Making What We Draw: Manufacturing with Ruled and Nurbs Curves

Kimo Griggs AIA; Kimo Griggs Architects, Somerville MA

REPORT

RESEARCH

Research engaged the following technologies:

Representative Digital Design Software:

Autocad (representing vector-based architectural design software)

- Rhino (representing Nurbs-based design development software)
- Solidworks (representing parametric design development software)
- FormZ (representing rendering software)

Representative Digital Manufacturing Technologies:

- Laser, and Water-Jet Cutters (representing plotter-type control mechanisms)
- Plasma Cutter (representing 2D Computer-Numeric-Control mechanisms)
- 3-axis CNC Router (representing 3D Computer-Numeric-Control mechanisms)

Research was conducted by designing, prototyping, and manufacturing actual building components or sculptural objects intended for use in a building environment, including examples of:

- 2.5 axis pattern cutting (cutting profiles in flat shapes)
- 2D laser cutting and bending (cutting flat shapes using HPGL, intended for further shaping)
- 2D plasma cutting and bending (cutting shapes using G-Code, and intended for further shaping)
- 2D plasma and water-jet cutting (cutting flat structural shapes not intended for further shaping)
- 2.5-axis plus 3-axis routing (cutting a flat shape after which a 3D shape was milled on one face)
- 3-axis routing on two opposite sides, with tool path variations (cutting a 3D shape using G-Code)

The design and manufacturing took place within a seamless digital environment, in which no paper documents were required to be produced during the production process.

The following pages will outline the findings and conclusions reached. The addendum includes a variety of images related to the individual projects, indicating the steps involved with producing each project where possible.

Computer Aided Design and Manufacturing (CAD/CAM) is a solid, workmanlike term that evokes a substantial presence and hard rules. Like many digital technologies, an inside view reveals a myriad of activities operating in an almost festive atmosphere of invention, creation, and daring bravura.

A substantial presence can indeed be found, represented by the large governmental agencies, large manufacturers, and well-established software giants that had a practical or financial interest in the development of CAD/CAM systems, but then there are the patternmakers, signmakers, machine-shop jobbers, woodworkers, jewelers, modelmakers, airplane builders, fabricators, sculptors, product designers…in short, an endless list of creative and productive parties who have been enlightened, emboldened, and made competitive through the use of some smaller piece of the immense digital design and manufacturing continuum. It's the Wild West out there, but it's connected upward, downward, and around the globe in astounding ways.

Continuity through a digital design and manufacturing process requires a few things: Digital Design Software that can produce Verifiable Entities that can be transferred to other digital environments; Manufacturing Software that can understand the designed entities such that it can define how a tool can work on them and then translate those definitions into a language a machine tool will understand; Control Software that can translate the machine tool language into precise electronic pulses to direct the machine tool in its work; and the Tool itself, usually controlled through the use of stepper or servo motors.

These general areas contain many specific solutions geared to specific families of shapes, materials, and manufacturing processes. For simplicity, continuity is often achieved by simply bundling software that works well together when a CAD/CAM system is specified, or a machine tool is purchased; the end result being that users in one area of design and manufacture my work very differently than users in another even though the components they use are recognizably the same. Manufacturers of sheet metal parts will work within environments unrecognizable to those in wire and tube-bending for example, who will be similarly removed from those working on a formula one racing car body.

VERIFIABLE ENTITIES

Within the world of Architectural Design, in which entities have always been defined through accurate, dimensioned scale drawings and the specification of material qualities, the need for verifiable entities is well understood. Entities can be defined in a number of ways using raster-based and vector-based programs, but the latter are vastly preferred, and most often used. Raster-based software defines images at a fixed scale by defining the color of each pixel within a grid of a defined size. Photoshop is a good example of such a program. Entities are not defined using equations, and have no control points or embedded editing capabilities, rendering them relatively useless for most manufacturing applications. I did not encounter a manufacturing application that used raster images directly aside from laser etching, although many applications appear to do so by invisibly translating a raster image into a vector image that can then be used to drive some digitally-enabled tools.

Digital Design Software is used to produce the vector-based drawings that form the basis most Computer-Aided-Manufacturing activities. Entities can be simply lines, arcs and points drawn in Autocad, NURBS curves - based shapes drawn in Rhino, or they can be fully-parametric 3D solids drawn in Solidworks or similar programs – the choice of what program to use is based more on design requirements than it is on down-stream manufacturing necessities. Vector-based programs allow geometric descriptions in the form of equations to be exported through common file formats such as dxf, iges, and step, among others. Entities defined by equations in one program can be read without error by another program, and can then be easily edited to add features, make them larger or smaller (patterns for metal casting are usually made larger that the original design, for example, to allow for the shrinkage associated with cooling metal). Common file formats continue to be developed, increasing the amount of information that can be reliably transferred from one program to another. The ability to transfer entities whose geometries are based on NURBS curves, for instance, allows virtually any shape to be translated without loss of accuracy – including very complex shapes that challenged traditional methods of manufacturing.

Ruled curves are particularly useful geometries where the final product will be composed of sheet materials, particularly when the geometries can be defined as being Developable - defined as being able to be flattened without distortion. Ruled curves are also useful when digitally-enabled tools are used to cut thicker stock using a straight tool, such as hot-wire foam cutter (for concrete form inserts or the basis for Exterior Insulation and Finish Systems (EIFS), or a multi-axis water-jet device, for example. Much of the work being produced by the Gehry office utilizes developable shapes, making it a relatively cost-effective approach to producing stunningly complex visual effects. Ruled curves can be produced in any 3D drawing package, but programs such as Rhino, Solidworks and their equivalents include specific commands and features that permit designs to be drawn using developable shapes, or that allow entities to be reshaped so that they can be reduced to developable shapes. The reduction of complex shapes to less complex developable shapes allows the widest range of numerically controlled tools to be utilized, such as plasma-cutters, laser cutters, water-jet cutters, cnc routers, edm wire cutters, hot-wire foam cutters, etc. The choice of specific tools will be based on size, shape, concerns for accuracy, speed and cost, and of course the material being utilized. While a great deal of editing can be done to the overall shape, typically very little intervention is allowed once the flat shapes have been defined, and most tools simply cut the shapes as closely as the technology allows.

NURBS Curves, or Non-Uniform Rational B Spline curves allow virtually any shape to be geometrically defined, and in such a way that accurate transfer between software environments is precise and complete. Shapes designed as solids or non-developable surfaces using programs such as Rhino and Solidworks are generally milled out of solid materials, or utilize molds that are milled using CAD/CAM software to produce G-Code that run downstream routing, milling and turning machine tools. They are also routinely used as the basis for rapid prototyping activities. These processes allow the surface appearance to be edited in a variety of ways without editing the original model by defining a tool path, which is then converted to the language required to run the machine tool.

Manufacturing Software allows a designed entity to be brought into a software environment familiar to a designer in appearance while actually applying different parameters. While the geometry can be seen, and often modified using graphic tools familiar to any draftsperson, the drawing tool set is usually limited, and some interesting things begin to be noticed as entities are worked on. An example: When a circle is drawn in a design environment such as Autocad or Rhino, the circle is simply a circle. When imported into a manufacturing program, the circle is a line that starts at 0 degrees, and goes in a counter-clockwise direction at a fixed distance from the origin, all the way to 360 degrees. This characteristic does not change the look and feel of the circle, but the added starting point, and the default position at 3 o:clock changes the nature of the circle. The program does this to allow down-stream manufacturing processes to take place. The purpose of the program is not to draw things, but to define what a tool needs to do in order to make the entity that has been brought into it. Tools need to go somewhere to do work, so the tool needs to define a point to start a circle, and a point to stop.

Manufacturing software provides many interesting opportunities, most of which remain to be explored by designers. Once an entity has been imported into a program such as BobCad or Mastercam, the purpose of the software is to produce a description of a tool path; a description of the path needed to move a tool in space and time to manufacture the desired entity. In essence, a geometric scaffold is built over and around the geometry imported from the design software environment, and a description is written that will cause a machine tool to recreate the scaffold. The original geometry is, essentially, rendered unnecessary at that point.

Creating a tool path within CAD/CAM software is where the rubber meets the road. Once a toolpath has been created, the movements of the actual machine tool are automatic, so it is in the definition of the tool path that some fun can be had. Most manufacturing tools operate within defined axes of movement – typically the X, Y and Z axes. Movement along those axes can be controlled by formulas, but for a variety of practical reasons most manufacturing software will divide curved shapes into small, discrete straight-line movements, or facets. Tools such as some lasers, knife cutters, plasma and abrasive jet cutters, for example, use Hewlett-Packard Graphics Language (HPGL) to control the movement of the tool – just like a plotter. The discrete motions are typically as small as 1,016 steps per inch, called plotter units (PLU), rendering them indistinguishable from continuous lines for all practical purposes. Other tools use languages such as G-Code, in which incremental motions along the different axes are simply described, along with the speed of the tool and other characteristics. The desire to break curves into discrete straight movements is sensible – files can be smaller and simpler, and no equations are required, thus reducing the need for fast processing times and allowing high manufacturing speeds. Someone has to decide just how to break up the curves, however, and what tools will be used to cut them. Designers generally, and unwittingly, hand this responsibility to manufacturers.

Tools that cut flat shapes are typically configured to follow shapes as closely as possible, and if they use HPGL there is little hope of engaging the manufacturing software to do much else. Tools that use G-Code, however, require manufacturing software be configured by choosing specific tool bits, and specifying values such as Tolerance, Stepover and Stepdown, and also the Toolpath Style.

Tolerance defines how closely a tool bit must follow a shape. A small tolerance, of perhaps .001 to .005 inches, will cause the tool to produce a very smooth and close rendition of the shape that was drawn, while a tolerance of .1 inch, for example, may produce a faceted appearance that is not easily predicted, although the effect is easily reviewed prior to manufacture. The facets are produced because a tolerance of .1 inch allows the tool to move as efficiently as it can within a range between 0.00 inches and 0.1 inches above, or outside the surface of the shape. The manufacturing software will choose the most efficient path – that being one that requires the fewest straight lines along a shape, and thus the fewest lines of descriptive text, and consequently the smallest file size (manufacturing software is frequently developed from early versions designed to use once-expensive memory as efficiently as possible). Fewer but longer straight lines form facets on the face of complex shapes. It is unusual to see faceted shapes being machined and used in the real world – I have not come across an example - but the potential for parametrically varying surface appearance seems rich.

Stepover and Stepdown define how close together individual passes of the cutting tool will occur with tools such as routers and milling machines as they slowly carve away a 3D shape. Like tolerance, this is seldom a concern when cutting flat shapes such as sheet metal or plywood, but it is a great concern for complex shapes in any material as it has a large effect on the visual appearance of the work. A small stepover will lead to a smoother overall appearance, while a larger stepover will generally lead to a rougher effect. Again, there is much room for experimentation here.

Toolpath Style defines the general movement pattern of a milling tool. Typical paths include: parallel cutting, spiral, project, radial, concentric, topographic, etc., and it is also possible to tell a tool to follow lines that are drawn on a surface, or to design your own toolpath. Pre-selected tool path definitions embody a good deal of someone else’s knowledge, so designing your own patterns can be perilous, but it is possible.

Modifications may be made to any or all of the configuration options without changing the definition of the shape that the toolpath is acting upon. In addition, different tool shapes may be chosen for efficiency or visual effect. A straight bit with a flat end will produce a different surface than a straight bit with a spherical end, for example, and a small bit size will lead to different visual effects than a large one, while also changing the speed at which work can be done by the machine tool. Experimenting with the different parameters can provide a wide range of appearances from a single design entity.

Control Software resides within a computer, and communicates with the controller box that lives between the computer and the actual machine tool. Text files, typically in the form of G-Code, are read by control software, which then commands the controller box to feed precisely controlled pulses to the motors that control the motion of the tool itself.

Computer-Numerical-Control is the system that actually controls the physical movements of the tool. Pulses generated by the controller are received by the stepper or servo motors to direct the tool to move to specific locations at specific speeds, to perform specific actions. The work is automatic during this phase, providing little opportunity for operator intervention.

CONCLUSION

The increase in designs that depend on complex digital geometries has caused some interesting relationships to develop between the design and construction sectors. Those relationships depend on the ability of designers to produce verifiable, and "watertight" digital models, and on the equal ability of vendors using digitally-enhanced machine tools to accurately render those designs in real materials. The goal of a completely seamless digital environment, from conception to completion remains a distant one in most cases, but a substantial amount of the routine work is now being accomplished via digital technologies, and there is every reason to think this growth will continue to grow in scope and capability.

The geometries that are used to design buildings can be chosen with preferred downstream manufacturing methods in mind. Flat shapes of almost any configuration can be easily manufactured using a wide variety of geometries and manufacturing tools, but the cost-effective production of complex shapes - and particularly complexly curved shapes – requires strategies that organize design objectives, material qualities, and manufacturing methods. These strategies are typically developed once the conceptual framework for a project has been established and general design intentions are known.

CASE STUDIES

Case Studies showing the bulk of the research are presented step-by-step in the following addendum. A variety of project types were reviewed or undertaken to test different combinations of design software, manufacturing software, and digital manufacturing tools.

2.5 axis pattern cutting – A Changing Pavilion for a Pool

This project for a demountable changing pavilion incorporated a flat pattern developed from the initials, H.H., found in a profusion of styles among the possessions within the main house. A particular pattern was chosen to be manipulated to form a flower shape, as well as buds and leaves that were distributed along the joint between vertical boards, which acted as a stem. A CNC router was used to cut the individual boards, which were then assembled manually to form the decorative surface.

Method:

Photograph original art (see Fig. 1 below)
Open Photo in Photoshop. Edit as desired.
Open Autocad. Insert photo as raster image. Scale to suit (it is not actually embedded, but linked)
In Autocad, Draw over photo to make new vector-based shapes
Modify vector-based shapes as desired, and apply to overall pavilion design
Identify individual boards to be cut on cnc router – still in Autocad
Save Autocad drawing containing individual boards to be cut using DXF format
Open DXF file in drawing window of BobCAD-CAM
Use drawing tools in BobCAD-CAM to draw the toolpath, offset ½ of the diameter of the router bit used to cut the part, which was ½"diameter for this work)
Open the Numeric Control (NC) window in BobCAD-CAM
Choose the offset lines for the tool path, indicating where the cutting action should start and stop, and where the tool will need to move up and down. The G-Code will be automatically written as commands are given (see Fig. 1 below).

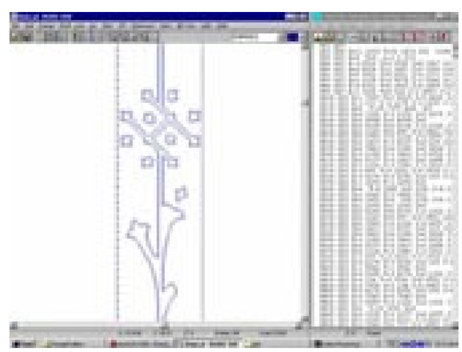

Figure 1. Original photo and CAD/CAM environment showing Drawing & NC window

When the tool path has been completely defined, save the G-Code file
Go to machine tool. Open G-Code file.
Place material on CNC Router. Begin program.
Repeat until all parts have been manufactured

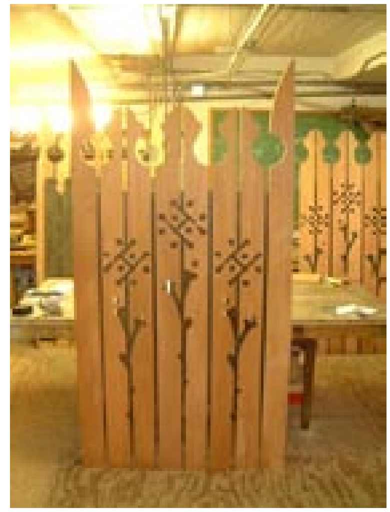

Figure 2. One side of the Pavilion, showing pattern

Case studies to come:

2D laser cutting and bending
2D plasma cutting and bending
2D plasma and water-jet cutting
2.5-axis plus 3-axis routing
3-axis routing on two opposite sides, with variations of the tool path
3D Printing (rapid prototyping) a wax model for ceramic-shell casting of a bronze part.