

зультаты макетирования. Система обеспечивает генерацию спецификаций сетевых архитектур с сохранением результатов в XML базе данных и генерацию html-отчетов с описаниями СА, с тестовой информацией по данным из XML базы данных.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Ивницкий, В.А. Теория сетей массового обслуживания / В.А. Ивницкий. – М.: Физико-математическая литература, 2004. – 772 с.
2. Андерсон, Д. Дискретная математика и комбинаторика / Д. Андерсон. – СПб.: Вильямс, 2004. – 960 с.

УДК 004.514.62

Никонюк А.Н.

Научный руководитель: к.т.н, доцент Костюк Д.А.

РЕАЛИЗАЦИЯ МОДУЛЯ РАСШИРЕНИЯ ДЛЯ ДИНАМИЧЕСКОГО НЕЛИНЕЙНОГО МАСШТАБИРОВАНИЯ В ОКОННОМ МЕНЕДЖЕРЕ COMPIZ

Задача манипулирования окнами в ситуации, когда их совокупная площадь существенно превышает разрешение устройства вывода, возникла почти одновременно с графическим интерфейсом пользователя. Среди способов облегчить навигацию в таких условиях можно упомянуть прокрутку рабочего пространства, позволяющую видеть его по частям, и иконификацию – «сворачивание» окна в пиктограмму, снабженную поясняющей подписью. С ростом вычислительных возможностей процессоров и разрешающей способности экранов появились графические оболочки и приложения, отображавшие вместо пиктограммы скрытого окна его уменьшенное изображение. Хотя такой подход сопряжен с рядом технических трудностей, возможность видеть одновременно, хотя бы с уменьшенной детализацией, изображение всего рабочего пространства дает пользователю ощутимые преимущества [1, 2]. Особенно это актуально в среде динамически изменяющихся объектов, к которым относятся все многозадачные графические среды современных операционных систем.

В последнее время проблема недостатка рабочей области получила дополнительное развитие из-за роста популярности портативных устройств – нетбуков и планшетных компьютеров, способных в той или иной степени запускать приложения, интерфейс и модель взаимодействия с пользователем в которых изначально рассчитаны на стандартное разрешение и размер экрана. Подобные устройства оказываются не только не способны разместить на экране нужное количество окон, но зачастую не могут показать целиком одно стандартное окно.

Нами разработана модель нелинейного масштабирования окон и реализующий ее модуль расширения для оконного менеджера CompiZ, позволяющие сократить размер занимаемой площади окна произвольного приложения с сохранением читаемости и доступности его содержимого.

Концепция нелинейного масштабирования окон и ее реализация для аппаратно-ускоренного манипулирования окнами

Концепция переменного масштаба смоделирована нами в виде деления рабочей области окна на две зоны: центральную 1 и периферийную 2. В зоне 1 отображается основная информация окна, представленная в единичном масштабе. В зоне 2 выводится изображение, искаженное в соответствии с функцией $\alpha(y)$, монотонно возрастающей

вдоль координаты y , по которой выполняется изменение масштаба, и имеющей область значений, принадлежащую интервалу $(0; 1]$. Таким образом, рабочая область делится на две части, и позиции пикселей, находящиеся во второй части, пересчитываются с переменным коэффициентом масштабирования [1].

Выбор функции изменения масштаба имеет существенное значение. В частности, использование функции $a(y) = y^{1/K}$ позволяет лучше видеть элементы, расположенные ближе к центральной области, и сильнее скрадывает находящиеся на большем удалении [2].

Все современные аппаратно-ускоренные менеджеры окон, (включая Comviz) используют возможности библиотеки OpenGL для передачи вычислительной нагрузки графическому акселератору. Одно из свойств OpenGL – объекты фреймбуфера – дает оконному менеджеру эффективный доступ к окнам неактивных приложений. Для приложения объекты фреймбуфера выглядят как обычные окна, а для оконного менеджера – как текстуры, которыми можно управлять с помощью обычных команд отрисовки мультитекстур (рис. 1).

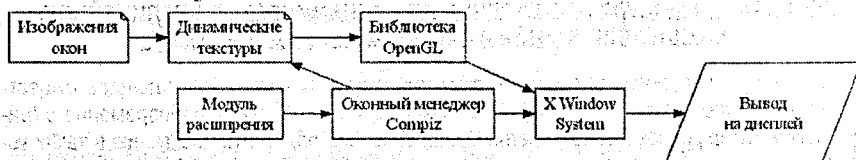


Рисунок 1 – Схема взаимодействия разработанного модуля с графической оболочкой и средствами поддержки аппаратно-ускоренной графики

Разработанный нами модуль *twist* отрисовывает измененные изображения окон на основе этой технологии. На рис. 2 приведена диаграмма зависимостей разработанного модуля к наборам функций оконного менеджера Comviz. Среди подключаемых модулей можно отметить модули сервисных функций ядра Comviz, подсистемы отрисовки сцены OpenGL, модуль композитных эффектов, а также модуль конфигурирования, обеспечивающий интеграцию пользовательских настроек разработанного модуля в общую систему настройки Comviz, и модуль опроса мыши (*mousepoll*). Использование последнего облегчает разработанному модулю задачу слежения за координатами указателя мыши. Модуль *mousepoll* берет на себя задачу контроля координат, централизованно считывая каждые несколько миллисекунд позицию указателя и тем самым избавляя систему от дублирования такой затратной операции несколькими модулями.

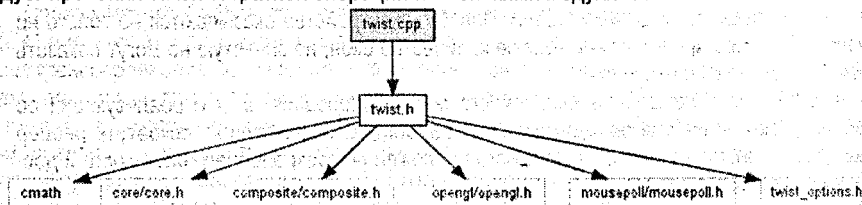


Рисунок 2 – Использование стандартных модулей Comviz

Типичный модуль Comviz состоит из трех функциональных частей: кода инициализации, обработчиков событий и сервисных функций. Код инициализации вызывается при загрузке модуля в память. При этом системе передаются указатели на конструкторы и деструкторы, вызываемые при создании и удалении контекста экрана и каждого из соот-

ветствующих экрану окон. Доступ к контексту экрана позволяет перехватывать события и дает возможность вклинуться в цепочку процедур обновления изображения, включающую подготовительную стадию, собственно отрисовку экрана (или экранов, если их несколько) и завершение отрисовки (рис. 3).



Рисунок 3 – Стадии отрисовки виртуальных экранов

Подготовительная стадия используется для выполнения различных подготовительных расчетов, которые имеет смысл выполнять перед началом отрисовки. Любой метод отрисовки в первую очередь должен вызывать базовую реализацию, обновляющую изображение экрана, а затем выполняет с полученным изображением любые действия, доступные средствами OpenGL; при этом он имеет доступ к устройству вывода, физическим координатам и всем открытым окнам, для каждого из которых существует соответствующая текстура. Стадия завершения отрисовки при необходимости используется для принудительной генерации события перерисовки экрана, благодаря чему выполняется отображение анимационных эффектов в реальном масштабе времени.

Алгоритм масштабирования окон

Разработанный нами модуль расширения активизируется при перетаскивании окна за пределы границы экрана (или за пределы рабочей области, если вдоль данной границы экрана расположена панель графической оболочки). Сжатие соответствующей части окна отображается в реальном масштабе времени в процессе дальнейшего перемещения, и т. о. пользователь имеет возможность регулировать коэффициент сжатия для достижения баланса между размерами и читаемостью содержимого окна.

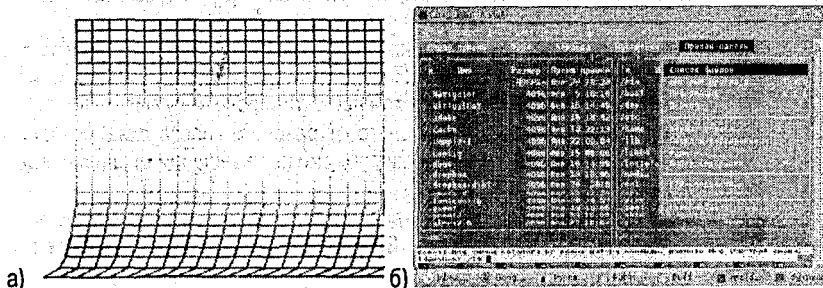


Рисунок 4 – Отображение окна с переменным масштабом: базовая трехмерная фигура (а) и наложенное окно в ортогональной проекции (б)

Для отображения нелинейного масштаба изображения применены два различных подхода.

В первом подходе, названном в настройках модуля «screw effect», текстура накладывается на базовую трехмерную фигуру, для которой выражение $a(y) = y^{1/k}$ определяет изменение z-координаты. Для доступа к текстуре скрытой за экраном части окна модуль предварительно выполняет масштабирование с коэффициентом, близким к единице, после чего создает копию части текстуры, соответствующей сжимаемой части. Дублированное изображение выводится поверх видимого фрагмента окна. На рис. 4а условно показан фрагмент базовой фигуры, для наглядности включающий как неискаженную об-

ласть 1, так и искаженную область 2. На практике, однако, строится только фрагмент фигуры, отвечающий искаженной области. Для вывода базовой фигуры и соответствующего результирующего изображения средствами OpenGL может применяться как ортогональная проекция (рис. 4б), для которой характерна лучшая читаемость содержимого окна, так и перспективная проекция, которая обладает большей визуальной выразительностью и может использоваться для наглядной демонстрации эффекта.

Во втором подходе (активируемом выбором опции «compress effect» в диалоге настроек) вместо наложения на криволинейную поверхность выполняется добавление промежуточных опорных вершин к фигуре базового прямоугольника окна и смещение части вершин в плоскости экрана в соответствии с функцией изменения масштаба. Базовая фигура является двумерной, вследствие чего подход позволяет воспроизвести только аналог ортогональной проекции, показанной на рис. 4б; однако он является менее затратным в отношении потребляемых вычислительных ресурсов.

Изменение масштаба окна затрагивает только его изображение. Для самого окна при этом не происходит никаких изменений в размерах, иначе была бы нарушена обратная совместимость с существующими приложениями. Поэтому, с точки зрения системы, все элементы управления окна сохраняют свои прежние координаты, и для взаимодействия с этими элементами необходимо помещать указатель мыши в ту область экрана, где они находились бы, сохраняя окно единичный масштаб [3, 4]. Т.о. необходимо либо блокировать события указателя мыши, относящиеся к сжатой области, либо выполнять пересчет координат. На текущий момент реализован промежуточный вариант решения проблемы. В процессе слежения за положением указателя мыши регистрируется помещение его в сжатую область окна и последующая смена расположения зоны сжатия окна так, чтобы исключить контакт указателя со сжатой областью.

Для реализации представленного алгоритма сжатия окон [5] модуль twist содержит следующие функциональные части:

- конструктор класса TwistScreen, производного от базового класса экрана Compiz, устанавливает обработчики событий для слежения за перемещением указателя мыши, а также для перерисовки экрана, необходимой при активированном screw effect;
- конструктор класса TwistWindow, производного от базового класса окна, создающий необходимые текстуры, а также обработчики событий перемещения мыши в пределах окна и перерисовки окна;
- обработчик перерисовки экрана, который вклинивается в цепочку отрисовки на финальной стадии DonePaintScreen (см. рис. 2) и генерирует событие перерисовки для каждого из присутствующих на экране окон;
- обработчик перерисовки окна, выполняющий в случае активации модуля для выбранного окна смещение опорных вершин для compress effect либо копирование фрагмента текстуры данного окна, соответствующей сжатой области 2, создание криволинейной поверхности рис. 4а и наложение на нее копии текстуры для screw effect;
- обработчик перемещений указателя мыши, проверяющий координаты текущей позиции на принадлежность сжатой части активного окна и меняющий расположение сжатой области так, чтобы избежать контакта данной зоны с указателем мыши.

Разработанный модуль динамического нелинейного масштабирования окон позволяет добиться более экономного использования площади экрана портативных устройств без модификации существующего прикладного программного обеспечения. За счет использования аппаратно-ускоренной графики удается выполнять пересчет изображений окон в реальном масштабе времени без увеличения нагрузки на центральный процессор.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Раскин, Дж. Интерфейс: новые направления в проектировании компьютерных систем. – СПб.: Символ-Плюс, 2003. – 272 с.
2. Гоманова, Е.В. Применение аналогии периферического зрения в аппаратно-ускоренном графическом интерфейсе пользователя / Е.В. Гоманова, Д.А. Костюк, К.Л. Костюк // Вестник БрГТУ. – 2007. – №5: Физика, математика, информатика. – С. 33–35.
3. Diomin, V.V. Grouped windows focus switching with variable scale factor / V.V. Diomin, D.A. Kostyuk // Computer Science & Engineering 2010 (CSE-2010): proc. of the IV International Academic Conference of Young Scientists. November 25-27 2010. – Lviv, Ukraine. – P. 32–33.
4. Костюк, Д.А. Модель мини-окон с динамическим отображением в аппаратно-ускоренном графическом интерфейсе / Д.А. Костюк, В.В. Дёмин // Вестник БрГТУ. – 2009. – №5: Физика, математика, информатика. – С. 71–74.
5. Никонюк, А.Н. Нелинейное масштабирование окон для экономии площади дисплея портативных устройств // Сучасні проблеми радіотехніки та телекомунікацій «РТ – 2011»: Матеріали 7-ої міжнар. молодіжної наук.-техн. конф., Севастополь 11–15 квітня 2011 р. – Севастополь: СевНТУ, 2011. – С. 362.

УДК 004.514.62

Никонюк А.Н.

Научный руководитель: к.т.н., доцент Костюк Д.А.

РЕАЛИЗАЦИЯ ФРЕЙМОВОГО МЕНЕДЖЕРА ОКОН С ФУНКЦИЯМИ НЕЛИНЕЙНОГО МАСШТАБИРОВАНИЯ

Фреймовый (или тайловый, от англ. tile – мозаика, черепица) оконный менеджер разбивает рабочее пространство экрана на взаимно непересекающиеся прямоугольные области — фреймы. Каждый фрейм используется для вывода информации отдельным приложением, в противоположность традиционным менеджерам окон, которые создают области вывода приложений по координатам и послыюно (окна), в рамках метафоры рабочего стола. Помимо полного использования всего видимого пространства, подход фреймовых оконных менеджеров позволяет минимизировать использование мыши за счет исключения необходимости перемещения окон и преимущественного использования клавиатурных комбинаций для переключения фокуса окна, уменьшая таким образом время, затрачиваемое на выполнение рутинных операций и повышая эффективность работы [1].

Хотя первый растровый графический интерфейс использовал именно метафору рабочего стола (в рамках языка Smalltalk, в Xerox PARC), тайлинг окон появился лишь немногим позже, в первой коммерческой системе Xerox Star (в сочетании с классическими плавающими окнами диалогов). В последующие 10 лет почти все популярные графические оболочки, включая Microsoft Windows и Digital Research GEM, проходили период экспериментов с тайлингом окон, однако разработчики раз за разом отказывались от него под давлением пользователей.

Хотя фреймовые оконные менеджеры не стали решением по умолчанию ни для одной популярной графической оболочки, многие приложения используют аналогичную метафору для отображения собственного функционала: почтовые программы, интегрированные среды разработки, боковые панели веб-браузеров, панель контекстной помощи в Microsoft Office. К варианту реализации тайлинга на языке разметки можно отнести HTML-фреймы. Разделение экрана на смежные части для одновременного отображения нескольких (обычно связанных) потоков видео или текстовой информации на одном дисплее широко применяется в программах трёхмерного моделирования (использование отдельных фреймов для одновременного просмотра и редактирования модели с двух сторон) в многопользовательских видеоиграх (для отображения игровых событий с