

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
Институт космических и информационных технологий  
Кафедра систем искусственного интеллекта

УТВЕРЖДАЮ  
Заведующий кафедрой

\_\_\_\_\_ инициалы, фамилия  
подпись  
« \_\_\_\_ » \_\_\_\_\_ 2017 г.

**МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ**

Визуальное представление многомерных эмпирических данных

09.04.02 Информационные системы и технологии

09.04.02.01 Информационно-управляющие системы

Научный руководитель	_____	доцент, канд. техн. наук	О. А. Попова
	подпись, дата		
Выпускник	_____		Е. С. Погорелый
	подпись, дата		
Рецензент	_____	ст. науч. сотр., канд., физ.-мат. наук	А. Н. Рогалев
	подпись, дата		
Нормоконтролер	_____		М. А. Аникьева
	подпись, дата		

Красноярск 2017

Продолжение титульного листа магистерской диссертации по теме:  
«Визуальное представление многомерных эмпирических данных»

Нормоконтролер

\_\_\_\_\_  
подпись, дата

М. А. Аникьева

## РЕФЕРАТ

Выпускная квалификационная работа в форме магистерской диссертации по теме «Визуальное представление и анализ эмпирических многомерных данных» содержит 86 страниц текстового документа, 2 приложения и 31 использованный источник.

ВИЗУАЛИЗАЦИЯ, АНАЛИЗ, МНОГОМЕРНОСТЬ, ДАННЫЕ, ВИЗУАЛЬНОЕ ПРЕДСТАВЛЕНИЕ, МНОГОМЕРНЫЕ ДАННЫЕ, ЭМПИРИЧЕСКИЕ ДАННЫЕ, ИНТЕРАКТИВНОСТЬ, МОДЕЛИРОВАНИЕ, ЧИСЛЕННЫЙ ВЕРОЯТНОСТНЫЙ АНАЛИЗ.

Актуальность темы обуславливается проблемой исследования и анализа различного рода данных. Во многих случаях происходит так, что значительная часть закономерностей и связей между различными параметрами изучаемого набора данных, в большей степени многомерных, скрыта. Целью магистерской диссертации являлось повышение качества и эффективности процедуры проведения анализа представленных данных.

Поставленная цель достигалась решением следующих задач:

- анализ темы исследования с целью выявления основных проблем исследуемой области;
- оценка эффективности существующих методов, применяемых для решения выявленных проблем;
- разработка алгоритма и программного модуля, предназначенного для повышения качества исследования представления многомерных данных;

В рамках магистерской диссертации были изучены теоретические аспекты визуально-интерактивного моделирования и представления многомерных данных и данных больших объемов, выявлены основные существующие проблемы, осуществлена разработка алгоритма представления многомерных данных в виде динамических сечений и программного модуля визуального моделирования многомерных данных, осуществлен поиск областей применения результатов данного исследования.

## СОДЕРЖАНИЕ

Введение.....	6
1 Анализ средств визуально интерактивного представления и моделирования многомерных данных.....	8
1.1 Многомерные данные и их представление .....	9
1.2 Визуально-интерактивное представление данных .....	12
1.3 Визуализация многомерных данных.....	14
1.4 Способы визуального представления данных .....	17
1.5 Представление многомерных данных на примере тессеракта .....	26
1.6 Проблемы, связанные с обработкой и представлением данных .....	37
2 Численный вероятностный анализ в области представления многомерных данных .....	40
2.1 Применение численного вероятностного анализа в области представления многомерных данных .....	40
2.2 Гистограммы как способ агрегирования многомерных данных.....	41
2.3 Понятие многомерных гистограмм.....	43
3 Разработка программного модуля для визуального моделирования многомерных данных.....	47
3.1 Концепция разрабатываемого программного модуля.....	47
3.2 Технологии, применяемые для реализации программы визуализации многомерных данных .....	49
3.3 Описание структуры и основных классов программного модуля .....	51
3.3.1 Основные классы .....	52
3.3.2 Классы управления .....	54
3.3.3 Ключевые объекты реализации программного модуля визуального представления многомерных данных .....	56
3.4 Графическое представление интерфейса программного модуля визуализации данных.....	58
Заключение .....	63

Список сокращений .....	64
Список использованных источников .....	65
Приложение А .....	69
Приложение Б .....	78

## ВВЕДЕНИЕ

В настоящее время, при решении многих практических задач, в том числе в задачах принятия решений, при обработке данных дистанционного зондирования земли [16], изучении крупно масштабных распределенных систем важную роль играет процесс обработки и анализа данных с целью последующей интерпретации полученной информации и построения рекомендаций для лица, принимающего решение [8]. Решение подобных задач существенно усложняется, когда данные характеризуются большими объемами и представляют собой сложный многомерный объект.

Принятие решений является одной из сложных и ответственных составляющих человеческой деятельности. Многообразие целей, сложность задач, ограничения и неопределенность требуют компьютерной поддержки процесса принятия решений. В настоящее время разработка компьютерных систем поддержки принятия решений представляет собой одно из интенсивно развивающихся направлений создания информационных систем [16]. В рамках поддержки решений используются различные технологии, среди которых следует указать на визуально-интерактивное моделирование. Массивы многомерных данных и данных больших объемов заключают в себе множество скрытых связей и закономерностей, зачастую недоступных при использовании имеющихся способов визуального анализа. За несколько прошедших десятилетий значительно возрос интерес к проблемам анализа многомерных данных, которые, в свою очередь, представляют собой данные, которые содержат информацию о трех или более признаках для каждого объекта. Сложность исследования подобного рода данных заключается в том, что, как правило, значительная часть закономерностей и связей между различными параметрами изучаемого набора данных скрыта [27]. Таким образом, крайне актуальной задачей является исследование, модификация и симбиоз существующих методов и способов представления и моделирования

многомерных данных, позволяющих осуществлять плодотворный визуальный анализ имеющихся наборов данных.

Целью данной работы являлось исследование и, по возможности, дальнейшая модификация и применение средств, методов и алгоритмов представления и моделирования различных данных больших объемов, что, в свою очередь, позволит повысить качество визуального анализа эмпирических многомерных данных.

Также, последовательно определялись и ставились задачи, необходимые для достижения определенной ранее цели работы, и позволяющие последовательно исследовать определенную темой работы тематику:

1) Провести проблемный анализ представленной тематики и исследование на тему необходимости изучения способов визуально-интерактивного представления и моделирования данных больших объемов.

2) Изучить теоретические аспекты визуально-интерактивного моделирования и представления эмпирических многомерных данных, обозначить основные существующие проблемы и сформулировать задачи исследования.

3) Разработать алгоритм, а впоследствии и программный модуль визуального представления многомерных данных в виде динамических сечений.

В итоге, выпускная квалификационная работа представляет собой 3 главы:

1) «Анализ средств визуально-интерактивного представления и моделирования многомерных данных», представляющая собой изучение и анализ теоретических аспектов согласно тематике работы.

2) «Численный вероятностный анализ в области представления многомерных данных» представляет описание средств ЧВА, доступных при обработке данных большой размерности.

3) «Разработка программного модуля для визуального моделирования многомерных данных», которая включает описание концепции разрабатываемого модуля построения динамических сечений, применяемых программных средств и процесса разработки.

## **1 Анализ средств визуально-интерактивного представления и моделирования многомерных данных**

Человеческая природа такова, что наше постижение мира, накопление знаний о нем, решение задач, возникающих перед человеком возможно двумя различными путями. Один из них позволяет работать с абстрактными цепочками символов, с текстами и т. п. Этот механизм мышления человеческого сознания обычно называют символическим, алгебраическим или логическим. Второй механизм мышления обеспечивает работу с чувственными образами и представлениями об этих образах. Его называют образным, геометрическим, интуитивным и т. п. [3].

Стремление человека выразить мысль, передать идею в форме графического изображения можно отметить еще с самых ранних времен. Графические изображения (на камне, холсте, бумаге, металле и других средствах отображения информации) давно используются в обучении. Экран компьютера как средство пассивного отображения графики не обладает принципиальной новизной. Применение графики в исследовательских работах не только увеличивает скорость передачи информации и повышает уровень ее понимания, но и способствует развитию таких важных для специалиста любой отрасли качеств, как интуиция, образное мышление.

Новой для сферы обучения является интерактивность компьютерной графики, благодаря которой в процессе анализа изображений существует возможность динамически управлять их содержанием, формой, размерами и цветом, рассматривать графические объекты с разных сторон, приближать и удалять их, менять характеристики освещенности и проделывать другие подобные манипуляции, добиваясь наибольшей наглядности. Таким образом, это возможность не только для пассивного созерцания графических иллюстраций, но и для активного исследования характеристик графических моделей изучаемых объектов или процессов [22].



## 1.1 Многомерные данные и их представление

За несколько прошедших десятилетий значительно возрос интерес к проблемам анализа многомерных данных, которые, в свою очередь, представляют собой данные, которые содержат информацию о трех или более признаках для каждого объекта. Сложность исследования подобного рода данных заключается в том, что, как правило, значительная часть закономерностей и связей между различными параметрами изучаемого набора данных скрыта. Таким образом, крайне актуальной задачей является исследование, модификация и симбиоз существующих методов и способов представления и моделирования многомерных данных, позволяющих осуществлять плодотворный визуальный анализ имеющихся наборов данных, ведь, в дополнение к той информации, которую можно извлечь из одномерных и двумерных наборов, многомерные данные можно использовать для получения информации о том, существует ли простая зависимость между этими признаками, насколько они взаимосвязаны, можно ли предсказать значение одной переменной на основании значений остальных. Важно отметить, что исследуется не только парные взаимосвязи признаков, но и зависимости относительно их совокупности, [8, 22, 23]. Примером многомерных данных является число граждан стран СНГ и Балтии, обучающихся в МГУ на 01.11.2003 г., по странам, приведенное ниже, в таблице 1, в которой содержатся данные о числе студентов, магистрантов, стажеров и аспирантов из республик бывшего СССР, обучающихся в МГУ.

Таблица 1 – Информация о иностранных студентах, обучающихся в МГУ

№	Страна	Студенты, чел.	Магистранты, чел.	Стажеры, чел.	Аспиранты, чел.	Всего, чел.
1	Азербайджан	57	2	0	4	63
2	Армения	32	2	0	1	36
3	Беларусь	89	0	1	1	91
4	Грузия	49	2	0	3	54

## Окончание таблицы 1

№	Страна	Студенты, чел.	Магистранты, чел.	Стажеры, чел.	Аспиранты, чел.	Всего, чел.
5	Казахстан	517	5	3	2	527
6	Кыргызстан	27	2	0	1	30
7	Латвия	26	1	0	3	40
8	Литва	37	0	0	0	37
9	Молдова	83	0	0	1	84
10	Таджикистан	13	0	0	1	14
11	Туркменистан	14	0	0	0	14
12	Узбекистан	29	2	1	0	62
13	Украина	1134	7	1	7	1149
14	Эстония	21	1	1	1	24
	Всего	2168	24	7	25	2224

Анализ данных, приведенных в таблице 1, позволяет сделать определенные выводы: число граждан стран СНГ и Балтики, обучающихся в МГУ, не пропорционально численности населения этих стран; наличие большого числа граждан Казахстана и Украины, обучающихся в МГУ, объясняется наличием Казахстанского и Черноморского филиалов Московского государственного университета; не все страны бывшего СССР перешли на подготовку бакалавров, магистров и так далее [26].

Значения переменных, которые регистрируются с помощью чисел, имеющих содержательный смысл, называют количественными данными. Данные, приведенные в столбцах № 3–7 таблицы 1, являются количественными. В то же время числа, приведенные в первом столбце таблицы 1, не являются количественными данными: они только указывают на номер государства при их алфавитном расположении. Данные числа не имеют содержательной интерпретации. С количественными данными можно выполнять операции над числами, такие, как вычисление среднего и оценку изменчивости [1].

Существуют различные способы представления МД, одним из которых являются нечеткие методы представления данных. Одной из основных причин возникновения идеи нечеткого множества является так называемый, принцип несовместимости, который заключается в том, что с увеличением размеров и сложности системы существенно усложняется ее моделирование с помощью известных математических выражений. Существенно возрастает число переменных и параметров, измерение и определение которых сильно затрудняется, а создание адекватной модели становится практически невозможным [2, 19]. Лотфи А. Заде предложил лингвистическую модель, которая использует не традиционные математические выражения, а слова, отражающие качество. Применение данного метода не обеспечивает аналогичную точность, которой обладают регулярные математические модели, но дает возможность создать приемлемое, качественное описание системы [11].

Многомерные данные могут быть представлены с помощью временных рядов, которые представляют собой собранный в разные моменты времени статистический материал о значении каких-либо параметров исследуемого процесса [7, 8, 20, 23]. Временные ряды позволяют наглядно продемонстрировать статистические данные, а также проанализировать их. Анализ временных рядов является совокупностью математико-статистических методов анализа, предназначенных для выявления структуры временных рядов и для их прогнозирования. Выявление структуры временного ряда необходимо для того, чтобы построить математическую модель того явления, которое является источником анализируемого временного ряда. Прогноз будущих значений временного ряда используется для эффективного принятия решений. Типичным примером временного ряда возможно назвать биржевой курс, при анализе которого пытаются определить дальнейшие изменения различных экономических показателей.

Кроме того, существует такой способ представления МД как гистограмма, которая является функцией, приближающей плотность вероятности некоторого распределения, построенной на основе выборки из него.

Данный метод представления данных применяется для сравнения значений в течение некоторого периода или же соотношения величин. Следует отметить, что гистограммы подходят в основном для визуализации данных на начальном этапе статистической обработки и при увеличении числа значений снижается наглядность представления [7, 11].

Такой метод представления МД как гистограммный временной ряд (ГВР) описывает ситуации, когда в течение каждого момента времени известны гистограммы, аппроксимирующие функции плотности некоторых случайных величин. Подобные ситуации возникают, когда необходима агрегация большого числа данных некоторые моменты времени. Во многих случаях гистограммы более информативны, чем, например, среднее значение. Области, где ГВР полезны, включают экономику, мониторинг окружающей среды и др. [8].

## **1.2 Визуально-интерактивное представление данных**

Применение графики в исследовательских работах не только увеличивает скорость передачи информации и повышает уровень ее понимания, но и способствует развитию таких важных для специалиста любой отрасли качеств, как интуиция, образное мышление.

Представление данных в графическом виде подразумевает использование различных технологий среди которых следует указать на визуально-интерактивное моделирование (ВИМ), представляющее собой метод, который предполагает использование динамического дисплея, в котором пользователь может изменять параметры модели во время обычной работы и анализировать их последствия [7, 27].

Для более точного понимания термина «Визуально-интерактивное моделирование» следует разобраться с понятием интерактивности, так как сам принцип так называемого «диалога» между машиной и экспертом (или пользователем), лежащий в основе понятия интерактивности, также неотделим от ВИМ. Одним из общих понятий данного термина является:

«Интерактивность — это принцип организации системы, при котором цель достигается информационным обменом элементов этой системы». Причем, элементами интерактивности являются все элементы взаимодействующей системы, при помощи которых происходит взаимодействие с другой системой/человеком (пользователем). Также, следует привести узкоспециализированные определения. Для ИС интерактивность есть ни что иное, как способность информационно-коммуникационной системы, активно и адекватно реагировать на действия пользователя. Подобное свойство считается признаком того, что система «умная», то есть обладает каким-то интеллектом. Если говорить о системах программирования, то здесь интерактивность принципиально ограничена этапом трансляции, который отделяет внесение изменений от их тестирования [16]. В интерактивных системах отдельного этапа трансляции нет; приложение состоит из одних и тех же объектов и во время разработки, и во время исполнения. Мало того, отсутствие разбиения на инструментальную и исполнительную среды позволяет использовать аналогичные средства и во время разработки, и во время исполнения, вследствие чего имеется возможность изменения работающего приложения и немедленного отслеживания результата данного изменения.

В случае с ВИМ, интерактивность «расширяется» до «визуальной интерактивности», которая представляет собой ни что иное, как визуальную интерактивную имитацию, что является имитацией решения, при которой конечный пользователь наблюдает развитие имитационной модели в анимационной форме с использованием графических представлений. Пользователь может взаимодействовать с имитационной моделью и проверять различные стратегии решения. Другими словами, интерактивность осуществляется посредством применения различных анимационных форм графического представления [11].

Важным преимуществом ВИМ является то, что при использовании визуального отображения у пользователей имеется возможность следить

за развитием событий, когда они происходили, что необходимо для выявления потенциальных ошибок [13].

Визуально-интерактивное моделирование может представлять статические и динамические системы. Статические модели проявляют визуальный образ результата одного альтернативного решения в определенный момент времени (при помощи компьютерных окон несколько результатов могут быть сравнены на одном экране). Динамические модели проявляют и отображают системы, которые развиваются во времени. Эволюция представляется при помощи анимации [17].

Одной из проблем ВИМ является визуализация данных — задача, с которой сталкивается в своей работе каждый исследователь. К задаче визуализации данных сводится проблема представления в наглядной форме данных эксперимента или результатов теоретического исследования. Традиционные инструменты в данной области (графики, диаграммы) недостаточно результативно справляются с задачей визуализации, когда возникает необходимость изобразить более трех взаимосвязанных величин, то есть, наглядно визуализировать многомерные данные [16, 27].

### **1.3 Визуализация многомерных данных**

Понятие «визуализация многомерных данных» описывает графические изображения, генерируемые программой, в которой контент изображения определяется считыванием цифровых данных. Под визуализацией данных понимается способ представления многомерного распределения данных на двумерной плоскости, при котором, по крайней мере, качественно отражены закономерности, присущие исходному распределению — кластерная структура, топологические особенности, внутренние зависимости между признаками, информация о расположении данных в исходном пространстве [16, 27] и т. д. Часто данные представляются в числовой форме. Существуют также программы, способные визуализировать понятия, заложенные в текстовых документах.

Такие программы организуют геометрические фигуры — точки, линии, круги и прямоугольники — таким образом, чтобы они представляли собой интерпретацию считываемых программой данных. Атрибуты — относительное расстояние, размер и цвет — отражают отношения между геометрическими фигурами. Исходя из определения понятия данных следует сделать вывод, что основной целью визуализации многомерных данных является, насколько возможно, наглядное представление данных в том виде, в котором пользователь сможет увидеть необходимые ему закономерности и связи между атрибутами (параметрами и т. д.) различных объектов [29].

Следует понимать, что визуализация данных — не только инструмент, позволяющий наглядно представлять различные закономерности, но, прежде всего, эффективное средство, предназначенное для более полного понимания абстрактных процессов. Сферы применения графической визуализации довольно широки, но в настоящий момент, наиболее широко, данный вид отображения данных используется для:

- представления информации, другими словами, наглядного изображения разного рода данных, с помощью которого возможно определить ранее неочевидные связи и закономерность, находящиеся в представляемых данных;
- изображения моделей различного генеза и итерационных архитектур;
- представления моделей;
- представления методов принятия решений;
- поддержки непосредственно алгоритмов визуализации;
- визуализации многомерных форм (без необходимости описания вспомогательных форм).

Визуальное интерактивное моделирование представляет собой ни что иное, как графическую анимацию, в которой системы и процессы представлены лицу, принимающему решения (ЛПР), в динамике, что дает возможность визуально представить результаты различных потенциальных действий. Данная технология с успехом используется в информационных системах принятия решений в области управления процессами и операциями.

ВИМ использует компьютерные графические отображения для представления воздействия различных управленческих решений [7, 16].

ВИМ-технологии часто применяются в экономическом и финансовом моделировании, помогая принимать те или иные решения, исходя из заданных условий, внешних факторов и параметров рынка.

В настоящее время визуализация данных является актуальным направлением развития ВИМ-технологий. Методы представления МД приобрели за последнее десятилетие значительную популярность в среде различных пользователей, поскольку они направлены на поддержку процессов принятия решений в различных сферах практической деятельности и имеют важное значение в задачах управления знаниями [10].

На данный момент существуют различные подходы и методы для обработки и представления многомерных данных больших объемов (такие как кластерный анализ, таксономия, распознавание образов, факторный анализ) [23], ввиду особой популярности у исследователей тематики представления и обработки различного рода многомерных данных, является закономерным то, что в настоящее время разработано большое число программных и алгоритмических средств визуализации многомерных структур. Прежде всего, к ним относятся универсальные системы визуализации *Data Explorer*, *RockWorks99* и *Slicer Dicer* [23].

Анализ многомерных данных, в свою очередь, представляет собой совокупность методов и алгоритмов, позволяющих получить максимально возможную информацию о массиве числовых данных, расположенных в некоторой области многомерного пространства. Многомерная визуализация прежде всего занимается изучением и созданием методов представлений многомерных структур и вопросами объединения этих методов в системы визуализации, позволяющие изучить многомерный объект с разных сторон [5].

Также, следует обозначить, что под визуализацией данных можно понимать такой способ описания данных, когда размерность их описания сокращается до двух измерений. Если в процессе сокращения размерности будет



сохранена существенная часть закономерностей, присущих данным (разбиение на классы, отношения соседства), то исследователь получает возможность наглядно представить себе исходный набор многомерных данных, сделать выводы об их распределении [25].

#### **1.4 Способы визуального представления данных**

В научно-технической документации применяются различные виды визуализации (ниже приведены примеры визуализации данных и их описание).

Таблица (*matrix*) показывает набор данных в виде заполненных его значениями ячеек, которые образуют собой строки и столбцы. Каждому столбцу и строке задан параметр, определяющий ячейку для значения [11].

Диаграммы связей показывают связи внутри набора данных, как правило, достаточно большого. Так, круговая диаграмма связей (*network diagram, arc diagram*). Показывает связи внутри набора данных в виде кольца, на котором расставлены значения. Значения связаны дугами или линиями, находящимися во внутренней области круга. При большом количестве значений они могут находиться и внутри кольца, хотя это менее наглядно. Связи также могут иметь направление.

Количественная диаграмма представляет собой изображение, рисунок, чертёж (графическое представление данных), позволяющее быстро оценить соотношение нескольких величин. Представляет собой геометрическое символическое изображение информации с применением различных приёмов техники визуализации [12].

Графики (рисунок 1) — это тип диаграмм, на которых полученные данные изображаются в виде точек, соединённых прямыми линиями. Точки могут быть как видимыми, так и невидимыми (ломаные линии).



Рисунок 1 — Пример графика

Также могут изображаться точки без линий (точечные диаграмм). Для построения диаграмм-линий применяют прямоугольную систему координат. На оси наносятся масштабы. Как правило, графики представляют собой двухмерные линейные графики одной или многих переменных.

Гистограммы (рисунок 2) позволяют увидеть, как распределены значения переменных по интервалам группировки, то есть, как часто переменные принимают значения из различных интервалов [20].



Рисунок 2 — Пример гистограммы

Таким образом, гистограмма представляет собой графическое изображение зависимости частоты попадания элементов массива числовых данных от соответствующего интервала группировки. Гистограмма представлена в виде прямоугольников, высота которых пропорциональна частоте, а ширину прямоугольников (интервал группировки)

обычно для удобства восприятия берут одинаковую. Особенно полезна гистограмма для большого числа данных, например, больше сотни значений. На простой гистограмме отображаются частоты значений одной переменной, а на составной гистограмме возможно отображать одновременно частоты нескольких переменных [18].

Столбчатая диаграмма, показанная на рисунке 3, используется для демонстрации количественных показателей переменных, также изображает статистические данные в виде вертикальных прямоугольников или трёхмерных прямоугольных столбиков. Каждый столбик изображает величину уровня данного статистического ряда. Все показатели выражены одной единицей измерения для сравнения показателей данного ряда [4].

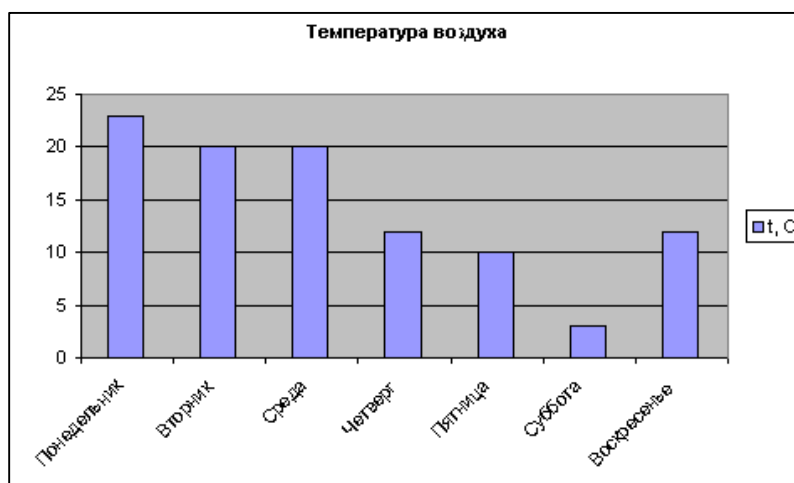


Рисунок 3 — Пример столбчатой диаграммы

Разновидностями столбчатых диаграмм являются линейные (полосовые) диаграммы. Они отличаются горизонтальным расположением столбиков. Столбчатые и линейные диаграммы взаимозаменяемы. Столбчатые диаграммы могут изображаться и группами (одновременно расположенными на одной горизонтальной оси с разной размерностью варьирующих признаков). Образующие поверхности столбчатых и линейных диаграмм могут представлять собой не только прямоугольники, но и квадраты, треугольники и т. д. [11].

Двухмерные диаграммы рассеяния используются для визуального исследования зависимости между двумя переменными  $X$  и  $Y$  (например, весом и ростом человека, рекламой и объемом продаж и т. д.).

Тепловая диаграмма (*heat map*) (рисунок 4) сравнивает значения внутри набора данных, закрашивая их одним из цветов в заранее выбранном спектре.

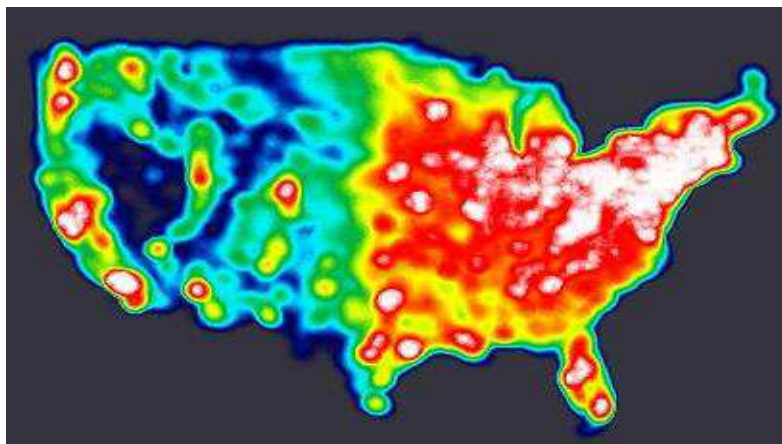


Рисунок 4 — Пример тепловой диаграммы

Основой является изображение или другая диаграмма, на которой расставлены значения. Цвет зависит от величины параметра и накладывается в виде пятен. Выбор цветового заполнения изображения зависит от пользователя, возможно представление в условных цветах, либо в оттенках серого, где градация цветового заполнения изменяется в зависимости от числового значения, т.е. для минимального и максимального значений в наборе данных задаются полярные цвета (например, синий и красный), либо полярные градации оттенков серого (белый и черный) [12].

Часто, если речь идет о наборе трехмерных данных, каждая величина в котором имеет три значения, возможно представление подобного набора данных в виде поверхности (рисунок 5).

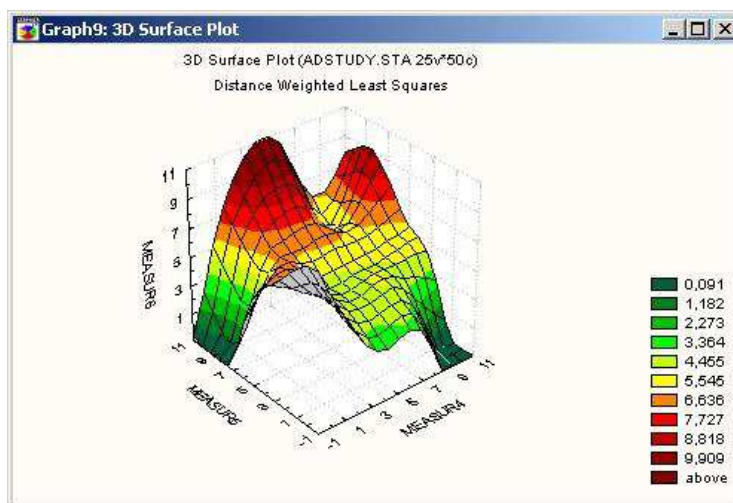


Рисунок 5 — Пример поверхности функции

Значения откладываются на трех координатных осях, впоследствии интерполируются. Данный способ зачастую применяется совместно с использованием условных цветов.

Трёхмерные диаграммы (рисунок 6) используется трёхмерная визуализация, спроецированная на плоскость, что придаёт ей отличительные черты или позволяет иметь общее представление об исследуемой области, в которой данная диаграмма применяется [31].

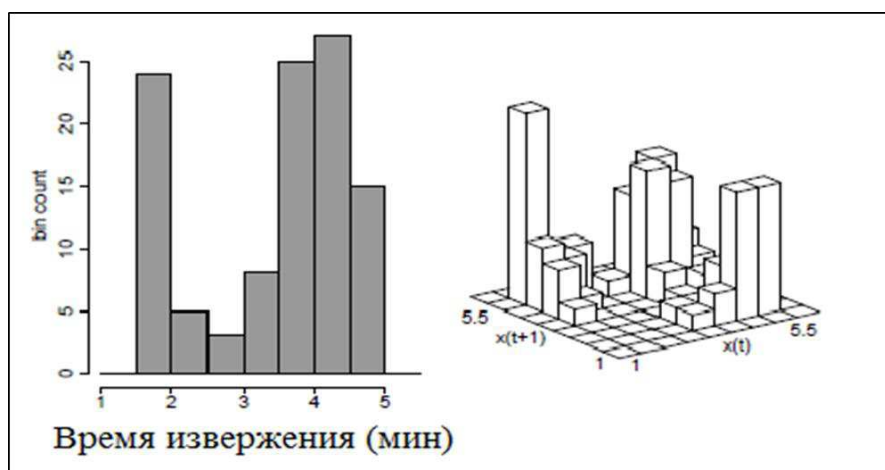


Рисунок 6 — Пример трехмерной диаграммы (справа)

Еще одной технологией визуализации является стереопара. Стереопара (рисунок 7) есть ни что иное, как пара параллельных плоских изображений

одного и того же объекта (сюжета), имеющая различия между изображениями, призванные создать эффект объёма [31].

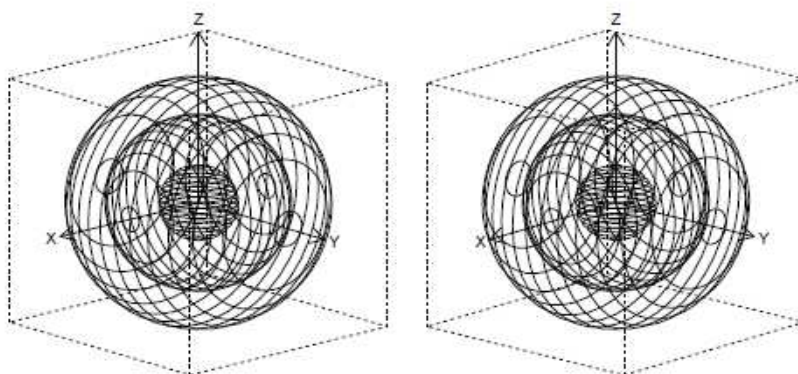


Рисунок 7 — Пример стереопары

Эффект возникает в силу того, что расположенные на разном удалении от наблюдателя части сюжета при просмотре с разных точек (соответствующих правому и левому глазу) имеют различное угловое смещение (параллакс). При рассматривании стереопары таким образом, чтобы каждый глаз воспринимал только изображение, предназначенное для него, создаётся иллюзия наблюдения объёмной картины (по принципу бинокулярного зрения) [31].

Попытки создания первых стереоизображений были предпринимались начиная с середины XIX века: для получения зрительного ощущения объёмной сцены достаточно спроектировать на сетчатки правого и левого глаз наблюдателей двух картин, соответствующих двумерным (плоским) изображениям, регистрируемым разнесенными в пространстве наблюдателями. Зачастую, создание подобных изображений связана с определенными техническими трудностями (одновременное ведение съемки с различных ракурсов, оцифровка трехмерных компьютерных анимаций двумя виртуальными видеокамерами и так далее), кроме того, процедура воспроизведения также вызывает определенные проблемы [14]. По-видимому, одним из первых решений на промышленной основе задачи статических анимированных стереоскопических визуализаций для массового

потребителя явилось создание стереоскопов и кинематоскопов — устройств, использующих для раздельного проектирования изображений стереопар в глаза наблюдателя при помощи методов геометрической оптики. Современной реализацией данного метода стали шлемы виртуальной реальности, использующие стереоскопический эффект, в которых «правое» и «левое» изображения создаются посредством двух миниатюрных экранов. Высокая стоимость подобных технических решений, сравнительно низкое разрешение устройств вывода графической информации, сложности в организации традиционного способа общения преподавателя с обучаемой аудиторией и возможность использования устройств лишь в индивидуальном режиме существенно сдерживают внедрение технологий, основанных на виртуальной реальности, в традиционное образование. Однако, следует признать то, что перспектива внедрения возможности глубокого погружения пользователя в виртуальный мир, позволяющий возможность интерактивного воздействия на действующие в нем законы, идентичные, либо отличающиеся, открывает недоступные ранее горизонты познания, творчества и обучения [14].

К методам формирования применимых для демонстрации в аудитории стереоскопических изображений относятся различные методы разложения изображений стереопар, проецируемых на экраны. Анаглифический метод (рисунок 8), основанный на хроматической селекции предназначенного для каждого из глаз изображений при помощи очков с красно-синими или красно-зелеными светофильтрами, является наиболее простым по техническому воплощению, ввиду простоты создания подобных стереоскопических изображений и относительно дешевых материалов для изготовления средств просмотра. Имея данные преимущества, анаглифический метод формирования стереоскопических изображений получил довольно широкое распространение и давно реализован на практике.



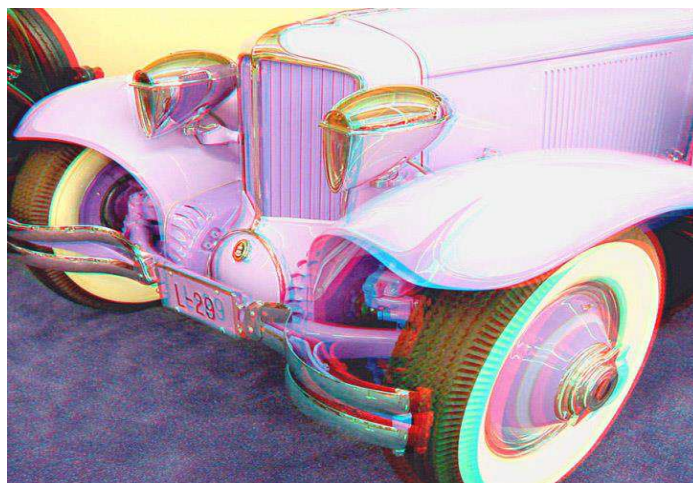


Рисунок 8 — Анаглифическое изображение

Передаваемое на экран изображение представляет собой суперпозицию двух квазимонохроматических слайдов, выполненных в цветах, определенных максимумами пропускания фильтров. Для создания подобных графических файлов используется программное обеспечение, имеющее достаточно простой интерфейс и не требующее значительных вычислительных ресурсов. В настоящее время существует широкий выбор программ, предназначенных для создания анаглифических стереопар. Данный метод обладает рядом существенных недостатков, основными из которых являются низкое качество цветопередачи и излишняя утомляемость пользователя, обусловленная отличной от естественной палитрой составляющих стереопару изображений. Формируемое объёмное изображение, благодаря эффекту, обусловленному бинокулярным смещением цветов, воспринимается однотонным или (при определённом соотношении яркостей) ахроматическим. Привыкание пользователя к специфическим условиям восприятия происходит достаточно быстро. Однако, после не столь долгого (около 15 минут) пребывания в анаглифических очках у пользователя на продолжительное (порядка получаса) время возникает ощущение дискомфорта от восприятия обычного мира и снижается цветовая чувствительность. Перечисленные негативные факторы явились основанием для признания нецелесообразности использования метода, как одного из способов демонстраций образовательной направленности.



С другой стороны, анаглифический метод признан пригодным для оперативной оценки качества объемного восприятия сцены при разработке стереоскопических демонстраций. Более подходящим для исследований возможностей применения стереоскопических трехмерных визуализаций в обучении был признан поляризационный метод. Изображения, составляющие стереопары, создаются направленными световыми пучками с линейными, либо круговыми поляризациями, разделение которых осуществляется с помощью поляризационных очков. Для создания на экране пары слабо разнесенных в пространстве изображений могут быть использованы обыкновенные мультимедийные проекторы, формирующие изображения в неполяризованных пучках света, которые после прохождения через поляризаторы, приходят в состояния с взаимно ортогональными поляризациями (линейными или круговыми) [14]. Использование линейной поляризации приводит к появлению зависимости качества разделения изображений от угла наклона головы наблюдателя, что не встречается при применении круговой поляризации. Подобными свойствами обладают металлические покрытия в случае нормального (или почти нормального) падения света на плоскость проецирования.

Использование технологий, чередующихся во времени демонстраций кадров стереопары, позволяет устранить необходимость использования двухпроекторной системы, приводящей к проблеме сведения изображений. Кадры, предназначенные для правого и левого глаз, могут разделяться при помощи активных очков затворного типа с модулируемой прозрачностью, поочередно пропускающих «правое» и «левое» изображения, либо пассивных поляризационных очков (через быстро переключаемое устройство, изменяющее состояние поляризации для каждого из последующих кадров, пропускается предметный световой пучок с проектора). Проблема сохранения стандартного качества изображения в таких схемах решается повышением частоты повторения кадров, что, однако, обуславливает качественно другие требования к системным требованиям демонстрационного ПК. Беспроводная синхронизация устройств,

осуществляется при помощи инфракрасных датчиков, сегодня не представляет больших проблем, но организационные вопросы установления единой частоты смены кадров и общего для разных моделей протокола обмена информацией между проекционным устройством и активными очками до сих пор остаются весьма актуальными. Образовательные учреждения, зачастую имеющие строго ограниченный бюджет, указывают на то, что стоимость дополнительного оборудования, такого как быстродействующий синхронный модулятор поляризации, либо наборы активных очков, оказывается соразмерной стоимости второго проектора для стереопары.

При использовании монитора при просмотре стереоскопических изображений существуют дополнительные к приведенному списку технологии, такие как размещение двух изображений в четных и нечетных строках экранной развертки с их последующим разделением при помощи экранной маски, обеспечивающей ортогональные поляризации излучения от строк, формирующих разделяемые изображения.

### 1.5 Представление многомерных данных на примере тессеракта

В данном разделе представлена программная реализация для отрисовки четырехмерного куба, отображаемая на плоскости монитора. Данное представление интересно тем, что позволяет увидеть в движении фигуру высокой размерности (в данном случае, порядка 4).

В первую очередь, необходимо начать с разъяснения, что представляет собой четырехмерное пространство. На рисунке 9 показано одно из представлений одномерного пространства, а, именно, ось  $Ox$ , любая точка на которой характеризуется одной координатой.

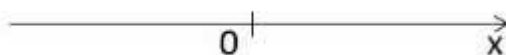


Рисунок 9 — Представление одномерного пространства

Далее, если провести ось  $OY$  перпендикулярно оси  $OX$ , получится двумерное пространство, то есть плоскость  $XOY$ , любая точка на которой характеризуется двумя координатами — абсциссой и ординатой (рисунок 10).

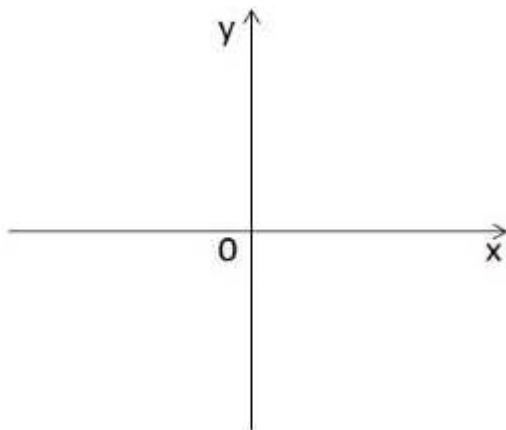


Рисунок 10 — Представление двумерного пространства

Если провести ось  $OZ$  перпендикулярно осям  $OX$  и  $OY$ , получится трёхмерное пространство, в котором у любой точки есть абсцисса, ордината и аппликата (рисунок 11).

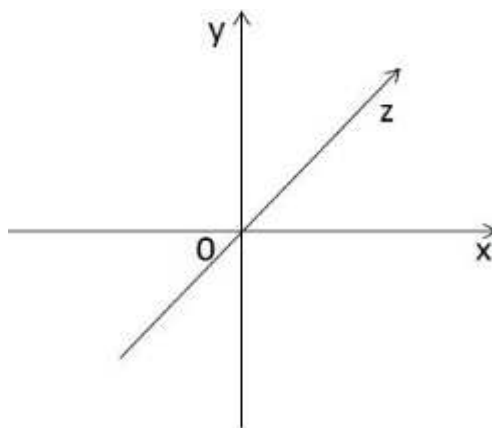


Рисунок 11 — Представление трехмерного пространства

Логично, что четвёртая ось,  $OQ$ , является перпендикулярной осям  $OX$ ,  $OY$  и  $OZ$  одновременно. Однако, невозможно точно построить подобную ось, и потому остаётся только попытаться ее представить (рисунок 12). У каждой точки в четырёхмерном пространстве есть четыре координаты:  $x$ ,  $y$ ,  $z$  и  $q$ .

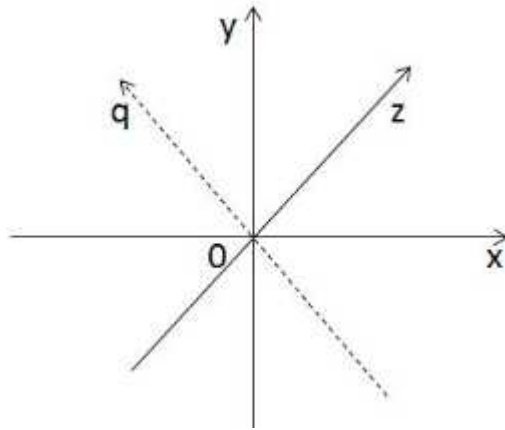


Рисунок 12 — Представление четырехмерного пространства

Теперь следует поговорить о том, как образуется куб в четырехмерном пространстве (тессеракт). На рисунке 13 изображена фигура одномерного пространства — линия.



Рисунок 13 — Линия в одномерном пространстве

Если сделать параллельный перенос данной линии вдоль оси  $OY$ , а затем соединить соответствующие концы двух получившихся линий, получится квадрат (рисунок 14).



Рисунок 14 — Квадрат, получившийся посредством параллельного переноса линии из рисунка 13 вдоль оси  $OY$

Аналогично, если сделать параллельный перенос квадрата вдоль оси  $OZ$  и соединить соответствующие вершины, то получится куб (рисунок 15).

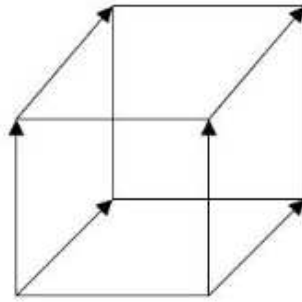


Рисунок 15 — Куб, образованный посредством параллельного переноса квадрата вдоль оси  $OZ$

Осуществление параллельного переноса куба вдоль координатной оси  $OQ$  и соединение вершин двух данных кубов, обеспечивает в результате визуальное изображение тессеракта или, другими словами, четырехмерного куба, представленное на рисунке 16. Следует заметить, что в тессеракте, как и трехмерном, линии и плоскости, образующие куб, либо параллельны, либо перпендикулярны друг другу.

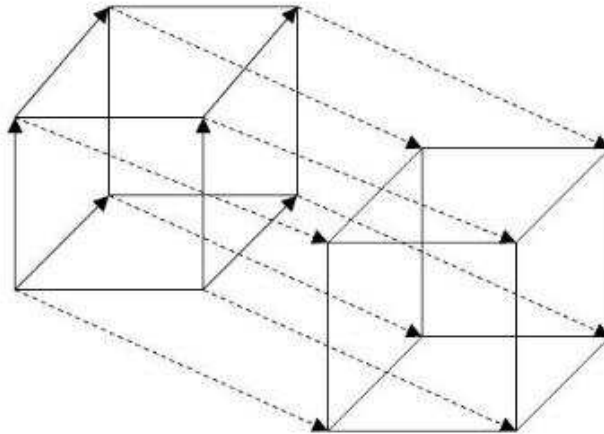


Рисунок 16 — Тессеракт, образованный посредством параллельного переноса куба вдоль оси  $OQ$

При возникновении необходимости отрисовать куб на плоскости, следует его спроецировать. Наглядное представление показано на рисунке 17.

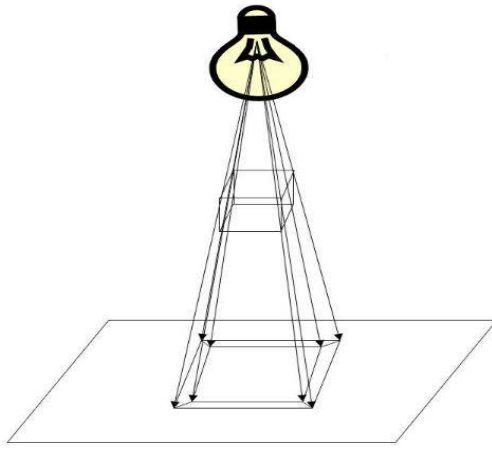


Рисунок 17 — Представление проекции куба на плоскость

Следует представить, что в воздухе над поверхностью висит каркасная модель куба (другими словами — проволочная, а над ней — лампочка. Если включить лампочку, обвести карандашом тень от куба, а потом выключить лампочку, то на поверхности будет изображена проекция куба. Если посмотреть на рисунок с лампочкой, видно, что все лучи сошлись в одной точке. Данная точка называется точкой схода и используется для построения перспективной проекции (кроме нее есть и параллельная, в случае, если все лучи в проекции параллельны друг другу. Результат — не создается ощущения объёма, однако, параллельная проекция легче, и, если точка схода достаточно сильно удалена от проецируемого объекта, то разница между данными двумя проекциями малозаметна) [6]. Чтобы спроецировать данную точку на данную плоскость, используя точку схода, нужно провести прямую через точку схода и данную точку, а потом найти точку пересечения получившейся прямой и плоскости. А для того, чтобы спроецировать более сложную фигуру, например, куб, нужно спроецировать каждую его вершину, а потом соответствующие точки соединить. Следует заметить, что алгоритм проекции пространства на подпространство можно обобщить для случая  $4D \rightarrow 3D$ , а не только  $3D \rightarrow 2D$ .

Как говорилось выше, человек, по своей природе, не имеет возможности точно представить, как выглядит ось  $OQ$ , равно как и тессеракт. Однако, возможно получить ограниченное представление о нём, спроецировав его на объём, а затем, нарисовав на экране компьютера.

Также, необходимо обратить внимание на построение проекции тессеракта. На рисунке 18 слева находится проекция куба на плоскость, а справа — тессеракта на объём.

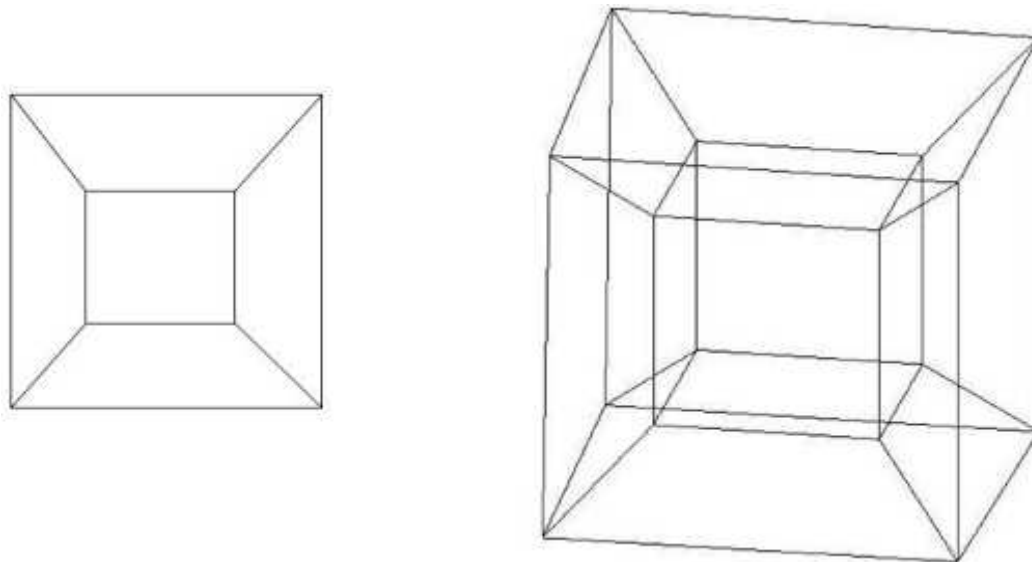


Рисунок 18 — Проекция куба на плоскость (слева)  
и тессеракта на объём (справа)

Данные проекции довольно схожи: проекция куба выглядит как два квадрата, маленький и большой, один внутри другого, и у которых соответствующие вершины соединены линиями. А проекция тессеракта выглядит как два куба, маленький и большой, один внутри другого, и у которых соответствующие вершины соединены. Но человек зрительно видел и знает куб, и имеет возможность с уверенностью сказать, что и маленький квадрат, и большой, и четыре трапеции сверху, снизу, справа и слева от маленького квадрата, на самом деле являются квадратами, при чём равными. Логично предположить, что и у тессеракта прослеживается подобная тенденция.

Данная программная реализация способна не только отрисовывать проекцию тессеракта на объём, но и вращать его. Следует сказать о вращении параллельно плоскости, для чего необходимо представить, что куб вращается вокруг оси  $OZ$ . Тогда каждая из его вершин описывает окружность вокруг оси  $OZ$  (рисунок 19).

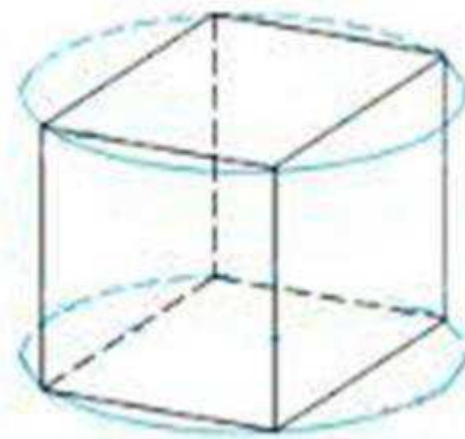


Рисунок 19 — Вращение куба вокруг оси  $OZ$

Окружность представляет собой фигуру плоскую. И плоскости каждой из данных окружностей параллельны между собой, и в данном случае параллельны плоскости  $XOY$ . То есть, следует говорить не только о вращении вокруг оси  $OZ$ , а ещё о вращении параллельно плоскости  $XOY$ . Как видно, у точек, которые вращаются параллельно оси  $XOY$  меняются только абсцисса и ордината, аппликата же остаётся неизменной и, кроме того, следует говорить о вращении вокруг прямой только в случае с трёхмерным пространством. В двумерном всё вращается вокруг точки, в четырёхмерном пространстве вокруг плоскости, в пятимерном — происходит вращение вокруг объёма. И если вращение вокруг точки человек может себе представить, то вращение вокруг плоскости и объёма — более трудоемкая процедура. А если говорить о вращении параллельно плоскости, то тогда в любом  $n$ -мерном пространстве точка может вращаться параллельно плоскости [6].

Следует сказать о матрице поворота, в которой, при умножении точки на собственно саму матрицу, получается точка, повёрнутая параллельно плоскости на угол  $\phi$ , что, в случае с двумерным пространством, выглядит подобным образом (1):

$$\begin{pmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{pmatrix} \quad (1)$$



Формулы (2) и (3) для нахождения координат точки, повернутой на угол  $\phi$ :

$$x_{a\phi} = \cos \phi \cdot x_a - \sin \phi \cdot y_a, \quad (2)$$

$$y_{a\phi} = \sin \phi \cdot x_a + \cos \phi \cdot y_a, \quad (3)$$

где  $x_a, y_a$  — абсцисса и ордината точки, которую необходимо повернуть;

$x_{a\phi}, y_{a\phi}$  — координаты уже повернутой точки.

Для трёхмерного пространства данная матрица поворота на угол  $\phi$  обобщается следующим образом (4):

$$\begin{pmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4)$$

Вращение параллельно плоскости  $XOY$ . Координата  $Z$ , как видно, не меняется, а меняются только  $X$  и  $Y$  (формулы (5), (6), (7)):

$$x_{a\phi} = \cos \phi \cdot x_a - \sin \phi \cdot y_a + z_a \cdot 0, \quad (5)$$

$$y_{a\phi} = \sin \phi \cdot x_a + \cos \phi \cdot y_a + z_a \cdot 0, \quad (6)$$

$$z_{a\phi} = x_a \cdot 0 + y_a \cdot 0 + z_a \cdot 1, \quad (7)$$

где, по сути,  $z_{a\phi} = z_a$ . Матрица поворота в данном случае выглядит (8):

$$\begin{pmatrix} \cos \phi & 0 & -\sin \phi \\ 0 & 1 & 0 \\ \sin \phi & 0 & \cos \phi \end{pmatrix} \quad (8)$$

Вращение параллельно плоскости  $XOZ$ , основано на формулах (9), (10) и (11):

$$x_{a\phi} = \cos \phi \cdot x_a - \sin \phi \cdot z_a + y_a \cdot 0, \quad (9)$$

$$y_{a\phi} = x_a \cdot 0 + y_a \cdot 1 + z_a \cdot 0, \quad (10)$$

$$z_{a\phi} = \sin \phi \cdot x_a + y_a \cdot 0 + \cos \phi \cdot z_a. \quad (11)$$

Для 4-го измерения матрица поворота выглядит следующим образом (12):

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi & 0 \\ 0 & \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (12)$$

Проекция тессеракта осуществляется по принципу, показанному на рисунке 20 и описанному далее.

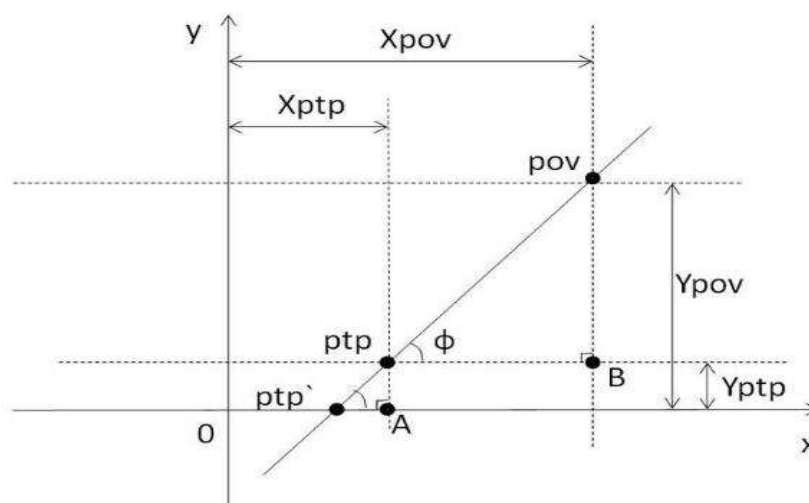


Рисунок 20 — Принцип построения проекции

Точка  $rov$  — *Point of view* (точка зрения),  $ptp$  — *Point to project* (точка, которую необходимо спроецировать), а  $ptp'$  — искомая точка на оси  $OX$ .

Углы  $rovptpB$  и  $ptpptp'A$  равны как соответствующие (пунктирная линия параллельна оси  $OX$ , прямая  $rovptp$  — секущая). Значение  $x$  в точке  $ptp'$  равно значению  $x$  в точке  $ptp$  за минусом длины отрезка  $ptp'A$ . Данный отрезок возможно найти из треугольника  $ptpptp'A$ :  $ptp'A = ptpA / tg\ ptpptp'A$ . Тангенс угла  $tg\ ptpptp'A = (Y_{rov} - Y_{ptp}) \cdot (X_{rov} - X_{ptp})$ . Таким образом:  $X_{ptp'} = X_{ptp} - Y_{ptp} / tg\ ptpptp'A$ .

Для того, чтобы спроецировать точку трёхмерного пространства на плоскость, рассматривается две плоскости —  $XOZ$  и  $YOZ$ , и для каждой из них решается данная задача. В случае четырёхмерного пространства следует рассматривать уже три плоскости:  $XOQ$ ,  $YOQ$  и  $ZOQ$ .

Принцип работы программы заключается в следующем:

- 1) инициализируются шестнадцать вершин тессеракта;
- 2) в зависимости от введённых команд тессеракт поворачивается;
- 3) происходит проецирование на объем;
- 4) в зависимости от введённых пользователем команд поворачивается проекция тессеракта на объем;
- 5) тессеракт проецируется на плоскость;
- 6) тессеракт отрисовывается.

Также, в программе, помимо поворота и проекции, имеется возможность отрисовки анаглифа, подсвечивания, для удобства просмотра, одной из граней красным цветом, регуляция координат точек — «глаз» и уменьшение или увеличение скорости поворота.

Рабочее окно программы, показанное на рисунке 21, представлено построенной каркасной моделью тессеракта, все действия по вращению куба, изменению его различных характеристик и характеристик вращения, а также включению режима стерео, осуществляются при помощи управляющих клавиш на клавиатуре.

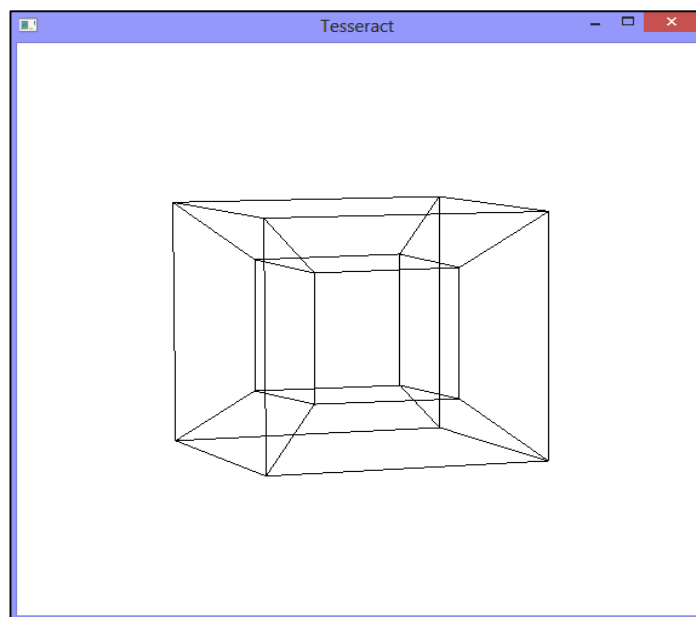


Рисунок 21 — Рабочее окно программы

Возможности программы позволяют отображать тессеракт в режиме стерео (рисунок 22), для полноценного просмотра получившегося эффекта, необходимо использовать соответствующие устройства (*3D*-очки и т. д.).

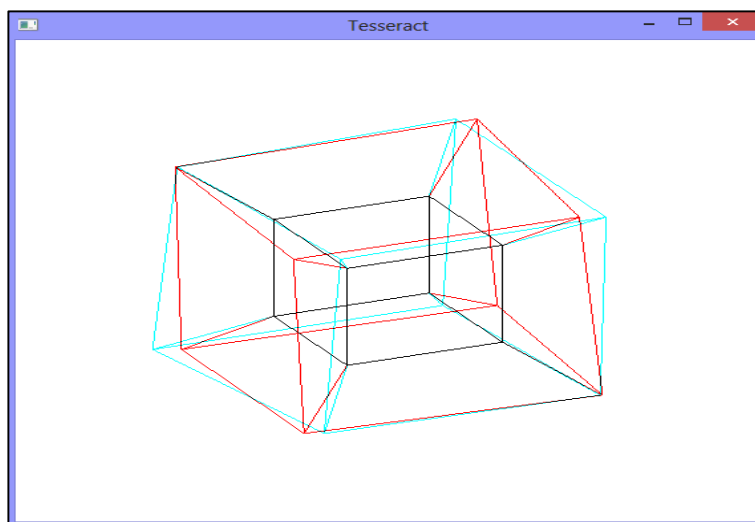


Рисунок 22 — Использование стереоэффекта

Подводя итог, следует сказать, что данная программа позволяет человеку, использующему ее, более основательно ознакомиться с четырехмерным пространством. Если рассуждать об изученных технологиях и способах визуализации данных, следует сказать, что многие из данных технологий

исторически зарекомендовали себя под конкретные типы и специфику данных. Другими словами, данные, свободно визуализируемые с помощью одних технологий, могут быть абсолютно нечитаемыми при попытке визуализации посредством других технологий и способов представления и отображения данных [28].

## **1.6 Проблемы, связанные с обработкой и представлением данных**

Одной из важных проблем анализа многомерных данных и последующей интерпретации полученных результатов является тот факт, что пространства разных размерностей топологически неэквивалентны, и, как следствие, любой вид отображения искажает геометрию многомерного множества. В этом случае теряется часть информации, или не выполняется свойство непрерывности представления. Это приводит к тому, что не удаётся построить методы, достаточные для всех задач визуализации. Поэтому разработка специализированных способов графического представления, выделяющих важную, необходимую информацию, представляет актуальную задачу. Кроме того, оказывается целесообразным использовать несколько методов, объединённых в систему визуализации [11, 15, 22].

В настоящее время можно выделить несколько подходов к анализу многомерных данных: классический (скалярный); подход функционального анализа; векторно-матричный; тензорный; многомерно-матричный. Классический или скалярный подход характеризуется тем, что имеющийся набор данных рассматривается состоящим из отдельных скалярных чисел или переменных. Математическая модель данных строится как соотношение, связывающее между собой отдельные скалярные переменные. Теоретическую основу классического подхода составляет классический математический анализ. Классический подход к анализу многомерных данных имеет следующие недостатки: громоздкость (необозримность) математических моделей многомерных данных, полученных с его помощью; отсутствие аналогии

с одномерным случаем; слабая формализованность; отсутствие теоретической общности; отсутствие алгоритмической общности, построенного на основе модели, для любого числа переменных их степеней без его модификации. Очевидно, что отсутствие теоретической общности влечет за собой отсутствие и алгоритмической общности. Однако теоретическая общность не обеспечивает алгоритмическую хотя бы потому, что модель, обладающая теоретической общностью, может плохо формализоваться [19, 22].

Для стандартного числа измерений реализация общей схемы анализа данных возможна лишь потому, что человек, существуя в трехмерном пространстве, обладает укладывающимися в сознание геометрическими образами и представлениями для пространств с числом измерений  $n \leq 3$ . Для многомерных пространств с числом измерений  $n > 3$  подобных геометрических образов у человека нет. Следовательно, не имеется и надежных методов графического представления данных в многомерном пространстве, кроме проецирования данных во вложенные пространства со стандартным числом измерений. Таким образом, при отсутствии возможности реализовать общую схему анализа данных, для случаев многомерных пространств остаются два основных направления исследования данных: методы и подходы, позволяющие понизить размерность исследуемого пространства до стандартной и попытки построения визуальной концепции для представления данных в графическом виде [15].

Следует сказать, что в настоящее время существует множество способов и методов обработки многомерных данных: анализ главных компонент, каноническая корреляция, кроме того, имеют место быть дискретно-косинусное преобразование и вейвлет-анализ: дискриминантный и факторный анализы, а также анализ соответствий, многомерное метрическое и неметрическое шкалирование и так далее [11].

Также, следует сказать, что каждая из имеющихся в распоряжении ученых технологий визуализации, будет становиться все менее информативной и «читаемой» по мере увеличения размерности представляемых данных.

Данная проблема, называемая «проклятием размерности», на данный момент, остается нерешенной и представляет интерес для многих исследователей по всему миру. Увеличение размерности данных ведет к увеличению сложности процесса обработки и снижению «ценности» данных, так как, при увеличении размера обрабатываемых данных, происходит увеличение количества так называемого «шума» относительно изначально необходимых данных. Д. Скотт сформулировал эту проблему как «проклятие размерности» [30].

По мнению многих исследователей одним из подходов к снижению объемов данных и их размерности является вычленение из «вороха» необработанных данных только их части, нужной для дальнейшего использования. Такой подход целесообразен тогда, когда уменьшение объема осуществляется без риска потери необходимой информации, или если возможные потери можно оценить на основе априорных и апостериорных оценок погрешностей используемых методов. В противном случае, имеют место необратимые последствия.

## **2 Численный вероятностный анализ в области представления многомерных данных**

Численный вероятностный анализ (ЧВА) представляет собой новый раздел вычислительной математики, предназначенный для решения различных задач со случайными входными данными. Основой численного вероятностного анализа являются понятие вероятностного расширения и численные операции над плотностями вероятности случайных величин. Численный вероятностный анализ направлен прежде всего на разработку методов представления, обработки, численных процедур моделирования и анализа данных, способствующих снижению уровня неопределенности в зависимости от ее типа, характера, специфических особенностей, объема и источников на всех стадиях информационного процесса, сопровождающего принятие управленческого решения [24].

### **2.1 Применение численного вероятностного анализа в области представления многомерных данных**

Предметом численного вероятностного анализа является решение различных задач со стохастическими неопределенностями в данных с использованием численных операций над плотностями вероятностей случайных величин и функций со случайными аргументами, для чего предлагается разнообразный инструментарий, включающий такие понятия как гистограммная арифметика, вероятностные, естественные и гистограммные расширения, гистограммы второго порядка. ЧВА представляет собой непараметрический подход и может успешно применяться для вероятностного описания систем в рамках визуально-интерактивного моделирования, повышая тем самым качество исследования систем [24].



Численный вероятностный анализ представляет собой раздел вычислительной математики, предметом которого является решение задач со стохастическими неопределённостями в данных в условиях элиторной и эпистимической неопределенности с использованием численных операций над плотностями вероятностей случайных величин и их функций. Необходимость разработки численных операций, определяющих предмет арифметики, определяется следующими основаниями. Одним из способов представления и описания стохастической неопределенности в данных является функция плотности вероятности. В рамках классической теории вероятностей приводятся формулы выполнения простейших арифметических операций над случайными величинами [21]. В реальности использование данных процедур или существенно затруднено, или практически невозможно. А, поскольку, при построении множества решений в задач визуально-интерактивного представления и моделирования многомерных данных необходимо осуществлять численные арифметические процедуры, в том числе возведение в степень, нахождение максимального и минимального значений и выполнение операций бинарных отношений, то разработка эффективных численных методов есть актуальная задача.

## **2.2 Гистограммы как способ агрегирования многомерных данных**

Одним из способов уменьшения объема данных является агрегация. Существуют различные подходы к агрегации данных. Например, в рамках численного вероятностного анализа рассматриваются различные подходы к представлению и агрегированию данных больших объемов [7]. ЧВА представляет собой эффективный инструмент извлечения знаний, который может успешно применяться в рамках технологии визуально-интерактивного моделирования с целью получения дополнительной информации для принятия эффективных решений [16, 27].

Термин «гистограмма» был введен известным математиком Карлом Пирсоном для визуальной поддержки статистического анализа для обозначения общей формы графического представления данных в виде столбцов как областей частотности в соответствии с масштабом их базиса. С другой стороны, гистограмму можно рассматривать как математический объект, который можно использовать для описания математических процедур и операций, сохраняя суть частотного распределения данных [22]. В общем случае гистограммой называется случайную величину, плотность распределения которой представлена кусочно-постоянной функцией. Гистограмма  $P$  определяется сеткой  $\{x_i | i = 0, \dots, n\}$ , на каждом отрезке  $[x_{i-1}, x_i]$ ,  $i = 1, \dots, n$  гистограмма принимает постоянное значение  $p_i$  [30].

Следует отметить важное свойство гистограмм, которое связано с понятиями агрегирование информации и агрегирование данных [8]. С этой точки зрения гистограмма выступает как система агрегированных данных, представляющих измерения характеристик какого-либо процесса или явления. С другой стороны, гистограмма представляет собой математический объект, как кусочно-постоянную функцию, заданную сеткой, на каждом отрезке которой гистограмма принимает постоянное значение. Простая и гибкая структура гистограмм позволяет использовать их для представления многомерных данных больших объемов с целью последующей агрегации и моделирования, использовать их в численных расчетах и для аналитических выводов. Например, ГВР описывает ситуации, когда в течение каждого момента времени известны гистограммы, исследуемых параметров [20]. Подобные ситуации возникают, когда необходима агрегация большого числа данных в некоторые моменты времени. Во многих случаях гистограммы более информативны, чем, например, среднее значение исследуемого параметра. Области, где ГВР полезны, включают экономику, мониторинг окружающей среды и так далее.

Таким образом, на основе применения ЧВА может быть реализован важный аспект обработки многомерных данных больших объемов,

который связан с визуальным (графическим) представлением и анализом многомерных структур данных, позволяющий изучить многомерный объект в условиях большого объема данных в условиях различных видов неопределенности. Представляется актуальным дальнейшее использование методов ЧВА для построения многомерных регрессионных зависимостей, а также для представления и обработки многомерных данных на основе гистограммных и интервальных функций распределения (*P-boxes*) [20].

Наряду с общими представлениями случайных величин своими плотностями в виде непрерывных функций, при необходимости агрегирования данных, имеет смысл рассматривать случайные величины, плотность распределения которых представляет гистограмму. Для решения подобных задач возможно также использовать гистограммы и гистограммы второго порядка [9].

Зачастую в прикладных задачах нет возможности получить точную функцию распределения случайной величины. В таких случаях задаются оценки плотности распределения как сверху, так и снизу, после чего подобные оценки удобно аппроксимировать интервальными гистограммами. Гистограмму называют интервальной, если ее функция распределения  $P(x)$  — кусочно-интервальная функция [14].

### 2.3 Понятие многомерных гистограмм

Наряду с интервальными гистограммами в рамках ЧВА предлагается использовать гистограммы второго порядка, т.е. такие гистограммы, каждый столбец которых — гистограмма [20]. Гистограмма второго порядка (ГВП) определяется как кусочно-гистограммная функция. ГВП так же, как и обычная гистограмма определяется сеткой  $\{z_i, i = 0, 1, \dots, n\}$  и набором гистограмм  $\{P_i, i = 1, 2, \dots, n\}$ . На каждом отрезке  $[z_{i-1}, z_i]$  ГВП принимает гистограммное значение  $P_i$ . Процедура построения ГВП выглядит следующим образом. Имеется ряд гистограмм  $\{Y_i, i = 1, 2, \dots, N\}$ . Каждому значению  $Y_i$  поставлена в соответствие вероятность  $p_i$ :  $\sum p_i = 1$ . Для простоты принято считать,

что гистограммы  $Y_i$  заданы на сетке  $\{z_i, i = 0, 1, \dots, n\}$  и  $Y_i$  на отрезке  $[z_{k-1}, z_k]$  принимает значение  $Y_{ik}$ . Таким образом, на каждом отрезке  $[z_{k-1}, z_k]$  имеется случайная величина  $Y_k$ , принимающая значения  $Y_{ik}$  с вероятностью  $p_i$ . Используя данные значения, на отрезке  $[z_{k-1}, z_k]$  создается гистограмма  $P_{z_k}$ .

Для осуществления численных операций над «неопределенными» переменными, заданными своими функциями плотности в виде ГВП, в условиях неопределенности, арифметика для них выглядит следующим образом. Пусть — ГВП определяются сетками  $\{v_i, i = 0, 1, \dots, n\}$ ,  $\{w_i, i = 0, 1, \dots, n\}$  и наборами гистограмм  $\{P_{x_i}\}$ ,  $\{P_{y_i}\}$ . Пусть  $Z=X*Y$ , где  $* \in \{+, -, \cdot, /, \uparrow\}$ . Гистограмма  $Z$  строится как ГВП. Задается сетка  $\{z_i, i = 0, 1, \dots, n\}$ , тогда гистограмма  $P_{z_i}$  на отрезке  $[z_{i-1}, z_i]$ , определяется по формуле (13):

$$P_{z_i} = \int_{\Omega_i} X(\xi)Y(\eta)d\xi d\eta, \quad (13)$$

где  $\Omega_i = \{(\xi, \eta) | z_i \leq \xi \cdot \eta \leq z_{i+1}\}$ .

Следует заметить, что на каждом прямоугольнике  $[v_{i-1}, v_i] \cdot [w_{j-1}, w_j]$  функция  $X(\xi)Y(\eta)$  — есть постоянная гистограмма  $P_{x_i} P_{y_i}$ . Интеграл от гистограммы по области есть значение гистограммы, умноженное на площадь области.

Далее следует пример работы гистограммной арифметики в случае сложения двух ГВП.

Пусть необходимо сложить две гистограммы второго порядка  $X$  и  $Y$ . Гистограммы  $X$  и  $Y$  порождены равномерными случайными величинами, заданными соответственно на отрезках  $[0, t_1]$  и  $[t_2, 2]$ , где  $t_1$  — равномерная случайная величина, заданная на отрезке  $[1, 2]$ ,  $t_2$  — равномерная случайная величина, заданная на отрезке  $[0, 1]$ . Результат сложения двух гистограмм представлен в виде гистограммы второго порядка  $Z$ , изображенной на рисунке 23. Носителем  $Z$  является отрезок  $[0, 4]$ , высота 1, значения плотности вероятности представлены оттенками серого.

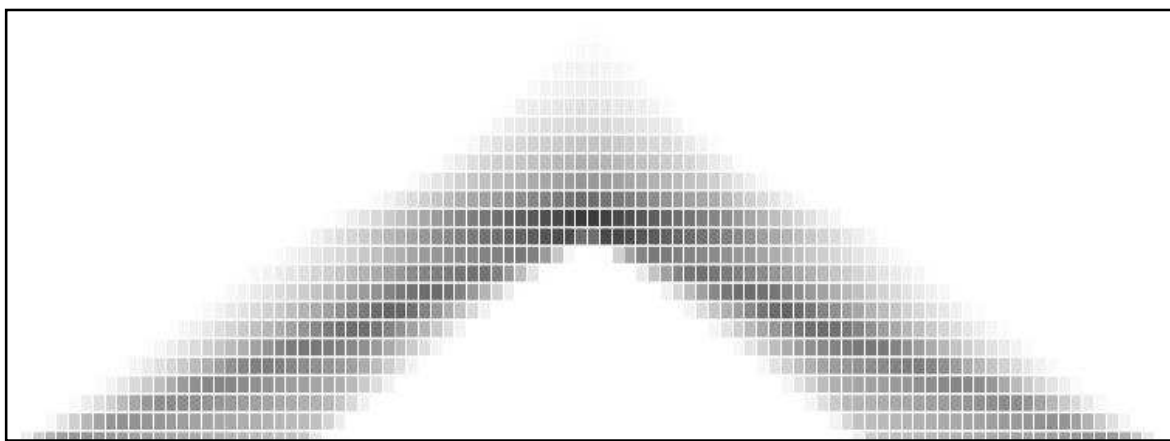


Рисунок 23 — Сумма двух гистограмм второго порядка  $Z=X+Y$

Использование гистограмм второго порядка в условиях неопределенности вероятностных характеристик параметров системы имеет широкую область применения к решению разнообразных практических задач. Например, к решению задач оценки эффективности инвестиционных проектов. В данном случае имеет место быть высокий уровень рыночной неопределенности, где стандартная финансовая модель не учитывает вероятностную природу характеристик оцениваемых показателей и соответственно не может дать достаточных оснований для принятия эффективных инвестиционных решений. Гистограммы второго порядка также могут успешно применяться в задачах оценки показателей надежности и оценки безотказной работы сложных технических систем, для изучения гидрологических и других систем [27]. Решение практических задач с использованием методов интерполяции и экстраполяции также лежит в сфере применения гистограмм второго порядка.

При численном моделировании часто приходится пользоваться различными зависимостями между входными и выходными величинами, а именно: как эмпирическими зависимостями, построенными на основе наблюдений, так и получающимися в процессе исследования. Для этого широко применяются методы интерполяции и экстраполяции. Например, задачи интерполяции возникают при подборе эмпирической формулы по данным

измерения. Здесь необходимо предостеречь от формального, слепого подбора подобной формулы только по измеренным значениям. Выбор вида формулы (многочлен, степенная функция, экспонента и т.д.) должен опираться не только на теоретическое обсуждение различных свойств изучаемой зависимости, но и на анализ имеющейся информации с точки зрения ее полноты и специфических особенностей присутствующей в ней неопределенности. Это значит проанализировать множество аспектов, связанных как с понятием «неопределенная информация», так и собственно самими численными методами, предназначенными для ее представления, обработки, моделирования, преобразования и извлечения знаний. С данной точки зрения, следует выделить два важных аспекта, которые следует иметь в виду исследователю при работе с неопределенными данными. Первый аспект связан с понятием “неопределенная информация” и проблемой изучения источников неопределенности и снижения уровня неопределенности в данных. Второй аспект связан с выбором численного метода для реализации вычислительных процедур, которые могли бы существенно помочь исследованию, а не оказаться источником ошибок.

### **3 Разработка программного модуля для визуального моделирования многомерных данных**

В ходе выполнения целей и задач научного исследования по теме: «Визуально-интерактивное представление и моделирование многомерных данных» возникло предположение о возможности представления данных большой размерности на компьютерном экране. Поводом для исследования данного вопроса послужило отсутствие компьютерных моделей, позволяющих непрерывно и в полном объеме отображать на экране монитора данные большой размерности.

#### **3.1 Концепция разрабатываемого программного модуля**

Необходимо понимать, что пространство, в котором существует человек, имеет трехмерный вид, то есть, каждую точку, лежащую в данном пространстве, возможно описать при помощи трех координат:  $x$  — длина,  $y$  — высота и  $z$  — ширина. Проблема состоит в том, что человек не способен видеть окружающее пространство в трехмерном виде так, «как оно есть». Человеческий глаз имеет возможность только строить проекцию окружающего трехмерного мира на сетчатку, которую, в свою очередь, человек и воспринимает за  $3D$  изображение объемного пространства вокруг себя. Другими словами, человек способен видеть только две координаты:  $x$  и  $y$ , третью координату  $z$  мозг рассчитывает самостоятельно, исходя из других параметров проекции видимого пространства, физических законов и самостоятельного опыта. Часто бывает так, что, выполняя различные практические задачи, человеку необходимо изучать какой-либо объект. Проблема в том, что исследуемый объект, как и пространство, в котором он находится, имеет трехмерную сущность. Следовательно, изучая, данный объемный объект, человеку необходимо знать, что происходит с объектом не только на видимом его участке, но и на скрытых от взора

исследователя (сбоку, слева, внутри). Как пример подобной практической задачи, возможно представить получение и отображение данных, часто используемое в медицине и других областях человеческой деятельности, называемое томографией. Томография, простыми словами, представляет собой получение послойного изображения внутренней структуры изучаемого объекта. Пример подобного способа отображения данных представлен на рисунке 24 на так называемых томограммах — последовательно сделанных срезах исследуемого объекта (в данном случае, человеческого мозга).

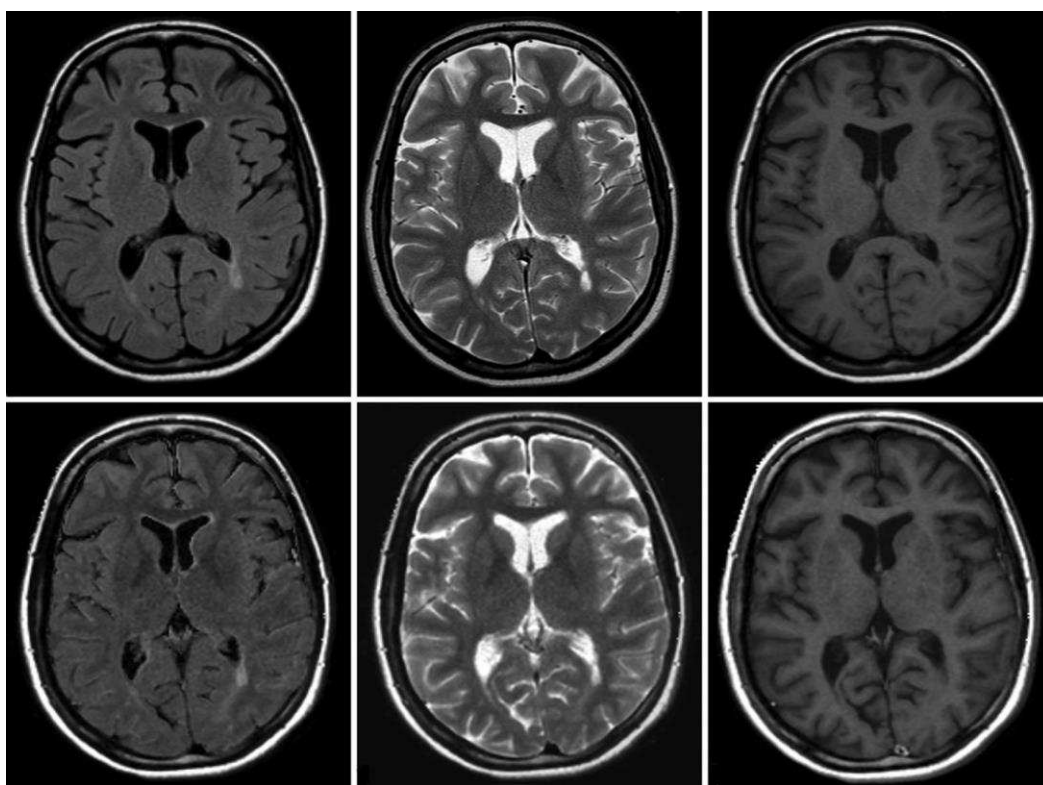


Рисунок 24 — Томограммы человеческого мозга

Используя данный вид получения и отображения данных, исследователь имеет возможность изучения внутренней составляющей объекта на статичных изображениях с помощью срезов, последовательно выполненных на одной из координатных осей через определенный интервал.

Однако, необходимо понимать, что данный вид представления данных имеет ряд недостатков, главным их которых является отсутствие возможности динамичного отображения данных. Другими словами, из-за того, что данные



представлены статично, нет возможности изучить объект с разных точек, увеличить интересующие участки или взглянуть на объект в целом, изменить ось среза и так далее.

Разрабатываемый программный модуль визуального моделирования многомерных данных предназначен прежде всего для того, чтобы на практике проверить возможность представления многомерных данных и данных больших объемов таким образом, чтобы отображаемый объем данных был комфортен и пригоден для исследования в рамках используемой компьютерной модели.

### **3.2 Технологии, применяемые для реализации программы визуализации многомерных данных**

Для реализации поставленных задач был выбран следующий стек технологий: язык *C#*, среды разработки *Visual Studio* и *Unity*. Данный выбор обусловлен обширной общественной поддержкой, которая выражается большим количестве обучающих материалов.

Язык программирования *C#* разрабатывался в конце XX века компанией *Microsoft* и позиционировался как основной язык разработки приложений для платформы *Microsoft .NET Framework*, является объектно-ориентированным и мультипарадигмальным языком. Основой для создания языка *C#* послужили наиболее признанные и популярные языки *C++* и *Java*, а также *Delphi*, *Smalltalk* и *Modula*. Таким образом, взяв за основу зарекомендовавшие себя с лучшей стороны черты лучших языков своего времени, разработчик *C#* Андерс Хейльсберг создал усовершенствованный язык, наполненный нововведениями и лишенный множества недостатков, присущих его предшественникам. Наиболее четкая связь прослеживается с *C*, у которого *C#* позаимствовал непосредственно синтаксис, многие операторы и ключевые слова, а значительно усовершенствованная объектная модель досталась от *C++*. Следует упомянуть, что хоть *C#*, как и *Java*, предназначен для получения переносимого кода, но прямым последователем *Java* не является. Главным образом, преимуществом

языка *C#* является его простота и доступность для обучения, влекущая за собой большое количество обучающих материалов, богатых коллекций библиотек, что обеспечивает возможность быстрой разработки различных приложений.

Линейка продуктов от компании *Microsoft*, коим является *Visual Studio*, представляет собой интегрированную среду разработки различного рода программного обеспечения и ряд других инструментальных средств. В данном пакете программ поддерживается реализация языка *C#* от корпорации *Microsoft*, а именно, *Visual C#*, которая включает в себя полнофункциональный редактор кода, фреймворк, компилирующий и выполняющий написанные программы, шаблоны проектов, конструкторы, отладчик и множество другого полезного при разработке программного обеспечения инструментария. Дополнительным преимуществом среды *Visual Studio* является то, что помимо использования стандартных инструментов, имеется возможность создания и подключения сторонних средств, что помогает функционально расширять возможности для разработки приложений практически на каждом уровне системы.

Поскольку имелась необходимость визуализации многомерных функций, появилась необходимость использования интегрированной среды разработки, направленной и ориентированной на создание трехмерных приложений и игр. Выбор данного инструментария был сделан в пользу среды *Unity*, продукта компании *Unity Technologies*. Впервые данный игровой движок был представлен общественности в 2005 году, на текущий момент среда разработки *Unity* имеет полноценную пятую версию, включающую весь необходимый для разработки интерактивных приложений с графикой, обрабатываемой в реальном времени, функционал, а именно: визуализатор, физический движок, систему выполняемых скриптов, анимации, ИИ, управление памятью и многопоточность. Преимуществами данной среды разработки являются:

- использование языка *C#*, что позволяет снизить временной порог вхождения в процесс работы с данным продуктом;

- кроссплатформенность, то есть возможность перенесения программного кода, написанного на визуализаторе *Unity*, на другие платформы с минимальными изменениями;

- поддержка разработчиком и *Community*, что означает достаточное количество подробного справочного материала и наличие официальной службы поддержки, оперативно реагирующей на возникающие вопросы.

Совместное использование представленного ранее инструментария позволило разработать программный модуль, позволяющий визуализировать и моделировать трехмерные функции для дальнейшего проведения качественного визуального анализа.

### **3.3 Описание структуры и основных классов программного модуля**

Разработанный программный модуль визуально-интерактивного моделирования и представления многомерных данных предназначен для повышения качества исследования представления многомерных данных в зависимости от размерности и объема. Для обеспечения повышения качества результатов визуального анализа при разработке программного модуля учитывались следующие требования:

- интуитивно понятный интерфейс (быстрота освоения пользователем функционала программы, минимальный объем настраиваемых параметров, требуемых для положительного результата визуального анализа данных);

- результативность решения поставленных задач пользователя (возможности программы позволяют решать поставленные задачи в процессе визуального анализа моделируемых многомерных данных);

- удобный доступ к полной функциональности;

- способность к модернизации (открытый программный код, дающий возможность свободного расширения функционала программы);

- автономность (нет привязки к рабочему месту);

- кроссплатформенность (возможность стабильного функционирования программного модуля на различных операционных системах);

- однозначность апробирования результатов работы программы;

Исходя из поставленных задач, сформирована структура программной реализации модуля визуализации данных, представленная на рисунке 25.

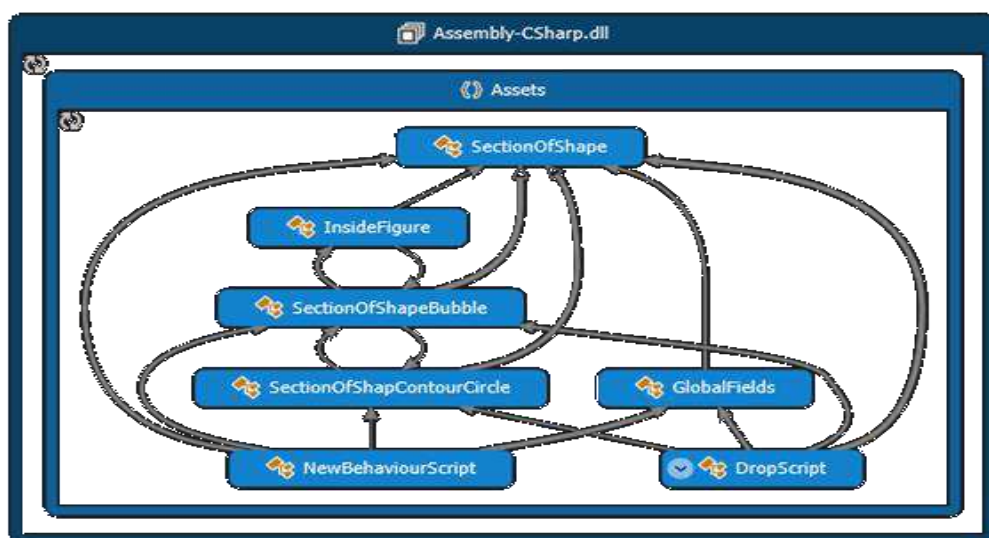


Рисунок 25 — Структура программного модуля

Структура программного модуля состоит из ряда основных и вспомогательных классов. В основных классах, представленных в подразделе 3.3.1, произведено описание и реализация основных действий над объектами. В подразделе 3.3.2 описываются классы управления, реализующие второстепенные функции.

### 3.3.1 Основные классы

В структуре программного модуля представлено 5 основных классов.

Класс *SectionOfShape* является абстрактным классом, который содержит общие описания плоскостей двух типов:

- срез в виде точечной сетки;
- срез в виде линий уровня.

Кроме того, данный класс реализует основные действия над плоскостью, такие как: создание плоскости и осуществление поворотов. Решение задач класса достигается выполнением следующих процедур:

1) Процедура *Create* (приложение А.1) в данном классе необходима для создания области отображения данных (среза), предназначенной для описания и хранения плоскостей в виде точечной сетки, либо линий уровня.

2) За задание позиции среза в пространстве отвечает процедура *SetPositionOffset*. Координатная ось движения среза определяется с помощью процедуры *SideRotateChange*.

Следующий основной класс *SectionOfShapeBubble* реализует срезы с точечной сеткой: создаёт точки, регулирует прозрачность точек на плоскости, контролирует цвет каждой точки и остающийся за ней шлейф в соответствии с положением точки в кластере.

1) Процедура *OpacityChane*, предназначенная для изменения прозрачности точек в сетке, предоставляет пользователю возможность анализировать пространство, скрытое ранее ввиду сплошного окраса точек.

2) Эффект изменения оттенка каждой отдельно взятой точки, в зависимости от удаления от центра фигуры, достигается применением процедуры *getColorForCylinder* (приложение А.2). Цвет точки меняется от красного (центр фигуры) к синему (край фигуры).

3) Процедура *SetColorBubbles* задает цвет всем точкам, другими словами, состоит из многочисленных итераций процедуры *getColorForCylinder*. Число итераций определяется количеством точек в сетке.

4) Время отображения точек, находившихся ближе к центру фигуры, определяется функцией *getTimeStartLife* (приложение А.3). Точки, имеющие синий оттенок отображаются меньшее количество времени.

5) Сохранение текущих параметров динамического среза осуществляется процедурой *putBools*.

6) Процедура *RepleceType* (приложение А.4) сохраняет созданные ранее срезы и выбранные параметры при изменении режима визуализации,

также происходит передача данных о срезе и их параметрах с изменением режима.

Класс *SectionOfShapContourCircle* реализует работу с линиями уровня: создаёт их, контролирует текущий размер линий.

1) Процедура *SetSizeCircle* (приложение А.5) производит изменение размера линий в зависимости от положения в пространстве, на середине оси движения линия имеет максимальный размер.

2) За задание позиции линий уровня в пространстве отвечает процедура *SetPositionOffset*. Координатная ось движения среза определяется с помощью процедуры *SideRotateChange*.

3) Класс *InsideFigure* контролирует отображение в сцене точек, которые находятся внутри фигуры.

4) Процедура *OnTriggerEnter* (приложение А.6) является перегрузкой стандартной функции из *Unity*, если объект (в терминах *Unity* частица визуального эффекта) входит в область сферы, то он отображается.

5) Если объект не входит в область фигуры, но он становится прозрачным (зануляется его альфа-канал), данное действие реализуется процедурой *OnTriggerExit* (приложение А.7).

Класс *GlobalFields* хранит в себе глобальные переменные такие как список созданных секций (*Sections*) и движущуюся секцию (*MovingSection*).

### **3.3.2 Классы управления**

За реализацию второстепенных функций, позволяющих расширять возможности программного модуля, отвечают классы управления.

Класс *NewBehaviourScript* осуществляет работу с динамическим срезом. Реализованы функции *LateUpdate* (приложение А.8) и *LookAtTarget* (приложение А.9), позволяющие управлять обзорной камерой. Переменные, представленные в приложении А.10 отслеживают положение камеры обзора. Основные исполнительные процедуры данного класса представлена далее:

1) Процедура *ValueChangeCheck* устанавливает цвета на основании отдалённости от центра фигуры при движении секции, изменяет либо цвета точек в сетке, либо размеры линий уровня в зависимости от выбранного режима визуализации, является управляющей процедурой, вызывающей другие функции, изменяет цвета плоскости в зависимости от ее положения в контексте динамического среза.

2) Изменения плоскости среза (оси *XYZ*) вызываются процедурой *SideSectionChange* (приложение А.12).

3) Включение отображения сетки только внутри кластера (фигуры) осуществляется процедурой *ToggleEnableCluster* (приложение А.13).

4) Изменение размера сетки (количества точек) только в динамическом срезе выполняется процедурой *ChangeCountElements* (приложение А.14).

5) Процедурой *ChangeToggleOfmodeSection* выполняется переход от сетки точек к линиям уровней (Приложение А.15).

Класс *DropScript* осуществляет работу со списком созданных плоскостей, за данный функционал отвечают следующие процедуры:

1) Процедура *ChangeCountElements* изменяет количество точек в сетке (во всех существующих срезах).

2) Отключение ненужных настроек при переходе к другому режиму выполняется процедурой *ChangeToggleOfmodeSection*.

3) Изменение прозрачности точек в выбранной плоскости происходит с помощью процедуры *ChangeOpasitySelectedSection*.

4) Процедура *ToggleEnableCluster* отображает только точки, находящиеся в кластере фигуры.

5) Процедура *Update* (приложение А.16) выполняется постоянно, после включения процедуры *Animation* (приложение А.17) сдвиг плоскости начинает отображаться, до того момента пока переключатель анимации находится в положении «true» (булева переменная). Процедура *Animation* устанавливает плоскость среза на край области отображения и начинает анимацию, двигая плоскость до противоположного края, после чего повторяет цикл.

### 3.3.3 Ключевые объекты реализации программного модуля визуального представления многомерных данных

В классе *SectionOfShapeBubble* представлено описание формулы задания цвета для плоскостей срезов в виде сетки точек (Приложение А.18), в котором изначально высчитывается расстояние между центром фигуры и точкой, для которой определяется цвет, дальше определяется процент относительно максимальной дистанции (от центра до края фигуры), затем следует перевод процентов в необходимый для дальнейших действий формат, после чего происходит задание цвета.

Необходимо, также, подробнее рассмотреть работу программы со срезами плоскостей фигуры. Срезы хранятся в классе *GlobalFields* в переменной *Sections*, далее добавляются в класс *DropScript* в переменную *AddNewOption*, удаляются процедурой *DeleteOption*. В тот момент, когда изменятся режим отображения, выполняется процедура изменения объектов в соответствии с выбранным режимом, функция *ChangeToggleOfmodeSection* выполняет данный процесс. Изменение прозрачности выбранной секции происходит только режиме сетки точек, при этом используется функция *ChangeOpasitySelectedSection*. Изменение количество точек в сетке всех срезов регулируется посредством выполнения функции *ChangeCountElements*.

Изменение оси движения динамической секции происходит в классе *NewBehaviourScript*. Отправной точкой в задаче изменения оси движения динамической среза служит функция *SideSectionChange*. В функции, в соответствии с выбранными параметрами, у объекта, который является динамическим срезом, изменяется свойство *SideRotate*. Изменение данного свойства вызывает функцию *SideRotateChange*, которая производит поворот и позиционирование динамического среза.

На каждую точку сетки добавляется компонент, который отслеживает нахождение точки в фигуре, а именно, пользовательский компонент *InsideFigure*. Данный компонент программного модуля использует стандартные возможности



среды *Unity* — выявление коллизии объектов: *OnTriggerEnter* и *OnTriggerExit*. Если точка находится внутри фигуры, то происходит окрашивание в соответствии с ее положением посредством функции *SetColorBubbles*.

При использовании линий уровня выяснилось, что в стандартных компонентах *Unity* нет понятия "Круг", поэтому появилась необходимость его создания и загрузки в приложение. Объект круга был создан в программе *Blender*. Загрузка объекта осуществляется в конструкторе класса *SectionOfShapContourCircle* стандартными средствами загрузки сторонних компонентов *Resources.Load*.

Изменение размеров кругов в соответствии с положением динамического среза происходит в классе *SectionOfShapContourCircle* в функции *SetSizeCircle*.

Для того, чтобы шлейф, идущий за движущимся срезом, исчезал после пересечения границы сферы, минимальное пороговое значение (0) сравнивается с текущим значением положения среза, при превышении порогового значения текущим, происходит прекращение создания шлейфа. Для создания возможности отображения только тех объектов, которые находятся внутри фигуры, используются стандартные возможности среды *Unity*, такие как: выявление объектов, состоящих в контакте (коллизия объектов), частицы компонента *ParticleSystem* и объекта, с которым требуется выявлять столкновения. Для создания шлейфа в данной программе используется стандартный компонент *ParticleSystem*, предварительно подвергшийся тонкой настройке. Шлейф реализован в режиме сетки точек и линий уровней.

Для отображения в режиме сетки точек к каждому объекту добавляется компонент. Основными особенностями является его настройка, а именно, установка стартовой скорости частиц (*Particl*) на нуль.

В программе имеется особое поведение шлейфа, такое, как отсутствие шлейфа у точек, находящихся вне кластера. Данные требования реализованы в классе *InsideFigure*.

В случае с сеткой точек имеется необходимость изменять цвет шлейфа, что реализовано в классе *SectionOfShapeBubble*, а именно, в функции *SetColorBubbles*.

В случае с линиями уровней особенными параметрами является отрисовка только крайних точек, для этого в настройках компонента *ParticleSystem* нужно установить параметры «*Shape*» на «*Cone*» и «*Emit from*» на «*Volume Shael*», что в программном коде реализовано в процедуре *SectionOfShapContourCircle*.

### 3.4 Графическое представление интерфейса программного модуля визуализации данных

Разработанный программный модуль визуально-интерактивного представления и моделирования многомерных данных представляет собой оконное приложение, управление возможностями программы реализуется клавиатурой, компьютерной мышью, либо сенсорной панелью ноутбука. При выполнении загрузки программа предлагает пользователю выбрать наиболее оптимальные параметры работы для имеющейся конфигурации рабочего ПК. Окно выбора параметров представлено на рисунке 26.

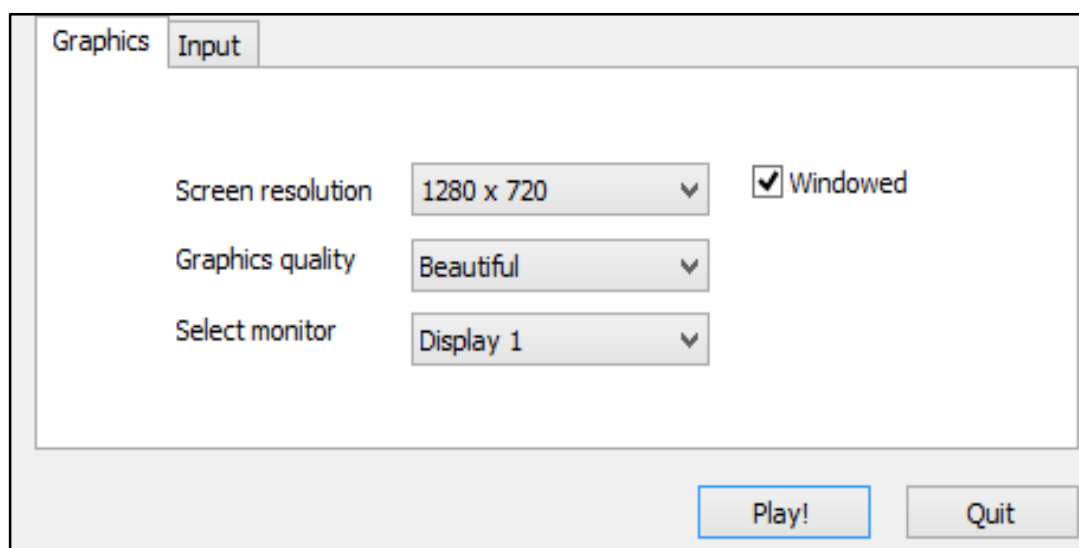


Рисунок 26 — Окно выбора параметров запуска

Рабочее поле программы, изображенное на рисунке 27, представляет собой:

1) область, предназначенную для моделирования объектов в многомерном пространстве;

2) панель настроек параметров визуально-интерактивного представления.

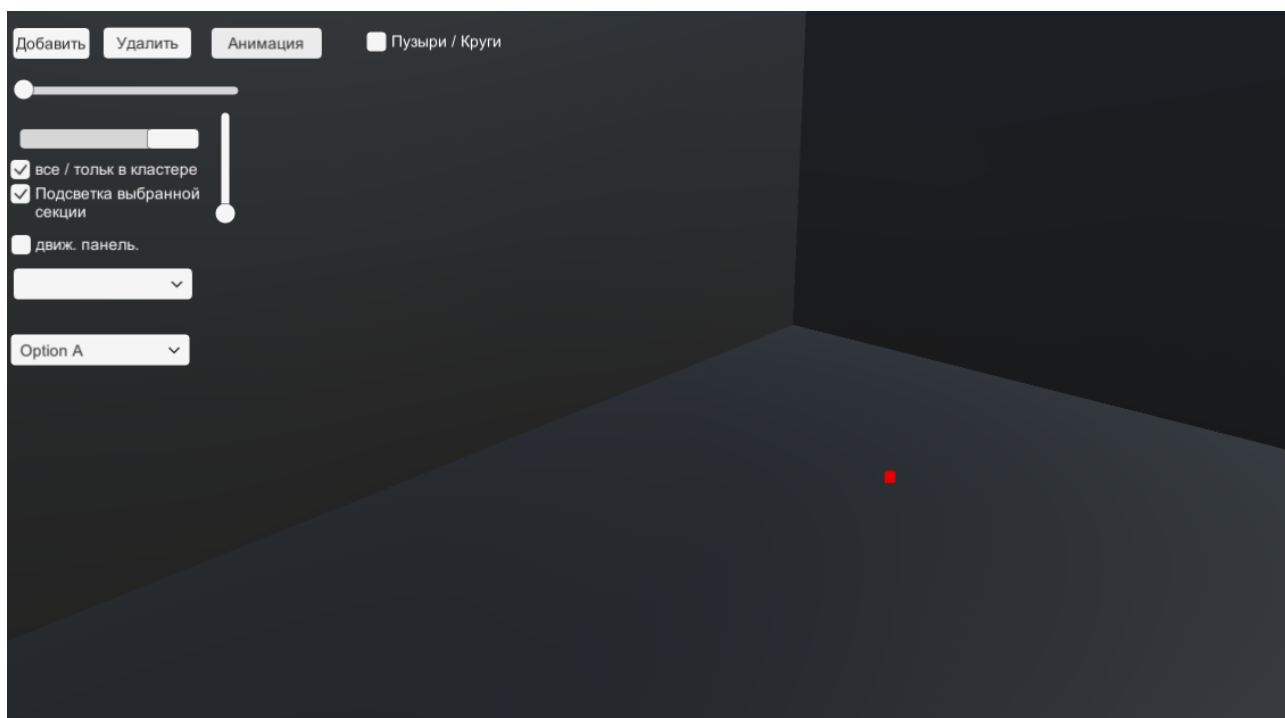


Рисунок 27 — Окно с рабочей областью и полем настроек параметров

В программном модуле возможно отображение объектов в двух режимах: срезы в виде сетки точек и линий уровней. Результат работы с программой в режиме сетки точек представлен на рисунке 28. В данном режиме возможно добавление, сохранение и удаление срезов фигуры с регулировкой прозрачности, выбором оси движения среза и настройкой размера сетки. Изменение оттенка каждой, отдельно взятой, точки сетчатого каркаса зависит от расположения одной относительно центра фигуры, чем ближе к краю фигуры находится точка, тем сильнее выражено изменение цвета точки от красного к синему, либо белому.

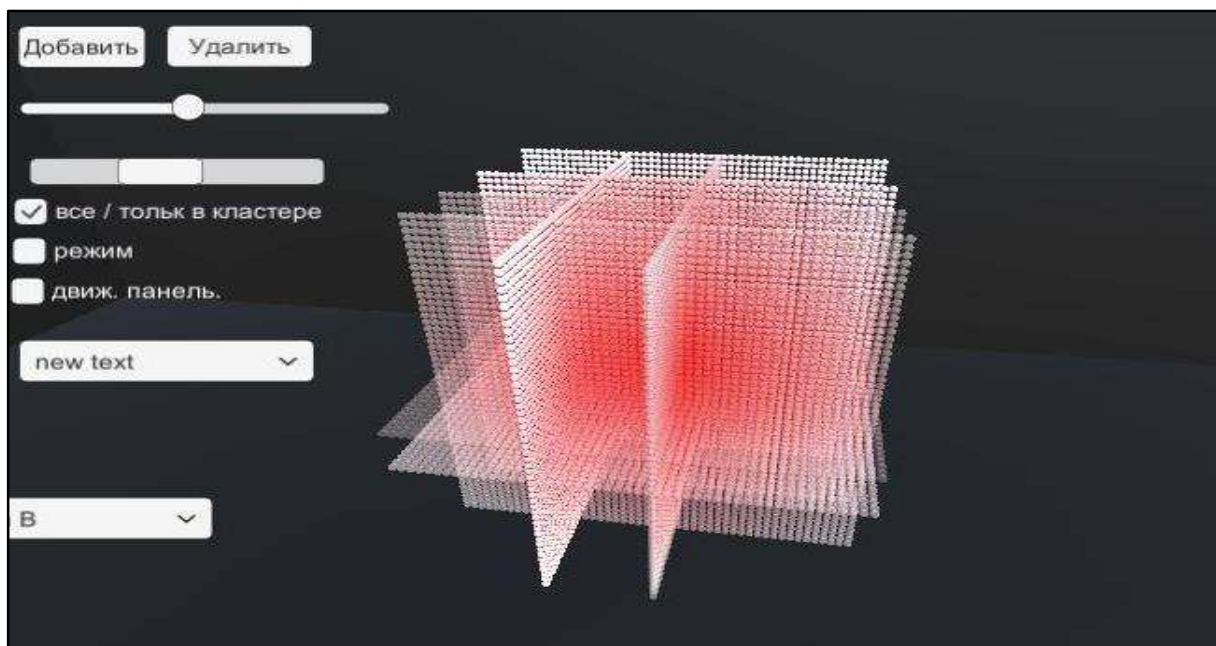


Рисунок 28 — Срезы в виде сетки точек

В режиме работы программы с линиями уровня, представленном на рисунке 29, показано отображение каркасной модели нескольких функций, представленных в виде сфер и помещенных одна в другую.

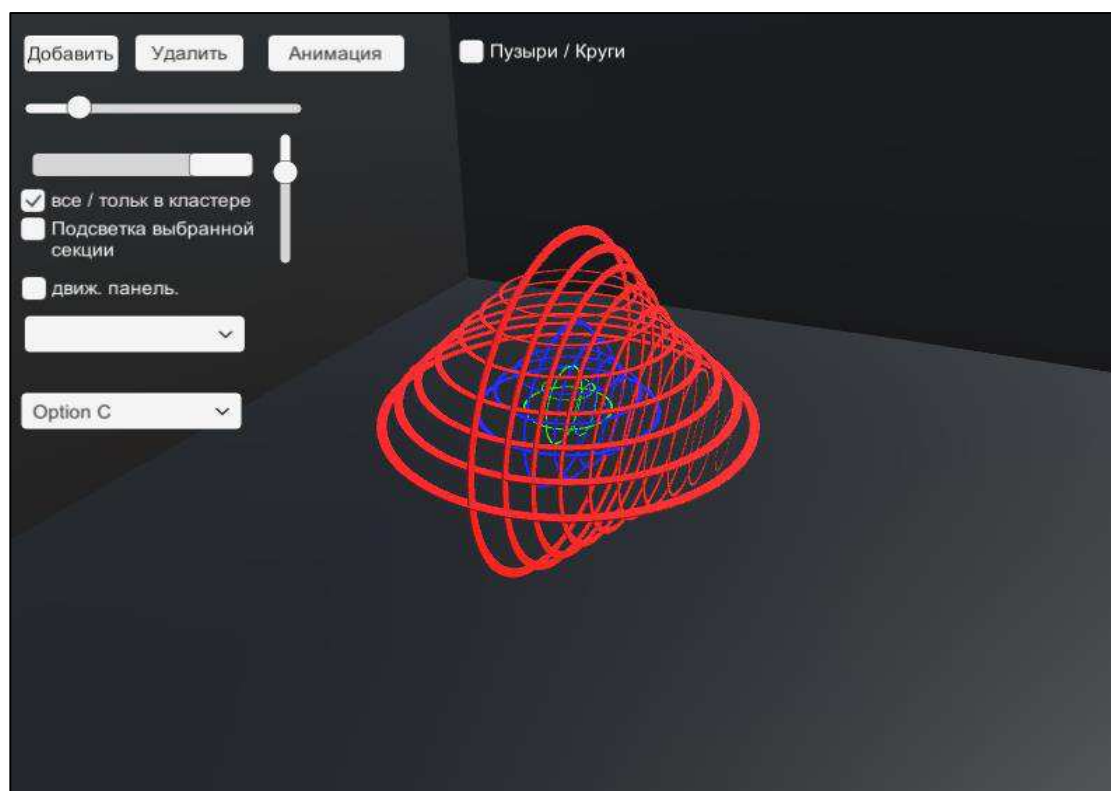


Рисунок 29 — Срезы в виде линий уровня

Также, в программном модуле визуализации данных реализована функция отображения шлейфа, следующего за двигающейся плоскостью среза. Особенность шлейфа, применительно к режиму отображения срезов в виде сетки точек, заключается в том, что точки, находящиеся ближе к центру фигуры, существуют больший отрезок времени, относительно находящихся с краю. В режиме отображения линий уровня шлейф реализован таким образом, что при движении среза с линиями, остающийся след имитирует предыдущие положения движущейся плоскости (рисунок 30).

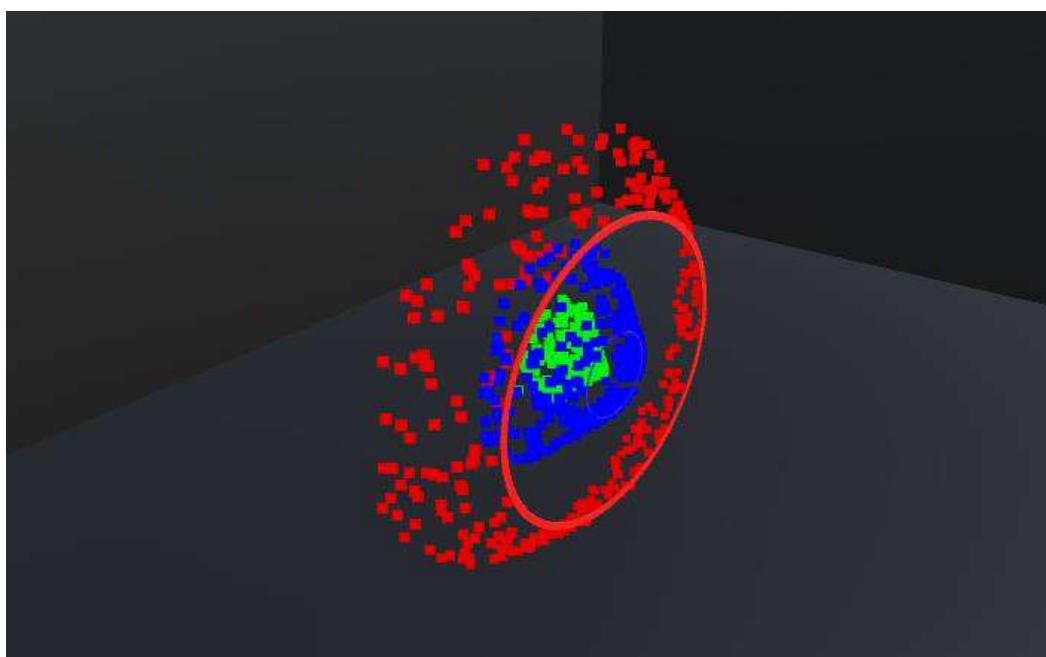


Рисунок 30 — Шлейф в режиме отображения линий уровня

Отображение шлейфа, остающегося за движущейся плоскостью, позволяет изучать визуализированный объект в динамике, без создания дополнительных статических срезов.

В программном модуле реализуется отображение объектов в режиме горизонтальной стереопары, что представлено на рисунке 31.

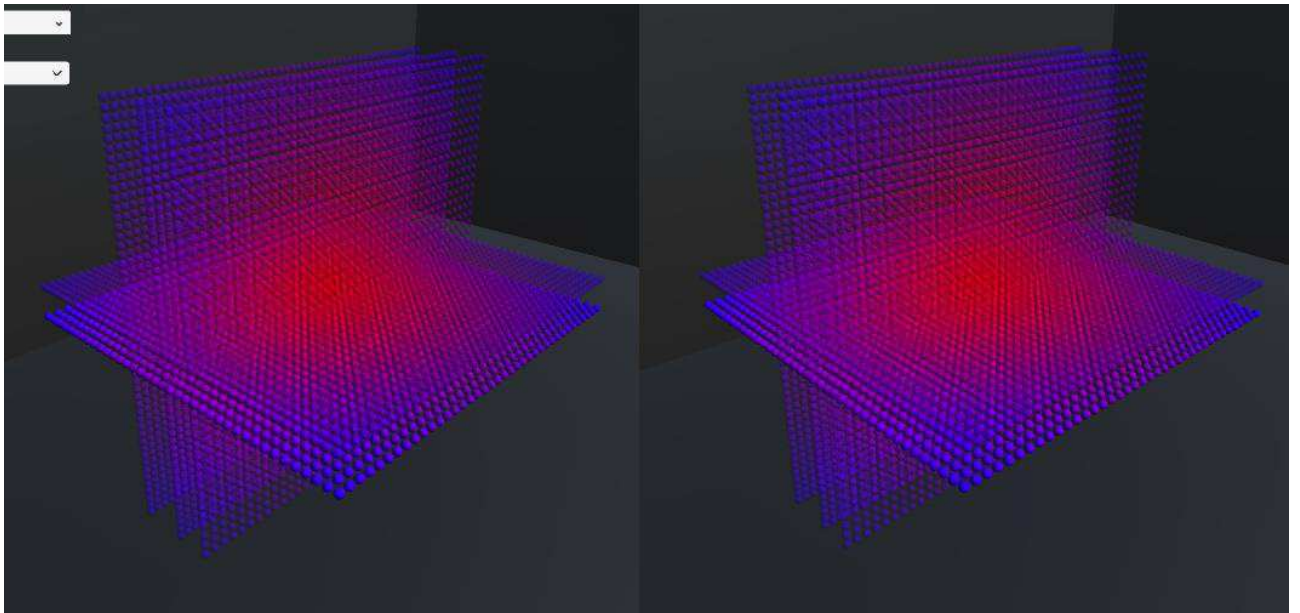


Рисунок 31 — Режим стереопары

Изображения с объектом расположены перед пользователем таким образом, чтобы сцены, включающая объект, при просмотре с разных точек (соответствующих правому и левому глазу), имели различное угловое смещение (эффект стереоскопического параллакса). При создании условий, что каждый глаз воспринимает только изображение, для него предназначенное, создаётся иллюзия наблюдения объёмной картины.

Таким образом, разработанный программный модуль обладает интуитивно понятным интерфейсом, который не требует значительного количества времени для освоения, позволяет решать поставленные задачи в процессе визуального анализа моделируемых многомерных данных и не ограничивает функциональных возможностей программы.

## ЗАКЛЮЧЕНИЕ

В ходе выполнения целей и задач исследования на тему визуального представления и анализа эмпирических многомерных данных изучены теоретические аспекты визуально-интерактивного моделирования и представления многомерных данных и данных больших объемов, обозначены основные существующие проблемы и сформулированы задачи исследования, осуществлена разработка алгоритма и программного модуля визуального моделирования многомерных данных, посредством построения динамических сечений, программный модуль протестирован на эмпирических данных.

По результатам исследований опубликованы статьи:

1. Погорелый, Е. С., Представление многомерных данных для ВИМ-технологий // Журнал «Научные исследования и разработки молодых ученых». 2015. № 7. С. 120-124.

2. Погорелый, Е. С., Проблемный анализ визуализации многомерных данных // Международная научная конференция «Молодежь и наука: проспект Свободный». 2016.

3. Погорелый, Е. С., Визуальное представление многомерных данных с использованием компьютерных моделей // Решетнёвские чтения: материалы XX Юбилейной междунар. науч.-практ. конф.: в 2 ч. / под общ. ред. Ю. Ю. Логинова ; СибГАУ — Красноярск, 2016. Ч. 2. С. 284–287.

4. Погорелый, Е. С., Визуально-интерактивное представление и моделирование многомерных данных // Международная научная конференция «Молодежь и наука: проспект Свободный». 2017.

Также, следует сказать, что в настоящее время область визуально-интерактивного представления и моделирования данных больших объемов представляет большой интерес для исследователей по всему миру. В пользу данного заключения говорит внушительное количество издаваемых публикаций, книг, монографий и пособий, проводимых конференций и представляемых на них докладов по данной тематике.

## СПИСОК СОКРАЩЕНИЙ

МД — Многомерные данные;

ГВР — Гистограммный временной ряд;

ВИМ — Визуально-интерактивное моделирование;

ИС — Информационная система;

ИИ — Искусственный интеллект;

ЛПР — Лицо, принимающее решение;

ЧВА — Численный вероятностный анализ;

ГВП — Гистограмма второго порядка.



## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Большаков, А. А., Керимов, Р. Н., Методы обработки многомерных данных и временных рядов // Москва, Горячая Линия — Телеком, 2007.
2. Бондарев, А. Е., Галактионов, В. А., Анализ и визуализация многомерных данных в задачах вычислительной газовой динамик // *Programming and Computer Software*. 2015. Т. 41. № 5. С. 247–252.
3. Бондарев, А. Е., Галактионов, В. А., Шапиро, Л. З., Анализ многомерных данных в задачах многопараметрической оптимизации с применением методов визуализации // ИПМ им. М.В. Келдыша РАН, Россия, Москва, 2012.
4. Булгаков, С. В., Агрегирование информационных моделей // *Перспективы Науки и Образования*, 2014, №3(9), С.9–13.
5. Васильев, В. Р., Волобой, А. Г., Вьюкова, Н. И., Галактионов, В. А. Контекстная визуализация пространственных данных // *Препринты ИПМ им. М. В. Келдыша*. 2004. № 56. 23 с.
6. Горбань, А. Н., Методы нейроинформатики // КГТУ, Красноярск, 1998.
7. Добронеец, Б. С., Попова, О. А., Гистограммный подход к представлению обработке данных космического и наземного мониторинга // *Известия ЮФУ. Технические науки*. 2014. № 6 (155). С. 14–22.
8. Добронеец, Б. С., Попова, О. А., Представление и обработка неопределенности на основе гистограммных функций распределения и P-boxes // *Информатизация и связь*. 2014. № 2. С. 23–26.
9. Добронеец, Б. С., Попова, О. А., Численный вероятностный анализ неопределенных данных: монография // Сибирский федеральный университет, Красноярск, 2014 — 168 с.
10. Дугина, Т. О., Туннельная модель как способ представления совокупностей простых и многомерных данных // *Известия высших учебных заведений. Поволжский регион. Технические науки*, 2015. № 3 (35). — С. 5-14.

11. Зиновьев, А. Ю., Визуализация многомерных данных // КГТУ, Красноярск, 2000. 180 с.
12. Кудж, С. А., Цветков, В. Я., Информационные образовательные единицы // Дистанционное и виртуальное обучение, 2014. №1. С.24–31.
13. Масленников, О. П., Мильман, И. Е., Сафиуллин А. Э., Бондарев, А. Е., Низаметдинов, Ш. У., Пилюгин, В. В., Разработка и развитие системы интерактивного визуального анализа многомерных данных // Труды 25-й Международной Конференции по Компьютерной Графике и Зрению ГрафиКон'2015, Протвино, Россия, 22–25 сентября 2015 г., С. 227–231.
14. Методы создания стереоскопических изображений: Электронный ресурс URL: <http://briefeducation.ru>
15. Муха, В. С., Анализ многомерных данных: проблемы, состояние, перспективы // Доклады БГУИИР, п. Бровки, 6, Минск, р. Беларусь, 2003.
16. Першина, Е. Л., Попова, О. А., Чуканов, С. Н., Интеллектуальные системы поддержки принятия решений: комплексы программ, модели, методы, приложения // Федеральное агентство по образованию, ГОУ ВПО «Сибирская государственная автомобильно-дорожная акад. (СибАДИ)». Омск, 2010.
17. Погорелый, Е. С., Визуальное представление многомерных данных с использованием компьютерных моделей // Решетнёвские чтения: материалы XX Юбилейной междунар. науч.-практ. конф., посвящ. памяти генерального конструктора ракетно-космических систем академика М. Ф. Решетнева: в 2 ч. / под общ. ред. Ю. Ю. Логинова ; Сиб. гос. аэрокосмич. ун-т. — Красноярск, 2016. Ч. 2. С. 284–287.
18. Погорелый, Е. С., Представления многомерных данных для ВИМ-технологий // Журнал «Научные исследования и разработки молодых ученых». 2015. № 7. С. 120–124.
19. Погорелый, Е. С., Проблемный анализ визуализации многомерных данных // Международная научная конференция «Молодежь и наука: перспектив Свободный». 2016.

20. Попова, О. А., Гистограммный информационно-аналитический подход к представлению и прогнозированию временных рядов // Информатизация и связь. 2014. № 2. С. 43–47.
21. Попова, О. А., Гистограммы второго порядка для численного моделирования в задачах с информационной неопределенностью // Известия ЮФУ. Технические науки, 06.2014.
22. Попова, О. А., Технология извлечения и визуализации знаний на основе численного вероятностного анализа неопределенных данных // Информатизация и связь. 2013. № 2. С. 63–66.
23. Попова, О. А., Численный вероятностный анализ для агрегации, регрессионного моделирования и анализа данных // Информатизация и связь. 2015. № 1. С. 15–21.
24. Попова О. А., Численный вероятностный анализ для оптимизационных задач гидроэнергетики // Известия ИрГУ, 2015. Т. 12. Серия «Математика». С.79–92.
25. Попова, О. А., Першина, Е. Л., Чуканов, С. Н., Интеллектуальные системы поддержки принятия решений: комплексы программ, модели, методы, приложения // Федеральное агентство по образованию, ГОУ ВПО "Сибирская гос. автомобильно-дорожная акад. (СибАДИ)". Омск, 2011.
26. Тихонов, А. Н. Концепция сетецентрического управления сложной организационно-технической системой // Иванников, А. Д., Соловьёв, И. В., Цветков, В. Я., Кудж, С. А. — М.: МаксПресс, 2010. 136 с.
27. Шрамм, Ф. К., Формозу, К.Т., Использование визуального интерактивного моделирования с использованием ПО по совершенствованию процесса принятия решений в системе обеспечения производства.
28. Цветков, В. Я. Информатизация: Создание современных информационных технологий. Часть 1. Структуры данных и технические средства // М., ГКНТ, ВНТЦентр, 1990. 118 с.
29. Dobronets, B. S., Krantsevich, A. M., Krantsevich, N. M. Software implementation of numerical operations on random variables // Журнал Сибирского

федерального университета. Серия: Математика и физика. 2013.  
Т. 6. № 2. С. 168–173.

30. Piatetsky-Shapiro, G., Machine Learning, Data Mining, and Knowledge Discovery: An Introduction // AAAI/MIT Press, 1996.

31. Scott, D. W., Multivariate density estimation: theory, practice and visualization // Rice University, Houston, Texas, 1993.

## ПРИЛОЖЕНИЕ А

### Программный код модуля визуализации

#### Листинг А.1 — Процедура *Create*

```
public void Create()
{
    Section = GameObject.CreatePrimitive(PrimitiveType.Cube);
    Section.transform.localScale = new Vector3(50, 50, 2);
    SideRotateChange();
    Section.GetComponent<Renderer>().material.shader =
    Shader.Find("Transparent/Diffuse");
    Section.GetComponent<Renderer>().material.color = new Color(1, 0,
    0, 0);
}
```

#### Листинг А.2 — Процедура *getColorForCylinder*

```
public Color getColorForCylinder(Vector3 positionShare)
{
    float dist = maxDist -
    Vector3.Distance(MainCube.transform.position, positionShare);
    dist = dist < 0 ? 0 : dist;
    float percentRED = dist / (maxDist / 100);
    float www = (100 - percentRED) / 100;
    Color result = new Color(1 - www, 0, www, Opacity);
    return result;
}
```

#### Листинг А.3 — Функция *getTimeStartLife*

```
float getTimeStartLife(Vector3 positionShare, Vector3
positionMainCube)
{
    float startLife = 0.5f;
    float maxDist = 43.30127f;
    float wdwd = Vector3.Distance(positionMainCube, positionShare);
    float dist = maxDist - wdwd;
```

```

dist = dist < 0 ? 0 : dist;
float percentRED = dist / (maxDist / 100);
float dwqs = 2f / 100f;
startLife += dwqs * percentRED ;
return startLife;
}

void putBools()
{

float CountElements = CountBubles - 2;
float factor = MaxElements / CountElements;
Vector3 centerPanelSection = Section.transform.position;
for (float x = 0; x <= CountElements; x++)
{
for (float y = 0; y <= CountElements; y++)
{
Vector3 Dot = new Vector3();
switch (_sideRotate)
{
case 0:
{
Dot = new Vector3(
x * factor + centerPanelSection.x - (MaxElements * 0.5f),
y * factor + centerPanelSection.y - (MaxElements * 0.5f),
centerPanelSection.z);
}
break;
case 1:
{
Dot = new Vector3(
centerPanelSection.x,
x * factor + centerPanelSection.y - (MaxElements * 0.5f),
y * factor + centerPanelSection.z - (MaxElements * 0.5f));
}
break;
case 2:

```

```

{
Dot = new Vector3(
x * factor + centerPanelSection.x - (MaxElements * 0.5f),
centerPanelSection.y,
y * factor + centerPanelSection.z - (MaxElements * 0.5f));
}
break;
default:
break;
}

```

#### Листинг A.4 — Процедура *RepleceType*

```

public SectionOfShapContourCircle RepleceType()
{
SectionOfShapContourCircle sdwdw = new
SectionOfShapContourCircle(MainCube, Claster, CenterPanelSection,
SideRotate, EnableClaster);
return sdwdw;
}

```

#### Листинг A.5 — Процедура *SetSizeCircle*

```

public void SetSizeCircle()
{
float Value = Vector3.Distance(RedCircle.transform.position,
MainCube.transform.position);
float ValueRedCircle = 25f - Value;
float ValueBlueCircle = 12.5f - Value;
float ValueGreenCircle = 6.25f - Value;
float OnePercentValueRedCircle = 25f / 100f;
float OnePercentValueBlueCircle = 12.5f / 100f;
float OnePercentValueGreenCircle = 6.25f / 100f;
float PercentRedCircle = ValueRedCircle / OnePercentValueRedCircle;
float PercentBlueCircle = ValueBlueCircle /
OnePercentValueBlueCircle;
float PercentGreenCircle = ValueGreenCircle /
OnePercentValueGreenCircle;
}

```

```

float OnePercentSizeRedCircle = 0.5f / 100f;
float OnePercentSizeBlueCircle = 0.25f / 100f;
float OnePercentSizeGreenCircle = 0.125f / 100f;
float ScaleRedCircle = PercentRedCircle * OnePercentSizeRedCircle;
float      ScaleBlueCircle      =      PercentBlueCircle      *
OnePercentSizeBlueCircle;
float      ScaleGreenCircle      =      PercentGreenCircle      *
OnePercentSizeGreenCircle;

RedCircle.transform.localScale = new Vector3(ScaleRedCircle,
ScaleRedCircle, ScaleRedCircle);

var ps = BlueCircle.GetComponent<ParticleSystem>();
var psg = GreenCircle.GetComponent<ParticleSystem>();

if (ScaleBlueCircle < 0)
ps.Stop();
else ps.Play();

if (ScaleGreenCircle < 0)
psg.Stop();
else psg.Play();

ScaleBlueCircle = ScaleBlueCircle < 0 ? 0 : ScaleBlueCircle;
ScaleGreenCircle = ScaleGreenCircle < 0 ? 0 : ScaleGreenCircle;

BlueCircle.transform.localScale = new Vector3(ScaleBlueCircle,
ScaleBlueCircle, ScaleBlueCircle);

GreenCircle.transform.localScale = new Vector3(ScaleGreenCircle,
ScaleGreenCircle, ScaleGreenCircle);}

```

### Листинг А.6 — Процедура *OnTriggerEnter*

```

void OnTriggerEnter(Collider other)
{
if (parent != null)
{
IsInside = true;

this.GetComponent<Renderer>().material.color =
((SectionOfShapeBubble)parent).getColorForCylinder(this.gameObject
.transform.position);}}

```



### Листинг А.7 — Процедура *OnTriggerExit*

```
void OnTriggerExit(Collider other)
{
    IsInside = false;
    this.GetComponent<Renderer>().material.color = new Color(1, 0, 0, 0);}
}
```

### Листинг А.8 — Процедура *LateUpdate*

```
void LateUpdate()
{
    float input = Input.GetAxis("Mouse ScrollWheel");
    if (input != 0)
    {
        cam_holder.fieldOfView *= 1 - input;
    }
    if (Input.GetMouseButton(1))
        _rotX += Input.GetAxis("Mouse X") * mouse_sens;
    _rotY -= Input.GetAxis("Mouse Y") * mouse_sens;
    LookAtTarget();}
    if (Input.GetMouseButton(2))
    {
        x_axis = Input.GetAxis("Mouse X") * mouse_sens;
        y_axis = Input.GetAxis("Mouse Y") * mouse_sens;
        cam_holder.transform.Rotate(Vector3.up, x_axis, Space.World);
        cam_holder.transform.Rotate(Vector3.right, y_axis, Space.Self);
    }
}
```

### Листинг А.9 — Процедура *LookAtTarget*

```
void LookAtTarget()
{
    Quaternion rotation = Quaternion.Euler(_rotY, _rotX, 0);
```

```

transform.position = target.position - (rotation * _offset);
transform.LookAt(target);
}

```

#### Листинг A.10 — Смещение камеры

```

target = MainCube.transform;
_rotY = transform.eulerAngles.y;
_rotX = transform.eulerAngles.x;
_offset = target.position - transform.position;

```

#### Листинг A.11 — Процедура *ValueChanged*

```

public void ValueChangeCheck()
{
    Debug.Log(SliderOfNewSection.value);
    if (MainCube.GetComponent<GlobalFields>().MovingSection != null)
    {
        MainCube.GetComponent<GlobalFields>().MovingSection.SetPositionOffset(SliderOfNewSection.value);
    }
}

```

#### Листинг A.12 — Процедура *SideSectionChange*

```

private void SideSectionChange()
{
    MainCube.GetComponent<GlobalFields>().MovingSection.SideRotate = SideSection.value;
}

```

#### Листинг A.13 — Процедура *ToggleEnableCluster*

```

public void ToggleEnableCluster()
{
    Destroy(MainCube.GetComponent<GlobalFields>().MovingSection.Section);
}

```

```

MainCube.GetComponent<GlobalFields>().MovingSection.EnableCluster =
EnableCluster.isOn;

MainCube.GetComponent<GlobalFields>().MovingSection.Create();
((SectionOfShapeBubble)MainCube.GetComponent<GlobalFields>().MovingSection).CreateBubbles();
}

```

#### Листинг А.14 — Процедура *ChangeCountElements*

```

public void ChangeCountElements()
{
((SectionOfShapeBubble)MainCube.GetComponent<GlobalFields>().MovingSection).CountBubbles = SliderOfCountElements.value;

Destroy(MainCube.GetComponent<GlobalFields>().MovingSection.Section);

MainCube.GetComponent<GlobalFields>().MovingSection.Create();
((SectionOfShapeBubble)MainCube.GetComponent<GlobalFields>().MovingSection).CreateBubbles();
}

```

#### Листинг А.15 — Процедура *ToggleEnableCluster*

```

public void ChangeToggleOfmodeSection()
{
if (ToggleOfmodeSection.isOn)
{
Destroy(MainCube.GetComponent<GlobalFields>().MovingSection.Section);

MainCube.GetComponent<GlobalFields>().MovingSection = new
SectionOfShapeBubble(MainCube, Cluster, centerCube,
SideSection.value, true, 1, true);
}
else
{
Destroy(MainCube.GetComponent<GlobalFields>().MovingSection.Section);

MainCube.GetComponent<GlobalFields>().MovingSection = new
SectionOfShapContourCircle(MainCube, Cluster, centerCube,
SideSection.value, true);
}}

```

### Листинг А.16 — Процедура *Update*

```
void Update()
{
    if (haveAnimation)
    {
        respawnPrefab.GetComponent<GlobalFields>().MovingSection.SetPositionOffset(OffsetAnimation);
        if (ToggleOfmodeSection.isOn)
        {
            OffsetAnimation += 2;
        }
        else { OffsetAnimation += 0.2f; }
        if (OffsetAnimation >= 50)
        {
            haveAnimation = false;
            OffsetAnimation = 0;
        }
    }
}
```

### Листинг А.17 — Процедура *Animation*

```
public void Animation()
{
    Vector3 startVector = new Vector3(40, -435, -1014);
    respawnPrefab.GetComponent<GlobalFields>().MovingSection.Section.transform.position = startVector;
    haveAnimation = true;}
}
```

### Листинг А.18 — Описание формулы задания цвета

```
public Color getColorForCylinder(Vector3 positionShare)
{
    float dist = maxDist - Vector3.Distance(MainCube.transform.position, positionShare);
    dist = dist < 0 ? 0 : dist;
    float percentRED = dist / (maxDist / 100);
}
```

```
float www = (100 - percentRED) / 100;  
Color result = new Color(1, www, www, Opacity);  
Color result = new Color(1 - www, 0, www, Opacity);  
return result;
```





Продолжение титульного листа магистерской диссертации по теме  
«Визуальное представление и анализ эмпирических многомерных данных»

*Замечаний нет.*

Нормоконтролер

*A* 15.06.17  
подпись, дата

М. А. Аникьева

## ОТЗЫВ

руководителя по магистерской диссертации  
Погорелого Евгения Сергеевича на тему «Визуальное представление и анализ  
эмпирических многомерных данных», представленную к защите по направлению  
09.04.02 — Информационные системы и технологии

1) Магистерская диссертация на тему: «Визуальное представление и анализ эмпирических многомерных данных» выполнена по заданию кафедры «Системы искусственного интеллекта» и посвящена актуальной задаче визуального представления эмпирических многомерных данных.

В диссертационной работе рассмотрены новые подходы разработки модуля представления эмпирических многомерных данных. Рассмотрены вопросы, связанные с отображением эмпирических многомерных данных в виде динамических структур. В работе использованы современные подходы, основанные на численном вероятностном анализе.

2) В магистерской диссертации решены следующие задачи:

- изучение публикаций по данной тематике;
- проанализированы методы визуализации данных;
- разработан алгоритм представления многомерных данных в виде динамических сечений;
- разработан программный модуль построения динамических сечений и их анализа на основе разработанного алгоритма;
- программный модуль протестирован на тестовых данных.

В процессе работы над диссертацией магистрант Погорелый Е.С. принял участие в ряде научно-практических конференций. По результатам исследований опубликованы четыре статьи.

Материалы работы, посвященные отображению эмпирических многомерных данных в виде динамических структур, возможно использовать в научно исследовательской работе при анализе эмпирических многомерных функций плотности вероятности.

Магистерская диссертация оформлена в соответствии с требованиями нормативных документов СФУ.

Выпускная квалификационная работа магистранта удовлетворяет требованиям, предъявляемым к магистерским работам СФУ, и может быть оценена на «отлично», а её автор Погорелый Евгений Сергеевич заслуживает присуждения квалификации магистр по направлению — Информационные системы и технологии.

Руководитель  
к.т.н., доцент.

Занимаемая должность:  
доцент каф. СИИ ИКИТ СФУ  
М. П.

О. А. Попова

«15» 06 2017 г.



## РЕЦЕНЗИЯ

на магистерскую диссертацию студента СФУ  
Погорелого Евгения Сергеевича  
на тему «Визуальное представление и анализ эмпирических многомерных  
данных», представленную к защите по направлению 09.04.02 —  
Информационные системы и технологии по программе 09.04.02.01 —  
Информационно-управляющие системы

Актуальность обусловлена необходимостью визуального представления эмпирических многомерных данных в виде, пригодном для последующего анализа.

Новизна работы заключается в использовании визуально-интерактивного моделирования.

Структура диссертационной работы. Работа содержит 86 страниц текстового документа, состоит из введения, трех глав, заключения, список литературы содержит 31 использованный источник.

Результатом данной магистерской диссертации является разработка программного модуля визуального представления эмпирических многомерных данных. Многомерные данные представляются в виде динамических сечений.

Апробация. По результатам диссертации опубликованы четыре печатные работы.

Замечания и предложения по диссертации. В работе есть определенная доля опечаток и неточностей.

Заключение. Изложенный в работе теоретический, графический и демонстрационный материал оформлен достаточно полно, в соответствии с требованиями нормативных документов СФУ. Работа выполнена в полном объеме и на достаточном уровне, в соответствии с поставленной целью. Погорелый Евгений Сергеевич заслуживает присвоения квалификации «магистр» по направлению 09.04.02 — Информационные системы и технологии.

Оценка отлично.

Рецензент

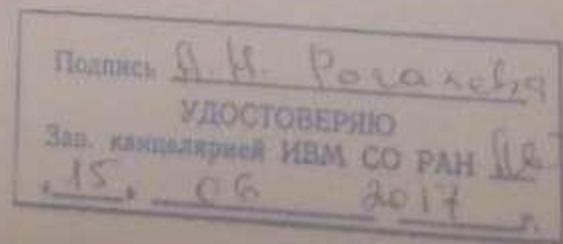
ст. науч. сотр., ИВМ СО РАН

к.ф.-м.н., доцент



А. Н. Роголев

«15» 06 2017 г.



Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
Институт космических и информационных технологий  
Кафедра систем искусственного интеллекта

УТВЕРЖДАЮ  
Заведующий кафедрой

подпись \_\_\_\_\_ инициалы, фамилия \_\_\_\_\_  
« \_\_\_\_\_ » \_\_\_\_\_ 2017 г.

**ЗАДАНИЕ  
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ  
в форме магистерской диссертации**

Студенту Погорелому Евгению Сергеевичу

Группа КИ15-02-1/1М направление (специальность): 09.04.02

Информационные системы и технологии

Тема ВКР: Визуальное представление и анализ эмпирических многомерных данных

Утверждена приказом по университету № 5072/с от 18.04.2017г.

Руководитель ВКР: О. А. Попова, доцент кафедры систем искусственного интеллекта, кандидат технических наук, Институт космических и информационных технологий Сибирского федерального университета.

Исходные данные для ВКР: диссертационные исследования, материалы с преддипломной практики, книги, научные журналы и статьи зарубежных и отечественных авторов, монографии по теме исследования.

Перечень разделов ВКР: введение, проблемный анализ темы исследования, численный вероятностный анализ в области представления многомерных данных, программная реализация модуля визуального моделирования многомерных данных, заключение.

Перечень графического материала: плакаты презентации.

Руководитель ВКР

  
\_\_\_\_\_

О. А. Попова

подпись

Задание принял к исполнению

  
\_\_\_\_\_

Е. С. Погорелый

подпись

« \_\_\_ » \_\_\_\_\_ 2017 г.



Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
Институт космических и информационных технологий  
Кафедра систем искусственного интеллекта

УТВЕРЖДАЮ  
Заведующий кафедрой

подпись \_\_\_\_\_ инициалы, фамилия \_\_\_\_\_  
«   » \_\_\_\_\_ 2017 г.

**ГРАФИК  
НАПИСАНИЯ И ОФОРМЛЕНИЯ ВЫПУСКНОЙ  
КВАЛИФИКАЦИОННОЙ РАБОТЫ  
в форме магистерской диссертации**

Студент: Погорелый Евгений Сергеевич

Группа: КИ15-02-1/1М направление (специальность): 09.04.02

Информационные системы и технологии

Тема выпускной квалификационной работы: Визуальное представление и анализ эмпирических многомерных данных

График выполнения выпускной квалификационной работы (ВКР) приведен в таблице 1.

Таблица 1 — График выполнения этапов ВКР

Наименование и содержание этапа	Срок выполнения	Примечание
Подбор литературы, ее изучение и анализ. Составление списка литературы по основным источникам	До 20 февраля 2016 г.	
Составление плана ВКР и согласование его с руководителем	До 15 марта 2016 г.	
Разработка и представление на проверку первой главы ВКР	До 20 марта 2016 г.	
Накопление, систематизация, анализ практических материалов	До 15 сентября 2016 г.	
Разработка и представление на проверку второй главы ВКР	До 20 декабря 2016 г.	
Разработка и представление на проверку третьей главы ВКР	До 10 марта 2017 г.	
Согласование с руководителем выводов и итогов ВКР	До 20 марта 2017 г.	
Доработка ВКР в соответствии с замечаниями и представление ее на кафедру	До 25 мая 2017 г.	
Разработка тезисов доклада для защиты	До 30 мая 2017 г.	
Прохождение рецензирования	До 12 июня 2017 г.	
Прохождение нормоконтроля	До 15 июня 2017 г.	
Завершение подготовки к защите с учетом рецензии	До 19 июня 2017 г.	