

- начальный список классов арматур сравниваем с отфильтрованным; если элемент присутствует в отфильтрованном списке, то удаляем его из начального списка;
- таким образом, получаем два списка: начальный подлежит удалению, второй содержит список используемых классов арматур.

**Шаг 3.** Далее следует дополнить полученный список используемых классов арматур недостающими классами из «белого списка».

Используемый подход представляет собой достаточно простую последовательность действий с точки зрения алгоритма. Однако внедрение подобного скрипта в деятельность проектной компании позволяет существенно облегчить рабочий процесс. Кроме того, рассмотренный в работе подход позволяет учитывать многие особенности разработки и хранения проекта в таких средах как AutoDesk Revit, AutoDesk Dinamo.

#### *Литература*

API для работы с базами данных [Электронный ресурс]. – Режим доступа: <https://www.revitapidocs.com/2019/>.

УДК 004.021

### **СПОСОБЫ ПРЕДСТАВЛЕНИЯ И РАЗМЕЩЕНИЯ ТРЕХМЕРНЫХ МОДЕЛЕЙ ДЛЯ ПРОТОТИПИРОВАНИЯ ЮВЕЛИРНЫХ ИЗДЕЛИЙ**

*А.В. Киселёв, Г.Н. Верхотурова*

*Научный руководитель Г.Н. Верхотурова*

*Уфимский государственный авиационный технический университет,  
г. Уфа, Российская Федерация*

В течение последних десятилетий технологии быстрого прототипирования являются одним из основных этапов изготовления ювелирных изделий. При создании прототипа изделия применение такого подхода позволяет сократить длительность технической подготовки новой продукции в 2–4 раза, а также снизить себестоимость продукции в единичном или мелкосерийном производстве в 2–3 раза.

К технологиям быстрого прототипирования относится 3D-фрезерование, которое повсеместно используется при производстве ювелирных изделий.

Процесс 3D-фрезеровки происходит на фрезерном станке с ЧПУ, управление которым происходит через компьютер. Система станка принимает на вход текстовый файл формата .stl, создаваемый инженерами в специальных САМ-программах. Далее система станка с ЧПУ рассчитывает движение иглы из полученного файла. Движущаяся по рассчитанной траектории игла отрезает от бруска воска лишнее, оставляя лишь восковый прототип [1].

В данной статье рассматривается процесс подготовки входного файла для станка с ЧПУ с помощью специального программного обеспечения, а именно представление и алгоритм размещения трехмерных моделей ювелирных изделий с минимизацией затраченных материалов.

**Способы представления трехмерных моделей в программе.** Проведенный анализ позволил выделить пять наиболее известных методов представления трехмерных объектов: воксельные модели, модели, основанные на изображениях, точечные представления, представление с помощью полигональной сетки и иерархическое представление.

Самым распространенным представлением на данный момент является полигональная сетка. Полигональная сетка — это список вершин, ребер и граней, определяющие форму объекта с множеством граней в трёхмерной компьютерной графике и объёмном моделировании.

Для представления в виде полигональной сетки существует огромное количество программ. Эти программы предоставляют возможности редактирования, передачи по сети и визуального представления модели с использованием аппаратной поддержки.

Полигональные сетки имеют одну важную особенность — поддержка связности модели. Благодаря этой особенности полигональные сетки часто применяют для того, чтобы описать большое число синтетических поверхностей.

Представить модель с помощью полигональной сетки можно множеством способов, меняя способы хранения вершин, ребер, граней.

Чаще всего, когда хотят визуализировать трехмерный объект, используют представление в виде списка граней, представляющего

объект как множество граней и множество вершин. Список граней позволяет производить явный поиск вершин грани и граней, окружающих вершину. Как преимущества данного метода можно выделить то, что с сеткой, представленной списком граней, довольно-таки просто отыскать вершины грани, потому что список всех граней также содержит информацию обо всех вершинах, связи которых представлены гранями из списка. Также в плюсы данного метода можно записать возможность динамического обновления формы. Происходит это с помощью графического процессора, в который загружаются грани как множество индексов вершин. Далее эти вершины посылаются как позиция/цвет/структуры нормалей. При этом не происходит обновления связанности граней. Недостатками данного является то, что ввиду не заданных явно ребер, для нахождения всех граней все же необходим поиск. Также возникают проблемы при выполнении таких операций как разрыв и объединение грани.

Вершинное представление представляет модель в виде множества вершин, которые соединены с другими вершинами. В качестве преимуществ такого представления можно выделить то, что оно является простейшим, однако информация о гранях и ребрах не выражена явно, поэтому, чтобы сгенерировать список граней для рендеринга, нужно обойти все данные и поэтому из-за этих недостатков оно достаточно редко используется.

Существуют представления, решающие проблему обхода от ребра к ребру. Одним из таких является «Крылатое» представление. Оно представляет собой упорядоченное множество граней вокруг ребра.

Если возникает необходимость подразделения поверхностей или интерактивного моделирования, то разумнее всего использовать «крылатое» представление. Однако следует быть готовым, что использование данного представления приведет к высоким требованиям к памяти программы, ввиду увеличивающейся сложности структуры [2].

STL-файл, используемый в 3D-фрезеровке, описывает набор треугольников — фасетов, каждый из которых представлен 4 значениями: координатами  $X$ ,  $Y$ ,  $Z$  и значением нормального вектора, указывающим направление наружу модели. Вершины описываются против часовой стрелки по направлению нормали [3].

Ввиду структуры .stl файла, наиболее предпочтительным способом представления является список граней, так как запись файла содержит все необходимые данные для использования данного метода.

**Размещение представленных трехмерных моделей.** Для решения проблемы размещения трехмерных моделей ювелирных изделий в пределах бруска воска, предлагается свести данную задачу к более простой задаче раскроя прямоугольника на множество многоугольников.

Параллелепипедная оболочка представляет собой набор из 8 точек, которые в совокупности образуют параллелепипед.

Данное решение разбивается на 3 подзадачи:

- нахождение параллелепипедной оболочки;
- нахождение минимальной выпуклой оболочки (МВО);
- нахождение эквидистанты к МВО.

Параллелепипедная оболочка представляет собой набор из 8 точек, которые в совокупности образуют параллелепипед.

Для нахождения данных точек, достаточно знать лишь две из них: точку с наименьшими координатами и точку с наибольшими координатами  $X, Y, Z$ . Все остальные точки формируются комбинированием координат этих главных точек (рис. 1).

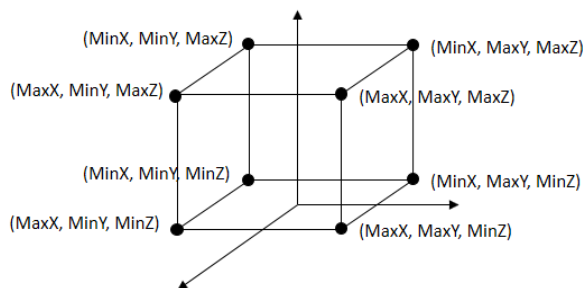


Рис. 1. Параллелепипедная оболочка

Полученная оболочка позволяет найти центр трехмерной модели, вследствие чего данную модель будет проще перемещать и поворачивать в пределах прямоугольника, используя простые матрицы поворота вокруг центра параллелепипеда.

Перед нахождением минимальной выпуклой оболочки (далее МВО) необходимо спроецировать все точки на плоскость  $xOy$ , так

как МВО представляет собой двумерный многоугольник, а все точки модели представлены тремя координатами.

МВО к множеству  $M$  точек называется кривая без самопересечений, такая, что внутри этой кривой лежат все точки множества  $M$ , с минимальным возможным периметром.

Существует множество алгоритмов нахождения МВО, среди которых:

- алгоритм Джарвиса;
- Грэхема;
- Эндрю;
- Чена.

В данной работе для реализации выбран алгоритм Джарвиса. Рассмотрим его подробнее.

Возьмем точку  $p_0$  нашего множества с самой маленькой  $y$ -координатой (если таких несколько, берем самую правую из них) и добавим ее в ответ.

На каждом следующем шаге для последнего добавленного  $p_i$  ищем  $p_i + 1$  среди всех недоавленных точек и  $p_0$  с максимальным полярным углом относительно  $p_i$  (если углы равны, надо сравнивать по расстоянию). Добавляем  $p_i + 1$  в ответ.

Если  $p_i + 1 = p_0$ , заканчиваем алгоритм [4].

Необходимость нахождения эквидистанты к фигуре обусловлена тем, что при размещении ювелирных изделий необходимо соблюдать технологическое расстояние между моделями, чтобы при вырезке изделия не возникло брака.

Алгоритм нахождения эквидистанты заключается в удалении всех точек, полученной на предыдущем шаге МВО, от центра на указанное расстояние.

Пусть  $Pnt$  — точка МВО,  $Ecv$  — эквидистанта данной точки, а  $Dist$  — расстояние, на которое  $Ecv$  удалена от  $Pnt$ , тогда  $Ecv$  находится с помощью формул:

$$Ecv.X = Pnt.X + \frac{Dist * Pnt.X}{\sqrt{Pnt.X^2 + Pnt.Y^2}}; \quad Ecv.Y = Pnt.Y + \frac{Dist * Pnt.Y}{\sqrt{Pnt.X^2 + Pnt.Y^2}}.$$

Последующее решение сводится к решению более простой задачи размещения полученных эквидистант моделей в пределах прямоугольника, заданного областью блока воска [5].

### *Литература*

1. Глебов М. 3D-Фрезерование и станок фрезерный с ЧПУ // Koloro Brand Design. — URL: <https://koloro.ua/blog/3d-tehnologii/3d-frezervovanie-i-stanok-frezernyj-s-chpu.html> (дата обращения: 06.01.2020).
2. Игнатенко А. Методы представления дискретных трехмерных данных // Лаборатория компьютерной графики при ВМК МГУ. — URL: [http://graphicon.ru/oldgr/ru/library/multires\\_rep/index.html](http://graphicon.ru/oldgr/ru/library/multires_rep/index.html) (дата обращения: 08.01.2020).
3. STL // PLMpedia. — URL: <http://plmpedia.ru/wiki/STL> (дата обращения: 13.01.2020).
4. Построение минимальных выпуклых оболочек // Хабр. — URL: <https://habr.com/post/144921/> (дата обращения: 13.01.2020).
5. Верхотурова Г.Н., Киселёв А.В. Об одном подходе к автоматизации процесса изготовления ювелирных изделий // The Proceedings of the 6th International Conference “Information Technologies for Intelligent Decision Making Support” (ITIDS2019), т. 2. — Уфа: Уфимский государственный авиационный технический университет», 2019. — С. 158–164.

УДК 004

## **10 ПРИНЦИПОВ ХОРОШЕЙ РЕАЛИЗАЦИИ BPM-СИСТЕМЫ В ОРГАНИЗАЦИИ**

*И.В. Климентьев*

*Научный руководитель Э.В. Егорова*

*Тольяттинский государственный университет, г. Тольятти,  
Российская Федерация*

Существует множество различных определений BPM-систем. Поэтому можно предположить наличие разногласий и путаниц в вопросе определения среди различных групп практиков. Для выяснения данного обстоятельства предложим следующее определение: BPMS (Системы управления бизнес-процессами) — это «набор прикладного программного обеспечения, которое дает возможность моделирования, выполнения, технического и оперативного контроля, и пользовательского представления бизнес-процессов и правил на основе интеграции существующих и новых функциональных возможностей информационных систем, которые организуются и интегрируются с помощью служб».