

УДК 004.925

А.Н. Романюк, д-р техн. наук, проф.,  
С.И. Вяткин, канд. техн. наук,  
О.В. Романюк, канд. техн. наук, доц.  
А.И. Черняк, канд. техн. наук, доц.Винницкий национальный технический университет, г. Винница, Украина  
rom8591@gmail.com

## Комбинированный метод рендеринга на основе светового поля и отображения текстур

На данном этапе развития компьютерной графики для обеспечения высокой производительности и реалистичности формирования изображений необходима разработка новых методов и средств. Моделирование и рендеринг на основе изображения позволяют использовать нескольких двумерных изображений для генерации новых двумерных изображений, минуя стадию ручного моделирования. Представлен метод рендеринга, с помощью которого непосредственно визуализируются ракурсы из неструктурированных входных изображений. Вход представляет собой набор исходных изображений вместе с их соответствующей позицией камеры оценки, а также примерный геометрический прокси для сцены. Разработан метод визуализации на основе изображений, который обобщает алгоритмы визуализации светового отображения полей и текстур. Метод позволяет выполнять рендеринг изображений от камер в произвольных конфигурациях. В случае обычной ориентации камер, алгоритм сводится к анализу светового поля. При наличии меньшего количества камер и хорошей приближенной геометрии, алгоритм работает с отображением текстур при различных ракурсах съемки.

**Ключевые слова:** рендеринг на основе изображений, световое поле, отображение текстур, визуализация.

DOI: 10.31474/1996-1588-2020-1-30-42-48

### Введение

Рендеринг на основе изображений стал популярной альтернативой традиционной трехмерной графике. В компьютерной графике и компьютерном зрении моделирование и рендеринг на основе изображений базируются на множестве двумерных изображений сцены для генерации трехмерной модели, а затем рассмотреть с новых ракурсов изображения сцены. Традиционный подход в компьютерной графике использует создание геометрической модели в 3D и проецирование ее на двухмерную плоскость. Компьютерное зрение, наоборот, в основном сосредоточено на выявлении, группировке и выделении признаков (ребер, граней, и т.д.), присутствующих на данном изображении, и их интерпретации. Моделирование и рендеринг на основе изображения позволяют использовать несколько двумерных изображений для генерации новых двумерных изображений, минуя стадию ручного моделирования.

Вместо того, чтобы рассматривать только физическую модель твердого тела, методы на основе изображений больше сосредоточены на моделировании света. Фундаментальное понятие – это функция освещения, которая является

параметризацией светового поля. Функция освещения описывает световые лучи, содержащиеся в данном объеме. Световой луч определяется положением, ориентацией, длиной волны и т.д. Методы на основе изображений аппроксимируют функцию освещения, чтобы сделать новый набор двумерных изображений из других. Учитывая высокую размерность этой функции, практические методы накладывают ограничения на параметры, с тем чтобы уменьшить это число.

Световое поле представляет собой вектор - функцию, которая описывает количество света, проходящего в каждом направлении через каждую точку пространства. Пространство всех возможных световых лучей задается пятимерной функцией, а величина каждого луча задается излучением. Если понятие ограничивается геометрической оптикой некогерентного света и объектов больших, чем длина волны света, то основным носителем света является луч. Мера количества света, проходящего вдоль излучения луча измеряется в ваттах (Вт) встерадиан (sr) на квадратный метр ( $m^2$ ). Стерадиан является мерой телесного угла и используется как мера площади поперечного сечения.

Параметризация луча в 3D-пространстве задается положением  $(x,y,z)$  и направлением

$(\theta, \varphi)$ . Излучение вдоль таких лучей в трехмерном пространстве называется пленоптической функцией [1]. Пленоптическая функция освещения является идеальной функцией, используемой в области компьютерного зрения и компьютерной графики, чтобы получить изображение сцены из любой возможной позиции просмотра под любым углом обзора в любой момент времени. Она фактически никогда не используется на практике в вычислительном отношении, но концептуально полезна для понимания других концепций в компьютерном зрении и графике [2]. Так как лучи в пространстве могут быть параметризованы с помощью трех координат  $X, Y, Z$  и двумя углами  $(\theta, \varphi)$ , это пятимерная функция.

Можно рассматривать время, длину волны и угол поляризации в качестве дополнительных переменных, получая многомерные функции. В пленоптической функции, если интересующая область содержит вогнутый объект, то свет, оставив одну точку на объекте, может перемещаться только на небольшое расстояние, прежде чем будет заблокирован другой точкой на объекте. Нельзя измерить эту функцию в такой области.

Однако если ограничиться местами за пределами выпуклой оболочки объекта, то есть в свободном пространстве, то можно измерить пленоптическую функцию, получая много фотографий с помощью цифровой камеры. Кроме того, в этом случае функция содержит избыточную информацию, потому что излучение вдоль луча остается постоянной от точки к точке вдоль его длины. Информация четырехмерной функции четырехмерного многообразия - 4D световое поле [3], Lumigraph [4]. Формально 4D световые поля определяются как яркости вдоль лучей в пустом пространстве.

Множество лучей в световом поле могут быть параметризованы различными способами. Из них наиболее распространенным является метод двух плоскостей параметризации. Хотя эта параметризация не может представлять все лучи, например, лучи параллельные двум плоскостям, если плоскости параллельны друг другу. Световые поля являются фундаментальным представлением света. Существует много способов создания световых полей с помощью компьютерных программ. В компьютерной графике световые поля, как правило, создаются либо путем рендеринга 3D-модели или с помощью фотографирования реальной сцены.

Таким образом, два эффективных метода рендеринга на основе изображений зависят от вида отображения текстуры [5] и светового поля / люмиграфа [3, 4]. Световое поле и алгоритмы

отображения текстуры во многих отношениях совершенно различны с точки зрения исходных данных. Рендеринг светового поля требует большой коллекции изображений, полученных с помощью камеры, центры которых лежат на регулярных двумерных патчах, и не делается каких-либо предположений о геометрии сцены. В отличие от этого подхода, в алгоритмах отображения текстуры предполагается относительно точная геометрическая модель, с небольшим количеством изображений от входных камер. Эти образы затем проецируются на геометрию для визуализации.

В данной работе исходим из предположения, что в своей основе эти два подхода являются вполне схожими. Оба метода предназначены для интерполяции значений цвета желаемого луча как некоторая комбинация входных лучей. В подходе [5] – эта интерполяция выполняется с помощью геометрической модели, чтобы определить, какие пиксели из каждого входного изображения "соответствуют" искомому лучу в выходном изображении. Из этих соответствующих лучей те, которые ближе всего к нужному лучу взвешиваются, чтобы внести наибольший вклад к интерполированному результату.

Визуализация светового поля [3, 4] может быть интерпретирована аналогичным образом. Для каждого желаемого луча  $(\theta, \varphi)$ .

$(s, t, u, v)$ , ищутся в базе данных изображений лучи, которые пересекаются вблизи некоторой точки  $(u, v)$  на "фокальной плоскости", и имеющие аналогичный угол к нужному лучу, измеряемый пересечением луча с "плоскостью камеры"  $(s, t)$ . В глубинно-скорректированном люмиграфе [4], фокальная плоскость эффективно заменяется приближенной геометрической моделью, что делает этот подход еще более похожим на подход [5].

### Постановка задачи

Учитывая эти связанные подходы к рендерингу на основе изображений, рассмотрим следующие вопросы. Существует ли обобщенная структура визуализации, которая охватывает все эти алгоритмы визуализации на основе изображений? Может ли такой алгоритм хорошо адаптироваться к различному числу входных изображений с камер в общих конфигурациях, а также иметь различные уровни геометрической точности?

В данной работе предлагается набор целей, которые должен иметь любой алгоритм визуализации на основе изображений. Ни один из

предыдущих алгоритмов на основе изображений не удовлетворяет одновременно всем этим условиям. Известные алгоритмы ведут себя достаточно хорошо при соответствующих допущениях по их вводу, но могут неудовлетворительно визуализировать, когда эти предположения нарушаются.

Сочетание точных геометрических данных с Ламбертовскими поверхностными свойствами обеспечивает высокое качество реконструкции из относительно небольшого количества исходных изображений. Процесс реконструкции включает поиск лучей от исходных камер.

С помощью высокоточной геометрической модели, видимость любой точки можно определить относительно конкретной исходной камеры. Если взгляд камеры на точку закрыт какой-то другой точкой на геометрической модели, то эта камера не должна использоваться в реконструкции искомого луча. По возможности, методы на основе изображений должны учитывать видимость при их реконструкции.

Неструктурированный ввод также желателен для рендеринга на основе изображений для приема входных изображений с камер. Оригинальный метод светового поля предполагает, что камеры располагаются в равномерно расположенных положениях на одной плоскости. Этот ограничивает применимость этого метода [6].

В статье [4] описывается система сбора данных, которая использует видеокамеры для получения входных изображений. Они применяются на этапе предварительной обработки, называемой ребинингом, на котором производится повторная выборка входных изображений от виртуальных камер-источников, расположенных на регулярной сетке. Этот процесс ребиннинга добавляет дополнительную процедуру к реконструкции и выборке. Этот дополнительный шаг имеет тенденцию ухудшать общее качество представления. Алгоритм поверхностного светового поля существенно зависит от проблемы с пересчетом.

Когда желаемый луч проходит через центр проекции исходной камеры можно тривиально реконструировать из базы данных лучей, при условии достаточно высокого разрешения входного изображения. Алгоритм с последовательностью эпиполя восстановит этот луч правильно без какой-либо геометрической информации. Алгоритмы светового поля и люмиграфа предназначены специально для поддержания этого свойства. Алгоритмы на основе текстур в реальном времени не обеспечивают это свойство, даже приблизительно, а значит, не будут работать правильно, когда имеется не точная геометрия.

Таким образом, метод должен быть

разработан специально с учетом вышеперечисленных целей. Рендеринг должен происходить с большим разнообразием входных данных. К ним относятся камеры, которые не находятся на общей плоскости, например, исходные изображения, полученные путем перемещения в сцене. Такая конфигурация была бы проблематичной для предыдущих подходов к рендерингу на основе изображений.

### Описание метода

Представлен метод рендеринга, с помощью которого непосредственно визуализируются ракурсы из неструктурированных входных изображений. Вход представляет собой набор исходных изображений вместе с их соответствующей позицией камеры оценки, а также примерный геометрический прокси для сцены.

Алгоритм рендеринга в реальном времени сначала оценивает "камеру поля сглаживания" на множестве вершин в нужной плоскости изображения и интерполирует это поле по всему изображению. Это поле сглаживания описывает, как каждая исходная камера взвешивается для восстановления заданного пикселя. Расчет этого поля основан на заявленных целях, а также включает факторы, связанные с угловой разницей между желаемым лучом, и которые имеются в данном наборе изображений. Учитывая поле сглаживания, каждый пиксель желаемого изображения затем восстанавливается с помощью средневзвешенного значения соответствующих пикселей в каждом взвешенном входном изображении. Камеры взвешиваются на основе сходства углов. Подход обобщается для разрешения и поля зрения. Заданный желаемый луч  $rd$ , пересекает поверхность модели в фронтальной точке  $p$ . Рассматриваем лучи  $ri$  от  $p$  к центрам  $Ci$ , каждой камеры  $i$ . Для каждой камеры определяем угол  $a(i)$ , как угловую разницу между  $ri$  и  $rd$  (рис. 1).

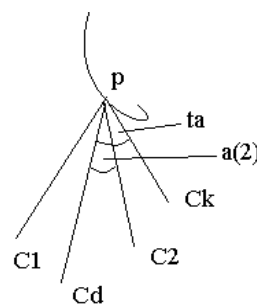


Рисунок 1 – В качестве угла используется угол  $K$ -ой самой дальней камеры  $t_a$

Когда угол  $a(i)$  равен нулю, необходимо получить максимальный вес смешивания  $w_a(i)$ , используемого для камеры  $i$ . Чтобы лучше удовлетворить согласованности эпиполя, этот максимальный вес должен быть бесконечным относительно весов всех остальных камер. Неясно, однако, когда именно  $w_a(i)$ , для конкретной камеры должен упасть до нуля. Например, один из способов определения веса плавного смешивания заключается в установке глобального порогового значения  $t_a$ . Затем, вес  $w_a(i)$  может уменьшиться с  $w_{max}$  до нуля, поскольку  $a(i)$  увеличивается с нуля до  $t_a$ . Такой подход оказывается неудовлетворительным при использовании неструктурированных вводных данных. Для того чтобы учесть нужные пиксели там, где нет угловых закрытых камер, нужно установить большой  $t_a$ . Но при использовании большого  $t_a$  будут смешиваться много камер в пикселях, где есть много угловых закрытых камер, давая излишне размытый результат. Одним из способов решения этой проблемы является использование метода интерполяции  $k$ -ближайшего соседа. То есть рассматриваем только  $k$  камер с наименьшим  $a(i)$  при восстановлении нужного луча. Всем остальным камерам присваивается нулевой вес. При таком подходе, необходимо позаботиться о том, чтобы  $w_a(i)$  конкретной камеры упал до нуля поскольку он покидает множество ближайших  $k$ . Достигается это с определением адаптивного  $t_a$ . Определяем локальный  $t_a$  как  $k$ -е наибольшее значение  $a(i)$  в наборе  $k$ -ближайших камер. Затем вычисляем весовую функцию, имеющую максимальное значение  $w_{max}$  при нуле и нулевое значение при  $t_a$ .

Вес смешивания, используемый в системе реального времени

$$w_a(i) = 1 - \frac{a(i)}{t_a} \quad (1)$$

Эта функция веса имеет максимум 1 и падает линейно до нуля при  $t_a$ , и поэтому она точно не удовлетворяет согласованности эпиполя. Последовательность эпиполя может быть усилена путем умножения  $w_a(i)$  на  $1/a(i)$  (или другими способами) за счет больших вычислений.

Затем нормализуем веса смешивания, чтобы суммировать до единицы

$$w'_a(i) = \frac{w_a(i)}{\sum_{j=1}^k w_a(j)} \quad (2)$$

Это взвешивание хорошо определено до тех пор, пока все  $k$  закрытых камер не равноудаленные. Для данной камеры  $i$ ,  $w'_a(i)$  является гладкой функцией. Помимо угловой разницы, необходимо добавить камерам использование метрик, основанных на разрешении и поле зрения.

С помощью этих различных  $a(i)$ , определяем комбинированную функцию  $a(i)$  как:

$$a_c(i) = \alpha \cdot a(i) + \beta \cdot a_{resolution}(i) + \gamma \cdot a_{fieldofview}(i) \quad (3)$$

где константы  $\alpha$ ,  $\beta$  и  $\gamma$  управляют относительной важностью разных  $a(i)$ . Константа может быть установлена в ноль, чтобы игнорировать  $\gamma a(i)$ . Затем можем определить  $w'_c(i)$ , используя интерполяцию  $k$ -ближайших соседей.

Даны проекционные матрицы эталонных камер, точка  $p$  и нормаль в точке  $p$ , можно предсказать степень рассогласования разрешения при использовании Якобиана плоской гомографии, связывающей желаемый вид с эталонной камерой. Этот расчет включает в себя большинство факторов, приводящих к рассогласованию разрешения, включая расстояние, наклон поверхности, фокусное расстояние и выходное разрешение. Для эффективности упрощаем это вычисление, рассматривая только расстояния от входных камер до изображаемой точки  $p$ . Кроме того, обычно имеем дело только с исходными лучами  $ri$ , которые значительно занижают наблюдаемую прокси-точку  $p$ . Конечно, передискретизация также может привести к проблемам (например, элайзинг), но правильное использование  $map$ -mapping позволяет избежать необходимости пересчета изображения за передискретизацию.

Таким образом, упрощенный вид функции будет

$$a_{resolution}(i) = \max(o, \|p - C_i\| - \|p - C_d\|) \quad (4)$$

где  $C_d$  - центр желаемой камеры.

Необходимо исключить использование лучей, которые падают снаружи поля зрения исходной камеры. Для этого надо включить использование функции:

$$a_{fieldofview}(i) = \begin{cases} 0, & \text{if } ri \text{ within field of view} \\ \infty & \text{otherwise} \end{cases} \quad (5)$$

Чтобы сохранить непрерывность, настраиваем функцию, чтобы она плавно увеличивалась к  $\infty$  по мере приближения  $ri$  к границе изображения  $i$ .

Используем блок глубины сложности, таким

образом, не нужно реализовывать вычисления видимости. Функция видимости присваивает  $\infty$  полностью невидимым точкам и малые значения для видимых точек. Необходимо обеспечить плавный переход из видимой области в невидимую.

Основная стратегия рендеринга в реальном времени заключается в оценке камеры поля смещения в разреженном наборе точек на плоскости изображения. Происходит триангуляция этих точек и интерполяция поля смешивания камеры над остальной частью изображения. Этот подход предполагает, что поле смешивания камеры достаточно гладкое.

Основные процедуры:

- очистка буфера кадров до нуля;
- выбор расположения образцов полей наложения камеры;
- триангуляция образцов полей смешивания.

Выбираем камеру поля смешивания в разреженном наборе местоположений в плоскости изображения. Выбираются места, соответствующие желаемым лучам обзора по простым правилам. Во-первых, проецируются все вершины геометрической модели в желаемый вид и используются эти точки в качестве выборочных местоположений. Для усиления согласованности эпиполя, затем добавляем образец на проекции каждой исходной камеры в нужном ракурсе. Наконец, включаем регулярную сетку образцов на требуемой плоскости изображения для получения достаточно плотного набор образцов, необходимых для фиксации пространственных изменений весов смешивания камеры.

Далее строим ограниченную модель триангуляции Делоне [7,8] образцов поля смещения. Во-первых, добавляем ребра геометрической модели в качестве ограничений на триангуляцию. Это ограничение не позволяет треугольникам пересекать две разные поверхности. Далее добавляем края регулярной сетки как ограничение на триангуляцию. Эти ограничения помогают сохранить триангуляцию от переворачивания, когда желаемая камера перемещается. Учитывая этот набор вершин и ребер ограничений, создаем ограниченную триангуляцию Делоне плоскости изображения. Алгоритм автоматически вставляет новые вершины на всех пересекающихся краях.

В каждой вершине триангуляции, вычисляем и храним набор камер с ненулевыми значениями. веса смешивания и связанные с ними веса смешивания. В вершине эти веса всегда равны единице. Для каждого образца может потребоваться хранение нескольких наборов весов местоположения, если луч выборки пересекает модель несколько раз. Треугольники соседние с этими образцами могут быть

отрисованы многократно на разных плоскостях.

Визуализируем желаемое изображение как набор проективно нанесенных треугольников на карту. Предположим, что существует общее число из  $m$  уникальных камер ( $k \leq m \leq 3k$ , где  $k$  – размер окрестности) с ненулевыми весами смешивания в трех вершинах треугольника.

Затем этот треугольник визуализируется  $m$  раз, используя текстуру из каждой  $m$  камер. Когда треугольник визуализируется с использованием одной из текстур исходной камеры, каждой из трех вершин назначается Альфа-значение, равное весу в этой вершине. Текстура устанавливается для проективной текстуры данных исходной камеры на визуализируемый треугольник. Для отбора проб лучей, пересекающих модель несколько раз, отображаются треугольники, связанные с этими образцами один раз для каждой плоской поверхности, которую они пересекают, с помощью алгоритма  $z$ -buffer для разрешения видимости.

## Результаты

Набор данных изображения (рис. 2) построен из 100 кадров видеопоследовательности, снятой с помощью цифровой видеокамеры. Камера откалибрована для восстановления фокусного расстояния и параметров радиального искажения объектива. Положения камер восстанавливаются с помощью методов получения структуры из движения. Используется одна плоскость для геометрической модели.

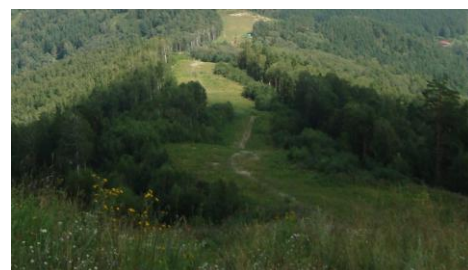


Рисунок 2 – Два виртуальных вида из 100 изображений, снятых цифровой камерой

Положение плоскости вычисляется на основе положений камер и положения трехмерной структуры точек, которые вычисляются во время

відеообробки. Площинка орієнтована паралельно площинкам зображення камери і поміщена на середньому відстані  $1/z$  від камер [1]. Так як камери розташовані приблизно по лінійній траєкторії, існує одна площинка, набір даних зображення показує паралакс тільки в одному вимірі. Тем не менше, ефект переконливий для моделювання видів поблизу висоти, на якій була встановлена відеокамера. Набір даних (рис. 3) був побудований так само, як і набір даних на рис. 2.



Рисунок 3 – Віртуальний вид з 100 зображень, знятих цифровою камерою

Побудовані неструктуровані люмінограми з коротких відеопослідовностей. Послідовність знімків показує залежність від ракурсу зближення і відображення на столешниці.

### Заклучение

Представлен метод візуалізації на основі зображень з неструктурованих вхідних даних. Алгоритм працює в режимі реального часу. Методика узагальнює люмінограф (lumigraph) і алгоритм візуалізації на основі текстур [9]. Реалізація в реальному часі має всі переваги структурованого рендерингу lumigraph, включаючи швидкість і фотореалістичну якість, дозволяючи при цьому використовувати геометричні моделі, неструктуровані вхідні камери і варіації роздільної здатності і поля зору.

### Список літератури

1. Blinn J.F. Models of Light Reflection for Computer Synthesized Pictures. In Computer Graphics, SIGGRAPH'77, 1977. Vol. 11. P. 192-198.
2. Blinn J.F. Simulation of Wrinkled Surfaces. In SIGGRAPH'78, 1978. P. 286-292.
3. Catmull E.A. Subdivision Algorithm for Computer Display of Curved Surfaces. PhD thesis, Univ. of Utah, 1974. Report UTEC-CSc-74-133.
4. Koenderink J.J., van Doorn A.J. Illuminance Texture Due to Surface Mesostructure. J. Opt. Soc. Am. A. 13(3):452-463, 1996.
5. Dana K.J., van Ginneken B., Nayar S.K., Koenderink J. J. Reflectance and Texture of Real-World Surfaces. ACM Transactions on Graphics. 18(1):1-34, 1999.
6. Vyatkin S.I. Complex Surface Modeling Using Perturbation Functions, Optoelectronics, Instrumentation and Data Processing, Volume 43, Number 3, 2007. P. 226-231.
7. Zhu S., Wu Y., Mumford D. Filters, Random Fields and Maximum Entropy Frame-Towards a Unified Theory for Texture Modeling. International Journal of Computer Vision. 27(2):107-126, 1998.
8. Wei L.-Y., Levoy M. Fast Texture Synthesis using Tree-Structured Vector Quantization. In Proceedings of Siggraph-2000. P. 479-488.
9. Leclerc Y.G., Bobick A.F. The Direct Computation of Height from Shading. In Proc. of IEEE Conf. on Comp. Vision and Patt. Recog, 1991. P. 552-558.
10. Faucher, O. Three-Dimensional Computer Vision: A Geometric Viewpoint. The MIT Press: Cambridge, MA USA, 1993. URL: <https://mitpress.mit.edu/books/three-dimensional-computer-vision>
11. Press W.H., Flannery B.P., Teukolsky S.A., and Vetterling W.T. Numerical Recipes in C. Cambridge Univ. Press, New York, 1988.
12. Efros A.A., Leung T.K. Texture Synthesis by Non-parametric Sampling. IEEE International Conference on Computer Vision, Corfu, Greece, 1999. URL: <http://www.faculty.idc.ac.il/arik/seminar2009/papers/efros-iccv99.pdf>

### References

1. Blinn, J.F. (1977), "Models of Light Reflection for Computer Synthesized Pictures. In Computer Graphics", SIGGRAPH'77, Vol.11, pp.192–198.
2. Blinn, J.F. (1978), "Simulation of Wrinkled Surfaces", In SIGGRAPH'78, pp. 286-292.
3. Catmull, E. (1974), "A Subdivision Algorithm for Computer Display of Curved Surfaces", PhD thesis, Univ. of Utah, Report UTEC-CSc-74-133.
4. Koenderink, J.J., van Doorn, A.J. (1996), "Illuminance Texture Due to Surface Mesostructure", J. Opt. Soc. Am. A., 13(3), pp. 452-463.

5. Dana, K.J., van Ginneken, B., Nayar, S.K., Koenderink, J. J. (1999), "Reflectance and Texture of Real World Surfaces", *ACM Transactions on Graphics*, 18(1), pp.1–34.
6. Vyatkin, S.I. (2007), "Complex Surface Modeling Using Perturbation Functions", *Optoelectronics, Instrumentation and Data Processing*, Volume 43, Number 3, pp. 226-231.
7. Zhu, S., Wu, Y., Mumford, D. (1998), "Filters, Random Fields and Maximum Entropy Frameworks Towards a Unified Theory for Texture Modeling", *International Journal of Computer Vision*, 27(2), pp.107-126.
8. Wei, L.-Y., Levoy, M. (2000), "Fast Texture Synthesis using Tree-Structured Vector Quantization", *In Proceedings of Siggraph-2000*, pp. 479-488.
9. Leclerc, Y.G., Bobick, A.F. (1991), "The Direct Computation of Height from Shading", *In Proc. of IEEE Conf. on Comp. Vision and Patt. Recog.*, pp. 552–558.
10. Faugeras, O. (1993), "Three-Dimensional Computer Vision: A Geometric Viewpoint", *The MIT Press: Cambridge, MA USA*, available at: <https://mitpress.mit.edu/books/three-dimensional-computer-vision>
11. Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T. (1988), "Numerical Recipes in C", *Cambridge Univ. Press*, New York.
12. Efros, A.A., Leung, T.K. (1999), "Texture Synthesis by Non-parametric Sampling". *IEEE International Conference on Computer Vision*, Corfu, Greece, September, available at: <http://www.faculty.idc.ac.il/arik/seminar2009/papers/efros-iccv99.pdf>

Надійшла до редакції 15.03.2020

**А. Н. РОМАНЮК<sup>1</sup>, С. И. ВЯТКИН, О. В. РОМАНЮК<sup>1</sup>, А. И. ЧЕРНЯК<sup>1</sup>**

<sup>1</sup>Вінницький національний технічний університет, м. Вінниця, Україна  
rom8591@gmail.com

#### **КОМБІНОВАНИЙ МЕТОД ВІЗУАЛІЗАЦІЇ НА ОСНОВІ СВІТЛОВОГО ПОЛЯ ТА ВІДОБРАЖЕННЯ ТЕКСТУР**

На даному етапі розвитку комп'ютерної графіки для забезпечення високої продуктивності та реалістичності формування зображень необхідна розробка нових методів і засобів. Моделювання та рендеринг на основі зображення дозволяють використовувати декілька двовимірних зображень для генерації нових двовимірних зображень, виключаючи стадію ручного моделювання. Представлено метод візуалізації, за допомогою якого безпосередньо візуалізуються ракурси з неструктурованих вхідних зображень. Вхід являє собою набір вихідних зображень разом з їх відповідною позицією камери оцінки, а також приблизний геометричний проксі для сцени. Розроблено метод візуалізації на основі зображень, який узагальнює алгоритми візуалізації світлового відображення полів і текстур. Метод дозволяє виконувати рендеринг зображень від камер у довільних конфігураціях. У випадку звичайної орієнтації камер алгоритм зводиться до аналізу світлового поля. При наявності меншої кількості камер і добре наближеною геометрією, алгоритм функціонує з відображенням текстур при різних ракурсах зйомки.

**Ключові слова:** рендеринг на основі зображень, світлове поле, відображення текстур, візуалізація.

**O. N. ROMANYUK<sup>1</sup>, S. I. VYATKIN, O. V. ROMANYUK<sup>1</sup>, A. I. CHERNYAK<sup>1</sup>**

<sup>1</sup>Vinnitsia National Technical University, Vinnitsia, Ukraine  
rom8591@gmail.com

#### **COMBINED METHOD OF VISUALIZATION BASED ON LIGHT FIELD AND TEXTURE RENDERING**

At this stage of computer graphics development some new methods and means are required to ensure high productivity and realism of image formation. Image-based modeling and rendering allow using multiple 2D images to generate new 2D images, excluding the manual modeling stage. The method of visualization is presented, by means of which the perspectives from the non-structured inputs are directly visualized. The input is a set of outputs stored together with their respective position of the evaluation camera, as well as approximate geometric proxy for the scene. An image-based visualization method which generalizes algorithms for visualization of light reflection of fields and textures is developed. The method allows the rendering of images from cameras in arbitrary configurations. In the case of the usual orientation of the cameras, the algorithm reduces to the analysis of the light field. With fewer cameras and good geometry approximation, the algorithm works by displaying textures at different shooting angles.

**Keywords:** image rendering, light field, texture rendering, visualization.