ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.2 , No.1, Pages : 383 – 387 (2013)
Special Issue of ICACSE 2013 - Held on 7-8 January, 2013 in Lords Institute of Engineering and Technology, Hyderabad

Performance Evaluation of Basic Compression Technique for Wireless Text Data

V. Bhagya Raju[1], Dr K. Jaya Sankar[2], Dr C.D. Naidu[3]

[1]Prof & HOD, ECE Dept
Vidya Vihar Institute of Technology
Andhra Pradesh, India.
vbhagya01@gmail.com

[2]Prof & HOD, ECE Dept
Vasavi College of Engineering
Andhra Pradesh, India.
kottareddyjs@gmail.com

[3]Principal
VNR VJIET
Andhra Pradesh, India.
principal@vnrvjiet.ac.in

Abstract: **The continuous growth of mobile, desktop and wired and wireless digital communication technologies has made the extensive use of the text data unavoidable. The basic characteristics of text data like transmission rate, bandwidth, redundancy, bulk capacity and co-relation among text data makes basic compression algorithms mandatory. The research exploration in the field of text data compression is huge. In this research paper we investigate the performance evolution of basic compression algorithms on text data. We are adopting RLE (Lossless) compression and its modified version of algorithm, named K-RLE (Lossy). The basic and proposed system architecture, design, complexity, and performance could be analyzed and compare using MATLAB Language, which is the quick tool to estimate performance of the system. The KRLE compression algorithm is basically used to compress the repeated unwanted data. This has implemented using ARM7 low cost embedded processor. To implement this project on the processor the external devices like 16X2 LCD, RESET circuit, RS-232 cable have interfaced to this controller.**

Keywords: **KRLE, Data compression, Lossy, MATLAB**

1. INTRODUCTION

The rapid growth of multimedia and networking technologies gives rise to numerous multimedia applications such as mobile, desktop, internet and video surveillance, satellite communication and webcams, consequently multimedia transmission has become a challenge issue. Due to the unique characteristics of real time wireless sensor text data such as large data size, high bandwidth and stringent real time requirements [1].the researchers have been forcing to use the proper compression algorithm to enhance the overall performance (compression ratio, saving percentage, compression time, entropy and code efficiency) of the system should be selected carefully for real time text transmission. Text compression is specialized discipline of electronic engineering as been gaining considerable attention on account of its applicability to various fields. Compression is art of representing information in compact form rather than it's original. Using the data compression method, the size of particular file can be reduced. Compressed transmission economizes bandwidth, computation and transmission--power, cost, and latency and therefore ensures cost-effectiveness during transmission. the application areas for such compression today range from mobile, TV and broadcasting of HD-TV up to very high quality applications such as professional digital video recording or digital cinema / large screen digital imagery and so on this as lead to enhanced interest in developing tools an algorithms for very low bit rate image coding and image quality [2].

1.1 COMPRESSION USING ARM 7 LOW COST EMBEDDED PROCESSOR

The LCD is interfaced to P1 port of the LPC2148 (P1.16 to P1.21) in 4 bit mode in which nibble by nibble DATA will be sent to the LCD. To receive the data entered from the PC the UART communication is used. To establish that communication RS-232 cable has connected through MAX 232 IC to the P0.0, P0.1 pins of the LPC2148 controller. The MAX IC placed in this board is used to match the RS-232 voltage levels to TTL logic levels to establish the proper communication between PC and processor.

The active RESET circuit has been designed to this processor is used to reset the total circuit whenever we want.
The working of total circuit is as fallows when we switch on the power supply the 3.3v goes to processor internal operation & the 5v for external peripherals like LCD, other devices. First In LCD the messages will be displayed like project name and student name and asks for the data to be entered.

When we entered the data the compression will start in very next step and displays the compressed data and decompressed data in all categories as shown above for the different data i.e. repeated data, partially repeated data, non repeated data. After completion of all these operation observed on the hardware modules we can conclude that the compression ratio is go on increases as precision factor increases and as well as increases the loss of data as shown in graphical representation.

1.2 IMAGE COMPRESSION

The purpose of compression is to code the image data into a compact form, minimizing both the number of bits in the representation, and the distortion caused by the compression .the importance of image compression is emphasized by the huge amount of data in raster images, a typical gray-scale image of 512 x 512 pixels, each represented by 8bits, contains 256 kilobytes of data. With the color information, the number of bytes is tripled. The video images of 25 frames per second, even a one second color film requires approximately 19 Megabytes of

Memory. To handle and process the above said data representation definitely one has to think of how to represent in terms of the encoded data the method is called compression, obliviously this technique becomes mandatory for any kind of present day digital image data processing.

2. Related Work:

Till now many scientists, research scholars, Engineers proposed many data compression algorithms that compress almost any kind of data, In that the best know are the family of ZIV-Lempel algorithms. If the method is lossless they retain all the information of the compressed data, they doesn't take advantage of the 2-D nature of the image data. Only small portion of the data space can be saved by a lossless compression method. now a day's lossy techniques are widely used in image compression, because they produce high compression ratio and saving ratio, of course there may be image quality degradation when reproduction of original image from compressed image, however the image quality could be improved by selecting the appropriate compression technique / algorithm based on the application requirements In lossy compression, always there would be tradeoff between the bit rate and the image quality.

A common characteristic of most images is that the neighboring pixels are correlated and therefore contain redundant information. Two fundamental components of compression are redundancy and irrelevancy. Redundancy reduction aims at removing duplication from signal source. Irrelevancy reduction omits parts of the signal that will not be noticed by the signal receiver. There are three types of redundancies, they are spatial redundancy means correlation among neighboring pixel values, coding redundancy is used when less than optimal code words are used, spectral redundancy means correlation between color planes and temporal redundancy means correlation between adjacent frames.

Image compression techniques reduce the number of bits required to represent an image by taking advantage of these redundancies. An inverse process called decompression is applied to the compressed image data to get the reconstructed image. The two main distinct structural blocks of typical image processing system are an ***encoder*** and a ***decoder*** as shown in figure 1. Image f(x,y) is fed into the encoder, which creates a set of symbols form the input data and uses them to represent the image. If we let n1 and n2 denote the number of information carrying units(usually bits) in the original and encoded images respectively, the compression that is achieved can be quantified numerically via the compression ratio.

$$CR = n1 / n2$$

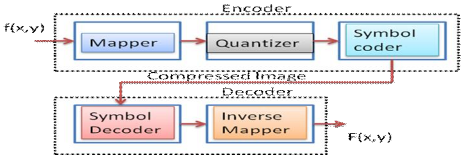


Figure 1

As shown in the figure 1, the encoder is responsible for reducing the coding, interpixel and psychovisual redundancies of input image. In first stage, the mapper transforms the input image into a format designed to reduce interpixel redundancies. The second stage, quantizer block reduces the accuracy of mapper's output in accordance with a predefined criterion. In third and final stage, a symbol decoder creates a code for quantizer output and maps the output in accordance with the code. These blocks perform, in reverse order, the inverse operations of the encoder's symbol coder and mapper block. As quantization is irreversible, an inverse quantization is not included in the figure 1.

The typical parameters, which are used to measure performance of the lossy image compression techniques / algorithms.
Compression Ratio is the ratio between the size of the compressed file and the size of the source file.
Compression Factor is the inverse of the compression ratio. That is the ratio between the size of the source file and the size of the compressed file.
Saving Percentage calculates the shrinkage of the source file as a percentage.
SP = (size before compression – size after compression) / size before compression

Compression Time

Time taken for the compression and decompression should be considered separately. Some applications like transferring compressed video data, the decompression time is more important, while some other applications both compression and decompression time are equally important. If the compression and decompression times of an algorithm are less or in an acceptable level it implies that the algorithm is acceptable with respective to the time factor. With the development of high speed computer accessories this factor may give very small values and those may depend on the performance of computers or machines.

3. Sensor Text Data Compression Techniques

The text compression techniques are broadly classified into two categories, they are: Lossless techniques and Lossy techniques. In our research work we consider one lossy compression (K-RLE) and one lossless compression algorithms (RLE) on hyper text data and data retrieved from temperature sensors and LDR.**For category1 (when K=0):**
Series 1 (For repeated data):
The output for this repeated data as shown below COMPRESSED DATA: 104204304404(8 bytes of data). The compression is high in this case
Series 2 (For partially repeated data):
The output for this partially repeated data as shown below COMPRESSED DATA: 103203303403503601(12 bytes of data).
The compression is low when compared to above case in this case

Series 3(For non- repeated data):

The output for this non repeated data as shown below COMPRESSED DATA: 101201301401501601701801901A01B01C01D01E01F01 (16 bytes of data). There is no compression in this case.

For category2 (when K=1):

Series 1 (For repeated data):

The output for this repeated data as shown below COMPRESSED DATA: 108308 (4 bytes of data) the compression is very high in this case when compared to category1

Series 2 (For partially repeated data):

The output for this partially repeated data as shown below COMPRESSED DATA: 106306504 (6 bytes of data) The compression is low when compared to above case in this case.

Series 3(For non- repeated data):

The output for this non repeated data as shown below
COMPRESSED DATA: 102302502702902B02D02F01 (16 bytes of data)
There is a little bit compression in this case.

For category2 (when K=2):

Series 1 (For repeated data):

The output for this repeated data as shown below COMPRESSED DATA: 10C404 (4 bytes of data) the compression is very high in this case when compared to category1

Series 2 (For partially repeated data):

The output for this partially repeated data as shown below COMPRESSED DATA: 109407 (4 bytes of data). The compression is low when compared to above case in this case.

Series 3(For non- repeated data):

The output for this non repeated data as shown below
COMPRESSED DATA: 103403703A03D03 (10 bytes of data)
There is a little bit BETTER compression in this case.

For category2 (when K=3):

Series 1 (For repeated data):

The output for this repeated data as shown below COMPRESSED DATA: 110 (2 bytes of data) the compression is very very high in this case when compared to previous all categories.

Series 2 (For partially repeated data):

The output for this partially repeated data as shown below COMPRESSED DATA: 10C504 (4 bytes of data). The compression is low when compared to above case in this case.

Series 3(For non- repeated data):

The output for this non repeated data as shown below
COMPRESSED DATA: 104504904D03 (8 bytes of data). There is a little bit BETTER compression in this case. When compared to previous categories. The overall comparison is as fallows as precision factor increases compression ratio increases as located in graphical representation.

3.1 LOSSLESS COMPRESSION TECHNIQUE

In lossless compression techniques, the original image/data can be perfectly recovered from the compressed (encoded) image/data. These are also called noiseless since they do not add noise to the signal (image).It is also known as entropy coding since it use statistics or decomposition techniques to eliminate or minimize redundancy. Lossless compression is used only for a few applications with stringent requirements such as medical imaging and sensor data processing, ***In our research work we consider the basic lossless compression technique named as Run Length Encoding (RLE),*** earlier researchers implemented the RLE compression algorithm on low cost, low power tinny embedded systems (based on 8bit/16bit microcontrollers) using ALP and respective EC programming for slowly varying sensor data for wired and wireless sensor networks (WSN) [4]. Even they evaluated the performance of RLE on Reconfigurable FPGA Architecture for above said applications [5]. Probably no one analyzed design exploration of image data compression using RLE. In our research work we analyze and evaluate the performance of RLE compression algorithm for image data applications based on MATLAB EDA Tools

3.1.1 RUN – LENGTH ENCODING:

- The Idea behind this algorithm is, If a data item d occurs n consecutive times in the input data we replace the n occurrences with the single pair nd.
- Run-Length Encoding (RLE) is a basic compression algorithm. It is very useful in case of repetitive and slowly varying data items.
- This is most useful basic compression algorithm on data that contains many such runs: for example, relatively simple graphic images such as icons, line drawings, and grayscale images.
- Which is a lossless data compression algorithm used for slowly varying sensor and image data.
- It is not useful with files that don't have many runs as it could double the file size.

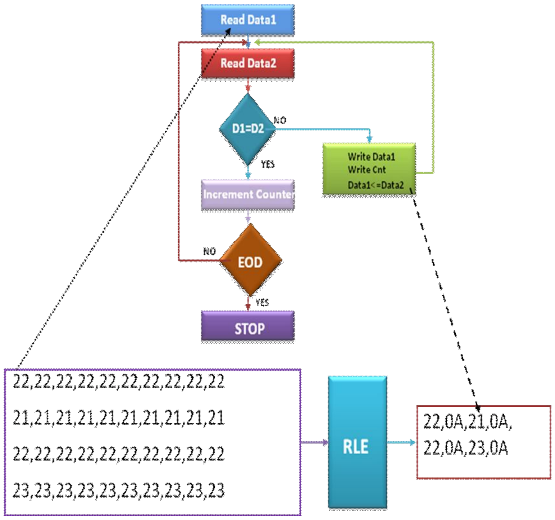


Figure 2: Flow Chart for Run Length Encoding:

3.2 LOSSY COMPRESSION TECHNIQUE

Lossy schemes provide much higher compression ratios than lossless schemes. Lossy schemes are widely used since the quality of the reconstructed images is adequate for most applications .By this scheme, the decompressed image is not identical to the original image, but reasonably close to it.

3.2.1 RUN LENGTH ENCODING WITH K-PRECISION:

The idea behind this new proposed algorithm is this: let K be a number, a data item d or data between d+K and d-K occur n consecutive times in the input stream, replace the n occurrences with the single pair nd. We introduce a parameter K which is a precision.

- If K = 0, K-RLE is RLE. K has the same unit as the dataset values, in this case degree.
- K-RLE is a lossy compression algorithm.
- This algorithm is lossless at the user level because it chooses K considering that there is no difference between the data item d, d+K or d-K according to the application.

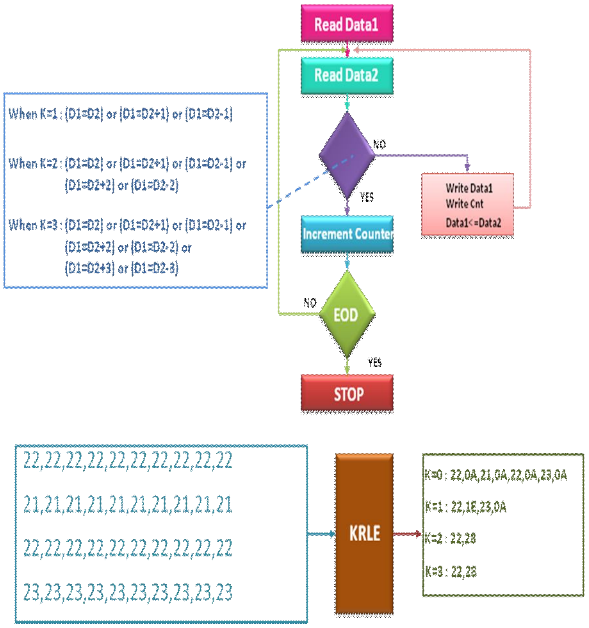


Figure 3: Flow Chart for Run Length Encoding with K - Precision:

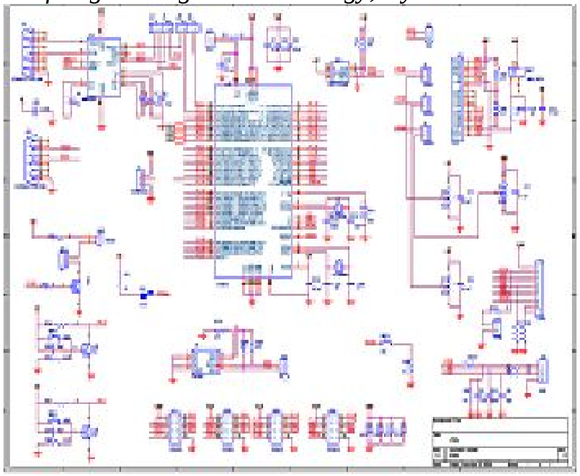


Figure 4: Schematic of ARM7 Processor with wireless sensor networks.

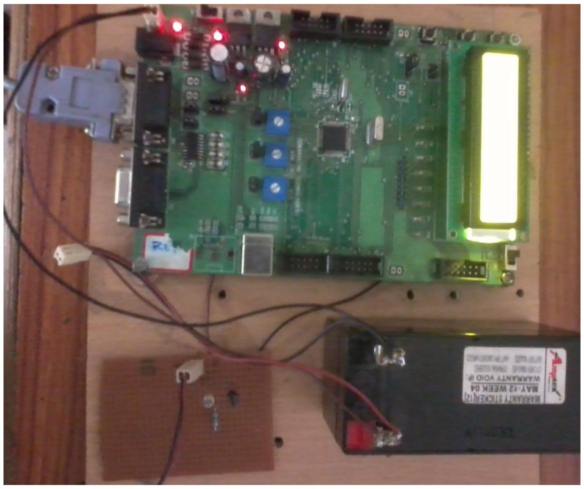


Figure 5: Development Board of ARM7 Processor with Wireless Sensor Networks

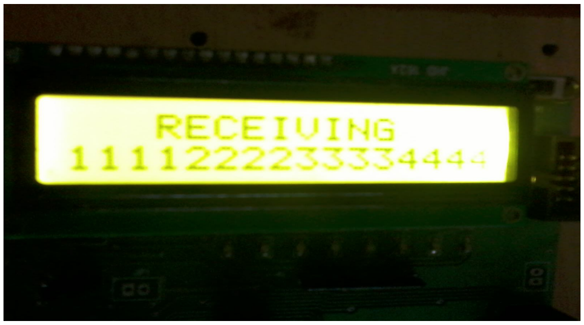
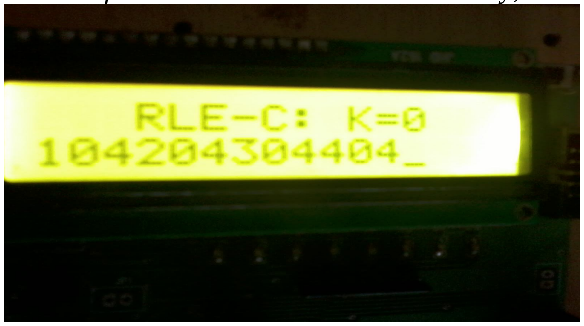
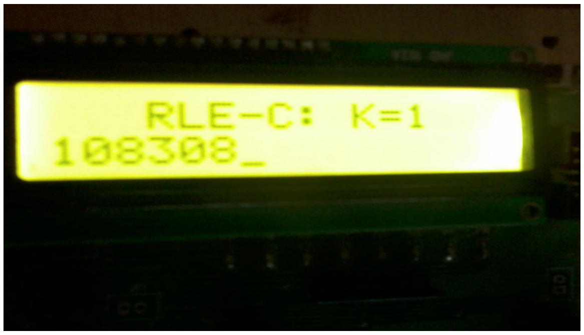


Figure 6: Photo Shot of Receiving Wireless Sensor Data ON LCD



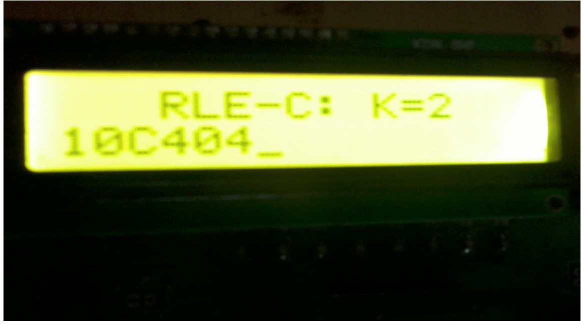
```
RLE-C: K=0
104204304404_
```

Figure 7: Photo Shot of Wireless Text Data Compression for K=0



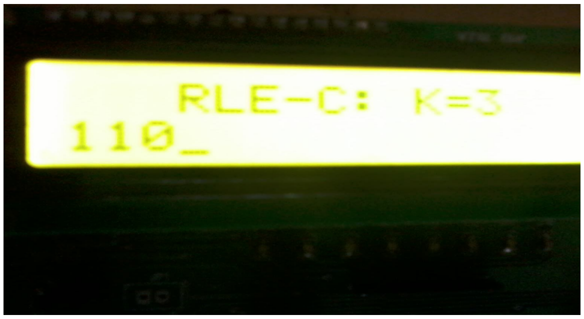
```
RLE-C: K=1
108308_
```

Figure 8: Photo Shot of Wireless Text Data Compression for K=1



```
RLE-C: K=2
10C404_
```

Figure 9: Photo Shot of Wireless Text Data Compression for K=2



```
RLE-C: K=3
110_
```

Figure 10: Photo Shot of Wireless Text Data Compression for K=3

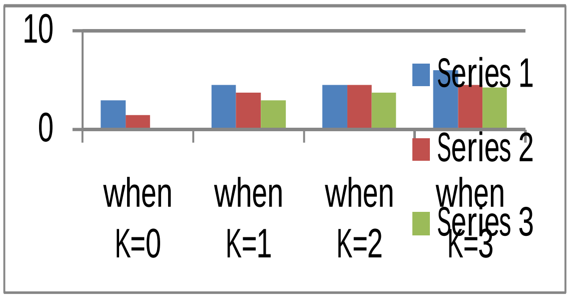


Figure 11: Compression of Text Data for Various Values of K

The data we are taking in order to compare the compression in all formats i.e when K-0,K=1,K=2,K=3 etc..

DATA 1: 1111 2222 3333 4444(16 bytes of repeated data)
DATA2:1112223334445556(16 bytes of partially repeated data)
DATA3:123456789ABCDEF (16 bytes of non repeated data)

REFERENCES

[1] Jayavrinda Vrindavanam, Saravanan Chandran and Gautam K. Mahanti, 2012, "A survey of image compression methods," *IJCA*.

[2] S.R Koditvwakku and U.S. Amarasinghe, "Comprasion of lossless data compression algorithms for text data," *IJCSE*, vol 1 no 4 410 – 425.

[3] B. Subramanyan, Vivek.M.Chhabria and T.G Sankar Babu, 2011, "Image encryption based on aes key expansion," *IEEE*.

[4] Eugene pamba cupo – Chichi, Herve Guyennet and Jean – Michel friedt, 2009, KRLE – "A new data compression algorithm for wireless sensor network," *IEEE*.

[5] Wail s. Elkilani, Hatem M. Abdul – Kadar, 2009, "Performance of encryption techniques for real time video streaming," *IEEE*.

[6] K. Arshak, E. Jafer, D. Mcdonagh and C.S.Ibala, 2007, "Modelling and simulation of wireless sensor system for health monitoring using HDL and simulink mixed enviroment," *IET computer*. *Digital tech* vol 1. No 5, September.