

УДК 004.921

ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ПРЕДТРЕНАЖЕРНОЙ И ТРЕНАЖЕРНОЙ ПОДГОТОВКИ ОПЕРАТОРОВ СЛОЖНЫХ ТЕХНИЧЕСКИХ СИСТЕМ

Работа выполняется при поддержке РФФИ, грант № 14-07-00020-а

*А.М. Гуацинтов, м.н.с.; К.А. Мамросенко, к.т.н., зав. отделом СИТ
(ЦВСИТ НИИСИ РАН, пр-т Нахимовский, 36, к. 1, г. Москва, 117218, Россия, kirillam@ya.ru)*

*В.Н. Решетников, д.ф.-м.н., профессор, зав. кафедрой КТ
(РГТУ им. К.Э. Циолковского, Берниковская наб., 14, г. Москва, 109240, Россия, kt-mati@mail.ru)*

Аннотация. Даются понятия тренажерно-обучающих систем, видов подготовки операторов. Приведены классификация, примеры использования тренажерно-обучающих систем. Показаны особенности моделирования в тренажерах поведения сложных технических систем.

Описаны новый метод воспроизведения разнородных видеоматериалов на гранях объектов виртуальной трехмерной сцены и архитектура декодера видеофайлов. Рассмотрены алгоритм синхронизации видео- и аудиоданных и методы загрузки декодированных видеокадров в память видеоадаптера. Определены ограничения, накладываемые на процесс одновременного воспроизведения видеофайлов в трехмерной виртуальной среде.

Ключевые слова: *тренажерно-обучающая система, ТОС, рендеринг, система визуализации, видеоматериалы, декодер видео, мультимедийные технологии.*

Под *тренажерно-обучающей системой* (ТОС) оператора *сложной технической системы* (СТС) будем понимать техническое средство для подготовки операторов, отвечающее требованиям методик подготовки, реализующее модель таких систем и обеспечивающее контроль за действиями обучаемого, а также для исследований [1].

Выделяются два вида подготовки: предтренажерная (теоретическая) и тренажерная. ТОС должна использоваться при проведении каждого из них. Теоретическая подготовка является базой, на которой в процессе тренажерной подготовки обучаемый формируется как специалист.

Для повышения эффективности подготовки на ТОС целесообразно использовать методы обучения посредством взаимодействия обучаемого с образовательными ресурсами, индивидуализированного обучения, а также методы, для которых характерно активное взаимодействие между всеми участниками процесса подготовки [2].

Для этапа теоретической подготовки ТОС должна обеспечить следующие возможности:

- получение фундаментальных знаний (для обучаемых с уровнем подготовки от 0 и выше);
- обсуждение с другими обучаемыми (в том числе и зарубежными) проблем, возникающих на этапах подготовки;
- работа с дополнительными источниками информации;
- проведение наблюдений;
- коррекция деятельности обучаемого со стороны преподавателя.

Пример представления информации в подсистеме предтренажерной подготовки показан на видео далее.

<http://youtu.be/3CDsMHzh-Co>



Использование тренажерных систем может давать ряд преимуществ по сравнению с обучением на реальных системах:

- безопасная отработка действий операторов при различных отказах;
- возможность отработки до 90 % выполняемых задач;
- возможность многократного повторения выполнения задач;
- гибкость учебного плана;
- снижение расхода ресурсов реальной системы;
- возможность работы тренажера до 23 часов в сутки;
- экологическая чистота.

Тренажер стыковки приведен на следующем видео [3].

<http://youtu.be/0eVBXK0vivA>



В зависимости от объема решаемых задач тренажеры могут быть комплексными, специализированными, процедурными.

Комплексные тренажеры предназначены для совместной подготовки операторов СТС в полном объеме алгоритмов их деятельности или одного оператора, деятельность которого в СТС осуществляется по нескольким специальностям.

Специализированные тренажеры обеспечивают формирование и отработку оператором навыков и умений по одному из видов деятельности либо обучение и подготовку одного оператора в объеме его функциональных обязанностей (при эксплуатации СТС двумя и более операторами).

Процедурные тренажеры обеспечивают формирование и отработку оператором навыков выполнения про-

цедур эксплуатации отдельных систем СТС и выполнения отдельных элементов различных видов деятельности. Данная классификация тренажерных систем обусловлена следующими обстоятельствами:

- стоимость самого комплексного тренажера СТС и стоимость эксплуатации тренажера (эта величина весьма велика);
- отработка отдельных частных процедур управления, задействующая малую часть функционала комплексного тренажера.

Таким образом, отработка отдельных частных процедур на комплексном тренажере, как правило, экономически невыгодна. Это и является предпосылкой к созданию специализированных и процедурных тренажеров.

Отдельно необходимо выделить программно-аппаратные комплексы для моделирования функционала новых СТС. Стоимость исправления ошибок на этапах испытаний опытного образца намного превышает стоимость исправления ошибок на этапе проектирования. Таким образом, основой для разработки новых СТС является моделирование функционирования СТС на этапах проектирования. [3, 4].

Занятия на тренажере для отработки некоторых групп навыков занимают меньше времени, чем отработка тех же навыков на реальной системе. Это обусловлено тем, что для приведения реальной системы в состояние S_1 , которое будет исходным для начала отработки навыка, необходимо выполнить некий алгоритм, затратив t_1 времени (причем t_1 может исчисляться часами, месяцами). В тренажере для перевода рассматриваемой системы в состояние, идентичное S_1 , необходимо только задать значения переменных параметров системы.

Для воссоздания ощущений реального мира при проведении тренировочных занятий необходимо воспроизвести все сигналы, получаемые оператором во время выполнения операций. При невозможности точного воспроизведения сигналов используются упрощенные модели рассматриваемых сигналов. Для ряда тренажеров некоторые группы сигналов могут не воспроизводиться ввиду их не востребуемости для данной тренировки, что приведет к снижению сложности тренажера, его удешевлению. Сигналы от тренажера должны быть согласованными, так как мозг человека для получения полной картины происходящего совмещает сигналы, получаемые различными органами чувств.

Для сигналов, получаемых посредством зрительных органов, в среднем принимается угол обзора $\pm 90^\circ$ по горизонтали, 50° и 70° по вертикали и угловое разрешение $1'$.

Информация, поступающая путем визуального наблюдения, воспринимается различно в зависимости от удаленности наблюдаемых объектов. Существует разделение на три зоны: до 1 метра, от 1 до 30 метров и свыше 30 метров от наблюдателя. В зоне, превышающей 30 метров, когда бинокулярное зрение ослабевает (оба глаза получают одинаковое изображение), для адекватного восприятия важны окклюзия и размеры.

Если экран помещен на расстоянии порядка 3 метров от обучаемого, а объект по сценарию тренировки удален от наблюдателя менее чем на 3 метра, то при использовании проекционных систем отображения информации может происходить рассогласование.

В тренажере передача зрительных сигналов о перемещении основана на том, что мозг человека с трудом различает изменения ситуаций (используя только зрительный канал), когда наблюдатель неподвижен, а окружающие объекты находятся в движении и когда движется наблюдатель, а объекты неподвижны. Поэтому генерируется движущаяся окружающая обстановка, создавая иллюзию перемещения наблюдателя (подвижная платформа способствует усилению восприятия движения).

Детализация воспроизведения окружающей обстановки напрямую связана с производительностью тренажера. Для повышения эффективности использования ресурсов тренажера необходимо определить, какие группы объектов наиболее важны для достижения целей тренировки, и следовательно, нужно детально воспроизводить именно эти группы [1].

ТОС, отвечающие требованиям методик подготовки персонала, как правило, содержат значительное количество информационных ресурсов. Одним из видов ресурсов, используемых при обучении, являются видеоматериалы.

Подсистема визуализации обеспечивает отображение результатов моделирования внешней среды и объекта управления с помощью устройств отображения информации. Отображение разнородных видеоматериалов в виртуальной трехмерной сцене – одно из требований к ТОС.

Воспроизведение видеоматериалов внутри виртуальной трехмерной сцены является сложной задачей, так как необходимо учитывать такие факторы, как производительность подсистемы визуализации, производительность видеокарты. Подсистема визуализации должна обеспечивать отображение трехмерной сцены с приемлемой частотой кадров (не менее 25 кадров в секунду) и при этом быть способной реагировать на внешние воздействия, в том числе на изменения параметров трехмерной сцены или загрузку дополнительных объектов. Для отображения нескольких видеоматериалов высокой четкости в виртуальной трехмерной сцене авторами была разработана и реализована архитектура декодера кадров видео, а для обеспечения требуемой производительности подсистемы визуализации при отображении видеоматериалов высокой четкости созданы методы эффективной передачи данных в видеопамять.

Для декодирования видеофайлов выбрана кросс-платформенная система FFmpeg, распространяемая по лицензии LGPL. Данная система является базовым средством декодирования видеоматериалов в операционных системах семейства Linux, а также используется в операционных системах Windows (в составе пакета FFDSHOW) и MacOS X. Основой системы FFmpeg является библиотека libavcodec, в которой хранятся различные кодеры и декодеры для работы с аудио- и видеoinформацией.

Структура видеофайла – контейнер, содержащий общую информацию (количество видео- и аудиопотоков, наличие субтитров, информация о разделах, метаданные – название, год создания, автор и т.д.) и сведения о различных потоках информации (видео, аудио, субтитры и т.д.). Некоторые контейнеры допускают использование только определенных кодеров для представления информации. Наиболее популярными контейнерами являются AVI, OGV, MP4, MKV, MOV, WMV.

В большинстве контейнеров данные представляются в виде пакетов (в некоторых форматах называемые chunks (куски) или segments (сегменты)). Пакеты хранят закодированную информацию определенного типа – видео, аудио, субтитры и т.д. В большинстве видеофайлов количество аудиопакетов значительно меньше количества видеопакетов: приблизительно 75 % видеопакетов и 25 % аудиопакетов. В зависимости от типа контейнера и используемого кодера расположение пакетов внутри контейнера может быть разным. Зачастую пакеты расположены в виде последовательности очередей пакетов разного типа, например, около 20 видеопакетов, расположенных подряд, затем несколько аудиопакетов, после которых снова подряд расположены около 20 видеопакетов.

Пакеты хранят в себе определенный сегмент аудио- или видеoinформации. В большинстве случаев в видеопакете хранится целый видеокادر, но в некоторых форматах (в частности, в формате MPEG 2) один видеокادر может состоять из нескольких пакетов. Это связано с тем, как кодер сохраняет информацию о видеокadre. Например, в формате MPEG 2 присутствуют три типа кадров: независимо сжатые кадры (I-кадры), кадры, сжатые с использованием предсказания движения в одном направлении (P-кадры), и кадры, сжатые с использованием предсказания движения в двух направлениях (B-кадры). Соответствующие группы кадров от

одного I-кадра до другого образуют GOP-Group Of Pictures-группу кадров.

Большинство форматов видеофайлов представляют видеоинформацию в цветовой модели YUV (Y – яркость, U и V – цветоразностные составляющие).

В пакетах, кроме закодированных данных и информации о размере и типе данных данного пакета, присутствует информация о времени, когда необходимо декодировать данный пакет (последовательности декодирования пакетов), и времени, когда декодированная информация должна быть отображена или воспроизведена.

Аудиоинформация видеофайла может быть закодирована с использованием различных форматов сжатия, таких как MP3, AAC, AC3, WMA и т.д. Для корректного воспроизведения аудиоинформации необходимо знать количество каналов (моно, стерео, 5.1, 7.1 и т.д.) и частоту дискретизации. После декодирования звукового пакета декодированные данные представляются в формате PCM (Pulse Code Modulation, импульсно-кодовая модуляция) и передаются звуковой карте для воспроизведения звука.

Для отображения видеоматериалов в подсистеме визуализации была выбрана кросс-платформенная система декодирования видеофайлов FFmpeg, распространяемая по лицензии LGPL. Пакет FFmpeg не предоставляет возможности для визуализации декодированных видеоизображений и воспроизведения звуковых данных. Для воспроизведения звуковых данных из декодируемого видеофайла был использован пакет SFML, распространяемый по лицензии zlib/png. Пакет SFML является кросс-платформенным и предоставляет возможности по работе с графикой, звуком, различными системами ввода информации и передачей данных по сети.

В качестве основы подсистемы визуализации ТОС был использован графический движок Horde3D, распространяемый по лицензии EPL. В Horde3D применена стратегия интеграции, отличная от графических движков OGRE 3D и Irrlicht. Данные графические движки предоставляют объектно-ориентированную библиотеку классов, с помощью которых пользователь может создать собственные реализации. В Horde3D используется более высокий уровень абстракции, чем в других движках, а основная функциональность доступна через небольшой процедурный интерфейс, в некоторых аспектах схожий с Microsoft Win32 API. Преимуществом простого C-интерфейса являются большая наглядность, простота изучения функциональности и особенностей движка, а также лучшая портируемость. Практически в любых языках программирования, в том числе и скриптовых, есть механизм доступа к функциям внешних библиотек, в то время как импорт классов намного сложнее или вовсе невозможен. Horde3D может быть использован в языках C#, Java, LUA, Python и др.

Из-за высокого уровня абстракции интерфейса прикладного программирования (API) пользователь не может добавлять новую функциональность, например новый тип узла сцены, без модификации исходных кодов движка. Для преодоления этого недостатка Horde3D предоставляет механизм расширений, который дает пользователю полный доступ к внутренней структуре движка. Расширение статически присоединяется к библиотеке движка и предоставляет свою функциональность через процедурный интерфейс [5].

Все потоки выполняются в адресном пространстве процесса. Выполняющийся процесс имеет как минимум один (главный) поток. Применение многопоточности наиболее эффективно при использовании многоядерных процессоров. На одноядерных процессорах многопоточное приложение тоже работает, но повышение производительности будет незначительным; возможно даже снижение производительности, если разные потоки одного процесса выполняют ресурсоемкие операции.

Хотя многопоточность способна существенно повысить производительность приложения, применять ее следует с осторожностью. Внедрение многопоточности существенно повышает сложность самого приложения и его отладки, а также при некорректном использовании может привести к таким проблемам, как состояние гонки (когда работа системы зависит от порядка выполнения частей кода), взаимная блокировка потоков (несколько потоков находятся в состоянии бесконечного ожидания ресурсов, занятых самими этими потоками), и к трудностям при обработке ошибок (исключений) внутри приложения.

Для оптимизации процесса декодирования видео используются отдельные потоки для сбора пакетов, декодирования видео- и аудиопакетов. Процесс воспроизведения звука также происходит в дополнительном потоке. Первоначально в качестве системы управления потоками был выбран пакет SFML, который предоставляет базовую функциональность управления многопоточностью на системах семейств Windows и Unix (Linux, MacOS). В целом система управления потоками SFML работала без сбоев, но было выявлено несколько недостатков. Система не позволяет выставить приоритет обработки создаваемому потоку, а также привязать создаваемый поток к конкретному процессорному ядру. Она предоставляет только один способ приостановки потока – сон, что делает невозможным точный прогноз времени, через которое потоку вновь будет выделено процессорное время. Например, если потоку дано указание «заснуть» на 5 мс, то нет гарантии,

что через 5 мс потоку будет выделено процессорное время, оно может быть выделено и через 15 мс. Данное обстоятельство зависит от степени загруженности процессора, реализации планировщика задач операционной системы и приоритета выполнения потока. С целью преодоления описанных выше недостатков пакетов процесс воспроизведения звука также происходит в дополнительном потоке. Для реализации многопоточности было принято решение интегрировать библиотеку управления потоками `thread` системы `boost`.

Система `boost` – это набор кросс-платформенных библиотек, расширяющих стандартные инструменты языка `C++` и распространяемых по лицензии `Boost Software License`. Часть библиотек являются кандидатами на включение в следующий стандарт языка `C++`.

Библиотека управления потоками `boost thread` обладает большей функциональностью, чем библиотека `SFML`, и предоставляет такие возможности, как управление потоками, созданными с помощью библиотеки `thread` и средствами операционной системы, использование условных переменных (`conditional variables`) и т.д. Использование средств операционной системы для управления созданным потоком позволяет реализовывать задание приоритета выполняемого потока, а также привязки потока к определенному ядру процессора. Применение условных переменных дает возможность приостанавливать поток при определенных условиях и в отличие от функции «сон», мгновенно восстанавливать работу потока.

При воспроизведении одного видео с разрешением вплоть до 1920 на 1080 (FullHD) в подсистеме визуализации либо нескольких видео с разрешением не выше 720 на 576 (576p, SD) время генерации одного кадра не превышает 40 мс (в зависимости от аппаратной платформы). Однако при одновременном воспроизведении нескольких видео высокой четкости наблюдается значительное увеличение времени генерации кадра подсистемой визуализации.

Для преодоления проблем с производительностью при одновременном воспроизведении нескольких видео высокой четкости было создано несколько методов. Первый метод состоял в том, чтобы копировать данные новых видеок кадров в пиксельные буферы в отдельном потоке, а затем передавать их в основном потоке в память видеокарты. Однако данный метод себя не оправдал, так как узким местом было не копирование информации в оперативной памяти, а передача данных в память видеокарты.

Второй метод основывался на использовании единого пиксельного буфера, который содержал бы кадры всех воспроизводимых видео. Преимуществом данного метода должно было стать то, что пиксельный буфер обновляется только один раз для всех воспроизводимых видео. Однако данный метод имел недостатки: необходимость сохранять предыдущие видеок кадры, чтобы избежать рассинхронизации видео и аудио, а также сложность реализации динамического расширения и уменьшения пиксельного буфера при добавлении или удалении видео в ходе работы приложения. Данный метод также не смог предоставить необходимой уровень производительности при обновлении нескольких видео.

Метод, позволяющий получить оптимальную производительность при воспроизведении нескольких видео высокой четкости, основан на применении приоритетов к обновлению воспроизводимых видеок файлов.

Для каждого видео применяется весовой коэффициент, отражающий общий приоритет видео и весовой коэффициент выполняемого действия. Общий приоритет видео основывается на разрешении видео – чем оно ниже, тем выше приоритет, так как видео меньшего размера обновляется быстрее. Существуют три общих приоритета видео – высокий (8), средний (4), низкий (0). Для видео с разрешением до 720 на 576 выставляется высокий приоритет, до 1280 на 720 – средний, до 1920 на 1080 – низкий. Весовой коэффициент выполняемой операции меняется каждый раз при обновлении видео в зависимости от действия, которое необходимо выполнить при следующей *итерации цикла* (ИЦ) генерации кадров подсистемы визуализации. Выполняемые операции имеют следующие весовые коэффициенты (обновление видеотекстуры имеет наибольший коэффициент): обновление видеотекстуры (3), обновление видеотекстуры и пиксельного буфера (2), обновление пиксельного буфера (1), исключение видео (0). Операция исключения видео выполняется в случае, когда на следующей ИЦ подсистемы визуализации нет необходимости обновлять видеотекстуру, а также все пиксельные буферы заполнены. Видео исключается из списка обновляемых, однако весовой коэффициент выполняемого действия меняется на высший для того, чтобы состояние видео было проверено на следующей ИЦ подсистемы визуализации.

Список обновляемых видео определяется на каждой ИЦ подсистемы визуализации. На основе общего приоритета видео и весового коэффициента выполняемого действия определяется позиция видео в списке обновления.

Алгоритм обновления видеотекстуры в подсистеме визуализации был также изменен для повышения производительности. Теперь первым производится обновление видеотекстуры, а не обновление пиксельного буфера. Так как операция обновления текстуры из пиксельного буфера является асинхронной, то она выполняется очень быстро, в то время как операция обновления информации в пиксельном буфере может вызвать

синхронизацию процессора и видеокарты, что приведет к задержкам. Также видеокarte после обновления пиксельного буфера может понадобиться некоторое время для обработки полученных данных. Поэтому при обновлении текстуры сразу после обновления пиксельного буфера может возникнуть задержка. Изменение алгоритма обновления видеотекстуры дает больше времени на обработку данных пиксельного буфера. После обновления пиксельного буфера одновременно с обработкой данных в пиксельном буфере видеокartой происходят расчет весового коэффициента действия для следующей ИЦ подсистемы визуализации и переход к обновлению следующей видеотекстуры.

Таким образом, под тренажерно-обучающей системой оператора СТС будем понимать техническое средство для подготовки операторов СТС, отвечающее требованиям методик подготовки, реализующее модель таких систем и обеспечивающее контроль за действиями обучаемого, а также для исследований [1]. ТОС, отвечающие требованиям методик подготовки персонала, как правило, содержат значительное количество информационных ресурсов. Одним из видов ресурсов, используемых при обучении, являются видеоматериалы.

Для реализации воспроизведения видеоматериалов в подсистеме визуализации ТОС были разработаны методы и алгоритмы для одновременного воспроизведения нескольких видео, в том числе высокой четкости, произведена оптимизация алгоритмов для достижения требуемого (не более 20 мс) времени генерации кадров в подсистеме визуализации при воспроизведении видеоинформации, разработана подсистема визуализации ТОС, внедрен программный модуль для декодирования видео, позволяющий обрабатывать видеофайлы на различных программно-аппаратных платформах.

Литература

1. Allerton D. Principles of flight simulation. Chichester U.K.: Wiley, 2009.
2. Васильев А.В. Дистанционное обучение как возможность расширения образовательного пространства в системе подготовки космонавтов. Звездный городок: РИО НИИ ЦПК им. Ю.А. Гагарина, 2009. С. 44–46.
3. К.А. Mamrosenko, V.N. Reshetnikov. Algorithms and methods in allocated training systems, Proceedings International conference on scientific research in open and distance education. Hanoi, Vietnam: The Gioi, 2008. Pp. 31–36.
4. Решетников В.Н., Мамросенко К.А. Методика и алгоритмы визуализации для обучающих модулей компьютерных тренажерно-обучающих систем: тр. Междунар. науч. конф., посвященной 80-летию со дня рожд. акад. В.А. Мельникова. М.: Первая Исслед. Лаборатория им. акад. В.А. Мельникова, 2009. С. 76–80.
5. Nicolas Schulz. Horde3D Documentation [Электронный ресурс]. URL: <http://www.horde3d.org/docs/manual.html> (дата обращения: 18.05.2011).
6. В.Н. Решетников. Космические телекоммуникации. Системы спутниковой связи и навигации. СПб: Ленинградское изд-во, 2010. 134 с.