Solid Modeling: a cousin of PM

CS535

Daniel G. Aliaga
Department of Computer Science
Purdue University

(slides based on Ajay Joneja, HKUST)

Solid Modeling

History:

CNC: ~1950

Mainframe Computers: ~1960's

BREP: 1970 (Baumgart)

CSG: 1974 (Ian Braid)

Computerized Drafting

Advantages:

Saves on storage/retrieval;

Easy modification, update;

Shortcomings:

Can't analyse the

strength, shape,

geometry, weight

center of mass, center of inertia

Popular Commercial tools: AutoCAD, CADKEY…

Constructive Solid Geometry (CSG)

Introduced: Ian Braid (Cambridge University, ~74)

Concepts:

Primitives: small set of shapes

Transformations: scaling, Rotation, Translation

Set-theoretic Operations Union, Intersection, Difference

*[**Euler operators**]*

Combinations of these → Solid part

Euler operators

U* (regular union)

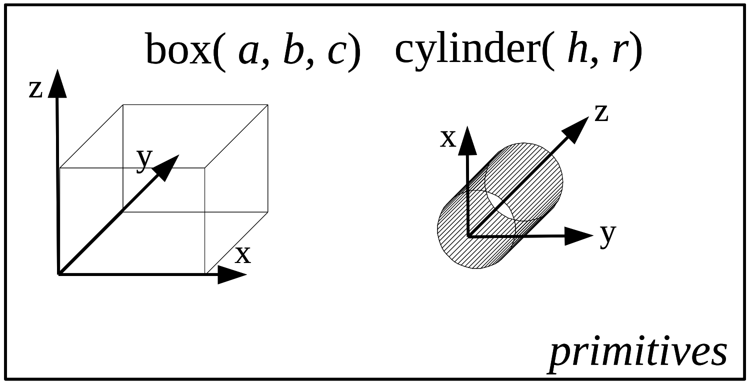
-* (regular difference)

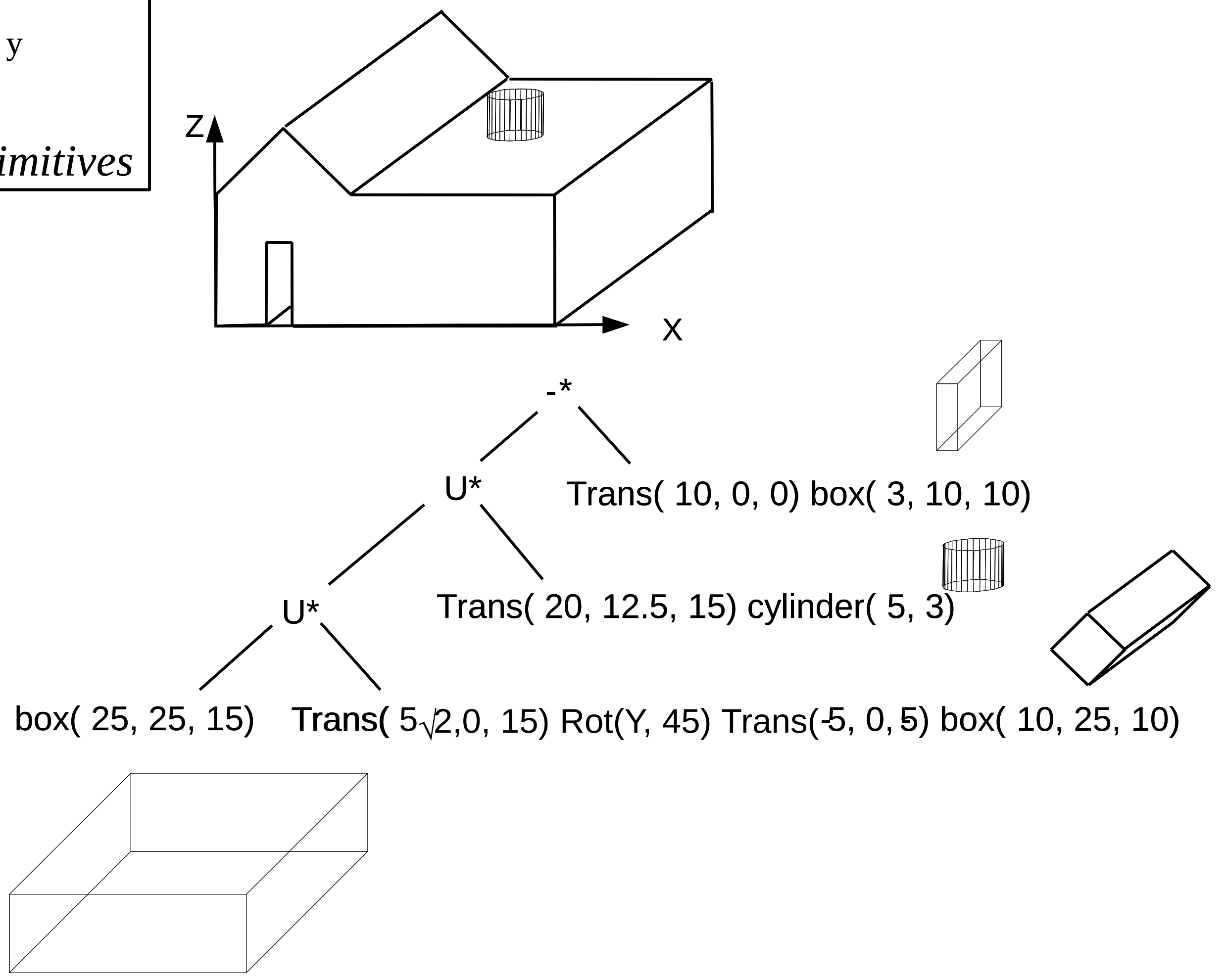
∩* (regular intersection)

CSG Tree:

Sequence of operators → design

Examples of CSG





Questions:

Can we use a different set of primitives ?

Is the CSG representation unique ?

[how to determine if two solids are identical ?]

Regularized operators

Is the set of 3D solids is closed with respect to (U, -, ∩)?

closure of a set S: *k*S

interior of a set S: *i*S

A U* B = *k i* (A U B)

A -* B = *k i* (A –B)

A ∩* B = *k i* (A ∩ B)

Why is closure over operations important?

uniform data structures

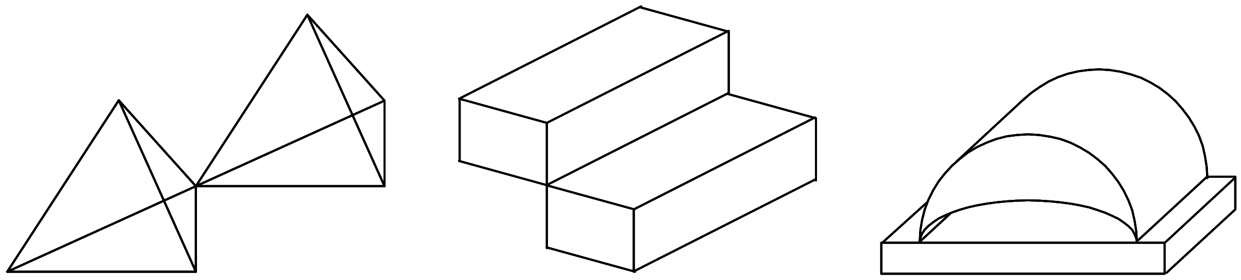
Regularized Euler Operators

Maintain solid as a *regular 2-Manifold*

2-Manifold regular solids

Open neighborhood of each point is similar to an open disc

Non 2-Manifold:



Problems with CSG

Non-Unique representation

Difficulty of performing analysis for some tasks

BREP (Boundary REPresentation)

What entities define the

Boundary of a solid ?

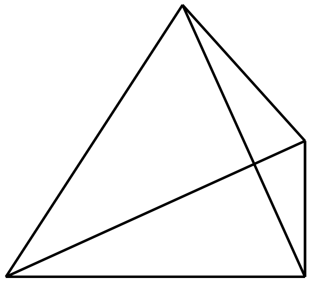
Boundary of surfaces?

Boundary of curves (edges) ?

Boundary of points ?

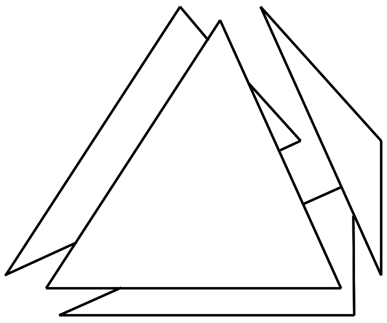
BREP

Boundary of a solid…



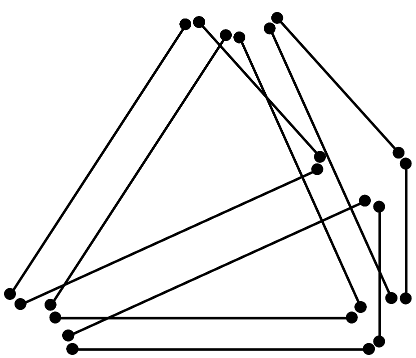
(a) Solid: bounded, connected subset of E^3

Boundary of surfaces…



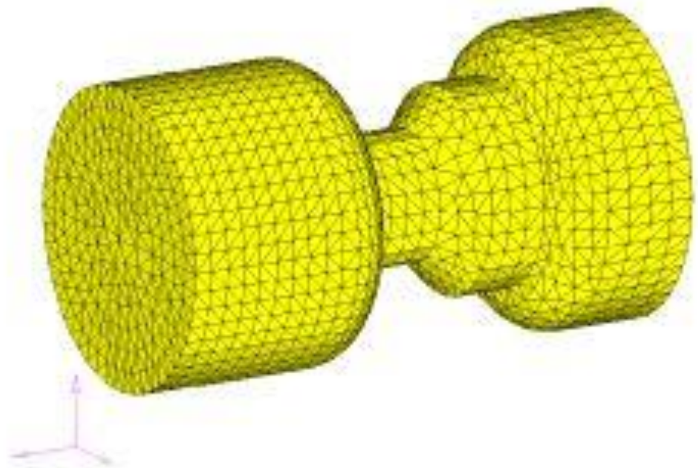
(b) Faces: boundary of solid
bounded, connected subsets of Surfaces

Boundary of curves (edges)…



(c) Edges: boundary of faces
bounded, connected subsets of curves

BREP: Polyhedral models



Using a Boundary Model

Compute Volume, Weight

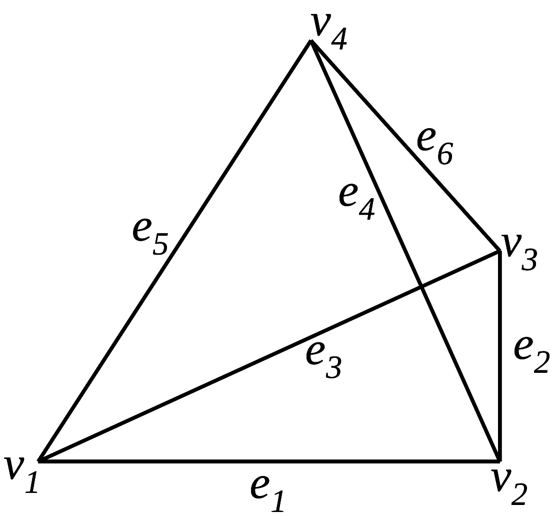
Compute Surface area

Point inside/outside solid

Intersection of two faces

…

An Edge-Based Model



Faces:

f_1	e_1 e_4 e_5
f_2	e_2 e_6 e_4
f_3	e_3 e_5 e_6
f_4	e_3 e_2 e_1

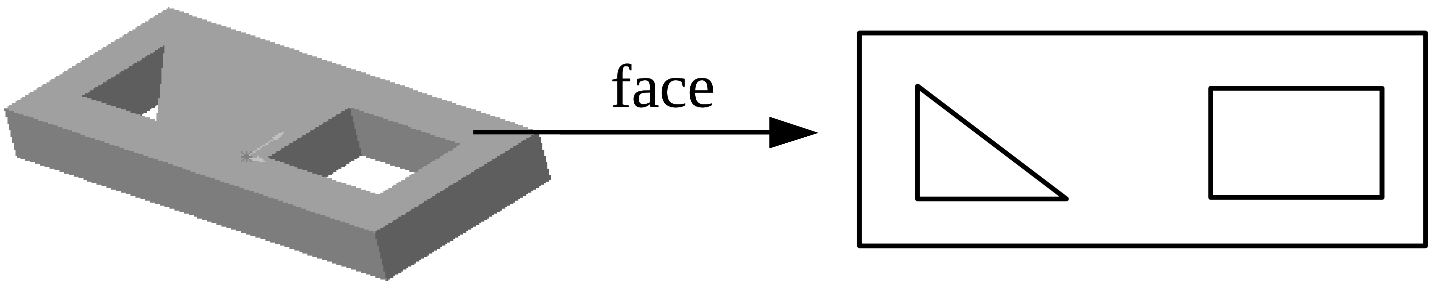
Edges:

e_1	v_1 v_2
e_2	v_2 v_3
e_3	v_3 v_1
e_4	v_2 v_4
e_5	v_1 v_4
e_6	v_3 v_4

Vertices:

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
v_6	x_6 y_6 z_6

Edge-Based Models: *inefficient algorithms*



Compute Surface Area:

1. Identify Loops
2. Compute area of each loop
3. Compute area of face

The Winged-Edge Data Structure

Efficient implementation of often-used algorithms

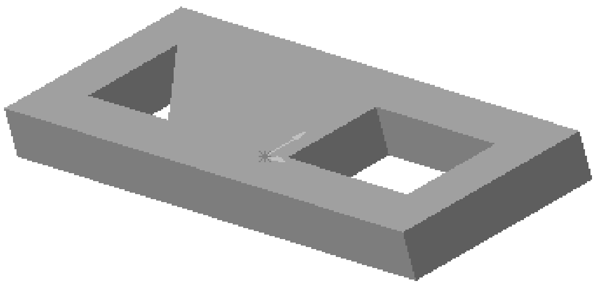
Area of Face

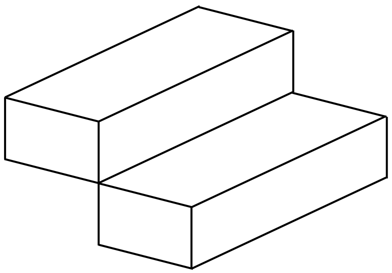
Hidden surface removal

Find neighbor-faces of a face

Observations

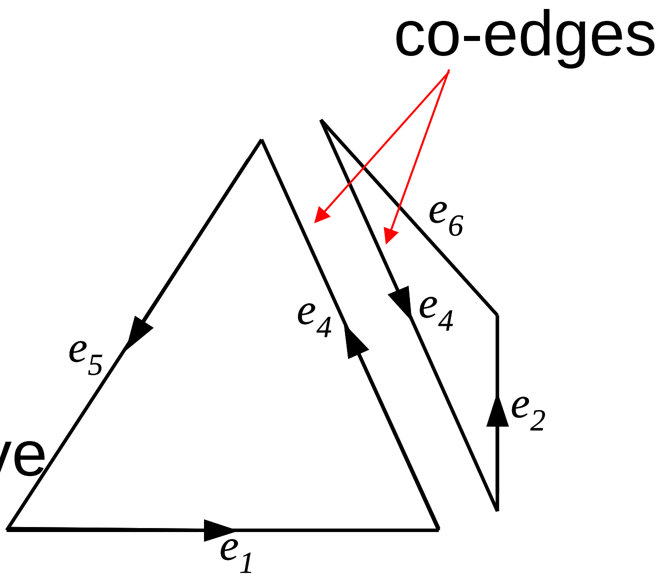
2-Manifold => Each edge is shared by exactly 2 faces



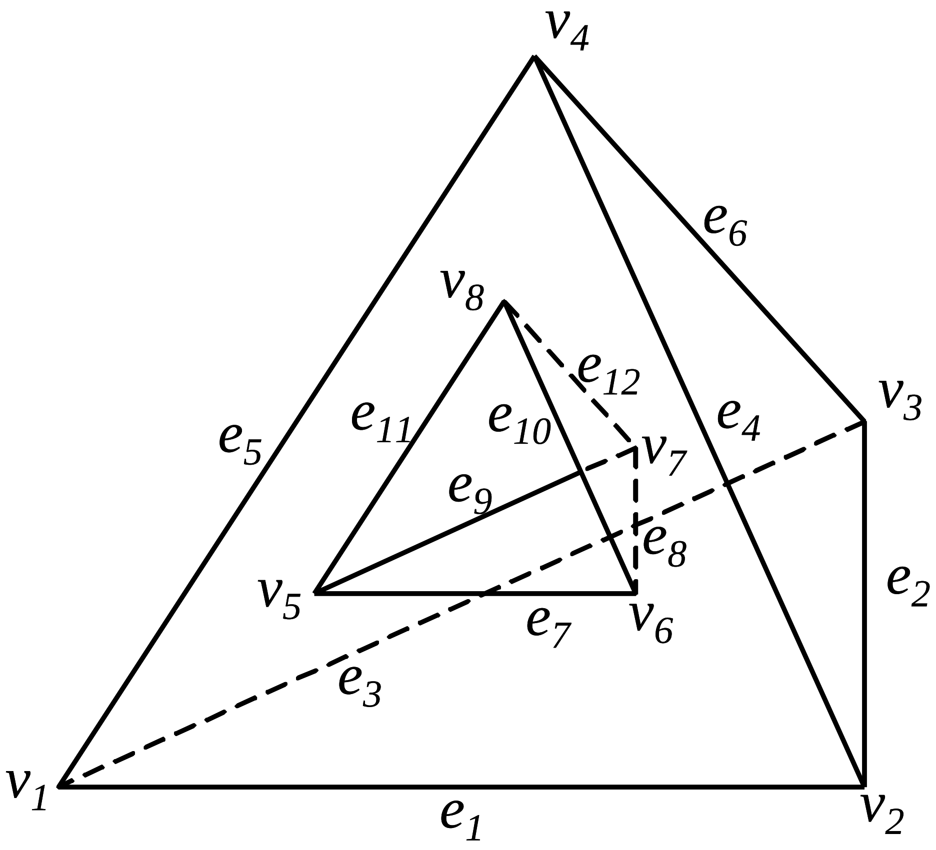


Face CCW convention =>

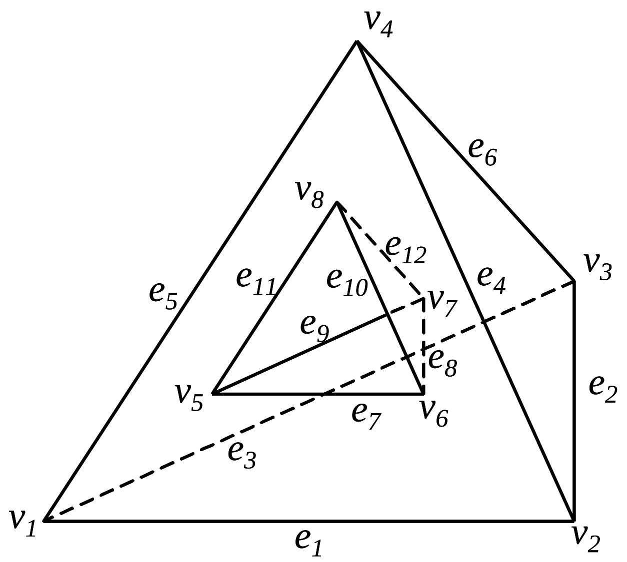
Each edge is once +ve, once -ve



BREP Example



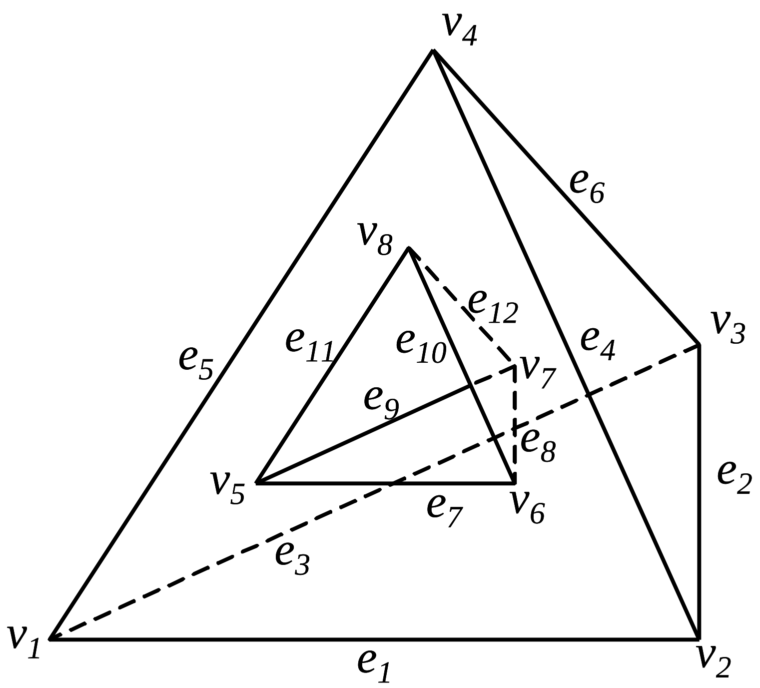
BREP Example



Vertices:

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
v_6	x_6	y_6	z_6
v_7	x_7	y_7	z_7
v_8	x_8	y_8	z_8

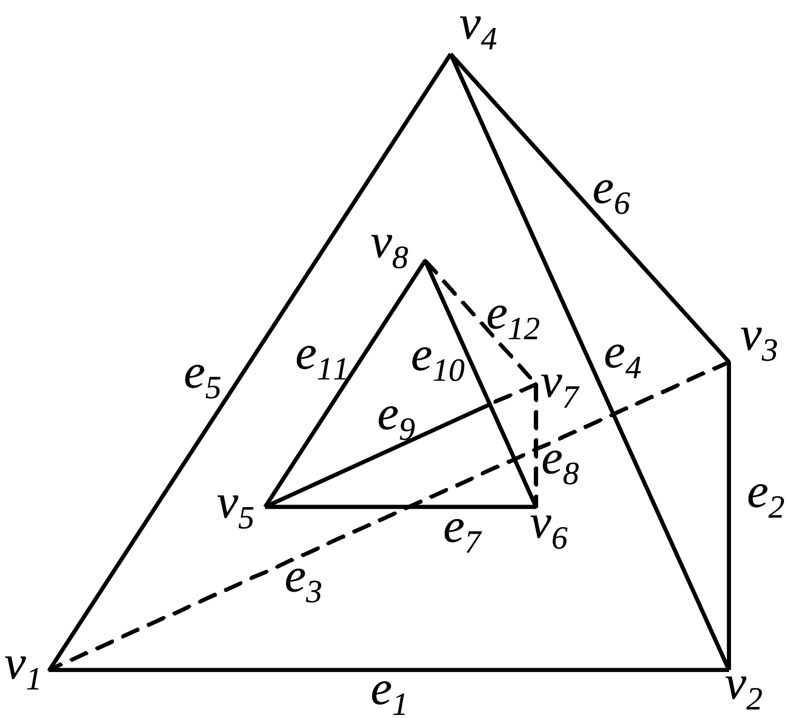
BREP Example..



Edges:

e_1	v_1 v_2
e_2	v_2 v_3
e_3	v_3 v_1
e_4	v_2 v_4
e_5	v_1 v_4
e_6	v_3 v_4
e_7	v_5 v_6
e_8	v_6 v_7
e_9	v_7 v_5
e_{10}	v_6 v_8
e_{11}	v_5 v_8
e_{12}	v_7 v_8

BREP Example…



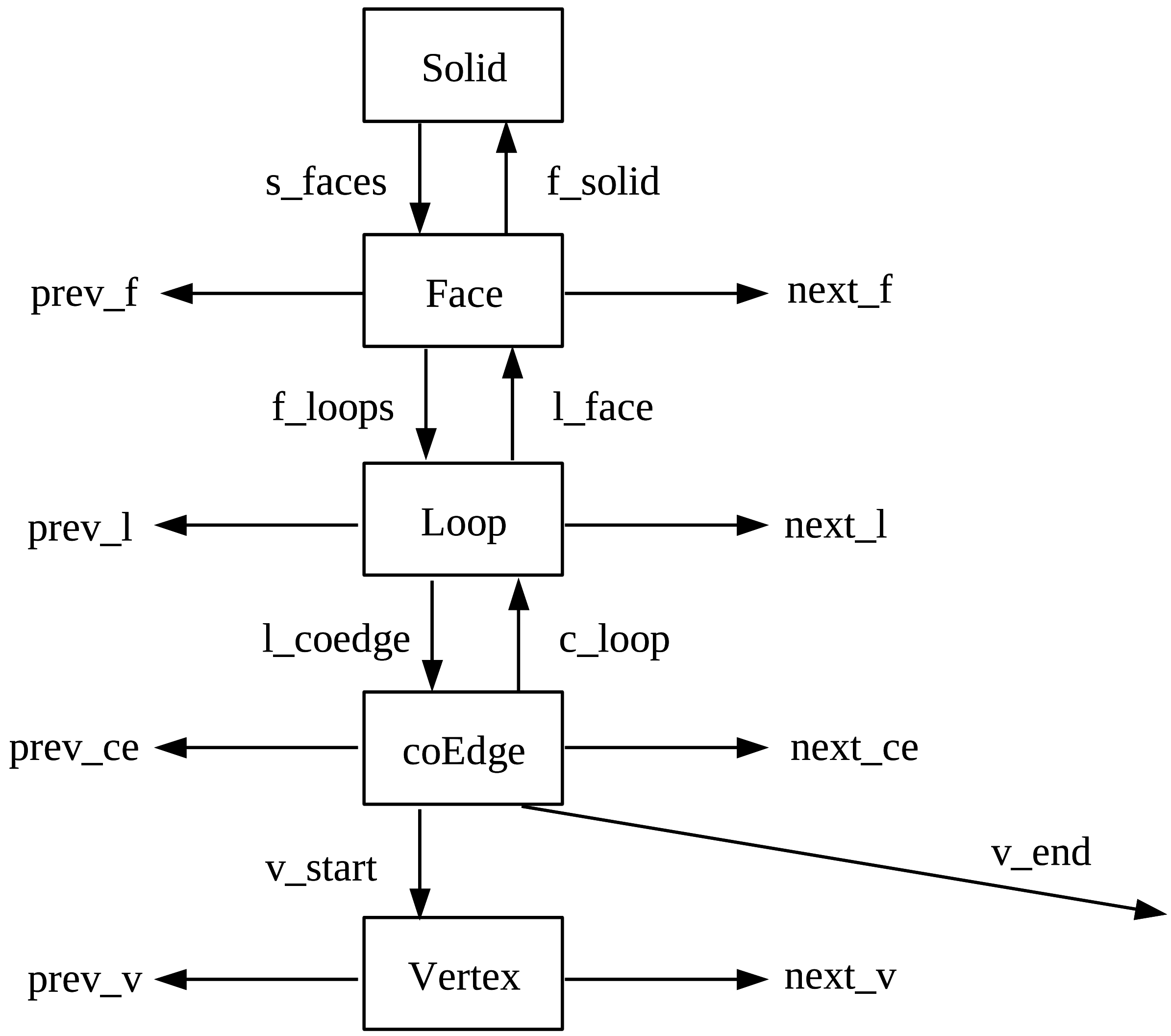
Faces:

f_1	l_1 l_2
f_2	l_3
f_3	l_4
f_4	l_5
f_5	l_6
f_6	l_7
f_7	l_8

Loops:

l_1	$+e_1$	$+e_4$	$-e_5$
l_2	$-e_7$	$+e_{11}$	$-e_{10}$
l_3	$+e_2$	$+e_6$	$-e_4$
l_4	$+e_5$	$-e_6$	$+e_3$
l_5	$-e_1$	$-e_3$	$-e_2$
l_6	$+e_7$	$+e_8$	$+e_9$
l_7	$+e_{10}$	$-e_{12}$	$-e_8$
l_8	$-e_{11}$	$-e_9$	$+e_{12}$

BREP: Winged edge data structure



BREP or CSG ?

Using: CSG is more intuitive

Computing: BREP is more convenient

Modern CAD Systems:

CSG for GUI (feature tree)

BREP for internal storage and API's

BREP: non-polyhedral models?

Same Data Structure, plus

For each edge, store equation

For each curved face, store equation

Why do we need to learn all this ?

(a) To anticipate when an operation will fail

(b) To allow us to write API's

CSG

- “InverseCSG: Automatic Conversion of 3D Models to CSG Trees”

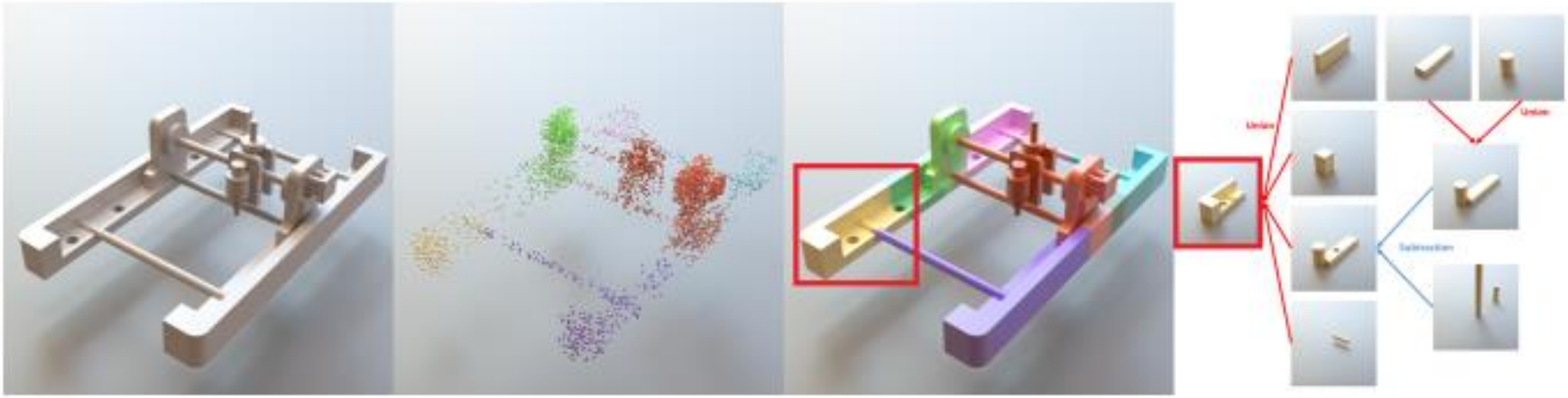


Fig. 1. We provide a system that takes as input a mesh file (left) and outputs its constructive solid geometry (CSG) representation. The three key ideas are a) use of carefully designed point samples to guide a purely discrete search problem (points, middle left), b) a divide-and-conquer algorithm to segment problems to ensure success (colors, middle left), and c) use of program synthesis techniques to solve the hard discrete search problem in each segment. The output CSG structure (middle right) correctly infers over 50 solid primitives and 18 boolean operators. A part of the solution (red box) is extracted for demonstration (right).