
КОМПЬЮТЕРНАЯ ГРАФИКА И ВИЗУАЛИЗАЦИЯ

УДК 004.946

360-ВИДЕО НА ОСНОВЕ ПРАВИЛЬНОГО ДОДЕКАЭДРА: ТЕХНОЛОГИЯ И МЕТОДЫ РЕАЛИЗАЦИИ В СИСТЕМАХ ВИРТУАЛЬНОГО ОКРУЖЕНИЯ

© 2021 г. П. Ю. Тимохин^{a,*}, М. В. Михайлюк^{a,**}, К. Д. Пантелей^{a,***}

^a ФГУ «ФНЦ Научно-исследовательский институт системных исследований РАН»,
117218 Москва, Нахимовский пр., д. 36, к. 1, Россия

*E-mail: webpismo@yahoo.de

**E-mail: mix@niisi.ras.ru

***E-mail: kpanteley@mail.ru

Поступила в редакцию 23.12.2020 г.

После доработки 25.12.2020 г.

Принята к публикации 14.01.2021 г.

В статье предлагаются новые технология и методы реализации панорамного видео с обзором 360 градусов, основанные на проекции виртуального окружения на правильный додекаэдр. Идея состоит в построении виртуальной панорамы, наблюдаемой зрителем, из прямоугольных снимков виртуального пространства, имитирующих внутреннюю поверхность додекаэдра. Разработан метод вычисления параметров проекции и ориентации 12 камер додекаэдра, основанный на геометрии “золотых прямоугольников”, метод и алгоритмы синтеза кадра 360-видео, основанные на оригинальной схеме упаковки пентагонов, а также метод и алгоритм визуализации прямоугольных снимков, обеспечивающий синтез непрерывной виртуальной панорамы. Предложенные решения реализованы в программном комплексе и апробированы на примере задачи визуализации полета по орбите МКС над земной поверхностью. Результаты исследования могут быть применены в системах виртуального окружения, видеосимуляторах, научной визуализации, виртуальных лабораториях, образовательных приложениях, видеонструкциях и др.

DOI: 10.31857/S0132347421030109

1. ВВЕДЕНИЕ

Во многих современных научных исследованиях и программах подготовки прикладных специалистов активно востребованы системы виртуального окружения [1]. Особенно это актуально для областей, где цена ошибки крайне высока: космос, авиация, медицина, нефтегазовая и атомная промышленность и др. [1–5]. Современные системы виртуального окружения являются достаточно сложными (нередко стационарными) программно-аппаратными комплексами, для запуска и настройки которых требуется участие VR-специалиста. Вместе с этим часто возникают ситуации, когда необходимо оперативно визуализировать уже пройденный процесс работы в виртуальном окружении, например, для анализа ошибок, допущенных при обучении в симуляторах, обмена результатами между удаленными исследователями, создания образовательных приложений и инструкций, и др. Одним из эффективных подходов является исследование и разработка новых технологий и форматов реализации видеозаписи виртуального окружения, при просмотре которой

зритель испытывал бы эффект погружения. Примерами являются технологии, основанные на различных форматах стерео, например, технология отложенного синтеза 4К-стереороликов в сложных динамических виртуальных сценах [6] или технология создания стереоанимаций на основе автостереоскопических мониторов [7, 8]. Благодаря бурному прогрессу экшн-камер (GoPro) и видеохостинга YouTube, в последнее время стало активно развиваться направление 360-видео [9], при котором съемка пространства выполняется с охватом 360 градусов по горизонтали и вертикали. Каждый кадр такого видео содержит изображение сферической панорамы, окружающей зрителя, что дает возможность пользователю произвольно менять направление взгляда в процессе воспроизведения видео и ощущать эффект присутствия в визуализируемом пространстве. Как известно, поверхность сферы невозможно отобразить на плоскость без искажений, поэтому изображение сферической панорамы обычно формируют из нескольких частей, которые при воспроизведении сшивают между собой. Однако, в виртуальном пространстве визуальные и гео-

метрические свойства объектов часто моделируются с помощью различных имитационных техник (для оптимизации расхода вычислительного ресурса), которые при крутом повороте ракурса снимающей виртуальной камеры могут давать отображения, проблематично сшиваемые между собой. В виду этого возникает задача разработки технологий и методов реализации 360-видео в системах виртуального окружения, направленных на обеспечение непрерывности синтезируемой 360-видеопанорамы виртуального пространства. Для решения этой задачи в данной работе предлагается новая технология, основанная на отображении сферической виртуальной панорамы на поверхность правильного додекаэдра, а также методы создания и воспроизведения такого 360-видео. Предложенное решение реализовано на языке C++ с использованием графической библиотеки OpenGL.

2. СОВРЕМЕННЫЕ ПОДХОДЫ К РЕАЛИЗАЦИИ 360-ВИДЕО

Анализ современных научных публикаций и докладов разработчиков ведущих IT-компаний (YouTube, Facebook, GoPro) показывает высокую востребованность новых технологий и методов реализации 360-видео. Можно выделить два основных направления, по которым ведутся исследования и разработки в данной области.

Первое направление включает в себя технологии и методы, в которых сферическая панорама разворачивается на плоскость с помощью одной или нескольких картографических проекций [10, 11]. Так, в популярном видеохостинге YouTube базовым форматом для добавления 360-видео является эквидистантная цилиндрическая проекция [12]. В ней масштаб отображаемой панорамы является истинным только вдоль экватора, а при приближении к полюсам он возрастает до бесконечного. Чтобы обеспечить более однородный масштаб, в исследовании [13] предлагается отображать сферическую панораму на горизонтальные полосы, длины которых уменьшены в соответствии с близостью к полюсам (при этом сами области полюсов записываются в виде двух отдельных (“крышек”).

Ко *второму направлению* относятся решения, в которых сферическая панорама отображается на некоторый выпуклый многогранник, который разворачивается на плоскость. Широкое распространение получила кубическая проекция (cube map projection) [14–17], при которой выполняется перспективное отображение панорамы на шесть граней куба. Хотя такой подход не дает вырожденные полюсы, как в эквидистантной проекции, детализация получаемого отображения все же имеет существенный разброс от центра до границ граней куба (до трех раз). Если не скомпенсиро-

вать такой разброс повышением разрешения всего отображения, то при проигрывании 360-видео будут наблюдаться артефакты “блочности”, что приведет к потере эффекта присутствия [14]. Для повышения качества получаемого 360-видео были также разработаны различные усовершенствования кубической проекции: равноугольная кубическая проекция [15], смещение центра съемки из центра куба [16], ориентирование граней куба на объекты переднего плана [17] и др. Несмотря на большую популярность кубической проекции, были также предложены и принципиально отличные способы отображения сферической панорамы. Так, в исследовании [18] панорама проецируется на ромбододекаэдр, а в публикации [19] разработчики из компании Facebook предложили проецировать панораму на грани тридцати пирамид видимости, равномерно охватывающих сферу. Несмотря на определенные преимущества перед кубической проекцией (уменьшение разброса детализации, сокращение объема видео-файла), данные подходы не получили развития в виду сложности реализации.

В ранних публикациях, посвященных разработке устройства для панорамного телевидения [20, 21], отмечается уникальное свойство правильного додекаэдра, делающее его лучшим выбором для съемки сферической панорамы. Правильный додекаэдр является единственным Платоновым телом, которое не только по форме близко к сфере, но и грани (пентагоны) которого близки к окружности. Благодаря этому свойству, разброс детализации отображения внутри пентагонов получается более сбалансированным, чем у куба или икосаэдра, а переход смежных ракурсов друг в друга – плавным (двухгранный угол около 116 градусов), что очень важно для непрерывной стыковки частей виртуальной панорамы [22]. В данной работе предлагаются новые технологии и методы реализации 360-видео в системах виртуального окружения, которые основаны на правильном додекаэдре и позволяют осуществлять синтез эффективного по площади кадра из отображений всех пентагонов, а также выполнять реконструкцию непрерывной панорамы из такого кадра.

3. ТЕХНОЛОГИЯ РЕАЛИЗАЦИИ 360-ВИДЕО НА ОСНОВЕ ПРАВИЛЬНОГО ДОДЕКАЭДРА

Рассмотрим задачу реализации 360-видео для наблюдателя, выполняющего исследование виртуального пространства, при котором он может изменить свое положение и направление взгляда. Обозначим через P_v , v и u позицию, направление взгляда и вектор “вверх” наблюдателя в некоторый момент времени в мировой системе координат (World Coordinate System, WCS [23]). Расположим вокруг наблюдателя воображаемый правиль-

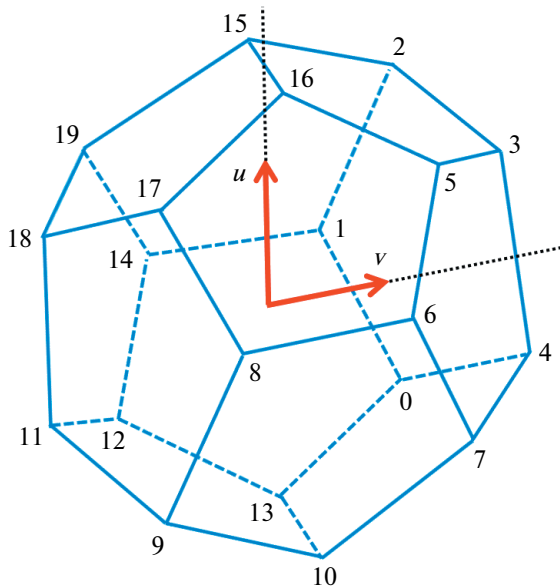


Рис. 1. Воображаемый додекаэдр.

ный додекаэдр, центр которого совпадает позицией P_V , а ориентация жестко связана с векторами $\mathbf{v}$ и $\mathbf{u}$ (см. рис. 1). В предлагаемой технологии реализации 360-видео мы вводим *камеры додекаэдра* – 12 одинаковых виртуальных камер $C_0, \dots, C_{11}$ с симметричными пирамидами видимости, которые расположены в позиции P_V наблюдателя и направлены в центры граней (пентагонов) воображаемого додекаэдра.

Предлагаемая технология включает в себя три этапа. На *первом этапе* выполняется вычисление параметров проекции и ориентации камер додекаэдра, при которых каждый пентагон будет вписан в область вывода направленной в него камеры. На *втором этапе* для каждого момента времени синтеза кадра 360-видео выполняется съемка виртуального пространства через камеры додекаэдра и упаковка полученных “снимков” в видеокадр оригинального формата. Первый и второй этапы реализуются на стороне создателя 360-видео (в системе виртуального окружения). На *третьем этапе* для каждого кадра 360-видео из упакованных “снимков” выполняется реконструкция внутренней поверхности додекаэдра, имитирующей видимую панораму виртуального пространства. Данный этап выполняется на стороне зрителя 360-видео (в плеере). Рассмотрим более подробно методы реализации этих этапов.

3.1. Метод вычисления параметров проекции и ориентации камер додекаэдра

Параметры проекции. Как отмечалось выше, камеры додекаэдра имеют одинаковые пирамиды видимости, симметричные по вертикали и гори-

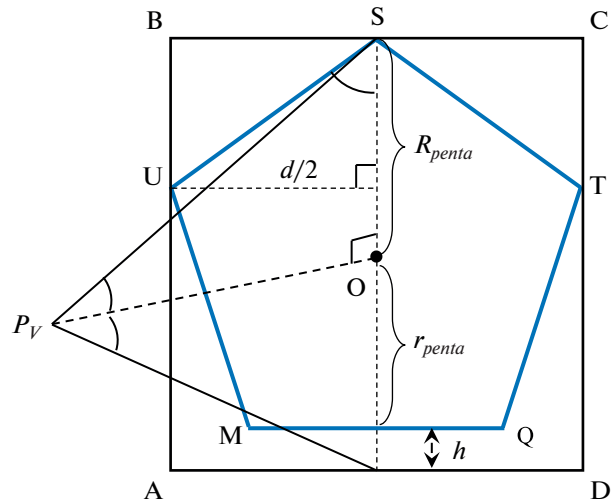


Рис. 2. Пентагон, вписанный в область вывода камеры додекаэдра.

зонтали. На рис. 2 показано расположение пентагона в области вывода C_i -й камеры додекаэдра, при котором их центры совпадают. Нетрудно заметить, что пентагон можно также вписать в область вывода и без расстояния h , если использовать камеру с наклонной пирамидой видимости (oblique frustum). Однако, в этом случае проекция виртуального пространства на нижней половине пентагона будет иметь больший масштаб, чем на верхней, что нарушает непрерывность стыковки снимков, получаемых от соседних камер додекаэдра.

Чтобы задать симметричную пирамиду видимости для каждой C_i -й камеры додекаэдра, вычислим угол γ вертикального раствора и отношение *aspect* ширины к высоте области вывода.

Из рис. 2 видно, что размеры области вывода C_i -й камеры должны быть пропорциональны размерам прямоугольника $ABCD$. Его центр O совпадает с центром пентагона $MQRST$, $AB = CD = 2R$, $BC = AD = d$, а AD отстоит от MQ на расстоянии $h = R - r$, где R, r – радиусы описанной и вписанной в пентагон окружностей, а d – диагональ пентагона [24]:

$$\begin{aligned} R &= (MQ/2) \operatorname{cosec}(\pi/5), \\ r &= (MQ/2) \operatorname{ctg}(\pi/5), \quad d = 2MQ \sin(3\pi/10). \end{aligned} \quad (1)$$

Тогда величину *aspect* области вывода камеры C_i можно выразить как

$$\operatorname{aspect} = \frac{BC}{AB} = \frac{d}{2R} = \frac{2 \sin(3\pi/10)}{\operatorname{cosec}(\pi/5)} \approx 0.951. \quad (2)$$

Угол γ найдем из прямоугольного $\Delta P_V OS$. Его катет $P_V O$ является радиусом сферы, вписанной в додекаэдр, и находится как $P_V O = (MQ/2) \operatorname{ctg}(\pi/5) \operatorname{tg}(\theta/2)$, где $\theta = 2 \arcsin$

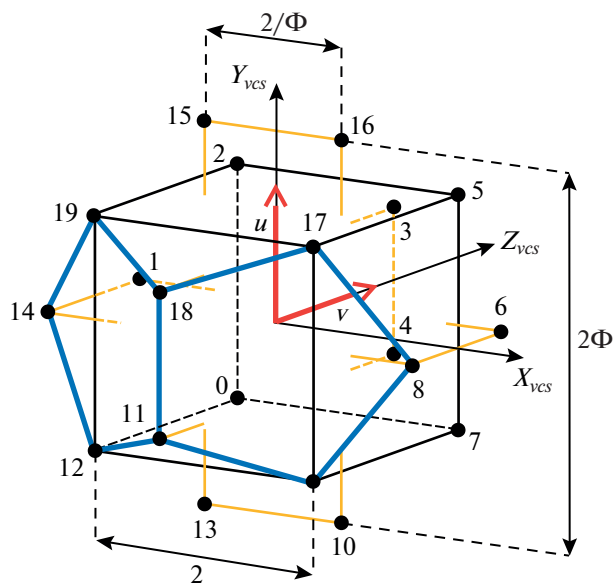


Рис. 3. Связь системы координат наблюдателя с воображаемым додекаэдром.

$(\cos(\pi/3)/\sin(\pi/5)) \approx 116.565^\circ$ – двугранный угол правильного додекаэдра [24]. Из ΔP_0OS :

$$\begin{aligned} \operatorname{tg}(\gamma/2) &= \frac{R}{P_0O} = \frac{\operatorname{cosec}(\pi/5)}{\operatorname{ctg}(\pi/5) \operatorname{tg}(\theta/2)} = \\ &= \frac{1}{\cos(\pi/5) \operatorname{tg}(\theta/2)} = \sec(\pi/5) \operatorname{ctg}(\theta/2), \end{aligned}$$

откуда

$$\gamma = 2 \arctg(\sec(\pi/5) \operatorname{ctg}(\theta/2)) \approx 74.8^\circ. \quad (3)$$

Параметры ориентации. В данной работе ориентация каждой C_i -й камеры задается аналогично ориентации наблюдателя с помощью вектора $\mathbf{v}_i$ взгляда и вектора $\mathbf{u}_i$ “вверх” в системе WCS. Чтобы определить эти векторы, пронумеруем вершины додекаэдра, как показано на рис. 1, и запишем через них последовательность H пентагонов: 0 – {4, 7, 10, 13, 0}, 1 – {0, 1, 2, 3, 4}, 2 – {4, 3, 5, 6, 7}, 3 – {7, 6, 8, 9, 10}, 4 – {10, 9, 11, 12, 13}, 5 – {13, 12, 14,

1, 0}, 6 – {12, 11, 18, 19, 14}, 7 – {1, 14, 19, 15, 2}, 8 – {3, 2, 15, 16, 5}, 9 – {6, 5, 16, 17, 8}, 10 – {9, 8, 17, 18, 11}, 11 – {18, 17, 16, 15, 19}. Вектор $\mathbf{v}_i$ направлен в центр i -го пентагона, а вектор $\mathbf{u}_i$ – из этого центра в некоторую, например, вторую, вершину i -го пентагона (порядковый номер этой вершины фиксируется для всех пентагонов).

Рассмотрим вычисление координат векторов $\mathbf{v}_i$ и $\mathbf{u}_i$. Для этого вначале определим связь левой системы координат наблюдателя (View Coordinate System, VCS) [23] с рассматриваемым додекаэдром. Начало системы VCS находится в центре додекаэдра, ось X_{vcs} проходит через середину ребра {6, 8}, ось Y_{vcs} – через середину ребра {15, 16} и совпадает с вектором $\mathbf{u}$ “вверх” наблюдателя, а ось Z_{vcs} – через середину ребра {3, 4} и совпадает с вектором $\mathbf{v}$ взгляда наблюдателя (см. рис. 3).

Обозначим через $P_0, \dots, P_{19}$ координаты вершин додекаэдра в системе VCS и найдем их значения. Для этого будем использовать “золотые прямоугольники” [25] – три взаимно-перпендикулярных прямоугольника (см. рис. 3), стороны которых связаны “золотым отношением” Φ (числом Фидия) [26]:

$$\Phi = (1 + \sqrt{5})/2 = 2 \sin(3\pi/10) = 2 \cos(\pi/5), \quad (4)$$

при котором большая сторона относится к меньшей так же, как их сумма относится к большей из них. Восемь вершин додекаэдра будут вершинами вписанного куба размера 2, а остальные двенадцать – вершинами “золотых прямоугольников”. Выпишем на основе рис. 3 табл. 1 координат $P_0, \dots, P_{19}$.

Обозначим через K_i координаты центра i -го пентагона, а через $\mathbf{v}_{vcs, i}$ и $\mathbf{u}_{vcs, i}$ – координаты вектора взгляда и вектора “вверх” камеры C_i в системе VCS. Запишем их выражения через координаты $P_0, \dots, P_{19}$:

$$K_i = \frac{1}{5} \sum_{j=0}^4 P_{n_{i,j}}, \quad \mathbf{v}_{vcs, i} = K_i, \quad \mathbf{u}_{vcs, i} = P_{n_{i,1}} - K_i, \quad (5)$$

Таблица 1. Координаты вершин додекаэдра в системе VCS

Координаты	Значения			
P_0, P_2, P_5, P_7	(-1 -1 1)	(-1 1 1)	(1 1 1)	(1 -1 1)
$P_9, P_{12}, P_{17}, P_{19}$	(1 -1 -1)	(-1 -1 -1)	(1 1 -1)	(-1 1 -1)
P_1, P_6, P_8, P_{14}	(-Φ 0 1/Φ)	(Φ 0 1/Φ)	(Φ 0 -1/Φ)	(-Φ 0 -1/Φ)
P_3, P_4, P_{11}, P_{18}	(0 1/Φ Φ)	(0 -1/Φ Φ)	(0 -1/Φ -Φ)	(0, 1/Φ -Φ)
$P_{10}, P_{13}, P_{15}, P_{16}$	(1/Φ -Φ 0)	(-1/Φ -Φ 0)	(-1/Φ Φ 0)	(1/Φ Φ 0)

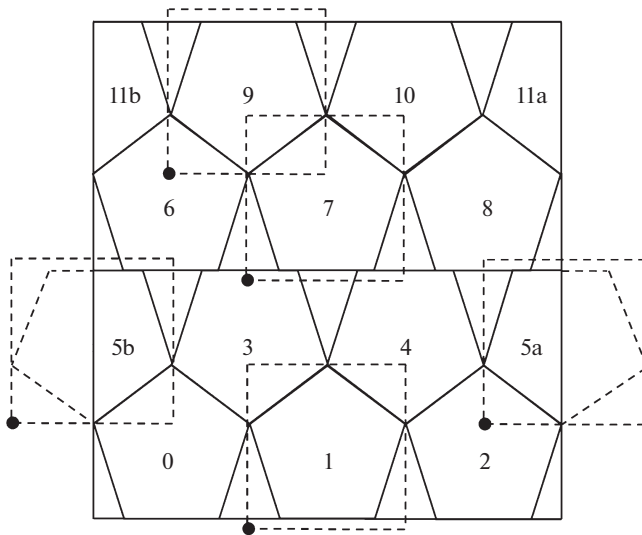


Рис. 4. Схема упаковки пентагонов в кадр 360-видео.

где $n_{i,j}$ — номер j -й вершины i -го пентагона из последовательности H . Обозначим через M_v верхнюю левую 3×3 -подматрицу матрицы преобразования координат из системы WCS в систему VCS [23]. Тогда координаты искомых векторов $\mathbf{v}_i$ и $\mathbf{u}_i$ в системе WCS можно вычислить как

$$\mathbf{v}_i = M_v^{-1} \mathbf{v}_{vcs,i}, \quad \mathbf{u}_i = M_v^{-1} \mathbf{u}_{vcs,i}. \quad (6)$$

Полученные в результате описанного этапа параметры проекции ($aspect$ и γ) и параметры ориентации ($\mathbf{v}_i$ и $\mathbf{u}_i$) мы устанавливаем для каждой виртуальной камеры C_i с помощью операторов `gluPerspective` и `gluLookAt` библиотеки OpenGL [27].

3.2. Метод синтеза кадра 360-видео

Синтез кадра 360-видео в данной работе предлагается выполнять на основе разработанной оригинальной схемы упаковки, состоящей из *прямых* и *перевернутых* пентагонов (см. рис. 4) — “снимков” виртуального пространства, полученных с помощью камер додекаэдра. Отметим, что рендеринг виртуального пространства мы выполняем в `rgba`-текстуру, исходно заполненную непрозрачным белым цветом, а перевернутые “снимки” получаем путем инвертирования знака вектора “вверх” (из формулы (6)) соответствующей камеры. На рис. 4 номерами обозначен порядок обхода камер додекаэдра, а пунктиром и черными кружками — ряд показательных случаев расположения их областей вывода. Как можно видеть, в предложенной схеме присутствуют разбитые пополам перевернутые пентагоны (5 и 11). Это достигается путем сдвига левого нижнего угла области вывода таким образом, чтобы нужная половина попадала в кадр (текстуру), а другая ее часть отсекалась границами кадра. На примере пентагонов 7 и 9 видно, что содержащие их области вывода будут перекрывать друг друга. Чтобы перевернутые “снимки” не затирали прямые, мы используем специальные корректирующие маски (см. рис. 5), состоящие из треугольников непрозрачного белого цвета (областей кадра, не несущих информации для пользователя) и треугольников непрозрачного черного цвета, представляющих собой места пересечения прямых и перевернутых “снимков”. Синтез прямого “снимка” в текстуре кадра реализует следующий алгоритм:

1. Рендерим треугольники 1–4 (см. рис. 5a) непрозрачного белого цвета.

2. Если номер снимка из отрезка [6, 8], то:

блокируем запись в `rgb`-каналы (с помощью оператора `glColorMask(0, 0, 0, 1)`),

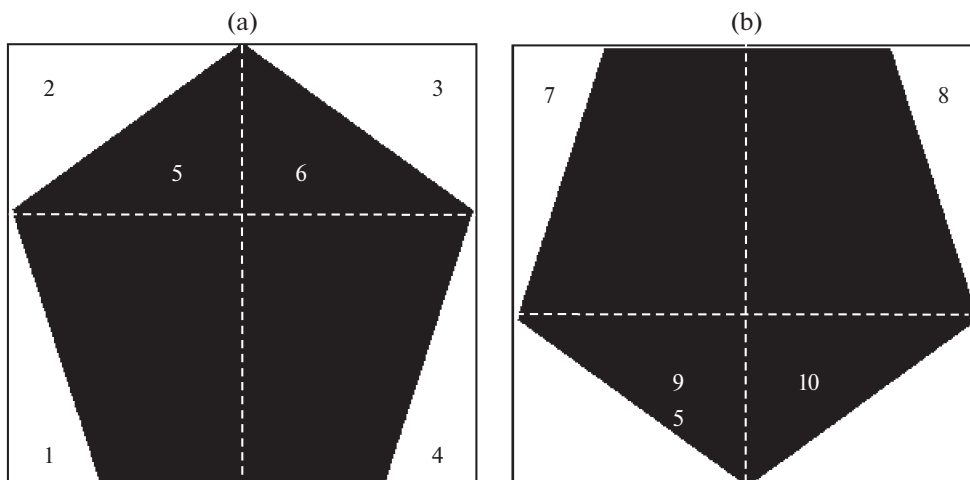


Рис. 5. Корректирующие маски для прямого (a) и перевернутого (b) пентагона.

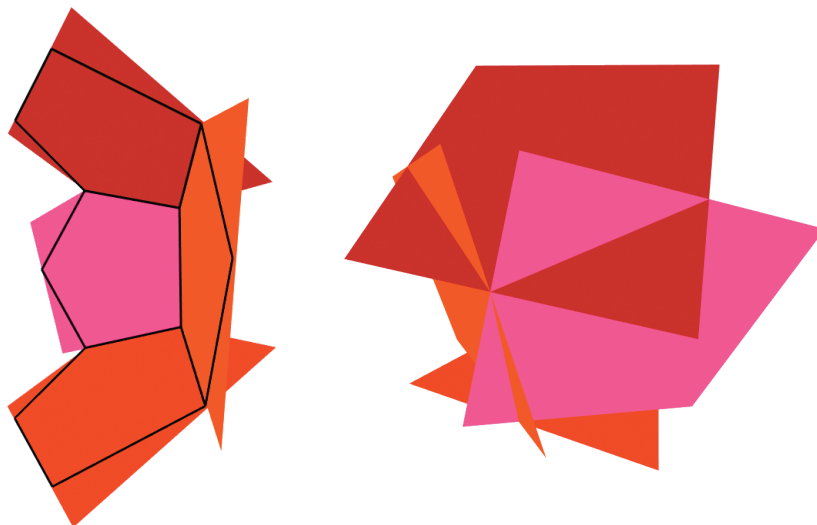


Рис. 6. Принцип реконструкции виртуальной панорамы.

рендерим в нижнюю часть области вывода прямоугольную полосу непрозрачного черного цвета, которая по высоте соответствует расстоянию h (см. рис. 2)

разрешаем запись в rgba-каналы.

3. Задаем режим альфа-смешивания `glBlendFunc(GL_ONE_MINUS_DST_ALPHA, GL_ONE)` [27].

4. Рендерим виртуальное пространство через соответствующую камеру додекаэдра.

5. Отключаем альфа-смешивание.

6. Блокируем запись в rgb-каналы.

7. Рендерим треугольники 5 и 6 (см. рис. 5a) непрозрачного черного цвета.

8. Разрешаем запись в rgba-каналы.

Конец алгоритма.

Синтез перевернутого “снимка” в текстуре кадра реализует следующий алгоритм

1. Рендерим треугольники 7, 8 непрозрачного белого цвета и треугольники 9, 10 непрозрачного черного цвета (см. рис. 5b).

2. Задаем режим альфа-смешивания `glBlendFunc(GL_ONE_MINUS_DST_ALPHA, GL_ONE)`.

3. Рендерим виртуальное пространство через соответствующую камеру додекаэдра.

4. Отключаем альфа-смешивание.

5. Если номер снимка из отрезка $[3, 5]$, то:

Разрешаем запись в rgba-каналы.

Рендерим в верхнюю часть области вывода прямоугольную полосу непрозрачного черного цвета, которая по высоте соответствует расстоянию h (см. рис. 2)

Конец алгоритма.

Отметим, что в приведенных алгоритмах рендеринг корректирующих треугольников и полосок выполняется в ортографической проекции. В результате выполнения для каждой камеры додекаэдра соответствующего ей алгоритма получим в rgba-текстуре упакованный кадр 360-видео.

3.3. Метод реконструкции виртуальной панорамы

Реконструкция виртуальной панорамы в данной работе выполняется в отдельной виртуальной сцене (*сцене плеера*). Оси мировой системы WCS этой сцены совпадают с осями системы VCS из раздела 3.1, за исключением оси $Z_{wcs} = -Z_{vcs}$. В начало системы WCS поместим зрителя, вокруг которого будем визуализировать 12 прямоугольников, имитирующих внутреннюю поверхность додекаэдра, как показано на рис. 6. Из рисунка видно, что на данном этапе добавка h , введенная в разделе 3.1, будет не видима, поэтому ее целесообразно не использовать. Все прямоугольники пронумерованы аналогично пентагонам воображаемого додекаэдра и жестко связаны с системой WCS — так же, как и додекаэдр с системой VCS: ось X_{wcs} проходит через середину ребра, образуемого пересечением 3-го и 9-го прямоугольника и т.д. Чтобы построить непрерывную виртуальную панораму, выполним следующие шаги.

Шаг 1. Вычисление координат вершин прямоугольников в системе WCS. Обозначим через A'_i, B'_i, C'_i, D'_i — координаты вершин i -го прямоугольника в системе WCS, а через $M'_i, U'_i, S'_i, T'_i, Q'_i$ и O'_i — координаты вершин и центра i -го пентагона, вписанного в $A'_i B'_i C'_i D'_i$, как показано на рис. 7. Координаты вершин пентагона можно легко найти из табл. 1, ин-

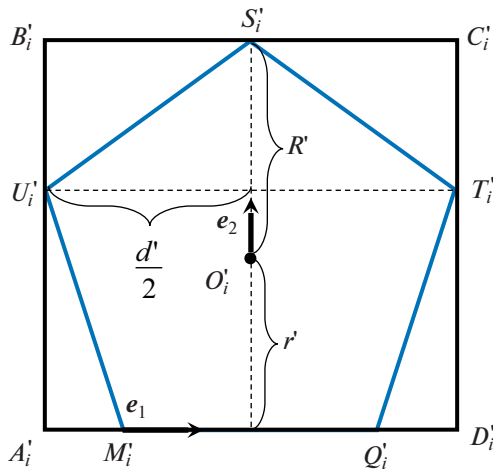


Рис. 7. Координаты вершин прямоугольника, имитирующего пентагон внутренней поверхности додекаэдра.

вертировав знак координаты z , а координаты центра — с помощью первой формулы из (5). Введем орты $\mathbf{e}_1 = \mathbf{M}'_i\mathbf{Q}'_i/|\mathbf{M}'_i\mathbf{Q}'_i|$ и $\mathbf{e}_2 = \mathbf{O}'_i\mathbf{S}'_i/|\mathbf{O}'_i\mathbf{S}'_i|$. Тогда искомые координаты вершин прямоугольника $A'_iB'_iC'_iD'_i$ можно вычислить как

$$\begin{aligned} B'_i &= S'_i - \frac{d'}{2}\mathbf{e}_1, & C'_i &= S'_i + \frac{d'}{2}\mathbf{e}_1, \\ A'_i &= B'_i - (R' + r')\mathbf{e}_2, & D'_i &= C'_i - (R' + r')\mathbf{e}_2, \end{aligned} \quad (7)$$

где d' , R' и r' — диагональ, радиусы описанной и вписанной окружности пентагона додекаэдра из рис. 3, вычисленные аналогично d , R и r из (1).

Таблица 2. Текстульные координаты вершин прямоугольников

Номера прямоугольников	Текстульные координаты			
	$\{s_0, t_0\}$	$\{s_1, t_1\}$	$\{s_2, t_2\}$	$\{s_3, t_3\}$
0, 1, 2	(0, 1)	(1, 1)	(1, 0)	(0, 0)
3, 4, 5, 11	(1, 0)	(0, 0)	(0, 1)	(1, 1)
6, 7, 8	(1, 1)	(0, 1)	(0, 0)	(1, 0)
9, 10	(0, 0)	(1, 0)	(1, 1)	(0, 1)

Шаг 2. Задание текстульных координат вершин прямоугольников. Обозначим через $\{s_0, t_0\}$, $\{s_1, t_1\}$, $\{s_2, t_2\}$ и $\{s_3, t_3\}$ текстульные координаты вершин A'_i , B'_i , C'_i , D'_i . Значения этих текстульных координат мы задаем с помощью табл. 2, учитывающей, прямой или перевернутый был “снимок” C_i -й камеры.

Шаг 3. Визуализация текстурированных прямоугольников. Создадим массив T из 12 текстурированных объектов, в которые будут считываться текстуры $w_i \times h_i$ пикселей из кадра 360-видео, где w_i , h_i — ширина и высота области вывода C_i -й камеры без добавки h . Обозначим через V_{coords} двумерный массив из 14 пар (10 целых пентагонов и 4 половины) координат (x, y) (в пикселах) левых нижних углов областей вывода камер додекаэдра, вычисленных на этапе синтеза кадра 360-видео. Визуализацию текстурированных прямоугольников реализует следующий алгоритм

1. Инициализируем $indx = 0$ — индекс пары координат в массиве V_{coords} .
2. Цикл по i от 0 до 11, где i — номер прямоугольника.
 Зададим ширину $frmW$ кадра видео 360, число $V_{coords}[indx][1]$ пропускаемых строк и число $V_{coords}[indx][0]$ пропускаемых пикселей в строке.
 Если $i > 5$, то $indx = (i + 1)$, в противном случае $indx = i$.
`glPixelStorei(GL_UNPACK_ROW_LENGTH, frmW).`
`glPixelStorei(GL_UNPACK_SKIP_ROWS, V_{coords}[indx][1]).`
`glPixelStorei(GL_UNPACK_SKIP_PIXELS, V_{coords}[indx][0]).`
 Загрузим текстуру в $T[i]$ -й текстурированный объект (с помощью оператора `glTexSubImage2D`).
 Если $(i == 5) \parallel (i == 11)$, // номера пентагонов, складывающихся из двух половин, то:
`glPixelStorei(GL_UNPACK_SKIP_ROWS, V_{coords}[indx+1][1]).`
`glPixelStorei(GL_UNPACK_SKIP_PIXELS, V_{coords}[indx+1][0] + w_i/2).`
 Загрузим текстуру в $T[i]$ -й текстурированный объект.
 Конец Если.
 Отообразим i -й прямоугольник с наложенным $T[i]$ -м текстурированным объектом.
 Конец цикла.

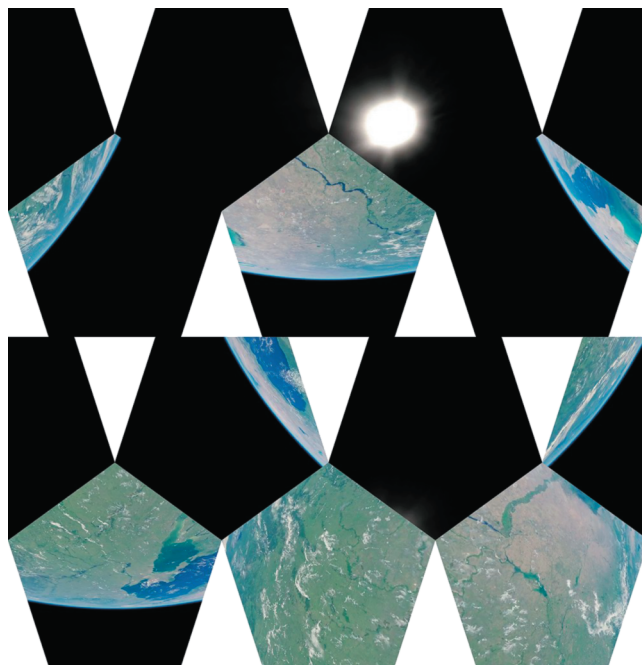


Рис. 8. Пример синтезированного кадра 360-видео с упаковкой пентагонов.

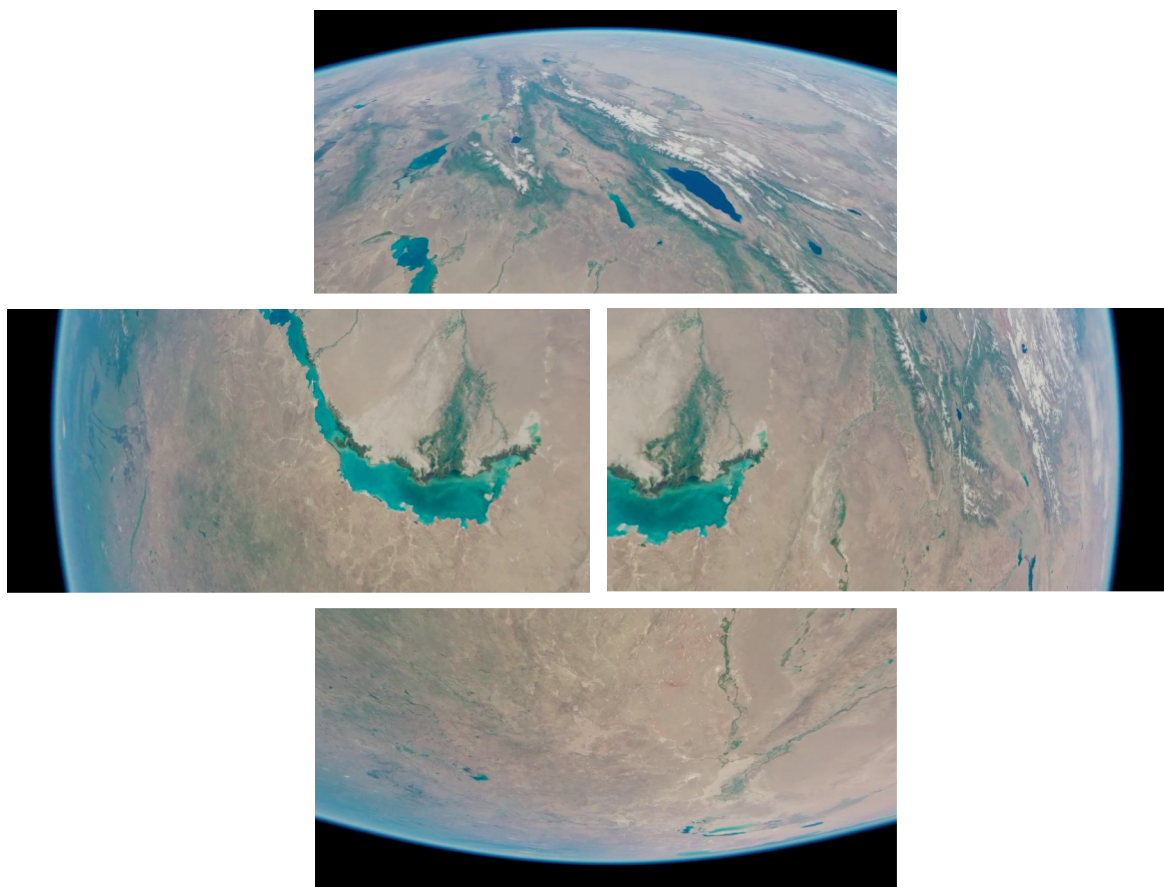


Рис. 9. Примеры полученных виртуальных панорам Земли

После выполнения трех шагов для каждого из 12 прямоугольников, имитирующих внутреннюю поверхность додекаэдра, перед зрителем в виртуальной сцене плеера будет сформирована непрерывная виртуальная панорама.

4. РЕЗУЛЬТАТЫ

Предложенные технология и методы были реализованы в программном комплексе, осуществляющем синтеза 360-видео на основе правильного додекаэдра в системах виртуального окружения. Созданный комплекс написан на языке C++ с применением графической библиотеки OpenGL и набора библиотек FFmpeg по кодированию/декодированию цифровых видеопоследовательностей. В состав предложенного программного решения входят модуль захвата 360-кадров, встраиваемый в систему визуализации, а также проигрыватель созданных 360-видео.

Разработанное решение было протестировано в подсистеме комплекса VirSim [1, 28], осуществляющей моделирование и визуализацию полета над Землей по орбите Международной космической станции (МКС). Было создано 360-видео полета вдоль участка трассы 3-го суточного витка МКС, проходящего через озеро Балхаш. Запись 360-видео выполнялась с частотой 25 кадров/с на основе стандарта H.264 сжатия видео и контейнера MP4. Размеры кадра 360-видео составили 3000×3077 пикселей. На рисунке 8 показан пример синтезированного кадра с реализованной схемой упаковки пентагонов. При воспроизведении 360-видео зритель может менять направление взгляда на 360 градусов по горизонтали и вертикали, а также увеличивать/уменьшать изображение. На рис. 9 показаны полученные виртуальные панорамы Земли для различных направлений взгляда.

Было выполнено сравнение качества проекций виртуального окружения, получаемых на основе правильного додекаэдра и куба. Мы оценивали изменение числа пикселей, которое может занимать проекция одного и того же отрезка на текстуре грани (додекаэдра и куба). Наибольшее число $r_{\max}$ пикселей такой отрезок будет занимать в области вершины грани, а наименьшее $r_{\min}$ — в центре грани. При этом у додекаэдра отношение $r_{\min}/r_{\max}$ составляет около 0.6, а у куба — только около 0.3 (в идеале, у сферы такое отношение равно 1). Таким образом использование проекции на правильный додекаэдр позволяет уменьшить до 2 раз потери детализации по сравнению с кубической проекцией. Также было проведено количественное сравнение уменьшения площади не-

информативных участков кадра с нашим предыдущим исследованием [22]:

$$\sigma_1 = 1 - \frac{S_{penta}}{S_{quad}} =$$

$$= 1 - \frac{10}{8\operatorname{tg}(36)\sin(54)(\operatorname{cosec}(36) + \operatorname{ctg}(36))} \approx 0.31, \quad (8)$$

$$\sigma_2 = \frac{S_{emptiness}}{S_{frame}} =$$

$$= \frac{\operatorname{tg}(36)}{\operatorname{cosec}(36) + \operatorname{ctg}(36) + 2\sin(72)} \approx 0.15, \quad (9)$$

где σ_1, σ_2 — доли неинформативных областей кадра 360-видео для исследования [22] и нашей текущей реализации, соответственно; S_{penta} — площадь пентагона; S_{quad} — площадь области вывода, содержащей пентагон (без добавки h); $S_{emptiness}$ — площадь всех пустот между пентагонами для кадра из рисунка 4, и S_{frame} — площадь кадра из рисунка 4.

Оценка показала, что за счет реализованной схемы упаковки пентагонов удалось уменьшить почти в два раза площадь неинформативных участков, что позволяет создавать более детализированные 360-видео при том же расходе битрейта видеопотока.

5. ЗАКЛЮЧЕНИЕ

В данной работе предложены решения, которые относятся к направлению исследования и разработки новых технологий и методов синтеза 360-видео в системах виртуального окружения, основанных на проекции виртуального пространства на грани выпуклых многогранников. Широко распространенная в настоящее время кубическая проекция обладает рядом недостатков, ограничивающих ее применение в системах виртуального окружения. В данном исследовании предлагается новая технология, основанная на проекции виртуального пространства на грани правильного додекаэдра. Технология включает в себя этап вычисления параметров проекции и ориентации камер додекаэдра, этап синтеза кадра 360-видео и этап реконструкции виртуальной панорамы. Для реализации этих этапов предложены методы и алгоритмы, основанные на оригинальной схеме упаковки “снимков” виртуального пространства. Созданные технология и методы были реализованы в программном комплексе и апробированы в подсистеме комплекса VirSim, осуществляющей моделирование и визуализацию

полета над Землей по орбите МКС. Апробация подтвердила эффективность и адекватность полученных результатов поставленной задаче. Полученные решения могут быть применены при развитии существующих и создании новых перспективных систем виртуального окружения, в видеотренажерах, системах научной визуализации, виртуальных лабораториях, образовательных приложениях и др.

Публикация выполнена в рамках государственного задания по проведению фундаментальных научных исследований (ГП 14) по теме (проекту) “34.9. Системы виртуального окружения: технологии, методы и алгоритмы математического моделирования и визуализации” (0580-2021-0012).

СПИСОК ЛИТЕРАТУРЫ

1. Михайлюк М.В., Мальцев А.В., Тимохин П.Ю., Страшнов Е.В., Крючков Б.И., Усов В.М. Система виртуального окружения Virsim для имитационно-тренажерных комплексов подготовки космонавтов // Пилотируемые полеты в космос, 2020. № 4 (37). С. 72–95.
2. Barladian B.K., Shapiro L.Z., Mallachiev K.A., Khoroshilov A.V., Solodelov Yu.A., Voloboy A.G., Galaktionov V.A., Koverninskii I.V. Visualization Component for the Aircraft Real-Time Operating System JetOS // Programming and Computer Software. 2020. V. 46. № 3. P. 167–175.
3. Lubimov M., Vidiger D., Shergin D. Limitless dynamic landscapes of 1 mm per pixel density, zooming included // SIGGRAPH 2020. 2020. Article № 6. <https://doi.org/10.1145/3407662.3407753>
4. Piedra J.A., Ojeda-Castelo J.J., Quero-Valenzuela F., Piedra-Fdez I. Virtual Environment for the Training of the Hands in Minimally Invasive Thoracic Surgery // IEEE 2016 8th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES), 2016. P. 1–4.
5. Страшнов Е.В., Мироненко И.Н., Финагин Л.А. Моделирование режимов полета квадрокоптера в системах виртуального окружения // Информационные технологии и вычислительные системы, 2020. № 1. С. 85–94.
6. Timokhin P.Yu., Mikhaylyuk M.V., Vozhegov E.M., Panteley K.D. Technology and Methods for Deferred Synthesis of 4K Stereo Clips for Complex Dynamic Virtual Scenes // Programming and Computer Software. 2021. V. 47. № 1. P. 67–75.
7. Andreev S., Bondarev A., Bondarenko A., Galaktionov V., Bondareva N. Constructing Stereo Images of Error Surfaces in Problems of Numerical Methods Verification // Proceedings of the 30th International Conference on Computer Graphics and Vision, 2020. V. 2744.
8. Andreev S.V., Bondarev A.E., Galaktionov V.A., Bondareva N.A. The problems of stereo animations construction on modern stereo devices // Scientific Visualization. 2018. № 10 (4). P. 40–52.
9. Corbillon X., Simon G., Devlic A., Chakareski J. Viewport-Adaptive Navigable 360-Degree Video Delivery // IEEE 2017 International Conference on Communications (ICC), 2017. P. 1–7.
10. Jiang H., He G., Yu W., Wang Z., Li Y. Scalable Video Coding Based on the User's View for Real-Time Virtual Reality Applications. In: Advances in Multimedia Information Processing – PCM 2017, LNCS 10736. Ed. by B. Zeng Springer, 2018. P. 766–775.
11. Yu M., Lakshman H., Girod B. A Framework to Evaluate Omnidirectional Video Coding Schemes // IEEE 2015 International Symposium on Mixed and Augmented Reality, 2015. P. 31–36.
12. Ray B., Jung J., Larabi M.-C. A Low-Complexity Video Encoder for Equirectangular Projected 360 Video Content // IEEE 2018 International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018. P. 1723–1727.
13. Li J., Wen Z., Li S., Zhao Y., Guo B., Wen J. Novel tile segmentation scheme for omnidirectional video // IEEE 2016 International Conference on Image Processing (ICIP), 2016. P. 370–374.
14. Тимохин П.Ю., Михайлюк М.В., Вожегов Е.М. Эффективные методы и алгоритмы синтеза видео 360 градусов на основе кубической проекции виртуального окружения // Труды ИСП РАН. 2020. Т. 32. № 4. С. 73–88.
15. Brown C. Bringing pixels front and center in VR video // Google AR and VR. March 14, 2017. <https://www.blog.google/products/google-ar-vr/bringing-pixels-front-and-center-vr-video/>.
16. Kuzyakov E., Liu S., Pio D. Optimizing 360 Video for Oculus. Facebook F8 developers conference, 2016. <https://developers.facebook.com/videos/f8-2016/optimizing-360-video-for-oculus/>.
17. Chen Z., Wang X., Zhou Y., Zou L., Jiang J. Content-Aware Cubemap Projection for Panoramic Image via Deep Q-Learning // MultiMedia Modeling. Lecture Notes in Computer Science. 2020. V. 11962. P. 304–315.
18. Fu C.-W., Wan L., Wong T.-T., Leung C.-S. The Rhombic Dodecahedron Map: An Efficient Scheme for Encoding Panoramic Video // IEEE Transactions on Multimedia. 2009. V. 11. № 4. P. 634–644.
19. Kuzyakov E., Pio D. Next-generation video encoding techniques for 360 video and VR. <https://engineering.fb.com/2016/01/21/virtual-reality/next-generation-video-encoding-techniques-for-360-video-and-vr/>.
20. McCutchen D. A dodecahedral approach to immersive imaging and display // Computer Graphics. 1997. P. 35–37.
21. McCutchen D. Immersive dodecahedral video viewing system. US Patent, No. 5703604. Dec. 30, 1997.
22. Timokhin P., Mikhaylyuk M., Panteley K. A technology to synthesize 360-degree video based on regular dodecahedron in virtual environment systems // Proceed-

- ings of the 30th International Conference on Computer Graphics and Vision. 2020. V. 2744.
<https://doi.org/10.51130/graphicon-2020-2-3-1>.
23. Хилл Ф. OpenGL. Программирование компьютерной графики. 2-е издание. Питер, 2002.
24. Zwillinger D. CRC Standard Mathematical Tables and Formulas. 33rd ed. CRC Press, 2018.
25. Vasilieva V.N. Golden Section and Golden Rectangles When Building Icosahedron, Dodecahedron and Archimedean Solids Based On Them // Geometry & Graphics. 2019. V. 7. № 2. P. 47–55.
26. Akhtaruzzaman M., Shafie A.A. Geometrical Substantiation of Phi, the Golden Ratio and the Baroque of Nature, Architecture, Design and Engineering // International Journal of Arts. 2011. V. 1. № 1. P. 1–22.
27. Segal M., Akeley K. The OpenGL Graphics System: A Specification. Version 4.6, Core Profile. The Khronos Group Inc., 2006–2018. <https://www.khronos.org/registry/OpenGL/specs/gl/glspec46.core.pdf>.
28. Михайлюк М.В., Тимохин П.Ю., Мальцев А.В. Метод тесселяции на GPU рельефа Земли для космических видеотренажеров // Программирование. 2017. № 4. С. 39–47.