Using Poxels for Reproducing Traditional Pierced Byzantine Jewellery*

Vasiliki Stamati and Ioannis Fudos
Department of Computer Science
University of Ioannina
GR45110, Ioannina, Greece
{vicky,fudos}@cs.uoi.gr

Sofia Theodoridou, Christina Edipidi
and Dimitris Avramidis
GeoAnalysis S.A
Thessaloniki, Greece
{sofia,cedip,ramid}@geoanalysis.gr

Abstract

We present ByzantineCAD, a feature-based parametric CAD system for designing pierced Byzantine jewellery. ByzantineCAD is an automated system where jewellery is designed according to the values of parameters and constraints that are determined by the end-user. In this paper, we introduce an approach to reproducing traditional pierced Byzantine jewellery using pierced voxels ("poxels"). Poxels are voxels with specific attributes and properties that are appropriately combined to create large complex pierced designs. We also present a scaling algorithm for enlarging pierced designs without altering the size of the poxels used to construct them and an algorithm for engraving arbitrary images in pierced jewellery.

1. Introduction

Modern CAD systems are very efficient and useful in designing different kinds of jewellery [4]. However, there are types of jewellery that are not easily designed with general purpose CAD jewellery systems. Such a category is traditional pierced Byzantine jewellery, which was handmade jewellery created using piercing. Pierced jewellery was created from thin sheets of gold on which designs were engraved with a thin sharp tool. After the outlining of the designs, holes following their shape were created and decorated with triangular carvings, using an iron chisel (Fig. 1 and [5, 1]).

This paper describes the development of ByzantineCAD, a parametric feature-based system [2, 3] for designing and manufacturing traditional jewellery. In particular, it makes the following technical contributions:introduces a novel approach to designing and manufacturing complex jewellery of a particular craftsmanship, presents a method for reproducing complex pierced designs using non traditional voxels called "poxels" and presents algorithms for scaling design objects using fixed size poxels and for engraving arbitrary images in pierced jewellery.

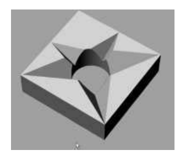

Figure 1. A pierced voxel

Many commercial 3D CAD systems have been developed for designing jewellery. However, none of these systems is appropriate for designing and creating jewellery of sophisticated craftsmanship, such as pierced jewellery. In the majority of these systems, designing is performed manually and usually the design steps cannot be programmed for automatic execution. Even in systems that step recording and parameterization is feasible, the tools provided are not adequate for constructing complex pierced objects. Furthermore, designing pierced jewellery using traditional CAD systems may lead to models with robustness problems, which are inappropriate for manufacturing. Also, editing parts of a pierced design in commercial CAD systems requires in depth knowledge of feature-based design and solid modeling techniques.

Our system is fully automated and easy to use even by the end users. In our system the parameter values are defined by the end user and the construction of the model is carried out by the system, based on the specified parameter values. The advantage of such a system is that the end user need not have designing skills or knowledge of using CAD systems.

[1] Work supported in part by a PENED Grant Nr 02ED329 and a PAVET Grant Nr 00BE3 by the General Secretariat of Research and Technology.We would like to thank Jack Ogden for his cooperation in designing the repertoire of pxels, so that it conforms with traditional jewellery. GeoAnalysis S.A. together with Polyline S.A. launched a company called ECHO which utilises among others the results of the byzantine project. ECHO creates byzantine jewellery collections which are based on the pierced technique. For more info see http://www.echo-jewellery.com.

2. ByzantineCAD: A parametric CAD system for the design of pierced Byzantine jewellery

2.1. Pierced voxels (Poxels)

By studying pierced Byzantine jewellery we discovered that the designs are made up of cylindrical holes with carvings around them. Each hole with the carvings around it can be considered a poxel (pierced voxel). Poxels can be used to create complex pierced designs.

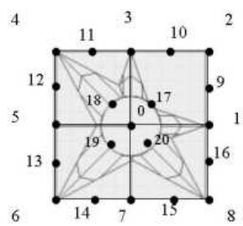

Figure 2. Valid points for carvings on a poxel

Poxels that are used to reconstruct pierced Byzantine jewellery have certain attributes. Specifically, a poxel is a solid made of a rectangular parallelepiped containing a cylindrical hole that is accompanied by at least one carving. The hole can be located at the center of the rectangular parallelepiped or in one of the 4 quadrants and a carving is directed from the cylindrical hole to a specific point. We assume the points depicted in Fig. 2. When the cylindrical hole is positioned at the center of the rectangular parallelepiped then the carvings can be directed to points 1-8. Otherwise, when the cylindrical hole is located in one of the 4 quadrants then the carvings can be directed at points 0-20, according to certain constraints which refer to the position of the hole in respect to valid carving directions. Additionally, there is only one poxel that is compact, with no holes or carvings.

Poxels are placed side-by-side, either on the top, bottom, right or left of each other, and unioned together. For two neighboring poxels to be unioned, the facet of the first must coincide exactly with the corresponding facet of the second. This way they form rows and columns of poxels. A poxel can be translated, rotated and bended.

Poxels, when combined appropriately, form different shapes and patterns. A continuous shape is formed by combining poxels whose carvings ends meet at a certain point.

We have developed a complete naming scheme for poxels previously described, in the sense that a unique name is provided for every acceptable configuration. The name assigned to each poxel describes the design created by the cylindrical hole and its carvings. The complete naming scheme is presented in [6].

2.2. Describing a pierced design with poxels

The information needed to construct a specific pierced design is recorded in a "layout description file" which is a text file that consists of the names of the poxels that make up the design in a row-wise format. Each pierced design can be thought of as a 2D matrix whose every entry corresponds to a poxel. The layout description file determines which poxel must be placed in each position of the matrix. Each time the name of a poxel is read from the description file, it is constructed, transformed (if necessary), translated to the proper location and unioned with the previous ones. Eventually a pierced plate representing the design is obtained.

A pierced design may be complex, meaning that it may contain a sequence of individual designs. For instance, the design may be a sequence of letters forming a word. In this case, the process of creating the plate representing the word is the same as for a single design. The layout description files of each individual design are read in parallel and the plate is created row-wise. The first line of the first letter is created, then the first line of the second letter is created and unioned with the first letter's first line and so on. From the pierced plates constructed, we can create different kinds of pierced jewellery, such as rings, bracelets and earrings.

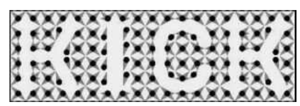

Figure 3. A pierced plate

3. A scaling algorithm for pierced designs

For designing jewellery depicting complex scenes it is important to have the capability of enlarging a pierced design without altering the size of the poxels used to construct it. For this reason we have developed a scaling method for enlarging pierced designs.

Since a pierced design can be considered a 2D matrix whose every entry contains a specific poxel, respectively, the scaled version of a design will be a larger 2D matrix containing more poxels. Scaling of a design is achieved by adding new rows or columns of poxels to the original design. This however may lead to problems with the symmetries of the design being scaled up. For instance, letter B is symmetric by a horizontal axis that goes through the middle of the design. If we add only one new row to letter B, it becomes asymmetric since the letter's shape is altered unintuitively. Therefore, the percentage of scaling that can be achieved is discrete, because of the need to preserve the symmetries that may exist in the original design. The restrictions for upwards scaling are: avoid adding only one row or column to the design and make sure the number of rows and columns is integer. As a consequence of the first restriction we choose to perform discrete scaling at a fixed factor. We choose a scale factor of 1.33 because it always results in adding two or more rows or columns. Therefore, from now on, we will refer to levels of scaling and not the scaling factor. Level 1 corresponds to scaling the design by a factor of 1.33.

We observe that as the design gets larger there is a need for thickening the engraved shape, so as to preserve its original form. We define as thickness factor T the ratio: $T = Rs/Ro$ where Rs is the number of rows in the scaled design and Ro is the number of rows in the original design. If $T > 1.5$, the thickness of the curves inside the design is not altered. If $1.5 < T \leq 2$ the thickness is increased by 50%, whereas if $T > 2$ the shape thickness is doubled (100% increase).

3.1. The scaling algorithm

Let us consider the Level 1 scaling of a letter of font size 6xc. When scaled up the design is transformed from a 6xc matrix to an 8xk matrix, where c and k are the number of columns of each matrix. The number of columns in the scaled design depends on the number of columns in the original design. For example, let us consider the scaling of letter O (Fig. 4).

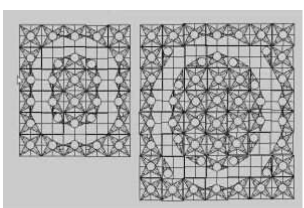

Figure 4. The letter O and its scaled version

The idea behind the scaling method is to gradually scan the initial design row by row using a sliding 2x2 window of poxels, scale individually the 2x2 windows of the design and then integrate smoothly the scaled overlapping parts to create the scaled version of the design. The combinations of poxels retrieved from the sliding window form different designs that can be categorized accordingly. An example is shown in Fig. 5. The categorization is based on the shape formed by the compact part of each poxel, meaning the part without a hole and carvings.

The original design is scanned using a 2x2 window that starts scanning the design row-wise from the upper left corner. The design is scanned from left to right, and from top to bottom. At each step the window is shifted to the right by one position, and when an entire row has been scanned, the window is initialized at the beginning of the next row. In general, the scaling method can be thought of as a transformation function whose input is a 2x2 scan window and

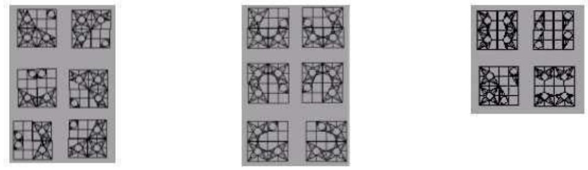

Figure 5. Different categories of shapes

output is a 2D matrix of various sizes, depending on the placement of the scan window in the scaled design. Below we present an outline of the algorithm. Steps 1 to 4 are explained in more detail later on.

```
for i=1 to n
 for j=1 to m
  step 1: Consider the 2x2 window of poxels:
          W[i,j]=[D[i,j],D[i+1,j],D[i,j+1],D[i+1,j+1]]
  step 2: Determine the new window Ws, that will be 2x2, 2x3, 3x2, or 3x3
          according to the category and position of the original window.
  step 3: Update the corresponding positions of Ds by placing the window
          Ws so as its upper left corner goes to [i,j]. If any such value
          conflicts with previous values of Ds then intergrate them so
          that the two overlapping windows join smoothly with each other.
 end for
end for
step 4: Search Ds for empty entries and fill them in with the neutral poxel.

where n is the number of rows,
      m is the number of columns,
      D is the matrix describing the original design,
      Ds is the matrix describing the scaled design,
      W is the sliding window, and
      Ws is the scaled sliding window
```

The time complexity of the algorithm is O(nm).

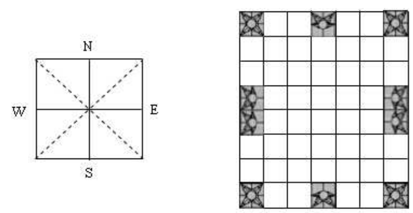

Figure 6. The datum points placed in the scaled design matrix

Before scanning and scaling, datum positions are marked in the scaled design matrix. We consider the poxels positioned at North, South, East, West, South-East, South-West, North-West and North-East as our datum points. These reference points are useful for making sure that we preserve the symmetries and that the various proportions of the shapes within the design are maintained. When the number of rows of the scaled design is even, reference points E and W are duplicated. Respectively, when the number of columns is even, points N and S are duplicated (Fig. 6).

Step1: A 2x2 window scan is performed that returns a combination of 4 poxels.

Step 2: The combination of poxels is scaled individually and then placed appropriately in the scaled design matrix. The scaling of the 2x2 block of poxels is such so that the relative position of the block in the original design is maintained in the scaled design and this is done in respect to the datum points. Also, if the block contains part of a curve of a shape the corresponding curve is scaled appropriately. Each combination is scaled in such a way so as to ensure that the initial shape form is preserved. The scaled form of a combination tries to approximate the original form as much as possible. For example, let's consider the combination in Fig. 7.

The 2x2 combination creates an approximately $\pi/3$ curve. The scaled version of this combination must also be a curve with the same angle. The size of the scaled combination is normally 3x3. However, the scaled window may be reduced to 3x2 or 2x3 (one column or one row truncated), or 2x2 (one row and one column truncated) if necessary. The way a 2x2 combination of poxels is scaled depends on the pattern created by them. By analyzing the original curve, we can see that it is created by poxels whose carvings create specific angles. If we consider that each poxel has a north, south, east and west orientation, then we can describe the angles through this orientation. The scaled version of the curve must have the same start and end orientation therefore an appropriate curve is drawn and then approximated with poxels, as best as possible.

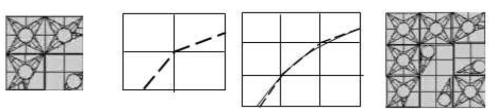

Figure 7. Scaling a 2x2 window

Step 3: The appropriate scaling for the specific combination determined in Step 2 is used to fill in the corresponding entries in the scaled matrix. This is placed in the scaled matrix so as to overlap previous scaled windows. Overlapping is used to ensure that the connection among neighbor cells is a valid one.

Step 4: The above steps are carried out row-wise until all of the design is scanned and scaled. If there are empty spaces in the scaled design matrix then these are filled with neutral poxels.

4. Engraving Images with Poxels

We have developed a technique for engraving images on pierced jewellery using poxels. The image is first converted to black and white and then we apply a principle similar to half-toning to render the black and white image by poxels. We use a 16x16 pixel sliding window which is mapped to a poxel. To encounter for aliasing effects the sliding window moves forward only by 8 pixels each time (row-wise and column-wise). The 16x16 black and white pixel area is partitioned in 4 quadrants and the number of black pixels in each quadrant is counted. According to this quantitative characterization of the pixel we choose a category of poxels to represent the pixel in the solid model. The exact poxel used is chosen to be compatible with the previously selected poxels (left and upper). The algorithm is presented below:

```
for (i=0; i<=n-8; i+=8) {
 for (j=0; j<=m-8; j+=8) {
  Consider the window consisting of the image pixels:
   W[k, l]= I[i+k, j+l], k, l= 0 to 15.
  Count the number of pixels in the four quadrants of the window:
   W[0..7, 0..7], W[0..3, 8..15], W[8..15, 0..7], W[8..15, 8..15]
  According to these counts select the category of poxels to use
   (center hole, or hole in one of the four quadrants)
  Select the specific poxel that is compatible with neighbor poxels
   that have been previously used (left and upper)
 }
}
```

5. Implementation and conclusions

ByzantineCAD was tested by creating STL models and sending them to a wax 3D printer. Wax models were manufactured and from these, metal prototypes were created (Fig. 8). We adopted a number of heuristics for enhancing the robustness of the models and for increasing the efficiency of the system. From the prototypes, certain issues regarding system parameters were re-evaluated and corrections were made so that we have a better final aesthetic result.

Pierced Byzantine jewellery is a unique kind of jewellery, mainly due to the special piercing technique through which it was created. This paper has introduced a novel approach to reproducing handmade objects of complex and sophisticated craftsmanship, by using non-traditional voxels(poxels). Poxels are appropriately combined to form complex pierced designs, thus reproducing a traditional technique that over time has faded away.

Figure 8. The metal prototype of a ring

References

[1] A. Geroulanou. *Diatrita*. Benaki Museum Publications, 1999.

[2] C. Hoffmann and R. Joan-Arinyo. On user-defined features. *Computer-Aided Design*, 5(30):321–332, 1998.

[3] G. Jense. Voxel-based methods for cad. *Computer-Aided Design*, 8(21):528–533, October 1989.

[4] L. Molinari and M. Megazzini. The role of cad/cam in the modern jewellery business. *Gold Technology*, (23):3–7, Spring 1998.

[5] J. Ogden and S. Schmidt. Late antique jewellery:pierced work and hollow beaded wire. *Jewellery Studies*, (4):5–12, 1990.

[6] V. Stamati and I. Fudos. A parametric feature-based cad system for reproducing traditional pierced jewellery. Technical Report 3-2004, Department of Computer Science, University of Ioannina, 2004.

Proceedings of the Computer Graphics International (CGI'04)
1530-1052/04 $20.00 © 2004 **IEEE**