

Здесь поле sig\_sw - флаг записи (0x80 - последний элемент набора; 0x40 - элемент свободен; 0x20 - режим сигнала IN; 0x10 - режим сигнала OUT; 0x08 - INOUT; 0x04 - снятие выходных реакций). Поле sig\_swf - флаг формата внутреннего представления данных, sig\_swvtv - флаг формата экранного представления данных, sig\_swe - флаг представления погрешности (см. dtds\_swe), sig\_idtxt - текстовый идентификатор сигнала, sig\_value - значение сигнала, sig\_value1 - значение сигнала (поле расширения), sig\_error - значение погрешности (см. dtds\_swe), sig\_error1 - значение погрешности (поле расширения). Здесь sig\_idtxt - предварительно определенная константа.

Таким образом, данные тестового вектора представляют из себя линейную область памяти, полностью определяемую структурой Описание\_ТНД. Набор ТНД есть матричная структура из элементов типа {< Префикс\_тестового\_вектора >, < Тестовый\_вектор >}. Длина элемента матричной структуры определена полем dtds\_recl, длина префикса тестового вектора полем dtds\_prfl. Соответственно начало тестового вектора с заданным номером задается смещением, которое определяется значениями соответствующих полей, например, как <Номер\_вектора> \* (<Длина\_вектора> + <Длина\_префикса\_вектора>), что может быть легко реализовано с использованием дескрипторных файлов. <Префикс\_вектора> включает поле <Номер\_такта> или <Время>, что позволяет на одной структуре реализовать как ТНДД так и набор данных, описывающих выходные реакции.

При реализации библиотеки необходимо также учитывать, что внутреннее представление форматов тестовых данных определяется в описании ТНД (поле sig\_swf). При поддержке экранного интерфейса выполняется преобразование данных согласно значению поля sig\_swvtv. При необходимости преобразования данных во время вызова моделирующей программы дополнительно может быть определена структура, аналогичная def\_dtds, но с замененными полями, что позволит с минимальными затратами обеспечить их преобразование.

**Заключение.** Сформулированы требования к организации хранения и обработки значений формирователей сигналов с учетом типовых задач, решаемых в ходе постановки имитационных экспериментов с проектами цифровых устройств, особенностей использования формирователей при моделировании. Применительно к языку

VHDL предложены форматы и структуры хранения информации в базе данных с учетом их привязки к проектам, портам, версиям архитектурной реализации. Это обеспечивает формирование структуры базы данных, ориентированной на поддержку и эффективное обеспечение задач моделирования.

Результаты создают основу для реализации соответствующих библиотек и для построения системы с автоматической настройкой модулей на интерфейс моделируемого проекта, что автоматизирует подготовку входных воздействий для проектируемого устройства, а в ходе моделирования - накопление и анализ реакций модели с выдачей результатов.

#### СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. VHDL для моделирования, синтеза и формальной верификации аппаратуры/Пер. с англ. - М.: Радио и связь, 1995. - 360 с.
2. Библио П.Н. Основы языка VHDL. - М.: "СОЛОН-Р", 2000. - 210 с.
3. Сергиенко А.М. VHDL для проектирования вычислительных устройств. - К.: "Корнейчук", 2003. - 208 с.
4. Армстронг Дж. Р. Моделирование цифровых систем на языке VHDL. - М.: Мир, 1992. - 175 с.
5. Поляков А.К. Языки VHDL и VERILOG в проектировании цифровой аппаратуры. - М.: "СОЛОН-Пресс", 2003. - 320 с.
6. Прихожий А.А., Муравьев Г.Л. Система проектирования цифровых СБИС с языком VHDL на ПП ЭВМ. Разработка и использование ПЭВМ// Труды международного симпозиума INFO' 89. - Мн., 1989. - т.2, ч.1. - 6 с.
7. Муравьев Г.Л. Автоматизированная система подготовки тестов // Сборник трудов научно-технической конференции "Новые технологии в машиностроении и вычислительной технике". - Брест: Брест. политехн. ин-т, 1996.
8. Дудкин А.А., Головкин В.А., Муравьев Г.Л. и др. Алгоритмы и подсистемы автоматизированного логического проектирования цифровых СБИС. Материалы по математическому обеспечению ЭВМ. - Мн., ИТК АН РБ, 1994. - 120 с.
9. Муравьев Г.Л., Шуть В.Н., Мухов С.В. Автоматизация построения имитационных моделей по их процессным испытаниям// Вестник БГТУ.- 2007.- № 5(47).- С. 39-42.

Материал поступил в редакцию 07.07.2008

MUHOV S.V., MURAVJOV G.L. Signal drivers library organization for digital device projects simulation

The approach to signal drivers library organization is considered. The library structure and data structures focused on realization by means of the programming languages of a high level are resulted.

УДК 004.514.62

Тавониус К.А., Костюк Д.А.

## ПРИМЕНЕНИЕ МАСШТАБНЫХ ПРЕОБРАЗОВАНИЙ ПИКТОГРАММ ДЛЯ УПЛОТНЕННОГО ОТОБРАЖЕНИЯ ФАЙЛОВОЙ СИСТЕМЫ

**Введение.** В последнее время все большее внимание уделяется попыткам использовать уменьшенный масштаб изображений, не находящихся в фокусе работы пользователя, для увеличения наглядности и интуитивности интерфейса [1]. Не в последнюю очередь оживление в данной области связано с ростом разрешающей способности дисплеев, делающей более информативным применение уменьшенных изображений объектов для предварительного просмотра (previews или thumbnails) - тумбнейлинга. Также, рост производительности систем позволяет добиться оптимального расчета положения и размера объектов. Ниже описана предпринятая нами попытка применения переменного масштабирования в контейнерах пиктограмм на примере файлового менеджера - типичного

приложения, интенсивно использующего данный вид элементов управления.

В последнее время масштабные преобразования начинают применяться в средствах навигации по файловой системе. Однако их использование ограничено одноуровневым тумбнейлингом - использованием сильно уменьшенных изображений или первых страниц документов вместо пиктограмм [2, 3].

Существует несколько аналогий (метафор), позволяющих эффективно представить пиктограммы переменного масштаба. В данной работе нами выбраны аналогии периферического зрения (ПЗ) и гравитационной аномалии (ГА).

Костюк Дмитрий Александрович, к.т.н., доцент кафедры «ЭВМ и системы» Брестского государственного технического университета.

Тавониус Кирилл Андреевич, студент факультета электронно-информационных систем Брестского государственного технического университета.

Беларусь, БрГТУ, 224017, г. Брест, ул. Московская, 267.

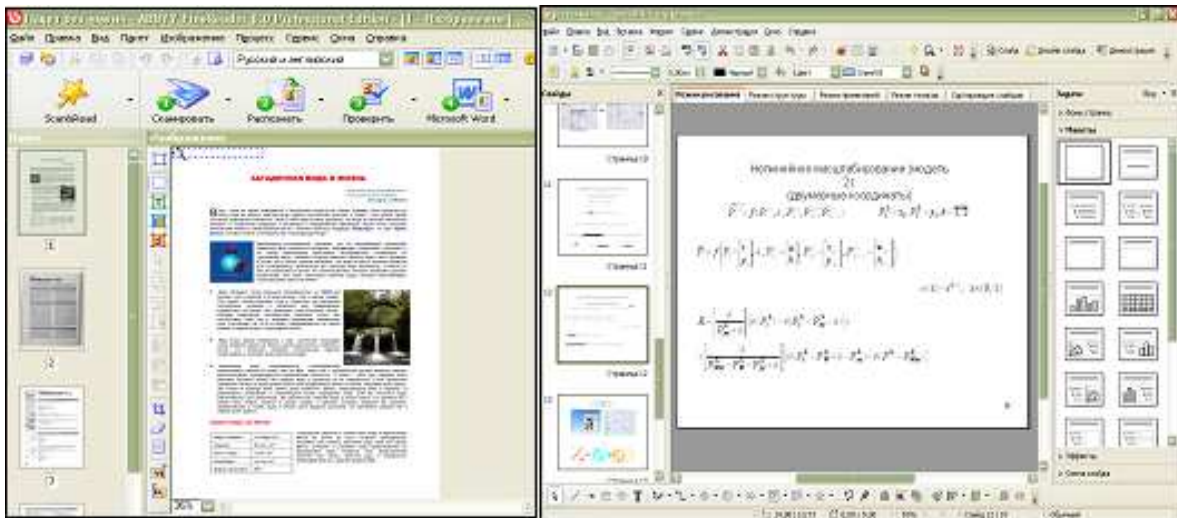


Рис. 1. Пример аналогии ПЗ в популярных программных продуктах

**1. Тумбнейлинг и аналогия периферического зрения.** Периферия сетчатки глаза характеризуется меньшей разрешающей способностью по сравнению с центральной областью, в результате чего человек хуже различает детали по краям поля зрения. Практика показывает, что использование аналогии ПЗ (размещение зон уменьшенной детализации в периферийных областях окна) повышает комфортность взаимодействия с интерфейсом программного продукта, делая его тем самым более конкурентоспособным на рынке ПО. Хотя напрямую тумбнейлинг часто не связывается дизайнерами интерфейсов с понятием ПЗ, характерные особенности модели ПЗ можно наблюдать в ряде популярных программных продуктов (рис.1). Интуитивность интерфейсов таких продуктов не вызывает проблем, т.к. задействуется природные способы ориентирования человека в пространстве [1].

Традиционным средством организации прокрутки, используемым в настоящее время всеми файловыми менеджерами, присущ ряд недостатков.

Согласно требованиям эргономики, полоса прокрутки (как наименее информативный элемент интерфейса) должна занимать минимум рабочего пространства и располагаться на его периферии. Однако это делает доступ к ней крайне невыгодным по времени, которое по закону Фитса [2] рассчитывается по формуле  $t = a + b \log_2(D/S + 1)$ , где  $D$  – расстояние от начальной позиции курсора до объекта;  $S$  – размер объекта вдоль линии перемещения курсора;  $a, b$  – константы, определяемые производительностью оператора. Кроме того, как элемент, расположенный на периферии окна, полоса прокрутки должна иметь хотя бы одну «бесконечную» размерность [2], что не может быть реализовано большей частью оконных менеджеров из-за постоянного присутствия рамки окна.

Попытка применения аналогии ПЗ в размещении пиктограмм файлов с переменным масштабом смоделировано нами в виде разделения рабочей области на три зоны: центральную 2 и периферийные 1 и 3 (рис. 3). В зоне 2, имеющей стандартное разрешение, расположены пиктограммы, с которыми пользователь работает непосредственно в данный момент. В зонах 1 и 3 разрешение пиктограмм понижается (соответственно увеличивается масштаб) в направлении от центра:

$$\Delta y = \begin{cases} C, y \in 2 \\ C \cdot \alpha(y), y \in 1, y \in 3 \end{cases}$$

где  $\Delta y$  – размер зерна (расстояние между центрами отдельных точек изображения),  $C$  – константа, определяющая единичный размер зерна, соответствующий области 2,  $\alpha(y)$  – функция пространственных искажений [4], монотонно возрастающая и имеющая область значений, принадлежащую интервалу (0; 1).

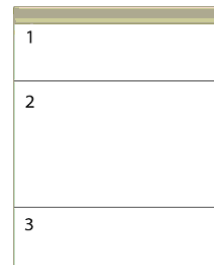


Рис. 2. Зонирование окна

В зоне 2 пиктограммы построчно размещаются в масштабе 1:1. Площадь пиктограммы в этом случае равняется  $M \times N$  (ширина и высота области вывода пиктограммы); таким образом, окно размером  $W \times H$  может вместить максимум  $(W/M)(H/N)$  пиктограмм. Если  $A$  (число пиктограмм для размещения) превышает данное, то  $A - (W/M)(H/N)$  пиктограмм попадает в дополнительные зоны 1 и 3.

В дополнительной зоне они размещаются в масштабе 1:2n, где  $n$  – номер строки. Нумерация строк начинается с единицы, и счёт идёт от зоны 2 (в зоне 1 – снизу вверх, а в зоне 3 – сверху вниз – см. рис. 3). Таким образом, в первой строке зоны 3 – размер пиктограммы равен  $(M/2)(N/2)$ , а в  $n$ -й строке зоны 3 –  $(M/2^n)(N/2^n)$ . На практике пиктограммы с размерами менее  $4 \times 4$  являются неинформативными. Поэтому для большей читаемости представляется целесообразным ограничить уменьшение размера пиктограмм, сведя правило пересчета для  $n$ -й строки к  $\max[(M/2^n)(N/2^n), (M/2^4)(N/2^4)]$ . При этом в  $n$ -ю строку попадает  $\min[2^n(W/M), 2^4(W/M)]$  пиктограмм. Аналогичная ситуация наблюдается с подписями к пиктограммам, традиционно содержащими имя файла, иногда сопровождаемое дополнительной сервисной информацией. Вывод текстовых подписей искусственно подавлялся при  $n > nT$  для более ясного восприятия содержимого окна. Значение  $nT$  выбирается так, чтобы усредненный коэффициент запечатывания знакоместа не превышал 70%.

После размещения пиктограмм проводится оценка на оптимальность. Критерием для проверки является разница между размерами дополнительной зоны и суммарными размерами области вывода пиктограмм в данной зоне. Если разница равна нулю, размещение оптимально. В противном случае итеративно увеличивается количество строк для каждого  $n$ , начиная со строк с меньшими номерами (рис. 3). Операция выполняется с увеличением  $n$  до получения оптимального размещения.

Прокрутка содержимого основной зоны осуществляется при использовании встроенных средств прокрутки мыши, или путем нажатия

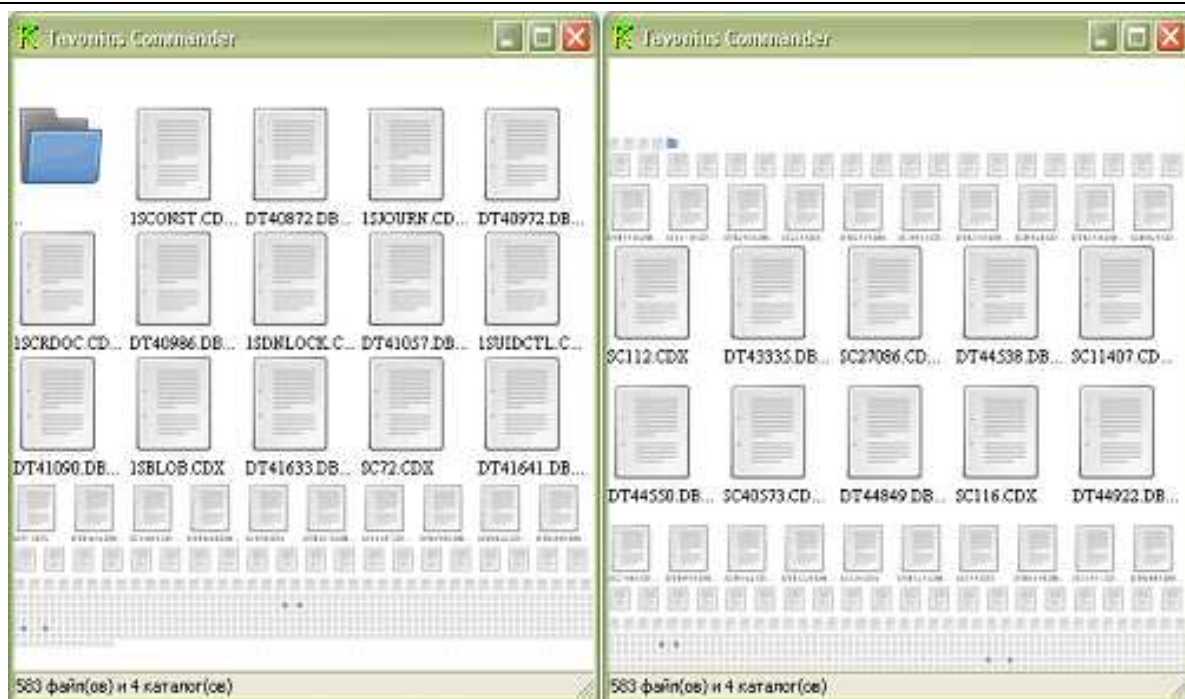


Рис. 3. Реализация прокрутки при большом количестве объектов

левой кнопки мыши в момент нахождения указателя в одной из дополнительных зон. При выполнении прокрутки содержимого основной зоны вниз последние  $W/M$  пиктограмм из основной зоны перемещаются в зону 3 и при этом приобретают масштаб 1:2. Соответственно, из зоны 1 последние  $W/M$  пиктограмм перемещаются в основную зону, принимая масштаб 1:1. После перемещения выполняется описанная выше проверка на оптимальность размещения пиктограмм.

При прокрутке в другую сторону имеет место обратная ситуация.

На рисунке 3 представлено приложение, разработанное для апробации представленного алгоритма. В качестве инструментария разработчика была использована среда быстрой разработки фирмы Borland (как удобный для быстрого прототипирования инструмент, находящийся «под руками»).

Потенциальные недостатки, присущие предложенной технологии, можно разделить на две категории: проблемы интуитивности использования пиктограмм переменного масштаба и проблемы производительности при их отрисовке. Если проблемы первой категории частично перекрываются естественностью восприятия аналогии ПЗ, позволяющей пользователю быстро освоиться в системе, то проблемы второй категории, особенно при работе с большим количеством файлов, вызывают вопросы.

На основе предложенного алгоритма было разработано тестовое программное обеспечение и выполнена оценка временных затрат, связанных с навигацией по файловой системе. Результаты тестов представлены на рис. 4. Как можно видеть, были апробированы два подхода к отрисовке пиктограмм – с использованием высокоуровневой библиотеки VCL/CLX и низкоуровневого интерфейса программирования ОС Windows GDI+. В эксперименте использовались пиктограммы размером 64x64 пиксела, со значениями  $M=N=80$ . Полигоном для экспериментов служила рабочая станция на основе процессора P4 2.0 ГГц.

Наблюдаемая зависимость потребления времени от количества подлежащих отрисовке пиктограмм близка к линейной. Следует отметить незначительность временных затрат на пересчет размещения пиктограмм по сравнению со временем отрисовки при использовании неоптимизированных средств вывода графики и их сравнимость с временем отрисовки на основе оптимизированного механизма GDI+. Согласно полученным данным, временной фактор не является значащим при определении целесообразности использования переменного масштаба пиктограмм.

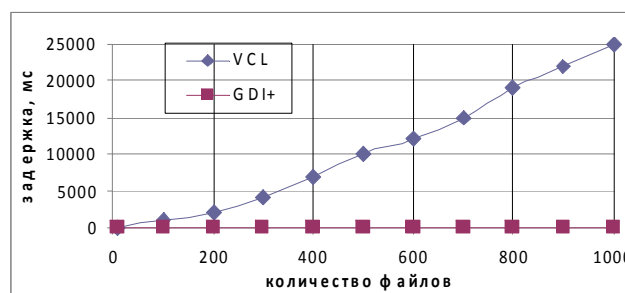
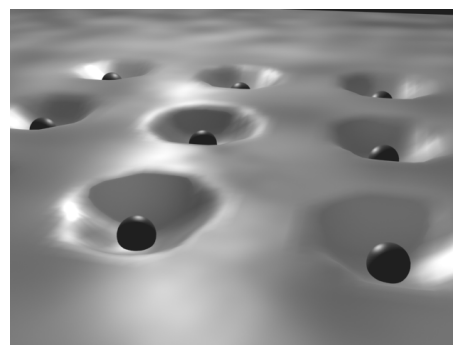


Рис. 4. Тест производительности алгоритма

**2. Тумбнейлинг и аналогия гравитационной аномалии.** Согласно рассматриваемой аналогии, масштаб пиктограммы  $S(W)$  является следствием деформации упругой ткани рабочего пространства (панели контейнера пиктограмм) под воздействием веса  $W$ , приходящегося на дискретную единицу площади. Всем пиктограммам контейнера присваивается единичный вес  $W_0$ , и в результате на каждом дискретном участке величина искривления поверхности определяется плотностью расположения пиктограмм в данной зоне, т.е. суперпозицией их весов  $\sum_{i=1}^N W_0$ . Визуально искривление поверхности представляется масштабом пиктограммы (глубиной тумбнейлинга).



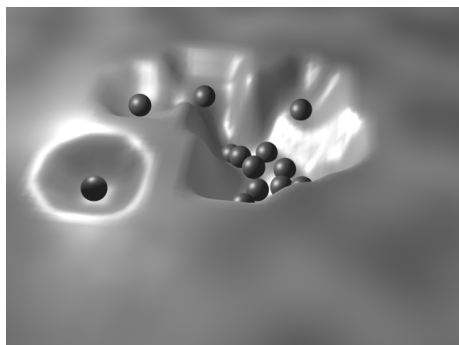


Рис. 5. Модель ГА

На рисунке 5 представлено схематичное изображение рассматриваемой аналогии. Для наглядности на иллюстрации деформация зоны расположения пиктограммы отображена не изменением ее масштаба  $s$ , а глубиной  $h$ , на которой пиктограмма расположена. При реализации же концепции ГА в графическом интерфейсе, пиктограммы оказываются одновременно видны благодаря переменному масштабу  $s(h)$ .

Предлагаемый подход также позволяет отказаться от использования полос прокрутки в контейнере. На рис. 6 представлено окно очередного тестового приложения, разработанного для апробации подхода. Как видно, для отображения ГА задействованы четыре позиции (число может варьироваться в зависимости от размеров контейнера). В рабочей зоне пиктограммы построчно размещаются в масштабе 1:1. Площадь пиктограммы в этом случае равняется  $M \times N$  (ширина и высота области вывода пиктограммы); таким образом, окно размером  $W \times H$  может вместить максимум  $(W/M)(H/N)$  пиктограмм. Если  $A$  (число пиктограмм для размещения) превышает данное, то число пиктограмм с единичным масштабом становится равным  $(W/M)(H/N) - 4$ , и, соответственно,  $A - (W/M)(H/N) + 4$  пиктограмм попадает в зону аномалии.

В приведенном примере аномалия отражает дополнительные объекты, из которых 12 имеют половинный масштаб, оказываются распознаваемыми, и таким образом доступны для выбора пользователем.

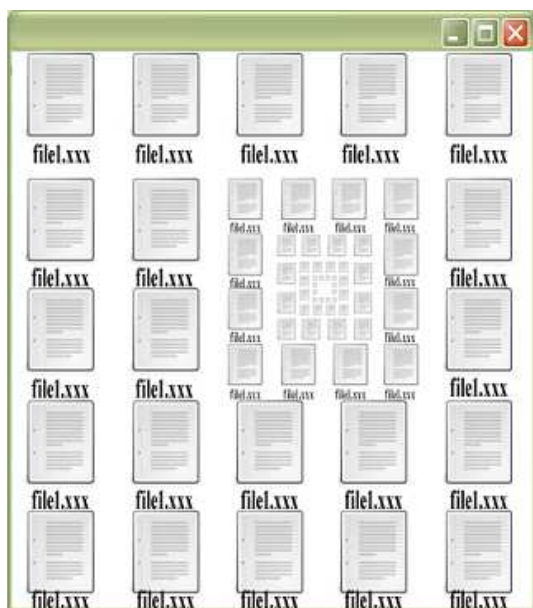


Рис. 6. Экспериментальная реализация ГА

Центр аномалии может использоваться для перемещения аномальной зоны в пределах контейнера. Если число  $A - (W/M)(H/N) + 4$  незначительно, перемещение аномалии позволя-

ет получить доступ ко всем элементам контейнера, однако при больших значениях этот способ неприменим.

Как видно, размещение объектов в зоне аномалии выполняется рекурсивно, и топологически каждый последующий уровень (глубина тумбнейлинга) повторяет предыдущий. Данный факт позволяет использовать панорамное масштабирование (ступенчатый «спуск» в аномалию) для доступа к находящимся в ней объектам. Благодаря кинематическим преимуществам повсеместно присутствующего в настоящее время колеса прокрутки данный метод представляется более перспективным с точки зрения пользовательского комфорта.

Поиск оптимальных алгоритмов для представления гравитационной аномалии выявил целесообразность использования структуры типа квадрантного дерева. Квадрантное дерево [5] позволяет представить двумерный объект в виде набора квадрантов различного размера путем рекурсивного деления исходного квадранта. Процесс деления иллюстрирует рис. 7. Как видно, объекты с номерами 1-5, 10, 19, 24 и выше отображаются в единичном масштабе, а объекты 6-9, 11, 16, 20, 23 и 29-30 – в половинном.

Сетка для отображения аномалии строится следующим образом. Создается исходный (корневой) квадрант, содержащий объект целиком внутри себя. Этот квадрант подразделяется на четыре квадранта путем деления каждой его стороны пополам. Затем квадранты классифицируются по положению относительно центра аномалии. Если квадрант прилегает к центру аномалии, он делится дальше. Процесс деления продолжается до тех пор, пока не будет удовлетворено требование по размещению всех пиктограмм.

На рис. 8 показан фрагмент квадрантного дерева, описывающего процесс деления и соответствующего аномалии, изображенной на рис. 7.

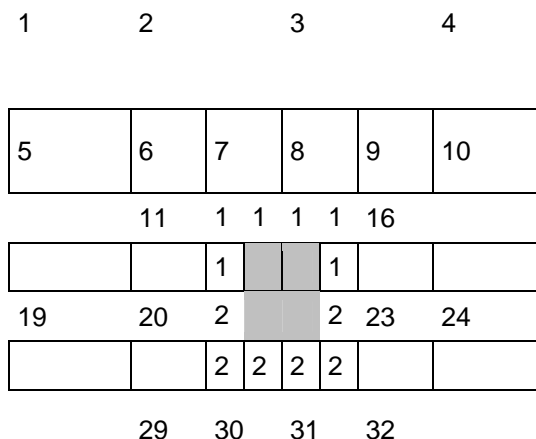


Рис. 7. Процесс деления на квадранты

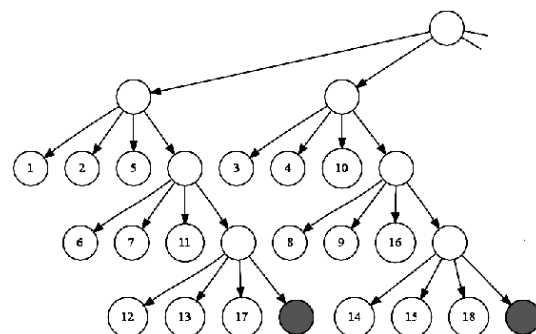


Рис. 8. Фрагмент квадрантного дерева

**СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ**

1. Борушко И.Н., Гоманова Е.В., Костюк Д.А. Применение модели периферического зрения в графическом интерфейсе пользова-

- теля. Современные информационные компьютерные технологии: сб. науч. ст. Гродно: ГрГУ, 2006. С. 22 – 27.
- Раскин Дж. Интерфейс: новые направления в проектировании компьютерных систем. СПб.: Символ-Плюс, 2003. 272 стр.
  - Тавониус К.А. Использование переменного масштаба пиктограмм для навигации по файловой системе. Современные проблемы математики и вычислительной техники: материалы V Республикан-

- ской научной конференции молодых учёных и студентов, Брест, 28-30 ноября 2007 г. – Брест: БрГТУ, 2007. С. 134 – 137.
- Прэртт У. Цифровая обработка изображений. Т. 1. М.: "Мир", 1982. 310 стр.
  - Yerry M.A., Shephard M.S. A Modified Quadtree Approach to Finite Element Mesh Generation. // IEEE Comput. Graph., and Appl., pp. 39-46, «Construction of Polyhedra and Geometric Modeling», CAD Group Document № 100, Cambridge University Computer

Материал поступил в редакцию 02.11.2008

**TAVONIUS K.A., KOSTIUK D.A. Scale transformation of icons for compact mapping of the filesystem**

A concept of variable scale of icons is proposed, which is copying the peculiarities of peripheral and perspective vision of humans and using the resolution of modern graphic interface. Two models are tested, the peripheral vision analogy and the one of gravity anomaly. A technical realization of spoken above models is proposed and partially tested. The ability to eliminate the latency increase of an interface caused by additional calculations for icons rescaling is demonstrated.

УДК 621.9.08

**Козак А.Ф., Костюк Д.А., Кузавко Ю.А., Петров Д.О.**

**РАСПРЕДЕЛЕННАЯ ЭЛЕКТРОННО-ИНФОРМАЦИОННАЯ СИСТЕМА МОНИТОРИНГА И ПРОГНОЗИРОВАНИЯ ПАВОДКА**

**Введение.** Защита территорий от наводнений важна с точки зрения комплексного использования и охраны природных ресурсов, является важнейшей социально-экономической и хозяйственной проблемой. Особенно ощутимо, а в отдельные годы катастрофично, влияние паводков в Республике проявляется в пойме р. Припять и ее притоках. Проведение противопаводковых мероприятий позволяет сокращать фактический ущерб от наводнений, однако требует значительных затрат и эксплуатационных расходов. Поскольку полностью исключить наводнения в Беларуси невозможно, первоочередная задача состоит в том, чтобы максимально приспособить хозяйственную деятельность к возможным экстремальным условиям, и тем самым минимизировать наносимый урон [1, 2].

В рамках решения этой проблемы нами разрабатывается распределенная программно-аппаратная система наблюдения и прогнозирования наводнений.

Хотя система разрабатывалась с ориентацией на контроль паводка в пойме р. Припять, однако легко перестраиваемая конфигурация сети АГУ позволяет максимально достоверно прогнозировать наводнения в бассейне любой реки, при наличии точек расположения АГУ, а также необходимых статистических и топографических данных [3].

**1. Мониторинг паводка.** В состав системы (рис. 1) входит единый информационный центр (ЕИЦ), обрабатывающий поток данных, поступающих с сети автономных гидрологических устройств (АГУ) измерения скорости течения реки и ее уровня, располагаемых в средних точках русла бассейна реки.

В качестве среды передачи данных от АГУ к ЕИЦ используется GSM-сеть [4]. ЕИЦ представляет собой вычислительный сервер, накапливающий информацию, поступающую по каналу сотовой связи, а также рассчитывающий и отображающий степень затопления поймы реки. Кроме результатов измерений АГУ для расчета могут быть применены оцифрованные статистические данные гидрологических измерений, проводившихся до разработки системы, математические модели движения водных потоков и грунтовых вод, а также оцифрованные трехмерные карты рельефа.

Конструкция АГУ включает [4] ультразвуковые пьезопреобразователи (УЗП), микроконтроллер, GSM-модем, системы питания и усиления сигнала.

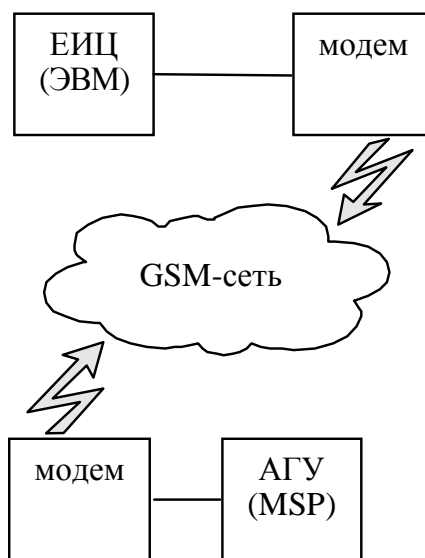


Рис. 1. Система мониторинга паводка

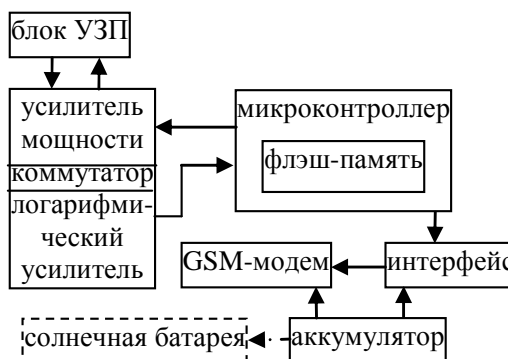


Рис. 2. Структура автономного гидрологического устройства

**Козак Александр Федорович**, ст. преподаватель кафедры «ЭВМ и системы» Брестского государственного технического университета.  
**Петров Дмитрий Олегович**, ассистент кафедры «ЭВМ и системы» Брестского государственного технического университета.  
 Беларусь, БрГТУ, 224017, г. Брест, ул. Московская, 267.  
**Кузавко Юрий Алексеевич**, к.ф.-м. н., с.н.с. Полесского аграрно-экологического института НАН Беларуси.  
 Беларусь, ПАЭИ НАН Беларуси, 224020, г. Брест, ул. Московская, 204.