Российская академия наук
Сибирское отделение
Институт систем информатики
им. А. П. Ершова

З. В. Апанович

ОТ РИСОВАНИЯ ГРАФОВ К ВИЗУАЛИЗАЦИИ ИНФОРМАЦИИ

Препринт
148

Новосибирск 2007

Методы визуализации информации широко применяются последние годы в таких областях как биологические науки, искусственный интеллект, анализ финансовой информации и т.д. Визуализация информации – это процесс преобразования больших и сложных видов абстрактной информации в визуальную форму. Универсальным средством представления такой информации являются графы. Данная работа имеет целью показать эволюцию методов визуализации графов в связи с потребностями визуализации информации и имеет обзорный характер.

Siberian Division of the Russian Academy of Sciences
A. P. Ershov Institute of Informatics Systems

Z. V. Apanovich

FROM GRAPH DRAWING TOWARDS INFORMATION VISUALIZATION

Preprint
148

Novosibirsk 2007

Last years methods of Information Visualization are getting more and more popular in such areas as biological sciences, artificial intelligence, business analytics, etc. Information Visualization is a process of transformation of large and complex abstract forms of information into visual form, strengthening user's cognitive abilities and allowing them to take the most optimal decisions. Graphs and trees are widely used for abstract information representation. The aim of this paper is to show evolution of graph drawing methods with respect to Information Visualization needs.

1. ВВЕДЕНИЕ

Визуализация информации имеет дело с размещением и интерактивным просмотром *абстрактных данных*, не имеющих явной физической интерпретации, часто очень больших, полуструктурированных или многомерных. Приложения визуализации информации возникают в таких областях, как информационные системы и программное обеспечение, биологические науки, искусственный интеллект, анализ финансовой информации и компьютерное ообучение. Универсальным средством представления абстрактной информации являются графы, поэтому методы визуализации графов и деревьев представляют собой теоретическую основу методов визуализации информации.

Наукоемкие продукты, использующие методы визуализации информации, существуют на мировом рынке в течение последних 10–15 лет. Имеются фирмы, поставляющие библиотеки и программные комплексы, ориентированные на визуализацию графов общего назначения (*Tom Sawyer software, ILOG software, Algorithmic Solutions Software GmbH, yWorks*), а также системы визуализации программного обеспечения (*Imagix Corporation, Absint и др.).* Вместе с тем появляется все больше фирм, специализирующихся на визуализации бизнес-информации, необходимой аналитикам различных предприятий и ориентированной на профиль тех или иных предприятий (Enterprise Solutions). Одной из старейших компаний этого направления является фирма *Inxight Software, Inc*, поставляющая средства визуализации информации для финансовых и биологических фирм. Наконец, совсем недавно появились компании, поставляющие на рынок продукты, использующие новые методы из области визуализации информации, так называемые методы *визуальной аналитики*. Среди растущего семейства инструментов визуальной аналитики процветает программное обеспечение фирмы *HiveGroup*, использующее визуализацию иерархических данных на основе *карты дерева* (Treemap) и поставляемое организациям, которым требуется ежедневный мониторинг сложной деятельности с участием тысяч продуктов, проектов и продавцов. Компания *Advanced Visual Systems (AVS)* поставляет на рынок продукты, позволяющие топ-менеджерам корпораций получать единую картину их бизнеса, а ведущая корпорация в области сервис-ориентированной архитектуры *TIBCO* приобрела недавно компанию *Spotfire* вместе с ее платформой Enterprise Analytics.

Данная работа имеет целью показать эволюцию методов визуализации графов в связи с потребностями визуализации информации и имеет обзорный характер.

2. ЗАДАЧА ПОСТРОЕНИЯ ИЗОБРАЖЕНИЯ ГРАФА

Стандартная формулировка задачи построения изображения графа очень проста: дан граф $G = (V, E)$. Требуется построить изображение его вершин и ребер на плоскости.

Такая задача существовала задолго до того, как появилось научное направление под названием Graph Drawing, и даже раньше чем Computer Science. В частности, возникновение направления «планарные графы» в теории графов связано с задачей «Можно ли построить на плоскости изображение графа без пересечений ребер?». Известно также, что еще в Древнем Риме было принято изображать генеалогические деревья в домах знатных римлян.

Тем не менее, возникновение *самостоятельного* направления Graph Drawing принято связывать с Первой Международной конференцией, которая прошла в Риме в 1992 году.

Аннотированная библиография *G. Di Battista, P. Eades, R. Tamassia, I.G. Tollis* [6] содержит ссылки на сотни статей, предлагающих различные классификации изображений и рассматривающих вопрос о том, что такое хорошее изображение графа.

Изображения графов можно классифицировать, например, в соответствии с *типом изображаемого графа*. Принято рассматривать отдельно изображения деревьев, изображения планарных графов, графов ориентированных и неориентированных.

Вторым общепринятым способом классификации изображений является классификация по *типу получаемого изображения***.** Список основных типов изображений графов, рассмотренных в аннотированной библиографии *G. Di Battista* et al. содержал следующие типы:

- прямолинейные,
- полилинейные,
- сетчатые,
- ортогональные,
- восходящие (нисходящие),
- плоские.

Позднее к этой группе присоединились круговые и радиальные изображения. При построении изображения графа принято руководствоваться *эстетическими критериями*. Примерами эстетических критериев являются:

- минимизация площади,
- минимизация количества пересечений ребер,
- минимизация количества сгибов ребер,
- минимизация количества наложений ребер и вершин,
- максимизация угловой резолюции,
- минимизация длины ребер (иногда – требование одинаковой длины),
- максимизация симметричности изображения,
- максимизация коэффициента формы (или: возможность управления коэффициентом формы).

Проблемой является то, что часто невозможно удовлетворить все критерии одновременно, и известно много примеров их взаимопротиворечивости. Поэтому возникают различные стили или методологии изображения, зависящие от конкретного приложения и доминирующего в этом приложении типа графа. Наиболее широко распространенными способами изображения графов на данный момент являются методы рисования неориентированных графов на основе физических аналогий [14, 21] и круговые алгоритмы [41], иерархические или поуровневые методы [44] для изображения ориентированных графов, а также ортогональные методы [45], применимые к неориентированным разреженным графам. Эти доминирующие типы изображений показаны на рис. 1.

Существует также отдельная группа общепринятых статических изображений деревьев, в которую входят такие изображения, как поуровневые, радиальные, круговые, hv-изображения и полилинейные изображения. Примеры стандартных изображений деревьев показаны на рис. 2.

3. ВИЗУАЛИЗАЦИЯ ИНФОРМАЦИИ НА ОСНОВЕ ВИЗУАЛИЗАЦИИ ГРАФОВ

Выделение *визуализации информации* в качестве самостоятельного направления связано с выходом в 1999 году книги под редакцией Stuart K. Card, Jock D. Mackinlay, и B. Shneiderman «Readings in Information Visualization: Using Vision to Think» [11]. Это направление объединило специалистов, труды которых до недавнего времени были разбросаны по та-

ким направлениям, как поиск информации, цифровые библиотеки, человеко-компьютерное взаимодействие, гипертекст и Интернет. С 2002 года издается журнал Information Visualization, в рамках ежегодной международной конференции IEEE InfoVis проводятся конкурсы программных систем, а с 2006 года проходит также и международный симпозиум IEEE VAST, посвященный научным и технологическим проблемам визуальной аналитики.

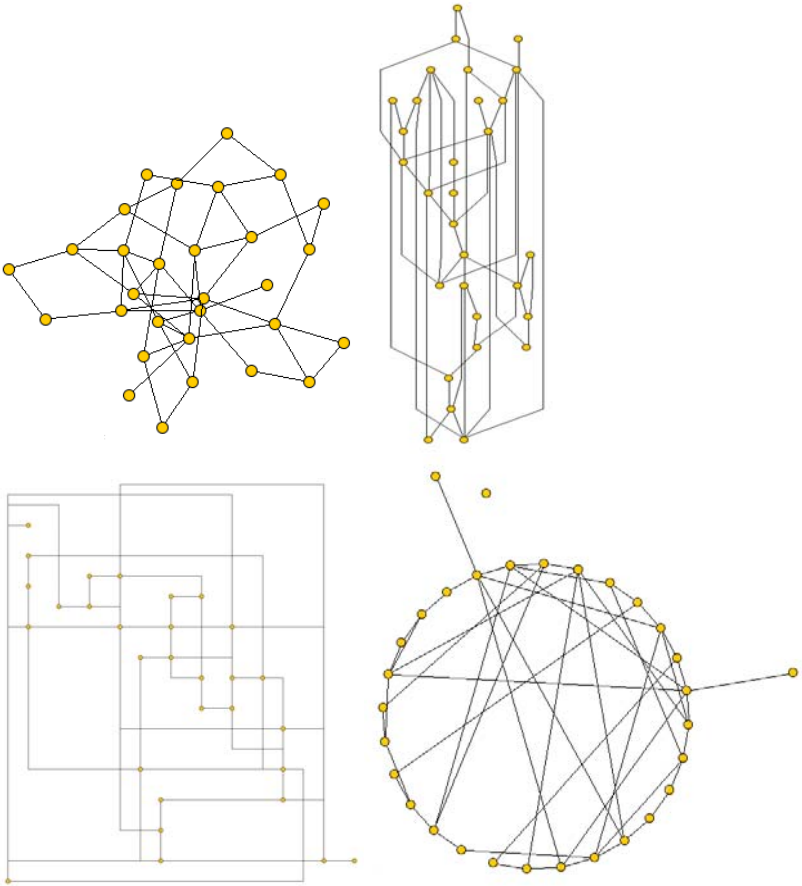


Рис. 1. Общепринятые статические изображения графов

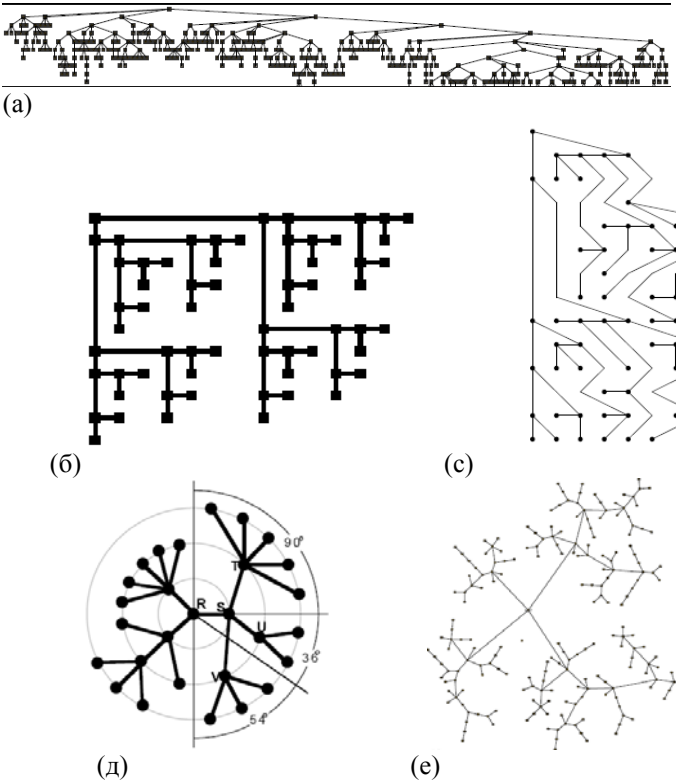


Рис. 2. Общепринятые статические изображения деревьев:
а) поуровневое, б) hv-изображение, с) полилинейное, д) радиальное, е) круговое

Одним из главных критериев оценки качества методов визуализации информации является прежде всего адекватность изображения заданному типу информации. В случае визуализации географических карт или карт дорог желательно, чтобы расположение вершин и ребер соответствовали географическим реалиям. При работе с различными таксономиями часто желательно ортогональное расположение ребер и меток, так как взаимодействие с изображением должно облегчать чтение сопутствующей информации; в то же время требование ортогональности не так важно для приложений, связанных с поиском шаблонов, например при визуализации телефон-

ных звонков или информации о доступе пользователей к различным сайтам.

При оценке качества алгоритмов или систем визуализации информации принято базироваться не столько на эстетических критериях, сколько на потребностях реальных пользователей. Для целей сравнения систем оценивается скорость и качество выполнения различных заданий группами пользователей при помощи систем визуализации [35–37]. С этой целью разрабатываются постоянно пополняемые таксономии заданий [42].

При визуализации информации возникают специфические технические проблемы. Одной из основных проблем является *объем изображаемых данных*. Алгоритм, который хорошо размещает несколько сотен вершин, совершенно необязательно будет так же хорош при работе с несколькими тысячами вершин. Существует не так уж много систем, которые действительно способны визуализировать очень большие графы (NicheWorks, H3Viewer, TopoLayout) [5, 30, 49]). К тому же размещение большого количества информации не всегда может быть полезным. Иногда это может ухудшать понимаемость и читаемость изображения.

Ключом к уменьшению количества деталей, размещаемых одновременно, является переход от статических методов размещения к интерактивным с применением навигации и методов фокус+контекст, включающих геометрическую или семантическую деформацию, кластеризацию, агрегацию и другие техники. При разработке интерактивных визуализаций существенной становится *временная сложность* алгоритмов Возникает необходимость в разработке алгоритмов, сложность которых близка к линейной. Также появляется новый критерий качества для динамических и интерактивных алгоритмов, называемый *предсказуемостью* или *сохранением ментальной карты*. Этот критерий требует, чтобы два разных прогона одного и того же алгоритма на одних и тех же или похожих данных давали похожий результат.

Далее будут рассмотрены две группы методов, характерных именно для визуализации информации, на примере иерархических структур данных, представимых деревьями. Считается, что хорошая визуализация иерархической информации помогает пользователю

- быстро находить нужный элемент в иерархии,
- понимать отношение элемента к его контексту,
- обеспечивать возможность прямого доступа к информации при вершинах.

Будет показано, как помогают решению этих задач методы типа *фокус+контекст* и методы визуализации численных атрибутов на основе *карты дерева*.

4. МЕТОДЫ ИЗ ГРУППЫ ФОКУС+КОНТЕКСТ

Методы фокус+контекст предназначены для взаимодействия с изображениями большого объема и позволяют совместить в одном изображении глобальный вид всей структуры и детали некоторого фрагмента, находящегося в фокусе.

Эти методы с переменным успехом применялись к большому количеству различных структур данных, включая графы, деревья, меню, календари, табличные данные и т.д. Одним из самых известных приложений является TableLens [38], пользовательский интерфейс для работы с табличными данными, выведенный на рынок фирмой Inxight в 2003 году. Полезность этого подхода для пользователей продемонстрирована экспериментально также на примере иерархических кластеризованных сетей [40].

Одним из первых примеров компьютерного метода фокус+контекст было геометрическое искажение, имитирующее линзу «рыбий глаз», введеное в 1982 году и называемое первоначально бифокальным дисплеем. Базовая идея состояла в искажении информационного пространства таким образом, чтобы элементы, находящиеся в фокусе, увеличивались в размере, а элементы на периферии - уменьшались.

4.1. Фокус+контекст на основе фильтрации вершин дерева

В 1986 году Furnas [15] описал класс методов, в которых вершины иерархической структуры автоматически включаются или удаляются из изображения в соответствии вычисляемой *степенью пользовательского интереса* (DOI). Он создал системы для просмотра и фильтрации структурированного программного кода, биологических таксономий и календарей, а также экспериментально продемонстрировал удобство этих динамические изображений для пользователя.

В модели Furnas функция *СтепеньИнтереса*(v) сопоставляет каждому элементу изображаемой структуры число, которое указывает, насколько пользователь может быть заинтересован в том, чтобы увидеть данный элемент. Первоначально пользователю предоставляется некоторое изображение, на котором он может выбрать одну из вершин в качестве фокусной.

При этом функция степени интереса для всех остальных вершин вычисляется по формуле:

СтепеньИнтереса(*v*) = *Априорная_значимость*(*v*) + *Расстояние_до_Фокуса*(*v*),

где *АприорнаяЗначимость*(*v*) – это априорная оценка важности вершины (*v*).

В случае, когда изображаемая структура является деревом, априорная значимость может вычисляться как расстояние от вершины v до корня дерева. Такой выбор априорной значимости хорошо соответствует, например, понятию важности того или иного индивидуума в иерархической структуре организации. Значение *Расстояние_До_Фокуса* соответствует в модели Фурнаса длине единственного пути от вершины *v* до фокусной вершины. После этого выбирается пороговое значение, и в изображение, выдаваемое на экран, включаются только вершины, *СтепеньИнтереса* которых превышает заданный порог (рис. 3).

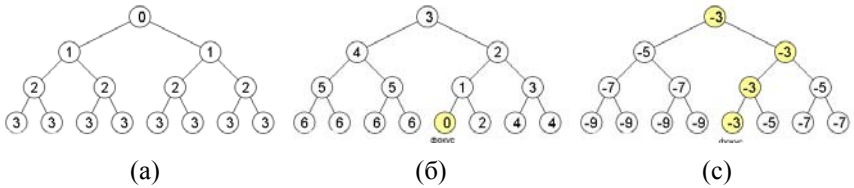


Рис. 3. Пример подсчета значения функции степени интереса:
(а) – расстояние до корня, (б) – расстояние до фокусной вершины,
(с) – окончательное значение функции степени интереса

В момент, когда пользователь выбирает в качестве фокусной любую другую вершину дерева, происходит перевычисление *СтепениИнтереса* для всех вершин, после чего из изображения удаляются вершины, *СтепеньИнтереса* которых оказалась ниже заданного порога, но зато в изображении появляются новые ветви, для вершин которых *СтепеньИнтерес*а возросла настолько, что превысила заданный порог. Таким образом изображение дерева изменяется вслед за изменяющимся интересом пользователя.

В работе Фурнаса было показано, что функция *СтепеньИнтерес*а, определенная для деревьев, обладает свойствами монотонности и выпуклости, что позволяет эффективную реализацию алгоритмов интерактивного размещения вершин дерева на основе введенной функции.

4.2. Обобщение подхода на случай графов

Заметим, что достаточно легко обобщить на случай произвольного графа первую компоненту функции *СтепеньИнтереса*. В качестве значения *Расстояние_до_Фокуса* можно использовать длину кратчайшего пути по графу от вершины до фокуса, а также расстояние в любой другой метрике, например, в евклидовой. Более сложной задачей являеется выбор значения *Априорная_Значимость* вершин, так как неудачный выбор этой функции может привести к потери связности изображаемого графа.

В 1992 году Sarkar и Brown [39] обобщили подход Фурнаса на случай произвольного графа, предложив использовать в качестве функции *Априорная_Значимость* один из численных атрибутов, связанных с каждой вершиной, а в качестве функции *Расстояние_до_Фокуса* – евклидово расстояние между вершинами уже существующего изображения.

В формализм рыбьего глаза были добавлены понятия *нормального изображения* и *деформированного изображения «рыбий глаз»*, таким образом, что *позиция*, *размер и уровень детализации* изображаемых вершин вычислялись на основе выбираемой пользователем функции искажения, расстояния объекта до фокуса и предписанной важности объекта в глобальной структуре.Предложенная ими модель выглядит следующим образом. Начальное изображение графа называется *нормальным изображением*, а позиции вершин этого изображения имеют *нормальные позиции* $P_{norm}(v)$. Каждая вершина помимо позиции имеет *размер* $S_{norm}(v)$, который соответствует длине стороны охватывающего прямоугольника вершины v. Также, каждой вершине v соответствует число, характеризующее ее априорную значимость . Координаты графа в искаженном изображении называются *координатами рыбьего глаза* P_{feye}. Помимо возможности удаления и добавления элементов в изображение структуры, Саркар и Браун ввели возможность управления размером и уровнем детализации изображения вершин в соответствии со следующими правилами:

1. Позиция вершины v в рыбьем глазе зависит от ее позиции в нормальном изображении и расстояния до фокуса f:

$P_{feye}(v, f) = F_1(P_{norm}(v), P_{norm}(f))$.

2. Размер вершины v в рыбьем глазе $S_{feye}(v, f)$ зависит от ее размера в нормальном изображении $S_{norm}(v)$, расстояния до фокуса $P_{norm}(f)$, и априорной степени интереса $API(v)$):

$S_{feye}(v, f) = F_2(S_{norm}(v), P_{norm}(f), API(v))$.

3. Уровень деталей, которые размещаются вместе с изображением вершины v в деформированном изображении $DTL_{feye}(v, f)$ зависит от размера

вершины после деформации $S_{feye}(v, f)$ и максимальной детали, которая может быть изображена:

$DTL_{feye}(v, f) = F_3(S_{feye}(v, f), DTLmax(v))$.

4. Наконец, решение о том, появится ли данная вершина в деформированном изображении, принимается на основе функции визуальной значимости $VW(v, f)$, зависящей от априорной значимости вершины v $API(v)$, и от расстояния между вершиной v и фокусом в нормальном изображении:

$VW(v, f) = F_4(P_{norm}(v), P_{norm}(f), API(v))$.

Большое семейство различных реализаций можно получить, выбирая по своему усмотрению функции F_1 - F_4.

В реализации Саркара и Брауна генерация деформированного изображения происходит в 2 шага. Сначала функция искажения применяется к нормальному изображению, с тем, чтобы вычислить координаты вершин в зависимости от расстояния до фокуса, а затем используется априорная важность каждой вершины для вычисления ее окончательного размера, уровня детализации и визуальной значимости.

В качестве функции искажения было предложено использовать функцию $G(x)$, которая монотонно возрастает и непрерывна на интервале $0 \le x \le 1$, при этом $G(0) = 0$, $G(1) = 1$.

Эксперименты осуществлялись с функцией $G(x) = (d + 1)x /(dx + 1)$, где константа d называлась *коэффициентом искажения.* На рис. 4 показан график функции $G(x)$ при значениях коэффициента искажениях $d = 0$ и $d = 5$. Если $d = 0$, то $G(x) = x$, поэтому изображение остается неизменным, в то время как при значении $d = 5$, наклон функции $G(x)$ вблизи фокуса становится больше единицы, а более отдаленные сегменты имеют наклон меньше единицы.

На рис. 6 показан пример применения геометрической деформации и семантической фильтрации применительно к карте США. Вершины изображают города, а ребра – дороги между этими городами. На рис. 6(а) показано нормальное изображение, а на рис. 6(б) – изображение «рыбий глаз», с коэффициентом искажения $d = 5$. В качестве функции *Априорная_Значимость* использовалась величина населения каждого города, а пороговое значение визуальной значимости было установлено равным 0.2 от максимального по всем городам населения.

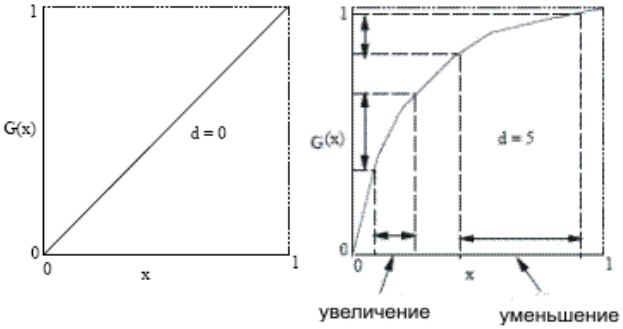


Рис. 4. График функции $G(x)$ при значениях коэффициента искажения $d = 0$ и $d = 5$

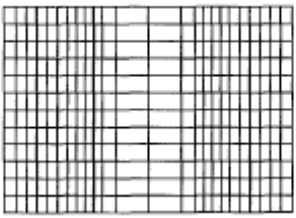
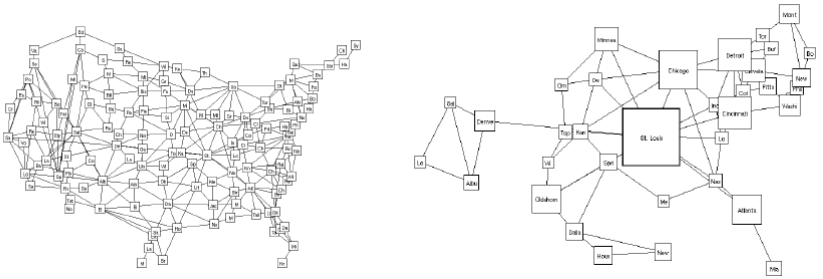


Рис. 5. Одномерное искажение изображение, получаемое при помощи функции $G(x)$



(а) (б)

Рис. 6. Пример нормального изображения и изображения «рыбий глаз» на примере карты США

При экспериментальной эксплуатации системы выяснилось, что применительно к географическим картам, декартова трансформация выглядела не очень естественно. Поэтому был предложен и второй тип деформации, так называемое круговое искажение, основанное на системе полярных координат с центром в фокусе.

Данная работа породила череду экспериментов с различными техниками искажения, которые подробно описаны в работе Y. K. Leung, M. D. Apperley [27], а в последнее время многофокусная версия данного алгоритма весьма успешно применяется для визуального сравнения филогенетических деревьев [43].

4.3 Геометрические искажения на основе гиперболической геометрии

Вышеописанная техника не зависит от алгоритма размещения и является отдельным шагом обработки при построении изображения графа. Взаимодействие с «рыбьим глазом» осуществляется изменением фокусной точки и/или изменением значения коэффициента искажения. Эта независимость имеет положительные и отрицательные моменты. Положительно то, что она позволяет модульную организацию программного обеспечения, в котором «рыбий глаз» – это отдельный шаг обработки, выполняемый между модулем размещения и реальной отрисовкой. При таком подходе построение искажения «рыбий глаз» может строиться существенно быстрее, чем работает алгоритм размещения, что важно для интерактивности. С другой стороны, возможно нарушение ограничений, задаваемых эстетическими критериями, например, возможно появление пересечений ребер, которых не было в нормальном изображении.

В качестве альтернативного варианта можно встроить возможности геометрического искажения непосредственно в алгоритм размещения, объединяя шаги размещения и деформации. Примером такого подхода является *StarTree* – гиперболический браузер, реализованный, запатентованный и выпущенный на рынок фирмой Inxight [23].

В этой программе функция визуализации реализована так, что фокусная вершина всегда размещается в центре окружности. Первоначально в центре окружности находится корень дерева, но размещение может быть гладко трансформировано таким образом, что в фокусе окажется любая другая выбранная вершина. Выбранная вершина перемещается в центр дисплея и увеличивается в размере, а вершины, удаленные от фокуса, уменьшаются в размере и размещаются ближе друг к другу. Изображение всегда остается в заданных границах, но только ограниченное количество связей из фокусной

вершины может быть видимо. Во всех случаях пространство, выделяемое каждой вершине является непрерывной функцией расстояния по дереву до вершины, находящейся в центре. Поэтому видимый контекст всегда включает несколько поколений родителей, братьев и сыновей, упрощая пользователю просмотр иерархии без опасения в ней заблудиться (рис. 7).

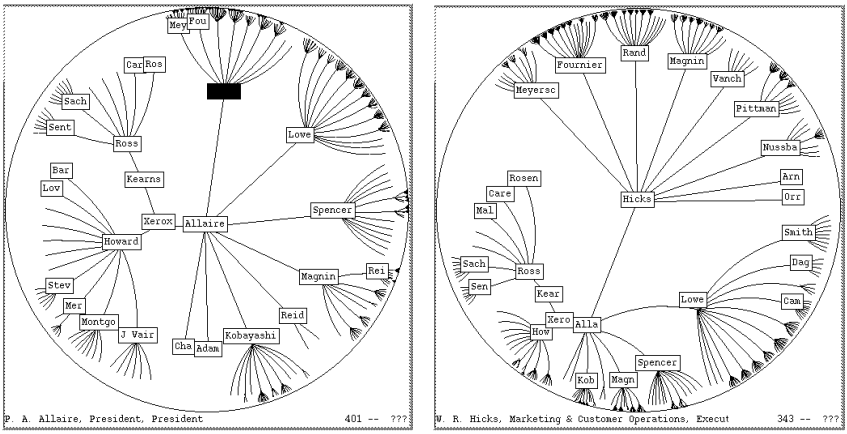


Рис. 7. Пример того, как при изменении фокусной вершины меняется гиперболическое изображение дерева

Данный подход использует свойства гиперболической геометрии. Основным достоинством этой техники является то, что такой браузер поддерживает эффективное взаимодействие с гораздо большими по объему иерархиями, чем стандартные браузеры. Это было продемонстрировано в 1997 году во время соревнования различных навигаторов, организованного в рамках конференции по человеко-компьютерному взаимодействию [29].

С математической точки зрения иерархия размещается на гиперболической плоскости, а затем эта плоскость отображается на круговую область дисплея. На гиперболической плоскости действуют законы неевклидовой геометрии, в которой параллельные линии расходятся в разные стороны, а длина окружности растет экспоненциально по отношению к радиусу ($C = 2\pi \sinh(r)$). Это дает два замечательных асимптотических свойства:

1) для небольшого радиуса пространство выглядит плоским;
2) для больших значений радиуса и длина окружности и ее площадь растут экспоненциально в зависимости от радиуса.

Это свойство и делает гиперболическое пространство идеальным для размещения в нем иерархических структур.

Существует 2 канонических способа отображения гиперболической плоскости на единичный диск евклидовой плоскости. В обоих случаях одна окрестность гиперболической плоскости находится в фокусе в центре диска, в то время как остальная часть гиперболической плоскости постепенно исчезает в перспективе по мере приближения к границе диска. Первая каноническая модель – это *конформная модель Пуанкаре*, которая сохраняет углы, но превращает линии гиперболического пространства в дуги единичного диска. Эта модель была использована для реализации двумерного гиперболического навигатора *StarTree*, и пользуется в настоящий момент невероятной популярностью у экспериментаторов (рис. 7).

Вторая каноническая модель – это *проективное отображение* или *модель Клейна*, которая отображает линии гиперболического пространства в прямые линии на плоскости, но зато искажает углы. Данный подход также был реализован в виде трехмерного Web-Браузера [30] и хорошо себя зарекомендовал при визуализации графов, для которых возможно создание осмысленных остовных деревьев на основе информации о предметной области (рис. 8).

Рис. 8. Пример трехмерного изображения графа на основе модели Клейна

В настоящее время гиперболические визуализаторы активно используются для просмотра больших коллекций данных [47]. Появляются новые

алгоритмы, такие как гиперболические самоорганизующиеся карты [32], а также гиперболическая версия силового алгоритма [22].

4.4. ДОИ-деревья для визуализации таксономий и органиграмм

С 2002 по 2006 год появилось сразу несколько экспериментальных систем (DOI-Tree, Tree-Plus, SpaceTree, TaxonTree), использующих описанные выше методы семантической фильтрации, геометрической и семантической модификации изображения [24, 25, 34]. Помимо этого в новых системах появились возможности, позволяющие управлять размером генерируемого изображения, располагая кластеризованное представление больших ветвей дерева в соответствии с заданными извне размерами. Последнее свойство позволяет использовать получаемые изображения в качестве модульных компонент информационных дисплеев и объединять их с другими элементами дисплея в приложении.

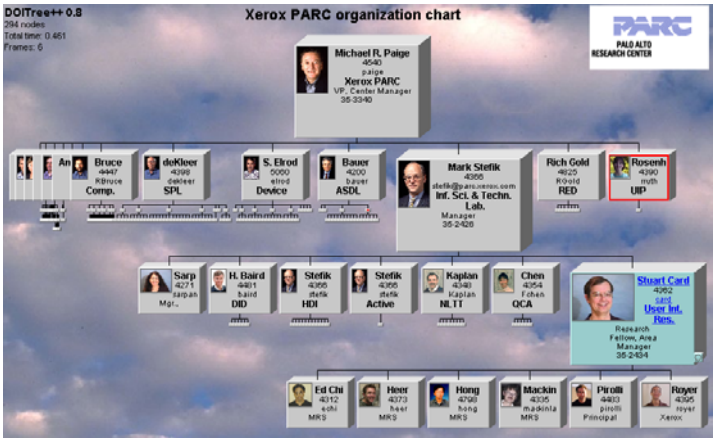


Рис. 9. Изображение органиграммы XEROX PARC в виде ДОИ-дерева

На рис. 9 показан пример использования DOITree для динамической визуализации организационной структуры XEROX Palo Alto Research Center, насчитывающей 400 вершин.

5. ВИЗУАЛИЗАЦИИ ЧИСЛЕННЫХ АТРИБУТОВ ИЕРАРХИЧЕСКИХ СТРУКТУР И БАЗ ДАННЫХ НА ОСНОВЕ КАРТЫ ДЕРЕВА (TREEMAP)

Метод визуализации иерархических структур, известный под названием Treemap (*карта дерева*) был введен B. Johnson и B. Shneiderman в 1991 году [19]. Несмотря на то, что с математической точки зрения метод не нов и известен в области САПР СБИС с 1982 года под названием *разрезающих планов (slicing floorplan)* [33], B. Shneiderman привнес новый взгляд на эту структуру, что привело и к появлению новых алгоритмов, неизвестных ранее. Первоначально метод был применен к визуализации распределения дискового пространства между файлами в иерархической структуре директорий, но постепенно получил широкое применение во многих областях визуализации от финансовой информации до результатов спортивных состязаний [8, 9, 18].

Этот вид визуализаций крайне эффективен при изображении численных атрибутов элементов (размер, стоимость, значение), организованных в большие иерархии. Базовая идея метода состоит в том, чтобы изобразить дерево, каждая вершина которого имеет имя и численный атрибут, в виде прямоугольника. Для изображения поддеревьев используется рекурсивная процедура разбиения прямоугольника, соответствующего всему дереву, на прямоугольники меньшей площади без незаполненных пространств и наложений. Для этого площадь прямоугольника разбивается горизонтальными и вертикальными линиями на прямоугольники, площадь которых пропорциональна значению данного атрибута. Такая разновидность метода называется методом *продольно-поперечных разрезов (Slice-and-Dice)*. На рис. 10 показан простой пример этого подхода.

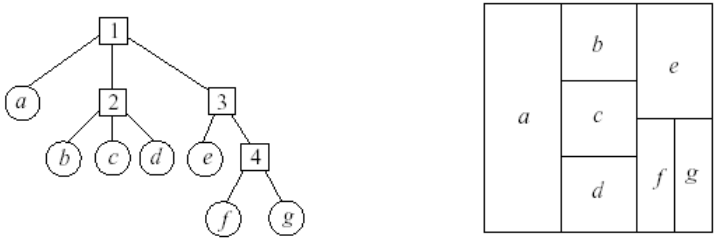


Рис. 10. Построение Карты дерева методом продольно-поперечных разрезов

Такой способ построения изображения очень эффективен, когда размер – это наиважнейший параметр структуры, который должен быть отображен. На рис. 11. показан общий план файловой системы, содержащей 1400 файлов, который позволяет с первого взгляда определить, где находится самый большой из файлов.

Тем не менее, у карты дерева существуют ограничения в применимости. Наихудший случай для них – это сбалансированное дерево, в котором каждая родительская вершина имеет одинаковое количество сыновей, а все листья имеют одинаковый размер. В этом случае визуализация превращается в бесполезное изображение квадратной сетки. Проблемой, специфической именно для метода продольно-поперечных разрезов, является то, что независимо от формы родительского прямоугольника направление разреза вычисляется только на основе информации об уровне вершины в дереве. В результате, прямоугольники часто режутся несколькими параллельными линиями и быстро становятся очень тонкими.

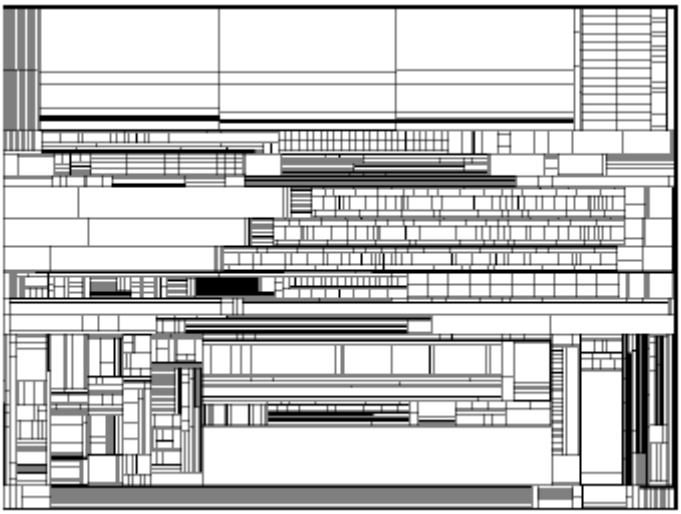


*Рис. 11***.** Общий план файловой системы, содержащей 1400 файлов

Поэтому во многих коммерческих приложениях используется алгоритм, основанный на *квадрифицирующей стратегии*, которая делает каждый прямоугольник как можно более квадратным, обычно размещая самые

большие квадраты в верхнем левом углу изображения, а самые маленькие – в нижнем правом углу. Это привлекательно визуально и полезно для понимания. Квадрифицированный алгоритм впервые был предложен в работе [9]. Другой алгоритм квадрификации, дающий аналогичное изображение, был использован для размещения на сайте финансового журнала Smartmoney.com интерактивной карты рынка, изображающей уровень капитализации для 1000 крупнейших мировых компаний (рис. 12).

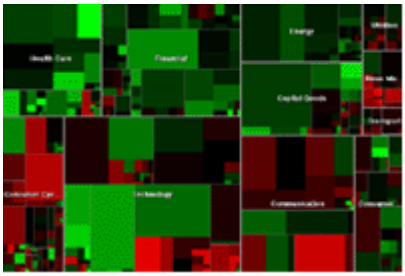


Рис. 12. Пример визуализации уровня капитализации предприятий при помощи квадрифицирующего алгоритма

Квадрифицирующий алгоритм ориентирован на поиск прямоугольников с хорошим коэффициентом формы, то есть таких, что отношение высоты к ширине каждого прямоугольника близко к единице. Данный метод был найден эмпирически, путем многочисленных экспериментов и основан на двух идеях.

1. Метод рассматривает разрезы только на одном уровне иерархии и пытается породить прямоугольники, близкие по форме к квадрату для множества братских вершин, беря за основу площади охватывающих прямоугольников. Предполагается, что квадратная форма охватывающих прямоугольников на одном уровне иерархии будет хорошей стартовой точкой для разрезаний на последующих уровнях.
2. На одном уровне иерархии разрешается иметь линии разреза, имеющие разные направления.

Алгоритм размещает прямоугольники в один ряд в еще незаполненной области изображения до тех пор, пока наихудший (самый высокий) коэффициент формы любого прямоугольника в текущем ряду продолжает улучшаться. Как только наивысший коэффициент формы среди прямоугольников достигает минимального значения, это означает, что добавление нового

прямоугольника к текущему ряду увеличило бы наихудший коэффициент формы. Поэтому текущий ряд фиксируется, и начинается новый ряд.

При создании нового ряда заново вычисляется направление размещения. Размещение происходит всегда вдоль более короткой стороны области размещения. Порядок, в котором добавляются прямоугольники, тоже имеет значение. Поскольку большие прямоугольники труднее разместить, чем маленькие, список прямоугольников перед размещением сортируется по убыванию площадей прямоугольников.

Вслед за квадрифицирующим алгоритмом быстро появилось огромное количество разнообразных стратегий построения карты дерева, таких как упорядоченные, кластеризованные, полосковые и квантованные карты дерева [19].

Карты дерева хорошо вписались в растущее семейство инструментов визуальной аналитики и бизнес-разведки. Приложения этого класса поставляются фирмой HiveGroup для анализа потребностей в таких областях, как страхование, управление продажами продуктов и производство нефти и газа.

В 2006 году представители фирмы Oracle провели контролируемое экспериментальное сравнение табличных представлений баз данных с программным обеспечением, поставляемым фирмой HiveGroup, после чего было сделано заключение, что «карты дерева должны быть включены в качестве стандартных графических компонент в приложения анализа данных и мониторинга уровня предприятия».

Версия Treemap 4.0, разработанная B. Shneiderman совместно с Catherine Plaisant, запатентована фирмой HiveGroup, и остается доступной для образовательных и исследовательских целей.

Вслед за картами дерева появилось большое количество модификаций этого подхода, использующих для изображения вершин графа геометрические формы, отличные от прямоугольников. Voronoi Treemaps [7] (карты дерева на основе диаграмм Вороного) используют сечения Вороного для вычисления размещения произвольных многоугольников, заполняющих пространство экрана. Этот подход ориентирован на улучшение выделения границ многоугольников, а также преодоление проблемы большого коэффициента формы. Весьма популярны также круговые карты дерева [48]. Эти методы активно используются для визуализации всевозможных иерархических структур, включая онтологии, тезаурусы, метрики программного обеспечения и т.д.

6. ЗАКЛЮЧЕНИЕ

Разработка наукоемких продуктов, использующих методы визуализации информации, требует специального обучения. Курсы по визуализации информации входят в программу Computer Science ведущих американских университетов, таких как Stanford и Berkeley.

Исторически так сложилось, что в Институте Систем Информатики СО РАН работает достаточно много специалистов, по роду деятельности связанных с алгоритмами визуализации графов. К этому направлению можно отнести все работы З. В. Апанович и А. Г. Марчука по автоматизации топологии САПР СБИС [1, 3, 4], работы М. А. Бульонкова и других сотрудников из Лаборатории смешанных вычислений по визуализации программного обеспечения [10], а также труды В. Н. Касьянова и В. А. Евстигнеева [2]. В течение последних четырех лет автором этой статьи читается курс, посвященный методам визуализации графов и методам визуализации информации, представляемой графами, в Новосибирском государственном университете. Все это позволяет надеяться, что накопленный потенциал будет способствовать внедрению методов визуализации информации на уровне российских предприятий, работающих в области ИТ.

СПИСОК ЛИТЕРАТУРЫ

1. Апанович З.В. Средства для работы с графами большого объема: построение и оптимизация компоновочных планов // Системная информатика: Сб. науч. тр. – Новосибирск: Изд-во СОРАН, 2006. – Вып. 10. Методы и модели современного программирования. – С. 7–58.
2. Касьянов В.Н., Евстигнеев В.А. Графы в программировании: обработка, визуализация и применение. – СПб.: БХВ-Петербург, 2003. – 1104 с.
3. Apanovitch Z., Bulyonkov M., Bulyonkova , Emelyanov P., Filatkina N., Ruyssen P. Using Floorplans for software Visualization // Bull. Novosibirsk Comp. Center. Ser. Computer Science. – Novosibirsk, 2006. – Iss. 24. – P. 27–44.
4. Apanovich Z.V. Marchuk A.G. An algorithm for standard cell and gate array placement // Proc. of EURO-ASIC-92, Paris, IEEE Computer Society Press, Los Alamitos, California, June 1992. – P. 416–421.
5. Archambault D., Munzner T., Auber D. Topolayout: Multi-level graph layout by topological features // IEEE Transactions on Visualization and Computer Graphics. – 2007. – Vol. 13, N 2. – P. 305–317.

6. Di Battista G., Eades P., Tamassia R., Tollis I.G. Algorithms for Drawing Graphs: an Annotated Bibliography // Computational Geometry, Theory and Applications. – 1994. – N 4. – P. 235–282.
7. Balzer M., Deussen O., Lewerenz C., Voronoi Treemaps for the Visualization of Software metrics// ACM Symp. on Software Visualization (SoftVis), 2005.
8. Bladh, T., Carr, D., Scholl, J. Extending tree-maps to three dimensions: a comparative study // Proc. of the 6th Asia-Pacific Conf. on Computer-Human Interaction (APCHI 2004). – 2004.
9. Bruls M., Huising K., van Wijk J. Squarified treemaps // Proc. of Joint Eurographics and IEEE TCVG Symp on Visualization (TCVG), 2000. – P. 33–42. – Available at: www.citeseer.ist.psu.edu/bruls99squarified.html
10. Bulyonkov M., Baburin D., Emelianov P., Filatkina N., Visualization facilities in program reengineering // Programming and Computer Software. – 2001. – Vol. 27, N 2. – P. 21–33.
11. Card S. K., Mackinlay J. D., Shneiderman B. Readings in Information Visualization: Using Vision to Think. – San Francisco: Morgan Kaufmann, 1999.
12. Card S. K., Nation D., Degree-Of-Interest Trees: A Component of an Attention-Reactive User Interface. – Palo Alto Reseaerch Center, 2002.
13. Collins C. Docuburst: Radial space-filling visualization of document content. – 2007. – (Tech. Rep. / Knowledge Media Design Institute; KMDI-TR-2007-1).
14. Eades P. A heuristic for graph drawing // Congress. Numerantium. – Vol. 42. – 1984 – P. 149–160.
15. Furnas G. W. Generalized Fisheye Views // Proc. of CHI'86. – 1986. –P. 16–23.
16. Gajer P., Kobourov S. G. GRIP: Graph dRawing with Intelligent Placement // Lect. Notes Comput. Sci. – 2001. –Vol. 1984. – P. 222–228.
17. Hachul, S. and Jünger, M.l. An Experimental Comparison of Fast Algorithms for Drawing General Large Graphs // Lect. Notes Comput. Sci. – 2005. –Vol. 3843. – P. 235–250.
18. Jin, L., Banks, D. C. TennisViewer: A Browser for Competition Trees // IEEE Computer Graphics and Applications. – Vol.17, N 4. – 1997. – P. 63–65.
19. Jonson B., Shneiderman B. Treemaps: a space-filling approach to the visualization of hierarchical information structure // Proc. of the Second Internat. IEEE Visualization Conf. – 1991.
20. Kadmon, N., and Shlomi, E. A polyfocal projection for statistical surfaces // Cartographic Journ. – 1978. – Vol. 15, N 1. – P. 36–41.
21. Kamada T., Kawai S. An Algorithm for Drawing general undirected graphs // Information Processing Letters. – 1989. – N 31. – P. 7–15.
22. Kobourov S.G., Wampler K. Non-Euclidean Spring Embedders // IEEE Transactions on Visualization and Computer Graphics. – 2005. – Vol. 11, N 6 – P. 757–767.
23. Lamping J., Rao R., Pirolli P. A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies // Proc. ACM Conf. Human Factors in Computing Systems. – 1995. – P. 401–408.

24. Lee B., Parr C. S., Campbell D., Bederson B. How users interact with biodiversity information using TaxonTree// Proceedings of the Working Conf. on Advanced Visual Interfaces Gallipoli, Italy. – 2004. – P. 320–327.
25. Lee B., Parr C. S., Plaisant C., Bederson B. B., Veksler V. D., Wayne D. Gray, C. Kotfila, TreePlus: Interactive Exploration of Networks with Enhanced Tree Layouts // IEEE TVCG (Infovis'06 Proc.). – 2006. – Vol. 12, N 6. – P. 1414–142.
26. Lee, B., Plaisant, C., Parr, CS., Fekete, JD., Henry, N. Task Taxonomy for Graph Visualization // Proc. of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization, Venice, Italy. – 2006. – P. 1–5.
27. Leung Y. K., Apperley M. D. A review and taxonomy of distortion-oriented presentation techniques // ACM Transactions on Computer-Human Interaction. – Vol. 1, N 2. – 1994. – P. 126–160.
28. Melancon G., Herman I. Circular Drawings of Rooted Trees. – 1998. – Available at: http://db.cwi.nl/rapporten.
29. Mullett K., Fry C., Schiano D. J. On Your Marks, Get Set, Browse! // CHI'97, (Panel).
30. Munzner T. Drawing large graphs with H3 Viewer and Site Manager // Lect. Notes Comput. Sci. – 1998. – Vol. 1547. – P. 384–393.
31. Munzner T., Guimbretiere F., Tasiran S., Zhang L., Zhou Y. TreeJuxtaposer: Scalable Tree Comparison using Focus+Context with Guaranteed Visibility // ACM Transactions on Graphics. – 2003. – Vol. 22, N 3. – P. 453–462.
32. Ontrup J., Ritter H. Large-scale data exploration with the hierarchically growing hyperbolic SOM // Neural Networks . – 2006. –Vol. 19, N 6–7 . –P.751–761.
33. Otten R.H.J.M. A new algorithm for floorplan design // Proc. 19th ACM/IEEE Design Automation Conf., Las Vegas, 1982. – New Jersey, 1982. – P. 101–107.
34. Plaisant, C., Grosjean J., Bederson. B. Spacetree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation // IEEE Symp. on Information Visualization. – 2002. – P. 57–64.
35. Purchase H. C., Which Aesthetic Has the Greatest Effect on Human Understanding? // Lect. Notes Comput. Sci. – 1997. – Vol. 1353. – P. 248–261.
36. Purchase H. C., Cohen R. F., James M., An Experimental Study of the Basis for Graph Drawing Algorithms // J. of Experimental Algorithms. – 1997. – Vol. 2, N 4.
37. Purchase H. C., Cohen R. F., James M., Validating Graph Drawing Aesthetics // Lect. Notes Comput. Sci. – 1995. – Vol. 1027. – P. 435–446.
38. Rao R. TableLens: A Clear Window for Viewing Multivariate Data. – 2006. – Available at: http://www.perceptualedge.com/articles/b-eye/tablelens.pdf.
39. Sarkar M., Brown M.H., Graphical Fish-eye views of graphs // Proc. of CHI '92 Conf. Human Factors in Computing Systems. – ACM Press, 1992. – P. 83–91.
40. Schaffer D., Zuo Z., Greenberg S., Bartram L., Dill J., Dubs S., Roseman M. Navigating Hierarchically Clustered Networks through Fisheye and Full-Zoom

Method // ACM Transaction on Computer-Human Interaction (TOCHI), 1996. – P. 162–188.
41. Six M., Tollis I. G. A framework for circular drawings of networks // Lect. Notes Comput. Sci. – 1999. –Vol. 1731. – P. 107–116.
42. Shneiderman, B., The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations // Proc. of IEEE. Symp. on Visual Languages, Los Alamos, 1996. – P. 336–343.
43. Slack J., Munzner T. Composite Rectilinear Deformation for Stretch and Squish Navigation // Transactions on Visualization and Computer Graphics, September 2006. – Vol. 12, N 5. – P. 901–908.
44. Sugiyama K., Tagawa S., Toda M. Methods for visual understanding of hierarchical system structures // IEEE Transactions on Systems, Man, and Cybernetics. – 1981. – Vol. SMC-11, N. 2. – P. 109–125.
45. Tamassia R. On embedding a graph in the grid with the minimum number of bends // SIAM. J. Comput. – 1987. – Vol. 16, N 3. – P.421–444.
46. Teoh S. T., Ma K.-L. RINGS: A Technique for Visualizing Large Hierarchies // Lect. Notes Comput. Sci. – 2002. – Vol. 2528. – P.268–275.
47. Walter J. A., Ontrup J., Wessling D., Ritter H. Interactive Visualization and Navigation in Large Data Collections using the Hyperbolic Space // Proc. of the 3rd IEEE Internat. Conf. on Data Mining (ICDM 2003), 19–22 December 2003. – P. 355–362.
48. Wang T.D., Parsja B. CropCircles: Topology Sensitive Visualization of OWL Class Hierarchies // Lect. Notes Comput. Sci. – 2006. – Vol. 4273. – P.695–708.
49. Wills G.J. NicheWorks-Interactive Visualization of Very Large Graphs // Lect. Notes Comput. Sci. – 1997. – Vol. 1353. – P. 403–414.

З. В. Апанович

ОТ РИСОВАНИЯ ГРАФОВ К ВИЗУАЛИЗАЦИИ ИНФОРМАЦИИ

Препринт
148

Рукопись поступила в редакцию 06.12.07
Рецензент Ю. А. Загорулько
Редактор Т. М. Бульонкова

Подписано в печать 28.12.07
Формат бумаги 60 × 84 1/16 Объем 1.6 уч.-изд.л., 1.75 п.л.
Тираж 60 экз.

Центр оперативной печати «Оригинал 2»
г.Бердск, ул. О. Кошевого, 6, оф. 2, тел. (383-41) 2-12-42