

ОБРАБОТКА ИЗОБРАЖЕНИЙ И СИГНАЛОВ

УДК 004.932.4

ПОВЫШЕНИЕ ВЫЧИСЛИТЕЛЬНОЙ ЭФФЕКТИВНОСТИ ПРОЕКТИВНОГО ПРЕОБРАЗОВАНИЯ ИЗОБРАЖЕНИЙ НА SIMD-АРХИТЕКТУРАХ

© 2019 г. А. В. Трусов^{1,*}, Е. Е. Лимонова^{2,3}, А. Р. Миргасимов⁴

¹ Московский физико-технический институт (государственный университет),
141701 Московская область, г. Долгопрудный, Институтский переулок, д. 9, Россия

² Федеральный исследовательский центр “Информатика и управление” РАН
119333 Москва, ул. Вавилова, 44/2, Россия

³ ООО “Смарт Энджинс Сервис”, 117312 Москва, проспект 60-летия Октября, д. 9, Россия

⁴ Компания “Rock Flow Dynamics”, 117418 Москва, Профсоюзная ул. 25А, Россия

*E-mail: trusov.av@phystech.edu

Поступила в редакцию 14.09.2018 г.

В работе предложен метод повышения вычислительной эффективности проективного преобразования изображений за счет сокращения числа проверок граничных условий преобразования. Эти проверки возникают, поскольку вследствие проективного преобразования на результирующем изображении могут быть точки, соответствующие точкам, выходящим за пределы исходного изображения. Предложенный метод позволяет эффективно задействовать SIMD-расширения современных вычислителей, позволяющие единообразно обрабатывать несколько элементов данных одновременно. Эксперименты показали, что таким образом можно ускорить алгоритм от 1.35 до 2 раз в зависимости от числа точек, попадающих за пределы исходного изображения.

Ключевые слова: проективное преобразование изображений, вычислительная эффективность, SIMD-архитектура, обработка изображений

DOI: 10.1134/S023500921901013X

ВВЕДЕНИЕ

Проективная геометрия возникла в XVII веке благодаря творчеству художников и архитекторов, стремившихся верно изобразить окружающий мир на их картинах и чертежах. Позже трудами Дезарга, Понселе и многих других ученых она превратилась в самостоятельную область математики, которая в наше время нашла практическое применение в техническом зрении и компьютерной графике.

Задачи представления объемного объекта в виде дискретного изображения, сопоставления нескольких кадров, полученных в результате съемки одного и того же объекта с разных ракурсов, или исправления искажений цифровых фотографий крайне важны для медицины (рентгенографии и компьютерной томографии (Cabral et al., 1994)), зрительных систем в автопилотах (Xu et al., 2000), компьютерной графики (McMillan, 1997), дополненной реальности (Tusegyan et al., 2002) и других областей. В решении всех этих задач важную роль играет проективное преобразование изображений. Поскольку многие из этих задач представляют не только теоретический, но и

практический интерес, на время работы алгоритмов, а значит, и на их вычислительную сложность, могут накладываться значительные ограничения. В частности, многие системы распознавания, такие как в работах (Skogyukina et al., 2015, 2017) или алгоритмы в работах (Fedorenko, Usilin, 2017; Kuznetsova et al., 2015) должны находиться в режиме реального времени. Поэтому вычислительно эффективные алгоритмы проективного преобразования, работающие с заданной точностью, представляют немалый интерес. Например, в работе (Musin, 1991) описана реализация, действующая быстро за счет уменьшения числа операций умножения и деления, составляющих основную вычислительную сложность алгоритма.

В этой работе рассматривается определенное геометрическое преобразование, так называемое проективное — переводящее прямые в прямые. Преобразование, обратное к проективному, так же является проективным, что легко позволяет использовать обратный метод (Hartshorne, 1967).

В работе предложено решение, которое в ряде случаев ускоряет работу алгоритма за счет уменьшения числа пикселей, требующих специальной

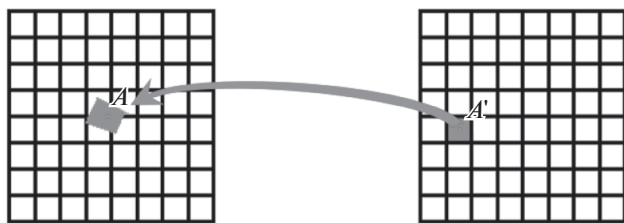


Рис. 1. Обратный метод вычисления проективного преобразования.

обработки. Для пикселей, прообразы которых целиком лежат внутри исходного изображения, не нужно проверять граничные условия. Для пикселей, прообразы которых целиком лежат вне исходного изображения, при определенных граничных условиях можно вообще никаких вычислений не производить. Кроме того, большая часть современных процессоров поддерживают SIMD (Single Instruction Multiple Data) принцип вычислений, который позволяет выполнять одну и ту же операцию над несколькими элементами данных одновременно, реализуя таким образом параллелизм на уровне данных. Предложенный метод может эффективно задействовать SIMD-инструкции, что позволяет дополнительно повысить вычислительную эффективность проективного преобразования.

АЛГОРИТМ ПРОЕКТИВНОГО ПРЕОБРАЗОВАНИЯ

Обратный метод. Задача проективного преобразования изображения является частным случаем задачи повторной дискретизации – преобразования дискретного изображения из одной системы координат в другую (Wolberg, 1994). Существуют два основных метода решения этой задачи: прямой и обратный. В первом случае для каждого пикселя исходного изображения выполняется поиск соответствующего ему положения на конечном изображении. В случае обратного преобразования каждый пиксель конечного изображения сопоставляется с пикселями исходного изображения с использованием обратного преобразования. В отличие от прямого, обратный метод гарантирует, что будет вычислено значение для каждого пикселя конечного изображения и поэтому чаще используется на практике.

Рассмотрим обратный метод решения задачи повторной дискретизации изображения. В этом методе для центра каждого пикселя конечного изображения A' с применением обратного преобразования находится соответствующая ему точка на исходном изображении A , и ее значение берется в качестве значения пикселя конечного изображения. Поскольку точка A может не совпадать

с центром какого-либо пикселя исходного изображения, значение в ней или восстанавливается по ближайшему пикселю, или интерполируется по близлежащим пикселям (линейным, квадратичным сплайновым или другим методом). Этот алгоритм и будет применяться в нашем решении (рис. 1).

Этот метод не совсем точен, поскольку используя его, мы приближаем значение пикселя конечного изображения значением в некоторой конкретной точке, в то время как в область пикселя конечного изображения могут отобразиться несколько пикселей исходного. Эту неточность можно снизить, вычисляя значение каждого пикселя конечного изображения по нескольким точкам внутри него (Wolberg, 1994).

Свойства проективного преобразования. Как было отмечено выше, проективное преобразование – это преобразование плоскости, переводящее прямые в прямые. В матричном виде оно задается как

$$\bar{u} = T\bar{x}, \tag{1}$$

где $T = (t_{ij})$, $i \in \{1, 2, 3\}$, $j \in \{1, 2, 3\}$ – матрица проективного преобразования, \bar{x} и \bar{u} – однородные координаты в плоскости исходного и конечного изображений соответственно. В координатном виде оно задается следующей формулой:

$$u = \frac{t_{11}x + t_{12}y + t_{13}}{t_{31}x + t_{32}y + t_{33}}; \quad v = \frac{t_{21}x + t_{22}y + t_{23}}{t_{31}x + t_{32}y + t_{33}}, \tag{2}$$

где (x, y) – координаты точки в плоскости исходного изображения, (u, v) – координаты ее образа в плоскости конечного изображения, при этом $\det(T) \neq 0$. В обратном методе используется обратное проективное преобразование. Обозначим его матрицу как $P = T^{-1}$, тогда выполняется равенство

$$x = \frac{p_{11}u + p_{12}v + p_{13}}{p_{31}u + p_{32}v + p_{33}}; \quad y = \frac{p_{21}u + p_{22}v + p_{23}}{p_{31}u + p_{32}v + p_{33}}. \tag{3}$$

В плоскости исходного изображения лежит прямая

$$t_{31}x + t_{32}y + t_{33} = 0, \tag{4}$$

образом которой является бесконечно удаленная прямая. Если она не пересекает выпуклый четырехугольник, то этот четырехугольник переходит в выпуклый четырехугольник в плоскости конечного изображения.

Получается, что если прямая (4) не пересекает исходное изображение, то его образом будет выпуклый четырехугольник в плоскости конечного изображения (рис. 2).

Граничные условия проективного преобразования. Как видно из рис. 2, некоторым пикселям на конечном изображении могут соответствовать



Рис. 2. Пример изображения проективного преобразования изображения.

На исходном изображении (слева) черным цветом обозначена прямая (4). Справа приведен результат проективного преобразования.

области, выходящие за границы исходного. Значения таких пикселей определяются граничными условиями задачи. Например, это может быть циклическое или симметричное повторение пикселей изображения за пределами его границ. Также может использоваться постоянное значение, т.е. считаться, что за пределами исходного изображения бесконечно много пикселей с одним постоянным значением (черный в примере на рис. 3) или значение ближайшего определенного пикселя изображения.

Эти условия достаточно сильно различаются между собой и могут потребовать дополнительных вычислений (например, в случае симметричного или циклического повторения значений), поэтому их проверка для каждого конкретного пикселя может быть достаточно вычислительно затратной.

Базовый алгоритм. В соответствии с предыдущими пунктами получаем базовый алгоритм проективного преобразования изображения.

Для каждого пикселя конечного изображения:

– находим координаты прообраза центра этого пикселя;

– находим значение в точке прообраза при помощи интерполяции и с учетом граничных условий.

Вычислительная оптимизация алгоритма проективного преобразования. Рассмотрим частный случай, при котором в плоскости исходного изображения прямая (4) не пересекает прямоугольник:

$$\Pi_o = \{x, y : x \in [-s, M + s], y \in [-s, N + s]\}, \quad (5)$$

где M – длина изображения, N – его высота, s – отступ, и координатные оси направлены вдоль сторон исходного изображения. Величина отступа определяется используемым методом интерполяции и в наиболее простых случаях (линейный метод или интерполяция по ближайшему пикселю) равна одному пикселю.

Рассмотрим еще один прямоугольник в плоскости исходного изображения

$$\Pi_i = \{x, y : x \in [s, M - s], y \in [s, N - s]\}. \quad (6)$$

Поскольку прямая (4) не пересекает (5), образами (5) и (6) являются выпуклые четырехугольники (5') и (6'),

$$\Pi_{o'} = T\Pi_o, \quad (5')$$

$$\Pi_{i'} = T\Pi_i \quad (6')$$

при этом (6') лежит внутри (5') в силу непрерывности преобразования (2) внутри прямоугольника (5).

Тогда все пиксели конечного изображения можно разделить на три класса:

– целиком лежит вне (5'). Если граничное условие постоянное значение, то для этих пикселей не нужно вычислять координаты прообразов центров.

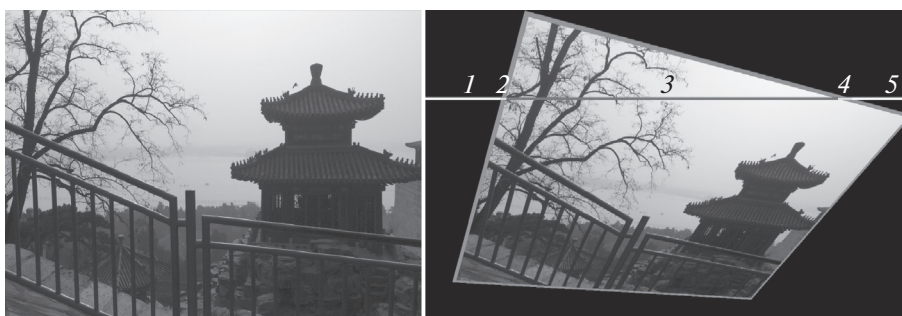


Рис. 3. Иллюстрация работы предложенного алгоритма.

Исходное (слева) и конечное (справа) изображения. На изображениях выделены области между прямоугольниками (5) и (6) и между их образами. На правом рисунке 1, 5 – отрезки внешних пикселей, 2, 4 – граничных, 3 – отрезок внутренних.

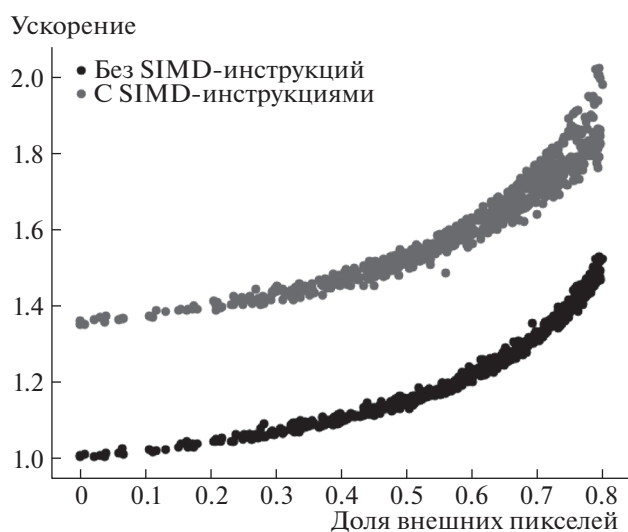


Рис. 4. Ускорение работы проективного преобразования по отношению к базовому алгоритму.

— лежит на границе (5'), или (6'), или между ними. Для такого пикселя нужно производить точные вычисления, согласно базовому алгоритму:

— целиком лежит внутри (6'). Для таких пикселей нет необходимости производить проверки, связанные с граничными условиями.

Таким образом, любая строка пикселей конечного изображения состоит не более чем из пяти отрезков пикселей одного из выше указанных типов (рис. 3).

ПРЕДЛОЖЕННЫЙ АЛГОРИТМ

Входными данными алгоритма являются матрица обратного преобразования (3), исходное изображение, а также набор параметров, таких как используемый метод интерполяции, граничные условия, информация о размерах конечного изображения. Выходными — конечное изображение.

Сам алгоритм состоит из следующих шагов:

1. Проверим, что преобразование (3) не вырождено и прямая (4) не пересекает прямоугольник (5). Если одно из этих условий нарушено, используем базовый алгоритм.

2. С применением прямого преобразования (2) находим координаты вершин четырехугольников (5') и (6').

3. Для каждой строки:

- (а) вычисляем границы отрезков пикселей одного типа по точкам пересечения строки с (5') и (6');

- (б) на отрезках внешних пикселей: в зависимости от граничных условий или задаем указанное значение пикселям, или выполняем вычисления,

согласно базовому алгоритму с учетом граничных условий;

- (с) на отрезках граничных пикселей вычисляем значения пикселей, согласно базовому алгоритму с учетом граничных условий;

- (д) на отрезке внутренних пикселей вычисляем значения пикселей, согласно базовому алгоритму без учета граничных условий.

Для получения большего ускорения вычисления на отрезке внутренних пикселей могут быть использованы SIMD-инструкции. Их использование упрощается, поскольку знаменатель преобразования (3) заведомо не обращается в 0, а значит, отсутствует необходимость в этой проверке. Таким образом, операции для соседних пикселей становятся однотипными и легко выполняются при помощи SIMD.

РЕЗУЛЬТАТЫ

Для получения экспериментальных данных было использовано изображение (рис. 3) размером 3264 на 2448 пикселей и набором из 700 матриц проективного преобразования, таких, что существенную часть конечного изображения составляют пиксели исходного. Для каждой матрицы время работы предложенного алгоритма определялось как медианное среднее по 21 эксперименту. В качестве базового метода использовался обратный метод вычисления проективного преобразования из пункта “Базовый алгоритм” без SIMD-инструкций. Было рассмотрено два варианта вычислительной оптимизации: алгоритм из пункта “Предложенный алгоритм” без и с использованием SIMD-инструкций внутри (6'). Ускорение посчитано как отношение времени работы оптимизированного алгоритма ко времени работы базового алгоритма.

В качестве граничного условия использовалось постоянное значение, поэтому время вычислений зависело от доли внешних пикселей. Результаты эксперимента представлены на рис. 4. Измерения проводились на процессоре Intel Core i7-7700HQ.

С использованием предложенного алгоритма (раздел “Предложенный алгоритм”) было получено ускорение в 1.4 раза при 80% внешних пикселей и практически никакого ускорения при их отсутствии. Ускорение на 5% было достигнуто при 20% внешних пикселей, что уже является значимым результатом для промышленных распознающих систем. Использование SIMD-инструкций еще сильнее ускорило работу алгоритма: в 1.35 раза при отсутствии внешних пикселей и в 1.7–2 раза при 80%.

ЗАКЛЮЧЕНИЕ

В этой работе предложен метод повышения вычислительной эффективности алгоритма проективного преобразования изображения, основанный на предварительном вычислении границ образа исходного изображения, который можно применить в том случае, если этот образ конечен. Было экспериментально показано, что в сочета-

нии с использованием SIMD-инструкций и постоянным граничным условием, он позволяет снизить время вычислений в 1.4–2 раза, в зависимости от числа внешних пикселей на конечном изображении.

Работа выполнена при частичной финансовой поддержке грантов РФФИ 17-29-03297 и 16-29-09508.

Increase of computational efficiency of projective image transformation on SIMD-architectures

A. V. Trusov^{a,#}, E. E. Limonova^{b,c}, and A. R. Mirgasimov^d

^a *Moscow Institute of Physics and Technology (State University),
141701 Moscow Region, Dolgoprudny, Institutsky Pereulok 9, Russia*

^b *Federal Research Center "Informatics and Management" of the Russian Academy of Sciences
119333 Moscow, Vavilova st. 44/2, Russia*

^c *LLC "Smart Engines Service", 117312, Moscow, Prospect 60-letiya Octyabrya 9, Russia*

^d *"Rock Flow Dynamics" company, 117418 Moscow, Profsoyuznaya st. 25A, Russia*

[#] *E-mail: trusov.av@phystech.edu*

The paper proposes a method for computational efficiency increase of projective image transformation, which is achieved by reducing the number of checks for the boundary options of the transformation. These checks occur because, as a result of the projective transformation, there may be points on the resulting image that correspond to points outside the original image. The proposed method makes it possible to effectively use SIMD-extensions of modern calculators, which allow to process several data elements at once. Experiments have shown that in this way, the algorithm can be accelerated from 1.35 to 2 times, depending on the number of points that fall outside the limits of the original image.

Key words: Projective image transformation, computational efficiency, SIMD architecture, image processing

REFERENCES

- Cabral B., Cam N., Foran J. Accelerated volume rendering and tomographic reconstruction using texture mapping hardware. *Proceedings of the symposium on Volume visualization*. 1994. P. 91–98.
- Fedorenko F., Usilin S. Real-time object-to-features vectorisation via Siamese neural networks. *Proc. of Ninth International Conference on Machine Vision*. 2017. V. 10341. P. 1–5.
- Hartshorne R. *Foundations of projective geometry*. 1967. 91 p.
- Kuznetsova E., Shvets E., Nikolaev D. Viola-Jones based hybrid framework for real-time object detection in multispectral images. *ICMV*. 2015. V. 9875. P. 1–6.
- McMillan L. *An image-based approach to three-dimensional computer graphics*. Ph.D. thesis, Citeseer. 1997.
- Musin O.R. Fast geometric transformations for image processing. *International Journal of Imaging Systems and technology*. 1991. V. 3 (3). P. 257–261.
- Skoryukina N., Chernov T., Bulatov K., Nikolaev D.P., Arlazarov V. Snapscreen: Tv-stream frame search with projectively distorted and noisy query. *Ninth International Conference on Machine Vision*. 2017. V. 10341. P. 103410Y.
- Skoryukina N., Nikolaev D.P., Sheshkus A., Polevoy D. Real time rectangular document detection on mobile devices. *Seventh International Conference on Machine Vision*. 2015. V. 9445. P. 94452A.
- Tuceryan M., Genc, Y., Navab N. Single-point active alignment method (spaam) for optical see-through hmd calibration for augmented reality. *Presence: Teleoperators & Virtual Environments*. 2002. V. 11 (3). P. 259–276.
- Wolberg G. *Digital Image Warping*. IEEE Computer Society Press, Los Alamitos, CA, USA, 1st edn. 1994.
- Xu J., Chen G., Xie M. Vision-guided automatic parking for smart car. *Proceedings of the IEEE Intelligent Vehicles Symposium*. 2000. P. 725–730.