

# АНАЛИЗ БЫСТРОДЕЙСТВИЯ И ВЫЧИСЛИТЕЛЬНОЙ СЛОЖНОСТИ АЛГОРИТМОВ 3D-РЕКОНСТРУКЦИИ С ТОЧКИ ЗРЕНИЯ ИХ ПРИМЕНИМОСТИ НА ПРОЦЕССОРАХ С НИЗКИМ ЭНЕРГОПОТРЕБЛЕНИЕМ

*А.В. Аргутин*

## THE ANALYSIS OF PERFORMANCE AND COMPUTING COMPLEXITY OF 3D-RECONSTRUCTION ALGORITHMS IN TERMS OF THEIR APPLICABILITY ON LOW POWER CONSUMPTION PROCESSORS

*A. V. Argutin*

Рассматриваются алгоритмы 3D-реконструкции и их применимость для решения задач компьютерного зрения на процессорах с низким энергопотреблением. Проведенный анализ быстродействия и вычислительной сложности алгоритмов позволяет определить их преимущества и сделать оптимальный выбор в зависимости от условий их реализации.

*Ключевые слова:* компьютерное зрение, 3D-реконструкция, алгоритмы.

3D-reconstruction algorithms and their applicability for computer vision problems solution on low power consumption processors are considered. The analysis of performance and computing complexity of 3D-reconstruction algorithms allows to determine their preferences and to make optimal choice depending on implementation conditions.

*Keywords:* computer vision, 3D-reconstruction, algorithms.

### Введение

В настоящее время задача 3D-реконструкции в реальном времени имеет множество решений [1]. Однако развитие робототехники как науки и появление современных портативных устройств, построенных на архитектуре ARM, привели к необходимости в решении задач компьютерного зрения на процессорах с низким энергопотреблением и, соответственно, меньшей производительностью. Задача 3D-реконструкции предполагает высокую алгоритмическую сложность вычислений, связанную с попиксельной переработкой кадров с двух видеопотоков, сравнение величин яркости в некоторой окрестности точки изображения.

Согласно [2], существует 3 наиболее распространенных алгоритма 3D-реконструкции с использованием стереопары: локальный, полуглобальный и глобальный – и около 15 способов расчета «цен совпадения» [3], используемых в этих методах. В данной статье описывается локальный метод, так как, согласно [2], полуглобальные методы [4] выполняются на 14 % дольше, а глобальные – (graph cuts) – на 50 %.

---

Аргутин Александр Вячеславович – аспирант кафедры электронных вычислительных машин, Южно-Уральский государственный университет; alex.argutin@gmail.com

### Алгоритмы 3D-реконструкции и анализ их эффективности

Локальный метод [5] представляет собой метод «скользящего окна», сканирующего изображения с левой и правой камеры и ставящего в соответствие окну  $\omega \in L$  с центром в точке  $(x_i, y_i)$  окну  $\omega' \in R$  с центром в точке  $(x_k, y_j)$ , где  $L, R$  – пространства левого и правого изображений. Для каждой пары окон вычисляется значение «цены совпадения», в дальнейшем минимизируемое для всех возможных значений disparity (далее – величины рассогласования). Сложность алгоритма «скользящего окна» –  $O(N*D*W)$ , где  $N$  – количество точек в исходном изображении;  $D$  – максимальная величина сдвига левого изображения относительно правого, или величина максимального рассогласования,  $W$  – количество точек, охватываемых скользящим окном.

Оптимизация локального алгоритма может быть выполнена за счет уменьшения одного из параметров –  $N, D$  или  $W$ , что приведет к следующим последствиям: уменьшение  $N$  снизит разрешение получаемой карты глубины и вызовет эф-

---

Argutin Alexander Vyacheslavovich – post-graduate student of Electronic Computers Department, South Ural State University; alex.argutin@gmail.com

факт либо понижающей качество дискретизации, либо урезания исходного фрагмента изображения. Уменьшение величины  $D$  уменьшит различаемое расстояние от камеры, так как величина рассогласования обратно пропорциональна квадрату расстояния от камеры до объекта. Уменьшение  $W$  приведет к снижению достоверности функции соответствия и увеличению количества ошибок по причине увеличения вероятности схожих окон. Наименьшая возможная величина  $W$  – это 1 пиксель, однако в таком случае некоторые алгоритмы теряют возможность построения карты глубины.

Фактором, влияющим на быстродействие алгоритма 3D-реконструкции не менее значительно, чем величины  $N$ ,  $D$  и  $W$ , является скорость работы алгоритма, рассчитывающего «цены совпадения» (или степени схожести) окон  $\omega$  и  $\omega'$ . Сложность некоторых алгоритмов, а также оценка количества операций процессора, требуемых для их реализации, приведены ниже.

AD [2] является алгоритмом с вырожденным окном (1 пиксель) и производит сравнение окон с помощью поиска наименьшей по модулю разности яркости пикселей в окнах  $\omega$  и  $\omega'$ . Таким образом, сложность алгоритма –  $O(1)$ , так как  $N$  всегда равно 1, а количество затрачиваемых операций напрямую зависит от способа реализации функции abs. Функция расчета «цен совпадения» [2], минимизация которой выполняется алгоритмом стереорекострукции, приведена ниже:

$C_{AD}(p, d) = |I_L(p) - I_R(p - d)|$ , где  $p$  – вектор координат пикселя  $[x \ y]$ ;  $d$  – вектор величин рассогласования по осям координат  $X$  и  $Y$ ;  $I_R$  и  $I_L$  – функции интенсивности цвета левого и правого изображений от вектора координат.

Для расчета «цены совпадения» при использовании высокопроизводительной функции расчета абсолютной величины [6] потребуется выполнить: 2 операции чтения из памяти, 2 операции вычитания, 1 операцию сложения, 1 операцию побитового исключающего ИЛИ (XOR), 15 операций сдвига, 2 операции присваивания, что в сумме дает 23 инструкции процессора. Несмотря на высокие показатели скорости работы, AD-алгоритм дает карту глубины очень низкого качества с большим количеством ошибок, из-за чего его использование в промышленности невозможно.

Алгоритмы SAD [5] и SSD [7] основываются на принципе работы алгоритма AD, но используют невырожденное окно и за счет этого получают более высокое (относительно AD) качество карты глубины. Размер используемого окна подбирается экспериментально, поскольку увеличение количества пикселей повышает количество шума в тестируемом окне, а уменьшение снижает количество информации, содержащейся в окне. Формулы расчета «цен совпадения» для алгоритмов SAD и SSD приведены ниже:

$C_{SAD}(p, d) = \sum |I_L(q) - I_R(q - d)|$ , где  $q \in Np$ ,  $Np$  – окрестность пикселя  $p$ ;

$C_{SSD}(p, d) = \sum (I_L(q) - I_R(q - d))^2$ , где  $q \in Np$ ,  $Np$  – окрестность пикселя  $p$ .

Для расчета «цены совпадения» потребуется выполнить:  $2 * N$  операции чтения из памяти,  $2 * N$  операции вычитания; в случае SAD:  $N * (15$  операций сдвига + 1 операцию сложения + 1 операцию побитового исключающего ИЛИ (XOR)); в случае SSD:  $N * (1$  операцию умножения, приравняваемую к 31 операции сдвига и 32 операциям сложения). В сумме, при расчете «цен совпадения» методом SAD с размером окна, равным  $9 * 7$  (для удобства сравнения с последующими методами), будет выполнено  $9 * 7 * (2 + 2 + 15 + 1 + 1) = 1323$  инструкции процессора, а при расчете методом SSD  $9 * 7 * (2 + 2 + 31 + 32) = 4221$  инструкция.

Метод Census [8] относится к группе непараметрических методов, так как основывается не на яркости пикселя непосредственно, а на локальном порядке следования яркостей пикселей. Зачастую реализация этого метода предполагает подготовку изображений, называемую трансформацией или фильтрованием. Метод выполняется в 2 этапа. На первом этапе каждое изображение преобразуется в census-матрицу. Каждый пиксель этой матрицы есть битовая маска,  $i$ -й бит которой единичный, если яркость  $i$ -го пикселя в окне  $\omega$  выше яркости пикселя в центре окна. Сложность алгоритма получения величины census линейна,  $O(N)$ , где  $N$  – количество пикселей в окне  $\omega$ , так как каждый пиксель в окне сравнивается по яркости с центральным пикселем окна лишь однажды. Формула вычисления «цены совпадения» приведена ниже.

$f(p, q) = \{1, I(q) > I(p),$  где  $I(p)$  – яркость пикселя в точке  $p$  изображения,

$0, I(q) > I(p)$

$\text{census}(p) = \|f(p, q)$ , где  $p, q \in \omega$ .

Одна из возможных реализаций на языке C расчета величины census приведена ниже:

```
uchar ctr = I[x][y];
uchar census = ((I[y-1][x-1]>ctr)<<7) |
((I[x][y-1]>ctr)<<6)
| ((I[x+1][y-1]>ctr)<<5) | ((I[x-1][y]>ctr)<<4) |
((I[x+1][y]>ctr)<<3) | ((I[x-1][y+1]>ctr)<<2) |
((I[x][y+1]>ctr)<<1) | ((I[x+1][y+1]>ctr));
```

Таким образом, для выполнения подготовки одной пары пикселей с левого и правого изображения необходимо выполнить:  $N$  операций чтения из памяти,  $N * (N - 1) / 2$  операций сдвига,  $N$  операций сравнения,  $N - 1$  операций побитового ИЛИ, 2 операции присваивания.

На размеры окна  $\omega$  накладываются архитектурные ограничения, связанные с моделью оперативной памяти. Количество бит в census должно быть максимально близко к кратному 8. Для окна размером  $9 * 7$  (census длиной 62 бита), на расчет величины census для двух пикселей (левого и правого изображений) потребуется:  $2 * (2 + 63 * 3 - 1 + 63 * (63 - 1) / 2) = 4286$  инструкций процессора.

Вторым этапом является подсчет расстояния Хемминга [9] между величинами *sensus* (в отличие от AD, где используется абсолютная величина разности). Из множества алгоритмов для архитектуры ARM при использовании окна  $9 \times 7$  более подходит реализация *popcount\_2*, не использующая 64-битного умножения [10]. Для вычисления расстояния Хемминга требуется выполнить: 63 операции сдвига, 5 операций побитового И, 5 операций сложения, 1 операцию вычитания, 6 операций присваивания. Таким образом, суммарное количество инструкций процессора, требуемое на расчет «цены совпадения» по алгоритму *Sensus*, равно  $4286 + (63 + 5 + 5 + 1 + 6) = 4366$ .

В данной статье не приводится анализ методов NCC, Z-(SAD, SAD, NCC), VT [11], Rank [12] и MI [2], как более сложных и, соответственно, более медленных, чем перечисленные выше.

### Заключение

Подводя итог, можно выделить метод SAD как наиболее быстрый из приведенных, однако, согласно [1], даже оптимизированные алгоритмы SAD и SSD не дают такой точности, как *Sensus*. Реализация алгоритма предподготовки для *Sensus* во многом определяет вычислительные затраты, поэтому может быть предпринята попытка оптимизации способа расчета величины *sensus* и/или способа накопления и сравнения битов, описывающих относительную яркость пикселей в окне  $\omega$ , однако на текущем этапе его использование затруднительно ввиду чрезмерного количества требуемых операций.

### Литература

1. *The Middlebury Stereo Vision Page, an evaluation of dense two-frame stereo algorithms.* – <http://vision.middlebury.edu/stereo/eval/>.
2. Hirschmüller, H. *Evaluation of Stereo Matching Costs on Images with Radiometric Differences* / H. Hirschmüller, D. Scharstein // *IEEE Transactions*

*on pattern analysis and machine intelligence.* – 2008. – V. 31, № 9. – P. 1582–1599.

3. Hirschmüller, H., *Evaluation of Cost Functions for Stereo Matching* / H. Hirschmüller, D. Scharstein // *IEEE Conference on Computer Vision and Pattern Recognition.* – 2007. – V. 1. – P. 1–8.

4. Hirschmüller, H., *Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information* / H. Hirschmüller // *IEEE Conference on Computer Vision and Pattern Recognition.* – 2005. – V. 2. – P. 807–814.

5. Cevahir, C. *Efficient Edge-Preserving Stereo Matching* / C. Cevahir, A.A. Alatan // *IEEE International Conference on Computer Vision Workshops ICCV 2011 Workshops, Barcelona, Spain, November 6–13, 2011.* – P. 696–699.

6. *Optimized abs function.* – [http://www.strchr.com/optimized\\_abs\\_function](http://www.strchr.com/optimized_abs_function)

7. Ambrosch, K. *Accurate hardware-based stereo vision* / K. Ambrosch, W. Kubinger // *Computer Vision and Image Understanding.* – 2010. – V. 114, № 11. – P. 1303–1316.

8. Baik, Y.K., *Fast Census Transform-based Stereo Algorithm using SSE2* / Y.K. Baik, J.H. Jo, K.M. Lee // *The 12th Korea Japan Joint Workshop on Frontiers of Computer Vision, Seoul National University Computer Vision Lab., Seoul, Feb.* – 2006. – P. 305–309.

9. Блейхут, П. *Теория и практика кодов, контролирующих ошибки* / П. Блейхут. – М.: Изд-во «Мир», 1986. – 576 с.

10. *The popcount algorithm.* – <http://wiki.cs.pdx.edu/forg/popcount.html>

11. Birchfield, S. *A pixel dissimilarity measure that is insensitive to image sampling* / S. Birchfield, C. Tomasi // *TPAMI.* – 1998. – V. 20, № 4. – P. 401–406.

12. Zabih, R. *Non-parametric Local Transforms for Computing Visual Correspondence* / R. Zabih, W. John // *Third European Conference on Computer Vision, Stockholm, Sweden, May 1994.* – 1994. – P. 151–158.

*Поступила в редакцию 22 июня 2012 г.*