

Верификация визуализации

Дмитрий Манаков, Владимир Авербух

Институт математики и механики имени Н.Н. Красовского УрО РАН,

Уральский федеральный университет

г. Екатеринбург

Аннотация

Статья посвящена проблемам верификации компьютерной визуализации на базе формальных или формализованных подходов. Верификация (проверка правильности) визуализации подразумевает наличие формальной модели, которая отличается от подобных моделей для слабо формализуемых явлений (например, от формальной верификации программного обеспечения), по большому счету, только предметной областью. В статье отражены возможные подходы к формированию моделей визуализации. Подробно рассмотрено применение множественного (бинарного) вида отображения для трех задач в области научной визуализации, информационной визуализации и визуализации программного обеспечения. Также рассматриваются возможности формализации визуализации и параллельной фильтрации данных на основе лингвистического и семиотического подходов.

Ключевые слова:

Верификация визуализации, валидация визуализации, анализ неопределенности

1. Введение

Проблемы верификации и валидации визуализации проявились в связи с развитием компьютерной визуализации, как самостоятельной области исследований. Внимание к этим проблемам вызвано, прежде всего, практикой использования визуализации в задачах компьютерного моделирования. Визуализация обеспечивает этап анализа и интерпретации данных, полученных в ходе моделирования. От точности и достоверности визуализации во многом зависит результат всей работы.

На первом этапе развития методов и средств визуализации многим казалось, что сам факт графического представления данных гарантирует быструю интерпретацию и верное понимание сути явлений. Однако достаточно быстро выяснилось, что визуализация может оказаться источником заблуждений и ошибочных решений. В начале 1990-ых годов была опубликована работа “*13 Ways to Say Nothing with Scientific Visualization*” [1], в которой в ироничной манере (в виде «вредных советов») рассматриваются актуальные на тот момент причины неудачного использования средств научной визуализации при отображении результатов компьютерного моделирования. Несколько позже были опубликованы сходные статьи, в частности, работа с ярким названием “*How NOT to Lie with Visualization*” [2], в которой среди прочих вопросов поднимается проблема выбора методик визуализации, обеспечивающих правильное отображение данных.

Среди прочих “советов” неудачливым визуализаторам авторы [1] писали об использовании сглаживания, как источника скрытия “неудобных” подробностей и особенностей решения. Также “советовали” не использовать универсальные системы визуализации, если надо обеспечить “красивый”, но малоинформативный, если не ошибочный графический вывод. Интересно, что в свое время мы получили несколько иной опыт по разработке специализированной системы визуализации. В задании заказчиков было указано, что система не должна использовать традиционные алгоритмы сглаживания, чтобы выявить важные особенности решения дифференциальных игр [3], тогда как наличие стандартного сглаживания в универсальных системах визуализации эту особенность увидеть не позволяла.

Публикации 1990-ых годов до сих пор представляют интерес в связи с обеспечением достоверной визуализации. В 2000-ых годах был опубликован ряд работ,

посвященных проблемам достоверности на основе учета факторов устойчивости и неопределенности визуализации и процессов рендеринга. В 2008 году была опубликована работа [4], в которой были поставлены проблемы валидации и верификации визуализации в связи с общими проблемами валидации и верификации компьютерного моделирования. Кроме этого в [4] рассмотрены источники ошибочных случаев визуализации, связанных с выбором тех или иных методик генерации графических выводов.

Начиная с 2012 года, проходят Европейские семинары по проблеме воспроизводимости, верификации и валидации в визуализации (The EuroRV3: EuroVis Workshop on Reproducibility, Verification, and Validation in Visualization) [5]. Анализ материалов прошедших семинаров показывает, что сообщения семинаров в целом охватывают следующие вопросы:

- устойчивость и неопределенность процессов визуализации,
- роль методик генерации изображения в обеспечении правильности визуализации,
- влияние “человеческого фактора”, включая восприятие и интерпретацию графических выводов.

Также проводятся исследования в области формализации и верификации средств человеко-компьютерного взаимодействия [6-8].

Визуализация является составной частью цикла компьютерного моделирования, наравне с разработкой математической модели и программированием. Любая из этих частей должна быть должным образом верифицирована. В среде исследователей, занимающихся теоретической информатикой, широко распространено узкое понимание термина «верификация» — только как формальной (доказанной математически) верификации. В дальнейшем под формализацией визуализации мы будем понимать именно ее математическую формализацию. Возможно и более чистое (узкое) рассмотрение теории визуализации в рамках конкретной науки, например, нейрофизиологии, когнитивистики или семиотики. Отметим, что в любом случае формализация не является самоцелью.

При рассмотрении компьютерной визуализацию, как самостоятельной дисциплины, необходимо построение ее ментального пространства со своей семантикой, прагматикой и базисом. Тогда любые два специалиста по визуализации смогут говорить на одном языке. Этот базис выбирается из достаточно широкой области знаний. Сейчас популярна идея карты той или иной области знаний и даже всей науки [9]. В нашем случае можно составить карту визуализации, где отображаемые факторы представляют собой не сумму знаний, а скорее их произведение, создавая, тем самым, новую сущность. На этой карте, кроме географической компоненты (например, указания, где работает тот или иной исследовательский коллектив), необходимо отображение и других, когнитивных данных. Например, не просто списков терминов или ключевых слов, используемых в публикациях, но их соотношение с другими подобными понятиями, разделенными, как географически, так и дисциплинарно. Для этих целей может служить метафора частично упорядоченного облака тегов [10]. Также возможно использование нечетких когнитивных моделей, основанных на формализации причинно-следственных связей между факторами (переменными, параметрами), характеризующими исследуемую систему. В последнем случае результатом формализации является представление системы в виде причинно-следственной сети, называемой нечеткой когнитивной картой [11].

2. О терминах и моделях

Верификация (проверка правильности) визуализации подразумевает наличие формальной модели, которая отличается от подобных моделей для слабо формализуемых явлений (например, от формальной верификации программного обеспечения), по большому счету, только предметной областью. Задача визуализации состоит в интерпретации результатов моделирования различных процессов и явлений. В рамках компьютерной визуализации востребована верификация задач представления больших

объемов сложно структурированных данных (big data). Эти задачи тесно связаны с восприятием человека и с когнитивными аспектами визуализации и поэтому слабо формализуемы.

На начальном этапе развития компьютерной визуализации были предложения использовать формальные оценки семантической информативности на основе анализа логических систем с использованием понятий “модели-интерпретации” и “модели и полумодели” обобщенной логической системы [12]. Близкий в какой-то мере подход к формализации связан с идеями ментальной модели [13], на основе которой возможна формальная верификация. В настоящей работе рассматривается формализация уровня полумодели, что, необходимо, в первую очередь, для понимания разработчиков систем визуализации. Кроме этого, ставится задача показать возможности применения в области визуализации стандартных в математическом плане решений, применяемых для других слабо формализуемых явлений.

Интерпретация результатов визуализации и интерактивных манипуляций восстанавливает (или создает заново) у пользователя набор когнитивных структур, в которых представляется картина явлений. Порождение таких структур по визуальным образам и есть процесс интерпретации. Этот процесс является обратным или, точнее, двойственным визуализации [14]. Очевидно, что любая формализация повышает уровень абстрактности, в то время как визуализация нацелена на снижение этого уровня через интерпретацию [15]. Метафора визуализации определяется как отображение понятий и объектов моделируемой прикладной области на некоторый изобразительный ряд. Использование метафор может обеспечить понижение уровня абстракции.

Существуют подходы к формализации визуализации на базе семиотического подхода и “смешанного”, основанного на использовании идей семиотики и теории категорий [16]. Выполнен семиотический анализ метафор и возможности их использования для представления информации [15]. Можно говорить о метафорической и/или семиотической верификации визуализации.

Кроме этого, существует, так называемая, наивная верификация компьютерного взаимодействия и визуализации, основанная на результатах тестирования или наивного байесовского подхода. В этом качестве можно рассмотреть традиционные в литературе требования обеспечения интуитивно понятного интерфейса. Интерфейс становится интуитивно понятным в случае, если его использование опирается на предыдущий опыт пользователя. Однако часто в реальных системах массового обслуживания пользователь сталкивается со сложностями выбора своих действий, так как цели интерфейса жестко детерминированы. Следовательно, свобода выбора при проектировании интерфейсов ограничена. “Интуитивно понятный интерфейс” может рассматриваться лишь как эталон низкого уровня, впрочем, как и естественный интерфейс. Заметим при этом, что знания низкого уровня кажутся очевидными или банальными, но это не означает, что они могут быть исключены из рассмотрения.

Верификация, как сравнение с эталоном широко применяется в научной визуализации, например, при разработке инженерных пакетов. Действительно, трудно представить, чтобы при решении одних и тех же уравнений одними и теми же методами, применении одинаковых видов отображения, получались бы разные изображения. Однако в принципе эта ситуация может возникать в результате накопления вычислительной погрешности. В работе [4] вводится понятие верифицируемой (верифицирующей) визуализации, которая отслеживает, как распространяется погрешность (неопределенность) на всем этапе вычислительного конвейера, включая визуализацию. Подобный подход в общем случае принято называть моделью с неопределенностью, а в частном - визуализацией с неопределенностью (Рис. 1), которую можно рассматривать, как пример некорректной задачи по начальным данным. Термины неопределенность и некорректность в данном случае можно рассматривать в качестве синонимичных.

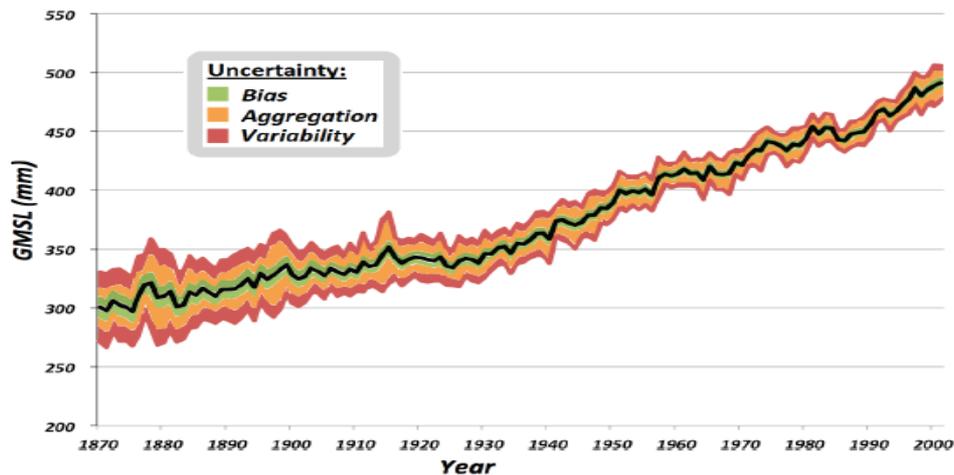


Рис. 1 Пример визуализации с неопределенностью [17].

Суммируя вышесказанное можно выделить качественные и количественные характеристики верификации или ввести две базисные функции или меры. Качественная – это степень формализации (наивная, метафорическая, формальная) и количественная, связанная с распространением неопределенности. Эти функции уместно именовать полнотой и точностью верификации.

Одновременно с верификацией визуализации необходимо рассматривать валидацию визуализации [4]. Валидация – это мера адекватности. В математическом моделировании адекватность определяется как соответствие теории (формальной модели) практике. Так формально правильная модель может быть неадекватной. Частую путаницу в определении верификации и валидации можно объяснить достаточно просто. На наш взгляд, наивная верификация эквивалентна валидации.

Ключевые вопросы относительно неопределенности в вычислительной науке, можно сформулировать в контексте визуализации как это сделано в работе [17]:

- Может ли быть визуализация надежным основанием для принятия решений?
- Как точность и валидация (адекватность) визуализации могут быть оценены?
- Какая уверенность или неопределенность может быть присвоена визуализации?

Рассматривая визуализацию, как решение обратной некорректной задачи, необходимо ответить на следующие вопросы.

- Существует ли решение?
- Является ли решение устойчивым? К этому же пункту отнесем и проверку на обусловленность или чувствительность (sensitivity).
- Является ли решение оптимальным?

В рамках нечеткой логики не приходится говорить об оптимальности решения, но можно оценить является ли предложенное решение по сравнению с известным, более или менее эффективным. По сути, это решение задачи минимакса. Эффективность – это скорость сближения двух решений или, в частном случае, порядок сходимости, как принято в вычислительной математике. Также очевидно, что эффективность это функция многих переменных. Она зависит, как, например, в случае параллельных вычислений, не только от количества данных, но и от количества процессоров. В каких-то случаях эффективность использования техники зависит от знаний и умений пользователя (человеческий фактор). В случае параллельных вычислений, кроме быстродействия, возможно рассмотрение нескольких метрик эффективности, например, эффективности по энергопотреблению. Используемое в экономике определение эффективности (продуктивности) через отношение продукта к источнику продукта (ресурсам) достаточно адекватно. Формально, это скорость или полный дифференциал. Также возможно рассмотрение данного отношения и с позиций нечетких множеств, так как экономическая

эффективность всегда меньше или равна единице, например, из-за того же закона сохранения массы.

Таким образом, непротиворечивая оценка эффективности возможна только на основе формальной модели. Это означает, что формальная верификация визуализации востребована не только для проектирования визуализации, но и для оценки ее эффективности. То есть, необходима формальная модель для оценки эффективности, в том числе и для оценки эффективности визуализации.

Эффективность в визуализации часто имеет субъективный характер. Это объясняется тем, что в визуализации выделяются, как минимум два действующих лица (актера): разработчик - автор визуального текста и пользователь – интерпретатор [15]. В деятельностном подходе один из ключевых вопросов – это определение цели. В математическом моделировании целеполагание рассматривается, как условие экстремальности. Очевидно, что у разработчика и пользователя цели могут не совпадать.

В качестве примера “наивного анализа” рассмотрим интерфейсы банкоматов и платежных систем. Разработчики заинтересованы, в первую очередь, в проверке правильности вводимой информации. В этом смысле применение древовидной структуры интерфейса оправданно. Проверка правильности ведет к увеличению длины дерева запроса. Для пользователя увеличение длины дерева запроса увеличивает вероятность ошибки. В лучшем случае при ошибке ему приходится начинать все сначала. Кроме того, для борьбы с «ботами» в платежных системах применяется изменение порядка действий и дополнительные запросы информации с картинок. Для пользователя важным фактором является быстрое действие и отсутствие стресса, вызванного отказом от обслуживания. Можно элементарно расширить интерфейс, введя запоминание профиля пользователя и выполнения соответствующих операций. Например, всегда снимать все деньги со счета. Наличие разных целей, скорее всего, приведет к возникновению несовместных условий, поэтому при формализации таких интерфейсов желательно рассматривать корпоративные игры.

В принципе верифицируемая визуализация то же самое, что и визуальная верификация, но с разным целеполаганием. В первом случае разработчик нацелен на то, чтобы предложить правильную визуализацию, а во втором на то, чтобы доказать правильность визуальными средствами.

К сожалению, нет общепринятой модели формальной верификации визуализации, включая терминологию, так как наследуются они из разных математических дисциплин [17]. Например, визуализацию рассматривают с позиции общей теории информации [18]. В этом случае используются два формализованных термина или две базисных функции - информативность и избыточность. В других работах по визуализации информативность – это наглядность, а избыточность – это выразительность. Причем доказать эквивалентность этих терминов не представляется возможным из-за частого отсутствия формальных определений, в отличие от общей теории информации. Возможно, родоначальником данных терминов является математическая логика, теоремы Гёделя о полноте и неполноте соответственно. Поэтому идея создания онтологии знаний (информации) востребована для оценки достоверности информации. Можно ввести, например, три вероятностные меры: онтология (происхождение определения в смысле неполной информации), гносеология (смысл), целеполагание (соответствие цели). Считая их независимыми, меры можно перемножить. Если полученное значение больше одной второй, то информацию можно считать достоверной. Данный подход аналогичен методу голосованию, но в котором один эксперт голосует за разные меры. Также очевидно, что мера достоверности (надежности) является двойственной мерой неопределенности. Необходимо отметить, что любое декларативное определение является незамкнутым по сравнению с формальным. (Напомним, что в теории категорий замыкание является одним из необходимых условий.) Следовательно, можно ввести бесконечно много мер оценки декларативного определения.

С одной стороны, чем больше мер оценки, тем лучше, но практически, их количество должно быть ограничено.

Суммируя вышесказанное, можно констатировать, что именование базисных функций несущественно по сравнению с тем, какими свойствами обладают эти функции. Выделение свойств можно рассматривать, как категоризацию. Формально, объекту ставится в соответствие вектор параметров. Проблема в том, что ни длина вектора, ни значения параметров точно не известны, впрочем, и имя объекта тоже приблизительное. Таким образом, необходимо использовать, как минимум вероятностную модель. Можно привести и другие варианты модели с неопределенностью, например, информационный разрыв (Info-Gap) теории принятия решений, который помогает в принятии решений в условиях неопределенности [19]. Это осуществляется на базе трех этапов моделирования, каждый из которых опирается на предыдущий: Первый начинается с моделирования ситуации, когда некоторые параметры неизвестны; Следующий оценивает параметр, который предполагается неправильным (неточным). Последний анализирует, насколько чувствительны (обусловлены) результаты модели к ошибке в этой оценке. Таким образом, верифицируемая визуализация должна обеспечить оценку участков чувствительности и робастности по параметрам модели в областях достижимости, оценивание аппроксимационных процедур управления или верификацию и анализ чувствительности результатов моделирования. В качестве примера визуального анализа чувствительности можно рассмотреть работы [20, 21], сравнительно недавно опубликованные в журнале “Научная визуализация”.

Первый вопрос, на который необходимо ответить, это вопрос о количестве базисных функций или о размерности пространства. В визуализации, кроме координат и времени, значимыми могут быть цвет, форма, структура объекта, знаки, символы и т.п. Поэтому термин – когнитивная размерность, используемый, например, в “usability” можно считать удачным в качестве дополнительной размерности, например, к декартовым координатам. Когнитивная размерность определяется количеством эвристик или термов, которые изначально считаются ортогональными или независимыми [22]. В данном случае близкими к терму можно рассматривать понятия *категория*, *класс*, *концепт*. Так в работе [18] рассмотрено применение при визуализации параллельных координат. В данном случае для экономической или затратной модели эффективность, являющаяся функцией многих переменных, отображается последовательно по каждой переменной и непрерывно. Приведенный в работе график эффективности, даже в случае рассмотрения полного дифференциала, является не совсем корректным, но дает представление о линейной зависимости между переменными.

Эвристический подход применим только, в рамках апостериорной неопределенности, например, для оценки качества или адекватности визуализации, понятие терм используется, как правило, в рамках аксиоматического подхода или априорной неопределенности. Вообще говоря, никто не гарантировал, что выбранные эвристики являются базисом, тем более ортогональным. Возникает стандартная (в частности, для математического распознавания образов, включая симплекс метод [20]) задача выбора информативных признаков – построение отображения $X^n \rightarrow X^m$, где n – количество критериев, m – размерность пространства. Также может быть построена “слабая” шкала по двоичным критериям таким как “хорошо–плохо”, “красиво–некрасиво”... Важно подобрать существенные критерии. Если средние значения (математическое ожидание) для двух эвристик близки, то и эти критерии оценки близки (они коррелируются). Следовательно, одну из эвристик можно не учитывать, исключить из базиса. Если нет дисперсии, то эвристика не существенна. Если эвристики подобраны удачно, есть повод перейти к аксиоматическому подходу. В этом смысле в визуализации сначала решается обратная некорректная задача, затем результат проверяется на прямой задаче (аксиоматический подход). Хотя применение аналогий никто не запрещает, прямая задача проще.

Вопрос о нормальности кажется очевидным, если не учитывать онтологические особенности. Так нормированную базисную функцию можно рассматривать, как меру, но термин базисная функция используется в функциональном анализе и в топологии, а понятие меры ассоциируется с теорией вероятности. Как более конструктивный пример, рассмотрим эквивалентность определений функции принадлежности (теория нечетких множеств) и вычислимой функции (вычислительная математика). Вопрос об определении алгоритма, по существу, равносителен вопросу об определении вычислимой функции. Вычислимые функции — это множество функций вида, $f: N \rightarrow N$, которые могут быть реализованы на исполнителе машин Тьюринга. Функция принадлежности: $\mu_x(x) \in \langle 0, 1 \rangle$ переводит множество само на себя, задает степень принадлежности элемента множеству, в отличие от классической теории, когда элемент точно принадлежит множеству, по сути, упорядочивает множество. Если область определения конечно дискретное множество, очевидно, что можно построить отображение на область определения натуральных чисел для вычислимой функции. По сути, функция принадлежности - это нормированная вычислимая функция, если не учитывать онтологические особенности. Исходя из эквивалентности определений, очевидно, что вычислительную математику, впрочем, как и визуализацию, можно рассматривать с позиций теории нечетких множеств или теории монотонной меры. Но остается актуальным вопрос о том, получим ли мы в результате подобных подходов новое знание? В вычислительной математике, скорее всего - нет, так как данная предметная область обладает высокой степенью формализации за исключением параллельных вычислений. В визуализации, скорее всего - да, так как данная предметная область имеет низкую или начальную степень формализации. Следовательно, для онтологии знаний желательно ввести меры новизны и актуальности, чтобы осветить на вышеизложенные вопросы.

Вопрос об ортогональности базисных функций или независимости случайных величин только кажется очевидным. Эта ортогональность востребована, например, при операции умножения. Если базисные функции не ортогональны, их можно разложить в ряд по формуле Тейлора или представить в аффинной форме (первой степенью многочлена, где первая производная заменена на “оператор отношения, связывающий числитель со знаменателем, в духе обозначений нечетких множеств, введенных Заде” [17]), а затем перемножить. Проблема как раз в том, что для слабо формализуемых явлений, как правило, отсутствует дифференцируемость, как следствие неаддитивность, нелинейность пространства. Все подобные рассуждения не отличаются высокой степенью прозрачности или инсайта, так как построены на отрицание и требуют переосмысления основных понятий, начиная с базиса, хотя бы с тех же позиций неопределенности. Определение NP-полной задачи также построено на отрицании. В этом смысле возникновение паранепротиворечивой логики, когда отрицание задается специфическим образом, не случайно, хотя прикладное значение подобной логики не очевидно. Если бы базисные функции были бы дважды дифференцируемы, можно было бы рассматривать задачу оптимального управления. Существование второй производной, также является, необходимым условием в вычислительных методах. Аналогичные проблемы возникают и с мерами: как посчитать условную вероятность, если зависимости не известны.

Довольно часто в литературе отмечается, что какие-либо меры или данные являются независимыми или тесно связанными. Такое разделение с двумя исходами возможно только в рамках Булевой логики. В любой другой логике, необходимо определить эти зависимости. Например, с точки зрения пропозиционной логики вычислительной математики можно определить эти зависимости через разложение в ряд. Можно рассмотреть функцию – “степень зависимости данных”. Если данные независимы, то степень зависимости равна нулю. Если зависимости есть, но они статические, то первая производная этой функции по времени равна нулю. Если зависимости динамические, то первая производная не равна нулю. В частности подобный подход применяется для классификации нейронных сетей по характеру настройки синапсов. Математический

аппарат различных логик, довольно развит, поэтому классификация по типу логики обоснованна. Также стандартными являются классификации по типу алгебры. Подобную классификацию можно найти в смежной области – верификации программного обеспечения [23].

На наш взгляд, рассматривать метафору, как “непрерывное” отображение исходного множества на целевое множество можно и с точки зрения топологии с целью достижения наибольшей общности в формализации. В частности, в топологии поднимается вопрос: равняется ли размерность топологического произведения сумме размерностей сомножителей, связанный с наследованием свойств. С точки зрения визуализации, желательно избегать ситуации, когда размерность произведения больше суммы размерностей сомножителей. Так применение трехмерного вида отображения для квазидвумерных данных не оправданно. В качестве примера можно привести применение трехмерной метафоры молекулы (См. Рис. 2) для представления “плоского” графа вызова. Идентификация функций может рассматриваться, как дополнительная когнитивная размерность к размерности декартовых координат. В данном примере вместе с масштабируемостью (информативностью) растет и избыточность изображения. Правда, некорректно требовать от разработчиков систем визуализации знание топологии, тем более, что применение теорем все равно будет не строгое, но элементарные (базисные) знания могут быть полезны, хотя бы, как аналогии. Кроме того, хотелось бы избежать тавтологий (в смысле доказательства теорем) при чтении подобной литературы.

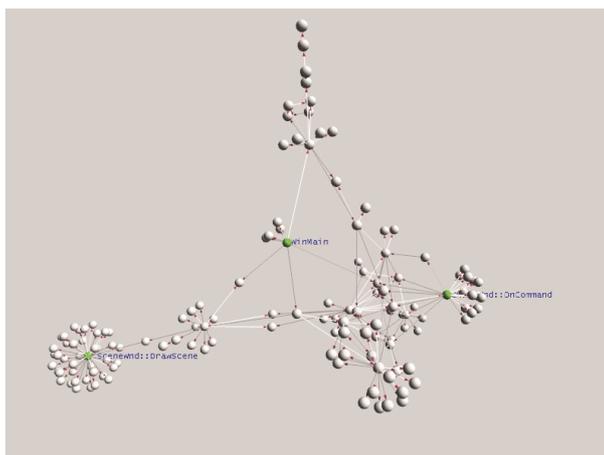


Рис. 2: Граф вызовов на базе метафоры молекулы [24].

Вопрос о существовании решения связан в первую очередь с топологией пространства. Топология равномерной сходимости, топология поточечной сходимости и компактно-открытая топология метризуемы [25]. В визуализации, как и в теории принятия решений, рассмотрение компакта обусловлено моделью с неопределенностью. Для существования решения достаточно выполнения свойства монотонной сходимости, в ряде случаев проще построить монотонно сходящуюся последовательность решений, например для задачи управления или взаимодействия, чем рассматривать компакт.

В области визуализации авторам известны три абстрактные метрики, связанные с процессом познания:

- * информационный разрыв – расстояние между тем, что мы знаем и тем, что должны знать, что бы решить поставленную задачу [19];
- * когнитивное расстояние – усилия пользователя, затраченные на интерпретацию [26];
- * инсайт (озарение, понимание) – цель визуализации, установление релевантных отношений между данными и существующей областью знаний [27] (целевой областью).

Необходимо отметить, что эти метрики являются опосредованными с точки зрения проверки адекватности. Так информационный разрыв можно измерять в количестве информации, усилия в калориях, а инсайт в изменение частоты пульса. Как

опосредованные метрики они совместимы с топологией числовой прямой. Возникает вопрос, если эти метрики независимы или более правильно не связаны между собой, то можно ли их считать эквивалентными? Или можно ли провести аналогию с утверждением в работе [25], что тонкая топология не зависит от выбора метрики. Или другими словами, что первично топология задачи или метрика.

Важность данного вопроса можно продемонстрировать на примере систем мониторинга параллельных программ, которые позволяют сбор и анализ различных метрик, совместимых с топологией числовой прямой. Инструментарий таких систем предполагает, что эти метрики изначально не зависимы (не связаны между собой), следовательно, пользователь всегда останется в рамках топологии графика, или другими словами для визуального анализа невозможно применить другой вид отображения, кроме вариаций графика. Таким образом, для повышения уровня инсайта, необходимо разрабатывать формальные модели как исходной, так и целевой областей, отслеживающие зависимости.

В области формальной верификации визуализации можно рекомендовать к изучению две работы [17] и [28], основанные на теории монотонной меры. По сути, это попытки применения теории вероятности, но в другом, топологическом (не вероятностном) пространстве, скорее всего, на бикомпактах. Кроме теории монотонной меры стоит упомянуть достаточно популярные в ряде прикладных областей информационный разрыв теории принятия решений [19] (по нашему мнению близкий к теории возможностей) и грубые множества [29].

Как утверждается в работе [17] теорию монотонной меры можно рассматривать, как обобщение теории очевидности, теории возможности и теории нечетких множеств. Следовательно, в визуализации с неопределенностью можно рассматривать три независимых типа неопределенности из различных источников, как аддитивные по отдельности. Также в работе вводится понятие функции отклонения или девиации, как расширение понятия нечеткого числа для случая больше, чем числовая прямая. Для наглядности можно сравнить Рис. 1 и Рис. 3. Как уже упоминалось, операция умножения задается в рамках аффинной арифметики, что обеспечивает инвариантность относительно аффинных преобразований изображения.

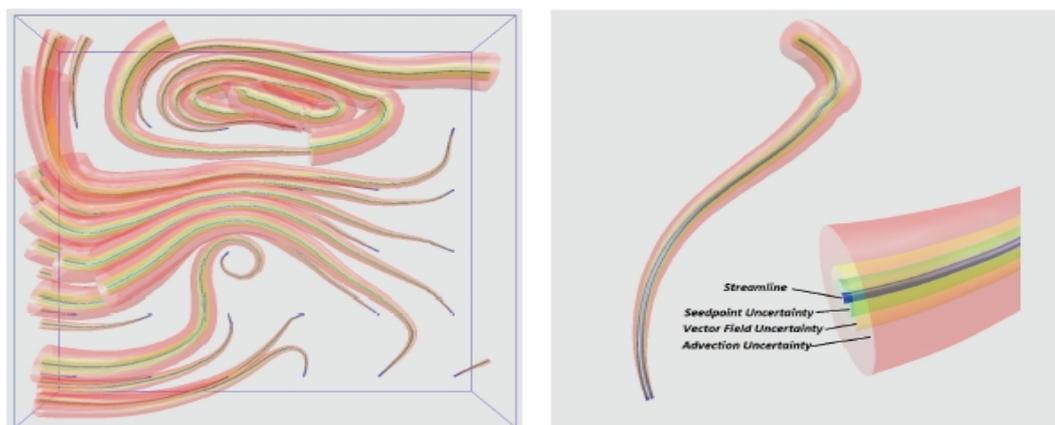


Рис. 3. Линии тока для набора данных по ураганам (дополнено неопределенностью) [17].

В работе [17] рассмотрены перспективы исследования приложения данной структуры для анализа чувствительности. Отмечается необходимость исследования возможности использования анализа встроенной неопределенности для ее управления, а не только для ее отображения. Обсуждаются также возможность применения этого подхода в области сжатия или фильтрации данных. Конечно, задача управления более интересная, но и более сложная. Так существует достаточно широкий пласт работ по

адаптивному интерфейсу, который можно рассматривать, как частный случай автономных вычислений, связанный с самооптимизацией под конкретного пользователя. Другой пример задачи управления – это планирование очереди задач.

Стоит отметить, что обозначения в работе [17] не совсем привычны, по сравнению с работой [28], но, впрочем, понятны. Практическая часть работы [28] связана с задачами компьютерного зрения, в частности с распознаванием контура. Решение этой задачи может быть получено из закона сохранения импульса, используя аддитивность массы и понятие энтропии, без теории монотонной меры. С другой стороны рассмотрение визуализации в топологии непрерывной сходимости также может быть неоправданным. Уравнение, в котором производная от знаний равняется функции от имиджа и знаний, в общем случае не имеет решения. В частном случае можно сделать вывод о применимости модели селекции к визуализации. Поэтому формализация тут не является самоцелью. Цель можно определить, например, в рамках когнитивной психологии, как построение ментального пространства визуализации со своей семантикой прагматикой и базисом.

Понятие нечеткого числа активно применяется в моделях с неопределенностью, в том числе и в визуализации [17], в рамках любого стандартного вычислительного метода. Стоит отметить, что сходимость этих методов была доказана не в рамках компактно-открытой топологии, а в топологии поточечной сходимости. Происхождение нечеткого числа проще всего интерпретировать с точки зрения теории возможности. Нечёткое число - это выпуклое, нормализованное нечёткое множество $\tilde{A} \subseteq R$, чья функция принадлежности, по крайней мере, кусочно-непрерывна и имеет функциональное значение $\mu_A(x) = 1$ на точно одном элементе. Запись нечеткого числа в виде интервала $[a,b]$ и в виде выражения a/b равнозначны. Операции над нечеткими числами, заданными диапазоном, выполняются поэлементно (в среднем). В этом смысле расширение понятия нечеткого числа в работе [17], в рамках аффинной арифметики введено с целью увеличения точности. Операцию умножения для расширения нечетких чисел можно задать и через всплески (вейвлеты). Напомним, что сжатие изображения, использующие вейвлет-преобразование, считается наиболее эффективным.

Можно ввести другое расширение или вариацию нечеткого числа – параметрическое или локально компактное нечеткое число. Продемонстрируем это понятие на примере параллельных вычислений. Сложность алгоритма задается иногда, как дробь, такая как, например, $2/3$, где 3 - сложность последовательного алгоритма, а 2 - сложность передачи данных. Вообще говоря, это могут быть не числа, а функции, зависящие от количества процессоров $[2,2]/[3,3](p)$. То есть параметрическое нечеткое число – это отношение двух базисных функций, зависящие от параметра, всегда меньше единицы (в противном случае задача не имеет решения или только замыкание параметрического нечеткого числа имеет функциональное значение равно единице). Также для существования решения необходимо, чтобы базисные функции были монотонными. Фактическое значение параметрического нечеткого числа может быть определено в результате динамической верификации, прогона программ. Для параметрического “темпорального” нечеткого числа можно рассмотреть меру скорости сближения, зависящую от изменения параметра или просто скорость, если параметр рассматривать, как время. Очевидно, что, когда скорость близка к нулю эффективность выше. Таким образом, мера эффективности двойственная мера скорости, то есть эффективность равняется единица минус скорость. Экономическая аналогия через отношение продукта к источнику продукта при условии что, продукт всегда меньше источника продукта, достаточно адекватна. С точки зрения математического моделирования - это модель с насыщением, возможна и интерпретация как “нечеткого” сходящегося ряда $o(2)/o(3)(p)$.

Выделение параметрического нечеткого числа оправданно частой встречаемости. Аналогичный пример – отношение латентности к скорости обмена. Стандартно оценка эффективности параллельных вычислений делается на основе модели первого

приближения, но в тоже время отмечается, что зависимости в этой модели явно нелинейные. Известно, что нелинейность, сильно влияет не только на точность, но и на адекватность модели. Так модель первого приближения не способна адекватно оценить масштабируемость распараллеливания. Таким образом, можно сделать вывод о необходимости применения моделей более высокого порядка точности. Так для эффективности, стремящейся к единице, напрашивается к рассмотрению модель с насыщением. Подобный вывод можно сделать и в области визуализации.

Рассмотрим достаточно простую интерпретацию параметрического нечеткого числа, при рассмотрении графиков базисных функций. Рис. 4 в [30] – это пример визуализации отражения хакерских атак на сайт, связанный с ростом числа IP-адресов. В принципе это задача робастного (с противодействием) управления, легко объяснимая с точки зрения теории возможности. Пусть a – это уверенность, что сайт функционирует нормально, график функции a должен соответствовать модели хищник-жертва с ограниченными ресурсами. Пусть b - это возможность того, что на сайт происходит атака, модель Мальтуса, экспоненциальный рост графика. Реальное значение количества IP-адресов всегда будет в “интервале” $[a,b]$ (зависящего от времени). Фактически нас интересуют не сами графики, а скорость сближения графиков $a/b(t)$ или эффективность контрмер. Также необходимо отметить, что у контрмер достаточно высокая латентность или плохая обусловленность системы, связанная с отсутствием дифференцируемости или с не возможностью отследить скорость изменение графика на достаточно малом промежутке времени. Атакующий также может проанализировать график и изменить стратегию атаки, что приведет к дальнейшему росту латентности. В этом смысле идея создания клона сайта кажется интересной, фактически атакующий будет наблюдать экспоненциальный рост, а реальная работа будет проходить в нормальном режиме.

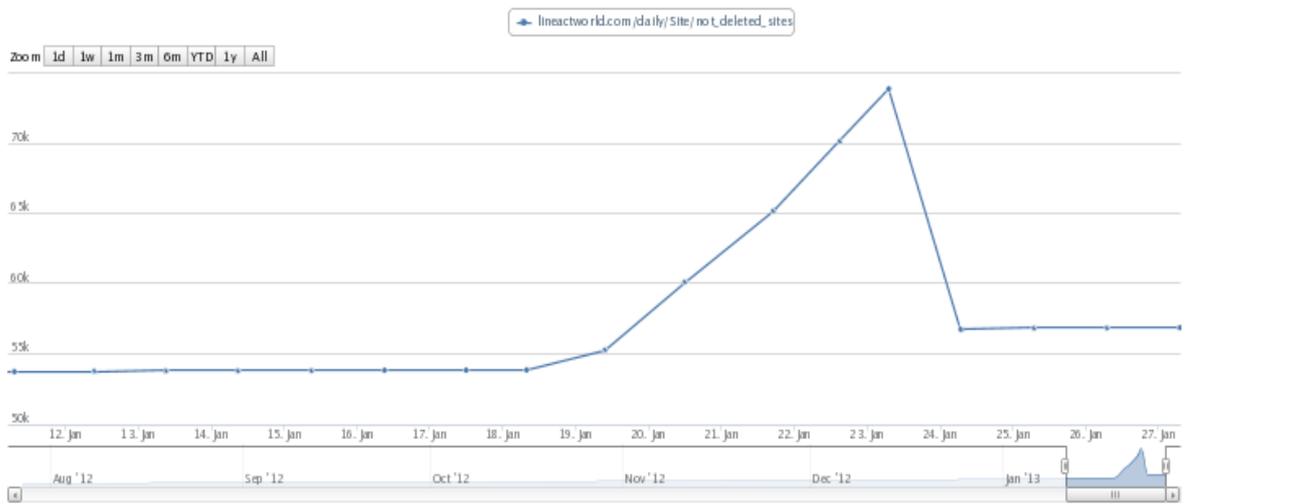


Рис. 4. Отражение хакерских атак на сайт, связанных с ростом IP-адресов [30].

О применение аксиоматического подхода в визуализации уже упоминалось. (См., например, работу [31].) Как естественное обобщение данного подхода часто рассматривают лингвистический подход, где определяется лингвистическая переменная:

$$\Omega = \langle \omega, T(\omega), U, G, M \rangle,$$

где ω - название переменной, T – терм - множество значений, т.е. совокупность ее лингвистических значений, U – носитель, G – синтаксическое правило, порождающее термы множества T , M – семантическое правило, которое каждому лингвистическому значению ω ставит в соответствие его смысл $M(\omega)$, причем $M(\omega)$ обозначает нечеткое подмножество носителя U .

Рассмотрение вопроса, что слабее - лингвистический подход, теория категорий, теория формальных типов или теория доказательств находится вне компетенции визуализации. Для прикладной области есть другие более важные вопросы:

- Как повлияет введение неопределенности на доказательство непротиворечивости?
- Можно ли построить формальное сравнение естественного интеллекта с искусственным (машинным), например, за счет сравнения определения лингвистической переменной и теоретико-множественного определения машины Тьюринга.
- Можно ли расширить лингвистический подход до семиотического подхода? Например, заменой семантического правила на правило интерпретации. Знаковый процесс (или семиозис) рассматривается на пятичленном отношении между знаком, его значением, его интерпретантом, контекстом, где знак встречается и, наконец, интерпретатором знака [15]. Или пятерка семиозиса та же, что и пятерка в лингвистическом подходе? По крайней мере, попытки ввести базисный знак по аналогии с базисной функцией для модели с неопределенностью кажутся оправданными. Только совокупность знаков надо сначала проинтерпретировать, а уж затем придумывать, как разложить в ряд. В математической семиотике вводится понятие системы знаков через требование однозначности смысла любого знака в системе. В качестве примеров здесь приводятся система дорожных знаков и язык морских сигналов флагами.

Рассмотрим, на какие вопросы отвечает работа [16], где рассматривается аксиоматический подход к визуализации, использующий теорию категорий и семиотику. **“Самой мощной концепцией в теории категорий является понятие коммутативности”** [16] В лингвистике определяется термин интенция, как коммуникативное намерение. Именно понятие коммутативности рассматривается, как повод для объединения теории категорий и семиотики. **“Полученное свойство «интенциональности» может использоваться для установления различия между истинными визуализациями и представлениями, управляемыми данными”**[16]. Аналогично, разделение на когнитивную и иллюстративную визуализацию. То есть свойство «интенциональности» проверяет, есть ли намерение (субъективное целеполагание), чтобы различить психические явления от физических.

Очевидно, что для замыкания цикла разработки недостаточно двух мер: информативности и выразительности или разделения визуализации на когнитивную и иллюстративную неизбежно введение к рассмотрению третьей меры или верифицируемой визуализации или обратной связи. Графически это можно отобразить в виде треугольника, подобного тому, который рисуется в семиотике. С учетом интерактивности и постановки задачи минимума нагляднее рисовать спираль. В связи с этим обобщим понятие *рефакторинга*. Рефакторинг модели (программы, базы данных, визуализации) – это уточнение параметров модели в результате обратной связи и постановки задачи минимума или верификации и валидации модели. Практическая часть работы [16] не выходит за рамки эвристического подхода или тестирования “класса” студентов в рамках теории категорий. К сожалению, авторы данной работы отказываются от понятия неопределенность, “чтобы не запутаться в рассуждениях”, просто в теории категорий нет понятия неопределенность. Хотя сделать это формально достаточно просто, если вместо графа рассматривать сеть, в том числе и нейронную.

Рассмотрим применение причинно-следственной сети, называемой нечеткой когнитивной картой, для интеллектуального анализа визуальной информации [11]. В данной работе используется термин *концепт*, который можно рассматривать, как нейрон или терм. Основное отличие нечеткой когнитивной карты от нейронной сети в том, что взаимодействие между нейронами рассматривается только попарное, как в теории категорий (фактически это матрица), и как следствие результирующее значение функция многих переменных (фактически это вектор). Таким образом, нечеткая когнитивная карта отражает не только тенденцию применения модели с неопределенность, но стремление перехода от дискретной математики к непрерывной. Аналогичная матрица возникает и

для нейронной сети, между двумя слоями. Очевидным требованием существования решения является неравенство определителя нулю. Такое преобразование, которое сохраняет топологические свойства, называют гомеоморфным [32]. Также стоит отметить, что матрица обмена сообщениями в параллельных вычислениях имеет аналогичную структуру, что и нечеткая когнитивная карта. В работе [11] экран делится на 64 части. В связи с этим хочется сказать о идеи проверки гипотез гештальт-психологии с точки зрения грубых множеств, где одно из ключевых понятий - это гранулярность.. В принципе, гранулу можно рассматривать, как обобщение интервала или расширение нечеткого числа в той же компактно-открытой топологии, но большей размерности, чем числовая прямая, Так, например, можно оценить минимальный размер кнопки через сумму вероятностей (попиксельно), при условии, что внутренность кнопки должна быть больше границы, по сути - это вариант неравенства Маркова. Подход с границами гранулированных множеств легко расширяем. Кнопка в виде круга лучше прямоугольной, так как имеет наименьшую границу. Для трехмерной графики гранула – графический примитив. Определены операции пересечения и объединения, следовательно, можно ввести любое количество базисных знаков и фикцию принадлежности нескольким знакам. (Та же кластеризация, стандартно принадлежность к кластеру определяется через минимум расстояния.) Также можно вывести алгоритм преобразования воксельной графики в полигональную, применяя закон сохранения импульса или его следствие – метод опорных векторов.

На этом мы заканчиваем обзор материалов, касающихся верификации визуализации. Конечно, представленные выше материалы могут быть дополнены. Но представляется, что обзор обеспечил приемлемый уровень понимания проблем для разработчиков визуализаций.

В следующей главе рассмотрим применение множественного вида отображения в визуализации с позиций нечеткой логики. Рассмотрение данного вопроса связано с проблемой интеграции, когда для верификации недостаточно одного вида отображения или одной модели или одной группы исследователей.

3. Множественный вид отображения

В сфере визуализации хорошо зарекомендовало себя применение комплексного или множественного вида отображения. Это понятие подразумевает использование в процессе интерпретации и взаимодействия нескольких разделенных видов отображения, между которыми устанавливаются взаимосвязи. В качестве примера такого вида отображения можно привести аналогию из области черчения, когда трехмерное тело представляется виде трех проекций.

Для рассмотрения и формализации данного примера можно ввести несколько групп базисных функций, таких как:

* Информативность и избыточность.

* Полнота и точность, обеспечивающие целостное восприятие и детализацию.

Результатом (целью) применения множественного вида отображения может являться не только сокращение объема информации, но и понижение размерности, в том числе и когнитивной. В последнем случае множественный вид отображения удобно рассматривать, как произведение бикомпактов, обеспечивающие наследование свойств (произведение бикомпактов есть бикомпакт).

Как пример явного разложения по двум базисным функциям можно привести множественный (бинарный) вид отображения, включающий в себя миникарту и основной вид отображения, для которого можно поставить задачу минимакса: с одной стороны, обеспечение целостного восприятия, но не точно, с другой – детально (более точно), но не всей информации (применение уровня детализации). Очевидно, что для бинарного вида отображения можно ввести две дополнительные операции над данными: алгебраическая сумма и геометрическая разность. На подобном подходе, например, построены некоторые алгоритмы распознавания. Для задачи применения множественного вида отображения,

наверное, возможна строгая формализация в рамках топологии или алгебры (например, решетка Стоуна). Далее речь идет только о формализации с использованием подходов “модели и полумодели” или ментальной модели в рамках нечеткой логики и теории возможности.

Более подробно остановимся на конкретном применении бинарного вида отображения для трех задач в области научной визуализации, информационной визуализации и визуализации программного обеспечения:

* Задача фильтрации трехмерной сетки. (Подраздел *Топология трехмерной сетки*).

* Задача информационной фильтрации, в частности отображения результатов поиска. (Подраздел *Применение облака тегов для информационной фильтрации данных*.)

* Применение бинарного вида отображения в области визуализации программного обеспечения. (Подразделы *Топологический анализ поведения циклов* и *Ментальная модель параллельных координат*.)

3.1 Топология трехмерной сетки

Под топологией сетки, обычно понимают структуру сетки, которая задается в виде геометрических примитивов (тетраэдров кубов и др. многогранников). В общем случае хотелось бы рассмотреть отображение сеточных данных (результатов вычислений) на достаточно широкий класс стандартных видов отображений именно с позиций топологического анализа, но пока ограничимся использованием нечеткой логикой с целью оценки эффективности на основе формальной модели. Эту задачу можно определить, как верификация сетки. Ее можно рассматривать, как частный случай визуализации данных большого объема. Данные большого объема определяются, как данные, которые невозможно отобразить как, в полном объеме, так и за приемлемое время. Один из основных вопросов задачи верификации сетки – это вопрос о метриках оценки эффективности.

Задача визуализации данных большого объема не имеет решения без постановки задачи фильтрации данных или семантически близких подходов, например, таких, как агрегация, редукция, кластеризация. Фильтрация данных (фильтр) – любая операция над данными, изменяющая их количество (“не хеширование”). В этом определении “не хеширование” рассматривается, как необходимое условие установление предпорядка. Формально можно говорить о построении идеала. Исходя из этого определения, сжатие изображения с потерей точности и рендеринг можно рассматривать, как частные случаи фильтрации данных. Как следствие возникает необходимость рассмотрения неполной информации или модели с неопределенностью.

В рамках концепции параллельной фильтрации данных можно рассматривать фильтры как преобразования, обеспечивающие целостное восприятие или детализацию. Правда, при этом возможна значительная потеря качества. Поэтому целесообразно использовать комплексные (множественные) виды отображения. Очевидно, что одновременный вывод на экран всего слишком большого объема данных влечет за собой снижение уровня детализации.

В качестве упрощенного примера рассмотрим применение фильтра - сечение плоскостью. Дана трехмерная сетка со скалярным значением в узлах сетки, которое стандартно отображается цветом. В этом смысле, метрика цветовой дифференциации совместима с топологией числовой прямой. Количество узлов сетки большого объема пусть будет N в кубе. Видеокарта способна обработать только количества данных порядка N в квадрате (то есть плоскость). Базисная функция полнота визуализации – это нечеткое число $[2,3](N)$. Проще говоря, если вывести N сечений плоскостью, то получится полное (равное единице) представление о трехмерной сетке. Точность определим через количество узлов сетки в примитиве визуализации. Наилучшая точность (уверенность) равна единице, возможность ограничена количеством узлов сетки. Рассмотрим параметрическое нечеткое число, как отношение функции полноты к функции точности с

коэффициентом нормирования $1/2$. Как уже отмечалось, это отношение можно рассматривать, как эффективность. Для данного примера максимальное значение эффективности равно единице достигается, когда отображается одна плоскость.

Эффективность в данном примере – это эффективность без учета задачи взаимодействия (управления). Выбор оптимального решения остается на пользователе, но в пределах двух интервалов. Для облегчения задачи управления часто говорят об автоматическом выделении особенностей, например области сгущения сетки. С точки зрения модели с насыщением можно говорить о выборе оптимальных начальных данных. Существование решения задачи управления могла бы обеспечить монотонная последовательность решений, в данном примере можно было бы вывести сначала одну плоскость, затем две и так далее N . Но в постановке задачи предполагалось, что у нас данные большого объема, то есть за раз точно можно отобразить только одну плоскость. Напрашивается решение выводить по одной плоскости, но достаточно быстро, например, со скоростью 25 кадров в секунду, тем самым, обеспечив иллюзию непрерывности. Такой режим управления можно назвать режим “радар”. В результате возникает необходимость в рассмотрении еще одной базисной функции – скорости дискретизации, зависящую от функций полноты и точности визуализации.

Все эти рассуждения говорят о том, что можно определить и верифицировать эффективность. В рамках той же модели можно применить и другие фильтры, а затем сравнить их эффективность. В специализированной системы визуализации данных [33] нами был предложен конвейер фильтров, включающий пространственную фильтрацию и фильтрацию «по значению». Функция пространственной фильтрации (полноты) реализована с помощью метафоры «альфа-сферы» (Рис. 5). Для фильтрации по значению (точность) введен дополнительный компонент пользовательского интерфейса - поле диапазонов. Диапазон представляет собой отрезок $[a, b]$, заданный на оси соответствующей характеристики, только на этом интервале происходит отображение и интерполяция данных. Метафора «альфа-сферы» - это тот же диапазон, но в сферических координатах, где функциональное значение полноты - прозрачность объектов, находящихся внутри области, прямо пропорционально удаленности объектов от центра области.

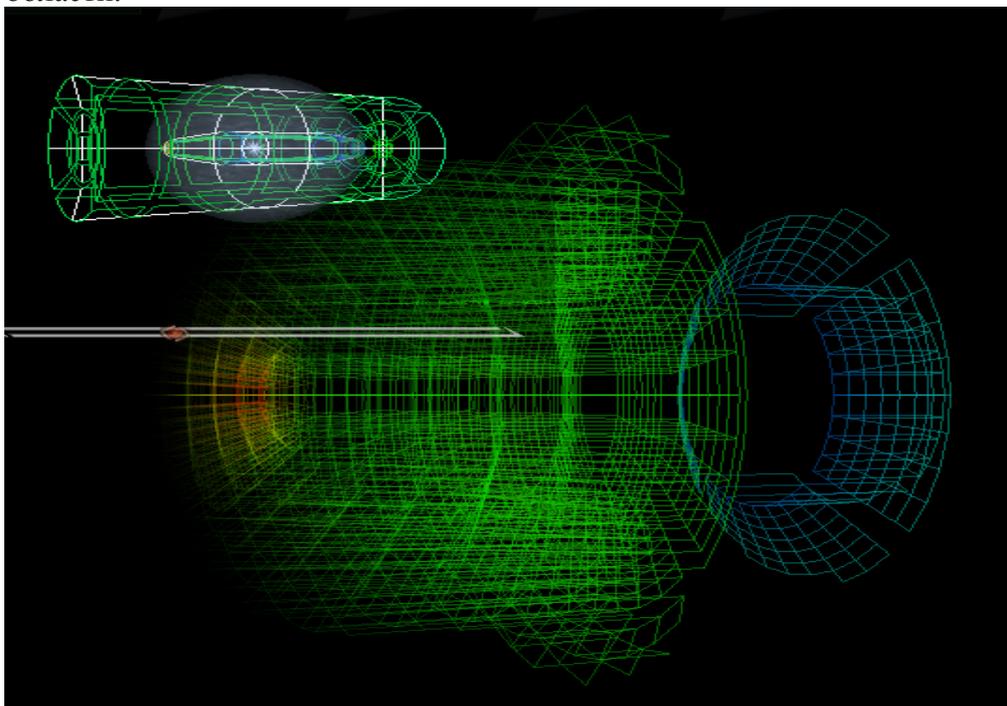


Рис. 5. Пространственная фильтрация с помощью метафоры «альфа-сферы» [33].

Видов отображения, используемых в научной визуализации для визуализации сеток, достаточно много, таким образом, множественных видов отображения еще больше. Сравнение и оценку которых целесообразно построить на основе формальных моделей, чтобы предложить оптимальный вариант. В работе [34] описан множественный вид отображения, предлагаемый в системе просмотра медицинских данных. Этот вид отображения включает два ортогональных сечения плоскостью с прокруткой и воксельный вид отображения. Плюсом данной системы является поддержка Web-технологий. Не заостряя внимание на удаленной и он-лайн визуализации, можно указать на то, что наличие данных свойств также можно рассматривать, как меры оценки эффективности и адекватности систем визуализации.

3.2 Применение облака тегов для информационной фильтрации данных

Параллельная фильтрация данных наряду с параллельным рендерингом активно применяется для сокращения объема визуализируемых (отображаемых) данных. В этом разделе рассмотрим задачу информационной фильтрации, а именно отображение результатов поиска. Вероятно, эта задача формально более простая по сравнению с отображением сеточных данных. Результаты поиска в Интернете плохо масштабируются, известным решением данной проблемы является кластеризация. Мы предлагаем другой подход, основанный на фильтрации данных [10] и обеспечивающий достаточно высокий уровень формализации или верификации. В этом случае для нас важен выбор задачи (информационной фильтрации данных). На этой задаче можно продемонстрировать общий подход (модель с неопределенностью). Она удобна для проверки адекватности модели, если основываться, например, на эвристическом подходе (тестирование). То есть мотивация здесь скорее научная, чем прагматическая. Решение данной задачи требует описание трех основных частей: архитектурного решения, вида отображения и формальной модели.

Архитектурное решение соответствует облачным вычислениям. Модуль-посредник перехватывает результаты поиска через Google API и реструктуризирует данные в виде ассоциативных массивов. В результате взаимодействие в клиентской части строится на основе хеширования, то есть практически без пересчета.

Вид отображения выбран бинарный, включающий миникарту и основной вид отображения – вывод непосредственно текста документа, реализация которого очевидно возможна через гиперссылку или в виде всплывающей подсказки. Поэтому основное внимание уделено реализации миникарты, по сути - это расширение облака тегов.

“Облако тегов или Облако меток или Облако ключевых слов (tag cloud, wordle) — визуальное представление списка ярлыков (или категорий). Частота упоминаний, поисков, ссылок в Интернете с определённого сайта неких слов, терминов, имён, отображается в специальной области в виде изображения этих слов в формате гиперссылок. Размер изображения тем больше, чем чаще использовался данный тег (слово, термин или имя)”. (Цитата из Википедии.)

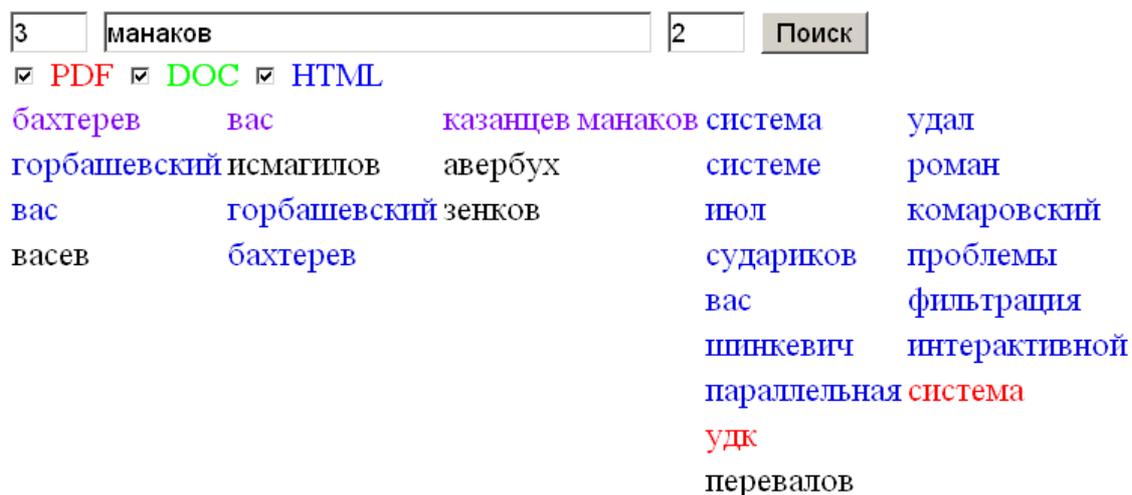
По сути, облако тегов - популярная в Интернете визуальная метафора, генерирующая вид отображения, где упорядоченному списку (например, слов по частоте встречаемости) ставится в соответствие двухмерное изображение в декартовых координатах, в котором можно варьировать позицию объекта (слова), его размер, цвет. Не трудно видеть, что это определение можно применить и для других случаев (квазитекстовых), например, посещаемость сайта (статистической информации ставится в соответствие карта мира, в данном случае возможно применение трехмерного изображения или сферических координат). В общем случае, можно рассмотреть нечеткое множество X , где для каждого элемента множества (объекта) задана функция принадлежности $\mu_X(x) \in \langle 0, 1 \rangle$.

Результаты поиска (нечеткое множество X) представлено в виде массива ассоциативных массивов, где центральный элемент массива – строка поиска (Рис. 6.). Формально задаются две функции принадлежности:

* ширина контекста или функция отклонения (количество слов слева и справа от строки поиска), реализующая фильтрацию данных в рамках компактно-открытой топологии;

* частота встречаемости слова, зависящая от типа документа с расширением pdf, doc, html соответственно отображается градацией цвета R, G, B , реализующая хеширование данных или брашинг (выделение цветом).

В данном случае фильтр можно рассматривать, как реализацию семантического запроса с неопределенностью. Результаты поиска также включают название документа, размещенные внизу, и гиперссылку, которые идентифицируются через взаимодействие с конкретным словом (ключом) и выделением черным цветом данной ветви.



[Манаков Д.В. - Сектор Визуализации ИММ УрО РАН](#)

[Соруководитель Д.В. Манаков. - Сектор Визуализации ИММ УрО ...](#)

Рис. 6. Облако тегов для информационной фильтрации данных [10].

Облако тегов может рассматриваться как граф, но двунаправленный (в компактно-открытой топологии) и отображаемый в виде таблицы. Именно представление данных в виде графа обеспечивает семантический зуминг. Как уже отмечалось, формализация нужна для оценки эффективности, которую определим как произведение двух мер масштабируемости (информативности) и избыточности. Масштабируемость – это отношение количества строк в облаке тегов к количеству в стандартной выдаче на поисковый запрос. Избыточность – это остаток сходящегося ряда (единица минус $1/2$ в степени n , где n – ширина контекста). В данном случае условной вероятностью можно пренебречь, так как зависимость этих мер достаточно слабая, что подтверждается опытным путем, например, при изменении ширины контекста на единицу. Очевидно, что контекстное облако тегов хорошо масштабируется, но облегчит ли оно интерпретацию результатов поиска. В рамках предлагаемой модели интерпретация должна упроститься, если пользователь, конечно, имеет представление о работе алгоритма MapReduce. Адекватность модели предполагалось проверить на основе тестирования, по единственному критерию - помогает ли контекстное облако тегов поиску. Желющие могут самостоятельно оценить перспективу проекта и качество взаимодействия по ссылке: <http://www.cv.imm.uran.ru/oblako>. Также в рамках исследования планировалось рассмотреть применение конвейера фильтров, например, для поиска соавторов. (На Рис. 6. видно, что информация избыточная, кроме фамилий есть другие слова.)

3.3 Топологический анализ поведения циклов

В качестве примера применения бинарного вида отображения в визуализации программного обеспечения рассмотрим работу [35]. Как видно на Рис 7. множественный вид отображения явно разбит на две группы и включает в себя основной вид отображения

- текст программы точно, но не полностью и миникарту, являющейся траекторией программы в виде облака точек, представленной полностью, но не точно. Также видно, что анализируются именно изменения. Эта система визуализации предназначена для анализа попадания данных в кэш при работе вложенных циклов. Кратко остановимся на том, как вычисляется траектория. Для этого используется метрика Левенштейна – количество изменений при переходе от одного картежа к другому (видимо посимвольно) и отображения полученного многомерного пространства на симплицальный комплекс (двумерное декартово пространство плюс цвет).

Почему же не возникает инсайта при взгляде на Рис. 7? В данном случае, как уже отмечалось, от пользователя требуется знание формальной модели, которая достаточно сложная. К тому же вряд ли метрика Левенштейна линейно связана с попаданием в кэш. Еще одна проблема связана с недоказуемостью формальной правильности циклов. Представляется, что, хотя бы на начальном этапе, циклы можно было бы заменить на стеки или итераторы.

```
File: MPM.cpp
1: for(unsigned ii=i; ii<=i+1; ii++){
2:   for(unsigned jj=j; jj<=j+1; jj++){
3:     g->mass(ii,jj) += g->S(ii, jj, mp->position(p))*mp->mass(p);
4:     g->momentum(ii,jj) += g->S(ii, jj, mp->position(p))*mp->mass(p)*mp->velocity(p);
5:   }
6: }

File: Grid.h
1: double S(int i, int j, const Point& p){ ... }
2: unsigned indexify(unsigned i, unsigned j) const { ... }
3: double S_x(int i, double x){ ... }
4: double S_y(int j, double y){ ... }
5: static double compute_shape_function(int cell, double position, double cell_size){
6:   // This is the distance of "position" from the position of "cell".
7:   const double cell_distance = std::abs(position - cell_size*cell) / cell_size;
8:   // Perform case analysis.
9:   if(cell_distance >= 1.0){
10:    return 0.0;
11:   }
12:   else{
13:    return 1.0 - cell_distance;
14:   }
15: }
```

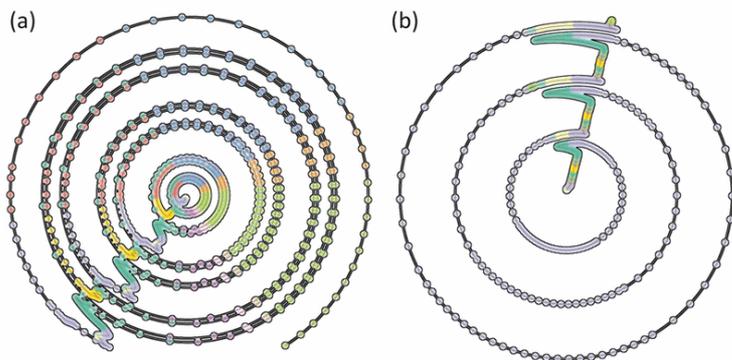


Рис. 7. Применение множественного вида отображения для анализа поведения циклов [35].

В настоящее время, все же, идея с траекторией программы, в том числе и параллельной, кажется перспективной. Оправданность такого подхода можно показывать на простых примерах, где возможна простая интерпретация графика. Например, тупик – это разрывная траектория (не определена производная или равна бесконечности), а узкое место – негладкая траектория. Таким образом, нужно определить “простые” правила интерпретации, которые можно рассматривать, например, как морфизмы в рамках теории категорий. С другой стороны в рамках топологий поточечной сходимости или компактно-открытой естественно применение в роли миникарты не траектории, а графа. Как пример простой интерпретации можно привести Рис. 8 [36]. Если у графа можно выделить независимые подграфы, то уровень параллелизма программы высокий. Формально это утверждение можно попытаться обосновать, например, применяя топологические теоремы об отделимости.

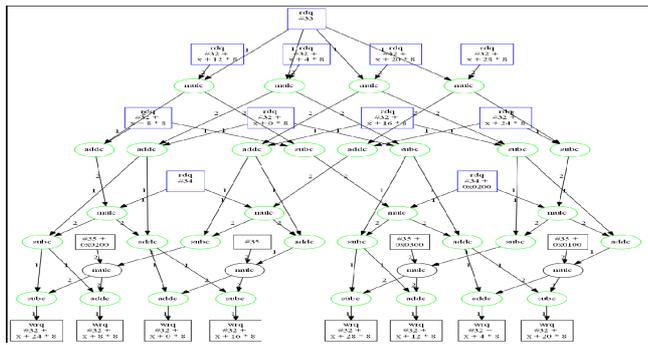


Рис.8. Граф программы с высоким уровнем параллелизма [36].

Также в работе [36] обсуждаются теоретические и практические результаты развития визуализации программного обеспечения. Необходимо отметить, что для отладки правильности и эффективности параллельных программ предложено достаточно большое количество метафор визуализации, которые, к сожалению, сложно интерпретировать. В данной области стандартным является применение множественного вида отображения, которое интуитивно связано с попытками разложения по нескольким базисным функциям. На наш взгляд, перспективным направлением является построение видов отображения на основе формальных моделей: нечеткой логики, топологического анализа, аксиоматического подхода. При этом не стоит забывать, что решения у данной задачи может и не существовать. Связано это в первую очередь с неполным знанием.

Далее рассмотрим ментальную модель параллельных координат. Эту модель можно считать полностью сформировавшейся, хотя некоторые понятия, имеющие одинаковый смысл, именуется по-разному, в зависимости от источника литературы.

3.4 Ментальная модель параллельных координат

Термин параллельные координаты активно используется в литературе [18,37]. На диаграмме с параллельными координатами множество осей размещаются параллельно друг другу, чтобы можно было выявить зависимости между переменными. Как отмечается в книге [37] цель информационной визуализации – рассказать историю. Из определения цели происходит термин сторинг (storing). Причем в качестве истории может выступать обычный график с пояснениями, а лучше с формальной моделью. Важно рассказать историю правильно, то есть поднимается вопрос о верификации. На наш взгляд, правильно рассказывать историю надо в следующей последовательности: формализация, визуализация, валидация.

Рассказывая о параллельных координатах, начнем с формального определения эффективности. Используемое в экономике определение эффективности через отношение продукта к источнику продукта (ресурсам) достаточно адекватно. Формально, это скорость или полный дифференциал. Также возможно рассмотрение данного отношения и с позиций нечетких множеств, очевидно, что экономическая эффективность всегда меньше или равна единице, например из того же закона сохранения массы. В данном случае для экономической или затратной модели эффективность, являющаяся функцией многих переменных, отображается последовательно по каждой переменной и непрерывно. Полученный график эффективности, даже в случае рассмотрения полного дифференциала, является не совсем корректным, но дает представление о линейной зависимости между переменными. Плюсом данного подхода можно считать то, что ресурсы между собой неаддитивны, в то время как, эффективность аддитивна. Следовательно, возможно развитие модели, например в сторону построения решетки.

В качестве иллюстрации в книге [37] приводится пример результативности баскетболиста, которая определяется, как *гол плюс пасс минус фол*. Эти показатели рассматриваются, как ресурсы и отображаются в виде параллельных координат. У

каждого игрока свой график результативности. Если номер игрока или его имя рассматривать, как константу фактически получается фазовое пространство или дифференциальное многообразие Рис. 9.

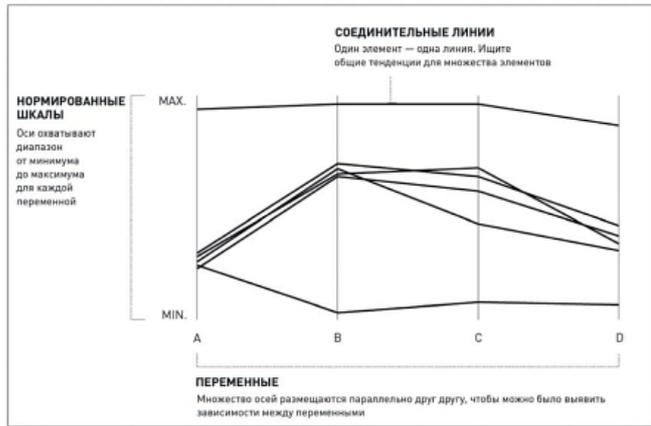


Рис. 9. Структура диаграммы с параллельными координатами [37].

Этот пример легко переносится в область визуализации программного обеспечения, где в качестве ресурсов рассматриваются, например, время чистого счета процессора или количество попаданий в кэш, вообще любая информация, которую можно собрать о программе. В случае параллельной программы цветом можно отображать номер процессора. Если отслеживать только минимальное и максимальное значения ресурсов, очевидна формализация и интерпретация в рамках теории возможности. При достаточно мелком разбиение, например, по времени счета или длине программы интервал значений ресурсов должен сойтись в точку, конечно, если выполняется свойство монотонной сходимости, и лучше к наиболее эффективному значению. Это и является целью отладки эффективности параллельных программ или правилом интерпретации. В результате получается вид отображения, известный в литературе, как информационная фреска или фреска Джердинга [38]. Также используется термин (Information Mural) [39]. (См. Рис. 10.)

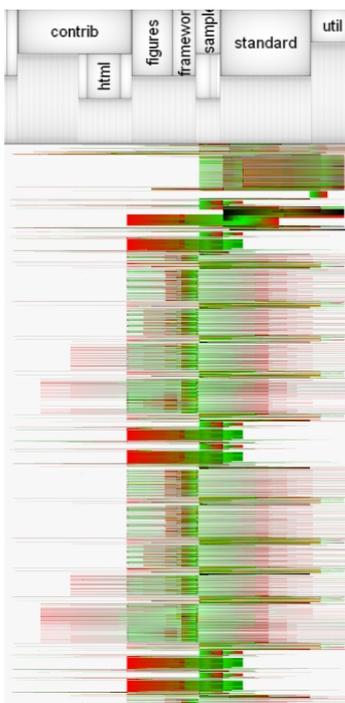


Рис. 10. EXTRAVIS информационная фреска [38].

EXTRAVIS – инструмент для визуализации больших трасс. Предлагаемые методы визуализации должны быть количественно оценены с целью понимания программы. Общие подходы в литературе разделяются на статические и динамические. Важное преимущество динамического анализа (отладки) – его точность, поскольку он рассматривает фактическое поведение системы. Среди недостатков его неполнота, так как собранные данные принадлежат сценарию, который был выполнен, и также существуют известные проблемы масштабируемости [38]. Поэтому было бы логично рассматривать виды отображения, связанные с динамической отладкой, например, информационную фреску, как основной вид отображения, а виды отображения, связанные со статической отладкой, как миникарту.

Работа [40] интересна, прежде всего, тем, что в ней предложена модель управления алгоритмами реального времени (*VA PB*). Семантика *VA PB* может быть определена, как набор кортежей (четверок):

$$VA \cdot PB = \{ \langle f_i, t_i, \eta_i, \bar{l}_i \rangle \}, \quad i=1, \dots, N,$$

где f_i – функциональная задача (действие), t_i – момент начала выполнения действия, η_i – длительность действия, \bar{l}_i – логический вектор, включающий набор значений условий в трехзначной логике, определяющий исполнение действия.

На первый взгляд определение лингвистической переменной шире, чем *VA PB*, так как там есть неопределенность в рамках нечеткой логики. Это не совсем так, в определении *VA PB* также присутствует неопределенность в логическом векторе. Как следствие наиболее существенным дополнением к параллельным координатам является то, что модель *VA PB* содержит возможность их самопересечения, что позволяет постановку задачи синхронизации процессов и реализацию одной и той же семантики программы с различными управляющими графами. Очевидно, что эти самопересечения параллельных координат можно интерпретировать в рамках той же теории возможности Рис. 11.

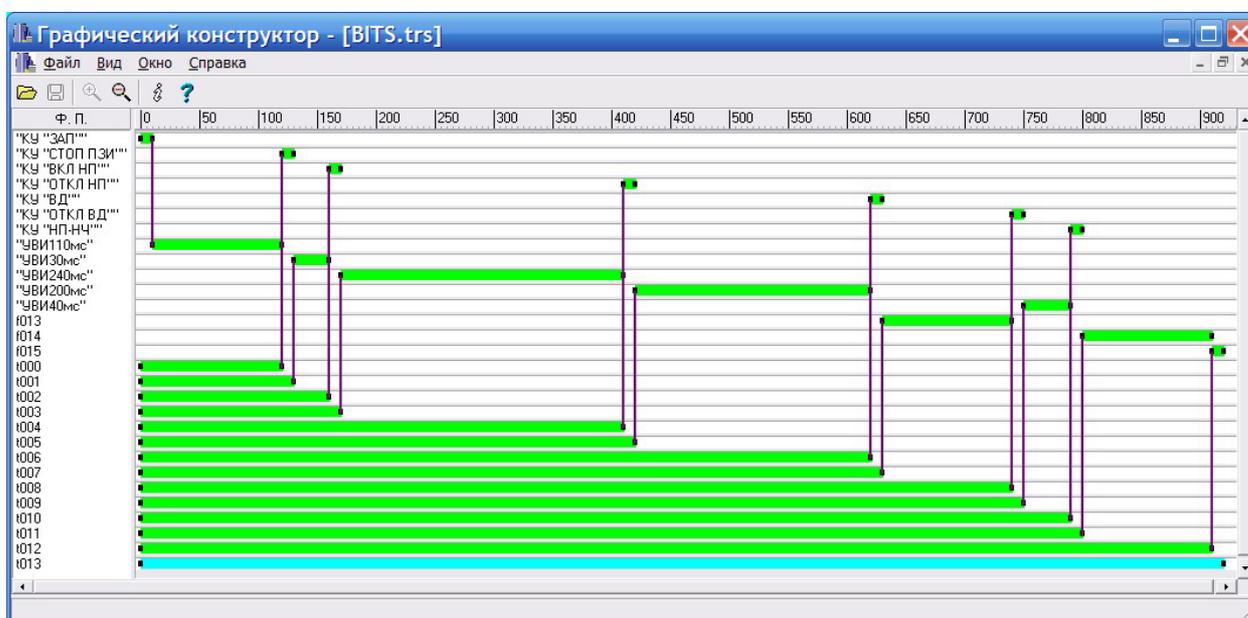


Рис. 11. Визуальная верификация циклограммы управляющего алгоритма [40].

Также стоит отметить, что в работе [40] используется термин визуальная верификация, а расширение информационной фрески названо циклограммой.

Несмотря на то, что и термины, и формальные подходы, упомянутые в этом разделе, имеют высокую степень синонимичности, ментальную модель параллельных координат можно считать сформировавшейся. На данном примере, мы попытались построить иерархию моделей и соответствующим им видам отображения, что,

несомненно, является одной из целей теории компьютерной визуализации. Метод параллельных координат, используемый в информационной визуализации для описания эффективности экономических моделей [18], может быть использован и при представлении данных об эффективности вычислений, так как непринципиально, анализируются ли мощность двигателя или количество попаданий в кэш. При достаточно мелком разбиении по длине программы или времени оправдано использование видов отображения, построенных на базе “информационной фрески”. В [40] представлен интересный пример использования циклограмм для “визуальной верификации” управляющих программ реального времени.

В [41] в рамках эмпирического подхода предлагается формализация понятий, связанных с масштабируемостью параллельных программ и рассматриваемых, как функции многих переменных. “Масштабируемость – свойство параллельной программы, характеризующие зависимость изменения динамических характеристик ее работы (в частности, эффективности) от изменения параметров запуска”. Например, при рассмотрении слабой масштабируемости эффективность – это функция многих переменных, которая зависит от количества процессоров и от количества данных. Данный подход позволяет использовать методики представления трехмерных объектов, характерных для научной визуализации. (См. Рис. 12.) Таким образом, возможно рассмотрение нескольких метрик эффективности. При этом эффективность по количеству процессоров – это частная производная по количеству процессоров, а эффективность по количеству данных – это частная производная по количеству данных.

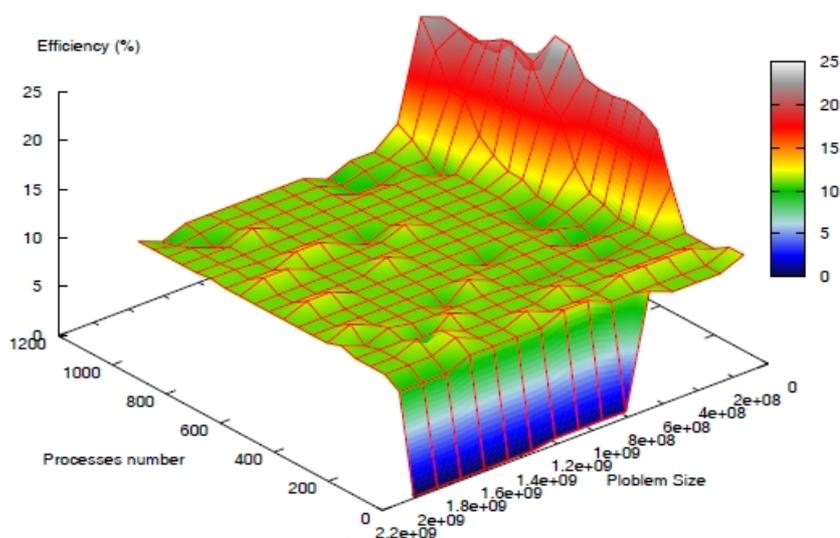


Рис. 12. Эффективность – это функция многих переменных, которая зависит от количества процессоров и от количества данных [41].

Представленное изображение может выступать в качестве визуального эталона при оптимизации программы, при отображении программы на целевой вычислитель или при решении одной задачи разными методами. В частности, последнее направление представлено в работе [42], которая посвящена решению СЛАУ с разными предобуславливающими матрицами. В этой работе рассматривается другая динамическая характеристика – скорость интерполяции (то есть та же, эффективность), зависящие от количества подобластей и параметров их наложения. Двумерная поверхность эффективности или траектория программы (Рис. 12) получена в результате серии запусков программы с разными параметрами.

Распараллеливание ориентировано, прежде всего, на задачи большого счета. Поэтому, ближайшей целью в оценке эффективности является переход от эмпирического подхода к экстраполяции. Также следует ввести и обосновать правила интерпретации.

Отметим, что рассмотрение двухпараметрического случая является достаточно распространенным. (Сравните Рис. 12 и Рис. 13).

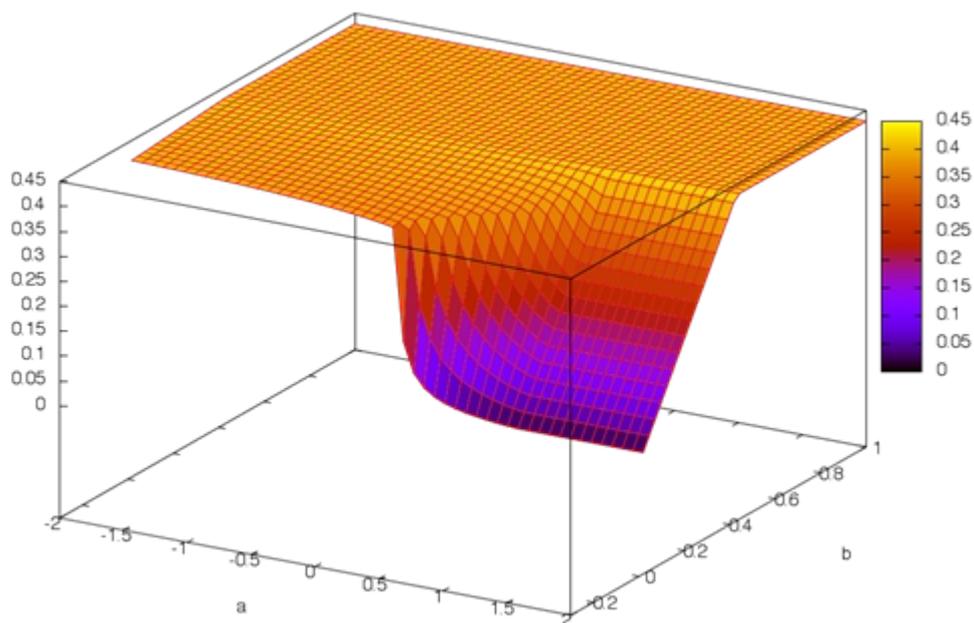


Рис. 13. График зависимости целевой функции от свободного параметра [20].

Аналогичным образом можно определить и масштабируемость визуализации, то есть необходимым условием эффективной визуализации является возможность изменения параметров программы визуализации. Технология фильтрации данных ориентирована, в том числе, и на решение данной задачи.

В следующей главе рассмотрим, что дает применение в лоб аксиоматического подхода к задаче параллельной фильтрации данных.

4. О возможности формализация визуализации и параллельной фильтрации данных на основе лингвистического и семиотического подхода

Прежде, чем перейти к формализации фильтрация данных необходимо представить ряд общетеоретических сведений, чтобы закрепилась ментальная модель. Примерно 10 лет назад сформировалась предметная область «визуальные супервычисления», была определена терминология и идеология области, созданы модельные и прототипные системы. Данная область должна была предоставить технологии, нацеленные на поддержку визуализации в параллельных и распределенных вычислениях [43]. Несколько позже сформировалась область облачных вычислений, которые предполагают распределенные вычисления, а также запуск задач (сервисов) и анализ их результатов с применением веб-технологий (“тонкий” клиент). Существует разделение параллельных и распределенных вычислений, основанное на том, что в первом случае характерен параллелизм по данным, а во втором - по задачам. Это разделение достаточно условно, значительно важнее их общность с точки зрения применяемых технологий программирования.

В работах [43-45] представлен обзорный материал по проблемам визуализации супервычислений, содержащий также информацию по дополнительным вопросам, таким, как статическая и динамическая отладка, алгоритмы с внешней памятью, объемный рендеринг, средства и функциональность веб-визуализации.

Рассмотрим ниже параллельную фильтрацию данных, технологию, использующую как интерактивную визуализацию, так и параллельное применение алгоритмов фильтрации данных. Предлагается формализация визуализации и параллельной фильтрации данных с использованием подходов, связанных с семиотикой и/или лингвистикой. Отметим, что ряд предлагаемых идей формализации пересекаются с теоретическими исследованиями в книге отечественного автора В.С. Файна [46], а также с довольно популярными в настоящее время идеями информационного разрыва [19].

Как причину возрастания роли визуализации в параллельных и распределенных вычислениях, обычно отмечают увеличение объема и усложнение структуры данных. Это, скорее, следствие развития техники. Главное, что параллельные вычисления нарушают последовательный цикл разработки программ. Не случайно в определении алгоритма выделяют входные и выходные данные, в параллельной реализации они нуждаются в постоянной декомпозиции, распределении, перераспределении и сборке. Можно приспособиться, адаптироваться к существующей ситуации, выделив параллельную часть (решателей) и последовательную (пред и постобработку) или развить, создавая новые технологии (интерактивную on-line визуализацию, параллельную фильтрацию данных). Обеспечение непрерывности и эффективности визуализации требует использования формальных методов.

Можно говорить о появлении нового термина - параллельная фильтрация данных или 'parallel data filtering'. Интуитивно понятно, что фильтрация связывает параллельную, интерактивную обработку данных с их on-line визуализацией. Параллельная фильтрация данных лежит в основе целого ряда разрабатываемых систем интерактивной визуализации в области параллельных вычислений. Предварительный отбор или фильтрация данных достаточно хорошо известны, так, например, существуют работы по информационной визуализации, связанные с фильтрацией для распределенных баз данных, а понятие фильтр используется в ряде графических библиотек (например, VTK), в первую очередь нас интересует анализ и описание применения этого подхода в рамках визуальных супервычислений.

Стандартный графический конвейер представляет собой последовательность процессов *фильтрации, геометрической обработки ("мэппирования"), рендеринга (растеризации картинки)*. Любой из этих трех процессов может быть реализована как параллельно, так и последовательно. Следовательно, возможны четыре основных варианта:

1) Все последовательно – соответствует off-line визуализации или постобработке. Стоит отметить вариант off-line визуализации с распараллеливанием, позволяющий применять в дальнейшем те же схемы, но, абстрагировавшись от счетного алгоритма, и иметь дело только с его выходными данными. Этот подход требует дополнительного цикла ввода-вывода, в ряде случаев время счета сопоставимо со временем вывода.

2) Все параллельно – полностью параллельный рендеринг. Результирующие изображение (растр) строится удаленно от станции визуализации. Этот подход удобен в следующих случаях: отфильтрованные данные велики для передачи по сети, выбранный вид отображения (алгоритм построения раstra по данным) требует емких вычислений. Среди недостатков: программный рендеринг (без использования аппаратных ускорителей), отсутствие интерактивности (нет возможности непосредственно взаимодействовать с графическими объектами). Чтобы сгладить последний, наиболее существенный недостаток применяют организацию данных со многими разрешениями или мультиразрешение (используется для уровня детализации) и организацию данных, зависящих от точки зрения (например, для вращения объекта).

3) Параллельная фильтрация. Наиболее эффективный подход. Позволяет использовать аппаратные возможности видеокарт или дополнительные аппаратный уровень параллелизма. Цель фильтрации сокращение объема данных. При такой постановке фильтрацию рассматривают, как результат работы параллельного фильтра.

Возможность применения интерактивности дает новое качество и позволяет определить фильтрацию в рамках теории информации, как интерактивный процесс, целью которого является получение необходимой (интересующей) информации за минимальное время.

Эта технология востребована для двух классов задач - визуализация данных большого объема и в задачах, требующих активного взаимодействия пользователя и системы в процессе визуального анализа, включая возможность изменения параметров задачи.

Фильтрация наиболее эффективный подход, использующий как интерактивность, так и генерацию проблемно-ориентированного метафайла, и параллельное применение алгоритмов фильтрации данных. Преимущества применения предобработки (фильтрации) на счетных узлах вычислителя достаточно очевидны: обрабатываемые для визуализации данные уже находятся на узлах вычислителя, что позволяет избежать дополнительных обменов; при визуализации можно учесть характерные особенности задачи; возможна интерактивная on-line визуализация большого объема данных; высокая скорость визуализации.

Можно выделить большое количество фильтров: децимация, многомерные проекции, сечение плоскостью, а также фильтры, ориентированные на конкретную задачу и модель визуализации. Поэтому можно определить две основные задачи - общее описание применения фильтров или технология фильтрации и разработка фильтров для конкретных задач.

4) Фильтрация и геометрическая обработка выполняются параллельно. Позволяет параллельно произвести триангуляцию и удалить невидимые примитивы. Этот подход сопоставим по эффективности с предыдущим, хотя в частных случаях можно добиться значительного ускорения. Для невыпуклого тела невидимая часть поверхности может составлять больше 50%, но при вращении объекта необходима подкачка недостающих примитивов. Алгоритм триангуляции достаточно сложный, то есть его имеет смысл распараллелить. Количество вершин имеет меньший размер, чем список треугольников. Для структурированных данных целесообразней пересчитать координаты на рабочей станции, чем их передать целиком (для прямоугольной сетки вместо трехмерного массива, достаточно передать одну вершину и шаг). В связи с этим в интерактивном режиме возникает задача не только задача минимизации данных, но и задача балансировки вычислений на сервере и клиенте.

Существуют попытки выделить в самостоятельные подходы естественные места разрыва графического конвейера, связанные с преобразованием координат. Можно рассмотреть и другие, более детальные схемы. Так в программируемом пиксельном конвейере выделяют вычисление освещенности и наложение текстур. Однако спорным вопросом остается оправданность их выделения в отдельные схемы распараллеливания конвейера визуализации.

Кроме существенных трудностей при реализации параллельных подходов основными недостатками являются: приостановка счетной задачи во время параллельного выполнения части графического конвейера и простой процессоров во время визуального анализа. Перспективными решениями являются автоматическое выделение особенностей, например области сгущения сетки и предвидение.

Вышеприведенная классификация аналогична разделению подходов по тому, какие данные передаются клиенту:

- проблемно-ориентированный подход (отфильтрованные математические данные) – соответствует параллельной фильтрации данных;
- аппаратно-независимый подход (графические примитивы) - фильтрация и геометрическая обработка выполняются параллельно;
- аппаратно-зависимый подход (коды устройства, видеокарты).

Рассматривая фильтрацию, как интерактивный процесс можно конвейер параллельных фильтров наложить на конвейер визуализации, включая триангуляцию,

децимацию, удаление невидимых примитивов, растеризацию, но все равно остается важным место прекращения параллелизма и визуализацию каких данных поддерживает клиент. Замена части графического конвейера на конвейер параллельных фильтров оправдана, если выполняется основное условие – достижение максимальной эффективности при минимуме затрат.

Закончив необходимую вводную часть, перейдем к формализации визуализации, и в частности, к формализации параллельной фильтрации данных на основе аксиоматического подхода.

Предлагается рассмотреть меру «*сложность интерпретации*», определяемую как количество элементарных понятий-сущностей в модели визуализации. Визуализация адекватна, когда нет проблемы интерпретации. То есть сущности (визуальному объекту) однозначно соответствует понятие (ментальный объект). Также существует сходимость. Например, для получения качественно нового знания в модель визуализации вводятся новых сущностей меньше, чем было. Формально, предлагается построить две монотонно возрастающие последовательности: одна отражает качественные характеристики, другая количественные и между элементами этих последовательностей установить соответствие.

Ключевая формула (1) отражает постулат адекватности визуализации - в процессе интерпретации пользователь извлекает смысл, заложенный в визуализации:

$$(1) S = \lim_{k \rightarrow \infty} I_k \left(\sum_{i=1}^n C_i \right)$$

S – смысл, предельное целеполагающее понятие.

I – оператор взаимодействия, назовем его итератор (интерпретатор). Мощность оператора $|I_k| = k$ (k - количество решений или правил интерпретации).

Пусть существует k решений, тогда $k+1$ строится группой экспертов [46]. Таким образом, вопрос о существовании решения в этой работе снимается на основе экспертного подхода.

$$M = \sum_{i=1}^n C_i$$

M – модель визуализации (метафора); рассматривается, как сумма (объединение) знаков C (семиотический подход) или термов (лингвистический подход) в некоторой топологии.

Очевидна постановка задачи минимума (2) для таких абстрактных метрик, как когнитивное расстояние или информационный разрыв. Напомним, что когнитивное расстояние определяется через усилия пользователя, затраченные на интерпретацию. Возможно и другое определение данной метрики, как расстояния между модельными объектами ввода и вывода и ментальными объектами. Можно провести аналогию сложности интерпретации или переопределенного когнитивного расстояния с таким понятием в области программирования как Колмогоровская или алгоритмическая сложность, определяемая, как минимальная длина текста, реализующая данный алгоритм.

$$(2) |S - \lim_{k \rightarrow \infty} I_k \left(\sum_{i=1}^n C_i \right)| \rightarrow \min$$

В этих формулах предполагается, что значение S (смысл или информация) неизвестно, но существует точная верхняя граница.

Для пояснения общей идеи можно привести пример, рассмотренный в работе [46]. Некоторые исследователи выдвигают тезис, что в рамках параллельных вычислений численные методы не адекватно решают поставленные задачи. Например, поднимается вопрос о существовании матриц большой размерности. Вполне возможно, что, начиная с некоторого N , они всегда плохо обусловлены. В преддверии “заоблачных” вычислений этот вопрос становится особенно актуальным.

Итак, пусть S – аналитическое решение дифференциального уравнения, а S_k – численное решение. В работе [47] рассматривается, как значения этих решений

коррелируются, при условии, конечно, что выполняется свойство монотонной или мажорируемой сходимости. В принципе этот инженерный подход является стандартным, так можно привести в качестве примера фильтр Калмана, где абсолютные координаты рекурсивно уточняются через относительные. Далее в [47] рассматривается предельный случай неопределенности, когда точное решение задачи отсутствует, и в наличии имеются только два вычисленных результата. Причем особый интерес вызывают два решения, полученных не с разным шагом (например, метод Рунге или как в [47] экстраполяция Ричардсона), а разными методами. Возникает вопрос, при каких условиях, кроме выполнения свойства монотонной сходимости и достаточно большого N можно вычислить относительную эффективность или скорость сближения двух решений?

$$(3) \dot{X} = (E - \mu S_k / S)X$$

Это уравнение имеет тот же вид, что и задача оптимального управления, но записанное в нормализованной форме. Знак μ показывает, что в отношении S_k / S присутствует морфологическая неопределенность. Уравнение (3) – это модель с насыщением (конечно, нельзя делить матрицу на функцию, важно, что соотношение $S_k / S \leq 1$ для оценки порядка системы). Таким образом, можно сделать вывод о существовании равновесного состояния. В работе [48] предложено однопараметрическое управление: пусть времени счета задачи соответствует интервал (error bar). Меняя длину интервала, можно изменить время перерасчета по формуле $1 - e^{-t/\tau}$.

Приведя ряд аналогий из области математического моделирования и нечетких множеств, вернемся к формализации визуализации и к уравнению (1), где $S_k = I_k \left(\sum_{i=1}^n C_i \right)$.

Смысл извлекается в процессе взаимодействия с моделью визуализации, представленной в виде системы знаков. В данном случае, можно использовать аналогию с сетями Петри, в которых действие сопоставимо с нашим интерпретатором.

В общем случае k не равно n . (Если рассматривать оператор, как матрицу, а набор знаков, как вектор, то при k не равно n получим недопустимую операцию.) Кроме того, существует вычислительная некорректность, связанная с невозможностью получения решения при большом объеме данных. Эта некорректность зависит от ресурсов $r(n)$, так как для задачи фильтрации наиболее критична нехватка памяти. Ресурсы можно учесть в модели с насыщением, которая подразумевает существование равновесного состояния. Так в формуле (1) существование предела отражает, например, что между взаимодействием и набором знаков достигается равновесие. Для решения задачи предположения о существовании равновесного состояния не достаточно. Необходимо предложить механизмы достижения равновесия.

Предел суммы равен сумме пределов, при условии, что последовательность знаков была монотонно возрастающей. Это требование можно усилить, потребовав однозначность смысла знака (4).

$$(4) \left| I_{k+1} \left(\sum_{i=1}^{n+1} C_i \right) \right| - \left| I_k \left(\sum_{i=1}^n C_i \right) \right| \leq 1, \text{ тогда справедлива формула:}$$

$$(5) S = \sum_{k=1}^{\infty} I_k \left(\lim_{i \rightarrow k} C_i \right) - \text{разложение по базисным знакам ведет к сумме решений совместных систем и, следовательно, к постановке корректной задачи.}$$

Рассмотрим, что дает применение формулы разложения по базисным знакам или раскрытие скобок в формуле (1) на простом примере когда, сумма знаков образует естественный язык. Базисными знаками для естественного языка являются буква, слово, словосочетание, предложение, абзац и т.п. Введем новый базисный знак – **граф понятий**. В данном случае смысл S эквивалентен понятию или термину.

Рассмотрим конкретное определение: “*фильтр – любая операция над данными, изменяющая их количество*”. Понятие *фильтр* зависит от терминов *операция*, *данные*, *количество*, для которых в свою очередь тоже можно дать определения, а в роли оператора I выступает глагол *изменять*. Таким способом можно построить граф понятий. Важно избежать рекурсий и учитывать контекст, так “*операция*” может быть и хирургической. Эти ограничения нужны для однозначности смысла знака. Глубина графа определяет индекс S смысла. Процесс интерпретации не может продолжаться бесконечно. В данном случае предел можно рассматривать, как стремление дать оптимальное определение конкретного понятия. Рассмотренный пример – это частный случай индуктивного подхода, но его анализ произведен в результате дедуктивного или аксиоматического подхода.

При переходе к анализу конкретных систем визуализации основная проблема состоит в определении операций *предел* и *сумма*. Далее рассмотри, что дает переход от предела последовательности к пределу функции (фильтра). Однозначность смысла знака (4) слишком сильное требование. Альтернативное преодоление некорректности – применение фильтров.

В нашем определении фильтра указано, что это любая операция над данными, изменяющая их количество (*не хеширование*). В этом определении *не хеширование* рассматривается, как необходимое условие установление предпорядка:

$$(6) \phi \left(\sum_{i=1}^n C_i \right) = \sum_{i=1}^m C_i, \text{ где } n \text{ не равно } m.$$

Параллельная фильтрации данных – интерактивный процесс над распределенными данными, где на каждом шаге применяется параллельный фильтр (ϕ), происходит передача отфильтрованных данных (tr) и их визуализация (v), имеющая вложенный цикл стандартного (непосредственного) взаимодействия. Параллельная фильтрации данных позволяет разбить взаимодействие на непересекающиеся группы. Например, для параллельного рендеринга это нельзя сделать, так как получится не сумма, а конкатенация $v(tr(\phi))$): $I = I(\phi) + I(tr) + I(v)$.

Рассмотрим только первый член этой суммы. Очевидно, что можно подобрать такой фильтр, при котором $k=m$, то есть система станет совместной. Таким образом, фильтрацию данных можно рассматривать как способ преодоления некорректности, альтернативный ранее приведенному способу, связанному с разложением по базисным знакам.

Целью фильтрации данных является получение необходимой (интересующей) информации за минимальное время. Возможна другая минимаксная постановка задачи аналогичная (2): целью фильтрации данных является получение максимума полезной информации с минимальными затратами $t(n)$ (времени вычисления, взаимодействия, интерпретации и т.п.). Следовательно, операции, которые являются фильтрами, над данными надо выполнять параллельно, минимизировать количество передаваемых данных, применять когнитивную визуализацию. В результате предполагается найти “оптимальное” управление ϕ для системы дифференциальных уравнений вида (3) с учетом ресурсов.

Таким образом, применение аксиоматического подхода к задаче параллельной фильтрации данных, позволяет сделать достаточно очевидный вывод, что данную задачу можно рассматривать, как задачу управления. Далее остановимся на вопросах, требующих дальнейшей детализации.

Фильтрация – интерактивный процесс, нацеленный на получение необходимого результата с помощью визуального анализа за минимальное время, в первую очередь за счет минимизации передаваемой информации. Следовательно, эта технология востребована для двух классов задач:

(1) визуализация данных большого объема ($k \ll n$),

и

(2) в задачах, требующих активного взаимодействия пользователя и системы в процессе визуального анализа (k сопоставимо с n).

Определение фильтра не противоречит условию $k > n$, в этом случае обратная задача не решена, и надо построить модель визуализации (увеличить количество знаков). Для первого класса можно рассмотреть следующую задачу минимакса: *применение множественного вида отображения с одной стороны, обеспечивающего целостное восприятие, но не точно, с другой – детально (более точно), но не всей информации (применение уровня детализации).*

Для задачи балансировки вычислений на сервере и клиенте можно применять вероятностный подход или модель с неопределенностью. Следовательно, адаптация метрик производительности к фильтрации в данном случае должна иметь аналогичный характер.

Важным вопросом является отличие параллельного фильтра от родительского, последовательного фильтра. Если операция выполняется над распределенными данными, которые не являются тесно связанными, то параллельный фильтр совпадает с последовательным. В качестве примера рассмотрим поиск по ключевому слову в Интернете. Предположим, есть слишком большой текст, который не входит на один жесткий диск. Разбив текст на два файла одинаковой длины, его можно поместить на две машины. При таком разбиении большая вероятность того, что некоторое слово будет разделено в два файла. При использовании стандартного фильтра поиска это слово не будет найдено. В данном случае решение довольно простое. Достаточно одного обмена между файлами. Возможно, немного увеличив длины файлов, хранить это слово целиком в обоих файлах (в последнем случае не нужен дополнительный обмен). Также можно запомнить исключение из правила поиска для разорванного слова. Предположим, возникла необходимость хранить отформатированный текст на 24 колонки в двух файлах по 12 колонок, попробуйте написать эффективный параллельный фильтр поиска. Видно, что при написании параллельного фильтра узким местом является точка или граница разрыва распределенных, связанных данных. Оценкой правильности реализации параллельного фильтра является получение того же результата, что и при выполнении последовательного фильтра, но с разумной погрешностью. С точки зрения когнитивной визуализации не обязательно стопроцентное совпадение результатов. Разрывы или возникновение артефактов на границе могут быть несущественны для интерпретации

Как уже отмечалось, информационный разрыв близок к теории возможности, но вероятно с более общей начальной постановкой задачи. В рассмотренном примере переход от топологии поточечной сходимости к топологии непрерывной сходимости, являющийся стандартным в математическом моделировании (например, та же модель хищник - жертва), обусловлен понятием предельной неопределенности. Все же приведем пример похожих рассуждений, которые ведутся именно с точки зрения нечеткой логики для задачи обработки текстовой информации [49]. В этом случае возникает задача по известным вероятностям истинности предложений из I оценить возможную вероятность истинности произвольного предложения. Также появляется подзадача - какие оценки $\mu : I \rightarrow [0, 1]$ являются совместными (т. е. возможными, реализуемыми).

5. Заключение

В области верификации визуализации применимы те же формальные модели, что и для других слабо формализуемых явлений, таких как программирование или экономика. В работе поднимаются вопросы, требующие решения, например:

* Построение иерархии моделей. Трудность в том, что модели, термины, языки программирования обладают высокой степенью синонимичности.

* Масштабируемость видов отображения. Супервычисления связаны с обработкой больших данных (Big Data), поэтому требуется развитие моделей для предельного случая.

* Задача интеграции, когда для верификации недостаточно одного вида отображения или одной модели или одной группы исследователей. Возможные пути решения этой задачи – рассмотрение топологического произведения, построение решеток, грубые множества. В качестве примера рассмотрен бинарный вид отображения.

Вопрос о полноте верификации остается открытым. На начальном этапе мы использовали подходы уровня “модели и полумодели” и/или ментальной модели. Несомненным плюсом формальной модели является то, что на ее основе можно провести численный эксперимент и произвести оценку результатов (валидацию) и тем самым проверить адекватность модели. Так в области визуализации программного обеспечения авторы предложили несколько вычислимых моделей для оценки эффективности параллельных программ. Первая модель связана с понятием темпорального нечеткого числа. Вторая модель является развитием идеи параллельных координат в направлении теории возможности. И, наконец, модель, основанная на рассмотрении скорости сближения, аналогично (3), которая представляется наиболее перспективной.

Кратко остановимся на постановке этой задачи.

Очевидно, что если есть обмены между процессорами, то должно быть насыщение при некотором числе процессоров. Траекторию X будем искать из уравнения $\dot{X} = (1 - O(p/n))X$, где p/n по смыслу энтропия (с ростом числа процессоров увеличивается степень беспорядка системы, а с ростом количества данных, как правило, увеличивается точность алгоритма). И мы хотим, чтобы на границе всегда выполнялось условие $\dot{X} = p$ – линейное ускорение, т. е. $H_p * X = p$ (Напомним, что, модель первого приближения; $\dot{X} = AX$ - это стандартный подход для плохо формализуемых систем, таких как, например, модель хищник – жертва)

H_p - матрица с замерами (интервалами) времени, где главная диагональ – последовательные алгоритмы (время чистого счета), другие элементы характеризуют обмены между процессорами. Размерность матрицы определяется количеством процессоров. Можно считать, что собственные значения отвечают за способ распараллеливания. Важно, что в итоге получаются матрицы определенных шаблонов, так для схемы мастер-рабочий не нулевыми являются главная диагональ и k-строка и k-столбец. Для конвейера или линейки процессоров получается трехдиагональная матрица, а для решетки пятидиагональная и т.д. Очевидным требованием существования решения является неравенство определителя нулю. Такое преобразование, которое сохраняет топологические свойства, называют гомеоморфным [32]. Система совместна. Можно найти ее решение. В этой постановке, как и в (3) содержится морфологическая неопределенность – надо определить расстояние между векторами размерности p и $p+1$. Вполне возможно, что эти модели с неопределенностью являются неадекватными, но, только, проведя численный эксперимент, получим меру неадекватности моделей, после чего можно провести рефакторинг моделей.

В заключении отметим, что хотя затронутые в нашей работе вопросы активно обсуждаются в научной литературе и отражены в большом корпусе текстов, включая статьи в Википедии, но в общем случае проблемы формальной (или хотя бы формализованной) верификации визуализации не решены и требуют новых усилий исследователей.

Литература

1. Globus A. Raible E. 13 Ways to Say Nothing with Scientific Visualization // IEEE Computer. Vol. 27, No. 7, July 1994. Pp. 86-88.
2. Rogowitz B.E., Treinish L.A. How NOT to Lie with Visualization // Journal Computers in Physics. Vol. 10 Issue 3, May/June 1996. Pp. 268-273.

3. Авербух В.Л., Байдалин А.Ю., Бахтерев М.О., Васёв П.А., Казанцев А.Ю., Манаков Д.В. Опыт разработки специализированных систем научной визуализации // Научная визуализация. Квартал 4. Том 2. Номер 4. 2010. Стр. 27-39.
4. Kirby R., Silva C. The need for verifiable visualization // IEEE Computer Graphics and Applications, 28(5) Sep 2008. Pp. 78–83.
5. The EuroRV3: EuroVis Workshop on Reproducibility, Verification, and Validation in Visualization. <http://www.eurorvvv.org/>
6. Dix A., Ghazali M., Gill S., Hare J., Ramduny-Ellis D. Physigrams: modelling devices for natural interaction // Formal Aspects of Computing. Volume 21, Number 6. December, 2009. Pp. 613–641.
7. Dix A.J. Formal Methods // The Encyclopedia of Human-Computer Interaction, 2nd Ed. Aarhus, Denmark: The Interaction Design Foundation. 2014. Available online at https://www.interaction-design.org/encyclopedia/formal_methods.html
8. Workshop on Formal Methods in Human Computer Interaction <https://sites.google.com/site/wsfomchi/>
9. Boyack K.W., Klavans R. Creation of a highly detailed, dynamic, global model and map of science // Journal of the Association for Information Science and Technology. Volume 65, Issue 4. April 2014. Pp. 670–685.
10. Манаков Д.В., Судариков Р.О. Облако тегов для информационной фильтрации данных // XIV Международная конференция “Супервычисления и Математическое Моделирование”. Тезисы. ФГУП РФЯЦ ВНИИЭФ. Саров. 2012, стр. 121-123.
11. Буртной М., Куземин А. Intellectual analysis of visual information // International Journal “Information Technologies & Knowledge”. Vol. 7. Number 2. 2013. Pp. 172-181.
12. Управление, информация, интеллект. (Под редакцией А.И. Берга, Б.В. Бирюкова, Е.С. Геллера, Г.Н. Поварова) М. Мысль. 1976.
13. Johnson-Laird P. Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness. Cambridge, MA. Harvard University Press. 1983.
14. Байдалин А.Ю., Исмагилов Д.Р. Средства представления структур в системах визуализации программного обеспечения // ГрафиКон'2006. Труды Конференции. Новосибирск. ИВМиМГ. 2006. Стр. 271-274.
15. Авербух В. Семиотический подход к формированию теории компьютерной визуализации // Научная визуализация, 2013. Квартал: 1. Том: 5. Номер: 1. Стр. 1-25.
16. Vickers P., Faith J., Rossiter N. Understanding Visualization: A Formal Approach Using Category Theory and Semiotics // IEEE Transactions on Visualization and Computer Graphics, vol. 19, no. 6. June, 2013. Pp. 1048-1061.
17. Fout N., Ma K.-l. Reliable Visualization: Verification of Visualization based on Uncertainty Analysis/ Tech. rep., University of California, Davis, 2012.
18. Dasgupta A., Chen M., Kosara R. Conceptualizing Visual Uncertainty in Parallel Coordinates // Comput. Graph. Forum 31(3). 2012. Pp. 1015-1024.
19. “Info-gap decision theory.” Available at: http://en.wikipedia.org/wiki/Info-gap_decision_theory
20. Ктитров С.В., Кокуев А.А., Кулябичев Ю.П., Крицына Н.А., Горелкин Г.А. Визуализации чувствительности решения задачи линейного программирования со свободными коэффициентами на основе дерева решений // Научная визуализация, 2014. Квартал: 4. Том: 6. Номер: 5. Стр. 8-13.
21. Жолобов Д.А. Использование технологии многомерного анализа для визуализации результатов анализа чувствительности в сложных процессах // Научная визуализация, 2014. Квартал: 4. Том: 6. Номер: 5. Стр. 45-60.
22. Green T.R.G., Petre M. Usability analysis of visual programming environments: a “cognitive dimensions” framework // J. Visual Languages and Computing, 7, 1996. Pp. 131-174.
23. Кулямин В.В. Методы верификации программного обеспечения. М. Институт Системного Программирования РАН, 2008.

24. Авербух В.Л., Байдалин А.Ю., Исмагилов Д.Р., Казанцев А.Ю. Трёхмерные методики визуализации программного обеспечения параллельных и распределённых вычислений // Труды ПАВТ'2008. Челябинск, ЮУрГУ, 2008. Стр. 283-288.
25. Нохрин С.Э. Некоторые свойства множественно-открытых топологий // Современная математика и её приложения. 2005. ИК АНГ. Т. 34. Общая топология. Стр.100-129.
26. Jacob R.J.K. Direct Manipulation // Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, (Atlanta, GA, 1986), N.Y. 1986. V.1, pp. 384-389.
27. North Ch. Toward Measuring Visualization Insight // IEEE Computer Graphics and Applications May/June 2006, Volume: 26, Issue: 3, pp. 20-23.
28. Каркищенко А.Н., Броневиц А.Г., Лепский А.Е. Неаддитивные меры: приложения к обработке информации с высокой неопределенностью // Вестник Южного научного центра РАН. Т.1. Вып. 3. 2005. Стр. 90-95.
29. Pawlak Z. Rough sets // J. Comput. Information Sciences, vol. 11, 1982. P. 341-345.
30. Васёв П.А., Согомоян М.С. Веб-система визуализации, анализа и мониторинга работы программ // ПАВТ-2013., ЮжУрГУ. Челябинск. Сборник Трудов. Том 2. С. 586.
31. Броневиц А.Г., Лепский А.Е. Аксиоматический подход к задаче нахождения оптимального полигонального представления контура/ Интеллектуальные системы, т. 9, вып.1-4, 2005, с.121-134.
32. "Neural Networks, Manifolds, and Topology."
Available at: <http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/>
33. Горбашевский Д.Ю., Казанцев А.Ю., Манаков Д.В. Параллельная фильтрация в системе визуализации параллельных вычислений // ГрафиКон'2006, 1-5 июля 2006. Россия. Новосибирск, Академгородок. Труды Конференции. Новосибирск. ИВМиМФ СО РАН. 2006. С. 333-336.
34. <http://slicedrop.com/>
35. Choudhury A.N.M.I., Bei Wang, Rosen P., Pascucci, V. Topological analysis and visualization of cyclical behavior in memory reference traces // IEEE Pacific Visualization Symposium, PacificVis 2012, Korea, February 28 - March 2, 2012. IEEE. 2012. Pp. 9-16.
36. Авербух В.Л., Анненкова О.Г., Бахтерев М.О., Манаков Д.В. Задачи визуализации программного обеспечения параллельных и распределённых вычислений// ПАВТ-2014, Южный федеральный университет 1-3 апреля 2014 г., г. Ростов-на-Дону. Сборник трудов С. 7-18.
37. Яу Н. Искусство визуализации в бизнесе. Издательство: "Манн, Иванов и Фербер" Москва. 2013. – 338 с.
38. Cornelissen B., Zaidman A., Van Rompaey B., van Deursen A. Trace Visualization for Program Comprehension: A Controlled Experiment // Proc. 17th IEEE Int'l Conf. Program Comprehension, 2009. Pp. 100-109.
39. Jerding D.F., Stasko J.T. The information mural: A technique for displaying and navigating large information spaces // IEEE Trans. Vis. Comput. Graph., 1998, 4(3). Pp. 257–271.
40. Тюгашев А.А., Богатов А.Ю., Шулындин А.В. Визуальный подход к верификации управляющих программ реального времени // Вестник Самарского государственного аэрокосмического университета № 1 (32). 2012. Стр. 219-225.
41. Теплов А.М. Об одном подходе к сравнению масштабируемости параллельных программ // Вычислительные методы и программирование. 2014. Т. 15. Выпуск 4. Стр. 697-711.
42. Гурьева Я.Л., Ильин В.П. Методы редукции и параллельные технологии решения СЛАУ // Труды международной научной конференции "Параллельные Вычислительные Технологии" (ПАВТ'2015). Екатеринбург, 31 марта - 2 апреля 2015. Челябинск, Издательский центр ЮУрГУ. 2015. Стр.114-122.
43. Brodlie K., Brooke J., Chen M., Chisnall D., Fewings A., Hughes C., John N.W., Jones M.W., Riding M., Roard N. Visual Supercomputing - Technologies, Applications and Challenges // Computer Graphics Forum, vol. 24, Number 2, 2. 2005. Pp.217-245.

44. Манаков Д.В. Анализ параллельных визуальных технологий // Вычислительные технологии. Том 12. N 1. 2007. Стр. 45-59.
45. Корж О. В. Методы параллельной визуализации научных данных. М. МГУ имени М.В.Ломоносова. 2011. (На правах рукописи)
46. Файн В.С. Распознавание образов и машинное понимание естественного языка. М. Наука, 1987.
47. Камм Дж. Р., Райдер У. Дж., Витковски У. Р., Тракано Т. Г., Бен-Хаим И. Анализ численных погрешностей метода в условиях недостаточности информации // XIV Международная конференция «Супервычисления и Математическое Моделирование». Тезисы. ФГУП «РФЯЦ ВНИИЭФ». Саров. 2012, стр. 101.
48. Sarkar A.; Blackwell A.F; Jamnik M.; Spott M. Interaction with uncertainty in visualisations // Eurographics/IEEE VGTC Conference on Visualization (EuroVis 2015) (The Eurographics Association). 2015. [doi:10.2312/eurovisshort.20151138](https://doi.org/10.2312/eurovisshort.20151138)
49. Пальчунов Д.Е., Яхьяева Г.Э. Нечёткие алгебраические системы // Вестник НГУ. Сер. матем., мех., информ., 10:3 (2010). Стр. 76-93.

Verification of visualization
Dmitry Manakov, Vladimir Averbukh
Institute for Mathematics and Mechanics named by N.N. Krasovskii
Urals Branch of Russian Academy of Science,
Urals Federal University
Ekaterinburg

Abstract

The paper is devoted the problems of verification of computer visualization based on formal or formalized approaches. Verification of visualization means existence of formal model. This model differs from similar models for poorly formalizable phenomena (for example, formal verification of the software) generally speaking only in application environment. Possible approaches to models of visualization are reported in the article. The applications of multiple (binary) views are considered in details for three tasks of Scientific Visualization, Information Visualization, and Software Visualization. Also formalization of visualization and data parallel filtration are considered. This formalization bases on linguistic and semiotic approaches.

Key words

Verification in Visualization, Validation in Visualization, Uncertainty Analysis

REFERENCES

1. Globus A. Raible E. 13 Ways to Say Nothing with Scientific Visualization // IEEE Computer. Vol. 27, No. 7, July 1994. Pp. 86-88.
2. Rogowitz B.E., Treinish L.A. How NOT to Lie with Visualization //Journal Computers in Physics. Vol. 10 Issue 3, May/June 1996. Pp. 268-273.
3. Averbukh V.L., Baydalin A.Yu., Bahterev M.O., Vasev P.A., Kazantsev A.Yu., Manakov D.V. Experience in Developing of Specialized Scientific Visualization Systems // Scientific Visualization. 2010. Quarter 4. Volume 2. Number 4. Pp. 27-39.
4. Kirby R., Silva C. The need for verifiable visualization // IEEE Computer Graphics and Applications, 28(5) Sep 2008. Pp. 78–83.
5. The EuroRV3: EuroVis Workshop on Reproducibility, Verification, and Validation in Visualization. <http://www.eurorvvv.org/>
6. Dix A., Ghazali M., Gill S., Hare J., Ramduny-Ellis D. Physigrams: modelling devices for natural interaction // Formal Aspects of Computing. Volume 21, Number 6. December, 2009. Pp. 613–641.

7. Dix A.J. Formal Methods // The Encyclopedia of Human-Computer Interaction, 2nd Ed. Aarhus, Denmark: The Interaction Design Foundation. 2014.
Available online at https://www.interaction-design.org/encyclopedia/formal_methods.html
8. Workshop on Formal Methods in Human Computer Interaction
<https://sites.google.com/site/wsfomchi/>
9. K.W. Boyack K.W., Klavans R. Creation of a highly detailed, dynamic, global model and map of science // Journal of the Association for Information Science and Technology. Volume 65, Issue 4, April 2014. Pp. 670–685.
10. Manakov D., Sudarikov R. Tag Cloud For The Information Data Filtration // XIV International Seminar “Super-Computations and Computer Simulations”. Sarov, 2012. Pp. 121-123.
11. Burtnoy M., Kuzemin A. Intellectual analysis of visual information // International Journal “Information Technologies & Knowledge”. Vol. 7. Number 2. 2013. Pp. 172-181.
12. Control, Information, Intellect. M. Misl’. 1976.
13. Johnson-Laird P. Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness. Cambridge, MA. Harvard University Press. 1983.
14. Bajdalin A., Ismagilov D. Structure representation in software visualization systems // International Conference Graphicon 2006, Novosibirsk, Akademgorodok, Russia. Pp. 271-274.
15. Averbukh V. Semiotic Approach to Forming the Theory of Computer Visualization // Scientific Visualization. 2013. Quarter 1. Volume 5. Number 1. Pp. 1-25.
16. Vickers P., Faith J., Rossiter N. Understanding Visualization: A Formal Approach Using Category Theory and Semiotics // IEEE Transactions on Visualization and Computer Graphics, vol. 19, no. 6. June, 2013. Pp. 1048-1061.
17. Fout N., Ma K.-I. Reliable Visualization: Verification of Visualization based on Uncertainty Analysis/ Tech. rep., University of California, Davis, 2012.
18. Dasgupta A., Chen M., Kosara R. Conceptualizing Visual Uncertainty in Parallel Coordinates // Comput. Graph. Forum 31(3). 2012. Pp. 1015-1024.
19. “Info-gap decision theory.”
Available at: http://en.wikipedia.org/wiki/Info-gap_decision_theory
20. Ktitrov S.V., Kokuyev A.A., Kulyabichev Yu.P., Kritsyna N.A., Gorelkin G.A. Visualization the sensitivity of the solution of a linear programming problem with the free parameters based on a decision tree // Scientific Visualization. 2014. Quarter 4. Volume 6. Number 5. Pp. 8-13.
21. Zholobov D.A. Using multidimensional analysis technologies for complex process sensitivity analysis visualization // Scientific Visualization. 2014. Quarter 4. Vol. 6. Number 5. Pp. 45-60.
22. Green T.R.G., Petre M. Usability analysis of visual programming environments: a “cognitive dimensions” framework // J. Visual Languages and Computing, 7, 1996. Pp. 131-174.
23. Kuliamin V. Methods of Software Verification M. Institute for System Programming of the Russian Academy of Sciences. 2008.
24. Averbukh V.L., Baydalin A.Yu., Ismagilov D.R., Kazantsev A.Yu. Three-dimensional Methods for Software Visualization of Parallel and Distributed Computing // Proceedings of PAVT-2008. Chelyabinsk. Pp. 283-288.
25. Nokhrin S. Some properties of multiple-open topology // Modern Mathematics and Applications. 2005. Vol. 34. General Topology. Pp. 100-129.
26. Jacob R.J.K. Direct Manipulation // Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, (Atlanta, GA, 1986), N.Y. 1986. V.1, pp. 384-389.
27. North Ch. Toward Measuring Visualization Insight // IEEE Computer Graphics and Applications May/June 2006, Volume: 26, Issue: 3, pp. 20-23.
28. Karkischenko A., Bronevich A., Lepskiy A. Non-additive measures: application to information processing with high uncertainty // Bulletin of the Southern Research Center of the RAS. Vol. 1: ISS. 3.2005. Pp. 90-95.
29. Pawlak Z. Rough sets // J. Comput. Information Sciences, vol. 11, 1982. P. 341-345.

30. Vasev P., Sogomonyan M. Web-system for Visualization, Analysis and Monitoring of Program Executions // Proceedings of PAVT-2013. Chelyabinsk. Volume 2. P. 586.
31. Bronevich A., Lepsky A. The axiomatic approach to the problem of finding the optimal polygonal contour representation // Intelligence Systems. Vol. 9, Issue 1-4. 2005. Pp. 121-134.
32. "Neural Networks, Manifolds, and Topology."
Available at: <http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/>
33. Gorbashvskiy D., Kazantsev A. Manakov D. Parallel Filtering in Parallel Computing Visualization System // International Conference Graphicon 2006, Novosibirsk, Akademgorodok, Russia. Pp. 333-336.
34. <http://slicedrop.com/>
35. Choudhury A.N.M.I., Bei Wang, Rosen P., Pascucci, V. Topological analysis and visualization of cyclical behavior in memory reference traces // IEEE Pacific Visualization Symposium, PacificVis 2012, Korea, February 28 - March 2, 2012. IEEE. 2012. Pp. 9-16.
36. Averbukh V., Annenkova O., Bakhterev M., Manakov D. The Tasks of Software Visualization for Parallel and Distributed Computing // Proceedings of Conference "Parallel computational technologies 2014", Southern Federal University, Rostov-on-Don. Pp. 7-18.
37. Yau N. Visualize This. The FlowingData Guide to Design, Visualization, and Statistics. Wiley Publishing. 2011.
38. Cornelissen B., Zaidman A., Van Rompaey B., van Deursen A. Trace Visualization for Program Comprehension: A Controlled Experiment // Proc. 17th IEEE Int'l Conf. Program Comprehension, 2009. Pp. 100-109.
39. Jerding D.F., Stasko J.T. The information mural: A technique for displaying and navigating large information spaces // IEEE Trans. Vis. Comput. Graph., 1998, 4(3). Pp. 257-271.
40. Tugashev A., Bogatov A., Shulyndin A. The visual approach to verification of control real time programs // Bulletin of Samara State Aerospace University. No. 1 (32). 2012. Pp. 219-225.
41. Teplov A. An approach to comparing the scalability of parallel programs // Numerical Methods and Programming. 2014. V. 15. Issue 4. Pp. 697-711.
42. Gurieva J.I., Ilyin V.p. Methods of reduction and parallel technologies solution SLAE // Proceedings of Conference "Parallel computational technologies 2015". Chelyabinsk, SUSU Publishing Center. 2015. Pp. 114-122.
- 43 Brodlie K., Brooke J., Chen M., Chisnall D., Fewings A., Hughes C., John N.W., Jones M.W., Riding M., Roard N. Visual Supercomputing - Technologies, Applications and Challenges // Computer Graphics Forum, vol. 24, Number 2, 2. 2005. Pp.217-245.
- 44 Manakov D. Analysis of Parallel Visual Technologies // Computer Technologies. Vol 12. N 1. 2007. Pp. 45-59.
- 45 Korzh O. Methods of Parallel Scientific Visualization. Moscow. MSU. 2011.
- 46 Fain V. Pattern recognition and machine understanding of natural language. M. Nauka, 1987.
- 47 Kamm Jr. Rider W., Witkowski W., Trakano T., Ben-Haim I. Numerical analysis of errors method under conditions of insufficient information // XIV International Seminar "Super-Computations and Computer Simulations". Sarov, 2012. Pp. 101.
48. Sarkar A.; Blackwell A.F; Jamnik M.; Spott M. Interaction with uncertainty in visualisations // Eurographics/IEEE VGTC Conference on Visualization (EuroVis 2015) (The Eurographics Association). 2015. [doi:10.2312/eurovisshort.20151138](https://doi.org/10.2312/eurovisshort.20151138)
49. Palchunov D., Yahyayeva G. Fuzzy algebraic systems// Bulletin of the NSU. Math., Mech. & Informatics. 10: 3 (2010). Pp. 76-93.