# Графичен модел

## Компютърна графика

Най-общо с терминът **графична** **информация** се обозначава всяка информация, която може да бъде представена чрез рисунка (образ, графика, изображение, картина, икона). Използването на компютър, като средство за представяне, синтез, обработка и изобразяване на графичната информация обособява направление, наричано **компютърна** **графика**. Трябва да се прави разлика между термините **представяне** и **изобразяване**. Под **изобразяване** се има предвид представяне на графична информация във вид на картина (образ, изображение, икона), т.е. чрез съобщения, които хората обикновено наричат „рисунки“. С терминът **представяне** се обозначава всяко друго съобщение, чрез което се представяграфична информация. Често вместо терминът „изобразяване“ се използва терминът **визуализация** на графичната информация. Макар, че терминът „компютърна графика“ отразява точно същността на това направление, в практиката се използва и терминът **машинна** **графика**. Крайният продукт и свързващо звено на компютърната графика е изображението. Следователно фундаменталните въпроси на които дава отговор компютърната графика са:

1. Как се представя изображение;
2. Как изображение се подготвя за визуализация;
3. Как се визуализира подготвеното изображение;
4. Как се осъществява взаимодействие с изображението;

На настоящия етап с терминът **компютърна** **графика** се обозначава **генерацията, представянета, обработката (оценката) и визуализацията на графични (геометрични) обекти с помощта на компютър, а също така и установяването на отношения (връзки) между геометричните обекти и неграфична информация.**

От гледна точка на целите на обработката на графична информация, компютърната графика се разделя на три основни области:

1. Изобразителна (геверативна) машинна графика;
2. Анализ на изображения;
3. Анализ на сцени;

## Обща постановка

Най-общо (генеративната) компютърната графика се занимава с **производството на картини** (изображения) чрез компютър. Което ще **рече последователност от команди за управление на периферно устройство, което генерира изображение**. Терминът изображение се използва за обозначаване на **физическата картина**, а последователността от команди е самото **представяне** на изображението, т.е. това е **програмата** за периферното устройство. От гледище на това, че и двете са съобщения, периферното устройство се разглежда като изпълнител, конвертиращ едно съобщение в друго. При изображението създадено **чрез набор от команди**, човекът-оператор трябва да има доста голяма квалификация, защото работи с представяне на абстрактния модел, а особено програмата, като набор от команди, е доста неестествена за работа. Когато изображение се генерира **чрез изчисляване** на някакви данни, човекът-оператор променя данните, а не директно програмата, но това също квалифицира операторът като програмист, защото трябва да борави директно със структрурата от данни представяща изображението. **Най-естественият подход е на човекът-оператор да се предоставят такива средства за директно манипулиране с изображението, чрез които той може да реализира естественият процес „рисуване“** – най-малкото създаване (изчертаване), отстраняване (изтриване). Можем да отличим шест характеристики на системите реализиращи този подход:

**Първо** процесът на създаване на изображение е последователност от заявени от оператора действия, като резултат от изпълнението на всяко действие е конкретно изображение, върху което може да бъде приложено ново действие. **Такава система се нарича диалогова** или **интерактивна система** за работа с графична информация.

На **второ** място **изображението** вече не **е** краен продукт, а **свързващо звено**. Изображението трябва да присъства като **структура от данни**. Времето за реакция на системата е сума от времената за обработка, конвертиране на данните и изпълнание на програмата на периферното устройство (най-често 1,2 секунди е приемливо). За различните периферни устройства се използват (реализират) различни генератори на програми, за да се реализира преносимост.

На **трето** място, ако искаме промяна, трябва да имаме **указване**. Това ще рече, че **между структурата от данни и изображението** (командите за периферното устройство) нашата система трябва да поддържа **взаимно еднозначно съответствие**.

На **четвърто** място работата ни само с примитиви е незадоволителна. Нашата система трябва да предоставя **средства за работа със структурирани изображения**, като при това те да са **реализирани** като самото изображение **– като структура от данни**.

На **пето** място, влагайки смисъл в (под)изображението, ние неминуемо трябва да го свържем с някакво **име**. Нашата система трябва да поддържа и връзки между имена и тяхното вътрешно представяне.

Не на последно място трябва да имаме входно-изходни устройства за произвеждане и манипулация с изображението. Тук удобството и бързината на работа е от първостепенно значение.

## Технически средства на Компютърната графика

С терминът технически средства компютърната графика се обозначават периферните устройства на компютъра подходящи за работа с графична информация. Основната функция, за която се използват е извеждане на изображение. Техническите средства се делят на:

1. устройства за извеждане на изображение, **графопостроители**
2. устройства за манипулация с изображение, **дисилейни системи**
3. устройства за въвеждане на данни или подпомагащи манипулацията на изображението, **диалогови** устройства

1. Най общо **графопостроителите** са устройства предназначение само за извеждане на графична информация. Те се делят на две:

1. устройства за получаване на твърдо копие (изображение)
2. устройства за диалогова работа (манипулация с изображението)

a) Устройствата за произвеждане на твърдо копие реализират рисуването върху лист посредством пишещ инструмент. Рисуването се реализира чрез последователност от команди за движение на пишещия инструмент (**устройства с произволно сканиране**) и/или хартията (**устройства с растрово сканиране**). Тук спадат устройства като плоския плоттер, барабанният плотер, електростатичен плотер и др.

b) За да бъде подходящо едно усторйство **за манипулация с изображението** е необходимо то да притежава две основни характеристики:

* времето за произвеждане на изображение да бъде малко; по възможност да не зависи от сложността на създаваната картина
* създаденото изображение да има такъв характер че да е възможна леснтата му замяна; дейността указване на елемент от изображението да може да бъде реализирана

Едно от най-разпространените устройства, които удовлетворяват тези изисквания е **Електронно Лъчевата Тръба (ЕЛТ)**. Тя представлява крушовидна вакумирана стъклена тръба. По разширения край на тръбата е нанесен слой луминофор. Изображенията произвеждани чрез ЕЛТ имат временен характер. Оттам ЕЛТ могат да бъдат разделени на ЕЛТ от тип памет (голям период на послесветене), ЕЛТ с рефреж (изображението се опреснява, поради малкия период на светене) и смесени ЕЛТ (управлява се интензивността на лъча). От гледище на елементите ЕЛТ се делят на: векторно-ориентирани (лъчът се отклонява непосредствено от произволна точка в друга такава, формирайки почти идеална права - вектор), растерно орентирани (лъчът се позиционира в строго определена последователност – мозайка; основната характеристика тук е разделителната способност), растерно-ориентирани с произволно сканиране (лъчът се отклонява в произволна точка), растерни ЕЛТ (редът на обхождане е фиксиран; независимо дали точка е осветена или не, лъчът се позиционира с определена интензивност) и др.

2. **Диалоговите устройства** са периферните устройства за въвеждане на данни и/или подпомагащи манипулацията с изображението (клавиши, клавиатура, мишка, кълбо, ръчка, скенер, сензорен панел и т.н.) Най-общо те се делят на превключватели и аналогово-цифрови преобразователи.

3. **Дисплейните системи** са устройства, реализиращи манипулация на изображението. Дисилейните системи имат две основни функции:

* бързо реализиране на изображението
* възможност за указване на елементи от текущо изведено изображение

Тези две функции определят две основни връзки: **правата** (визуализацията; система – оператор) и **обратната** **връзка** (получаване на информация; оператор – система). Следователно **диалогът може да се дефинира като последователност от алтернативно сменящи се обратна и права връзка**. Състав на дисплейната система:

* устройство за визулизиране на изображението
* едно или няколко диалогови устройства, обикновено мишка и клавиатура
* памет за записване на командите за периферните устройства, чието изпълнение визулизира изображение
* процесор, наречен дисилеен процесор

**Дисилейнят процесор** (ДП) е централния елемент на всяка дисилейна система. Неговите функции определят възможностите за манипулация с изображение на дисилейната система като цяло. ДП се състои от **дисилеен контролер и дисилеен генератор**. Дисилейният генератор е цифрово-аналогов преобразовател който превежда всяка команда в терминате на езика на ЕЛТ. Дисплейният контролер е по същество устройството, което реализира функциите на дисплейният процесор:

* изпълнение на програма, записана в дисилейната памет
* дешифриране на команда и според видът й, определяне и инициализиране на съответната схема на дисплейня генератор или предаване към ЦП на искана информация
* следене на състоянията на диалоговите устройства и съответно отразяване на това състояние като положение на курсора върху екрана чрез дисилейнят генератор
* откриване на определени състояния на входните устройства и прекъсване на работата на ЦП

Дисплейните системи могат да бъдат със запомняща тръба, с векторни дисплеи с регенерация, с растерни дисплеи с регенерация и др.

## Методология на графичното моделиране

Крайният продукт и фундаменталното свързващото звено в компютърната графика е изображението, но във всяка система за работа с графична информация се налага едно и също изображение да бъде представяно по няколко начина. Налага се да се поддържа **йерархия от представяния**. **Всяко представяне е нов, по-конкретен модел на определен аспект на абстрактния модел** на оператора за изображението. При всяко конвертиране не трябва да бъде губена информация, а само определени аспекти на представянията да бъдат конкретизирани. Най-конкретния модел ще бъде реализиран в програмата на дисплейния процесор. Той трябва да съдържа информация за:

* визуализационните характеристики на елементите, от които е съставено изображението
* геометричните характеристики, като форма, размери, местоположение, ориентация
* структурните характеристики на изображението
* семантичните характеристики на изображението (евентуалните имена на свързване)

Тъй **като процесът на преминаване от ниво в ниво е линеен**, изпълнението на операция ще протича в две фази:

* фиксиране в най-високото ниво на представяне на запис за необходимостта от изпълнение на конкретната операция
* изпълнение на конкретна операция в подходящото за нейното изпълнение ниво на представяне

Следва, че всяка система за работа с графична информация представлява **йерархична система от виртуални машини, всяка от които получава като вход някакви данни и произвежда изходни данни**, явяващи се вход на следващата виртуална машина. **В компютърната графика е прието всяка такава виртуална машина** (съвкупността от структурата на входните данни и алгоритмите върху тях) **да се обозначава с терминът процесор**.

Възможностите за манипулация с изображение (възможностите за графично моделиране) обуславят йерархия от **пет процесора**. Тази йерархия трябва да се реализира явно или неявно във всяка система за работа с графична информация. **Именно съвкупността от тези виртуални машини и отношенията между тях определят методологията за графично моделиране**.

В основата на йерархията стои процесорът, чрез който се произвежда изображение. **Дисплейният процесор** е програмно-хардуерна система, можеща да изпълнява последователност от команди и като резултат да произвежда изображение.

**Геометричният процесор**, заема следващото по-високо ниво в йерархията. Целта му е реализирането на геометричните преобразувания над изображението.

**Структурният процесор**, който реализира групирането на елементите на изображението в подизображения, и **семантичния процесор**, който свързва неграфичната информация (имената) с изображението могат да бъдат наречени **с общото наименование езикови процесори**.

Най-високото ниво в йерархията заема **диалоговият процесор**. Той е средството, чрез което командите на оператора се конвертират в изображение. Понеже диалогът се осъществява от диалоговият процесор, но чрез диалоговите устройства (които са част от дисплейната система) се налага **двата процесора да имат задължителна обратна връзка**. Нещо повече, най-разумният подход налага **диалоговият процесор да има достъп до всички нива на представяне на изображението**.

Не трябва да се забравя, че така въведените пет процесора, като понятия имат чисто логическа същност. Дефинирането им преследва единствено обособяването на обозрими области, с цел по-лесното обсъждане на спецификата на данните, подходящата структура на данните и алгоритмите, които трябва да се реализират върху тях. Във всяка система могат да бъдат реализирани различни подмножества от функциите на тези процесори.

## Изображение. Дисплеен процесор.

Най-общо изображение е рисунка, генерирана от някакво изходно периферно устройство. **Изображението** евсъщност **модел**, **отразяващ най-общите свойства на рисунките**, които могат да се получат чрез изходни периферни устройства. **Рисунката** е съвкупност от обособени от средата за рисуване области. Визуализационните характеристики са единственото средство за отделяне (обособяване) на области от средата за рисуване. **Среда за рисуване** от друга страна е най-естествено да се разглежда като множество от точки с еднакви визуализационни характеристики, точно както и всеки елемент на рисунката може да бъде разгледан по този начин и следователно **изображението** **също може да се определи като крайно множество от точки**, всяка от които има определена визуализационна характеристика.

Между пикселите на Електронно Лъчевата Тръба на екрана и точките съставящи изображението трябва да съществува взаимно-еднозначно съответствие. То се установява на базата на еднакви имена (координати).

Ако понятието **изображение** се разглежда като **конкретно състояние на екрана**, се говори за **растерна компютърна графика**.

Ако разглеждаме **изображението като подмножество на екранната област**, вече говорим за **векторна компютърна графика**. В тази дефиниция е отразена възможността изображението да се описва като съвкупност от множества от точки, като всяко множество се дефинира чрез отношението, в което трябва да се намират точките (отсечката е отношение на точки), а не чрез изброяване на точките, съставящи това множество (както при растерната графика).

**Дисплейният процесор** е виртуална машина, изпълняваща определена програма, и като резултат произвежда някаква рисунка чрез графопостроител. Дисплейният процесор изпълнява програма и произвежда битова карта (ново състояние на екрана), която е вход за дисплейната система (тя го възпроизвежда). Оттук можем да разглеждаме **дисплейният процесор като алгоритъм конвертиращ представянето на изображението като програма в представяне на изображението като битова карта**. Достатъчно е да опишем елементите и структурата на програмата (дефиниционната област), елементитие и структурата на битовата карта (областта от стойности) и алгоритъмът, по който те се съпоставят. Тук трябва да разгледаме и термина екранна област, като логическо представяне на множество от наредени двойки числа. Всяка такава двойка се нарича точка, а числата, които я съставят наричаме координати. Важно е да се отбележи, че с екрана е свързана ляво-ориентирана декартова координатна система.Тъй като лъча (на ЕЛТ) обхожда пикселите на екрана отляво на дясно и отгоре надолу, мястото на пикселите се определя по формулата

**k = (y\*M) + x** , където **0 <= x <= M** ; **0 <= y <= N** ; **0 <= k <= M\*N+M+N** ;

В компютърната графика се е наложил терминът **дисплеен файл** наместо **дисплейна програма**. Дисплейната програма трябва да описва елементи като

* типа, местоположението и ориентацията в екранното пространство на графичните примитиви
* визуализационните характеристики на елементите на изображението
* структурата, групите от графични примитиви
* семантиката на изображението (имената)

Основна характеристика на **дисплейната програма** е, че тя е **линейна**. **Графичните примитиви** биват три вида **– точки, отсечки и знаци**.

**Точките** се задават по два начина. Единият е чрез параметри описващи координатите на екранната област или **абсолютните координати**. В другия случай като параметър се задават координатите на вектор, който добавен към координатите на текущата точка определя координатите на точката за визуализация (**относителни координати**).

**Отсечките** също се задават по два начина. С координати на две точки – начална и крайна (**абсолютни координати**) или пак с координатите на вектор, един вид отместване спрямо координатите на текущата точка (**относителни координати**). Изпълнението на командата с такива параметри се свежда до определяне и визуализиране на крайно множество от точки от екранната област, които лежат най-близко до графиката на реалната точка. Използват се и специални алгоритми за намаляне на т.н. „стълбищен“ ефект.

**Знаците** се разглеждат като подмножества на някаква правоъгълна област от точки. Те също са фиксирано множество от точки, зададени спрямо локална координатна система.

**Графичен обект** е множество от примитиви, притежаващи еднакви визуализационни характеристики и статус, и идентифицируеми чрез едно име. Графичните обекти са средство за структриране на изображението.

**Визуализационните характеристики** се описват чрез команди, имащи области на действие до следващата команда от този вид. Обикновено всеки графичен обект определя свои собствени статус и визуализационни характеристики.

Дисплейната програма може да съдържа и подпрограми. Тези подпрограми се наричат **символи**. **Символът** не трябва да съдържа команди за установяване на визуализационни характеристики, наместо това символите наследяван статуса и визуализационните характеристики на иницииращият ги графичен обект.

Във всеки един момент, върху екранната област има една указана точка, кординатите на която са известни на дисплейният процесор. Указването или **идентификацията на графичен обект** е установяване на принадлежност на тази точка с един (или повече) графични обекти.

## Геометричен процесор

Основното предназначение на геометричният процесор е **реализирането на геометрични преобразувания** над изображението. Понеже вътрешното представяне на изображението може да бъде разбито до **съвкупност от точки и/или вектори**, то и функциите на геометричния процесор ще се свеждат до **геометрични операции над точка и вектор**. Екранната област не е удобна за работа, защото предполага работа с цели положителни числа в даден интервал. Затова въвеждаме понятието **световна област**, в което ще боравим и манипулираме с изображението посредством реални числа. Описанието на точка или вектор в световната област се извършва чрез двойка декартови координати, които не са удобни за геометрични преобразувания. Наместо тях използваме **хомогенни** (още еднородни**) координати и матричните умножения** като основни средства за реализиране функциите на геометричния процесор**. Елементите на наредената тройка [X,Y,H] (H != 0) се наричатхомогенни** **координати на точка в двумерното пространство**. Елементите на наредената тройка [Vx,Vy,0] се наричатхомогенни координати на вектор в двумерното пространство. Хомогенните координати се разглеждат като **вектор-ред** (с което вече можем и да ги умножаваме с матрици).

* Преход от декартови към хомогенни координати**: X = x, Y = y, H = 1;**
* Обратен преход (от хомогенни към декартови): **x = X/H, y = Y/H, H != 0;**
* Преобразуванието **X = X/H, Y = Y/H, H = 1;** се нарича **нормализация на хомогенни координати.**

**Световната област** тук ще разглеждаме като **пространство от равноправни точки**, именувани чрез координатите си, а **изображението** като **подмножество на това пространство**. Най-общо под **геометрично преобразование** върху точка или вектор ще разбираме **получаване на нови координати на точка или вектор, като функция на старите**. Общ вид на матрицата на трансформации и основните й елементи:

|  |  |  |
| --- | --- | --- |
| **a** | **b** | **p** |
| **c** | **d** | **q** |
| **m** | **n** | **s** |

**a,b,c,d** реализират мащабиране, въртене и отражение;

**m,n** са за преместване (отместване по x и y)

**p, q** получават проекции;

**s** е мащабен множител;

**Транслационна матрица** – променя местоположението на точка, но запазва разстоянието между две точки подложени на една и съща трансформация.

|  |  |  |
| --- | --- | --- |
| **1** | **0** | **0** |
| **0** | **1** | **0** |
| **m** | **n** | **1** |

|  |  |  |
| --- | --- | --- |
| **1** | **0** | **0** |
| **0** | **1** | **0** |
| **-m** | **-n** | **1** |

**T(m,n) = T-1(m,n) =**

**Ротационна матрица** (около координантото начало) запазва непроменени ъглите и разстоянията между две точки или два вектора подложени на една и съща трансформация.

|  |  |  |
| --- | --- | --- |
| **cos(a)** | **sin(a)** | **0** |
| **-sin(a)** | **cos(a)** | **0** |
| **0** | **0** | **1** |

|  |  |  |
| --- | --- | --- |
| **cos(a)** | **-sin(a)** | **0** |
| **sin(a)** | **cos(a)** | **0** |
| **0** | **0** | **1** |

**R(a) = R-1(a) =**

**Ротационна матрица** (около произволна точка с координати m,n) запазва непроменени ъглите и разстоянията между две точки или два вектора подложени на една и съща трансформация.

|  |  |  |
| --- | --- | --- |
| **cos(a)** | **sin(a)** | **0** |
| **-sin(a)** | **cos(a)** | **0** |
| **m\*(1-cos(a)) + n\*sin(a)** | **n\*(1-cos(a)) - m\*sin(a)** | **1** |

**R(a,m,n) =** = T(-m,-n) \* R(a) \* T(m,n)

местим, завъртаме, връщаме

**Мащабираща матрица** с мащабни множители **u,v**. Мащабирането променя дължината на вектор (разширява при множители по-големи от 1 или го свива в обратния случай). Променя разстоянията между точките пропорционално и запазва ъглите, в случай че **u = v.** Ако множителите имат различна стойност, множеството от точки (изображението) ще бъде деформирано, тъй като се получава различна степен на мащабиране в различните направления – променят се и разстоянията и ъглите. Точката с координати (0,0) не се влияе.

|  |  |  |
| --- | --- | --- |
| **u** | **0** | **0** |
| **0** | **v** | **0** |
| **0** | **0** | **1** |

|  |  |  |
| --- | --- | --- |
| **1/u** | **0** | **0** |
| **0** | **1/v** | **0** |
| **0** | **0** | **1** |

**S(u,v) = S-1(u,v) =**

**Обобщено мащабиране** с мащабен множител s > 0. Трансформацията не влияе на векторите.

|  |  |  |
| --- | --- | --- |
| **1** | **0** | **0** |
| **0** | **1** | **0** |
| **0** | **0** | **1/s** |

|  |  |  |
| --- | --- | --- |
| **1** | **0** | **0** |
| **0** | **1** | **0** |
| **0** | **0** | **s** |

**SS(s) = SS-1(s) =**

**Матрица на отражение** (флип на изображението около x или y, огледално изображение)

|  |  |  |
| --- | --- | --- |
| **-1** | **0** | **0** |
| **0** | **1** | **0** |
| **0** | **0** | **1** |

|  |  |  |
| --- | --- | --- |
| **1** | **0** | **0** |
| **0** | **-1** | **0** |
| **0** | **0** | **1** |

**Ox = Oy =**

Смяна координатната система:

**[X,Y,1] = [x,y,1]\*T(-a,-b)\*R(-q)** или **[X,Y,1] = [x,y,1]\*T-1(a,b)\*R-1(q);** където q е ъгълът който сключват осите (x,y) на старата и новата (X,Y) координатни системи, а Т и R са трансформации.

За да може световната област да се изобрази адекватно върху екранната върху всички точки, подлежащи на визуализация, преди да бъде отрязана дробната част на координатите им, трябва да бъде приложено мащабиране с мащабни множители **M/A и N/B (A/B = M/N)**; M,N кореспондират с екранната област, а A и B е правоъгълника, до който се оразмерява световната област.

В компютърната графика възможността за обособяване на части от изображението в световната област се нарича **определяне на прозорец на видимост**, а областта – **прозорец**. Изображението от прозореца е нужно да се визуализира върху част от екранната област. Това означава, че операторът трябва да има средство за определяне на част от екранната област, в която да се визуализира изображението. Това средство се нарича **определяне на екранен прозорец за видимост** а областта – **екранен прозорец** (или отверстие за гледане). От гледна точка на геометричния процесор, съществени са операциите **отрязване** (относно правоъгълник, изпъкнал многоъгълник, външно отрязване, отрязване спрямо произволен многоъгълник) и операцията **налагане**.

За да бъдат реализирани функциите на геометричният процесор, трябва да бъде използван или разработен пакет от подпрограми, чрез които да бъдат **реализирани ефективно решенията на голямо количество геометрични задачи**. Затова се изисква класификация на **геометричните обекти**: точкови множества, вектори, координетни системи; **и операциите над тях**: генерация на обект, информация за обект, векторна алгебра, геометрични преобразувания, координатни трансформации, класификация на точкови множества и изчисляване на сечения на точкови множества. В езиците за програмиране ги описваме с **матрици и матрични операции**.

## Структурен процесор

Хората имат свои виждания за структурата на изображението, като с всяка част свързват допълнителен смисъл. Поради тази причина е нужно да обособим виртуалната машина структурен процесор с функции за поддръжка на структура в изображението. Поддържането на структура в изображението означава възможност за установяване или премахване на отношения между елементи на изображението, т.е. това е процес, на формиране и състав на система от примитивни елементи, чрез установяване на отношения. Проблемите, с които трябва да се справи структурният процесор са следните:

1. Проблемът **за вида на отношенията**, чрез които се формират структурните елементи на изображението.
2. Елементите на такова множество могат да бъдат примитиви или множества, чиито елементи също са множества и т.н. до произволна дълбочина. От гледна точка на структурите от данни, установяването на такава йерархия не представлява проблем. Но всяка такава структурата от данни е представяне на изображение, което трябва да бъде конвертирано в дисплейна програма, т.е. структурата на дисплейната програма **налага ограничения върху отношенията**, които операторът може да установи между елементите на изображението, а това ще се отрази и върху възможните структури на данните. Следователно, вторият проблем е **проблемът за допустимите отношения** между структурните елементи на изображението от гледна точка на структурата на дисплейната програма.
3. На трето място възниква проблемът с **откриването на указан елемент от изображението**, тъй като елементите, тъй като елементите на изображението могат да бъдаг съставни, а може да бъде указан само примитив.
4. И не на последно място възниква **проблемът с прозорците**, доколкото структурата от данни, представяща изображението е една, а може да бъде визуализирана по различен начин.

Главната идея на структурния процесор е да **поддържа даннова структура аналитичен граф**, чрез която се моделират семантичните и синтактичните отношения на принадлежност. Изображението се разглежда като **съвкупност от структурирани елементи**. Основните операции, които трябва да се реализират по отношение на работата с елементите са:

* добавяне на елемент към структурата от данни
* отстраняване на елемент от структурата от данни

## Семантичен процесор

Приложно ориентирана надсройка в системите за работа с графична информация. Основни функции: свързване на структури от данни и алгоритми с имената от семантична даннова структурара и изпълнение на тези алгоритми. Основна задача: да помага за изграждането на неграфични модели и да ги преобразува в графични изображения.

Възможностите на семантичния процесор определят в голяма степен сферата на приложимост на графичната система

За да бъде широко приложим семантичният процесорр трябва да поддържа универсални модели (математически обекти) Осн. му ф-ии са три: а) представяне на мат. обекти; б) решаване на мат. задачи; в) визуализация на мат. обекти.

Почти всички широко разпространени графични системи включват семантичен процесор с възможност за описание и визуализация на функции, поради тяхното изключително широко използване.

## Диалогов процесор

От възможностите на диалоговия процесор зависи до най-голяма степен приемането или отхвърлянето на всяка система за работа с графична информация. За да има диалог е необходима реализацията на **права и обратна връзка**. Правата връзка се изразява **в предаване към системата на команда в резултат на чието изпълнение по обратната връзка се получава изображение**.От тази гледна точка диалогът може да се разглежда като **последователност от алтернативно сменящи се права и обратна връзка**. Можем да обобщим три фундаментални базови понятия: **устрайства за диалог, събитие и рамка.**

Устройствата за диалог се делят на два типа: **активни и пасивни**.

Като **пасивни** се определят онези устройства, които **във всеки момент от времето могат да бъдат в едно от множество равноправни състояния**. Промяната на такова устройство няма да носи голяма информация за процесора.

Като **активни** се определят онези устоиства, които **във всеки момент от времето могат да бъдат в едно от две неравноправни състояния**. Промяната на състояние на такова устройство носи значителна информация и се нарича **събитие**. Множеството състояния в които може да бъде едно устройство могат да бъдат определени като **пространство на състоянията на устройството**. За диалоговия процесор са важни точно тези пространства на състоянията на устройствата за диалог.

Най-общо пространствата на състоянията се делят на

* **структурирани** – декартово произведение на пространства на състояния (напр. мишка)
* **неструктурирани** – **дискретни** (напр. клавиш) и **скалирани** (напр. аналогово-цифров преобразувател)

Основната роля на диалоговият процесор се заключва в **адекватното реагиране на състоянията на активните устройства за диалог**. Те формират среда на диалог.

**Събитие** – промяна на състоянието на активно устройство. Понеже **диалоговия процесор** реагира на това събитие, можем да заключим, че основната му характеристика е, че той **е краен автомат**. Начинът за установяване на настъпило събитие е реализиран чрез **механимзва на прекъсване** (диалоговия процесор бездейства до настъпване на събитие, когато се активира и риализира действие по обработката на прекъсването). Съществуват и други методи, те имитират програмно същия механизъм. За да може диалоговият процесор да стане напълно независим, **концепцията за пространство на състояния** може да бъде обобщена и заменена в концепцията за **пространство на събития** (събитийно програмиране).

Понятието **рамка** отразява същността на процеса създаване на изображение. Възможностите за работа с рамки се обособяват в отделна подсистема – рамков процесор. Той използва съществено функциите на структурният, геометричният и дисплейният и някои възможности на семантичният процесор. Изпълнява функции по създаване, наредба, отстраняване и търсене на рамка. Установяването на принадлежнаст на пиксел към рамка става чрез обхождане в ред обратен на визуализационния и вземаме първата рамка, съдържаща пиксела (най-близката визуално).

Системите за работа с **прозорци** са системи реализиращи **графичен интерфейс**. **Прозорец** означава правоъгълна област от екрана, с която се свързва програма. Входните данни, командите за управление и резултатите от работата се предават само чрез прозореца. Прозорецът определя среда на комуникация. **Понятието графичен интерфейс включва**:

* възможност за комуникация чрез прозорец
* възможност за задаване на команда чрез свързване на пиктограма (икона, рисунка)
* възможност за манипулация с прозорци

## Стандартизация на графичен вход и изход

## Системи за графично моделиране

# Геометрично моделиране

## Подход за геометрично моделиране

Понятието **геометрично моделиране** включва **теории, методи и системи**, насочени към създаване на информационно пълни представяния на тримерни реални обекти, **които дават възможност да се изчисли** (по възможност автоматично) **всяко добре определено геометрично свойство на обектите**, които те описват. Някои от основните термини в общата теория на моделирането са **обект**, **атрибут**, **модел**, **дефиниция** и **оценка**. **Обект** се нарича някакъв реален (физически) или концептуален (мислен) обект, който ще трябва да бъде моделиран чрез средствата на информатиката. **Атрибут** се нарича всяко уместно (съществено) свойство на обект т.е. атрибути се наричат само онези свойства на обектите, които имат отношение към целта на предприетото моделиране. Всеки екземпляр се дефинира като съвкупност от атрибути. Трябва да се прави разлика между **атрибутен клас** и **атрибут**. Например, общият атрибутен клас за всички геометрични фигури е формата, но всяка геометрична фигура има конкретна форма – кръгла, квадратна и други т.е. характеризира се с конкретен атрибут. **Модел** **на обект** се определя като обект, притежаващ същите уместни атрибути като оригинала. Моделът е **точен** (пълен) , ако стойностите на всички негови уместни атрибути са същите, като тези на оригинала. Два пълни модела на един и същи обект са неразличими един от друг. Винаги съществува някакво **крайно множество от атрибути**, от което, чрез някакъв метод могат да бъдат изведени всички други атрибути. Това множество от атрибути се нарича **определение** (дефиниция) **на обекта**, или просто **модел**.

Основната цел на геометричното моделиране е създаване на представяния (модели) , позволяващи да се изчисли всяко добре дефинирано геометрично свойство на моделирания обект . Пътят за достигането на тази цел включва два последователни етапа :

Реалност Изчислителна геометрия

**метод на моделиране**

**метод на представяне**

**S**

тримерни твърди тела

**М**

математически обекти (множества от точки)

**R**

езикови структури (представяния)

Физически абстакни обекти

обекти

Ще ни интересуват само онези атрибутни класове, чиято съвкупност формира в съзнанието на всеки човек понятието (тримерно) **твърдо тяло** (съвкупността, от които формира реалността). Тази съвкупност от атрибутни класове се нарича **геометрична информация**. Реалните обекти, притежаващи определените свойства, се наричат **физически обекти**. На второ място трябва да бъдат определени онези **математически обекти**, които **представят** геометрична информация. Очевидно, математическите обекти ще бъдат описани на езика на математиката, който не е подходящ за компютърно моделиране. Поради тази причина се налага вторият етап в процеса на моделиране, през който математическите обекти се съпоставят на обекти, описани със средствата на информатиката, т.е. **информатични модели** (представяния). Същността на геометричното моделиране като научно направление се заключва именно в разработването и реализацията на различни методи за представяне на геометрична информация.

## Геометрична информация

Реалните тримерни твърди тела притежават безброй много свойства, но от гледната точка на геометричното моделиране, само някои от тях са съществени, т.е. само някои свойства трябва да присъстват като атрибути в създаваните модели. Преди всичко, **всяко тяло е обособено от средата в която се намира**, т.е. то заема някаква ограничена част от пространството. Това свойство ще наречем **ограниченост**. Обикновено повърхността на телата се възприема като граница, която отделя тялото от пространството, т.е. телата се характеризират с **граница**, която определя еднозначно коя част от пространството се разглежда като тяло. Частта от пространството, определена като тяло обикновенно се нарича **вътрешност** на тялото, а съвкупността от границата и вътрешността се нарича **тяло**. Същественото е , че границата трябва да определя **еднозначно** вътрешността на тялото. Друго съществено свойство на реалните тримерни тела е, че те са хомогенно тримерни (**обемни**), т.е. всяко тяло има тримерна вътрешност, а границата не може да има висящи или изолирани части. Това свойство ще наричаме **хомогенна тримерност**. Друго съществено свойство е, че телата не се разглеждат като група от тела, т.е. от всяка точка на вътрешността може да се тостигне до произволна друга точка на вътрешността на тяло, без да се присича граница. Това свойство ще наричаме **свързаност**. В практиката тримерните твърди тела се описват като крайна съвкупност от части на границата и крайно множество от отношения между тези части. Съвкупността от частите на границата и отношения между тези части се означава като **форма на тялото**. Формата на телата не зависи от местоположението и ориентацията им в пространството, т.е телата могат да се местят в пространството без да се променя формата или размерите им. Това свойство ще определим като **твърдост**. Моделите трябва да отразяват следните свойства:

* ограниченост
* еднозначност на границата
* хомогенна тримерност
* свързаност
* твърдост
* крайност на описанието

За моделиране на единични обекти и установяване на тъждество или различие между тях е достатъчно всеки модел да съдържа трите атрибутни класа **форма**, **метрика** **и {местоположение** **и ориентация}**, т.е това са атрибутните класове , формиращи обектите на геометрията. Тези три атрибутни класа се наричат **геометрична информация**.

**G = ({s},{m},{p}), s** - множество от пространствени форми, **m –** множество от метрически характеристики, определящи размерите, **p –** множество от параметри задаващи местоположението и ориентацията;

Геометричните обекти са множество от точки в Евклидово пространство Е3. Обикновенно едно точково множество се описва спрямо координатна система. Координатната сиситема спрямо която описваме модела наричаме **локална координатна система** (собствена за обекта ), а нейния център се нарича **полюс на обекта** (на модела). Обикновенно полюса е вътрешна точка за модела, а местоположението му и вида на локалната координатна система се избират от съображения за удобство на описанието. **Местоположението и ориентацията** на едно точково множество в Е3 се дефинират или задават като местоположение или ориентация на **локалната му координатна система** спрямо глобалната. От нейното представяне можем да индоцираме съответното точково множество, което тя описва като включим формата в уравнение и намерим неговите корени. Обикновенно в информатиката намирането на корените на уравнението се опосредства с така наречения **предикат за принадлежност на точка към множество**. Тогава нашите информатични модели на реалните тримерни обекти ще бъдат съвкупността от геометрична информация и предикат за принадлежност.

## Математическо пространство

## Схеми за представяне

Представяне на **абстрактен обект** ще наричаме синтактично коректна символна структура, изградена от знаците на някаква азбука, съгласно някакви синтактични правила. Множеството от синтактично коректни предстявяния се нарича **пространство на представянията** и се обозначава с **R**. Следователно, пространството на представянията може да се разглежда като език, породен от някаква граматика. Но от всички правилни представяния не всички са верни. Въвеждат се правила, по които на всяко R-множество се съпоставя представяне. Съвкупността от тези правила се нарича схема на представяне. В практиката за различни цели се използват различни схеми на представяне. Свойства на схемите на представяне:

* Мощност – характеризира се чрез количеството на видовете повърхности, които могат да се използват за описание на границата.
* Действителност на представянията- Областта от стойности на всяка схема за представяния са действителни представяния. Възниква въпросат как да се определи дали синтектично коректно представяне е действително. Единият подход е, да се избере такава граматика, че всички синтактично коректни представяния, породени от тази граматика да бъдат действителни. Но за да има практическа приложимост този подход схемата на представяне трябва да бъде взаимно-еднозначна. Другият подход , е да се допусне използването само на такива конструктивни математически опрерации, резулатат от които е R- множество. Тогава правилата за извършване на тези операции могат да се заложат в алгоритмите , установяващи действителността на представяне, като представянето отразява начина на получаването на описаното R-множество, т.е. да включва и операциите.
* Недвусмисленост – представянето трябва да съдържа достатъчно информация за различаване на описан абстрактен обект от всички други абстрактни обекти, а това може да стане само ако схемата на представяне е недвусмислена.
* Еднозначност- в практиката на геометричното моделиране не се използват еднозначни схеми на представяне. Причините са две : обикновенно две представяния се считат за еднакви с точност до наредбата на техните елементи ; различните представяния могат да съответстват на различно позиционирани в пространството, но конгруентни копия на един и същ реален обект.
* Сбитост- сбитите представяния съдържат малко излишни данни, но сбитостта не трябва да се преследва на всяка цена .
* Лекота на създаване – Тя е важно свойство на всяка схема на представяне. Обикновено се смята, че сбитите схеми на представяне притежават и свойството лекота на създаване.
* Ефективност в приложенията – представянията на реалните обекти се използват за данни на алгоритми, изчисляващи различни свойства, в частност, създаващи графични изображения.

Конвертиране наричаме процеса на превод на едно представяне в една схема в друго представяне в друга схема. За да може представяне на обект в една схема за представяне да бъде конвертирано в представяне на същият обект в друга схема на представяне, двете представяния трябва да бъдат съгласувани.Две схеми на представяне S и SS са еквивалентни, ако всяко представяне r от S притежава еквивалентно представяне rr от SS и обратно. Следователно, две недвусмислени схеми на представяне са еквивалентни тогава и само тогава, когато имат само една и съща дефиниционна област. Теоретично, конвертирането на представяне на две не двусмислени схеми е възможно всякога, при положение, че схемите са еквивалентни. В практиката това се проявява в наличието на алгоритми, конвертиращи от схема S в схема SS, но не и обратно. Единичното пространство на представянията е заменено с йерархия от пространства на представянията, свързани чрез схеми за представяне, които могат могат да бъдат наречени подсхеми на представяне. Всяко пространство на представяне от по-ниско ниво може да се разглежда като реализация на съответното пространство на представяне от по виско ниво.

## Схеми „екзепляри и чисти примитиви“

Тези схеми на представяне се основават **на идеята за фамилия от обекти**, като всеки член от фамилията се различава от останалите **чрез стойностите на няколко параметъра**. Всяка фамилия от обекти се нарича **генериращ примитив**, а всеки индивидуален обект от фамилията се нарича **екземпляр на примитива**. Генериращият примитив се представя като име, последвано от N-торка от променливи величини, а всеки екземпляр на примитив - като N+1-торка от стойности. В термините на програмирането чистият примитив е дефиниция на подпрограма изчисляваща някакво свойство, а екземпляр на чист примитив е оператора на изпълнение на тази програма.

Procedure XX1

A,B:real;

Var c:real

Begin

c:=A+B

end.

Основният недостатък на тази схема e липсата на средство за комбиниране на екземпляри на примитиви, с цел-получаване на нови, по-сложни обекти. По принцип, тези схеми са **сбити** и **лесни** за използване, но те предполагат стандартизация, а в практиката са удобни само, ако дефиниционната им област може да бъде сведена до каталог от много на брой фамилии, като всяка фамилия се описва с малко на брой параметри. Друг голям недостатък на тази схема е трудността на създаване на алгоритми за обработка на тези представяния. В практиката тези схеми рядко се използват самостоятелно.

## Схеми „изброяване на заетото пространство“

## Схеми „разбиване на клетки“

Тримерна триангулация на многостен с равнинни лица, имащи линейни граници, се нарича **декомпозиция** на многостена на тетраедри, чийто стени са равни са равнинни триъгълници, които от своя страна са оградени от ребра, чийто крайни точки са върхове на триангулацията.

Разбиването на клетки е обобщение на триангулацията, кадето клетките могат да имат произволен, но краен брой лица. Дефиницията е подобна, но термините тетраедър и лице са заменени съответно с **3-клетка** и **2-клетка**. 3-клетките са топологично еквивалентни на октахедрон, и са ограничени от произволен, но краен брой 2-клетки. 2-клетките са множества, топологично еквивалентни на многоъгълник, и са ограничени от произволен, но краен брой ребра. Телата могат да бъдат представени като се разбият на клетки, и се представи всяка клетка от разбиването. Следователно, схемите “изброяване на заетото пространство” са частен случай на схемите “разбиване на клетки” тъй като при тях всички клетки трябва да бъдат кубове и да лежат във фиксирана мрежа. Дефиниционната област на всяка система „разбиване на клетки” се определя от типа на клетките, използвани в схемата. Тези схеми са **недвусмислени** , но не и **еднозначни**, тъй като всяко тяло може да бъде разбито на клетки по много начини. Най-общо представянията на тези схеми не са нито сбити, нито лесни за създаване, но върху тях е възможно да се реализират много полезни алгоритми, изчисляващи различни топологични характеристики на обектите.

## Схеми „кодиране с осмично дърво“

## Схеми „конструктивна геометрия на твърди тела“

## Схеми „запълваща продукция“

Базовата идея, върху която са изградени тези схеми е**: всяко множество, което се движи през пространството, замита тяло, което може да бъде представено чрез представяне на движещо се тяло и треакторията му на движение**. Запълващата продукция може да бъде дефинирана математически в термините или на декартово произведение, или на обединение на безкраен брой от множества. Обекновено тези схеми се използват в най-простия си вид, когато движещото се тяло е двумерно R-множество, лежащо в дадена равнина, а треакторията му на движение се задава чрез отсечка или дъга от окръжност. За да се опише тримерно R-множество е достатъчно вектора на отсечката, или вектора на допирателната към окръжността да не бъдат компланарни на равнината на двумерното R-множество. Треакторията на движение може да бъде описана чрез закона за постъпателно или въртеливо движение и интервала от време, през което се извършва движението. Трябва движението да не бъде ускорително, т.е. параметъра за време да участва във формулата на закона за движение от първа степен. В този случай, алгоритъма за класифициране на точка относно описаното множество е дефиниран, тъй като във всеки момент от времето, всяка точка от пространството принадлежи не еднозначно на описаното тяло.

## Схеми „описание на границата“

Тези схеми предполагат описания **само** на **границата**. Но самото описание не определя от коя страна на границата е вътрешността на тялото. Схемите се използват много широко в компютърната графика, и много често геометричното моделиране се отъждествява с тази схема на представяне. Чрез тези схеми, тялото се представя чрез сегментиране на неговата граница на краен брой ограничени подмножества, наречени лица. Обикновено, по същия начин се представя и всяко лице, т.е. чрез ребрата, които формират неговата граница, а ребрата се представят чрез върховете си . Най-често тази схема се използва за представяне на многостени с равнинни лица. В този случай лицата са равнинни многоъгълници, ребрата – отсечки, а върховете-точки. Храктеристики:

1. С всяка схема е асоциирано някакво крайно множество от типове повърхнини които се наричат примитивни.
2. Лицата трябва да удовлетворяват следните условия:

* Всяко лице е подмножество на границата на описвания обект.
* Обединението на всички лица е равно на границата на обекта.
* Всяко лице е подмножество на примитивна повъхност.
* Всяко лице е хомогенен двумерен топологичен многоъгълник.

Създаването и използването на тези схеми предполага отговарянето на следните въпроси : Какво е лице ?, Как се представят лицата ?, Какво е ребро ? , Как се представят ребрата ?, Какви излишни данни трябва да бъдат поддържани ? Тези схеми имат голям описателен потенциал и покриват широки дефинициони области. Ако представянето на лицата е недвусмислено, то и схемата е недвусмислена , но не е еднозначна. Представянията чрез такива схеми са многословни, но имат много голяма ефективност в приложенията . Съществуват алгоритми, чрез които се установява действителността на представянията.

## Хибридни схеми

Практически всякога се използват схеми, проектиране чрез комбиниране на различни видове схеми. Тези схеми се наричат **хибридни** или **нехомогенни** . Най-често се използват схеми, чиито представяния са конструктивни дървета, чиито листа са или N-торки, представяши примитиви, или гранични представяния на непримитивни тела, или представяния за запълваши продукции. Но хибридните схеми пораждата и сериозни практически проблеми, тъй като алгоритмите за изчисляване на различни свойства се проектират и реализират твърде тежко. Поради това се препоръчва хибридните представяния да се хомогенизират , а алгоритмите да се проектират за единични хомогенни схеми. Но хогенизацията на хибридна схема е частен случай на конвертиране на хомогенни схеми една в друга. Следователно, възможността за конвертиране на представянията на една хомогенна схема в представяния на друга хомогенна схема има голямо практическо значение (известни са алгоритми за точно конвертиране; реализира се приближено конвертиране в представяния на схемата „ изброяване на заетото пространство” ) .

## Системи за геометрично моделиране

# Алгоритми за визуализация

## Обща постановка

## Алгоритъм на Робъртс

## Алгоритъм на Варнок

## Алгоритъм, използващ Z-buffer

## Алгоритъм за поредово сканиране

## Еректични образи и модели. Прост модел на осветяване

## Сенки

## Глобален модел и осветяване чрез третиране на лъчи