

Стохастическая семантика больших данных (параллельные вычисления и визуализация)

Д.В. Манаков^{1,А}, П.А. Васёв^{2,А}

Институт математики и механики им. Н.Н. Красовского УрО РАН

¹ ORCID: 0000-0001-6852-8096, manakov@imm.uran.ru

² ORCID: 0000-0003-3854-0670, vasev@imm.uran.ru

Аннотация

В первую очередь в работе рассмотрена задача верификации (формализации) системы онлайн визуализации и параллельных вычислений с точки зрения динамических систем как развитие теории вычислительной сложности для случайных процессов. Рассмотрение задач, связанных с действительно большими данными, неизбежно ведёт к использованию блочного подхода, который применяется и в теории информации и стохастических дифференциальных уравнениях. В качестве естественной метафоры выбраны сигнальные графы – это граф, в узлах которого определена спектральная функция, в рассматриваемых примерах это функция цвета, высоты или количества данных. В параллельных вычислениях блок можно ассоциировать с вычислительным юнитом (процессором) и рассмотреть задачу максимизации энтропии (производительности). В разрабатываемой системе онлайн визуализации и параллельных вычислений для геометрического распараллеливания можно реализовать и сравнить стационарный случайный процесс (равновероятные сообщения, реализованные с использованием широко вещания) и установившийся случайный процесс (сообщения точка-точка), которые имеют разные аналитические решения. В совокупности это позволяет сделать вывод, что предложенная реализация стационарного процесса имеет определенную новизну, кроме того она задумывалась как более удобная для автоматизированного распараллеливания. Также рассмотрены задачи автоматической балансировки нагрузки (задача интерполяции) и оптимальной масштабируемости параллельных вычислений (задача экстраполяции). В области верификации визуализации сделано не так много – предложена визуализация сеток, рассматриваемая как параметризованная модель белошумного случайного процесса. Конечно, данную работу нельзя считать завершенной, но направление, которое авторы назвали стохастическая семантика очевидно является перспективным. Авторы намерены вплотную заняться установившимися возмущенными процессами в области визуализации, в том числе с учётом человеческого фактора (приведены наброски формализации в виде обсуждения).

Ключевые слова: сигнальные графы, динамические системы, балансировка нагрузки, энтропия, визуализация цифровой модели поверхности.

1. Введение

Существует такое теоретическое направление как верификация программного обеспечения (ПО). Поскольку визуализация технически тоже является вычислением, хотелось бы иметь общую математическую модель верификации как для ПО, так и для визуализации. Верификация визуализации – формальное (математическое) доказательство правильности визуализации. Разрабатываемые математические модели должны ответить на вопрос правильно ли мы решили задачу, поставленную пользователем, оценить качество и эффективность визуализации, перспективы развития современных

тенденций в области визуализации. В частности, виртуальной реальности (VR), web-визуализации, онлайн визуализации, больших данных, параллельных вычислений.

Для верификации в качестве базовой модели выбраны стохастические дифференциальные уравнения (СДУ), как наиболее общее решение, учитывающее шум, в сочетании с теорией информации. Эту гипотезу назовем стохастической семантикой.

Первоначальной моделью обработки изображения является модель шумоподавления, которая получила широкое распространение, в частности, из-за распространения таких пакетов как “Stable Diffusion“, генерирующих изображение по тексту. В работе [1] описаны три варианта шумоподавления:

1. Denoising Diffusion Probabilistic Models (DDPM) - вероятностная диффузионная модель шумоподавления;
2. Noise Conditioned Score Networks (NCSNs) – оценочные сети с условными шумами;
3. Stochastic Differential Equations (SDEs) – СДУ, диффузионный процесс.

Как отмечается, первые две модели являются частным случаем диффузионного процесса, вероятно, которые можно получить, воспользовавшись Марковским свойством, чего и не хватает с точки зрения верификации. В принципе и для других работ по этой тематике характерен инженерный подход, то есть показывается на примерах, что модель рабочая, производится валидация при недостаточной верификации (обычно используется правдоподобие). Наибольшее распространение получил подход – оценочные сети с условными шумами. В этой же работе [1] сказано, что он используется для решения транспортной задачи, а именно генерации суперразрешения. Этот подход также известен как baking (запекание, модель отжига). Также можно сослаться, например, на запекание нормалей в Blender.

В принципе, те же самые подходы используются или могут применяться и для распознавания речи, но в этом случае спектр состоит из фонем, а относительно визуализации спектр – это базис цветов, например, RGB.

Можно порекомендовать к изучению книгу [2] по обработке изображений на сигнальных графах (частный случай СДУ) с достаточно прозрачной математикой. Сигнальный граф – сеть, в узлах которой заданы значения функции, следовательно, можно определить операторы дифференцирования в узлах и ребрах графа и перейти к СДУ в частных производных. Изначально сигнальные графы применялись в электротехнике, где рассматриваемые подходы являются задачей автоматического управления, цель которого построить передаточную функцию, например, по формуле Мэйсона. Для балансировки нагрузки также применяется метод диффузии [3].

В тоже время хотелось бы, чтобы кроме теории присутствовал определенный прагматизм – применение моделирования для решения конкретных задач. В этой работе представлены две задачи, одна в области параллельных вычислений, другая в области визуализации: формализации динамической системы онлайн визуализации и параллельных вычислений (ДС) на сигнальных графах, визуализация сеток как параметризованная модель бел шумного случайного процесса.

2. Стохастическая семантика

Рассмотрим визуализацию (интерактивный процесс и анимацию) с точки зрения динамических систем. Определим визуальный процесс одновременно как параллельный процесс, рассматриваемый как взаимодействующие последовательные процессы (Хоара) с точки зрения программирования и как случайный процесс с точки зрения математического моделирования. Случайный процесс – параметризованный набор (семейство) случайных величин [4]. Но в отличие от типовых моделей СДУ вводятся два линейно независимых параметра: количество данных и время (две случайные величины) $\omega = \omega_N \times \omega_t$, где \times - декартово произведение или каррирование в зависимости от модели, теория информации как раз нужна для того, чтобы от произведения перей-

ти к сумме, воспользовавшись свойством аддитивности энтропии. То есть случайный процесс определим как композицию функций:

$$f(t, N, \omega) = f_t(t, \omega_t) \circ f_N(N, \omega_N),$$

причем количество (объем) данных N может быть сколь угодно большим, например, не входит в оперативную память одного процессора.

В общем случае необходимо рассматривать сумму подобных композиций, например, композицию для таких двух типов случайных событий, как обмен сообщениями и реактивные вычисления. Отметим некоторые отличия общего случая от теоремы Колмогорова-Арнольда, утверждающей, что каждая многомерная непрерывная функция может быть представлена в виде суперпозиции непрерывных функций одной переменной:

1. В работе используется вероятностный подход (рассматриваются случайные процессы) – вероятно можно (надо в дальнейшем) перейти от СДУ и теории информации к weak (слабый) КАМ [5] и теории среднего поля;

2. Функция многих переменных усилена спектральным свойством – фактически рассматриваются векторные функции многих переменных;

3. Фильтрация данных рассматривается как частный случай реактивных вычислений, для конвейера фильтров важен порядок выполнения фильтров;

4. В отличие от количества информации оценка качества информации имеет субъективный характер. Важной задачей является формальное определение, понижающее уровень субъективности, таких понятий как контекст, качество, когнитивность и перцептивность информации, так чтобы при решении задач можно было бы воспользоваться СДУ или weak КАМ. В области визуализации часто рассматривается решение обратной задачи (в теории категорий вводится понятие обратного предела), например, можно предложить такой вид отображения, чтобы он являлся непрерывным с точки зрения зрительного восприятия (перцептивная непрерывность).

Для верификации программного обеспечения, как и для визуализации наравне с денотационной семантикой, структурами событий можно использовать и СДУ. Стохастическую семантику можно определить как игровую семантику с шумом, а можно сразу использовать СДУ. Хотя эти два подхода эквивалентны и относительно программирования рассматривается модель потока данных, но в результате получаются разные математические модели.

В основу верификации положен переход от декларативных определений к формальным (верифицируемым) определениям. В работе [6] семиотическое определение метафоры визуализации рассматривается как непрерывное отображение исходного множества на целевое множество. При этом в стандартное определение метафоры по Лакоффу [7] добавлено только свойство непрерывности. Аналогичным и наиболее известным подходом в области верификации программного обеспечения является денотационная семантика Скотта [8]. В результате применения аксиоматического (семиотического) подхода к формализации визуализации и параллельной фильтрации данных получен некоторый аналог задачи стохастического управления с морфологической неопределенностью. Полученный результат скорее является ментальным (полезным для понимания природы явления), чем математическим. В данной работе рассматриваются задачи в рамках априорной неопределенности, а именно СДУ и теории информации.

Рассмотрение задач, связанных с действительно большими данными, неизбежно ведет к использованию блочного подхода, который применяется и в теории информации и стохастических дифференциальных уравнениях. Большие данные – предельный (на данный момент) случай обработки данных, при котором универсальные подходы к анализу и визуализации не работают или неэффективны. Тогда в качестве больших данных могут рассматриваться многомерные и многокатегориальные данные, данные большого объема, данные с неполной информацией (модель с неопределенностью). Предельный случай формирует вызовы, на которые необходимо ответить, чтобы двигаться дальше. Решение возникающих проблем приводит к тому, что сегодняшние

“большие данные” завтра становятся нормой [9]. При анализе и визуализации больших данных рассмотрение предельной неопределенности – неопределенности, которая имеет конечный предел в конкретной метризуемой топологии, является вынужденным. Поскольку разрабатываются вычислимые модели, поясним это определение на примере вычислимой функции, в определение которой используется специальный элемент, означающий неопределённость, соответствующая случаю, когда алгоритм зависает. Поскольку рассматривается процесс, алгоритм не может зависнуть, он всегда интерактивно разрешается в результате отладки правильности и эффективности программы. Почему же до сих пор не предложена асимптотическая теория алгоритмов или автоматов, появление которой прогнозировал Р. Л. Стратонович?

3. Параллельные вычисления

Параллельные вычисления являются генератором больших данных. В ряде случаев, вместо того чтобы хранить все данные, предпочтительнее их обработать средствами визуальной аналитики, а затем принять решение о том нужны ли эти данные? Как следствие, возникает потребность в реализации онлайн визуализации. А если необходимо хранить данные, то их надо размещать не на клиенте, а в облаке или на параллельной файловой системе, как следствие, возникает потребность в реализации удаленной визуализации. Задачи онлайн и удаленной визуализации архитектурно очень близки, поэтому должна существовать единая среда разработки таких программ. Как уже отмечалось, визуализация является тоже вычислением, поэтому разрабатывается единая среда онлайн визуализации и параллельных вычислений. В работе [10] основное внимание уделено валидации этой среды, в том числе задаче балансировки нагрузки средствами визуальной аналитики. В данном случае речь пойдет о верификации, а именно о формализации динамической системы онлайн визуализации и параллельных вычислений. Основное внимание уделено оценке эффективности параллельных программ, учитывающая количество данных, а именно рассматриваются задачи максимизации энтропии для блоковых моделей, где блок – процессор имеет физические и логические уровни.

Прежде чем перейти к формализации, перечислим основные спецификации среды онлайн визуализации и параллельных вычислений:

1. Универсальность системы-онлайн визуализации и параллельных вычислений;
2. Рассмотрение с точки зрения динамической система;
3. Применимость параметрических (параметризованных) моделей, в том числе управление через параметры.

Во-первых, технически визуализация является вычислением (конечно, с некоторыми особенностями), поэтому можно предложить некий универсальный способ программирования, включая язык программирования, который позволит эффективно реализовывать алгоритмы визуализации, их связь с параллельными вычислительными программами, а также сами эти программы, то есть произвольные вычислительные алгоритмы.

Во-вторых, эта среда построена вокруг идеи представления параллельного процесса вычисления в форме множества зависимых задач [11]. Зависимость задач означает, что входные аргументы для некоторой задачи определяются результатами вычисления других задач. Список задач и зависимостей между ними определяется пользовательским алгоритмом, который может иметь как форму конечного вычисления, так и случайного процесса. Задержки, возникающие перед приемом и передачей данных, будем рассматривать как шум, который в идеале стремится к нулю. Задачи поступают на вход процессу планирования, который определяет, на каком вычислительном узле следует выполнять ту или иную задачу. Затем, по мере разрешения зависимостей, задачи выполняются. Естественно, что от качества используемого алгоритма планирования зависит эффективность всего параллельного вычисления. Соответствие задачи узлу будем

называть назначением. Набор таких назначений и зависимости между задачами будем называть планом выполнения. Подчеркнем, что план выполнения строится во времени, по мере поступления задач и их аргументов (данных). В итоге в работе [10] опытным путем предложена формула балансировки нагрузки, одна из задач данной работы показать, что формула соответствует задачи интерполяции на сигнальных графах [2]. Для которых должно быть определено значение функции в узлах. Эту функцию можно рассматривать как зависящую от времени, но реальное значение времени передачи данных можно получить только после определенного шага выполнения программы, то есть офлайн, что неприемлемо. Поэтому в данном случае рассматривается функция, зависящая от количества данных (аналогично вычислительной сложности). Действительно, сложность параллельного алгоритма задаётся иногда как отношение сложности передачи данных к сложности последовательного алгоритма, например, как $O(N^2)/O(N^3)$. Конечно, это грубая экспертная оценка. Возникает вопрос: можно ли уточнить эту оценку, например, определив передаточную функцию через каноническое разложение, например, Тейлора?

В-третьих, наиболее важной особенностью среды онлайн визуализации и параллельных вычислений является возможность управления как вычислениями, так и визуализацией через параметры. Например, параллельный рендеринг можно организовать не как цикл, зависящий от положения камеры, а как реактивное вычисление. Подобные подходы не только экономят память, но и востребованы для исследовательских задач, кроме того они формализуемы с применением СДУ. В качестве примера, будет рассмотрена визуализация сеток как параметризованная модель белозумного случайного процесса. В частности, для визуализации цифровой модели поверхности.

Перейдем к формализации динамической системы онлайн визуализации и параллельных вычислений. Кроме формального описания этой автоматизированной системы программирования, как уже отмечалось, нас интересуют две задачи: автоматическая балансировка нагрузки и оценка эффективности параллельных вычислений, учитывающая количество данных.

3.1. Формальное описание динамической системы онлайн визуализации и параллельных вычислений

Давайте рассмотрим конструкцию универсальной алгебры, которую назовем структурой процессов: $P\mathcal{U}(\mathbb{N}, +, \circ, t\{\omega_i\})$, где \mathbb{N} – множество натуральных чисел, на которое отображаются идентификаторы объектов, $t\{\omega_i\}$ – логическое время, зависящее от списка случайных событий.

Цель автоматической балансировки нагрузки определим следующим образом: для любого процесса pi из структуры процессов построить бинарное гомоморфное отображение G – сигнальный граф, который отображает процесс на СО-ОПН (Concurrent Object-Oriented Petri Nets [12]) с минимальным шумом (задержками): $G: pi \rightarrow CO-OPN$. Дальнейший анализ показал, что рассматриваются польские пространства.

Польское пространство – пространство, гомеоморфное полному метрическому пространству со счётным плотным подмножеством. В частности, в области визуализации рассматривается пример для сепарабельного банахова пространства. Таким образом, существует возможность перехода к weak КАМ.

СО-ОПН – параллельная объектно-ориентированная сеть Петри – одна, наиболее известная реализации алгебраической сети Петри. Сеть Петри – дискретная динамическая система (направленный двудольный граф), специально придуманная для параллельных вычислений. Поскольку любое дерево, включая синтаксическое дерево, представимо в виде двудольного направленного графа, гомоморфный образ этого отображения G ничем другим кроме модификаций сетей Петри являться не может. Мотивацией применения сигнальных графов является получение автономной системы дифференциальных уравнений. Аналогичным подходом к применению сигнальных графов

является графо-символическое программирование [13]. (Можно привести и другие примеры автоматизации программирования на графах, например, информационные графы, вероятно список работ по данной тематике обширный, но авторы делают акцент не на программирование, а на формализацию, в частности на развитие теории вычислительной сложности для параллельных вычислений). На наш взгляд, введение типа дуги (последовательная, параллельная, терминирующая) является лишним, поскольку есть понятие пути графа. С точки зрения автоматизации программирования и с целью избегания излишнего копирования данных, а возможно и попадания в кэш должно выполняться правило максимизации пути сигнального графа (графа задач) на каждом процессоре, которое можно представить как функцию цены. В сети Петри есть понятие перехода, аналогично в графо-символическом программировании используется метка предиката, а в динамической системе онлайн визуализации и параллельных вычислений на нижнем уровне реализованы стандартные языковые конструкции фьючерсы (future, promise, обещания) и миксины (mixin, примеси), но в параллельном варианте. Обещание – это предикат, зависящий от типа данных (параметров объекта), в данном случае, есть прямая аналогия с раскрашенными функциональными сетями Петри. Примесь – это предикат, зависящий от номера процессора, следовательно, зависящий от схемы распараллеливания. В контексте больших данных реализация примесей является существенным дополнением к сетям Петри. Далее будет показано, что с точки зрения теории информации эффективная реализация примесей должна основываться на ширококовещании и векторной маршрутизации потоков, в частности на буферизации сообщений. Некоторые примеры примесей также будут приведены.

Логическое время определяет порядок событий. В ДС возможны два типа событий – реактивные вычисления и обмен сообщениями. Поскольку энтропия является аддитивной, то с точки зрения теории информации эти две задачи можно рассматривать независимо, например, можно ввести два логических времени. Хотя авторов интересует формализация задач, связанных с обменом сообщениями, немного остановимся на реактивных вычислениях.

В реактивных вычислениях случайным событием является (онлайн или офлайн) изменение значения параметра. По аналогии с абстрактными типами данных, такой подход авторы называют абстрагированием параметров. Самой простой реализацией онлайн визуализации, но вероятно не самой эффективной, является офлайн изменение параметра. То есть все данные текущей итерации находятся на клиенте, пока пользователь занимается визуальной аналитикой, на вычислителе можно посчитать следующую итерацию. В этом случае нет проблемы с установлением порядка событий таким образом можно ввести два ортогональных логических времени. Формально абстрагирование параметров можно рассматривать как лямбда-апликацию и применять, например, аппарат теории категорий. Хотя монада определяется как функтор с дополнительной структурой в контексте больших данных, между ними существуют принципиальные отличия. Для функторов не существует проблемы асимптотической сходимости, достаточно, чтобы параметры были линейно независимыми или чтобы выполнялось свойство мультипликативности (в теории категорий используют термин каррирование). Для монад можно рассматривать сходимость, если определен повторный интеграл через кратный интеграл, например, для решеток, кривых, заполняющих пространство, мозаик, случайных графов, иначе редукция синтаксического дерева или графа задач – это задача дискретной оптимизации на графах. Фактически в ДС обещание – это монада. Таким образом, структура процессов является не просто конструкцией универсальной алгебры, а алгебраической системой. Её сигнатурой называется набор функциональных и предикатных символов с их арностями.

Задачи, связанные с обменом сообщениями, будут рассмотрены с позиции априорной неопределенности, хотя авторы и предполагают ответ. Так оценка эффективности параллельных вычислений – это неоднородная задача Марковского управления, где правая часть уравнения равна количеству блоков (процессоров) в блочной модели. А

задача автоматической балансировки нагрузки (алгоритм планирования) соответствует задаче интерполяции на сигнальных графах. Алгоритм планирования - это алгоритм, который выбирает, на каком исполнителе следует выполнять ту или иную задачу. Поскольку объем оперативной памяти на разных классах процессоров разный (а быть может заранее и не известен), то как решение можно рассматривать транспортную задачу, например, с переменным числом выходов. Вроде бы существует решение этой постановки, как задачи автоматического управления – построения передаточной функции с использованием полинома Гурвица, но в ДС применяется другой подход, если памяти не хватает, то задача переносится на другой исполнитель, в результате возможна разбалансировка нагрузки. В целях обобщения моделирования далее исполнителя будем называть процессором.

3.2. Задача автоматической балансировки нагрузки

Давайте рассмотрим задачу интерполяции на сигнальных графах. Для начала в области визуализации, см. Рис. 1.

Пусть функция $f^0: \mathcal{V}_0 \rightarrow \mathbb{R}$, где $\mathcal{V}_0 \subset \mathcal{V}$ подмножество вершин графа с известными значениями. Задача интерполяции сводится к рассмотрению уравнения: $\Delta f(v_i) = 0$ на $\mathcal{V} \setminus \mathcal{V}_0$. В первоисточнике [2] Лапласиан задается специфическим образом для изотропного и анизотропного диффузионного процесса (фактически переопределяется оператор набла), что для дальнейших рассуждений не существенно. В работе рассмотрены простые примеры распараллеливания – изотропные диффузионные процессы. Авторы считают, что такие примеры распараллеливания как: задачи дискретной оптимизации на графах и решение СЛАУ методом Холецкого являются анизотропными диффузионными процессами.



Рис.1. Схема интерполяции [2].

Алгоритм планирования должен динамически распределять граф задач по процессорам, так чтобы они не простаивали. Для задачи [14] распараллеливания одномерного массива на линейке процессоров с теневой гранью шириной в единицу, то есть на каждом шаге итерации вычислялась функция: $f(mu[i - 1], mu[i], mu[i + 1], \sigma)$, σ – среднее значение функции на предыдущем шаге. Опытным путем в сочетании с визуальной аналитикой была получена следующая формула балансировки нагрузки [10]:

$$E_i = |\mathcal{V}_i \setminus \mathcal{V}_i^0| + k \ln(queue_i),$$

где математическое ожидание нагрузки процессоров – среднее значение нагрузки процессора E_i (очередная задача назначается исполнителю с наименьшим значением E_i), $|\mathcal{V}_i \setminus \mathcal{V}_i^0|$ - количество аргументов задачи, отсутствующих в кэше исполнителя, $queue_i$ - текущий размер списка назначенных и пока не решенных заданий (очереди сообщений). Вычислительная сложность пирамидальной (оптимальной) обработки очереди имеет логарифмическую зависимость. Одной из проблем было подбор коэффициента $k = 0.1$. Можно предположить, что он является характеристикой архитектуры распараллеливания и равен отношению (физической) пропускной способности канала к скорости вычисления процессора.

Чтобы перейти к общему случаю, давайте рассмотрим другую схему геометрического распараллеливания: задано равномерное распределение данных (двумерного массива) по процессорам, $N \times N$ - количество данных на каждом процессоре, N – переменный параметр, $f_i = N^2$, $\nabla f_i = 2N$. Задачу интерполяции для балансировки нагрузки запишем в следующем виде:

$$\nabla E_i = \nabla |2N(\mathcal{V}_i \setminus \mathcal{V}_i^0)| + \nabla k \ln(\text{queue}_i) = 0.$$

Очевидно, что для рассматриваемого примера $2N$ – длина сообщения. В соответствии с Марковским свойством диффузионных процессов [4] $\lim_{i \rightarrow \infty} \nabla k \ln(\text{queue}_i) = 0$ (то есть рассматривается белый шум, общее название - задача Пуассона, другое слагаемое соответствует задаче Дирихле).

Стохастический процесс обладает Марковским свойством, если условное распределение вероятностей будущих состояний процесса зависит только от нынешнего состояния, а не от последовательности событий, которые предшествовали этому.

Вместо математической формулировки Марковского свойства достаточно ссылки на следующий пример.

Пример 7.3.4. Броуновское n -мерное движение, конечно, является решением стохастического дифференциального уравнения: $dX_t = dB_t$.

Таким образом, производящий оператор A процесса B_t (броуновское n -мерное движение) имеет вид:

$$Af = \frac{1}{2} \Delta f.$$

Для общего случая достаточно правдоподобно выглядит следующая формула балансировки нагрузки:

$$E_i = |d * \nabla f_i(\mathcal{V}_i \setminus \mathcal{V}_i^0)| + k \ln(\text{queue}_i),$$

где d – ширина теневой грани, $\mathcal{V}_i \setminus \mathcal{V}_i^0$ – количество ожидаемых сообщений, а не количество аргументов задачи отсутствующих в кэше исполнителя, поскольку в целях оптимизации необходимо использовать буферизацию сообщений.

Актуальным направлением в программировании является векторная маршрутизация потоков. Кратко остановимся только на буферизации сообщений. На практике нужно рассматривать среднюю длину сообщения в количестве буферов (полезный размер буфера является константой). Очевидно, что буферизация коротких сообщений значительно увеличивает скорость обменов, а для длинных сообщений она не хуже, следовательно, для сообщений произвольной длины также выгодно использовать буферизацию. Поскольку командам синхронизации свойственна высокая латентность коротких сообщений, а буферизация не всегда возможна, то возникает желание отказаться от синхронизации, далее будут рассмотрены недетерминированные сообщения.

Хотя формула балансировки нагрузки выглядит достаточно правдоподобно со точки зрения эксперта программиста, авторы обещали рассмотреть поставленные задачи в рамках априорной неопределенности, а именно СДУ и теории информации.

3.3. Теории информации и модель обмена сообщениями

При любой формализации неизбежна определенная идеализация. Хотя авторы акцентируют внимание на модели обмена сообщениями, эта модель применима и для других параллельных архитектур: для общей памяти и ускорителей, с несущественными дополнениями. Моделирование основано на применении монографии Р. Л. Стратоновича: “Теория информации” [15], который принципиально отказался от использования специальных терминов теории информации с целью обобщения её и термодинамики. Авторы напротив намерены пояснять некоторые формулы в терминах теории информации и СДУ. Р. Л. Стратонович рассматривал кодирование (передачу) информации, авторы рассматривают передачу (обмен) сообщениями, что в принципе одно и то же.

Максимальное значение энтропии называется пропускной способностью (информационной емкостью) канала (вычислителя, параллельной программы) без помех. Рассмотрим задачу максимизации энтропии (производительности) при максимизации пути сигнального графа на каждом процессоре, которая, действительно, напоминает постановку транспортной задачи дробно-линейного программирования: максимизация паросочетаний при максимизации потока.

Понятие “сообщения” без ущерба для теории можно заменить на понятие “случайной величины”, понятие “последовательности сообщений” на “случайный процесс”. Так количество информации в контексте теории вероятности представляется в виде средней энтропии:

$$I = H_{\xi} = - \sum_{\xi} P(\xi) \ln P(\xi),$$

где ξ – дискретная случайная величина, а $P(\xi)$ – её распределение вероятности.

Эта формула является следствием (в асимптотическом смысле) формулы Хартли, для неравновероятных событий, которая в виде случайной энтропии (то есть энтропия – случайная величина) представлена как:

$$H(\xi) = -\ln P(\xi),$$

с условием нормировки: $\sum_{\xi} P(\xi) = 1$.

Естественно, средняя энтропия является средним значением случайных энтропий:

$$H_{\xi} = MH(\xi).$$

Авторы предпочли бы изложение теории информации сразу в терминах СДУ. Так, в случае непрерывной случайной величины можно ли вместо суммы использовать интеграл Ито, ведь известно, что дифференциальная энтропия неограниченна? И решение тоже известно: надо рассматривать нормализованную энтропию, которую в контексте нечетких множеств называют невероятностной энтропией, или: надо рассматривать производную Радона-Никодима.

При определении структуры процессов ранее упоминалось свойство аддитивности энтропии.

Теорема 1.3. Если случайные величины ξ_1, ξ_2 независимы, то полная (совместная) энтропия H_{ξ_1, ξ_2} распадается на сумму энтропий:

$$H_{\xi_1, \xi_2} = H_{\xi_1} + H_{\xi_2}.$$

Теорема 1.4. Энтропия обладает свойством иерархической аддитивности:

$$H_{\xi_1, \dots, \xi_n} = H_{\xi_1} + H_{\xi_2 | \xi_1} + \dots + H_{\xi_n | \xi_1, \dots, \xi_{n-1}},$$

где $H_{\xi_2 | \xi_1}$ – условная энтропия.

Это свойство в практическом плане используется при реализации миксинов. Рассмотрим задачу для которой надо посчитать математическое ожидание Mf_i , где f_i – среднее значение функции на i -процессоре. Исходя из последовательного варианта с целью повышения эффективности можно предложить попарное пирамидальное суммирование, но в этом случае сообщения будут точка-точка, то есть неравновероятными, а канал несимметричным. Возможно для небольшого числа процессоров этот вариант будет более эффективным, чем реализация с учетом свойства иерархической аддитивности: Используя ширококовечание, суммирование надо вести на каждом процессоре, при этом дерево выбора на каждом процессоре будет своё, но условная энтропия на каждом процессоре все равно будет стремиться к нулю. Данная схема распараллеливания близка к схеме мастер-рабочий без синхронизации, для которой каждый процессор является и мастером, и рабочим.

Для реализации дерева выбора каждый процессор как минимум должен иметь два (физических) двунаправленных канала связи, где k – количество каналов процессора аналогично числу букв алфавита (для многоядерной архитектуры k – количество ядер, но дерево выбора направлено в другую сторону, аналогично пирамидальному суммированию). Кроме того, существует не совсем корректное мнение, что результат суммирования зависит от порядка суммирования, в случае энтропийной устойчивости все реализации суммирования будут приблизительно равны.

Закон Амдала иллюстрирует ограничение роста производительности вычислительной системы с увеличением количества вычислителей (процессоров), который по формулировке является хорошо изученным распределением Бернулли. То есть закон Амдала является частным случаем с точки зрения теории информации и не учитывает обмен сообщениями. Благодаря свойству аддитивности энтропии эти две задачи можно рассматривать независимо. Кроме того, для геометрического распараллеливания при фиксированном количестве данных доля последовательных вычислений обратно пропорциональна количеству процессоров, то есть стремится к нулю при количестве процессоров стремящимся к бесконечности, следовательно, при оценке производительности надо учитывать только обмен сообщениями. При рассмотрении общего случая понадобится не только равноправность процессоров, но и равновероятность сообщений (асинхронные недетерминированные сообщения, стационарный процесс по количеству процессоров). Поскольку наибольшее распространение получила кластерная архитектура, в которой обмены осуществляются через общую шину (можно рассматривать и k-дерево), то подобная идеализация оправдана, кроме того каналы являются двусторонними, то можно считать, что длина пути для асинхронных сообщений равна единице, а для синхронизированных обменов она равна двум. Длина пути - это константа, зависящая от типа сообщения и архитектуры вычислителя, на которую надо умножить среднюю длину сообщения.

На самом деле, равновероятность сообщений не обязательна. Можно рассматривать детерминированные сообщения (установившийся процесс), например, решетка данных на решетке процессоров, но это достаточно узкий класс задач. Перспективой расширения класса задач является рассмотрение анизотропных диффузионных процессов и кратных интегралов.

Дать общую формулировку свойства асимптотической эквивалентности неравновероятных возможностей (сообщений) равновероятным помогает понятие энтропийной устойчивости семейства случайных величин.

Семейство случайных величин $\{\eta^n\}$ является энтропийно устойчивым, если отношение $H(\eta^n)/H_{\eta^n}$ при $n \rightarrow \infty$ сходится по вероятности к единице. Это значит, что каковы бы не были $\varepsilon > 0$, $\eta > 0$ найдется такое $N(\varepsilon, \eta)$ что будет выполнено неравенство:

$$P\{|H(\eta^n)/H_{\eta^n} - 1| \geq \varepsilon\} < \eta,$$

при любом $n > N(\varepsilon, \eta)$.

В определении подразумевается, что все $0 < H_{\eta^n} < \infty$ и H_{η^n} не убывает с ростом n . Обычно $H_{\eta^n} \rightarrow \infty$.

Факт асимптотической равновероятности можно сформулировать при помощи понятия энтропийной устойчивости в виде следующей теоремы.

Теорема 1.9. Если семейство случайных величин $\{\eta^n\}$ является энтропийно устойчивым, то множество реализаций каждой случайной величины можно разбить на два подмножества A_n и B_n таким образом, что

1. Суммарная вероятность подмножества A_n исчезает:

$$P(A_n) \rightarrow 0 \text{ при } n \rightarrow \infty;$$

2. Реализации второго подмножества B_n становятся относительно равновероятными в смысле соотношения

$$\left| \frac{\ln P(\eta) - \ln P(\eta')}{\ln P(\eta)} \right| \rightarrow 0 \text{ при } n \rightarrow \infty, \eta \in B_n, \eta' \in B_n;$$

3. Число M_n реализаций множества B_n связано с энтропией H_{η^n} соотношением

$$\ln M_n / H_{\eta^n} \rightarrow 1 \text{ при } n \rightarrow \infty.$$

Приведем некоторые комментарии относительно энтропийной устойчивости. С точки зрения СДУ определение энтропийной устойчивости можно рассматривать как обобщение интеграла Ито через предел по вероятности: Отношение $H(\eta^n)/H_{\eta^n}$ - отно-

сительная энтропия или расстояние Кульбака-Лейблера для равномерного распределения. В этом случае используется обозначение $H_{\xi}^{P/Q}$.

В теории информации основное внимание уделено текущему кодированию, но также рассмотрен и другой подход, который можно назвать блочным. При нем подлежит кодировке конечная совокупность (блок) элементарных сообщений. Если блок является энтропийно устойчивой величиной, то вероятность потери части реализаций сообщений достаточно мала. Как уже отмечалось, вместо кодировки сообщений можно рассматривать передачу сообщений, а также максимальное значение энтропии называется пропускной способностью канала без помех. Рассмотрим задачу максимизации энтропии в случае блочного подхода. В качестве комментария отметим, что эта задача эквивалентна задаче минимизации расстояние Кульбака-Лейблера, также её называют задачей максимального правдоподобия. В нашей интерпретации блок – это процессор, который должен быть энтропийно устойчивым, то есть ПРОЦЕССОР=МАРКОВСКИЙ ПРОЦЕСС, с учётом аппаратной и программной реализаций так же, как и весь вычислитель. Если рассматривать задачу отказоустойчивости для Марковского процесса, то для её реализации достаточно одного дополнительного процессора (средняя вероятность выхода из строя одного процессора очень мала), причем события, связанные с заменой одного процессора на другой, не влияют на производительность вычислений (без учета сохранения и восстановления данных).

Часть задачи, реализуемая на процессоре (блоке), должна быть Марковским процессом.

Приведем некоторые примеры блочных Марковских процессов.

Авторы постоянно ссылаются на геометрическое распараллеливание, поскольку в этом случае интерпретация является достаточно наглядной. Данные расположенные в блоке являются открытым множеством, поскольку есть теневые грани. Суммарная мера на границе множества должна быть меньше меры внутренности множества, по сути это вариант неравенства Чебышева, активно используемом при доказательстве теорем.

В принципе можно рассматривать граф (задач на процессоре) достаточно произвольной структуры. Без умаления общности можно рассмотреть производную Радона-Никодима, введя две меры, а, следовательно, и случайный процесс:

$$E \leq X \rightarrow \frac{1}{X} \leq \frac{1}{E(X)} \ll 1,$$

где X – мера определенная в узлах графа и зависящая от их количества, E – мера, определенная на дугах (границах) графа.

Эти соотношения справедливы и для сигнальных графов, например, количество ближайших соседних вершин. И применимо и в других областях знаний, например, для визуализации графов, в частности антологии знаний. “Для гиперграфа h с заданным набором узлов X , заданным набором ребер E и набором соответствующих значений матрицы инцидентности $\{a\}$ информация $I(h)$ имеет следующий вид [16]”:

$$I(h) = \frac{1}{|X|} \log_{\{a\}} |E|.$$

Хотя матрица инцидентности используется для редукции графов, включая сети Петри, на взгляд авторов, использование определителя матрицы инцидентности в основание логарифма не обосновано.

Распишем задачу минимизации расстояния Кульбака-Лейблера:

$$\begin{aligned} \lim_{X \rightarrow \infty} D_{KL} \left(\frac{1}{X} \middle| \frac{1}{E(X)} \right) &= \sum \lim_{X \rightarrow \infty} \frac{1}{X} \ln \left(\frac{E(X)}{X} \right) \\ &= \sum \lim_{X \rightarrow \infty} \frac{1}{X} \ln(E(X)) - \sum \lim_{X \rightarrow \infty} \frac{1}{X} \ln(X) = \sum (I(h) - 0) \end{aligned}$$

Фактически $I(h)$ – относительная энтропия, а условная энтропия стремится к нулю.

Известным примером является случай, когда дерево выбора ограничено сверху k -деревом, в теории информации получившим название – оптимальное кодирование дешифруемых кодов Крафта.

Теорема 2.3. Можно указать такой способ кодирования (передачи) равнораспределённых независимых сообщений, что

$$l_{\text{cp}} < \frac{H_{\xi}}{\ln D} + 1.$$

В нашей интерпретации l_{cp} – средняя длина пути между узлами вычислителя, D – количество каналов процессора (в оригинале количество букв в алфавите). В случае обмена сообщениями через общую шину можно считать, что $D = \infty$, следовательно, средняя длина пути между узлами вычислителя равна единице, что соответствует экпертному подходу. В случае блочного подхода применима следующая теорема.

Теорема 2.4. Существуют такие способы кодирования бесконечного сообщения, что средняя длина элементарного сообщения может быть сделана сколь угодно близкой к $\frac{H_{\xi}}{\ln D}$.

Подобные оценки для средней длины пути применимы не только для аппаратной архитектуры вычислителя, но и на логическом уровне для графов в пространстве R^D . Полезными ссылками являются: внеядерные алгоритмы (реструктуризация данных по k -дереву) [17]; Полиномиальная приближенная схема для задачи о цикловом покрытии графа фиксированного размера k в евклидовом пространстве произвольной фиксированной размерности (дискретная оптимизация) [18].

Поскольку процессор сам с собой не обменивается сообщениями, можно ввести понятие субэнтропии, например, для равновероятных сообщений, аналогично определению субфакториала. Субфакториал $!n$ определяется как количество беспорядков порядка n , то есть перестановок n -элементного множества без неподвижных точек. В случае асимптотической сходимости равновероятность сообщений не обязательна, как и рассмотрение субэнтропии. В работе [19] показано, что в классической теории информации субэнтропия двойственная субэнтропии фон Неймана, определенная через перестановки попарной разности собственных значений, является точной нижней границей пропускной способности канала и её вычисление соответствует задаче минимизации расстояния Кульбака-Лейблера.

В зависимости от задачи длина сообщения между разными процессорами может быть не одинаковой. В этом случае основной характеристикой является средняя длина сообщения. Рассмотрим прямой способ вычисления максимальной энтропии для этого примера, что соответствует параграфу 3.1. Пусть имеются m процессоров V_1, \dots, V_m , которые передают сообщения соответственно длиной $l(1), \dots, l(m)$. Суммарная длина сообщения будет $L = l(1) + \dots + l(m) = l_{\text{cp}} m$. Зафиксируем эту длину и будем подсчитывать число $M(L)$ различных реализаций данной длины. Максимальная информация достигается, когда все из $M(L)$ возможностей равновероятны. При этом

$$\frac{H_L}{L} = \frac{\ln M(L)}{L}.$$

Взяв предел при $L \rightarrow \infty$ получим энтропию, рассчитанную на единицу длины. То есть достаточно рассмотреть решение в случае асимптотической сходимости линейного однородного уравнения, которое имеет вид: $M(L) = C e^{\lambda L}$. Решение имеет единственный корень с максимальной действительной частью λ_m . Итак, число различных реализаций длины L имеет вид:

$$M(L) \approx C_m e^{\lambda_m L}.$$

Тот же самый результат можно получить из решения первой вариационной задачи. Ограничимся только рассмотрением дискретного канала без помех (общей постановкой задачи). Система $[Y, c(y), \alpha]$ полностью характеризует дискретный канал без помех, $y \in Y$, функция штрафов $-c(y) \leq \alpha$ (микрочаноническое распределение). В частности $c(y) \leq L$. Удобнее рассматривать каноническое распределение: $\sum c(y)P(y) \leq \alpha$. Пропускную способность C или информационную ёмкость канала $[Y, c(y), \alpha]$ определим, как максимальное значение энтропии $C = \sup_{P(y)} H_y$. Таким образом пропускная способность канала определена как решение вариационной задачи. Отметим, что она эквива-

лентна задаче минимизации рисков (временных задержек). Эквивалентность микроканонического и канонического распределений доказана. В контексте данной работы закон Амдала – это микроканоническое распределение, а соответствующим ему каноническим распределением является распределение Бернулли. Также подчеркнём, что рассматривается задача максимизации производительности (энтропии) параллельных вычислений.

В качестве иллюстрации важности средней длины сообщения приведем пример поведения производительности параллельной программы при фиксированном количестве данных и переменном количестве процессоров. При росте количества процессоров средняя длина сообщения в количестве буферов может уменьшиться на единицу, вследствие экспоненциальной зависимости от длины сообщения, должен наблюдаться скачок производительности. Таким образом, длина сообщения может зависеть так же как и доля последовательных вычислений от количества процессоров (переменного параметра по сути аналогичного времени). Следовательно, надо рассматривать решение линейного неоднородного уравнения, а для начала выписать функцию штрафов.

3.4. Оптимальная масштабируемость параллельных вычислений (блочный подход)

В контексте высокопроизводительных вычислений существует два показателя масштабируемости:

1. Сильная масштабируемость — показывает, как меняется время решения задачи с увеличением количества процессоров (или вычислительных узлов) при неизменном общем объёме задачи;
2. Слабая масштабируемость — показывает, как меняется время решения задачи с увеличением количества процессоров (узлов) при неизменном объёме задачи для одного процессора (или узла).

Оптимальная масштабируемость (оценка производительности параллельных вычислений, учитывающая как количество данных, так и количество процессоров) – это неоднородная задача Марковского управления, где правая часть уравнения равна количеству блоков (процессоров) в блочной модели. Конечно, это очередная правдоподобная гипотеза.

Под оптимальной масштабируемостью будем понимать случай, когда пропускная способность канала определена как решение вариационной задачи, то есть энтропию можно выписать в явном виде. Для начала надо определить функцию штрафов (для дискретного канала), зависящую от двух параметров количества данных и количества процессоров – p :

$$c(N, p) = s \sum_{i=1}^p l_i(N, p) + \sum_{i=1}^p \alpha_i(N, p),$$

где α_i – доля последовательных вычислений итого процессора, s – средняя длина пути (напомним, что для равновероятных сообщений она равна единице). Кроме того, ранее предполагалось использование сигнальных графов, как развитие идей вычислительной сложности, то есть $l_i(N, p) = \nabla f_i(N, p)$, что приведет к рассмотрению второй вариационной задачи.

Далее постараемся найти подобную функцию штрафов в теории информации. Прочитируем параграф 3.5 – метод потенциалов в случае большого числа параметров: “Функция штрафов зависит от числового параметра и является дифференцируемой по этому параметру.” В нашем случае по количеству процессоров (блоков) – p . Авторы не видят большого смысла переписывания известных формул в других обозначениях. Всё же приведем одно определение. Функция

$$B(N) = - \frac{\partial c(N, p)}{\partial p}$$

называется случайным внутренним (эндогенным) термодинамическим параметром, сопряженным с внешним (экзогенным) параметром p . Далее в том же параграфе рас-

смотрен пример с двумя случайными переменными, в нашем случае это длина сообщения и доля последовательных вычислений, которые находятся из решения системы двух уравнений с двумя неизвестными. Таким образом, аналитическое решение рассматриваемой задачи известно.

Аналитическое решение может, например, использоваться для определения помех в вычислителе. Следовательно, возникает потребность в разработке адекватных моделей обработки статистических данных производительности параллельных вычислений в том числе и для следующей задачи. До сих пор рассматривались только стационарные процессы. Используя миксины, в ДС для графа задач можно реализовать стационарный процесс с некоторыми накладными расходами относительно доли последовательных вычислений (очевидно, что должно выполняться неравенство Чебышева, чем больше данных в блоке, тем меньше доля накладных расходов, вероятно, надо рассматривать канал с помехами (параграф 7)) и чтения сообщений (напомним, что размер очереди сообщений стремится к бесконечности, что соответствует белому шуму). В случае геометрического распараллеливания можно реализовать и сравнить стационарный процесс и установившийся процесс (сообщения точка-точка). Аналитическое решение для последнего в теории информации тоже известно (хотя термин - установившийся процесс не используется и параллельные вычисления не рассматривались). Для этого вводится понятие информации связи (в нашем случае дуги графа), естественно, через условную вероятность, через неё же определяется симметричный канал, естественно, применяя перестановки, соответственно параграфы 6 и 8. При этом условное распределение на выходе канала при фиксированном входном сигнале предполагается известным (фактически строится передаточная функция).

Пропускной способностью канала $[P(y|x), c(x)]$ называется максимальное значение информации связи между входом и выходом:

$$C = C[P(y|x), c(x)] = \sup_{P(x)} I_{x,i,j}.$$

В случае полностью симметричного канала (решетка процессоров) формула выглядит достаточно простой (8.4.9):

$$C = \ln M^{1/p_j} - M \ln^{1/p_j}.$$

В случае равномерного распределения формула имеет следующий вид

$$C = \ln M \frac{Q(dy)}{P(dy)} - M \ln \frac{Q(dy)}{P(dy)},$$

где $Q(dy)$ есть вспомогательная мера, на основании которой производится перестановка интервалов.

Но эта формула выполняется разве что на ядре (в активной зоне) или в случае общей памяти, поскольку в рассмотренных случаях [14] функция штрафов зависит от одной переменной. Для геометрического распараллеливания можно сократить зависимости до одной переменной, учитывая, что при фиксированном количестве данных доля последовательных вычислений обратно пропорциональна количеству процессоров, то есть стремится к нулю (но не ноль) при количестве процессоров стремящимся к бесконечности. Чем больше данных на процессоре, тем сбалансированность вычислений лучше, что ведёт к определённому противоречию.

В практическом плане более важным является соблюдение определённых правил программирования при разработке параллельных программ, начиная с бинарного гомоморфного отображения и заканчивая установившимся процессом. Если график сильной масштабируемости соответствует логарифмической функции, то параллельную программу можно считать установившимся процессом по количеству блоков (если строго, то надо рассматривать линейную задачу фильтрации [4]: системы с шумом и измерений с шумом). С целью повышения производительности можно рассмотреть следующую задачу [6] экстраполяции (прогноза [4]) до момента остановки: $N(p-1)$ известно, надо найти $N(p)$ при максимизации энтропии, которую авторы как раз и называют неоднородной задачей Марковского управления. Поскольку общее решение

неоднородного уравнения является суммой фундаментальной системы решений и частного решения, рассмотрение неоднородной задачи автоматически ведет к увеличению энтропии и понижает размерность системы однородных уравнений (в нашем случае до одного). Неоднородную задачу можно получить из случайной замены времени (в [6] рассматривалась функция энтропии $p/N(p)$ или дробная линеаризации на ядре) $L = l_{cp}p$, дифференцируя по средней длине сообщения. На границе должно выполняться условие, соответствующие линейному ускорению (максимизации энтропии) $\dot{X} = p$.

Поскольку обрабатываются статистические данные, надо перейти к Марковским цепям (решению линейной системы алгебраических уравнений), что и будет является наилучшей аппроксимацией задачи максимизации эффективности параллельных вычислений модели обмена сообщениями: $H_p X = p$, где H_p – матрица обмена сообщениями (матрица смежности), на главной диагонали указано среднее время последовательных вычислений блока, а другие элементы соответствуют среднему времени передачи сообщений между процессорами, предполагается, что схема распараллеливания не меняется (не зависит от номера итерации).

Матрица обмена сообщениями будет близка к симметричной, поскольку ответное сообщение той же длины предположительно должно занимать немного больше времени. В таких случаях рассматривается задача Дирихле со стохастическим управлением вида регулятор с неполной информацией о состоянии системы – назовём её задача стохастического управления информацией. Но зато у этой матрицы будет другая симметрия, связанная со сбалансированными вычислениями, схемой распараллеливания и определением симметричного канала.

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 i -строка и i -столбец (i – номер процессора, на котором расположен мастер). Для конвейера или линейки процессоров получается трёхдиагональная матрица, для решетки пятидиагональная и т.д.

Подробнее рассмотрим схему мастер-рабочий. Матрица смежности имеет следующий вид:

$$H_p = \left| \begin{array}{cccc|c} a & b & \dots & b & 1 \\ b & a & & 0 & 2 \\ \vdots & & \ddots & & \vdots \\ b & 0 & \dots & a & p \end{array} \right.$$

Система не совместна, только когда $a_{1,1} = 0$ - время без учёта подготовки и сохранения данных, то есть мастер не занимается счётом. Из последнего уравнения легко найти x_p

$$dx \approx x_p = \frac{p-bx_1}{a}.$$

В пространстве L_2 приращение энтропии (коэффициент сдвига в диффузионном процессе)

$$dx = \overline{X_p} - \overline{X_{p-1}} \approx x_p.$$

С учетом вычислительной сложности алгоритма $a = f(N)$ (коэффициент диффузии), а $b = \nabla f(N)$, например, $f(N) = N^3$ и балансировки загрузки по количеству процессоров приращение энтропии имеет вид:

$$dx = \left(\frac{p}{N}\right)^3 p - O\left(\frac{1}{N}\right) \rightarrow dx \approx \frac{p^4}{N^3}.$$

Получается, что чем сложнее алгоритм, тем он лучше распараллеливается (масштабируется). Если положить $\frac{p^4}{N^3} = 1$ (максимальное значение энтропии равно единице), то получим выражение N через p . Конечно, приведённая оценка является грубой, но можно рассмотреть параметризованную модель белом шумного случайного процесса, поскольку функция энтропии всегда $p/N(p) \ll 1$ (подробнее в разделе про визуализа-

цию). Эта функция является гармонической функцией (свойство среднего), следовательно, рассмотрение задачи о движущейся границе является обоснованным, которая СДУ называется задачей Якоби [4].

В качестве того, что функция штрафов выписана правильно, можно привести сравнение графиков бета-функции распределения Дирихле и эффективности параллельных вычислений для задачи решения СЛАУ методом Холецкого [20] в случае сбалансированных вычислений, Рис. 2.

Пусть $X = (X_1, \dots, X_K) \sim Dir(a)$ – распределение Дирихле и $a_0 = \sum_{i=1}^K a_i$, тогда $E[X_{i|a}] = \frac{a_i}{a_0}$.

$$B(x, y) = \int_0^1 t^{x-1} (1-t)^{y-1} dt$$

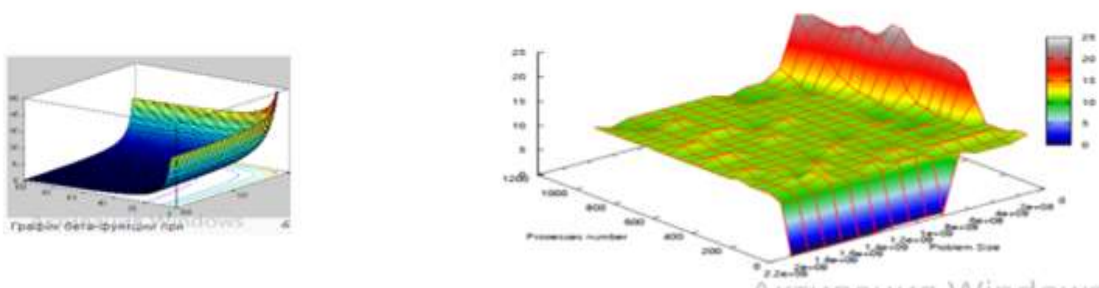


Рис. 2. Слева – Бета-функция, справа – эффективность параллельных вычислений

С точки зрения визуальной верификации графики похожи.

4. Примеры применения блочного подхода в других областях знаний

Блочный подход в СДУ применяется достаточно часто, например, в глобальной модели сейсмичности земли [21], но количество блоков является фиксированным (например, количество литосферных плит). К сожалению, авторам не удалось найти примеров, когда количество блоков является переменным параметром. А эта задача является актуальной, не только для задачи прогноза производительности параллельных вычислений, но и для других задач экстраполяции (в общем случае задач стохастического управления информацией), которые приведены ниже.

Рассмотрим аналогичный пример из области экономики – анализ данных Счётной Палаты. Сгенерирована матрица поставок по (автомобильным) номерам регионов (Рис. 3) и определена гармоническая функция – отношение суммы поставок в регионе к количеству фирм в регионе, её значение между регионами отображены в виде сфер в соответствующих элементах матрицы. Видно, что перестановкой строк из этой матрицы можно получить матрицу близкую к симметричной с учетом того, что заказывают всегда больше, чем надо (определена точная верхняя граница). Хотя значение регулятора является важным в экономике, оно связано и с нормой прибыли и уровнем коррупции, рассмотрим только задачу сокращается ли информационный разрыв между Москвой и другими регионами. Определим меру информационный разрыв, как отношение многомерного расстояния к географическому расстоянию, которое является вспомогательной мерой в производной Радона-Никодима). Если рассматривать отношение информационных разрывов в задаче экстраполяции, то вспомогательная мера сократится. Если мера отношения будет возрастать, то информационный разрыв будет сокращаться и наоборот. А теперь рассмотрим эту же задачу, когда появился новый регион. Она

уже была рассмотрена – количество блоков (регионов) является переменным параметром. Можно рассматривать и типовые задачи кластеризации, но с точки зрения диссипативных систем – образования новых кластеров. Эта же модель применима для анализа распространения информации в интернет сетях, где в качестве параметра выступает не количество регионов, а количество информационных каналов (блогов и т.п.).

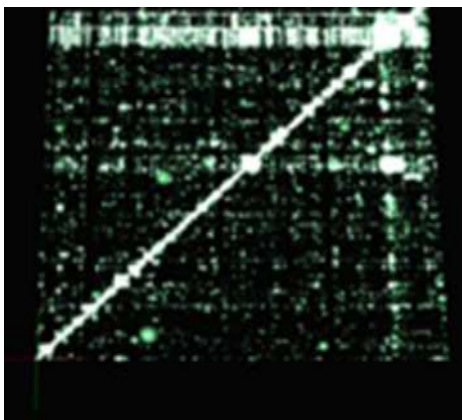


Рис. 3. Матрица поставок по регионам

Далее рассмотрим некоторые примеры задач стохастического управления визуальной информацией, сосредоточив основное внимание на применении блочного подхода в визуализации.

5. Стохастическое управление визуальной информацией

Синкхорновские (Sinkhorn) нейронные сети, в основе которых лежит теорема Синкхорна [22], используются для решения широкого класса транспортных задач; суперразрешение, сравнение двух дистрибутивов. Утверждается, что эти сети лучше по быстродействию и числу параметров, чем генеративные сети максимального правдоподобия. Далее будет рассмотрена возможность применения СДУ для задач на (сигнальных) графах как альтернатива традиционному применению нейронных сетей. Для параллельных вычислений функция, определенная на сигнальных графах, зависела от количества данных в этом параграфе зависимость от высоты (карта высот). Сначала будет рассмотрена задача визуализации цифровой модели поверхности (ЦМП) с фиксированным количеством блоков и переменным количеством данных в каждом блоке. Для этой задачи максимальное количество блоков зависит от количества данных, которое ограничено памятью видеокарты, то есть является константой. В продолжение будет рассмотрена задача распознавания жестов младенцев на одном блоке как сравнение двух дистрибутивов (видеопотоков) здорового младенца и возможно с отклонениями в будущем. В принципе для задач распознавания количество блоков может быть переменным параметром, а не константой. В перспективе ставится цель решения задачи детекции объектов как композиция этих двух задач (этих двух случайных процессов) с разными видами неоднородности: количество блоков и высота.

5.1. Визуализация сеток как параметризованная модель белого шумного случайного процесса

Блок – это элемент хранения данных ЦМП (аналогично для задачи распознавания), матрица размером $N \times N$, в каждой точке которой определена функция высоты над уровнем моря и константный цвет, соответствующий классу объекта, то есть задан сигнальный граф на решетке. ЦМП представляется набором таких матриц или блочной матрицей. При этом блок имеет иерархическую структуру – квадродерево, которая от-

ражает разные уровни детализации по точности. Основной спецификацией визуализации является её применение в VR (трехмерная графика), то есть в банаховом пространстве (отображаются не точки, а интервалы), из-за этого между блоками образуются “стыки” или “дырки”, которые вызывают трудности при реализации алгоритмов рендеринга. Они должны особым образом обрабатывать границы блоков. В Гильбертовом пространстве (для растровой графики) таких проблем нет. Основное отличие трехмерного графического подхода от стандартного, применяемого в СДУ заключается в том, что предел ступенчатой функции должен быть определен не в топологии поточечной сходимости, а в компактно-открытой топологии, что сделано с целью построения непрерывного отображения (визуализации) с точки зрения зрительного восприятия.

При облёте ЦМП подгружаются блоки с учётом функции минимума расстояния между камерой и блоком сначала с наименьшей точностью, в дальнейшем с улучшением точности. Приложение визуализации выполнено средствами Web-GL, при этом реализовано абстрагирование шейдеров, которое аналогично абстрагированию параметров, то есть шейдер – это функция, зависящая, например, от положения камеры. Фактически реализованы реактивные вычисления на уровне видеокарты. Конечно данное направление интересно, но как уже отмечалось рассматриваться не будет.

Приведем виды отображения, реализованные на блоке. Самое простое – это облако точек, которое рассматриваться не будет, поскольку не является непрерывным отображением с точки зрения зрительного восприятия, которое в преобразование Лапласа соответствует “оригиналу”. На Рис. 4. для сравнения приведены два малоразличимых между собой вида отображения, но с разной моделью визуализации. Слева стандартная полигональная графика – барицентрические координаты. Справа визуализация столбиками. Столбик – это метафора. Этот вид отображения иногда называют статистической призмаграммой (четырёхугольная призма) - трехмерный аналог диаграммы. С точки зрения математика – это обратное преобразование для “изображения” в преобразование Лапласа, что и требуется показать, для начала по построению. (Визуализацию часто рассматривают как решение обратной задачи, скорее надо говорить о решении сопряженной задачи).

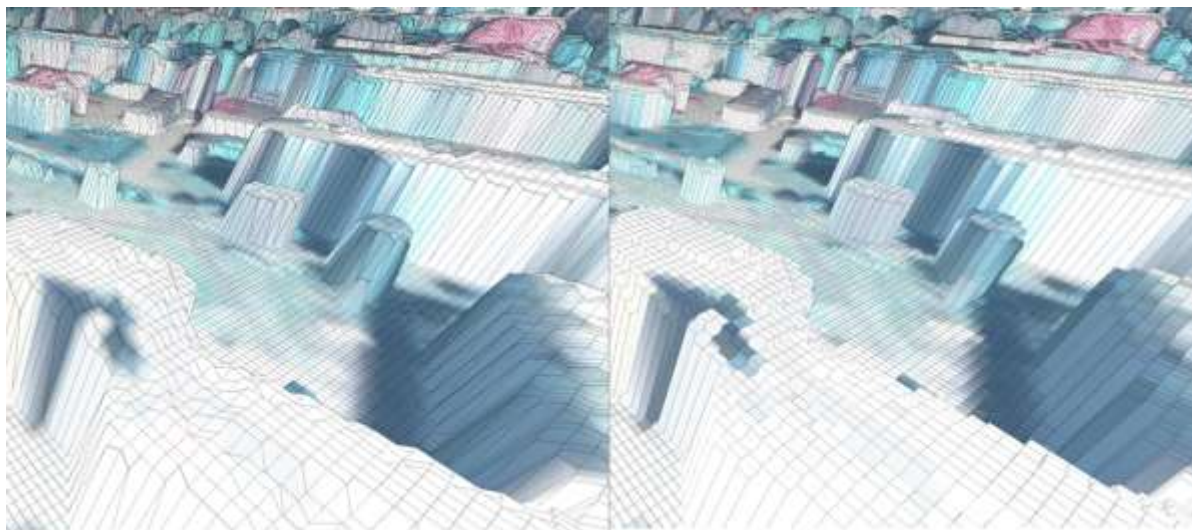


Рис. 4. Задачи визуализации цифровой модели поверхности, слева поверхность изображена треугольниками, справа – столбиками

На Рис. 5. показано как строится статистическая диаграмма для двумерного случая. Формальное описание будет приведено ниже, пока что ограничимся декларативным с использованием когнитивной визуализации. Исторически термин когнитивная визуализации происходит из решения математических задач графическим способом [23]. Статистическая диаграмма - это направленная ступенчатая функция, функция высоты,

математическое ожидание значения которой соответствует середина отрезка, интервала (показано красной точкой), что соответствует интегралу Стратоновича. В трехмерном пространстве двумерный интервал с учетом детализации и направления нормалей принято называть микрогранью. Например, в компьютерной визуализации микрогрань используется для отображения шероховатой поверхности. Очевидна связь микрогрань с конусом нормалей. С учётом уровня детализации (точности) можно рассмотреть кратный интеграл Стратоновича по пространственным переменным (стационарный процесс), в пределе равный двойному. В дополнение на виде отображения, так же прорисовываются “ступеньки” – частные производные (показано красным отрезком), что сделано с целью непрерывности отображением с точки зрения зрительного восприятия. Стоит подчеркнуть, что для проекции на плоскость двойной интеграл Ито нарисовать столбиками не получится, в отличие от проекции на сферу или цилиндр, применение которых планируется при реализации волнового уравнения рендеринга как диффузионного процесса. Известно, что для одностороннего преобразования дисперсия стремится к бесконечности. Конечно, можно было бы отображать значение высоты в квадрате, но такой рисунок не имеет смысла. Кроме того, перспективная проекция и аффинные преобразования изображения являются линейными, поэтому не влияют на дисперсию (например, для дисперсии Алона), что важно при непосредственно взаимодействием с ЦМП.

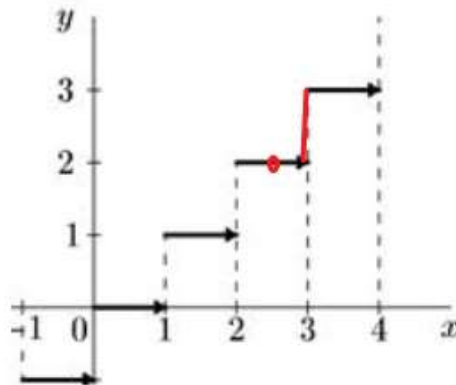


Рис. 5. Статистическая диаграмма - ступенчатая функция.

Стоит отметить, что в качестве базового вида отображения выбрана визуализация столбиками, потому что, во-первых, в ЦМП много вертикальных линий – ступенек, во-вторых, меньше площадь дырок и они расположены вертикально, а не в горизонтальной плоскости как в случае с полигональной графикой, Рис. 6.

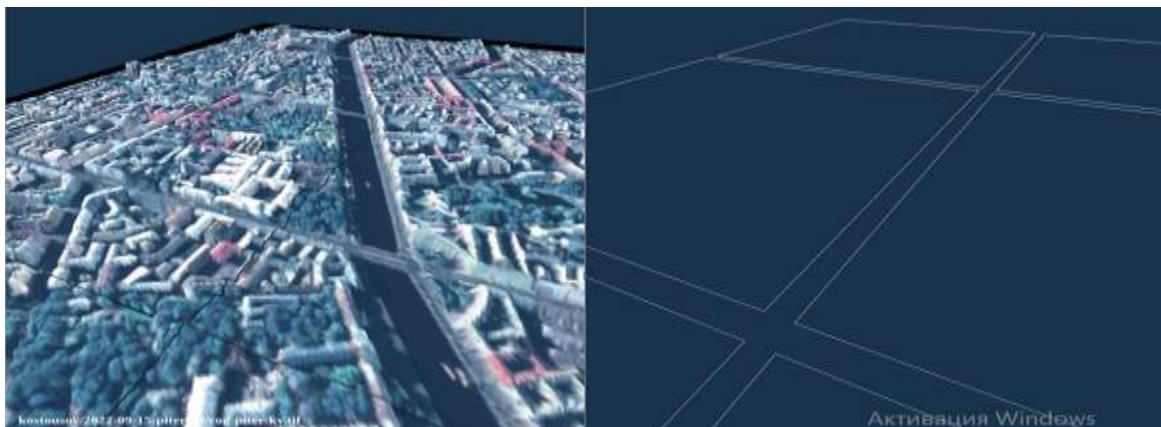


Рис. 6. Слева видны артефакты (синие линии, цвета фона на границах блоков), справа показано разбиение на блоки.

Возникает вопрос можно ли убрать артефакты в изображение, например, рассмотрев транспортную задачу на границе блоков. В случае полигональной графики ответ очевиден: можно ввести фиктивную линию (теневую грань) на границе, и взять среднее значение высоты в точке (производную). Конечно, здесь есть определённые трудности в смысле программирования, на которых останавливаться не будем. Но этот подход в случае визуализации столбиками не работает, Рис. 7. приведен для сравнения артефактов эти двух видов отображений.

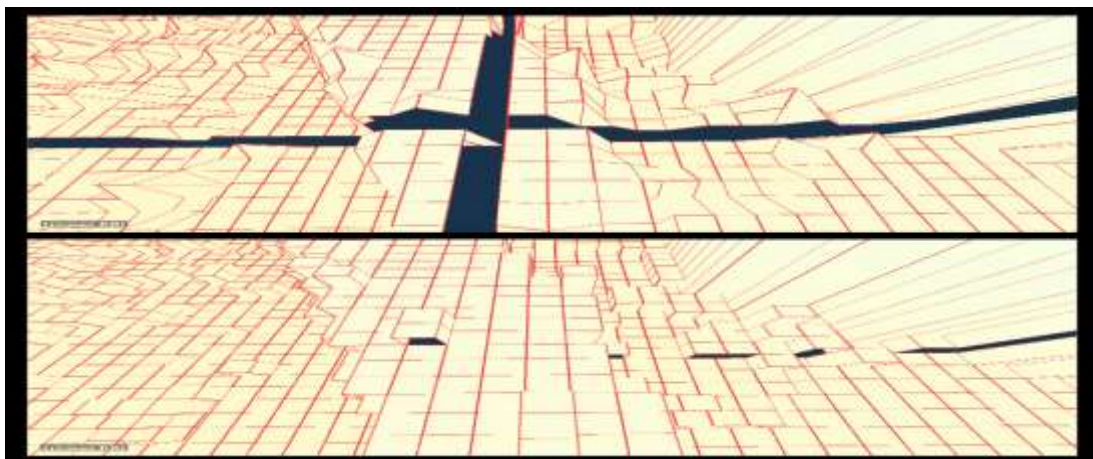


Рис. 7. Вверху артефакты в случае полигональной графики, внизу в случае визуализации столбиками

Чем же являются дырки в случае визуализации столбиками? Это интегральная метрика - метрика Вассерштейна для частных производных:

$$\sum_{v_i \in \bar{V}_j} \|\nabla f(v_i)\|_2^2,$$

где v_i – узел графа, $f(v_i)$ – высота, \bar{V}_j – внутренняя граница блоков.

Интегральная метрика является более информативной, чем векторное поле. Вряд ли математики рассматривали задачу убирания дырок, наверное, здесь можно поколдовать с распределением Дирихле или со стохастическим управлением, но подобное решение всё равно приведет к обменов между блоками. Вероятно, можно обойтись без обменов, приблизительно доопределив частную производную на границе симметрично вниз с предыдущих ячеек, поскольку для неравенства Маркова, являющемся частным случаем неравенства Чебышева, выполняется правило трех сигм.

Этот же подход визуализации ЦМП применим и для трехмерных сеток, когда вместо (графического) фильтра проекция интерактивно применяется фильтр сечения плоскостью (или сечения сферой для волнового уравнения рендеринга), как следствие возникает необходимость определения формулы Ито для фильтров. Так же сетка не обязательно должна быть регулярной, её всегда можно реструктурировать по окто-дереву. В многомерном случае используется матрица рассеивания, которая напрямую связана с определением полностью заданного случайного процесса (см. Рис. 8.), где вместо “оригинала” (облака точек) используется визуализация столбиками.

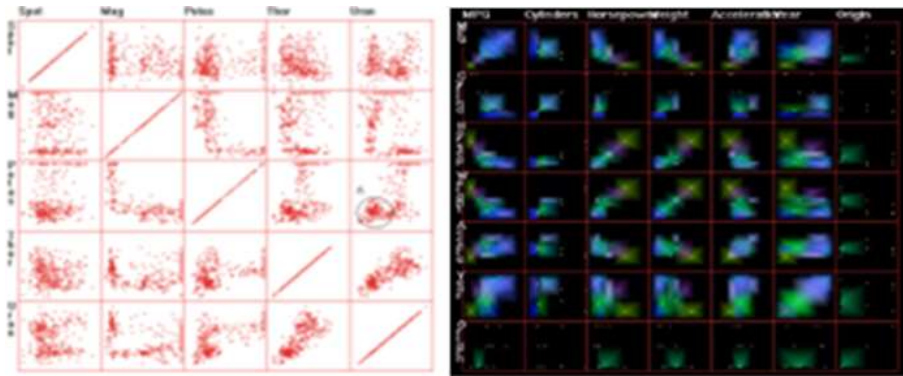


Рис. 8. Матрица рассеивания, слева для облака точек [24], справа для параллельных координат [25], которые можно рассматривать как полный дифференциал [26].

На этом декларативное описание задачи закончим, формализация в основном будет базироваться на монографии [27] Д. Ф. Кузнецова: “Некоторые вопросы теории численного решения стохастических дифференциальных уравнений Ито”.

5.2. Формализация задачи визуализации сеток

Начнем с обобщение интеграла Ито. Чтобы перейти к интегралу предел суммы определяется особым способом с использованием ступенчатой функции [4]:

$$\sum_i f(t_i^*, \omega) (B_{i+1} - B_i)(\omega)$$

Интеграл Ито: $t_i^* = t_i$ - левый конец отрезка. Обозначается:

$$\int_S^T f(t, \omega) dB_t(\omega).$$

Функция $f(t, \omega)$ является измеримой, согласованной и:

$$E\left(\int_S^T f(t, \omega)^2 dt\right) < \infty$$

Важное свойство интеграла Ито состоит в том, что он является мартингалом.

Интеграл Стратоновича: $t_i^* = (t_{i+1} - t_i)/2$ - середина отрезка.

Можно определить **обобщение интеграла Ито** через предел по вероятности:

$$\int_0^t f(s, \omega) dB_s(\omega) = \lim_{n \rightarrow \infty} \int_0^t f_n(s, \omega) dB_s(\omega),$$

где $f, f_n \in W_H$, f_n - ступенчатые функции: $\int_0^t |f_n - f|^2 ds \rightarrow 0$ по вероятности (по мере относительно P).

Очевидно, что определение энтропийной устойчивости так же является обобщением интеграла Ито через предел по вероятности. Аналогичное определение используется в [26]:

Определение 1.2 Последовательность случайных величин $\xi_k(\omega)$ называется сходящейся с вероятностью единица или почти наверное к случайной величине $\xi(\omega)$: $\xi_k \xrightarrow{п.н.} \xi$ при $k \rightarrow \infty$, если:

$$P\{\omega: \xi_k \rightarrow \xi \text{ при } k \rightarrow \infty\} = 1.$$

Эта последовательность также называется фундаментальной сходящейся с вероятностью единица. Фундаментальность последовательности случайных величин является необходимым и достаточным условием существования её предела, что называется критерием Коши.

Рассмотрим отличие стандартного определения полностью заданного случайного процесса от определения в банаховом пространстве.

Случайный процесс считается полностью заданным, если заданы его конечномерные распределения – набор функций распределения, которые определены для любого $k \geq 1$ соотношениями:

$$F_{\xi}(x_1, \dots, x_k, t_1, \dots, t_k) = P\{\cap_{j=1}^k \{\xi(t_j, \omega) < x_j\}\},$$

где $x_j \in R^j$.

Справедливо и обратное утверждение, установленное Колмогоровым:

Если функции $F_{\xi}(x_1, \dots, x_k, t_1, \dots, t_k)$ при всех $k \geq 1$ удовлетворяют условиям:

1. $F_{\xi}(x_1, \dots, x_k, t_1, \dots, t_k)$ - является совместной функцией распределения k случайных величин;
2. справедливо тождественное равенство:

$$F_{\xi}(x_1, \dots, x_k, t_1, \dots, t_k) \equiv F_{\xi}(x_{i_1}, \dots, x_{i_k}, t_{i_1}, \dots, t_{i_k})$$

для любой перестановки i_1, \dots, i_k чисел $1, \dots, k$;

3. $\lim_{x_k \rightarrow +\infty} F_{\xi}(x_1, \dots, x_k, t_1, \dots, t_k) = F_{\xi}(x_1, \dots, x_{k-1}, t_1, \dots, t_{k-1})$,

тогда существует такой случайный процесс $\xi(t, \omega)$, совместными функциями распределения которого являются функция:

$$F_{\xi}(x_1, \dots, x_k, t_1, \dots, t_k).$$

Таким образом, совокупность совместных функций распределения значений случайного процесса $\xi(t, \omega)$ является его исчерпывающей характеристикой.

Если функции распределения $F_{\xi}(x_1, \dots, x_k, t_1, \dots, t_k)$ имеют конечную смешанную k -производную, то существуют совместные плотности распределения значений случайного процесса ξ_t в соответствующие моменты времени:

$$p_{\xi}(x_1, \dots, x_k, t_1, \dots, t_k) = \frac{\partial^k}{\partial x_1 \dots \partial x_k} F_{\xi}(x_1, \dots, x_k, t_1, \dots, t_k).$$

В банаховом пространстве матрица частных производных, как и ковариационная матрица, является несимметричной (производная слева не равна производной справа). Для симметричной матрицы количество перестановок равно $k!$, а для несимметричной - 2^k . В общем случае надо применять теорему Синкхорна [21]. Гильбертово пространство является банаховым, следовательно, можно рассматривать симметричные матрицы в пределе. В этом случае как раз и возникают кратные интегралы. Очевидно, что детализация на одном блоке задает сжимающее отображение, следовательно, можно применять теорему Банаха о неподвижной точке. Стандартно, для численного решения задач применяются Гауссовы процессы, центрирование фундаментальной последовательности и преобразование Лапласа.

Случайный процесс называется гауссовским, если все его совместные плотности распределения являются гауссовскими:

$$p_{\xi}(x_1, \dots, x_k, t_1, \dots, t_k) = \frac{1}{2\pi^{\frac{k}{2}} |K|^{\frac{1}{2}}} \exp\left(-\frac{(x-m)^T K^{-1} (x-m)}{2}\right),$$

где $x = (x_1, \dots, x_k)^T$, $m = (m_1, \dots, m_k)^T$, $m_i = M\{m_i\}$, ковариационная матрица $K = K^T > 0$.

Процесс $\overset{\circ}{\xi}_t = \xi_t - M\{\xi_t\}$ называется центрированной составляющей процесса ξ_t .

Функция: $R_{\xi}(t_1, t_2) = M\left\{\overset{\circ}{\xi}_{t_1} \overset{\circ}{\xi}_{t_2}\right\}$ называется корреляционной функцией процесса ξ_t ,

причем:

$$D\{\xi_t\} = R_{\xi}(t, t),$$

где $D\{\xi_t\}$ - дисперсия случайного процесса ξ_t

Для того, чтобы функция $R_{\xi}(t)$ при $t \in (-\infty, +\infty)$ была корреляционной функцией стационарного в широком смысле случайного процесса ξ_t , удовлетворяющего условию:

$M\{(\xi_{t+\tau} - \xi_t)^2\} \rightarrow 0$ при $\tau \rightarrow 0$, необходимо и достаточно, чтобы она допускала представление:

$$F_{\xi}(\omega) = \iint_{-\infty}^{\infty} e^{it\omega} dF_{\xi}(\omega),$$

где $F_{\xi}(\omega)$ - произвольная неотрицательная ограниченная монотонно неубывающая функция, непрерывная слева.

$F_{\xi}(\omega)$ называется спектральной функцией, если она абсолютно непрерывна и

$$F_{\xi}(\omega) = \iint_{-\infty}^{\infty} S(u) du,$$

где $S(u)$ спектральная плотность процесса ξ_t .

Очевидно, что частным случаем спектральной функции (например, RGB) является преобразование Фурье. Преобразование Фурье со сдвигом (точное значение не совпадает с его магматическим ожиданием) принято называть преобразованием Лапласа. Таким образом визуализация столбиками есть обратное преобразование Лапласа (сопряженный вид отображения с преобразованием Лапласа) в банаховом пространстве.

Очевидно, что задачу визуализации ЦМП можно свести к параметризованной модели белошумного случайного процесса [27] параграф 1.3, положив $\frac{p}{N} = \mu \rightarrow 0$, где количество блоков $p = const$, μ – малый параметр. Анализ динамики интерактивной визуализации при случайных внешних воздействиях пользователя сводится к исследованию вероятностных и статистических свойств решений систем дифференциальных уравнений, возмущённых случайными процессами (стохастическому управлению). Система дифференциальных уравнений для параметризованной модели белошумного случайного процесса имеет вид:

$$\dot{x}_t = a(x_t, u(t), t) + \sum(x_t, t) \frac{1}{\mu} \xi_{\frac{1}{\mu^2}}; x_0 = x(0).$$

Задача визуализации ЦМП является стационарной линейной (подробнее [27] параграф 1.3) и потому не очень интересная с точки зрения математики.

Определённую ценность имеет и разработка метафор визуализации и взаимодействия сопряженных с математической моделью. Визуализация столбиком не такая уж элементарная метафора с учётом блочного подхода. Кроме того, она порождает другую метафору – интегральную метрику для частных производных как альтернатива векторному полю.

6. Обсуждение перспектив развития решения задач стохастического управления визуальной информацией

В рассматриваемых примерах возмущение в параметризованной модели белошумного случайного процесса связано с количеством данных: вычислительной точностью, длиной очереди задач, длиной сообщений. С точки зрения автоматического управления задача визуализации ЦМП является стационарной линейной и потому не очень интересная с точки зрения математики, но сам блочный подход является перспективным, потому что есть более сложные задачи, например, разработка тренажеров с обратной связью, задача детекции на карте высот, рассмотрение которой начнем с задачи распознавания жестов младенцев.

6.1 Задача распознавания жестов младенцев

Существует значительное количество работ по применению нейронных сетей для распознавания поведенческих шаблонов во временных рядах, в том числе и для жестов, но их использование не даёт гарантии, что поставленная задача решена правильно, особенно в случае, когда валидация имеет субъективный характер. С другой стороны, идея совмещения искусственного и математического интеллектов заманчива. Кроме уже упомянутой модели шумоподавления, стоит отметить новый тренд решения ДУ, особенно обратных задач: нейронные операторы Фурье и Колмогорова-Арнольда. Фактически будет рассматриваться задача Коши, то есть при каких условиях задача распознавание жестов младенцев имеет решение.

Многолиственный скелет определяется как многолистная фигура (плоская фигура с самопересечениями) [28]. Будет рассматриваться нетривиальная задача, когда многолистные скелеты младенцев анатомически подобны.

Срединная ось плоской фигуры представляет собой множество внутренних точек фигуры, каждая из которых имеет, по меньшей мере, две ближайшие граничные точки.

Известны решения задачи коммивояжера для графов Делоне (Евклидово минимальное остовное дерево [18]), также графы Делоне (диаграммы Вороного) применяются для распознавания жестов (считается, что расстояние между узлами графа (многолиственного скелета) не меняется). Для решения поставленной задачи важным предположением является, то что узел графа (сустава) имеет площадь (задача с неопределенностью). В этом случае срединная ось получена как объединение прямых и параболических отрезков плоской фигуры, например, для сустава локтя, схематично представлена на Рис. 9.

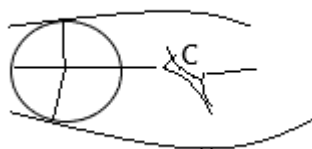


Рис. 9. Срединная ось сустава (с) – вогнутый четырехугольник, вершины которого соединены параболами

Численное, достаточно точное построение субдифференциала в трехмерном пространстве, как и срединной оси проблематично, поэтому будет рассматриваться другая задача – нахождение списка собственных значений линейно независимых подмножеств (окрестностей разных суставов) для карты высот, к которой применено преобразование Лапласа. Список собственных значений фактически определяет субэнтропию. Если два дистрибутива здорового и потенциально больного младенца разделяются, то у них должен быть разный набор передаточных функций (жестов), следовательно, и собственных значений. Дистрибутив представляет собой карту высот, изменяющуюся во времени. Берем два близких по времени кадра вычитаем один из другого получаем матрицу с большим количеством “нулевых” элементов, имеющую блочную структуру. К блоку применяем преобразование Лапласа. Строим по нему передаточную функцию, которая существует по теореме Синкхорна [21]:

Если A – матрица размера $n \times n$ со строго положительными элементами, то существуют диагональные матрицы D_1 и D_2 со строго положительными диагональными элементами такие, что $D_1 A D_2$ дважды стохастическая матрица. Матрицы D_1 и D_2 уникальны по модулю умножения первой матрицы на положительное число и деления второй на то же число. [21].

Простой итерационный метод приближения к двойной стохастической матрице состоит в том, чтобы поочередно масштабировать все строки и все столбцы матрицы A , чтобы в сумме получить 1.

Поскольку рассматривается задача на остовных графах с неопределенностью, возможно, необязательно брать разность кадров, достаточно нахождение собственных значений для бесконечной последовательности кадров (для каждого кадра по отдельности).

Можно сделать вывод, что задача распознавания жестов младенцев является перспективной в плане решения, хотя известного математического аппарата явно недостаточно.

Авторов интересуют и другие постановки задач, связанные с применением стохастической семантики в области визуализации, имеющие как прикладной, так и теоретический характер. Например, формула Ито для графических фильтров (параллельная фильтрация данных), рассмотрение волнового уравнения рендеринга как параметризованной модели белозумного случайного процесса (геометрическое решение).

Авторы намерены вплотную заняться установившимися возмущенными процессами в области визуализации с учётом человеческого фактора, начиная с реализации простых однопараметрических тестов (графических фильтров), количество которых долж-

но быть достаточно большое, так чтобы можно было вычислить скрытые зависимости (условные вероятности) между параметрами, например, воспользовавшись теоремой Колмогорова-Арнольда или ковариационным МНК. Давайте попробуем свести рассматриваемую задачу к решению системы ОДУ с помехами и нелинейностью типа композиции фильтров. (Хотя, в области визуализации предпочтительнее рассматривать гаусовские возмущенные процессы, то есть уравнения в частных производных, поскольку преобразование Фурье и всплески являются общепринятыми в данной области). В том числе надо выдвинуть правдоподобные гипотезы, чтобы определить контекст, качество, когнитивность и перцептивность информации, имеющие субъективный оттенок.

6.2 Методика оптимизации параметров в реактивных вычислениях для оценки профессиональной деятельности

Интерактивную визуализацию можно рассматривать как случайный процесс, важным свойством которого является асимптотическая сходимость. Следовательно, у профессионального пользователя изменение некоторого параметра должно стремиться к оптимальному значению.

6.2.1. Реактивные вычисления и параллельная фильтрация данных

Фильтрация данных рассматривается как частный случай реактивных вычислений. Фильтрация данных – любая операция над данными, изменяющая их количество. Следовательно, добавление объектов и детализация изображения являются фильтрами, а подавление шума в изображении – нет (белый шум не является помехой). Наибольшее распространение получила однопараметрическая фильтрация данных – так называемый слайсинг [9], например, сечения плоскостью (сферой), или изоповерхности. Так, параллельные сечения плоскостью имеют вид:

$$\sum_i Mf(Ax + By + Cz \in [d_i, d_{i+1}]) = f$$

где f – спектральная функция, заданная в узлах сетки (графа), необязательно регулярной, i – числовой параметр (количество блоков), аналогичный времени, фиксированная длина интервала Δd – определяет помехи, связанные с точностью вычисления (измерения), а выбор количества блоков определяет реактивные вычисления, например, взаимодействие реализовано с помощью слайдера. Если перейти к пределу по i , стремящемуся к бесконечности, то получится обобщение интеграла Ито через предел по вероятности (как отмечалось, в случае трехмерной визуализации удобнее рассматривать сходимость фундаментальной последовательности в сепарабельном банаховом пространстве).

Рассмотрим параллельную фильтрацию данных. С точки зрения параллельных вычислений принадлежность некоторому интервалу является выборкой, которую удобно реализовывать как конвейер или pipe – стандартная конструкция в разрабатываемом языке программирования. Далее данные надо передать на компьютер-клиент и отобразить на экран, причем это отображение должно быть непрерывным в некотором смысле, что и является основной темой данного раздела. По сути данный подход является формализованным обобщением архитектуры MVC (Model-View-Controller, Модель - Вид (отображения) - Контроллер). В частности, шаблон наблюдатель в контексте данной работы – это стационарный возмущенный процесс, для которого характерно равновероятность сообщений (реакций). В объектно-ориентированном программировании наиболее естественный способ реализации реактивных вычислений состоит в том, чтобы к методам и полям объектов добавить реакции, которые автоматически пересчитывают значения, и другие реакции зависят от изменений этих значений. С целью оптимизации производительности желательна редукция графа (синтаксического дерева, сети Петри), для этого необходимо хранить историю, что ведет к дополнительным накладным расходам (помехам). С целью упрощения модели пока редукция графа рассматриваться не будет, кроме того фильтрация данных является редукцией более вы-

сокого уровня. Если сопоставить долю реактивных вычислений доли параллельных вычислений, а выполнение реакций с передачей сообщений, то в обоих случаях очевидна общность математических моделей, в частности параметризованной модели бел шумного случайного процесса, следовательно, и общность синтаксиса в языке программирования. В MVC архитектуре с помощью реактивного программирования можно реализовать автоматическое отображение изменений из Model в View и наоборот из View в Model. Как отмечалось, и параллельные (распределенные) вычисления и визуализация вносят свои помехи.

6.2.2. Формула Ито для фильтра сечения плоскостью

На практике наибольшее распространение получили серия сечений плоскостью, параллельных одной из координатных плоскостей, в следствие простоты выборки. Конечно, для выбранных значений можно было бы применить преобразование Лапласа (являющимся бесконечно дифференцируемым), но авторы намерены свести задачу к ОДУ. Рассмотрим возмущенный процесс положив, например $\Delta d = \Delta z$, так чтобы возмущенная плоскость проходила через середину параллельной плоскости. Так как возможны два варианта пересечения, получается, что порядок переменных важен: результирующее множество получится открытым по x и замкнутым по y или наоборот. Для простоты будем рассматривать решетку (регулярную сетку) с количеством данных $f = N^3$. Целью данного анализа является оценка количества данных, например, для того чтобы зарезервировать память на ГПУ, при применении фильтра сечения плоскостью (в общем случае произвольного фильтра). Очевидно возможны два варианта: проекция ближайших соседних узлов на возмущенную плоскость (фильтр), принадлежавших окрестности Δz выше плоскости и по середине плоскости. В первом случае дополнительным количеством данных является N , что похоже на интеграл Ито, а во втором - $2N$, что похоже на интеграл Стратоновича. Данный подход будем называть экспертным.

Теорема: Пусть $f(N, \vec{x})$ возмущенный случайный процесс (Ито) по количеству данных. Если процесс Ито является мартингалом, то для любого фильтра $\varphi \in C^2$ дважды непрерывно дифференцируемого $\varphi f(N, \vec{x})$ является мартингалом.

Доказательство: Рассмотрим только схему доказательства. Пусть фильтр φ – производящий оператор (дважды непрерывно дифференцируемый). Далее, как для одномерной формулы Ито [4].

Следствием формулы Ито является инвариантность относительно суммы интегралов. Экспертный подход в случае процесса Ито дает следующую формулу:

$$f = N^3 \rightarrow f'(B_N) = N^2 + N,$$

где N является ds – интегралом. .

Из чего не трудно предложить формула Ито для фильтра сечения плоскостью (производную по фильтру):

$$f'(B_N) = (\vec{n}, \nabla \prod x_i)|_{x_i=N} + N.$$

Поскольку значение ds -интеграла равно N , количество данных не может быть слишком большим. Например, когда точность вычислений сопоставима с точностью арифметических операций, в постоянно расширяющемся открытом множестве произойдет некоторый скачек (про нейронную сеть в подобном случае говорят, что она переобучилась), что возможно даст некоторую другую трактовку обобщенной теоремы термодинамики [15].

Теперь вместо фильтра сечения плоскостью рассмотрим, например, сечение цилиндром. У авторов не настолько развито воображение, чтобы всегда использовать экспертный подход. Можно ли вместо скалярного произведения нормали и градиента сразу использовать Якобиан (производящий, характеристический (Гессиан) операторы)? То, что Якобиан является мартингалом вроде тоже очевидно. Для визуализации направление нормали важно. Рассмотрение фильтрации данных как дифференцирования многомерного случайного процесса по фильтру имеет определенные перспективы для визуализации многомерных данных (визуализация с неопределенностью, отоб-

ражать ds -интеграл) и оценки вычислительных методов с точки зрения СДУ. Перспективные направления исследования стохастической визуализации: визуализация неопределенности, стохастическая модель рендеринга (явное выделение особенностей), рассмотрение случайного процесса не по количеству данных, а, например, по кривизне (например, задана возмущенная поверхность вращения надо найти возмущенную ось вращения).

6.2.3. Основные определения и демонстрационные примеры в методике оптимизации параметров

В реактивных вычислениях изменение длины интервала приведет к автоматическому пересчету функции, например, в узлах плоскости. В основе формализации лежит аналогия, заключающаяся в том, что реактивные вычисления можно рассматривать как частное решение дифференциального уравнения или фазовую траекторию, в котором изменяются начальные данные (значения параметров). (Авторы и ранее активно использовали понятие траектории программы [29]). Так параллельные сечения плоскостью – это множество возмущённых траекторий, где каждая плоскость является множеством возмущённых траекторий. Ещё раз подчеркнем, что дискретным временем в этом случае является количество плоскостей (количество блоков). Траекторию с учетом событий будем обозначать $f(\cdot, \omega)$, пример стохастической траектории программы представлен на Рис. 10. Для сравнения на Рис. 11. представлена траектория программы (множества достижимости), явно зависящая от времени - $f(\cdot)$ [30].

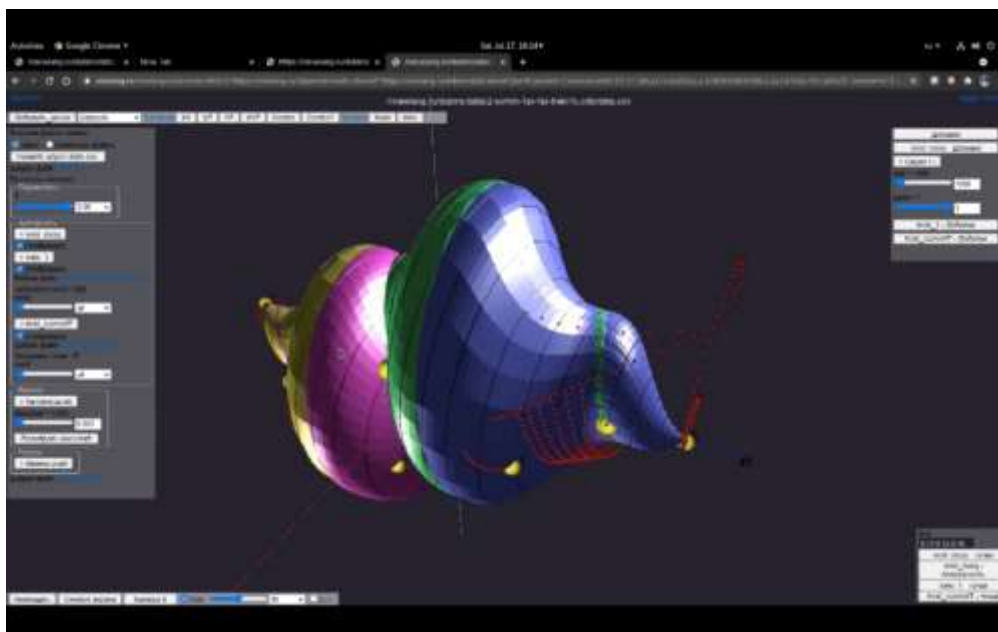


Рис. 10. Стохастическая траектория программы

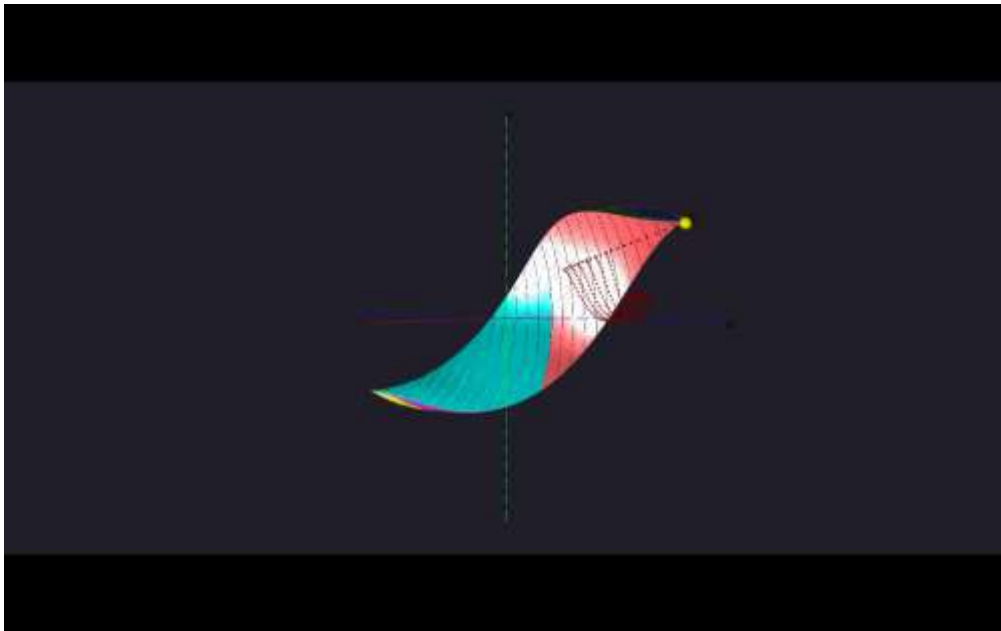


Рис. 11. Траектория программы (множества достижимости), явно зависящая от времени

Пора переходить к определению понятий, имеющих субъективную окраску:

1. Контекст (информации) полностью описывается конвейером фильтров, функцией штрафов и помехами. Так монотонная задача в смысле профессионального подхода с точки зрения ДС – это типичный параллельный конвейер:

2. Качество информации – это относительная, невероятностная (нечёткая) энтропия (в нечёткой энтропии вероятности заменены на функцию принадлежности, значение которой определяет эксперт);

3. Когнитивность информации – это точная нижняя граница качества информации;

4. Персептивность информации – это точная верхняя граница качества информации, в качестве которой предполагается использовать метрику PSNR (peak signal-to-noise ratio - отношение пикового сигнала к шуму, где шум – это среднеквадратичная ошибка).

Опираясь на примеры визуализации, дадим некоторые пояснения к этим определениям. В таких случаях рассматривается задача Дирихле со стохастическим управлением вида регулятор с неполной информацией о состоянии системы - задача стохастического управления визуальной информацией. Известно, что для её решения должна быть определена точная верхняя граница, следовательно, точная нижняя граница в плане моделирования не интересна.

Метрика PSNR активно используется в области визуализации, например, в модели шумоподавления Рис. 12. [2].



Рис.12. Original image and Noisy image, PSNR=29.38dB.

Для сжатого изображения при PSNR=40-50dB, изображение считается хорошего качества. Даже при достаточно высоком уровне шума человек сможет определить, что на

рисунке изображена женщина., поэтому авторы определили когнитивность информации как точную нижнюю границу качества информации. Но авторов интересуют возмущенные процессы, поэтому более наглядным примером является отображение ступенчатой функции.

Представьте отображение непрерывной функции с несколькими локальными экстремумами на экран, где параметром разбиения является количество интервалов. Хотя когнитивность информации в плане моделирования не особо интересна, наверное, точная нижняя граница качества информации соответствует минимуму шума (дисперсии) между точками локальных экстремумов и их проекциями на интервалы. Поскольку экран состоит из пикселей, то точная верхняя граница будет определена на польском пространстве (пространство, гомеоморфное полному метрическому пространству со счётным плотным подмножеством). Следующие рассуждения направлены на понижение точной верхней границы. В изображение можно добавить толщину линии (условие Лифшица) так, чтобы оно было плотным на конечном подмножестве (все толстые интервалы касаются друг друга). Такие изображения будем называть непрерывными с точки зрения зрительного восприятия (перцептивно непрерывными). Компенсаторная функция зрительного восприятия человека хорошо развита, следовательно, можно допустить, чтобы среднее значение шума было меньше толщины линии. Фактически это распределение Бернулли. Такие изображения будем называть компенсаторно-непрерывными. Примером таких изображений является Рис. 7.

Поскольку метрика PSNR применяется для двух близких изображений, возникает необходимость в фиксации изображения, выбранного экспертом. Соответствующее этому изображению значения параметра будем считать оптимальным, его потом можно уточнить по результатам тестирования. Очередная гипотеза заключается в следующем: если человек выбрал значение параметра меньше оптимального, то у него больше развито левое полушарие (логическое мышление), если больше, то правое. Далее приведен пример экспертного подхода в области визуализации. Гибридный вид отображения рассматривался для реализации облёта ЦМП, но применяться не стал, так как возмущенные гауссовские процессы хорошо справились с этой задачей.

Гибридный вид отображения в центре (в области прямого взгляда) имеет хорошую точность (ядро), а на периферии точность хуже (носитель). Эта задача формулируется следующим образом: максимизация информации (качества визуализации) при фиксированном размере памяти GPU. Далее рассмотрим фундаментальные последовательности, считая, что в одном графическом примитиве находится один узел графа. Так как графические примитивы занимают разный объём, то такая постановка задачи имеет смысл.

В основе моделирования лежит экспертный подход:

1. Качество визуализации равно невероятностной энтропии;
2. Строится антитонное распределение Рис. 13., где V - воксели, T - 3D-текстуры, P - полигональная графика, C - облако точек;
3. Качество визуализации определим, как функцию принадлежности:

$$\mu(V) = 1, \mu(T) = 0.9, \mu(P) = 0.8, \mu(C) = 0.7.$$

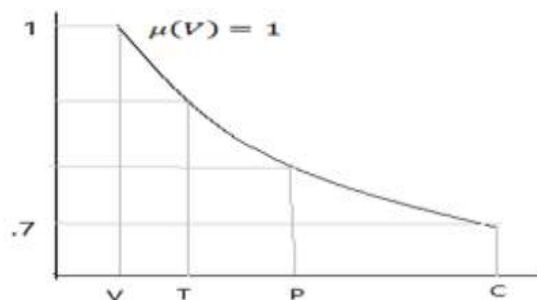


Рис. 13. Антитонное распределение

Рассмотрим множество перестановок ядро - носитель в блочной модели. Будем считать, что у той перестановки качество визуализации выше, у которой соответственно выше невероятностная энтропии (принцип максимизации). Невероятностная энтропии является интегральной характеристикой нечеткости нечеткого множества. Она вычисляется по стандартным формулам [31] – это нормализованная энтропия Шеннона:

$$H = -\frac{1}{\ln N} \sum_{i=1}^N p_i \ln p_i, \text{ где } p_i = \frac{\mu_A(x_i)}{\sum_{i=1}^N \mu_A(x_i)}$$

Промежуточной целью методики оптимизации параметров является разработка однопараметрических тестов.

Рассмотрим фильтр – серия слоёв Рис. 14. (аналогичный тест можно реализовать для фильтра – нарезка). Разработан и запрограммирован набор преобразований визуального представления для повышения качества восприятия – серия слоёв [30]. Он позволяет взглянуть на внутреннюю часть поверхности, а зрительное восприятие человека может аппроксимировать поверхность на пустых интервалах (рассматривается компенсаторная непрерывность). Можно посчитать метрику, которую авторы назвали информационный разрыв:

$\sup_p \frac{PSNR_{in}}{PSNR_{ex}}$, где p – количество блоков (не пустых интервалов), in – внутренняя часть поверхности, ex – внешняя часть поверхности.

Очевидно, что для одноцветного цилиндра достаточно трех непустых интервалов. Таким образом, можно определить случайный процесс (интерактивную визуализацию), который сопоставляет количеству цветов и кривизну поверхности количество интервалов и длину интервала, (которую можно посчитать по количеству интервалов). То есть пользователю ставится задача изменить количество слоёв, которое должно стремиться к оптимальному. Таким образом, можно отслеживать три квазиобъективных параметра: отклонение от оптимального количества слоёв (можно добавить и отсечение плоскостью, но для начала лучше один параметр), количество итераций, которое определяет скорость сходимости, и двоичный параметр (присутствия) – изменения ракурса.



Рис. 14. Визуальный образ множества достижимости с добавленными преобразованиями – отсечение по оси u и серия слоёв по оси ϕ [30].

Цель программирования – создать набор таких тестов. В контексте зрительного восприятия авторам известно некоторое число таких тестов, недостаточное для моделирования: метафора восхода солнца, параллакс движения, изменение контрастности.

В заключение данного раздела добавим пару очевидных определений. Комбинаторная функция (человека) в случае независимых переменных (параметров) описывается количеством информации (свойство аддитивности информации), а в случае зависимых переменных теоремой Колмогорова-Арнольда, поскольку рассматриваются только непрерывные отображения. Компенсаторная функция – это взвешенная сумма с пере-

менными весами (человек автоматически определяет эти веса в зависимости от ситуации, как отмечалось можно рассмотреть распределение Бернулли).

7. Заключение

Для верификации программного обеспечения, как и для визуализации в данной работе использованы СДУ, теория информации, сигнальные графы. Этот подход назван стохастической семантикой. Важно, что визуальный процесс является параллельным процессом (взаимодействующие последовательные процессы Хоара) с точки зрения программирования и случайным процессом с точки зрения математического моделирования.

Рассмотрение задач, связанных с действительно большими данными, неизбежно ведёт к использованию блочного подхода. В параллельных вычислениях блок можно ассоциировать с процессором и рассмотреть задачу максимизации энтропии (производительности). В разрабатываемой динамической системе онлайн визуализации и параллельных вычислений для геометрического распараллеливания можно реализовать и сравнить стационарный случайный процесс и установившийся случайный процесс, которые имеют разные аналитические решения. Это позволяет сделать вывод, что предложенная реализация стационарного процесса имеет определенную новизну.

В области верификации визуализации сделано не так много – предложена визуализация сеток, рассматриваемая как параметризованная модель белошумного случайного процесса. Авторы интересуют и другие постановки задач, связанные с применением стохастической семантики в области визуализации. Особо хочется отметить исследовательскую серию работ по обобщенному вычислительному эксперименту [28].

Конечно, данную работу нельзя считать завершённой, но направление, которое авторы назвали стохастическая семантика очевидно является перспективным. Авторы намерены вплотную заняться установившимися возмущёнными процессами в области визуализации, в том числе с учётом человеческого фактора (приведены наброски формализации в виде обсуждения для задачи детекции и методики оптимизации параметров в реактивных вычислениях для оценки профессиональной деятельности).

Список литературы

1. Croitoru F-A., Hondru V., Tudor Ionescu R., and Shah M. Diffusion models in vision: a survey, // IEEE Transactions on pattern analysis and machine intelligence 14(8): 2022. P. 1-25.
2. L'ezoray O. and Grady L Image Processing and Analysis with Graphs: Theory and Practice, CRC Press, July 2012. Graph Signal Processing and Applications.
3. Cheng-Zhong Xu, Francis C., Lau M. Optimal parameters for load balancing with the diffusion method in mesh networks. // Parallel processing letters, Volume 4, 1994 P. 139-147.
4. Oksendal B. (2003). Stochastic differential equations: an introduction with applications. Berlin: Springer. ISBN 3-540-04758-1.
5. Fathi A. (2009). Weak KAM theory in lagrangian dynamics. Cambridge studies in advanced mathematics.
6. Манаков Д., Авербух В. Верификация визуализации // Научная визуализация 2016. Кв. 1. Том 8. N: 1. С. 58 - 94.
7. Lakoff G. The contemporary theory of metaphor // Metaphor and thought. (2nd ed.). Cambridge: Cambridge university press, 1993, P. 202-251.
8. Scott D.S. Data types as lattices // Proceedings of the international summer institute and logic colloquium, Kiel, in Lecture notes in mathematics. Springer-Verlag. 499. P. 579-651.
9. Авербух В.Л., Манаков Д.В. Анализ и визуализация “больших данных” // Труды международной научной конференции “Параллельные Вычислительные Технологии”

(ПаВТ'2015). Екатеринбург, 31 марта - 2 апреля 2015. Челябинск, Издательский центр ЮУрГУ. 2015. С.332-340.

10. Васёв П.А. Визуализация работы алгоритма планирования параллельных задач. // *ГрафиКон 2023: 33-я Международная конференция по компьютерной графике и машинному зрению*, 19-21 сентября 2023, Москва: труды. С. 341-353. DOI: <https://doi.org/10.20948/graphicon-2023-341-353>

11/ Котов В.Е., Проблемы развития параллельного программирования // *Труды Всесоюзного симпозиума "Перспективы системного и теоретического программирования"*. Новосибирск, 1979. С. 58-72

12. Biberstein O., Buchs D., and Guelfi N. // *Object-oriented nets with algebraic specifications: The CO-OPN/2 formalism.* / Agha G., De Cindioand F, and Rozenberg G. editors, *Advances in Petri nets on object-orientation*, LNCS. Springer-Verlag, 2001

13. Кудрин К.А., Коварцев А.Н., Прохоров С.А. Методы автоматизации отладки в технологии графо-символического программирования // *Сборник научных трудов "Информационные системы и технологии"*, Самара, 1996. С. 75-79.

14. Averboukh Y. Lattice approximations of the first-order mean field type differential games. *Nonlinear Differ. Equ. Appl.* 28, 65 (2021). DOI: 10.1007/s00030-021-00727-2

15. Стратонович Р.Л. Теория информации. М.: Сов. радио, 1975. 424 с.

16. Baimuratov I., Nguyen T., Golchin R., Mouromtsev D. Developing non-empirical metrics and tools for ontology visualizations evaluation and comparing. *Scientific Visualization*. 2020. Vol. 12. No. 4. P. 71-84. DOI: 10.26583/sv.12.4.07.

17. Shih M., Zhang Y., Ma K.-L., Sitaraman J., Mavriplis D. Out-of-Core Visualization of TimeVarying Hybrid-Grid Volume Data // *IEEE Symposium on Large Data Analysis and Visualization*. 2014. P. 93 – 100.

18. Незнахина Е.Д. PTAS для задачи Min-k-SCCP в евклидовом пространстве произвольной фиксированной размерности // *Труды Института математики и механики УрО РАН*. 2015. Т. 21. N: 3. С. 268-278.

19. Datta N., Jozsa R., Dorlas T. Benatti F. Properties of Subentropy, // *J. Math. Phys.* 55 (2014) 062203. doi: 10.1063/1.4882935

20. Теплов А.М. Об одном подходе к сравнению масштабируемости параллельных программ // *Вычислительные методы и программирование*. 2014. Т. 15. Выпуск 4. С. 697-711

21. Розенберг В.Л. Сферическая блоковая модель динамики и сейсмичности литосферы: современное состояние и перспективы развития // *Современные методы оценки сейсмической опасности и прогноза землетрясений: III Всероссийская научная конференция с международным участием*, 25 -26 октября 2023, Москва: материалы. Москва: ИТПЗ РАН, 2023. С. 224-228.

22. Sinkhorn, Richard. (1964). A relationship between arbitrary positive matrices and doubly stochastic matrices // *Ann. Math. Statist.* **35**. P. 876–879. doi:10.1214/aoms/1177703591

23. Зенкин А. А. Когнитивная компьютерная графика / Под ред. Д.А. Поспелова. М.: Наука, 1991. 192 с.

24. Cui Q., Ward M., Rundensteiner E., and Yang J. Measuring data abstraction quality in multiresolution visualizations // *IEEE TVCG*, 12(5): 2006. P. 709–716,

25. Yang J., Ward M. and Rundensteiner E. Hierarchical exploration of large multivariate data sets // *Data visualization: The state of the art 2003*, P. 201–212.

26. Манаков Д.В. Модели абстракции данных: выборка (параллельные координаты), фильтрация, кластеризация // *Научная визуализация*, 2019, Кв.1. Том 11. N: 1. С. 139 - 176, DOI: 10.26583/sv.11.1.11

27. Кузнецов Д.Ф. Некоторые вопросы теории численного решения стохастических дифференциальных уравнений Ито // *Дифференциальные уравнения и процессы управления*. 1998. N 1. 367 с.

28. Мехедов И.С. Многолистная плоская фигура и её серединная ось // *Известия вузов. Математика*. 2011. N 12. С. 42–53.

29. Vasev P., Bakhterev M., Manakov D., Porshnev S., Forghani M. // On expressiveness of visualization systems' interfaces/. Scientific visualization/ 2022 14.5: P. 77 - 95, DOI: 10.26583/sv.14.5.06

30. Vasev P.A., Fedotov A.A. Visualization of three-dimensional reachable set for the Dubins car / Proceedings of the All-Russian Conference with international participation “Control Theory and Mathematical Modeling”, dedicated to the memory of Professor N.V. Azbelev and Professor E.L. Tonkov, Izhevsk. 2020, P. 264–265.

31. Кононюк А.Е. Дискретно-непрерывная математика. (Множества (нечеткие)). — В 12-и кн. Кн 2, ч.2— К.: Освіта України. 2012. 452 с.

32. Alekseev A.K., Bondarev A.E., Pyatakova Yu.S. On the use of canonical decomposition for visualization of the results of parametric calculations // Scientific visualization, 2023, Q. 4.Vol. 15. N: 4. P. 12 - 23, DOI: 10.26583/sv.15.4.02

Stochastic Semantics of Big Data (Parallel Computing and Visualization)

D. V. Manakov^{1,A}, P. A. Vasev^{2,A}

Krasovsky Institute of Mathematics and Mechanics, Ural Branch of the Russian Academy of Sciences

¹ ORCID: 0000-0001-6852-8096, manakov@imm.uran.ru

² ORCID: 0000-0003-3854-0670, vasev@imm.uran.ru

Abstract

Considering problems involving truly big data inevitably leads to the use of a block approach which is also used in both information theory and stochastic differential equations. As a natural metaphor the graph signals were chosen. This is a graph in nodes, of which a spectral function is defined in the examples considered this is a function of color (RGB), height or amount of data. In parallel computing, a block can be associated with a computing unit (processor) and consider the problem of entropy (performance) maximization. In the developed on-line visualization and concurrent computing system for geometric parallelization, it is possible to implement and compare a stationary random process (equiprobable messages implemented using broadcasting and mixins) and a steady-state random process (point-to-point messages), which have different analytical solutions. Together, this allows concluding that the proposed implementation of a stationary process has a certain novelty; in addition, it was intended to be more convenient for automated parallelization. The problems of automatic load balancing (interpolation problem) and optimal scalability of parallel computing (extrapolation problem) are also considered. Not much has been done in the field of visualization verification for example a mesh visualization has been proposed to be considered as a parameterized model of a white-noise random process. Of course, this work cannot be considered complete, but the direction that the authors called stochastic semantics is obviously promising.

The authors intend to take a close look at the established perturbed processes in the field of visualizations including those that take into account the human factor (the sketches of the formalization in the form of a discussion are given).

Keywords: signal graphs (graph signals), dynamic systems, load balancing, entropy, visualization of a digital surface model.

References

1. Croitoru F-A., Hondru V., Tudor Ionescu R., and Shah M. Diffusion models in vision: a survey, // IEEE Transactions on pattern analysis and machine intelligence 14(8): 2022. P. 1-25.
2. L'ezoray O. and Grady L Image Processing and Analysis with Graphs: Theory and Practice, CRC Press, July 2012. Graph Signal Processing and Applications.
3. Cheng-Zhong Xu, Francis C., Lau M. Optimal parameters for load balancing with the diffusion method in mesh networks. // Parallel processing letters, Volume 4, 1994 P. 139-147.
4. Oksendal B. (2003). Stochastic differential equations: an introduction with applications. Berlin: Springer. ISBN 3-540-04758-1.
5. Fathi A. (2009). Weak KAM theory in lagrangian dynamics. Cambridge studies in advanced mathematics.

6. Manakov D., Averbukh V. Verification of visualization // Scientific visualization 2016. Quarter 1. Volume 8. N: 1. P. 58 - 94. [In Russian]
7. Lakoff G. The contemporary theory of metaphor // Metaphor and thought. (2nd ed.). Cambridge: Cambridge university press, 1993, P. 202-251.
8. Scott D.S. Data types as lattices // Proceedings of the international summer institute and logic colloquium, Kiel, in Lecture notes in mathematics. Springer-Verlag. 499. P. 579-651.
9. Averbukh V. L., Manakov D. V. Analysis and visualization of "big data" // Proceedings of the International Scientific Conference "Parallel Computing Technologies" (2015). Yekaterinburg, March 31 - April 2, 2015. Chelyabinsk, SUSU Publishing Center. 2015. pp.332-340. [In Russian]
10. Vasev P. A. Visualizing the operation of the parallel task planning algorithm // GrafiKon 2023: 33rd International Conference on Computer Graphics and Machine Vision, September 19-21, 2023, Moscow: trudy, pp. 341-353. [In Russian] DOI: <https://doi.org/10.20948/graphicon-2023-341-353>
11. Kotov V.E., Problems of parallel programming development // Proceedings of the All-Union Symposium "Prospects for System and Theoretical Programming". Novosibirsk, 1979. pp.58-72. [In Russian]
12. Biberstein O., Buchs D., and Guelfi N. //Object-oriented nets with algebraic specifications: The CO-OPN/2 formalism. / Agha G., De Cindioand F, and Rozenberg G. editors, Advances in Petri nets on object-orientation, LNCS. Springer-Verlag, 2001
13. Kudrin K. A., Kovartsev A. N., Prokhorov S. A. Methods of debugging automation in graph-symbolic programming technology //Collection of scientific papers "Information systems and technologies", Samara, 1996. pp. 75-79. [In Russian]
14. Averboukh Y. Lattice approximations of the first-order mean field type differential games. Nonlinear Differ. Equ. Appl. 28, 65 (2021). DOI: 10.1007/s00030-021-00727-2
15. Stratonovich R.L. Theory of Information, Moscow: Sov. radio, 1975. , 424 p. [In Russian]
16. Baimuratov I., Nguyen T., Golchin R., Mouromtsev D. Developing non-empirical metrics and tools for ontology visualizations evaluation and comparing. Scientific Visualization. 2020. Vol. 12. N. 4. P. 71-84. DOI: 10.26583/sv.12.4.07.
17. Shih M., Zhang Y., Ma K.-L., Sitaraman J., Mavriplis D. Out-of-Core Visualization of TimeVarying Hybrid-Grid Volume Data // IEEE Symposium on Large Data Analysis and Visualization. 2014. P. 93 – 100.
18. Neznakhina E.D. PTAS for the Min-k-SCCP problem in a Euclidean space of arbitrary fixed dimension //Proceedings of the Institute of Mathematics and Mechanics of the Ural Branch of the Russian Academy of Sciences, 2015, vol.21, no. 3, pp.268-278.
19. Datta N., Jozsa R., Dorlas T. Benatti F. Properties of Subentropy, // J. Math. Phys. 55 (2014) 062203. doi: 10.1063/1.4882935
20. Teplov A.M. On one approach to comparing scalability of parallel programs / / Computational methods and programming. 2014. Vol. 15. Issue 4. pp. 697-711
21. Rozenberg V. L. Spherical block model of dynamics and seismicity of the lithosphere: current state and development prospects // Modern methods of seismic hazard assessment and earthquake prediction: III All-Russian Scientific Conference with International Participation, October 25-26, 2023, Moscow: materials. Moscow: ITPZ RAS, 2023. pp. 224-228.
22. Sinkhorn, Richard. (1964). A relationship between arbitrary positive matrices and doubly stochastic matrices // Ann. Math. Statist. 35. P. 876–879. doi:10.1214/aoms/1177703591
23. Zenkin A. A. Cognitive computer graphics / Ed. D. A. Pospelov. Moscow: Nauka, 1991. 192 p. [In Russian]
24. Cui Q., Ward M., Rundensteiner E., and Yang J. Measuring data abstraction quality in multiresolution visualizations // IEEE TVCG, 12(5): 2006. P. 709–716,
25. Yang J., Ward M. and Rundensteiner E. Hierarchical exploration of large multivariate data sets // Data visualization: The state of the art 2003, P. 201–212.

26. Manakov D.V. Data abstraction models: sampling (parallel coordinates), filtering, clustering // Scientific visualization 2019, Q. 1. Vol. 11. N: 1. pp. 139-176, [In Russian] DOI: 10.26583 / sv.11.1.11
27. Kuznetsov D.F. Some issues of the theory of numerical solution of Ito stochastic differential equations // Differential equations and control processes. 1998. N 1. 367 p. [In Russian]
28. Mekhedov I.S. Multi-sheeted plane figure and its median axis // News of universities. Mathematics. 2011. N 12. P. 42–53. [In Russian]
29. Vasev P., Bakhterev M., Manakov D., Porshnev S., Forghani M. // On expressiveness of visualization systems' interfaces/. Scientific visualization/ 2022 14.5: P. 77 - 95, DOI: 10.26583/sv.14.5.06
30. Vasev P.A., Fedotov A.A. Visualization of three-dimensional reachable set for the Dubins car / Proceedings of the All-Russian Conference with international participation “Control Theory and Mathematical Modeling”, dedicated to the memory of Professor N.V. Azbelev and Professor E.L. Tonkov, Izhevsk. 2020, P. 264–265.
31. Kononyuk A.E. Discrete-continuous mathematics. (Sets (fuzzy)). - In 12 books. Book 2, part 2 - K.: Education of Ukraine. 2012. 452 p. [In Russian]
32. Alekseev A.K., Bondarev A.E., Pyatakova Yu.S. On the use of canonical decomposition for visualization of the results of parametric calculations // Scientific visualization, 2023, Kv.4.Vol. 15. N: 4. P. 12 - 23, DOI: 10.26583 / sv. 15. 4. 02