A Parametric Feature-based CAD System for Reproducing Traditional Jewelry

Vasiliki Stamati and Ioannis Fudos

University of Ioannina, {vicky,fudos}@cs.uoi.gr

ABSTRACT

We introduce an automated parametric CAD system for the design of pierced medieval jewellery, 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 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, traditional jewellery, voxels, CAD, scaling algorithm.

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. Editing and redesigning are feasible in a 3D CAD environment, as opposed to the difficult task of fixing 2D design sketches. 3D rendering helps the artist correct or redesign parts of the model that are unsatisfactory, or are to be determined by the customer [3]. Also, in a 3D CAD system the designer has various tools to assist him/her in designing a piece of jewellery, including transformations, primitive solids, and embedded libraries of jewellery. 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.

However, 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, which is jewellery that is designed and created by hand using piercing, a traditional Byzantine technique.

Pierced Byzantine jewellery is gold jewellery 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 and the special aesthetic effect [4]. Pierced jewellery was created from thin sheets of gold on which designs were engraved with a thin sharp tool. After outlining of the designs, holes were created that followed the shapes of the designs and they were decorated with triangular carvings, using an iron chisel (Fig. 1) [5, 6].

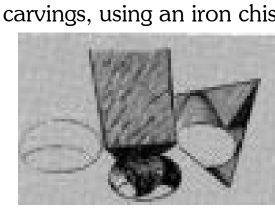

Fig. 1. (left) Using a chisel to create carvings around a hole; (right) A pierced Byzantine bracelet.

In this paper we present ByzantineCAD, a parametric CAD system suitable for the design of pierced jewellery. The system is automated and parametric [7, 8], meaning that the user-designer merely 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 customer. We introduce a feature-based [9] and voxel-based [10, 11] approach to designing jewellery, by defining elementary structural elements with specific attributes and properties that are used as building blocks to construct complex pierced designs.

This paper describes the development of a parametric feature-based CAD system for designing and manufacturing jewellery.

In particular, it 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 how to produce robust solid models ready for 3D printing.

Section 2 refers to the reasons for which current commercial CAD systems are not suitable for the design of this particular family of jewellery. Section 3 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 4 presents a method for scaling pierced designs, while maintaining a fixed size for the basic components used to construct them. Section 5 presents our experience in developing this software. Section 6 provides conclusions.

2. CAD SYSTEMS FOR JEWELLERY DESIGN

Many commercial CAD systems have been developed for the purpose of designing jewellery. Some of the more popular CAD systems for jewellery design are JewelCAD [12], JewelSpace [13], Matrix [14], ArtCAM Jewelsmith [15],TechGems [16], JCAD [17] and CADjewel 3 [18]. Most of these systems are parametric and feature-based [19]. They provide graphical interfaces with excellent rendering capabilities. Most of these systems provide built-in libraries of settings and cut gems and stones and advanced feature-based design tools [20]. 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 [21]. 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 sophisticated craftsmanship. In most of these commercial systems, designing is performed manually using various tools and usually the design steps cannot be programmed to be executed automatically. 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, a method which is necessary for reconstructing pierced jewellery in an automated user friendly manner.

Our system 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, 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.

Finally, editing parts of a pierced design in commercial jewellery design systems requires in depth knowledge of feature-based design and solid modeling techniques.

3. BYZANTINECAD: A PARAMETRIC CAD SYSTEM FOR THE DESIGN OF PIERCED BYZANTINE JEWELLERY

By studying the craftsmanship of pierced jewellery it is apparent that a pierced design can be represented by defining and combining appropriate structural elements. These 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. 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.

3.1 An Approach to Designing Pierced Jewellery Using Structural Elements

3.1.1 Description of a Structural Element

By studying pierced jewellery we discovered that the design is made up of cylindrical holes with carvings around them. Each hole with the carvings around it can be considered 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. 2). 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 any of the four quarters.

3.1.2 The Repertoire and Naming Scheme of the Structural Elements

We introduce a complete naming scheme for the structural elements in the sense that a unique name is provided for every acceptable configuration. The name of a structural element describes the design created by the carvings and the cylindrical hole. The name is a string consisting of characters and numbers where the first two characters of the string describe 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 top of the cylinder in the square upper face of the structural element. CT stands for center and corresponds to the circle placed in the center of the square area. In the other cases, letters describe the quadrant where the circle is located (R=right, L=left, U=upper, L=lower). The letters describing the position of the hole are followed by a sequence of numbers describing the directions of the carvings. Specifically, we assume points in a structural element as depicted in Fig. 2.

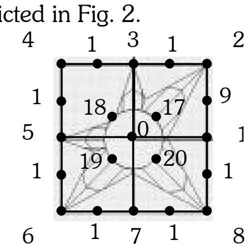

Fig. 2. The valid direction points for carvings in a structural element

Each carving is directed from the cylindrical hole towards one of these points. For instance, the name of the structural element in Fig. 2 is CT23568. The order of the position numbers in the sequence is not significant, meaning that the structural element CT23568 is the same as CT23586.

In the case where the position of a point is described by a two-digit number, such as 19, this number is prefixed with the sign $. For instance, the name RL14$19 refers to a structural element whose hole is located in the lower right corner and contains 3 carvings directed to positions 1, 4 and 19.

3.1.3 Validity Rules for Structural Elements

The number of different structural elements that can be created by a hole and carvings is quite large. Not all of these possible elements are valid for use in creating pierced designs. 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 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 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.

b. Two carvings cannot be directed to points belonging to the same quadrant.

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.

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 called “SP” (Solid Parallelepiped), because it cannot be named according to the naming scheme.

From the group of structural elements that satisfy these conditions in the specific library of design that we have developed for ByzantineCAD we used only a subset, because the elements belonging to this subset are sufficient for the pierced designs that we have built. The rest of the valid structural elements can be used in developing alternative fonts and new patterns.

3.2 Representing a Pierced Design

3.2.1 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. Description files are simple text files containing in a row-wise format the names of the structural elements that the design consists of.

Each design can be thought of as a two-dimensional matrix (Fig. 3 (right)) whose every entry corresponds to a structural element. The layout description file determines the structural element that must be placed in each position of the matrix.

The first line of a description file records the size of the matrix corresponding to the pierced design and it is expressed as [#rows] [#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 the 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. Rows are separated by empty lines. An example of a description file is shown in Fig. 3. This file describes a pierced letter K. The size of the letter is 6x5 blocks. In the example the first row of letter is constructed as follows: element

CT246 is placed in position [1][1], CT248 is placed in position [1][2], CT2468 is placed in [1][3] and so on.

3.2.2 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

```
6*5

CT246
CT248
CT2468
CT246
CT248

CT467
CT1278
CT2468
LU6
CT268

CT3467
CT123
CT245
RL2
CT2468
```

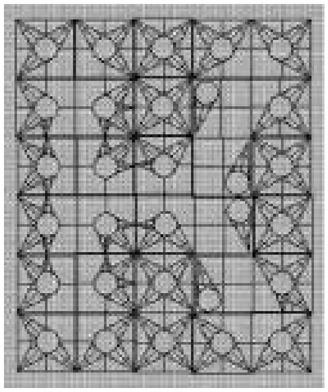

Fig. 3. (left) The description file for the letter K; (right) its pierced representation.

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 structural elements already placed vertically in the current column and l, k are the height and length of the structural element. Every time a structural element is placed, it is unioned with the previous ones. Eventually, a pierced plate representing the design is created.

3.2.3. 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. 4). 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.

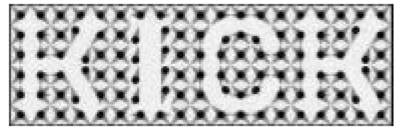

Fig. 4. A pierced plate

3.3 The Creation of Different Kind of Jewellery

ByzantineCAD is capable of designing rings, bracelets, necklaces and earrings. Rings and bracelets are designed in a similar way: the pierced plate with engraved figures is constructed and then 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, that passes through the center of the pierced 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 altered, whereas the rest of the plate is compressed or stretched, in order to create the pierced cylinder (Fig. 5).

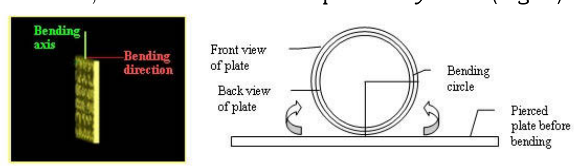

Fig. 5. (left) The directions defined for bending the plate; (right) A cross section of the plate during bending.

For creating a ring or bracelet, certain parameters have to be defined by the end user. The user defines the size of the structural elements, the size of the ring and the designs that are to be engraved on the jewellery. The designer may also choose to add a beaded border around the ring or bracelet. After the values for the parameters are set, the system checks the validity of the parameters [22], to see if a model can be created based on the specified values. The parameters are connected with the following equation:

$$p = c \times s \,, \tag{3}$$

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 integer. The system can slightly modify the given parameter 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. 6). Each shape can be considered a combination of two parts: a solid shape and a beaded border made up of bullet-like solids.

Fig. 6. Some shapes of necklaces and earrings.

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 creates 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, in such a way so that they slightly overlap. Then the bullets are unioned with the solid shape. We would like to create pierced jewellery with these shapes. This can be done 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. 7). This results in a pierced jewel in the desired shape. 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:

$$j \geq c \times s \,, \tag{4}$$

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.

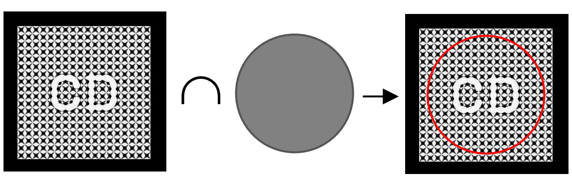

Fig. 7. Intersecting a cylinder and the pierced plate.

4. 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 (Fig. 8).

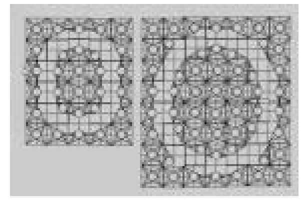

Fig. 8. (left) The original pierced design for letter O; (right) A level 1 scaled version of the letter O.

The percentage of 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, it becomes asymmetric, because if its added to the upper or lower half of the design, then the letter's shape is altered unintuitively. Also if it's added in the middle, the design becomes unproportionally thicker at that point and therefore alters its original shape. These restrictions are best expressed by the following rules for upwards scaling: a) avoid adding one row, or one column, and b) the number of rows and columns is 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 the 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.

4.1 The Scaling Algorithm

A pierced design is represented by a 2-dimensional matrix whose every entry contains a link to a structural element. 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. 8). 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 calculated to be 7 (7 is the closest integer to 5*1.33= 6.65), therefore an 8×7 matrix is created and it remains to be filled in with the structural elements of the scaled design.

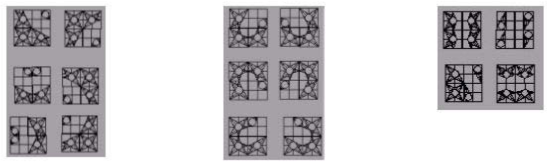

Fig. 9. Different categories of 2x2 windows. Categories (L-to-R) CURVES, CURVE ENDS and LINES.

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 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. 9. The original design is thus 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. Below we present an outline of the algorithm. Steps 1 to 4 are explained in more detail later:

```
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
              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
              the new scaled matrix Ds by placing the
              magnified window Ws so as its upper left
              corner goes to [i, j]. If any such value
              conflicts with previous values of Ds 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,m is the number of rows and columns
respectively, 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
```

Tab. 1. The scaling algorithm

The time complexity of the algorithm is O(nm). 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 useful for making sure that we preserve the symmetries and that the various proportions of the shapes within the design are maintained. Fig. 10 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.

Step1: Every time a window scan is performed, a combination of four 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:

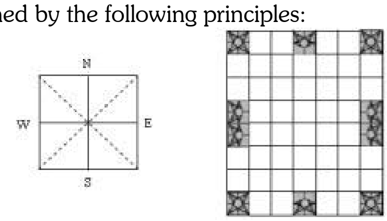

Fig. 10. Datum positions are marked in the scaled design matrix.

- 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. However, according to the above principles the size of the scaled window may reduced to 3×2 or 2×3 (one column or one row truncated), or 2×2 (one row and one column truncated).

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 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 the neutral structural element CT2468.

In the following we explain how the algorithm will scale the design of Fig. 8. The way a 2×2 combination of structural elements is scaled depends on the pattern created by the elements. For instance, the structural elements may create curves, straight lines and other designs. The scaled form of a combination tries to approximate the original form as much as possible.

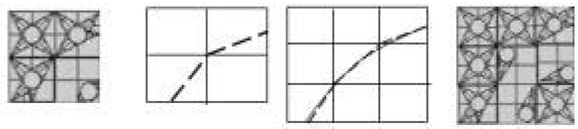

Fig. 11. The process of scaling a curve combination.

Let's consider, for instance, the first window combination of the letter O. This combination creates an approximately $\pi/3$ curve. The scaled version of this combination must also be a curve with the same angle. By analyzing the original curve, we can see that it is created by two structural elements whose carvings create specific angles (Fig. 11). If we consider that each structural element has a north, south, east and west orientation, then we can describe the angles through this orientation. Specifically, the structural element in position [2, 1] of the original design matrix, which marks the beginning of the curve, contains an angle with a S→NE orientation, whereas the element in position [1, 2] which continues the curve, contains a SW→E angle. The starting orientation of the curve is S and the ending orientation is E. Therefore the scaled version of this curve must have the same start and end orientation. After marking these points, the curve connecting them is drawn and then approximated with structural elements, as best as possible.

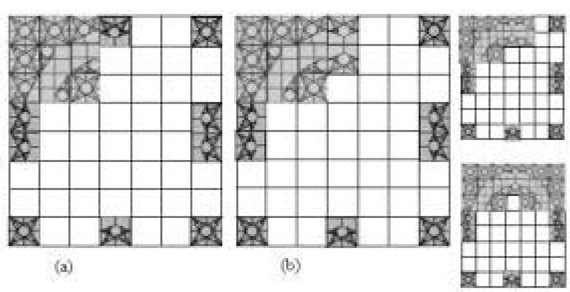

Fig. 12. The scaling of the letter O.

After the scaled version of the 2×2 combination window is determined, it is placed in the correct position in the scaling matrix (Fig. 12 (a)). 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. 13. 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 (Fig. 12 (b)). It 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. 12 (b)). This position is currently occupied by a RL6 element. The orientation of this element is SW→E. The overlapping element in this position is the structural element RL17, with an orientation of S→E. To determine the structural element that will fill the position, we examine the neighboring elements already placed. From Fig. 12 (a), (b) we observe that position [3, 2] is occupied by an element whose orientation is S→NE. Therefore, the element in position [2, 3] must have an orientation beginning with SW. The structural element that satisfies this condition is the one already in this place, therefore is not replaced.

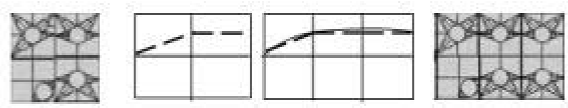

Fig. 13. A combination of structural elements and a possible scaled version.

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 scaling process is again executed as described above, with the difference that the datum points examined in scaling this row are those of located at the West and East points of the scaled design matrix.

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 CT2468 (Fig. 8).

5. IMPLEMENTATION ISSUES

ByzantineCAD was implemented under the Microsoft Visual C++ programming environment using ACIS solid modeling libraries by Spatial [23]. 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. 14). 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. In the case of constructing a ring, on a Windows 2000 Server platform with two 2GHz 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.

Fig. 14. (left) The GUI of ByzantineCAD; (right) the rendered model

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 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. 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 had various flaws, such as naked edges, blind holes, and overlapping ring ends. 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, such as gaps or seams among the individual subplates, which later caused flaws in the STL model. Therefore, the process presented in section 3.2.3 was finally adopted for the creation of a sequence of pierced designs. However, when the plate is very large (for example in large bracelets), other techniques are used for 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. 15). From the prototypes, certain issues regarding the system parameters were re-evaluated and corrections were made so that we have a better final aesthetic result.

Fig. 15. A metal prototype of a pierced ring

6. 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, 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.

7. ACKNOWLEDGEMENTS

This work is supported by PENED Grant Nr. 02ED329 by the General Secretariat of Research and Technology. 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. ECHO creates Byzantine jewellery collections which are based on the pierced technique. For more info see http://www.echo-jewellery.com.

8. REFERENCES

[1] Molinari L. C. and Megazzini M. C., The role of cad/cam in the modern jewellery business, *Gold Technology*, Vol. 23, Spring 1998, pp 3–7.

[2] Malagoli M. G., Cad/cam technology - transforming the goldsmith's workshop, *Gold Technology*, Vol. 34, Spring 2002, pp 31–35.

[3] Bohm W., Design opportunities through production technology, *Gold Technology*, Vol. 23 Spring 1998, pp 8–11.

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

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

[6] Ogden J., *Ancient Jewellery*, London, 1992.

[7] Hoffmann C. M. and Rossignac J., A road map to solid modeling, *IEEE Transactions on Visualization and Computer Graphics*, Vol. 1, No. 2, March 1996, pp 3–10.

[8] Anderl R. and Mendgen R., Parametric modeling and its impact on solid modeling applications, in *Proceedings of the third ACM Symposium on Solid Modeling and Applications*, May 17-19, Salt Lake City, Utah, USA, 1995, pp 1-12.

[9] Hoffmann C. M. and Joan-Arinyo R., On user-defined features, *Computer-Aided Design*, Vol. 5, No. 30, 1998, pp 321–332.

[10] Chiu W. K. and Tan S. T., Using dexels to make hollow models for rapid prototyping, *Computer-Aided Design*, Vol. 7, No. 30, June 1998, pp 539–547.

[11] Jense G. J., Voxel-based methods for cad, *Computer-Aided Design*, Vol. 8, No. 21, October 1989, pp 528–533.

[12] JewelCad, http://www.jcadcam.com, Jewellery CAD/CAM Ltd.

[13] Jewelspace, http://www.jewelspace.net, Caligari Software.

[14] Matrix, http://www.gemvision.com/html/matrix.html, Gemvision.

[15] Artcam Jewelsmith, http://www.artcamjewelsmith.com, Delcam.

[16] Techgems 3.0., http://www.techjewel.com, TechJewel.

[17] Jcad 3, http://www.jcad3.com , Takumi Pro.

[18] Cadjewel, http://www.cadsource.com.au/ software, Cadsource.

[19] Shah J. and Mantyla M., *Parametric and Feature-Based CAD/CAM,* John-Wiley and Sons Inc., 1995.

[20] Van Houten F., Salomons O. and Kals H., Review of research in feature-based design, *Journal of Manufacturing Systems*, Vol. 2, No. 12, 1993, pp 113–132.

[21] Galloni M., Folini F. and Cugini U., Rapid prototyping and parametric cad systems, in *27th International Symposium on Automotive Technology and Automation (ISATA)*, 31 October - 4 November, Aachen, Germany, 1994, pp 51–57.

[22] Hoffman C. and Kim K., Towards valid parametric cad models, *Computer-Aided Design*, Vol. 1, No. 33, January 2001, pp 81–90.

[23] Acis, http://www.spatial.com, Spatial.