Расширение метода иерархических уровней детализации для динамических сцен с детерминированным характером событий

В.А. Семёнов[1,2,3], В.Н. Шуткин[1], В.А. Золотов[1], С.В. Морозов[1,4]
sem@ispras.ru|v451ly@ispras.ru|vladislav.zolotov@ispras.ru|serg@ispras.ru
[1]Институт системного программирования им. В.П. Иванникова РАН, Москва, Россия;
[2]Московский физико-технический институт (национальный исследовательский университет), Долгопрудный, Россия;
[3]Национальный исследовательский университет «Высшая школа экономики», Москва, Россия;
[4]Московский государственный университет имени М.В. Ломоносова, Москва, Россия

Рендеринг больших трёхмерных сцен с убедительным уровнем реализма является серьёзной проблемой компьютерной графики. Одним из распространённых подходов к решению этой проблемы является использование различных уровней детализации (LOD) для объектов сцены в зависимости от их удалённости от наблюдателя. Более эффективным для больших сцен является поход с иерархическими уровнями детализации (HLOD), когда уровни детализации создаются не для каждого объекта индивидуально, а сразу для больших групп объектов. Однако данный метод сталкивается с трудностями, когда в сцене происходят изменения. В данной работе рассматривается класс сцен с детерминированным характером событий и приводится метод для их эффективного рендеринга, основанный на использовании иерархических динамических уровней детализации (HDLOD). Описываются алгоритмы генерации HDLOD и их применение при визуализации сцен.

Ключевые слова: *рендеринг, динамические сцены, полигональные модели, уровни детализации.*

Extension of HLOD Technique for Dynamic Scenes with Deterministic Events

V.A. Semenov[1,2,3], V.N. Shutkin[1], V.A. Zolotov[1], S.V. Morozov[1,4]
sem@ispras.ru|v451ly@ispras.ru|vladislav.zolotov@ispras.ru|serg@ispras.ru
[1]Ivannikov Institute for System Programming of the Russian Academy of Sciences, Moscow, Russia;
[2]Moscow Institute of Physics and Technology, Dolgoprudny, Russia;
[3]National Research University Higher School of Economics, Moscow, Russia;
[4]M.V. Lomonosov Moscow State University, Moscow, Russia

Rendering of large 3D scenes with a convincing level of realism is a challenging computer graphics problem. One of the common approaches to solving this problem is to use different levels of details (LOD) for scene objects, depending on their distance from the observer. Using hierarchical levels of detail (HLOD), when levels of details are created not for each object individually, but for large groups of objects at once, is more effective for large scenes. However, this method faces great challenges when changes occur in the scene. This paper discusses a specific class of scenes with a deterministic nature of events and introduces a method for effective rendering of such scenes based on usage of so-called hierarchical dynamic levels of details (HDLOD). Algorithms for generating HDLOD and their use for visualization of the scenes are also described.

Keywords: *rendering, dynamic scenes, polygonal models, level of details.*

1. Введение

Рендеринг больших трёхмерных сцен с убедительным уровнем реализма является серьёзной проблемой компьютерной графики. Зачастую такие сцены состоят из тысяч или миллионов полигональных моделей, созданных или полученных с очень высоким уровнем детализации. Объекты сцены могут дополняться динамическим поведением, описывающим их появление, движение или исчезновение в соответствии с детерминированным расписанием или случайно происходящими событиями.

Детальные модели с точно определенными расписаниями не всегда являются необходимыми, а зачастую даже мешают эффективному рендерингу сцен с достаточной для интерактивности частотой кадров на современном графическом оборудовании. Высокая производительность рендеринга особенно критична для многих систем CAD/CAM/CAE, функциональность которых включает как отображение сложных сцен, так и манипулирование индивидуальными объектами в пространстве и во времени.

Одним из перспективных подходов к рендерингу сложных сцен является упрощение моделей и использование их серий разного разрешения с подходящими для текущего положения камеры уровнями детализации (LOD). Это направление исследований в компьютерной графике имеет давнюю историю, начавшуюся с введения концепции LOD Джеймсом Кларком в 1976 году. Реализация этой концепции подразумевает, что подходящие уровни детализации будут выбраны и отображены таким образом, что более точные представления будут использоваться для близких объектов, а более грубые — для дальних объектов. Хотя концепция LOD допускает более широкую интерпретацию, основные усилия исследователей были сосредоточены на преобразованиях полигональных представлений.

Были разработаны многочисленные алгоритмы упрощения, преследующие одну общую цель: сократить количество полигонов в сетке, при этом сохранив детали и характеристики оригинала, насколько это возможно. В целом, алгоритмы различаются по основной операции прореживания, используемой для преобразования сетки, и по метрике ошибки, используемой для измерения вносимых искажений. По используемой операции прореживания можно выделить следующие алгоритмы: удаление вершины, стягивание ребра, исключение грани, кластеризация вершин, объединение вершин и другие. Алгоритмы с удалением вершины обычно используют схему с

приоритезацией вершин, согласно которой вершины последовательно выбираются, их соседние грани удаляются, а получающееся отверстие заново триангулируется [11, 12]. Алгоритмы кластеризации вершин используют вспомогательную решетку, окружающую упрощаемую полигональную сетку. В каждой ячейке решетки вершины объединяются в одну новую вершину, инцидентные грани обновляются соответствующим образом [8]. Алгоритмы стягивания рёбер итеративно выбирают ребро, которое трансформируется в одну вершину с вырождением соседних граней [4, 5, 7]. Следует особенно выделить алгоритм [3]. За счёт использования как стягивания рёбер, так и объединения вершин, а также квадратичной метрики ошибки этот алгоритм показывает хорошую производительность и качество упрощения. При этом, что немаловажно, он может применяться для сеток, которые не являются односвязными двумерными многообразиями.

В настоящее время для решения задач визуализации больших сцен широко применяются иерархические уровни детализации (HLOD) [1, 2]. В отличие от традиционных уровней детализации, иерархические предоставляют упрощённые представления не для индивидуальных объектов, а для целых групп, организованных в многоуровневые иерархии. Вместо анализа индивидуальных объектов и выбора подходящего уровня детализации для каждого из них становится возможным обрабатывать сразу целые группы объектов при обходе иерархии уровней детализации. Работа с группами объектов позволяет достичь большей степени упрощения, сократить время обхода дерева сцены и количество вызовов отрисовки.

Однако иерархические уровни детализации затруднительно применять для произвольных динамических сцен. Каждый раз, когда объекты появляются, исчезают или двигаются, их представления должны быть пересчитаны. Время центрального процессора, необходимое для пересчётов, как правило, больше времени графического процессора, необходимого для рендеринга сцены, что делает иерархические уровни детализации бесполезными для динамических сцен. Известные попытки оптимизировать пересчёты благодаря использованию инкрементальных обновлений и их параллельному исполнению на многопроцессорных системах с разделяемой памятью не привели к значимому успеху [2].

В данной статье будут рассматриваться динамические сцены с детерминированным характером событий, описывающих появление и исчезновение объектов сцены. Для эффективной визуализации сцен такого типа предлагается новый подход, названный иерархическими динамическими уровнями детализации (HDLOD). Используя априорное знание о событиях в сцене, данный подход позволяет создавать иерархические уровни детализации, не требующие пересчёта при анимации сцены. Предлагаемый подход принципиально отличается от методов использования уровней детализации при рендеринге сцен с типовыми геометрическими моделями и предопределенными шаблонами поведения, например, сцен моделирования пешеходных потоков [9].

2. Иерархические динамические уровни детализации

Пусть сцена $S(t)$ определена в трёхмерном евклидовом пространстве E^3 на моделируемом временном периоде $t \in [0,T]$ и представлена как линейный список объектов $s(g_s, f_s) \in S$, имеющих фиксированные пространственные позиции и неизменные геометрические представления $g_s \subseteq E^3$. Статус присутствия объектов в сцене определяется их функциями поведения $f_s(t)\colon [0,T] \to \{0,1\}$ таким образом, что функция $f_s(t)$ принимает единичное значение, если объект s присутствует в сцене в момент времени t, и нулевое значение — если отсутствует. В дальнейшем будем называть такие сцены детерминированными псевдо-динамическими. Следует отметить, что движущиеся непрерывно объекты могут быть смоделированы путём дискретизации непрерывного пути и имитации последовательных событий появления и исчезновения экземпляров объекта вдоль этого пути. Предполагается, что в сцене перемещается относительно малое количество объектов, или даже что такие движения вообще отсутствуют. Несмотря на это ограничение, рассматриваемый класс сцен имеет множество индустриальных применений. В частности, это визуальное планирование и моделирование сложных строительных проектов, городских инфраструктурных программ, автоматизированное сборочное машиностроение.

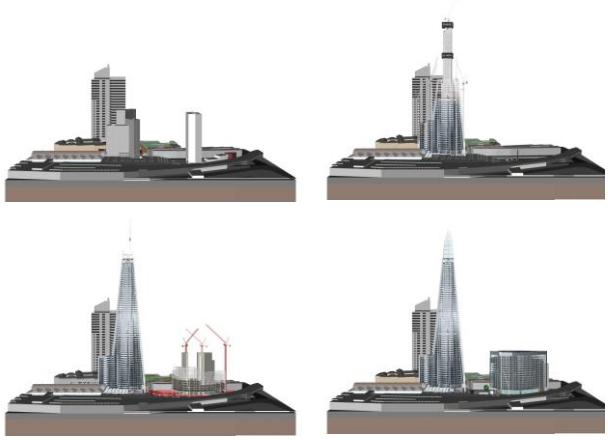

Рис. 1. Пример псевдо-динамической сцены.

На рис. 1 показан пример псевдо-динамической сцены, моделирующей строительство небоскрёба в соответствии с предварительно подготовленным планом проекта. По мере изменения времени моделирования строительные элементы и оборудование устанавливаются на строительной площадке или удаляются. Элементы ландшафта остаются неизменными на протяжение всего моделируемого периода. Такие виды поведения воспроизводятся при помощи стандартных функций, представленных на рис. 2.

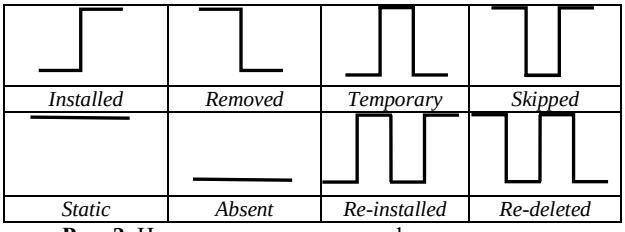

Рис. 2. Некоторые стандартные функции поведения.

В дальнейшем будем предполагать, что объекты сцены геометрически представлены как наборы треугольников с вершинами, ссылающимися на общее множество точек. Не делается никаких предположений насчёт твердотельности объектов или многообразности их граничных представлений, поскольку часто лишь так называемый "полигональный суп" предоставляется для целей рендеринга без гарантий каких-либо топологических свойств.

Будем называть иерархическими динамическими уровнями детализации (HDLOD) дерево кластеров $C(G,F) = \{c(g_c, f_c), \prec\}$, представленное множеством кластеров $c(g_c, f_c)$ с приписанными геометрическими представлениями g_c и функциями поведения $f_c(t)\colon [0,T] \to [0,1]$. В отличие от функций поведения объектов $f_s(t)$,

которые принимают значения 0 или 1, функции поведения кластеров $f_c(t)$ принимают значения на отрезке от 0 до 1, используя таким образом понятие частичной истины. Действительно, поскольку некоторые из объектов кластера могут присутствовать в сцене в некоторый момент времени, в то время как другие могут в этот же момент отсутствовать, невозможно вынести однозначный вердикт о статусе присутствия всего кластера. Для кластеров также определено отношение агломерации $\prec$ таким образом, что $c' \prec c$ тогда и только тогда, когда кластер $c' \in C$ является прямым потомком (в дереве) кластера $c \in C$. Листья дерева являются точно заданными индивидуальными объектами. Внутренние узлы представляют собой кластеры, агрегирующие геометрию и поведение объектов в соответствующих поддеревьях. Корень — это кластер, содержащий наименее точное и наиболее упрощённое представление всей сцены. На рис. 3 показан пример несбалансированного HDLOD дерева с узлами различной степени.

Каждый кластер $c \in C$ хранит не только геометрическое и поведенческое представление, но также и такие производные атрибуты, как: ограничивающий параллелепипед b_c, размер или мощность w_c, а также пространственная погрешность ε_c и временная погрешность γ_c, которые будут определены ниже. Фактически эти атрибуты насчитываются при генерации иерархических динамических уровней детализации и используются при их отображении. Таким образом, каждый HDLOD кластер представляется как кортеж $c(g_c, f_c, b_c, w_c, \varepsilon_c, \gamma_c)$.

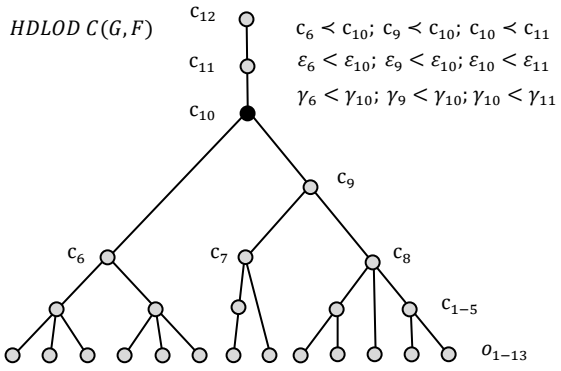

Рис. 3. Дерево HDLOD и некоторые агломерационные и алгебраические соотношения

Пространственную погрешность ε_c можно определить как абсолютную погрешность, которая устанавливает допустимое максимальное локальное отклонение геометрического представления кластера c от агрегированного представления объектов $o \prec .. \prec c' \prec c$:

$$\varepsilon_c = \max_{c' \prec c}(\varepsilon_{c'}) + D_H\left(g_c, \bigcup_{c' \prec c} g_{c'}\right).$$

Здесь погрешность ε_c определена рекуррентно с использованием метрики Хаусдорфа $D_H(A,B)$, которая представляет собой наибольшее из всех расстояний от точки из одного множества до ближайшей точки другого множества:

$D_H(A,B) =$
$max\{max_{x\in A}\, min_{y\in B}\, D(x,y), max_{y\in B}\, min_{x\in A}\, D(x,y)\}$,
где A, B — замкнутые множества точек и $D(x,y)$ — функция метрики в Евклидовом пространстве.

Временную точность γ_c можно определить как максимальное отклонение функции поведения кластера от индивидуальных функций поведения оригинальных объектов:

$$\gamma_c = \max_{c' \prec c}\left(\gamma_{c'} + D_F(f_c, f_{c'})\right),$$

где расстояние $D_F(f_A, f_B)$ вычисляется с использованием функциональной метрики:

$$D_F\left(f_A(t), f_B(t)\right) = \frac{1}{T}\int_0^T |f_A(t) - f_B(t)|dt.$$

Данные параметры используются для оценки пространственной близости объектов, отклонения геометрии и близости временных поведений. Пространственные погрешности вычисляются в процессе упрощения геометрии кластера. Временные погрешности должны быть вычислены вместе с функцией поведения кластера. Для вычисления функции поведения кластера предлагается использовать следующую формулу:

$$f_c(t) = \frac{\sum_{c' \prec c} w_{c'} f_{c'}(t)}{\sum_{c' \prec c} w_{c'}}.$$

Использование взвешенных сумм позволяет в большей степени учитывать поведение значимых объектов и надлежащим образом воспроизводить поведение кластера. Если функция $f_c(t)$ принимает единичное значение, значит все объекты кластера присутствуют в сцене в момент времени t. Если функция принимает нулевое значение, все объекты кластера отсутствуют в момент времени t.

Для каждого кластера необходимо принять решение о том, стоит ли отображать его, проигнорировать, или использовать более точные дочерние представления. Для этого вычисляется зависящая от времени пространственная погрешность $\delta_c(t)$ (пространственная ошибка, вызванная как геометрическим, так и временным упрощением):

$$\delta_c(t) = \begin{cases} 0, & \text{при } f_c(t) = 0 \\ \varepsilon_c, & \text{при } f_c(t) = 1 \\ \varepsilon_c + (1 - f_c(t)) \sum_{c' \prec c} w_{c'}, & \text{при } \frac{1}{2} \le f_c(t) < 1 \\ \varepsilon_c + f_c(t) \sum_{c' \prec c} w_{c'}, & \text{при } 0 < f_c(t) < \frac{1}{2} \end{cases}$$

При визуализации сцены совершается обход дерева кластеров. В каждом узле атрибуты кластера анализируются для определения необходимости дальнейшего обхода поддерева. Проверяется, присутствует ли кластер в сцене в указанное время моделирования ($f_c(t) > 0.5$), попадает ли ограничивающий параллелепипед b_c кластера в конус видимости, а также является ли погрешность $\delta_c(t)$ кластера достаточной для получения необходимого качества. Если все условия удовлетворены, представление кластера немедленно выбирается для отображения. В данном случае всё поддерево может быть исключено из обхода. Если третье условие не удовлетворяется, продолжается обход поддерева кластеров и дочерние узлы проходят такие же проверки, пока не будут достигнуты листовые узлы с точно заданными объектами сцены. Псевдокод описанного алгоритма представлен на рис. 4.

```
PROCEDURE DISPLAY (HDLOD tree, VIEW view, TIME time,
                   RESOLUTION resolution)
{
DISPLAY CLUSTER (ROOT (tree), view, time, resolution)
}

PROCEDURE DISPLAY CLUSTER (CLUSTER node, VIEW view, TIME time,
                           RESOLUTION resolution)
{
IF (VALUE OF (BEHAVIOR FUNCTION (node), time) EQUAL 0)
    RETURN
ELSE IF (IS OUTSIDE FRUSTUM (BOUNDING BOX (node), view))
    RETURN
ELSE IF (VALUE OF (DELTA FUNCTION (node), time) /
         DISTANCE (node, view) < resolution)
    RENDER (GEOMETRY (node), view))
ELSE
    {
    SET OF CLUSTER children = CHILDREN NODES (node)
    FOR_EACH (CLUSTER child IN children)
        DISPLAY CLUSTER (child, view, time, resolution)
    }
}
```

Рис. 4. Псевдокод алгоритма отображения HDLOD.

3. Генерация HDLOD

Иерархические динамические уровни детализации могут быть сгенерированы автоматически при помощи предложенного метода, который использует иерархическую (снизу-вверх) кластеризацию и многоуровневый контроль точности. Метод начинает с индивидуальных объектов и последовательно группирует их во всё большие и большие кластеры, пока не будет достигнуто желаемое количество уровней. Корневые кластеры могут быть в дальнейшем ещё сильнее упрощены, если потребуется отображать эту сцену как часть более сложной композиции. Кластеризация должна проводиться при строгом контроле точности, результирующее дерево HDLOD должно удовлетворять множеству требований и условий касательно ожидаемого количества уровней детализации, степени узлов, пространственной плотности и перекрытия дочерних кластеров, их временной близости и снижения сложности кластеров с повышением уровня.

К сожалению, классические методы кластеризации не могут быть напрямую применены к проблемам генерации HDLOD. Предъявляемые требования достаточно сложны и могут противоречить друг другу. Это препятствует математической формализации метрических функций и критериев связности, необходимых для методов кластеризации [13]. Неприемлемо высокая вычислительная сложность этих методов является другой причиной, препятствующей адаптации классических результатов.

Например, наивная реализация агломеративной кластеризации имеет временную сложность $O(n^3)$ и требует $O(n^3)$ памяти, что делает её бесполезной даже для простых сцен. Более быстрая иерархическая кластеризация с временной сложностью $O(n^2)$ и потреблением памяти $O(n^2)$ также не подходит для решения обсуждаемых проблем [13].

Поэтому предлагается формировать деревья HDLOD, руководствуясь другими принципами. Процесс кластеризации разделяется на шаги, на каждом из которых сформированные кластеры удовлетворяют определённым требованиям по точности. По мере того как делаются новые шаги, эти требования ослабляются таким образом, чтобы гарантировать завершение процесса после определённого количества шагов, равного числу уровней детализации L. Лишь активные кластеры участвуют в этом процессе в противовес пассивным кластерам, которые уже были проанализированы и сгруппированы.

На каждом запланированном шаге метода l $(1 \le l \le L)$ делается попытка сформировать новые кластеры с погрешностями $\varepsilon_c \le \varepsilon(l)$, $\gamma_c \le \gamma(l)$. Для этого пороговые значения $\varepsilon(l)$, $\gamma(l)$ предварительно насчитываются таким образом, что они монотонно увеличиваются с увеличением уровня $\varepsilon(l-1) < \varepsilon(l)$, $\gamma(l-1) < \gamma(l)$. Пороговые погрешности могут быть выбраны следующим образом:

$$\varepsilon(l) = w^{1-l/L}/R,\ \ \gamma(l) = 1/\tau + (1-1/\tau)\cdot w^{1-l/L},$$

где w — это размер наименьшего из объектов сцены, отнесённый к размеру всей сцены (например, диагональ параллелепипеда, ограничивающего все объекты сцены за весь период моделирования, в том числе временно установленные и удаляемые), R — эффективное разрешение (число пикселей по одной из осей, делённое на поле зрения камеры) и τ — эффективная временная дискретизация, используемая для анимационных целей (типичный временной шаг между последовательными событиями, отнесённый ко всему моделируемому периоду).

На каждом шаге метода представители из числа активных кластеров выбираются с использованием кривых пространственного заполнения и соответствующих упорядочений Гильберта [10]. С одной стороны, это позволяет выбирать представителей во всём объеме сцены, с другой — локализовать их в плотно заполненных областях. Далее для каждого представителя осуществляется поиск соседей как в пространстве, так и во времени. Соседи должны удовлетворять условиям по пространственной и временной близости, а также гарантировать формирование родительского кластера с запланированной точностью. Если для текущего представителя не удалось найти подходящих соседей, он исключается из анализа на текущем шаге, но будет участвовать в следующих. Функция поведения родительского кластера определяется путем вычисления взвешенной функции поведения $f_c(t)$, приведённой выше. Результирующее геометрическое представление родительского кластера g_c получается объединением полигональных моделей дочерних кластеров. Оно должно быть упрощено для достижения необходимого снижения сложности с повышением уровня, например, с использованием упомянутого алгоритма [3]. Псевдокод описанного алгоритма представлен на рис. 5.

Используя предварительно вычисленные и развёрнутые индексы, кластеризация может быть осуществлена за $O(n \log n)$, где n — количество объектов сцены. В сравнении с классическими методами кластеризации, упомянутыми выше, этот результат является гораздо более приемлемым для больших сцен. Для быстрого поиска соседей могут применяться различные индексные структуры [10], в частности, для псевдо-динамических сцен показывают хорошую производительность регулярные динамические октодеревья [6]. В данной реализации использовались простые упорядочения по каждой из координат.

```
PROCEDURE GENERATE HDLOD (SCENE scene, INTEGER levels,
                          HDLOD tree)
{
SET OF CLUSTER active = NULL, next = NULL
FOR_EACH (OBJECT object IN OBJECTS (scene))
    {
    CLUSTER cluster = FORM CLUSTER (object)
    ADD TO (cluster, tree)
    ADD TO (cluster, active)
    }
FOR_EACH (INTEGER step = 1 TO levels)
    {
    REAL epsilon, gamma
    COMPUTE LEVEL THRESHOLDS (scene, levels, step, epsilon,
                              gamma)
    WHILE (NOT EMPTY (active))
        {
        CLUSTER representative = SELECT REPRESENTATIVE (active)
        SET OF CLUSTER neighbors = FIND NEIGHBORS (active,
            representative, epsilon, gamma)
        IF (IS EMPTY (neighbors))
            {
            ADD TO (representative, next)
            REMOVE FROM (representative, active)
            }
        ELSE
            {
            SET OF CLUSTER children
            ADD TO (representative, children)
            FOR_EACH (CLUSTER neighbor IN neighbors)
                ADD TO (neighbor, children)
            CLUSTER cluster = CREATE CLUSTER (children)
            SIMPLIFY (cluster, epsilon, gamma)
            ADD TO (cluster, tree)
            ADD TO (cluster, next)
            REMOVE FROM (representative, active)
            FOR_EACH (CLUSTER neighbor IN neighbors)
                REMOVE FROM (neighbor, active)
            }
        }
    COPY (next, active)
    EMPTY (next)
    }
}
```

Рис. 5. Псевдокод алгоритма генерации HDLOD.

4. Вычислительные эксперименты

Для того чтобы апробировать введённую концепцию HDLOD, а также предложенный метод для автоматической генерации и визуализации HDLOD, была проведена серия вычислительных экспериментов. Были измерены значения времени рендеринга кадра при навигации по заданной сцене при фиксированных моментах времени моделирования, а также среднее время кадра для анимации на протяжении

всего периода моделирования. В качестве тестовой была выбрана представленная на рис. 1 динамическая сцена строительства небоскрёба. Модель состоит из 79396 строительных элементов, представленных полигональными сетками, суммарно содержащими 3632126 треугольников. План строительства включает 67099 активностей, каждая из которых отвечает за определённые работы на строительной площадке и порождает соответствующие события в динамической сцене.

Для оценки эффективности иерархических динамических уровней детализации и исключения фактора отсечения конусом видимости эксперименты были проведены для камеры, расположенной на различных расстояниях от сцены. В первом положении камера была размещена на максимально близком расстоянии, при котором сцена была видна полностью. В других положениях камера размещалась на таких расстояниях, что сцена занимала одну четвёртую и одну шестнадцатую области экрана. Были выбраны положения времени в начале и в конце моделируемого периода, а также в одной третьей и в двух третях модельного периода. Вычислительные эксперименты проводились на компьютере типичной конфигурации: Intel Core i7-4790 CPU (3.6 GHz), 16 GB of RAM, GeForce GTX 750 Ti (2 GB).

	1/1 экрана	**1/4** экрана	**1/16** экрана	Без HDLOD
Начало	7.02	1.5	0.66	40.88
1/3 периода	8.7	2.75	0.71	82.9
2/3 периода	26.47	2.93	0.74	121.3
Конец	28.17	3.15	0.76	164.83
Анимация	18.63	2.87	0.72	105.4

Таблица 1. Время кадра (в миллисекундах) при визуализации сцены строительства небоскрёба.

В таблице 1 приведены результаты измерений производительности в ходе описанных экспериментов. В последней колонке содержатся результаты, полученные при рендеринге индивидуальных объектов без использования иерархии упрощённых представлений HDLOD. Очевидно, что применение HDLOD значительно повышает производительность, и по мере удаления камеры от сцены достигаемый эффект растёт. При приближении камеры к отдельному объекту или группе объектов эффект от применения HDLOD несущественен, поскольку доминирующим фактором становится отсечение сцены конусом видимости. Подобное поведение наблюдается как при навигации по статической сцене, зафиксированной в выбранные моменты времени, так и при анимации сцены. Проведенная серия экспериментов была распространена на другие задачи визуального моделирования проектов строительства, городских инфраструктурных программ, машиностроительных процессов. Ее результаты подтверждают высокую эффективность и масштабируемость предложенного метода.

5. Заключение

Таким образом, в данной статье была представлена концепция иерархических динамических уровней детализации (HDLOD) и методы для их автоматической генерации и визуализации. В отличие от традиционных методов уровней детализации, подходящих только для статических сцен, разработанный метод применим к широкому классу детерминированных псевдо-динамических сцен, возникающих в многочисленных индустриальных приложениях. Дальнейшая работа будет посвящена исследованию алгоритмических вариантов разработанного метода, а также его обобщениям для сцен с непрерывно движущимися объектами.

6. Литература

[1] Cesium 3D Tiles. Beyond 2D Tiling. 2016. https://cesium.com/presentations/files/FOSS4GNA2016/3DTiles.pdf

[2] Erikson, C. et al., 2001. HLODs for Faster Display of Large Static and Dynamic Environments. I3D '01 Proceedings of the 2001 symposium on Interactive 3D graphics. New York, USA, pp. 111-120.

[3] Garland, M. and Heckbert, P.S., 1997. Surface simplification using quadric error metrics. SIGGRAPH '97 Proceedings of the 24th annual conference on Computer graphics and interactive techniques. New York, USA, pp. 209-216.

[4] Guéziec, A., 1995. Surface Simplification With Variable Tolerance. 2nd Annual International Symposium on Medical Robotics and Computer Assisted Surgery, Baltimore, USA, pp. 132-139.

[5] Hoppe, H., 1996. Progressive Meshes. SIGGRAPH '96 Proceedings of the 23rd annual conference on Computer graphics and interactive techniques. New York, USA, pp. 99-108.

[6] Morozov, S. et al, 2018. Indexing of Hierarchically Organized Spatial-Temporal Data Using Dynamic Regular Octrees. Petrenko A., Voronkov A. (eds) Perspectives of System Informatics. PSI 2017. Lecture Notes in Computer Science, vol 10742, pp. 276-290.

[7] Ronfard, R. and Rossignac, J., 1996. Full-range approximation of triangulated polyhedra. Computer Graphics Forum, Vol. 15, No. 3, pp. 67-76.

[8] Rossignac J. and Borrel P., 1993. Multi-resolution 3D approximations for rendering complex scenes. Falcidieno B., Kunii T.L. (eds) Modeling in Computer Graphics. IFIP Series on Computer Graphics. Springer, Berlin, Heidelberg, Germany.

[9] Rudomin, I. et al, 2014. Hierarchical level of detail for varied animated crowds. Visual Computer Vol. 30(6-8), pp. 949-961.

[10] Samet, H., 2006. Foundations of Multidimensional and Metric Data Structures. Morgan Kaufmann, Burlington, USA.

[11] Schroeder, W.J. et al., 1992. Decimation of Triangle Meshes. SIGGRAPH '92 Proceedings of the 19th annual conference on Computer graphics and interactive techniques. New York, USA, pp. 65-70.

[12] Soucy, M. and Laurendeau, D., 1996. Multiresolution Surface Modeling Based on Hierarchical Triangulation. Computer Vision and Image Understanding, Vol. 63, No. 1, pp. 1-14.

[13] Xu, D. and Tian, Y., 2015. A Comprehensive Survey of Clustering Algorithms. Annals of Data Science, Vol. 2, pp. 165-193.

Об авторах

Семёнов Виталий Адольфович, д.ф.-м.н., проф., зав. отделом Института системного программирования им. В.П. Иванникова РАН. E-mail: sem@ispras.ru.

Шуткин Василий Николаевич, аспирант Института системного программирования им. В.П. Иванникова РАН. E-mail: v451ly@ispras.ru.

Золотов Владислав Александрович, к.ф.-м.н., н.с. Института системного программирования им. В.П. Иванникова РАН. E-mail: vladislav.zolotov@ispras.ru.

Морозов Сергей Вячеславович, к.ф.-м.н., в.н.с. Института системного программирования им. В.П. Иванникова РАН. E-mail: serg@ispras.ru