

Тихомиров В.А., Кутузов Д.Ф.
V.A. Tikhomirov, D.F. Kutuzov

АВТОМАТИЗАЦИЯ ОПРЕДЕЛЕНИЯ МИНИМАЛЬНЫХ ГАБАРИТОВ 3D МОДЕЛЕЙ В CAD СИСТЕМЕ UNIGRAPHICS

ATUOMATED DEFINITION OF MINIMAL DIMENSIONS OF 3D MODELS IN THE 'UNIGRAPHICS' CAD-TYPE SOFTWARE



Тихомиров Владимир Александрович – кандидат технических наук, доцент, заведующий кафедрой МОПЭВМ Комсомольского-на-Амуре государственного технического университета. E-mail: kmopevm@knastu.ru.

Mr. Vladimir A.Tikhomirov – PhD in Engineering, Associate Professor, Head of the Department, Komsomolsk-on-Amur State Technical University (Komsomolsk-on-Amur). E-mail: kmopevm@knastu.ru.

Кутузов Денис Федорович – инженер, ЗАО «Флора», системный администратор.
E-mail: d@dmsoft.ru.

Mr. Denis F.Kutuzov – engineer, JSC “Flora”, System Administrator. E-mail: d@dmsoft.ru

Аннотация: В статье описывается алгоритм и его программная реализация в среде CAD системы Unigraphics, позволяющий автоматизировать процесс определения минимальных габаритов 3D моделей и их сборок. Программный модуль внедрен на Комсомольском-на-Амуре авиационном производственном объединении им. Ю.А.Гагарина и используется при раскрое материала и проектировании упаковочной тары.

Summary: The paper describes an algorithm and its software implementation in the CAD environment of ‘Unigraphics’ package that automates the process of definition of minimal dimensions of a 3D model and of its subsequent assembly. The program module has been implemented at the JSC ‘Yury Gagarin Aviation Plant’ (Komsomolsk-on-Amur), and is used for material cutting and design of packaging.

Ключевые слова: минимальные габариты, 3D модель, автоматизация САПР, программирование в Unigraphics, функции Open API Unigraphics.

Keywords: minimal dimensions, 3D model, CAD automation, Unigraphics programming, Open API Unigraphics.

УДК 658.012

Проблема автоматизации определения габаритов 3D моделей в CAD системах неоднократно поднималась на страницах специализированных изданий [1]. Каждый автор решал эту задачу по-своему. Но все предложенные решения, к настоящему времени, имеют весьма тяжеловесные алгоритмы, требующие длительного времени выполнения на компьютере. Для практического применения в повседневной работе технолога это не приемлемо.

В статье рассматривается новый алгоритм автоматизированного определения габаритов 3D моделей (деталей и сборок) в среде CAD системы Unigraphics, имеющий минимальное (несколько секунд) время работы на персональном компьютере и обеспечивающий достаточную для производства точность решения задачи.

Общие положения

Система моделирования Unigraphics (UGS) – достаточно мощный инструмент для построения трехмерных моделей. Однако невозможно построить систему, которая решала бы *все* запросы производителей. Поэтому во многих CAD системах, и в частности в UGS, присутствует механизм, с помощью которого квалифицированный пользователь может разрабатывать собственные, встраиваемые в систему, программные модули, решающие нужные пользователю задачи. К таким механизмам в UGS относятся программирование GRIP и Open API [2].

Расширенные возможности открываются при использовании Open API (набора библиотек и подпрограмм, позволяющих внешнему приложению получить доступ к объектам модели UGS). Open API предоставляет возможность программисту создать собственное приложение (динамическую библиотеку), запустить его в среде UGS и манипулировать объектами CAD системы из этого приложения.

На производстве, при работе с системой UGS, появилась необходимость определения габаритов 3D моделей, создаваемых в этой системе. Выяснилось, что такая функция не предусматривается в UGS, и пользователь тратил время на анализ детали, построение дополнительных объектов, вычисление расстояний. Задача пользователя усложнялась при необходимости проведения измерений составных деталей и сборочных единиц.

Проблема нахождения габаритов детали

В современных CAD системах обычно нет инструментов для нахождения минимальных габаритов созданных 3D моделей. В то время как необходимость получения этих данных очевидна – взять хотя бы задачу о проектировании упаковки для изделия или раскрое материала в заготовительно-штамповочном производстве.

Для упрощения работы проектировщика необходимо автоматизировать вычисление габаритов деталей. На рис. 1 приведён пример детали с нанесенными габаритами.

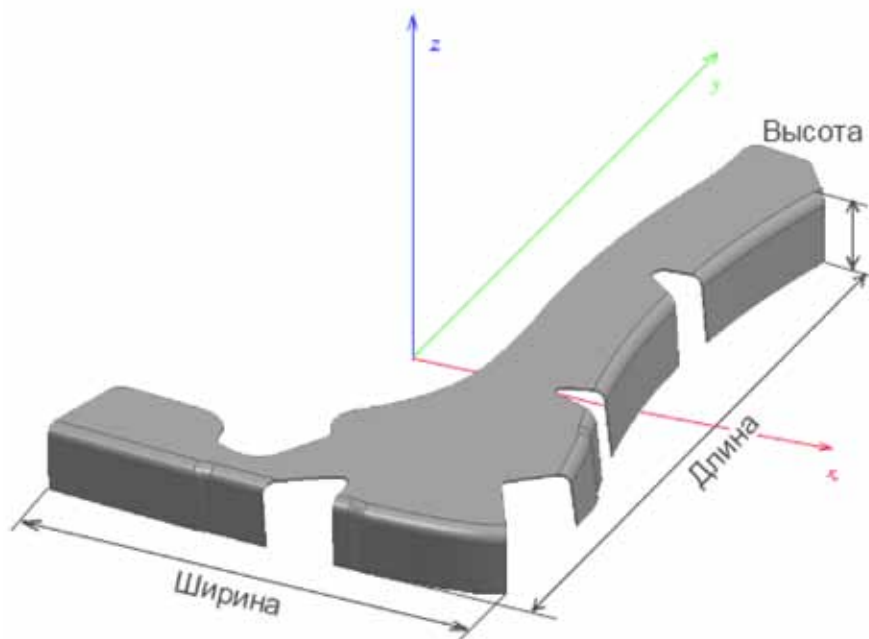


Рис. 1. Габариты модели в рабочей системе координат

В сущности, необходимо найти минимальный параллелепипед, охватывающий 3D модель, и замерить его стороны. Для построения охватывающего параллелепипеда необходимо найти нижнюю «правую», верхнюю «левую» точку модели. После вычисления модуля разности их координат мы получим длину, ширину и высоту модели. Нужный параллелепи-

пед САД системы вычисляют в текущих координатах своей рабочей сцены. Однако совсем не обязательно, что полученный параллелепипед будет минимальным, ведь 3D модель в текущих координатах сцены может располагаться произвольно и совсем не оптимально.

Для построения алгоритма определения оптимальных габаритов были решены две задачи: разработаны приемы нахождения координат диагональных точек охватывающего параллелепипеда (min, max) и предложена последовательность перебора поворота 3D модели в текущей системе координат для нахождения её оптимального положения в пространстве.

Первая задача могла решаться двумя путями:

1. Перемещением координатных плоскостей параллельно самим себе и определением наличия пересечения этих плоскостей с 3D моделью. Предлагалось пройти по трём измерениям детали с неравномерным шагом и на каждом шаге строить плоскость, перпендикулярную измерению. Для проверки наличия пересечения плоскости и 3D модели предлагалось строить сечение модели плоскостью и если сечение не 0, то деталь не входит в ограничивающий параллелепипед, иначе – входит. В последнем случае выполняются операции уточнения результата, с заданной точностью уменьшая шаг перемещения секущей плоскости. Такой алгоритм способен вычислить габариты абсолютно любой детали, но недостатком этого метода является затраченное время на построение плоскостей, сечений и вычисление шага. На рис. 2 представлен процесс измерения детали таким алгоритмом, для наглядности с равномерным шагом.

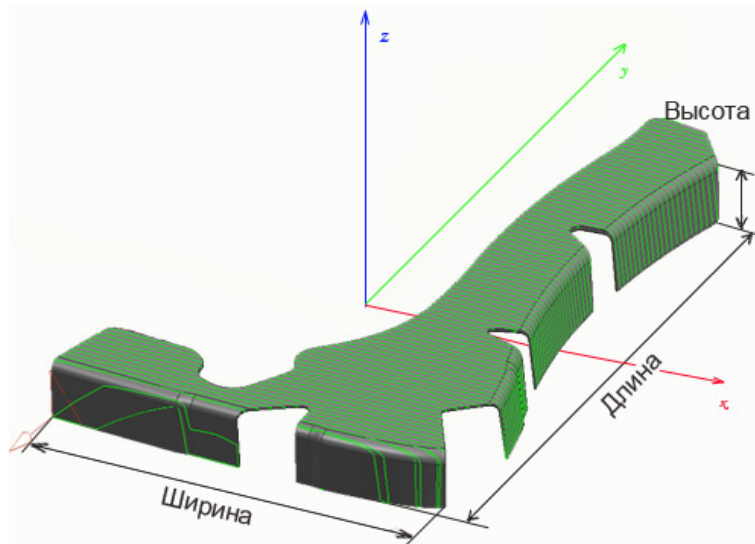


Рис. 2. Замер габаритов пересечением модели координатной плоскостью

2. Преобразованием 3D модели к фасеточному (полигональному) виду и перебору координат всех точек фасеточной модели. Каждая твердотельная модель в системе UGS прорисовывается с помощью функций OpenGL [3], соответственно она состоит из полигонов. При получении всех полигонов возможен расчёт габаритов. Однако не гарантировано, что координаты точек полигона будут равны точкам в выбранной системе измерений. Также неизвестно, какими функциями UGS предоставляет доступ к полигонам детали, на самом низком уровне, т.е. непосредственно к прорисовке. Каждую деталь в системе можно перевести в фасеточную модель (функция UF_FACET_facet_solid), то есть представить деталь в виде полигональной сетки. На рис. 3 приведена деталь с нанесёнными фасетами. После преобразования детали, необходимо получить количество фасет (функция UF_FACET_ask_n_facets_in_model). Идентификаторы всех фасет получаем в цикле функцией UF_FACET_cycle_facets. Каждый идентификатор фасеты можно опросить на координаты точек (функция UF_FACET_ask_vertices_of_facet). Для работы этой функции ей необходимо

передавать указатель на уже выделенную память под массив, а получить его размерность удобнее всего функцией `UF_FACET_ask_max_facet_verts`, которая возвращает максимальное количество вершин в одной фасете в выбранной фасеточной модели. Количество вершин в текущей фасете определяется с помощью функции `UF_FACET_ask_num_verts_in_facet`. В результате перебрать все точки модели и найти из них крайние (max, min), не составляет труда.

В ходе исследований для решения задачи был выбран второй путь, как наиболее эффективный по временным критериям. При его использовании достаточно было перебрать все точки детали и найти из них самую нижнюю правую и самую верхнюю левую. Вычислить модуль разности каждой координаты двух точек и получить длину, ширину и высоту (габариты параллелепипеда). Однако наблюдались ситуации, в которых крайняя точка (относительно рабочей системы координат) детали находилась неправильно. Это происходило, если крайняя граница тела была закруглена, и не точка, а, к примеру, сплайн лежал на самом краю 3D модели по одной из осей координат. В таких ситуациях пришлось уточнять имеющиеся краевые точки с помощью первого предложенного метода, но с неравномерным шагом. Уточнения шли до тех пор, пока шаг не становился равен введенному значению.

Для возможности измерения группы деталей необходимо перебрать все выделенные пользователям детали и проделать с каждой из них изложенный выше метод.

Данный метод применим к твёрдотельным, листовым деталям и их граням, именно такие детали вызывают затруднение измерения габаритов.

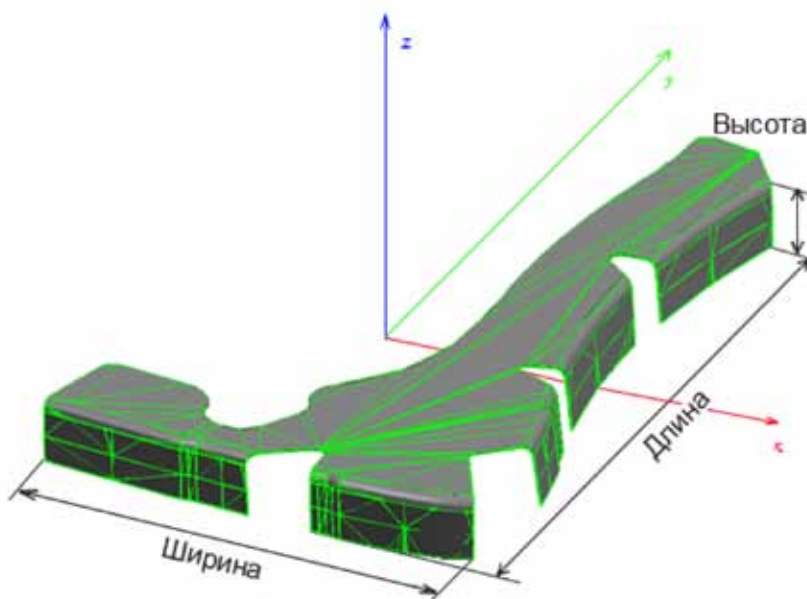


Рис. 3. Разбивка модели на фасеты

Нахождение габаритов сборки

Во многих случаях требуется найти габариты не одной детали, а целой сборки. В таком случае описанный выше алгоритм расширяется.

Из множества объектов в сборке необходимо найти тот, у которого хотя бы одна точка находится на максимальном удалении по оси. Соответственно необходимо искать и ту деталь, хотя бы одна точка которой находится на минимальной координате той же оси. Поиск ведётся для каждой из осей WCS (рабочей системы координат), то есть найденных объектов получится шесть. К каждому из них применяется вышеописанный алгоритм.

Условная реализация описанного алгоритма нахождения диагональных точек охватывающего параллелепипеда представлена в листинге 1 (псевдокод).

Алгоритм Get_polus_part

Input: tags, count_tags
 Output: x,y,z – габариты детали

```

Begin
Set polus_tags[0..5] = tags[0]
UF_FACET_ask_vertices_of_facet(tags[0])
Set polus_vertex[0..1] = vertices[0];
  For i:=1 to count_tags-1 do
    UF_FACET_ask_vertices_of_facet(tags[i])
    For j:=0 to tags[i].count_vertices-1 do
      If vertices[j].x<polus_vertex[0].x then begin
        Set polus_vertex[0].x:=vertices[j].x
        Set polus_tags [0]:=tags[i]
        End if
      If vertices[j].x>polus_vertex[1].x then begin
        Set polus_vertex[1].x:=vertices[j].x
        Set polus_tags[1]:=tags[i]
        End if
      ... повторение для осей Y и Z
    End for
  End for

  Exactly (polus_tags [0],+1) // Уточнение максимума по X (WCS)
  Exactly (polus_tags [1],-1) // Уточнение минимума по X (WCS)
  ... повторение для осей Y и Z
End Get_polus_part
    
```

Разработанная программа позволяет измерить габариты одной детали, группы деталей или составной детали, используя второй метод. Деталь может быть построена по любому методу, разрешены дополнительные построения, округления, сшивка, карманы и др.

Нахождение минимального габарита

Чтобы полученный охватывающий параллелепипед был минимальным, был разработан алгоритм поворота координатных осей CAD системы.

Для нахождения угла поворота, при котором габариты тела будут минимальными, необходимо поворачивать тело по трем осям. Координатные оси достаточно вращать по каждой из осей на угол $(180 - \Delta\alpha)$ градусов, где $\Delta\alpha$ - точность вычисления угла вращения (на рис. 4 отражены этапы вращения тела по трем осям). На вращение модели в системе координат UGS и замер габаритов модели в них тратится весьма много времени, поэтому предлагается преобразовывать модель в фасеточный тип, получать все точки модели и далее проводить все операции над массивом этих точек, за счет чего значительно ускоряется работа алгоритма.

В листинге 2 представлен условный код «вращения» системы координат для определения ее оптимального положения.

Листинг 2

Алгоритм angle_force

Input: tag
 Output: angle
 Begin
 Получить все точки объекта

```
For az:=0 to 179 do
  Rotate_Cs(points, {0,0,1},az)
  //Поворот всех точек на угол az по оси Z
  If (get_gabarit(points)<last_gabarit) then
    Set last_gabarit= get_gabarit(points)
    Set angle.z=az
  End if
End for
Поворот непосредственно детали на угол angle.z по оси Z
...
Далее предыдущий цикл, но по оси Y и X
End angle_force
```

Поворот каждой точки осуществляется без участия функций UGS (или другой CAD системы). Примененная математическая модель поворота точек в новой системе координат (в псевдокоде) представлена в листинге 3.

Листинг 3

Алгоритм Rotate_Cs

Input: dPoint[3], Axis[3], dAngle

Output: x,y,z

begin

// Производим нормирование

Set k:= sqrt(Axis[0]*Axis[0] + Axis[1]*Axis[1] + Axis[2]*Axis[2]);

Set Axis[0]:= Axis[0] / k;

Set Axis[1]:= Axis[1] / k;

Set Axis[2]:= Axis[2] / k;

Set z = dPoint[2] * (cos(dAngle) + (1-cos(dAngle))*Axis[2]*Axis[2])
+dPoint[0] * ((1-cos(dAngle))*Axis[2]*Axis[0]
+Axis[1]*sin(dAngle))+dPoint[1] * ((1-cos(dAngle))*Axis[2]*Axis[1]
+ Axis[0]*sin(dAngle));

Set x = dPoint[2] * ((1-cos(dAngle))*Axis[2]*Axis[0]
+Axis[1]*sin(dAngle)) + dPoint[0] * (cos(dAngle)
+ (1-cos(dAngle))*Axis[0]*Axis[0])
+dPoint[1] * ((1-cos(dAngle))*Axis[0]*Axis[1]
- Axis[2]*sin(dAngle));

Set y = dPoint[2] * ((1-cos(dAngle))*Axis[2]*Axis[1] -
Axis[0]*sin(dAngle)) +dPoint[0]
*((1-cos(dAngle))*Axis[0]*Axis[1]
+ Axis[2]*sin(dAngle)) +dPoint[1] * (cos(dAngle)
+ (1-cos(dAngle))*Axis[1]*Axis[1]);

end Rotate_Cs

Если использовать инструменты UGS [2] для выполнения поворота модели на рабочей сцене CAD системы (см. рис. 4), то для достаточно сложной модели потребуется достаточно много времени (от 1 до 5 мин). При этом невозможен поворот моделей, явно зависящих от других построений, UGS сам не в состоянии это осуществить (без нарушения целостности модели).

В связи с этим, в представленном алгоритме «вращается» не сама модель, а её не параметризованная копия. Это существенно ускоряет процесс вращения и ликвидирует исключительные ситуации.

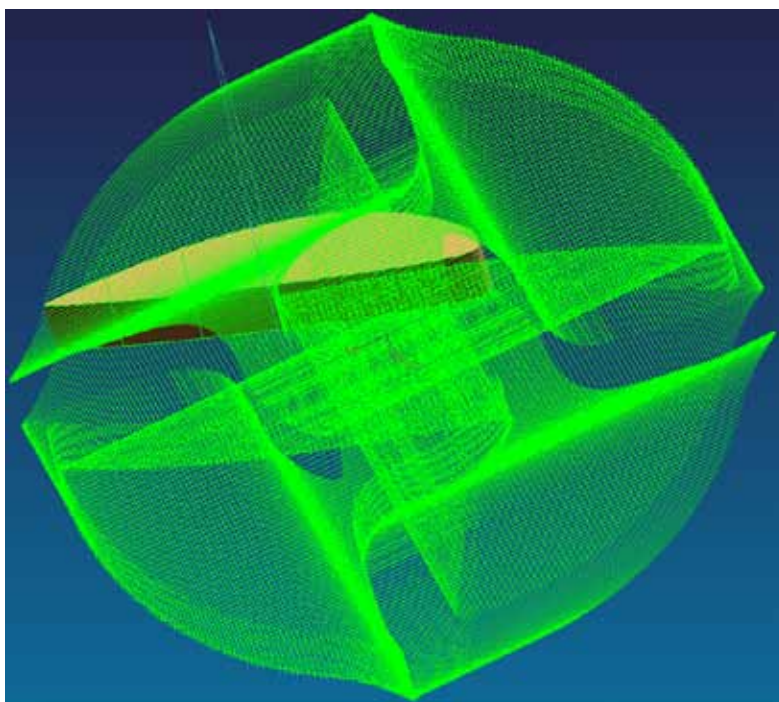


Рис. 4. Вращение модели по оси Z средствами UGS

Написанная программа способна вычислять габариты как параметризованных, так и непараметризованных моделей. Даёт пользователю возможность измерить в САД системе Unigraphics габариты любой твёрдотельной, листовой 3D модели или их сборок с минимальными затратами. Существует возможность определения габаритов отдельной грани(ей) детали. Разработано специальное меню для вызова программы из панели инструментов UGS.

Программа внедрена в работу технологических служб Комсомольского-на-Амуре авиационного производственного объединения им. Ю.А. Гагарина. Имеется акт внедрения.

ЛИТЕРАТУРА

1. Ермолин, Е. Автоматическое определение минимальных габаритов деталей / Е. Ермолин, М. Богданов, Р. Кусков, К. Лыков, А. Попова // САПР и графика. – 2007. – № 12. – С. 24-28.
2. Краснов, М. Unigraphics для профессионалов / М. Краснов, Ю. Чигишев. – М.: ЛОРИ, 2004. – 141 с.

ПРИМЕЧАНИЯ

1. Unigraphics NX – Программирование//ESG.SPB.RU : сервер группы компаний бюро ESG. 2010. URL: http://www.esg.spb.ru/win/Pr_products/Unigraphics/Unigraphics_NX/UG_program.htm (дата обращения: 30.03.2010).