

УДК 004.92

**СИСТЕМА ВЫСОКОТОЧНОГО  
ТРЕХМЕРНОГО МОДЕЛИРОВАНИЯ РЕЗКИ ДЛЯ  
ПЯТИКООРДИНАТНЫХ ФРЕЗЕРНЫХ  
ОБРАБАТЫВАЮЩИХ ЦЕНТРОВ**

А.С. Бессараб, Е.С. Чердынцев

Томский политехнический университет  
E-mail: [intaabs@gmail.com](mailto:intaabs@gmail.com)**Бессараб Андрей Сергеевич**, магистрант кафедры оптимизации систем управления Института кибернетики ТПУ.E-mail: [intaabs@gmail.com](mailto:intaabs@gmail.com)  
Область научных интересов: программирование, компьютерное моделирование, компьютерная графика.**Чердынцев Евгений Сергеевич**, канд. техн. наук, доцент кафедры оптимизации систем управления Института кибернетики ТПУ.E-mail: [es@tpu.ru](mailto:es@tpu.ru)

Область научных интересов: компьютерная графика, теория графов, дистанционное образование, мультимедийные сети.

Предложен подход к реализации программного обеспечения для высокоточной трёхмерной визуализации резки заготовки на пятикоординатном фрезерном станке с ЧПУ. Рассматриваемое программное обеспечение позволяет моделировать пятикоординатную обработку заготовки разнообразными видами фрез под любым углом. Следует отметить, что в процессе разработки не использовались сторонние методики, все алгоритмы и подходы полностью авторские. Предлагаемое решение включает в себя: модель данных для эффективного хранения и обработки объема заготовки; модель данных режущего инструмента как тела вращения с любым сложным образующим контуром; эффективные алгоритмы точного вырезания (вычитания) материала фрезой; алгоритмы преобразования модели данных, в которой заготовка хранится и обрабатывается, в модель данных для визуализации в трехмерном пространстве. Работоспособность разработанных алгоритмов демонстрируется в реализованном программном

обеспечении. Алгоритмы позволяют выполнять расчет и визуализацию кадров в реальном времени. Программная реализация написана с использованием следующего инструментария: C++, MS Visual Studio, Direct3D, TrollTech Qt.

**Ключевые слова:**

Числовое программное управление, пятикоординатная обработка, компьютерная графика, визуализация.

**Введение**

В наше время на высокотехнологичных предприятиях, занимающихся изготовлением деталей на станках с числовым программным управлением (ЧПУ), используется специальное программное обеспечение: системы автоматизированного проектирования CAD (computer-aided design) и системы автоматизированного машиностроения CAM (computer-aided manufacturing) [1–3].

На первом этапе в CAD-среде происходит создание трехмерной модели. Затем происходит её преобразование в CAM-среду, контролирующую процесс изготовления, связанный с обработкой на станке.

CAM-среда позволяет получать управляющую программу, проводить визуализацию резки и другие действия. Управляющая программа содержит набор команд для выполнения траекторий режущего инструмента и дополнительных технологических действий. Данное программное обеспечение можно условно назвать «виртуальным станком». Таким образом, появляется возможность, перед тем как отправить управляющую программу на «реальный станок», сначала проверить её на «виртуальном станке», на котором посредством трехмерной графики визуализируется весь процесс срезания материала с заготовки, можно произвести замеры и другие действия. Подобная предварительная проверка позволяет избежать недостаточное или лишнее снятие мате-

риала, экономить ресурсы, а также предотвратить проблемы, способные привести к критическому ущербу для станка [4].

### Постановка задачи

Цель исследования заключается в разработке эффективных моделей данных, а также соответствующих алгоритмов и программного обеспечения для реализации виртуальной модели 5-координатного фрезерного станка, позволяющей проводить проверку выполнения управляющих программ. Для решения данной задачи условно выбрана модель станка японского производства Fanuc-6M (рис. 1).

На рис. 2 с целью наглядности показан упрощенный вид 5-координатного станка Fanuc-6M как набор подвижных звеньев, образующих кинематические пары. Станок можно условно представить как две кинематические цепи:

- стол (на рис. 1, слева), служащий для удержания детали и имеющий одну поступательную ось  $Y$  (стол может двигаться вертикально) и две вращательных оси  $A$  и  $C$ ;
- шпиндель с вращающимся режущим инструментом (на рис.1, справа), способный двигаться поступательно в горизонтальной плоскости по осям  $X$  и  $Z$ .



Рис. 1. Пятикоординатный станок Fanuc-6M

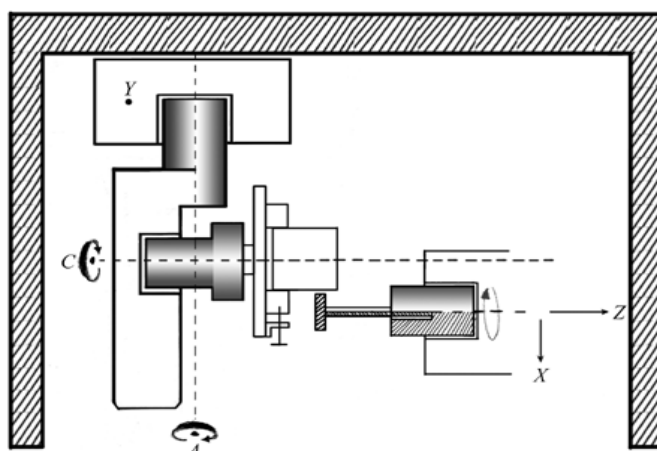


Рис. 2. Упрощенный вид 5-координатного станка Fanuc-6M в виде набора подвижных звеньев

Реальными условиями современного производства выдвигаются следующие требования к создаваемому программному обеспечению:

1. Заготовка
  - форма: параллелепипед, цилиндр, усеченный конус;
  - максимальный размер: 320×320×320 мм;
  - точность обработки: 5 микрон.
2. Режущий инструмент
  - должны поддерживаться формы режущего инструмента: как простые (цилиндрические сверла и фрезы), так и сложные (с одновременным содержанием цилиндрических, конических участков, закругленных кромок определенного радиуса);
  - система должна давать возможность пользователю настраивать все необходимые размеры режущего инструмента.
3. Прочее
  - поддержка стандартного языка управляющих программ g-code;
  - должна быть обеспечена постоянная проверка нахождения режущего инструмента внутри рабочей зоны с целью предотвратить порчу дорогостоящего оборудования;
  - система должна позволять инженеру в удобной форме производить замеры на обрабатываемой детали.

Изначально требуемая поддержка сложных форм режущего инструмента и демонстрация полученной программной реализации представлена на рис. 3.

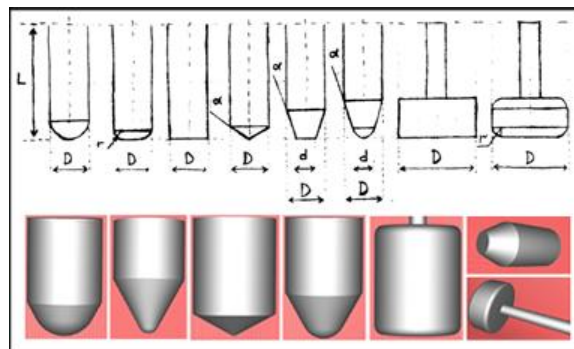


Рис. 3. Требуемые и реализованные формы режущего инструмента

### Проблема эффективного хранения и обработки большого количества частиц заготовки

В качестве одного из требований к поставленной задаче было задано, что программа моделирования резки должна обеспечивать обработку заготовки, предельные размеры которой могут достигать 320×320×320 мм, а также точность обработки должна достигать 5 микрон (0,005 мм).

Для оценки объемов информации, с которыми предстоит работать, представим условно, что рассматриваемая заготовка состоит из множества кубических частиц со стороной 5 микрон. Определим, какое количество таких частиц будет приходиться на куб размером 320×320×320 мм:

$$N = \frac{320000}{5} \cdot \frac{320000}{5} \cdot \frac{320000}{5} = 2,621 \cdot 10^{14} = 262100000000000.$$

Учитывая, что для хранения каждой из координат (x, y и z) частицы потребуется как минимум три вещественных переменных (4 байта), найдем требуемые объемы информации:

$$N = 2,621 \cdot 10^{14} \cdot 3 \cdot 4 = 3,146 \cdot 10^{15} \text{ байт} = 3 \cdot 10^9 \text{ Мб} = 2,93 \cdot 10^6 \text{ Гб} = 2930000 \text{ Гб}.$$

В результате расчетов оказалось, что в случае представления заготовки как множества простых частиц придется иметь дело с чрезмерно большими объемами информации, которые просто невозможно уместить на стандартных современных носителях, и на последовательный перебор и обработку которых уйдет очень много времени. Стоит напомнить, что в настоящий

момент на большинстве компьютеров в мире применяются жесткие диски с примерным объемом 50–1000 Гб и оперативной памятью 0,256–8 Гб.

Следовательно, необходимо использовать более эффективную форму представления информации.

#### **Карта высотных отрезков как эффективная модель для хранения и обработки большого количества частиц материала**

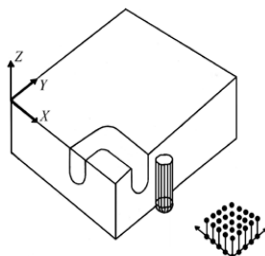
Для эффективного представления материала как набора частиц в информационных системах часто используются так называемые карты высот.

Рассмотрим простые случаи использования карт высот для различных видов резки. Например, при резке на токарном станке, заготовку удобно хранить как карту высот, представляющую собой одномерный массив расстояний относительно оси вращения (рис. 4).



**Рис. 4.** Карта высот для тела вращения

Для обработки заготовки на 3-х координатном станке удобно подходит двухмерная карта высот (рис. 5), т. е. данный подход приемлем только в том случае, если режущий инструмент не совершает бокового снятия материала. Если, в соответствии с рис. 5 считать, что фреза располагается в пространстве не параллельно оси  $z$ , а под некоторым углом, то данный вид карты высот уже неприемлем для использования, так как в ходе «виртуальной» резки у заготовки будут исчезать участки, которые при реальном процессе обработки остаются.

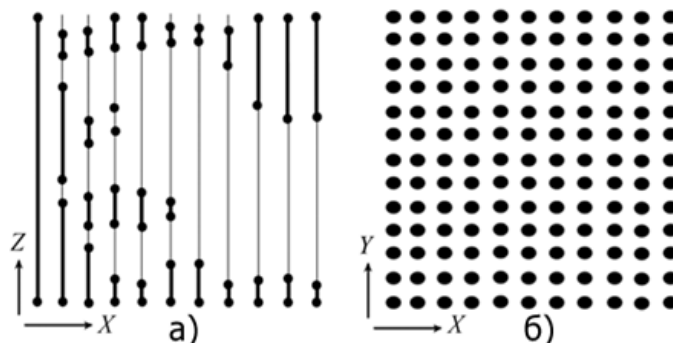


**Рис. 5.** Карта высот, используемая при обработке на трехкоординатном станке

В данном исследовании требуется виртуально обрабатывать заготовку на 5-координатном станке, а это значит, что резка заготовки может происходить с любой стороны детали и при любом положении фрезы в пространстве, что значительно усложняет задачу математически.

В результате планирования, в качестве самого эффективного способа хранения заготовки был выбран способ представления, условно обозначаемый в данной работе как «карта высотных отрезков».

Графическое представление карты высотных отрезков представлено на рис. 6.



**Рис. 6.** Карта высотных отрезков как форма хранения структуры детали для обработки на 5-координатном станке: а) вид сбоку; б) вид сверху

Карта высотных отрезков напоминает собой двумерную карту высот, однако в каждой ячейке (далее называемой высотой) хранится не одно значение высоты, а множество отрезков, где каждый представляется парой значений: расстоянием от основания до низа отрезка и расстоянием от основания до вершины отрезка.

Каждый набор высотных отрезков представляет собой не массив, а связанный список (стандартная форма хранения последовательности данных в программировании) для удобного удаления и вставки отрезков внутри последовательности.

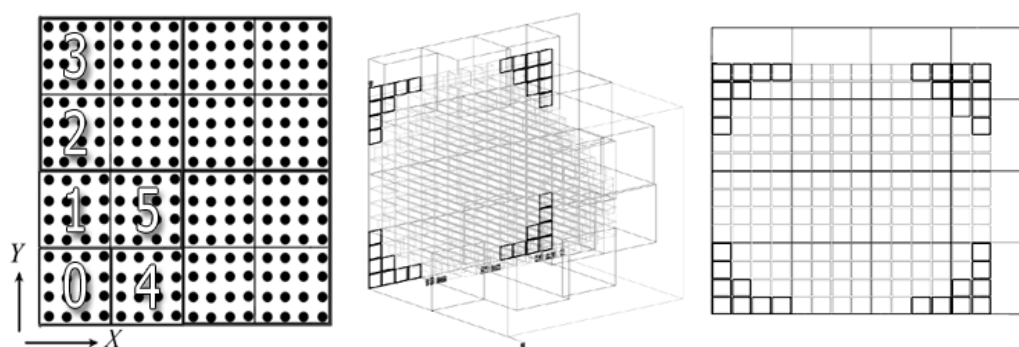
Карта высотных отрезков позволяет значительно сократить обрабатываемые объемы информации, а также ввести удобные способы оптимизации скорости вычислений.

### Оптимизация обработки заготовки

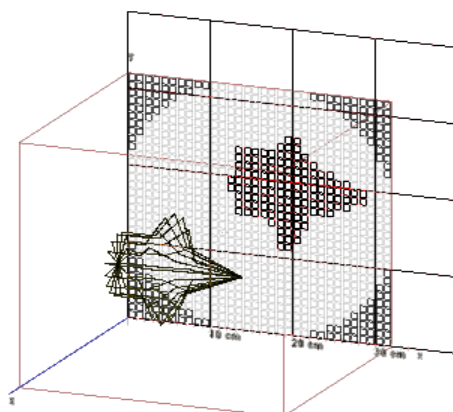
Заготовка детали представляет собой большое количество информации. Если каждый кадр обрабатывать всю заготовку, потребуется длительное время, поэтому лишние участки нужно отсекают, для этого используется разбиение пространства заготовки на секторы.

Каждый сектор хранит указатели на участки материала, находящиеся внутри его объема (рис. 7).

При обработке режущим инструментом карты высотных отрезков в вычислениях участвуют только высотные отрезки принадлежащие секторам, попавшим в пересечение с режущим инструментом, остальные отрезки отбрасываются (рис. 8). Так достигается существенная оптимизация по скорости.



**Рис. 7.** Разбиение пространства заготовки на секторы (здесь секторы отмеченные темным цветом, не содержат материала и не подлежат обработке)



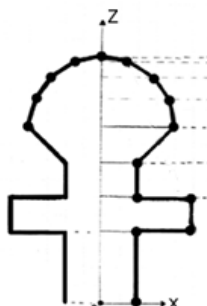
**Рис. 8.** Демонстрация секторов, попавших в пересечение с заготовкой

### Реализация режущего инструмента

Очень важной и трудоемкой частью разработки является необходимость резать заготовку режущим инструментом любой формы.

Самым эффективным в плане универсальности и быстродействия оказался вариант представления режущего инструмента в виде тела вращения с образующим контуром в виде ломаной линии.

Суть данного подхода заключается в том, что режущий инструмент условно имеет свою локальную систему отсчета и представляет собой объем, состоящий из некоторого количества цилиндрических и конических участков, формируемых контуром в виде ломаной линии (рис. 9).



**Рис. 9.** Общее представление режущего инструмента как множества цилиндрических и конических участков

### Обработка заготовки режущим инструментом, необходимость перевода координат режущего инструмента из 5-координатной системы отсчета к 3-х мерной системе отсчета, связанной с заготовкой

В ходе исследования были разработаны авторские алгоритмы для нахождения пересечений. По найденным точкам пересечения, их количеству (и др. факторам) производится изменение структуры детали.

Кратко резку (вычитание материала математически) можно описать следующим образом. Как видно из рис. 2, при обработке заготовки на пятикоординатном станке (относительно наблюдателя, стоящего рядом) одновременно движется заготовка (оси Y, C, A) и режущий инструмент (оси X, Z). Для расчетов к заготовке и режущему инструменту необходимо назначить локальные системы отсчета. В алгоритмах резки для расчета пересечений (аналитическая геометрия) используются преобразования из одного базиса в другой (линейная алгебра). Для этого используются специальные матрицы перехода.

Сначала проверяется, в какие сектора заготовки попал режущий инструмент. При этом координаты режущего инструмента переводятся в локальную систему координат заготовки, и производится расчет пересечений объемов.

Затем происходит переход к процессу вычитания материала из объема заготовки. Из тех секторов, в которые попала фреза, координаты точек каждого высотного отрезка переводятся в локальную систему координат режущего инструмента, производится нахождение точек пересечения, отсечение отрезков, попавших внутрь фрезы, и возврат полученных координат обратно в локальную систему координат заготовки.

### Визуализация структуры материала

Современные видеоадаптеры для визуализации графических элементов используют треугольники (полигональная графика). Модель трехмерного объекта является сеткой из треугольников, соединенных углами. На треугольник может быть наложена текстура, хранящаяся в файле-изображении. Механизм визуализации объектов треугольниками заложен в видеоадаптерах на аппаратном уровне, что обеспечивает высокую производительность.

Двухмерная и трехмерная графика основана на законах линейной алгебры и аналитической геометрии (матрицы и вектора). Существует глобальное пространство с абсолютной (глобальной) системой отсчета. В этом пространстве могут быть объекты, заданные либо в абсолютных координатах, либо в относительных координатах (за счет матричных преобразований смещения, поворота, растяжения, отражения и др.). В пространстве обязательно есть камера, которая определяется положением, направлением, углом обзора, глубиной видимости и др. Данные параметры камеры формируют матрицы вида и проекции, которые используются при расчете сцены. Сценой является определенная зона пространства, попавшая в поле обзора камеры, имеющее форму усеченной пирамиды (рис. 10). Объекты, оказавшиеся снаружи этой зоны, отбрасываются.

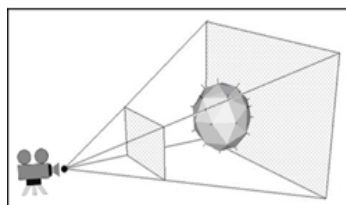


Рис. 10. Камера и сцена в трехмерной графике

Следующим этапом исследования является преобразование структуры, в которой заготовка хранится (карта высотных отрезков), в структуру, основанную на треугольниках (и других примитивах) – для визуализации. Данная задача оказалась самой сложной частью исследования.

Простейшими способами визуализации, которые пригодны только для отладки и разработки проекта, но не для пользователя программы, являются визуализация линиями и точками. Визуализация линиями позволяет изобразить структуру заготовки в том виде, в каком она хранится (карта высотных отрезков). Снимок, демонстрирующий такую визуализацию, представлен на рис. 11.

Визуализация является самым затратным по времени звеном программы. Визуализация с помощью точек является самой быстрой из всех и позволяет оценить возможности всех остальных алгоритмов программы, почти исключив затраты на визуализацию. При визуализации точками (рис. 12) рисуются одна или обе вершины каждого высотного отрезка, но не рисуются боковые части.

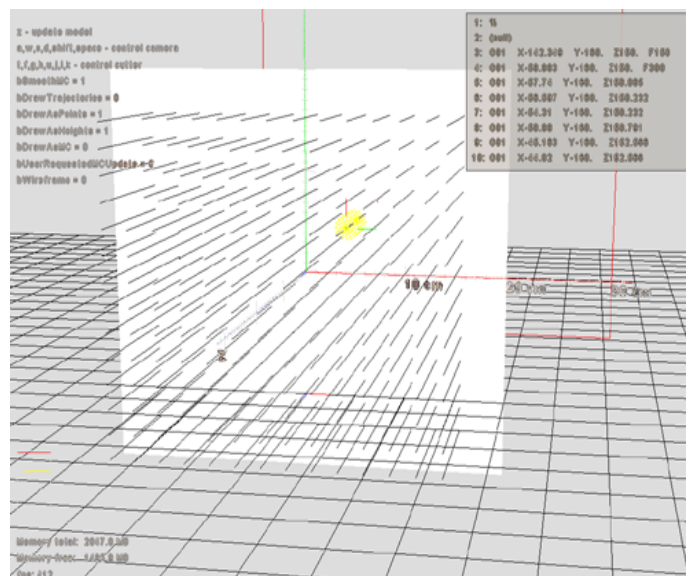


Рис. 11. Визуализация с помощью линий для отладки

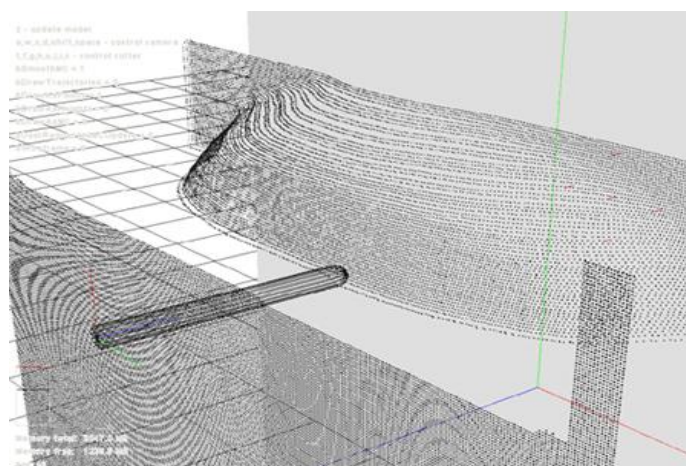
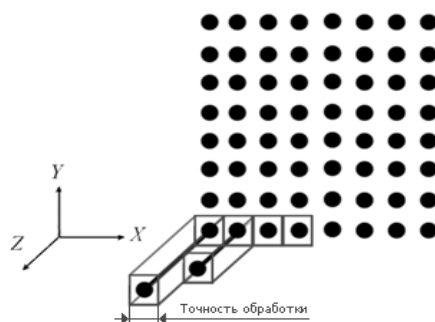


Рис. 12. Визуализация с помощью точек для отладки

### Визуализация с помощью параллелепипедов и эффекта тумана

Одним из вариантов визуализации треугольниками является отрисовка с помощью параллелепипедов и эффекта тумана. При отрисовке заготовки в трехмерном пространстве каждый высотный отрезок с карты высот представляется параллелепипедом, как это показано на рис. 13. Далее все высотные отрезки в виде параллелепипедов отрисовываются в пространстве с использованием эффекта тумана. Эффект тумана используется в компьютерной графике для постепенного уменьшения видимости объектов по мере их удаления и отбрасывания объектов отрисовки, полностью скрывшихся в тумане. Эффект тумана заложен на аппаратном уровне в видеоадаптеры. В данном же проекте эффект тумана используется в нестандартной форме и придает своеобразное затенение и позволяет видеть рельеф обрабатываемой заготовки.





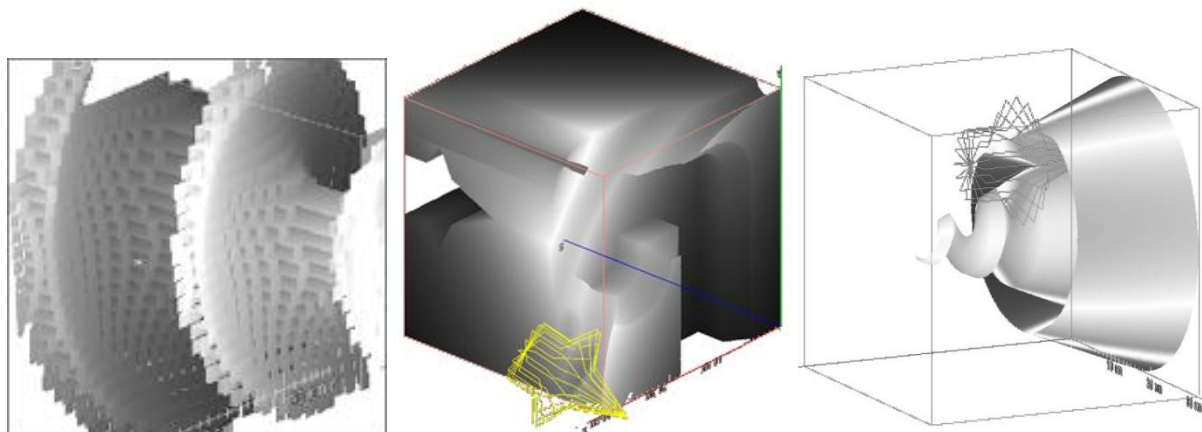
**Рис. 13.** Представление высотных отрезков в виде параллелепипедов

Для оптимизации по вычислительным затратам оказалось эффективно выводить не все грани параллелепипедов, а только те, чья лицевая сторона наиболее сильно повернута к наблюдателю (камере) (рис. 14). В случае большой точности обработки часто бывает достаточно визуализировать только лишь одну сторону, что дает большой прирост скорости визуализации, почти без потерь качества картинки.

В данном методе вершины геометрии не содержат векторы нормалей освещения из-за их ненужности, что ускоряет загрузку данных в видеопамять.

Положительная черта данного подхода заключается в том, что преобразование высотного отрезка к параллелепипеду, состоящему из треугольников, является очень быстрой операцией.

К отрицательному моменту можно отнести то, что визуализируются не только участки поверхности заготовки, но и все внутренние части материала, что не позволяет визуализировать заготовку с большим уровнем точности. Кроме того, такая визуализация часто дает не очень наглядную видимость поверхности рельефа и требует определенной настройки со стороны пользователя, которую трудно рассчитать аналитически и производить автоматически, что делает этот метод не очень практичным.



**Рис. 14.** Представление части детали посредством множества параллелепипедов с визуализацией самых существенных граней

### Визуализация с помощью алгоритма Marching Cubes

В ходе исследования были произведены попытки реализовать визуализацию с помощью известного алгоритма построения изоповерхности Marching Cubes [5, 6]. Данная попытка не привела к желаемому результату, потому лишь отметим, что данный алгоритм требует плавных переходов значений скалярного поля, в противном случае появляется ступенчатость, как показано на рис. 15. Были произведены попытки реализовать послойное сглаживание скалярного поля путем нахождения контуров и нанесения сглаженных линий на контур, как показано на рис. 16, но избавиться от ступенчатости полностью не удалось (рис. 17). В результате данный алгоритм был отвергнут как не подходящий для реализации целей работы.

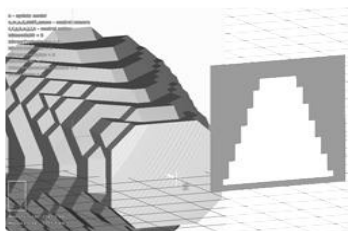


Рис. 15. Пример первой реализации алгоритма Marching Cubes

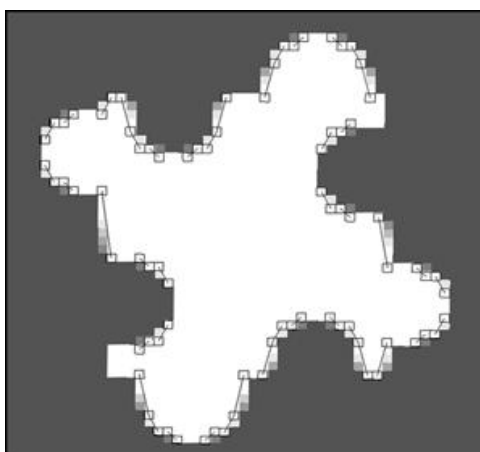


Рис. 16. Демонстрация реализации специального сглаживания на одном слое скалярного поля

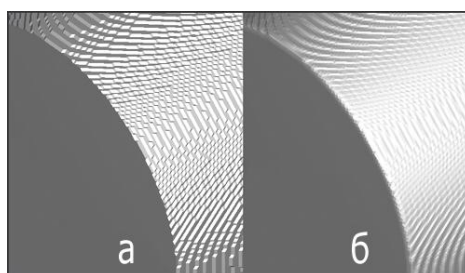


Рис. 17. Сравнение качества изображения: а) без использования сглаживания; б) с использованием сглаживания

### Визуализация карты высотных отрезков путем соединения отрезков по соседству

В ходе исследования был найден авторский способ для превращения карты высотных отрезков в структуру для визуализации треугольниками. Программная реализация показана на рис. 18 и 19. В данном способе структура данных для визуализации отображается с той же точностью, как и структура, в которой заготовка хранится, что важно при моделировании высокоточной резки. Кроме того, идеально используются нормали поверхностей для освещения. Суть подхода состоит в том, что у каждого высотного отрезка есть вершины «А» и «Б», и каждая из вершин хранит дополнительную информацию о соединении с соседними вершинами, которая обновляется в каждом кадре по необходимости. По данной информации между соседними высотными отрезками строятся внешние поверхности материала. Кроме того, данный способ позволил хранить и показывать необработанные и обработанные участки отдельно.

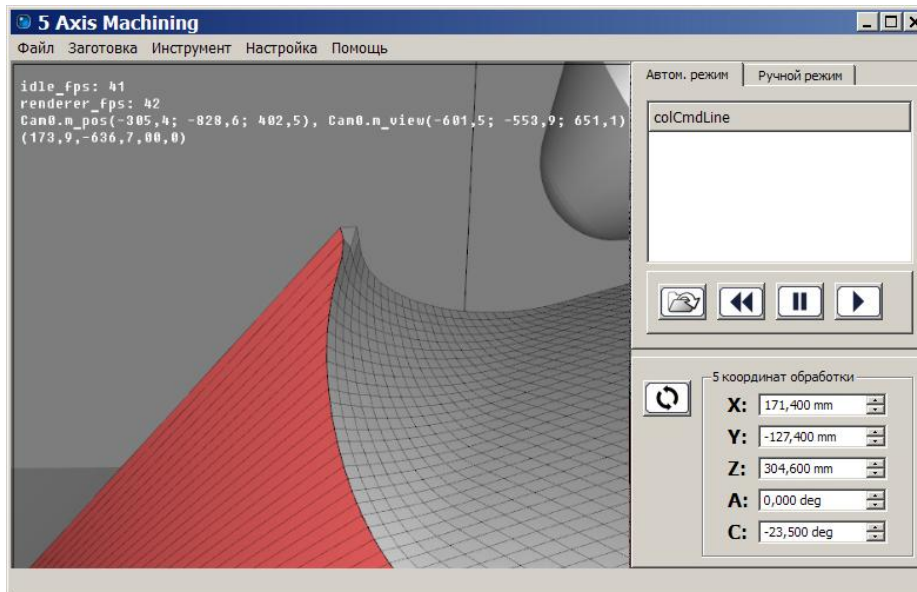


Рис. 18. Получаемая поверхность при визуализации карты высотных отрезков путем соединения отрезков по соседству

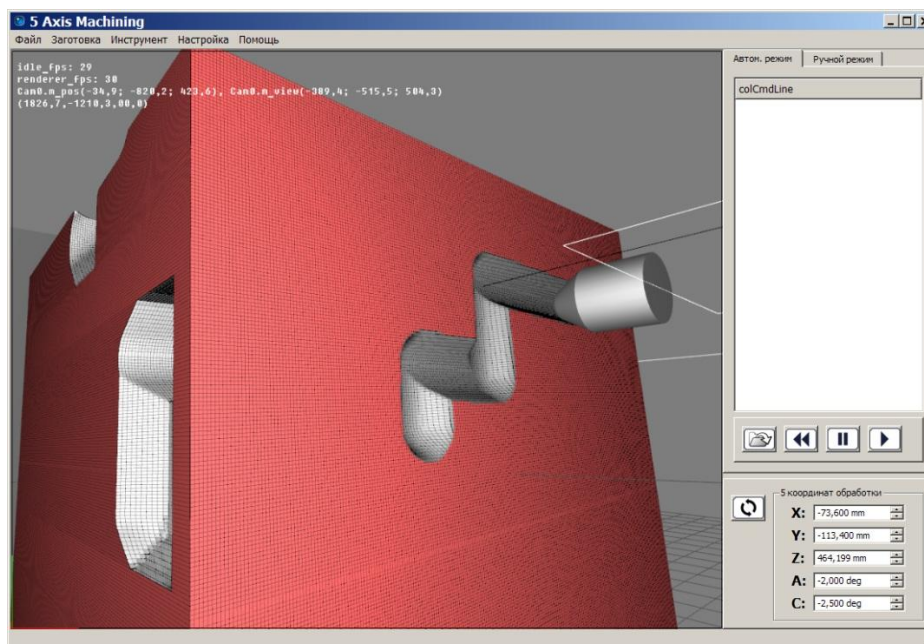


Рис. 19. Общий вид программной реализации

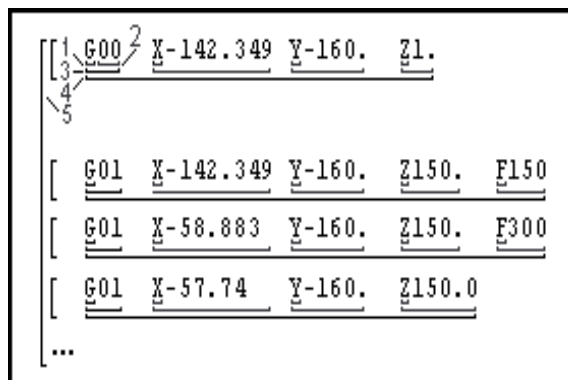
### Реализация командного управления путем создания синтаксического анализатора для управляющих программ на языке g-code

Поскольку целью исследования является разработка модели «виртуального» станка с ЧПУ, то необходимым является присутствие командного управления.

Станок с ЧПУ работает автоматически, выполняя последовательность команд в управляющей программе. Управляющая программа хранится в обычном текстовом файле. Самым распространенным языком программирования управляющих программ является язык G-code.

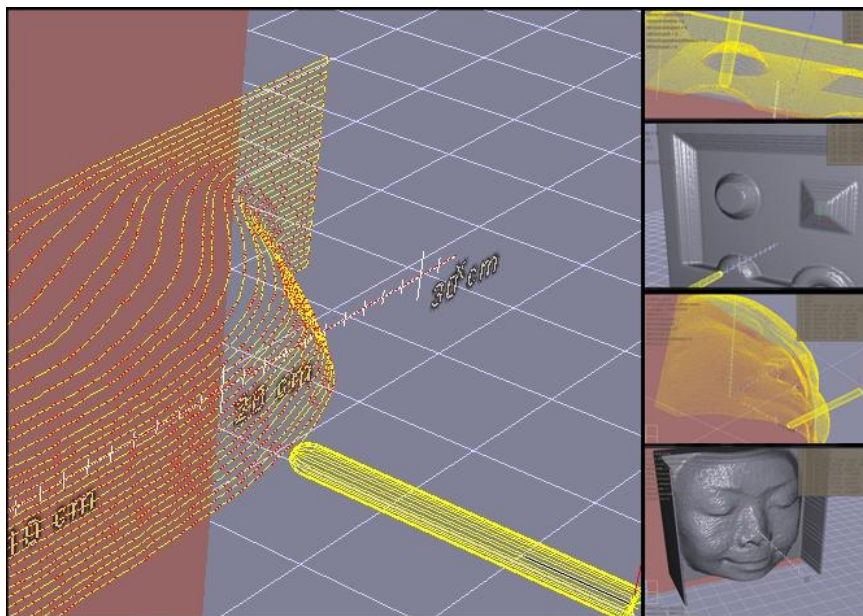
Для чтения управляющей программы реализован синтаксический анализатор (парсер) G-code программы [7]. Парсер производит просмотр текста программы, обрабатывая его по лексемам, распознавая их и записывая обнаруженные данные в удобную для использования

структуру (синтаксическое дерево), как показано на рис. 20. После того, как структура распознанной программы подготовлена, её можно выполнять.



**Рис. 20.** Принцип, по которому синтаксический анализатор распознает код и сохраняет распознанную информацию для дальнейшего удобного использования: 1 – символ, определяющий, является ли данная лексема командой или её параметром; 2 – число, являющееся либо номером команды, либо типом параметра; 3 – лексема кода программы; 4 – строка файла программы; 5 – массив строк

Демонстрация считанной из файла G-code программы и обработка заготовки показаны на рис. 21.



**Рис. 21.** Демонстрация считанной из файла G-code программы и обработка заготовки

### Анализ достигнутых результатов

Для демонстрации скорости обработки программной реализации был проведен ряд тестов. Результаты тестирования приведены в таблице. В связи с особенностями алгоритмов значение длины заготовки не влияет на время обработки, поэтому данное значение везде равно константе.

При проведении тестов использовался компьютер малой производительности следующей конфигурации:

- Intel® – совместимый одноядерный процессор 3.0 GHz;
- 2 Гб оперативной памяти;
- 256 Мб видеопамати.

**Таблица.** Результаты тестирования производительности системы

Ширина заготовки, мм	Высота заготовки, мм	Длина заготовки, мм	Точность обработки, мм	Скорость расчета и визуализации, кадров/сек
320	320	320	0,200	31
200	200	320	0,200	27
100	100	320	0,050	24
50	50	320	0,010	21
30	30	320	0,005	19

Из полученных результатов видно, что в настоящий момент небольшую заготовку размером 30×30×320 с точностью 5 микрон (точность качественных станков с ЧПУ) удается обрабатывать в реальном времени на малопроизводительном компьютере.

Основные результаты данного исследования:

- полностью авторские алгоритмы резки и визуализации заготовки полноценно поддерживают пятикоординатную обработку, заготовка может обрабатываться под любым углом;
- разработанная модель является «виртуальной» копией станочной системы Fanuc-6M, однако может быть адаптирована под другие фрезерные станки.
- алгоритмы позволяют использовать фрезы с любыми участками профилей (цилиндрические, конические, сферические, эллиптические);
- все алгоритмы обладают хорошей оптимизацией в плане быстродействия, что позволяет проводить большое количество вычислений в реальном времени;
- пользователь имеет возможность вести свободный обзор заготовки в пространстве (перемещение, вращение, масштабирование) в любой момент обработки;
- алгоритмы визуализации заготовки позволяют получить реалистичное освещение вырезаемого рельефа;
- поддерживается исполнение управляющих программ ЧПУ на языке G-код.

### Заключение

В ходе исследования удалось реализовать основные требования поставленной задачи. На данный момент программная реализация демонстрирует работоспособность алгоритмов и хорошие результаты. В то же время еще есть много возможностей сделать систему эффективнее.

### СПИСОК ЛИТЕРАТУРЫ

1. Ловыгин А. Современный станок с ЧПУ и CAD/CAM система. – М.: Эльф ИПР, 2006. – 322 с.
2. Статья о CAD-системах // Интернет-энциклопедия «Википедия». URL: [http://en.wikipedia.org/wiki/Computer-aided\\_design](http://en.wikipedia.org/wiki/Computer-aided_design) (дата обращения: 12.10.2012).
3. Статья о САМ-системах // Интернет-энциклопедия «Википедия». URL: [http://en.wikipedia.org/wiki/Computer-aided\\_manufacture](http://en.wikipedia.org/wiki/Computer-aided_manufacture) (дата обращения: 12.10.2012).
4. Вернер Д.В. Причины использования ПО для верификации и оптимизации NC-программ // Сайт разработчика САПР ООО «Евразия Ltd». URL: <http://www.sapr2000.ru/prensa3.html> (дата обращения: 12.10.2012).
5. Лоренсен В.Е. Алгоритм Marching Cubes для реконструкции трехмерной поверхности. – Л.: Энергоатомиздат, 1983. – 732 с.
6. Anderson B. An Implementation of the Marching Cubes Algorithm. URL: [http://www.cs.carleton.edu/cs\\_comps/0405/shape/marching\\_cubes.html](http://www.cs.carleton.edu/cs_comps/0405/shape/marching_cubes.html) (дата обращения: 15.11.2012).
7. Статья о синтаксическом анализе // Интернет-энциклопедия «Википедия». URL: <http://ru.wikipedia.org/wiki/> (дата обращения: 12.10.2012).

Поступила 14.11.2012 г.