

Визуализация информации на основе графовых моделей*

В.Н. КАСЬЯНОВ

Институт систем информатики им. А.П. Ершова СО РАН

e-mail: kvn@iis.nsk.su

В докладе дается обзор основных существующих методов и систем визуализации информации на основе графовых моделей. Особое внимание уделяется вопросам визуализации информации большого объема и сложной структуры.

1. Введение

Визуализация информации — это процесс преобразования больших и сложных видов абстрактной информации в интуитивно понятную визуальную форму. Универсальным средством такого представления структурированной информации являются графы. Графы применяются для представления любой информации, которую можно промоделировать в виде объектов и связей между объектами. Поэтому визуализация графовых моделей является ключевой компонентой во многих приложениях в науке и технике, а методы визуализации графов представляют собой теоретическую основу методов визуализации абстрактной информации. Методы и средства визуализации графов и графовых моделей широко используются в таких областях, как информационные системы и программное обеспечение, биологические науки, искусственный интеллект, анализ финансовой информации, компьютерное обучение и многие другие.

В зависимости от применения элементы (вершины и ребра) графа должны изображаться различными способами. Например, вершины могут быть нарисованы в виде точек, кругов, прямоугольников или других геометрических фигур, или представлены неявно — через имена, которыми вершины помечены. Аналогично имеется большое разнообразие рисования ребер: например, в виде отрезков прямых, ломаных линий или кривых. Граф может рисоваться на плоскости или в трехмерном пространстве. Он может изображаться целиком, частично или иерархически, например, путем стягивания некоторых его подграфов в вершины, которые могут раскрываться по требованию. Изображения графа могут быть не только статическими, но и интерактивными, поддерживающими различные способы навигации, адекватные потребностям пользователя. Интерактивная визуализация может быть вызвана не только динамическим характером работы с визуальным представлением графа в приложении, но и большим размером визуализируемого графа. Если число элементов графа велико, его обработка может занимать неприемлемо большие ресурсы или даже достигать предельных возможностей используемой для визуализации платформы. Даже если возможно разместить и показать все элементы большого графа, часто возникают проблемы наглядности и удобства, поскольку в таком изображении графа становится невозможным различать его

*Работа выполнена при частичной финансовой поддержке Российского фонда фундаментальных исследований (грант РФФИ N 12-07-0091).

элементы (вершины и дуги) и их взаимосвязи. Интерактивная визуализация превращает статическую демонстрацию визуального представления информации в непрерывный процесс взаимодействия пользователя с информацией через её визуальное отображение и доступные ему навигации. Пользователь может исследовать, рассматривать, открывать, узнавать и манипулировать данными через визуальные метафоры.

Активная разработка методов и средств визуализации графов началась во второй половине 80-х годов прошлого столетия. Это было связано, с одной стороны, с возросшей потребностью оперировать с большими объемами информации и сложными структурами данных, которые достаточно естественным образом представляются графами, а с другой стороны, с существенным прогрессом в развитии аппаратных средств, позволившим сделать графический интерфейс и компьютерную графику удобным и эффективным средством общения человека с компьютером. В настоящее время вопросам визуализации графов посвящена обширная литература (см., например, [1],[2],[3],[4],[5],[6]), а на рынке широко представлены наукоемкие программные продукты, использующие методы визуализации информации на основе графовых моделей.

Исследования в области визуализации графов и графовых моделей проводятся широким фронтом и имеют обширные и все более разнообразные сферы приложения. В работе [5] перечисляются в качестве основных источников, содержащих результаты этих исследований, такие журналы, как Journal of Graph Algorithms and Applications (JGAA), IEEE Transactions on Visualization and Computer Graphics, Computational Geometry : Theory and Applications и Algorithmica, а также труды таких регулярных конференций, как Graph Drawing Symposia (GD), ACM Conference on Human Factors in Computing Systems (CHI), ACM Symposium on User Interface Software and Technology (UIST), IEEE Information Visualization Conference (InfoVis), Eurographics Workshop on Scientific Computing. Такая разбросанность исследований по журналам и конференциям, относящимся к весьма разной тематике, усложняет поиск подходящих методов визуализации при создании систем, использующих визуализацию графовых моделей, а также затрудняет объединение полученных результатов в единую теорию и/или структурированное множество методологий.

В докладе дается обзор основных существующих методов и систем визуализации информации на основе графовых моделей. Доклад начинается с рассмотрения задачи визуализации графов, требований к изображениям графа и основных понятий (изобразительное соглашение, эстетичность и ограничение), связанных с формализацией понятия качественного способа рисования графа. Изучаются вопросы изображения деревьев — важного подкласса графов, широко используемого для представления различных иерархий, таких как генеалогические деревья, организационные схемы или поисковые деревья. Описываются основные методы, позволяющие получать удовлетворительные решения задач статической визуализации графов, такие как планаризация, использование физических аналогий, поуровневые (или Сигаяма-подобные) и потоковые методы. Рассматриваются методы рисования графовых моделей большого размера и сложной структуры, а также вопросы интерактивности и навигации при визуализации графов. Доклад завершается кратким описанием существующих систем визуализации.

2. Задача визуализации графовых моделей

Изображение (drawing) (или, как иногда говорят, *представление*, *рисунок*, *расположение*, *укладка* или *раскладка*) графа на плоскости (или в пространстве) — это отображе-

ние вершин и ребер графа в множество точек плоскости (или пространства). Понятно, что один и тот же граф можно визуализировать разными способами, причем качество одного и того же изображения может по-разному оцениваться разными индивидуумами, а разные приложения могут требовать разные способы визуализации графа. Например, при работе с различными таксономиями часто желательна ортогональное расположение ребер и меток, так как взаимодействие с изображением должно облегчать чтение сопутствующей информации, а при визуализации карт дорог нужно, чтобы расположение вершин и ребер соответствовали географическим реалиям. Поэтому главным критерием оценки качества методов визуализации информации является адекватность изображения графовой модели заданному типу информации и характеру её использования.

Понятие качественного способа рисования графа формализуется с помощью таких понятий, как изобразительное соглашение, эстетический критерий и ограничение.

Изобразительное соглашение — это одно из основных правил, которому должно удовлетворять изображение графа, чтобы быть допустимым. Например, при рисовании блок-схемного представления программ можно использовать соглашение о том, что все вершины должны изображаться прямоугольниками, а дуги — ломаными линиями, состоящими из вертикальных и горизонтальных звеньев. При этом, конкретный вид соглашения реального применения может быть достаточно сложен и включать много деталей, касающихся изображения. Широко используются следующие изобразительные соглашения: *поллинейное* изображение предполагает, что каждое ребро графа рисуется в виде ломаной линии, *прямолинейное* изображение характеризуется тем, что каждое ребро представляется с помощью отрезка прямой, *ортогональное* изображение предполагает, что каждое ребро графа изображается в виде ломаной линии, состоящей из чередующихся горизонтальных и вертикальных сегментов, *сетчатое* изображение предполагает, что все вершины, а также все точки пересечения и сгибы ребер находятся в узлах координатной сетки, образованной прямыми, параллельными координатным осям и пересекающимися их в точках с целочисленными координатами, *плоское* изображение предполагает отсутствие точек пересечения у линий, изображающих ребра, *восходящее* (соответственно *нисходящее*) изображение имеет смысл по отношению к ациклическому орграфу и предполагает, что каждая дуга изображается ориентированной кривой, координаты точек которой монотонно изменяются в направлении снизу вверх (соответственно сверху вниз) и слева направо.

Средства навигации и интерактивности являются существенными также при визуализации информации большого объема. Сам алгоритм размещения графа не может преодолеть проблемы, вызванные громадными размерами визуализируемых графов, возникающих в различных приложениях.

Эстетические критерии специфицируют такие свойства изображений, которые желательнее применять в такой наибольшей степени, как только это возможно, чтобы повысить наглядность изображения. Широко используются следующие эстетические критерии: минимизация пересечений, минимизация сгибов, минимизация области размещения, максимизация разрешения, минимизация общей длины ребер, минимизация длины ребра, унификация длин ребер, минимизация числа сгибов на ребре, унификация сгибов, максимальная симметричность, минимизация коэффициента сторон. Большинство из используемых эстетических критериев, как задачи оптимизации, являются сложными для решения с вычислительной точки зрения. Поэтому обычно при построении изображений используются различные эвристики и стратегии приближенных решений.

Ограничения. Если соглашения и эстетические критерии формулируются по отношению ко всему графу и его изображению, то ограничения относятся к отдельным подграфам и частям изображений. Наиболее часто рассматриваются следующие ограничения: центр, когда требуется разместить заданную вершину ближе к центру изображения, внешность, когда требуется разместить заданную вершину на внешней границе изображения, кластер, когда требуется разместить заданное подмножество вершин рядом друг с другом, последовательность слева-направо (сверху-вниз), когда требуется нарисовать заданный путь горизонтально слева-направо (соответственно сверху-вниз).

3. Визуализация деревьев

Наиболее простым и широко используемым для решения задач визуализации информации подклассом графов являются деревья. Алгоритмы размещения деревьев имеют наименьшую сложность и являются наиболее простыми для реализации.

Корневые деревья часто используются для представления иерархий, таких как генеалогические деревья, организационные схемы или поисковые деревья. К их изображениям часто применяют дополнительные соглашения, эстетические критерии и ограничения. Например, при соглашении *включения* (inclusion) вершины корневого дерева изображаются прямоугольниками, а отношения отец-сын представляются включением одного прямоугольника в другой, а соглашение *опрокидывания* (tip-over) подобно классическому соглашению нисходящего плоского изображения корневого дерева, однако допускает, чтобы сыновья некоторых вершин располагались не горизонтально, а вертикально.

Простой и эффективный метод построения нисходящего плоского изображения корневого дерева T состоит в использовании поуровневого расположения дерева, при котором вершины p глубины i имеют y -координату $y(p'') = -i$, а x -координаты присваиваются таким образом, чтобы разность $x(p'') - x(p')$ имела тот же знак, что и разность $x(\text{отец}(p'')) - x(\text{отец}(p'))$.

Некоторой вариацией поуровневого изображения дерева является его *радиальное* расположение, в котором уровни имеют вид концентрических окружностей, а поддерева занимают секторные сегменты. Алгоритмы, применяемые к корневым деревьям, можно использовать для получения допустимых радиальных изображений свободных деревьев, размещая вершины свободного дерева на концентрических окружностях по уровням относительно теоретико-графового центра дерева.

Гиперболическое размещение деревьев (или других классов графов) является одной из таких форм раскладки, которая разрабатывалась под влиянием вопросов визуализации графов большого размера и интерактивности. Существует два канонических способа отображения гиперболической плоскости на внутренность круга евклидовой плоскости: конформная модель Пуанкаре, которая сохраняет углы, но превращает линии гиперболического пространства в дуги единичного круга, и модель Клейна, которая отображают линии гиперболического пространства в прямые линии на плоскости, но зато искажает углы. В обоих случаях одна окрестность гиперболической плоскости находится в фокусе в центре круга, в то время как остальная часть гиперболической плоскости постепенно исчезает в перспективе по мере приближения к границе круга.

Метод визуализации, получивший название *дерево-карта* (treemap), использует соглашение включения и является крайне эффективным 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 численный атрибут, в виде прямоугольника (или некоторой другой геометрической фигуры) таким образом, чтобы площади изображений вершин дерева были пропорциональны их значениям атрибута.

4. Методы рисования

Большинство методологий рисования графа основывается на следующих двух простых наблюдениях: эстетические критерии часто противоречат друг другу и таким образом поиски компромиссов неизбежны; даже если эстетические критерии не конфликтуют, часто алгоритмически трудно удовлетворить всем им одновременно. Имеется ряд методов, которые позволяют получить удовлетворительные решения задач визуализации графов; основными среди них являются следующие.

Планирование. Плоские расположения графов, т.е. без пересечений рёбер, обычно более привлекательны, чем неплоские. Например, они весьма важны в технологиях печатных плат с точки зрения минимизации размещения.

К сожалению, на практике многие графы не имеют плоских укладок, т.е. не являются *планарными* (planar). Можно попытаться сделать граф планарным, удаляя из него либо как можно меньше ребер (эта проблема является *NP*-полной), либо те ребра, чья вставка впоследствии может создать наименьшее число пересечений. Эта проблема минимизации пересечений является в общей постановке *NP*-трудной, но есть некоторые эвристики, которые позволяют получать вполне удовлетворительные результаты.

Некоторые планарные графы можно нарисовать на плоскости таким образом, чтобы каждая граница грани являлась выпуклым многоугольником. Такое представление возможно для графа только тогда, когда границы всех граней являются простыми циклами. Граф, не являющийся двусвязным, не имеет выпуклого представления, а для любого 3-связного графа существует выпуклое представление (теорема Татта).

Использование физических аналогий. Эти методы интерпретируют граф при построении его изображения как физическую систему с силами между вершинами и пытаются минимизировать энергию системы для получения хорошего рисунка. Такого типа алгоритмы используются для рисования произвольных (разреженных) сетей, таких как блок-схемы, графы программного планирования, графы телефонных вызовов и т.п. Они также применяются для кластерных изображений.

Данный подход достаточно популярен, поскольку физические аналоги, с одной стороны, делают алгоритмы рисования достаточно прозрачными для понимания и простыми для реализации, а с другой, приводят к алгоритмам, дающим весьма хорошие расположения графов. В частности, эксперименты показали, что во многих случаях алгоритм рисования, когда каждое ребро рассматривается как пружина, а вершины считаются одинаково заряженными частицами, между которыми действуют силы отталкивания, строит симметричные изображения графа.

Поуровневые или Сугияма-подобные методы. Наиболее широко используемыми алгоритмами для рисования ациклических орграфов являются алгоритмы, относящиеся к классу, предложенных Сугиямой. Они производят поуровневые (или иерархические) изображения, пытаясь также минимизировать количество пересечений или размер области размещения. Выбор этого подкласса для рисования можно объяснить двумя причинами. Во-первых, преимущественное большинство реальных графов, встречающихся в программировании, являются ациклическими, а, во-вторых, любой ориентированный

(и тем более неориентированный) граф может быть преобразован к ациклическому графу путем смены или задания ориентации у части его ребер.

Данный подход является крайне интуитивным и позволяет разбить задачу нахождения расположения ациклического графа на три достаточно независимых шага, реализация которых опирается на свои методы и использует различные эстетические критерии.

Потоковые методы. Проблема минимизации числа сгибов может эффективно решаться путем сведения ее к задаче потока в сети, по крайней мере в тех случаях, когда зафиксирована топология размещения. Те же самые методы могут применяться для максимизации углов между ребрами.

5. Визуализация больших графов

Размер визуализируемого графа является ключевой проблемой при визуализации графов. Если число элементов графа велико, его обработка может выходить за границы производительности или даже достигать предельных возможностей используемой для визуализации платформы. Даже если возможно разместить и показать все элементы большого графа, возникают проблемы наглядности и удобства, поскольку становится невозможным различать вершины и дуги. По существу, проблема удобства возникает даже раньше, чем проблема различия элементов. Хорошо известно, что охватывающий и детальный анализ данных в графовой структуре наиболее прост, когда размер демонстрируемого графа небольшой.

Очевидно, что выбор подходящего изображения ребер может иногда уменьшить *запутанность* (clutter) изображения. В частности, это можно сделать за счет замены полилинейных линий, изображающих ребра, на гладкие кривые. *Конфлюэнтное рисование* (confluent drawing) ребер, *рёберная кластеризация* (edge clustering), *рёберные связки* (edge bundles) и *выборка* (sampling) — другие подходы к уменьшению визуального загромождения изображения.

Вообще говоря, нет смысла проверять планарность графа, состоящего из нескольких сотен вершин, и пытаться минимизировать количество пересечений ребер. Часто более очевидным и практическим решением является простое размещение покрывающего дерева графа и добавление оставшихся ребер к уже имеющемуся изображению.

Другим популярным подходом к визуализации больших графов является переход от 2D к 3D изображению. Дополнительное третье измерение позволяет более гибко размещать элементы графа и вообще обходиться без пересечений ребер, но приводит к новым проблемам, поскольку современные устройства вывода по своей природе являются двумерными и поддерживают ограниченные разрешение и область визуализации.

При работе с большими графами часто используются специальные методы, позволяющие уменьшать количество деталей, размещаемых одновременно. Вместо статического размещения также используются различные интерактивные методы с применением навигации и методов выделения (фокус+контекст), включающих геометрическую или семантическую деформацию, кластеризацию, агрегацию и другие техники.

В 1986 году Фурнас описал класс методов, в которых вершины иерархической структуры автоматически включаются или удаляются из изображения в соответствии автоматически вычисляемой степенью пользовательского интереса, зависящей от априорной значимости данной вершины и её расстояния в дереве от вершины, выбранной пользователем в качестве фокуса. 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 1992 году Саркар и Браун обобщили данный подход на случай произвольного графа, предложив использовать в качестве функции расстояния до фокуса — евклидово расстояние между вершинами уже существующего изображения.

Кластеризация (clustering) в общем случае — это процесс разбиения заданной выборки объектов (ситуаций) на подмножества, называемые кластерами (clusters), так, чтобы каждый кластер состоял из схожих объектов, а объекты разных кластеров существенно отличались. Кластеризация позволяет уменьшать количество видимых элементов графа с сохранением при этом его глобальной структуры. Конкретным примером, где её применение весьма оправдано, являются так называемые графы малых миров (small-world graphs) — графы, которые имеют небольшой средний кратчайший путь между вершинами, но высокую степень кластеризации (по сравнению с усреднённым графом соответствующего размера). Это свойство присуще многим графам реальных приложений, таких как социальные сети, нейронные сети, программные системы, сети электроснабжения, базы знаний с перекрестными ссылками и Интернет.

6. Интерактивная визуализация

Рассмотренные выше методы используются для рисования статических графов. Однако, есть ряд применений, требующих построения интерактивных изображений графов. Среди них: инструменты отладки, сохранение документов, модули отношений сущностей, схемы СБИС, а также Web-графы.

Средства навигации и интерактивности являются существенными также при визуализации информации большого объема. Сам алгоритм размещения графа не может преодолеть проблемы, вызванные громадными размерами визуализируемых графов, возникающих в различных приложениях.

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 исполнения одного и того же алгоритма на одних и тех же или похожих данных давали похожий результат.

Обычно процесс создания интерактивной визуализации распадается на два последовательных этапа. Первый этап состоит в отображении исходных данных на геометрическую плоскость в виде статического графового изображения. Содержанием второго этапа является собственно навигация, обеспечивающая пошаговое изменение изображения и позволяющая достичь той формы представления информации, которая нужна пользователю в данный момент. Как правило, на этом этапе применяются различные методы поддержки взаимодействия пользователя с графовым изображением, построенном на первом этапе, но иногда реализация второго этапа — это многократное выполнение первого этапа с измененными требованиями к получаемому графовому изображению. При этом помимо стандартных методов навигации, таких как открытие (и закрытие) новых окон, масштабирование, прокрутка или инкрементальный просмотр, широко используются специальные методы взаимодействия с графовой информацией, такие как фильтрация, выделение, кластеризация и различные деформации.

Методы деформации *фокус+контекст* (focus+context) предназначены для взаимодействия с изображениями большого объема и позволяют совместить в одном изображе-

нии глобальный вид всей представленной структуры (графа, дерева, меню, календаря, таблицы и т.д.) и детали некоторого её фрагмента, находящегося в фокусе. Другой пример деформации — так называемая метафора *резинового листа* (rubber-sheet), при которой дисплей действует как резиновый лист с закреплёнными границами, в котором растягивание одних областей приводит к сморщиванию других.

В процессе взаимодействия пользователь может иметь возможность открывать и закрывать новые окна для рассмотрения тех или других частей визуализируемого графа, а также осуществлять различные манипуляции с изображением, представленным в том или в другом окне. Часто оказывается полезным предоставлять пользователю отдельное окно для общего изображения большого графа с показом на нем частей графа, визуализируемым в других окнах.

7. Системы визуализации

Преимущества представления сложных структур и процессов графами становятся более ощутимыми при наличии хороших средств их визуализации. Поэтому неслучайно в последнее время в мире растет интерес к созданию и использованию систем визуализации графовых моделей, которые условно можно разделить на два класса.

К первому (более широкому) классу относятся узкоспециальные системы, которые ориентированы на графовые модели с определенной семантикой и топологией.

Второй класс — это универсальные системы визуальной обработки графовых моделей, такие как daVinci, GraphEd, Graphlet, GraVis, VCG, GLT, Higes, aiSee, yEd.

Наряду с графовыми визуализаторами и графовыми редакторами широко распространены и используются различные библиотеки классов, такие как LEDA, fGraph, Graph Drawing Server. Они позволяют описывать семантику графовой модели путем создания производных классов с дополнительными атрибутами и обычно включают множество других полезных функций, связанных как с визуализацией графов, так и с выполнением на них различных теоретико-графовых алгоритмов и алгоритмов рисования.

Список литературы

- [1] Касьянов В. Н., Евстигнеев В. А. Графы в программировании: обработка, визуализация и применение. — СПб.: БХВ-Петербург, 2003. — 1104 с.
- [2] Касьянов В. Н., Касьянова Е. В. Визуализация графов и графовых моделей. — Новосибирск: Сибирское Научное Издательство, 2010. — 123 с.
- [3] Di Battista G., Eades P., Tamassia R., Tollis I.G. Graph Drawing: Algorithms for Visualization of Graphs. — Prentice Hall, 1999. — 397 p.
- [4] Drawing Graphs. Methods and Models.— Berlin: Springer, 2001. — 312 p. — (Lect. Notes Comput. Sci.; 2025).
- [5] Herman I., Melancon G., Marshall M. S. Graph visualization and navigation in information visualization: a survey // IEEE Trans. on Visualization and Computer Graphics. — 2000. — Vol. 6, N 1. — pp. 24–43.
- [6] Kasyanov V.N. Kasyanova E. V. Information visualization based on graph models // Enterprise Information Systems. — 2013. — Vol. 7, N 2. — pp. 187–197.