

УДК 004.925: 004.932: 004.942: 004.946

ВЫЧИСЛЕНИЕ КАРТЫ ГЛУБИНЫ СТЕРЕОИЗОБРАЖЕНИЯ НА ГРАФИЧЕСКОМ ПРОЦЕССОРЕ В РЕАЛЬНОМ ВРЕМЕНИ

Котюжанский Л.А.

ФГАОУ ВПО «Уральский федеральный университет имени первого Президента России
Б.Н. Ельцина», Екатеринбург, e-mail: nexus077@gmail.com

Предложен скоростной метод вычисления карты глубины стереоизображения, каждый пиксель которого содержит информацию о расстоянии до стереокамеры, для реализации на графическом процессоре (ГПУ). Данное решение базируется на использовании нескольких скользящих окон адаптивной структуры с пред- и постобработкой входных и выходных изображений. Высокая производительность достигается за счет архитектуры современных ГПУ, наиболее подходящих для решения задачи вычисления карты глубины локальными методами. Проведенное сравнение времени работы алгоритма на центральном процессоре (ЦПУ) и на ГПУ демонстрирует высокую масштабируемость предложенного решения. Высокая скорость позволит использовать данный метод в задачах реального времени, например, в робототехнике и создании естественных интерфейсов. Обсуждаются пути возможного усовершенствования данного метода.

Ключевые слова: пиксель, диспаратет, карта глубины, стереопара, текстура

CALCULATION OF THE STEREOIMAGE DEPTH CARD ON THE GRAPHIC PROCESSOR IN REAL TIME

Kotjuzhanskij L.A.

Ural Federal University named after the first President of Russia B.N.Yeltsin,
Ekaterinburg, e-mail: nexus077@gmail.com

The high-speed method of calculation of depth card of the stereoimage which each pixel contains the information on distance up to a stereocamera, for realization on graphic processor (GPU) is offered. The given decision is based on use of several sliding windows of adaptive structure with pre-and postprocessing of entrance and target images. High efficiency is achieved due to architecture modern GP, the most suitable for the decision of the problem of depth card calculation by local methods. The carried comparison of an operating time of algorithm on central processor (CP) and on GP shows high scalability of the suggested decision. High speed will allow to use the given method in problems of real time, for example, in a robotics and creation of natural interfaces. Ways of probable improvement of the given method are discussed.

Keywords: pixel, disparity, depth card, stereopair, texture

Одна из важных задач стереозрения – процесс преобразования двух плоских изображений в трехмерную сцену с восстановлением информации о *глубине* каждой точки плоского изображения (расстоянии от стереокамеры до соответствующей точки реальной сцены).

Зная информацию о глубине (функция двух переменных) можно генерировать 3D модели ландшафта и других природных объектов для использования в различных приложениях, таких как виртуальная реальность, симуляция полета, робототехника. В частности, знание об удаленности точек изображения от реального прообраза позволяет делать захват опорных точек движущегося объекта для получения трехмерных координат, что важно в индустрии спецэффектов, кино и телевидении. Захват координат опорных точек позволяет строить уникальные бесконтактные человеко-машинные интерфейсы, управление в которых осуществляется движениями пальцев рук или вообще жестов в пространстве на некотором удалении от стереокамеры.

На данный момент известны активные и пассивные методы восстановления информации о глубине реальной сцены. Ак-

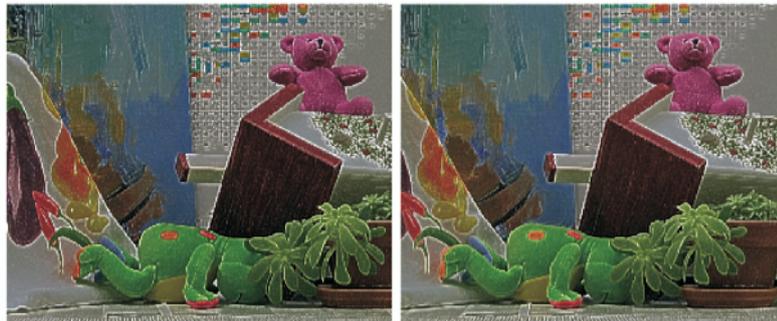
тивные методы используют ультразвуковые преобразователи или лазерное освещение рабочего пространства, дающие на выходе быструю и точную информацию о глубине [2, 6]. Однако у этих методов есть ограничения по отношению к диапазону измерений и стоимости аппаратных компонентов.

Пассивные методы, основанные на компьютерном зрении, обычно реализуются более простыми и недорогими сенсорами, определяющими расстояние. Такие методы способны генерировать информацию о глубине по полученной паре изображений и параметрам двух камер [1, 7]. Одна из фундаментальных проблем стереозрения заключается в установлении точного соответствия между левым и правым изображением стереопары. Под соответствием понимается расстояние (диспаратет) между пикселями одного и того же объекта на левом и правом изображении. Существующие локальные методы по вычислению карты диспаратета основаны на принципе «скользящего окна» [8]. Они характеризуются определенной сбалансированностью скорости-качества, хорошо масштабируются и реализуются на таких параллельных архитектурах, как видеокарты.

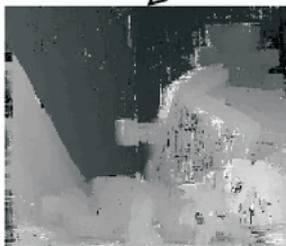
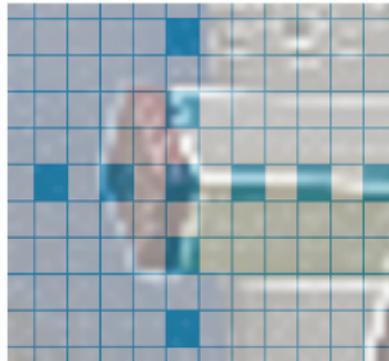
Входные данные - два ректифицированных изображения с левой и правой камеры



Предобработка входных изображений пиксельным шейдером
Вычисление свертки входного изображения с наложением на входное изображение



Пиксельный шейдер вычисления диспаратета.
Используется два адаптивных разряженных окна с весовыми коэффициентами и общей точкой пересечения, с шагом разрежения 1,2,4 или 6.
Вычисляется взвешенная сумма всех элементов каждого окна. Вес обратно пропорционален расстоянию.
Где сумма конкретного элемента - сумма квадратов разностей на левом и правом изображении.
Выбирается окно с минимальной суммой. Операция повторяется с заданным смещением d адаптивных окон по x .
Индекс смещения d с минимальной суммой признается как диспаратет для пикселя левого изображения.



Первичная карта глубины

Шейдер постобработки первичной карты глубины.
Используется медианный фильтр уменьшения шума



Рис. 1. Общий порядок действий для вычисления карты глубины

Цель данной работы заключается в разработке метода и программного модуля вычисления карты глубины приемлемого качества за кратчайшие временные интервалы в районе от 10 до 40 миллисекунд на недорогом оборудовании. Такие требования продиктованы тем, что данный модуль предназначен для взаимодействия с другими программными модулями, в задачи которых входят трекинг объектов, идентификация лиц и жестов. Все модули должны работать в реальном времени на одном ПК в едином программном комплексе машинного зрения для захвата движения человека и бесконтактного управления мультимедиа.

Общий порядок вычислений представлен на рис. 1. Он состоит из предобработки входных данных, вычисления диспаратета

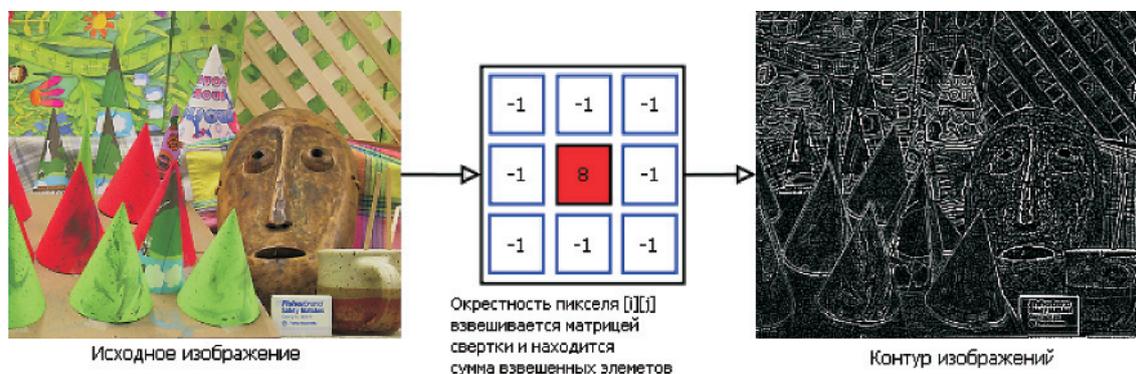


Рис. 2. Выделение контуров на изображении

Далее вычисляется линейная комбинация градиентного и исходного изображений с варьируемым параметром k (рис. 3).

Использование такой предобработки значительно уменьшает вероятность ошибки вычисления глубины пикселя, особенно при наличии на стереопаре однородных, малоcontrastных областей (рис. 4).

Диспаратет пикселя d по определению равен расстоянию между соответствующими пикселями левого и правого изображений (установление соответствия выполняется построчно перебором пикселей на выбранной строке изображений y)

$$d = |x_r - x_l|.$$

Здесь x_l – положение выбранного пикселя фиксированной строки y на левом изображении, x_r – положение соответствующего пикселя этой строки на правом изображении.

Чтобы найти x_r для каждого элемента (x, y) строки y правого изображения вычисляется сумма стоимостей пикселей C с окрестностями, как показано в [8].

$$C(x, y, d) = \sum_{(i, j) \in W(x, y)} P_c(i, d, j),$$

пикселя и конечной постобработки выходных данных медианным фильтром.

Первый шаг вычислений – получение градиентных изображений стереопары методом свертки (рис. 2) [5]

$$C_i(x, y) = \sum_{(i, j) \in N(x, y)} I_l(i, j) \cdot M(i, j).$$

Здесь C_i – обработанный пиксель левого изображения; $I_l(i, j)$ – исходный пиксель на левом изображении; $N(x, y)$ – окрестность 3×3 пикселей; $M(i, j)$ – элемент известной матрицы свертки

$$M = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}.$$

где $W(x, y)$ – одно- или двумерная окрестность (окно) пикселей с центром в точке (x, y) .

Под стоимостью понимается модуль разности левого (выбранного) и правого (подбираемого) пикселей:

$$P_c(x, d, y) = |p(x, y) - q(x - d, y)|.$$

Здесь p – пиксель на левом изображении; q – пиксель на правом изображении; d – перебираемый диспаратет.

Далее из всех полученных методом перебора значений суммы стоимостей выбирается минимальное значение, соответствующее положению пикселя с координатами $(x_l - d, y)$ на правом изображении. Такой принцип в различных вариациях используется во многих работах, в частности в работе с реализацией адаптивного окна [8]. Так, наиболее похожие пиксели и их окрестности на левом и правом изображении вдоль горизонтальной линии будут формировать минимальную стоимость.

В локальных методах, при увеличении размеров скользящего окна, с одной стороны, уменьшается вероятность ошибки определения диспаратета пикселя, а с другой стороны, значительно возраста-

ет объем вычислений, что делает данный метод малопригодным для расчетов на стереопарах высокого разрешения в ре-

альном времени. Кроме того, появляется эффект «раздувания» переднего плана (рис. 4).



$$P = (1 - k) \cdot I + C \cdot k$$

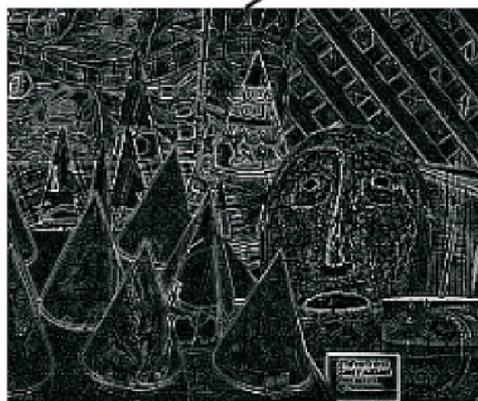


Рис. 3. $k = 0,5$; P – обработанный пиксель, I – пиксель исходного изображения; C – пиксель градиентного изображения

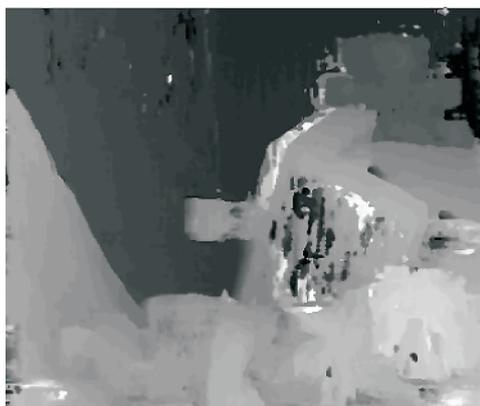


Рис. 4. Преимущества предобработки входных данных – вычисление карты глубины с использованием предобработки (слева)

Если использовать окна с адаптивной структурой, можно получить достаточно хорошее качество карты глубины, но при этом вновь требуются значительные объемы вычислений.

В предлагаемом методе для достижения оптимального баланса между скоростью и качеством используется два одномерных разреженных окна – вертикальное и горизон-

тальное. Каждое окно строится из центра рассматриваемого пикселя. Под разрежением понимается, что окно состоит не из ближайших друг к другу пикселей, а из пикселей, находящихся на некотором заданном расстоянии друг от друга, например, два, четыре, шесть пикселей. Это позволяет существенно увеличить скорость вычислений при незначительном увеличении ошибки (рис. 5).

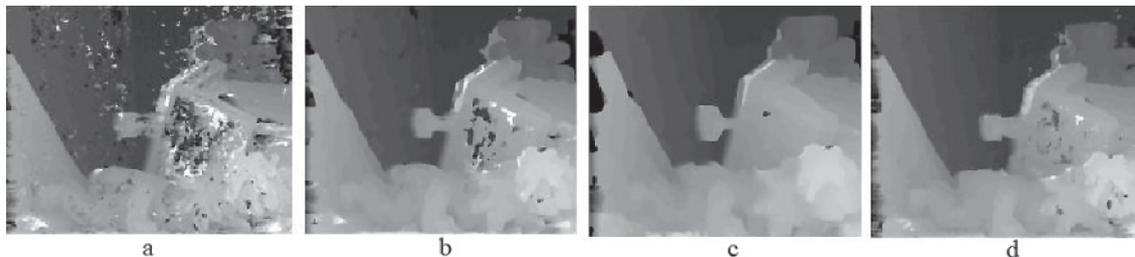


Рис. 5. Результаты вычислений для различных вариантов метода построения карты глубины: а – окно 3×3 – время вычисления 7,6 мс; б – окно 10×10 – 125 мс; с – окно 21×21 – 700 мс; д – два одномерных разреженных окна 20×1 , с шагом разрежения 4 пикселя – 11,7 мс

Для расчета стоимости двух одномерных окон используются следующие формулы:

$$C_h(x, y, d) = \sum_{i \in W_h(x, y)} P_c(i, d, y) \cdot \frac{1}{|x-i|};$$

$$C_v(x, y, d) = \sum_{j \in W_v(x, y)} P_c(x, d, j) \cdot \frac{1}{|y-j|};$$

$$C_d = \min \{C_h, C_v\}.$$

Здесь W_h и W_v – горизонтальные и вертикальные окрестности пикселя с центром в точке (x, y) ; C_h – стоимость горизонтального окна; C_v – стоимость вертикального окна; C_d – итоговая стоимость в точке (x, y) .

Подчеркнем, что операция суммирования в этих формулах выполняется по индексам суммирования, меняющимся с шагом, не равным единице (эффект прореженности окна (окрестности)).

В настоящий момент в области ГПУ вычислений наиболее активно развиваются технологии CUDA и OpenCL, обеспечивающие высокую гибкость и удобство в работе при взаимодействии между ЦПУ и ГПУ. Так, примеры успешной реализации параллельных алгоритмов на перечисленных выше технологиях показаны в [3]. Однако в данной работе был выбран язык шейдеров [4] GLSL, реализующий обработку

графических данных, таких как текстуры, полигоны, фрагментными и вершинными программами. Выбор в пользу GLSL обусловлен тем, что входными и выходными данными разработанного алгоритма являются изображения. Отдельно стоит отметить, что GLSL работает корректно почти на всех платформах и ГПУ, в то время как CUDA будет работать только на картах Nvidia, а текущая реализация OpenCL имеет различия в реализациях AMD и NVIDIA как синтаксически, так и с точки зрения скорости выполнения ряда функций. Отметим, что тестирование алгоритма проводилось на тестовых изображениях базы Middlebury (<http://vision.middlebury.edu/stereo/data/>).

В процессе разработки данного метода обозначились пути возможного улучшения качества и оптимизации скорости вычислений. Так, для практически полного удаления шумов и повышения четкости границ объектов на карте глубины можно использовать сегментацию входного изображения. Это позволит находить карту глубины не для каждого конкретного пикселя изображения, а для набора сегментов-плоскостей изображения. Другой подход – обработка предложенным алгоритмом только интересующих регионов изображения либо вычисление значения глубины только для движущихся объектов в кадре стереокамеры.

Список литературы

1. Fehn C., Cooke E., Schreer O., Kauff P. 3D analysis and imagebased rendering for immersive TV applications // *Signal Process. Image Commun.* – 2002. – Vol. 17, № 9. – P. 705–715.
2. Feiner S., MacIntyre B., Seligmann D. Knowledge-based augmented reality // *Commun. ACM.* – 1993. – Vol. 36, № 7. – P. 53–62.
3. Harris J. Fast Fluid Dynamics Simulation on the GPU / J. Harris // http://developer.nvidia.com/GPUGems/gpugems_ch38.html.
4. Harris J., Prins J., Nyland L. Fast N-Body Simulation with CUDA / J. Harris / Prins J / Nyland L // GPU Gems 3 Chapter 31.
5. http://processing.org/learning/topics_test/convolution.html.
6. Iddan G.J., Yahav G. 3D imaging in the studio and elsewhere // *Proc. SPIE.* – 1994. – Vol. 4298. – P. 48–55.
7. Kanade T., Yoshida A., Oda K., Kano H., Tanaka M. A stereo machine for video-rate dense depth mapping and its new applications // *Proc. CVPR'96.* – 1996. – P. 196–202.
8. Kanade T., Okutomi M. A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment // *Proceedings of the 1991 IEEE International Conference on Robotics and Automation (ICRA '91).* – 1991. – P. 1088-1095.

References

1. Fehn C., Cooke E., Schreer O., Kauff P. 3D analysis and imagebased rendering for immersive TV applications // *Signal Process. Image Commun.* 2002. Vol. 17, no. 9, pp. 705–715.

2. Feiner S., MacIntyre B., Seligmann D. Knowledge-based augmented reality // *Commun. ACM.* 1993. Vol. 36, no. 7. pp. 53–62.
3. Harris J. Fast Fluid Dynamics Simulation on the GPU / J. Harris // http://developer.nvidia.com/GPUGems/gpugems_ch38.html
4. Harris J., Prins J., Nyland L. Fast N-Body Simulation with CUDA / J. Harris / Prins J / Nyland L // GPU Gems 3 Chapter 31.
5. http://processing.org/learning/topics_test/convolution.html.
6. Iddan G.J., Yahav G.. 3D imaging in the studio and elsewhere // *Proc. SPIE.* 1994. Vol. 4298, pp. 48–55.
7. Kanade T., Yoshida A., Oda K., Kano H., Tanaka M. A stereo machine for video-rate dense depth mapping and its new applications // *Proc. CVPR'96.* 1996, pp. 196–202.
8. Kanade T., Okutomi M. A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment // *Proceedings of the 1991 IEEE International Conference on Robotics and Automation (ICRA '91).* 1991, pp. 1088–1095.

Рецензенты:

Гребенников В.И., д.ф.-м.н., профессор, главный научный сотрудник Института физики металлов УрО РАН, г. Екатеринбург;
 Готлиб Б.М., д.т.н., профессор, заведующий кафедрой «Мехатроника» ФГБОУ ВПО «Уральский государственный университет путей сообщения», г. Екатеринбург.
 Работа поступила в редакцию 24.02.2012.