PIXEL PROCESSING IN ADVANCED TEXTILE DESIGN

FOR

Advent Electromagnetic Technologies,
Coimbatore

Project report submitted on partial fulfillment of the requirements
for the award of the degree of

Master of Computer Applications

of

Bharathiar University

by

D. HEMALATHA
(Reg. No. 0038M1026)

Under the guidance of

Mr. K. R. BASKARAN B.E., M.S.,
Department of Computer Science and Engineering

Department of Computer Science and Engineering
KUMARAGURU COLLEGE OF TECHNOLOGY
(Affiliated to Bharathiar University)
Coimbatore – 641 006

CERTIFICATE

Department of Computer Science and Engineering

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE – 641 006.

CERTIFICATE

PROJECT REPORT 2002 - 2003

Certified that this is a bonafide report of
the project work done by

D. HEMALATHA

(Reg. No. 0038M1026)

Mr. K. R. BASKARAN B.E., M.S.,	**Prof. S. THANGASAMY, Ph.D.,**
Project guide	Head of the Department,
Computer Science & Engineering	Computer Science & Engineering

Place: Coimbatore

Date: 16/4/2003

Submitted for viva-voce examination held on

16/4/2003

Internal Examiner

External Examiner 16/4/03

A E T

Beat Crime
Before
it Beats you

ADVENT ELECTROMAGNETIC TECHNOLOGIES

2, Valluvar Nagar, Peelamedu, Coimbatore - 641 004, Tamilnadu, India
Phone / Fax : 91-0422-576297, 590857 **E-mail** : ckraj@md2.vsnl.net.in

7th April 2003

TO WHOMSOEVER IT MAY CONCERN

This is to certify that Ms D.HEMALATHA of Kumaraguru College of Technology,Coimbatore has undergone her project work in our organization from Dec 2002 to March 2003. The title of the project is "Pixel Processing In Advanced Textile Design".

We are pleased to certify that she has successfully completed her project. During the period of project work, her conduct was good.

We wishes her all the very best in her endeavours.

For Advent Electromagnetic Technologies

C.Kanakaraj
Chief Executive

DECLARATION

DECLARATION

I, hereby declare that the project entitled **'PIXEL PROCESSING IN ADVANCED TEXTILE DESIGN'** submitted to Bharathiar University as the project work of Master of Computer Applications degree is a record of original work done by me under the supervision and guidance of Mr.Shashidaran Nair Project Manager, Advent Electromagnetic Technologies,Coimbatore and Mr.K.R.Baskaran,B.E.,M.S,Kumaraguru College of Technology,Coimbatore and this project work has not found the basis of award of any Degree/Diploma/Fellowship/Associateship or similar title to any candidate of any University.

Place: Coimbatore

Date: 7-4-2003

D.Hemalatha

Signature of Guide

Mr.K.R.Baskaran,B.E.,M.S,
Assistant Professor,
Kumaraguru College of Technology,
Coimbatore-641006.

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

I am bound to express my gratitude to **Dr.K.K.Padmanaban, B.Sc (Engg)**, M.Tech., Ph.D, Principal, Kumaraguru College of Technology, Coimbatore, for constant encouragement throughout my course.

I wish to thank **Prof.Dr.S.Thangasamy, B.E (Hons), Ph.D**, Head, Department of Computer Science and Engineering, for supporting me to pursue new goals and ideas.

I admit my heartfelt thanks to my course coordinator, **Mr.A.MuthuKumar,Ph.D**, Assistant Professor, for being supportive during the tenure of my project.

I express my earnest gratitude to my internal project guide, **Mr.K.R.Baskaran,B.E.,M.S**, Assistant Professor, for his creative support and timely advice in completing this project.

I extend my sincere thanks to **Mr.C.Kanakaraj,M.B.A**, Managing Director, Advent Electromagnetic Technologies,Coimbatore, for providing me this opportunity and encouraging me throughout the project.

I am greatly indebted to **Mr.K.ChandraMohan,M.E**, Manager, South India Textile Research Association, for his technical support and guidance.

I owe much to **Mr.C.Sashidharan Nair,M.C.A**, Project Manager, Advent Electromagnetic Technologies,for his immense support in completing this project.

Last but not the least, I extend my heartfelt thanks to all my well wishers for their cooperation in my endeavor.

SYNOPSIS

SYNOPSIS

The project entitled "**Pixel Processing in Advanced Textile Design**" comprises of five modules, namely

1. Image Acquisition.
2. Array Extraction.
3. Grid Generation.
4. Mapping pixel data onto grid.
5. Image Fabrication.

Image Acquisition module involves obtaining the image details such as the resolution of the image file and the pixel data. The image file should contain the .bmp format. The resultant file is stored as a data file.

Array Extraction module involves the extraction of pixel values from the data file. The values are indexed on the basis of color value of the pixel. The indexed value is stored in an integer array.

In the Grid Generation module, Graph sheet like grid is spawned. The generated grid is restricted to a specified size.

Mapping pixel data onto grid involves the mapping of array data onto the grid precisely which facilitates the enlarged portion of the image. The mapped grid is saved as a file with the extension of .tiff.

The final module called Image Fabrication sends the enlarged images to the printer. The outcome is a monochrome image which is the final output for the fabric design process.

CONTENTS

CONTENTS

INTRODUCTION

1. INTRODUCTION

1.1 ABOUT THE ORGANIZATION

Advent Electromagnetic Technologies, a software development cum training center witnessed its humble beginning a way back fifteen years. Advent Electromagnetic Technologies believes in a world of quick changing frontiers to be a successful enterprise. They are committed to reach out their clients with IT products and solutions specific to their business needs.

Advent Electromagnetic Technologies use their expertise and experience to help clients to anticipate, initiate and manage changes better than others. This commitment and supportive relationship has helped clients to win.

Advent Electromagnetic Technologies believes in recognizing and encouragement the merit of its people as individuals. The environment thus facilitates freethinking, experimentation and innovation. With an emphasis on personal responsibility, Advent Electromagnetic Technologies provides an atmosphere that supports original thinking.

The most striking feature of this growing organization is the commitment of the staffs and project guides, who are always trying to give the best of their knowledge to the students.

1.2 INTRODUCTION TO THE SYSTEM

The project **"PIXEL PROCESSING IN ADVANCED TEXTILE DESIGN"** empowers a dynamic change in the field of Textile which would enrich the enhancement of fabric design. It assures easier development of designs in a easier, accurate and exquisite models. The manual process is obviously a time consuming and a hectic process to the user. The project is developed in such a manner that reduces the existing textile design work which is manual at present, to a maximum extent and produces the result more precise and concise.

The project minimizes the work of the user by acquiring the image at ease. And also, it drastically reduces the work in each and every move of the project, such as getting the pixels, enlarging them, mapping them, and sending them to the printer. Most of the manual process which is been done now are transferred to internal memory processing and computerized automatically. The workload of the user is lessened and design could be achieved efficiently.

The project design is sketched out into modules namely, Image acquisition, Augmentation and Fabrication. Each and every module is developed such that it could be efficient, if integrated as a whole or as an individual performance.

Being an excellent system side development tool, C was used in order to deliver improved support for reuse, tool interoperability and resource management.

1.2.1 Introduction to Image Processing

Digital Image Representation

Monochrome images refers to a two-dimensional function like f(x, y) where x and y denote spatial coordinates and the value of f at any point (x, y) gives the brightness at that point. A digital image is an image f(x, y) that has been discretized both in spatial coordinates and brightness. A digital image can be considered a matrix whose row and column indices identify a point in the image and the corresponding matrix element value identifies the gray level at that point. The elements are called as picture elements or pixels.

Steps in Image Processing

Image Acquisition

Image acquisition is a process to acquire the image. It requires an imaging sensor and the capability to digitize the signal produced by the sensor. If the output is not already in digital form, an analog to digital converter digitizes it. The formats of the image to be acquired are determined by the application.

Preprocessing

The key function of preprocessing is to improve the image in ways that increase the chances for success of the subsequent operations to be performed on the image. It typically deals with the techniques for enhancing, contrast, removing noise, isolating regions etc.

Segmentation

The key role of this function is to partition the input image. In character recognition, the role of segmentation is to extract characters from the background.

Recognition and Interpretation

Recognition is the process that assigns a label to an object based on the information provided by its descriptors. Interpretation involves assigning meaning to an ensemble of recognized objects.

Pixels

Pixels are the basic building blocks of all bitmap images. These are individual elements combined together to form a picture. It is a smallest individual unit of the image. The term pixel represents the dots on the computer screen or the dots a printer can produce and the individual elements of a bitmap graphics.

Image Aspect Ratio

Image aspect ratio refers to the number of horizontal and vertical pixels that go into making up an image. Any .bmp file will have a certain number of pixels in the horizontal and vertical dimensions. This ratio is often referred to as the size of the image and is usually written in the form 800x600.

Bit Depth

A bit is smallest unit of memory used by the computer and it can have a value on (or) off. In a bitmap graphic, the color of each pixel is recorded by the computer using bits. The more bits the computer uses more colors we get. The number of bits the computer uses for one pixel is the bit depth.

1.3 SOFTWARE OBJECTIVE

The software developed includes the acquisition of the image, enlarging it to the specified size, save as separate image files and printing it which supports the development of discriminating fabric design.

1.4 EXISTING SYSTEM

1.4.1 Initial study

The .bmp image which is got as input is to be enlarged as such. For this purpose, the image file is retrieved. Based on the image's width and resolution of the printer, the image is divided manually into separate segments. The grid generation is also done manually where the segmented image has to be mapped, which is a tedious and a time consuming process. The file name for each mapped grid is entered by the user while saving. The mapped grid is saved in .bmp format which consumes more memory.

Problem Definition

- Manual processing of pixels sounds hectic.
- Requires more time.
- Lack of accuracy
- Possibility of errors is superfluous.
- Damage of the file will affect the output.

1.4.2 Detailed Study

Detailed evaluation is done on the present system and data collection based on some key questions such as,

What must be done to solve the problem?
What are the facts?

Logical model of the system is developed and flow diagram is designed.

1.5 PROPOSED SYSTEM

1.5.1 Feasibility Study

Depending on the results of the initial investigation, the survey is expanded to a more detailed feasibility study. During the study, the problem definition is crystallized and aspects of the problem to be included in the system are determined.

Statement of the problem

Manual image segmentation is tedious and time consuming. Probability of errors is possibly very high. Storage of large number of bitmap files of high capacity may drag down the machine efficiency.

Conclusion

The survey concludes that the system is feasible and it meets all the user requirements.

1.6 SOFTWARE REQUIREMENT ANALYSIS

Selecting a system begins with requirement analysis. The software selection focuses on reliability, functionality, flexibility and other criteria. Software selection is a critical aspect of system development

Criteria for software selection

Reliability

It is a probability that the software will execute for a specified time period without a failure. It relates to ease of enhancement and ability to consistent results. Software reliability brings up the concept of modularity. The high degree of modularity has the capacity to operate in many machine configurations.

Functionality

Functionality is a definition of the performance, efficiency and the other factors that the user requires in the finished product.

Capacity

Capacity refers to the capability of software package to handle the user's requirements for different image resolutions.

Flexibility

It is a measure of effort required to modify an operational program. One feature of flexibility is adaptability, which is a measure of the ease of extending the product.

PROGRAMMING ENVIRONMENT

2. PROGRAMMING ENVIRONMENT

2.1. SOFTWARE REQUIREMENTS

Language	: C, MATLAB
Operating System	: Windows 98, DOS

2.2. HARDWARE REQUIREMENTS

Processor	: Pentium III
CPU Speed	: 799 MHz
RAM	: 256 MB
CDROM	: 52X
Hard Disk	: 20 GB
Keyboard	: 101 Keys

2.3. ADDITIONAL REQUIREMENTS

Printer	: Line Printer (4"x5") 300 dpi

SYSTEM DESIGN

3. SYSTEM DESIGN

3.1 INPUT DESIGN

The goal of the input design is to make the data entry easier, logical and free from errors.

Initially, to enter various data required for the system, an input screen is introduced .The home screen is available to the user in the form of a menu which instructs him to enable the services required. (Figure 1)

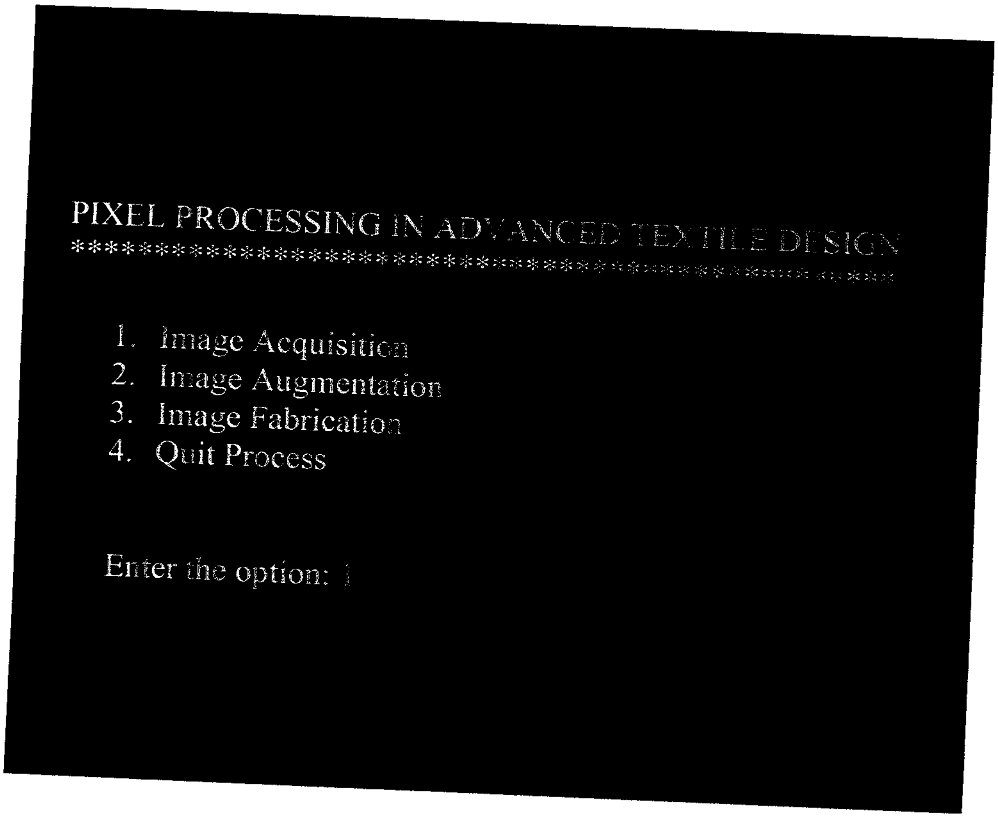
The input screen displays a menu as said with four options, namely

1. Image Acquisition
2. Image Augmentation
3. Image Fabrication
4. Quit process.

The user is provided with an option of selecting an appropriate service from the list values.

On defining the menu, if user selects the option 1, the acquisition of image is done with the input of image name .If the user selects the option 2, the image is shown enlarged on a specified scale. If the user selects the option 3, the image file will be printed. On selecting option 4, the control is returned to the main IDE.

If user selects option 2 or 3 without acquiring the image, an error condition fires which instructs the user to input the name of the image file.



```
PIXEL PROCESSING IN ADVANCED TEXTILE DESIGN
**************************************************

    1.  Image Acquisition
    2.  Image Augmentation
    3.  Image Fabrication
    4.  Quit Process


    Enter the option: 1
```

Figure 1

1. Image Acquisition

In the Image Acquisition process, the image file name in the .bmp format is given as input. The file should reside in its home directory. If the file is present in any other hard drive, the path name should be specified along with the file name. The width, height and pixel data of the image is read from the image header. The data is stored in a binary file.

2. Image Augmentation

A Grid is generated. The image file is enlarged to a specified scale on the grid. The pixel data of the image is mapped on to the grid. Then the corresponding image is saved as a .tiff file. The process is repeated till every single one of the pixel is mapped. The corresponding enlarged .tiff files are displayed for the user.

3. Image Fabrication

Number of .tiff files is generated from a .bmp files are kept in account. Then all the files are sent to the printer from the first grid image generated to the last grid belonging to the same .bmp file. Each individual grid image is identified by a unique filename. After all the files have been printed the fabrication process is terminated.

3.2 OUTPUT DESIGN

The user is constantly provided with control messages during every event that is invoked. Outputs generated by the system during the process are promptly displayed to the end users.

The acquired image is placed on the grid which displays the enlarged portion of the image. Based on the resolution of the printer, the enlarged images are sent to the printer. The processed image is a monochrome image with two colors ‘Black and White’. Different symbols are used to identify the colors.

SYSTEM DEVELOPMENT AND IMPLEMENTATION

4. SYSTEM DEVELOPMENT AND IMPLEMENTATION

4.1 SYSTEM DEVELOPMENT

C – An Overview

'C' is a programming tool used in my system. It has high machine efficiency. The language provides a number of header files, library functions, file systems, etc .It also offers a complete, easy development environment. It is one of the best platforms for professional software developers.

In many respects, 'C' has become *lingua franca* of programmers. C language, elements are easily translated to other languages. The important reason for using C is simply one of efficiency.

Header Files

To start with a graphics on the screen, we need a header file called GRAPHICS.H and a library file called GRAPHICS.LIB. The header file contains definitions and explanations of all the functions and the constants needed. The graphics functions are kept in graphics library file. The first thing needed to do before carrying out any drawing activity is to switchover to the graphics mode. This is done by making a call to initgraph() function. This function figures out the further added header files are

Stdio.h – Standard I/O functions
Conio.h – Display property functions
Dos.h - DOS operations
Math.h-Mathematical operations
Malloc.h-Memory allocation operations
String.h-String operations

System functions

Built-in Functions

The system development includes both the user-defined and built-in functions.

initgraph() is a built-in function to switch over to the graphics mode. The project inculcates EGAVGA.BGI file as our graphics driver.

For pixel operations, we use putpixel() and getpixel() functions. These functions are used to get the pixel value and process the same.

imagesize() function is used to allocate memory for image to save in disk.

malloc() is used to allocate memory.

getimage() and putimage() are used to store and retrieve the image from memory.

closegraph() is used to unload the graphics driver.

restorecrtmode() is used to take the screen back to the mode that existed prior to the calling of the initgraph.()

I/O Functions

fopen() opens a data file.
fgetc() gets the character from a file.
fgets() gets the string from a file.
fprintf(),fwrite() writes data to file.
fputc() puts a character to a file.

fputs() puts a string to a file.

fseek() moves to the specified position of the cursor.

ftell() gives the file pointer position.

fread() read data from file.

fclose() to close a file from current execution.

User defined Functions

There are many user defined functions. Let us have a outlook of some main functions following,

gen_grid() is a function responsible for generating grid as and when it is called.

extract() is used to extract pixel data from file.

map_pixel() is used to map the image onto the grid. There are eight different sub functions defined in the main function such as,

map_bk() to map black color pixel.

map_bl()) to map blue color pixel.

map_gn() for green and so on…

display() is used to display the enlarged image.

save _image() to save a .tiff image file.

print_image () is used to print the image.

Matlab

Matlab is a software package for high-performance numerical calculations and visualization. It provides an interactive environment .It provides hundreds of built-in functions for technical computation, graphics and animation. It also provides easy extensibility with its own high-level programming language. There are several optional toolboxes. It provides a collection of functions for processing. The basic building blocks of matlab are the matrices. The fundamental data type is the array.

Several algorthims are used in the implementation of the system, namely

* Bresenham's Line Drawing Algorithm

* Halftonic Algorithm

Bresenham's Line Drawing Algorithm

This algorithm is used to draw lines by determining the pixel positions. The Bresenham's algorithm finds the closest integer coordinates to the actual line path using integer arithmetic. To draw a line, we oblige two points x1, y1 and x2, y2. For slant line the intermediary pixels on the line path is calculated.

Procedure

1. End point are to be specified.(x1,y1) and (x2,y2).
2. The left most point is selected as first point for display.
3. Calculate dx=x2-x1 and dy=y2-y1.
4. Let p1=2dy-dx , then if p1<0 ,the next point to be set as (x1+1,y1+1).
5. Increment the x coordinates by unit steps. At position (xi+1) , the coordinate to be selected, yi+1, is either yi or yi+1, depending on whether (pi<0) or (pi>=0). The calculation for each parameter p depends on the last one.

If pi<0 the form for next parameter is

pi+1 = pi + 2dy

If pi>=0 , the next parameter is

pi+1 = pi + 2(dy-dx)

If pi+1<0 , the next y coordinate to be selected is yi+1.Otherwise select (yi+1)+1

6. Repeat the procedures in step 5 until the x coordinates reaches x2.

```
         Procedure bress_line(x1,y1,x2,y2 :integer);
             Var
                  dx,dy.x,y,x_end,p,const1,const2:integer;
             begin
                  dx :=abs(x1-x2);
                  dy :=abs(y1-y2);
                  pi :=  2 * (dy-dx);
                  const1  := 2 * dy;
                  const2  := 2 * (dy-dx);
            if  x1>x2  then
            begin
                  x:=x2,y:=y2;
                  x_end:=x1;
            end;{if x1>x2}
            else
            begin
                  x:=x1;y:=y1;
                  x_end:=x2
           end;if{x1<=x2}
    set_pixel (x,y);
    while x<x_end  do begin
               x:=x+1;
               if  p<0 then p:=p+const1
    else begin
               y:= y+1;
               p:=p+const2
          end;{else begin}
          set_pixel (x,y)
          end  {while x<x_end}
end;{bress_line}
```

Halftonic Algorithm

The Halftonic Algorithm is actually meant for display purpose. The basis of halftonic algorithm is 'Error Diffusion Technique'. This algorithm calculates error levels for each and every position values. When error reaches maximum then that pixel value will be set on. If the error is not great enough leave the pixel off.

Procedure

* Input image with rows and columns.
* Define sum of the errors propagated to position due to prior assignments.
* Define the maximum error that can be generated.
* Define error distribution function.
* Set sum of errors and errors generated as zero for all rows and columns.
* Calculate the total propagated error at position due to prior assignments

Error Propagation = Sum (i=1,j=1..n(cijEg((m-i+1),(n-j+1))))

- Sum the current pixel value and the total propaget error

T=I(m,n)+Ep(m,n)

- Check error value if maximum then set pixel on.
- Calculate error generated at current location
- Calculate error generated at current location

Eg(m,n) = threshold

Dithering Technique

This dithering technique is used for printing graphics images. This technique takes in image arrays .Let us take two arrays of size 100 each. To print a row with 200 elements two arrays must be joined. The combined array is converted into raster scan array .Raster scan array is nothing but an array of characters with eight rows .Each row will have 200 elements. In which one column has eight, eight byte numbers. To print the image, each row is sent to the printer in the form of bytes for graphics printing.

4.2 SYSTEM IMPLEMENTATION

System Implementation includes the tasks where the theoretical design is turned in to a working system and then to monitor the operation of the system to ensure that it continues effectively and efficiently. It involves careful planning, investigation of the current system and it's constraints on implementation. The system is implemented only after the thorough test is done on each of its subsystem and checking for, whether it is working according to the specification.

The project **"Pixel Processing in Advanced Textile Design"** comprises of five modules,

(i) Image Acquisition and Preprocessing.
(ii) Array Extraction
(iii)Grid generation
(iv)Pixel Mapping onto grid
(v) Pixel Fabrication

4.2.1 Image Acquisition and preprocessing

In the initial stages of Image Acquisition, the name of the bitmap file is got as input. The image file should reside in the home directory where the software exists. If the software is found elsewhere in the hard drive, the program is coded such that the location of the pathname is identified and given by the user.

The header of the image is read which contains the data relating to the resolution of the image file. The resolution includes the width and height of the image. Along with that, bits per pixel and pixel values are also read from the image header. This information is stored in a data file to which a file pointer is proclaimed. Using the file pointer, the data is retrieved from the file which concludes the acquisition process.

4.2.2 Array Extraction

In the Array Extraction module, take bitsperpixel as reference, where restricted number of bits (say 8bpp) are taken at once and stored in a buffer. The buffer value is checked and indexed and the corresponding integer value is stored in an array. This procedure is continued for consecutive extraction of pixels.

In the case of color image, the index value in an array ranges from 0 to 9.Each number represent a single color.

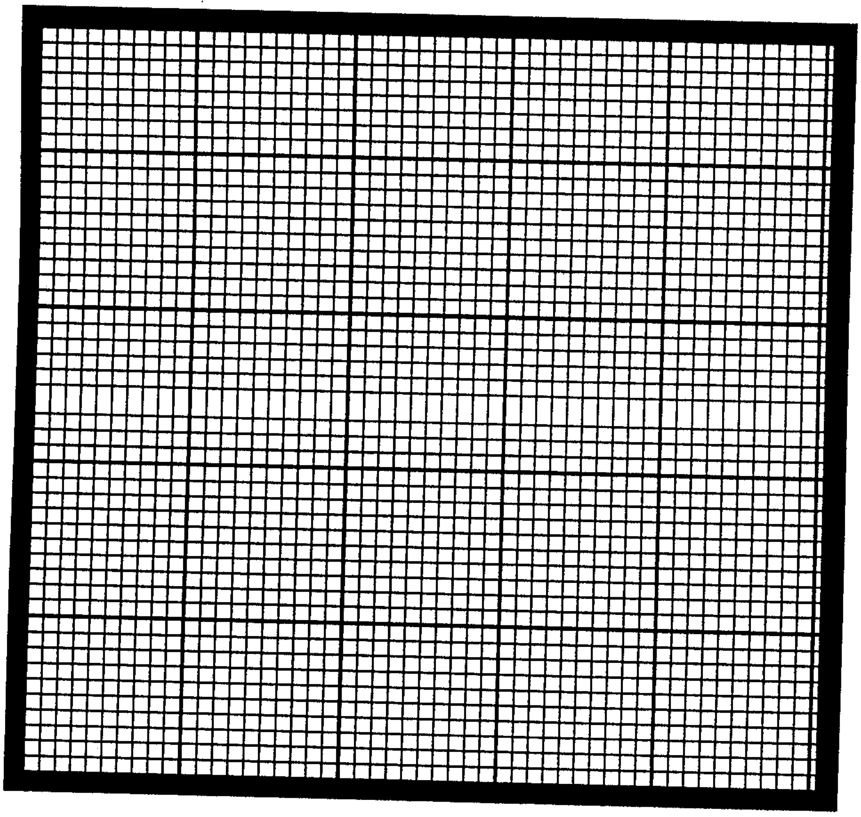
4.2.3 Grid Generation

A 2-D Integer array is of size 410x410(i,j) is declared. 'i' represents rows and 'j' represents columns .Initially all the array elements are assigned zero value in order to avoid garbage values .

Starting from the first row (say 0), the entire column values are converted to 1 till the last column. The same process is repeated for all the rows till the last row in multiples of 5(ie., i+5).After converting consecutive 10 rows, the 11th row is enabled(ie,, set with value 1).the same process is repeated till the last 410-ith row.

The same procedure is repeated for segregation of entire columns starting from first row to last row. For each and every 100 pixels, a grid is generated. The separation of grid is done automatically till the last pixel is found.

Grid figure



4.2.4 Pixel mapping onto grid

In this module, the extracted integer array and the grid array are used for pixel mapping. The indexed pixel values are checked whose values range from 0-9.The corresponding mapping functions are enabled. If the value of array element is 0, the row and column values are got as input. With that as reference, the area to be shaded is calculated.

For black color pixel, the first column and the last row in the box is left as zero and all the other pixels a re converted to 1.Different shading patterns are used for other colors.

The mapped file is saved as a .tiff image. A character array is set and the number of sub files is taken into account and it is stored in that array. Each time a grid is generated, it is stored as a sub file and file array is updated

4.2.5 Pixel Fabrication

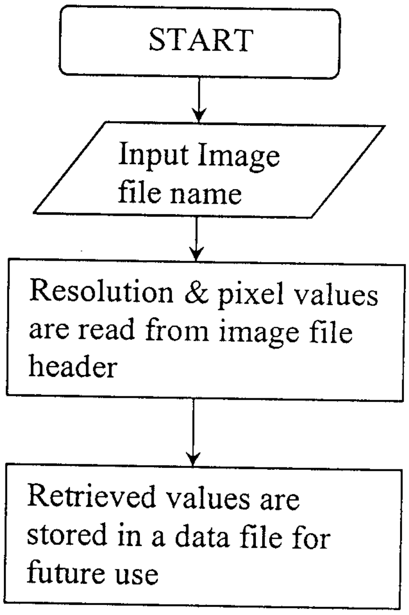
In this module, the mapped grid is stored as .tiff file. The pixel vales are stored in an array of type short. Then the image filename and array are passed to the printing function.

Initially, the printer is set to the graphics mode, and then the array values are passed to the printer file buffer. Initially the values of all the pixels present in the first row are sent to the buffer and printed, then the second row and so on.

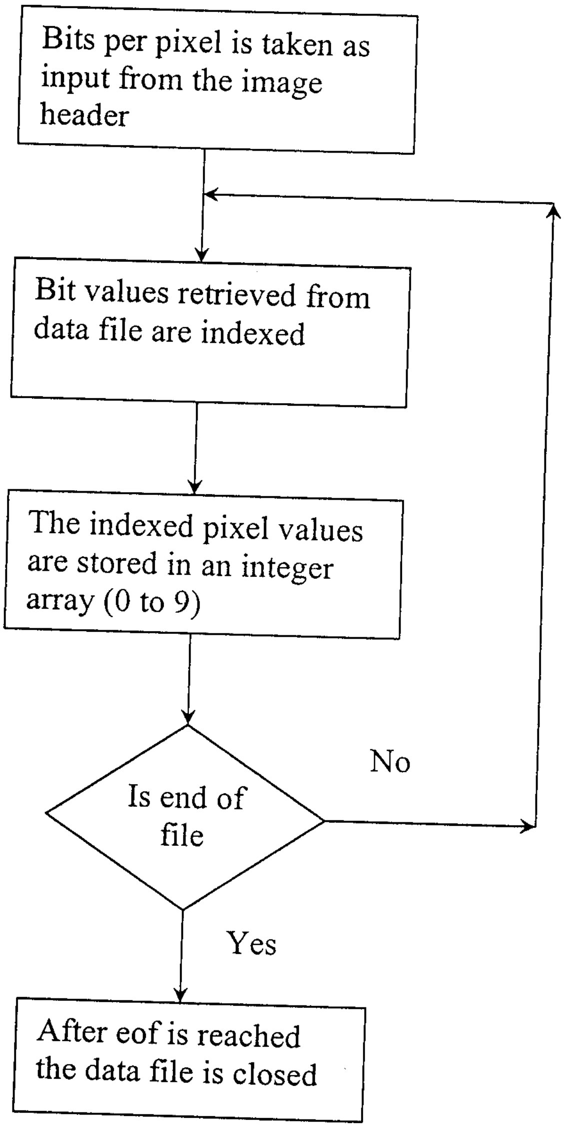
In a line printer, any number of rows can be printed continuously, but width is restricted based on the resolution of the printer. All the files .tiff files are sent to the buffer and it is printed. After printing, graphics mode is set off.

4.3 FLOW DIAGRAM

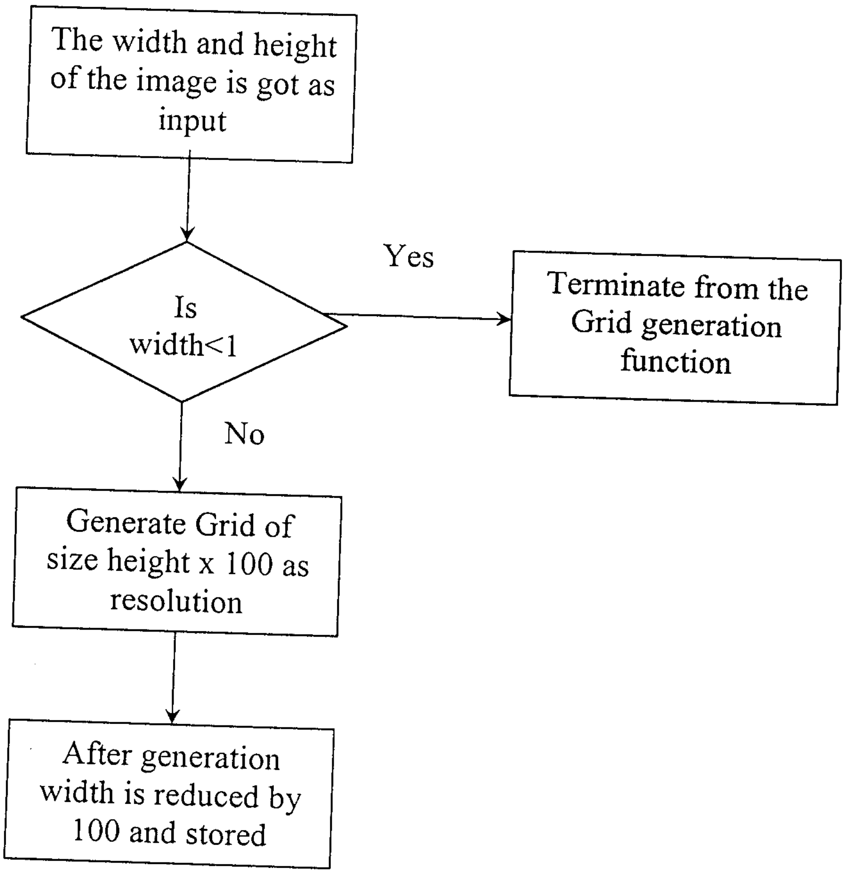
Image Acquisition


START
Input Image file name
Resolution & pixel values are read from image file header
Retrieved values are stored in a data file for future use

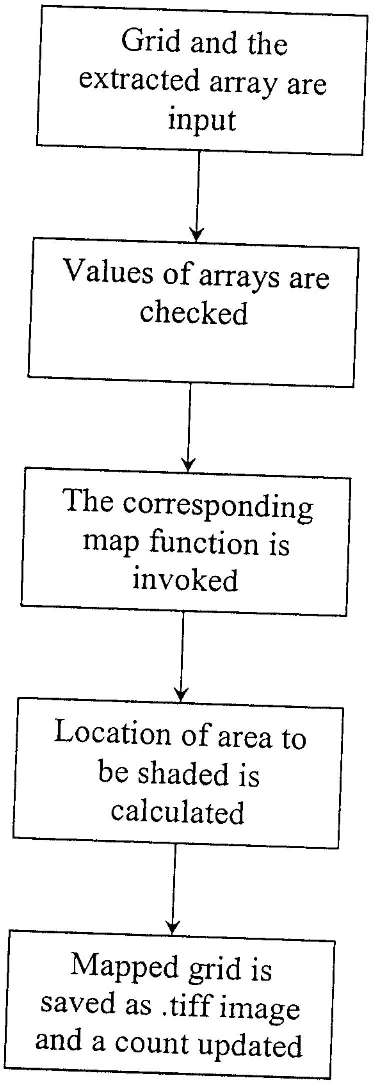
Array Extraction


Bits per pixel is taken as input from the image header
Bit values retrieved from data file are indexed
The indexed pixel values are stored in an integer array (0 to 9)
Is end of file
No
Yes
After eof is reached the data file is closed

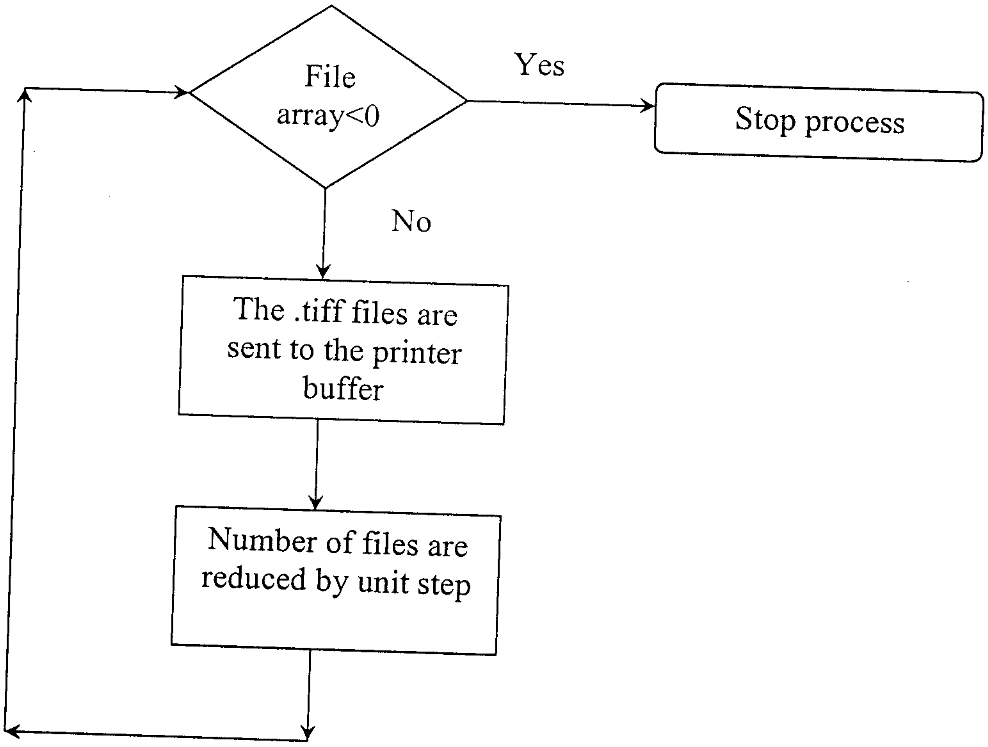
Grid Generation


The width and height of the image is got as input
Is width<1
Yes
Terminate from the Grid generation function
No
Generate Grid of size height x 100 as resolution
After generation width is reduced by 100 and stored

Pixel Mapping onto Grid


Grid and the extracted array are input
Values of arrays are checked
The corresponding map function is invoked
Location of area to be shaded is calculated
Mapped grid is saved as .tiff image and a count updated

Pixel Fabrication


File array<0
Yes
Stop process
No
The .tiff files are sent to the printer buffer
Number of files are reduced by unit step

TESTING

5. TESTING

5.1 Introduction

Testing is a mandatory and continuous process for the successful completion of any project Testing is done at various levels. Integration testing follows Unit Testing. Here, different dependant units are assembled and tested for any bugs that may surface due to the integration of the modules.

A final testing is done to access the correctness of the whole software as such. Here several test cases are taken and real life situations are created while testing. The program timings are also noted and compared with present or express results.

5.2 Unit Testing

In the acquisition module, initially the pixel values in the image header were directly acquired each and every time for processing which was found tiresome. So, it was rectified by taking all the pixel values together and storing in a file.

In case accessing image pixels, the values were taken in bits. So, in order to avoid confusions, indexing was introduced which would convert the bit values into integers.

While generating grid, all the pixel values were used for generation. It was found inefficient to store the file. So, with the help of arrays, the grid is generated.

While saving the graphical output after mapping, the usage of getimage() and putimage() functions made the system to hang on. This paved the way to store the image file in a prescribed format.

5.3 Integration Testing

While integrating two or three modules, several forms of type mismatch problems, variables undefined and null point declaration occurred. While mapping pixels onto the grid, the location of pixel calculations ended in chaos. It has been overcome by introducing Line drawing algorithms which simplified the problem in a much better way.

While integrating the part of mapping the grid and saving it, the problem encountered was 'Out of memory' when saving bitmap images. As it utilizes more memory space, the bitmap image was converted to .tiff image.

The Regression testing, the last step in the Integration Testing was carried out to find out various complications that aroused in the individual modules.

5.4 Control Structure Testing

Condition Testing

The Condition testing is a test case design method that exercises the logical conditions contained in a program module. The various condition testing strategies carried out were as follows,

- **Branch Testing**

  For all compound condition C, the true and false branches of C and every simple condition in C were executed at least once.
- **Branch and Relational operator Testing**

  This technique was performed to guarantee the detection of branch and relational operator errors in a condition.
- **Loop Testing**

  All simple loops, nested loops, concatenated loops and unstructured loops were tested, bugs were detected and corrected.

CONCLUSION

CONCLUSION

The software entitled as **“Pixel Processing in Advanced Textile Design”** is implemented to replace the manual system effectively. As and when compared to the existing system, the software was found much faster and the results were accurate and precise.

The project finally can be used for enlarging bitmap files without breaking the pixels. The pixels are exactly mapped on to the grid and printed in a well-organized manner. The resultant image offers exquisite fabric design for the user.

FUTURE SCOPE

FUTURE SCOPE

The future holds a lot to offer to the development and refinement of this project. The Software is designed in such way that it will be helpful for any further enhancement.

The pixel range can be widened in future. More number of colors can be introduced. Printers with different resolutions can be introduced.

The software is developed only for line printers. This could be made possible for varied printers in future enhancement.

BIBLIOGRAPHY

BIBLIOGRAPHY

1. **Yashvant Kanetkar**, "Let us C", BpB Publication, Inc.1999.

2. **John Corrigan** , "Computer Graphics, Secrets and Solutions", BPB publication, Inc. 1994.

3. **Nelson Johnson**, "Advanced Graphics in C", Osborne McGrae Hill, Inc.1987.

4. **Roger . T & Steven**, "Advanced Graphics in C & C++", BPB Publications, Inc. 1993.

5. **Donald Hearn & M.Pauline Baker**, "Computer Graphics", Prentice Hall International, Inc. 1986.

6. **William K. Pratt**, "Digital Image Processing", A Wiley Interscience Publication, Inc. 1999.

7. **Roger . T & Steven**, "Graphics in C ", BPB Publications, Inc. 1993.

8. **Rafael.C.Gonzalez,** "Digital Image Processing", Pearson Education Asia,Inc.1998

9. **Yashvant Kanetkar**, "Pointers in C" BpB Publication, Inc.1999.

GLOSSARY

GLOSSARY

IDE –Integrated Development Environment

TIFF –Tagged Image File Format

BMP- Bitmap picture

dpi –dots per inch

ppi-pixels per inch

bpp-bits per pixel

ANNEXURES

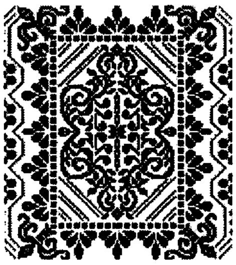
APPENDIX A

ANNEXURES

APPENDIX A OUTPUT SCREENS

IMAGE NAME	:	design1.bmp
RESOLUTION	:	100X100

ORIGINAL IMAGE



ENLARGED IMAGE

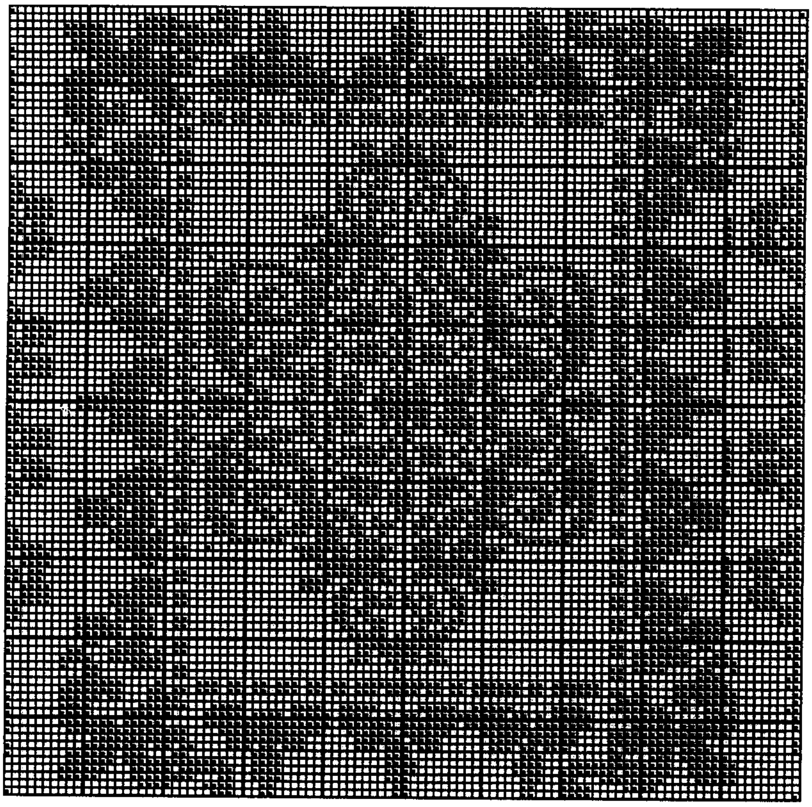
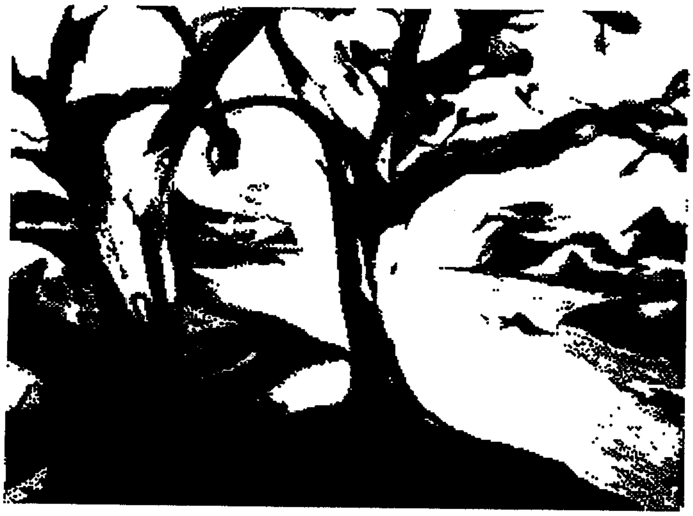


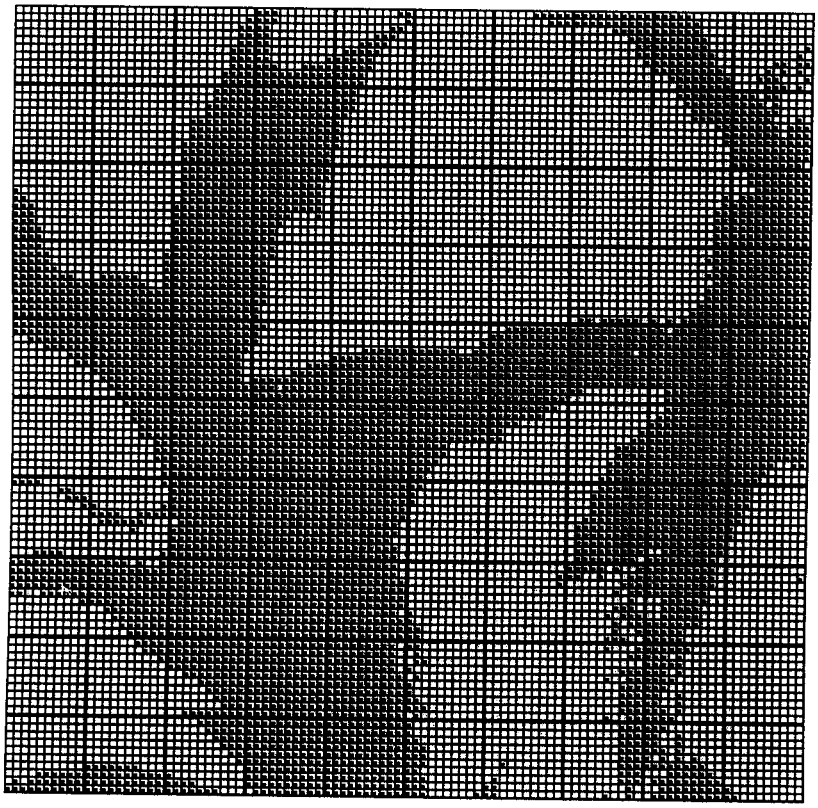
IMAGE NAME : design1.bmp
RESOLUTION : 258x350

ORIGINAL IMAGE

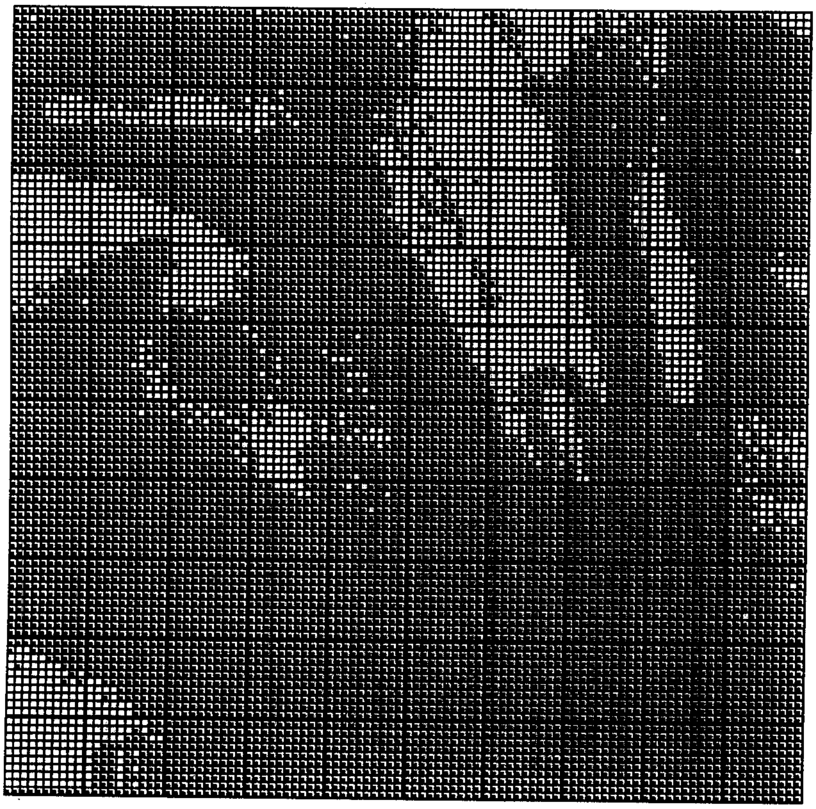


ENLARGED IMAGE

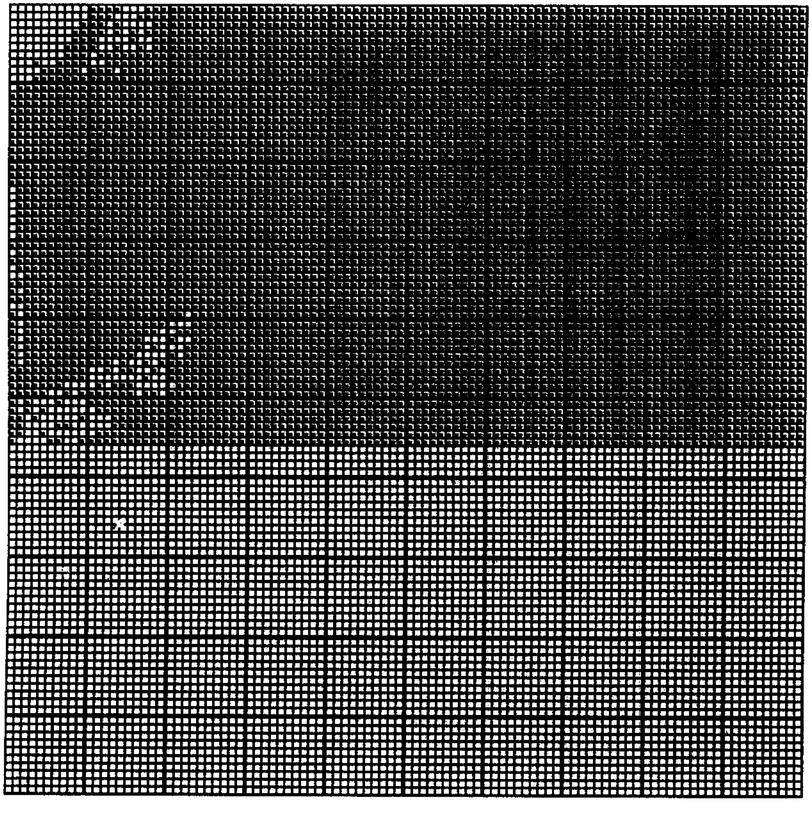
PART 1



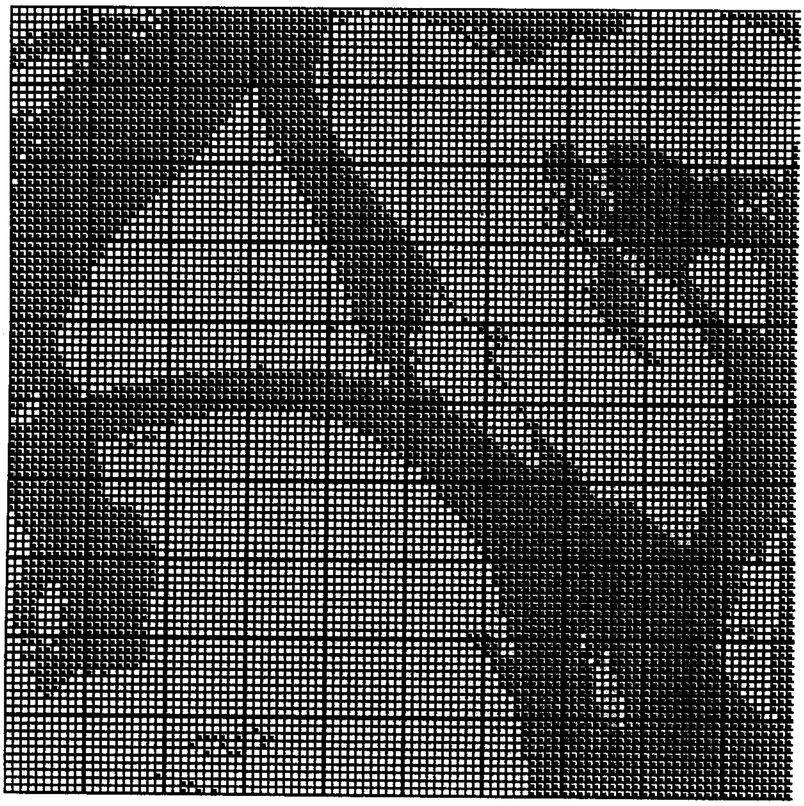
PART 2



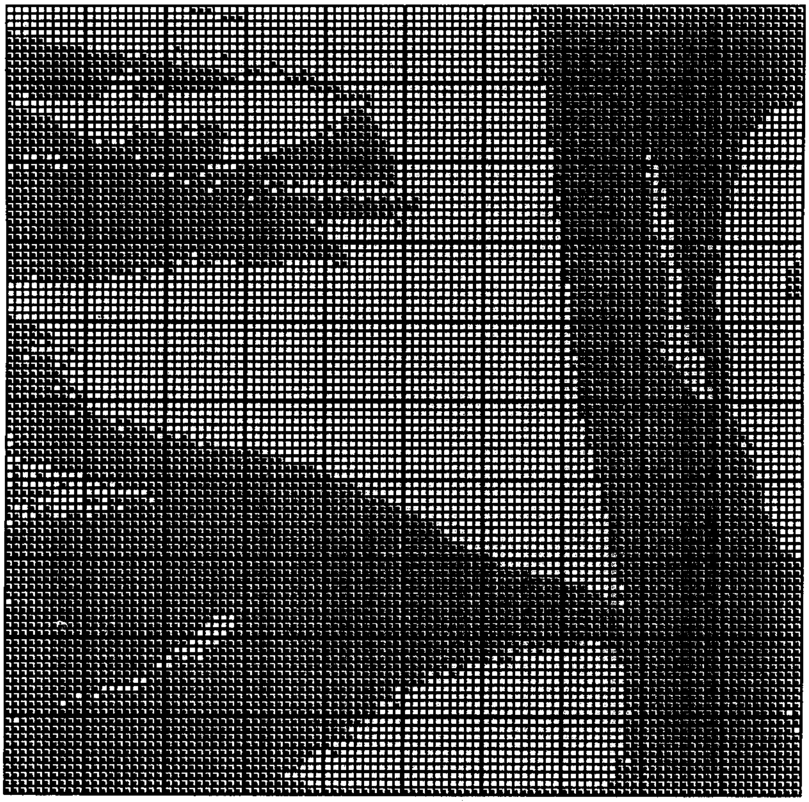
PART 3



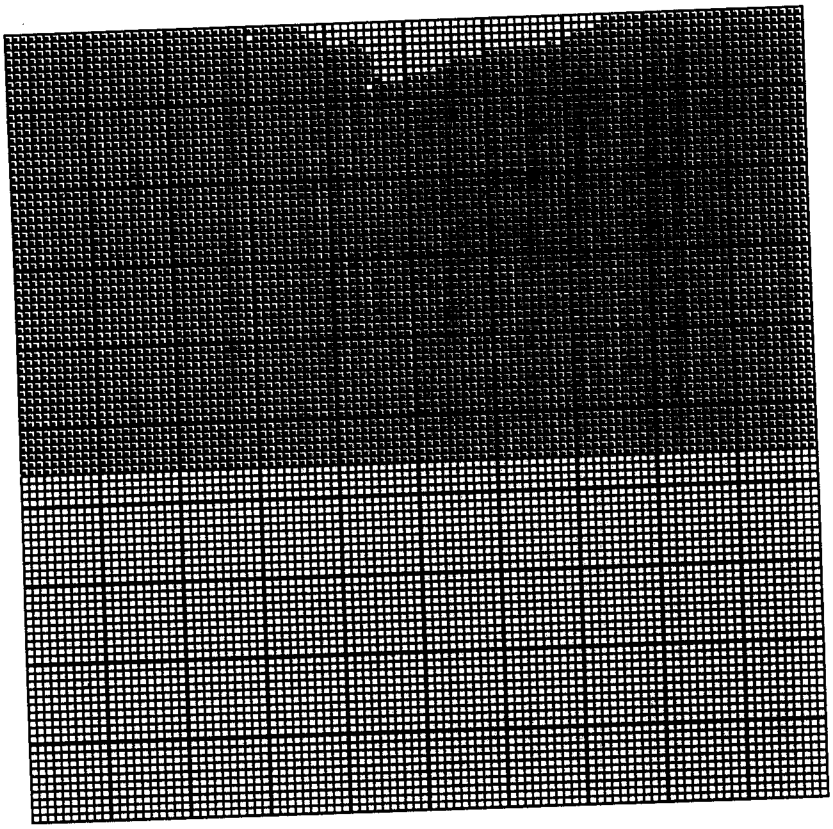
PART 4



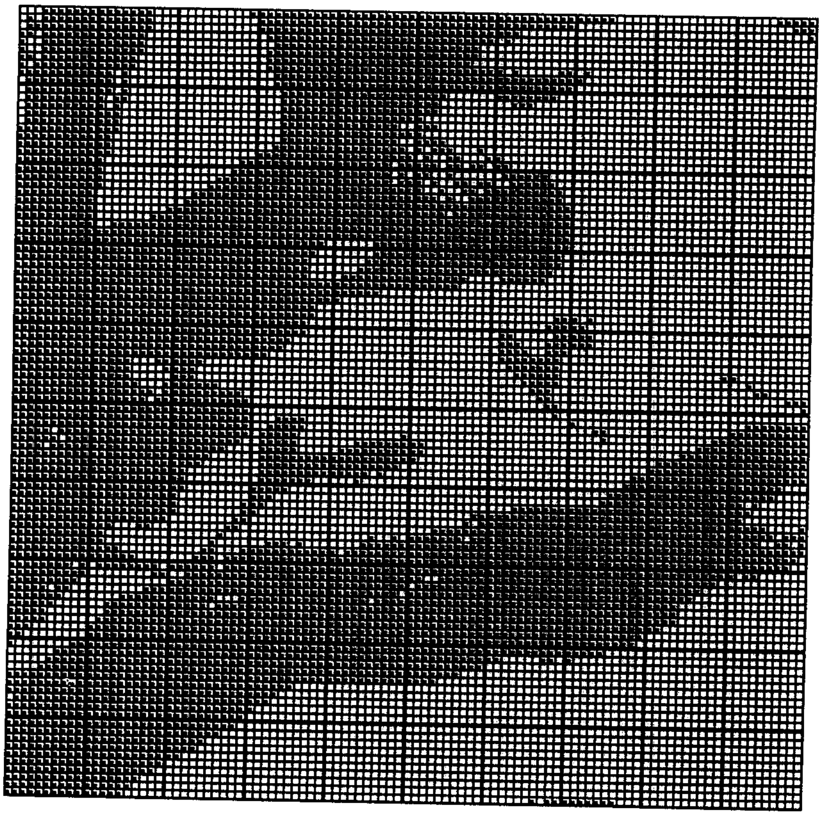
PART 5



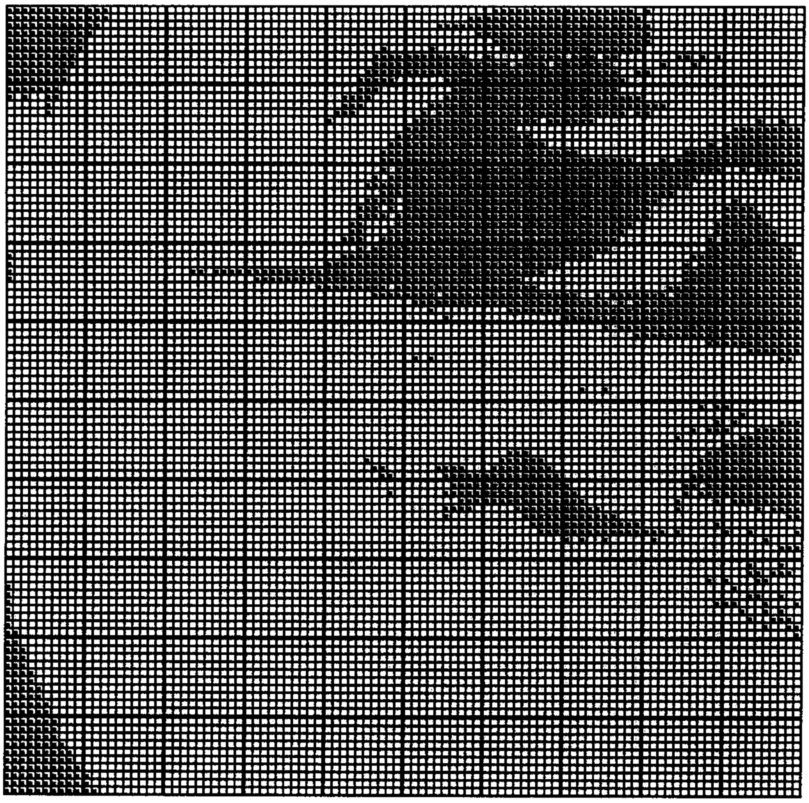
PART 6



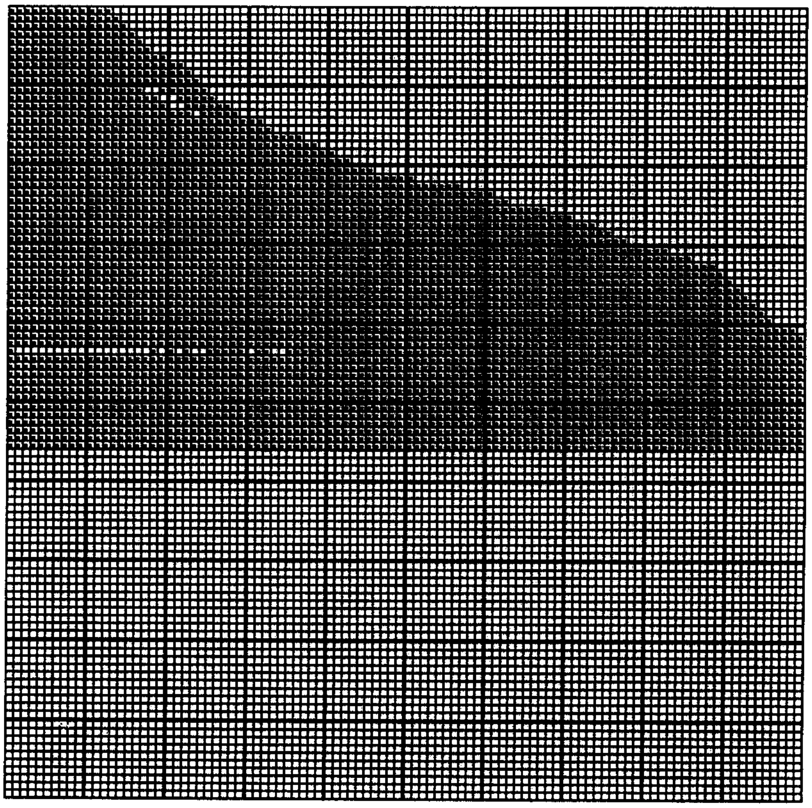
PART 7



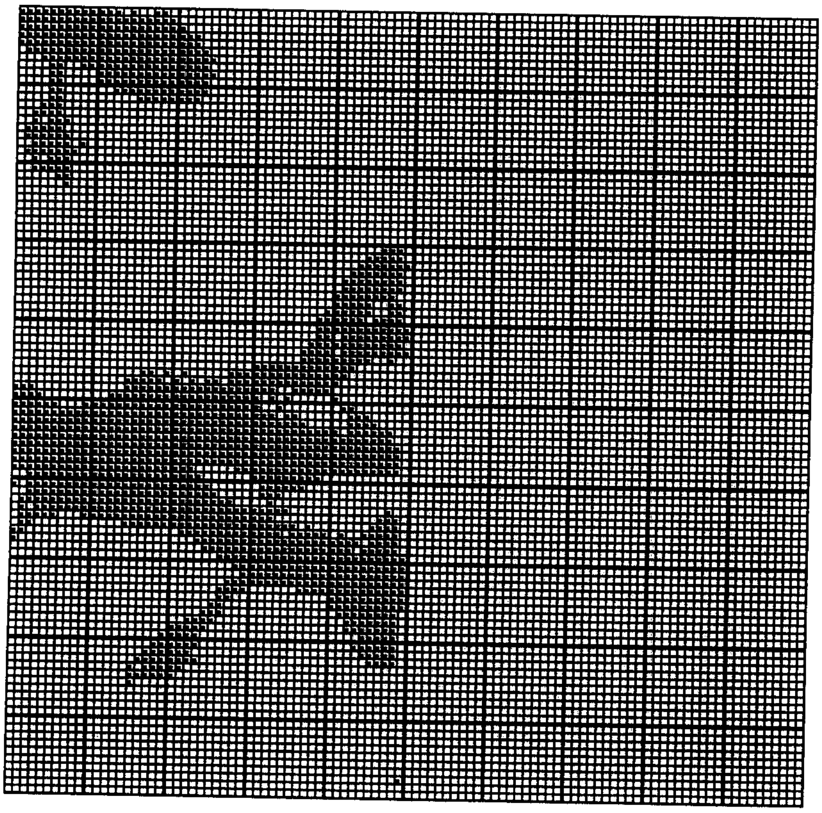
PART 8



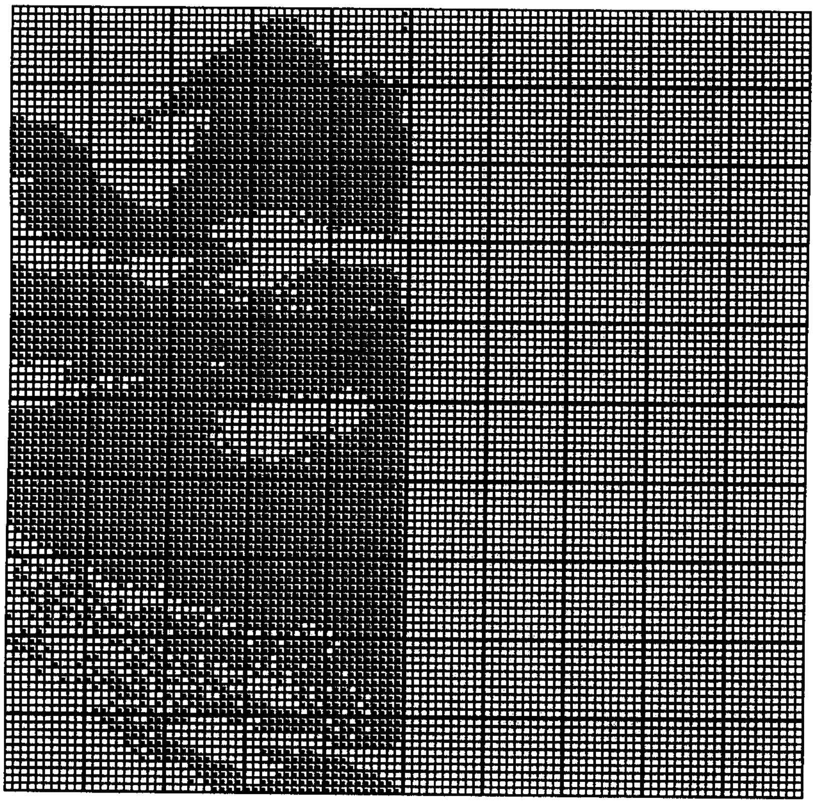
PART 9



PART 10



PART 11



PART 12