\partial y_i} = \begin{cases} 0 & \text{если } m + sim(x_l, x_n) - sim(x_l, x_p) < 0 \\ \frac{x_{n_i} * ||x_l||^2 - (x_l, x_n) * x_{l_i}}{||x_l||^3 * ||x_n||} & \text{иначе и } y = x_n \\ -\frac{x_{n_i} * ||x_l||^2 - (x_l, x_p) * x_{l_i}}{||x_l||^3 * ||x_p||} & \text{иначе} \end{cases} \quad (17)$$

В случае, если значение на паре с негативным участком довольно близ-

ко к значению с позитивным участком, должен быть наложен "штраф" - градиент функции отличен от нуля. В ином же случае, если разность между этими значениями больше m, "штрафа" накладываться не должно.

Обучение происходило на расширенной части тренировочного датасета Middlebury. Все изображения были сначала переведены в оттенки серого, а затем стандартизированы путем вычета среднего и делением на дисперсию. Также тренировочный датасет был расширен путем отражения пар изображений по горизонтали и вертикали, а также использования того факта, что многие датасеты Middlebury были сняты при различных услових освещения. В качестве оптимизатора для градиентного спуска использовался RMSProp [18] с параметрами $alpha = 0.0001, mu = 0.99, eps = 1e-8$:

$$w(t) = w(t-1) - \alpha * \triangle w(t)$$
$$\triangle w(t) = \frac{\partial E}{\partial w}(t) / \sqrt{MeanSquare(w,t) + \epsilon}$$
$$MeanSquare(w,t) = mu * MeanSquare(w, t-1) + (1 - mu)\frac{\partial E}{\partial w}(t)^2$$

В тренировочный набор включались только неперекрытые участки изображений. Задание x_p происходило однозначно, тогда как x_n заданы путем небольшого смещения влево или вправо относительно участка x_p:

$$q = (x - d \pm rand[2,6], y)$$

Всего обучение происходило на 10 млн. троек участков, на протяжении 10 эпох с размером mini-batch = 128 (четность особо важна ввиду выбранной реализации). Так как выбранный фреймворк пока что не реализован на GPU, обучение происходило в 8 потоков и заняло около 50 часов.

Финальная функция стоимости, используемая нами на основе CCHC:

$$C_{CNN}(p,q) = \frac{1}{2}(-sim(Win_{I_r}(p), Win_{I_t}(q)) + 1)$$

Функция стоимости (принимающая значение из [0,1]) также, как и все

остальные, была нормализована к диапазону $[0., 255.]$. Как изображение-источник, так и изображение-цель были расширены отступами в $WINSIZE/2$ пикселей с интенсивностью 0 (zero-padding) в каждом и 4 основных направлений, чтобы правильно применять свертку по краям изображений.

Стоит отметить, что дальнейшим шагом в оптимизации реализации этого метода может послужить оптимизация многократной свертки перекрывающихся участков изображения, что в текущей реализации не предусмотрено в виду того, что на вход НС подается всегда лишь участок изображения, а не все изображение целиком.

3.2. Экстраполяция перекрытых регионов

Еще одной проблемой в стереосопоставлении является борьба с перекрытиями. Перекрытые пиксели обычно видны лишь на одном изображении, так что невозможно оценить информацию о смещении на основе перекрытой пары пикселей. Однако, во многих областях, например, 3D-моделировании (что было сделано нами при построении 3D-моделей лиц людей), необходимо назначить обоснованные значения смещений перекрытым пикселям. Поэтому ввиду этой необходимости нами были использованы техники по борьбе с перекрытиями, одна из которых является скрещиванием методов [3] и [6], а вторая - нашим развитием идеи первой.

3.2.1. Линейная экстраполяции перекрытых регионов

Неверно назначенные смещения можно поделить на две категории: перекрытые пиксели и "несоответствия" (неперекрытые неверно назначенные пиксели). Найти множество всех неверно назначенных пикселей можно с помощью left-right проверки. Она состоит в проверке согласованности (рассматривается ошибка перепроектирования: сначала происходит проекция с источника на цель, затем - обратно) назначенных значений смещений на источнике и цели с определенной точностью ϵ. Был взят $\epsilon = 1px$. Помимо пикселей, не прошедших left-right проверку

Рис. 9: Маски Invalidated-регионов без описанного преобразования по борьбе с шумом (слева) и с ним (справа)

(LR), в это множество приписывались пиксели, не прошедшие blayer test [3], но после нескольких простых морфологических трансформаций с целью избавления от лишних шумов. Таким образом, множество неверно назначенных смещений $Invalidated$ строится следующим образом:

$$Invalidated = LR \cup 4connected(morph(Blayer))$$

Однако сгенерированная таким образом маска допускала много ошибочно не отброшенных (в виду ошибки "несоответствия") пикселей. Для борьбы с этой проблемой нужно было устранить мелкие шумы в маске (рис. 9), внутри Invalidated-регионов. Для поиска контуров границ выбран алгоритм, предложенный Satoshi Suzuki [17]. Он позволяет находить не только сами контуры, но и их иерархию. Поиск происходил по самым внешним контурам, далее расматривались их потомки. Если площадь внешнего контура во много раз (мы выбрали $\alpha = 140$) превосходит площадь внутреннего потомка ($S_{outer}/S_{inner} > \alpha$), то внутренность внутреннего контура рассматривается как шум (его индекс помещается в $invalidatedContours$):

$$Invalidated' = Invalidated \bigcup_{i \in invalidatedContours} S_i$$

Также всем пикселям, не прошедшим blayer test, запрещалось распространять на этапе минимизации функционала свое неверное значение смещения. Сделано это путем приравнивания члена $V(D_p, D_q)$ в фор-

муле 3 нулю для пикселей, являющимися соседями пикселей, не прошедших blayer test. Эти пиксели, скорее всего, на данном этапе получат неверное значение смещения, однако затем мы все равно переопределим его, включив их в множество $Invalidated$.

Далее нужно отличать перекрытые пиксели и "несоответствия", вызванные ошибкой в назначении смещений. Hirschmuller [6] нашел оценочный критерий для этого (см. рис. 10). Пусть точка $p1$ источника проектируется в точку $q1$ цели. Точку $p1$, для которой функция со значениями смещений, определенная на соответствующей эпиполярной линии цели $e_{bm}(q1, d)$ и всегда проектирующая в исходную точку источника $p1$ (на рисунке $e_{bm}(p1, d)$), не пересекает функцию смещений D_m на изображении-цели, будем считать перекрытой, иначе - "несоответствием". Можно сказать, что эквивалентно, что в перекрытую точку не проектируются другие точки цели. Пиксели, являющиеся прямыми соседями перекрытых, также считаем перекрытыми. Далее необходимо экстраполировать значения для перекрытых пикселей исходя из информации, которой обладают его соседи из "заднего фона", для которых значения назначены корректно. Для пикселей, для которых были допущены "несоответствия", необходимо уже интерполировать их значения значениями с переднего плана.

Для этого "распространяем" правильно назначенные значения от соседей вдоль основных восьми направлений. Стоит отметить, что априорно задается основное направление. Оно используется для того, чтобы значения, распространяемые не из основных направлений, не учитывались, если расстояние до источников распротранения слишком велико. Для всех основных датасетов мы использовали направление left-to-right (им соответствуют значения u_{p1} и u_{p2} из всех восьми значений):

$$D'_p = \begin{cases} D_p & \text{если p не в Invalidated множестве} \\ median_i(u_{pi}) & \text{если p - "несоответствие"} \\ min_i(u_{p1}, u_{p2}, & \\ secmin_j(u_{pj} \setminus (u_{p1} \cup u_{p2}))) & \text{если p перекрыт} \end{cases}$$

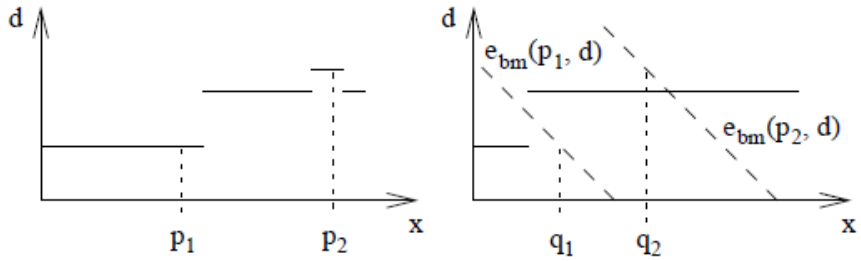

Рис. 10: Отличие перекрытия и "несоответствия"

3.2.2. Экстраполяции перекрытых регионов путем голосования

Логичным продолжением описанной идеи является разработка алгоритма по борьбе с перекрытиями не только на основе геометрических соображений, но и на основе цветовой составляющей. Еще одной идеей является распространять правильно назаначенные значения не только от одного пикселя, но и от его соседей. Все эти эвристики получили дальнейшее обобщение.

Далее используется та же маска неверно определенных смещений $Invalidated'$, что была получена в предыдущем пункте.

Введем функцию вероятности схожести (на самом деле, это скорее весовая функция) значений смещений двух пикселей l и k как

$$w(l,k) = e^{-\frac{EuclDist^2(l,k)}{\alpha} - \frac{EuclDist^2(I(l),I(k))}{\gamma}}$$

Значения $I(l), I(k)$ представляют собой векторы из 3 компонент в формате RGB, EuclDist - функция расстояния в евклидовом пространстве. Процесс распространения верных смещений разбивается на два этапа: начальный этап распространения смещений от пикселей, не принадлежащих набору $Invalidated'$, и распространения смещений между элементами $Invalidated'$. Для начала положим $d^0(k) = D_k$, т.е. проинициализируем теми же значениями, что и карта смещений, полученная после этапа минимизации. Вычислим значения функции-носителя на

начальном этапе (на основе значений его соседей $N(l)$):

$$Supp^0(l,d) = \sum_{k \notin Invalidated, k \in N(l)} w(l,k) * \mathbb{1}_{d^0(k)=d} \tag{18}$$

Теперь определим основные приближения после начального этапа (смещения и соответствующие им значения вероятностей):

$$d^0(l) = max_d Supp^0(l,d)$$

$$Supp^0(l) = Supp^0(l, d^0(l))$$

Значения $Supp^0(l)$ характеризуют "надежность" вычисленных значений $d^0(l)$. Чем больше $Supp^0(l)$, тем больше вероятность, что пикселю l назначено верное смещение $d^0(l)$. После этого этапа следует итеративный этап распространения смещений уже между самими перекрытыми пикселями:

$$Supp^i(l,d) = \sum_{k \in Invalidated, k \in N(l)} S^{i-1}(k) * w(l,k) * \mathbb{1}_{d^{i-1}(k)=d} \tag{19}$$

$$Norm^i(l,d) = \sum_{k \in Invalidated, k \in N(l)} w(l,k) * \mathbb{1}_{d^{i-1}(k)=d}$$

Вторая функцию нужна для нормализации значений функции-носителя. Начальная сумма значений функции-носителя должна сохраняться на протяжении всего итеративного процесса. Почти аналогично начальному вычисляются смещения и соответствующие им значения вероятностей после данного этапа:

$$d^i(l) = max_d Supp^i(l,d) \tag{20}$$

$$Supp^i(l) = \frac{Supp^i(l, d^i(l))}{Norm^i(l, d^i(l))}$$

Суммирование значений весов для значения $d^0(k) = d$ в формуле 19 реализовано с помощью unordered map. Это удобно с той точки зрения, что получение наибольшего значения в формуле 20 может быть реализовано за один проход по всем окрестным пикселям.

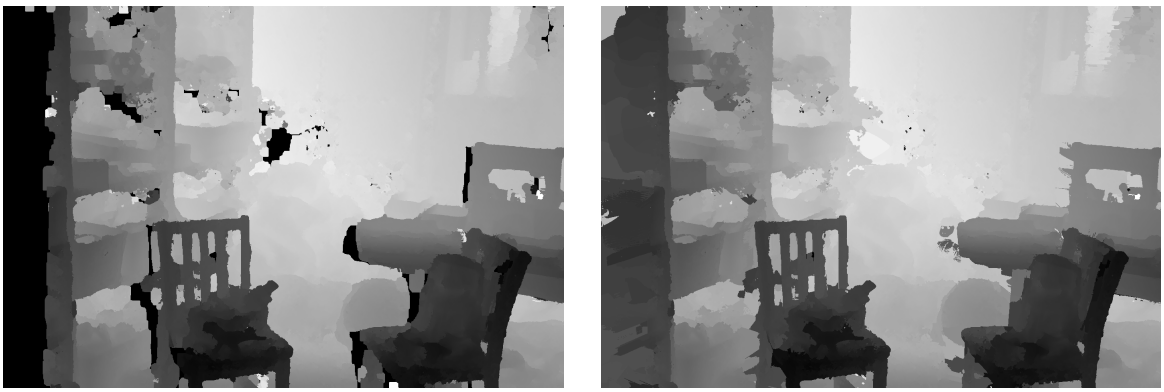

Рис. 11: Карта смещений после первой итерации и в конце метода

Был выбран размер окна на первой начальной итерации 20×20 и 12×18 на последующих. Стоит отметить, что размер окна непосредственно влияет на число итераций, поскольку зачастую распространение смещений должно происходить на довольно большие расстояния. Было выбрано число итераций $IterNumber = 13$, $\alpha = 140$, $\gamma = 50$. Сложность процесса $O(W * H * N)$, где N - размер окна. Примечательно, что сложность не зависит от размера вектора возможных смещений. Однако на датасетах с большим разрешением (например, Middlebury 2014) процесс довольно долгий, поскольку окно должно быть выбрано довольно большим, чтобы верные смещения должны быть распространены до всех перекрытых регионов.

3.3. Адаптивные веса на основе градиента изображения

В MGM член $V(d, d')$ в формуле 9 выглядит следующим образом:

$$V(d, d') = \begin{cases} 0 & \text{если } d = d' \\ P1 & \text{если } \mid d - d' \mid = 1 \\ P2 & \text{иначе} \end{cases} \tag{21}$$

Мы попробовали использовать технику адаптивных весов $P1$ и $P2$, в зависимости от модуля градиента относительно точек p и $p - x$ (используя

идею [2]):

$$P_i = \frac{P_{iO}}{1 + |I(p) - I(p - x)|/W_{P_i}}, i = 0..1 \quad (22)$$

$p - x$ - непосредственный сосед p, согласно MGM (9). Параметр W_{P_i} контролирует возможное уменьшение P_i. Данная модификация использует наблюдение, что достаточные изменения градиента являются признаком возможного "скачка" в глубинах, вызванного границей объекта.

Однако, как показали результаты экспериментов (см. раздел "Результаты экспериментов"), данная эвристика плохо сработала с MGM, возможно, потому что MGM уже на своем уровне хорошо "штрафует" данные скачки (большая стоимость назначения такому "соседу" того же смещения) путем рассмотрения перпендикулярных направлений.

4. Результаты экспериментов

Для сравнения результатов, полученных применением той или иной части алгоритма, были стандартизированы применяемые алгоритмом параметры. Так, мы положили $P1 = 80$, $P2 = 320$ в формуле 21 (как это рекомендовано авторами MGM в общем случае), а значения функции стоимости перемасштабировали к $[0; 255]$.

Были использованы две функции оценки ошибок - bad 1.0 (пиксели, для которых отклонение смещения по сравнению с ground truth не более пикселя) и avgerr (среднее отклонение от ground truth в пикселях). Для тестирования был выбран датасет Middlebury 2014 в разрешении 1/2 от максимального, так как это наболее близко отражает наш дальнейший эксперимент с лицами людей (стоит обратить внимание, что сам университет Middlebury использует в таком случае немного другую метрику, сначала перемасштабируя результат к полному разрешению и уже затем сравнивая его с ground truth в полном разрешении). Сравнение результатов производится с результатами алгоритма, полученными с помощью функции стоимости census без использования сторонних алгоритмов уточнения результата (например, медианного фильтра), если явно не указано обратное.

Стоит отметить, что новая функция стоимости MIC превосходит в точности как census, так и MI. Результаты экспериментов представлены в таблицах 1, 2. Наша эвристика о том, что данная функция сочетает в себе преимущества обеих функций стоимости, оказалась верна на всех наборах изображений, что говорит об устойчивости MIC по сравнению с вышеназванными функциями. График, демонстрирующий зависимость точности в терминах bad 1.0 (слева) и avgerr (справа) от постоянной W_{MI} при использовани функции стоимости MIC приведен на рисунке 12.

Функция стоимости CNN оказалась намного точнее census, MI, MIC. Однако в следствие большого объема вычислений и, на данный момент, отсутствия реализации на GPU, подходит лишь для датасетов, требующих высокой точности, но не требующих большой скорости вычисле-

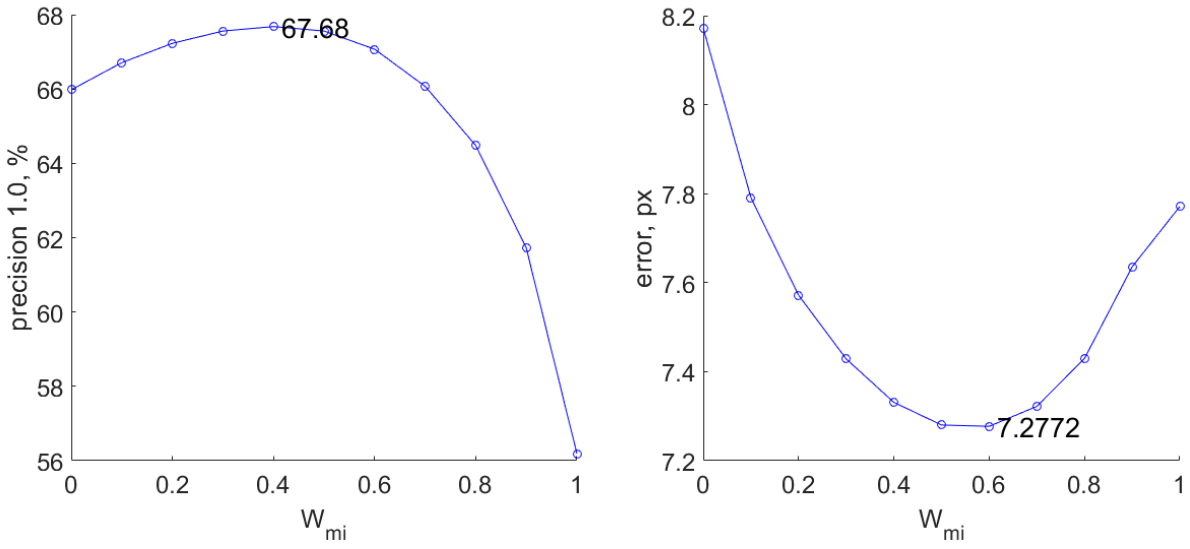

Рис. 12: Графики, демонстрирующие зависимость точности в терминах bad 1.0 (слева) и avgerr (справа) от постоянной W_{MI} при использовани функции стоимости MIC

ний - например, для реконструкции лиц людей. Например, в среднем на одной из стереопар из упомянутого датасета Middlebury 2014 в разрешении 1/2 на процессоре Intel Core i7 3630QM работа нейронной сети занимает 910 секунд. В таблице 3 приведены результаты при использовании CNN и линейной экстраполяции (колонка "Наш метод", это наш метод, показавший наилучшие результаты) в "официальной" метрике Middlebury с перемасштабированием в полное разрешение, которые можно сравнивать с результатами на официальном сайте Middlebury. Также для сравнения приводятся результаты, полученные с помощью лучшего для каждого датасета метода, опубликованного в базе Middlebury, а также использования SGM вместо MGM. В общем рейтинге Middlebury наш метод занимает пятое место при использовании всего изображения как маски и восьмое место при рассмотрении лишь неперекрытых регионов.

Экстраполяция перекрытых регионов, в свою очередь, также продемонстрировала прирост точности на всем изображении, и многократный прирост на перекрытых регионах. Сюрпризом стал тот факт, что линейная экстраполяция оказалась точнее интерполяции на основе голосования. Это может объясняться тем, что в большинстве случае по-

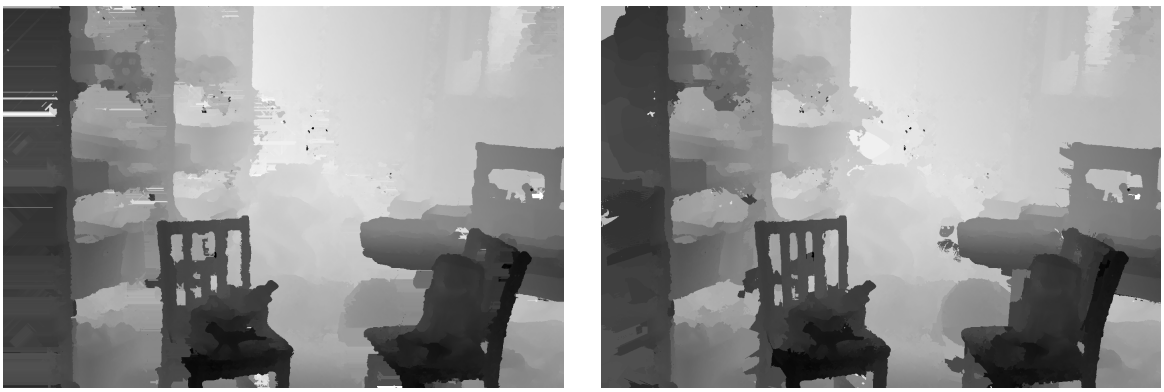

Рис. 13: Карта смещений с использованием линейной (слева) и "голосующей" экстраполяции (справа)

верхность предмета параллельна плоскости камеры, а предметы находятся на значительном расстоянии, что позволяет использовать геометрические соображения, например линейную экстраполяцию. Учитывая, что сложность линейной экстраполяции всего $O(W \times H)$, можно сказать, что ее использование является рекомендуемым. Однако если взглянуть на сравнение карт смещений при использовании предложенных методик (рис. 13, 14), можно заметить, что в случае "голосующего" метода назначенные смещения менее "линейны" и более осмыслены в местах, где плоскость объекта не является параллельной камере. Сравнение результатов приведено в таблицах 4, 5.

В таблицах 6 и 7 приведены обобщенные результаты совместного использования новых функция стоимости и линейной экстраполяции.

Как уже было сказано, использование динамических весов $P1$ и $P2$ вместе с MGM не оправдало себя. В таблицах 8 и 9 приведены результаты с использованием двух значений констант W_{P_i}, дающие наилучшие результаты. На графиках 15 изображена зависимость точности в терминах bad 1.0 (слева) и avgerr (справа) от постоянной W_{P_i} при использовани адаптивных весов. Можно видеть, что при стремлении постоянной к бесконечности, ошибки стабилизируются.

Таблица 1: Точность в терминах ошибки bad 1.0 (отклонение не более пикселя), полученная при использовании функции стоимости Census, Mutual Information and Census (с весами $w_{MI} = 0.4$ и $w_{MI} = 0.5$), Mutual Information, CNN на датасете Middlebury. При измерениях не использовались никакие алгоритмы уточнения результата(например, борьба с перекрытиями)

Датасет	Census, %	MIC 0.4, %	MIC 0.5, %	MI, %	CNN, %
Adirondack	75.26	77.96	77.72	65.17	86.02
Jadeplant	61.99	62.88	62.70	51.61	63.90
Motorcycle	78.75	79.04	78.76	64.68	84.90
Piano	74.92	75.03	74.53	54.23	78.52
Pipes	71.74	72.45	72.28	63.29	78.25
Playroom	62.13	64.84	64.55	51.56	70.08
Playtable	45.01	47.88	48.25	48.08	72.90
Recycle	76.72	79.36	79.29	63.98	82.87
Shelves	43.12	45.62	45.89	39.97	57.18
Взв. среднее	**65.98**	**67.68**	**67.56**	**56.19**	**75.27**

Таблица 2: Ошибка avgerr (среднее отклонение от gt в пикселях), полученная при использовании функции стоимости Census, Mutual Information and Census (с весами $w_{MI} = 0.4$ и $w_{MI} = 0.5$), Mutual Information, CNN на датасете Middlebury. При измерениях не использовались никакие алгоритмы уточнения результата(например, борьба с перекрытиями

Датасет	Census, px	MIC 0.4, px	MIC 0.5, px	MI, px	CNN, px
Adirondack	3.91	3.42	3.40	2.12	3.18
Jadeplant	21.24	18.94	18.79	21.56	20.36
Motorcycle	4.92	4.94	4.96	5.67	4.56
Piano	4.24	3.81	3.82	4.78	3.66
Pipes	9.23	8.97	9.03	10.01	8.11
Playroom	10.55	9.56	9.53	10.20	9.65
Playtable	10.21	9.00	8.82	7.61	5.82
Recycle	3.24	2.94	2.92	3.74	2.76
Shelves	7.25	5.42	5.26	5.32	5.82
Взв. среднее	**8.17**	**7.33**	**7.28**	**7.77**	**5.82**

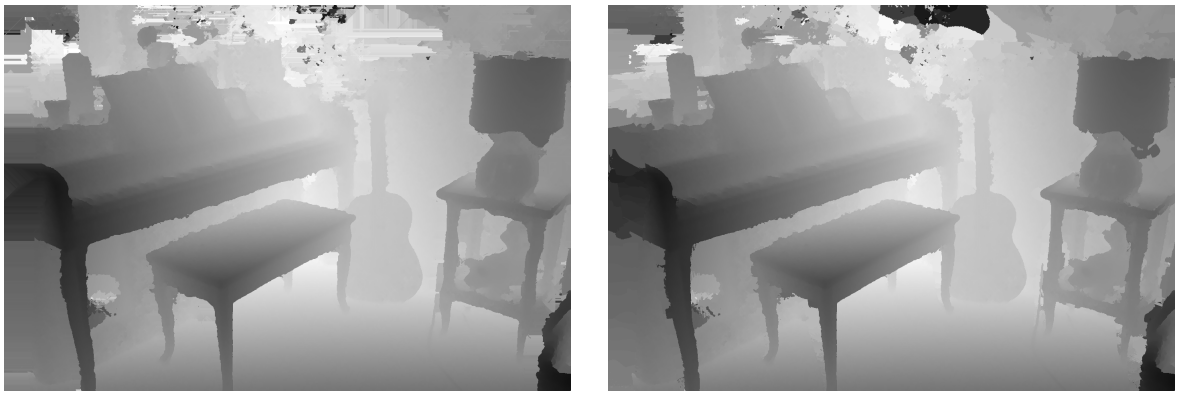

Рис. 14: Карта смещений с использованием линейной (слева) и ”голосующей” экстраполяции (справа)

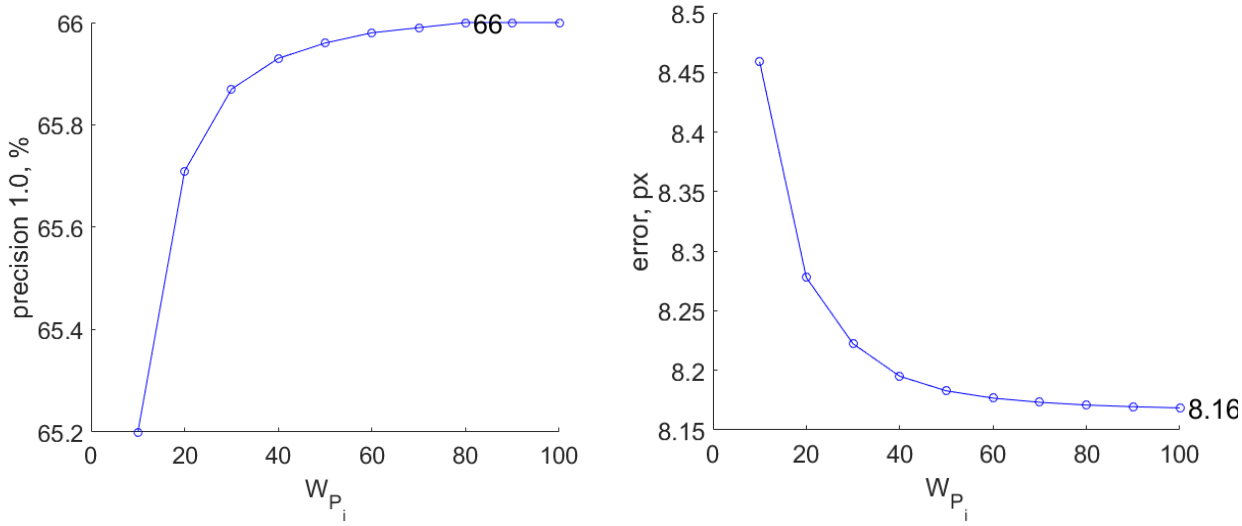

Рис. 15: Графики, демонстрирующие зависимость точности в терминах bad 1.0 (слева) и avgerr (справа) от постоянной W_{P_i} при использовани адаптивных весов

Таблица 3: "Официальная" метрика ошибки Middlebury bad 1.0 (с масштабированием к полному разрешению), полученная с помощью функции стоимости CNN и линейной экстраполяции (колонка "Наш метод"). Также для сравнения приводятся результаты, полученные с помощью лучшего для каждого датасета метода, опубликованного в базе Middlebury, а также применения SGM вместо MGM.

Датасет	Все изображение, %			Не перекрытые регионы, %		
	Наш м-д	Лучший	SGM	Наш м-д	Лучший	SGM
Adirondack	79.18	83.60	78.88	82.65	89.0	82.33
Jadeplant	59.44	60.30	59.28	74.03	76.00	73.66
Motorcycle	82.65	82.80	82.30	88.09	90.34	88.00
Piano	73.20	75.90	72.76	78.20	80.90	77.80
Pipes	75.76	77.70	75.29	88.11	91.16	87.83
Playroom	58.74	66.30	58.37	67.39	74.80	67.06
Playtable	66.05	74.40	65.71	70.68	81.40	70.45
Recycle	75.98	78.20	75.55	79.72	83.90	79.52
Shelves	49.92	57.70	49.62	51.59	61.30	51.18
Взв. среднее	**69.36**	**73.00**	**69.01**	**75.83**	**80.70**	**75.54**

Таблица 4: Точность в терминах ошибки bad 1.0 (отклонение не более пикселя), полученная на датасете Middlebury с помощью линейной экстраполяции и "голосующего" метода экстраполяции

Датасет	Все изобр., %			Перекрытые рег., %		
	Без	Лин.	Голос.	Без	Лин.	Голос.
Adirondack	75.26	79.16	78.17	07.11	56.45	44.75
Jadeplant	61.99	65.44	66.15	05.85	20.75	23.64
Motorcycle	78.75	84.05	82.65	09.17	54.28	36.40
Piano	74.92	77.26	76.60	10.52	32.80	25.92
Pipes	71.74	75.50	75.62	07.07	25.44	25.19
Playroom	62.13	64.13	63.72	06.66	15.92	13.28
Playtable	45.01	46.44	44.51	05.59	17.56	13.77
Recycle	76.72	80.10	79.74	05.21	42.14	31.59
Shelves	43.12	46.90	46.16	07.94	36.45	29.28
Взв. среднее	**65.98**	**69.30**	**68.66**	**07.06**	**30.81**	**25.94**

Таблица 5: Ошибка avgerr (среднее отклонение от gt в пикселях), полученная на датасете Middlebury с помощью линейной экстраполяции и "голосующего" метода экстраполяции

Датасет	Все изобр., px			Перекрытые рег., px		
	Без	Лин.	Голос.	Без	Лин.	Голос.
Adirondack	3.91	1.93	2.16	27.60	6.44	9.85
Jadeplant	21.24	15.25	16.88	63.82	44.96	50.96
Motorcycle	4.92	2.38	2.56	31.60	10.16	12.80
Piano	4.24	2.19	2.59	25.80	6.16	5.91
Pipes	9.23	5.17	6.00	36.89	16.92	19.57
Playroom	10.55	5.09	3.64	54.77	19.12	11.49
Playtable	10.21	7.37	7.80	33.24	11.77	13.40
Recycle	3.24	1.38	1.76	25.60	4.40	9.47
Shelves	7.25	5.67	6.28	16.04	6.29	9.37
Взв. среднее	**8.17**	**5.05**	**5.41**	**39.98**	**18.46**	**20.45**

Таблица 6: Суммарная таблица демонстрирующая точность в терминах ошибки bad 1.0 (отклонение менее 1px) различных комбинаций предложенных методов по отношению к начальному применению лишь функции стоимости census (используемый метод экстраполяции - линейный)

Датасет	Census only, %	MIC 0.4 only, %	CNN only, %	Census+ occl. ref., %	MIC 0.4+ occl. ref., %	CNN+ occl. ref., %
Adirondack	75.26	77.96	86.02	79.16	81.06	90.21
Jadeplant	61.99	62.88	63.90	65.44	65.97	67.30
Motorcycle	78.75	79.04	84.90	84.05	84.10	90.01
Piano	74.92	75.03	78.52	77.26	77.23	80.68
Pipes	71.74	72.45	78.25	75.50	76.24	81.58
Playroom	62.13	64.84	70.08	64.13	66.14	71.57
Playtable	45.01	47.88	72.90	46.44	48.40	75.41
Recycle	76.72	79.36	82.87	80.10	83.04	86.46
Shelves	43.12	45.62	57.18	46.90	47.91	60.37
Взв. среднее	**65.98**	**67.68**	**75.27**	**69.30**	**70.52**	**78.54**

Таблица 7: Суммарная таблица демонстрирующая ошибку avgerr (среднее отклонение от gt в пикселях) различных комбинаций предложенных методов по отношению к начальному применению лишь функции стоимости census (используемый метод экстраполяции(occl. ref) - линейный)

Датасет	Census only, px	MIC 0.4 only, px	CNN only, px	Census+ occl. ref., px	MIC 0.4+ occl. ref., px	CNN+ occl. ref., px
Adirondack	3.91	3.42	3.18	1.93	1.55	1.23
Jadeplant	21.24	18.94	20.36	15.25	14.94	15.45
Motorcycle	4.92	4.94	4.56	2.38	2.32	2.08
Piano	4.24	3.81	3.66	2.19	1.99	1.68
Pipes	9.23	8.97	8.11	5.17	4.83	4.51
Playroom	10.55	9.56	9.65	5.09	3.29	3.20
Playtable	10.21	9.00	5.82	7.37	6.74	2.92
Recycle	3.24	2.94	2.76	1.38	1.28	1.20
Shelves	7.25	5.42	5.82	5.67	4.73	4.62
Взв. среднее	**8.17**	**7.33**	**7.00**	**5.05**	**4.54**	**4.04**

Таблица 8: Точность в терминах ошибки bad 1.0 (отклонение от gt менее пикселя) , полученная на датасете Middlebury при использовании адаптивных весов P1 и P2 на основе градиента, используя $W_{P_2} = 70$ и 50

Датасет	Без использования, %	Dep. weights 70, %	Dep. weights 50, %
Adirondack	75.26	75.13	75.07
Jadeplant	61.99	62.19	62.22
Motorcycle	78.75	78.78	78.73
Piano	74.92	75.01	75.03
Pipes	71.74	71.81	71.80
Playroom	62.13	62.13	62.09
Playtable	45.01	44.67	44.51
Recycle	76.72	76.81	76.83
Shelves	43.12	43.16	43.15
Взв. среднее	**65.98**	**65.99**	**65.96**

Таблица 9: Ошибка avgerr (среднее отклонение от gt в пикселях), полученная на датасете Middlebury при использовании адаптивных весов P1 и P2 на основе градиента, используя $W_{P_2} = 70$ и 50

Датасет	Без использования, px	Dep. weights 70, px	Dep. weights 50, px
Adirondack	3.91	3.93	3.94
Jadeplant	21.24	21.15	21.15
Motorcycle	4.92	4.89	4.90
Piano	4.24	4.26	4.26
Pipes	9.23	9.17	9.17
Playroom	10.55	10.54	10.54
Playtable	10.21	10.32	10.38
Recycle	3.24	3.23	3.23
Shelves	7.25	7.27	7.28
Взв. среднее	**8.17**	**8.17**	**8.18**

5. Применение стереометода на изображениях лиц людей

Как было выяснено в ходе работы, не существует общедоступного стерео-датасета с хорошо откалиброванными камерами, главным объектом съемки в котором были бы лица людей. Помимо этого, встает вопрос, как измерять качество применения используемого алгоритма при отсутствии ground truth.

В следствии данных проблем, было решено прибегнуть к использованию плохооткалиброванного датасета, предоставленного [8] Бирмингемским университетом. Погрешности ректификации после калибровки камер оказались намного больше, нежели погрешности при проведении планарной ректификации с неизвестными параметрами камер с помощью метода [12]. Также алгоритм предоставил оценку внутренних параметров камер, необходимых для соблюдения масштаба между осями при построении 3D модели.

Основной проблемой при построении карты глубин человеческого лица на основе двух камер служат перекрытия. В следствии этого был использован описанный ваше алгоритм для решения этой проблемы. В качестве основного направления экстраполяции было выбрано bottom-to-top, так как оно точнее всего применимо для экстраполяции на наборе плоскостей, составляющих форму лица. Перекрытые участки, экстраполяция которых была проведена на основе слишком удаленных регионов, были выброшены из рассмотрения и не участвовали в 3D проектировании (сравнение полученных результатов с применением экстраполяции и без приведено на рисунке 16).

При проектировании в 3D использовались оба вида (и источник, и цель). Сделано это с помощью проектирования обоих видов на одну расширенную плоскость источника. Каждой точке такой плоскости может соответствовать несколько точек цели, что выражается в том, что каждой такой точке может соответствовать несколько Z-координат. На рисунке 17 изображена проекция карты смещений правого изображения на плоскость левого (вид левого), а также совместное распределение то-

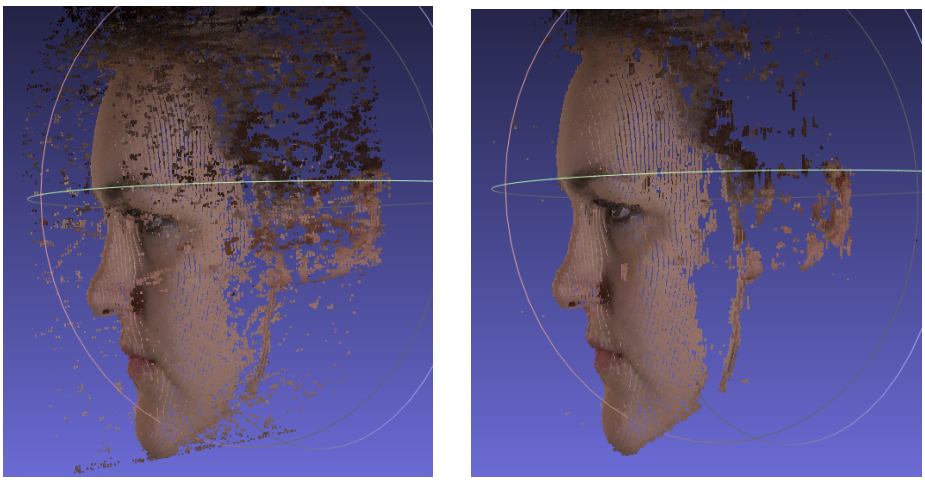

Рис. 16: Сравнение полученных результатов с применением экстраполяции (справа) и без (слева), функция стоимости census

чек в пространстве, где красным обозначены точки левого вида, синим - правого.

Дальнейшее проектирование происходило по следующим формулам для точки P (сначала введем необходимые обозначения для матриц внутренних параметров $calib_l$ и $calib_r$ соответственно левой и правой камер):

$$calib_l = \begin{bmatrix} f_{xl} & 0 & c_{xl} \\ 0 & f_{yl} & c_{yl} \\ 0 & 0 & 1 \end{bmatrix}, \qquad calib_r = \begin{bmatrix} f_{xr} & 0 & c_{xr} \\ 0 & f_{yr} & c_{yr} \\ 0 & 0 & 1 \end{bmatrix},$$

$$v = calib_l * \begin{bmatrix} P_x \\ P_y \\ 1 \end{bmatrix},$$

где f_{xl}, f_{xr} - фокусные расстояния по оси X проективной плоскости соответственно левой и правой камер (аналогично для оси Y, учитывая, что сенсорные элементы камер могут быть не квадратными); c_{xl}, c_{xr} - главные точки левой и правой камер соответственно по оси X (аналогично для оси Y).

Тогда координаты (x, y, z) точки P в пространстве будут:

$$z = \frac{f_{xr}}{P_x + P_{disp} - (calib_r * v)_0} \qquad x = z * v_0 \qquad y = z * v_1$$

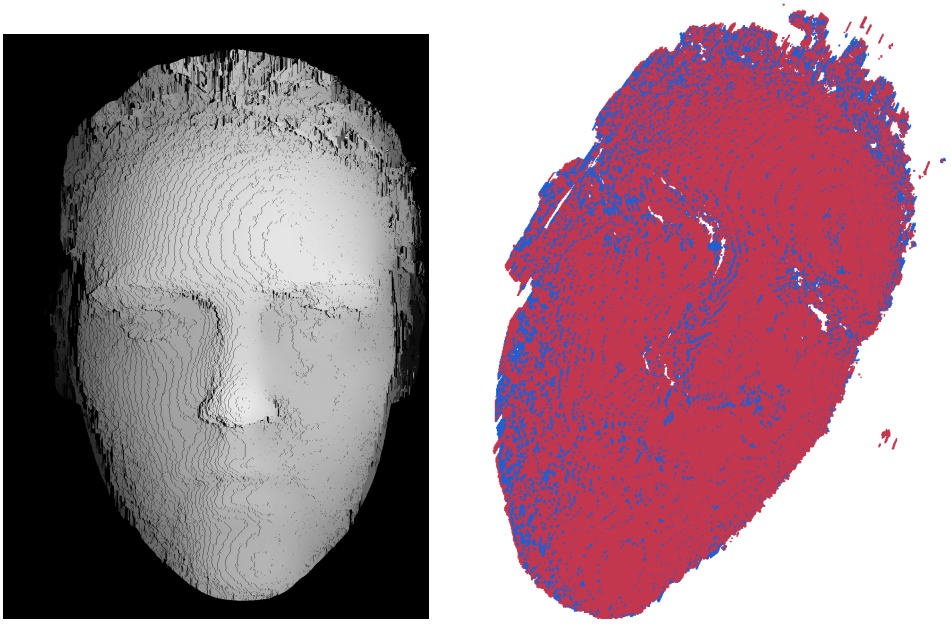

Рис. 17: Проекция карты смещений правого изображения на плоскость левого и совместное распределение точек в пространстве

Экспорт модели происходит в формате PLY с заданием цвета для вершин. Помимо этого, было триангулировано облако точек по следующей схеме (результатом служил меш). Рассмотрим четыре пикселя-соседа в шести возможных конфигурациях (рис. 18).

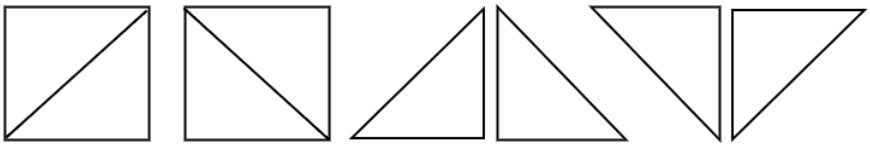

Рис. 18: Возможные конфигурации соседних пикселей

Когда два соседних пикселя имеют глубину, отличающуюся более чем на заданную величину, наблюдается разрыв глубин. Эта заданная величина задается заранее, как минимально возможная разница глубин, считающаяся разрывом поверхностей. Если разрыв есть, треугольник не должен быть создан. Таким образом, для четырех соседних точек рассматривались лишь те, между которыми разрыва нет. Если между всеми четырьмя из них нет разрывов, будет создано два треугольника, что изображено на первых двух вариантах рисунка 18. Причем из этих двух вариантов будет выбран тот, в котором разница между глубинами соединенных диагональных точек наименьшая. В

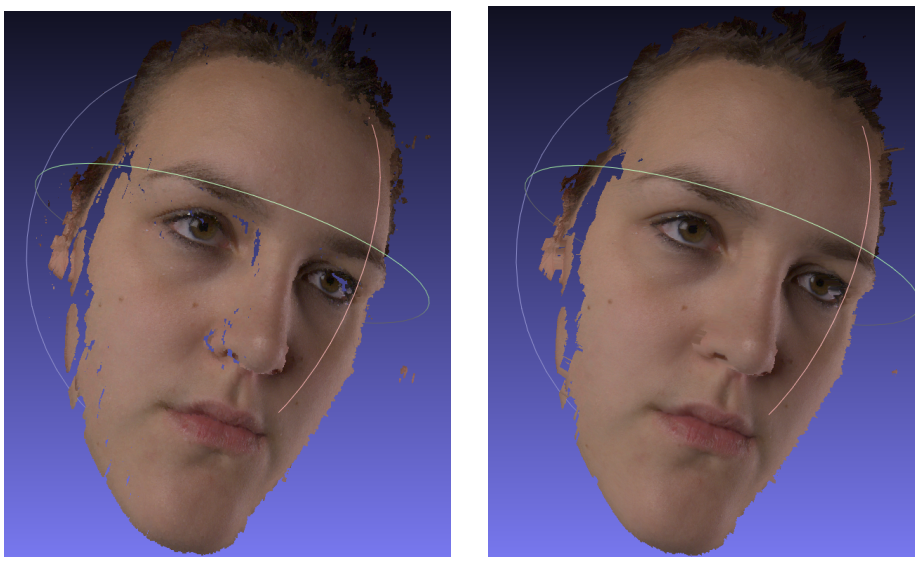

Рис. 19: Облако точек и примененная к нему триангуляция

противном случае, если есть три точки, между которыми разрыва нет, будет создан один из четырех последних треугольников на рисунке 18. На рисунке 19 приведено сравнение результатов в виде облака точек и примененной к нему триангуляции.

В дальнейшем полученные полигональные сетки были импортированы в среду MeshLab для лучшей визуализации результатов. Были проверены две функции стоимости - census и MIC. Использование CCNC было оставлено для дальнешей работы, поскольку для достижения наилучших результатов стоит обучить CCNC непосредственно на изображениях лиц людей, что в отсутствие ground truth, не представляется возможным. На данном этапе были проанализированы только визуальные характеристики реконструированных поверхностей (более формальным результатам посвящен следующий пункт работы). Использование функции стоимости MIC продемонстрировало бóльшую визуальную устойчивость к шумам (неформально, наличие шумов прежде всего характеризуется метрикой avgerr, нежели bad 1.0, что было выяснено в следующей части работы), нежели census (см. рис. 20, 21).

Многие авторы, рассматривая в качестве стереопары изображения лиц людей, вводят дополнительные ограничения на гладкость, чтобы избежать шумов на источнике или цели. Например, что глубина двух соседних пикселей не должна отличаться более чем на один пиксель.

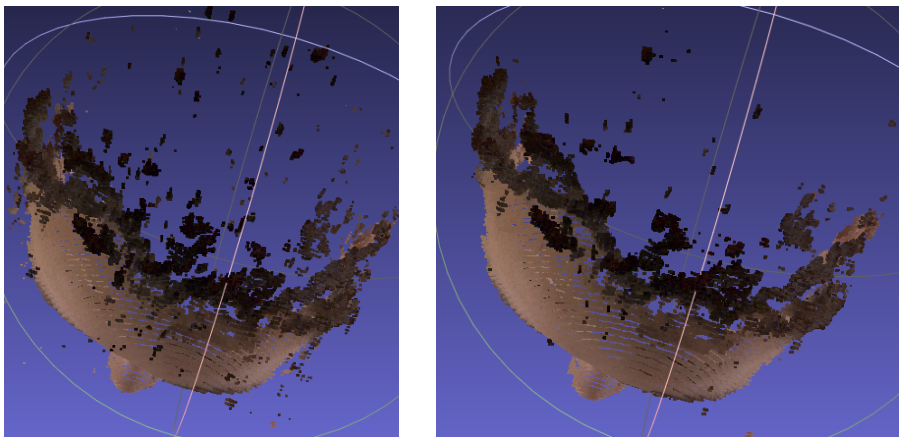

Рис. 20: Рисунок демонстриует бóльшую визуальную зашумленность результата при использовании функции стоимости census (слева), нежели MIC (справа) , без триангуляции

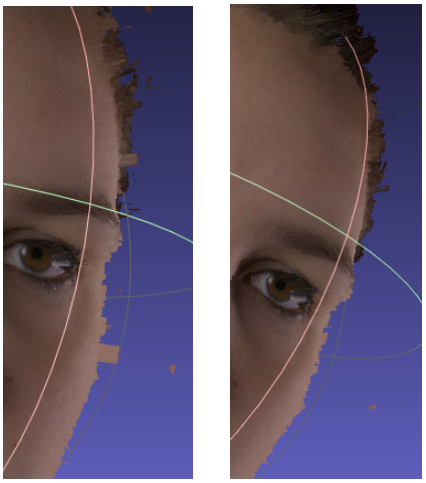

Рис. 21: Рисунок демонстриует бóльшую визуальную зашумленность результата при использовании функции стоимости census (слева), нежели MIC (справа) , с триангуляцией

Это ограничение можно имитировать путем правильного выбора параметров $P1$ и $P2$ в формуле 21. Первый из этих параметров контролирует стоимость "прыжка" в смещении на единицу, тогда как второй - стоимость "прыжка" произвольной глубины. Таким образом, сделав $P1$ малым относительно функции стоимости, а $P2$ - относительно большим, можно контролировать гладкость решения. При выборе двух ракурсов камер, как у нас, гладкость может теряться лишь рядом с участками носа и ушей.

Мы начали с использования $P1$=160 и $P2$=320. Затем значение $P1$

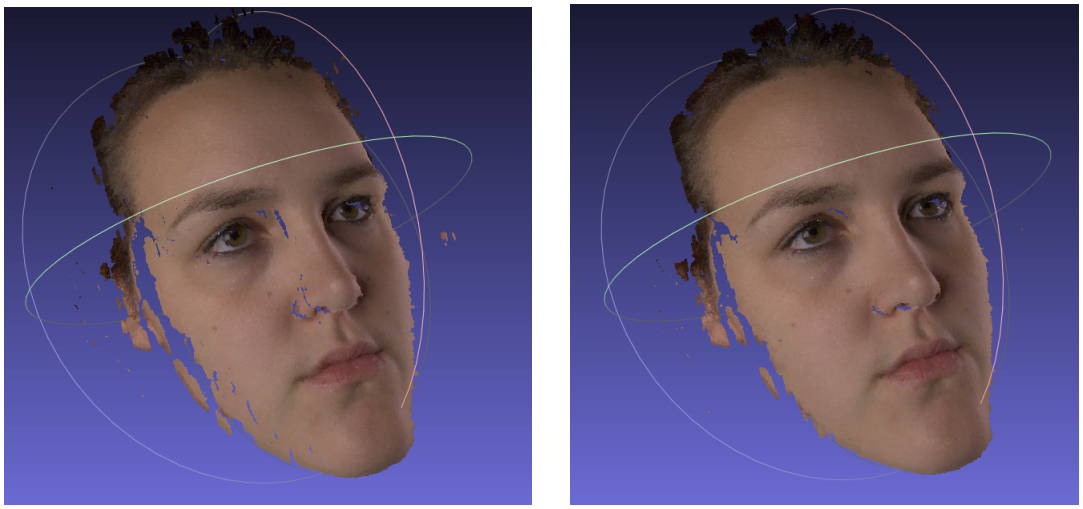

Рис. 22: Рисунок демонстриует бóльшие разрывы реконструированной поверхности в области носа при использовании $P1$=160 и $P2$=320 (слева), нежели $P1$=80 и $P2$=320 (справа)

было уменьшено до 80. Можно заметить, что эвристика полностью работает: многие участки лица, имеющие разрывы при изначальном выборе коэффициентов, стали менее разрывны. Также заметные улучшения наблюдаются при увеличении значения $P2$ до 520. Сравнения результатов приведены на рисунках 22 и 23.

5.1. Применение стереометода на изображениях лиц людей с известной 3D моделью

В конце работы нами все же был получен один набор изображений (который не является открытым), содержащий ground truth 3D модель лица и полученный с помощью хорошо откалиброванных камер [5]. Мы спроектировали глубины точек 3D модели на левую и правую плоскости камер, предварительно ректифицировав изображения и, соответственно, получив новые параметры камер. Также была построена карта смещений и карта перекрытий, как это сделано во всех классических датасетах.

По итогам анализа полученных результатов, оказалось, что функция Census оказалась точнее MIC по метрике bad 1.0 (отклонение не более пикселя, табл. 10), однако по метрике avgerr (которая демонстри-

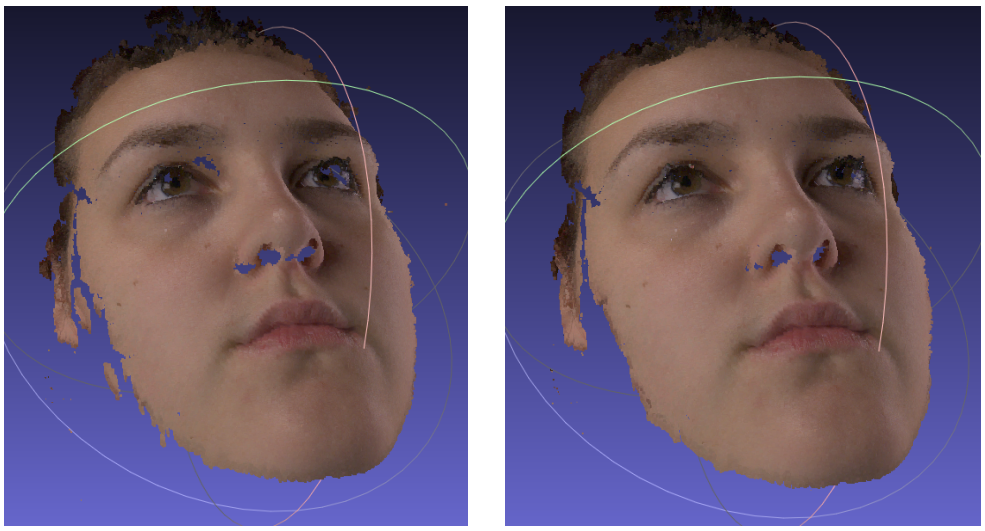

Рис. 23: Рисунок демонстриует бóльшие разрывы реконструированной поверхности в области носа при использовании $P1{=}80$ и $P2{=}320$ (слева), нежели $P1{=}80$ и $P2{=}520$ (справа)

рует зашумленность и более заметна глазу при просмотре 3D модели, таблица 11) MIC все же оказалась точнее. Функция стоимости CNN в экспериментах не участвовала, так как отсутствовало достаточное число дополнительных изображений для тренировки нейронной сети.

Наши предыдущие рассуждение о влиянии выбора весов $P1$ и $P2$ на точность подтвердились в обеих метриках. Неожиданностью стало то, что несмотря на зашумленность модели без использования экстраполяции, точность при использовании экстраполяции (как линейной, так и голосующей) в терминах bad 1.0 метрики оказалась ниже. Это можно связать с низким перепадом в глубинах точек модели лиц, тогда как экстраполяция демонстрирует лучшие результаты, когда такие перепады существенны.

Таблица 10: Результаты применения (по метрике отклонение не более пикселя) метода с различными параметрами на наборе изображений, содержащем лицо человека

Выбранные параметры			Все изобр., %			Перекрытые регионы, %		
Ф-я стоимости.	P1	P2	Без обр.	Лин.	Голос.	Без обр.	Лин.	Голос.
MIC	160	320	69.83	68.85	66.71	5.56	4.74	4.11
MIC	80	320	75.86	74.71	72.72	6.00	4.96	2.94
MIC	80	520	76.43	74.88	73.40	5.95	4.78	3.16
Census	160	320	71.75	71.04	69.03	4.39	4.42	3.14
Census	80	320	76.55	75.86	74.19	4.80	4.94	2.84
Census	80	520	77.82	76.73	75.25	5.39	4.82	3.21

Таблица 11: Результаты применения (по метрике avgerr) метода с различными параметрами на наборе изображений, содержащем лицо человека

Выбранные параметры			Все изобр., px	Неперекрытые регионы, px
Ф-я стоимости	P1	P2	No occl.	No occl.
MIC	160	320	4.43	2.97
MIC	80	320	3.60	2.30
MIC	80	520	3.71	2.28
Census	160	320	7.71	5.56
Census	80	320	5.56	3.59
Census	80	520	4.24	2.65

Заключение

В ходе работы были выполнены следующие задачи:

1. Адаптация MGM [4] под библиотеку OpenCV
2. Разработка и реализация алгоритма оценки стоимостей на основе синтеза алгоритмов Census и Mutual Information [6], дабы добиться устойчивости к сложным радиометрическим изменениям на изображениях
3. Разработка и реализация функции стоимостей на основе сиамких сверточных нейронных сетей
4. Разработка и реализация алгоритма поиска перекрытий и устранение их проявлений (а вместе с тем и повышения точности самого алгоритма)
5. Использование адаптивных весов на основе градиента
6. Проверка полученного алгоритма на наборах изображений Middlebury, а также на наборах изображений лиц людей и анализ полученных результатов

В результате нашей работы было показано, что предложенные функции стоимости в сочетании с MGM устойчиво работают на датасете Middlebury (в метриках bad 1.0 и avgerr), представляющем трудности для всех стерео-алгоритмов, по точности превосходя стандартную для MGM функцию стоимости census. Функция стоимости на основе CCHC демонстрирует наилучшую точность, однако в следствие большого объема вычислений и отсутствия реализации на GPU, подходит лишь для датасетов, требующих высокой точности, но не требующих большой скорости вычислений. Линейная экстраполяция перекрытий продемонстрировала многократный прирост точности в перекрытых регионах, что в сочетании со сложностью ее применения $O(W * H)$ делает ее использование рекомендуемым.

Применение предложенных алгоритмом на изображениях лиц людей показало, что результаты, полученные при использовании различных функций стоимости на данном наборе изображений, различны для метрик avgerr и bad 1.0. Выбор той или иной функции стоимости в данном случае должен зависеть от метрики, в которой необходимо получить наилучший результат. Использование экстраполяции на данном датасете не оправдало себя. В качестве дальнейшей работы оставлено обучение и применение новой функции стоимости CNN на наборах изображений, содержащих лица людей.

Список литературы

[1] 3D face recognition using passive stereo vision / N. Uchida, T. Shibahara, T. Aoki et al. // Image Processing, 2005. ICIP 2005. IEEE International Conference on. — Vol. 2. — 2005. — Sept. — P. II–950–3.

[2] Banz C., Pirsch P., Blume H. EVALUATION OF PENALTY FUNCTIONS FOR SEMI-GLOBAL MATCHING COST AGGREGATION // ISPRS. — 2012. — Vol. XXXIX-B3. — P. 1–6. — URL: http://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XXXIX-B3/1/2012/.

[3] Bleyer Michael, Gelautz Margrit. Simple but effective tree structures for dynamic programming-based stereo matching // In VISAPP. — 2008. — P. 415–422.

[4] Facciolo Gabriele, de Franchis Carlo, Meinhardt Enric. MGM: A Significantly More Global Matching for Stereovision. — 2015. — http://dev.ipol.im/ facciolo/mgm/.

[5] High-Quality Single-Shot Capture of Facial Geometry / Thabo Beeler, Bernd Bickel, Paul Beardsley et al. // ACM Trans. on Graphics (Proc. SIGGRAPH). — 2010. — Vol. 29, no. 3. — P. 40:1–40:9.

[6] Hirschmuller Heiko. Stereo Processing by Semiglobal Matching and Mutual Information // IEEE Trans. Pattern Anal. Mach. Intell. — 2008. — . — Vol. 30, no. 2. — P. 328–341.

[7] Inamoto Naho, Saito Hideo. Intermediate view generation of soccer scene from multiple videos // Proceedings - International Conference on Pattern Recognition. — 2 edition. — 2002. — Vol. 16. — P. 713–716.

[8] Jiang Xiaoyue, Schofield Andrew J., Wyatt Jeremy L. Computer Vision – ECCV 2010: September 5-11, 2010, Proceedings, Part IV. — 2010. — P. 58–71. — ISBN: 978-3-642-15561-1.

[9] Kolmogorov V., Zabih R. Computing visual correspondence with occlusions using graph cuts // Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on. — Vol. 2. — 2001. — P. 508–515 vol.2.

[10] Li Y., Huttenlocher D. P. Learning for stereo vision using the structured support vector machine // Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. — 2008. — June. — P. 1–8.

[11] Min D., Sohn K. Cost Aggregation and Occlusion Handling With WLS in Stereo Matching // IEEE Transactions on Image Processing. — 2008. — Aug. — Vol. 17, no. 8. — P. 1431–1442.

[12] Monasse Pascal. Quasi-Euclidean Epipolar Rectification // Image Processing On Line. — 2011. — Vol. 1.

[13] Pattern Recognition: 36th German Conference, GCPR 2014 / Daniel Scharstein, Heiko Hirschmüller, York Kitajima et al. / Ed. by Xiaoyi Jiang, Joachim Hornegger, Reinhard Koch. — Cham : Springer International Publishing, 2014. — P. 31–42. — ISBN: 978-3-319-11752-2.

[14] Pattern Recognition: 36th German Conference, GCPR 2014, Münster, Germany, September 2-5, 2014, Proceedings / Amnon Drory, Carsten Haubold, Shai Avidan, Fred A. Hamprecht / Ed. by Xiaoyi Jiang, Joachim Hornegger, Reinhard Koch. — Cham : Springer International Publishing, 2014. — P. 43–53. — ISBN: 978-3-319-11752-2. — URL: http://dx.doi.org/10.1007/978-3-319-11752-2_4.

[15] Pearl Judea. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. — San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 1988. — ISBN: 0-934613-73-7.

[16] Scharstein D., Pal C. Learning Conditional Random Fields for Stereo // Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on. — 2007. — June. — P. 1–8.

[17] Suzuki Satoshi. Topological Structural Analysis of Digitized Binary Images by Border Following. — 1985. — Vol. 30 (Issue 1) of Computer Vision, Graphics, and Image Processing. — http://www.sciencedirect.com/science/article/pii/0734189X85900167.

[18] Tieleman Tijmen, Hinton Geoffrey. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude // COURSERA: Neural Networks for Machine Learning. — 2012. — Vol. 4. — P. 2.

[19] Zbontar Jure, LeCun Yann. Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches // CoRR. — 2015. — Vol. abs/1510.05970. — URL: http://arxiv.org/abs/1510.05970.

[20] Zhang L., Seitz S. M. Estimating Optimal Parameters for MRF Stereo from a Single Image Pair // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2007. — Feb. — Vol. 29, no. 2. — P. 331–342.

[21] Zhu Ke, d'Angelo Pablo, Butenuth Matthias. EVALUATION OF STEREO MATCHING COSTS ON CLOSE RANGE, AERIAL AND SATELLITE IMAGES // ICPRAM 2012. — 2012. — P. 379–385.