

1. Организация диалога в САПР

Диалоговые языки служат средством оперативного взаимодействия проектировщика с ЭВМ, при котором происходит чередование запросов и ответов между человеком и ЭВМ в реальном масштабе времени. Диалог человека с ЭВМ в САПР рассматривается как метод решения задачи, при котором человек знает и ставит задачу проектирования, а ЭВМ используется для ее решения. В процессе диалога с ЭВМ создается цифровая модель проектируемого объекта (ЦМО), выполняются расчет и анализ характеристик объекта, формируются необходимая для его изготовления документация или программы для управления автоматами, например, станками с ЧПУ. Одним из основных требований к диалоговым языкам является близость к естественным для человека.

Многие операции по формулированию и корректировке ЦМО могут быть представлены в виде различных комбинаций следующих четырех действий:

1. выбора элемента,
2. ввода нового элемента,
3. удаления элемента,
4. изменения характеристик элемента или его связей.

По способам ввода команд различают более десяти типов представления языка диалога, среди которых наибольшее распространение получили языки типа «запрос — ответ» на основе:

1. директив пользователя;
2. выбора альтернативных возможностей;
3. заполнения пользователем форматов, представляемых машиной на экране дисплея.

В *директивных* языках основным форматом представления операторов является текстовая строка, а основным устройством ввода — алфавитно-цифровая клавиатура. Недостаток — пользователь должен помнить все элементы словаря данного языка и правила формирования предложений.

В языках, основанных на процедуре *выбора альтернативных возможностей*, конструирование предложения происходит путем указания предоставляемых на экране дисплея элементов словаря.

Процедура трансляции предложений языка существенно упрощается, если синтаксический разбор начинается не по окончании ввода всего предложения, а по мере ввода его отдельных членов. В таком случае пользователю предоставляется не весь словарь, а только та его часть, из которой выбирается очередной член предложения.

Языки с такой организацией ввода получили название *диалоговых языков со сменными наборами команд* (СНК-языки).

Диалоговые языки, основанные на использовании графических изображений и устройств ввода графических данных, называют *диалоговыми графическими языками* (ДГ-языками).

Для учета психологических факторов и улучшения процесса взаимодействия человека с ЭВМ в диалоговых графических системах (ДГС) используют ряд

приемов в организации языка изображений и сервисных программных средств. Поле экрана графического дисплея обычно разделяют на ряд областей по функциональному назначению: главную (рабочую), в которой воспроизводится собственно графическое представление объекта проектирования; процессов, предназначенную для отображения ключевых слов команд пользователя, допустимых в данном состоянии системы; графических данных — для отображения стандартных или построенных ранее графических объектов, используемых в качестве элементарных для построения сложных изображений; сопровождения диалога, в которой выводятся системные указания, пользователю, вопросы системы и диагностические сообщения; контроля данных — для вывода эхо-отображения при вводе данных с клавиатуры.

Программное обеспечение диалоговых графических подсистем САПР

Диалоговые графические подсистемы САПР (ДГП) предназначены для обеспечения непосредственного участия проектировщика в процессе автоматизированного проектирования с целью выработки оптимального решения и выпуска качественной проектной документации на основе рационального распределения функций между человеком и ЭВМ при формировании, контроле и редактировании цифровой модели объекта проектирования, при выборе методов и алгоритмов моделирования его функций, а также при оценке результатов этого моделирования и автоматизированного проектирования в целом.

В ДГП выделяют четыре типа программ:

- программы, которые строят, видоизменяют и поддерживают модель за счет добавления, удаления и замены содержащейся в ней информации;
- программы, осуществляющие просмотр моделей для извлечения информации, по которой делают графический вывод;
- программы, которые осуществляют просмотр модели для извлечения информации, используемой при анализе поведения или работы модели;
- программы, которые используются обычно тремя предыдущими группами программ для вывода информации из модели (результаты расчетных программ, сообщения пользователю и обработки управляющей информации, введенной пользователем).

В программном обеспечении унифицированных ДГП в общем случае можно выделить следующие функциональные программные процессоры.

Языковой процессор предназначен для интерпретации входных алфавитно-цифровых и графических диалоговых языков в форматы единого командного протокола взаимодействия с прикладным программным процессором.

Прикладной программный процессор выполняет семантическую интерпретацию введенных команд по формированию и корректировке цифровой модели объекта или чертежа.

Процессор геометрического моделирования предназначен для преобразования данных из модели объекта в геометрическую модель, которая зависит от требуемого вида представления модели объекта (структурная схема, трехмерное изображение, разрезы, чертежи и т. п.).

Проблемно-ориентированный графический процессор обеспечивает формирование графической модели изображения объекта в требуемом виде со всеми элементами оформления в соответствии с действующими стандартами на проектно-графическую документацию.

Процессор базового графического ввода-вывода обеспечивает преобразование графической модели данных в команды конкретных графических устройств для визуализации изображения на экране дисплея или документирования на графопостроителе.

2. Видеоконтроллеры, их стандарты для ПЭВМ типа IBM PC.

Первым стандартным видеоадаптером для IBM PC являлся **MDA** (Monochrome Display Adapter - монохромный адаптер дисплея). Он работал в текстовом режиме с разрешением 80x25 (720x350, матрица символа - 9x14), поддерживает пять атрибутов текста: обычный, яркий, инверсный, подчеркнутый и мигающий. Частота строчной развертки - 15 кГц. Интерфейс с монитором - цифровой. Объем видеопамати 4 кБ.

В восьмидесятые годы конкуренцию MDA составлял **HGC** (Hercules Graphics Card - графическая карта Hercules) - тот же самый MDA плюс монохромный графический режим 720x348.

Первым видеоадаптером для IBM PC, поддерживающим цветные графические режимы являлся **CGA** (Color Graphics Adapter - цветной графический адаптер). Он работает либо в текстовом режиме с разрешениями 40x25 и 80x25 (матрица символа - 8x8), либо в графическом с разрешениями 320x200 или 640x200. В текстовых режимах доступно 256 атрибутов символа - 16 цветов символа и 16 цветов фона (либо 8 цветов фона и атрибут мигания), в графических режимах доступно четыре палитры по четыре цвета каждая в режиме 320x200, режим 640x200 - монохромный. Интерфейс с монитором - цифровой.

EGA (Enhanced Graphics Adapter - улучшенный графический адаптер) - дальнейшее развитие CGA. Добавлено разрешение 640x350, что в текстовых режимах дает формат 80x25 при матрице символа 8x14 и 80x43 - при матрице 8x8. Количество одновременно отображаемых цветов - по прежнему 16, однако палитра расширена до 64 цветов. Структура видеопамати сделана на основе так называемых битовых плоскостей - "слоев", каждый из которых в графическом режиме содержит биты только своего цвета, а в текстовых режимах по плоскостям разделяются собственно текст и данные знакогенератора. Совместим с MDA и CGA. Интерфейс с монитором - цифровой.

MCGA (Multicolor Graphics Adapter - многоцветный графический адаптер) - введен фирмой IBM в ранних моделях PS/2. Добавлено разрешение 640x400 (текст), что дает формат 80x25 при матрице символа 8x16 и 80x50 - при матрице 8x8. Количество воспроизводимых цветов увеличено до 262144. Помимо палитры, введено понятие таблицы цветов, через которую выполняется преобразование 64-цветного пространства цветов EGA в пространство цветов MCGA. Введен также

видеорежим 320x200x256, в котором вместо битовых плоскостей используется представление экрана непрерывной областью памяти объемом 64000 байт, где каждый байт описывает цвет соответствующей ему точки экрана. Интерфейс с монитором - аналогово-цифровой.

VGA (Video Graphics Array - множество, или массив, визуальной графики) - расширение MCGA, совместимое с EGA, введен фирмой IBM в средних моделях PS/2. Фактический стандарт видеоадаптера с конца 80-х годов. Добавлен текстовый режим 720x400 для эмуляции MDA и 16 цветный графический режим 640x480 с доступом через битовые плоскости. Совместим с MDA, CGA и EGA, интерфейс с монитором идентичен MCGA.

IBM 8514/a - специализированный адаптер для работы с высокими разрешениями (640x480x256 и 1024x768x256), с элементами графического ускорителя. Не поддерживает видеорежимы VGA. Интерфейс с монитором аналогичен VGA/MCGA. Широкого распространения не получил.

IBM XGA - следующий специализированный адаптер IBM. Расширено цветовое пространство (режим 640x480x64k), добавлен текстовый режим 132x25 (1056x400). Как и IBM 8514/a не получил широкого распространения.

SVGA (Super VGA - "сверх"-VGA) - расширение VGA с добавлением более высоких разрешений. Видеорежимы добавляются из ряда 800x600, 1024x768, 1152x864, 1280x1024, 1600x1200 - большинство с соотношением 4:3. Цветовое пространство расширено до 65536 (High Color) или 16.7 млн (True Color). Также добавляются расширенные текстовые режимы формата 132x25, 132x43, 132x50. Первоначально каждая фирма предлагала собственную реализацию SVGA видеокарт несовместимых друг с другом. Это было неудобно как разработчикам программ, так и пользователям. Поэтому в 1992 году VESA (Video Electronics Standards Association - ассоциация стандартизации видеоэлектроники) разработала спецификацию VBE (VESA BIOS Extension - расширение BIOS в стандарте VESA). VBE является унифицированным стандартом программного интерфейса с VESA-совместимыми картами - при работе через видео-BIOS он позволяет обойтись без специализированного драйвера карты. Это фактический стандарт видеоадаптера в данное время. На данный момент существует версия VBE 3.0.

3. Текстуры в машинной графике.

Задача текстурирования формулируется таким образом: есть грань - согласно предположениям, треугольная - с наложенной на нее текстурой. То есть каждая точка грани окрашена цветом соответствующей ей точки в текстуре. Текстура накладывается линейным образом. Есть точка экрана с координатами на экране (sx,sy), принадлежащая проекции грани. Требуется найти ее цвет, то есть цвет соответствующей этой точке экрана точки текстуры. А для этого надо найти координаты текстуры для этой точки - точнее, для той точки, проекцией которой на экран является наша (sx,sy).

Аффинное текстурирование.

Пусть есть некоторая точка плоскости экрана (sx,sy), соответствующая точке пространства (x,y,z), тогда (u,v) – соответствующая точка текстуры.

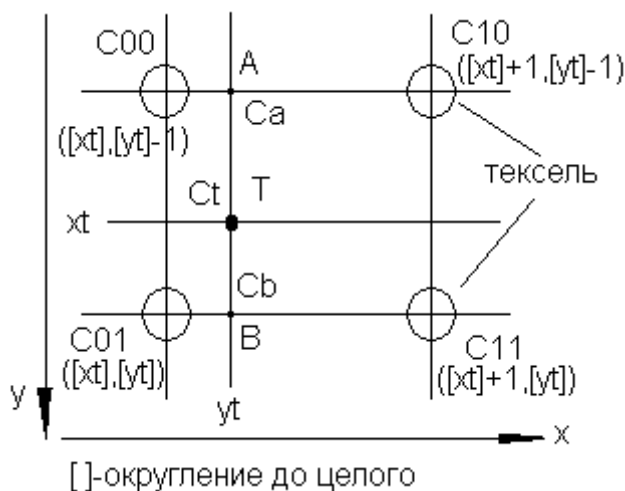
Этот метод текстурирования основан на приближении u , v линейными функциями. Итак, пусть u - линейная функция, $u = k_1 * s_x + k_2 * s_y + k_3$. Можно посчитать k_1 , k_2 , k_3 исходя из того, что хотя бы в вершинах грани u должно совпадать с точным значением - это даст нам три уравнения, из которых быстро и просто находятся эти коэффициенты, и потом считать u по этой формуле. При выводе в растр многоугольника, он заполняется текстурными точками (текселями) горизонтальными полосами сверху вниз. Так происходит поточечное текстурирование.

Билинейная фильтрация текстур.

Текстура - это 2D картинка, а 2D картинка в свою очередь - набор замеров цвета через какие-то промежутки. В реальной же жизни цвет не меняется скачком через каждый, например, миллиметр, а является какой-то непрерывной функцией от положения. При обычном текстурировании мы получаем координаты в текстуре, округляем их до ближайшего целого числа и выбираем нужный цвет из текстуры. То есть мы как бы берем значение цвета в самой близкой к рисуемой точке сетки замеров цвета, поэтому у нас цвет резко меняется, оставаясь непрерывным между узлами сетки, поэтому возникает эффект больших квадратов.

При билинейной фильтрации цвет всего-навсего линейно интерполируется между узлами сетки замеров.

Используя линейную интерполяцию вычислим интенсивность в точке A:



$$Ca = C00(1 - (xt - [xt])) + C10(([xt] + 1) - xt)$$

$$Cb = C01(1 - (xt - [xt])) + C11(([xt] + 1) - xt)$$

$$Ct = Ca(1 - (yt - ([yt] - 1))) + Cb(([yt] + 1) - yt)$$

Мипмэппинг.

Если полигон относительно сильно удален или повернут, так, что соседним пикселям на экране соответствуют сильно разнесенные точки текстуры, то возникают всякие неприятные артефакты. Они возникают потому, что при текстурировании мы выбираем лишь какую-то одну точку текстуры, а реально в экранный пиксел будет проецироваться несколько текселов.

Поэтому для удаления артефактов используется мипмэппинг. Для каждой текстуры заранее создается несколько ее копий уменьшенного размера ($1/4$, $1/8$, и

так далее), а далее при текстурировании используется либо сама текстура, либо подходящая уменьшенная копия.

Схема построения – берется базовая текстура и каждые ее четыре точки усредняются до одной.

При использовании мипмэппинга текстурирование осуществляется следующим образом:

- начиная с базового уровня определяется количество текселей на один пиксель
- если выясняется, что на 1 пиксель приходится не кратное 4 число текселей, используют линейную фильтрацию текстур (выборка текселей из 2-х соседних уровней текстур)