



# *Компютърна графика и ГПИ*

Представящи схеми

*доц. д-р Александър Пенев*

# *Представящи Схеми*



# Схеми на Представяне

- ❖ Екземпляри на чисти примитиви (Parameterized primitive instancing);
- ❖ Изброяване на заетото пространство (Spatial occupancy enumeration);
- ❖ Разбиване на клетки (Cell decomposition);
- ❖ Кодиране с осмично дърво (Octree);
- ❖ Конструктивна геометрия на твърди тела (CSG);
- ❖ Описание на границата (B-Rep);
- ❖ Функционално представяне (F-Rep);
- ❖ Запълващи продукции (Sweeping);
- ❖ Метатопчета (Metaballs);
- ❖ Системи частици (Particle systems);
- ❖ Хибридни схеми.

# *Екземпляри на чисти примитиви*

## *(Parameterized primitive instancing)*



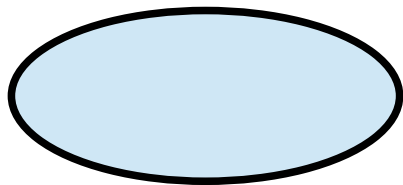
## Екземпляри на чисти примитиви

Тези схеми на представяне се основават на идеята за фамилия от обекти, като всеки член на фамилията се различава от останалите чрез стойностите на няколко параметъра.



## Екземпляри на чисти примитиви

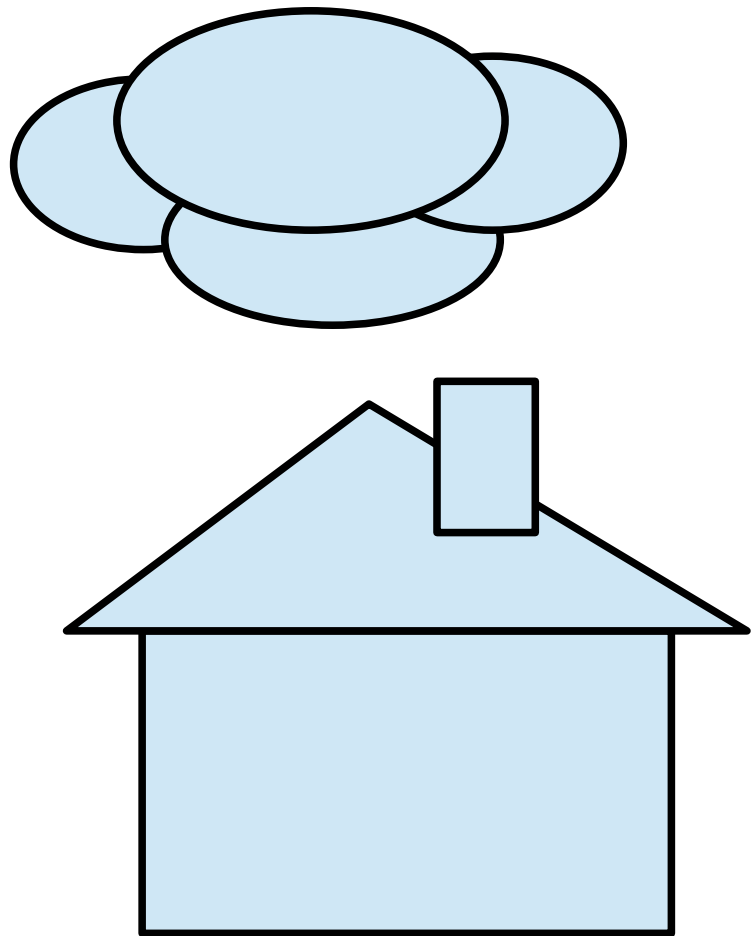
- ❖ Генериращ примитив – всяка фамилия от обекти. Представя се като име последвано от N-торка от променливи величини;
- ❖ Индивидуален обект – всеки обект от фамилията се нарича екземпляр на примитива и се представя като N+1-торка от стойности (или име последвано от N стойности).



ELLIPSE(A, B)

(ELLIPSE, 5, 2) или ELLIPSE(5, 2)

## Пример



Елипса (1.00, 3.00, 4.44, 5.26)

Елипса (3.00, 4.00, 7.44, 6.26)

Елипса (5.60, 2.70, 9.00, 5.00)

Елипса (2.37, 2.10, 7.50, 5.00)

Правоъгълник (2.70, 10.30, 9.70, 14.30)

Триъгълник (6.20, 7.30, 1.70, 10.30,  
10.70, 10.30)

Правоъгълник (6.60, 7.00, 7.90, 9.00)

## Недостатъци на схемата

- ❖ Липсата на средство за комбиниране на екземпляри на примитиви, с цел получаване на нови по-сложни обекти;
- ❖ Трудността на създаване на алгоритми за обработка на тези представяния, защото при добавяне на нов вид примитив, обикновено се налага преработката на всички алгоритми (но въпреки това в повечето случаи новите алгоритми са просто разширение на старите с поддръжка и на новия примитив).



## *В практиката*

- ❖ Схемите са сбити и лесни за използване;
- ❖ Удобни само, ако дефиниционната им област може да бъде сведена до каталог от малко на брой фамилии;
- ❖ Използват се съвместно със схеми, базирани на някаква конструктивна операция;
- ❖ В 2D се използват правоъгълници, елипси, линии, многоъгълници и др. В 3D се използват паралелепипеди, елипсоиди, сфери, линии, равнини, цилиндри, конуси, пирамиди, многостени и др.

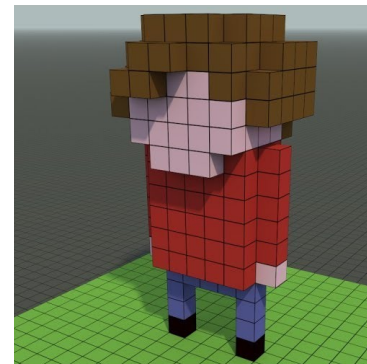
# *Изброяване на заетото пространство (Spatial occupancy enumeration)*

# Изброяване на заетото пространство

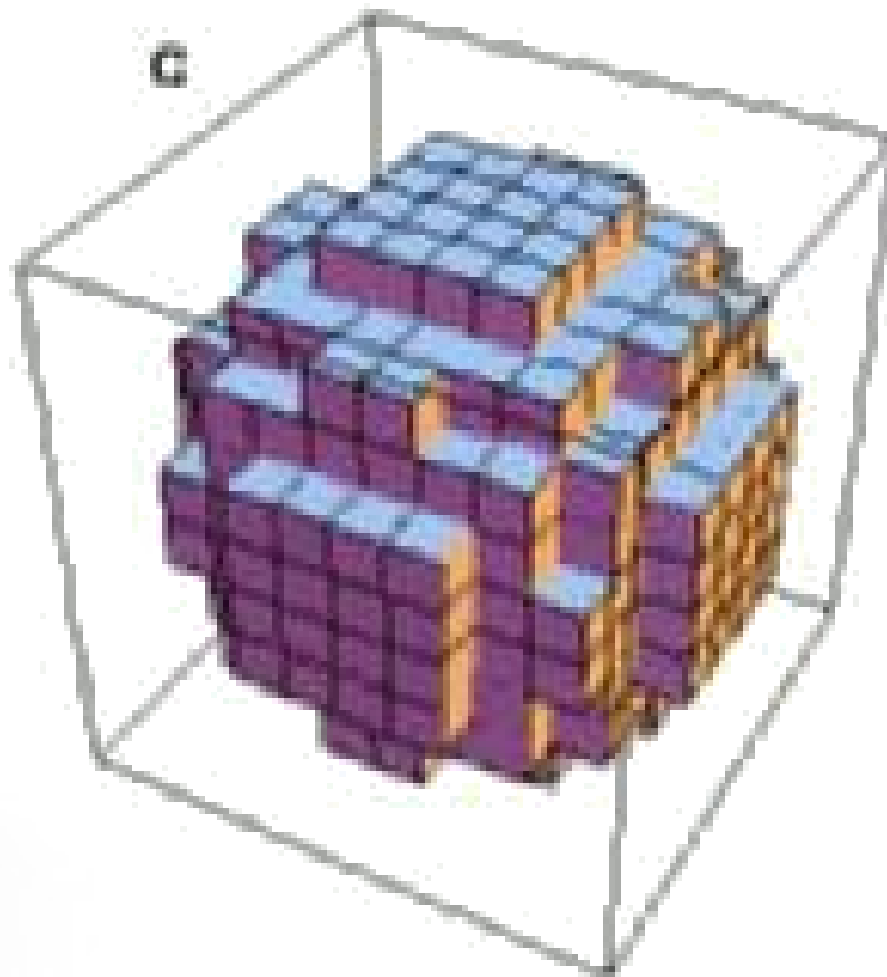
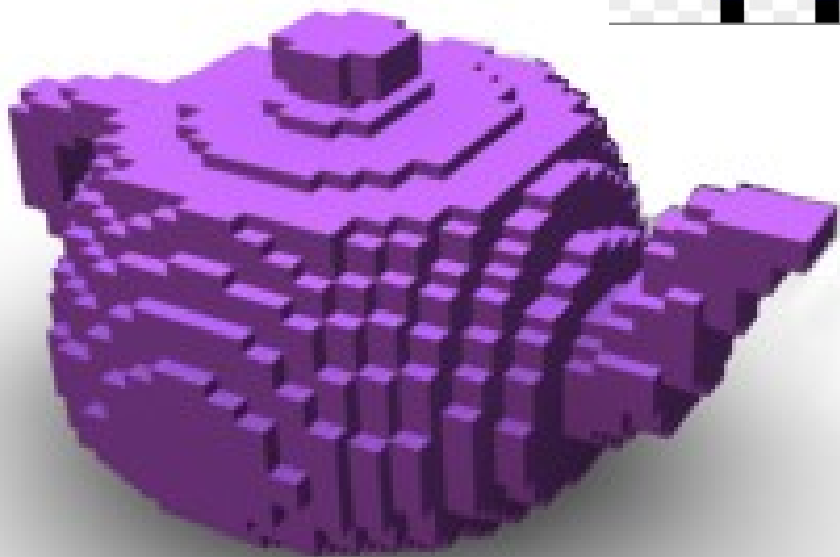
Всяко представяне на тяло, чрез тези схеми представлява списък на пространствени клетки, които тялото заема.

Пространствените клетки обикновено се наричат воксели  
(voxel = **v**olumetric **p**ixel).

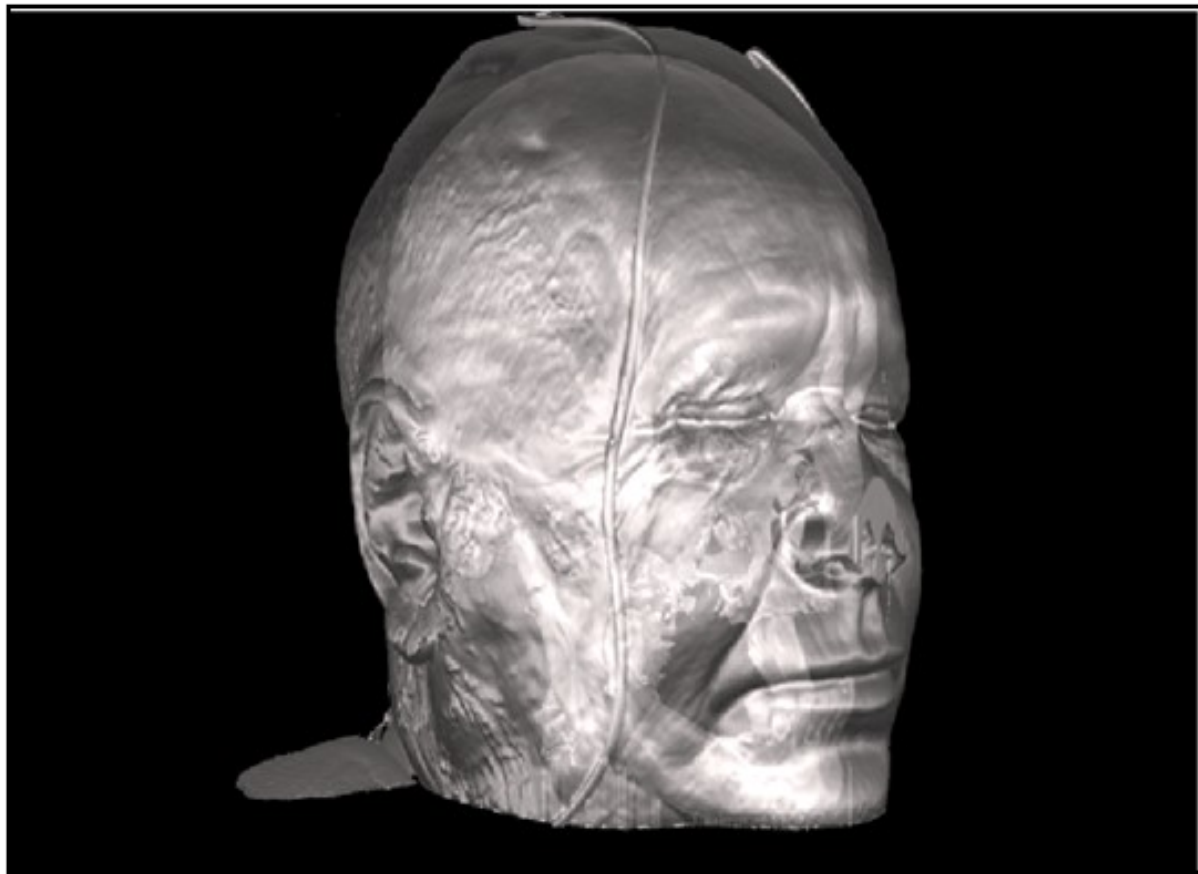
В двумерния (2D) случай клетките се наричат пискели  
(pixel = **p**ictures **e**lement).



# Примери



## *Пример – 3D изображение получено от томограф*



Получените данни от томографа (благодарение на сложен физичен процес и математическа обработка) могат да се визуализират, да се правят различни разреза и анализи и др.

# Свойства

- ❖ Кубове са наредени в пространствена мрежа;
- ❖ Всяко представяне на тяло чрез такава схема е (ненаредено) множество от наредени тройка числа;
- ❖ Към тройката числа може да има и други визуални характеристики, като плътност, цвят и др. Ако се описват всички клетки в отнапред известната мрежа, може да се използва подхода (bitmap/битмап) при който координатите не се съхраняват, а се съхраняват само визуалните характеристики, но винаги в определена последователност;
- ❖ Тези схеми имат твърде много общи свойства със схемите, базирани на разбиване на телата на части с непресичаща се вътрешност.



# *Разбиване на клетки*

## *(Cell decomposition)*



## *Разбиване на клетки*

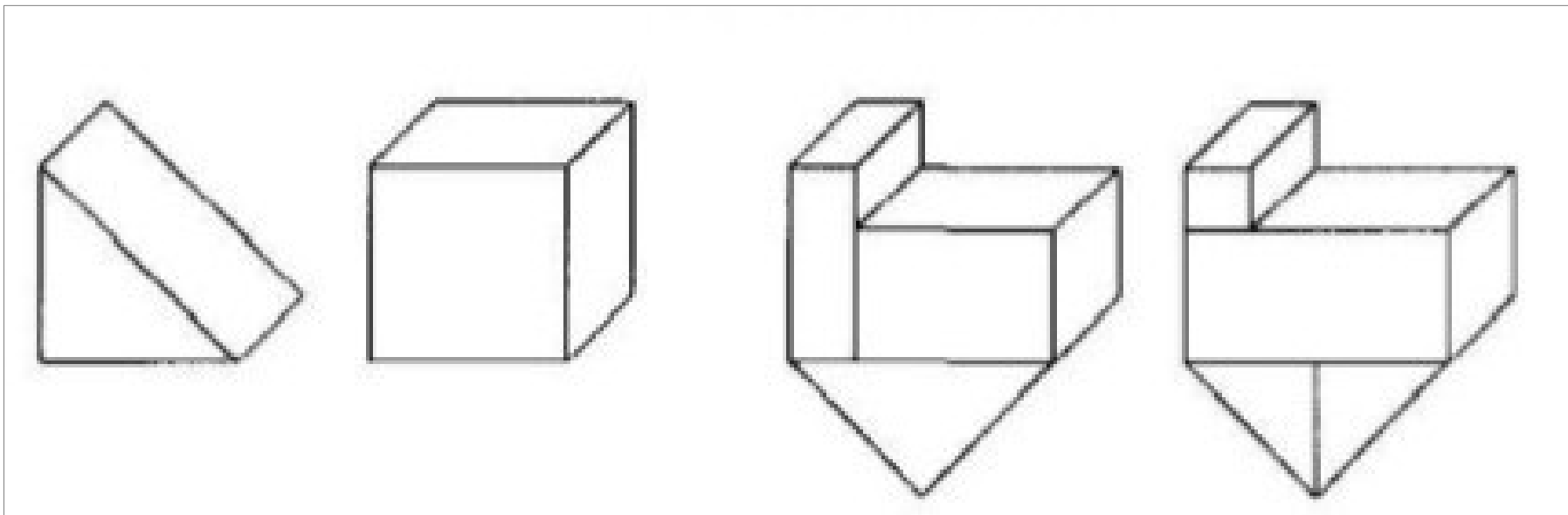
Всяко представяне на тяло, чрез тези схеми представлява списък на пространствени клетки, които тялото заема.

Пространствените клетки може да бъдат с различна форма и размери.





## Пример



Обърнете внимание, че представянето с тези схеми може да не е еднозначно. Тялото в дясно на примера може да се разбие по много начини на различни клетки, но това обикновено не е проблем.

# Свойства

- ❖ Схемите **Изброяване на заетото пространство** са частен случай на схемите **Разбиване на клетки**;
- ❖ Понеже клетките може да са с произволна форма, то за разлика от подхода в изброяване на заетото пространство, е необходимо да се описва геометрията на целите клетки (например всички върхове на многостена, вместо само центъра му). Това в повечето случаи води до използване на повече памет;
- ❖ Често за клетки се използват тетраедри;
- ❖ Използват се за създаване на специални ефекти, симулации и други.

# *Кодиране с осмично дърво*

## *(Octree)*

## Кодиране с осмично дърво

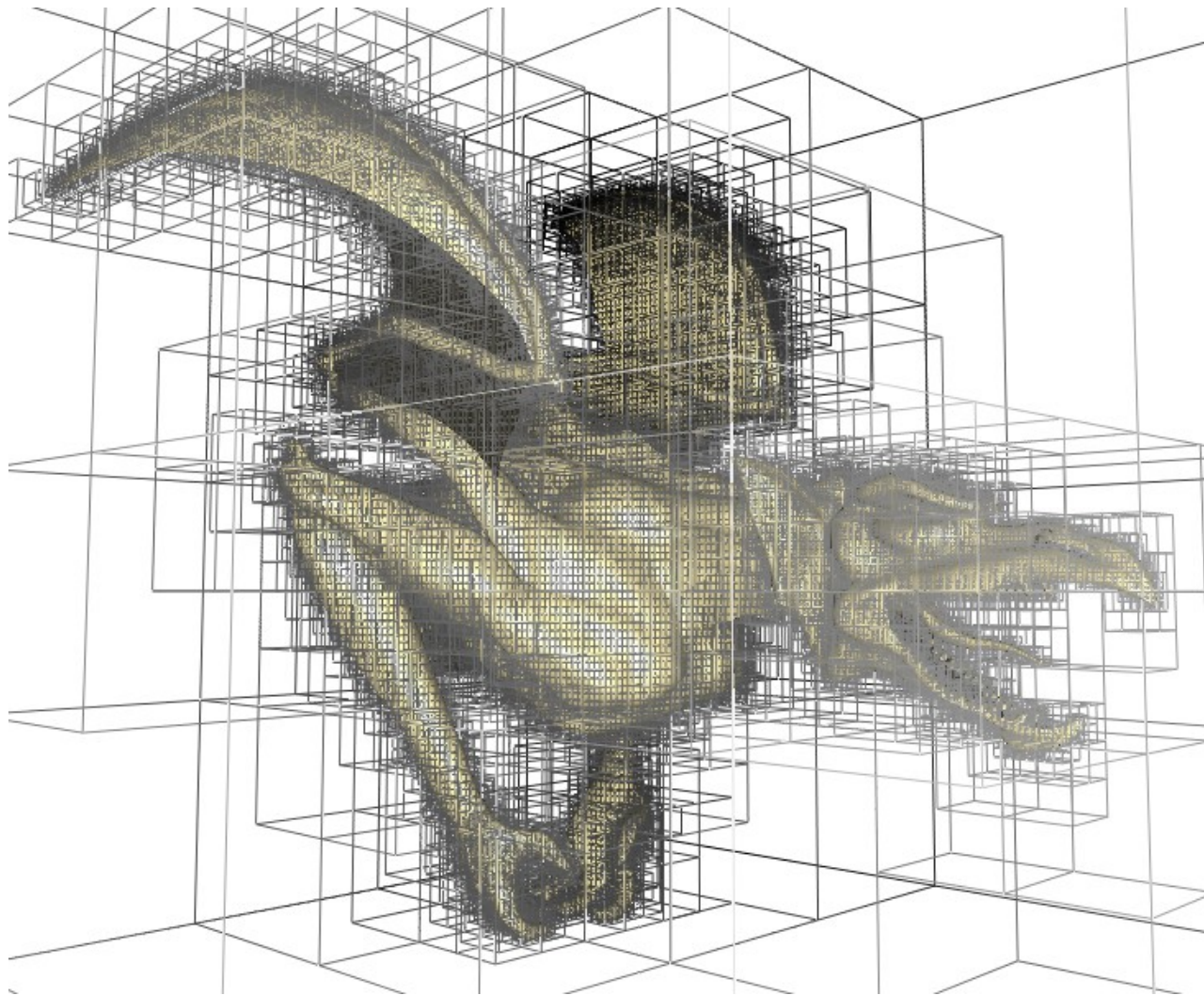
Всяко представяне на тяло, чрез тези схеми може да се представи с дървовидна структура (осмично дърво), като всяко ниво в дървото може да се разгледа като различна детайлизация на модела.

Всеки възел в дървото може да е:  
Празен (**E**mpy), Запълнен (**F**ull) или  
Частично запълнен (**P**artial).

**E** и **F** са листа в дървото, а **P** е под дърво.

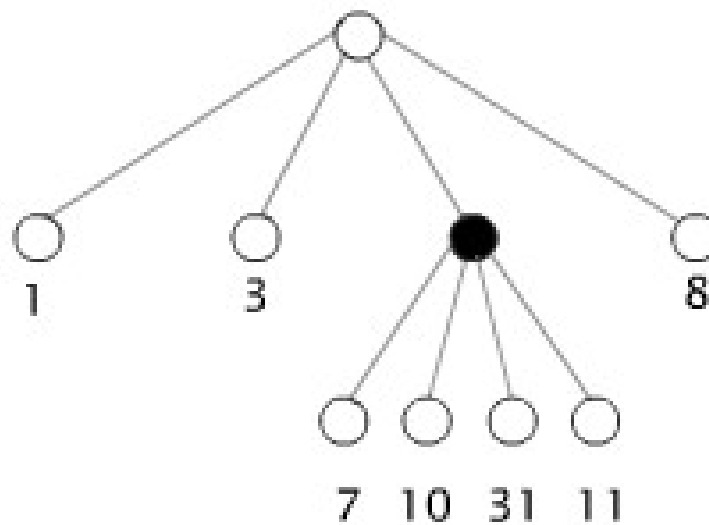
## Пример

Описанието на тялото започва от един голям куб, след което той се дели на 8 октанта, всеки от които е или пълен **F** или празен **E** (при тези случаи приключваме описанието на съответния подкуб). В противен случай разделянето продължава.

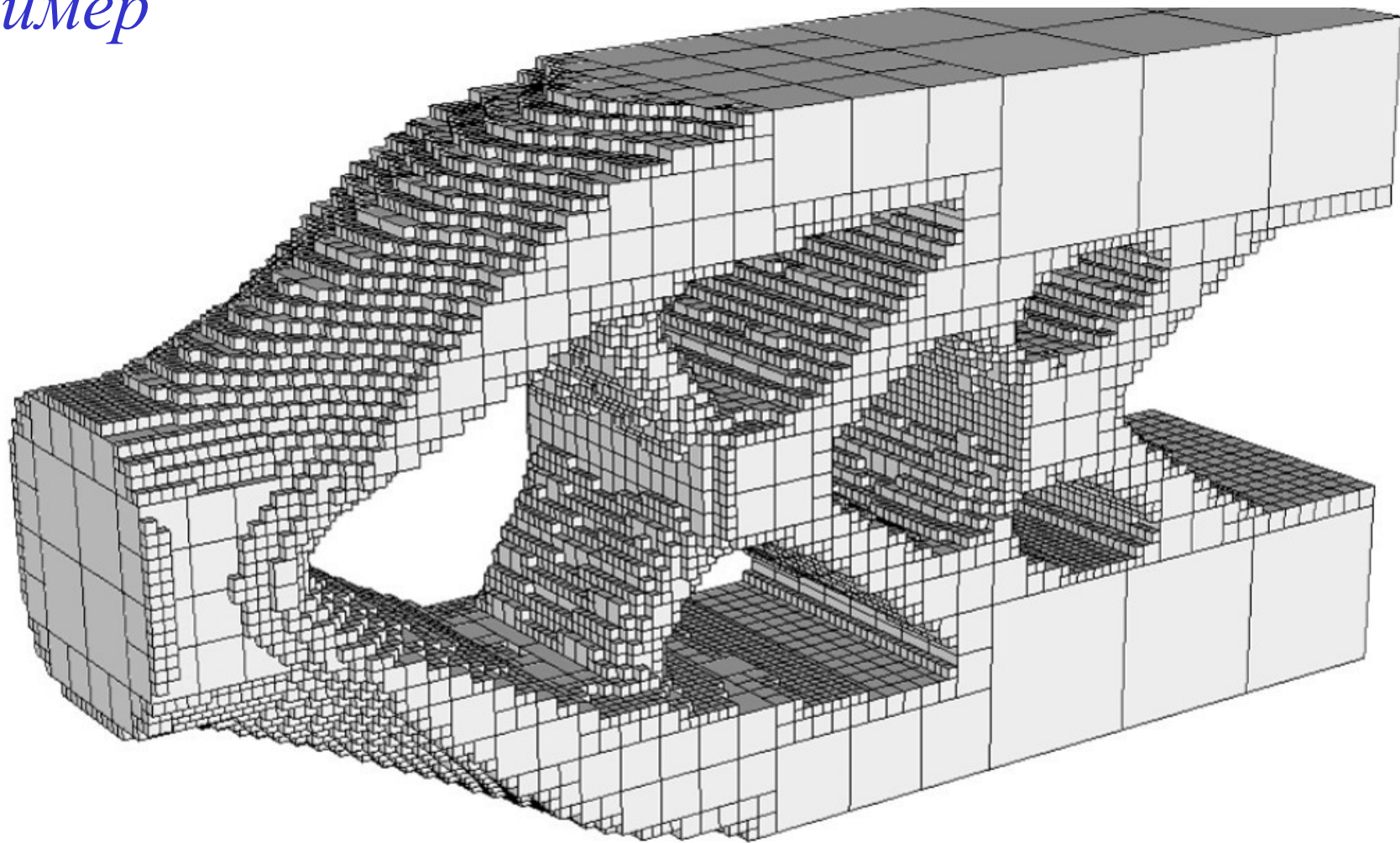


## Пример (четвъртично дърво)

1	1	3	3
1	1	3	3
7	10	8	8
31	11	8	8



# Пример



# Свойства

- ❖ В двумерния вариант схемата се нарича кодиране с четвъртично дърво. Едномерният – двоично дърво.
- ❖ Обикновено представянето е по-сбито в сравнение с Изброяване на заетото пространство;
- ❖ Много често осмичното дърво (както и други подобни на него дървета) се използват за ускоряващи структури в някои алгоритми за визуализация и др.;



# *Конструктивна геометрия на твърди тела*

*(Constructive Solid Geometry, CSG)*

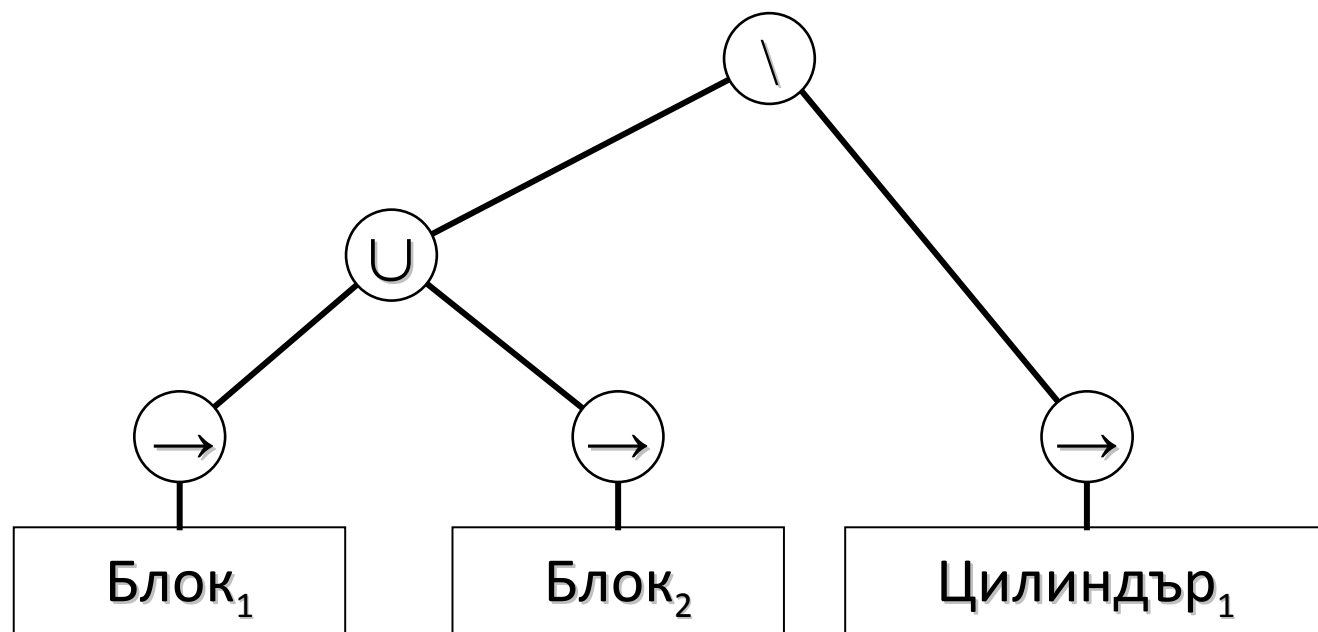
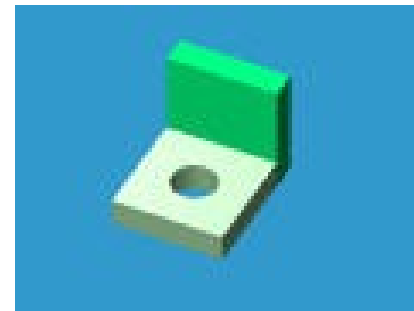
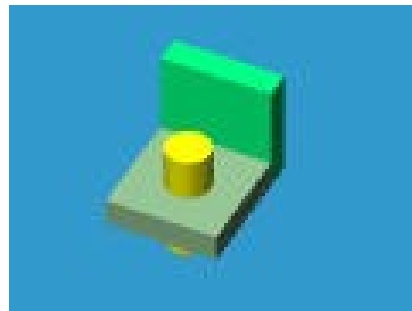
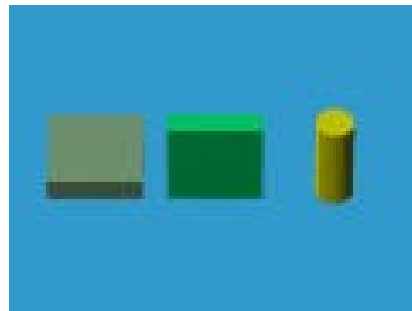
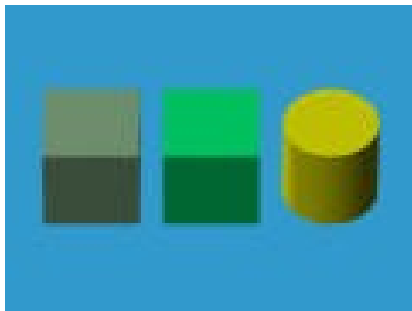


# *Конструктивна геометрия на твърди тела*

Метод за моделиране на твърди тела, който комбинира използването на прости форми, наречени примитиви за изграждането на сложни модели посредством булевите оператори: Обединение, Разлика и Сечение.



# Конструктивна геометрия на твърди тела



Конструктивно дърво:

Състои се от възли с примитиви, възли с трансформации и възли с теоретико-множествени операции.

# Свойства

- ❖ За лесно генериране на описание се използва и конструктивно дърво;
- ❖ Схемите са подходящи за лесно създаване на модели от човек;
- ❖ Много често се използват в CAD/CAM системите;

# *Описание на границата*

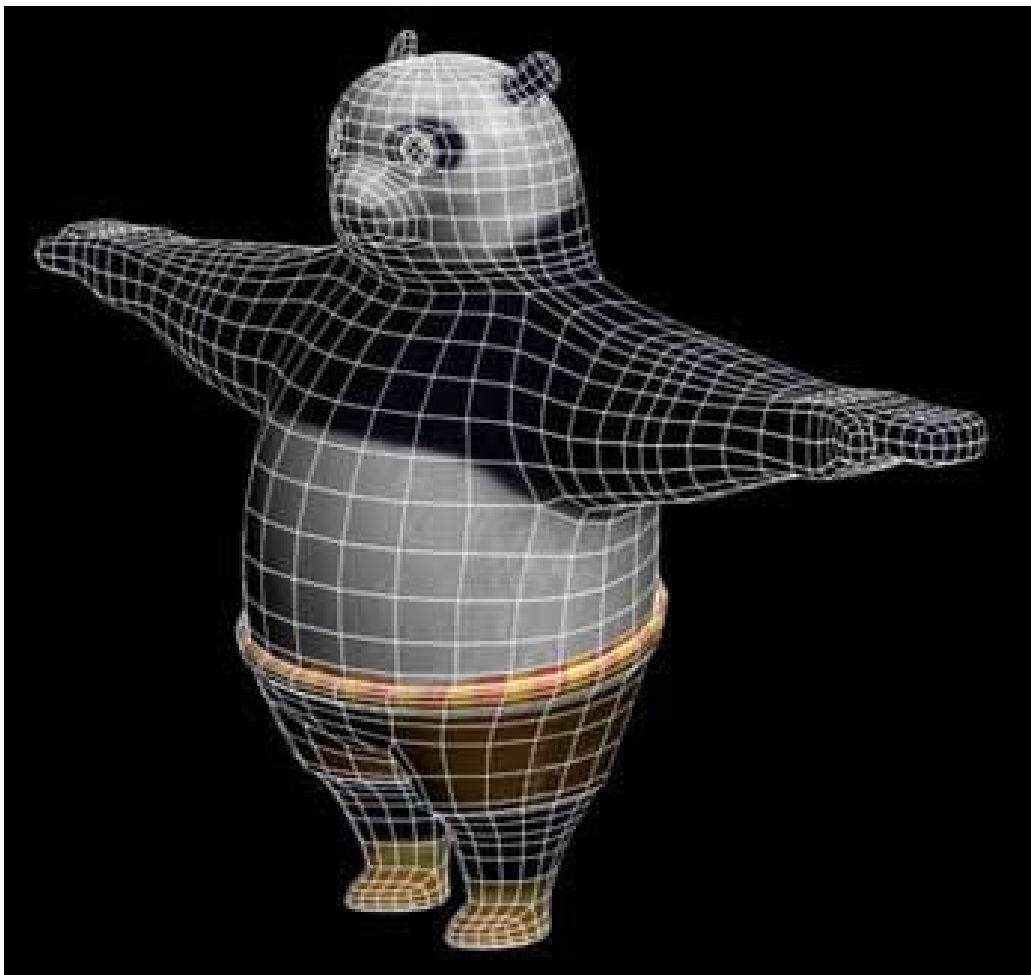
## *(Boundary Representation, B-Rep)*

## Описание на границата

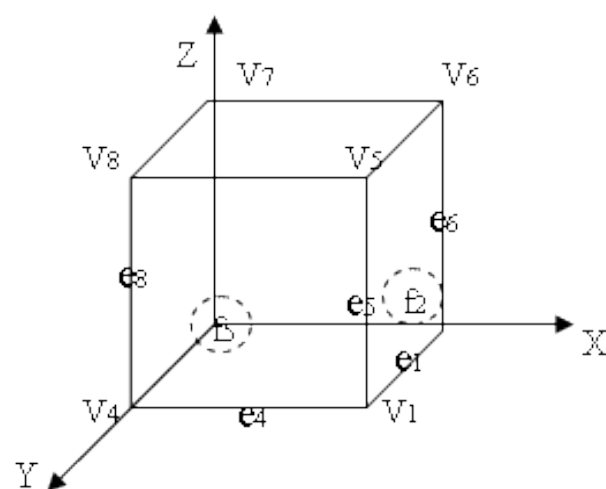
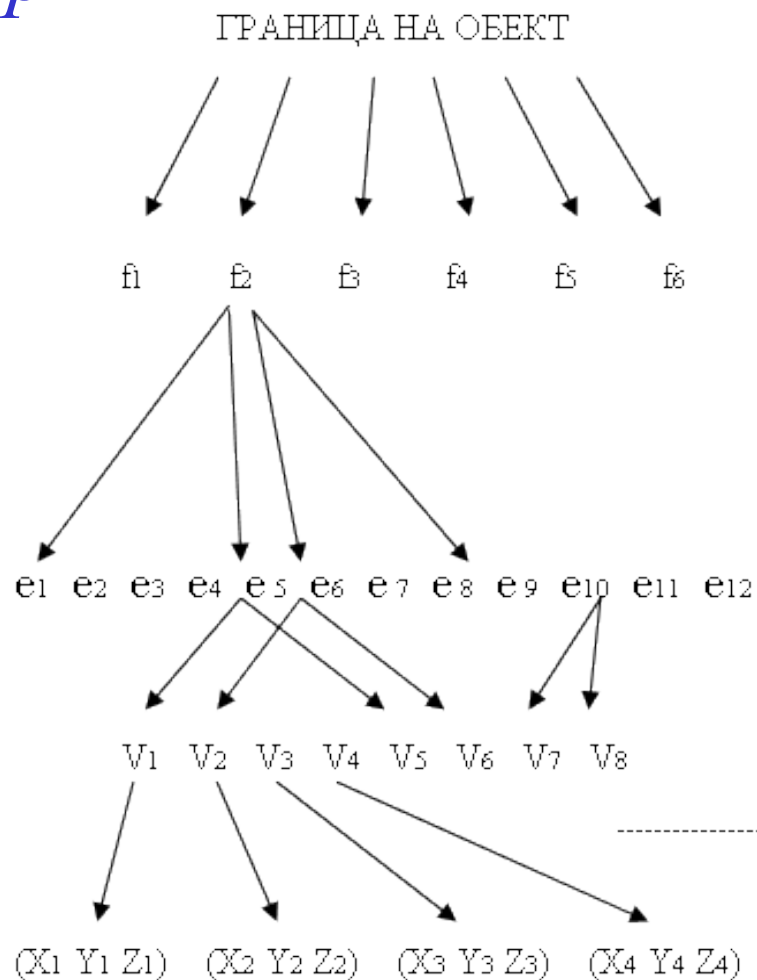
Моделите се представят като обединение на някакви пространствени лица (наречени още „кръпки“), описващи границата на модела.

Най-често за такива лица се използват триъгълници или четириъгълници.

# Пример 1



# Пример 2



КОМБИНАТОРНИ  
СТРУКТУРИ

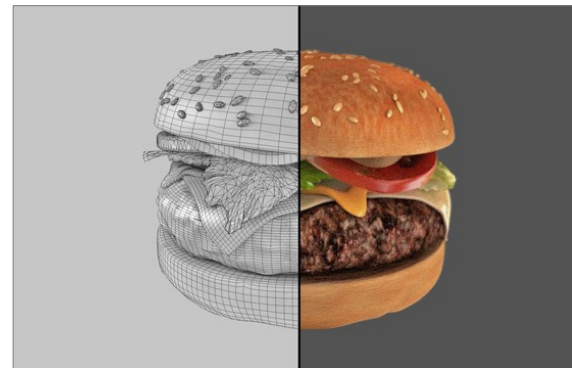
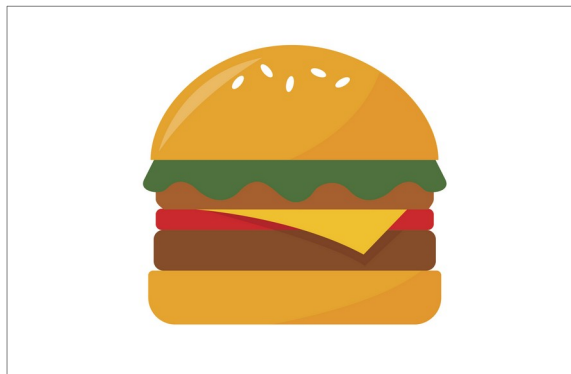
МЕТРИЧНА  
ИНФОРМАЦИЯ





# Свойства

- ❖ Една от най-често използваните представящи схеми в момента;
- ❖ Схемите са подходящи за лесно създаване на модели от човек;
- ❖ Съществуват ефективни алгоритми за обработка и визуализация.



# Функционално представяне

*(Functional Representation, F-Rep)*

## Функционално представяне

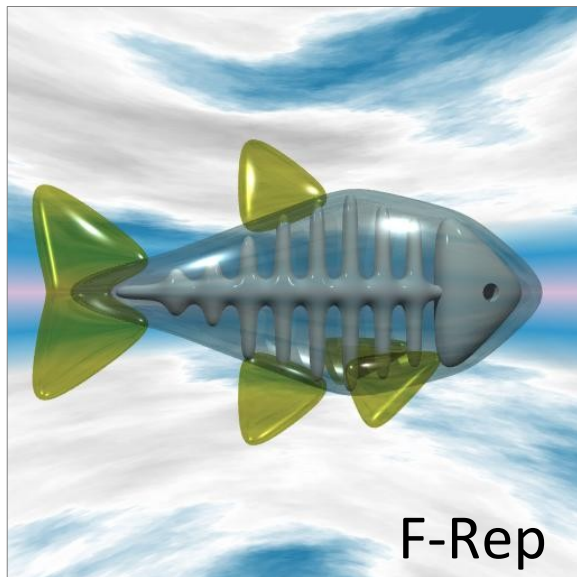
Телата се описват неявно чрез тримерна функция  
 $F(x,y,z)$

Точките в пространството принадлежащи на тялото  
отговарят на условието  $F(x,y,z) < 0$

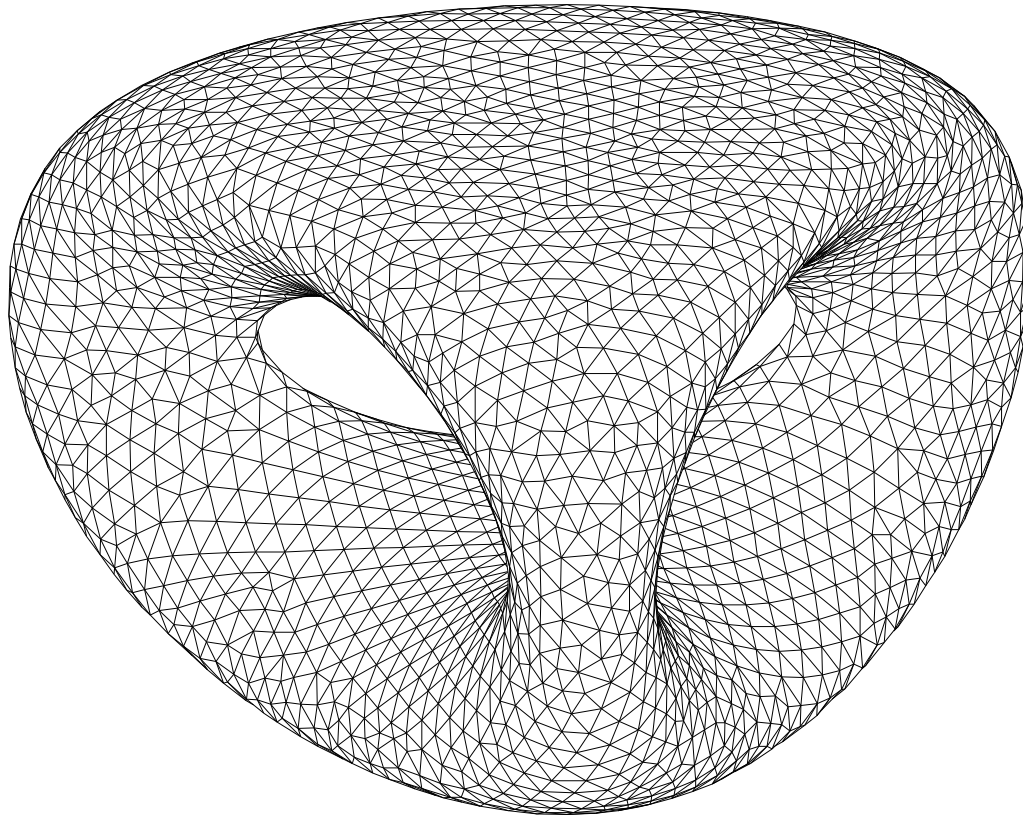
$F(x,y,z) = 0$  е границата на тялото.



# Пример (F-Rep)



## Пример (Genus 2) – полионизирана повърхност



$$2y(y^2 - 3x^2)(1 - z^2) + (x^2 + y^2)^2 - (9z^2 - 1)(1 - z^2) = 0$$

# *Запълващи продукции*

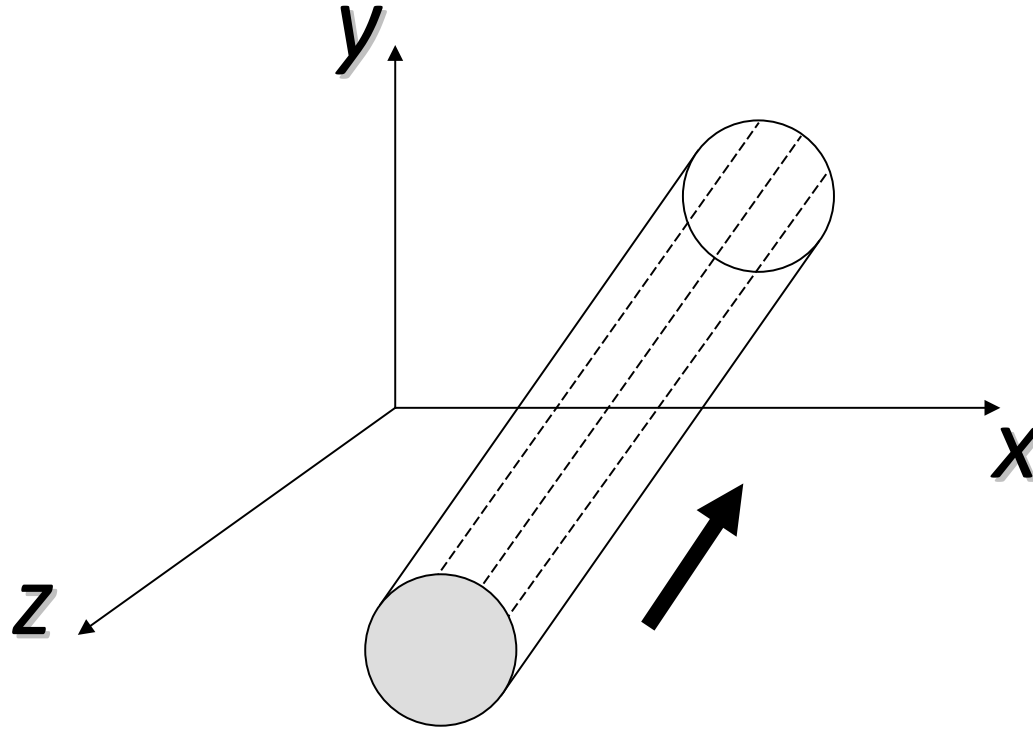
## *(Sweeping)*

# *Запълващи продукции*

Всяко множество, което се движи през пространството, “замита” тяло, което може да бъде представено чрез представяне на движещото се тяло и траекторията му на движение.

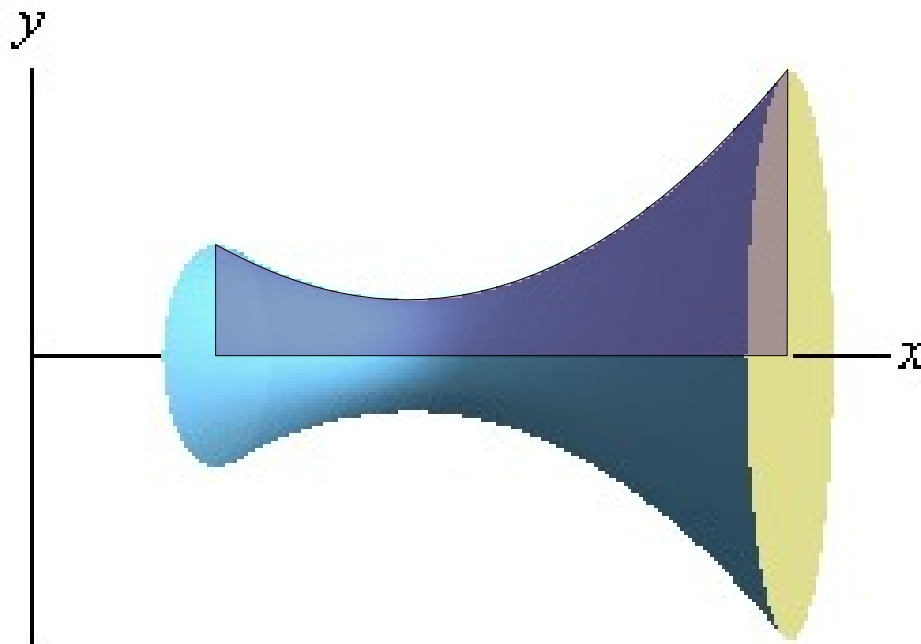
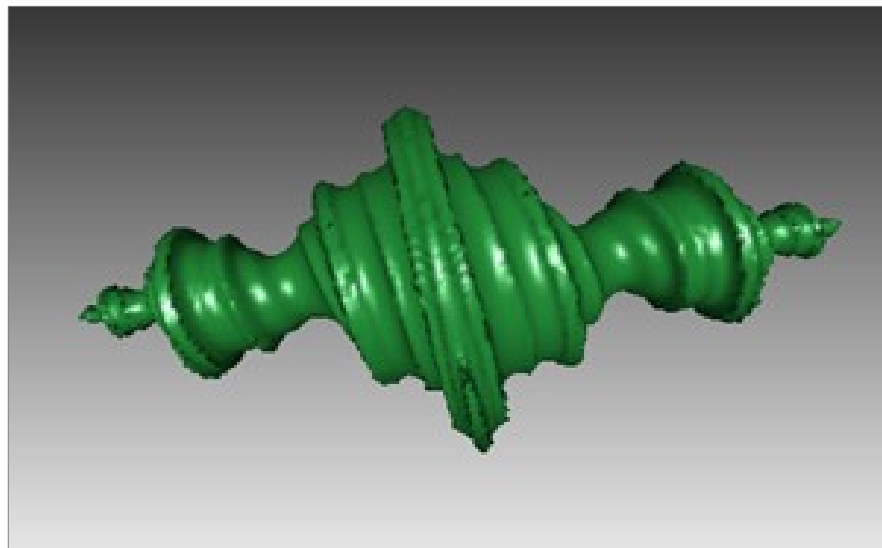


# Запълващи продукции





# Пример – ротационни тела



# *Метатопчета*

## *(Metaballs)*



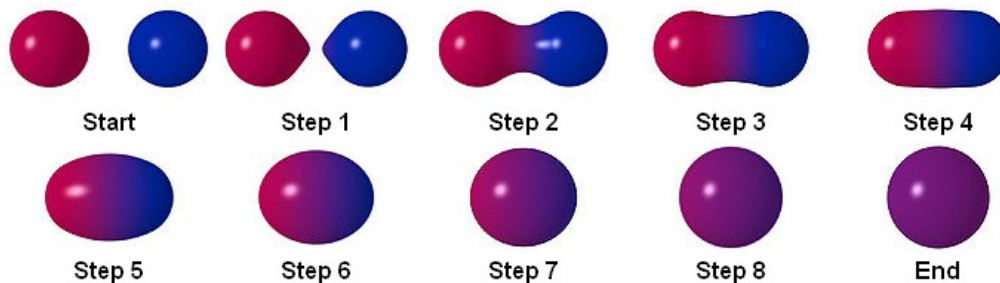
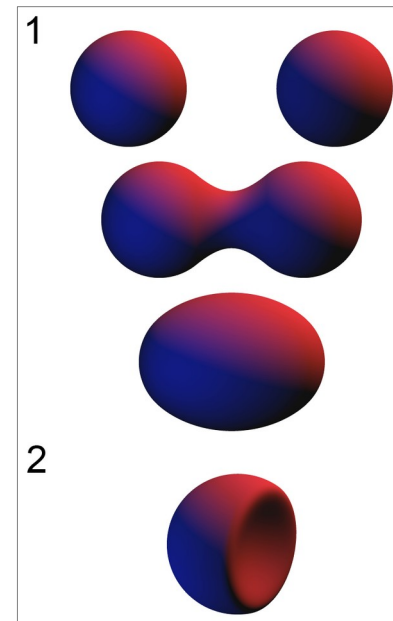
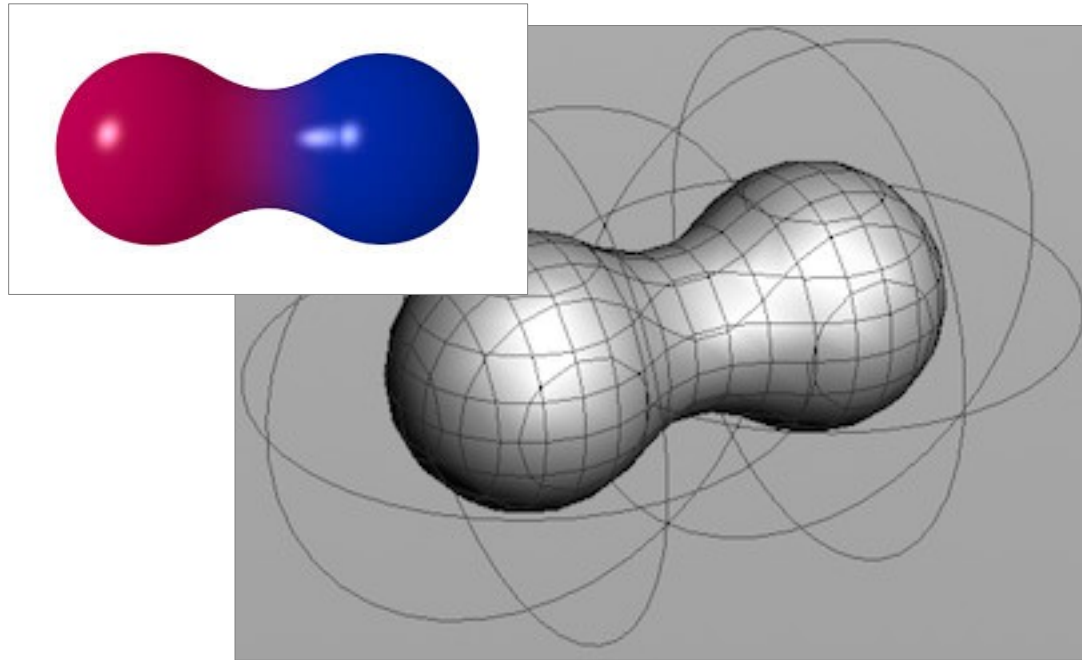
# Метатопчета

Всяко „тяло“ се представя като съставено от множество малки сфери, които когато са близо една до друга се сливат гладко помежду си.

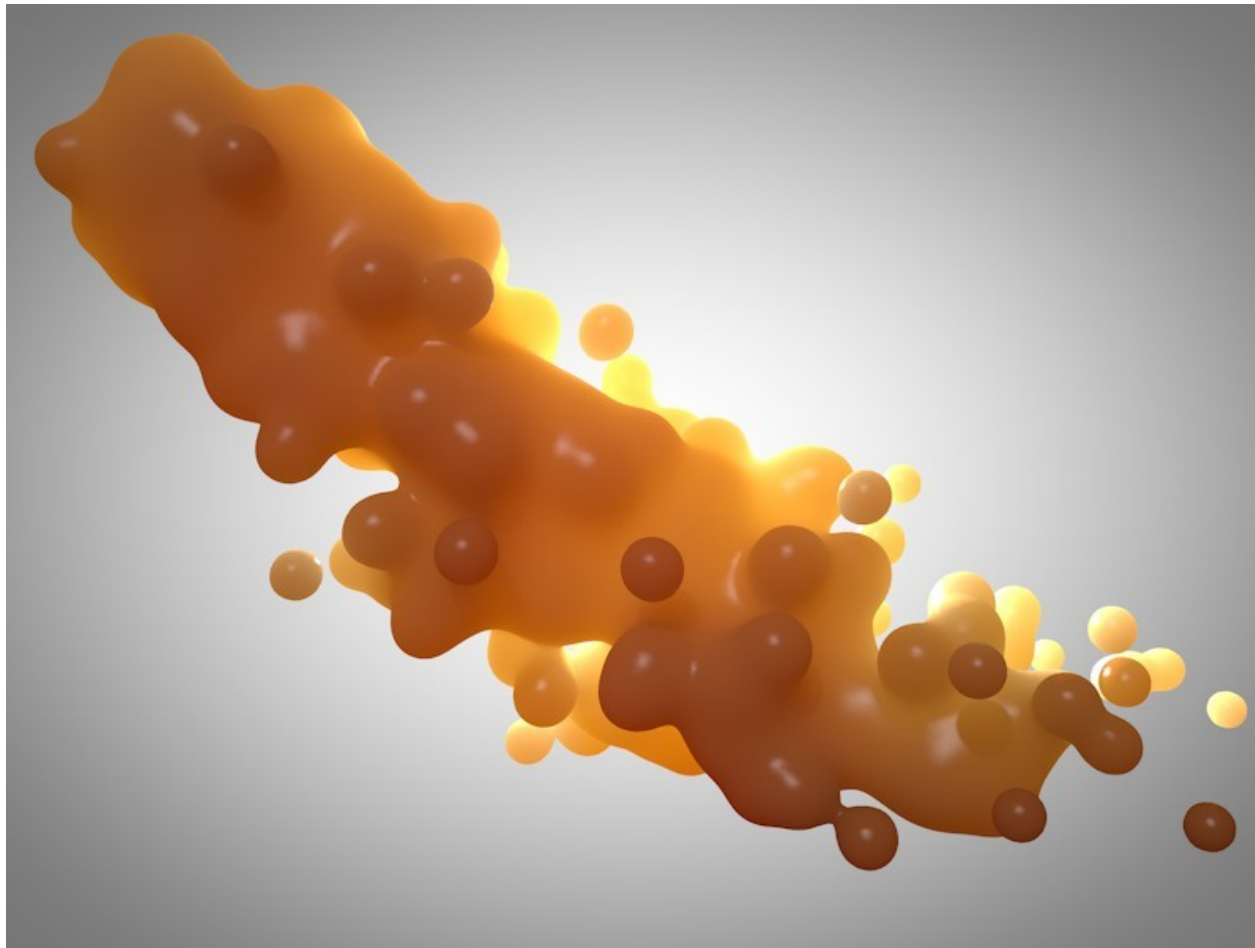
Много често тези схеми се използват за създаване на специални ефекти или „меки“ тела.



# Метатопчета



# Метатопчета



# *Системи частици*

## *(Particle systems)*



# Системи частици

Всяко „тяло“ се представя като съставено от множество малки частици (точки, триъгълници, фасети, ...), които се създават, премахват и/или променят характеристиките си по някакви закони.

Много често тези схеми се използват за създаване на специални ефекти или за моделиране и визуализация на природни стихии, като огън, пушек, сняг, дъжд, торнадо и др.

# Системи частици





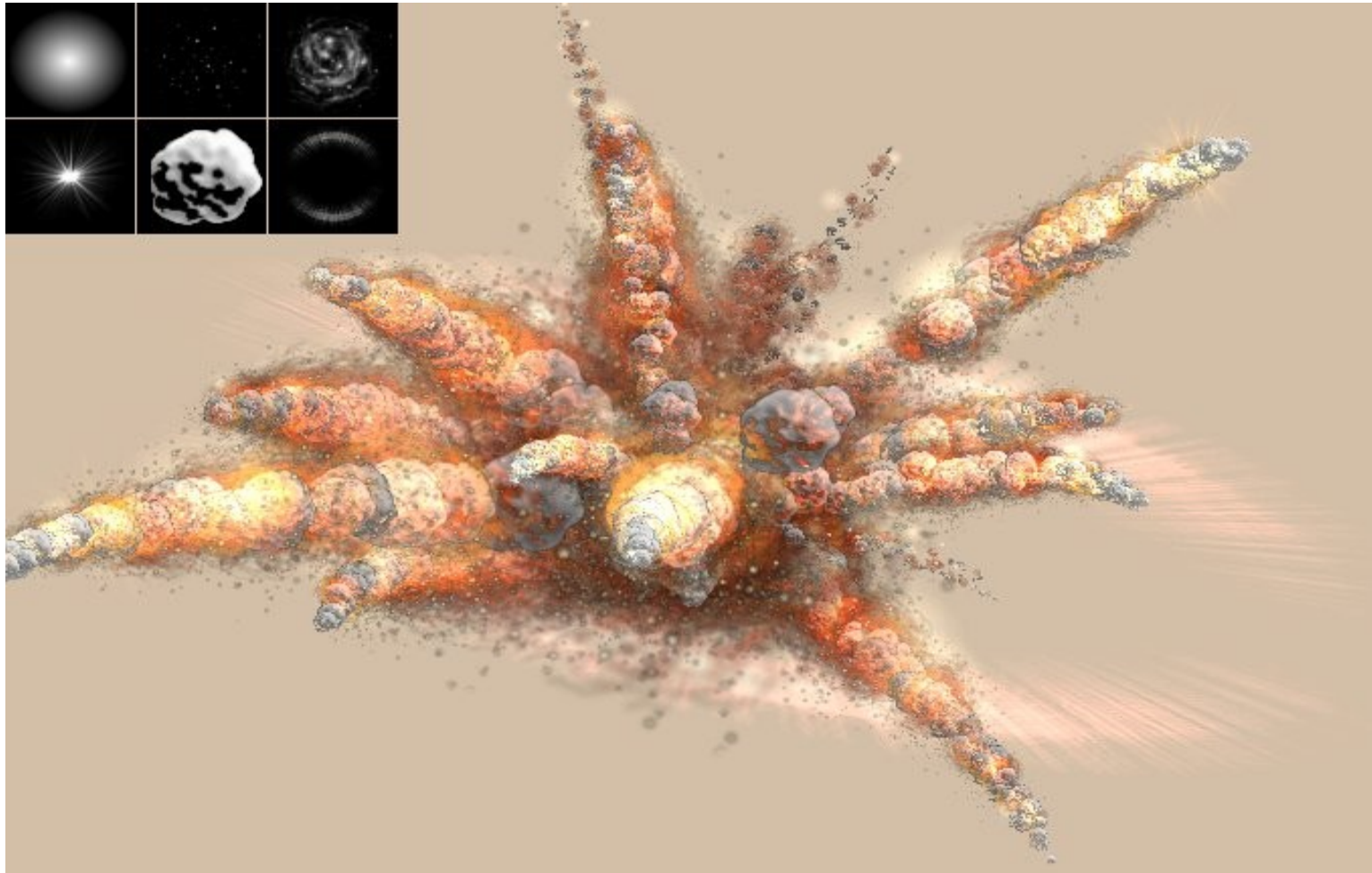
# Системи частици



# Системи частици



# Системи частици



# Системи частици (коса)



Рапунцел и разбойникът:

Косата е моделирана със специален много сложен математически модел, който позволява на аниматорите лесно да постигнат естественост на движенията ѝ.





## *Системи частици (козина)*



# Въпроси?

*apenev@uni-plovdiv.bg*

