

## Specification of VRML in Color Rapid Prototyping

Ling Wai Ming and Ian Gibson\*

Department of Mechanical Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong

**Abstract** – The STL or Stereolithography format, established by 3D systems, gathers the geometric data of the model in a number of triangular surfaces. It can be in ASCII or binary format, and is a *de facto* standard in the Rapid Prototyping (RP) world. RP has developed greatly over the last ten years. In particular, improvement in materials has meant greater part accuracy and strength, which in turn has increased the range of functional applications. Future applications of RP will focus on rapid tooling and direct manufacturing. Direct manufacturing in particular may see much benefit from the incorporation of color into models. For color RP, besides designing new hardware to add color into the prototypes, it is necessary to redefine the CAD software for adding and accurately positioning color onto the model. STL cannot effectively store this kind of information. Among the existing data file formats, VRML is an acceptable one that is complimentary to existing RP processes. This paper acts as a review to discuss several methods of using VRML for coloring model data. This paper will also discuss the problems occurred in coloring the layer contours of the RP model.

**Keywords:** VRML, Color, data interface, Rapid Prototyping

### 1. Introduction

Rapid prototyping (RP) is a process that is capable of converting a 3D CAD design into a physical object by means of layer-by-layer powder, solid or liquid-based manufacturing methods [1]. Generally, RP models are considered to be homogeneous in nature and therefore only the geometric description of the model is essential to the process. Common interfaces for storing geometric information of the model data include IGES [2, 3], HPGL [4], CT Data [5] and one of the most popular, STL [6]. STL is popular because models are defined in a simple manner and so transfer between different software systems is relatively straightforward.

Recently there has been increased interest in developing new RP systems for making multiple-material products. An example of this is Jepson [7], who is developing the SLS process, called Multiple Material Selective Laser Sintering (M<sup>2</sup>SLS), to permit the material composition of components to be varied in a controlled manner. The LENS<sup>TM</sup> process can also be used to fabricate graded or layered multiple-material parts [8]. Parts of this nature that exhibit variable material properties, or heterogeneous models, cannot be easily represented in a 3D format for transfer using the above-mentioned data exchange interfaces.

The addition of color into a model can also be regarded as a change in the material properties of the model. Michael Rees [9] states that color can be a

means for communication through association. Color RP models have already been used to highlight features in medical and related areas. It is an additional tool for communication of concepts and ideas; the contrasting effect of different colors provides a means for distinguishing features in a much easier way than when using monochromatic processes. In Wohlers State of the Industry Report 2000 [10], Z Corporation announced that the first color 3D Printing machine, Z402C, will soon come onto the market. For this new color 3D printer, besides the original ink-jet printing head for giving binder to the powder selectively, there is also an additional printing head for printing specific color to the desired position. This machine uses VRML file to store both the geometric and color information. A similar color printing mechanism is also being developed in TNO Institute of Industrial Technology (in the Netherlands) and by Buss Müller Technology in Germany.

Since color RP is fast becoming a viable commercial process with the realistic prospect of new machines coming on to the market, there is a need to develop a new data interface that can be used to store the color information of the model. In addition to this, to keep pace with the latest work on multiple-material RP machines, a new interface that can be used to store the material information of the model should also be developed. The existing data interfaces, however, are not designed to store such information.

This paper results from a need to provide a useful data exchange format for models that can be processed within a system that is a modification of the SLS process, designed to incorporate color [11, 12]. This system uses a color printer to apply coloured inks onto powder during the conventional SLS process. During this investigation,

\*Corresponding author:  
Tel : +852-2859-7901  
Fax : +852-2858-5415  
E-mail : igibson@hkucc.hku.hk

it was found that there has been work carried out on the development of a colored or 'extended STL' file format [6]. The color handling capacity of this approach is limited to coloration of individual triangles and therefore can only be used to distinguish large geometric features using color. TNO Industrial Prototyping Production Development Division has also developed its own STL format that can be used to store the color information of RP models. Basically it uses the unused space in the file to store that information. However, for complex color format like bitmapped graphics or special color pattern like gradual change in color, this new file format may find difficulties to store such information. Therefore, a new standard for generating RP part files that include complex color information must therefore be considered.

There are two main objectives in this paper. One is to analyse the existing CAD software for storing the color information of the model. The other one is to discuss the framework for creating slice files for color system.

## 2. Definition of color

Color is a material property. It is the visual effect resulting from the eye's ability to distinguish different wavelengths (ranges from 400 and 700 nm) of light. The apparent colors of an object depend on the wavelengths of light that it reflects [23]. There are several formal systems, or photometric interpretations, to specify and represent colors. The most widely recognised system includes RGB (Red, Green and Blue primaries). This is an additive primary system and is popularly used in

representing computer graphics. The CYM(K) (Cyan, Yellow, Magenta, (and black)) system is a subtractive primary system. This is used in ink-jet printers for mixing colors. HSB (Hue, Saturation and Brightness) is a model based on human perception of color. To incorporate color into RP, there should be an interface to store the color information generated during the CAD design of the model, which can be precisely related to the geometric description of the model. Among the existing interfaces, STEP (Standard for the Exchange of Product Data) can store both the material and geometric information of the model [13, 14]. However, it is too rich to apply for rapid prototyping. The size of STEP files can be very large, and the extraction of geometric and color data of the model from the file is quite complex due to its "document-like" structure. It is generally considered to be inconvenient to use. VRML (Virtual Reality Modelling Language) is more preferable as its format is much simpler and has already been widely adapted for graphical representation of 3D objects over the Web. Normally, VRML is used to store 24-bit color (true color) in RGB format, but it is also possible to use the HSB format to store color information.

## 3. Overview of VRML

VRML was conceptualised in late 1994. The VRML 1.0 specification was established in 1995 and is based on the Open Inventor file format [15]. It has evolved through several versions of the language (1.0, 1.0c, 1.1 and 2.0). In 1997, VRML 2.0 was presented to the

**Table 1.** Common VRML browsers

Browser	Comments	Platform
3Space Assistant from TGS (Template Graphics Software)	Interactive viewer for VRML, VRML 97, DXF and Open Inventor files	Windows 9X Windows NT
Actify's 3D View (Actify Corporation)	Support multiple CAD formats: Pro/E, Catia, Inventor, DWG/DXF, IDEAS, Unigraphics, SolidWorks, Solid Edge, Parasolid, IGES, STL, VRML 1.0/2.0	Windows 95 Windows 98 Windows NT Windows 2000
Blaxxun Contact 4.4 (blaxxun interactive, Inc.)	Can support VRML 1.0, 2.0 and 97	Windows 95/98, Windows Me, Windows 2000 and Windows NT
CASUS Presenter (Fraunhofer Institute for Computer Graphics)	Supports VRML 97 and free for noncommercial use	SGI IRIX, Sun Solaris, Linux, Windows 95, 98 and NT
Cortona (ParallelGraphics)	The VRML client for MS Internet Explorer and Netscape Navigator. It fully supports VRML 97	Windows 95 Windows NT Windows CE Macintosh
Cosmo Player (Platinum)	Supports VRML 1.0 and 97. It is one of the most popular browsers	Macintosh SGI Windows 3.x Windows 95 Windows NT
Solid View (Solid Concepts Inc.)	Reads STL, VRML, 3D DXF and other formats and converts to STL, VRML, as well as other formats	Windows 3.x, 95, 98 and NT

International Standards Organisation (ISO) and after minor changes, VRML 97 was established and was ratified by ISO as an international standard in December 1997 (ISO/IEC DIS 14772-1, 1997). VRML is a file format for describing interactive 3D objects and worlds. It is also an interchange format for integrated 3D graphics and multimedia [16]. One aim of VRML is to make it easy to incorporate 3D models into virtual environments. Models can be created with geometry, texture and color. In addition, these models can be animated and sound effects can also be incorporated using this file format. VRML graphics are displayed using a VRML browser. VRML is widely used for communication of model data over the internet and over intranets and therefore VRML browsers are normally displayed within internet browsers (e.g. Internet Explorer and Netscape Navigator). There are about 32 VRML browsers in the market [17] and some of them are shown in Table 1. Most of the browsers listed can show any version of VRML file.

There has been some notable work investigating the incorporation of VRML into RP. Bauer, Fadel and Kirshman [18] suggested using VRML as a RP model interface. Yanshuo [19], on the other hand, developed an SFF (Solid Freeform Fabrication) process planning and virtual prototyping (using OpenGL) system that can read model information from VRML directly. This is then used to generate the laser path-planning data for the SALDVI (Selective Area Laser Deposition Vapor Infiltration) process for prototyping. In relation to VRML browsers, Solid View and Actify's 3D View can change conventional CAD file formats like STL and IGES to the VRML format.

#### 4. Structure of VRML

VRML uses a branch structure as shown in the graph of Fig. 1. The header information is required in every VRML file. This header always starts with **#VRML V2.0 utf8**. The identifier "utf8" indicates a clear text encoding that allows for international characters to be displayed in ISO/IEC 14772 using the UTF-8 encoding defined in ISO/IEC 10646-1. VRML is made up of nodes, which contain mathematical descriptions of three-dimensional (3D) points, lines, surfaces, text strings, and solids [20]. Different shapes can be grouped together by the **Group** node to form a more complex shape. The **Group** node is then divided into a number of **children** nodes, which contain the shapes that make up the group. Each **children** node is then subdivided into a number of smaller nodes like **Shape** node and **geometry** node to describe the geometry and material information of the model.

#### 5. Use of VRML to store the geometric data of the model

Yanshuo and Marcus [19] stated the structure of VRML

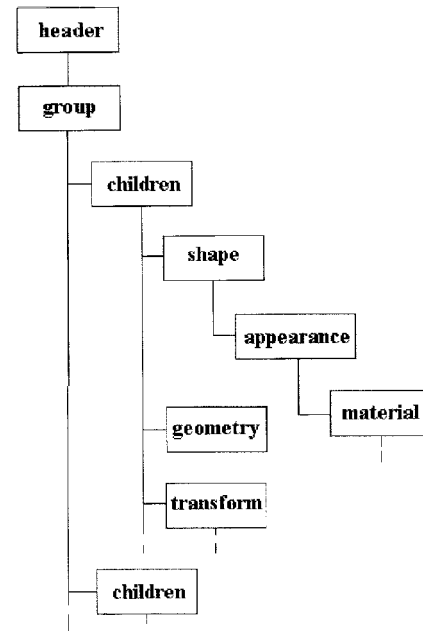


Fig. 1. VRML branch structure.

to store the geometry information of a CAD model. Basically the vertex coordinate of each face is stored in the **IndexedFaceSet** node, which is a sub-branch of the **children** node. The **IndexedFaceSet** node includes the **coord** exposed field which specifies the **Coordinate** node for providing a list of 3-D coordinates. The X, Y, and Z coordinates in the **Coordinate** node is implicitly numbered starting at 0. These numbers are called coordinate indices. There is also a **coordIndex** field that specifies the list of coordinate indices used to describe the face. This also includes the -1 coordinate index for face separators. There is also a **ccw** field to indicate whether the faces are indexed in clockwise or counter-clockwise order. This field will determine the direction of the solid face as shown in Fig. 2. The default value of the **ccw** field is counter-clockwise, which means that the vertex of each face is counted in

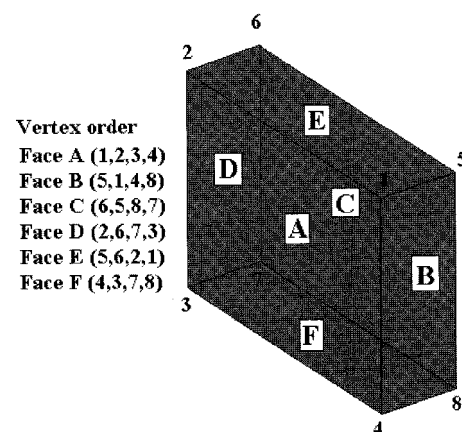


Fig. 2. Direction of solid face in counter-clockwise direction.

counter-clockwise direction.

In VRML, besides this IndexedFaceSet method, there are other ways to store geometric information of the model. This IndexedFaceSet is the most favourite one because it can store model of any shape and has been widely used in CAD software like SolidWorks or Rhineoceros.

### 6. Use of VRML to store the color data of the model

In VRML, it is possible to add color to the surface of the model and there are basically three methods to create and store the color information:

#### 6.1. Adding color binding to the surface

VRML has a **color** field that uses the **Color** node to give the color used to decorate the faces of the solid. This field is included in the **IndexedFaceSet** node. This field contains the RGB color in a normalised form. The normalised RGB color contains three floating-point values, each ranging from 0.0 to 1.0 respectively. The first value of normalised RGB color is red, the second one is green and the third one is blue. All colors are made by the combination of these three primary colors. If any value of red, green or blue is set to 0.0, it means that this color is turned off. Inversely, if any value of red, green or blue is set to 1.0, it means that this color is turned on completely. The **colorIndex** field gives a list of indices used to determine which color in the **Color** node is applied to the surface. Similar to **coordIndex**, it includes the -1 coordinate index to separate the faces. The **colorPerVertex** field uses a **TRUE** or **FALSE** statement to specify which colors in **Color** node are to be used for each face or each vertex of the face. The default field value is **TRUE**. Table 2 shows the cases when the **colorIndex** is empty and when it is not empty:

In the Appendix it shows a VRML color binding example. The **colorPerVertex** in this case is set to **FALSE**. That is, colors in the **color** node are applied to model faces. Fig. 3 is the model displayed through the CORTONA VRML CLIENT VERSION 2.0 browser. [21] Fig. 4 is the model in which the **colorPerVertex** field is set to **TRUE**. Colors in the **color** node are pasted on the vertices of each face and each point in the face is the interpolation of the vertices that join up

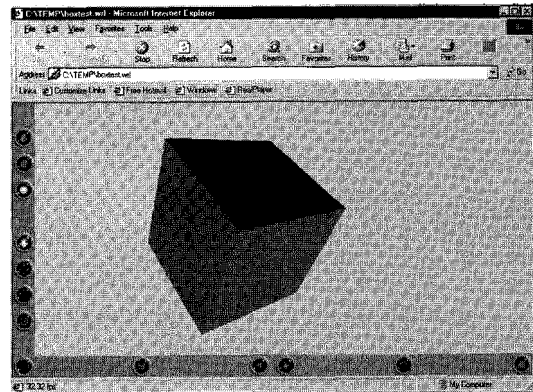


Fig. 3. Color binding model (**colorPerVertex** field is set to **FALSE**).

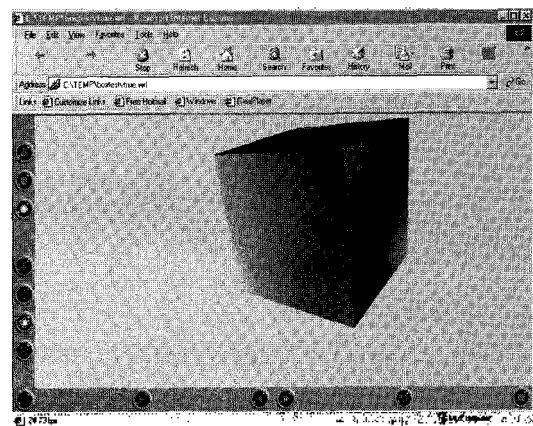


Fig. 4. Color binding model (**colorPerVertex** field is set to **TRUE**).

the face.

#### 6.2. Coloring the whole shape

If rendering of the whole shape with a single color is desired, it is much simpler to use the **emissiveColor** field in the **Material** node. This field also stores the normalised RGB color.

#### 6.3. Texture mapping to the faces

Fig. 5 shows a model in which faces are mapped with textures. In the **Appearance** node it contains the texture field and can be set to **ImageTexture** and **PixelTexture** nodes. In the **ImageTexture** node there is a **url** field that can be used to attach the JPEG, PNG or

Table 2. Cases for the **colorIndex** field

Color index is empty	Color index is not empty
When the <b>colorPerVertex</b> value is <b>FALSE</b> , colors in the color list match the faces in order, one by one.	When the <b>colorPerVertex</b> value is <b>FALSE</b> , color indexes in the color index list match the faces in order, one by one.
When the <b>colorPerVertex</b> value is <b>TRUE</b> , colors in the color list match the coordinate indexes in the <b>coordIndex</b> field in order, one by one.	When the <b>colorPerVertex</b> value is <b>TRUE</b> , color indexes in the color index list match coordinate indexes in the <b>coordIndex</b> field in order, one by one.

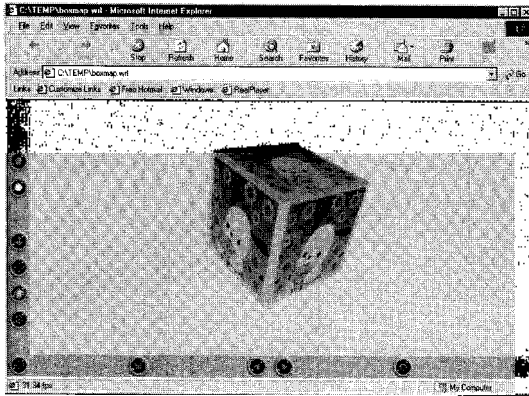


Fig. 5. Texture Mapping Model.

GIF format file images to any model surfaces. In the **PixelTexture** node, you can specify the individual pixel colors to make an image and you can apply these images to faces or shapes.

#### 6.4. Selection of methods to store color information

There are a number of methods to store the color information of an RP model using VRML. Among them **Color** node has the greatest capability to store the color information. For color information like single color on each surface, multiple color on each surface or for complex color structure like gradual change in color, we can use this function to store such information. For color format like bitmapped graphics, we can use **ImageTexture** node to store them.

## 7. Discussion

One of the advantages of VRML files is that they contain less redundant data when compared with STL files. In VRML files, it is not necessary to include the normal vectors as the vertices of all faces are taken in only counter-clockwise direction. STL files may use the same X, Y and Z coordinates many times over while VRML files used the coordinate index instead of the coordinates, which can save a lot of space. VRML files are very convenient in storing the color information of a model.

For RP processes, when a part is designed using a CAD system, the geometric information of the model is gathered in the data interface like an STL file. This file contains a series of simple polygons to represent the surface of the model. The file is then transferred to RP machines where there will be a conversion of this faceted information to a number of slices showing the contour of the model, layer by layer (Fig. 6). Cross-sections of the part are built according to these slice layers until the whole physical shape is made.

There are many existing techniques for performing the slicing process. Chalasani used a simple geometric algorithm to slice the STL file [22]. At the beginning

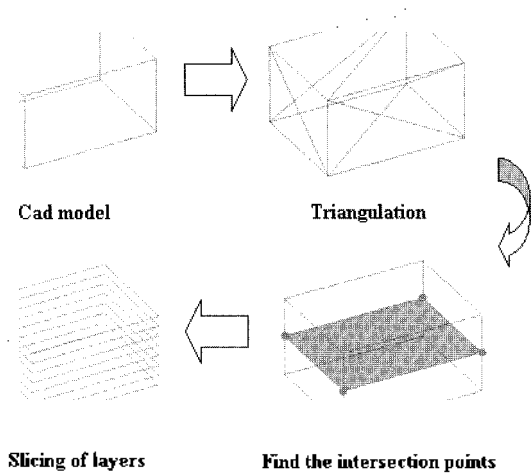


Fig. 6. Slicing Process.

the STL file is read in. A multiplier is used on the vertex coordinates of all the triangles (usually a large number - approximately 5000) to convert them into integers so that the computer can process at a faster time. All the flat triangles are removed and they are used to fill in the top and bottom of the part later. For each z-layer, there is a verification of the intersection line between the plane and the triangle. After finding all intersection lines on the same plane, they will be sorted in a head-to-tail fashion to make up the contour. This contour may be hatched to a specific pattern for further applications.

For color RP layer processing, there should be two types of slice files. The first type includes the contour of the model by layers, while the second one is used to gather the colors of the layer contour. The algorithm is shown in Fig. 7.

Similar to the STL file, after a VRML is read in, the vertex coordinates of all the polygons is multiplied to integers for easier calculation. The flat faces are again taken out of the slicing process and they will be placed back at the end of the process to cover the top and bottom gap of the part. After finding and sorting the intersection lines within a plane, the slicing process is divided into two parts. Besides taking the geometric information of the contour in each layer to the geometry slice files, the color information of each point that is beyond the intersection lines is taken and stored in the color slice files. The point color depends on the color method implemented on the model's faces. If the **IndexPerVertex** value is false, the point color is equal to the color of the face that includes the point. If the **IndexPerVertex** value is true, the point color is equal to the linear interpolation of colors of vertices that make up the face. If the color method is texture mapping, the point color will be calculated from a mapping function from the texture image to the face. If at a certain Z position of the model there is a flat surface, the color on the flat surface will then be placed back to the slice

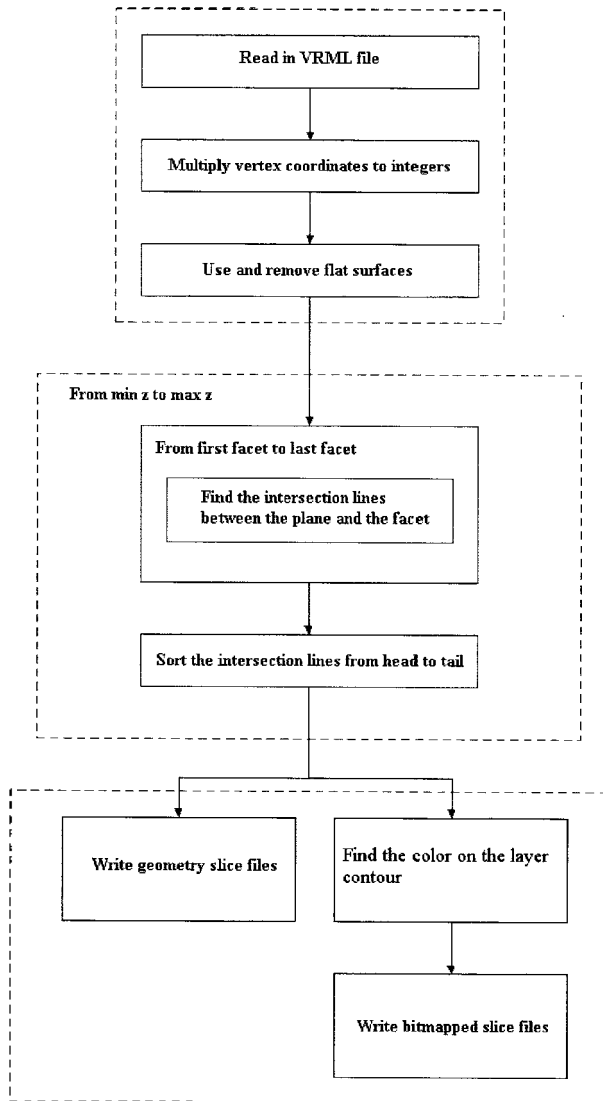


Fig. 7. Slicing algorithm of coloured model.

file at that position.

The above algorithm is suitable for RP process that has hybrid mechanisms. Take for example, in Kira

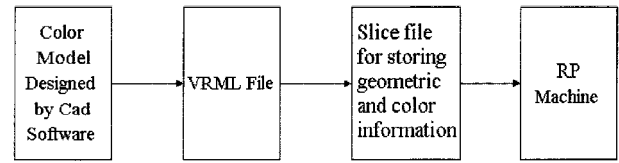


Fig. 9. Flowchart for storing color and geometric information of model by VRML only.

process, it makes use of carbon ink to bind the paper together and also outline the layer contour of the model, while it has the cutting knife for giving the shape of the model. Slicing in his case will be divided into two formats to drive the two mechanisms separately (Fig. 8). For the color 3D printer, however, it just takes advantages of color inks to give up the shape and the color to the model simultaneously. The whole process is a single mechanism. The model in this case, is just needed to be sliced into one format which contains both the geometric and color information of the contour layer (Fig. 9)

As discussed in the previous section, in VRML there are basically three methods to represent the model's color. The color binding method is mainly suitable for models in which the faces are different from each other. If it is necessary to add a bitmapped graphics texture to the surfaces, the texture mapping method should be used. An advantage of VRML is that it can be used to store more than one model in a single file, which is different from STL in which only one solid model's information can be stored. For models of different colors, it is possible to use the method of adding a suitable value in the **emissiveColor** field in the **Material** node. This could be quite useful in distinguishing different parts in an assembly (Fig. 10).

There are many problems during coloring the layer contour in the model. Assume the method used is similar to the 3DP process, where colored liquid ink is the transfer median and it will be added on to the slice layer in each time. As it is necessary to ensure that the liquid median is pasted to the desired position on the

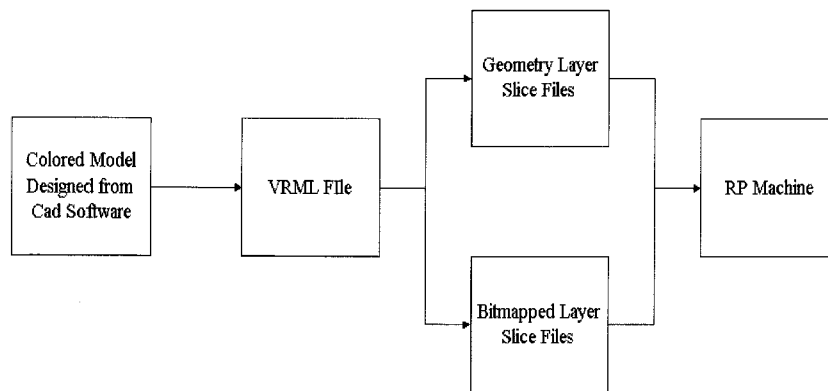


Fig. 8. Flowchart of storing model's color and geometric information using two slice formats.

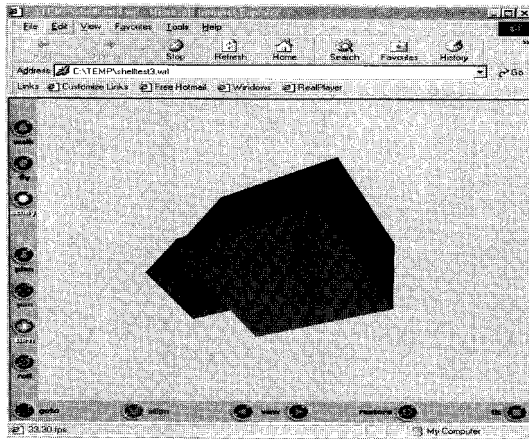


Fig. 10. Assembly model.

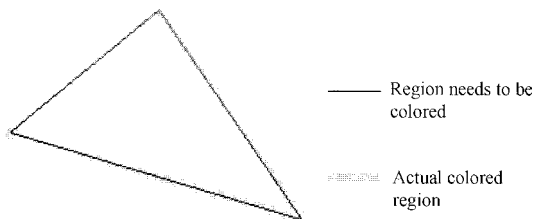


Fig. 11. Actual colored region on layer contour.

slice layer, you need to do an offset of liquid on the layer, i.e. to increase the width of liquid path. This approach will inevitably cause contamination of some regions that do not form parts of the component (Fig. 11).

As the width of the paths is increased, it may cause a blending of colors and form a undesired color on the intersection between them (Fig. 12a). In this example, two faces of an object have different colors and intersect at a common vertex. This vertex becomes a point where blending occurs. This phenomenon becomes more severe in regions that have sharp corners (Fig. 12b), and between the outer and inner contours, e.g. for thin-walled components (Fig. 12c). These can be considered as special cases, where different settings are needed to ensure good quality results from a color RP system. Therefore care should be taken when determining the path offset.

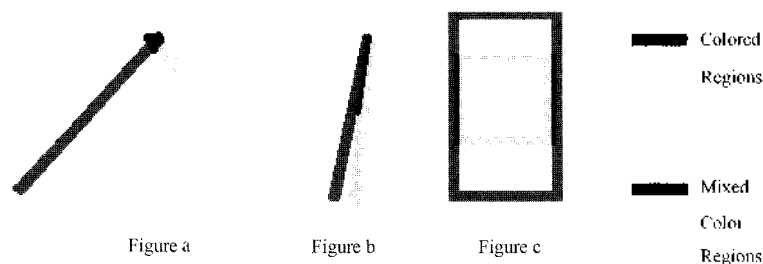


Fig. 12. Mixed colors at a) Intersection of paths b) Sharp corner c) between outer and inner layer contours.

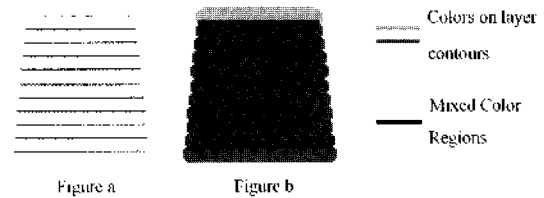


Fig. 13. a) Little penetration b) Too much penetration.

Besides the problems on the individual slice layer, there are also some problems between the slice layers themselves. Penetration of transfer medium is an important factor in controlling the quality of color on the model. Low penetration will make a gap between colors while too much penetration will mix the colors together (Fig. 13a & b). In this example, the top and bottom of the object are different colour to the side.

### 8. Conclusion

VRML is a suitable data interface for use in color RP systems. It can store the geometric information of the model in a similar way to the *de facto* industry standard STL. On the other hand, VRML stores less redundant data than STL, and can store two or more parts simultaneously in a single file. More importantly, VRML can be used to effectively store the color information of the parts. VRML has already been used in a limited sense in the Z Corporation's color 3D Printing system. There are basically three coloring methods and each can be applied to a specific coloring condition. Since VRML contains all the features of STL, whilst STL does not have VRML's advantages as far as color RP is concerned, it is considered that VRML may replace STL as the *de facto* standard for file transfer in the RP market. Once color parts are defined, slicing parts for RP becomes a matter of geometric segmentation accompanied by color offsetting. Because of potential problems with calibration of these separate stages of color RP, and because of the need to consider ink penetration through a layer, the offsetting process requires special consideration. This consideration includes attention to color blending or merging where different color features meet.

## 9. Acknowledgement

We thank the Department of Mechanical Engineering, the University of Hong Kong and the financial support from the Hong Kong Research Grant Council.

## References

- [1] Chua C. K. and Leong K. F. (1997), *Rapid Prototyping: Principles and Applications in Manufacturing*, John Wiley & Sons, inc.
- [2] Reed, K., et. al. (1990), *Initial Graphics Exchange Specification (IGES) version 5.0*, CAD-CAM Data Exchange Technical Centre.
- [3] Li, J. (1992), "Improving stereolithography parts quality-Practical solutions," *Proceedings of the Third International Conference on Rapid Prototyping*, 171-179.
- [4] Jamieson, R. and Hacker H. (1995), "Direct Slicing of CAD Models for Rapid Prototyping," *Rapid Prototyping Journal*, 1(2), 4-12.
- [5] Swaelens B. and Kruth J.P. (1993), "Medical applications of rapid prototyping techniques," *Proceedings of the Fourth International Conference on Rapid Prototyping*, 107-120.
- [6] Swaelens B. and Kilburn P. (1999), "STL Based CAD Work: Do it with Triangles," *Proceedings of the 8<sup>th</sup> European Conference on Rapid Prototyping and Manufacturing*, 21-28, July 6<sup>th</sup> - 8<sup>th</sup>.
- [7] Jepson L., et. al. (1999), "Initial Developing of a Multi-Material Selective Laser Sintering Process (M<sup>2</sup>SLS)," *Proceedings of the 8<sup>th</sup> European Conference on Rapid Prototyping*, 367-384.
- [8] Griffith M. L., et. al. (1997), "Multi-Material processing by LENS," *Solid Freeform Fabrication Proceedings*, 387-394.
- [9] Rees M. (1998), "Color," *Prototyping Technology International*, April-June issue, 36-40.
- [10] Wohlers T., *Wohlers Report 2000*: pp. 87-88.
- [11] Gibson I. and Ling W. M. (1999), "Incorporation Colour and Rapid Prototype Models," *Proceedings of the 8<sup>th</sup> European Conference on Rapid Prototyping and Manufacturing*, 385-394, July 6<sup>th</sup> - 8<sup>th</sup>.
- [12] Gibson I., et. al. (2000), "Multiple material systems based on the SLS process," *8<sup>th</sup> European Conference on Rapid Prototyping*, 3&4 May.
- [13] Gilman C. R. and Stephen J. R. (1995), "The Use of STEP to Integrate Design and Solid Freeform Fabrication," *Solid Freeform Fabrication Proceedings*, 213-220.
- [14] Haescong J. L. and Lee B. Y. (1999), "Slicing STEP-based CAD Models for CAD/FP Interface," *Solid Freeform Fabrication Proceedings*, 171-178.
- [15] <http://tcfa.unige.ch/guides/vrml/sig-graph-tutorial/notes/preface.htm>
- [16] Ames A. L., et. al., *VRML 2.0 Sourcebook: Second Edition*, 1997.
- [17] <http://www.web3d.org/vrml/browpi.htm>
- [18] Fadel G. M. and Kirschman C. (1996), "Accuracy issues in CAD to RP translations," *Rapid Prototyping Journal*, 2(2), 4-17.
- [19] Yanshuo W., et. al. (1997), "The Use of VRML to Integrate Design," *Solid Freeform Fabrication Proceedings*, 669-676.
- [20] <http://www.vrml.org/Specifications/VRML97/>
- [21] <http://www.parallelgraphics.com/products/cortona/>
- [22] Chalasani K. L., et. al., "Algorithm to Slice 3D Shapes for Reconstruction in Prototyping Systems," *Proceedings of 1991 ASME Computers in Engineering Conference*, pp. 209-216, August 1991.
- [23] Zollinger H., *Color Chemistry: Synthesis, Properties and Applications of Organic Dyes, and Pigments- Second, revised edition*, VCH Verlagsgesellschaft mbH, D-6940 Weinheim (Federal Republic of Germany), 1991.

## [Appendix]

```
#VRML V2.0 utf8
# The VRML 97 example
# 1-9-2000
# Designed by Ling Wai Ming
Group {
  children [
    Shape {
      appearance Appearance {
        material Material { }
      }
      geometry IndexedFaceSet {
        coord Coordinate {
          point [
            -2.50 2.50 2.50,
            2.50 2.50 2.50,
            2.50 2.50 -2.50,
            -2.50 2.50 -2.50,
            -2.50 -2.50 2.50,
            2.50 -2.50 2.50,
            2.50 -2.50 -2.50,
            -2.50 -2.50 -2.50,
          ]
        }
        coordIndex [
          0,1,2,-1,
          0,2,3,-1,
          4,6,5,-1,
          4,7,6,-1,
          3,2,6,-1,
          3,6,7,-1,
          0,5,1,-1,
          0,4,5,-1,
          3,4,0,-1,
          3,7,4,-1,
          1,6,2,-1,
          1,5,6,-1
        ]
      }
      colorPerVertex FALSE
      color Color {
        color [
          0.0 0.0 1.0,
          0.0 1.0 0.0,
          1.0 0.0 0.0,
          1.0 1.0 0.0,
          0.0 1.0 1.0,
        ]
      }
    }
  ]
}
```



```
1.0 0.0 1.0
]
}
colorIndex [
0,0,
1,1,
2,2,
3,3,
4,4,
5,5
]
}
}
]
}
```

[VRML color binding example]

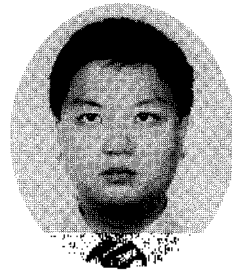
---

**Ling Wai Ming** received a BEng in 1997, and a PhD in 2002 from the Department of Mechanical Engineering, the University of Hong Kong. His research interests include the VRML, color Rapid Prototyping and Selective Laser Sintering (SLS).

---

**Ian Gibson** received a BSc in 1984, and a PhD in 1988 from the Department of Electronic Engineering, the University of Hull, UK. He is currently Associate Professor at The University of Hong Kong, Department of Mechanical Engineering, where he specializes in Advanced Manufacturing Technology, Rapid Prototyping, CAD/CAM and Product Development.

---



Ling Wai Ming



Ian Gibson