IMAGE COMPRESSION USING SAND ALGORITHM

Karimella Vikram¹, Dr. Niraj Upadhyaya², Dr.A.Goverdhan³, T.Madhavi Kumari⁴

¹Associate Professor and HOD, Dept of CSE, Medak College of Engineering and Technology, Kondapak, Medak Dist
Research Scholar, CSE Dept., Jawaharlal Nehru Technological University, hyderabad-515002, Andhra Pradesh, India
Email: vikramkariella@yahoo.com
²Professor, Dept. of CSE, JBIET Moinabad, hyderabad, Andhra Pradesh, India
Email: drniraju@gmail.com
³Professor, Dept. of CSE JNTUH
⁴Professor, Dept. of CSE JNTUH

ABSTRACT

In today’s world transfer of information plays a vital role. Today the term information is not just confined to literal data. Images and videos convey a lot more than text messages. But transmission and storage of images have their own setbacks. The major one being the space and time complexity and this is where the compression of images comes into picture. Ours is one such algorithm indigenously developed by me, which has the potential to compress the images with out any loss of any information. This makes the compression algorithm ideal for domains where loss of information is unacceptable. One major application can be in the transfer of information on stellar objects by non-terrestrial telescopes, spacecrafts and space probes, which sends back vital information and it is also necessary to ensure the bandwidth and also the cost of transmission is low. It can also be employed in bio-telemetry and other related fields where information transmitted has to be reliable (without any loss) for faithful analysis of the data and at the same time the cost factor should also be considered. We have an ideal solution for all these domains. We haven’t confined our self to the theoretical proof of this algorithm; rather we have also come up with a practical working model of the algorithm which implements a section of the algorithm to show its practical feasibility. The software developed works on the basis of Sand
algorithm; it is demo software, which can be implemented on 8 bit (256 color .bmp) color images.

I. INTRODUCTION

AND works on the principle of elimination of repetitions, it literally absorbs pixels of the same nature (color) as the name suggests. In other words the pixels of the same nature converges to a single point this reduces the number of pixels which in turn reduces the size of the image thereby increasing the rate of transmission. The coding of the information is so done that during the decompressing process not a single bit of information is lost.

II. ILLUSTRATION

Any digitized image can be broken into pixels and the number of pixels per area is proportional to the resolution of the image (fig 0). These pixels hold information for recreating the image. If the picture is a 16 bit image then each pixel would hold 16 bits of information as to what color has to be displayed. For example if you were to take a photograph of yourself in a studio with a red screen as a background then the region except your image in the photograph would be red. It is this region that the sand takes advantage of, it sucks all the pixel into almost a single pixel. The absorption process has to take place in such a way that it can be recreated without any loss. Thus the compression is a sequential process the flow diagram of which is shown in fig.

A. Algorithm
The flow diagram shown above gives an idea of the various processes involved in the compression process each of which will be discussed in the following sections. As it is known, that every picture is made up of thousands of pixels and each pixel has its colour and these pixels in combination form the image. That would mean each pixel requires 32 bits to represent a pixel. The picture below shows an illustrative view of the arrangement of pixels in a picture.

![Zoomed View of the Pixel Arrangement](image)

The compression process can be broadly classified into two sections where one process follows the other and they are termed as Latitudinal compression Longitudinal compression. To understand the compression process in a better way let us take a segment of the picture and implement the compression process on that. The block in fig (1) shows a section of a picture with pixels arranged, each circle represents a pixel.

![Latitudinal Compression](image)

Fig-(1) Latitudinal Compression

The block shown in fig (1) represents a block that would undergo latitudinal compression. In the latitudinal compression process the block is first divided into rows (5 rows in this example), each row is processed separately. The processing involves taking the first pixel and comparing them with the consecutive pixel. Depending on the result of the comparison, further processing is carried out. If the color of first pixel does not match with the second pixel, both the pixels are retained. If both pixels tend to have the same color then the first pixel is retained while the second pixel is eliminated or absorbed. This process is applied to each row of the picture. Employing the same logic on the fig (1) we get fig(1.1). Taking the first row we have two blue pixels that are absorbed into one, likewise in the fourth row, three red pixels are absorbed into one pixel (absorbed regions are represented by dots). Similar process is carried out through out. This process is applied to each row of the picture. Employing the same logic on the fig (1) we get fig(1.1). Taking the first row we have two blue pixels that are absorbed into one, likewise in the fourth row, three red pixels are absorbed into one pixel (absorbed regions are represented by dots). Similar process is carried out through out. The fig(1.1) shows the pixel block after the application of latitudinal compression.
Latitudinal Compression

After the latitudinal compression, the next step involves the creation of “Single bit absorption table“ (SBA table). It is this table, which gives information regarding the absorption pattern of the pixels. A simple logic is followed for this, Absorbed pixels in the block are represented as “0” and the unabsorbed pixels are represented as “1”. Thus we get the table as seen in fig. The size of the pixel is proportional to the size of the block.

Single Bit Absorption Table (for Latitudinal Comp)

<table>
<thead>
<tr>
<th>1</th>
<th>1</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Fig

2) Longitudinal Compression

This is similar to the latitudinal compression the only difference being that the pixel block after the latitudinal compression is taken and is divided into columns and the compression process is carried out vertically (column wise) columns and the compression process is carried out vertically (column wise)

This block is obtained from fig-2.1 by discarding the dots (absorbed pixels)

Transmission / storage box

<table>
<thead>
<tr>
<th>1</th>
<th>1</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Fig
3) Efficiency of Compression

The efficiency of the compression depends completely on the picture and the arrangement of the pixels. The efficiency varies from picture to picture. Considering the blocks taken here the efficiency would be given as follows. If the picture taken here is a 24 bit color picture then the memory occupied by the pixel block before the compression process would be

\[ Siz_{\text{before}} = (N \times W) \]

Where,
- \( N \) = Number of pixels in the block.
- \( W \) = Weight of each pixel.

Thus,

\[ Siz_{\text{before}} = (25 \times 24) = 600 \]

After the compression process,

\[ Siz_{\text{after}} = (Nr \times W) + OH \]

Where,
- \( Nr \) = Number of pixels remaining after Compression
- \( OH \) = Over head due to the SBA tables

\[ OH = (2 \times N) = (2 \times 25) = 50. \]

\[ Siz_{\text{after}} = (13 \times 24) + 50 = 362 \]

Thus, this yields a reduction of nearly 40% of the original size of the block. Practically higher compression ratio is realizable considering the number of pixels in the pixel block to be much higher.

B. Retrieving your Image

So far we have been talking about compressing the image. More important than this is retrieving the image without any loss. As said earlier the information at the receiving end would be the resultant block (fig 3.1) and the two tables (fig 3.2,3.3). We need to create the Image using these three information units. The decompressing algorithm works conversely to that of the compression algorithm. It is responsible for decoding the information from the received “Transmission box”. It makes use of a simple concept; the processing takes place as follows. The two tables are analyzed bit by bit; if the bit in both the tables is “1” then the pixel is got from the “Resultant block”. This process is depicted in the diagram shown below, where the shaded (yellow) region represents the bit being processed. As seen since both pixels are “1” the pixel from the resultant block is moved to the “Recreation block”. The second condition is that when the bit in both the SBA table is “0” then the pixel gets the color of the pixel on the left in the recreation block. The example for which is shown below. The third condition is encountered when the bit in the first SBA table is “1” and the second table has the value as “0”. In such a case the pixel gets the color of the pixel above it in the “Recreation table”. The example for which is show below. Note that the “Resultant block” comes into picture only when both the table have the data as “1”. For all other conditions the “Recreation box” itself acts as the source for the pixels. The illustration for the third condition is shown below. The processing is carried out in the similar fashion; every bit The two tables are analyzed bit by bit; if the bit in both the tables is “1” then the pixel is got from the “Resultant block”. This process is depicted in the diagram shown below, where the shaded (yellow) region represents the bit being processed. As seen since both pixels are “1” the pixel from the resultant block is moved to the “Recreation block”. The second condition is that when the bit in both the SBA table is “0” then the pixel gets the color of the pixel on the left in the recreation block. The example for which is shown below.
block” The second condition is that when the bit in both the SBA table is “0” then the pixel gets the color of the pixel on the left in the recreation block. The example for which is shown below. The third condition is encountered when the bit in the first SBA table is “1” and the second table has the value as “0”. In such a case the pixel gets the color of the pixel above it in the “Recreation table” The example for which is show below. Note that the “Resultant block” comes into picture only when both the table have the data as “1”. For all other conditions the “Recreation box” itself acts as the source for the pixels. The illustration for the third condition is shown below. The processing is carried out in the similar fashion; every bit

<table>
<thead>
<tr>
<th>1</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The second condition is that when the bit in both the SBA table is “0” then the pixel gets the color of the pixel on the left in the recreation block. The example for which is shown below

<table>
<thead>
<tr>
<th>1</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The third condition is encountered when the bit in the first SBA table is “1” and the second table has the value as “0”. In such a case the pixel gets the color of the pixel above it in the “Recreation table” The example for which is show below. Note that the “Resultant block” comes into picture only when both the table have the data as “1”. For all other conditions the “Recreation box” itself acts as the source for the pixels. The illustration for the third condition is shown below. The processing is carried out in the similar fashion; every bit in the SBA table is interpreted based on the decompression rules described earlier. Thus ultimately the transmitted image can be regenerated as follows

Recreated Image

C. Compression Criteria

After every compression process the algorithm checks if the criteria is satisfied which as given below

\[(Siz\_after + OH) < Siz\_before\] only If this criteria is satisfied the compression process carried out.

D. Exponential Reduction:

The compression process can be further be aggravated by a process of exponential reduction where your “Resultant block” can be further reduced by the same compression process until the
sand criteria is broken. Not only the pixel can be compressed but also the table can be reduced the table which is in the binary format can be grouped into “W” bits (24 bits) to form a colored pixel which can be rearranged and then can be compressed to reduce the total overhead. Further more like the longitudinal and latitudinal compression cross compression is also possible. The type of compression to be adopted depends the arrangement of the pixels and the efficiency it yields.

III. CONCLUSION

The abstract presented here gives a overview of the compression idea developed by me. The Sand algorithm is a promising idea waiting to be tapped practically. The algorithm has extensive scope in various fields of research and analysis. Though this algorithm cannot match the efficiency of other lossy compression formats like (JPEG). It would be an ideal choice in areas where quality of information cannot be compensated for the cost. This algorithm will enable us to cut cost and increase the rate of transmission of images without any adverse effect on the quality of the image.

REFERENCES


AUTHORS

K.Vikram is working as a Vice-Principal at Medak College of Engineering and Technology (MCET), Hyderabad, India. He has received MCA and ME.(Computer Science and Engineering) from Anna University. Presently, he is a Research Scholar. He has published and presented good number of technical papers in National and International Conferences. His main research interests are Software Engineering, Data Mining, and Network Security.

Dr. Niraj Upadyay is working as a Dean of CSE Dept. at JBIET. He has received B.Tech and MS.(Computer Science and Engineering) from BITS and Ph.D. He has published and presented good number of technical papers in National and International Conferences. His main research interests are Software Engineering, Data Mining, and Ne
Dr. Govardhan has been conferred CSI Chapter Patron Award (2010-2011), Shining Image of India Award 2011, Seva Chakra Puraskar 2011, Bharat Jyoti Award 2010, Rajiv Gandhi Excellence Award 2010, Best Citizens of India 2010 Award, Life Time Achievement Gold Medal Award 2010, Rashtriya Vidya Saraswati Puraskar 2010, Eminent Educationist Award 2010 and Certificates of Excellence for Outstanding services, Achievements, Contributions, Meritorious Services, Outstanding Performance and Remarkable Role in the field of Education and Service to the Nation He has been listed as One among the • Top Three Faculty in JNTU Hyderabad, made by Outlook Survey for the year 2008 • He was awarded Telugu Vignana Parithoshikam, by the Government of Andhra Pradesh for B.E(CSE) program • He was nominated as the Best Student for the year 1987-88 in A.P. Residential Junior College (APRJC), Nagarjuna Sagar, A.P. • He was awarded Merit Scholarship by the Government of Andhra Pradesh for securing first position in School. He was awarded PADMA JYOTHI for securing First Position in School in 10th standard. • He is a Member in Executive Council, JNTUH, He is a member of Standing Committee for Academic Senate, JNT University Hyderabad and Academic Advisory Committee (AAC), UGC-Academic Staff College and Sports Council, JNT University Hyderabad • He is the Chairman for Post Graduate Board of Studies (BOS) in Computer Applications, Yogi Vemana University, Kadapa, Member on various BOS (for CSE and IT, UG and PG), including JNT University Hyderabad, Adikavi Nannaya University, Rajahmundry, Satavahana University, Karimnagar and VR Siddhartha Engineering College, Vijayawada and Sree Nidhi Institute of Science and Technology, Hyderabad. • He is the Vice-Chairman for CSI Hyderabad Chapter • He is a Member on the Editorial Boards for Eight International .Mrs. Thoomati Madhavi Kumari obtained B.Tech. in Electronics & Communication Engineering and M.Tech. in Computer Science from JNT University. Presently Mrs. Madhavi is pursuing Ph.D. in Data security for natural languages using FPGA. Mrs. Madhavi joined the faculty of ECE Department of JNU College of Engineering, Kukatpally, Hyderabad as Assistant Professor and served the department for 13 years. Later she was appointed as Assistant Director, UGCASC, JNT University, Hyderabad Mrs. Madhavi, in the ECE Department, conducted two faculty development programs in collaboration with IBM and Intel. In the department, she was associated with the implementation of AICTE sponsored project on computer networks. Mrs. Madhavi guided many M.Tech. and MCA