Computer-Aided Design 37 (2005) 431–449

COMPUTER-AIDED DESIGN

www.elsevier.com/locate/cad

A parametric feature-based CAD system for reproducing traditional pierced jewellery☆

Vasiliki Stamati, Ioannis Fudos*

Department of Computer Science, University of Ioannina, P.O. Box 1186, GR 45110 Ioannina, Greece

Received 22 November 2003; received in revised form 23 July 2004; accepted 29 July 2004

Abstract

In this paper, we introduce ByzantineCAD, a parametric CAD system for the design of pierced medieval jewellery, which is jewellery created by piercing, a traditional Byzantine technique. ByzantineCAD is an automated parametric system where the design of a piece of jewellery is expressed by a collection of parameters and constraints and the user's participation in the design process is through the definition of the parameter values. We present an approach to designing traditional pierced jewellery using a voxel-oriented feature-based Computer Aided Design paradigm: a large complex pierced design is created by appropriately placing elementary structural elements. We also present a scaling algorithm for enlarging pierced designs without altering the size of the elementary structural elements used to construct them.

Keywords: Feature-based design; Parametric design; Traditional jewellery; Voxel based design; Font scaling; CAD

1. Introduction

Computer Aided Design (CAD) systems are widely used in most industries and are increasingly used in jewellery manufacturing [1,2]. While manual design of jewellery is still in wide use, this approach is both cumbersome and time consuming when compared to designing using 3D CAD systems. Furthermore with 3D CAD systems, the designed model is realistically rendered from various viewpoints, giving the designer a clear optical understanding of the final result.

Editing and redesigning are feasible in a 3D CAD environment, as opposed to the difficult task of fixing hand-drafted 2D design sketches. 3D rendering helps the artist correct or redesign parts of the model that are unsatisfactory [3]. In addition, it is possible to redesign based on the customer feedback. Besides, in a 3D CAD system the designers have various tools to assist them in designing a piece of jewellery, including transformations, primitive solids, embedded libraries of jewellery parts such as cut stones and gems and ring settings. Finally, an important advantage of CAD systems is that models can be passed on directly to rapid prototyping machines for the manufacturing of jewellery using layering techniques.

Despite the effectiveness of current 3D CAD systems for jewellery, there are categories of jewellery that are not easily designed even with modern CAD systems. An example of such a type is traditional Byzantine jewellery. Byzantine jewellery were designed and created by hand using piercing, a traditional Byzantine technique, and the designs illustrated there were very detailed and complex. In this paper, we present ByzantineCAD, a parametric CAD system suitable for the design of pierced Byzantine jewellery. The system is automated and parametric [4,5], meaning that the user-designer sets some parameter values and ByzantineCAD creates the jewellery model that corresponds to the specified values. This provides the designer with the ability to rapidly create custom-designed jewellery, based on the preferences of the customers such as including their initials on a ring. ByzantineCAD introduces a feature-based [6] and voxel-based [7,8] approach to designing jewellery, through the definition of elementary

☆ Work supported by a GSRT grant for Industrial Research and Development (Grant No. 1273). This work was carried out in cooperation with GeoAnalysis SA.

* Corresponding author.

E-mail addresses: vicky@cs.uoi.gr (V. Stamati), fudos@cs.uoi.gr (I. Fudos).

0010-4485//$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2004.07.004

structural elements with specific attributes and properties that are used as building blocks to construct complex pierced designs.

In particular, the paper makes the following technical contributions:

- introduces a novel feature-based approach to designing and manufacturing complex jewellery of a particular craftsmanship,
- presents a method for illustrating complex designs using non traditional voxels (such as elementary pierced solids),
- presents an algorithm for scaling design objects using fixed size pierced designs,
- describes our experience in the development of this software and in particular provides the reader with the heuristics adopted on how to optimize the time consuming solid manipulation operations and on how to produce robust solid models ready for 3D printing.

The remainder of this paper is structured as follows. Section 2 describes pierced Byzantine jewellery and the particular way in which they were created. In Section 3, we refer to the reasons for which current commercial CAD systems are not suitable for the design of this particular family of jewellery. Section 4 describes the approach used by ByzantineCAD to approximate a pierced design. Pierced designs are approached using a feature-based method, in which appropriate structural elements are defined and combined in ‘layout description files’. Section 5 presents a method for scaling pierced designs, while maintaining a fixed size for the basic components used to construct them. Section 6 describes an extension of ByzantineCAD, to include custom solid designs in a pierced surrounding. Section 7 presents our experience in developing this software. Section 8 provides conclusions.

2. Pierced Byzantine jewellery

Pierced Byzantine jewellery are gold jewels with pierced designs that were made along the coastlines of the eastern Mediterranean Sea during the period 3rd–7th century A.D. Their originality is due to the particular processing technique that is used for their creation resulting in a special esthetic effect [9]. Pierced jewellery was created from thin sheets of gold. The designs were engraved on these sheets of gold with a thin sharp tool. After the outlining of the designs, holes following their shape were created and these were decorated with triangular carvings, using an iron chisel (Fig. 1) [10,11]. By studying pierced Byzantine jewellery, one can distinguish certain types, shapes and characteristics.

The piercing technique was applied to various types of jewellery, such as necklaces, earrings, rings and bracelets (Fig. 2) [9]. This technique was also used to create pierced

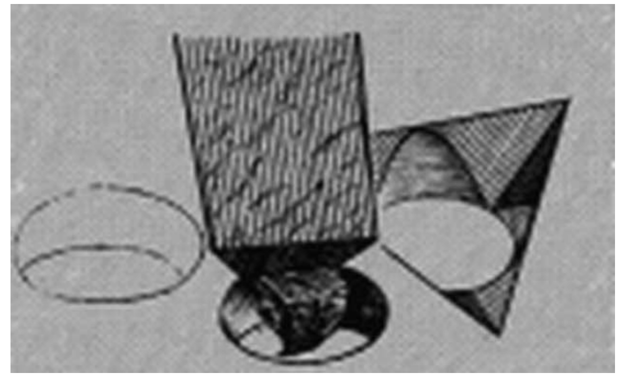


Fig. 1. Using a chisel to create carvings around a hole.

crowns, pins used for holding tunics, belts and brooches. Jewellery such as earrings and pendants on the necklaces were usually shaped as rectangles, circles, hexagons, half moons (lunes) and teardrops. Many pieces of jewellery were decorated with sparsely placed solid beads, or with a beadlike sequence wrapped around the piece of jewellery.

The designs used in Byzantine jewellery were specific. Usually, they were representations of nature or of human life. Frequently used nature inspired themes were different forms of birds, peacocks in particular, animals, especially dolphins, and plants, such as clovers and vine leaves. Many pieces of jewellery represented scenes combining nature and human beings (Fig. 3) [9]. Byzantine jewellery had often representations of letters or religious symbols such as the cross. The letters used belonged either to the Greek or Latin alphabet and gave a personal touch to the jewellery.

Pierced Byzantine jewellery is interesting for many reasons. There is a limited number of such jewellery preserved until today. The technique used to create them is a very sophisticated form of craftsmanship, as are the designs featured on them, and this technique has faded over the years. Furthermore, their esthetic effect is unique. There are other techniques that result in jewellery with engraved designs, but they differ in the way that they are processed and the final esthetic result. An example of such a technique is braiding plain or decorative wire to create the lace-like effect of pierced jewellery. The difference of the processing technique can be observed in the details of the designs and the back side of the jewellery. When the jewellery is

Fig. 2. A pierced Byzantine bracelet.

Fig. 3. A pierced bracelet featuring people and birds.

pierced, the traces of the tool used for the piercing are apparent, and the back side of the jewellery is an ensemble of holes in a solid surrounding.

Our goal is to deploy CAD techniques to reproduce this type of jewellery.

3. CAD systems for jewellery design

Many commercial CAD systems have been developed for the purpose of designing jewellery. Table 1 provides an overview of some of the most popular jewellery design systems. A large number of these systems are parametric and feature-based [12]. They provide graphical interfaces with excellent rendering capabilities. The majority of these systems provide built-in libraries of settings and cut gems and stones and advanced feature-based design tools [13]. Some systems provide advanced functionality such as Matrix [14] that provides the use of builders for recording design steps and for defining parameter values for parts to be used in the process. Also, the majority of these systems have the capability of exporting models to rapid prototyping machines [15]. All of these systems provide various tools for making jewellery design a simpler and less time-consuming process.

However, none of these systems is appropriate for designing and creating pierced Byzantine jewellery or other jewellery of this kind of detailed engraving. For crafting such jewellery we need to create a variety of very small structural elements and use them as building blocks to construct designs in a way similar to the use of voxels for representing solids.

In particular, for constructing such jewellery, where the fine aesthetic result is achieved through the detailed engraving, we need systems with the following capabilities:

(1) create very accurate and robust solid models of small voxel-sized pierced structural elements. This entails the direct invocation of solid modeling libraries or the use of geometric constraints.
(2) perform automatic placement of the large number of structural elements to create more complicated designs.
(3) incorporate this design to a solid model corresponding to a ring, bracelet, necklace, etc.

In most commercial CAD systems for jewellery, designing is performed manually using various tools and usually the design steps cannot be programmed to be executed automatically and accurately. This means that each different piece of pierced jewellery would have to be created basically from the beginning by hand which does not conform to requirement (1) above. Even in systems that step recording and parameterization is feasible, the tools provided are not adequate to provide for a voxel-based construction of objects. To satisfy this requirement we could use other generic CAD tools for this purpose. For example we could define a structural element with appropriate accuracy if we use a geometric constraint modeler (e.g. ProEngineer). Defining all the repertoire of structural

Table 1
Popular jewellery CAD systems and their characteristics

CAD system name	Characteristics of the system
JewelCAD [16]	Very good rendering techniques, capability of modifying and repairing STL files. The user can define her/his own libraries, calculate the weight and volume of a piece of jewellery, export models to rapid prototyping machines
JewelSpace [17]	A feature-based system with: realistic rendering, built-in libraries of gems and settings, support of many file formats. It provides a variety of tools for designing jewellery, and the capability of exporting models to rapid prototyping machines
Rhinoceros [18]	A general purpose system for which plug ins have been developed for designing jewellery. It is compatible with other jewellery design systems. It provides NURBS technology, repair tools and export operations directly to rapid prototyping systems
Matrix [14]	A parametric feature-based jewellery design system based on Rhinoceros. It provides specialized libraries with gems, stones and jewellery settings, realistic rendering techniques, and builders that record and carry out a sequence of design steps. It can export models for rapid prototyping
ArtCAM Jewelsmith [19]	ArtCAM provides powerful modeling tools, stl export capabilities for rapid prototyping, advanced ring creation tools (built-in settings, easy ring size adaption), precision manufacturing capabilities
TechJems 3.0 [20]	A plug-in for Rhinoceros with jewellery designing tools such as: built-in libraries of gems, cuttings, materials and settings, advanced tools such as a chain and ring creator, dummy tool (e.g. a dummy bust for necklace designing), pave tools (for placing objects perpendicular to a surface), weight calculation tools, and STL export capabilities
Jcad3/Takumi Pro [21]	Built-in libraries of cut gems, realistic rendering techniques and a variety of designing tools
CADJewel 3D [22]	It is a system based on Rhinoceros and therefore has the same characteristics as Rhinoceros

elements in such a system is a cumbersome but feasible procedure. Furthermore, placing all the structural elements side by side to create a plate which will then be transformed into a free form shape (requirements (2) and (3) above) is a procedure that requires the use of customized macros that invoke low level solid modeling operations.

In general, editing parts of a pierced design in commercial jewellery design systems requires in depth knowledge of feature-based design and solid modeling techniques. In contrast, our system offers an easy procedure for creating and using these small sized structural elements. ByzantineCAD is fully automated and easy to use even by the end users. In our system one defines the basic parameters that refer mostly to the appearance, size and content of the final product and then the construction of the specified model is carried out by the system. The advantage of such a system is that the end user need not have designing skills or knowledge of using CAD systems.

Also, by parameterizing the process of creating pierced jewellery, it is very easy to modify characteristics of the jewellery such as the size and the designs represented. Furthermore, designing a pierced jewel using a traditional CAD system may lead to models with robustness problems, which are inappropriate for manufacturing, creating therefore the need for repairing tools and techniques. The technique used in ByzantineCAD leads to robust models that can be directly sent to rapid prototyping machines for manufacturing without any further intervention or repair.

In a nutshell, our system combines the usual parametric and feature-based CAD paradigms using a voxel-like representation for designing pierced jewellery.

4. ByzantineCAD: a parametric CAD system for the design of Pierced Byzantine jewellery

By studying the craftsmanship of pierced Byzantine jewellery we have concluded that a pierced design can be represented by defining and combining appropriate structural elements. These structural elements are placed side by side, either on top, bottom, right or left of each other, and unioned into a new object. For two neighboring elements to be unioned, the facet of the first element must coincide exactly with the corresponding facet of the second element. A pierced design can be defined if we know which structural elements are required and in what order they have to be placed. The corresponding information for each pierced design is stored in a file called the 'layout description file'.

4.1. An approach to designing pierced jewellery using structural elements

4.1.1. Description of a structural element

The conformity of the structural elements with traditional jewellery was carefully examined in cooperation with Dr Jack Ogden, a specialist in the materials and techniques of ancient and historic precious metals and jewellery [10,11]. The design of pierced Byzantine jewellery is not just made up of cylindrical holes, but the holes have carvings around them. Each hole with the corresponding carvings around it is considered for the purposes of reconstruction as a structural element. Each structural element is a solid made of a rectangular parallelepiped with a cylindrical hole and the corresponding carvings around the hole (Fig. 4). According to the aesthetic rules that characterize traditional Byzantine jewellery, all structural elements have the same size but differ in the position of the hole and the carvings around it. The hole can be located either in the center of the parallelepiped or in the center of any of the four quarters. Note that, in terms of computer aided design and manufacturing the cylindrical hole can be positioned anywhere in the rectangular parallelepiped; the above restriction follows from careful interpretation of the traditional artistic patterns used.

The diameter of the cylindrical hole is determined based on the aesthetic result and the manufacturing constraints and is proportional to the size of the structural element. If the hole is smaller then the carvings are thinner and not quite distinct, whereas if the hole is larger then the carvings are too thick and there is not enough 'solid' area to contribute to the design shape. For instance, if the structural element is 1 by 1 by 0.5 mm, then an appropriate diameter for the hole is determined to be 0.4 mm (40% of the length of the top square surface).

4.1.2. The repertoire and representation of the structural elements

We introduce a complete representation for the structural elements in the sense that a sequence of strings representing a structural element is provided for every acceptable configuration and every such sequence of strings corresponds to unique solid model for a structural element. To determine a structural element we need to describe the design created by

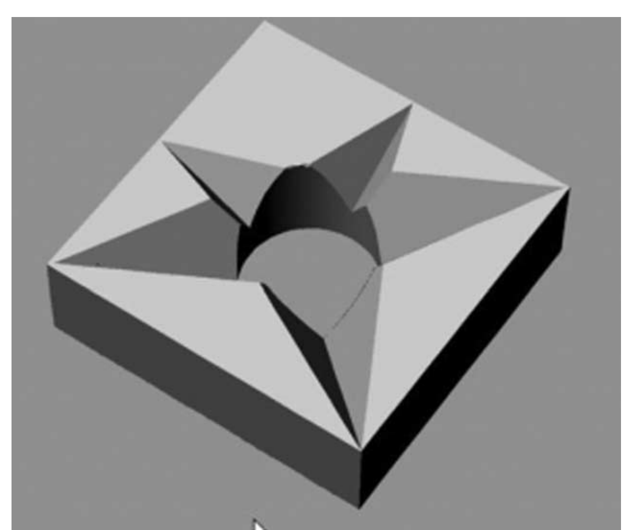


Fig. 4. A structural element.

the carvings and the cylindrical hole. The representation sequence consists of a position keyword and a sequence of numbers. The position keyword describes the position of the cylindrical hole in the rectangular parallelepiped. The position of the hole is determined by positioning the circular area that corresponds to the cross section of the cylinder in the square upper face of the structural element. The position keyword 'Center' corresponds to the circle being placed in the center of the square area. In other cases the position keywords describe the quadrant where the circle is located. For instance, the keyword 'RightUpper' corresponds to a structural element whose cylindrical hole is located in the center of the right upper quadrant of the solid. The position keyword is followed by a sequence of numbers describing the placement of the carvings. The numbers are separated by spaces. Specifically, we have a predefined set of points where the end points of a carving may be placed. The set of points is illustrated in Fig. 5.

Each carving is directed from the cylindrical hole towards one of these points. The carvings are all parts of the same tetrahedron as all carvings were being constructed using the same instrument (chisel). Thus we note that the projection of the tetrahedral carving is always a triangle whose upper vertex coincides with the carving end point and the corresponding angle is always the same (approximately $\pi/3$). The bisector of this angle goes through the center point of the cross section of the cylindrical hole. For instance, the carvings in the structural element of Fig. 6 are directed towards points 1, 3, 4, 6 and 8. Therefore, the representation sequence of the structural element is 'Center 1 3 4 6 8'. With this description, scheme we can represent all valid structural elements and ensure that each representation corresponds to a unique structural element. The order of the position numbers in the sequence is not significant, meaning that the structural element 'Center 1 3 6 4 8' is the same as 'Center 1 3 4 6 8'. This is due to the fact that the sequence simply describes the set of the positions of the carving, without implying any order. However, by convention, we assume that the carvings are described starting at point 1 and moving counterclockwise in the structural element.

Aside from the structural elements described above, there is also a structural element that is compact, with no hole or carvings. This structural element is described by a single keyword ('Solid').

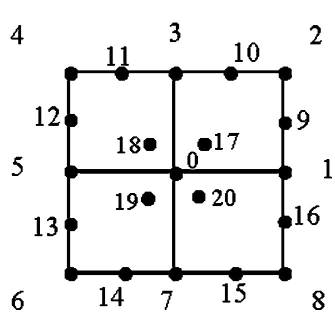


Fig. 5. The valid direction points for carvings in a structural element.

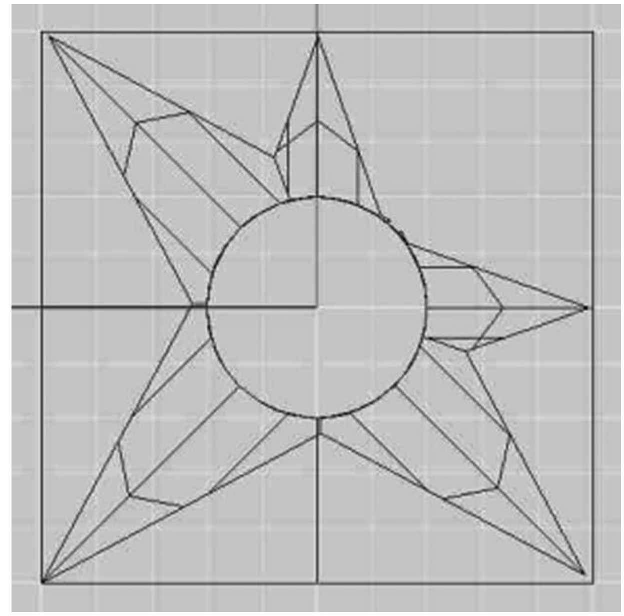


Fig. 6. The structural element 'Center 1 3 4 6 8'.

4.1.3. Validity rules for structural elements

A large number of different structural elements can be created by a hole and various carvings. Not all of these feasible structural elements are valid for use in creating pierced designs. The restrictions concerning the carving directions were determined based on aesthetic and artistic rules. Theoretically, carvings can be directed towards any points on the boundary of the structural elements and we could have any number of carvings. However, in this case, the style of the designs created would not conform to the rules of traditional craftsmanship of pierced Byzantine jewellery. Also when carvings are directed at points that are very close together then the aesthetic result is not appropriate. When the points are more spaced out then the carvings are more distinct and the design is clearer. In order to determine which structural elements are valid for creating pierced designs, the following rules are applied:

(1) If the hole is positioned in the center of the rectangular parallelepiped then,
   i. Each carving starts from the hole.
   ii. A carving cannot be directed towards the hole. The valid points to which a carving can be directed are points 1–8.
(2) If the hole is positioned in one of the four quadrants then,
   i. Each carving starts from the hole
   ii. Each carving is directed away from the hole
   iii. The valid points to which a carving can be directed are 0–20. This, however, is constrained by the position of the hole.
      a. The points that belong exclusively to the quadrant where the hole is located are not valid points for carving ends. For instance, if the hole

is located in the quadrant ‘RightLower’, then points 8, 15, 16, and 20 are not valid end points for carvings, whereas 7 and 1 are, since they both belong to other quadrants.

b. Two carvings cannot be directed to points belonging to the same quadrant. In the previous example, if there is a carving pointing to position 2, then points 0, 1, 3, 9, 10 and 17 are not valid points for a second carving in the element.
c. There can be only one carving pointing to an internal point around the center point 0. This means, for example, that if a carving points to position 18, then there cannot be another carving pointing to positions 17, 19, 20 and 0.

In the library of designs that we have developed for ByzantineCAD we used a subset of the set of structural elements that satisfy these rules. Some characteristics of the structural elements that belong to the subset used so far in ByzantineCAD are (a) the structural elements with holes located in the center have at most five carvings, and (b) the structural elements with the hole in one of the quadrants do not have more than three carvings. As we will see in Section 5.1, however, we can use structural elements not belonging exclusively to the subset, but are part of the set of valid structural elements, to resolve conflicts that arise during scaling. All the rest of the valid structural elements that do not belong to the subset used by ByzantineCAD are available to be used in developing alternative fonts and new patterns. In Appendix A we present the figures of the structural elements used in ByzantineCAD for creating pierced designs.

4.2. Describing a pierced design

4.2.1. The layout description files

Each pierced design is a combination of structural elements. Therefore, every design can be described using a ‘layout description file’, a file where the information needed to construct a specific pierced design is stored.

Each design can be thought of as a two-dimensional matrix (Fig. 7) whose entries correspond to structural elements. The layout description file determines the structural element that must be placed in each position of the matrix.

The first line of a layout description file states the size of the matrix corresponding to the pierced design. The size of the design is expressed as [#rows]$\times$[#columns], expressing the number of rows and the number of columns of structural elements used to create the overall design. The size is followed by a row-by-row description of the design, which is expressed by the names of the structural elements and the corresponding transformations (if any). Each row is described from left to right. An example of a description file is shown in Fig. 8. This file describes pierced letter K. The size of the letter is 6$\times$5 blocks. In Fig. 8 rows in the layout description file are separated by empty lines to illustrate where a new row starts. However, empty lines or spaces are of no syntactic importance.

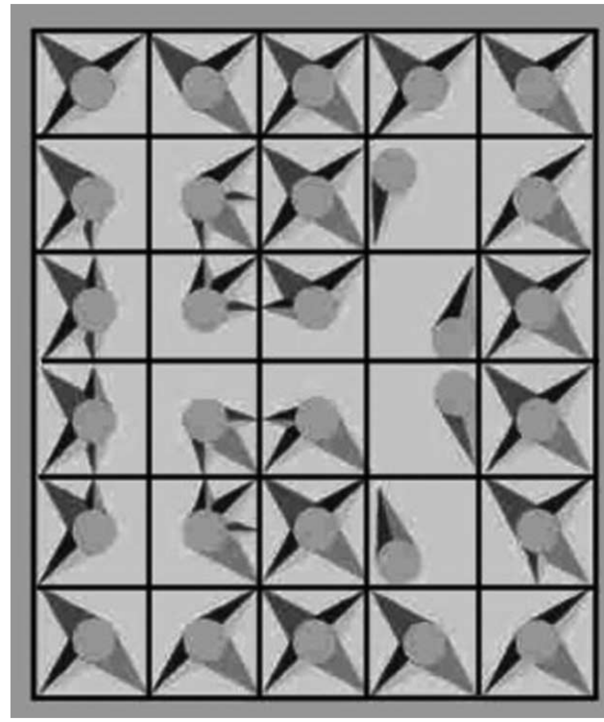


Fig. 7. The letter K as a 2-dimensional matrix.

The layout description files of the letters of the Greek and Latin alphabet have been embedded in ByzantineCAD, along with some characteristic designs found in traditional Byzantine jewellery. It is possible for additional designs to be used by the system, as long as their layout description files are provided. The end-user can use an ordinary text editor to construct such designs by manually recording the sequence of structural elements that form the design. In addition, the system provides a user-interface where the end-user can manually select valid structural elements and combine them to create new designs. After a new design is created, the user needs to store the sequence as a simple text file (the layout description file) and then this file can be imported to the ByzantineCAD library to enhance the repertoire of available pierced designs.

4.2.2. The process of constructing a pierced design

A pierced design is created by using its layout description file. Each time the name of a structural element is read, it is created, transformed (if necessary) and then translated to the proper location. The horizontal and vertical translations of the element are calculated using the equations:

$$x = h \times l \tag{1}$$

$$y = v \times k \tag{2}$$

where x, y are the horizontal and vertical translations, respectively, h is the number of structural elements already placed horizontally in the current row, v is the number of

```
6*5

Center 2 4 6
Center 2 4 8
Center 2 4 6 8
Center 2 4 6
Center 2 4 8

Center 4 6 7
Center 1 2 7 8
Center 2 4 6 8
LeftUpper 6
Center 2 6 8

Center 3 4 6 7
Center 1 2 3
Center 2 4 5
RightLower 2
Center 2 6 8

Center 3 4 6 7
Center 1 7 8
Center 5 6 8
RightUpper 8
Center 2 4 6 8

Center 3 4 6
Center 1 2 3 8
Center 2 4 6 8
LeftLower 4
Center 2 4 7 8

Center 4 6 8
Center 2 6 8
Center 2 4 6 8
Center 4 6 8
Center 2 6 8
```

The first row of K is constructed as follows:

Center 2 4 6: is placed in position [1][1]

Center 2 4 8: is placed in position [1][2]

Center 2 4 6 8: is placed in position [1][3]

Center 2 4 6: is placed in position [1][4]

Center 2 4 8: is placed in position [1][5]

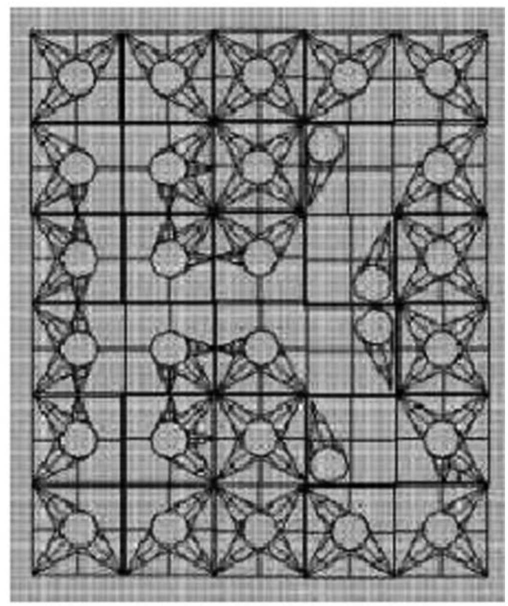


Fig. 8. (left) The description file for the letter K and (right) the pierced representation of letter K.

structural elements already placed vertically in the current column and l, k are the height and length of the structural element.

Each time a structural element is placed, it is unioned with the previous ones. Eventually, a pierced plate representing the design is created.

4.2.3. The process of constructing a sequence of pierced designs

The pierced design on a piece of jewellery can be 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.

Specifically, the layout description files of each individual design are read in parallel and the plate is created row-wise (Fig. 9). First 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.

4.3. The creation of different kind of jewellery

ByzantineCAD is capable of designing rings, bracelets, necklaces and earrings. The process of producing ring and bracelet models is the following: the pierced plate with engraved figures is constructed and then it is bended in a circle around the y-axis passing through the horizontal center of the plate. We consider a plane created by two direction vectors, the bending direction and the bending axis (Fig. 10), that passes through the center of the pierced

Fig. 9. A pierced plate.

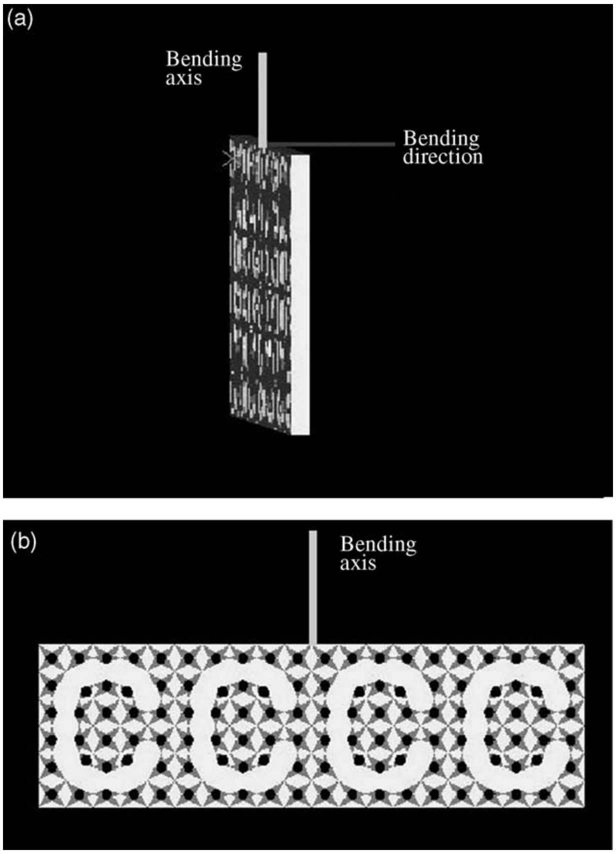


Fig. 10. The directions defined for bending the pierced plate.

plate. The plane divides the plate into two sections. Each section is bended by 180° around the bending axis, towards the bending direction, on a circle. During the bending, the parts of the plate that are located on the circle are not scaled, whereas the rest of the plate is compressed or stretched, in order to create the pierced cylinder (Fig. 11). The points on the back surface of the plate are compressed whereas the points on the front surface are stretched. The points contained between these surfaces, excluding the ones that coincide with the circle, are stretched and compressed proportionally to their distance from the center surface.

For creating a ring or a bracelet, certain parameters have to be defined by the end user. Specifically, the user defines the size of the structural elements, the size of the ring or the bracelet and the designs that are to be engraved on this piece of jewellery. The designer may also choose to add

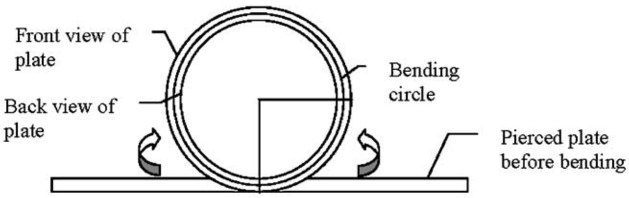


Fig. 11. Cross section of pierced plate during bending.

a beaded border around the ring or bracelet. After the values for the parameters are set, the system checks the validity of the parameters [23], to see if a model can be created based on the specific values. The parameters are

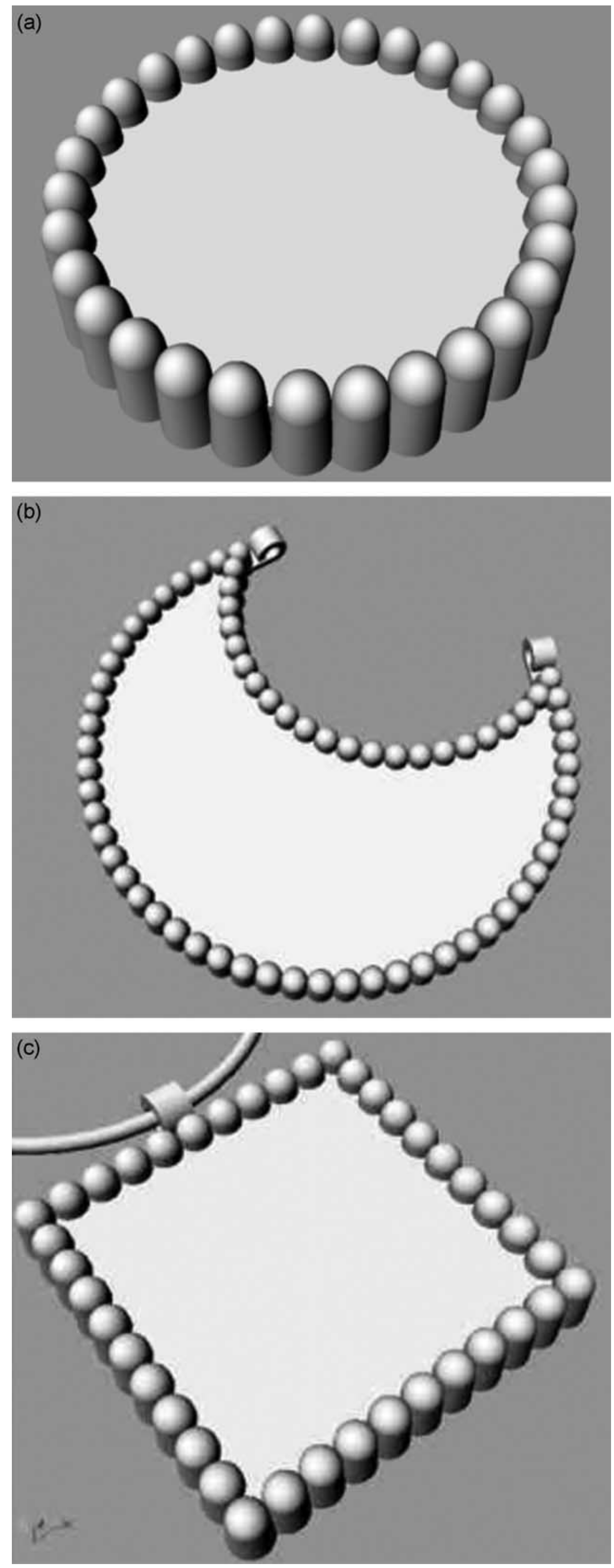


Fig. 12. Some shapes of necklaces and earrings.

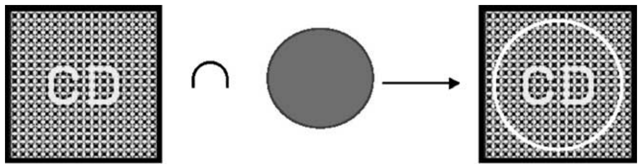


Fig. 13. Intersecting the pierced plate with a cylinder.

related by the following equation:

$$p = cs \quad (3)$$

where p is the size of the plate, c is the number of columns in the plate and s is the size of the structural element.

The number of columns c is an integer, therefore the ratio p/s should be an integer. The system can slightly modify the given parameters so as to ensure that c is an integer. The parameter p cannot be modified, because rings and bracelets have a small fixed set of size values. Thus parameter s is adjusted accordingly.

Earrings and necklaces are created in a number of different shapes, and are decorated with a beaded border (Fig. 12). Each of the above shapes can be considered as a combination of two parts: a solid shape and a beaded border made up of bullet-like solids. Each bead of the border is created by unioning a cylinder with a sphere of the same diameter. The sphere is placed so that its center coincides with the center of the circle that consists the top surface of the cylinder. This creates a solid shaped as a bullet. These bullets are then placed almost tangent to the solid shape around the solid shape and side-by-side to each other, in such a way so that they slightly overlap. Then the bullets are unioned with the solid shape. To create pierced jewellery with these shapes we proceed by constructing a large pierced plate and by ‘cutting’ out of the plate the desired shape. Therefore, we construct an enlarged pierced plate containing different designs, a solid with the desired shape, and we perform an intersection operation between them (Fig. 13). This results in a pierced jewel in the desired shape (Fig. 14).

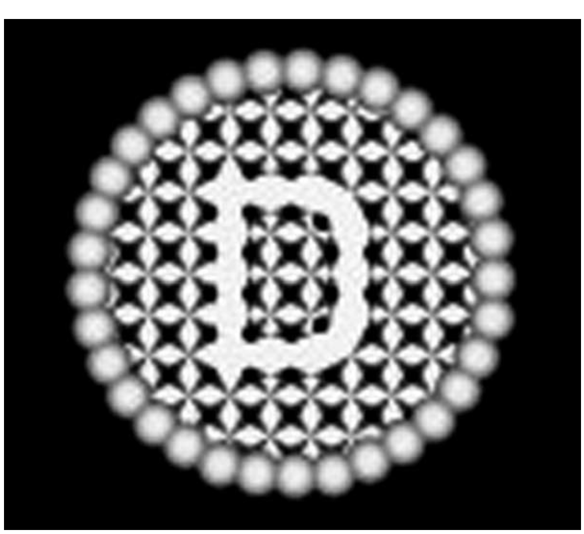


Fig. 14. A pierced earring in the shape of a circle.

The parameters that have to be defined by the end user are: the shape of the jewel, its size, the size of the structural elements and the designs that are to be put on the jewel. The relation that is checked and modified for validity is:

$$j = cs \quad (4)$$

where j is the size of the piece of jewellery, c is the number of columns in the plate and s is the size of the structural element.

5. An algorithm for scaling pierced patterns

For designing jewellery depicting complex scenes it is important to have the capability of enlarging a pierced design without altering the size of the structural elements used to construct it. Having this capability we may include, for instance, different font sizes in the same design. For this reason we have developed a scaling method for pierced designs.

As mentioned earlier a pierced design is thought of as a 2-dimensional matrix whose every entry contains a structural element. Respectively, the scaled version of a design will be a larger 2-dimensional matrix of structural elements (Fig. 15).

5.1. Scaling levels and thickness factor

The scaling that can be achieved is discrete, because of the need to preserve symmetries that may exist in the original design. For instance, letter B is symmetric by a horizontal axis that goes through the middle of the design. A design is scaled by means of new rows and columns added to it. If we add only one new row to letter B, the letter becomes asymmetric, because if the row is added to the upper half or the lower half of the design, then the letter’s shape is altered unintuitively. Also if it is added in

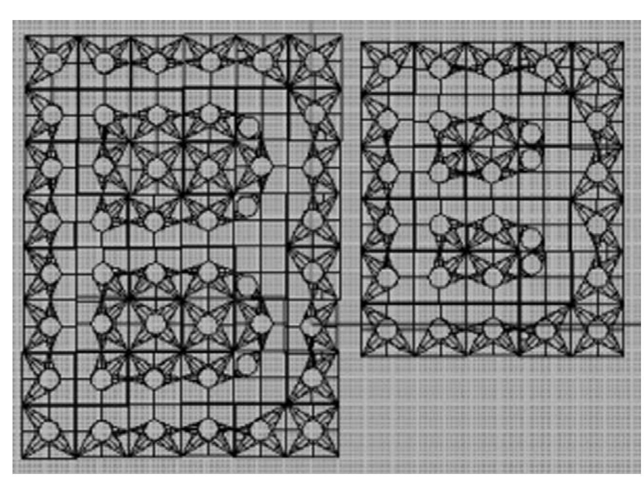


Fig. 15. The letter B in its (left) scaled and (right) original form.

the middle, the design becomes unproportionally thicker at the middle and therefore its original shape is modified. These restrictions are best expressed by the following rules for scaling upwards:

- avoid adding one row, or one column, and
- the number of rows and the number of columns must be integer

As a consequence of the first rule 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 to a scaling factor. Level 1 corresponds to scaling the design by a factor of 1.33, Level 2 corresponds to a scale factor of 1.66 and so on.

A pierced design is represented by a 2D matrix whose entries correspond to structural elements. For instance let us consider the Level 1 scaling of a letter of font size $6\times c$. When scaled to Level 1 a design is transformed from a $6\times c$ matrix to an $8\times k$ matrix (8 is the closest integer to $6 * 1.33=7.98$), where c and k are the number of columns of each matrix. The number of columns in the scaled design depends on the original number and is calculated in the same manner.

For example, let us consider the scaling of letter O (Fig. 16). The size of the design, according to its description file, is 6×5. We would like to scale it up to Level 1, which means that the new design will have 8 rows. The number of columns, based on the original design, is

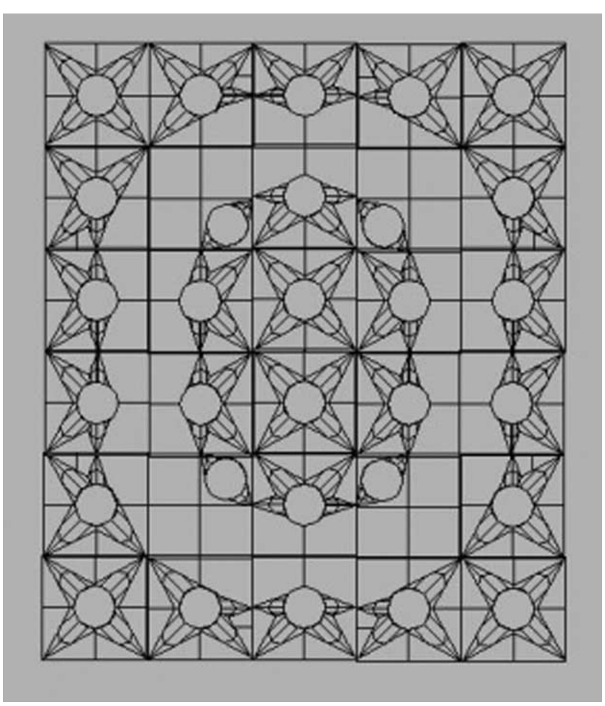


Fig. 16. The pierced letter O.

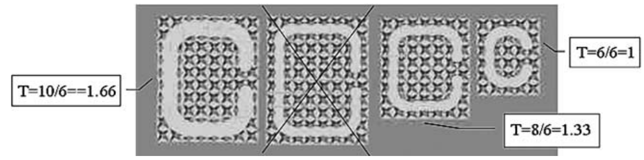


Fig. 17. Different scaled versions of the letter C.

calculated to be 7 (7 is the closest integer to $5 * 1.33=6.65$), therefore a 8×7 matrix is created and it remains to be filled in with the structural elements of the scaled design.

We observe that while scaling a design, as the design gets larger, there is a need for thickening the engraved shape, so as to preserve its original form. In Fig. 17 we see an original design of the C (rightmost design), which is scaled to two different levels (Level 1: 8 rows and Level 2: 10 rows). There are two different versions of Level 2 scaling. The crossed out version of letter C is not valid, because there is a distinct difference in the font style, compared to the original design. Therefore, the interior shape in Level 2 has to be thickened. We define as thickness factor T the ratio:

$$T = \frac{\text{Number of rows in scaled design}}{\text{Number of rows in original design}} \tag{5}$$

In the initial pierced design, the solid area that forms the shape that is created by two structural elements corresponds to approximately 90% of the whole area covered by the structural elements. Therefore the thickness factor is used in combination with this initial thickness to determine the thickness that the scaled design must have.

The thickness H of the scaled design is determined by the product of the initial thickness $H_0=0.9$ and the thickness factor. The quantum of the thickness increase is 0.5. Thus the discrete thickness H_d is expressed by the following equations:

$$H = H_0 T \quad H_\mathrm{d} = \mathrm{round}(2H)/2 \tag{6}$$

For example, if the initial design consists of 6 rows, and the second level scaled design consists of 10 rows, then the product of the thickness factor (10/6) and the initial thickness (0.9) is 1.5, which is the thickness the scaled design must have.

If $H_\mathrm{d}=1$, the thickness of the curves inside the design is not altered. If $H_\mathrm{d}=1.5$ the thickness is increased by 50%, if $H_\mathrm{d}=2$ the shape thickness is doubled (100% increase) and so on (Fig. 17).

5.2. The scaling algorithm

The idea behind the scaling method is to gradually scan the design row by row using a sliding 2×2 window of structural elements, scale individually the 2×2 windows

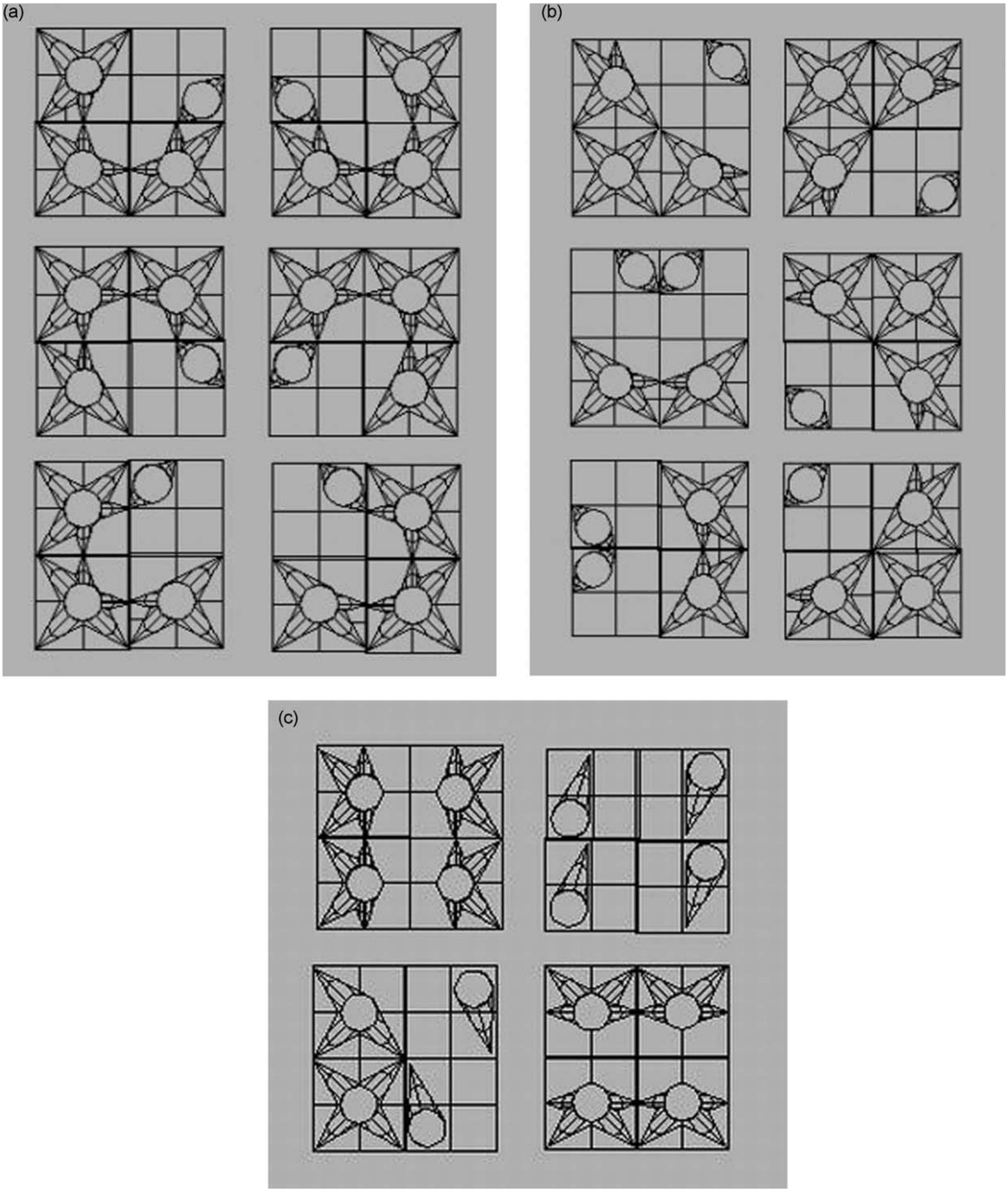


Fig. 18. Different categories of 2×2 structural element windows. Categories: (from left-to-right) “CURVE ENDS”, “CURVE SEGMENTS” and “LINE SEGMENTS”.

of the design and then integrate smoothly the scaled overlapping parts to create the scaled version of the design. The combinations of the structural elements form different designs that can be categorized accordingly. An example is shown in Fig. 18.

The original design is thus scanned using a 2×2 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.

Below we present an outline of the algorithm. Steps 1–4 are explained in more detail later:

The scaling algorithm

for i = 1 to n

for j = 1 to m

step 1: Consider the 2x2 window of structural elements: W[i, j]= [D[i, j], D[i+1, j], D[i, j+1], D[i+1, j+1]]

step 2: Determine the new magnified window W_s. This window will be 2x2, 2x3, 3x2 or 3x3 according to the category and position of the original window.

step 3: Update the corresponding positions of the new scaled matrix D_s by placing the magnified window W_s so as its upper left corner goes to [i, j]. If any such value conflicts with previous values of D_s then integrate them so that the two overlapping windows join smoothly with each other

end for

end for

step 4: Go through Ds searching for empty entries and fill them in with the neutral structural element.

where n is the number of rows and m is the number of columns, D is the matrix describing the original design, D_s is the matrix describing the scaled design, W is the sliding window, and W_s is the scaled sliding window.

The time complexity of the algorithm is O (nm), where n is the number of rows and m is the number of columns of the original design.

Before scanning and scaling, datum positions are marked in the scaled design matrix. We consider the structural elements positioned at North, South, East, West, South–East, South–West, North–West and North–East as our datum 'points'. These reference points are used for ensuring that symmetries are preserved and that the various proportions of the shapes within the design are maintained. Fig. 19 depicts the eight structural elements used as reference points. When the number of rows of the scaled design is even reference points E and W are duplicated. Respectively, when the number of columns of the scaled design is even reference points N and S are duplicated.

Step 1: Every time a window scan is performed, a combination of 4 structural elements is returned.

Step 2: This combination is scaled individually and placed appropriately in the scaled design matrix. The scaling of the 2×2 block of structural elements is determined by the following principles:

- The relative position of the block in the original design should be maintained in the scaled design.
- The datum points should be respected.
- If the block contains part of a curve of a shape the corresponding curve should be scaled appropriately.

The size of the scaled combination is normally 3×3 for Level 1 scaling. However, according to the above principles the size of the scaled window may be reduced to 3×2 or 2×3 (one column or one row truncated), or 2×2 (one row and one column truncated). For the other levels of scaling, the size of the scaled combination is determined proportionally.

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 new scaled matrix so as to overlap previous scaled windows. The overlapping is used to ensure that the connection among neighboring cells is a valid one. When two overlapping structural elements are not the same, then we have a conflict which has to be resolved.

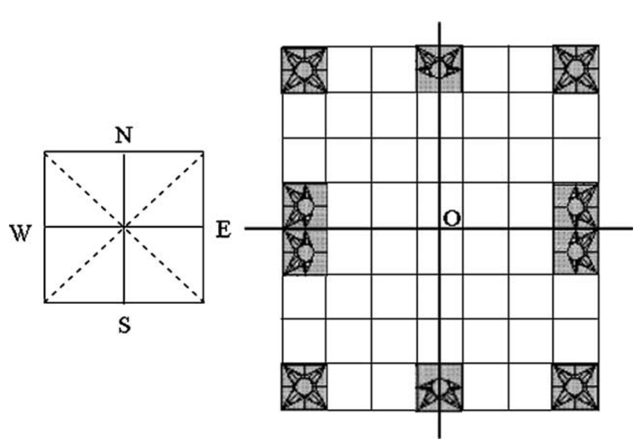


Fig. 19. Datum positions are marked in the scaled design matrix and O is the center of the coordinate system created by the datum axes.

5.2.1. Overlapping and conflict resolution

Suppose S is the 2×2 sliding window which we get from step 1. After scaling this combination of structural elements we obtain a scaled version S' of the sliding window. The scaled sliding window S' is placed in the scaled design matrix positioned in such a way so as to overlap previously scaled sliding windows. Suppose P' is the previously scaled window on the left of S', P'' is the previously scaled window located directly above S'. If S' is being placed in the first row of the design matrix then we overlap the last column of P'' with the first column of S'. If S' is placed in one of the other rows then we not only overlap with P″ but also with P″ by placing the first row of S′ over the last row of P″.

After appropriately placing S′ we check to see if any conflicts have occurred. A conflict occurs when overlapping elements do not match exactly. In this case, one of the overlapping elements must be picked to occupy the scaled design matrix position.

Conflict resolution algorithm

segments, ‘line intersections’ and other designs. The scaled form of a combination tries to approximate the original form as much as possible. Let’s consider, for instance, the first window combination of the letter O. This combination creates a curve that forms an approximately $\pi/3$ angle with the horizontal axis. The scaled version of this combination must also be a curve with the same angle.

Compare the structural elements that overlap:

Step 1: For all the structural elements that are neighbors (top, bottom, left, right, and diagonal positions) to the conflicting structural element do:

If the neighboring element does not have carvings whose ends point to the conflicting element, then ignore it.

Else consider the end points of the carvings that are directed toward the conflicting position.

end for

Step 2: Determine the valid hole positions for the structural element that is to fill the conflicting position, based on which part of the structural element must be solid area so as to continue the shape design that is created by all the neighbors.

Step 3: Choose the most approximate element based on the orientations the carving ends must have in order to meet up with the carving end points of the neighboring elements that contribute to the shape design and the possible hole positions determined in step 2. First we examine the conflicting elements to see if any of the two are appropriate, otherwise we examine the rest of the set.

A structural element can always be found to occupy the conflicting position so that it conforms to the neighboring elements and continues the shape design correctly. A large number of cases are covered by the mirrored and rotated versions of a structural element. The other cases are covered by using structural elements that belong to the more general group of valid structural elements that were defined in Section 4.1.3, from which we only use a subset in ByzantineCAD. In cases where a structural element with a hole in the center is used, we can ensure the continuation of a ‘curve’ shape because we have all possible combinations of carvings. Otherwise, by using elements whose hole is not in the center, we can handle shapes where possibly there is a change in the shape angle-wise.

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 the neutral structural element ‘Center 2 4 6 8’.

Example: In the following we explain how the algorithm will scale the design of Fig. 16. In general, the scaling method can be thought of as a transformation function through which a $6\times c$ design is scaled up to a larger one. The input of the function is the 2×2 scan window and the output is a 2-dimensional matrix of various sizes, according to the placement of the window combination.

The way a 2×2 combination of structural elements is scaled depends on the pattern created by the elements. Specifically, the structural elements can create curves, straight lines (horizontal, vertical and diagonal), line

In order to scale each individual 2×2 sliding window, we consider a global coordinate system created by the axes passing through the datum points and intersecting at point O in the scaled design matrix (Fig. 19). We scale each 2×2 sliding window combination with respect to this global coordinate system and we look to preserve the end points of the curve created by the structural elements and the internal carving end points because when structural elements that have carvings whose end points meet are combined together, then ‘continuous curves’ are formed. For instance, in Fig. 20 we see that the outer curve is created by two structural elements, and the end points of the curve are the points P1 and P2, whereas the internal carving point is P3. We scale the end points and match them with structural elements whose carvings fit appropriately. Structural elements that have carvings that point to these end points are selected, based on their hole position and the number of carvings that they have. Initially, we try to scale each 2×2 combination of structural elements up to its 3×3 form. In the case where structural elements cannot be found to continue and complete the shape being scaled, then we reduce the size of the scaled version by one row or one column and repeat the above process.

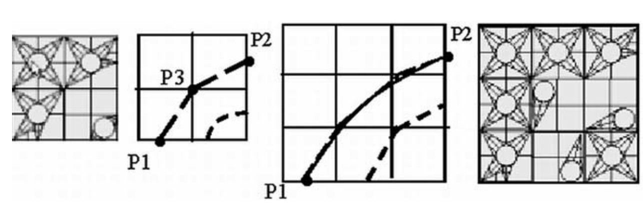


Fig. 20. The process of scaling a curve combination.

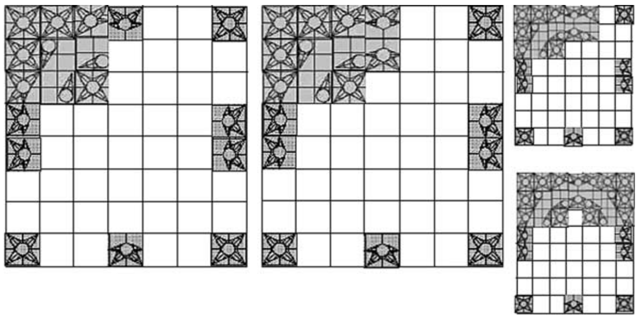


Fig. 21. The scaling of the letter O: (a) the first scaled window combination (3×3) is placed starting from [1][1], (b) the second scaled combination (2×2) is placed in [1][3], by overlapping the last column of the combination of (a), (c) the third combination of elements (2×2) is placed starting from position [1][4], (d) the first row of the initial design has been scaled.

After the scaled version of the 2×2 combination window is determined, it is placed in the correct position in the scaling matrix. In Fig. 21a we see in the upper left corner that a 3×3 scaled window has been placed starting from position [1][1] of the scaled design matrix. The window is shifted one position to the right and the next combination is examined. This combination is the extension of the previous curve to a horizontal line. Theoretically, the specific form could be scaled to the form shown in Fig. 22. However, each quadrant of the design is scaled up until the datum point is reached and then it is connected with the datum's element. In our example, the element in position [1][2] of the scan window is the datum component, therefore the scaled version of this combination must not exceed that mark. Consequently, the scaled version of the specific form is the same 2×2 combination. In Fig. 21b we can see that a second scaled window combination has been placed in the scaled design matrix, starting from position [1][3]. The combination is placed in the design matrix by using overlapping. The combination to be placed is always positioned in such a way as to overlap the last column or row of the previous combination with the first column or row, respectively, of the newly placed one. This overlap, however, results in a conflict of overlapping elements. The conflicting element is located in position [2][3] of the scaled design matrix (Fig. 21b). This position is currently occupied by a 'RightLower 6' structural element. The overlapping element in this position is the structural element 'RightLower 1 7'. To determine the structural element that will fill the position, we examine the neighboring elements already placed. From Fig. 21a and b we observe that position [3][2] is occupied by a structural element whose carving points to the lower left corner to the conflicting element. Therefore, the element in position [2][3] must have a carving that is directed towards that point. The structural element that satisfies this condition is the one already occupying the position, therefore the structural element placed in this position is not replaced.

Fig. 22. A combination of structural elements forming a curve end and a horizontal line, and its possible scaled version.

The window is again shifted in the design matrix and the next combination is a horizontal line ending with the beginning of a curve. This form is the mirror image of the previous combination, meaning that we can scale it in two different ways. In this case the next 2×2 combination is examined. If the next combination is scaled to 3 columns, then the current combination is kept 2×2. Otherwise it is scaled to its 2×3 version. In our case, it is used in its 2×2 form and it is placed in the scaled design matrix.

The window is shifted for the last time in the first row. The last combination is analogous with the first one, and the same methodology is followed.

The scaling window is shifted down one row and it is initialized to the first column. The combination examined is a curve end continued by a vertical line. The scaling process for this combination is again analogous with that of the second step in the first row, only that in this case the form can be scaled downwards by one row (Fig. 23). However, the position [2][1] of the combination window is marked by a datum point element, therefore the accepted form is that of the 2×2 window. This combination must now be placed in the scaled design matrix. The combination to be added is placed so that its first row overlaps the last row of the combination above it. This overlap, however, results in a conflict of overlapping elements. The conflicting element is the one placed in the position [3][2]. This position is filled with the element 'RightLower 2', whereas the overlapping element is 'RightLower 1 7'. To determine which element will finally occupy the position, we examine the carving end points of the neighbors. The elements beside (east side) and above diagonally (northeast side) the conflicting element are examined and based on its carving end points, the proper one is chosen. The element that wins the conflict is the one already occupying the corresponding entry.

This method is continued until all of the design is scanned and scaled. If there are empty spaces in the scaled design matrix, then these are filled with the neutral structural element 'Center 2 4 6 8' (Fig. 24).

The scaling algorithm can also scale down a pierced design, if we use the reverse process. However, this is usually

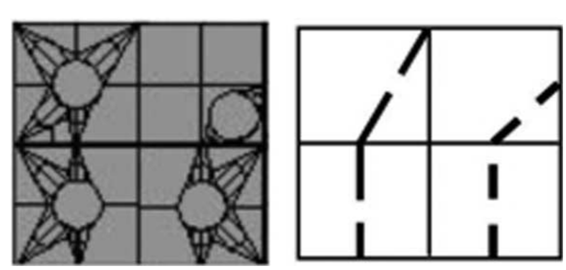


Fig. 23. The analysis of a combination of a curve and a vertical line.

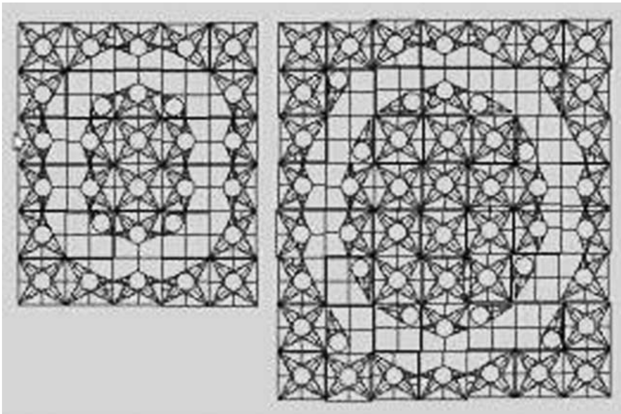


Fig. 24. The letter O and its scaled version.

not needed since the idea behind the scaling algorithm is to create pierced designs of minimal size and to scale up in order to enlarge the designs as desired.

6. Extending ByzantineCAD to encapsulate custom solid figures in a pierced surrounding

In traditional Byzantine jewellery there were pieces in which solid non-pierced designs were placed in a pierced environment (Fig. 25) [9]. Therefore we have expanded our system to have the capability of embedding solid designs in pierced surroundings.

To integrate a solid design with a pierced surrounding, certain conditions have to be satisfied:

(a) The design must be provided in a format that is compatible to the system. For ByzantineCAD the internal solid modeling format used is the ACIS [24] SAT format. The solid design can be created using CAD or modeling systems capable of exporting designs in ACIS SAT format. The designs demonstrated in this section were created using the general purpose CAD system Rhinoceros.

Fig. 25. Byzantine jewellery showing 2 peacocks and a cross.

(b) The center of the design imported is by convention located at the origin in the global coordinate system.
(c) The dimensions of the solid design imported to ByzantineCAD are predefined. We consider the predefined dimensions of the solid to be $10\times10\times2$ units. ByzantineCAD users must be able to express in a simple manner the desired dimensions of the solid design in the piece of pierced jewellery. Therefore we assume that the dimensions of the solid design are expressed through the number of structural elements that correspond to the volume the non-pierced design will occupy.

For instance, suppose we want to include the design in Fig. 26 as part of a necklace.

Originally, the dimensions of the imported solid design are $10\times10\times2$ units. We provide the system with the dimensions of the design, referring to the number of structural elements needed to create the matrix it will be embedded in. For example, if the desired size of the design is 3×3 structural elements (we assume that the width of the design is constant), then the design is scaled appropriately so that its horizontal size is equal to $3\times$(structural element length) and its vertical size is equal to $3\times$(structural element height).

Suppose we would like to insert a solid design into pierced jewellery. Initially, the designer selects the designs, pierced or solid, to be included in the piece of jewellery. ByzantineCAD creates the plate replacing the positions of the solid designs with neutral structural elements “Center 2 4 6 8”, as is shown in Fig. 27.

After the creation of the plate, the files containing the solid designs are loaded by the system and each design is scaled to the appropriate size. A rectangular parallelepiped with the same size as the plate is created and placed at the center of

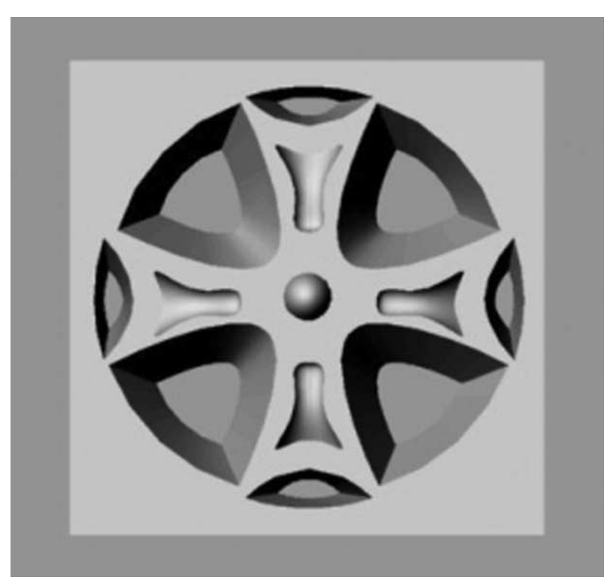


Fig. 26. A solid design featuring a cross.

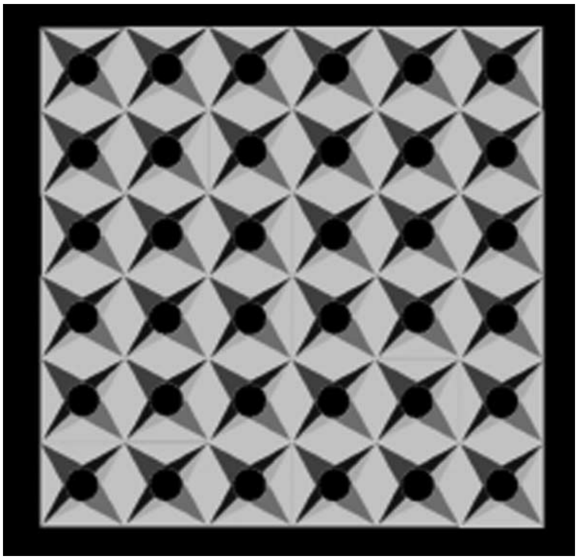


Fig. 27. A complex structure made up of neutral structural elements.

the neutral plate. It is subtracted from the plate (Fig. 28) and the solid design is placed in the empty space that is left and unioned with the rest of the plate (Fig. 29).

7. Implementation issues

ByzantineCAD was implemented under the Microsoft Visual C++ programming environment using ACIS solid modeling libraries by Spatial [24].

ByzantineCAD is a user-friendly system that provides the designer with various capabilities. The user-designer interacts with the GUI of the system and provides the parameter values and then the system creates the model based on these values (Fig. 30).

The CAD system uses the ACIS (SAT) solid modeling format for the internal representation and is capable of exporting to stereolithography (STL) format. The system renders the SAT models, whereas the STL model is ready to be submitted to a rapid prototyping machine, for manufacturing the wax model.

A few problems were encountered during the implementation of ByzantineCAD. The main problems were the long execution time the system needed to create the final model and the robustness of the STL model.

Increasing the memory of the system improved considerably the execution time, since disk swapping was avoided.

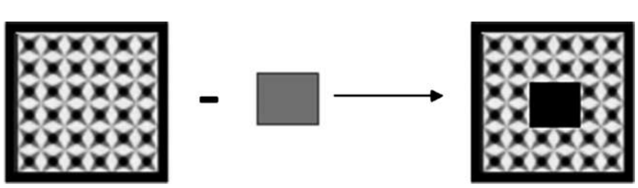


Fig. 28. Subtracting a cubic solid from the pierced plate. The custom solid design is inserted in the empty space.

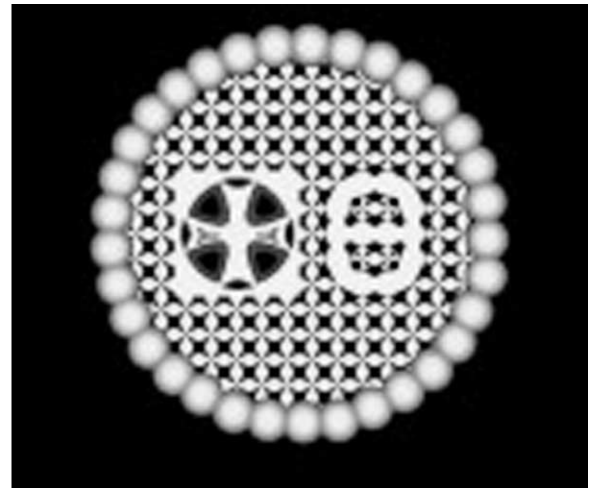


Fig. 29. An example of pierced jewellery with a solid design.

In the case of constructing a ring, on a Windows 2000 Server platform with two 2 GHz processors and 1 GB RAM, the execution time break down is approximately 20% for plate creation, 60% for the bending process and 20% for the STL file creation.

Some of the robustness problems were resolved by changing the tolerances of the system. Specifically, the system variables modified are the smallest representable number and the computer precision. The smallest number representable, that was initially 10–6, was changed to 10–8, whereas the computer precision was changed from 10–11 to approximately 2.22×10–16.

Initially, the structural elements were designed manually using the CAD system Rhinoceros (version 1.1), and exported in SAT format. These elements were imported in our system and used to create the pierced design plates. However, this approach proved both time consuming and error prone. Specifically, during the creation of pierced rings, the data structures used to represent the plate (which was later on bended to form the ring) required a lot of memory leading to disk swapping during the bending operation. Also the ring created from the above plate, with the bending process, had various flaws, such as naked edges, blind holes, and overlapping ring ends. Both of these problems of the models were handled by constructing the structural elements using solid modeling operations.

Also, certain heuristics were adopted to create robust models. Initially, the process for constructing a plate of pierced designs was based on the idea of creating the plate for each design separately, and then unioning all these individual plates to form the final one. When this method was implemented, the final plate model had various robustness flows, meaning that there were either gaps or seams among the individual subplates, which later caused flaws in the STL model. Therefore, the process presented in Section 4.2.3 was finally adopted for the creation of a sequence of pierced designs. However, when the plate is very large (for example in large bracelets and earrings), other techniques are used for

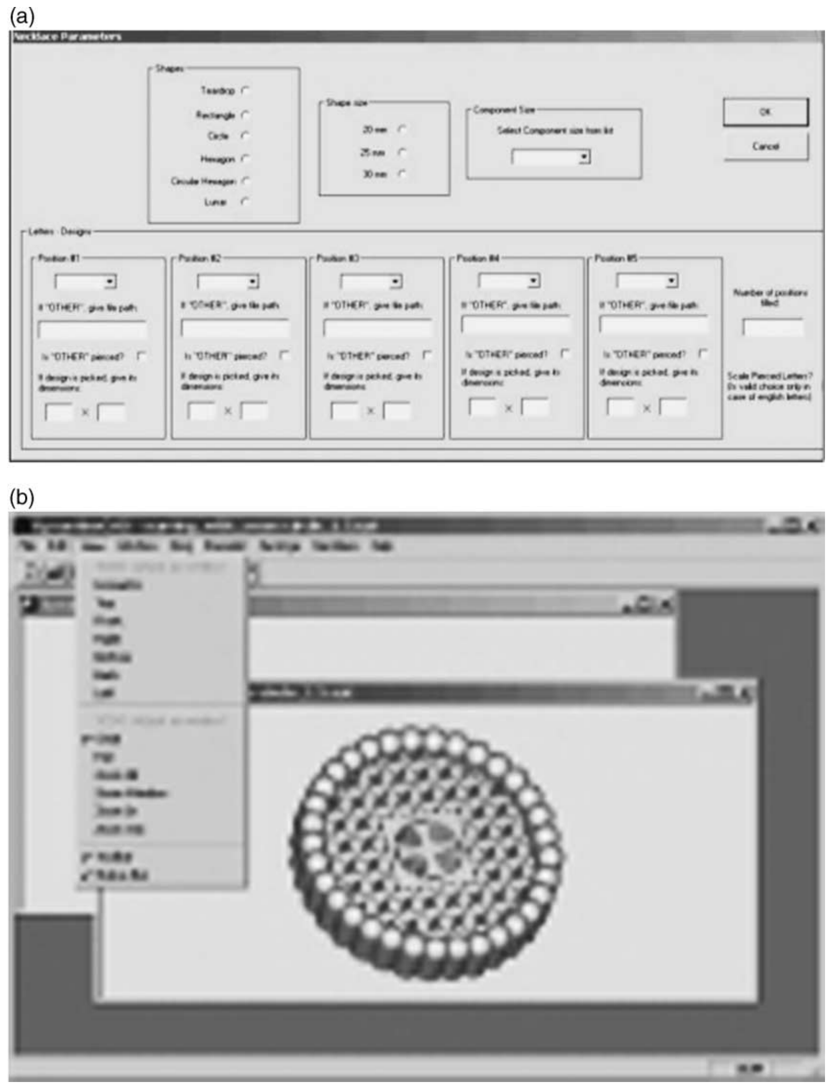


Fig. 30. The GUI of ByzantineCAD: (left) The necklace parameter window, (right) the rendered model.

the creation of the plate. For instance, in the case of a large neutral plate, one row may be created and then replicated to create the remaining rows, which are then all unioned together.

ByzantineCAD was tested by creating STL models and by sending these models to a wax 3D printer. Wax models were manufactured and from these, metal prototypes were created (Fig. 31). From the prototypes created, certain issues regarding the system parameters were re-evaluated and corrections were made so that we have a better final aesthetic result.

8. Conclusions

Pierced Byzantine jewellery is a unique kind of jewellery, mainly due to the special piercing technique through which it is created. This paper has introduced a novel CAD approach to designing handmade objects of complex and sophisticated craftsmanship. We have presented ByzantineCAD,

Fig. 31. A metal prototype of a pierced ring.

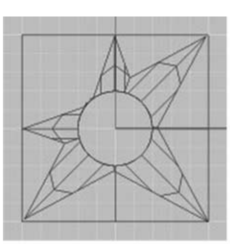
a parametric system for the design of pierced Byzantine jewellery. This system provides the user with the capability of designing custom pierced jewellery in an easy-to-use and efficient manner, using a parametric feature-based design concept. The final piece of jewellery is produced by applying a sequence of operations on a number of elementary solids. An algorithm for scaling pierced patterns and designs has been introduced to enlarge pierced figures without altering the size of the structural elements used to construct them. We have also presented a number of heuristics for enhancing the robustness of the models and for increasing the efficiency of the system.

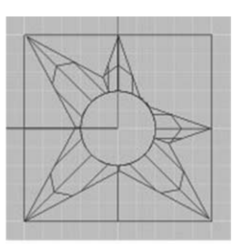
Acknowledgements

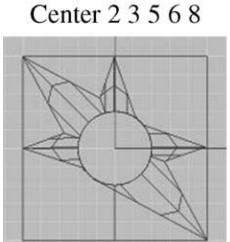
We would like to thank the personnel of Geoanalysis and in particular Sophia Theodoridou, Christina Edipidi and Dimitris Avramidis for testing ByzantineCAD and for manufacturing wax and metal prototypes of the STL models. Also, we would like to thank Jack Ogden for his cooperation in designing the repertoire of structural elements, so that it conforms with traditional jewellery. GeoAnalysis S.A. together with Polyline S.A launched a company called ECHO which utilizes among others the results of the Byzantine project. For more information see http://www.echo-jewellery.com.

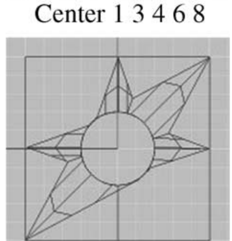
Appendix A

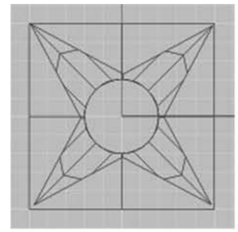
In this appendix we present the set of structural elements that are used in ByzantineCAD to create traditional pierced Byzantine jewellery. The structural elements in this set can be rotated to obtain more structural elements.

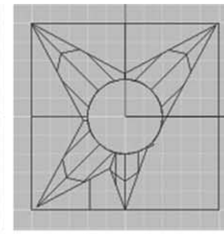

Center 2 3 5 6 8

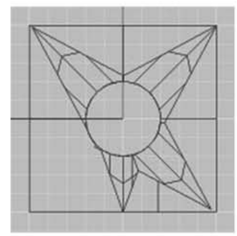

Center 1 3 4 6 8

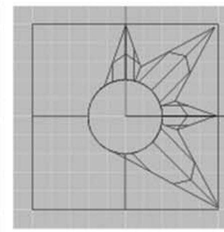

Center 1 3 4 5 8

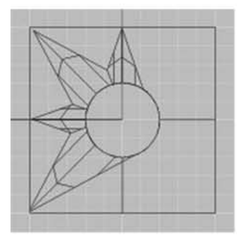

Center 1 2 3 5 6

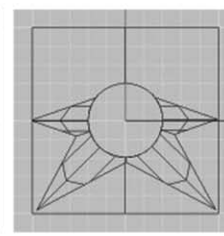

Center 2 4 6 8
(neutral element)

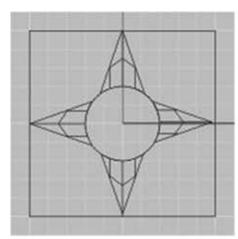

Center 2 4 6 7

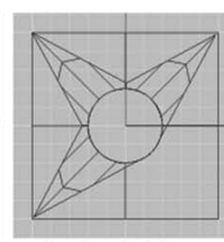

Center 2 4 7 8

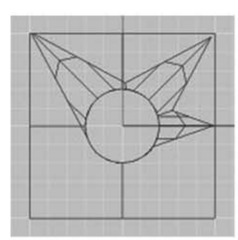

Center 1 2 3 8

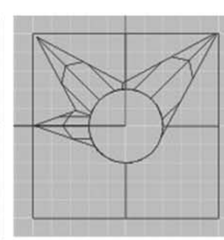

Center 3 4 5 6

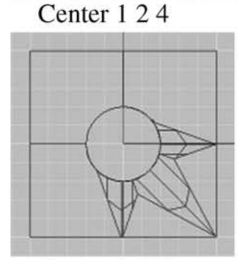

Center 1 5 6 8

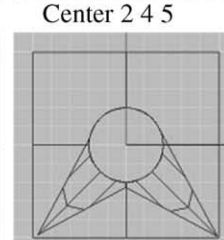

Center 1 3 5 7

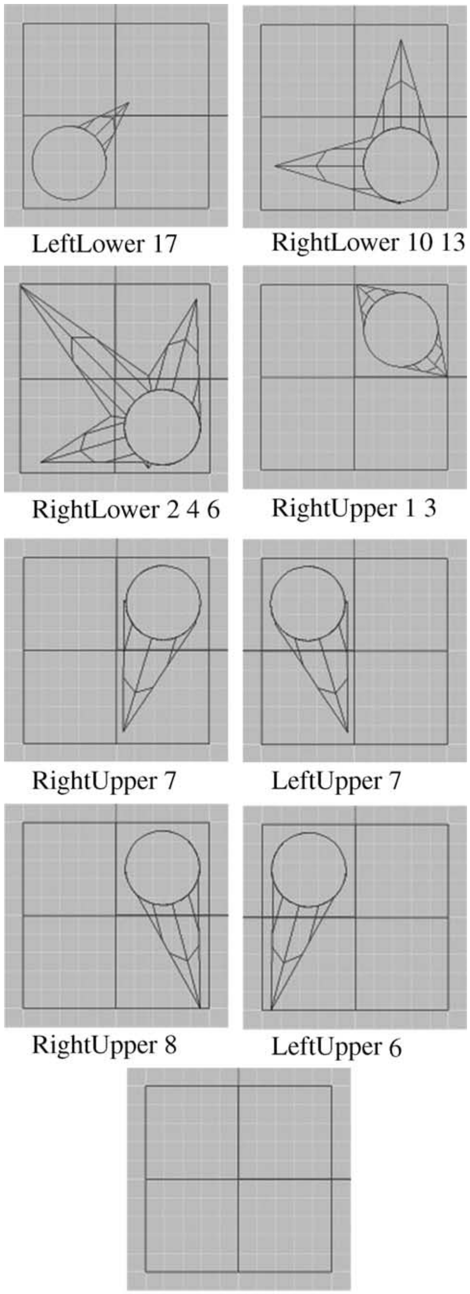

Center 2 4 6


Center 1 2 4


Center 2 4 5


Center 1 7 8


Center 6 8



References

[1] Malagoli MG. CAD/CAM technology—transforming the Goldsmith's workshop Gold Technology; No. 34. Berlin: Springer; 2002 pp. 31–35.
[2] Molinari LC, Megazzini MC. The role of CAD/CAM in the modern jewellery business Gold Technology; No. 23. Berlin: Springer; 1998.
[3] Bohm W. Design opportunities through production technology Gold Technology: No. 23. Berlin: Springer; 1998 pp. 8–11.
[4] Hoffmann CM, Rossignac J. A road map to solid modeling. IEEE Trans Visual Comput Graph 1996;2(1):3–10.
[5] Anderl R, Mendgen R. Parametric modeling and its impact on solid modeling applications Solid Model 1995. Salt Lake City, Utah.
[6] Hoffman CM, Joan-Arinyo R. On user-defined features. Comput-Aided Des 1998;30(5):321–32.
[7] Chiu WK, Tan ST. Using dexels to make hollow models for rapid prototyping. Comput-Aided Des 1998;30(7):539–47.
[8] Jense GJ. Voxel-based methods for CAD. Comput-Aided Des 1989; 21(8):528–33.
[9] Geroulanou A, Diatrita, Benaki Museum Publications, Athens 1999.
[10] Ogden J, Schmidt S. Late antique jewellery: pierced work and hollow beaded wire. Jewell Studies 1990;4:5–12.
[11] Ogden J. Ancient Jewellery, London 1992.
[12] Shah JJ, Mantyla M. Parametric and feature-based CAD/CAM. New York: John Wiley and Sons; 1995.
[13] Salomons O, van Houten F, Kals H. Review of research in feature-based design. J Manufact Syst 1993;12(2):113–32.
[14] Gemvision, Matrix, http://www.gemvision.com/html/matrix.html.
[15] Folini F, Galloni M, Cugini U. Rapid prototyping and parametric CAD systems 27th International Symposium on Automotive Technology and Automation (ISATA) (Aachen, Germany 31 October–4 November, 4 1994 pp. 51–57.
[16] Jewellery CAD/CAM Ltd. JewelCAD, http://www.jcadcam.com.
[17] Caligari Software. JewelSpace, http://www.jewelspace.net.
[18] Robert McNeel, and Assoc. Rhinoceros - NURBS modeling for Windows, http://www.rhino3d.com.
[19] Delcam. ArtCAM Jewelsmith, http://www.artcamjewelsmith.com.
[20] TechJewel. TechGems 3.0, http://www.techjewel.com.
[21] Takumi Pro. JCAD 3, http://www.jcad3.com.
[22] Cadsource. CADjewel, http://www.cadsource.com.au/software.
[23] Hoffmann CM, Kim K. Towards valid parametric CAD models. Comput-Aided Des 2001;33(1):81–90.
[24] Spatial, ACIS, http://www.spatial.com.

Vasiliki Stamati is a PhD candidate in the Department of Computer Science at the University of Ioannina, Greece. She received her MS degree in 2003 and her BS degree in 2001 from the University of Ioannina, both in Computer Science. Her research interests include computer-aided design, feature-based and parametric design, solid modeling and reverse engineering.

Ioannis Fudos is Assistant Professor of Computer Science at the University of Ioannina, Greece. He received his PhD degree from the Department of Computer Science at Purdue University, USA in 1995. His research interests include geometric constraint solving, image retrieval, graphics and visualization systems, computer-aided design and reverse engineering. He has also obtained several research grants in reverse engineering, computer-aided design and geometric constraint solving from national and European Union agencies.