

DOI: 10.15514/ISPRAS-2019-31(4)-4

Технология и методы отложенного синтеза 4К-стереороликов для сложных динамических виртуальных сцен

П.Ю. Тимохин, ORCID: 0000-0002-0718-1436 <webpismo@yahoo.de>

М.В. Михайлюк, ORCID: 0000-0002-7793-080X <mix@niisi.ras.ru>

Е.М. Вожегов, ORCID: 0000-0003-2676-1206 <vozhegovem@icloud.com>

К.Д. Пантелей, ORCID: 0000-0001-9281-2396 <kpanteley@mail.ru>

ФГУ «ФНЦ Научно-исследовательский институт системных исследований РАН»,
117218, Россия, г. Москва, Нахимовский просп., 36, к.1

Аннотация. В статье рассматривается задача записи управляемой исследователем стереовизуализации сложной динамической виртуальной сцены в видеопоследовательность стереопар (стереоролика) сверхвысокого разрешения. Предлагается технология отложенного синтеза стереороликов, которая позволяет создавать такие стереоролики, не нарушая масштаб реального времени визуализации. Технология включает в себя построение в масштабе реального времени сценария процесса визуализации и офлайн-преобразование сценария в стереоролик. В работе рассматриваются методы реализации этих этапов на примере задачи стереовизуализации изоповерхности насыщенности вытесняющей жидкости. В исследовании предлагается разработанный оригинальный файловый формат «scg» сценария визуализации, основанный на чанковой структуре данных, который реализует компактное представление соседних одинаковых кадров. Преобразование файла сценария в последовательность 4К-стереопар выполняется с помощью технологии внеэкранный рендеринг виртуальной сцены, а добавление стереопар в стереоролик – с помощью набора открытых библиотек FFmpeg обработки цифровых видеозаписей. В основе стереоролика используется медиаконтейнер MP4 и стандарт H.264 сжатия видео (оба являются частями международного стандарта MPEG-4). Предложенные технология и методы отложенного синтеза 4К-стереороликов реализованы в программном комплексе визуализации результатов моделирования неустойчивого вытеснения нефти из пористых сред. С помощью данного программного комплекса был синтезирован 4К-стереоролик, иллюстрирующий процесс изменения изоповерхности насыщенности вытесняющей жидкости на стадии развития процесса неустойчивого вытеснения нефти. Проведенная апробация подтвердила адекватность созданных решений поставленной задаче. Разработанные решения могут быть использованы в виртуальных лабораториях, при построении систем виртуального окружения, систем научной визуализации, в образовательных приложениях и др.

Ключевые слова: стерео; визуализация; виртуальная сцена; сценарий; кодирование видео; захват видео; реальное время; внеэкранный рендеринг; 4К

Для цитирования: Тимохин П.Ю., Михайлюк М.В., Вожегов Е.М., Пантелей К.Д. Технология и методы отложенного синтеза 4К-стереороликов для сложных динамических виртуальных сцен. Труды ИСП РАН, том 31, вып. 4, 2019 г., стр. 61-72. DOI: 10.15514/ISPRAS-2019-31(4)-4

Благодарности: Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 16-29-15099-офи_м.

Technology and methods for deferred synthesis of 4K stereo clips for complex dynamic virtual scenes

P.Yu. Timokhin, ORCID: 0000-0002-0718-1436 <webpismo@yahoo.de>

M.V. Mikhaylyuk, ORCID: 0000-0002-7793-080X <mix@niisi.ras.ru>

E.M. Vozhegov, ORCID: 0000-0003-2676-1206 <vozhegovem@icloud.com>

K.D. Panteley, ORCID: 0000-0001-9281-2396 <kpanteley@mail.ru>

Federal State Institution «Scientific Research Institute for System Analysis» of RAS,
36, build. 1, Nakhimovskiy Avenue, Moscow, 117218, Russia

Abstract. The paper considers the task of capturing controlled by a researcher stereo visualization of a complex dynamic virtual scene into a stereopair videosequence (stereoclip) of ultrahigh resolution. An efficient technology of deferred synthesis of stereoclips is proposed. It allows to create stereoclips without violating a real-time visualization. The technology includes real-time constructing of scenario of visualization process and offline-transforming the scenario to stereoclip. In the paper, methods to realize these stages are considered for the task of stereovisualization of saturation isosurface of displacing liquid. For this, original file format «scg» of visualization scenario is developed, based on «chunk» data structures. The format developed provides a compact representation of neighboring repeated frames. Transforming scenario file to a sequence of 4K-stereopairs is carried out by means of an offscreen rendering of virtual scene, and adding stereopairs to a stereoclip is performed using a number of open-source FFmpeg libraries designed for processing digital video content. For video recording media container MP4 and video compressing standard H.264 are used. Proposed technologies and methods of 4K-stereoclips deferred synthesis are implemented in a program complex for visualization of simulation results of unstable oil displacement from porous media. By means of the program complex a 4K-stereoclip is created, which illustrates the evolution of the isosurface during the process of unstable oil displacement. The approbation results confirmed the adequacy of the proposed solution to the task. Developed solutions can be used in virtual laboratories, in constructing of virtual environment systems and scientific visualization systems, in educational applications etc.

Keywords: stereo; visualization; virtual scene; scenario; video encoding; video capture; real-time; offscreen rendering; 4K

For citation: 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. Trudy ISP RAN/Proc. ISP RAS, vol. 31, issue 4, 2019. pp. 61-72 (in Russian). DOI: 10.15514/ISPRAS-2019-31(4)-4

Acknowledgements. The reported study was funded by RFBR, project number 16-29-15099.

1. Введение

В настоящее время во многих научных экспериментах востребована трехмерная визуализация сложных динамических виртуальных сцен [1, 2] в масштабе реального времени (с частотой смены кадров не менее 25 раз в секунду). Это особенно актуально в тех областях, где исследование реального объекта сопряжено с его разрушением, а цена нового образца крайне высока. Примером является нефтегазовая сфера, в частности, виртуальные эксперименты по неустойчивому вытеснению нефти из пористых сред [3-5].

Значительно повысить информативность и качество виртуальных экспериментов позволяет стереовизуализация виртуальных сцен [6] в сверхвысоком разрешении (4K, Ultra HD). Выполнение такой визуализации в реальном времени является сложной вычислительной задачей, для решения которой необходимы видеокарты high-end класса и специализированное программное обеспечение, что препятствует обмену полученными результатами в научном сообществе. Эффективным выходом является запись процесса визуализации виртуальной сцены в виде видеопоследовательности из стереопар (стереоролика). Стереороликами можно легко обмениваться между исследователями и воспроизводить их с помощью программ-стереоплееров [7] на персональных компьютерах и мобильных устройствах. Проблема состоит в том, что при использовании внешних программ

видеозахвата [8] визуализация каждого кадра фактически приостанавливается на некоторое время, необходимое для считывания и кодирования изображения, что приводит к нарушению режима реального времени синтеза изображений и появлению рывков в видеоролике. Особенно это заметно при выполнении исследователем различных управляющих воздействий на процесс визуализации – вращения и приближении виртуальной сцены, переключении между ракурсами и др. С целью уменьшения временных затрат на захват кадров активно развиваются аппаратные кодировщики видео (NVIDIA NVENC [9]), в которых алгоритмы сжатия распараллеливаются на многоядерных графических процессорах (GPU). По сравнению с кодированием видео на центральном процессоре (CPU) это дает существенный прирост скорости в задачах сжатия готовых видеопоследовательностей [10], однако, в задачах визуализации сложных динамических виртуальных сцен, где интенсивно используется GPU [4], наблюдается эффект взаимного торможения процессов кодирования и визуализации. В этой связи возникает задача разработки технологий синтеза стереороликов *in situ*, т.е. непосредственно в системе визуализации, основанных на захвате первичной информации (динамических параметров визуализации виртуальной сцены), объем которой значительно меньше, чем объем вторичных данных (синтезированных изображений).

В данной работе предлагается технология *отложенного* синтеза 4К-стереороликов, основанная на построении в масштабе реального времени сценария управляемой исследователем визуализации виртуальной сцены и преобразовании сценария в стереоролик в офлайн режиме. Предлагаемое решение реализуется на языке C++ с использованием открытого комплекса Qt средств разработки приложений, графической библиотеки OpenGL и набора открытых библиотек FFmpeg обработки цифровых видеозаписей.

2. Технология отложенного синтеза стереороликов

Пусть имеется некоторая система визуализации, которая может выполнять в масштабе реального времени рендеринг виртуальных сцен в моно и стерео режимах (горизонтальная стереопара, side-by-side [6]). Исходно в систему загружена необходимая виртуальная сцена и установлен моно режим. Предлагаемая технология синтеза стереороликов включает в себя два этапа: а) захват параметров визуализации и б) непосредственно синтез 4К-стереоролика.

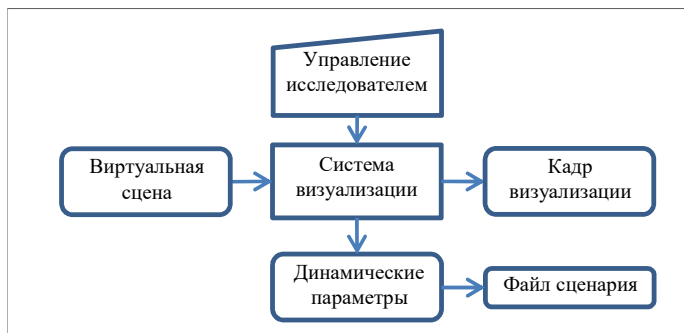


Рис. 1. Схема построения сценария стереоролика
Fig.1. The scheme of stereoclip scenario construction

На **первом этапе** из системы визуализации в масштабе реального времени считывается набор *динамических параметров* визуализации виртуальной сцены (см. рис. 1). По каждому такому набору можно однозначно восстановить состояние виртуальной сцены на момент синтеза соответствующего кадра. Совокупность всех захваченных неповторяющихся подряд наборов динамических параметров в данной работе называется *сценарием* визуализации. По окончании этапа захвата формируется файл с записанным в него сценарием.

На **втором этапе** (синтеза 4К-стереоролика) в системе визуализации выполняется специальное покадровое воспроизведение файла сценария, при котором в отдельном созданном внеэкранном (невидимом) буфере кадра выполняется синтез 4К-стереопары, а в экранный (видимый) буфер кадра помещается ее уменьшенная копия, вписанная по размерам в главное окно визуализации (см. рис. 2). Каждая синтезированная 4К-стереопара считывается из внеэкранного буфера, кодируется и добавляется в файл стереоролика, а уменьшенная копия выводится на экран для визуального контроля процесса синтеза стереоролика. Данный процесс может выполняться уже не в масштабе реального времени в зависимости от выбранного кодировщика видео и вычислительной мощности CPU и GPU компьютера.

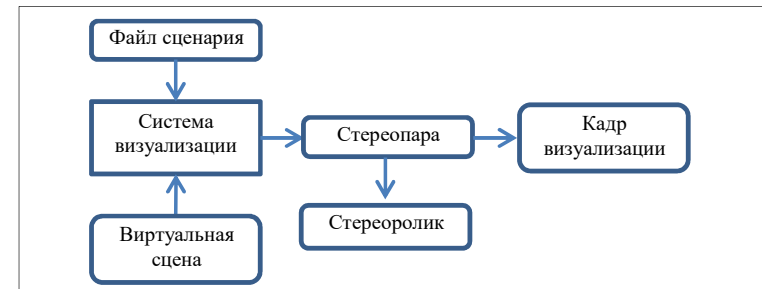


Рис. 2. Схема преобразования сценария в стереоролик
Fig. 2. The scheme of transforming scenario to a stereoclip

Далее рассмотрим методы реализации описанных этапов на примере задачи визуализации результатов неустойчивого вытеснения нефти из пористых сред [6], в частности, визуализации поверхности постоянного значения поля насыщенности вытесняющей жидкости (*изоповерхности*).

2.1 Метод построения сценария визуализации

В рассматриваемой задаче параметрами процесса визуализации являются:

- положение и ориентация модели изоповерхности;
- параметры виртуальной камеры;
- параметры источника освещения;
- параметры материала модели изоповерхности;
- постоянное значение поля насыщенности (изоуровень);
- номер визуализируемого шага моделирования.

Для захвата этих параметров в данной работе разработан формат файла сценария визуализации (*scr-файл*), который включает в себя заголовок (с общей информацией) и блок данных со значениями параметров визуализации (рис. 3). Заголовок scr-файла содержит следующие данные:

- идентификатор *scrId* = 0x00726373 («scr» в кодировке ASCII);
- число *numFrames* кадров в сценарии;
- интервал *queryInterval* опроса значений параметров визуализации, в мс;
- размер *scrSize* блока данных *scr*-файла, в байтах.

В блоке данных *scr-файла* хранятся значения параметров визуализации только для неповторяющихся подряд кадров. Если кадр повторяется несколько раз, то параметры визуализации записываются только для первого кадра, а последующие повторные кадры просто подсчитываются. Для этого используется структура «пакет кадра», включающая в себя:

- размер *blockSize* блока данных пакета, в байтах;
- число *numRepeats* повторов подряд текущего кадра;
- блок данных пакета со значениями параметров визуализации.

Блок данных пакета состоит из «чанков». «Чанк» – это структура данных, которая содержит идентификатор *id* объекта, размер *size* (в байтах) и поле *data* значений параметров визуализации. В поле *data* «чанка» мы записываем значения параметров визуализации (одного или группы параметров, объединенных по смыслу). Использование таких структур данных дает возможность эффективно добавлять в *scr-формат* новые параметры визуализации, модифицировать и удалять их, сохраняя обратную совместимость между форматами (если система визуализации не может распознать какой-то идентификатор «чанка», то его поле *data* просто пропускается).

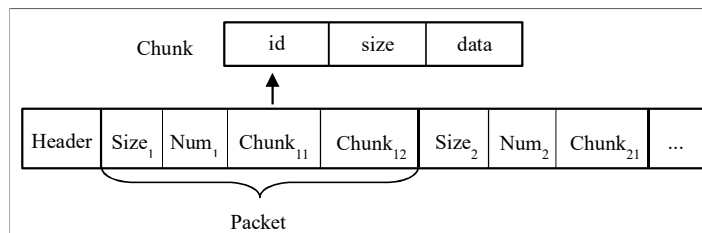


Рис. 3. Структура *scr-файла*
Fig. 3. *Scr-file structure*

Для рассматриваемой задачи реализованы следующие типы «чанковых» структур:

- *SMVChunk* – модельно-видовой матрицы, задающей положение и ориентацию модели изоповерхности в системе координат камеры (*id* = 0, *size* = 64, *data* – 16 элементов модельно-видовой матрицы);
- *SCamChunk* – камеры (*id* = 1, *size* = 16, *data* – вертикальный угол раствора камеры, отношение ширины кадра к высоте (*аснект*), расстояния до ближней и дальней плоскостей отсечения);
- *SLightChunk* – направленного источника освещения (*id* = 2, *size* = 64, *data* – направление источника освещения, интенсивности фоновой, диффузной и зеркальной составляющих освещения);
- *SMaterialChunk* – материала модели изоповерхности (*id* = 3, *size* = 52, *data* – цвета фоновой, диффузной и зеркальной составляющих материала, а также коэффициент блеска);
- *SIsoLevelChunk* – изоуровня (*id* = 4, *size* = 4, *data* – постоянное значение скалярного поля насыщенности вытесняющей жидкости);
- *SStepChunk* – визуализируемого шага моделирования (*id* = 5, *size* = 4, *data* – номер визуализируемого шага моделирования).

Захват параметров визуализации осуществляется с некоторой частотой v_1 опроса (в размах в секунду), где $25 < v_1 \leq v_2$, а v_2 – наименьшая частота синтеза изображений виртуальной сцены, которая зависит от вычислительной мощности видеокарты. Во время каждого опроса считываются значения всех параметров визуализации и сравниваются со значениями предыдущего опроса. Если текущее значение какого-то параметра визуализации повторяется,

то оно не записывается в файл сценария. На рисунке 4 изображена схема опроса на примере модельно-видовой матрицы модели изоповерхности, которая рассчитывается для каждого кадра на GPU. Отметим, что в более сложных многообъектных виртуальных сценах для сокращения размера файла сценария положения и ориентации виртуальных объектов можно записывать в виде кватернионов.

В данной работе опрос значений параметров визуализации реализуется с помощью таймера (класс *QTimer*) из библиотеки Qt. Сигнал периодического срабатывания таймера связывается с разработанной функцией формирования пакета с помощью оператора *connect*. Получаемые в результате работы этой функции пакеты кадров мы будем накапливать в динамический массив *V* байтов.

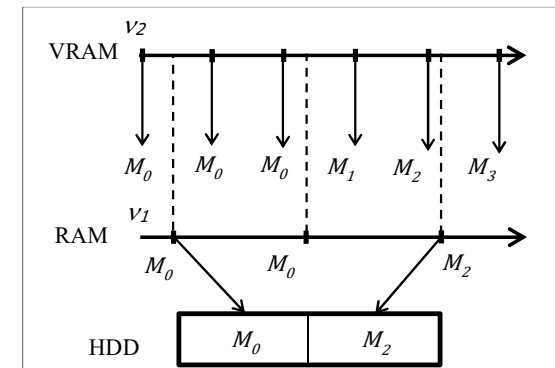


Рис. 4. Опрос модельно-видовой матрицы модели изоповерхности
Fig. 4. *Querying of modelview matrix of isosurface model*

Построение сценария визуализации реализует Алгоритм 1.

1. Запишем: *scrSize* = 0, *numFrames* = 0, *blockSize* = 0, *numRepeats* = 0, *PACKET_HEADER* = *sizeof(blockSize)* + *sizeof(numRepeats)*, где *sizeof* возвращает размер переменной в байтах.
Инициализируем флаг нового неповторяющегося кадра: *gotFrame* = *false*.
2. Создадим «чанки» *SMVChunk*, *SCamChunk*, *SLightChunk*, *SMaterialChunk*, *SIsoLevelChunk*, *SStepChunk* с пустыми полями *data*.
3. Запустим таймер опроса с помощью функции *start()*.
4. Цикл опроса с интервалом *queryInterval*
blockSize = 0.
Считаем значение *isoLevel* изоуровня из системы визуализации.
Если *isoLevel* ≠ значению *data* «чанка» *SIsoLevelChunk*, то:
 Если *gotFrame* ≠ *true*
 Добавим в массив *V* *blockSize* и *cntFrame*.
 gotFrame = *true*.
 Запишем *isoLevel* в поле *data* «чанка» *SIsoLevelChunk*.
 Добавим «чанк» *SIsoLevelChunk* в массив *V*.
 blockSize = *blockSize* + *sizeof(SIsoLevelChunk)*.
Проверим значения остальных параметров визуализации аналогично *isoLevel*.
gotFrame = *false*.
Если *blockSize* ≠ 0, то:
 numRepeats = 0.
 Обновим в массиве *V* значения *blockSize* и *numRepeats* текущего пакета.
 scrSize = *scrSize* + *PACKET_HEADER* + *blockSize*.
В противном случае:
 numRepeats = *numRepeats* + 1.

Обновим в массиве V значение $numRepeats$.
 $numFrames = numFrames + 1$.
Конец цикла.
5. Остановим таймер опроса с помощью функции $stop()$.
6. Создадим scr-файл.
7. Запишем в scr-файл $scrId$, $numFrames$, $queryInterval$, $scrSize$ и массив V .
8. Закроем файл.

Алгоритм 1. Построение сценария визуализации
Algorithm 1. Construction of the visualization scenario

Чтобы просмотреть захваченный процесс визуализации, в данной работе также создан режим воспроизведения файла сценария. Как и в Алгоритме 1, данный режим реализуется с помощью таймера. При этом сигнал периодического срабатывания таймера связывается с разработанной функцией проигрывания сценария. Предварительно мы загружаем все пакеты из файла сценария в байтовый массив K размера $scrSize$ (из заголовка scr-файла). Воспроизведение сценария реализует Алгоритм 2.

1. Инициализируем следующие переменные:
 $gotFrame = false$, // флаг нового неповторяющегося кадра.
 $packetsCnt = 0$, // счетчик считанных байт из блока данных scr-файла.
 $frameCnt = 0$, // счетчик проигранных повторных пакетов (кадров).
 $curSize = 0$, // число считанных байт из блока данных пакета.
 $CHUNK_HEADER = sizeof(id) + sizeof(size)$. // размер заголовка «чанка», в байтах.
2. Запустим таймер проигрывателя с помощью функции $start()$.
3. Цикл проигрывания с интервалом $queryInterval$
3.1. Если $packetsCnt == scrSize$, то:
Выйдем из цикла проигрывания с помощью функции $stop()$.
3.2. Если $gotFrame \neq true$, то:
Считаем $blockSize$ и $numRepeats$ текущего пакета из массива K .
 $curSize = 0$.
Цикл по «чанкам» в пакете, пока $curSize \neq blockSize$.
Считаем id и $size$ текущего чанка.
Считаем из поля $data$ текущего «чанка» значение параметра визуализации согласно его id .
Обновим в системе визуализации значение считанного параметра.
 $curSize = curSize + CHUNK_HEADER + size$.
Конец цикла.
 $packetsCnt = packetsCnt + PACKET_HEADER + blockSize$.
Если $numRepeats \neq 0$, то $gotFrame = true$;
В противном случае:
 $frameCnt = frameCnt + 1$.
Если $frameCnt == numRepeats$, то $gotFrame = false$, $frameCnt = 0$.
3.3. Визуализируем сцену с обновленными параметрами.
Конец цикла.

Алгоритм 2. Воспроизведение сценария визуализации
Algorithm 2. Playback of the visualization scenario

2.2 Метод преобразования сценария в 4К-стереоролик

Для создания 4К-стереоролика необходимы эффективные медиаконтейнер и алгоритм кодирования видео, поддерживающие работу с большими объемами видеоданных (около 16 млн. пикселей на кадр). В данном исследовании предлагается использовать контейнер MP4, являющийся частью международного стандарта MPEG-4 (MPEG-4 Part 14, ISO/IEC 14496-14:2003) сжатия цифрового аудио и видео. В таблице 1 приведено сравнение MP4 с распространенными контейнерами AVI (Microsoft), MOV (Apple) и MKV (Open Source). Из

таблицы видно, что MP4 превосходит по функционалу контейнеры AVI и MKV, в частности за счет поддержки В-кадров (они позволяют значительно сжимать размер кодируемой видеопоследовательности). Другим важным преимуществом является поддержка покadroвого редактирования видео (технология Edit-in-place), что позволяет изменять отдельные участки видео (например, разрезать или склеивать) без повторного сжатия всей видеопоследовательности. Контейнер MOV близок к MP4 по функциональности, однако ориентирован в первую очередь на использование в операционных системах компании Apple, в то время как MP4 является отраслевым стандартом и имеет более широкую поддержку. Также немаловажной отличительной чертой контейнера MP4 является поддержка онлайн-трансляции.

Таблица 1. Сравнение видео контейнеров
Table 1. Comparing of video containers

Контейнер	Поддержка кодека H.264	Edit-in-place	Поддержка В-кадров
MP4	mp4 - официальный стандарт контейнера для кодека H.264	да	да
AVI	затруднена в связи с ограниченной поддержкой В-кадров	нет	отсутствует в исходном формате, реализуется «хакерским» путем
MOV	да	да	да
MKV	да	нет	да

Контейнер MP4 поддерживает ряд современных видеокодеков, из которых для решения рассматриваемой задачи был выбран кодек H.264. Данный кодек также является частью международного стандарта MPEG-4 (MPEG-4 Part 10) и, хотя по эффективности сжатия он уступает таким кодекам, как H.265/HEVC и AV1, является самым распространенным, так как уже прошел многолетний путь внедрения в отрасль. В течение этого периода стандарт постоянно модернизировался, получил аппаратное декодирование в большинстве плееров и в итоге хорошо зарекомендовал себя на практике. Особенностью видеокодека H.264 является работа с макроблоками (группами пикселей изображения) размерами от 16x16 до 4x4 [11], что накладывает соответствующие требования к ширине и высоте кодируемого видео. В рассматриваемой задаче 4К-стереопара имеет размеры, кратные 16 (7680x2160 пикселей). В данной работе запись стереопар в стереоролик реализуется с помощью набора открытых библиотек FFmpeg (Fast Forward MPEG) [12]. Из всего набора для решения рассматриваемой задачи используются следующие библиотеки: libavcodec (кодеры и декодеры видео и аудио), libavformat (мультиплексоры и демультиплексоры медиаконтейнеров), libswscale (функции масштабирования изображений и преобразования цветовых пространств и форматов пикселей) и libavutil (генераторы случайных чисел, математические и мультимедиа утилиты и др.).

Кодек H.264 имеет большое количество настроек, которые позволяют управлять качеством и скоростью кодирования видеопоследовательности. К ним относятся битрейт, количество кадров в каждой группе последовательных изображений потока (GOP), параметр сложности оценки движения, коэффициент компрессии видеопотока, профиль видеокодека и другие. В данной работе эти параметры задаются в структуре AVCodecContext из библиотеки FFmpeg. Чтобы начать процесс кодирования видео с помощью FFmpeg, необходимо выполнить ряд предварительных действий: задать формат выходного видео (AVOutputFormat), создать контекст ввода-вывода для записи видео (AVFormatContext), создать поток записи видео (AVStream), связать поток с кодировщиком H.264 (AVCodec), а также инициализировать структуры AVFrame, SwsContext, AVPicture для обработки видеок кадров и структуру AVPacket для добавления кадров в контейнер MP4. Более подробное описание реализации этих этапов можно найти в [12].

Процесс записи 4К-стереоролика включает в себя модифицированное проигрывание файла сценария (см. Алгоритм 2), в котором используется функция стереовизуализации виртуальной сцены в формате side-by-side [6]. Исходными данными, как и в Алгоритме 2, является массив K пакетов кадра, загруженный из файла сценария. Преобразование сценария в стереоролик реализует Алгоритм 3.

1. Создадим внеэкранный буфер F кадра размера $(2w_s) \times h_s$ пикселей, где w_s и h_s - ширина и высота 4К-кадра в моно режиме.
2. Создадим буфер I (типа $QImage$ из библиотеки Qt) размера $(2w_s) \times h_s$ пикселей для хранения стереопары в оперативной памяти.
3. Зададим прямоугольник $P_0P_1P_2P_3$, вписанный по ширине в главное окно визуализации размера $w_f \times h_f$:
Вычислим высоту h_i прямоугольника $P_0P_1P_2P_3$ в системе координат нормализованного объема видимости (Normalized Device Coordinate System, NDCS):
 $h_i = 2a_s a_f$, где $a_s = 2w_s / h_s$ - аспект 4К-стереопары, $a_f = w_f / h_f$ - аспект главного окна визуализации.
Запишем координаты вершин прямоугольника $P_0P_1P_2P_3$ (в системе NDCS координат нормализованного объема видимости):
 $P_0 = (-1, 0.5h_i), P_1 = (-1, -0.5h_i), P_2 = (1, 0.5h_i), P_3 = (1, -0.5h_i)$.
4. Откроем файл для записи видео с помощью функции `url_fopen` из FFmpeg.
5. Цикл по i от 1 до $numFrames$
Считаем i -ый пакет кадра из массива K и обновим в системе визуализации значения считанных параметров (см. пп. 3.1, 3.2 из Алгоритма 2).
Выполним синтез текстуры S с изображением стереопары:
Активируем внеэкранный буфер F кадра.
Визуализируем виртуальную сцену в стереорежиме side-by-side.
Деактивируем внеэкранный буфер F кадра.
Добавим стереопару в MP4-контейнер:
Выгрузим из видеопамати текстуру S в буфер I стереопары.
Конвертируем с помощью функции `sws_scale` из FFmpeg RGB-стереопару (буфер I) в YUV-изображение (структура $AVFrame$), где Y -яркость, а U и V - цветоразностные компоненты.
Закодируем YUV-изображение в байтовый массив данных H.264 с помощью функции `avcodec_encode_video` из FFmpeg.
Добавим массив данных H.264 в структуру отдельного сжатого кадра $AVPacket$ (контейнер MP4).
Отобразим стереопару в главном окне визуализации:
Установим область вывода с помощью функции `glViewport(0, 0, w_f, h_f)`.
Визуализируем прямоугольник $P_0P_1P_2P_3$ с наложенной текстурой S .
Конец цикла.
6. Закроем файл для записи видео с помощью функции `url_fclose`.

Алгоритм 3. Преобразование сценария в стереоролик
Algorithm 3. Transforming the scenario to a stereoclip

По окончании работы Алгоритма 3 формируется MP4-файл с результирующим 4К-стереороликом в базовом стереоформате side-by-side, который может быть воспроизведен на поляризационной стереоустановке с помощью программы-стереоплеера, например, Stereoscopic Player, а также преобразован (в этом же плеере) в ряд других популярных видов стерео (анаглифическое, чересстрочное и др.).

3. Результаты

Предложенные в данной работе технология и методы отложенного синтеза 4К-стереороликов были реализованы в программном комплексе визуализации результатов моделирования неустойчивого вытеснения нефти из пористых сред [13]. С помощью данного комплекса было выполнено исследование стадии развития неустойчивости вытеснения

нефти водой, в частности, изменения формы изоповерхности насыщенности вытесняющей жидкости. Исходными данными являлась последовательность из 65 трехмерных массивов значений насыщенности вытесняющей жидкости, полученных в результате пошагового моделирования процесса неустойчивого вытеснения нефти на расчетной сетке размера 100^3 ячеек. В процессе исследования для каждого из 65 шагов моделирования выполнялись построение и визуализация трехмерной полигональной модели изоповерхности. Исследование включало в себя динамическое изменение ориентации и масштаба модели изоповерхности, а также ее перестроение для различных постоянных значений насыщенности. Визуализация модели изоповерхности проводилась при разрешении Ultra HD 4K (3840x2160) на персональном компьютере с процессором Intel Core i7 950 3.06 ГГц, RAM 12Гб, видеокартой NVIDIA GeForce GTX 1080 Ti (VRAM 11 Гб, 3584 ядер). Средняя частота визуализации составила около 100 кадров в секунду.

Был выбран участок исследования длительностью около 2-х минут, для которого был записан сценарий визуализации с интервалом опроса равным 10 мс. Временные затраты на захват динамических параметров визуализации составили крайне малые значения (менее 1 мс на кадр) и не оказали заметного влияния на частоту визуализации. Размер полученного scf-файла сценария составил около 370 Кб. На основе созданного файла сценария был выполнен синтез 4К-стереоролика, демонстрирующего изменение формы изоповерхности вытесняющей жидкости. На рис. 5 изображен процесс синтеза 4К-стереоролика в системе визуализации. На рис. 6 изображено проигрывание полученного в результате 4К-стереоролика в Stereoscopic Player в режиме анаглифического стерео.

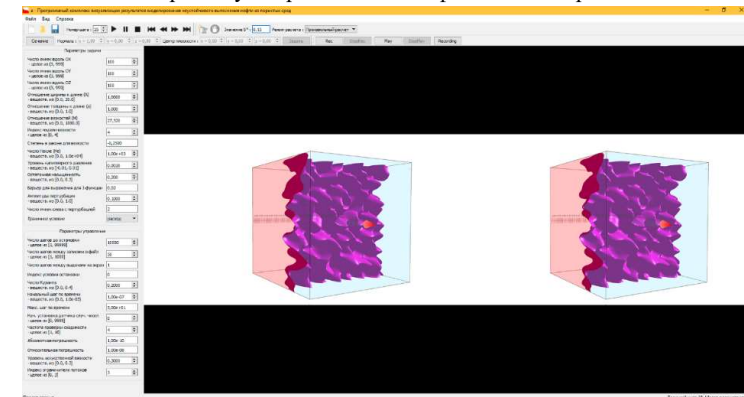


Рис. 5. Синтез 4К-стереоролика в системе визуализации
Fig. 5. Synthesis of 4K-stereoclip in a visualization system

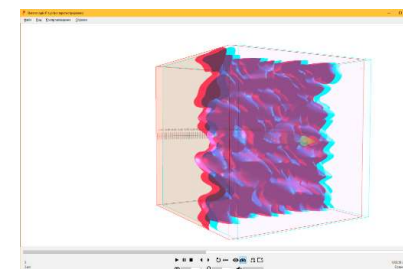


Рис. 6. Проигрывание синтезированного 4К-стереоролика в Stereoscopic Player
Fig. 6. Playing of synthesized 4K-stereoclip in Stereoscopic Player

4. Заключение

В статье рассмотрена задача записи стереовизуализации сложной динамической виртуальной сцены в 4К-стереоролик. Использование внешних программ видеозахвата является проблематичным, т.к. при визуализации каждого кадра заметная часть времени тратится на считывание и кодирование стереопары, что приводит к появлению рывков в стереоролике. В этой связи целесообразно выполнять синтез стереороликов непосредственно в системе визуализации (in situ) на основе захвата значений динамических параметров визуализации.

В данной работе предложена технология отложенного синтеза стереороликов, которая позволяет записывать 4К-стереоролики, не нарушая масштаб реального времени визуализации. Технология включает в себя этап захвата всех неповторяющихся подряд наборов динамических параметров процесса визуализации (сценария) и этап преобразования сценария в стереоролик, который может выполняться уже в офлайн режиме.

В статье предложены методы реализации этих этапов на примере задачи стереовизуализации изоповерхности насыщенности вытесняющей жидкости. Реализация включает в себя разработанный оригинальный формат «scg» файла сценария, основанный на «чанковых» структурах данных, которые дают возможность эффективно расширять и изменять формат, сохраняя обратную совместимость.

В работе описаны созданные эффективные алгоритмы построения и воспроизведения сценариев в системе визуализации, в которых повторяющиеся подряд пакеты значений параметров визуализации просто подсчитываются, а не записываются. Этап преобразования файла сценария в 4К-стереоролик реализуется с помощью технологии внеэкранного рендеринга виртуальной сцены и набора открытых библиотек FFmpeg обработки цифровых видеозаписей. В работе приведено сравнение распространенных медиаконтейнеров, в результате которого по ряду преимуществ выбирается контейнер MP4 и стандарт H.264 сжатия видео, а также описан алгоритм преобразования сценария в стереоролик, в котором реализован визуальный контроль процесса синтеза стереоролика.

Предложенное в статье решение было реализовано в системе визуализации результатов моделирования неустойчивого вытеснения нефти из пористых сред. Был синтезирован 4К-стереоролик, демонстрирующий изменение формы изоповерхности насыщенности вытесняющей жидкости на стадии развития процесса неустойчивого вытеснения нефти. Полученные результаты подтвердили адекватность предложенного решения поставленной задаче и его применимость для виртуальных лабораторий, систем виртуального окружения и научной визуализации, и др.

Список литературы / References

- [1]. Puzyrkov D.V., Podryga V.O., Polyakov S.V. Parallel processing and visualization for results of molecular simulation problems. *Trudy ISP RAN/Proc. ISP RAS*, vol. 28, issue 2, 2016, pp. 221-242. DOI: 10.15514/ISPRAS-2016-28(2)-15.
- [2]. В.И. Гонахьян. Алгоритм удаления невидимых поверхностей на основе программных проверок видимости. *Труды ИСП РАН*, том 30, вып. 2, 2018 г., стр. 81-98. DOI: 10.15514/ISPRAS-2018-30(2)-5 / Gonakhchyan V.I. Occlusion culling algorithm based on software visibility checks. *Trudy ISP RAN/Proc. ISP RAS*, vol. 30, issue. 2, 2018, pp. 81-98 (in Russian).
- [3]. V.F. Nikitin, L.I. Stamov, E.I. Skryleva, V.V. Tyurenkova, M.V. Mikhaylyuk. Computer visualization of fluid displacement instability in porous medium. In *Proc. of the 68th International Astronautical Congress (IAC 2017)*, 2017, pp. 1-5.
- [4]. Mikhaylyuk M., Timokhin P. Effective GPU-based section visualization in isosurface of saturation of displacing liquid in a porous medium. In *Proc. of the 2018 International Conference on Engineering Technologies and Computer Science (EnT 2018)*, 2018, pp. 57-60.
- [5]. И.В. Козлов, Е.И. Скрылева. Математическое моделирование и обработка эксперимента по вытеснению нефти водой из неомомских песчанников. *Вестник кибернетики*, 2016, № 2, стр. 138-

- 145 / Kozlov I.V., Skryleva E.I. Mathematical modeling and data processing of water-oil displacement in neocomian sandstone. *Proceedings in Cybernetics*, 2016, issue 2, pp. 138-145 (in Russian).
- [6]. М.В. Михайлюк, А.В. Мальцев, П.Ю. Тимохин. Методы стереовизуализации результатов моделирования неустойчивого вытеснения нефти из пористых сред. *Труды НИИСИ РАН*, том 8, № 2, 2018, стр. 125-129 / Mikhaylyuk M.V., Maltsev A.V., Timokhin P.Yu. The methods for 3D stereo visualization of data obtained in simulation of unstable oil displacement from porous media. *Trudy NIISI RAN/Proc. of SRISA RAS*, vol. 8, issue 2, 2018, pp. 125-129 (in Russian).
- [7]. Stereoscopic Player. Available at: <http://www.3dtv.at/>, accessed 25.06.2019.
- [8]. Open Broadcaster Software. Available at: <https://obsproject.com/>, accessed 25.06.2019.
- [9]. NVIDIA VIDEO CODEC SDK. Available at: <https://developer.nvidia.com/nvidia-video-codec-sdk>, accessed 25.06.2019.
- [10]. А.Г. Кушниренко, А.В. Мальцев, М.В. Михайлюк, А.А. Прилипко, П.Ю. Тимохин, М.А. Торгашев. Сжатие разделенных видео потоков в задачах дистанционного обучения. *Известия Академии инженерных наук им. А.М. Прохорова*, 2015, № 2, стр. 3-10 / Kushnirenko A.G., Maltsev A.V., Mikhaylyuk M.V., Prilipko A.A., Timokhin P.Yu., Torgashev M.A. The compression of separated video streams for distance education tasks. *News Academy of Engineering Sciences A.M. Prokhorov*, 2015, issue 2, pp. 3-10 (in Russian).
- [11]. Гук И. Особенности сжатия видеоданных по рекомендации H.264. Компоненты и технологии, 2006, № 2, стр. 1-10 / Guk I. Features of video data compression under H. 264/MPEG 4 part 10 recommendation. *Components & Technologies*, 2006, issue 2, pp. 1-10 (in Russian).
- [12]. FFmpeg. A complete, cross-platform solution to record, convert and stream audio and video. Available at: <https://ffmpeg.org/>, accessed 25.06.2019.
- [13]. Программный комплекс визуализации результатов моделирования неустойчивого вытеснения нефти из пористых сред (ПО «Визуализатор неустойчивого вытеснения»). РФ, № 2019614787, дата заявки 30.04.2019, дата опубликования 13.05.2019. Доступно по ссылке: <https://www1.fips.ru/register-web/action?acName=clickRegister®Name=EVM>, дата обращения: 25.06.2019 / Program complex for visualization of simulation results of unstable oil displacement («Vizualizator of unstable oil displacement»). Russian Federation, № 2019614787, request 30.04.2019, publication 13.05.2019 (in Russian). Available at: <https://www1.fips.ru/register-web/action?acName=clickRegister®Name=EVM>, accessed 25.06.2019.

Информация об авторах / Information about authors

Петр Юрьевич ТИМОХИН – старший научный сотрудник НИИСИ РАН. Сфера научных интересов: компьютерная графика, визуализация.

Petr Yurievich TIMOKHIN – Senior Researcher of SRISA RAS. Research interests: computer graphics, visualization.

Михаил Васильевич МИХАЙЛЮК – доктор физико-математических наук, профессор, главный научный сотрудник НИИСИ РАН. Сфера научных интересов: компьютерная графика, визуализация, системы виртуального окружения.

Mikhail Vasilievich MIKHAYLYUK – Doctor of Physico-Mathematical Sciences, Professor, Chief Researcher of SRISA RAS. Research interests: computer graphics, visualization, virtual environment systems.

Евгений Михайлович ВОЖЕГОВ – ведущий программист НИИСИ РАН. Сфера научных интересов: компьютерная графика.

Evgeniy Mikhaylovich VOZHEGOV – Leading programmer of SRISA RAS. Research interests: computer graphics.

Клим Денисович ПАНТЕЛЕЙ – ведущий программист НИИСИ РАН. Сфера научных интересов: численное моделирование.

Klim Denisovich PANTELEY – Leading programmer of SRISA RAS. Research interests: numerical simulation.