Computer-aided Design and Manufacturing

Adrian Bowyer

A.Bowyer@bath.ac.uk

Fab Academy, 30 September 2009
The Polya Urn – seemingly irrelevant...

1. Place one red and one white ball in the urn
2. Take a ball out at random
3. Replace it
4. Add another ball of the same colour
5. Go to 2...

When there are, say, 1000 balls in the urn, what is the expected ratio of reds to whites?
The Polya Urn

A Martingale stochastic process
The Polya Urn

The gradient is random
Computer Aided Design

As soon as computer graphics became possible (late 1960s; Evans & Sutherland), people started to write programs to do engineering drawing:
Computer Aided Design

Great! Let's get the program to do the perspective view automatically.

Oh dear! It's impossible...
Why is it impossible?

We need an *unambiguous* 3D representation.
Computer Aided Design

In fact, we need an unambiguous 3D representation for all automated downstream processes such as:

1. Volume
2. Mass
3. Surface area
4. Moments of inertia
5. Strength
6. Flexibility
7. Heat and fluid flow
8. Fields, currents and fluxes
9. Mechanical integrity and fit
10. Manufacture, and
11. Pictures
Computer Aided Design

Idea Number One – The Boundary Representation (Ian Braid, Bruce Baumgart et al.)

Extended Euler-Poincaré formula:

\[ \text{Faces} + \text{Vertices} - \text{Edges} - \text{Rings} = 2(\text{Shells} - \text{Holes}) \]
Computer Aided Design

Idea Number One – **The Boundary Representation** (Ian Braid, Bruce Baumgart *et al.*)

The extended Euler-Poincaré formula allows us to test the *topology* for solidity, but...

```
Faces = 3
Vertices = 0
Edges = 2
```

It can't do non-polyhedral shapes directly.
Computer Aided Design

Idea Number One – **The Boundary Representation** (Ian Braid, Bruce Baumgart *et al.*)

Curved surfaces: **Bi-parametric patches**

\[ P(s, t) = (x(s, t), y(s, t), z(s, t)) \]

The most common patch is the Non-uniform Rational B-spline: NURBS.
Computer Aided Design

Idea Number One – The Boundary Representation (Ian Braid, Bruce Baumgart et al.)

Curved surfaces: **Bi-parametric patches**

We can now stitch these together to make objects as a patchwork quilt.
Computer Aided Design

Idea Number One – **The Boundary Representation** (Ian Braid, Bruce Baumgart *et al.*)

But, if we change the geometry, the topology can become nonsense.
Computer Aided Design

Idea Number One – **The Boundary Representation** (Ian Braid, Bruce Baumgart *et al.*)

These problems have been solved (largely...).

_Some commercial B-Rep Geometric Modellers:_

**ACIS** (Spatial Corp.)

**Parasolid** (Siemens PLM Software)

**Open CASCADE** (Open CASCADE S.A.S.)
Computer Aided Design

Idea Number Two – The Constructive Solid Geometry Representation (Ari Requicha, John Woodwark et al.)
Idea Number Two – **The Constructive Solid Geometry Representation** (Ari Requicha, John Woodwark *et al.*)

The design is represented as an operator tree with geometric primitives at the leaves.
Idea Number Two – **The Constructive Solid Geometry Representation** (Ari Requicha, John Woodwark *et al.*)

J: $Ax + By + C \leq 0$ etc.

(Convention: negative is solid)

Rectangle: $J \wedge K \wedge L \wedge M$
Computer Aided Design

Idea Number Two – **The Constructive Solid Geometry Representation** (Ari Requicha, John Woodwark *et al.*)

Infinite cylinder, $I$: $x^2 + y^2 - r^2 \leq 0$

Infinite planar half-space, $P$: $Ax + By + Cz + D \leq 0$

Cylinder with ends: $I \cap P_1 \cap P_2$
Computer Aided Design

Idea Number Two – **The Constructive Solid Geometry Representation** (Ari Requicha, John Woodwark *et al.*)

And look! We can mix all the algebra and set-theory up in the tree.

Remembering the convention that negative is solid:

![Diagram of Constructive Solid Geometry Representation](image)

**CSG** \( \rightarrow \) **Functional Representation**
Computer Aided Design

Idea Number Two - **The Functional Representation**

We can put in any operations and functions that we like: \( \sin(...) \), \((...)^3\), \(\ln(...)\) and so on.

This gives us all the bendy surfaces that parametric patches gave B-Rep, and more.
Computer Aided Design

Idea Number Two - **The Functional Representation**

We will (almost) always have a valid unambiguous solid, but...

*We don't know where it is or what it is shaped like.*

We have to **evaluate** it.

Disc: \((x - a)^2 + (y - b)^2 - r^2 \leq 0\)

Point: \((x_1, y_1)\)

Evaluate: \((x_1 - a)^2 + (y_1 - b)^2 - r^2\)

→ - : inside
→ 0 : on surface (caution: f.p. arithmetic)
→ + : outside

A point membership test.
Computer Aided Design

Idea Number Two - **The Functional Representation**

Evaluating rectangles rather than points.

Discrete: $(x - a)^2 + (y - b)^2 - r^2 \leq 0$

Rectangle: $([x_{low}, x_{high}], [y_{low}, y_{high}]) = (X_i, Y_i)$

Evaluate: $(X_i - a)^2 + (Y_i - b)^2 - r^2$

→ [-, -]: definitely inside
→ [-, +]: *May* straddle surface
→ [+ ,+]: definitely outside

An interval membership test.

It is **conservative**.
Idea Number Two - The Functional Representation

Evaluating the entire shape using intervals.

Recursive spatial division (e.g. a quad tree, or – as here – a binary tree).

Rectangle A is entirely void (\(=\) null set).

Rectangle B is entirely solid (\(=\) universal set).

Rectangle C contains surface.
Idea Number Two - The Functional Representation

Pruning the tree.

The hatched area is $C \wedge J \wedge K \wedge L \wedge M$

In Q: $C$ gives $null$

$\rightarrow null \wedge J \wedge K \wedge L \wedge M$
$\rightarrow$ whole thing is $null$ in Q

In P: $K$ gives $null$

$\rightarrow$ whole thing is $null$ in P

In R: J, K, and L give $universal$

$\rightarrow C \wedge true \wedge M$
$\rightarrow$ we $MAY$ have surface in R
Computer Aided Design

Idea Number Two - The Functional Representation

Pruning the tree.

The Great Bath

Aquae Sulis

Roman Britain
### Boundary Representation vs. Functional Representation

<table>
<thead>
<tr>
<th>B-Rep</th>
<th>F-Rep</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bad:</strong></td>
<td><strong>Bad:</strong></td>
</tr>
<tr>
<td>Calculating volume (hence mass) is hard</td>
<td>Making pictures is hard</td>
</tr>
<tr>
<td>Numerical accuracy problems</td>
<td>Calculating surface area is hard</td>
</tr>
<tr>
<td>The data structures use lots of memory</td>
<td>F-rep primitives are not local</td>
</tr>
<tr>
<td>Input can be messy without</td>
<td>The model is unevaluated</td>
</tr>
<tr>
<td>lots of extra software</td>
<td>Hard to triangulate the surface</td>
</tr>
<tr>
<td>The model is evaluated</td>
<td><strong>Good:</strong></td>
</tr>
<tr>
<td><strong>Good:</strong></td>
<td><strong>Good:</strong></td>
</tr>
<tr>
<td>Easy to triangulate the surface</td>
<td>Calculating volume (hence mass) is easy</td>
</tr>
<tr>
<td>Making pictures is easy</td>
<td>Easy(ish) to make numerically robust</td>
</tr>
<tr>
<td>Calculating surface area is easy</td>
<td>Input is pretty straightforward</td>
</tr>
<tr>
<td>B-rep primitives are local</td>
<td>The model is unevaluated</td>
</tr>
<tr>
<td>The model is evaluated</td>
<td>The data structures are compact</td>
</tr>
</tbody>
</table>
Computer Aided Design

Boundary Representation vs. Functional Representation

Right! We can now take our preferred representation and go away and design Lamborghini, wave guides and art galleries!

Errr. No. We need a user-interface. And that typically takes much more code and many more person-hours than the geometric modeller.
Computer Aided Design

One aspect of the User Interface – Geometric Constraints
Computer Aided Design

One aspect of the User Interface – **Geometric Constraints**
Computer Aided Design

One aspect of the User Interface – Geometric Constraints

We end up with a large hierarchical non-linear geometric system to solve on-the-fly as the user is designing.
Computer Aided Design

Some commercial systems

AutoCAD (Autodesk Inc.)

SolidWorks (Dassault)

Pro-Engineer (Parametric Technology)
Computer Aided Design

All commercial systems are B-Rep.

They grew out of 2D drafting

B-Rep is quick to render on ancient computers

Neither of these matters today

But the Polya urn started out with more red balls than white...
Seeing what's there

There is one predominant technology – **the depth buffer**.

If everything had been F-Rep, not B-Rep, then the predominant technology would probably be ray-tracing.
Seeing what's there

**The depth buffer.**

Entirely implemented in hardware courtesy of:

1. Silicon Graphics and flight simulators, then
2. Computer games

For each pixel store both colour and **depth**

Send it **3D** coloured triangles in any order

Surfaces in front win
Computer Aided Manufacturing

Cutting away and building up
Offsetting straight lines and circular arcs is pretty straightforward.
Computer Aided Manufacturing

Offsetting flat F-Rep is very easy:

\[ J: Ax + By + C \leq 0 \quad \text{etc.} \]

\[ \rightarrow \]

\[ J': Ax + By + C - R \leq 0 \quad \text{etc.} \]

Offset rectangle: \( J' \land K' \land L' \land M' \)
Computer Aided Manufacturing

Offsetting parametric patches in B-Rep is surprisingly easy:

\[ P(s, t) = (x(s, t), y(s, t), z(s, t)) \]

\[ W(s, t) = \frac{\partial P}{\partial s}, \quad V(s, t) = \frac{\partial P}{\partial t} \]

\[ N(s, t) = W \times V \]

Tool centre is at \( P + rN \) (\( N \) normalised).
Offsetting pixel data is very easy:

Region growing

Set each white pixel within $r$ of a black one red...

Very memory hungry in 3D

→

Quad/Oct-tree + interval tricks

Now it's not so easy as it was...
Computer Aided Manufacturing

So. Offsetting is pretty much solved. What's the problem?

**Toolpath collisions**

Not too hard to detect.

But what does the computer do about it?

Using a smaller tool sometimes works, but is inefficient.

And what happens when the tool shaft hits the work?
Computer Aided Manufacturing

3D Toolpath collisions are not a problem at all for **additive manufacturing**.

This is the main reason that it is considered such a powerful technology.
Data exchange

How do we get data from CAD A to CAD B?

**ISO 10303 - STEP**

- A definition language – Express
- Implementation methods
- Conformance tests
- Etc.

The biggest standard within ISO...
Data exchange

How do we get data from CAD to a numerically-controlled machine?

G-Codes

; GCode generated by RepRap Java Host Software
; Created: 2009-09-13:11-29-21
G21 ;metric
G90 ;absolute positioning
T0; select new extruder
G28; go home
M104 S190.0 ;set temperature
;!LAYER: 1/48
M107 ;cooler off
G4 P20 ;delay
G1 Z0.0 F5.0 ;z move
G1 X1.7 Y2.2 F3000.0 ;horizontal move
Data exchange

How do we get data from CAD to additive-manufacturing machines?

STL files

solid
  facet normal n1 n2 n3
  outer loop
    vertex v11 v12 v13
    vertex v21 v22 v23
    vertex v31 v32 v33
  endloop
endfacet
facet normal 0.0 1.0 0.0
  outer loop
    Vertex 7.9 -2.63 12.2
    Vertex -18.4 -2.63 0.45
    Vertex 3.35 -2.63 5.5
  endloop
endfacet
endsolid
In Conclusion

Neither CAD nor CAM is like logic or mathematics, but they are like the engineering they serve:

They don't always give the answer you want (nor even the correct answer) even if you put correct data in at the start. But they work almost all the time.

The predominant technologies are not necessarily the best nor the most elegant. They got here today by Polya-urn style historical accident.
References from the **Dawn of Time**...

Type most of the terms in this lecture into Google Scholar and you'll find all the modern stuff. But where did it all start?


Bruce G. Baumgart, Winged edge polyhedron representation., Stanford University, Stanford, CA, 1972
