An Automatic System Of Segmentation, Recognition And Detection For Any Image By UML

Md Mobin Akhtar, Lecturer, Computer Science Afif, Shaqra University
Md. Ashraf Siddiqui, Lecturer, Computer Science Hafouf, King Faisal University
Najmul Islam, Project/ Research Associate, Indian Institute of Technology (IIT) Delhi

ABSTRACT
Unified Modeling Language, a standard language for designing and documenting a system in an object-oriented manner. UML is not a programming language it is basically a modeling language it’s a language by which Technical architects can communicate with developers. It’s a blue print of our source code. In Our Paper UML model has been designed for the compression of any image using transform method because compressed image take a very less transmission time for sending image from one node to another node and at receiving end after decompression re find the original image. The class diagram and sequence diagram for the compression of an image are depicted in this paper. Some run length coding technique is also reported in this paper with activity diagram.

Keywords:
Run Length Code, Discrete cosine transform, Transform, Class, sequence

1. INTRODUCTION
UML is an influential modeling language used to represent the research problems visually. The Unified Modeling Language (UML) is a visual language for capturing software designs and patterns. It’s a language by which one can express design of software architecture. It’s a blue print of our source code. By the use of UML, Real time system based on UML is described by Selic and Rumbaugh [1]. The first represented of UML in the field of telecommunication sector is described by Holz [2]. The concept of UML was invented by the Greddy Booch et al. [3]. The UML may be used to visualize, specify, construct, and document the artifacts of a software intensive system. The UML is more than just a bunch of graphical symbols. Rather, behind each symbol in the UML notation is a well-defined semantics. The present research work is based on the design of the UML model how the image is compressed at the source end in the network at the time of transforming from one node to another node. By the use of proposed model one can easily develop software for compress any image of any quality very easily. MATLAB tools are available for development of any software related of digital image processing.

2. PROCESS
Let us first define the process which may be the group or block of instructions of program, macro, sub programs and subroutines. For defining the process, there is a need of the processing element. The processing element is defined as a stereotype and generally used to handle the concurrent process executing in the parallel and distributed environments. The following Fig. 1 shows the definition of processing unit.

Figure 1. Definition of Processing Unit
process_create()
process_delete()
process_update()
process_join()
process_suspend()
process_synchronize()

Figure 2. Class definition of process
A class diagram captures the static relationships of your software. A class is represented by a rectangular box divided into compartments. Compartment is an area in the box to write information. The first compartment holds the name, the second holds the attributes and the third is used for the operations.

The instance of the process is defined by the use of object abc which is shown in Fig 3:

<processing_unit>>
abc:process

Figure 3. Instance of class
The set of the instances of the class process is modeled by the use of multiple objects which is shown below:

<<processing_unit>>
_process

Figure 4. Multiple instances of object

3. LOAD THE IMAGE
Process is use to load the image. In spatial coordinate system image is stored in the frame buffer of size NXN dimension. Each cell contain the coordinate of pixel (x,y).In the gray scale scheme each pixel use 8 bits to store the information but in the colored scheme, three plane are there for each R,G and B. The monochrome image f(x,y) is discredited both in spatial coordinates and gray level values to obtain the digital image. A digital image can be represented as matrix whose row and columns are used to locate a point in the image and corresponding element values give the gray level at that point.

In The MATLAB the function
image = imread(.file name.) is use to load the image and we can access the each pixel by using this method. F(x,y) = image[(x,y)] x vary from 0 to N and y vary from 0 to N.

4. TRANSFORMATION OF THE IMAGE
In the Image Processing Discrete cosine transform, the DCT is used in JPEG image compression, MJPEG, MPEG, DV, the two-dimensional DCT-II of N x N blocks are computed and the results are quantized and entropy coded. In this case, N is typically 8 and the DCT-II formula is applied to each row and column of the block. The result is an 8 x 8 transform coefficient array in which the (0,0) element (top-left) is the DC (zero-frequency) component and entries with increasing vertical and horizontal index values represent higher vertical and horizontal spatial frequencies.

Figure 5:DCT-II (bottom) compared to the DFT (middle) of an input signal (top).

There are different frequency transform approaches as Fourier transform use for image enhancement and image reconstruction, Discrete cosine transform widely use for image compression. The popular transform used for image compression is discrete cosine transform so it is discussed in this paper in detail. It is define as

\[ C(u,v) = a(u)b(v) \sum f(x,y) \cos((2x+1)\pi u/2N) \cos((2y+1)\pi v/2N) \]

Where
x and y vary from 0 to N-1 and u and v varies from 0 to N-1

Where
\[ a(u) = \sqrt{1/N} \text{ for } u = 0 \text{ and } \sqrt{2/N} \text{ for } u = 1 \text{ to } N-1 \]
\[ b(v) = \sqrt{1/N} \text{ for } v = 0 \text{ and } \sqrt{2/N} \text{ for } v = 1 \text{ to } N-1. \]

f(x,y) is image in spatial coordinate system.

Cosine transform is real and orthogonal i.e \( C^{-1} = C^T \) and it has a excellent energy compaction for
correlated data. In MATLAB the instruction used to transform the spatial coordinate into Discrete cosine transform is as Image1 = Dct (image)

5. EXTRACTION APPLICATION
Compression is useful because it helps reduce resources usage, such as data storage space or transmission capacity. Compression is a process of reducing the amount of data required to represent a given quantity of information or it means to remove the redundant data in the information. Various redundancy methods are there for compression of an image as variable length coding and run length coding. For variable length coding, if variable length of bits is used to represent a pixel then one can eliminate redundancy compared to the fixed length bits of a pixel. In this paper Run length coding is explained in detail.

5.1 Image Data Compression
Image data compression is important for
1. Image archiving e.g. satellite data.
2. Image transmission e.g. web data
3. Multimedia applications e.g. desk-top editing
4. Image data compression exploits redundancy for more efficient coding

A data compression technique for storing raster data. Run-length encoding stores data by row. If two or more adjacent cells in a row have the same value, the database stores that value once instead of recording a separate value for each cell. The more adjacent cells there are with the same value, the greater the compression.

5.2 Run Length Coding
To compress any data either in text or character run length encoding is a data compression algorithm that is supported by nearly all bitmap file formats, such as Tiff, BMP file. Run Length Encoding is both easy to implement and quick to execute, making it a good alternative to either using a complex compression algorithm or leaving image data uncompressed.

By reducing the physical size of a repeating string of characters Run length Encoding (RLE) works in this manner. Uncompressed, a character run of 15 X characters would normally require 15 bytes to store: XXXXXXXXXXX. The same string after RLE encoding would require only two bytes: 15X. The 15X code generated to represent the character string is called an RLE packet. Here, the first byte, 15 is the run count and contains the number of repetitions. The second byte, A is the run value and contains the actual repeated value in the run. A new packet is generated each time the run character changes, or each time the number of characters in the run exceeds the maximum count. Assume that our 15-character string now contains four different character runs:

AAAAAAbbhhXXXXXt. Using run-length encoding this could be compressed into four 2-byte packets: 6A3b5X1t

Thus, after run-length encoding, the 15-byte string would require only eight bytes of data to represent the string, as opposed to the original 15 bytes. In this case, run-length encoding yielded a compression ratio of almost 2 to 1. Let us take another example to encode this string:

Xtmprsqzntwlfb

RLE schemes are simple and fast, but their compression efficiency depends on the type of image data being encoded. A black-and-white
image that is mostly white, such as the page of a book, will encode very well, due to the large amount of contiguous data that is all the same color. An image with many colors that is very busy in appearance, however, such as a photograph, will not encode very well. This is because the complexity of the image is expressed as a large number of different colors. And because of this complexity there will be relatively few runs of the same color. One can easily decode the run length code at the time of decompression.

5.3 Variants on Run-Length Encoding
Run length encoding is a principle to compress repeated strings in a stream of strings. There are a number of variants of run-length encoding. Image data is normally run-length encoded in a sequential process that treats the image data as a 1D stream, rather than as a 2D map of data. In sequential processing, a bitmap is encoded starting at the upper left corner and proceeding from left to right across each scan line (the X axis) to the bottom right corner of the bitmap but alternative RLE schemes can also be written to encode data down the length of a bitmap (the Y axis) along the columns. To encode a bitmap into 2D tiles or even to encode pixels on a diagonal in a zig-zag fashion as shown in Fig 8d. This RLE variant works well only with real-world images that contain many subtle variations in pixel values.

RLE encoder always stops at the end of each scan line of bitmap data that is being encoded. There are several benefits to doing so. Encoding only a simple scan line at a time means that only a minimal buffer size is required. Encoding only a simple line at a time also prevents a problem known as cross-coding. Cross-coding is the merging of scan lines that occurs when the encoded process loses the distinction between the original scan lines. Cross-coding is sometimes done, When an encoder is encoding an image, an end-of-scan-line marker is placed in the encoded data to inform the decoding software that the end of the scan line has been reached. This marker is usually a unique packet, explicitly defined in the RLE specification, which cannot be confused with any other data packets. End-of-scan-line markers are usually only one byte in length, so they don’t adversely contribute to the size of the encoded data.

6. UML MODEL FOR COMPRESSION OF IMAGE
Step involved in compressing an image is as follows.

1. Load image
2. Take a sub image of size 8x8
3. Apply cosine transform for this sub image
4. Then these coefficients are normalized using the Normalized matrix and apply run length coding Technique to compress the image.
5. Repeat the same procedure until all image are Scanned.

UML is authoritative device for modeling of object oriented software system. UML class Diagram is shown in Figure-9 which shows how the classes are associated to each other for compressing any image. In the Class diagram one can see that how message passing take place among the different classes like Source Image data 8x8 Block, Forward
DCT, Quantized, Entropy Encoder and compress image data. It also shows that how long time that all classes involve in Execution. Activity diagram shown in fig 7, which shows organized steps implicated in compressing any image of any type.

Figure 9: UML Class Diagram for compression of image.

Figure 10: UML Activity diagram for compression of image.

7. UNIFIED MODELING LANGUAGE FOR EXTRACTING AN IMAGE

At receiving end extracting an image is required for finding an original image. Therefore in compressed file apply inverse run length coding for getting again 8x8 images after that apply quantization and inverse DCT transform and merging all the 8x8 sub images to get the decompress file as original image.

Unified Modeling Language diagram is depicted in the Fig-11 for decompressing a compressed file.

Figure 11: UML Class Diagram for uncompressing of image

8. CONCLUSION AND UPCOMING SCOPE OF WORK

All the function of image processing can be without difficulty implemented with the help of Unified Modeling Language such as image segmentation, detection of line in the image, pattern recognition in the image as well as water marking in the image. This modeling technique will be suitable for software developer to develop software regarding image processing.

REFERENCES