

уровнях детализации ГПИ, где нет возможности вместить все изображение целиком и потому необходима прокрутка. Причина в общеизвестном недостатке полос прокрутки с точки зрения эргономики.

Использование нелинейного масштабирования периферийных областей вместо прокрутки позволяет визуально ориентироваться при поиске элемента, находящегося вне центральной (рабочей) части окна.

#### **Интерфейс с аппаратно-ускоренной графикой**

Традиционный способ вывода окон экрана в выделенные им области – использование прямого или обратного алгоритма закрашивания. В случае прямого алгоритма каждый последующий слой наносится на предыдущий. Сначала отрисовывается фон, а затем каждое окно поверх предыдущего, как бы нанизываясь на ось Z, направленную перпендикулярно плоскости экрана. В случае обратного алгоритма сначала выводится самое верхнее окно, а затем каждое следующее, используя прямоугольники отсечения – таким образом, чтобы каждый пиксель наносился на экран только один раз. Данные алгоритмы имеют существенный недостаток – в некоторых случаях алгоритм не может точно вычислить порядок расположения окон относительно оси Z, что не позволяет реализовать полноценный трехмерный интерфейс.

При появлении поддержки аппаратных возможностей современного видеоадаптера в отрисовке окон наметились сдвиги. Например, расширение Composite для X Window System по-прежнему использует все тот же закрашивающий алгоритм, но, отрисовка окон делается во внеэкранный, невидимую область видеопамати. Затем оконный менеджер составляет из этих окон изображение для видимой части. Поскольку оконный менеджер имеет в своем распоряжении содержимое всех окон, становится возможным организовать частичную прозрачность и некоторые другие эффекты.

На основе анализа экспериментальных интерфейсных решений, использующих аппаратное ускорение трехмерной графики, разработана реализация предлагаемых моделей, позволяющая использовать основанные на них рабочие среды с уже существующим программным обеспечением без модификации последнего, за счет перехвата изображения и его обработки в реальном масштабе времени графическим процессором стандартной видео-карты. Для реализации моделей ПЗ было выбрано расширение Xgl системы X Window System и кодовая база оконного менеджера Comriz [3]. Выбор в пользу Xgl обоснован его более стабильной работой по сравнению с аналогом и меньшей избирательностью к аппаратному обеспечению. На выбор Comriz повлияла его модульная структура, позволяющая реализовать желаемые возможности в отдельных модулях, обходясь минимальными изменениями остального кода. Так, на основе его расширяемой архитектуры относительно несложно реализовать код, который будет выполнять масштабирование окон по нужному алгоритму, соответствующему разработанному в данной работе моделям.

#### **ЛИТЕРАТУРА:**

1. Раскин Дж. Интерфейс: новые направления в проектировании компьютерных систем. СПб.: Символ-Плюс, 2003. 272 стр.
2. Прэтт У. Цифровая обработка изображений. Т. 1. М.: "Мир", 1982. 310 стр.
3. Tamponi E. Communication between Xorg, Xgl, and an OpenGL client, through libGL and the GLX Protocol. 2006. <http://principe.homelinux.net>

УДК 004.514.62

**Борушко И.Н., Гоманова Е.В.**

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

#### **АНАЛОГИЯ ПЕРИФЕРИЧЕСКОГО ЗРЕНИЯ В ГРАФИЧЕСКОМ ИНТЕРФЕЙСЕ ПОЛЬЗОВАТЕЛЯ**

Периферическое зрение (ПЗ), называемое часто также боковым или палочковым зрением, играет важную роль в ориентировании человека в окружающей среде. Соглас-

но [1] зона ПЗ более чем вдвое перекрывает угол обзора прямого зрения

Световые лучи [2] от рассматриваемых предметов проходят через оптическую систему глаза (роговицу, хрусталик и стекловидное тело) и попадают на фоторецепторы сетчатки – колбочки и палочки. Основная масса колбочек сосредоточена в центральной части сетчатки, называемой желтым пятном. По мере удаления от него количество колбочек уменьшается, а число палочек возрастает, в результате чего на периферии сетчатки имеются только палочки. Колбочки, обладающие высокой разрешающей способностью участвуют в точном восприятии формы, цвета и деталей предмета. Палочки, имеющие малую разрешающую способность, как раз и обеспечивают ПЗ. Из-за их меньшей разрешающей способности форма предмета, находящегося в зонах ПЗ воспринимается менее четко. Таким образом, центральное зрение дает возможность рассматривать мелкие детали и опознавать предметы, а периферическое служит в основном для ориентирования в пространстве, обнаружения предметов и восприятия различных движений.

При работе с документами, которые представляют собой статические объекты, палочковое зрение играет вспомогательную роль, делая ориентацию в рабочей области более наглядной, уменьшая нагрузку на запоминание расположения документов и их элементов. Однако в динамически изменяющихся системах его роль возрастает. Так, при вождении автомобиля, т.е. в типичной динамической системе, боковой обзор и в некоторой степени отслеживание показателей приборов на приборной панели требуют от водителя развитого ПЗ, только благодаря которому он может в ряде случаев избежать аварийной ситуации.

Рабочая область, представленная графическим пользовательским интерфейсом (ГПИ), также является динамической системой основанной на событиях. В том числе это касается и таких типично «офисных» задач, как работа с документами и навигация в сети Интернет. Поэтому доступ ко всей площади виртуального рабочего пространства хотя бы с меньшей детализацией при организации ГПИ безусловно важен.

Человек, работающий одновременно с большим количеством документов, предпочитает располагать их на какой-либо доступной поверхности, имеющей большую площадь. Применение этого подхода в компьютерной технике в неизменном виде требует значительных материальных затрат. Так, использование дополнительных дисплеев (иногда в большом количестве) можно встретить в тех условиях, где это дает существенную выгоду, а затраты на аппаратную часть не играют главенствующую роль – например, в центрах наблюдения, интегрирующих потоки визуальной информации от значительного числа видеокамер. В задачах, характерных для персональных компьютеров этот подход не нашел широкого распространения. Хотя современные операционные системы позволяют использовать одновременно два дисплея, подключаемые к видео адаптеру и дающие возможность работать с виртуальной областью, увеличенной по крайней мере в два раза, число пользователей, задействующих данную технологию, невелико. Отчасти из-за дороговизны, отчасти из-за недостаточного удобства.

Анализ вспомогательных средств навигации, применяемых в ГПИ, показывает, что в основе многих из них в большей или меньшей степени заложена аналогия с ПЗ человека.

Традиционные решения включают в себя:

- минимизацию окон в пиктограммы (в привычном пользователю виде – панель задач),
  - виртуальный рабочий стол, больший, чем область экрана,
  - «пейджер» - программа, обеспечивающая доступ к нескольким виртуальным столам.
- Наиболее распространены первый и третий варианты, причем в ряде случаев они используются совместно.

В первом случае пользователь может «минимизировать» или «свернуть» окно, в результате чего оно не отображается на экране непосредственно, а вместо этого присутствует в специальной области в виде пиктограммы – мини-изображения, назначенного данному окну и соответствующей ему программе разработчиком. В панели задач при

существуют кнопки с пиктограммами для всех приложений: и минимизированных, и отображаемых в рабочей области. Кроме контейнера свернутых окон панель задач служит средством переключения между окнами приложений. Особенно актуально последнее для случая, когда окна перекрывают друг друга, и требуется получить доступ к окну, скрытому под соседними. Док (появившийся впервые в ОС «NextStep» и распространенный оттуда на некоторые графические интерфейсы, характерные для ОС семейства Unix) выполняет похожие функции, но предоставляет ряд дополнительных возможностей - например смена пользователем пиктограмм для различных окон, «пришвартовывание» пиктограмм к специальной области, служащей для быстрого либо автоматического запуска приложений, возможность разработчику программы выводить дополнительную информацию рядом с пиктограммой.

Пиктограммы минимизированных окон уже долгое время занимают место лидера среди средств увеличения виртуального рабочего пространства. Аналогия с ПЗ наблюдается в данном случае в наиболее чистом виде: схематичные изображения объектов, расположенные на периферии экрана соответствуют уменьшенной разрешающей способности в периферийных областях сетчатки. И, вероятно не в последнюю очередь популярность данных интерфейсных решений вызвана их привычностью. Однако вместе с очевидными достоинствами данный подход имеет два типичных недостатка: *малый размер* (по определению) соответствующей области экрана, что усложняет выбор при большом количестве элементов, а также *непригодность для динамического слежения* за состоянием минимизированного окна.

Действительно, хотя ряд разработчиков пытается манипулировать пиктограммой и/или соответствующей ей подписью для отображения происходящих в программе событий, тем не менее результат получается весьма ограниченным и не слишком наглядным. Особенно это касается информации, выводимой в подписи (дублирующей текст заголовка окна), которая сразу становится нечитаемой при увеличении числа пиктограмм на панели задач.

Виртуальный рабочий стол, превышающий по размеру видимое изображение и доступный пользователю по частям, через окно, снабженное средствами прокрутки - вероятно, наименее часто используемый способ расширения рабочей области. Нелюбовь пользователей связана со сложностью навигации, отсутствием наглядности (меньше возможностей задействовать визуальную память для поиска нужного окна) и несогласованностью с другими элементами навигации, такими как док или панель задач. Следует также отметить, что в данном подходе полностью игнорируется привычка человека рассчитывать на ПЗ.

Программы-пейджеры в настоящее время стали стандартным элементом ГПИ Unix-подобных систем и также доступны в качестве отдельных коммерческих продуктов для ОС семейства Windows. Принцип напоминает предыдущее решение. Рабочая область, в несколько раз большая видимого изображения, поделена на части, называемые страницами или виртуальными рабочими столами (последнее название более распространено в настоящее время): Пейджер показывает в отведенной ему области экрана схематичное изображение виртуальных рабочих столов (а обычно - еще и имеющихся на них окон).

Пейджер позволяет легко переключаться с одного рабочего стола на другой, а также «перетаскивать» окна приложений между рабочими столами.

Пейджер может быть самостоятельным программным продуктом, а может быть реализован и как часть менеджера окон. Второй вариант предполагает более легкую интеграцию с ГПИ, так как облегчается задача слежения за позицией и размером окон. Первый вариант требует от менеджера окон некоторого стандартного способа предоставления необходимой информации.

Недостатки пейджера – схематичность изображений окон, не позволяющая получить хотя бы отдаленное представление об их содержимом, слабая пригодность для отслеживания динамических событий.

Кроме описанных, всем рассмотренным решениям свойственны еще и два общих недостатка: *отсутствие прямой аналогии* с ориентацией человека в реальной среде, что требует некоторого обучения и привыкания, а также – *необходимость вспомогательных элементов* ГПИ для навигации, что несколько усложняет интерфейс.

Большинство экспериментальных ГПИ так или иначе основываются на представлении информации в трехмерном пространстве с целью расширения рабочего пространства пользователя «вглубь». При этом для пользователя удобство переключения между окнами важнее того, какое количество окон одновременно умещается на экране. В идеале от него должен требоваться только выбор нужного окна, как в традиционном двумерном интерфейсе.

На данный момент существует несколько решений интеграции трехмерной графики в ГПИ различных ОС. Ни одно решение пока не получило распространения в качестве законченного программного продукта, т.е. все рассмотренные ранее разработки являются в той или иной мере экспериментальными.

Для ОС семейства Windows были найдены четыре разработки.

**MaW3** – прототип трехмерного ГПИ, основанного на оконной системе Windows 95. Пользователь находится в начале трехмерного туннеля и его взгляд направлен вглубь. Окна могут быть расположены на произвольной глубине туннеля, и чем дальше они находятся от начала, тем меньше становятся. В дополнение есть возможность располагать некоторые или все окна на стенах туннеля для более эффективного использования рабочего пространства (поскольку окна при этом «сжаты» проекцией, в их размещении также может быть усмотрена аналогия с ПЗ). В правой части MaW3 находится аналог пейджера – панель, представляющая план туннеля (вид сверху) для быстрого переключения между окнами и изменения их положения относительно начала туннеля.

**Win3D** – продукт компании ClockWise Technologies, еще в 2001 году оформившей патент на систему для создания трехмерного графического интерфейса для ОС Windows. В настоящее время компания предлагает разработку трехмерных интерфейсов на заказ, модифицируя Win3D в зависимости от пожеланий клиента. Во всех случаях используется виртуальное трехмерное пространство, заполненное различными объектами (в ряде случаев копирующее интерьеры реальных помещений). При несомненной эстетической привлекательности, данные решения воспроизводят скорее игровую, чем рабочую среду.

**SphereXP** – похожий проект, более приближенный к нуждам оконного интерфейса. В качестве поверхности, на которой располагаются окна, выбран участок невидимой сферы большого диаметра. На читаемости текста трехмерность сказывается не лучшим образом. Кроме того, человек испытывает наибольший комфорт при работе с документами, расположенными на плоской поверхности, ортогональной зрительной оси (именно так устроены рабочие места в реальном мире).

**Aero Glass** (ранее - Avalon) – учитывает данное обстоятельство. Его разработчики не злоупотребили трехмерностью виртуального пространства, сосредоточившись на использовании аппаратных ресурсов видеоадаптера для улучшения дизайна. Результатом, однако, явился драматический рост системных требований (включая требования к видеоадаптеру), и даже некоторое ухудшение эргономики оконного интерфейса из-за выбранных дизайнерских концепций.

Под Unix-подобные операционные системы также существует ряд аналогичных решений. Эти решения базируются на стандартной среде ГПИ X Window System и открытой библиотеке OpenGL, и распространяются с исходным кодом, что заметно облегчает анализ технических особенностей их реализации.

**Looking Glass** разработка компании Sun Microsystems, переданная сообществу open source. Функционирует в одном из двух режимов: в виде самостоятельной рабочей среды в отдельном окне (доступен крайне ограниченный набор демонстрационных приложений) и в виде оконного менеджера. Окна в Looking Glass выглядят как полупрозрачные стеклянные пластины, внутри которых вставлено что-то напоподобие листка бумаги с содержимым окна. Окна приобретают фокус (становятся непрозрачными), если на них навести курсор и тут же его теряют, когда курсор оказывается за рамками окна. При щелчке мышью в рабочей области окна оно "всплывает" на передний план, становясь чуть крупнее других (остальные окна наоборот становятся чуть меньше). Как видно, в данном случае также упор сделан скорее на дизайнерские решения, чем на попытку оптимизировать ГПИ за счет дополнительных аппаратных ресурсов.

**Melisse** – проект, нацеленный на превращение рабочего стола X Window System в полноценное трехмерное пространство [3]. Перед пользователем предстают окна с дополнительными кнопками в заголовках, с помощью которых осуществляется свободный поворот окна в любом направлении. Если в Looking Glass все окна самостоятельно находят свои места в пространстве, то, работая с Melisse, пользователь должен сам регулировать угол поворота окна.

**Compiz** оконный менеджер, частично основанный на коде оконного менеджера Metacity. Структура представлена в модульном виде: возможности вынесены во внешние модули. Большая часть модулей предназначена для организации различных декоративных эффектов: прозрачность и тень, плавное появление и исчезновение окон и т.д. Однако есть несколько модулей, реализующие более интеллектуальные задачи. Например, с помощью модуля Scale при нажатии на заранее определенную клавишу все окна на текущем рабочем столе масштабируются до нужных размеров и выстраиваются так, чтобы все они были видны пользователю. При этом содержимое окна изменяется динамически, и если в нем происходят какие-либо события, это в реальном масштабе времени отражается в уменьшенной копии. Выбор окна мышью поднимает его поверх остальных и делает активным, а все остальные окна возвращаются на прежние позиции. Модуль Switcher также обеспечивает переключение между окнами по их уменьшенным изображениям. Следует отметить, что богатые возможности технологии использованы по большей части в декоративных целях (а некоторые декоративные эффекты, поражающая экстравагантным видом, в конечном итоге затрудняют работу).

Таким образом, можно говорить о программных способах увеличения размера рабочей области за счет виртуального изменения разрешающей способности отдельных областей экрана. Средства навигации, наиболее четко соответствующие модели ПЗ – панель задач и пейджер – демонстрируют наибольшую популярность среди пользователей. В то же время альтернативные интерфейсы, основанные на аппаратно-ускоренной трехмерной графике, в меньшей степени ориентированы на выполнение данной задачи. В своем большинстве они страдают от недостатка прагматического подхода, а при их разработке практически не учитываются соображения эргономики ГПИ. Данные обстоятельства сводят на нет потенциальные преимущества от использования пост-обработки изображения средствами графического процессора. Однако представляется весьма целесообразным и несложным с технической точки зрения применение аппаратного ускорения для уменьшения рассмотренных недостатков традиционных средств навигации – их малой динамичности, интуитивности и выразительности.

#### ЛИТЕРАТУРА:

1. Парфенов В. Дизайн аудио-видео среды. Оптические расчеты. 2004  
<http://www.homeav.ru/HomeCin/OptiCalc.html>
2. Прэтт У. Цифровая обработка изображений. М. «Мир», 1982

УДК 004.514.6

Гашко Р.В.

Научный руководитель: Костюк Д.А.

## ПРИМЕНЕНИЕ МОДЕЛИ ИНТЕРФЕЙСА КАЯ КРАЗУЭ ДЛЯ ОПТИМИЗАЦИИ ОФИСНЫХ ПРИЛОЖЕНИЙ

В работе рассмотрены проблемы организации интерфейса современных офисных программных пакетов на примере популярных текстовых процессоров. Предложен вариант оптимизации интерфейса с помощью применения модели Кая Краузе. Анализируются основные положения данной модели и их влияние на эффективность рассматриваемых приложений.

Среди проблем интерфейса современных офисных пакетов, исследователи обычно отмечают перегруженность пользовательского интерфейса (особенно это заметно в отношении инструментальных панелей), а также отсутствие решений, стимулирующих пользователя повышать свою квалификацию и, соответственно, выбирать более эффективные способы решения задач [1-2].

Современный текстовый процессор представляет собой сложный комплекс программ с огромным числом функций, каждую из которых разработчикам приходится делать доступной через графический интерфейс. Однако неоднократно проводившиеся наблюдения за пользовательской активностью показывают, что большая часть пользователей игнорирует такое многообразие возможностей, обходясь привычным минимумом, тем самым лишая себя возможности заметно облегчить и ускорить выполняемую работу. К сожалению, инерционность типичного пользователя отчасти поощряется главенствующим на рынке подходом, согласно которому менее используемые или требующие большей квалификации возможности приложения скрываются в менее заметных или более труднодоступных областях интерфейса. В результате приложение выглядит более простым и привлекательным, не уменьшая число возможностей программного продукта. Однако при этом не поощряется повышение пользовательской квалификации, и вероятность использования дополнительных возможностей приложения оказывается невысока.

В частности, в [2] анализируются приемы работы со стилями текста, диктуемые наиболее популярными текстовыми процессорами (рассмотрены Microsoft Word 2003 и OpenOffice Writer 2.0). Поддержка стилей реализована в любом современном текстовом процессоре. Стили позволяют пользователю один раз определить некоторый вариант форматирования текста, включая параметры шрифта, выравнивания, интервалов и др., а затем применять созданный стиль к различным блокам текста. Своим происхождением стили обязаны различным языкам разметки. Использование стилей в текстовом процессоре позволяет:

1. Упростить соблюдение единого стиля оформления документа,
2. Быстро изменить оформление документа, отредактировав стиль.

Как видно, выигрыш от использования стилей возрастает по мере увеличения объемов текста. Однако разработчики относят данный механизм к категории функций, предназначенных для квалифицированных пользователей. Как следствие осмысленное применение стилей не является обязательным, а средства их редактирования зачастую тщательно скрыты от глаз неопытного пользователя (Microsoft Word, Abiword, KWord, Corel WordPerfect и др.). Поэтому большинством пользователей данный механизм игнорируется, в результате чего, во-первых, после набора объемного текста приведение его к нужной схеме оформления оказывается трудоемкой задачей, а во-вторых, оказывают