

Быстрый алгоритм построения промежуточных кадров для многоракурсных стерео дисплеев

Гришин С.В., Ватолин Д.С.

МГУ им. Ломоносова, факультет Вычислительной Математики и Кибернетики,
Лаборатория Компьютерной Графики
{sgrishin, dmitriy}@graphics.cs.msu.ru

1. АННОТАЦИЯ

Вычисление видеопотока для многоракурсных дисплеев (далее стерео-дисплеев) накладывает жесткие ограничения на вычислительную сложность алгоритмов, что делает многие из быстрых методов стерео-зрения неприменимыми для решения этой задачи. В данной статье описывается быстрый алгоритм вычисления видеопотока для стерео-дисплеев, пригодный для использования в системах реального времени. Высокая вычислительная эффективность разработанного алгоритма достигается за счет использования метода оценки движения, традиционно применяемого в задачах сжатия и обработки видео. Метод оценки движения, основанный на поиске соответствия между блоками кадров, позволяет с высокой скоростью вычислять смещения для точек кадра. Достоинством разработанного алгоритма является возможность обработки наложений возле краев кадра, что не свойственно быстрым алгоритмам стерео-зрения.

Keywords: stereo-vision, 3D display, motion estimation, occlusions processing.

2. ВВЕДЕНИЕ

Задача восстановления трехмерной структуры сцены по двум изображениям (далее задача стерео-зрения) находит свое применение во многих областях, например телеконференциях, навигации и управлении роботами, виртуальной реальности и т.д. В последнее время интерес к данной области возрос благодаря появлению устройств, способных воспроизводить объемные изображения без использования очков, примером таких устройств могут служить многоракурсные дисплеи или 3D-дисплеи. Для работы таких устройств требуется несколько ракурсов воспроизводимой сцены, однако, как правило, исходно имеется лишь два. Для получения недостающих ракурсов могут быть использованы алгоритмы стерео-зрения. Согласно общепринятой классификации [1] алгоритмы решения задачи стерео делятся на две группы: глобальные и локальные. В глобальных методах для получения результата используются ограничения на данные всего кадра или каждой его строки. Локальные методы, напротив, используют ограничения, действующие в локальной окрестности обрабатываемого пикселя. Методы первой группы [2, 3] обеспечивают результат высокой точности, однако работают, как правило, очень медленно и не применимы в системах реального времени. Алгоритмы второй группы [4] работают значительно быстрее глобальных методов, но дают менее качественный результат. При построении видеопотока, воспроизводимого на стерео-дисплеях, для каждой

пары исходных кадров требуется рассчитать смещения для нескольких (обычно 7-15) промежуточных кадров, что повышает требования к вычислительной сложности алгоритмов. Поскольку вычислительная сложность методов стерео-зрения увеличивается линейно с ростом числа вычисляемых промежуточных кадров, многие локальные методы становятся также неприменимыми в приложениях реального времени.

В данной статье описывается алгоритм получения видеоданных, предназначенный для использования в многоракурсном дисплее. Особенностью данного алгоритма является его высокая скорость, которая делает возможным его использование в системах реального времени. Данный алгоритм позволяет получать кадры с хорошим визуальным качеством (см. рис. 3-5), для чего используются методы, описанные в предыдущих работах, посвященных преобразованию частоты кадров [5, 6].

3. ОПИСАНИЕ АЛГОРИТМА

Видеоданные, воспроизводимые на стерео-дисплее, состоят из нескольких видеопотоков, каждый из которых должен соответствовать определенному углу зрения. В числе этих видеопотоков есть два исходных видеопотока, которые были сняты двумя камерами с разных точек. При этом считается, что камеры были синхронизированы и откалиброваны (подробнее о калибровке см. [1]). Корректное выполнение требований синхронизации гарантирует, что камеры производят захват кадров в одни и те же моменты времени. Геометрия положения камер и сцены – стандартная для задач стерео-зрения (каждая точка сцены проецируется в строку изображения с одним и тем же номером для левой и правой камер) [1]. Вычисление видеопотока для стерео-дисплея производится пошагово: на каждом шаге вычисляются промежуточные кадры, соответствующие определенной паре кадров из исходных видеопотоков. При этом для вычисления этих промежуточных кадров используются два кадра, снятых с камер в один момент времени.

Пусть искомый видеопоток состоит из N видеопоследовательностей (рис. 1). Из них две – исходные, значит общее число промежуточных видеопотоков равно $N-2$. Таким образом, в искомом видео потоке каждому моменту времени t соответствует N кадров одинакового размера. Обозначим видеопоследовательности, снятые на левую и правую камеры, как $I_L(x,t)$ и $I_R(x,t)$ соответственно, где $I_K(x,t)$ – яркость точки x на кадре с номером t , снятом камерой $K \in \{L, R\}$. Тогда искомый видеопоток можно представить в следующем виде:

$$V(n, x, t) = \begin{cases} I_L(x, t), & n = 1 \\ I(x, t) = f(x, t, I_L(x, t), I_R(x, t)), & 1 \leq n \leq N - 1, \\ I_R(x, t), & n = N \end{cases}$$

где $n \in \{1..N\}$ – номер видеопоследовательности в искомом видеопотоке;

t – момент времени;

$f(x, t, I_L(x, t), I_R(x, t))$ – функция, зависящая от кадров, снятых левой и правой камерами.

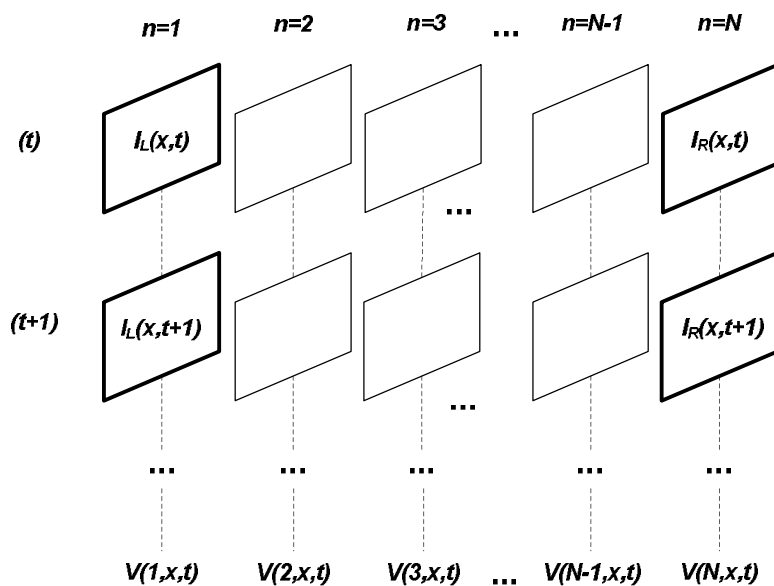


Рис. 1. Схема искомого видеопотока

Можно выделить следующие основные шаги алгоритма вычисления видеопотока для стерео-дисплея:

1. получение на вход очередной пары кадров $I_L(x, t)$ и $I_R(x, t)$;
2. вычисление поля смещений между кадрами $I_L(x, t)$ и $I_R(x, t)$;
3. вычисление промежуточных кадров $V(n, x, t)$ для $1 \leq n \leq N - 1$;
4. переход на шаг 1.

3.1. Вычисление поля смещений

Поле смещений, или векторов движения, является основной информацией для вычисления промежуточных кадров $V(n, x, t)$, $1 \leq n \leq N - 1$. Оно используется для определения позиций каждой точки исходных кадров $I_L(x, t)$ и $I_R(x, t)$ на промежуточных кадрах (см. например [2]). Для вычисления векторного поля в разработанном алгоритме используется метод оценки движения (ME, Motion Estimation), применяемый в обработке/сжатии видео. Использованный метод оценки движения (ОД) был ранее разработан для применения в задаче преобразования частоты кадров [5, 6]. Данный метод ОД основан на поиске соответствия между блоками (block-matching algorithm) и позволяет с высокой вычислительной эффективностью получать поле векторов движения для блоков кадра. Для его адаптации под задачу стерео-зрения потребовалось несколько модификаций.

Первая состоит в изменении способа подачи кадров на вход алгоритма ОД. Обычно, алгоритм ОД считывает кадры последовательно, в порядке, в котором они расположены во входном видеопотоке. Причем, при вычислении поля векторов движения между текущим (t) и предыдущим ($t-1$) кадрами используется поле векторов, вычисленное для кадров ($t-1$) и ($t-2$). Для того чтобы такая схема кэширования поля векторов работала и в разработанном алгоритме, кадры на вход алгоритма ОД подаются парами: сначала пара кадров $I_L(x,t)$ и $I_R(x,t)$, затем – $I_L(x,t+1)$ и $I_R(x,t+1)$ и т.д.

Вторая модификация заключается в том, что поиск векторов движения в вертикальном направлении не производится. Это позволяет сэкономить время, уменьшив область поиска, не потеряв при этом в точности результирующего векторного поля. Потери точности не происходит из-за структуры исходных видеопотоков, обусловленной геометрией сцены – все смещения между кадрами возможны только в горизонтальном направлении (см. [1]).

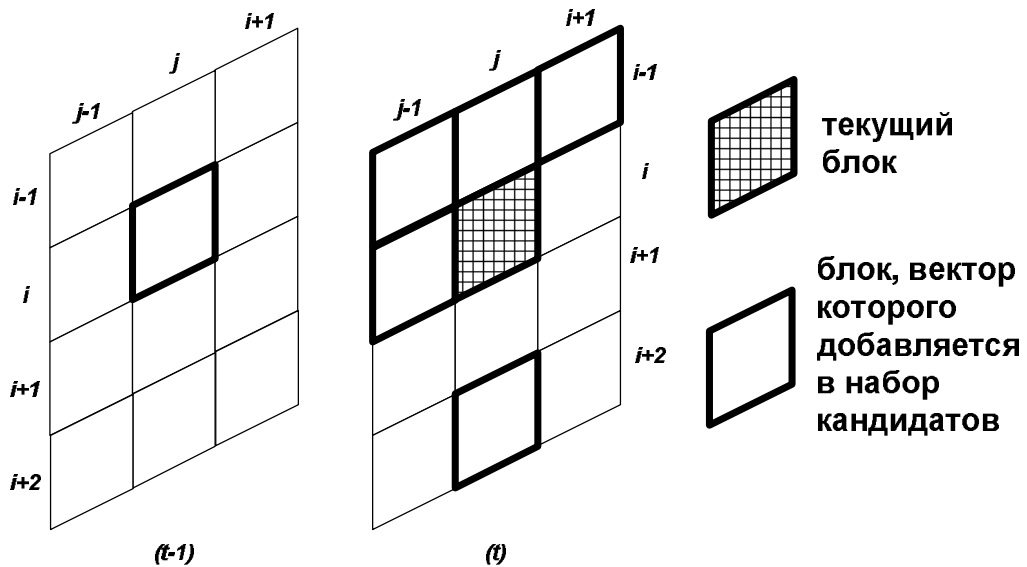


Рис. 2. Пример набора кандидатов

В традиционных методах стерео-зрения много времени уходит на вычисление смещений между кадрами. Отчасти это обусловлено тем, что часто используется метод полного перебора, в котором для поиска вектора движения в каждой точке кадра проверяются все возможные смещения. Использование метода ОД позволяет существенно сэкономить время, поскольку в нем используется несколько приемов оптимизации поиска. Среди этих приемов:

- методы шаблонного поиска, в которых проверяются не все, а лишь ограниченное число точек в окрестности данной;
- использование векторов-кандидатов (рис. 2). Поскольку параметры движения коррелированы в пространстве и во времени, часто вектор движения в очередной точке кадра совпадает или близок к вектору движения в соседней точке. Поэтому часто достаточно проверить один из векторов соседних точек и

перейти к поиску смещения для следующей точки. Проще это объяснить на примере. Пусть $v(n, x, t)$ – вектор смещения точки x на кадре t из видеопотока n . Тогда с большой вероятностью среди векторов $v(n, x, t-1)$, $v(n, x - (1, 0)^T t)$, $v(n, x + (1, 0)^T t)$, $v(n, x - (0, 1)^T t)$, $v(n, x + (0, 1)^T t)$ и т.д. есть совпадающий или очень близкий к $v(n, x, t)$ вектор. В используемом алгоритме ОД применяется специальный метод выбора векторов-кандидатов, позволяющий минимизировать число проверяемых кандидатов и быстро найти подходящий вектор смещения.

3.2. Вычисление промежуточных кадров

Вычисление промежуточных кадров производится с использованием метода, описанного в [6] для получения интерполированных кадров, с небольшими модификациями. Во-первых, как и в случае с методом ОД, изменен способ подачи кадров на вход метода: кадры считываются не последовательно, а парами $I_L(x, t)$ и $I_R(x, t)$. Во-вторых, существенно изменен метод обработки наложений. В, отличие от алгоритма, описанного в [6], наложения обрабатываются только возле краев кадра. Это обусловлено тем, что в данной задаче для интерполяции промежуточных кадров могут использоваться только два изображения – $I_L(x, t)$ и $I_R(x, t)$, в то время как исходный метод производит интерполяцию с использованием четырех кадров. Таким образом, в измененном методе интерполяции поиск и обработка наложений производится только возле краев кадра по тому же алгоритму, что и в исходном методе. Важно заметить, что традиционные методы стереозрения, использующие обработку наложений, являются очень медленными и неприменимы в задачах реального времени. Поэтому, несмотря на то, что обработка наложений производится не на всем кадре, ее наличие является важным достоинством разработанного метода.

4. РЕЗУЛЬТАТЫ

Скорость неоптимизированной реализации алгоритма на компьютере с процессором Intel Pentium IV 2.4GHz при обработке видео с размером 352x288 составляет 7.4 кадров/с. Предполагается, что скорость оптимизированной версии при тех же условиях будет, как минимум, в 2-3 раза выше текущей. Помимо относительно высокой скорости (алгоритмы стереозрения редко работают быстрее 1 кадр/с), алгоритм демонстрирует хорошее визуальное качество промежуточных кадров (см. рис. 4). Как уже было замечено, дополнительным достоинством алгоритма является обработка наложений возле краев кадра.



Рис.3. Кадр, снятый первой камерой



Рис. 4. Вычисленные промежуточные кадры (зеленые линии наложены на кадры для удобства сравнения)

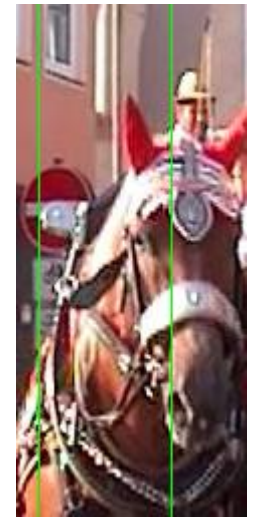


Рис. 5. Кадр, снятый второй камерой

5. ЛИТЕРАТУРА

1. Z.Brown, D.Burschka, D.Hager, "Advances in Computational Stereo", *Proc. IEEE Transactions on Pattern Analysis and Machine Intelligence* (25), No. 8, August 2003, pp. 993-1008
2. А.А. Лукьяница, "Эффективный алгоритм восстановления промежуточных ракурсов по стереопаре", *Труды конф. Graphicon-2006*, стр. 375-377, Июль 2006
3. С. Lawrence Zitnick, Takeo Kanade, "A Cooperative Algorithm for Stereo Matching and Occlusion Detection", *Proc. IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7), pp. 675-684, July 2000
4. L. De Stefano, M.Marchionni, S.Mattocchia, "A Fast Area-based Stereo Matching Algorithm", *Proc. Image and Vision Computing* 22(12), pp. 983-1005, 2004
5. Д. Ватолин, С. Гришин, "Алгоритм преобразования частоты кадров на основе интерполяции скомпенсированных кадров", *мат. девятого научно-практического семинара "Новые информационные технологии в автоматизированных системах"*, с. 32-46, март 2006
6. Гришин С.В., Ватолин Д.С., "Алгоритм N-кратного увеличения частоты кадров видео на основе пиксельной компенсации движения с обработкой наложений", *мат. конф. Graphicon*, стр. 112-119, Июль 2006

6. Об авторах

Ватолин Дмитрий Сергеевич — к.ф.-м.н., старший научный сотрудник лаборатории компьютерной графики ВМиК МГУ. Основатель сервера «Все о сжатии» <http://compression.ru/>. E-mail: dmitriy@graphics.cs.msu.su.

Гришин Сергей Викторович — аспирант, сотрудник лаборатории компьютерной графики ВМиК МГУ. E-mail: sgrishin@graphics.cs.msu.su.