
Solid Modeling Data Structures

How does the computer store solid models?

MEEM5408 Design Automation: Theory and Implementation

MichiganTech

What is a “Data Structure”?

- In computer programming, it is a **grouping of variables**.
- Some of the variables can be references to other data structures.
- For example: a point and line can be represented as:

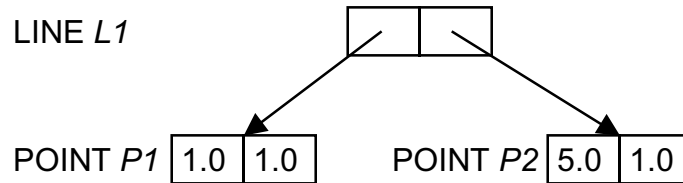
```
struct POINT { real x, real y };  
struct LINE { POINT *start, POINT *end };
```

MEEM5408 Design Automation: Theory and Implementation

MichiganTech

Using the Data structure

This diagram helps to visualize the use of data structures. Think of each block as a memory location.

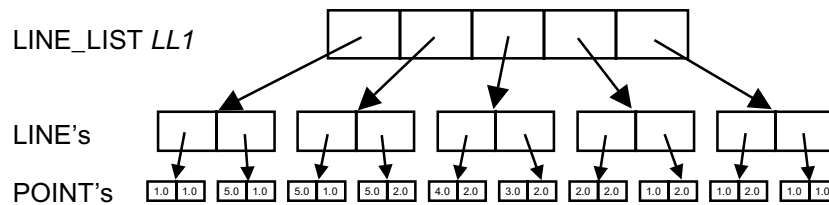


A program could then use this information to draw the corresponding geometry or do other calculations such as length.

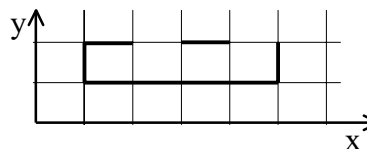
Lists – Arrays

An “array” can hold a list of data items.

E.g., `struct LINE_LIST {LINE *array[5]};`



This information can be drawn as:



Linked List: a different kind of list

- “Linked list” data structures are common in Geometric Modeling.
- They can be represented by the structure:

```
struct LINK
{ XXX *element, LINK *next };
```

where *XXX* is the type of entity being listed.

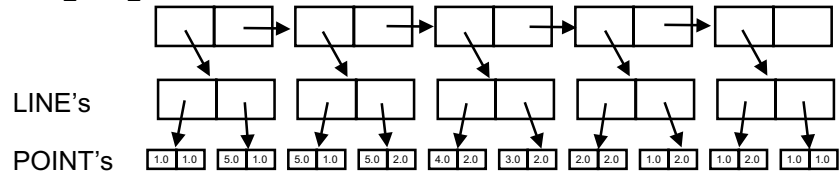
- Each list link always has a pointer to the next one.

A linked list of lines

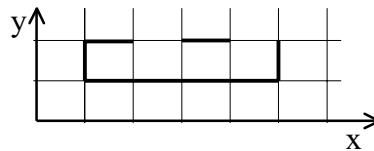
E.g. `struct LINE_LIST_LINK`

```
{ LINE *line, LINE_LIST_LINK *next };
```

LINE_LIST_LINK LN1



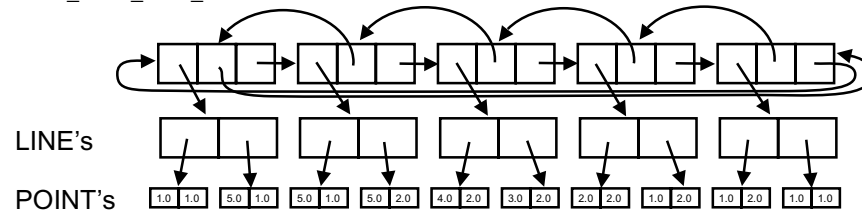
This information can also be drawn as:



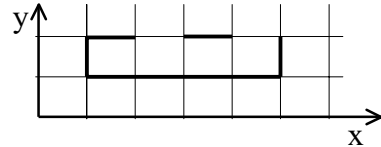
Doubly linked lists

```
E.g. struct LINE_LIST_DBL_LINK    { LINE *line,  
    LINE_LIST_DBL_LINK *prev, LINE_LIST_DBL_LINK *next};
```

LINE_LIST_DBL_LINK LD1



This information can also
be drawn as:



MEEM5408 Design Automation: Theory and Implementation

MichiganTech

Types of Solid Modeling Representations

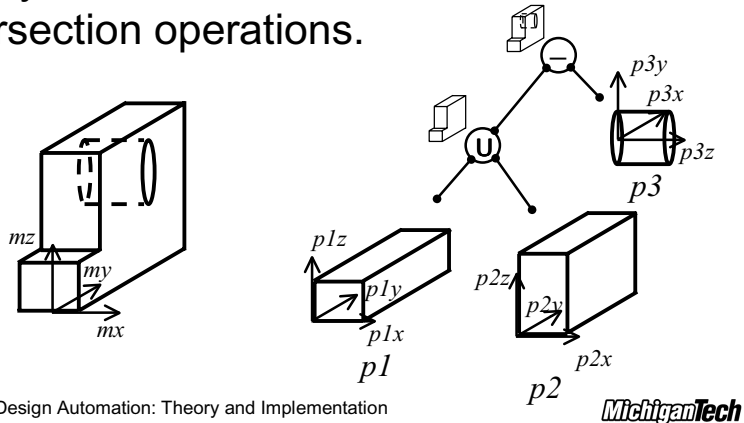
- Constructive Solid Geometry (CSG)
- Boundary Representation (B-Rep)
- Decomposition Representations

MEEM5408 Design Automation: Theory and Implementation

MichiganTech

CSG Tree Structure

- The CSG solid model is made up of a binary tree of union, subtraction and intersection operations.



MEEM5408 Design Automation: Theory and Implementation

MichiganTech

CSG Data Structure

```

struct operator { integer op_type, // 0 – union, 1 – intersect, 2 – subtract
                 left_type, // 0 – operator, 1 – primitive
                 right_type; // 0 – operator, 1 – primitive
void *left_struct, // left operator or primitive
    *right_struct, // right "
    *parent_struct; // parent "
};

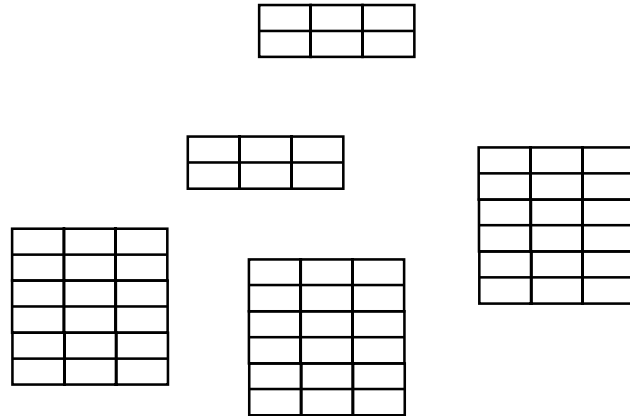
struct primitive { integer prim_type; // 0 – block, 1 – cylinder, 2 – ...
                  real pos_x, pos_y, pos_z, // position
                    dirx_x, dirx_y, dirx_z, // x orientation
                    diry_x, diry_y, diry_z, // y orientation
                    dirz_x, dirz_y, dirz_z; // z orientation
                  real params[10]; // parameters w, h, etc.
};
    
```

MEEM5408 Design Automation: Theory and Implementation

MichiganTech

CSG Data Structure

- The previous CSG solid can be represented in the following diagram

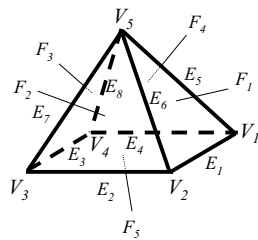


MEEM5408 Design Automation: Theory and Implementation

MichiganTech

Simple B-Rep Data Structure

A pyramid and tables to describe the pyramid.



Face Table

Face: Edges

$F_1: E_1, E_5, E_6$
 $F_2: E_2, E_6, E_7$
 $F_3: E_3, E_7, E_8$
 $F_4: E_4, E_8, E_5$
 $F_5: E_1, E_2, E_3, E_4$

Edge Table

Edge: Vertices

$E_1: V_1, V_2$
 $E_2: V_2, V_3$
 $E_3: V_3, V_4$
 $E_4: V_4, V_1$
 $E_5: V_1, V_5$
 $E_6: V_2, V_5$
 $E_7: V_3, V_5$
 $E_8: V_4, V_5$

Vertex Table

Vertex: Coord.

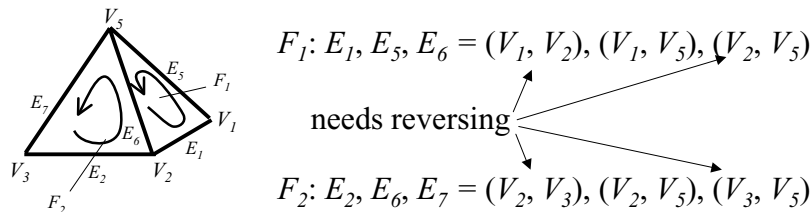
$V_1: x_1, y_1, z_1$
 $V_2: x_2, y_2, z_2$
 $V_3: x_3, y_3, z_3$
 $V_4: x_4, y_4, z_4$
 $V_5: x_5, y_5, z_5$

MEEM5408 Design Automation: Theory and Implementation

MichiganTech

Limitations

- Can only represent planar surfaces
- No internal face boundaries (no holes)
- Path traced along edge of face has incorrect vertex ordering for some faces



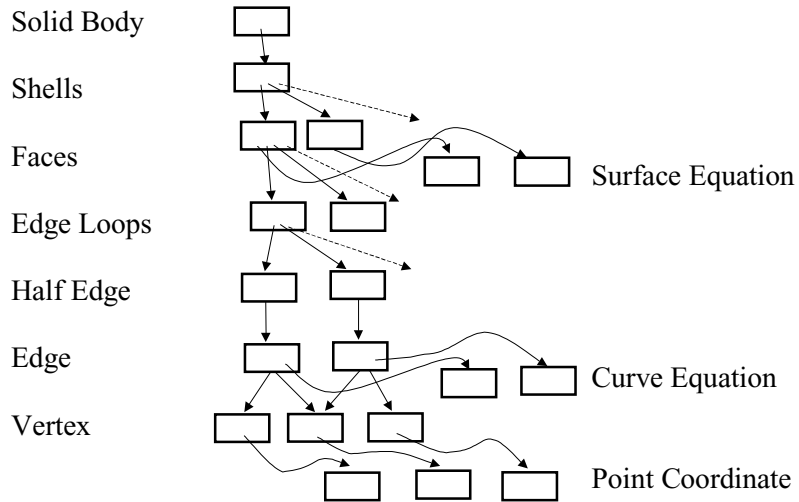
Each edge will need reversing for one face but not the other.
 MEEM5408 Design Automation: Theory and Implementation *MichiganTech*

Half-Edge B-Rep structure

The half-edge B-Rep structure solves these problems by:

- Curve and surface equations are associated with edges and faces.
- Each face can have multiple loops.
- Each face is considered to own half of an edge, and a separate data structure identifies if the vertex ordering should be reversed.

Half-Edge Data Structure



MEEEM5408 Design Automation: Theory and Implementation

MichiganTech

Decomposition Representations

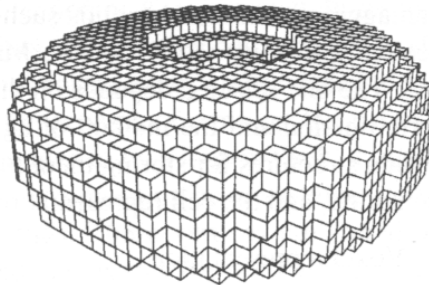
- Voxel Representation
- Octree Representation
- Cellular Decomposition

MEEEM5408 Design Automation: Theory and Implementation

MichiganTech

Voxel Representation

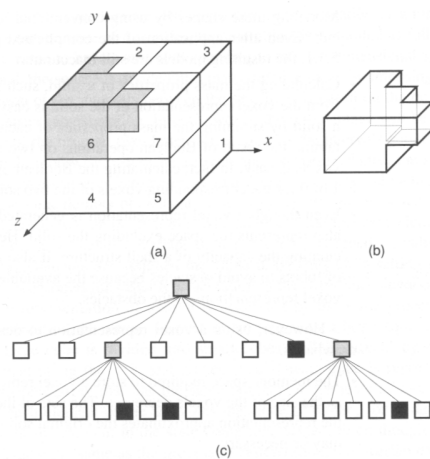
In this representation a cubic volume is divided into a three dimensional grid of smaller cubes, known as voxels (volume elements).



From K. Lee, "Principles of CAD/CAM/CAE Systems," Addison-Wesley, 1999
MEEM5408 Design Automation: Theory and Implementation *MichiganTech*

Octree Representation

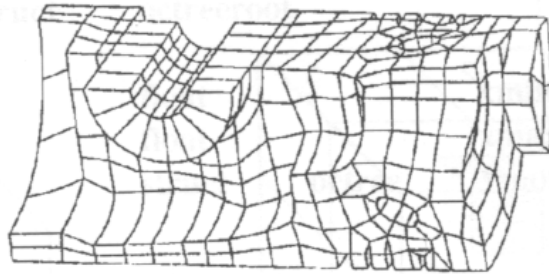
- Similar to the voxel representation, except that the structure is hierarchical, rather than using an array.
- A larger cube is divided into 8 smaller cubes only if the larger cube contains points both inside and outside the solid being approximated.



From K. Lee, "Principles of CAD/CAM/CAE Systems," Addison-Wesley, 1999
MEEM5408 Design Automation: Theory and Implementation *MichiganTech*

Cell Representation

- In this representation, the cells can have any shape (don't have to be cubes).
- Finite Elements are one example.



From K. Lee, "Principles of CAD/CAM/CAE Systems," Addison-Wesley, 1999
MEEM5408 Design Automation: Theory and Implementation *MichiganTech*