

Эффективный алгоритм восстановления промежуточных ракурсов по стереопаре

А.А. Лукьяница,

Московский государственный университет им. М.В. Ломоносова, Москва, Россия
luk@ic.msu.ru

Аннотация

При трехмерной съемке реальных объектов, как правило, доступны только два изображения, образующие стереопару. Для некоторых приложений нужно иметь большее число ракурсов. В настоящей работе предлагается эффективный алгоритм реконструкции произвольного числа промежуточных ракурсов трехмерной сцены по известной стереопаре. Результаты численных экспериментов продемонстрировали высокое качество предложенного алгоритма, а также его высокую скорость, которая позволяет использовать разработанный метод в реальном времени.

Ключевые слова: Стереозображение, динамическое программирование, промежуточный ракурс.

1 Введение

В настоящее время появились устройства, способные воспроизводить объемные изображения: многоракурсные дисплеи и проекционные системы. Для работы с этими устройствами требуется заданное число ракурсов воспроизводимой сцены – несколько проекций, получаемых при заданных положениях камеры. В натурной съемке получить много ракурсов не представляется возможным; как правило, дело имеют только с двумя, которые составляют стереопару. Поэтому возникает проблема реконструкции необходимого числа промежуточных ракурсов по известной стереопаре – т.е. изображений, которые были бы получены аналогичной камерой, если бы она занимала промежуточные положения между камерами, используемыми при стереосъемке. Большинство из известных алгоритмов позволяют решить эту задачу путем выполнения реконструкции трехмерной поверхности с последующим построением ее проекций (см., например, [1]). Эти методы могут быть применены только в режиме off-line из-за большой вычислительной сложности. В настоящей работе предлагается эффективный алгоритм, пригодный к использованию в реальном времени. Предлагаемый алгоритм решает поставленную задачу в два этапа. На первом этапе производится нахождение сопряженных точек, то есть точек на левом и правом изображении, являющихся проекциями одной и той же области трехмерного пространства. Существует много алгоритмов, позволяющих найти сопряженные точки. Например, в работе [2] используется статистический подход, основанный на методе максимизации ожидания. В работе [3] для этой цели применяется набор гауссовских функций, покрывающих изображение. В работе [4] поиск сопряженных точек осуществляется с помощью генетического алгоритма. В настоящей работе предлагается быстрый алгоритм, проводящий сопоставление точек на двух снимках с помощью метода, основанного на динамическом программировании. На втором этапе по найденным точкам

осуществляется аппроксимация промежуточных ракурсов и заполнение областей, для которых не нашлось соответствия. Такие области возникают по той причине, что некоторые фрагменты трехмерной сцены видимы одной камерой и невидимы другой, поэтому для таких фрагментов не существует сопряженных точек. Предложенный алгоритм позволяет эффективно заполнить эти области путем проведения специальной аппроксимации.

2 Постановка задачи

Пусть нам дана стереопара – т.е. два изображения одной и той же сцены, зафиксированные стереокамерой. Обозначим изображение, полученное левой камерой, буквой L , а изображение правой камеры – буквой R . Введем ортогональную систему координат, согласованную с камерами: ось x направим вправо через центры изображений, ось y – вниз вдоль вертикальной оси изображений, а ось z дополняет их до правой тройки. Начало координат выберем таким образом, чтобы изображения находились в плоскости Oxy , а центры изображений были симметричны относительно плоскости Oyz (см. Рис 1).

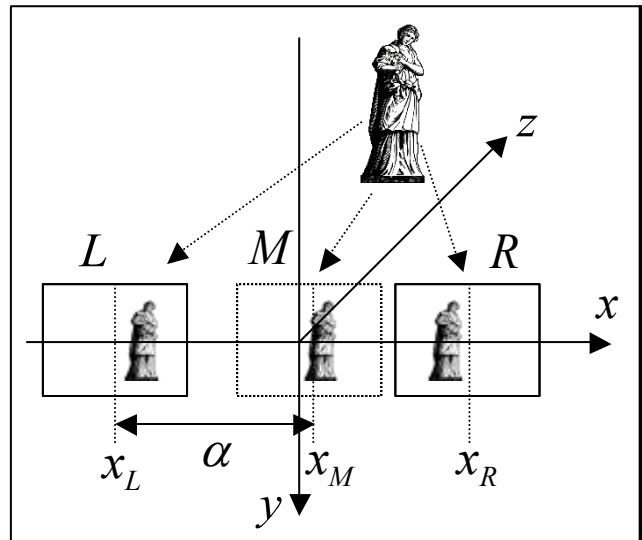


Рис 1. Геометрия стереосъемки. Сцена зарегистрирована двумя камерами - L и R . Следует построить промежуточный ракурс M .

Пусть центр левого изображения в выбранной системе координат имеет координаты $(x_L, 0, 0)$, а центр правого - $(x_R, 0, 0)$. Задача состоит в том, чтобы построить изображение в плоскости Oxy с центром в некоторой точке

$(x_M, 0, 0)$, наиболее близкое к изображению, которое было получено камерой, чья оптическая ось проходила бы через точку $(x_M, 0, 0)$. Предлагаемый метод состоит из двух этапов: нахождение сопряженных точек на левом и правом изображениях; аппроксимация промежуточного изображения по сопряженным точкам и заполнение областей, для которых соответствие не является однозначным.

3 Алгоритм нахождения сопряженных точек

Пусть каждое изображение содержит $K \times N$ пикселей соответственно по горизонтали и вертикали. Задача нахождения сопряженных точек состоит в следующем: для каждого пикселя левого изображения L_{ki} , $k=1, \dots, K$, $i=1, \dots, N$, следует найти пиксель R_{kj} , $j=1, \dots, N$ на правом изображении, соответствующий той же точке регистрируемой сцены. Предположим, что камеры были установлены на одной горизонтали и имеют одинаковый коэффициент усиления. Тогда горизонтальные строки фотоприемников совпадают с эппольярными линиями, поэтому сопряженные точки должны находиться на строках, имеющих одинаковый номер. Это позволяет записать

$$L(k, i) \approx R(k, j + \Delta_j),$$

где Δ_j - смещение, зависящее от номера пикселя.

Нахождение сопряженных точек состоит в вычислении смещений для всех пикселей. Далее для простоты мы будем опускать первый индекс, поскольку он имеет одно и то же значение для сравниваемых изображений и подразумевать, что описываемый ниже процесс выполняется для каждой пары соответствующих строк на левом и правом изображениях. Для нахождения подходящих кандидатов в сопряженные пиксели следует сравнивать некие области Ω в окрестности этих пикселей, что существенно повышает устойчивость алгоритма. Действительно, в изображениях одного и того же участка сцены под различными ракурсами могут появиться отличия, связанные с взаимным расположением источников света и элементов сцены. Кроме того, некоторые области могут быть видны из одной камеры и невидны для другой. Пусть сравниваемая область Ω имеет размеры $k \times n$, тогда в окрестности j -го пикселя на правом изображении следует просканировать область большего размера, $m \times l$, $m \geq k$, $l > n$. В процессе сканирования вычисляется функция расстояния $d(i, j)$, характеризующая близость изображений, находящихся в окрестности Ω_i пикселя $L(i)$ и в окрестности Ω_j пикселя $R(j)$. В качестве функции расстояния обычно выбирают сумму разностей интенсивностей соответствующих пикселей из Ω_i и Ω_j :

$$d(i, j) = \sum_{m=-k/2}^{k/2} \sum_{l=-n/2}^{n/2} |L(k+m, i+l) - R(k+m, j+l)|$$

или сравнивают градиенты изображений из Ω_i и Ω_j [5].

На практике лучше использовать сочетание этих методов, поскольку они имеют различную эффективность для

различных типов изображений. Наиболее часто различными авторами используется метод сравнения интенсивностей, но при обработке изображений с однородными пространственными объектами, как, например, стена дома, этот метод приводит к появлению плато у функции $d(i, j)$, что затрудняет выбор сопряженных точек. Разность градиентов в этом случае позволяет учесть «более тонкие» детали изображений.

Для выбора сопряженных точек из всех возможных претендентов нужно удовлетворить двум условиям. Во-первых, очевидно, что номера индексов сопряженных точек должны минимизировать суммарные отличия сравниваемых областей на левом и правом изображении. Второе условие связано с тем, что величина смещения не может уменьшаться по мере возрастания индекса j , т.е. $\Delta_j \geq \Delta_{j-1}$. Это условие следует из геометрических свойств стереопары. Для удовлетворения этим условиям предлагается следующий алгоритм, основанный на методе динамического программирования.

Введем две вспомогательных функции: вещественную ψ и целочисленную P . Первая из них нужна для накопления суммарных отличий сравниваемых областей, а вторая предназначена для хранения соответствующих индексов. Алгоритм динамического программирования состоит из трех этапов.

1. Инициализация.

$$\psi(i, 1) = d(i, 1), \quad i = 1, \dots, N,$$

$$P(i, 1) = i, \quad i = 1, \dots, N.$$

2. Индуктивный переход

$$\psi(i, j) = \min_{1 \leq k \leq i} \psi(k, j-1) + d(i, j), \quad i = 2, \dots, N, j = 1, \dots, N,$$

$$P(i, j) = \arg \min_{1 \leq k \leq i} \psi(k, j-1), \quad i = 2, \dots, N, j = 1, \dots, N.$$

3. Обратный ход.

$$J(N) = \arg \min_{1 \leq i \leq N} \psi(i, N),$$

$$J(j) = P(J(j+1), j+1), \quad j = N-1, \dots, 1.$$

В результате будет построен массив индексов для сопряженных точек J , по которому для любого пикселя левого изображения $L(i)$ сопряженный пиксель правого изображения определяется как $R(J(i))$.

4 Нахождение промежуточных ракурсов

Введем параметр $\alpha = \frac{x_M - x_L}{x_R - x_L}$, который характеризует

относительную степень близости восстанавливаемого ракурса к левому изображению. Далее, для всех значений $i = 1, \dots, N$, $j = 1, \dots, N$ вычислим:

$$k = \lceil \alpha j + (1 - \alpha) J(j) \rceil$$

$$M(i, k) = \alpha L(i, j) + (1 - \alpha) R(i, J(j))$$

Как видно из этих соотношений, некоторые точки промежуточного ракурса могут оказаться незаполненными. Это происходит из-за того, что соответствующие фрагменты объекта трехмерной сцены видны одной камерой и

невидимы другой. Пусть в реконструируемом изображении в какой-либо i -й строке оказались незаполненными подряд идущие пиксели с номерами $j_0, \dots, j_0 + m$. Здесь возможны два случая:

1. Если $J(j_0 - 1) = J(j_0 + m + 1)$, то эти пиксели видимы только левой камерой, поэтому они заполняются соответствующими значениями из левого изображения:

$$M(i, j) = L(i, j), \\ j = j_0, \dots, j_0 + m.$$

2. Если $J(j_0 - 1) < J(j_0 + m + 1)$, то эти пиксели видимы только правой камерой, поэтому они заполняются соответствующими значениями из правого изображения:

$$M(i, j) = R(i, J(j)), \\ j = j_0, \dots, j_0 + m.$$

Предложенный алгоритм был реализован в виде программы, по которой было проведено большое число численных расчетов с различными стереопарами. На рисунке (Рис 2) приведены результаты восстановления двух промежуточных ракурсов, соответствующих значениям параметра $\alpha = 1/3$ и $2/3$:

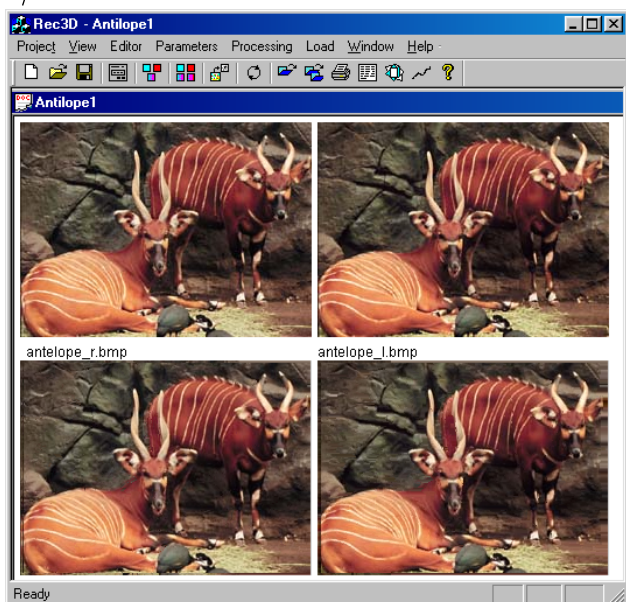


Рис 2. Результаты реконструкции двух промежуточных ракурсов. Верхние изображения являются стереопарой, нижние соответствуют значениям параметра $\alpha = 1/3$ (левое) и $2/3$ (правое).

5 Заключение

В настоящей работе предложен эффективный алгоритм, позволяющий проводить реконструкцию промежуточных ракурсов для известной стереопары. Результаты численных экспериментов продемонстрировали высокое качество предложенного алгоритма, а также его высокую скорость,

которая позволяет использовать разработанный метод в реальном времени.

6 Библиография

- [1] F.Pedersini, P.Pigazzini, A.Sarti, S.Turbaro. 3D area matching with arbitrary multiview geometry. *Signal Processing: Image communication*, Vol.14, 1998, pp.71-94.
- [2] A.R. Mansouri, J. Konrad. Bayesian Winner-Take-All Reconstruction of Intermediate Views from Stereoscopic Images, *IEEE Transaction on Image Processing*, Vol. 9, № 10, October 2000, 13 p.
- [3] G-Q Wei, W.Brauer, G.Hirzinger. Intensity- and Gradient-Based Stereo Matching Using Hierarchical Gaussian Basis Functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.20, № 11, November 1998, pp. 1143-1160.
- [4] J.Chai, S.D.Ma. Robust epipolar geometry estimation using genetic algorithm. *Pattern Recognition Letters*, Vol.19, 1998, pp.829-838.
- [5] Z.N. Li, G. Hu. Analysis of Disparity Gradient Based Cooperative Stereo. *IEEE Transaction on Image Processing*, Vol. 5, № 11, November 1996.

Об авторе

Андрей Александрович Лукьяница – кандидат физ.-мат. наук, ст. научный сотрудник факультета вычислительной математики и кибернетики Московского государственного университета им. М.В. Ломоносова.

Адрес: Москва, 119899, Воробьевы горы, МГУ, 2-й учебный корпус, факультет ВМиК, кафедра автоматизации научных исследований.

E-mail: luk@ic.msu.su

Effective Algorithm for Reconstruction of Intermediate Views from Stereopair

Abstract

A multiple object views for 3D systems are often needed while in practice just stereopair is available. A new effective algorithm for reconstruction of intermediate views from stereopair is proposed. Numerical simulations demonstrate fine quality of presented algorithm, and its high speed make it suitable for real time applications.

Keywords: *Stereopair, dynamic programming, intermediate views.*

About the author

Andrey Lukyanitsa is a senior researcher at Moscow State University, Department of Computational Mathematics and Cybernetics. His contact email is luk@ic.msu.su.