Talking about Shapes

Malcolm Sabin* CMath MIMA

This article is based on the IMA Summer Lecture I gave on 26 June 2013. It explores how people have communicated shape in the past and how they do it now. It closes by suggesting that new methods ought to be invented to take advantage of the very large memories now typically available. Having worked in the aircraft industry, my main focus is on smooth shapes, like aircraft, ships and cars.

1 Traditional shape languages

There is a purely analogue format, the *template* in which some physical object basically says: 'What you want is the same shape as me.' This would have been used in the building of medieval cathedrals. There is also a purely discrete format, which I will call the *recipe*, in which a list of instructions tells you what materials to start with, what tools to use and what to do with them.

The engineering drawing is a kind of hybrid, with both analogue (if lines look perpendicular they are intended to be perpendicular) and digital aspects. 'Do not scale' means that you do not measure off the drawing, but instead read the explicit dimensions.

2 Lofting

Until the mid-1930s, the aircraft industry used a technology which is essentially templating, but which supports the description of 3D shapes. This is based on descriptive geometry [1], invented (or at least formalised) by Monge in about 1800 to help Napoleon design fortifications.

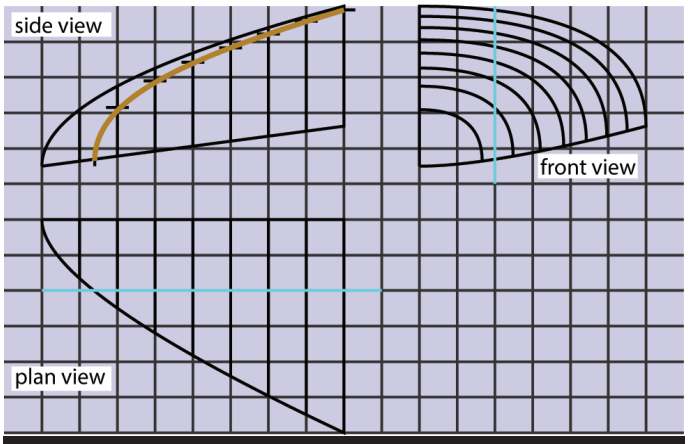

Figure 1: Traditional lofting

The structure of aircraft from the early 1920s was based on shipbuilding technology, in which the *skin* is wrapped around a set of *frames* which define the shape. These are structural members in parallel planes, and it is possible to draw their shapes in a single view. Copies of this single view, an end view in the case of the fuselage, can be cut to shape and used as traditional templates for the manufacture of the frames themselves. This process is called *lofting*, and the people doing the drawing *loftsmen*. The drawing was done at full scale even for aircraft over 100 feet long.

To get a smooth aerodynamic shape it is necessary not only for the frame shapes to be individually smooth, but for them to be in some sense consistent. This is achieved by a process of cross-fairing in which two other sets of parallel sections (vertical sections parallel to the plane of symmetry and horizontal sections) would be constructed by laying off distances from the datum planes to the various sections and plotting these in the new view, fairing them by using *splines* which were springy pieces of wood, very consistent along their length in cross section. Working to and fro between multiple views allowed the surface to be made very smooth (see Figure 1).

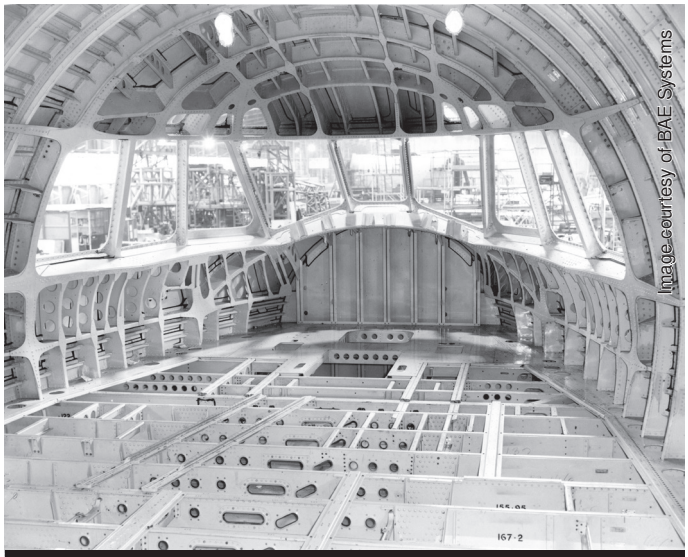

Figure 2: Aircraft frames enforce shape.

3 Conic lofting

In 1944 Roy Liming described the process of using *directrices* [2] which were the longitudinal curves defining the overall shape – the silhouettes in side and plan view – these being defined in terms of conic sections in those two views. The silhouette in plan view we now recognise as being a fairly complicated curve, a quartic algebraic curve, the intersection of two quadrics, which does not have a rational polynomial parametrisation, but all the calculations could be done by at worst solving a quadratic. Once the directrices were in place the *generators* could be defined as being conic sections passing through the directrices and with appropriate horizontal and vertical tangents at those points. Again all the calculations involved at worst square roots.

This was made possible by desk calculators becoming available which could do about one operation per second, and with enough registers to hold the coefficients of a given quadratic so that they did not need to be re-entered for every point computed. These were decimal registers, but a figure of about 100 bytes is a good upper estimate of the amount of memory required. Such machines became available in the mid-1930s, thus making *conic lofting* possible.

Conic lofting was adopted very quickly because the office space needed to loft an airliner full scale is much larger, and thus more expensive, than that needed for a small team of calculating machine operators. The arguments are well stated by Shelley [3].

* Numerical Geometry Limited

I was privileged to work (during the late 1960s) at what must have been the only aircraft company in the western world who still used manual lofting. Why had they not bothered? I conjecture that it was because they owned their own factory and the cost of the accommodation for lofting was not therefore as visible as if it had been rented.

4 Parametric surfaces

In the 1960s significantly more powerful computers were becoming available. A few hundred thousand operations per second and memories of a 100 kB could be bought.

There were also ideas in the academic community about a new sort of surface. In fact these ideas were not new, having been invented by Gauss to help him survey Germany in the early 1800s, but they were being formulated in a new way. Carl de Boor described a system using interpolating spline functions in 1963. Steve Coons described a system which defined a four-sided surface patch in terms of the four arbitrary parametric boundary curves in 1964 (published by MIT in 1966). Bézier and de Casteljau in Europe were also exploring the same ideas.

The idea was that you described a surface as a map from a piece of a plane into 3D. The x-, y- and z-coordinates of a surface point were all calculated from the u- and v-coordinates of the point in the *parameter plane*.

Both traditional lofting and conic lofting had regarded their output as being the *set of lines*. The new method thought of the surface as something continuous, so that the frames were only plane sections, which could be computed when you wanted them. In fact conic lofting surfaces could have been thought of as parametric ones, but nobody noticed that at the time.

My own contribution to this, apart from building a system [4] which was being used on real aeroplanes by 1970, was probably that I picked up (from a Ford Motor Company paper) the idea of using Newton's method to do the one difficult thing, which is to get from the coordinates of the point you were interested in to the values of u and v, and of applying it in four dimensions, not just two, to calculate intersections of two surfaces [6], a task which the conic lofting approach presented as being a difficult one. The question, of course, was that it was all very well using general parametric surfaces, but what should the actual equations be?

4.1 Solid body transforms

One of the intuitive properties of shape is that it does not alter when you turn an object around. Therefore much of a representation should remain invariant under such operations. Under conic lofting it does not.

The linear form

$$P(u,v) = \Sigma_i C_i f_i(u,v)$$

does have that property. If P is represented in a Cartesian coordinate system, then it can be expressed in another Cartesian system by

$$P' = R.P$$

which is a linear operator. P' is calculated from P which is itself calculated from u and v. Therefore P' is still a parametric surface.

Further, in the above equation, it is possible to apply the rotation operator to the coefficients C_i first, by taking the rotation operator inside the summation, and then use the rotated coefficients with no point-by-point rotation in the inner loops of the code. The actual representation of the rotated surface is of exactly the same kind as the original, but merely has rotated coefficients.

4.2 Tensor product basis functions

General functions of two variables are not particularly easy to choose. Nor are the coefficients particularly intuitive 'handles' for designing surfaces with.

If the bivariate basis functions are chosen to be the product of two univariate functions,

$$f_{ij}(u,v) = f_i(u)g_j(v)$$

the basis is the tensor product of the two univariate bases, which gives the coefficients some kind of rectangular structure.

The use of f and g for the functions in the two directions is just to stress that they do not have to be chosen from the same set of functions. In fact, of course, they often were.

4.3 Summation to unity

If each of the univariate basis function sets has the property that

$$\Sigma_i f_i(u) = 1$$

$$\Sigma_j g_j(v) = 1$$

then the bivariate function set also has the property

$$\Sigma_{ij} f_{ij}(u,v) = 1$$

and this leads to the property that the coefficients transform as points under a solid body transform. If they transform that way then they *are* points, and their effect becomes potentially intuitive. We start calling them *control points*.

4.4 Well-distributed maxima

If the maxima of the basis functions in any one set are well distributed in parameter values, then the effect of a particular control point being moved will be felt most strongly in the part of the parameter plane (and therefore the part of the surface) where the maximum of its basis function is, and therefore that part of the surface stays relatively close to its control point. If the basis functions are bell-shaped, then that localisation will be intuitively predictable.

My system used interpolating splines. In fact it was exactly Carl de Boor's formulation, though I did not find out about that until afterwards. No, it was not quite, because we also allowed control of the first derivatives at the control points. Bézier's work became known in the UK around 1970, and his functions also had the property of good distribution of the maxima.

4.5 Finite support

The last step in the evolution of good basis functions for surface design was what we thought then to be the generalisation of Bézier curves to B-splines. Bézier functions have finite support in the sense that you only evaluate them within a unit domain, but each of them influences the whole of that domain. The B-splines each influence only a part of that domain, and so it is possible to edit one end of the shape without the other end altering at all.

4.6 Non-uniformity

There are two final aspects in the development of what is now the de facto standard in engineering. Non-uniformity allows the knots, the places where finite support basis functions have their edges, to be unequally spaced in the parameter plane. This was extremely important in the interpolating spline context, where two closely spaced interpolated points with widely spaced parameter values could cause a loop in the interpolating curve. It is much less important in the B-spline context, except in the limiting case, where having two coincident knots can achieve a deliberate reduction in the degree of continuity, thus allowing creases to be modelled.

4.7 Rational basis functions

Parametric polynomial functions are not able to model pieces of circle, or any other conic except the parabola and straight line. The vendors of the CAD systems of the mid-1970s felt this to be a big problem, particularly when selling to companies still using conic lofting. Adding denominators in what we now know as NURBS allowed them to claim the ability to model conic sections. Yes they could, although NURBS cannot model conic lofting surfaces, because the directrices typically used are not rational parametric curves at all.

4.8 Offsets

Although there are good reasons for choosing these specific equation forms, the generality should not be forgotten when building software. In particular, we can calculate a point on an offset surface progressively. First a point on the base surface is determined from the chosen parameter values. Because the derivatives of the surface point with respect to the parametric variables are also determinable, the tangent plane and the local surface normal can be computed. A point on the offset surface is produced by moving up this surface normal the required amount. Because this complete process computes an offset surface point from the parametric coordinates, the offset surface is itself a parametric surface. It can be interrogated like any other, provided that the interrogation code does not make any assumptions to exploit the specific structures described above.

This may have been one of the reasons why this technology was quite quickly taken up. At the time (1970s) there was a lot of competition in the civil aircraft market, and an important aspect of competitiveness is the question of lead time. Being able to make wind tunnel models quickly is a good way of not losing time in bringing your product to market, and making wind tunnel models is certainly eased by being able to drive the centre of a numerically controlled machine cutter along the intersections of surfaces offset by the shape of the cutter.

This technology is still in use in aircraft design [5]. There are more modern methods (subdivision surfaces and triangulations) that have not been adopted by the aircraft industry, but are used in animation and manufacturing.

5 Subdivision surfaces

Subdivision curves were first described by Georges de Rham, in 1947. One of his colleagues had a student who did his work experience at a hammer factory and described the process of making hammer handles:

> You start with a rectangular section piece of wood, and then on each face mark out lines at 1/3 and 2/3 the way across it. Cut off the corners. Then repeat the process by marking on each of the eight faces lines 1/3 and 2/3 the way across it, and remove the corner again. Repeat this process using filing and sandpaper until the handle is smooth.

After some study de Rham determined that the curve did not have an equation, although the shape is well determined. At the midline of every face that is constructed there is a line on the limit surface, but the curvature there is infinite. The CAD community did not discover de Rham's work for several years.

George Chaikin attended a conference in 1974 at which most of the CAD academic community was present. He presented a really nice way of plotting curves (see Figure 3).

> You start with a polygon defined as a sequence of vertices. Then you compute a denser polygon by computing the points at 1/4 and 3/4 of the way along each edge. Repeat this process until it is smooth enough.

Figure 3: The Chaikin subdivision construction.

At that conference Rich Riesenfeld and Robin Forrest both recognised that the limit curve was a quadratic B-spline. They then went on within weeks to see that there were analogous constructions for the (equal-interval) B-splines of higher degrees as well. Jeff Lane and Rich Riesenfeld went on to show that you could do all B-splines by just taking averages. (Add two numbers and then shift right.)

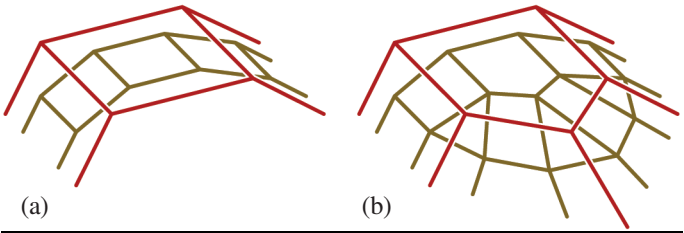

Figure 4: (a) Two-way subdivision (Chaikin). (b) Quadratic subdivision.

In 1979 two papers appeared in the journal *Computer-Aided Design (CAD)*, one by Ed Catmull and Jim Clark, the other by Daniel Doo and myself. Both pointed out that the surface analogues of subdivision curves could handle situations where the network of control points was not rigidly rectangular. A control point could be at the join of other than four facets; a facet could have other than four vertices (see Figure 4). The Catmull–Clark paper described both a biquadratic analogue and a bicubic one, by suggesting averaging rules to be applied at a non-4-valent vertex. At regular vertices the normal Lane–Riesenfeld B-spline rules would be applied. The Doo–Sabin paper suggested a way of understanding the behaviour of the limit surface in the neighbourhood of the limit point corresponding to such a point. Armed

with the ability to analyse, it also suggested a better set of rules for the biquadratic case.

These were a mathematical curiosity until 1997, when Tony deRose joined Pixar and Pixar used the Catmull–Clark surfaces for a short animated film called *Geri's Game*. They have used subdivision ever since, and this technology has become the effective standard in the computer animation industry [7].

The aircraft industry has not picked it up, presumably because aircraft surfaces do not need anything except regular grids of control points for either fuselages or wings. It is slightly more surprising that the car industry has not, because their shapes include more interesting features. However, this makes the point that even if a technology exists, it will not be taken up unless it provides sufficient economic advantage.

It was eventually taken up by the animation industry because for nice graphics it is really important to be able to run the edges of the control polyhedron along locally extruded directions in the target surface. If you run across them you get very visible effects called *lateral artefacts*, and the features on Pixar's characters were not limited to nice regular check patterns.

6 Triangulations

In the early to mid-1980s a new manufacturing technology began to be available. Layer manufacture worked, in the first instance, by using a laser to cause polymerisation in a thin layer of monomer. Once some pattern was set in this thin layer, more monomer was added and the laser polymerised a slightly different pattern in the next layer. By adding hundreds of layers, solid shapes could be built up from their cross-sections. By this time the computers available for controlling the machines had enough memory and processing power that they could determine the laser path in real time from a sufficiently simple surface representation. This avoided having to compute and then transmit hundreds of contours from a *part-programming* computer to the control computer.

This technology was called *rapid prototyping* even though the actual manufacture was a lot slower than machining, because it took the lead time of programming a tool-path out of the process of getting a prototype made from a model.

The representation from which the laser paths were computed was a triangulation, holding the coordinates of all the vertices of all of a set of triangles covering the surface. There is a more efficient representation which holds the vertex coordinates in one array and the subscripts of the vertices for a given triangle in another. For speedy navigation across the surface, indexes are also needed, to give access to the neighbouring triangles from any vertex and to neighbouring triangles from any triangle.

If we want an accuracy of one part in 10^6 (not unreasonable for engineering modelling even though the actual manufacture may be less accurate) the number of triangles for a smoothly surfaced object will be of the order of 10^6. This needs about 100 MB of memory to hold the data. Clearly really coarse models can be held in a lot less, while enormous models like the scanned version of Michelangelo's sculpture *David* can take up to 1000 GB.

Layer manufacture is maturing to the point where actual functional parts can be made from sintered metal rather than just stylists' prototypes from polymer. This representation has also become the one of choice for systems for part-programming numerically controlled machine tools.

7 What's next?

Looking over this historical view, we can see that the increasing memory size of computers has made different representations feasible every 25 to 30 years (see Figure 5). Some of these representations have made possible significant economic advantages and have been taken up quickly.

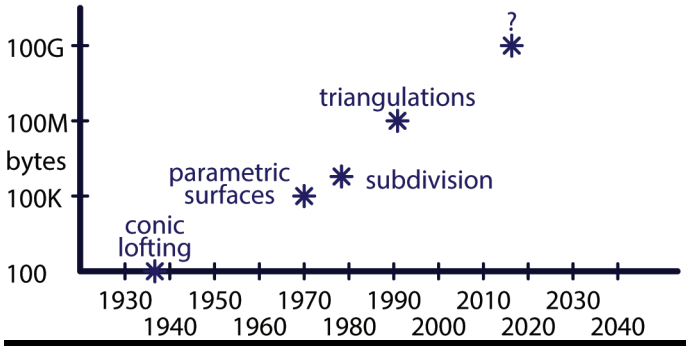

Figure 5: How we might use the computing power now available?

Workstations with 100 GB of memory will be commonplace on engineers' desks in just two or three years from now. The question we should be asking is whether we can invent a new representation which can exploit such memory sizes to good economic effect.

The 3D bitmap is probably two further generations away, because a grid of size 10^6 on each side requires about 2^{57} bytes, but oct-tree ideas give a compressed form of this. The interesting thought is that while parametric surfaces answer directly questions of the form, 'Where is this object?', and implicit surfaces those of the form, 'Is this object near here?', the bitmap approach (extended a bit) answers questions of the form, 'What is near here?' The complementarity of these questions means that we can expect to see multiple representations in use, with corresponding problems of maintaining consistency.

Acknowledgement: The author would like to thank Rebecca Waters for her help with this article.

Further reading

The references below are all reasonably accessible for the general mathematically literate reader. The journal *Computer Aided Geometirc Design (CAGD)*, published by Elsevier, is the main channel for new results in this area. At the forthcoming IMA conference on *Mathematics of Surfaces*, to be held on 13–15 September in Birmingham, one of the invited speakers is the world expert on using 19th century geometries to make it possible to build interestingly shaped buildings at reasonable cost.

References

1 Robertson, R.G. (1966) *Descriptive Geometry*, Pitman.
2 Liming, R. (1944) *Practical Analytic Geometry with Applications to Aircraft*, Macmillan.
3 Shelley, J.H. (1944) *The Development of Curved Surfaces for Aerodesign*, Gloster Aircraft Company.
4 Sabin, M.A. (1971) An existing system in the aircraft industry, *Proc. Roy. Soc. Lond. A*, vol. 321, pp. 197–206.
5 Farin, G. (2002) *Handbook of Computer Aided Geometric Design*, North Holland.
6 Patrikalakis, N.M. and Maekawa, T. (2002) *Shape Interrogation for Computer Aided Design and Manufacturing*, Springer.
7 Warren, J. and Weimer, H. (2002) *Subdivision Methods for Geometric Design*, Morgan Kaufmann.