Numerous Image Transaction with Reduced Space using Encoding Compression Technique

D. Rajagopal, K. Thilakavalli, J. Satheesh, A. Syedali Fathima

1. Assistant Professor, Department of Computer Science, K.S.R College of Arts and Science, Tiruchengode, Namakkal Dt, Tamil Nadu, India – 637 215.
2. Assistant Professor, Department of Computer Science, K.S.R College of Arts and Science for women, Tiruchengode, Namakkal Dt, Tamil Nadu, India – 637 215.
3. Assistant Professor, Department of Computer Science, K.S.R College of Arts and Science, Tiruchengode, Namakkal Dt, Tamil Nadu, India – 637 215.
4. Assistant Professor, Department of Computer Science and Engineering, M.Kumarasamy College of Engineering, Tamil Nadu, India

E-mail: sakthiraj2782007@gmail.com, thilaksathyav2782007@gmail.com, j.satish@ksrcas.edu, safathima07@gmail.com

ABSTRACT

Most organizations maintaining the documents as image format. For the purpose they are handling those images, they want to spend large memory space to their computers. If an image has large size, they cannot able to attach and send that image to others through E-mail (Electronic Mail). While the communication, the hackers can able to theft the data from the image. In this paper tells how the numerous images can store into small memory space and how the numerous images can share with security.

Keywords: Image, Electronic Mail, Secret Key, Memory.

1. INTRODUCTION:

In an image processing, the compression is most widely technique. Compression is the technique for reduce the image size of bytes without any losses of source image reality and quality. The reduction of the size of an image allows, storing more number of images in a given amount of memory space. It is used to share the more images into a single E-Mail transaction. The objective of compression is to reduce the number of bits as much as possible, while keeping the resolution and the visual quality of the reconstructed image as close to the original image as possible [1]. While sharing the bunch of images the attacker hack this bunches to get secret information, which has the image. Inserting a unique secret code of each bunch has to avoid the secret information hacking.

2. COMPRESSION:

Data compression technique algorithm can be classified into two types. They are
1. Lossless compression algorithm
   Lossless compression algorithms remove only the redundancy of the source content, while reconstructing the source content does not have any change.
2. Lossy compression algorithm
   Lossy compression algorithm is compressing the source content, while reconstructing the source content losses some content of the source [4].

2.1 LOSSLESS COMPRESSION ALGORITHM:

Lossless compression algorithm are listed below
1. Repetitive Sequence Suppression
   Straight forward to understand, implement.
   1.1 Simple Repetition Suppression:
      • Replace series with a token and a count number of occurrences.
      • Usually need to have a special flag to denote when repeated token occurs.
2. Run-Length Encoding(RLE):
   This encoding method is frequently applied to graphics-type images. Basic RLE approach (e.g. for images):
   • Sequences of image elements X1:X2:::Xn (Row by Row)
   • Mapped to pairs (c1; 11); (c2; 22);::; (cn; ln)
     Where ci represent image intensity or color, li is length of the ith run of pixels.
3. Pattern Substitution:
   This is a simple form of statistical encoding. Substitute frequently repeating pattern.
4. Entropy Encoding:
It is based on information theoretic techniques.

4.1 Shannon-Fano algorithm:
This is the basic information theoretic algorithm. It is used to finite token stream. The count of the symbol in the stream has used in the compression.

4.2 Huffman Coding:
It is based on the frequency of occurrence of a data item. Lower number of bits encodes frequent data.

4.3 Arithmetic Coding:
It is widely used entropy coder. Extend the source to binary and assign each symbol to the range. Subdivide the range for the first token given the probability of next etc.

5. Lempel-Ziv-Welch (LZW) algorithm:
LZW algorithm used dictionary with 4k entries, first 256(0-255) are ASCII codes [2].

2.2 LOSSY COMPRESSION ALGORITHM:
Lossy transform codecs, samples of picture or sound are taken, divided into small segments, transformed into a new basis space, and quantized. The resulting quantized values are then entropy coded.

Lossy predictive codecs, previous and/or subsequent decoded data is used to predict the current sound sample or image frame. The error between the predicted data and the real data, together with any extra information needed to reproduce the prediction, is then quantized and coded.

2.3 LOSSY VERSUS LOSSLESS:

<table>
<thead>
<tr>
<th>LOSSY</th>
<th>LOSSLESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Produce a much smaller compressed file</td>
<td>Not much smaller</td>
</tr>
<tr>
<td>Most often used for compressing sound, images or videos</td>
<td>Used for compressing sound, images or videos</td>
</tr>
<tr>
<td>Transparent</td>
<td>Opaque</td>
</tr>
</tbody>
</table>

3. PROPOSED SYSTEM:
Proposed system has been implemented one of the Lossless algorithm, which the name is arithmetic coding algorithm. This algorithm has derived from the Entropy encoding lossless algorithm. For storing n number of images into a small size of memory, each image has been saved as varbinary (MAX) format in database. Filestream can store unstructured binary data (image data) into the database. Varbinary (MAX) column to tell the data has to be stored directly on the file system. Filestream provides an easy way to handling BLOB (Binary Large Object) data with database transactional consistency and thus improves the performance [5]. The size of the file is limited by the disk space that makes this feature more useful. Varbinary (MAX) can store images with maximum size of 2GB. The large image has been saved into single byte using the proposed system. The Varbinary(MAX) format images cannot display the images directly. So the hackers and unauthorized users can’t see the image content. Proposed system can able to store numerous images among less memory usage. The proposed system users only can view and handle numerous images with security.

4. ENTROPY ALGORITHM:
Entropy is a lower bound on the average number of bits needed to represent the symbols (the data compression limit).

- Entropy coding methods:
  - Aspire to achieve the entropy for a given alphabet,
  - A code achieving the entropy limit is optimal

BPS: Bits Per Symbol

\[ BPS = \frac{|\text{encoded message}|}{|\text{original message}|} \]

\[ H(P_1...P_n) = -\sum P_i \log_2 P_i \]

Where
\[ P_i \] is occurrence probability of symbol \( S_i \) in the input string.
\[ \sum_{i\in A} P_i = 1 \]

Entropy of a set of elements \( e_1, ..., e_n \) with probabilities \( P_1, ..., P_n \) [3].

4.1 ARITHMETIC CODING ALGORITHM:
Assigns one (normally long) codeword to entire input stream and Reads the input stream symbol by symbol, appending more bits to the codeword each time.

Codeword is a number, representing a segmental subsection based on the symbols and
probabilities and Encodes symbols using a non-integer number of bits. For Example

Figure: 4.1.1 Example Coding of BCAE

<table>
<thead>
<tr>
<th>Coding of BCAE</th>
<th>pA = pD = 0.25, pB = 0.4, pC = 0.15</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.25</td>
</tr>
<tr>
<td>B</td>
<td>0.4</td>
</tr>
<tr>
<td>C</td>
<td>0.15</td>
</tr>
<tr>
<td>D</td>
<td>0.25</td>
</tr>
<tr>
<td>E</td>
<td>0.3875</td>
</tr>
</tbody>
</table>

Any number in this range represents BCAE

Figure: 4.1.2 Mathematical Definition of BCAE

4.2 ARITHMETIC ENCODING

Initially \(L=0, R=1\) when encoding next symbol, \(L\) and \(R\) are refined

\[
L \leftarrow L + R \sum_{i=1}^{n} p_i, \quad R \leftarrow R \cdot p_i
\]

At the end of the message, a binary value between \(L\) and \(L+R\) will unambiguously specify the input message. The shortest such binary string is transmitted.

In the previous example

Any number between 385625 and 3875 (discard the 0)

Shortest number -386, in binary: 110000010

BPS = (9 bits / 4 Symbols) = 2.25

Two possibilities for the encoder to signal to the decoder end of the transmission:

1. Send initially the number of symbols
2. Assign a new EOF symbol in the alphabet, with a very small probability, and encode it at the end of the message.

The order of the symbol in the alphabet must remain consistent throughout the algorithm. In order to decode the message, the symbols order and probabilities must be passed to the decoder. The decoding process is identical to the encoding. Given the codeword (the final number), at each iteration the corresponding sub-range is entered, decoding the symbol representing the specific range.

Figure: 4.2.1 Example of Arithmetic Decoding BCAE

Arithmetic coding manages to encode symbols using non-integer number of bits. One codeword is assigned to the entire input stream, instead of a codeword to each individual symbol. This allows arithmetic coding to achieve the Entropy lower bound.

5. DISTRIBUTION ISSUES:

Symbol distributions were known in advance. The input string cannot know. Therefore, the arithmetic coding and Huffman have an adaptive version. Using this version distributions are updated as the input string is read and it can work online.

6. EXPERIMENTAL RESULT:

The images are encrypted and stored into the database using proposed system. While using proposed, system to encrypt large size of image can be reduced from 42086 kb to 32 kb. The amount of reduced image size is multiples of 1315. The medium size of image can be reduced from 2304 kb to 48 kb. The amount of reduced image size is multiples of nearly 50. The small size of image can be reduced from 428 kb to 44 kb. The amount of reduced image size is multiples of below 10. The experimental result has listed in the table.
7. CONCLUSION:

This paper concludes that, using the proposed system can encrypt the large size of images can be stored and shared effectively with security. In the future, for making the system adaptive to the changing environment, it will be necessary that the system will be produced above the average, which is provided by the proposed system.

8. ABOUT THE AUTHORS

D. RAJAGOPAL has completed his Bachelor of Computer Science degree and completed his Master of Computer Applications degree in Periyar University in the year 2003 and 2006 respectively. He completed his Master of Philosophy in PRIST University in the year of 2012. He has 3 years and 10 months experience in the field of software developing and 4 years 4 months experience in the field of teaching. He delivered more than 10 seminars & training programs for different academic institutions. He published a paper in an international journal. His research area of interest is computer networks (Wireless & Wired), Image processing, Mobile computing, Data mining, Software programming (OOPS).

K. THILAKAVALLI has completed her Bachelor of Physics degree in Bharathiar University in the year of 2006. She completed her Master of Computer Applications degree in Anna University in the year 2009. She completed her Master of Philosophy in PRIST University in the year of 2010. She has 5 years experience in the field of teaching. She published 3 international conference papers, a national conference paper. She published a paper in an international journal too. Her research area of interest is image processing, Computer networks (Wireless & Wired), Mobile computing, Data mining.

J. SATHEESH has completed his Bachelor of Computer Science degree in Bharathiar University and completed his Master of Computer Applications degree in Anna University in the year 2007 and 2010 respectively. He is doing his Master of Philosophy in Bharathiar University. He has 2 years 8 months experience in the field of teaching. His research area of interest is Image processing, Data mining.

A. SYEDALI FATHIMA has completed her Bachelor of Engineering in Syed Ammal Engineering College, Anna University in the year 2009, and she completed her Master of Engineering degree in Kalasalingam University in the year of 2012. She has 3 years experience in the field of teaching. Currently she is working as an Assistant Professor in the Department of Computer Science and Engineering in M.Kumarasamy College of Engineering. Her research area of interest is Image processing, Computer networks, Mobile computing, Data mining.

9. REFERENCES


ANNEXURE - I

CODE FOR ENCODING:

Dim mystream As New ADODB.Stream
Dim tmp As String
Dim Fn As Integer

Public Sub picSave(sObject As Object, Rs As Recordset, sField As String, Spath As String)
    SavePicture sObject, Spath
    mystream.Open
    mystream.Type = adTypeBinary
    mystream.LoadFromFile Spath
    Rs(sField) = mystream.Read
    mystream.Close
End Sub

CODE FOR DECODE:

Dim mystream As New ADODB.Stream
Dim tmp As String
Dim Fn As Integer

Public Sub picRetrieve(Rs As Recordset, sField As String, sObject As Object, Spath As String)
    mystream.Open
    mystream.Type = adTypeBinary
    mystream.Write Rs(sField)
    mystream.SaveToFileSpath,
    adSaveCreateOverWrite
End Sub

CODE FOR CHANGE RESOLUTION:

Dim DevChg As DEVMODE
Dim ResultCh As Long
Dim Ask As Integer
Dim iWidth As Single
Dim iHeight As Single

Public Sub ChangeRes()
iWidth = 1280
iHeight = 1024

'Retrieve info about the current graphics mode on the current display device
ResultCh = EnumDisplaySettings(0&, 0&, DevChg)

'set the new resolution, pixels height and pixels width
DevChg.dmFields = DM_PELSWIDTH Or DM_PELSHEIGHT
DevChg.dmPelsWidth = iWidth
DevChg.dmPelsHeight = iHeight

'change the display setting
ResultCh = ChangeDisplaySettings(DevChg, CDS_TEST)
Select Case ResultCh
Case DISP_CHANGE_RESTART
    Ask = MsgBox("You must restart your computer, and then open the program again." & vbCrLf & "Do you want to restart now?", vbYesNo + vbSystemModal)
    If Ask = vbYes Then Call ExitWindowsEx(EWX_REBOOT, 0)
Case DISP_CHANGE_SUCCESSFUL
    Call ChangeDisplaySettings(DevChg, CDS_UPDATEREGISTRY)
    'Case Else
    'computer not support
    'MsgBox "Mode not support.", vbSystemModal, "Screen Resolution..."
End Select
End Sub

Dim mystream.Close
sObject.Picture = LoadPicture(Spath)
End Sub