

Визуализация городской обстановки пленоптическим методом

Ю. С. Борисов

Институт автоматки и процессов управления ДВО РАН,

Владивосток, Россия

borisoff@dvo.ru

Аннотация

Описывается IBR алгоритм, позволяющий визуализировать большие пространственные сцены. В качестве входной информации используются изображения, полученные обычной цифровой камерой. Эти изображения переводятся в панорамные виды, которые являются четырехмерной параметризацией полной пленоптической функции, для их хранения. С использованием простой аппроксимирующей геометрии из этих структур строятся новые изображения с виртуальной камеры.

В статье формулируются условия, при которых сцены можно визуализировать предложенным способом. Приводятся результаты как для смоделированных, так и для реальных сцен.

Ключевые слова: Рендеринг по изображениям, Пленоптическая функция, Панорамные виды.

1. ВВЕДЕНИЕ

Традиционной задачей компьютерной графики является построение виртуальной окружающей среды с высокой реалистичностью. Стандартный полигональный подход в этом случае требует наличия подробной и качественной трехмерной модели. В случае построения городской среды, подобные модели приобретают огромные по объему данных размеры, требующие больших усилий на стадии построения и высоких компьютерных мощностей на этапе визуализации. Недостатком также является зависимость скорости рендеринга сцены от ее сложности. В последнее десятилетие реальной альтернативой становятся методы, основанные на изображениях (IBR методы). Одной из первых систем, которая использовала последовательность фотографий, была QuickTimeVR [1]. Она позволяла выполнить круговой обзор сцены с определенной точки пространства. В [2] был предложен подход, который позволял построить гладкую интерполяцию между изображениями, используя движение пикселей (оптическое течение) во время перемещения камеры с одного положения в другое. Другим направлением в IBR методах является использование пленоптической функции $f(x, y, z, t, \varphi, \theta, \lambda)$ [3], которая описывает полный световой поток в каком-либо регионе. В [4] было рассмотрено представление пленоптической функции в виде набора цилиндрических проекций, с дальнейшим вычислением несоответствия между этими проекциями для того, чтобы построить новый вид с позиции, не совпадающий с позициями существующих видов. В отличие от цилиндрической проекции, в [5],[6] была предложена четырехмерная параметризация пленоптической функции, при которой каждый луч сцены определялся через координаты его пересечения с некоторыми заданными плоскостями. Ясно, что в этом случае вся сцена должна быть ограничена некоторым объемом, следовательно, область

применения этих методов ограничена визуализацией одиночных объектов. Дальнейшим развитием этой идеи является Unstructured Lumigraph [7], в котором авторы соединили подход Lumigraph'a с VDTM (View-dependent texture mapping [8]), подходом, требующим относительно точной геометрической модели и небольшого количества изображений. В [9] был рассмотрен алгоритм для осуществления прохода в виртуальной среде. В качестве входных использовались изображения, полученные с помощью всенаправленной камеры, установленной на моторизованное шасси. Изображения снимались через небольшие расстояния, вследствие этого основными задачами этого подхода были управление устройством получения изображений, компрессия данных и их эффективное использование.

Целью настоящей работы является построение системы, позволяющей выполнять визуализацию больших пространственных сцен городской среды, основываясь на информации, полученной с помощью обычной фотокамеры. Подобная система предоставляет удобный и качественный способ для выполнения виртуальных проходов по городу. Исходя из размеров визуализируемого пространства и, как следствие, количества усилий для получения изображений, мы выдвигаем следующие требования на алгоритм: простота получения исходной информации, ограничение на размер памяти, необходимой для ее хранения и высокая скорость визуализации.

2. ОПИСАНИЕ МЕТОДА

Поскольку в своей работе мы визуализируем ограниченный класс сцен, то можно сформулировать дополнительные условия, упрощающие решение нашей задачи:

1. Движение камеры ограничено некоторой траекторией (улица), следовательно, изображения достаточно снимать только в некоторых точках этой траектории.
2. Движение камеры происходит на одной высоте от поверхности.
3. Поскольку объекты (дома), в основном, находятся на значительном удалении от камеры, не требуется точная визуализация мелких деталей.
4. Рассматриваются только статические сцены.

Учитывая поставленные условия, пленоптическая функция $f(x, y, z, t, \varphi, \theta)$ редуцируется к четырехмерной функции $f(x, y, \varphi, \theta)$. Данное представление является естественным представлением при визуализации виртуальных пространств. Кроме того, оно удобно для задания движения виртуальной камеры. Основной идеей предлагаемого метода является получение значений пленоптической функции в некоторых точках (x, y) заданной траектории для $0^\circ \leq \varphi < 360^\circ$,

$-\psi/2 \leq \theta \leq \psi/2$, где ψ - угол вертикального раствора камеры. Назовем их *панорамными видами*. Эти виды сохраняются и в дальнейшем используются для построения изображений с виртуальной камеры. Как и в [7] при генерации новых изображений мы используем простейшую прокси-геометрию, достаточную при указанных выше ограничениях.

Работу нашей системы можно условно разделить на три этапа: получение изображений, приведение к определенному виду для хранения и получение новых видов в зависимости от положения виртуальной камеры. Причем поскольку первые два этапа выполняются на стадии предварительной обработки, то они не являются критичными по времени выполнения, в то время как основными требованиями на построение новых видов являются получение изображений без видимого искажения на высоких скоростях.

2.1 Получение входных изображений

Для получения изображений используется стандартная цифровая фотокамера. Она является калиброванной, то есть внутренние параметры считаются известными.

Для выполнения съемки камера закрепляется на штативе на определенной высоте. При этом мы уменьшаем количество ее степеней свободы относительно поворота с трех до одной – поворота относительно вертикальной оси, проходящей через оптический центр камеры. Кроме этого, на камеру крепится шкала для измерения текущего угла поворота. Внешний вид используемой установки показан на рис.1.

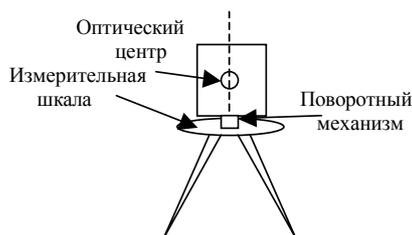


Рис 1: Устройство для получения изображений.

При помощи указанной установки получаем изображения окружающей обстановки через определенные углы α_i . Значения этих углов определяются по показаниям измерительной шкалы. Необходимо, чтобы полученные изображения покрывали все 360° , и при этом давали некоторое перекрытие друг друга. Большее перекрытие дает более качественные результаты, но требует дополнительной работы по съемке изображений. Количество требуемых изображений зависит от горизонтального раствора камеры и требуемого перекрытия отдельных изображений. Следующим шагом алгоритма является уточнение положений камеры. Поскольку получение изображений ведется в ручном режиме, возможно возникновение неточностей при определении угла поворота камеры. Для корректировки угла используется следующий алгоритм: идем с заданным шагом $\Delta\alpha$ от $\alpha' - \beta$ до $\alpha' + \beta$, где α' - предварительно вычисленные положения, β - подгоночная величина (выбирается в пределах $3^\circ - 5^\circ$). В качестве уточненного угла выбираем такой угол α , который

максимизирует корреляцию перекрывающихся регионов. Использование описанного подхода позволяет обойтись без поиска особенностей на изображениях для сопоставления их положения относительно друг друга.

2.2 Построение панорамных видов

Вторым этапом работы метода является приведение полученных изображений к панорамным видам для дальнейшего хранения и последующего построения изображений с нового положения камеры. Для этого нам необходимо для каждого луча панорамного вида C определить, какой точке на изображении I он соответствует (рис. 2). Имея эти точки, мы можем затем произвести смешивание их интенсивностей для определения цвета этого луча.

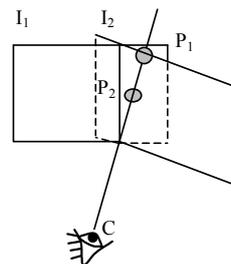


Рис 2: Определение точки на изображении по заданному лучу. I_1, I_2 - изображения, P_1, P_2 - соответствующие точки на изображениях.

При этом получаем точное соответствие геометрии с отдельных изображений, а последующее смешивание необходимо из-за возможного отличия освещения на разных снимках. Каждая полученная панорама сохраняется как JPEG-изображение. Ее положение и ориентация заносится в общую базу изображений.

Суммируя вышесказанное, алгоритм перевода полученных изображений в панорамный вид можно представить следующим образом:

1. Внутренние параметры камера, полученные с процесса калибровки: $imWidth, imHeight$ – размеры изображения; $FOVY$ – угол раствора камеры по вертикали. Параметр $FOVX$ – горизонтальный раствор – вычисляется по формуле
$$FOVX = 2 \arctan\left(\frac{imWidth}{2f}\right), \text{ где } f = \frac{imHeight}{2 \tan\left(\frac{FOVY}{2}\right)}$$
2. Проходим по углу φ (от 0 до 360) с требуемым шагом (в экспериментах выбиралось 0.1 и 0.05 градуса).
3. Для каждой камеры, входящей в построение панорамы выполняем проверку, попадает ли текущий угол φ в раствор камеры: $-\frac{FOVX}{2} + An_i < \varphi < \frac{FOVX}{2} + An_i$, где An_i - поворот i -ой камеры.
4. Если точка попадает, то вычисляем ее горизонтальное положение на изображении по формуле $x = f \tan(\alpha)$. α - относительный угол на

камере: $\alpha = \varphi - An_i$. Запоминаем номер камеры и абсциссу точки p на изображении.

- Идем по вертикальному углу θ от $-\frac{FOVY}{2}$ до $\frac{FOVY}{2}$ с тем же шагом, что и этапе 2. Для камеры, сохраненной на этапе 4, вычисляем ординату точки $q = \sqrt{f^2 + x^2} \tan(\theta)$. Если точка попадает на изображение $-\frac{imHeight}{2} \leq q \leq \frac{imHeight}{2}$, то запоминаем ее положение (p, q) и цвет $c = PixelColor\left(p + \frac{imWidth}{2}, q + \frac{imHeight}{2}\right)$.
- Считаем для каждой точки ее удаленность от центра по горизонтали $d = |p|$.
- Вычисляем коэффициенты смешивания $A_i = \frac{|F(d_{max} - d_i)|}{D}$, где $D = \sum_i |F(d_{max} - d_i)|$. В этих формулах F - полиномиальная функция смешивания.
- Заносим в точку панорамы (φ, θ) цвет $c = \sum_i A_i c_i$.
- Полученную панораму помещаем на общий план. Для этого указываем положение панорамы (x, y) и точку, соответствующую нулевому углу (для правильной ориентации одной панорамы относительно другой).

2.3 Построение новых видов

Основным и заключительным этапом работы метода является получение нового вида по существующим данным и заданному положению виртуальной камеры (x, y, φ) .

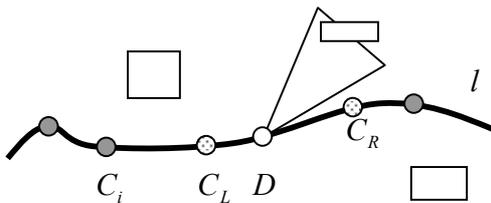


Рис. 3: Выбор исходных камер для построения вида. l - траектория, на которой снимались панорамные виды; C_i - набор панорамных видов, D - положение виртуальной камеры, C_L, C_R - выбранные виды

Поскольку съемка изображений происходит вдоль определенных линий (улиц), то для генерации каждого нового вида требуется не более чем два панорамных вида, между которыми в данный момент находится камера (рис. 3).

Выбрав два необходимых вида, производим смешивание с использованием дополнительной прокси-геометрии. В наших экспериментах выбирались три плоскости, две из которых расположены вертикально, параллельно траектории

движения, а третья - параллельно земле на расстоянии, соответствующем высоте штатива. Алгоритм построения нового вида выглядит следующим образом (рис. 4):

- Для каждой точки нового вида D строим луч α .
- Находим пересечение этого луча с аппроксимирующей плоскостью π , полученная точка - точка M^1 .
- Для двух выбранных видов C_L и C_R находим лучи α_L и α_R , проходящие через центр вида и точку M^1 .
- Цвета, соответствующие этим лучам на панорамах, смешиваем с коэффициентами, равными относительной удаленности виртуальной камеры от панорамного вида.

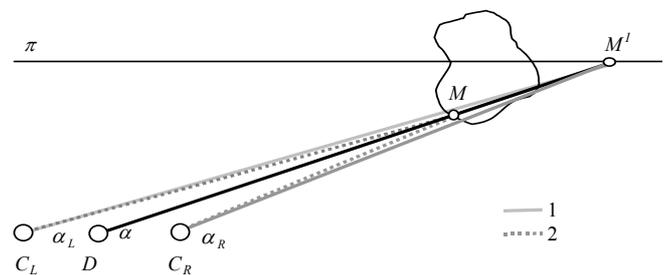


Рис 4: Нахождение необходимых лучей в панорамных видах.

1 - рассчитанный луч, 2 - точный луч.

Несмотря на то, что построение нового вида поточечное, оно дает приемлемую скорость выполнения.

Как видно на рис. 4, вследствие несовпадения аппроксимирующей плоскости с объектами среды, рассчитанный луч расходится с лучом, соответствующим точной геометрии, что приводит к появлению эффекта размытия на построенном изображении. Для уменьшения указанного эффекта, при невозможности определения точного расстояния до объектов или при неравномерном их расположении, необходимо снимать панорамные виды через меньшие расстояния. Поскольку для городской обстановки среднее расстояние до объектов составляет несколько десятков метров, то расстояние между снимаемыми панорамами может выбираться в пределах нескольких метров.

3. РЕЗУЛЬТАТЫ

Предложенный подход был реализован программно на языке C++ в среде WINDOWS. Были проведены пробные эксперименты как для виртуальных сцен, смоделированных в пакете трехмерной графики, так и для реальных объектов.

Для виртуальной сцены бралась камера с горизонтальным раствором в 50° , для каждой панорамы снималось 10 снимков размером 800×600 . Панорамы строились с шагом в 0.1 градус. Функция смешивания $F(x) = x^2$. Расстояние между отдельными панорамами выбиралось в $1/8$ от расстояния до строений. При построении новых видов мы брали всего две

плоскости, не учитывая нижней горизонтальной плоскости. На рис. 7а видно, что в этом случае возникает сильное несоответствие по поверхности пола.

При проведении экспериментов на реальных сценах, для получения изображений использовалась цифровая камера Samsung Digimax 210. Каждая панорама снималась в виде двенадцати изображений размером 1600x1200 пикселей. Некоторые из снятых изображений показаны на рис. 5.

Панорама собиралась с шагом в 0.05 градуса. В качестве функции смешивания выбиралась функция $F(x) = x^3$. Как видно на рис.6, такой выбор функции F с перекрытием отдельных изображений в 30% дает гладкую склейку даже при значительном отличии в яркости соседних изображений.

В сцене «Кабинет» строились две панорамы через расстояние 30 см. Аппроксимирующие плоскости выбирались на расстоянии в 2.5 м от камеры. Новые виды строились при перемещении виртуальной камеры вдоль линии, соединяющей положения снятых панорам. Объекты, отстоящие от плоскости стен, показывают размытие на новом виде до величины в три пиксела, что является практически незаметным при постоянном движении камеры.

Сцена «Коридор» снималась как 10 панорам на прямой линии в центре коридора через расстояние 40 см. Поскольку в этом случае мы могли точно измерить положение до стен, причем все объекты лежат практически точно на этих плоскостях, то новые виды на этой траектории видны без искажений.

Все тесты по построению панорам и новых видов выполнялись на компьютере Athlon-1200. Время на построение одной панорамы – 16 сек. Скорость получения нового вида – 10 fps. Размер каждой панорамы – около 500 кб.

Проведенные эксперименты показывают, что предложенный метод дает хорошие результаты в случае сцены, которая хорошо аппроксимируется несколькими плоскостями (комната, коридор, прямая улица). Однако в случае сцен с сильно разбросанными в ней объектами на новых видах появляются значительные искажения, приводящие к сильному размытию отдельных участков, и, как следствие, требующие снятия панорам через гораздо меньшие расстояния, что, в свою очередь, увеличивает усилия по получению исходной информации.

С учетом отмеченных недостатков и ограничений дальнейшие усилия по совершенствованию системы будут направлены на следующее:

1. Автоматическое сопоставление ориентации отдельных панорам на этапе 2. В текущей версии системы положение каждой задается вручную.
2. Уточнение прокси-геометрии, как по полученным изображениям, так и с использованием дополнительной информации (например, локальных планов застройки), и дальнейший переход от аппроксимирующих плоскостей к более сложным поверхностям.
3. Пока мы рассматривали только статические сцены. В условиях реальных городских улиц возможно появление локальных «мусорных» изображений

движущихся объектов, которые необходимо устранять.

4. Увеличение быстродействия за счет перехода от поточечного построения новых видов к полигональному.

4. БИБЛИОГРАФИЯ

- [1] Chen S. *QuickTime. VR - An Image-Based Approach to Virtual Environment Navigation. SIGGRAPH'95.* 29-38, 1995.
- [2] Chen S., Williams L. *View interpolation for image synthesis. ACM SIGGRAPH 1993,* 279–288, 1993.
- [3] Adelson E.H., Bergen J. *The Plenoptic Function and the Elements of Early Vision. Computational Models of Visual Processing, MIT Press, Cambridge, MA,* 3-20, 1991.
- [4] McMillan L., Bishop G., *Plenoptic Modeling: An Image-Based Rendering System. SIGGRAPH 95,* 39-46, 1995.
- [5] Gortler S., Grzeszczuk R., Szeliski R., Cohen M. *The lumigraph. SIGGRAPH 96,* 43–54.
- [6] Levoy M., Hanrahan P., *Light Field Rendering, SIGGRAPH'96,* 31-42, 1996.
- [7] Buehler C., Bosse M., McMillan L., Gortler S., Cohen M. *Unstructured Lumigraph Rendering. ACM SIGGRAPH 2001,* 425–432, 2001.
- [8] Debevec P., Taylor C., Malik J. *Modeling and rendering architecture from photographs. SIGGRAPH 96,* pages 11–20.
- [9] Aliaga D., Carlbom I. *Plenoptic Stitching: A scalable method for reconstructing 3D interactive walkthroughs. ACM SIGGRAPH 2001,* 443–450, 2001.

Об авторах

Борисов Юрий Сергеевич, аспирант лаборатории машинной графики Института автоматизации и процессов управления ДВО РАН.

Адрес: Владивосток, 690090, ул. Радио, 5, ИАПУ

E-mail: borisoff@dvo.ru

Visualization of city environment by means of plenoptic method

Abstract

We describe an image based rendering algorithm that allow visualizing large spatial scenes. Images acquired by ordinary digital camera are used as inputs. These images are transformed into panoramic views, which are 4D parameterization of the full plenoptic function, for further storing. Using simple proxy geometry, we can create novel view from these structures. The cases when our approach can be used are formulated. The results of our algorithm for both synthetic and real scenes are demonstrated.

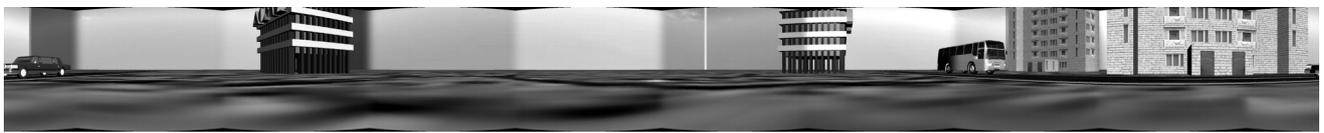
Keywords: *Image-based rendering, Plenoptic function, Panoramic views*

About the author

Yuriy Borisov is a Ph.D. student at Institute for Automation and Control Processes. His contact email is borisoff@dvo.ru.



Рис. 5: Три из двенадцати изображений, снятые для сцены «Кабинет» (этап 1)

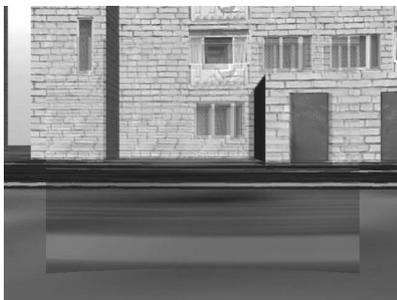


(a)



(b)

Рис 6: Панорамные виды (a) Для виртуальной сцены «Город» (b) Для сцены «Кабинет» (этап 2)



(a)

(b)

(c)

Рис 7: Новые виды, полученные алгоритмом (этап 3)

(a) виртуальная сцена «Город» (b) сцена «Кабинет» (c) сцена «Коридор»