

РАЗДЕЛ V. СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

```
nco_out=nco_mod-((0x20000000-  
(nco_in>>2))>>1); // nco_mod-(1/4-nco_in/4)/2  
nco_mod=(nco_out&0x80000000)+nco_out-  
(nco_out&0x40000000); // mod om 0.5
```

```
if (nco_out<=0)  
{  
    overflow = 0x7FFFFFFF;  
} else overflow=0;  
//----- end of NCO -----//
```

Выводы

В результате комплексного подхода к оптимизации программы ЦСП был достигнут выигрыш по среднему времени выполнения более чем в два раза по сравнению с неоптимизированной программой. Благодаря использованию машинно-зависимых средств (intrinsics) компилятора языка C для семейства ЦСП TMS320VC55XX ЦОС программа сохранила высокий уровень переносимости на процессор с другой системой команд. При переносе следует заменить эти средства на соответствующие другому компилятору языка C, а в случае их отсутствия подключить виде внешних функций. Дальнейшая оптимизация программы может быть достигнута путем частичного или полного перевода программы на язык ассемблера, однако при этом переносимость будет полностью потеряна.

СПИСОК ЛИТЕРАТУРЫ

1. Касьянов, В.Н. Оптимизирующие преобразования программ./ В.Н. Касьянов– М. : Наука, 1988.
2. Сергиенко, А.Б. Цифровая обработка сигналов./ А.Б. Сергиенко. – СПб.: Питер, 2003.
3. TMS320C55x optimizing C/C++ compiler : User's guide, SPRU281F, 2003. [Электрон.ресурсы]. URL: <http://focus.ti.com/lit/ug/spru281f/spru281f.pdf>
4. Gardner, F.M. A BPSK/QPSK timing error detector for sampled receivers. // F.M. Gardner IEEE Trans. Commun.- 1986.- Vol. 34, P. 423–429.
5. Nezami, M.K. RF architectures & digital signal processing: Aspects of digital wireless transceivers. // M.K. Nezami IEEE Trans. Commun. - 2003.
6. Farrow, C.W. A continuously variable digital delay element. // IEEE Int. Symp. "Circuits and Systems".- Espoo, Finland, 1988.- P. 2641-2645.
7. Фильтры Фарроу (Farrow filters) на примере фильтра третьего порядка. Ресэмплинг сигналов. // DSPSYSTEM. Теория и практика цифровой обработки сигналов. [Электрон.ресурсы].URL: <http://www.dspsystem.narod.ru/content/farrow>
8. Modulo operation. // Википедия: свободная электронная энциклопедия: на английском языке. [Электрон. ресурсы]. URL: http://en.wikipedia.org/wiki/Modulo_operation

М.н.с., аспирант **Василевский В.В.**, тел. 8-960-990-1248, e-mail: altego@omgtu.ru; инженер, **Панюшкин В.А.**, тел. 8-908-792-30-07, e-mail: panyushkin@rtuisd.ru; инженер, **Пузырёв П.И.**, тел. 8-908-105-48-51, e-mail: puzyrev@rtuisd.ru Омский государственный технический университет, НИЧ, кафедра «Радиотехнические устройства и системы диагностики» (г. Омск).

УДК 681.2.001.66:004.5

ОБЩИЕ ПОДХОДЫ ПРИ РАЗРАБОТКЕ СПЕЦИАЛИЗИРОВАННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ВИРТУАЛИЗИРОВАННЫХ ИЗМЕРИТЕЛЬНЫХ ПРИБОРОВ

С.Ф. Дмитриев, Д.Н. Лященко, А.В. Новоженков, А.В. Ишков

Авторами выработаны общие подходы, лежащие в основе создания специализированного программного обеспечения виртуализированных измерительных приборов. Общие подходы при разработке приборных приложений основаны на использовании интерфейса низкого уровня для доступа к буферу с данными в wave-формате, микшерной подсистемы Windows для контроля параметров звукового устройства и модульном принципе построения алгоритма прибора. Приведено описание алгоритмов и основных функций wave-модуля, модуля инициализации и калибровки и модуля основного цикла приборов ИЭНМ-20М и ВВТ-2М.

Ключевые слова: виртуализированный измерительный прибор, программное обеспечение, интерфейс низкого уровня, микшерная подсистема, программный модуль.

Современные подходы к реализации измерительного прибора (ИП) базируются на нескольких положениях, связанных с общими

концепциями измерений, средствами их осуществления, объектами и их моделям, а также метрологическом эксперименте [1, 2].

ОБЩИЕ ПОДХОДЫ ПРИ РАЗРАБОТКЕ СПЕЦИАЛИЗИРОВАННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ВИРТУАЛИЗИРОВАННЫХ ИЗМЕРИТЕЛЬНЫХ ПРИБОРОВ

При разработке любого ИП сначала необходимо определиться с самой измеряемой характеристикой или сигналом, поступающим от объекта измерения так, чтобы он обладал максимальной информативностью, затем выбрать сам метод измерения (сравнения исследуемого сигнала с эталоном или шкалой), который в общем виде может быть абсолютным, прямым, относительным или косвенным, потом разработать новое или применить известное средство измерений (**СИ**) - техническое устройство, имеющее нормированные метрологические показатели и реализующее выбранный ранее метод, и, наконец, первичный преобразователь - устройство, перерабатывающее полученную от объекта измерительную информацию в форму, удобную для дальнейшего преобразования, передачи, хранения и обработки.

Учитывая обстоятельство постоянного преобразования измерительного сигнала, часто все блоки ИП относят к преобразователям (первичным, промежуточным, передающим и пр.), которых в любом приборе оказывается как минимум четыре [3].

Из сказанного выше понятно, что практически любой ИП содержит в себе специализированную архитектуру преобразовательных блоков с определенными характеристиками, создается под конкретные измерительные задачи и модели объектов измерения и не может быть универсальным СИ.

При сопряжении ИП с ЭВМ, последняя, как правило, используется для вторичной обработки окончательно преобразованного сигнала либо как мощное средство визуализации измерений.

В то же время возможности современной вычислительной техники и периферийных устройств ЭВМ, несущих в своем составе многоканальные и высокоскоростные АЦП/ЦАП, позволяют совсем иначе реализовать схему сопряжения ЭВМ с измерительным прибором.

Если в классической схеме сопряжения измерительного прибора с ЭВМ, СИ и ПК представляли собой отдельные системы, то предлагаемая и реализуемая нами концепция виртуализированного измерительного прибора (**ВИП**) основана на реализации основных преобразовательных и измерительных функций СИ в специализированном программном обеспечении (**ПО**) конкретного измерительного программно-аппаратного комплекса - самоисполняемой компьютерной программе, не требующей установки на ПК дополнительных (кроме операционной) сред и связи ее с первичным датчиком-преобразователем, по-

средством имеющихся в ЭВМ АЦП/ЦАП, портов и пр. [4, 5].

В настоящей работе описаны некоторые общие подходы, позволяющие эффективно разрабатывать специализированное ПО для различных ВИП, использующих в качестве АЦП/ЦАП звуковой адаптер ПК.

Постановка задачи

Схемы программно-аппаратных комплексов, подобных ВИП, уже сейчас реализуются, например, в среде разработки специализированного ПО LabView, либо приемами раздельной программной обработки data-файлов, содержащих измерительную информацию или параметры различных объектов стандартными средствами Windows [6]. Однако, необходимость постоянного обращения к оболочке среды разработки или ее библиотекам либо «ручного» копирования и обработки файлов с данными (при втором подходе) снижают скорость измерений, увеличивают систематическую и случайную погрешность измерений и сильно ограничивают возможности обработки измерительных сигналов встроенным в такие среды ПО. Кроме того, это лишь макеты (виртуальные приборы), неспособные самостоятельно запускаться на ПК при отсутствии на них предварительно установленной среды разработки, что существенно ограничивает их области применения, увеличивает стоимость и ставит в зависимость от лицензируемого программного обеспечения (ПО), их разработчиков.

Особенности программной реализации

Первый общий подход к разработке специализированного ПО для ВИП, заключается в использовании интерфейса низкого уровня. В отличие от MS DOS, термин «низкоуровневое» в Windows означает не манипуляцию регистрами, прерываниями и каналами DMA, а просто самый элементарный уровень сервиса, обычно предоставляемого непосредственно драйвером устройства. В нашем случае таким устройством является звуковой адаптер (звуковая карта) ЭВМ.

Низкий уровень позволяет приложениям получать доступ к буферам, содержащим воспроизводимые или записываемые звуковые данные, работать с внутренней структурой файлов, содержащих звуковую информацию, а также использовать другие дополнительные возможности [7].

Для работы со звуковым адаптером в ОС Windows нет необходимости программировать на уровне портов ввода/вывода, прерываний и каналов прямого доступа. Весь необ-

РАЗДЕЛ V. СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

ходимый интерфейс (высокого или низкого уровня) предоставляется приложению DLL-библиотекой `mmsystem.dll` [8].

В `mmsystem.dll` определены функции низкого уровня (Low-Level Functions) и функции высокого уровня, представляющие собой интерфейс для управления средой (MCI).

Для разработки специализированного ПО нами также используются два стандартных драйвера звукового устройства: драйвер для ввода звука (Waveform Input Driver) и драйвер для вывода звука (Waveform Output Driver). Эти драйверы поставляются вместе со звуковым адаптером, устанавливаются после установки ОС и позволяют Windows обмениваться со звуковыми устройствами wav-файлами, а при установке Codecs-ов и файлами других форматов. Для непосредственного доступа к буферам, содержащим звуковые данные, приложение использует интерфейс низкого уровня, обеспечиваемый в `mmsystem.dll` несколькими функциями с префиксом имени `wave` (`waveInOpen`, `waveOutOpen`, `waveOutWrite`, `waveAddBuffer` и т.д.). Таким образом, любое создаваемое под Windows ПО автоматически будет иметь полный контроль над процессом воспроизведения/записи, так как оно само должно подготовить блоки данных в оперативной памяти.

В отличие от интерфейса MCI, где многие параметры воспроизведения/записи принимаются по умолчанию, интерфейс низкого уровня требует учета всех деталей этого процесса, позволяя получить большую гибкость и возможность работы со звуковыми данными в реальном времени, необходимые для реализации приборов, использующих звуковую карту ПК как АЦП/ЦАП.

Так как звуковое устройство, используемое нами как высокоскоростное АЦП/ЦАП в составе программно-аппаратного комплекса, в Windows может одновременно использоваться не только специализированным приложением, но и любой другой программой или самой ОС, то требуется надежный контроль его переменных параметров в процессе работы виртуализированного прибора (любое из приложений, обращающихся к адаптеру через `mmsystem.dll`, может изменить параметры входного и выходного сигнала, уровень громкости записи/воспроизведения, частоты каналов и пр.). Эта задача является серьезным препятствием при реализации любых программно-аппаратных комплексов через встроенный звуковой адаптер ЭВМ, зачастую сводя на нет все усилия разработчиков.

Второй общий подход к разработке специализированного ПО для ВИП, заключается в использовании микшерной подсистемы Windows для получения приложением возможности жесткого контроля параметров звукового устройства [7].

Базовый звуковой интерфейс (MME) включает три класса устройств - Wave (собственно цифровой звук), MIDI (синтезаторы и устройства управления в стандарте MIDI) и Mixer/Aux (микшеры и дополнительные устройства), отражая архитектуру типового адаптера, содержащего тракт цифрового звука, синтезатора и микшера, соединяющего выходы источников звука с входами приемных устройств.

Микшеры, как вполне самостоятельные блоки звуковых адаптеров и виртуальных синтезаторов, также управляются в MME отдельной подсистемой, но в отличие от дополнительных устройств, в отношении которых из Windows доступно только управление громкостью, микшер позволяет осуществлять приложению коммутацию источников и приемников звука, регулировку уровня, панорамы, тембра и других параметров звука, смешивать несколько источников звука в единый звуковой сигнал, осуществлять контрольное прослушивание и корректировку сигнала, на одной или нескольких линиях.

Стандартное микшерное приложение ОС Windows - `SndVol32` предоставляет собой унифицированный интерфейс пользователя с любым возможным микшерным устройством, при добавлении в систему нового устройства, имеющего драйвер микшера, оно также может управляться с помощью этого стандартного приложения.

Кроме возможности получения уведомлений, открывание микшера гарантирует его существование до завершения сеанса работы. Открытый микшер невозможно удалить из системы до тех пор, пока он не будет закрыт, а неоткрытый микшер может быть удален в любой момент времени. Вдобавок, доступ к открытому микшеру происходит по ключу, который не изменяется в течение всего сеанса работы, а к неоткрытому - по номеру, который также может смениться при установке или удалении других микшерных устройств. В отличие от драйверов Wave/MIDI, драйверы микшерных устройств чаще всего допускают параллельное открывание микшеров несколькими процессами. При открывании микшера подсистема возвращает его идентификатор, или ключ (тип - `HMIXER`), по которому затем происходит обращение к микшеру.

ОБЩИЕ ПОДХОДЫ ПРИ РАЗРАБОТКЕ СПЕЦИАЛИЗИРОВАННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ВИРТУАЛИЗИРОВАННЫХ ИЗМЕРИТЕЛЬНЫХ ПРИБОРОВ

Таким образом, в микшерной подсистеме может возникать только два типа событий - изменение состояния линии (line status change) и изменение состояния элемента управления (control status change). О наступлении каждого из этих событий подсистема уведомляет специализированное ПО с тем, чтобы она обновила состояние органов управления и индикаторов на экране либо сообщила пользователю о нарушении параметров настройки ВИП, если обновление невозможно.

Все необходимые константы, типы, структуры и прототипы функций микшерной подсистемы Windows определяются в файле

mmsystem.h, который по умолчанию включается в компиляцию из общего файла windows.h, дополнительные константы определены в файле mmreg.h, а интерфейсные функции импортируются из библиотеки winmm.lib [8].

С использованием описанных выше подходов управления звуковым адаптером, его настройками и связи со стандартными библиотеками ОС Windows, нами был разработан блок универсального ПО для ВИП, использующих встроенный звуковой адаптер ПК как АЦП/ЦАП (рисунок 1).

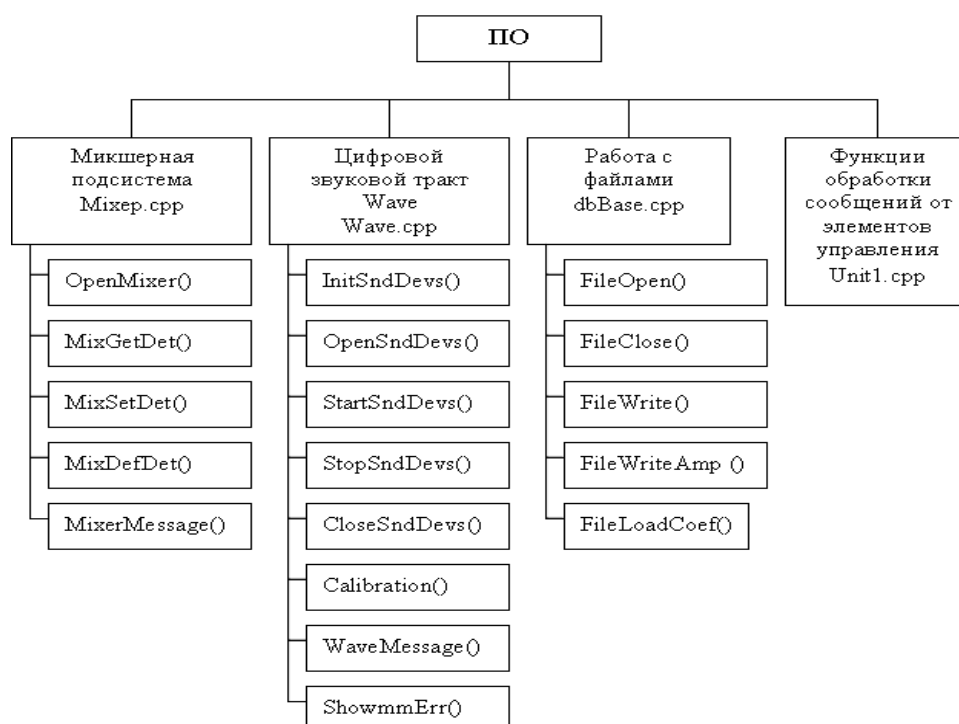


Рисунок 1 - Функциональная схема wave-блока универсального ПО для ВИП

Программное обеспечение для работы с wave-данными было разработано на языке высокого уровня C++ и тестировано на ОС MS Windows 95/98/2000/XP/Vista, имеет интерактивный оконный интерфейс, обеспечивает оцифровку измерительных сигналов звуковой частоты в режиме реального времени с последующей их передачей специализированному ПО ВИП.

Разработанный универсальный wave-модуль ПО можно условно разделить на четыре функциональных элемента: - управление микшерной подсистемой Windows (Mixer.cpp), - управление стандартным звуко-

вым устройством Windows (Wave.cpp), - функции для работы с файлами (dbBase.cpp), - функции обработки сообщений элементов управления (Unit1.cpp).

Функции каждого из модулей ПО при работе могут, кроме того, вызывать встроенные функции среды разработки Borland C++ Builder v. 6.0, а так же API-функции

Третий подход к реализации ПО ВИП основывается на модульном принципе его организации как ИП прямого действия [2, 3], при последовательной обработке звуковых wave-данных в режиме реального времени. Тогда ВИП будет содержать следующие модули: - запуска программы; - запуска wave-блока; - инициализации и калибровки устройства; - запуска основного (измерительно-

РАЗДЕЛ V. СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

преобразовательного) цикла программы; и - блок завершения программы.

Результаты анализа работы устройств

Рассмотрим алгоритмическую реализацию этих подходов на примере разработанных нами ВИП, использующих для измерений метод вихревых токов (МВТ) - измерителя электропроводности неферромагнитных материалов (ИЭНМ-20М) и виртуализированного вихретокового толщиномера (BBT-2М).

При запуске специализированного ПО этих приборов сначала происходит их инициализация (рисунок 2).



Рисунок 2 - Алгоритм инициализации и калибровки

После инициализации происходит калибровка устройства, которая заключается в замере уровня нуля на сигнальной катушке первичного преобразователя, одинакового для обоих приборов - вихретокового трансформаторного преобразователя накладного типа (ВТПНП).

Выбор звукового устройства включает в себя проверку на наличие в системе звуковых устройств, а так же определение их названия и количества. Осуществляется это при помощи функции `InitSndDevs()`.

При открытии устройства функцией `OpenSndDevs()`, происходит открытие выбранного wave-устройства, а так же инициализация входных и выходных буферов. При

этом создается один выходной буфер размером $SPS \cdot BPS / 8$ байт, где SPS – частота дискретизации, BPS – число бит в выборке.

В нашем случае для выходного сигнала частота дискретизации 44100 Гц/48000 Гц, число бит в выборке 16. Частота генерируемого сигнала составляет 16000 Гц. Сигнал является монофоническим. Так же мы указывается, что в цикле воспроизведения будет только один буфер. В отличие от входного буфера нам необходимо два выходных. Пока один буфер заполняется новыми данными, мы осуществляем необходимую обработку данных второго буфера. Размер буферов определяется как фиксированная часть секунды исходя из частоты дискретизации (44100 Гц/48000 Гц), числа бит в выборке (16 бит) и числа каналов (1 канал).

В нашем случае размер буфера составляет $SPS \cdot BPS / (8 \cdot 10)$ (1/10 секунды). Частота входного сигнала 16000 Гц. Далее, при помощи функции `FillBuf (int freq, int amp)`, происходит заполнение выходного буфера значениями синусоидального сигнала заданной амплитуды и частоты.

Установка уровня выходного сигнала осуществляется двумя способами, первый это задание амплитуды выходного сигнала при заполнении выходного буфера (параметр `amp` функции `FillBuf (int freq, int amp)`), а второй это установка уровня громкости системного микшера, при помощи микшерной подсистемы.

При вызове функции `OpenMixer(UINT n)`, где параметр `n` - это номер открытого wave-устройства, происходит установка регуляторов громкости следующих линий: - регулируемый вход оконечного усилителя мощности громкоговорителей; - выход ЦАП цифрового тракта адаптера; - регулируемый выход микрофонного предусилителя.

Запуск процесса генерации и считывания сигнала определенной частоты сопровождается вызовом функции `StartSndDevs()`, которая в свою очередь вызывает функции `waveInStart()`, `waveOutWrite()`.

Запуск процесса считывания данных со входа звуковой платы начинается с открытия устройства записи вызовом функции `waveInOpen`. Через параметр `dwCallback` мы передаем функции `waveInOpen` адрес функции обратного вызова. Для этих целей функцией `AllocateHWND` создается невидимое окно для отработки сообщений устройства ввода. По мере заполнения буферы будут передаваться невидимому окну, где и будет осуществляться их обработка.

ОБЩИЕ ПОДХОДЫ ПРИ РАЗРАБОТКЕ СПЕЦИАЛИЗИРОВАННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ВИРТУАЛИЗИРОВАННЫХ ИЗМЕРИТЕЛЬНЫХ ПРИБОРОВ

Калибровка включает в себя выставление уровня нуля, одновременно, в это же время осуществляется проверка наличия подключенного датчика и правильности его подключения к звуковой плате, которая происходит посредством анализа входных данных (рисунок 2).

После того как инициализация и калибровка будут удачно завершены, начинается основной цикл работы программы в функции DataWaveIn() (рисунок 3).

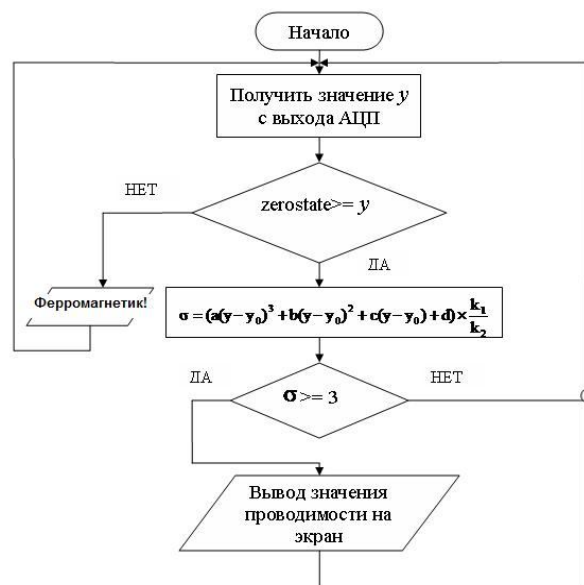


Рисунок 3 - Алгоритм основного цикла ВИП
ИЗНМ-20М

В ходе выполнения основного рабочего цикла работы программы происходит постоянное считывание уровня сигнала с выхода АЦП микрофонного входа звуковой карты. На основе получаемых данных осуществляется разделение металлов на ферромагнитные и неферромагнитные. Если уровень нуля начинает увеличиваться, то это говорит о том, что ВТП был установлен на ферромагнитный объект, и на экран выводится надпись «Ферромагнетик!».

Если же уровень нуля начинает снижаться, то это значит, что ВТП установлен на неферромагнитный объект, и в этом случае производится расчет электропроводности. Если $\sigma > 3$, то результат измерения выводится на экран.

При завершении работы программы происходит закрытие открытого звукового устройства и ассоциированного с ним микшера. Так же происходит и освобождение памяти, которая выделялась под входные и выходные буферы ПО.

Отличие алгоритма основного цикла ВИП ВВТ-2М от приведенного на рисунке 3, заключается только в виде используемой аппаратной функции измерительного преобразователя, которая для толщины диэлектрического покрытия на неферромагнитном основании описывается не полиномической, а экспоненциальной зависимостью.

Заключение

Поставленная в работе задача разработки специализированного программного обеспечения виртуализированных измерительных приборов в виде самоисполняемых в ОС Windows exe-файлов была реализована с использованием нескольких общих подходов.

Во-первых, это использование интерфейса низкого уровня, предоставляемого в Windows драйвером звукового адаптера ЭВМ.

Во-вторых, использование микшерной подсистемы Windows для получения приложением возможности жесткого контроля параметров звукового устройства.

В-третьих, реализация программного обеспечения на модульном принципе, в режиме реального времени с архитектурой измерительного прибора прямого действия.

С использованием указанных подходов были разработаны действующие виртуализированные приборы ИЗНМ-20М и ВВТ-2М, реализующие различные варианты метода вихревых токов.

СПИСОК ЛИТЕРАТУРЫ

1. Марков, Н.Н. Конструкция, расчет и эксплуатация измерительных инструментов и приборов. / Н.Н. Марков, Г.М. Ганевский. - М.: Машиностроение, 1981.
2. Савченко, В.Т. Измерительная техника [Текст] / В.Т. Савченко. - М.: Высшая школа, 1974.
3. Димов Ю.В. Метрология, стандартизация и сертификация [Текст] / Ю.В. Димов. - СПб.: Питер, 2008.
4. Ишков, А.В. Современная концепция сопряжения измерительных приборов с ЭВМ. // Измерение, контроль, информатизация: Материалы восьмой междунар. научн.-техн. конф. [Текст] / Ишков А.В., Дмитриев С.Ф. - Барнаул: Изд-во АлтГТУ, 2007. С. 3-6.
5. Дмитриев, С.Ф., Панов С.Г., Ишков А.В. Виртуализированные приборы, использующие метод вихревых токов, для контроля параметров различных объектов и окружающей среды. // Ползуновский альманах. № 2. 2008. С. 15-20.
6. Жарков, Ф.П., Каратаев В.В., Никофоров В.Ф., и др. Использование виртуальных инструментов LabView. - М.: Радио и связь, 1999.
7. Гордеев, О. Программирование звука в Windows. - М.: BHV, 2007.
8. Страуструп, Б. Язык программирования C++. - СПб.: Невский Диалект, 1999.

К.т.н., доцент Дмитриев С.Ф., тел. (3852) 24-76-18, e-mail: dmitrlm@mail.ru, Алтайский государственный университет; ст. преп., аспирант Лященко Д.Н., тел. (3852) 62-83-80; e-mail: LDN-ray@yandex.ru, Алтайский государственный аг-

рарный университет; инженер, аспирант Ново-женов А.В., тел. (3852) 48-57-14, Алтайский госуниверситет; .к.х.н., д.т.н., профессор Ишков А.В., e-mail: olg168@rambler.ru, Алтайский государственный аграрный университет (г. Барнаул)

УДК 004.92

ДИНАМИЧЕСКОЕ РЕГУЛИРОВАНИЕ УРОВНЯ ДЕТАЛИЗАЦИИ ТРЕХМЕРНЫХ МОДЕЛЕЙ СЛОЖНЫХ ИЗМЕРИТЕЛЬНЫХ ПРИБОРОВ

В.М. Калачев, Н.Н. Решетникова

Предложен алгоритм, который позволяет динамически регулировать уровень детализации полигональных моделей сложных 3D-объектов в интерактивных сценах. Визуальные артефакты при упрощении полигональных сеток не возникают.

Ключевые слова: 3D-объект, полигональная модель, уровень динамической детализации, интерактивность

Современные мощности персональных компьютеров позволяют решать задачи компьютерной графики на качественно новом уровне. Становится возможным моделирование и отображение на экране компьютера пространства, окружающего жизнь человека, и организация взаимодействия с виртуальным пространством. Несмотря на рост вычислительных мощностей, персональные компьютеры и даже специализированные графические станции не в состоянии отображать полноценные трехмерные изображения аналогичные тем, которые «формирует» человеческий глаз, поддерживая при этом интерактивность происходящего.

Поэтому одной из основных проблем визуализации интерактивных трехмерных сцен является производительность конкретной системы отображения виртуального пространства, так как вычисления, которые предваряют визуализацию трехмерного образа, достаточно тяжеловесны. Реализованные в полигональной технологии ограничения, в соответствии с которыми любой трехмерный объект создается из набора полигонов (треугольников, однозначно определяющих плоскость в пространстве) и наложенных на эти треугольники текстурных изображений, лишь частично решают проблему.

Постановка задачи

Одним из методов, позволяющих ускорить процесс визуализации трехмерных объектов, является представление этих объектов с различной степенью визуального разрешения или различными уровнями детализации (Level Of Detail, **LOD**) [1]. Уровень детализа-

ции есть некоторая аппроксимация начальной модели объекта с использованием меньшего количества геометрических деталей - полигонов.

Как известно качество изображения, воспринимаемое глазом, имеет свойство ухудшаться с увеличением дистанции до объекта. Именно на этом свойстве основана большая часть современных LOD алгоритмов. Исходя из текущего расположения наблюдателя, алгоритм регулирует уровень детализации поверхности модели объекта.

При реализации видео - независимого LOD алгоритма, создается специальная структура, из которой в процессе визуализации в реальном времени извлекается нужный объект с определенным количеством полигонов. Преимуществом здесь является плавность изменения детализации, а также возможность поэтапного ее увеличения. При этом уровень детализации для каждого объекта определяется точно, а не выбирается из нескольких заранее созданных вариантов, тогда и полигонов используется ровно столько, сколько требуется. Использование такого метода приводит к экономии вычислительных ресурсов.

Алгоритмическая реализация

Рассмотрим в обобщенном виде работу видео-независимого алгоритма упрощения полигональных сеток по шагам.

Шаг 1. После запуска алгоритма, выполняется проверка полигональной сетки целевого объекта на присутствие в ней полигонов, которые имеют четыре и более вершин. Если такие полигоны имеются, то алгоритм произ-