УДК 004.021

АЛГОРИТМЫ АНАЛИЗА ИЗОБРАЖЕНИЙ В СИСТЕМАХ СТЕРЕОЗРЕНИЯ РЕАЛЬНОГО ВРЕМЕНИ

А. А. Крыловецкий, С. И. Протасов

Воронежский государственный университет

Поступила в редакцию 09.11.2010 г.

Аннотация. В настоящее время вычислительная мощность персональных компьютеров позволяет решать в реальном времени задачи стереосопоставления (stereo matching) изображений, формирования и передачи трёхмерных объектов. В работе рассматриваются подходы к оптимизации ранее известных алгоритмов на базе модели случайного Марковского поля, приложение параллельных вычислений, а также предлагаются оригинальные алгоритмы обработки и анализа изображений в системах стереозрения реального времени. Исследованы характеристики алгоритмов с точки зрения вычислительной сложности и достоверности результата.
Ключевые слова: стереосопоставление, системы стереозрения реального времени, модель случайного Марковского поля.
Annotation. Nowadays computational capabilities of personal computers allow to solve stereo matching problem and problems of three-dimentional object reconstruction and transfer in real time. The article discusses approaches to optimization of existing algorithms based on Markov Random Fields, application of parallel computing to the problem and also offers novel algorithms of image processing and analysis in real time stereo-vision systems. The paper presents results of algorithms research in terms of computational complexity and output reliability.
Keywords: stereo matching, stereo-vision systems of real time, Markov Random Fields.

ВВЕДЕНИЕ

Построение трехмерных моделей по набору изображений традиционно разделяют на две последовательные задачи: стереосопоставление и получение пространственных структур по дальнометрическим данным (рис. 1).

Рис. 1. Этапы формирования трехмерной модели по стереопаре.

Входными данными для алгоритмов стереосопоставления является пара ректифицированных (rectified) и избавленных от дисторсий изображений [1]. В работе представлен алгоритм, позволяющий осуществлять быструю ректификацию изображений на основании матрицы соответствия эпиполярных линий.

Задачей алгоритма стереосопоставления является получение дальнометрических данных (ДД) [2]. На основании ДД могут быть построены дальнометрические изображения (ДИ) – изображения, на которых данные о расстоянии до сцены кодируются цветом или яркостью точек (см. рис. 4). Одним из наиболее наглядных типов ДИ является карта диспаритетов (disparity map) [3]. Как указано в [4], большинство алгоритмов стереосопоставления состоят из четырех этапов:

– инициализация оценок (подбор веса каждого пикселя для различных гипотез диспаритетов);

– агрегирование оценок (группировка начальных оценок в пространстве);

– оптимизация диспаритетов (выбор значения диспаритета, оптимизирующего целевую весовую функцию);

– очистка диспаритетов (пост-обработка для устранения ошибок).

Большинство существующих алгоритмов стереосопоставления можно отнести либо к локальным, либо к глобальным. Основное их различие заключается в шаге оптимизации диспаритетов. Локальные алгоритмы используют принцип «победитель забирает все» (WTA), просто выбирая значение диспаритета, минимизирующее агрегированный вес. Глобальные ал-

горитмы оптимизируют более сложные целевые функции, учитывающие зависимости между пикселями, и поэтому требуют для вычисления оптимального решения использования глобальных методов оптимизации, таких как метод распространения доверия [5] или вычисления разреза графа [6]. Вообще говоря, локальные алгоритмы являются менее затратными по времени вычислений, а глобальные алгоритмы формируют более точные дальнометрические данные. Все лучшие с точки зрения точности алгоритмы стереосопоставления относятся к глобальным [7]. Выполнение этих алгоритмов на персональных компьютерах занимает несколько десятков минут для вычисления карты диспаритетов с 16 уровнями по изображениям с разрешением 288 x 384; таким образом, они не применимы в реальном времени. Большинство систем стереозрения, работающих в реальном времени, используют локальные решения на базе GPU [8], однако существуют более универсальные решения в реальном времени, использующие CPU. В предложенной работе рассматриваются модификации локальных алгоритмов стереосопоставления на базе модели случайного Марковского поля, такие как градиентная метрика, предварительная фильтрация изображений и пост-фильтрация диспаритетов, в приложении к задаче стереозрения в реальном времени.

С точки зрения формального описания, задачу построения сеточной модели (gridding) по множеству точек в трехмерном пространстве можно представить следующим образом: получить поверхность **S'**, которая аппроксимирует неизвестную поверхность **S** с использованием множества точек в трехмерном пространстве и с заданной величиной допустимого отклонения и плотностью расположения точек [9]. Полигональная сетка представляет собой совокупность соприкасающихся ребрами граней, обычно треугольных, в совокупности являющихся многогранником. Построение аппроксимирующих сеточных моделей также является крайне ресурсоемкой задачей. Для построения сеточных моделей по неструктурированным данным используют воксельные алгоритмы, такие как marching cubes [10] и его модификации [11,12]. Так как дальнометрические данные, получаемые на выходе алгоритма стереосопоставления, являются структурированными, то построение по ним сеточной модели можно упростить за счет априорного знания о взаимном расположении точек в пространстве.

Следует отметить, что для построения трехмерной модели большое количество вычисленных диспаритетов является избыточным: априорно выбранная детализация модели задает верхнюю границу количества диспаритетов, достаточных для построения модели. Таким образом, предварительное сокращение областей поиска точек для алгоритма стереосопоставления может заметно уменьшить объем вычислений. Основанием для введения описываемой модификации может служить низкая достоверность диспаритетов, получаемых с использованием монотонных областей изображения. Предварительный отбор точек сокращает количество входных данных на каждом из последующих этапов алгоритма.

Описываемые в работе алгоритмы были реализованы на платформе .NET 2.0 (MONO) (язык C# 3) с использованием небезопасного (unsafe) кода, а также на языке ANSI-C с применением компилятора GCC. Испытания проводились под управлением 64-разрядных версий ОС Windows 7 и Ubuntu 10.04 на 4-ядерном процессоре Core i7 с включенной поддержкой технологии Hyper-threading. Для проверки алгоритмов использовались изображения из базы данных института Middlebury [13].

1. БЫСТРАЯ РЕКТИФИКАЦИЯ ИЗОБРАЖЕНИЙ

Одной из центральных задач в области машинного зрения является определение ориентации и расположения камеры в пространстве по изображению, полученному с ее помощью. Рассмотрим наиболее часто используемую математическую модель калибровки камеры. Пусть P – некоторая точка трехмерного пространства, а I – ее проекция на плоскость изображения. Путь в некоторой однородной системе координат P описывается вектором $[p_x, p_y, p_z, 1]^T$, а точке I соответствует вектор $[u, v, 1]^T$. Тогда общую модель камеры можно описать с помощью 3×4 матрицы M, задающей отображение точек трехмерного пространства, попадающих в угол обзора камеры, на плоскость изображения.

$$\lambda I = MP, \qquad (1)$$

где λ – некоторая нормирующая константа.

Решением задачи калибровки являются вычисленная матрица M [14].

Данные калибровки для каждой из камер чрезвычайно важны, однако не являются ис-

черпывающими для подготовки системы к функционированию. Необходимо также описать связь между координатными системами камер. Такой связью может служить матрица G_{LR}, такая что $P_R = G_{LR}P_L$, описывающая преобразование координат точки P_L в системе первой камеры в координаты P_R второй камеры.

Получив матрицу M для каждой из камер, а также матрицу G_{LR}, мы можем перейти к ректификации изображений. Под ректификацией в данном случае подразумевается установление соответствий между эпиполярными линиями на стереопаре, вдоль которых будет выполняться поиск в ходе работы алгоритма стереосопоставления. Для нахождения этих сечений мы использовали следующий метод. Исходя из общей особенности матрицы M, все точки $P_{L'}$, лежащие на некоторой прямой L, проходящей через центр линзы, будут иметь в проекции на изображение одни и те же координаты I. Этим, в частности, объясняется эффект наложения (occlusion effect). Используя выражение (1) и считая координаты I какого-либо пикселя изображения известными, мы можем восстановить уравнение прямой L, такой, что

$$\forall(P : \lambda I = MP)[P \in L]. \qquad (2)$$

Выберем линию поиска (scanline) на левом изображении. Обычно, для удобства и ускорения работы алгоритма, выбирают прямую, совпадающую с горизонтальной строчкой пикселей изображения [15]. Далее для двух точек I_1 и I_2, принадлежащих этой прямой $N(I_1, I_2)$ рассчитаем уравнения соответствующих им прямых L_1 и L_2. Используя матрицу G_{LR}, повернем эти прямые в координатную систему правой камеры. Описанная полученными прямыми плоскость сечет плоскость правого изображения по прямой $N(I_3, I_4)$. Эта прямая и будет искомой линией поиска для правой камеры. Получив уравнения соответствующих прямых для стереопары, мы можем построить дискретную функцию $E : N^2 \to N$, определяющую для каждой пары (x_R, y_L) значение y_R. Получившуюся прямоугольную матрицу можно рассчитать однократно и использовать для быстрой ректификации изображений.

2. ИНТЕГРАЛЬНЫЕ ГРАДИЕНТНЫЕ ИЗОБРАЖЕНИЯ

Введем понятие интегрального градиентного изображения (ИГИ). Под ИГИ будет понимать особым образом сформированное изображение: для всех точек исходного изображения рассчитывается сумма модулей градиента по каждому компоненту цвета в модели RGB. Данная характеристика описывает, насколько «немонотонной» окружностью обладает та или иная точка изображения.

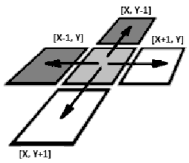

Рис. 2. Иллюстрация для расчета ИГИ

В целочисленном представлении значение яркости каждого пикселя ИГИ находится следующим образом:

$$F(x,y) = \sum_{i,j=\pm1} \sum_{C\in\{R,G,B\}} \left|D_{(i,j),C}(x,y)\right|,$$
$$D_{(i,j),C}(x,y) = (P_{x+i,y+j})_C - (P_{x,y})_C. \qquad (3)$$

Индексы (i, j) однозначно задают вектор компоненты базиса, на который проецируется градиент, а функция D соответствует значению проекции вектора градиента с точностью до линейного коэффициента. Данная функция неотрицательна, имеет целочисленные значения, для 24-битного представления цвета не превышающие 3060 (для хранения достаточно 2 байт) и обращающиеся в ноль в случае абсолютно монотонной окрестности точки. Кроме того, данная функция быстро вычисляется, благодаря тому, что не использует операций с плавающей точкой.

Отбор точек в модели RGB осуществляется с помощью характеристической функции $F : P_{(x,y)} \to R_+$ на множестве точек $P_{(x,y)} = (P_R, P_G, P_B)_{(x,y)}$ изображения, описывающей особенность точки, а также порог, который необходимо превысить данной функции для того, чтобы точка попала в сокращенную область поиска. В нашем случае роль характеристической функции играет ИГИ: значение ИГИ в каждой точке дает нам значение функции.

3. АЛГОРИТМЫ СТЕРЕО-СОПОСТАВЛЕНИЯ

В качестве функции подобия точек в алгоритме стереосопоставления будем использовать

$$\Phi(P^L_{x,y}, P^R_{u,v}) = \sum_{C\in\{R,G,B\}} \left(k_1 \left|P^L_{x,y} - P^R_{u,v}\right| + k_2 \sum_{i,j=\pm1} \left|D^L_{(i,j),C} - D^R_{(i,j),C}\right|\right), \qquad (4)$$

(а) (б)

Рис. 3. ИГИ. а) исходное изображение; б) градиентное изображение

где верхние индексы L и R относят точки соответственно к левому и правому изображениями, а коэффициенты k_1 и k_2 используются для приведения значений функции к интервалу [0, 1] и задания «значимости» компонент цвета центрального пикселя. В приведенных далее исследованиях применялись экспериментально выбранные значения коэффициентов $k_1 = k_2 = 0{,}2$.

Каждый их выделенных подэтапов обладает высокой вычислительной сложностью. Для оценки сложности подэтапов алгоритма введем следующий вектор входных значений:

$$I = \langle w, h, Tp, c, N_1, N_2, Th, d \rangle, \qquad (5)$$

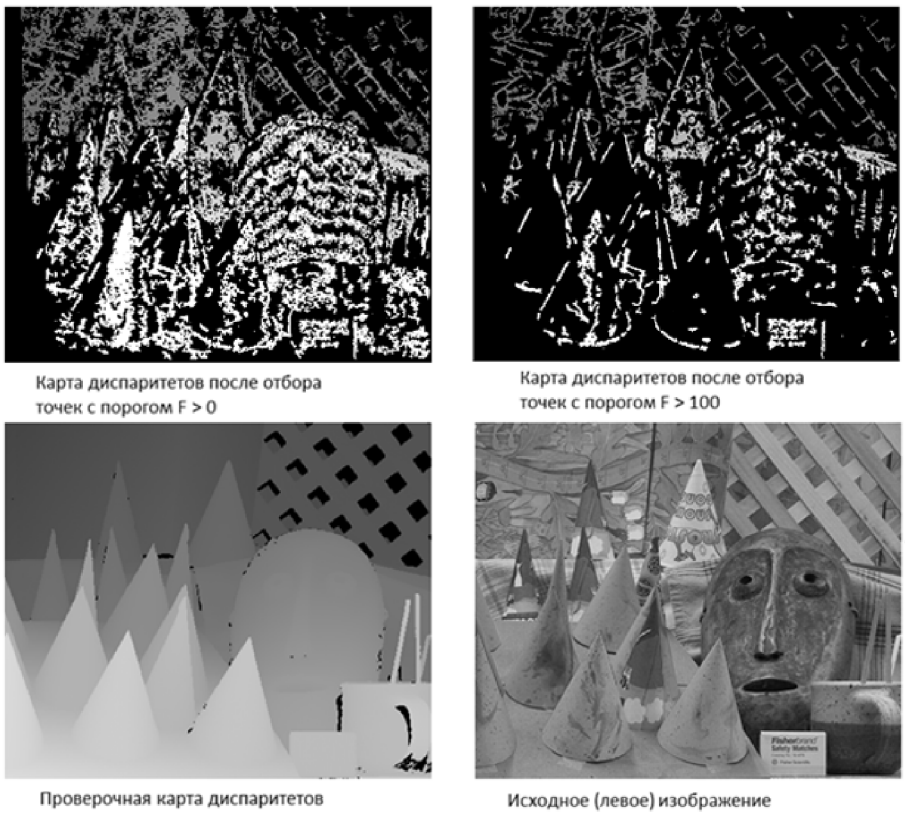

Рис. 4. Влияние предварительного отбора точек на площадь карты диспаритетов.

где w, h – ширина и высота входных изображений; Tp – размерность данных, представляющих координаты точек; c – количество компонент цвета изображения; N_1, N_2 – количество точек, полученных при предварительной фильтрации изображения, такое что $N_i \le wh$; Th – количество задействованных процессоров или ядер процессора; d – размерность значений для характеризующей функции точки.

Подэтап предварительного отбора точек изображения будет требовать для выполнения времени, прямо пропорционального площади изображения

$$S_0 \in \Theta(w \cdot h) \tag{6}$$

и требовать количества оперативной памяти, равного сумме памяти, необходимой для хранения двух изображений, и памяти необходимой для хранения

$$M_0 \approx 2 \cdot w \cdot h \cdot c + \\ +(N_1 + N_2) \cdot d + const \tag{7}$$

машинной памяти (в байтах), однако, вследствие введенного марковского свойства предложенный алгоритм должен хорошо распараллеливаться, и для Th-ядерной системы время вычислений будет обратно пропорционально количеству задействованных ядер

$$Sp_0 \approx \frac{S_0}{Th}. \tag{8}$$

Для иллюстрации оценки был произведен замер времени работы алгоритма на 1, 2, 4 и 8 (была включена технологии HT) ядрах процессора, результаты которого приведены на рис. 5.

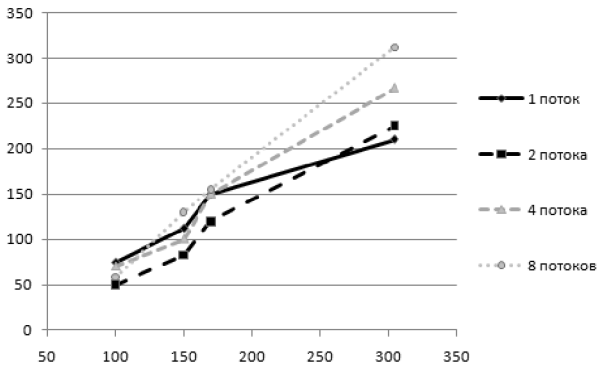

Рис. 5. Зависимость времени работы алгоритма предварительного отбора точек от площади входного изображения (cones)

График подтверждает зависимость (6), однако не соответствует предположению (8). Можно сделать вывод, что ввиду небольшого времени работы алгоритма накладные расходы на сознание объектов потоков на платформе .NET нивелируют выгоду от распараллеливания. Таким образом, для изображений с невысоким разрешением (максимальное рассматриваемое разрешение – 826px × 740px) оптимальным будет отказ от распараллеливания алгоритма предварительного отбора точек.

При вычислении характеристической функции (3) для описания каждой точки изображения используется вектор из 15 целочисленных значений. Для уменьшения времени поиска диспаритетов предлагается в ходе предварительного отбора точек сократить размерность представления точки или преобразовать данные к виду, позволяющему осуществить сравнение с использованием формулы (4) существенно быстрее, исключив повторные обращения к области памяти, содержащей изображения. Для этого в ходе отбора точек предлагается вычислять и хранить значение функции D (3).

После того, как была сокращена область поиска, пара изображений подается на вход алгоритма стереосопоставления. Условие ректифицированности изображений и Марковское свойство модели также позволяют построить предположение о хорошей распараллеливаемости задачи стереосопоставления. Можно предположить, что для алгоритма с учетом уменьшенной области поиска будут справедливы следующие оценки временной сложности: время работы алгоритма сверху ограничено высотой одного изображения, умноженного на квадрат ширины изображения (при полном переборе пикселей двух эпиполярных линий), а время работы алгоритма снизу ограничено случаем равномерного распределения областей поиска по обоим изображениям ($\frac{N_1}{h}$ и $\frac{N_2}{h}$ пикселей в строке соответственно).

$$S_1 \in O(w^2 \cdot h); \\ S_1 \in \Omega\left(\frac{N_1 \cdot N_2}{h}\right). \tag{9}$$

Объем используемой машинной памяти можно оценить (в байтах) как сумму памяти, необходимой для хранения входных данных, и памяти, необходимой для хранения результирующей карты диспаритетов, площадь которой не больше площади любой из областей поиска:

$$M_1 \approx (N_1 + N_2) \cdot d + \\ + Tp \cdot \min(N_1, N_2) + const. \qquad (10)$$

Опираясь на аналогичную (6) гипотезу предположим время работы алгоритма в случае параллельного выполнения [18] оценивается величиной:

$$Sp_1 \approx \frac{S_1}{Th}. \qquad (11)$$

Зависимость правильно определяемых алгоритмом диспаритетов от объема входных данных (после предварительного отбора точек) изображена на рис. 4. Представленная зависимость показывает некоторое снижение процента корректности определения диспаритетов при уменьшении объема входных данных, однако видно, что при снижении объема входных данных в 2 раза потеря точности на тестовом изображении составляет не более 2%.

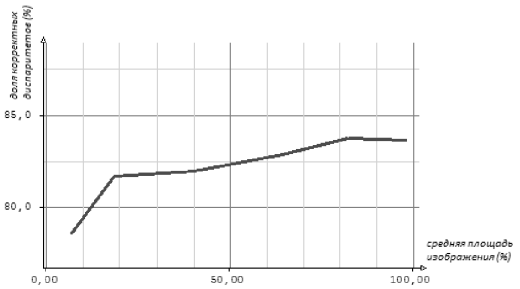

Рис. 6. Зависимость доли корректно определяемых диспаритетов от объема входных данных (aloe)

Для иллюстрации предположений (9, 11) построим зависимость времени работы алгоритма стереосопоставления от величины $\frac{N_1 \cdot N_2}{h}$, поскольку эта величина описывает нижнюю границу вычислительной сложности алгоритма, а значит в лучшем случае результирующий график будет близок к линейной зависимости (рис. 7).

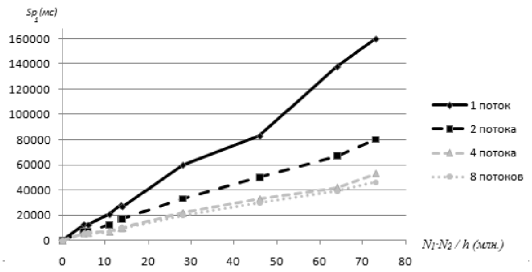

Рис. 7. Зависимость времени работы алгоритма стерео-сопоставления от объема входных данных (cones)

Из рис. 7 видно, что время работы алгоритма Sp_1 прямо пропорционально параметру $\frac{N_1 \cdot N_2}{h}$, то есть лежит вблизи нижней границы оценки временной сложности. Кроме того, распараллеливание задачи дает существенный выигрыш во времени. Лучшие результаты для 8 потоков при использовании 4-ядерного процессора можно объяснить применением технологии Hyper-threading, однако выигрыш по отношению к 4 потокам является несущественным (10–15 % времени).

Одним из предполагаемых способов улучшения точности определения диспаритетов является метод встречного стереосопоставления [17]. Данный метод заключается в том, что соответствия для пары изображений ищутся в обоих направлениях: на правом изображении для левого и наоборот. Корректно определенными считаются только те соответствия, которые одинаковы для обоих проходов. Оценка временной сложности такого алгоритма будет ограничена сверху величиной, прямо пропорциональной высоте одного изображения и кубу ширины изображения (время поиска соответствий для всех пикселей одной строки умноженное на время верификации), а нижняя оценка описывается равномерным распределением областей поиска по изображениям сходно с (9)

$$S_1^* \in O(w^3 \cdot h); \\ S_1^* \in \Omega\left(\frac{N_1^2 \cdot N_2}{h^2}\right). \qquad (12)$$

Оценка временной сложности для параллельного вычисления будет аналогична (11). На рис. 8 видно, что время работы такого алгоритма увеличивается примерно в 2 раза, а количество корректно определяемых диспаритетов

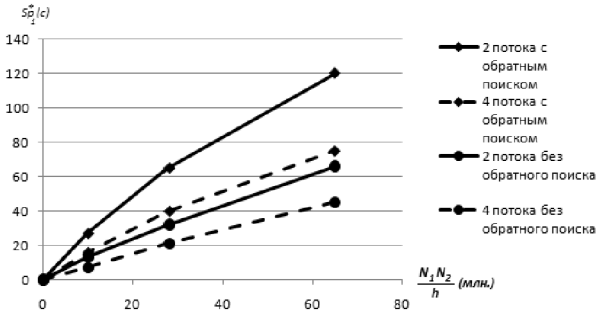

Рис. 8. Сравнение работы алгоритма стерео-сопоставления с использованием обратного прохода и без (aloe)

после определенного объема входных данных начинает уменьшаться. Можно предположить, что корректно определенные диспаритеты выбрасываются из-за достаточно высокого процента ошибок.

Таким образом, использование подобной модификации алгоритма не вносит видимых улучшений в результат, хотя и сокращает количество ошибочных определений диспаритетов.

4. ПОСТ-ФИЛЬТРАЦИЯ ДИСПАРИТЕТОВ

Еще одним способом уменьшения доли неверно определенных диспаритетов является пост-фильтрация. Для решения этой задачи нами был разработан алгоритм, опирающийся на гипотезу связности диспаритетов [18]. Основной идеей данного алгоритма являет послойная фильтрация диспаритетов и исключение связных областей с площадями, не превышающим заданный порог, которые можно принять за несистематическую ошибку вычислений. На рис. 9 показаны зависимости характеристик выходных данных от площади исключаемых областей. Слева показано, как снижается количество корректно определенных диспаритетов по отношению к площади изображения, а справа – как увеличивается доля корректно определяемых диспаритетов. Зависимости, представленные на рис. 9, показывают, что при потере корректно вычисленных диспаритетов в 10% для порога алгоритма в 15 пикселей можно снизить долю ошибочно определенных диспаритетов до 3%. Время работы алгоритма существенно зависит от степени сегментированности изображения. Временная сложность алгоритма прямо пропорционально площади изображения, а также прямо пропорциональна количеству фильтруемых слоев:

$$S_2 \in \Theta(w \cdot h \cdot disp), \qquad (13)$$

где *disp* – размер диапазона диспаритетов, для которых осуществляется фильтрация; ‘та величина либо известна заранее, либо равна разности максимального и минимального определенных диспаритетов.

Особенность реализации алгоритма позволяет выполнять фильтрацию слоев диспаритетов в параллельных потоках. Таким образом, оценка временной сложности алгоритма будет обратно пропорциональна числу *Th* ядер в системе. В табл. 1 приведена зависимость времени работы алгоритма от количества задействованных ядер процессора.

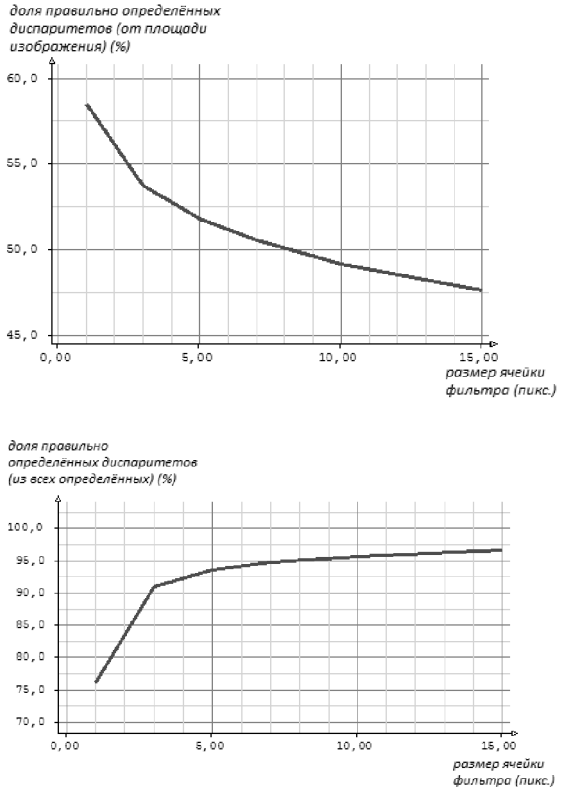

Рис. 9. Характеристики алгоритма пост-фильтрации (aloe)

Таблица 1

Зависимость времени работы алгоритма пост-фильтрации от количества задействованных ядер процессора (изображение aloe)

число ядер	1	2	4	8*(HT)
время (с)	4.702269	3.0741758	2.0141152	1.8721071

При использовании предварительной фильтрации изображений вместе с пост-фильтрацией диспаритетов оценка временной сложности равна максимумам верхней и нижней оценок подэтапов:

$$S_{sum} = \max(S_1, S_2, S_3) = S_1. \qquad (14)$$

На рис. 10 приведено сравнение результатов работы алгоритма, использующего предложенные нами интегральные градиентные изображения и соответствующую пороговую функцию отбора точек (3) и функцию подобия (4) с результатами алгоритма, использующего корреляцию девятиточечной окрестности и отбор точек по значению дисперсии в окрестности. Результаты для обоих алгоритмов

приведены после пост-фильтрации с размером ячейки 15.

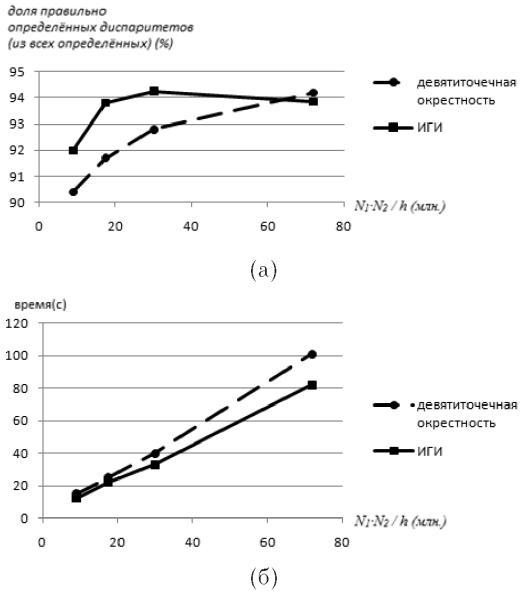

Рис. 10. Сравнение работы градиентного алгоритма с алгоритмом, использующим восьмиточечную окрестность

Из рис. 10 можно сделать вывод, что использование ИГИ позволяет уменьшить объем вычислений в ходе стереосопоставления при сохранении доли верно определяемых диспаритетов.

5. ВЫДЕЛЕНИЕ МНОЖЕСТВА ТОЧЕК ДЛЯ ПОСТРОЕНИЯ МОДЕЛИ

Для входных изображений даже с невысоким разрешением количество точек полной карты диспаритетов существенно превышает необходимое для построения грубой трехмерной модели в реальном времени, что позволяет обратиться к применению отбора некоторого множества точек, на основании которых можно будет рассчитать, например, сеточную трехмерную модель. Так как отбор точек напрямую связан с последующей детализацией модели, то одним из возможных критериев, применяемых к отбору точек можно считать их «равномерность» распределения в пространстве. Для этого на получаемую карту диспаритетов можно наложить сетку с определенным размером ячейки и для каждой ячейки оставлять только одно значение диспаритета. Выбор оставляемой точки можно осуществлять по критерию близости к центру ячейки или по минимуму ошибки сопоставления. На рис. 11 представлена характеристика алгоритма отбора точек. Можно заметить, что при двукратном сокращении объема входных данных выходной объем данных алгоритма не уменьшается. Это хорошо оправдывает выбранный метод предварительного ограничения области поиска. Для ячеек, в которые не попала ни одна точка из карты диспаритетов, можно применять линейную интерполяцию и экстраполяцию.

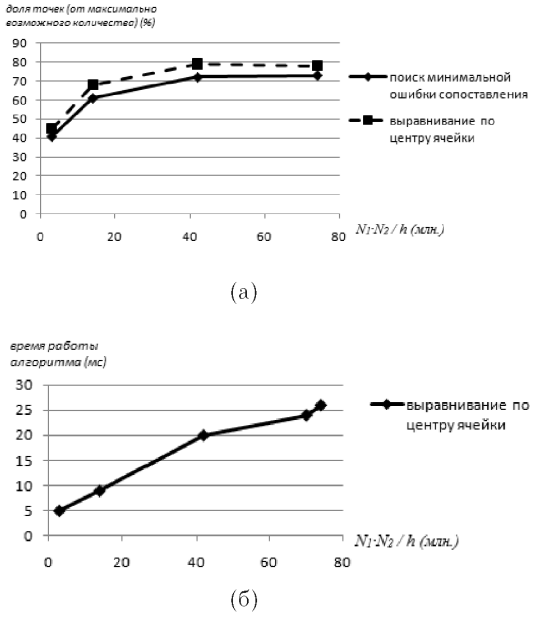

Рис. 11. Отбор точек

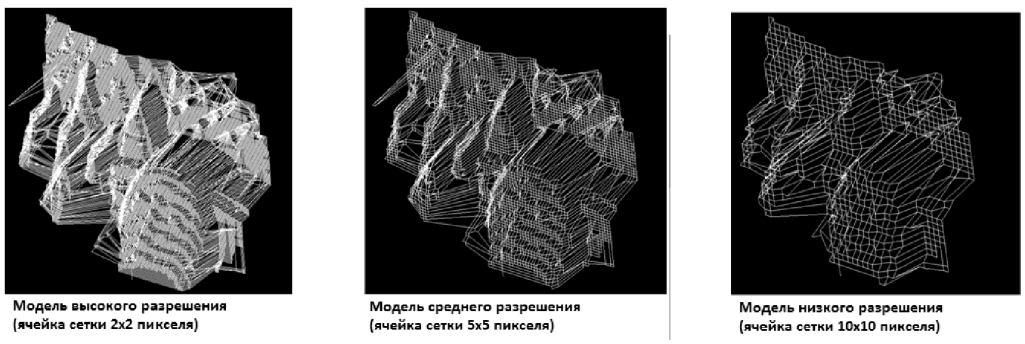

Рис. 12. Сеточные модели, построенные на основании карт диспаритетов

На рис. 12 представлены варианты сеточной моделей, полученные с использованием карт диспаритетов для рис. 3 и ячеек отбора 2×2, 5×5 и 10×10.

ЗАКЛЮЧЕНИЕ

В настоящей работе рассмотрены различные подходы к решению задачи анализа изображений в системах стереозрения реального времени на основе модели случайного Марковского поля. Предложен алгоритм быстрой ректификации изображений, основанный на предварительном расчете матрицы соответствий для эпиполярных линий. На этапе подготовки входных данных предложено производить сокращение площади исходных изображений. Для этого введено понятие ИГИ в сочетании с пороговой функцией для отбора точек входных изображений. Использование ИГИ позволило осуществить уменьшение площади входных изображений таким образом, что процент верно определяемых диспаритетов не уменьшился, а время выполнения алгоритма стереосопоставления сократилось прямо пропорционально количеству отобранных точек. Показано, что использование распараллеливания вычислений является эффективным только для этапов стереосопоставления и пост-фильтрации. Анализ применения параллельного алгоритма в задаче предварительного отбора точек показал его неэффективность. Рассмотрен встречный метод стереосопоставления, применяемый для уменьшения относительной ошибки. Показано, что он позволяет сократить процент ошибок только на очень малом количестве отобранных точек, а с их увеличением объем выходных данных перестает возрастать. На этапе пост-обработки карт диспаритетов предложен метод фильтрации, основанный на гипотезе связности диспаритетов, позволяющий избавиться от несистематических ошибочных выбросов в результирующих данных. Рассмотрен принципиальный подход к построению трехмерной модели по неполной карте диспаритетов.

СПИСОК ЛИТЕРАТУРЫ

1. *Вахитов А.Т.* Обзор алгоритмов стереозрения / А.Т. Вахитов, Л.С. Гуревич, Д.В. Павленко. Стохастическая оптимизация в информатике, Вып. 4 / под ред. О. Н. Граничина. – СПб. : Издательство С.-Петербургского университета 2008. – 299 с.

2. *Крыловецкий А.А.* Построение виртуальных моделей по дальнометрическим данным / А.А. Крыловецкий, Г.А. Карапиш, И.С. Черников // Телематика'2008. – (http://tm.ifmo.ru/tm2008/prog_2008.pdf).

3. *Papadimitriou D.* Epipolar line estimation and rectification for stereo image pairs / D. Papadimitriou, T. Dennis // IEEE Transactions on Image Processing, April 1996. 5(4): pp. 672–676.

4. *Scharstein D.* A taxonomy and evaluation of dense two-frame stereo correspondence algorithms / D. Scharstein, R. Szeliski // IJCV, 2002.

5. *Pearl J.* Reverend Bayes on inference engines: A distributed hierarchical approach // Proceedings of the Second National Conference on Artificial Intelligence, 1982. AAAI-82: Pittsburgh, PA. Menlo Park, California: AAAI Press. pp. 133–136.

6. *Cormen T.* Introduction to Algorithms / Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. // MIT Press and McGraw-Hill, Second Edition. pp. 563, 655, 1043.

7. Middlebury stereo evaluation online system. (http://vision.middlebury.edu/stereo/eval/).

8. *Yu W.* Real time stereo vision using exponential step cost aggregation on GP / W. Yu, T. Chen, J. C. Hoe // IEEE International Conference on Image Processing (ICIP), 2009.

9. *Хромова Е.Н.* Воссоздание поверхности по произвольному набору точек. Подзадача построения плоскости, наименее удаленной от совокупности точек / Е.Н. Хромова, Д.П. Пауков, Е.А. Башков // II Республіканська наукова конференція студентів , аспірантів та молодих вчених «Комп'ютерний моніторинг та інформаційні технології» Донецк ДонНТУ, 2006 г.

10. *Lorensen W.* Marching Cubes. A high resolution 3D surface construction algorithm / William E. Lorensen, Harvey E. Cline // Computer Graphics, Vol. 21, Nr. 4, July 1987.

11. *Esteve J.* Approximation of a Variable Density Cloud of Points by Shrinking a Discrete Membrane / Jordi Esteve, Pere Brunet, Alvar Vinacua // Universitat Politecnica de Catalunya, Barcelona, August 2005.

12. *Hoppe H.* Surface Reconstruction from Unorganized Points / Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, Werner Stuetzle // Proceedings of the 19th annual conference on Computer graphics and interactive techniques, 1992, pp. 71–78.

13. *Scharstein D.* Learning conditional random fields for stereo / D. Scharstein, C. Pal // IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), Minneapolis, MN, June 2007.

14. *Кравцов А.* Общая формулировка задачи внешней калибровки камеры / А. Кравцов, В. Вежневец. // Компьютерная графика и мультимедиа. – 2003. – (www.ict.edu.ru/ft/002404/num1krav.pdf).

15. Стерео-реконструкция / А. Конушин [и др.] // МГУ, ВМиК. – 2008. – (http://courses.graphicon.ru/main/vision2008).

16. *Falcou J.* A Parallel Implementation of a 3D Reconstruction Algorithm for Realtime Vision / J. Falcou, J. Serot, T. Chateau, F. Jurie // Parallel Computing, 2005.

17. *Murray D.* Using real-time stereo vision for mobile robot navigation / Don Murray, Jim Little // Autonomous Robots, Springer Netherlands, Volume 8, Number 2 / 2 (Апрель) 2000 г, pp. 161–171.

18. *Henkel R.* Classical Cooperative Schemas // R. Henkel. – (http://axon.physik.uni-bremen.de/research/stereo/Coop/index.html).

Крыловецкий Александр Абрамович – доцент каф. Цифровых технологий Воронежского государственного университета e-mail: krylovetsky@gmail.com

Протасов Станислав Игоревич – аспирант каф. Цифровых Технологий Воронежского государственного университета e-mail: krylovetsky@gmail.com

Krylovetsky Alexander Abramovich – Associate Professor of department Digital technologies, Voronezh State University e-mail: krylovetsky@gmail.com

Protasov Stanislav I. – Postgraduate department of Digital Technologies, Voronezh State University e-mail: krylovetsky@gmail.com