

Институт систем информатики им. А. П. Ершова СО РАН
пр. Академика Лаврентьева, 6, Новосибирск, 630090, Россия
E-mail: apanovich@iis.nsk.su

МЕТОДЫ НАВИГАЦИИ ПРИ ВИЗУАЛИЗАЦИИ ГРАФОВ *

Визуализация информации – это процесс преобразования больших и сложных видов абстрактной информации в визуальную форму. Универсальным средством представления такой информации являются графы. В данной статье дается обзор методов навигации при визуализации графов. Рассматриваются такие методы навигации как семантическая и структурная фильтрация вершин, геометрическое и семантическое искажение изображения и кластеризация вершин графа.

Ключевые слова: интерактивная визуализация информации, граф, навигация, фокус + контекст.

Введение

Методы визуализации информации предназначены для поддержки интерактивных представлений абстрактных данных, часто очень больших, полуструктурированных или многомерных, не имеющих явной физической интерпретации с целью усиления когнитивных возможностей пользователя. Интерактивная визуализация информации – это другой подход к обработке информации в информационных системах, которая превращается в непрерывный процесс взаимодействия с информацией через визуальное отображение. Пользователь может исследовать, рассматривать, открывать, узнавать и манипулировать данными через визуальные метафоры.

Приложения визуализации информации возникают в таких областях, как информационные системы и программное обеспечение, биологические науки, искусственный интеллект, анализ финансовой информации и компьютерное обучение.

Наукоемкие продукты, использующие методы визуализации информации, присутствуют 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). Одна из старейших компаний этого направления, «Inxight Software, Inc», поставлявшая средства визуализации информации для финансовых и биологических фирм, стала недавно подразделением SAP. 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.

Что же касается России, то за исключением немногих публикаций, посвященных рисованию графов [Апанович, 2006; 2007; Бульонков и др., 2001; Касьянов, Евстигнеев, 2003], русскоязычная литература по визуализации информации практически отсутствует. Поэтому автор надеется, что данный обзор, посвященный последним достижениям в области интерактивных методов визуализации информации, может способствовать развитию этого важного направления.

* Работа выполнена при финансовой поддержке РАН (грант 14/9 – «Разработка моделей и методов построения информационных систем, основанных на формальных, логических и лингвистических подходах»).

Универсальным средством представления абстрактной информации являются графы, поэтому методы визуализации графов и деревьев представляют собой теоретическую основу методов визуализации информации. Процесс создания интерактивной визуализации включает два важных шага. Первый шаг состоит в отображении данных на геометрическую плоскость. Это фундаментальная проблема визуализации решается чаще всего при помощи методов рисования или размещения графа. К сожалению, ни один алгоритм размещения не может справиться в одиночку с графами большого объема, используемыми в приложениях. Поэтому вторым неотъемлемым элементом интерактивной визуализации является навигация, обеспечивающая пошаговое изменение изображения и позволяющая достичь той информации, которая нужна определенному пользователю.

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. Этой «простой» проблеме посвящены тысячи научных публикаций, рассматривающих различные алгоритмы построения изображений графов, классифицирующих эти изображения и оценивающих их качество. Например, список основных типов изображений графов, рассмотренных в аннотированной библиографии [Di Battista et al., 1994] содержал такие типы изображений графов, как прямолинейные, полилинейные, сетчатые, ортогональные, восходящие (нисходящие) и плоские. В последующие годы к этому списку добавились круговые и радиальные изображения. Все эти типы изображений относятся к классу «диаграммы связей вершин», поскольку факт наличия связи между вершинами графа изображается при помощи линии, соединяющей эти вершины. С точки зрения потребностей визуализации информации все более популярными становятся методы изображения графов из класса «заполнения пространства», изображающие связь между вершинами при помощи геометрической вложенности их изображений. Этот подход позволяет повысить объем изображаемых данных и чрезвычайно полезен в приложениях, требующих визуального сравнения размеров атрибутов.

При построении изображения графа принято руководствоваться эстетическими критериями, такими как:

- минимизация площади;
- минимизация количества пересечений ребер;
- минимизация количества сгибов ребер;
- минимизация количества наложений ребер и вершин;
- максимизация угловой резолуции;
- минимизация длины ребер (иногда – требование одинаковой длины);
- максимизация симметричности изображения;
- максимизация (или управляемость) коэффициента формы.

В силу того что часто невозможно удовлетворить все критерии одновременно и известно много примеров их взаимопротиворечивости, возникают различные стили или методологии изображения, зависящие от конкретного приложения. Наиболее широко распространенными способами изображения графов на данный момент являются:

- «силовые» [Fruchterman, Reingold, 1991] и круговые [Six, Tollis, 1999] методы для неориентированных графов;
- иерархические (поуровневые) методы [Sugiyama et al., 1981] для ориентированных графов;
- ортогональные методы [Tamassia, 1987] для неориентированных разреженных графов.

Специфические проблемы визуализации информации

Одним из главных критериев оценки качества методов визуализации информации является адекватность изображения заданному типу информации. В случае визуализации географических карт или карт дорог желательно, чтобы расположение вершин и ребер соответствовало 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 телефонных звонков или же информации о доступе пользователей к различным сайтам.

Что касается технических проблем, одной из основных остается *объем изображаемых данных*. Алгоритм, который хорошо размещает несколько сотен вершин, совершенно необязательно будет так же хорош при работе с несколькими тысячами. Для графов большого объема известно очень немного специализированных систем [Archambault et al., 2007; Gajer, Kobourov, 2001; Nachul, Jünger, 2005], да и не всегда одновременное размещение большого количества информации 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*.

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 *предсказуемость* (сохранение ментальной карты). Этот критерий предполагает, что два разных прогона одного и того же алгоритма на одних и тех же или похожих данных должны давать похожий результат.

Считается, что хорошая визуализация информации помогает пользователю быстро находить нужный элемент в иерархии, понимать отношение элемента к его контексту и обеспечивать возможность прямого доступа к информации при вершинах. Бен Шнайдерман [Shneiderman, 1996] так сформулировал «мантру» интерактивной визуализации: «Сначала общий план, потом масштабирование и фильтрация, затем детали по требованию», имея в виду действия, которые должен быть способен выполнять пользователь при взаимодействии с системой. Помимо стандартных методов навигации, таких как масштабирование и прокрутка, существуют и специальные методы взаимодействия с графовой информацией. Эти методы можно разбить на три большие подгруппы: семантическая и / или структурная фильтрация вершин, геометрическое или семантическое искажение изображения, абстракция (кластеризация) вершин.

Фильтрация вершин графа

Под семантической фильтрацией вершин понимается удаление из графа (или из изображения графа) подмножеств вершин либо на основе значений индивидуальных атрибутов вершин либо исходя из структурных свойств графа с целью сокращения визуальной сложности изображения. На данный момент общепринятыми являются две модели фильтрации: фильтрация «рыбий глаз» и глобальная фильтрация. Глобальная фильтрация осуществляется при помощи функции значимости вершин, в то время как фильтрация «рыбий глаз» использует помимо значимости вершин еще и расстояние до фокусной вершины. Кроме этого, глобальная фильтрация удаляет вершины насовсем, а фильтрация «рыбий глаз» делает их временно невидимыми.

Модель фильтрации «рыбий глаз» была предложена Фурнасом [Furnas, 1986] применительно к иерархическим структурам, в частности деревьям. Он описал класс методов, в которых вершины автоматически включаются или удаляются из изображения иерархической структуры в соответствии вычисляемой степенью пользовательского интереса (СИ), создал системы для просмотра и фильтрации структурированного программного кода, биологических таксономий и календарей, а также экспериментально продемонстрировал удобство этих динамических изображений для пользователя.

В модели Фурнаса функция СИ(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. Функция степени интереса для всех вершин вычисляется по формуле:

$$СИ(v) = \text{Априорная_значимость}(v) + \text{Расстояние_до_Фокуса}(v),$$

где Априорная_значимость(v) (АЗ(v)) – это априорная оценка важности вершины (v).

В случае, когда изображаемая структура является деревом, $A3(v)$ 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 $\text{Расстояние_до_Фокуса}(v)$ 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, степень интереса которых превышает заданный порог (рис. 1).

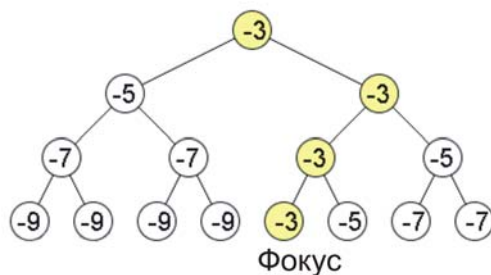


Рис. 1. Подсчет функции степени интереса

В момент, когда пользователь выбирает в качестве фокусной любую другую вершину дерева, происходит перевычисление СИ для всех вершин, после чего из изображения удаляются вершины, СИ которых оказалась ниже заданного порога, но зато в изображении появляются новые ветви, для вершин которых СИ возросла настолько, что превысила заданный порог. Таким образом, изображение дерева изменяется вслед за изменяющимся интересом пользователя. В работе Фурнаса было показано, что функция СИ, определенная для деревьев, обладает свойствами монотонности и выпуклости, что позволяет эффективную реализацию алгоритмов интерактивного размещения вершин дерева на основе введенной функции. С 2002 по 2006 г. появилось сразу несколько экспериментальных систем (DOI-Tree, Tree-Plus, SpaceTree, TaxonTree), использующих описанные выше методы семантической фильтрации [Card, Nation, 2002; Lee et al., 2004; Lee, Parr et al., 2006; Plaisant et al., 2002].

Для того чтобы применить фильтрацию «рыбий глаз» не к дереву, а к графу, можно взять в качестве функции расстояния расстояние по графу от вершины до фокуса, т. е. $D(u, v_f) = d(u, v_f)$. Что же касается функции априорной значимости ($A3$), то ее значение можно выбирать либо на основе значений индивидуальных атрибутов вершины, либо на основе структурных свойств графа. Выбор конкретной функции зависит от приложения. Один из примеров вычисления функции $A3$ на основе значения атрибутов будет рассмотрен ниже. Если же в приложении визуализации важны структурные свойства графа, то желательно, чтобы функция $A3$ вершины показывала, насколько вершина явно или неявно связана с другими вершинами. Для измерения этого свойства хорошо подходят меры *центральности*, введенные для изучения свойств социальных сетей [Freeman, 1978]. Наиболее распространенными являются три меры центральности: *степень*, *близость* и *промежуточность*. В случае, когда эти меры нормализованы так, что их значения находятся в промежутке от 0 до 1, они называются *относительными мерами центральности*.

Центральность степени $C_D(u)$ – это просто количество ребер, инцидентных данной вершине. При нормализации это значение делится на $n - 1$:

$$C_D(u) = \frac{\text{deg}(u)}{n - 1}.$$

Центральность близости $C_C(u)$ оценивается как сумма кратчайших путей, соединяющих данную вершину со всеми остальными вершинами:

$$C_C(u) = \frac{\sum_{v \in V} d(u, v)}{n - 1},$$

где $d(u, v)$ – это кратчайший путь между вершинами u и v , который равен количеству ребер в нем.

Центральность промежуточности $C_B(u)$ вычисляется как отношение количества кратчайших путей между двумя вершинами, проходящих через вершину u , к количеству всех возможных кратчайших путей в графе: $C'_B(u) = \sum_{j=1}^n \sum_{k=1}^{j-1} \frac{g_{jk}(u)}{g_{jk}}$, а ее нормализованное значение –

$C_B(u) = \frac{C'_B(u)}{(n-1)(n-2)/2}$, где g_{ik} – это количество кратчайших путей между вершинами j и k , а $g_{jk}(u)$ – это количество кратчайших путей, содержащих вершину u .

Известно, что данные три меры центральности могут давать противоречивые результаты на одном и том же графе. Поэтому в качестве меры важности вершины разумнее использовать комбинированное значение: $A3(u) = w_1 CD(u) + w_2 CC(u) + w_3 CB(u)$.

Для нормализации функции расстояния, разделим ее значение на величину, равную максимальному расстоянию по всем парам вершин, что соответствует диаметру графа

$$D = \max d(u, v_j).$$

Тогда функция степени интереса для графа имеет вид

$$СИ(u) = A3(u) - D(u, v_j) = A3(u) - d(u, v_j)/D.$$

При такой нормализации, значение $СИ(u)$ принадлежит отрезку $[-1, 1]$.

Введя пороговое значение t , мы получим множество видимых вершин:

$$V' = \{u | u \in V \wedge СИ(u) \geq t\}.$$

В результате такой фильтрации будут показаны все вершины в окрестности фокусной вершины, в то время как из более удаленных вершин отрисуются только наиболее важные. Если для приложения существенным требованием является связность изображаемого графа, то при формировании множества видимых вершин следует также проверять, что в него попали все точки сочленения и мосты исходного графа.

Применение к изображениям графов геометрических искажений «рыбий глаз»

Саркар и Браун [Sarkar, Braun, 1992] обобщили подход Фурнаса на случай произвольного графа, предложив использовать в качестве функции $A3$ один из численных атрибутов, связанных с каждой вершиной, а в качестве расстояния до фокусной вершины – евклидово расстояние между вершинами уже существующего изображения. 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 $A3(v)$. 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 от расстояния до фокуса f :

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

2. Размер вершины v в деформированном изображении «рыбий глаз» $S_{feye}(v, f)$ зависит от ее размера в нормальном изображении $S_{norm}(v)$, расстояния до фокуса $P_{norm}(f)$ и $A3(v)$:

$$S_{feye}(v, f) = F_2(S_{norm}(v), P_{norm}(f), A3(v)).$$

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:

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

4. Наконец, решение о том, появится ли данная вершина в деформированном изображении, принимается на основе значения *Визуальная Значимость* = $VW(v, f)$, зависящего от $A3(v)$ и от расстояния между вершиной v и фокусом в нормальном изображении:

$$VW(v, f) = F_4(P_{norm}(v), P_{norm}(f), A3(v)).$$

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 два 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. Например, для построения *декартова* искажения «рыбий глаз», x и y координаты каждой вершины v вычисляются независимо по следующей формуле (функция F_1):

$$P_{\text{feye}}(v, f) = \left[G\left(\frac{D_{\text{norm}_x}}{D_{\text{max}_x}}\right) D_{\text{max}_x} + P_{f_x}, G\left(\frac{D_{\text{norm}_y}}{D_{\text{max}_y}}\right) D_{\text{max}_y} + P_{f_y} \right],$$

где D_{max_x} и D_{max_y} – это расстояния по горизонтали и по вертикали между границей экрана и фокусом изображения в нормальных координатах, а D_{norm_x} и D_{norm_y} – расстояния по горизонтали и по вертикали между вершиной v и фокусом в нормальных координатах. Отношения $D_{\text{norm}_x} / D_{\text{max}_x}$ и $D_{\text{norm}_y} / D_{\text{max}_y}$ принимают значения в интервале между 0 и 1. В качестве функции

G , называемой *функцией искажения*, использовалась функция $G(x) = \frac{(d+1)x}{dx+1}$, которая монотонно возрастает и непрерывна на интервале $0 \leq x \leq 1$; при этом $G(0) = 0$, $G(1) = 1$. Константа d называется *коэффициентом искажения*.

На рис. 2 показан пример нормального и декартова искажения «рыбий глаз» применительно к карте 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.

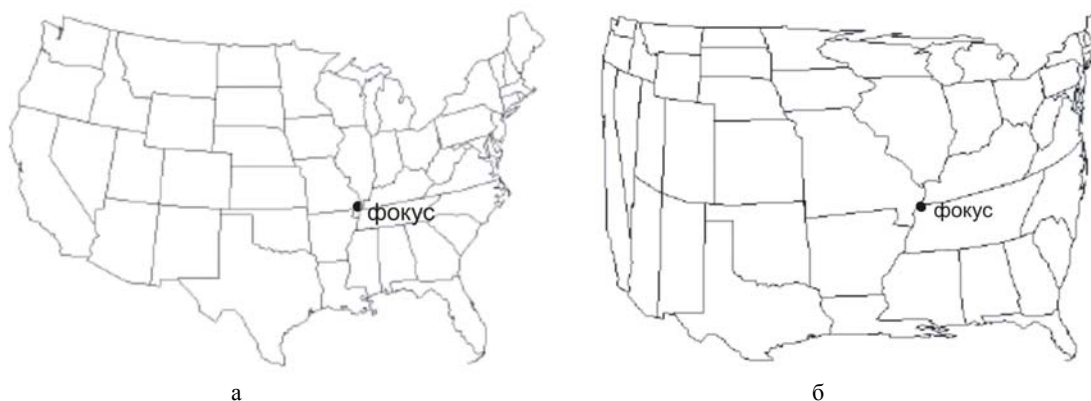


Рис. 2. Нормальное изображение (а) и декартово искажение «рыбий глаз» (б) для карты США

Функция трансформации Саркара и Брауна $G(x)$ является частным случаем применения функций нелинейного увеличения, математическую основу которых заложили Кэдмон и Шломи [Kadmon, Shlomi, 1978]. Они предложили использовать для описания искажений изображений две функции: трансформации и увеличения. Функция трансформации представляет собой функцию искажения, которая растягивает или сжимает изображение. Функция увеличения является производной функции трансформации и явно представляет значения увеличения, присутствующие в функции трансформации неявно (рис. 3).

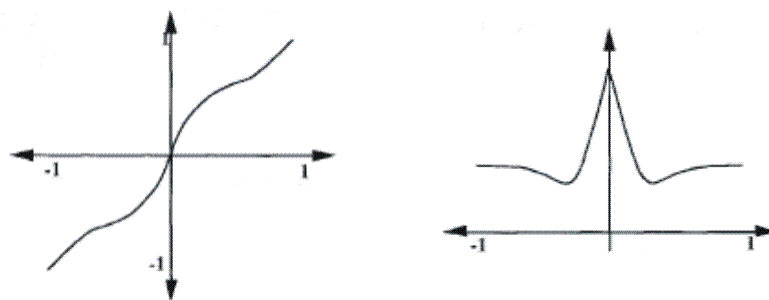


Рис. 3. Функции трансформации и увеличения

Кривизна функции увеличения управляется двумя множествами параметров, одно множество управляет увеличением вблизи фокуса, а второе – изменением уровня увеличения по мере удаления от фокуса. На рис. 4 показан результат применения этих функций в одномерном и двумерном случаях. «Провалы» функции увеличения служат для компенсации высокого коэффициента увеличения в области, окружающей фокусную точку. Для того чтобы получить изображение с несколькими центрами увеличения, функция увеличения должна иметь множественные пики.

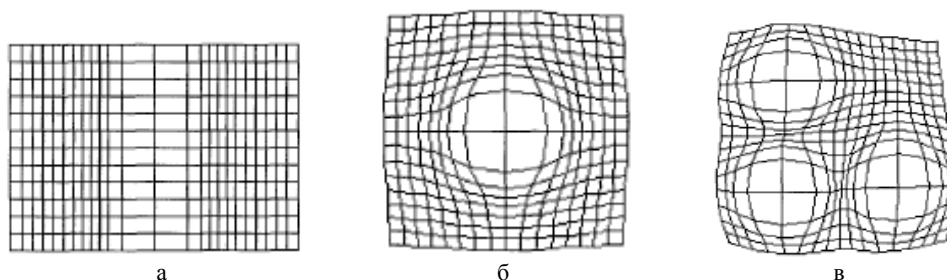


Рис. 4. Результат применения функции трансформации в одномерном (а) и двумерном (б) случаях. Многофокусная деформация изображения (в)

Функция трансформации Саркара и Брауна $G(x)$ является типичным представителем класса одномерных функций нелинейной трансформации. Хотя эта функция не является слишком затратной с вычислительной точки зрения, у нее есть неудобное требование нормализации координат перед применением. В качестве альтернативного варианта можно подобрать другую функцию $h(x)$, не нуждающуюся в нормализации. Таких функций существует немало. Можно, например, использовать гиперболическую функцию $h(x) = \tanh(x)$. Также ее можно снабдить параметром β , управляющим степенью увеличения: $h(x, \beta) = \tanh(x, \beta)$, и она не нуждается в предварительной нормализации. Еще одной функцией, дающей результат, аналогичный $\tanh(x)$, является функция $h(x, \beta) = [2,0 / (1,0 + e^{-2\beta x})] - 1,0$. Эта функция на многих компьютерах вычисляется более эффективно, чем $\tanh(x)$.

С практической точки зрения, основным недостатком навигации при помощи функций нелинейного увеличения является отсутствие инструментов, позволяющих явно управлять выбором одной или нескольких фокусных областей, количеством пространства, выделяемым для каждой области и уровнем увеличения. Все эти параметры вычисляются неявно на основе задаваемого пользователем коэффициента искажения. Помимо этого, все рассмотренные выше функции нелинейного увеличения (например, гиперболический тангенс) можно аппроксимировать при помощи кусочно-линейных функций. Такие аппроксимации обладают большей производительностью и допускают более широкий класс трансформаций. Поэтому в последние годы все большей популярностью пользуется альтернативный механизм навигации фокус + контекст, использующий метафору «резиновый жгут».

Метафора «резиновый жгут» и «аккордеонные изображения» деревьев

Одним из первых примеров системы, использующей для навигации метафору «резиновый жгут» является StretchTool [Sarkar, Reiss, 1992]. Вместо управления функцией искажения пользователю было предложено манипулировать резиновым жгутом при помощи *рукояток*, которые можно располагать и перемещать на экране. Часть экрана, окруженная рукоятками с четырех сторон, называется *областью*. При перемещении рукоятки область, расположенная по одну сторону от рукоятки, растягивается, а по другую сторону – сжимается. С каждой областью связывается горизонтальный коэффициент растяжения s_x и вертикальный коэффициент растяжения s_y . Горизонтальный (соответственно вертикальный) коэффициент растяжения равен отношению исходной ширины (соответственно высоты) области к ширине (высоте) области после растяжения: $s_x = W_{old} / W_{new}$, $s_y = H_{old} / H_{new}$. В растягиваемой области коэффициент растяжения больше единицы, а в сжимаемой – меньше единицы. Предположим, что в результате растяжения области текущая позиция западной рукоятки равна W_p , а ее смещение – W_d . Аналогично S_p и S_d будут соответствовать текущей позиции и смещению южной рукоятки. Тогда любая точка $p(x, y)$ внутри прежней области отображается в точку $p'(x, y)$ растянутой области при помощи операций масштабирования, в которых в качестве коэффициента масштабирования используется горизонтальный или вертикальный коэффициент растяжения:

$$\begin{aligned}x' &= W_p + s_x(x - (W_p - W_d)), \\y' &= S_p + s_y(y - (S_p - S_d)).\end{aligned}$$

На рис. 5 показано, как меняются координаты точек после смещения западной рукоятки области вправо. Таким образом, функция трансформации, автоматически вычисляемая системой при смещении рукояток, является кусочно-линейной, а функция увеличения – кусочно-постоянной.

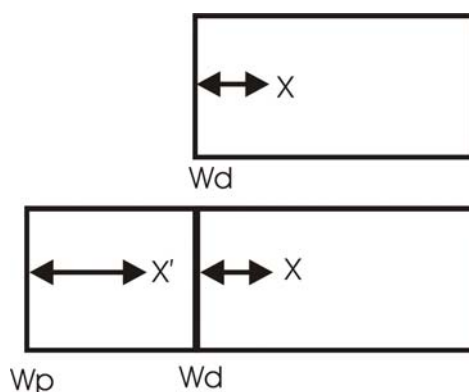


Рис. 5. Изменение координат точки после растяжения области

В работе [Slack, Munzner, 2006] описан базовый алгоритм навигации под названием *аккордеонное изображение*, использующий метафору «резиновый жгут» и позволяющий визуальное сравнение структуры деревьев, насчитывающих сотни тысяч или даже миллионы вершин (рис. 6). Хотя визуальная метафора, предлагаемая пользователю, достаточно интуитивна, инфраструктура, поддерживающая такую навигацию, весьма непростая. Используется иерархическая структура данных, которая хранит относительные границы областей, для того чтобы быстро вычислять их абсолютные позиции. При отрисовке абсолютные позиции границ вычисляются только для видимых областей. За счет того что количество отрисовываемых объектов существенно меньше, чем их общее количество, при обновлении состояния навигации не требуется перевычисление позиций всех элементов изображения, даже если деформация изменила позиции всех объектов. Этот алгоритм имеет две специфические особенности:

- все рукоятки, называемые далее *линиями разреза*, генерируются автоматически, избавляя пользователя от необходимости создавать их вручную;

- каждая последующая деформация применяется к состоянию, описывающему предыдущую трансформацию, благодаря чему все ранее совершенные действия продолжают влиять на изображение.

Автоматическая генерация рукояток. Для реализации процесса навигации каждый геометрический объект должен быть окружен четырьмя автоматически сгенерированными рукоятками. Поэтому при инициализации аккордеонного изображения дерева автоматически создается сетка размером $n \times t$, где n равно количеству листьев дерева, а t – высоте дерева, а затем все геометрические элементы изображения дерева распределяются по ячейкам этой сетки. Процесс автоматической генерации рукояток называется дискретизацией и включает четыре шага:

- размещение дерева;
- генерация сетки;
- распределение ребер дерева по узлам сетки;
- создание таблицы соответствия между геометрическими объектами и рукоятками.

Наименьшая из сгенерированных областей называется *атомарной областью*. Последовательность атомарных областей образует *коллективную область*. Алгоритм навигации основан на деформации коллективных областей. Единичная деформация, применяется к коллективной области и требует двух деформаций (справа и слева от линии разреза), но каждая из коллективных областей может состоять из гораздо большего количества атомарных областей.

Если бы система навигации работала с абсолютными позициями линий разреза, то она требовала бы $O(n)$ обновлений для корректного изменения координат всех линий разреза. Вместо этого система использует *бинарное дерево разрезов*, хранящее относительное пространственное упорядочение деформируемых областей, ограниченных линиями разреза (рис. 7). В каждой вершине дерева хранится *отношение разреза*, число в диапазоне от 0 до 1, равное отношению размера области левого сына к размеру всей отцовской области:

$$split\ ratio = left_son_area / parent_area.$$

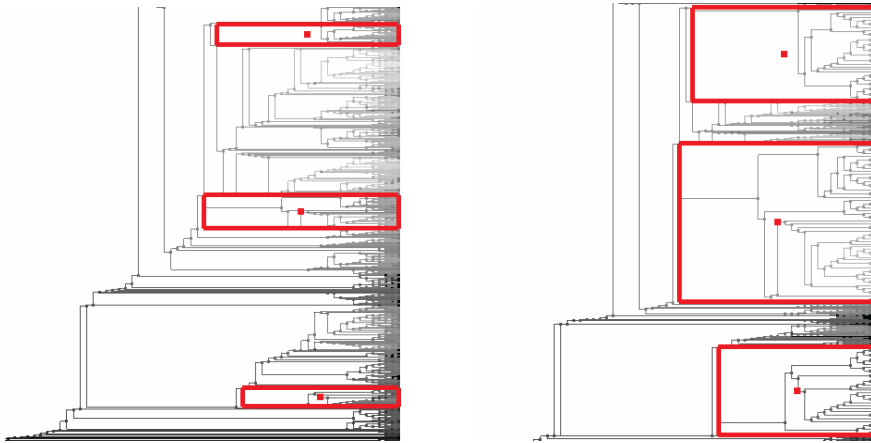


Рис. 6. Пример трехфокусной навигации «Резиновый жгут» для филогенетического дерева

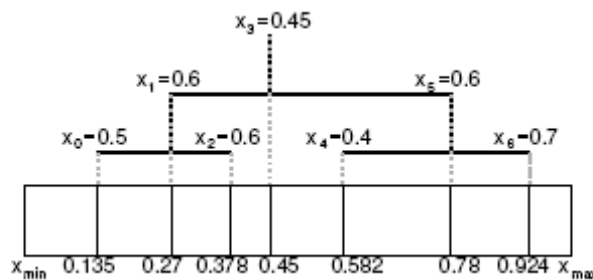


Рис. 7. Бинарное дерево разрезов для пересчета координат

Абсолютную координату любой линии разреза можно вычислить при помощи коэффициентов разреза, встречающихся на пути от корня бинарного дерева до вершины, соответствующей данной линии разреза. Вычисление абсолютных позиций осуществляется спуском по дереву, начиная с корня, так что абсолютные позиции предков могут быть использованы при вычислениях позиций потомков. Следовательно, в сбалансированном бинарном дереве абсолютные позиции всех вершин можно вычислить за время $O(\log n)$.

Благодаря этому свойству, базовый алгоритм навигации растяжений и сжатий, реализующий композицию последовательных прямолинейных деформаций, состоит в том, что получает на входе *новые абсолютные* позиции границ k коллективных областей, изменивших свой размер, а на выходе он выдает *новые относительные* позиции границ областей, общее количество которых не превышает $q \leq k \log n$. При перемещении границ k коллективных областей, при общем количестве n атомарных областей, сложность этого алгоритма оценивается как $O(k \log n)$. Данный подход обладает хорошей масштабируемостью и использовался для визуального сравнения не только филогенетических деревьев, но и последовательностей генов. Радиальная версия этого алгоритма навигации реализована в системе DocuBurst, предназначенной для визуализации документов [Collins, 2007; Yang et al., 2003].

Визуальная абстракция графа на основе кластеризации

При визуализации больших графов не всегда имеет смысл размещать одновременно все вершины и ребра. Возможность уменьшения количества видимых элементов с сохранением при этом глобальной структуры графа может оказаться весьма полезным. Такая трансформация изображения называется *визуальной абстракцией графа*. Стандартным способом построения таких абстракций является применение различных методов кластеризации [Archambault et al., 2007; Gajer, Kobourov, 2001; Nachul, Jünger, 2005]. В то же время пользователь может быть заинтересован в том, чтобы увидеть детали тоже, поэтому нужен приемлемый способ интеграции деталей с контекстом.

Специфическим примером, где данный подход весьма оправдан, являются *графы малых миров*, у которых средний кратчайший путь между вершинами невелик, но степень кластеризации по сравнению со средним графом такого же размера высока. Это свойство может быть идентифицировано во многих графах реального мира, таких как социальные сети, нейронные сети, программные системы, сети электроснабжения, базы знаний с перекрестными ссылками и Интернет.

Основным инструментом визуализации таких графов являются силовые методы размещения, которые обычно настроены на минимизацию разницы длины ребер. Но для кластеризованных графов более естественным было бы изображение, в котором ребра внутри кластера короче, чем межкластерные связи. Недавно Ноак [Noack, 2004] предложил силовой алгоритм для вычисления таких размещений. При генерации размещения разработанной им программой 1-LinLog расстояние между кластерами обратно пропорционально их *сцеплению*. Сцепление между двумя кластерами C_1 и C_2 – это мера их связанности, определяется как $E(C_1, C_2) / |C_1||C_2|$, где $E(C_1, C_2)$ означает общее количество ребер, соединяющих эти кластеры. Благодаря этому свойству можно использовать геометрические расстояния в качестве метрики кластеризации, так как наиболее сцепленные кластеры геометрически находятся ближе друг к другу. Вооруженные этим алгоритмом, ван Хам и ван Вийк [Ham, Wijk, 2004] предложили серию интерактивных методов, предназначенных специально для визуализации графов малых миров. Они использовали размещение, вычисляемое системой LinLog, для построения дендрограмм, в которых вершины соответствуют кластерам вершин, затем ввели функцию степени абстракции, зависящую от положения кластера в дендрограмме и его расстояния до фокусной точки в размещении, и предложили метод гладкого перехода от одного уровня абстракции к другому на основе функции степени абстракции (СА).

Агломеративная кластеризация графа G вычисляется снизу вверх, используя меру расстояния $d(i, j)$ между кластерами i и j . Кластер k описывается четверкой (c_k, r_k, d_k, f_k) , где c_k – это центр кластера; r_k – его радиус; d_k – расстояние между его подкластерами; f_k – родительский кластер. Первоначально каждой вершине с номером i присваивается значение $(p_i, r_{base}, 0, nil)$. Затем кластеры итеративно сливаются выбором на каждом шаге двух кластеров i и j

наименьшим расстоянием $d(i, j)$ и их слиянием в новый кластер k . Для всех ребер e_x , инцидентных вершинам i или j , создается новое ребро e_{kx} . Если ребро уже существует, возрастает его вес $w(e_{kx})$.

Позиция c_k кластера k вычисляется как среднее арифметическое позиций его листовых вершин $l \in LN(k)$. Радиус r_k пропорционален квадрату количества листовых вершин $(|LN(k)|)^2$. В качестве функции расстояния $d(i, j)$ между кластерами берется среднее расстояние между листовыми вершинами двух кластеров. В качестве расстояний между членами кластеров берется длина ребер, а для несвязных вершин расстояние устанавливается в бесконечность. Такой способ не позволяет объединять кластеры, которые не связаны, но расположены близко геометрически. Хотя эстетически не самый красивый, метод длины ребер дает хорошие кластеры.

В результате кластеризации создается бинарная иерархия, или *дендрограмма*, в которой уровень иерархии для кластера k равен расстоянию d_k между его сыновними кластерами. Простой способ создания абстракции для кластеризованных графов состоит в том, чтобы определить глобальную степень абстракции $CA \in [0, 1]$ и отрисовывать только те кластеры k , для которых $d_k \leq d_{\text{root}} \cdot CA < d_{jk}$. Для создания непрерывного перехода от одного изображения к другому при изменении CA можно интерполировать переход от размера и позиции кластера k к размеру и позиции отцовского кластера при помощи локального параметра $\lambda_k = ((CA \cdot d_{\text{root}}) - d_k) / (d_{jk} - d_k)$, где λ_k – это параметр, указывающий величину интерполяции между позицией и размером кластера k и позицией и размером его отца f_k на основе CA в этой точке. Для всех видимых кластеров $0 \leq \lambda_k \leq 1$, потому что $d_k \leq d_{\text{root}} \cdot CA < d_{jk}$.

Для интерактивного просмотра графа используется разбиение изображения на три концентрических области, центром которых является вершина f , выбираемая в качестве фокуса. Исходя из расстояния $s = |p - f|$ точки p до фокуса f выделяется фокальная область $A = \{p: s(p) \leq R_f\}$, в которой вершины не абстрагируются, но подвергаются геометрическому искажению. В области $B = \{p: R_f < s(p) \leq R_{CA}\}$ вершины все больше абстрагируются по мере удаления от фокуса, а в оставшейся области C уровень абстракции остается постоянным.

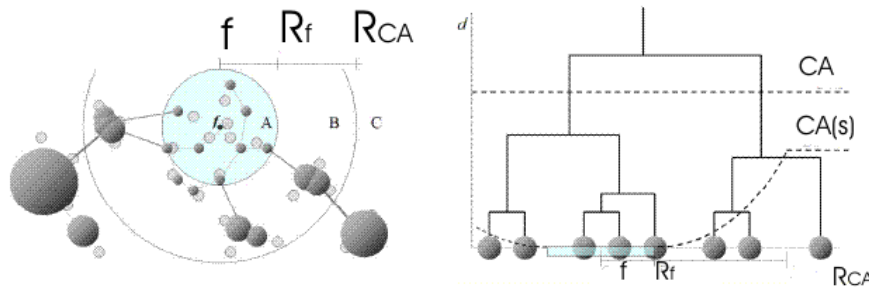


Рис. 8. Геометрическое искажение и функция CA

В качестве функции геометрического искажения в фокусной области A предпочтительна такая функция Z , которая увеличивает элементы изображения, близкие к фокусу, но не искажает их, когда $s \geq R_f$. Подходящей функцией является функция Саркара и Брауна:

$$Z(s) = \frac{(z+1)s}{\frac{zs}{R_f} + 1}, \text{ если } s < R_f,$$

$$Z(s) = s, \text{ если } s \geq R_f,$$

где z – это коэффициент, который определяет количество увеличения в окрестности фокуса. Размер вершин остается постоянным, меняется только их координата, а вместе с этим и длина ребер. Это дает визуальный эффект плотно упакованного кластера вершин, растягиваемого вокруг фокуса при помощи линзы «Рыбий глаз».

Для изображения точек вне фокусной области, используется функция семантической абстракции $CA(s) \in [0, 1]$, зависящая от расстояния s до фокуса f :

$$CA(s) = 0, \text{ если } s < R_{f\bar{b}},$$

$$CA(s) = a(s), \text{ если } R_f < s \leq R_{CA},$$

$$CA(s) = a(R_{CA}), \text{ если } R_{CA} < s.$$

Выбор возрастающей функции a произволен, это может быть линейная или кубическая функция. Результирующее изображение графа включает только наивысшие кластеры в иерархии k , для которых верно соотношение $d_k \leq d_{\text{root}} \cdot CA(|s_k - f|) \wedge d_{\text{root}} \cdot CA(|s_{fk} - f|) < d_{fk}$. Для того чтобы два различных действия увеличения (одно – геометрическое, а второе – семантическое) были понятнее пользователю, границы области геометрического искажения явно указываются на изображении.

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. В течение последних четырех лет автором этой статьи читается курс, посвященный методам визуализации графов в Новосибирском государственном университете. В настоящий момент заканчивается подготовка к публикации первой части учебного пособия, соответствующего данному курсу.

Список литературы

Апанович З. В. Средства для работы с графами большого объема: построение и оптимизация компоновочных планов // Системная информатика: Сб. науч. тр. Новосибирск: Изд-во СО РАН, 2006. Вып. 10: Методы и модели современного программирования. С. 7–58.

Апанович З. В. От рисования графов к визуализации информации. Препринт № 148. Новосибирск: ИСИ СО РАН, 2007. 16 с.

Бульонков М. А., Бабурин Д. Е., Емельянов П. Г., Филаткина Н. Н. Средства визуализации при перепроектировании программ // Программирование. М.: Наука, 2001. №2. С. 21–33.

Касьянов В. Н., Евстигнеев В. А. Графы в программировании: обработка, визуализация и применение. СПб.: БХВ-Петербург, 2003. 1104 с.

Archambault D., Munzner T., Auber D. Topolayout: Multi-level Graph Layout by Topological Features // IEEE Transactions on Visualization and Computer Graphics. 2007. Vol. 13. No. 2. P. 305–317.

Di Battista G., Eades P., Tamassia R., Tollis I. G. Algorithms for Drawing Graphs: an Annotated Bibliography // Computational Geometry, Theory and Applications. 1994. No. 4. P. 235–282.

Card S. K., Mackinlay J., Shneiderman B. Readings in Information Visualization: Using Vision to Think. 1999.

Card S. K., Nation D. Degree-Of-Interest Trees: A Component of an Attention-Reactive User Interface // Palo Alto Research Center May 22–24, 2002.

Collins C. Docuburst: Radial Space-Filling Visualization of Document Content // Technical Report KMDI-TR-2007-1, Knowledge Media Design Institute. 2007.

Freeman L.C. Centrality in Social Networks: Conceptual Clarification // Social Networks. 1978. No. 1. P. 215–239.

Fruchterman T. M. J., Reingold E. M. Graph Drawing by Force-Directed Placement Software // Practice and Experience. 1991. Vol. 21. No. 11. P. 1129–1164.

Furnas G. W. Generalized Fisheye Views // Proceedings of CHI'86. New York: ACM Press, 1986. P. 16–23.

Gajer P., Kobourov S. G. GRIP: Graph dRrawing with Intelligent Placement // Lect. Notes Comput. Sci. 2001. Vol. 1984. P. 222–228.

Hachul S., Jünger M. L. An Experimental Comparison of Fast Algorithms for Drawing General Large Graphs // Lect. Notes Comput. Sci. 2005. Vol. 3843. P. 235–250.

- Kadmon N., Shlomi E.* A Polyfocal Projection for Statistical Surfaces // Cartographic Jour. 1978. Vol. 15. No. 1. P. 36–41.
- Lee B., Parr C. S., Campbell D. et al.* How Users Interact with Biodiversity Information Using TaxonTree // Proceedings of the working conference on Advanced visual interfaces Gallipoli, Italy. 2004. P. 320–327.
- Lee B., Parr C. S., Plaisant C. et al.* TreePlus: Interactive Exploration of Networks with Enhanced Tree Layouts // IEEE TVCG (Infovis'06 proceedings). 2006. Vol. 12. No. 6. P. 1414–142.
- Lee B., Plaisant C., Parr C. S. et al.* Task Taxonomy for Graph Visualization // Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization. Venice, Italy, 2006. P. 1–5.
- Noack A.* An Energy Model for Visual Graph Clustering // Lect. Notes Comput. Sci. 2004. Vol. 2912. P. 425–436.
- Plaisan C., Grosjean J., Bederson B.* Spacetree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation // IEEE Symposium on Information Visualization. 2002. P. 57–64.
- Sarkar M., Brown M. H.* Graphical Fish-eye Views of Graphs // Human Factors in Computing Systems, CHI '92 Conference Proceedings. ACM Press, 1992. P. 83–91.
- Sarkar M., Reiss S. P.* Manipulating Screen Space with Stretch Tools: Visualizing Large Structure on Small Screen // Technical Report CS-92-42 / Dept. of Comp. Sci., Brown U., Providence, RI, September 1992.
- Six M., Tollis I. G.* A Framework for Circular Drawings of Networks // Lect. Notes Comput. Sci. 1999. Vol. 1731. P. 107–116.
- Shneiderman B.* The Eyes Have It: A Task by Data Type. Taxonomy for Information Visualizations // Proc. of IEEE. Symposium on Visual Languages. Los Alamos, 1996. P. 336–343.
- Slack J., Munzner T.* Composite Rectilinear Deformation for Stretch and Squish Navigation // Transactions on Visualization and Computer Graphics, September 2006. Vol. 12. No. 5. P. 901–908.
- 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. No. 2. P. 109–125.
- Tamassia R.* On Embedding a Graph in the Grid with the Minimum Number of Bends // SIAM. J. Comput. 1987. Vol. 16 (3). P. 421–444.
- Ham F., van, Wijk J. J., van* Interactive Visualization of Small World Graphs // Proc. IEEE Symp. Information Visualization, IEEE CS Press. 2004. P. 199–206.
- Yang J., Ward M., Rundensteiner E. et al.* InterRing: A Visual Interface for Navigating and Manipulating Hierarchies // Information Visualization. 2003. Vol. 2. P. 16–30.

Материал поступил в редколлегию 20.08.2008

Z. V. Apanovich

Navigation Methods for Graph Visualization

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. This paper is a survey of navigation methods for graph visualization. Such techniques as logical filtering, geometric distortion, semantic zooming and clustering are considered.

Keywords: interactive information visualization, graph, navigation, focus + context.