

Object Cloning

Naga Arjun.V¹, Dileepan chakkaravarthi.A²

Department of Engineering Design, Anna University, Chennai, India^{1,2}

ABSTRACT: This project “*OBJECT CLONING*” mainly focuses in cloning any symmetrical object from any remote area. First, software is created which is meant to generate CNC codes for lathe automatically and it sends the generated codes to any part of the world with the help of internet in a fraction of second, this also can be said as live object transfer.

The software has a scanning window which is used to scan the object and to find the dimensions of the object. The object is captured using a video camera and stored as an image. This image is then scanned for its dimensions using image processing technique with the help of scanning software.

These dimensions are used to write the special standardised codes meant for the *CNC Lathe*, the *G & M codes*. Hence the codes to make the object are generated. The generated codes are transferred through internet to any part of the world in a fraction of seconds. The specialty of this project is, the scanning, the automatic generation of CNC codes, sending the generated codes to any remote area all these things happens in a fraction of seconds.

KEYWORDS: Image processing, scanning window, object transfer

I. INTRODUCTION

E-mail (electronic-mail) may be defined as sending mailing through internet; transferring money through internet is called e-commerce; buying a product through internet is called e-buy, like wise controlling machines with the help of internet is called e-control.

In our project we are taking the advantage of using “e-control” to clone a symmetrical object in any remote area. High speed of code generation and the live object transfer are the next step development of automation. This project involves in developing software called “scanning window” for scanning the object to get its exact dimensions and to write CNC G and M codes automatically. The software is written in “visual basic” language

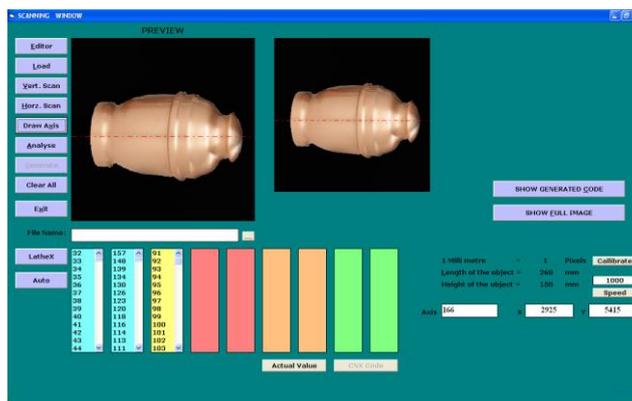
II. CONSTRUCTION

The object is captured as an image with the help of a digital camera, the distance between the object and the camera is fixed by calibrating the image to get its original size. The image processing is done with a help of a computer and the scanning of the image is done by creating software called scanning window. The scanning consists of vertical scanning and horizontal scanning to get the dimensions of the object. The software also contains a code generator which generates the CNC codes automatically with the help of the scanned dimensions of the object. Then the generated codes are sent to the desired place with the help of internet by using client/server or master/slave technique.

III. WORKING PRINCIPLE

The object which is to be cloned is kept in a black background with perfect lightings. A high resolution digital camera is used to capture the image of the object; the resolution of camera will determine the quality of the image. The captured image is stored in a computer for image processing and scanning. The image is then scanned to get the dimensions of the object, scanning is done by creating software called scanning window which is written in visual basic language. The scanning consists of vertical scanning to determine the length of the object and horizontal scanning to calculate the height of the object. The method of determining the height and length of the object is clearly explained in

the design calculations. The principle of scanning is by identifying the color of the pixels. The scanning gives the dimensions and coordinates of the object which is used in writing the CNC codes for CNC lathe automatically. The generated codes are sent to the desired location with the help of internet especially with client/server technique. The RGB color module and client/ server module is given below.



M CODES

GROUP	FUNCTIONS
M00	Program stop
M01	Optional stop
M02	End of program
M03	Spindle on (clockwise)
M04	Spindle off (counter clock wise)
M05	Spindle off
M06	Turret index
M08	Coolant on
M09	Coolant off
M30	End of program and rewind
M98	Subprogram
M99	End of subprogram

RGB color model

In the RGB model, each color appears in its primary spectral components of red, green, and blue. This model is based on a Cartesian coordinate system. The color subspace of interest is the cube in which RGB values are at three corners; cyan, magentas, and yellow are at three other corners; black is at the origin; and white is at the corner farthest from the origin. In this model, the gray scale (points of equal RGB values) extends from black to white along the line joining these two points. The different colors in this model are points on or inside the cube, and are defined by vectors extending from the origin. For convenience, the assumption is that all color values have been normalized so that the cube is the unit cube. That is, all values of R, G, B are assumed to be in the range [0,1].

Images represented in the RGB color model consist of three component images, one for each primary color. When fed into an RGB monitor, these three images combine on the phosphor screen to produce a composite color image.

The number of bits used to represent each pixel in RGB color pixel [that is, a triplet of values (R, G, B)] is said to have a depth of 24 bits (3 image planes times the number of bits per plane). The term full-color image is used often to denote a 24 bit RGB color image. The total number of colors in a 24-bit RGB image is $(2^8)^3 = 16,777,216$. The 24-bit RGB color cube corresponding to the diagram.

The cube shown in fig is a solid, composed of the $(2^8)^3=16,777,216$ colors mentioned in the preceding paragraph. A convenient way to view these colors is to generate color planes (faces or cross sections of the cube). This is accomplished simply by fixing one of the three colors and allowing the other two parallel to the GB-plane is the plane (127, G, B) for G, B=0, 1, 2,,255. Here we used the actual pixel values rather than the mathematically convenient normalized values in the range [0,1] because the former values are the ones actually used in a computer to generate colors. Figure shows tat an image of the cross sectional plane is viewed simply by feeding the three individual component images into a color monitor. In the component images, 0 represents black and 255 represents white (note that these are gray scale images). Finally, figure shows the three hidden surface planes of the cube in figure generated in the same manner.

It is of interest to note that acquiring a color image is basically the process shown in figure in reverse. A color image can be acquired by using three filters, sensitive to red, green, and blue, respectively. When we view a color scene with a monochrome camera equipped with one of these filters, the result is a monochrome image whose intensity is proportional to response of that filter.

Repeating this process with each filter produces three monochrome images that are the RGB component images of the color scene. (In practice, RGB color image sensors usually integrate this process into a single device.) Clearly, displaying these three RGB component images in the form shown in figure would yield an RGB color rendition of the original color scene.

While high-end display cards and monitors provide a reasonable rendition of the color in a 24-bit RGB image, many systems in use today are limited to 256 colors. Also, there are numerous applications in which it simply makes no sense to use more than a few hundred, and sometimes fewer, colors. Given the variety of systems in current use, it is of considerable in-son ably independently of viewer hardware capabilities. This subset of colors is called the set of safe RGB colors, or the set of all-systems-safe colors. In internet applications, they are called safe Web colors or safe browser colors.

On the assumption that 256 colors is the minimum number of colors that can be reproduced faithfully by any system in which a desired result is likely to be displayed, it is useful to have an accepted standard notation to refer to these colors. Forty of these 256 colors are known to be processed differently by various operating systems, leaving only 216 colors that are common to most systems.

Client-Server Model

A server is a computer program that offers a service over a network. A server accepts an incoming request, forms a response, and returns the result to the requester. For the simplest services, each request arrives in a single datagram and the server returns a response in another datagram.

An executing program becomes a client when it sends a request to a server and waits for a response. Because the *client-server* model is a convenient and natural extension of inter process communication used on a single machine, it is easy for programmers to build programs that use the model to interact.

Servers can perform simple or complex tasks. For example, a time-of-day server merely returns the current time whenever a client sends the server a packet. A web server receives requests from a browser to fetch a copy of a Web page; the server returns the requested page to the browser.

Usually, servers are implemented as application programs. The advantage of implementing servers as application programs is that they can execute on any computing system that supports TCP/IP communication. For example, if the load on a server increases, the server can be moved to a faster CPU. Technologies exist that allow a server to be replicated on multiple physical computers to increase reliability or performance.

IV. FUTURE DEVELOPMENT

Complicated components can also be cloned by developing the software to scan three-dimensionally. The software can further be developed to clone any asymmetrical objects. Objects with threads can be cloned and transferred by measuring the pitch and number of threading.

REFERENCES

- [1] "Cnc Programming And Maintenance".....Psg Course Material
- [2] "Robotics".....Groover
- [3] "Digital Image Processing".....Rafael C. Gonzalez & Richard E. Woods
- [4] "Internet Working With Tcp/Ip".....Douglas E.