

Структурно-параметрический синтез интерфейса геометрического компьютера

О.А. Бабушкина ¹

¹ Санкт-Петербургский государственный университет телекоммуникаций, пр. Большевиков д. 22, корп. 1, Санкт-Петербург, 193232, Россия

Аннотация

Статья посвящена рассмотрению ряда возможных структурных и параметрических композиций, способных в совокупности составить инструмент автоматизированного геометрического проектирования, предназначенного для решения геометрических, инженерных и педагогических задач. Понятийный аппарат работы основан на трудах петербургской геометрической школы. Моделирование процесса работы геометрической машины в виде конструктивной схемы позволяет визуализировать этапы ее работы, начинающиеся с получения информации от объекта и заканчивающиеся построением его модели. Анализ существующих работ выявил три основных направления в области обработки геометрических данных: посредством ПЛИС (FPGA), графических процессоров (GPU) и микроконтроллеров. Реализация показанных конструкций в виде аналитической модели на высокоуровневом языке программирования python позволила сделать выбор наиболее подходящей из них для первой итерации реализации геометрического компьютера и спланировать дальнейшие шаги по его модернизации.

Ключевые слова

Научная визуализация; конструктивное геометрическое моделирование; геометрический эксперимент; проективная геометрия.

Structural-Parametric Synthesis of the Geometric Computer Interface

O.A. Babushkina ¹

¹ St.-Petersburg State University of Telecommunications, 22 bld. 1, Prospect Bolshhevikov, St.-Petersburg, 193232, Russia

Abstract

The article is devoted to the consideration of a number of possible structural and parametric compositions that together can form an automated geometric design tool designed to solve geometric, engineering and pedagogical problems. The conceptual apparatus of the work is based on the works of the St. Petersburg geometric school. Modeling the operation process of a geometric machine in the form of a constructive diagram allows you to visualize the stages of its work, starting with obtaining information from the object and ending with the construction of its model. An analysis of existing work has revealed three main areas in the field of processing geometric data: using FPGA, GPU or microcontrollers. The implementation of the shown structures in the form of an analytical model in the high-level programming language python made it possible to choose the most suitable of them for the first iteration of the implementation of a geometric computer and plan further steps for its modernization.

Keywords

Scientific visualization; constructive geometric modeling; geometric experiment; projective geometry.

ГрафиКон 2023: 33-я Международная конференция по компьютерной графике и машинному зрению, 19-21 сентября 2023 г., Институт проблем управления им. В.А. Трапезникова Российской академии наук, г. Москва, Россия

EMAIL: oksenyu123@yandex.ru (О.А.Бабушкина)

ORCID: 0009-0003-6043-5372 (О.А.Бабушкина)



© 2023 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

1. Введение

Увеличению точности и скорости процесса проектирования в течение долгого времени способствовало только расширение штата инженеров-проектировщиков. Развитие информационных технологий в современном мире влечет за собой прямую корреляцию экономических показателей производств и внедрения автоматизации во все сферы их организационной структуры. Разработка систем автоматизированного проектирования – прямое следствие потребности в увеличении показателей эффективности производств в области проектирования и моделирования.

В настоящее время можно выделить два вектора развития систем автоматизированного проектирования, критерием отнесения к которым служит размерность пространства объекта проектирования. Системы моделирования объектов трёхмерного пространства служат, как правило, для разработки предметных моделей. Яркими представителями систем такого рода являются AutoCAD, КОМПАС, Maya, 3D StudioMax и другие. Ко второму направлению развития автоматизации геометрического проектирования следует отнести программные продукты, позволяющие создать модели многомерных объектов, предстающие в виде работающей геометрической машины, разделяющей входные и исходящие потоки геометрических данных. Для такого рода геометрических моделей программные комплексы должны не только составлять многосвязную геометрическую конструкцию, но и обеспечивать автоматическую работу по преобразованию геометрических – в том числе и числовых – данных. В качестве примера таких программных продуктов можно привести пакет программных продуктов Графор, геометрический пакет ФАП-КФ, Geometer's Sketchpad.

В системе Графор используется алгоритмический язык Фортран IV, ограничивающий типы переменных до числовых и строковых данных, явное задание структурных типов переменных невозможно. Также причинами, ограничивающими использование пакета Графор, можно назвать: отсутствие графического интерфейса (возможна только текстовая форма кодирования алгоритмов); организация геометрической конструкции возможна только при наличии ее аналитического эквивалента; невзаимосвязанность процессов написания кода, трансляции в объектно-ориентированную программу и окончательной отладки кода.

ФАП-КФ, как и Графор, является процедурным расширением языка Фортран, однако принципиальное отличие его заключается в том, что в нем реализована возможность представления геометрических объектов в виде структурированных переменных без необходимости решения данной задачи пользователем. Остальные ограничения ФАП-КФ, в целом, схожи с концепцией Графор.

Geometer's Sketchpad – система автоматизации программирования задач, однако процесс программирования скрыт от пользователя. Геометрическая переменная ассоциируется с выводом объекта на монитор, а сложные геометрические построения предстают в виде функций, параметрами которых являются геометрические переменные. Генерируемая системой программа представляет собой последовательность элементарных взаимосвязанных геометрических построений. Есть возможность использования скриптов пользователем для реализации нужных команд при часто повторяющихся однотипных построениях. Решаемые и поддерживаемые задачи Geometer's Sketchpad формулируются в терминах евклидовой геометрии, в связи с чем не используются геометрические понятия расширенной проективной плоскости, есть возможность задать не более двух аргументов геометрических функций, в системе не предусмотрено конструирование геометрических объектов производных типов и методов их обработки. Данная система относительно недорога, адаптирована под рынок РФ, интегрирована с MacOS/Windows.

Обобщив сказанное, можно сделать вывод о недостаточном уровне развития программно-инструментальных средств автоматизации геометрического моделирования, направленных на решение многомерных геометрических задач. Таким образом, актуальной становится задача развития новой методологии проектирования, объединяющей современные концепции и достижения процедурного и логического программирования.

Таким образом, целью данной работы является структурно-параметрический синтез интерфейса геометрической машины и реализация полученной структуры для тестового

объекта. Мотивация: разработка более гибкого инструмента геометрического проектирования, способного принять сигнал от формы и преобразовать по законам формы.

2. Проектирование геометрической машины

2.1. Понятийный аппарат

Определение понятия «Машина» с кибернетической точки зрения дано в работе [1] как устройство, с помощью которого набор основных параметров претворяется в соответствующий сигнал. Таким образом, если объект моделирует другой объект по определенному циклу информации с учетом ограничений, то два этих объекта являются тождественными машинами и моделью друг друга. Геометрическая модель в данном контексте преобразует введенные данные в выходные параметры на основе заблаговременно установленных правил и определенной конструкции, организующей каждый такт работы в виде четкой последовательности графических операций.

Отсюда можно сделать вывод, что геометрическая конструкция, позволяющая на основе фиксированной последовательности операций связывать между собой несколько параметров, выявляет многомерный характер объекта, а кроме того, работает как детерминированная машина с входом, внутренним устройством и выводом.

Впервые понятие машины как устройства некоего замкнутого преобразования было дано в работе [21]. В дальнейшем данная идея распространяется на геометрические явления в работах К. И. Валькова [1, 2]. Далее эта позиция развивается в работах петербургской геометрической школы [20, 22-23] и др.

Если предположить, что между объектами a, b, c установлено следующее соотношение $R: a = f(b, c)$, то можно представить это соотношение в виде $OUT = f(IN)$, где IN – входные параметры, OUT – вывод работы машины, f – функция, задающая вид отношений.

В свою очередь объекты a, b, c также зависят от ряда параметров, соотношения R можно записать так: $R(a, b, \dots c) \leftarrow f(l, m, \dots n)$.

Таким образом, можно рассматривать f как преобразователь с закрытой внутренней структурой. Если входных параметров нет, $IN = []$, получаем способ записи значений объектов-констант.

Пусть совокупность отношений R обладает совместным составом и ни один из объектов каждого отношения не обладает циклической зависимостью от самого себя. Именованную совокупность таких отношений условимся называть алгоритмом. Алгоритм включает в себя все отношения и объявленные в них объекты. Вычислительная работа алгоритма состоит в организации и выполнении всех вычислительных работ всех отношений и обеспечивает нахождение всех объявленных выходных параметров. Конструирование геометрической машины заключается в модификации состава отношений алгоритма. Определены две основные операции модификации:

Включение отношения в состав алгоритма

Исключение отношения из состава алгоритма.

Так как проектирование геометрической машины предполагает визуальный характер работ, требуется исследование условий, при которых обеспечивается корректность состава отношений.

Исходя из поискового характера процесса проектирования геометрических машин, следует понимать, что полученная модель, основанная на совокупности отношений, может выглядеть избыточно сложной, нерациональной.

В работе [20] представлены правила, позволяющие корректировать полученные модели с учетом рациональности соотношений.

2.2. Обзор существующих решений

В работе [4] представлен алгоритм ускорения глубокого обучения сверточных нейронных сетей на основе FPGA. Данная разработка позволяет проводить 5 гигаопераций в секунду

(GOPS) с повышением рабочей тактовой частоты на 30%, в 2–4 раза выше при использовании всего 25% мощности вентиляционной матрицы.

В работе [5] разработана программная среда для обработки изображений на базе FPGA для облегчения взаимодействия пользователя с низкоуровневой системой.

В работе [6] представлена архитектура, базирующаяся на FPGA, способная моделировать и выполнять сложные расчеты в реальном времени для трехмерных поверхностей.

В диссертации [7] разрабатываются методы ускорения расчетов в квантовой химии на GPU.

В работе [8] разработан алгоритм квантования, обеспечивающий отсутствие потери точности при высоком уровне разреженности. За счет этого работа сверточных нейронных сетей на GPU становится более эффективной

В работе [9] показан способ моделирования многослойных трехмерных фигур переменной толщины и анализа на GPU облака точек пересечения данных фигур и зазоров между ними. Результат визуализируется в Unreal Engine, который может работать с большим облаком точек.

В работе [10] описывается способ повышения производительности анализа геометрических структур нейронными сетями CUDA на GPU с использованием метода Монте-Карло.

В работе [11] спроектирована система полива, базирующаяся на параллельной системе микроконтроллеров. Данная работа интересна с точки зрения распараллеливания обработки сигналов, но всё равно достаточно трудоёмка для требуемых количеств параллельных вычислений.

В работе [13] на ПЛИС спроектирован умножитель целых чисел, учитывающий знаки.

В работе [20] разработана и описана система Симплекс, способная работать с мнимыми и бесконечно удаленными объектами.

Исходя из анализа работ, опубликованных в последние 10 лет по теме статьи в российских и зарубежных научных журналах, можно сделать вывод о наибольшей продуктивности ПЛИС (FPGA) и GPU в качестве преобразовательного центра геометрической машины. Ниже в статье показано, что точность аппроксимации геометрических образов, полученных в результате работы геометрической машины, требует большого количества параллельных вычислений, в связи с чем в качестве преобразовательного центра для первой итерации реализации интерфейса геометрической машины выбрана ПЛИС (FPGA). Очень часто для оптимизации вычислений на вентилях и увеличения рабочей тактовой частоты применяются сверточные нейронные сети, что делает еще привлекательнее возможность использования данного инструмента.

2.3. Моделирование процесса работы геометрической машины

На рисунке 1 показана графическая модель процесса моделирования объекта геометрической машиной.

Задача геометрической машины – взаимное однозначное кодирование и переход от физических сигналов к соответствующим им геометрическим образам, возможность проведения различных дальнейших простых и сложных геометрических манипуляций с объектом (аффинные, проективные преобразования, инверсия) [3]

На основе анализа работ зарубежных и российских коллег было выделено три основных варианта электронных вычислительных компонентов, пригодных для решения прикладных и теоретических геометрических задач:

- Графический процессор GPU
- Система микроконтроллеров
- ПЛИС \ FPGA

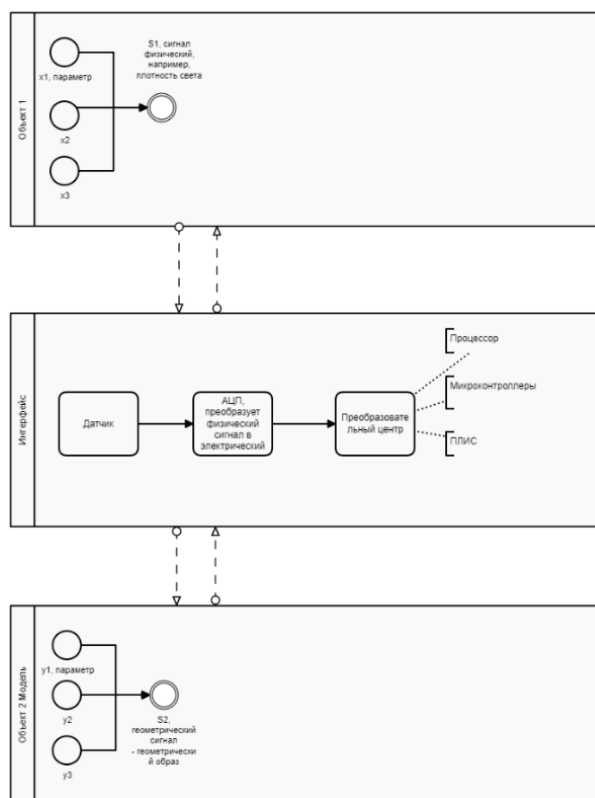


Рисунок 1 – Визуализация процесса построения модели геометрической машиной

2.4. Анализ возможных реализаций интерфейса

- ПЛИС\FPGA

Достоинства: параллельные вычисления, возможность программировать вентили на достаточно распространенном языке моделирования (Verilog\VHDL и другие), средние объемы памяти и вычислительных мощностей по сравнению с микроконтроллерами и GPU

Недостатки: для проведения вычислений нужно проводить математическую подготовку данных

Используется для: большого количества параллельных вычислений (в том числе геометрических) в работах [4-6]

- Система микроконтроллеров

Достоинства: удобно программировать, доступные дешевые решения

Недостатки: мало памяти, вычислительных мощностей, последовательные вычисления

Используется для: последовательных расчетов в работах [11,12], не используется в сфере моделирования геометрических явлений.

- GPU

Достоинства: большие объемы памяти и вычислительных мощностей, есть доступные библиотеки и среды моделирования геометрических явлений

Недостатки: уступают параллельным вычислениям по скорости.

Используется для: большого количества конвейерных вычислений (в том числе геометрических) в работах [7-10].

3. Отладка механизма работы геометрической машины

Для отладки процесса проектирования геометрической машины был разработан алгоритм на высокоуровневом языке программирования python.(рисунок 2).

№	Алгоритм работы геометрического компьютера	Текущая реализация
1	Преобразование физической формы в пиксельную	Web-камера
2	Преобразование растрового изображения в вектор (для удобства работы в образах аналитической геометрии)	Библиотека <code>opencv</code> python
3	Линейное преобразование вектор-вектор	Библиотека <code>opencv</code> python
4	Обрезание картинки	Библиотека <code>opencv</code> python
5	Аналитическое преобразование объекта	Аналитическая модель на высокоуровневом ЯП работы FPGA, GPU и системы микроконтроллеров

Рисунок 2 – Алгоритм работы геометрической машины. На первом этапе должно происходить принятие сигнала от физической формы (фотографирование) и оцифровка изображения объекта посредством web-камеры, с цифровым изображением будет проводиться дальнейшая работа. На втором этапе производится перевод растрового изображения в векторную форму, которая на третьем и четвертом этапах подвергается, если это необходимо, линейным преобразованиям (поворот, отражение, сжатие, растяжение) и обрезке. С подготовленной векторной формой на пятом этапе проводится эмуляция работы преобразовательного центра геометрической машины

В качестве тестового объекта была выбрана фотография объекта в форме окружности. Данная фотография с помощью библиотеки `opencv` была обработана алгоритмом на языке `python`, имитирующем прием геометрической машиной сигнала от формы. Сначала были получены пиксельные контуры фигуры и, соответственно, матрица чисел из 0 и 255, с которой можно в дальнейшем работать и получать координаты точек контура. Далее был получен центр фигуры (O) и самая дальняя от центра точка-пиксель (C). Полученный отрезок стал направляющим вектором для основания полярной системы координат. Следующим шагом был введен параметр точности аппроксимации. От точки C начинается откладывание векторов. Чем выше значение точности аппроксимации, тем больше векторов в системе (рисунок 3).

Таким образом, можно сделать вывод, что чем большей точности переноса контура в систему мы хотим достичь, тем больше параллельных вычислений нужно провести. Отсюда получаем, что самая подходящая структура для реализации нужного функционала должна базироваться на ПЛИС\FPGA.

Так как ПЛИС оперирует только сигналами 0 и 1, то для дальнейшей работы был осуществлен переход от полярных координат к декартовым при помощи библиотек `numpy`, `math` и `opencv` по формуле:

$$r = \sqrt{x^2 + y^2}$$

$$\cos \varphi = \frac{x}{\sqrt{x^2 + y^2}}$$

где r , $\cos \varphi$ – полярные координаты, x , y – декартовы координаты точек.

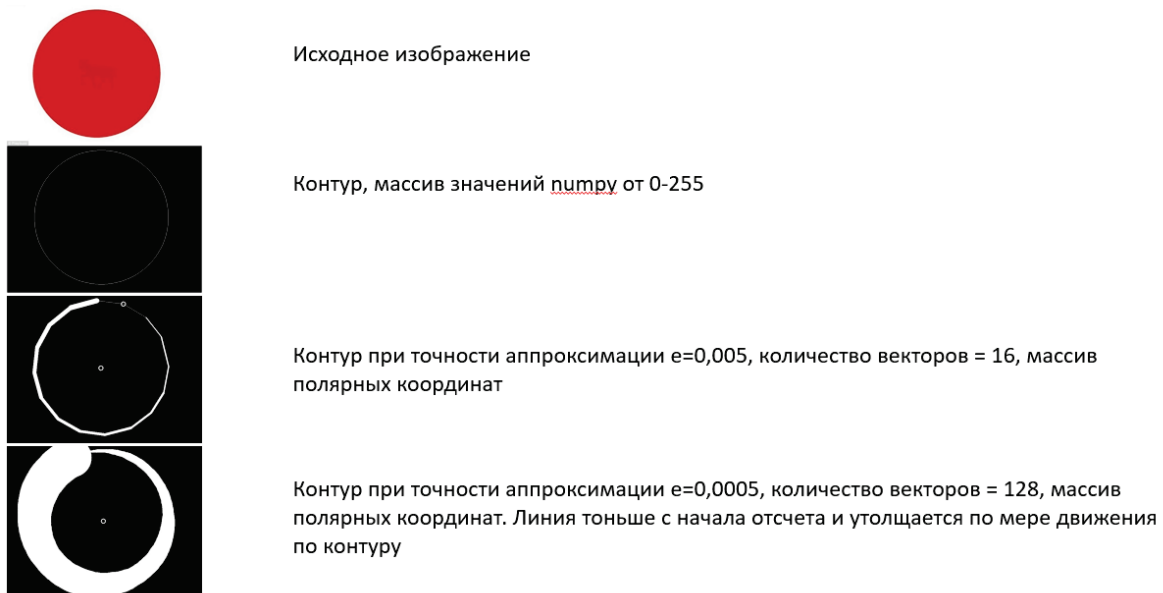


Рисунок 3 – Результаты считывания контуров тестового объекта с различными точностями аппроксимации. Толщина линий показывает направление построения линий

Для работы на FPGA Altera C4E6E10 нужно преобразовать полученные координаты в двоичную систему счисления для дальнейшей работы. В работе [13] спроектирован умножитель целых чисел на ПЛИС с учетом знаков. На основе этой работы проектируется матричный умножитель для возможности воспроизведения ряда геометрических манипуляций с тестовой фигурой. В книге [17] спроектированы сумматор, полусумматор и иные операторы.

Для представления координат и скаляра в двоичной системе счисления с учетом знака используется данная схема:

$$A = -a_{m-1}2^{m-1} + \sum_{i=0}^{m-1} 2^i a_i$$

$$x = -x_{n-1}2^{n-1} + \sum_{i=0}^{n-1} 2^i x_i$$

Тогда:

$$A * x = x_{n-1}a_{m-1}2^{m+n-2} + \sum_{i=0}^{m-2} \sum_{j=0}^{n-2} x_j a_i 2^{j+i} - \sum_{i=0}^{m-2} x_{n-1}2^{i+n-1} a_i - \sum_{j=0}^{n-2} a_{m-1}2^{i+m-1} x_j$$

где A- скаляр, x- координата.

Возможно, для оптимизации данного процесса будет протестировано смещение координатных осей с целью избежать работы со знаками координат и работать только с положительными числами.

Ниже (рисунки 4 и 5) представлена схема умножителя. В последующем планируется доработка данной схемы для работы с вещественными числами. Для точности аппроксимации (с7) = 0,0005 при допущении, что координаты в двоичной системе счисления пятиразрядные, нужны параллельные вычисления на 2 944 вентилях.

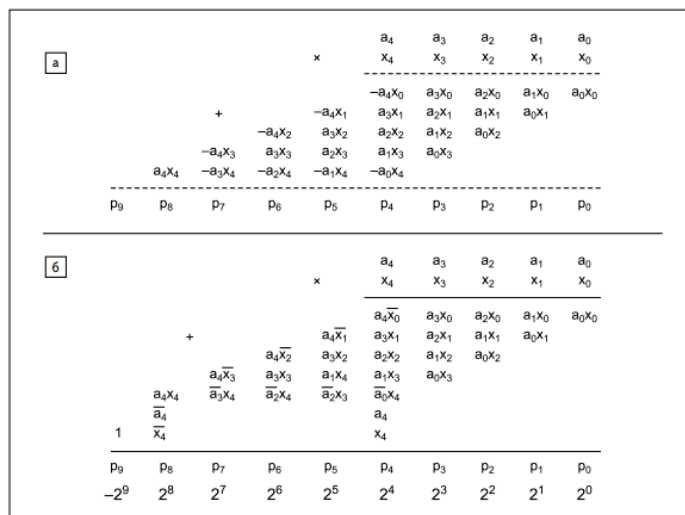


Рисунок 4 – Схема умножения скаляра на вектор в двоичной системе счисления

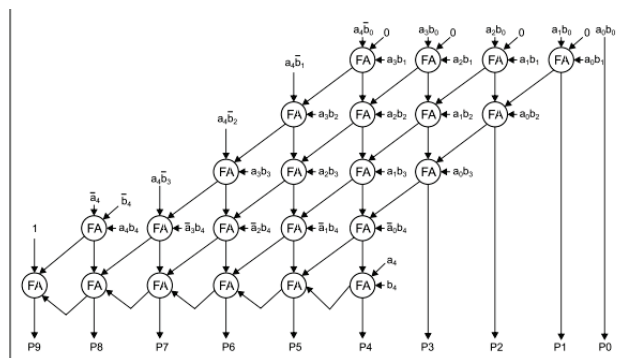


Рисунок 5 – Алгоритм действия умножителя на вентилях с учетом знака

4. Заключение

Была разработана аналитическая модель геометрической машины на python для отладки алгоритмов:

- Нахождения контуров физического объекта на фотографии
- Аппроксимации линии контура (точки, соединенные отрезками). Планируется протестировать второй подход: найти геометрическое место как можно большего количества точек контура
- Преобразований для координат векторов на ПЛИС (умножение на скаляр)

5. Список источников

[1] Вальков К.И. Введение в теорию моделирования: Монография. - Ленинград: ЛИСИ, 1973. - 150 с.
 [2] Вальков К.И. Лекции по основам геометрического моделирования: Учебное пособие. – Ленинград: ЛГУ им. А.С. Пушкина, 1970. - 180 с.
 [3] Заславский А.А. Геометрические преобразования. — М.: МЦНМО, 2004. — 86 с. 2-е изд., стереотипное.
 [4] Xu Yang. FPGA Implementation of a deep learning acceleration core architecture for image target detection / Xu Yang, Chen Zhuang, Wenquan Feng, Zhe Yang, Quiang Wang // Applied Sciences. – 2023. – 13. – 41-44.

- [5] D. Crookes. Design and implementation of a high level programming environment for FPGA-based image processing / D.Crookes, K.Benkrid, A.Bouridane, K.Alotaibi, A.Benkrid // IEE. – 2000. – 4. – 377-384.
- [6] D. Chaikalas. A real-time FPGA architecture for 3D reconstruction from integral images / D. Chaikalas, N.P. Sgouros, D. Maroulis // Elsevier. – 2010. – 21. – 9-16.
- [7] Nathan Luehr. GPU ACCELERATED QUANTUM CHEMISTRY: Ph. D., Re-distributed by Stanford University under license with the author. Stanford, 2015. – 192.
- [8] Marcin Pietroń. Speedup deep learning models on GPU by taking advantage of efficient unstructured pruning and bit-width reduction / Marcin Pietroń, Dominic Zurek, Bartolomei Sniezinsky // Elsevier. – 2023. – 67. – 24-32.
- [9] Harshil Shah. GPU-accelerated geometric algorithms for computational modeling and simulation.: A dissertation submitted to the graduate faculty in partial fulfillment of the requirements for the degree of DOCTOR OF PHILOSOPHY. Ames, Iowa, 2021. – 114.
- [10] Linus Mossberg. GPU-accelerated Monte-Carlo geometry processing for gradient-domain method: Monography. – Norrköping: Department of science and technology Linköping University, 2021. – 77.
- [11] Aderemi A. Atayero. Design and Construction of a MicrocontrollerBased Automatic Irrigation System /Aderemi A. Atayero, Adeyemi S. Alatishe // WCECS. – 2015. – 21. – 11-16.
- [12] Осипов Г.А., Кудинова Т.В. Сравнительный анализ микроконтроллеров K1986BE92Q1 и STM32F103C8 в качестве устройств обработки сигналов // Информационно-измерительные и управляющие системы. 2021. Т. 26. № 2. С. 47–53.
- [13] Строгонов А. Проектирование умножителя целых чисел со знаком методом правого сдвига и сложения в базе ПЛИС // Компоненты и технологии. – 2014. – 1. – 94-100.
- [14] Интернет-ресурс. Документация к библиотеке OpenCV URL: https://docs.opencv.org/4.x/db/deb/tutorial_display_image.html.
- [15] Р. Беллман. Введение в теорию матриц.: Учебное пособие. – М.: Наука, 1969. –36с.
- [16] Акчурин А.Д., Юсупов К.М., Колчев А.А. ОСНОВЫ РАБОТЫ В СРЕДЕ QUARTUS II. – Казань: КФУ, 2017. – 49 с.
- [17] Антонов А.А. Цифровой синтез: практический курс / под общ. ред. А. Ю. Романова, Ю.В. Панчула. – М: ДМК Пресс, 2020 – 556с.
- [18] Пеклич В.А. Мнимая начертательная геометрия. Учеб. пособие / В.А.Пеклич – М.: изд.-во АСВ, 2007 – 104 с.
- [19] Гирш А.Г. Наглядная мнимая геометрия. М.: Маска, 2008. 200 с.
- [20] Волошинов Д.В. Конструктивное геометрическое моделирование: теория, практика, автоматизация / Денис Волошинов. — [Saarbrücken]: LAP LAMBERT Academic Publishing, [2011] .— 355 с.
- [21] Эшби У. Введение в кибернетику. М.:1959. – 432с.
- [22] Вальков К. И. Моделирование и формализация. Л.: ЛИСИ, 1984. – 85с.
- [23] Вальков К. И. Общие принципы конструирования геометрических алгоритмов//Геометрические модели и их применение: Межвуз.сб.науч.тр. Ярославль, 1990. – С.5-8.