DATA COMPRESSION

PROJECT WORK DONE AT

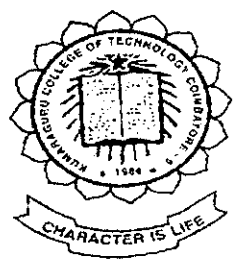
SakthiSri Infotech Pvt Ltd, Coimbatore

PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF
MASTER OF COMPUTER APPLICATIONS
OF BHARATHIAR UNIVERSITY, COIMBATORE.

SUBMITTED BY
P.Rajendhiran
Reg.No : 9938M0625

GUIDED BY
Mr.K.Sivanarulselvan M.Sc.,M.C.A.,PGD P.M.I.R.,M.Phil



Department of Computer Science & Engineering
Kumaraguru College of Technology
Coimbatore – 641 006

May - 2002

CERTIFICATE

Department of computer Science & Engineering

Kumaraguru College Of Technology

(Affiliated to the Bharathiar University)

Coimbatore – 641 006

CERTIFICATE

This is to certify that the project work entitled
DATA COMPRESSION

Done by

P.Rajendhiran
Reg.No : 9938M0625

Submitted in partial fulfillment of the requirements for the award of the degree of
Master of Computer Applications of Bharathiar University.

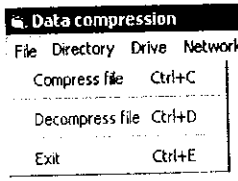
Professor and Head

Internal Guide

Submitted for the University Examination held on 09.05.2002

Internal Examiner

External Examiner



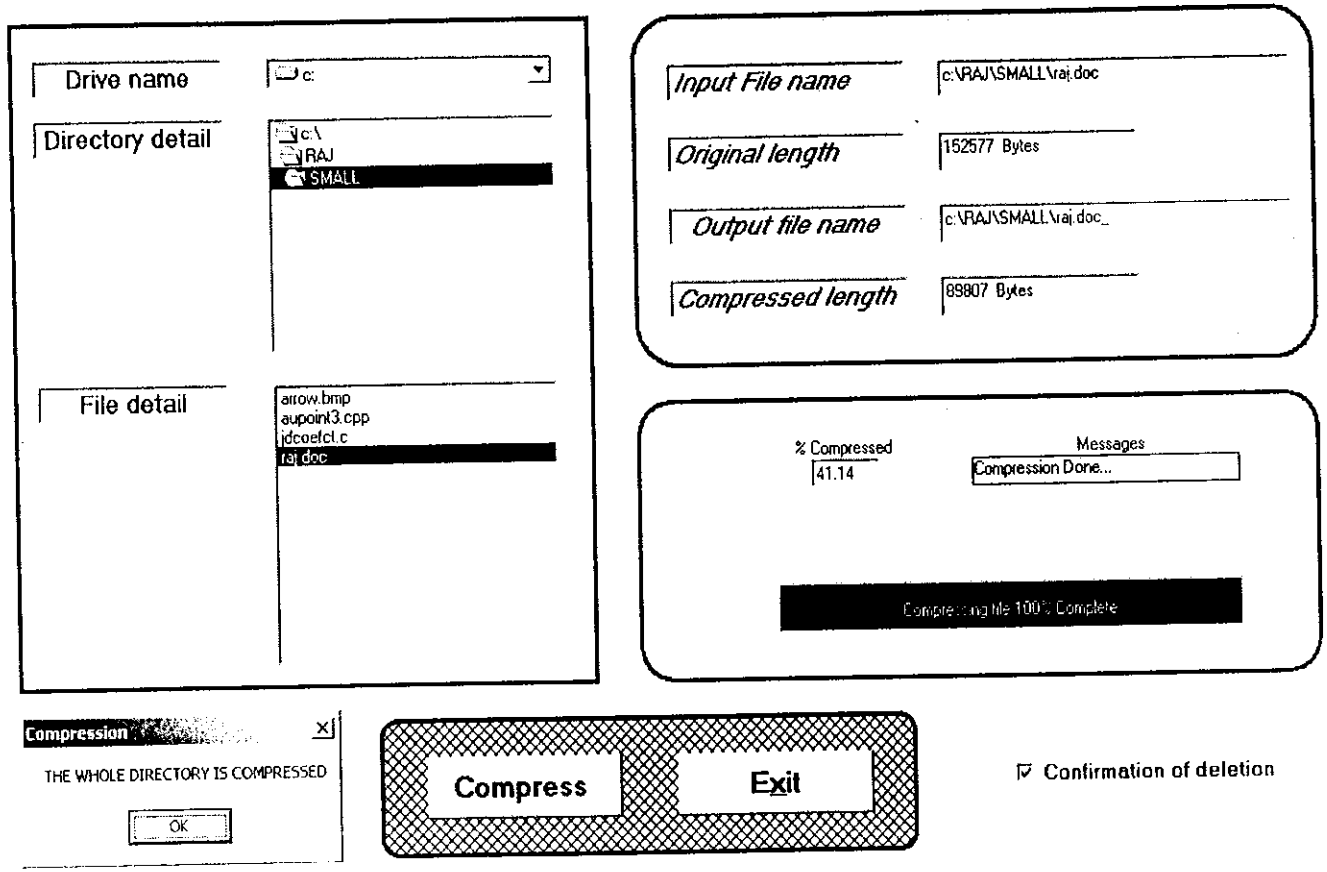
A PROJECT ON COMPRESSION OF DATA

Developed

at

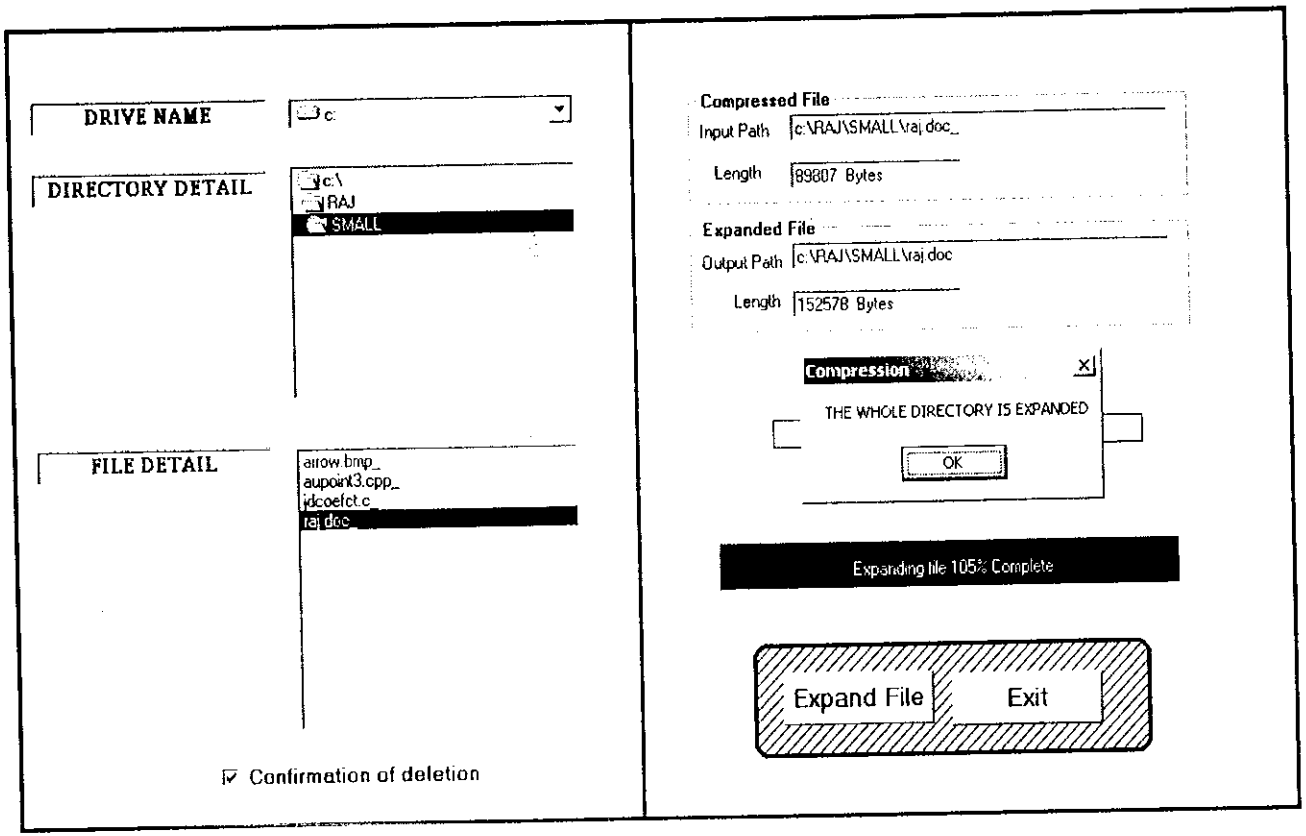
SAKTHISRI INFOTECH (P) Ltd.

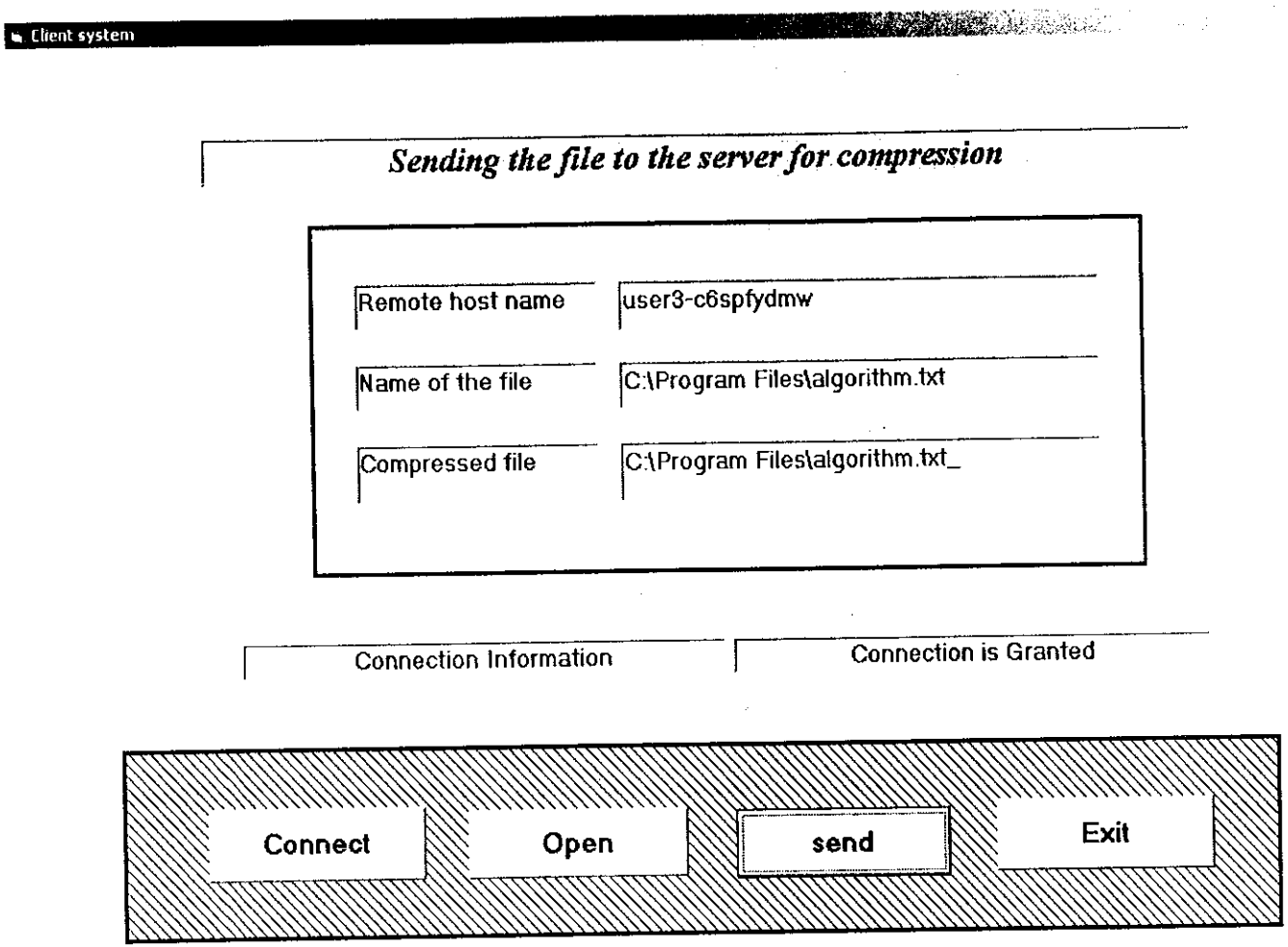
Compression of the particular directory



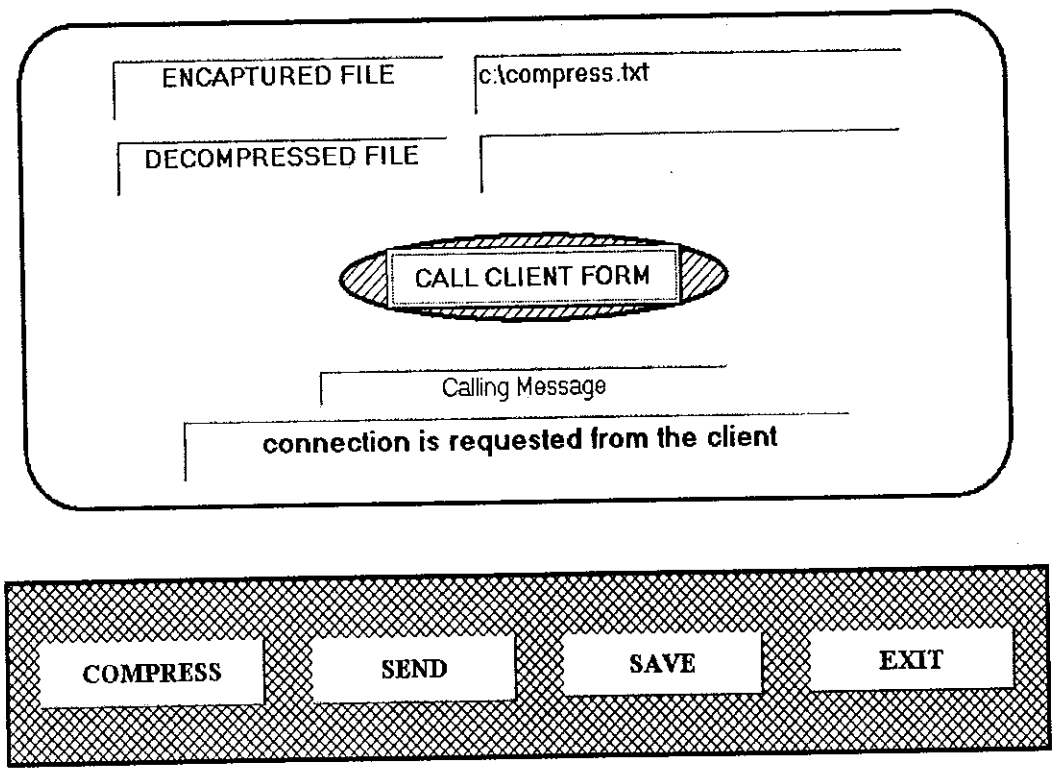
Directory decompression

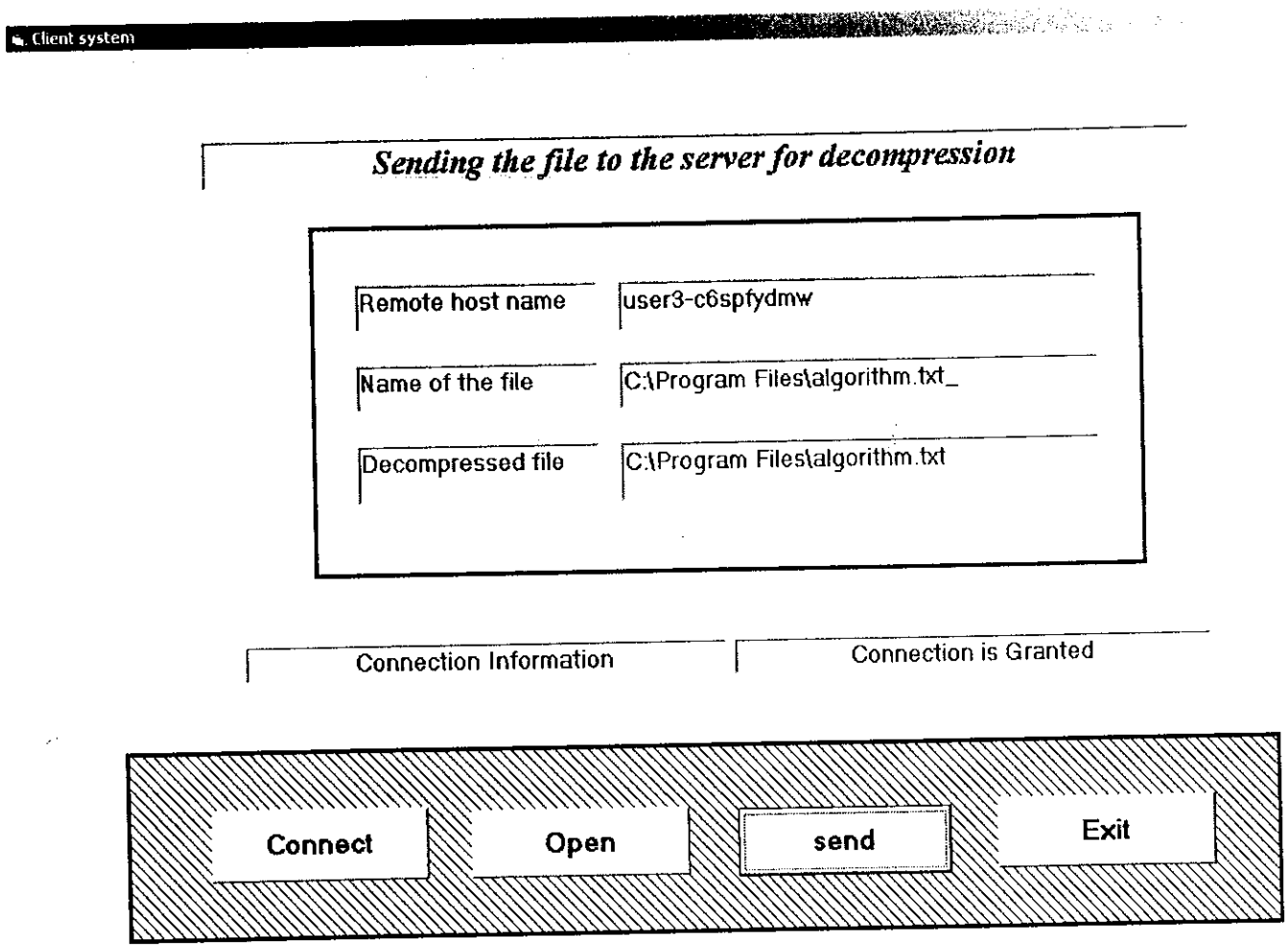
Decompression of the particular directory




Client system
Sending the file to the server for compression
Remote host name
user3-c6spfydmw
Name of the file
C:\Program Files\algorithm.txt
Compressed file
C:\Program Files\algorithm.txt_
Connection Information
Connection is Granted
Connect
Open
send
Exit

Encapturing the file from the client for compression




Client system
Sending the file to the server for decompression
Remote host name
user3-c6spfydmw
Name of the file
C:\Program Files\algorithm.txt_
Decompressed file
C:\Program Files\algorithm.txt
Connection Information
Connection is Granted
Connect
Open
send
Exit

SakthiSri Infotech

TO WHOMSOEVER IT MAY CONCERN

This is to certify that Mr.P.RAJENDARAN final year [illegible] (Master of Computer Applications) student of Kumaraguru College of Technology, Coimbatore has successfully completed the project titled "DATA COMPRESSION" during the period January 2003 to April 2003.

Since the source code is of strict confidentiality it cannot be provided in any format.

We wish him all success in future endeavors.

For Sakthisri Infotech Pvt. Ltd.

V. Pradeep Kumar
Project Co-ordinator

DECLARATION

DECLARATION

I hereby declare that the project entitled **"DATA COMPRESSION"**, 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.K.Sivanarulselvan** M.Sc.,M.C.A.,PGD P.M.I.R., M.Phil, Lecturer of **Kumaraguru College of Technology, Coimbatore** and **Mr.V.Pradeep** B.E, **of SakthiSri Infotech Pvt Ltd, Coimbatore** and this project work has not found the basis for the award of any Degree/Diploma/Associate ship/ Fellowship or similar title to any candidate of any university.

Place: Coimbatore

Date : 30.04.2002.

Raamdhaip

Signature of the Student

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

I express my heartful thanks to **Dr.K.K.Padmanabhan, B.Sc(Engg)., M.Tech., Ph.D.,** Principal, Kumaraguru College of Technology, for having given me the opportunity to serve the purpose of my education

I am indebted **to Dr.S.Thangasamy, Ph.D** Professor & Head of Department Of Computer Science and Engineering, for his valuable guidance and useful suggestions during the course of project.

I am deeply indebted to my project Guide **Mr.K.Sivanarulselvan M.Sc., M.C.A., PGD P.M.I.R., M.Phil.,** Lecturer of Department of computer Science and Engineering, Kumaraguru College of Technology, for his helpful guidance and valuable support given to me throughout the project.

With immense pleasure, I express my esteemed gratitude to **Mr.N.Venkatesan,** Managing director of Sakthisri Infotech Pvt Ltd, for providing me the opportunity to do the project in reputed organization.

I take privilege of expressing my sincere thanks to my external guide **Mr.V.Pradeep Kumar B.E,** for his keen interest and efforts in guiding and encouraging me throughout the project and also providing all necessary resources needed for the project in the organization.

I would like to thank all the staff members of the Department of Computer Science and Engineering of my college, for their constant encouragement and guidance throughout the course.

P.Rajendhiran

SYNOPSIS

SYNOPSIS

This project entitled **"DATA COMPRESSION"** has been developed for sakthisri infotech private limited at coimbatore.

Data compression is a term that refers to the process of transforming a body of data to a smaller form, from which the original can be restored.

Data compression has many uses, especially in modern networking environments. File transfer time can be significantly reduced by compressing large files. This might prove useful for retrieving database information within a company's internal network, or for downloading images on a internet web page. Compression is also useful for backing up files as it reduces the space required for information storage. This system consist of five menu options . They are

1. File
2. Directory
3. Drive
4. Network
5. Small data compress

Every option having the two sub options that are compress and decompress.

The Huffman algorithm has been implemented in this software. It is names after D.A. Huffman, who published a paper in 1952 entitled "A Method for the construction of Minimum Redundancy Codes" describing the following algorithm.

Huffman compression is a statistical data compression technique which gives a reduction in the average code length used to represent the symbols of an alphabet. The Huffman code is an example of a code which is optimal in the case where all symbols probabilities are integral powers of 1/2. A Huffman code can be built in the following manner:

1. Rank all symbols in order of probability of occurrence.
2. Successively combine the two symbols of the lowest probability to form a new composite symbol; eventually we will build a binary tree where each node is the probability of all nodes beneath it.
3. Trace a path to each leaf, noticing the direction at each node

The file option is used to compress and decompress the single file ,in the same way the other options are used compress or decompress the directory and drive.

The Networking option is the very useful thing in this package. It is not necessary to keep this package in all of systems which are under networking. It is sufficient to keep the package only on the server. we can retrieve the file from the client systems using this network option. And also we can compress or decompress the file and send to the particular client system. So we can save the lot of memory areas .

The main disadvantage of this algorithm is that this one can not compress the small files (size less than 1 kb) with maximum compression ratio. So the fifth and final option is used to solve this problem. The new algorithm is developed for this one.

The idea behind this new algorithm is to build a dictionary of 160 common letter pairs. As the package scans the file to compress, it tries to find pairs of letters in the dictionary. If it succeeds, the program stores the index of the pair in the uncompressed file. If the pair isn't in the dictionary, the algorithm stores the first letter in the uncompressed file. The algorithm continues processing the file until it has all been compressed. To decompress the file it will follow the steps reversely.

This Data Compression software has been developed using Visual Basic 6.0 Professional Edition under Windows 98 Operating System.

CONTENTS

CONTENTS

INTRODUCTION

1. INTRODUCTION

1.1 Project Overview

This project entitled "Data compression" is very useful to compress or decompress the file, directory, drive. The data compression technique can be classified as Lossy or Lossless based on the way they achieve compression. In the case of lossless coding, Compression does not result in any loss of information in the original data. This is unlike lossy coding techniques that loss certain amount of information for achieving better compression ratio. These lossy coding can be used in applications where loss of information is tolerable and high compression is required. A lossless coding technique is applied where the information content of the file has to be retained .

The example for the loss less compression algorithms are Huffman compression and LZ77 compression .JPEG is the example for the lossy compression program.

So the Huffman algorithm is the lossless algorithm. So we can compress any type of file with out the loss of its single information. By using this algorithm we can compress the files with the minimum ratio of 30%. And we can compress the files with the maximum ratio of 80%.

This Software involves following steps,

- Choosing the Application among File, Directory, Drive, Network and Small data compress .
- If it is File, select the file to be Compressed and set the path to store the Compressed file. And give the detail of the deletion of the original file. Compress the Selected file by using specific options.
- If it is a Directory than select the Drive from the particular drive .After selecting the Directory give the confirmation of the deletion of the original file .Than here it is not necessary to give the output path .It will automatically put the compressed file in the specified path. The option Drive is the same one as Directory.

- The option Network is used to compress the client system files from the server. Here we send the file from client system with specified input path and the output path to save the compressed data . The server will compress the client file and compressed file will be sent to client by server.
- Small data compress is the final option that is used to compress the data with the size of less than 700 bytes. Here also we give the input path of the file to be compressed and output path to store the compressed data.

1.2 Need for computerization

Compression is the art of significantly reducing the physical size of a block of information offering the following capabilities such as

(i) Storing more information on the same media.

(ii) Reducing the transfer time of data on a network.

(iii) Improving the usage and reproduction of file with minimum degradation.

Electronic Transmission of uncompressed data can be very time consuming. Sending a 10 MB file would take approximately 45 minutes in a conventional phone line and a modem operating at 28,800 bits per second . Data compression technique reduce the size (and hence cost) of any archive and can decrease transfer time.

Data compression operations reduce the data content of file just before the storage or transmission.

1.3 Huffman coding

Huffman code takes the advantage of the statistical redundancy present in the source. Instead of uniformly assigning the bits to all the symbols as the case in standard pulse code modulation, here the bit allocation depends on the probability of the occurring symbols. Less bits are allocated to the symbols occurring more frequently and more bits are allocated for the bits occurring less frequently, there by reducing the number of bits

transmitted for a given file. Huffman has devised an algorithm for generating variable length code words for a given source symbol probabilities. This algorithm is optimum in the sense that average number of binary digits required to represent the source symbols in a minimum. It also forms code words that satisfy the prefix condition, i.e., no code word is a prefix to another codeword. This condition ensures unique decidability of the received sequence.

Description of the algorithm

Huffman algorithm uses a frequency table and data structure called binary tree for constructing a variable length code. Here , the binary tree is referred to as code tree. In a code tree each node consists of two child nodes or none. The nodes with no child nodes are called leaves of the tree and they represent the symbols. These nodes are linked to one another based on the probabilities. i.e., frequencies of the symbols .The Huffman tree is so formed that the bits allocated for the most probable symbol is the least and is maximum for the least probable symbol. This results in shorter code for the message or the image to be coded . The procedure to construct the huffman code tree is described below.

The first step is to calculate the frequencies of the given file. A file consists of intensity values ranging from 0-255. So the frequency table is formed by counting , how many tomes each intensity value is occurring in the file. These form the leaves of the code tree. It is convenient to arrange the symbols in ascending or descending order. The next step is to combine the two symbols . A '0' is assigned to the left child and a '1' is assigned to the right child to differentiate these child nodes. Now this parent node will have the sum of the frequencies of the child nodes and the symbol set is lee by one. The above step is carried out repeatedly on the resultant parent nodes and the symbols , while allocating one bit to the child nodes formed at every step, until all the symbols are reduced to a single super symbol. This the root of the code tree which represents the message itself and will have a frequency value that is the sum of the frequencies of all the symbols in the given file.

To form the codeword for a given symbol, the tree is traversed for the root to the leaf having that symbol while reading off the bils assigned to the branches in the path from the root to the leaf. This ensures that no code word will have another symbol code word as its prefix. And also while constructing the tree, the lowest frequency nodes are combined to form a parent node by adding one level of branching .This makes the path length of the low frequency symmbolc longer and hence longer code word length for low probability symbols.

1.4 Organization Profile

The company Sakthisri Infotech Pvt limited is one of the leading software consultancies in Coimbatore. Mr.N.Venkatesan is the Managing Director of this company. He had been in the computer field for the past 15 years.

The organization has developed software for various concerns located in and around coimbatore. The organization has more than 200 clients for whom they develop various software.

The organization currently has 18 staff members. In these, three persons are in sales , two in customer support and ten in software development.

The main focus of Sakthisri Infotech Pvt limited is the business of software training, software development and consultancy services, Projects and Products.

Technology Services

Sakthisri Infotech Pvt Ltd provides project management and contract services for the analysis, design, programming, and implementation of Internet Commerce, Client Server, and enterprise-wide business applications. They develop Internet and data communications systems along with legacy systems to GUI interface middleware applications that utilize the latest Java based technologies. They have experience with a wide range of platforms, networks, databases, and languages.

Staffing Services

Sakthisri Infotech Pvt Ltd recruiting process starts with a complete understanding of the needs which are then entered into the system which scans a number of databases for qualified matches, then they contact the candidates, ensure their resume and skills are accurate and up to date, establish their interest in your position, and agree on the terms and availability. Sakthisri Infotech Pvt Ltd performs technical interviews and follows up with a minimum of five reference checks.

SYSTEM STUDY AND ANALYSIS

2. SYSTEM STUDY AND ANALYSIS

2.1 Existing system - Limitations

In Kaashyap radiant systems ltd, presently available compression algorithms are written in c and pascal in Unix/PC environment. The existing system was developed using the LZ77 compression algorithm. The existing algorithms, which are written in C and pascal works only on the text files. There is no networking facility is available with the existing system.

Limitations of the Existing system:

- We can compress only the text files using the existing system. It is not possible to compress the image or another kind of files.
- We can not get the compression ratio more than 40% of input file.
- And the network facility is not available with the existing system, so we want to keep the package in each and every system. It makes a wastage of memory.
- It is not possible to compress the files with the memory size of less than 700 bytes.
- Here in the existing system, we can compress only the single text file, it is not possible to compress the directory or drive.

2.2 Proposed System

Keeping the above requirement, the proposed system is a complete package, which has the facility to compress any kind of file. Huffman algorithm has been implemented in the Proposed system. The proposed system has the facility of networking. TCP protocol has been used to transmit the message from one system to another system in this software. This TCP protocol is Connection oriented protocol. So

when we connect our system with others, this protocol creates one virtual circuit between these two systems. Through this circuit only all the data transmission takes place.

Visual Basic provides a powerful environment for developing graphical user interfaces than the c or pascal.so the new system has the excellent user-friendly ness.

The new system must have to satisfy all of the disadvantages of the existing system. And the new system must have friendly user interaction. The provision for the display of the user interactive messages, providing tools to study the file characteristics are the main aims of the proposed system.

2.3 Requirements on new system

Existing system has lot of disadvantages. In order to overcome these limitations of the existing system and to improve the performance of the existing system, this new system has been proposed and developed. Some important features are available with the new system. That are given below

- We can compress all type of files using the new system. It is possible to compress the image or another kind of files.
- We can get the compression ratio more than 40% of input file.
- And the network facility is available with the new system, so we do not want to keep the package in each and every system. It makes a savage of memory.
- It is possible to compress the files with the memory size of less than 700 bytes.
- It is possible to compress the directory or drive in the new system.
- The new system is an excellent user-friendly.

2.4 User Characteristics

User should be aware of something about Compress and decompress, Algorithm, and networking. User should know the visual basic.

3. PROGRAMMING ENVIRONMENT

3.1 Hardware Configuration

- **Operating System** : **Windows 98**
- **CPU** : **Pentium III 850 MHz**
- **RAM** : **64 MB RAM**
- **Hard Disk drive** : **10 GB**
- **Cache Memory** : **256 KB**
- **Floppy Disk Drive** : **1.44 MB**
- **Monitor** : **SVGA SAMTRON 45Bn Color**
- **Keyboard** : **104 Keys**
- **Mouse** : **Logitec serial mouse**

3.2 Description of Software & tools used

Front End : Visual Basic 6.0

Plat Form : Windows NT

About Microsoft Visual Basic 6.0

Visual Basic is an ideal programming language for developing sophisticated professional applications for Microsoft Windows. It makes use of Graphical User Interface for creating robust and powerful applications. Coding in GUI environment is quite a transition to traditional, linear programming methods where the user is guided through a linear path of execution and is limited to a small set of operations.The most significant features included are as follows.

Native Code Compilation

VB6.0 is the first version of Visual Basic to offerability to generate executables in native code format. Native code is code that can be executed directly by the operating system without requiring the assistance of another layer of software known as the interpreter to carry out their instructions. Visual Basic programs can now be compiled to native stand-alone code, if one requires, one can still compile to Pseudo-code(P-code).

Creating ActiveX Controls

ActiveX control creation is an exciting new VB capability that greatly expands the range of software a Visual Basic programmer can provide. ActiveX controls serve as the building blocks for other programs and Web pages. It is a set of component technologies that can be integrated by other applications.

Interface Enhancements

You can now configure almost all aspects of your working VB environment to suit the way you like to work.

Add-ins

The much more robust and fully Object-Oriented Visual Basic hierarchy makes it easy to create your own add-ins and wizards and seamlessly integrate them with the VB IDE.

Firing Events

In Visual Basic 6.0, custom events are a reality. Events are fired using the Raise Event statement and received with the help of the With Events keyword.

Multiple Projects

It is possible to work with multiple projects simultaneously.

Reason for choosing Visual Basic

Visual Basic is an integrated development environment in which one can develop, run, and debug event driven application. Visual Basic provides a powerful environment for developing graphical user interfaces. The available graphic algorithms are efficient enough and give a good performance in visual basic. The development of the application which requires both graphical user interface combined with graphic algorithms are met with Visual Basic and has therefore been selected to develop this project.

About Winsock ActiveX Control

The **Winsock** control is an ActiveX control that Provides a way for applications to communicate using the TCP/IP or UDP protocols. A **Winsock** control uses the underlying network connection or infrared port to transfer data. Because a **Winsock** control can act as a client that connects to a server application or as a server that provides connections to network clients, the first step in using a **Winsock** control is to determine whether the control will act as a client or as a server.

Winsock server applications using the TCP/IP protocol set the **Protocol** property to **sckTCPProtocol** and the **LocalPort** property to the port number the control will use to receive data. After the port number is set, the **Winsock** control can listen for data arriving at the port. To configure a **Winsock** control to listen on the port, call the **Listen** method.

When a client attempts to connect to the server application, the **Winsock** control generates the **ConnectionRequest** event. To accept the connection request, call the **Accept** method. If the **Winsock** control can successfully accept the connection, it starts the **Connect** event and sets the **RemoteHostIP** and **RemotePort** properties with the IP address and port number of the connecting client. After a connection is made, the control sets the **State** property to **sckConnected**. You then can begin to transfer data.
A connected Winsock control receives data through the connection. When data is sent to the server, the Winsock control generates the DataArrival event, indicating that the data

can be read from the control. To read the incoming data from the control, you can use the GetData method.

To send data through a connected Winsock control, you canuse the SendData method. As the data is sent through the control to the IP address and port number specified in the RemoteHostIP and RemotePort properties, the Winsock control generates the SendProgress event to notify you of the number of bytes sent and the number of bytes left to send. When the data transfer is complete, the control generates the SendComplete event.

After all data transfer has been completed, you can close a TCP/IP connection by calling the Close method. When a connection closes, the control generates the Close event. In addition to the Connect and Close events, the Winsock control can initiate the Error event to notify you of network errors. Unlike control errors or code errors, network errors do not cause a run-time error, but only raise the Error event.

To create a client application, set the RemoteHost and RemotePort properties, then call the Connect method. Once the connection to the server is established, you can transfer data as usual by using the SendData and GetData methods.

Windows NT

Windows NT has many features that place it in the upper ranks of Operating systems for micro-computers and workstations. Windows NT is a multitasking, multithreading, and scalable operating system with easy graphical user interface and compatibility with DOS and Windows 95. The major features of Windows NT are listed below.

- Portability of programs to other machines.
- Multitasking and Multithreading.
- Multiprocessor support.
- Scalability so as to increase performance.

- A new file system called NTFS to improve performance, reliability and security.
- Internet services.

Multi processor Support

Windows NT supports the use of more than one processor on a complete running NT. Windows NT provides support for computers with symmetric, multiprocessor setups.

Multi platform Support

Windows NT runs on a number of powerful desktop platforms. You can run Windows NT on RISC based platforms such as R3000, R4000, DEC Alpha AXP machines.

Multitasking and Multithreading

Windows NT truly lets you do more than one task at a time. Windows NT uses a pre-emptive multitasking scheme to manage multiple applications. Rather than applications co-operatively releasing control to the CPU to other applications. NT's pre-emptive multitasking system lets the CPU manage its own time.Windows NT is a secure system and it offers the flexibility of securing individual files instead of entire directories or drives.

4. SYSTEM DESIGN & DEVELOPMENT

This package is developed to compress any kind of file. The package is developed in visual basic 6.0 . The files are available in the binary format and any file can be loaded to compress on the algorithm, when the compression is completed the resulting image is save in the name given by the user. The various member functions and the major modules that are designed are described below in the various designs.

4.1. Input Design

Input design is the part of overall system design that requires careful attention and it is most important phase. Input to the system is very important and it should be validated. According to the input only we can get accurate output. So input data should be validated before processing starts. Objectives during input design are as follows,

a. Achieve high level accuracy
b. Ensure input is free of ambiguity

Input design involves converting the user originated inputs into a computer based format. The aim of input design is to make data entry easier and logical error free. It helps us to filter errors in the input data.

It involves procedures for capturing data, verifying and then passing them on to system. After choosing input medium, attention is focused on designing of error handling, control, and grouping and validation procedures.

During application development, care has been taken to make our system extremely user friendly and organize our screens such that the possibilities of making error are maintained.

In this package there are five menu options are available. If it is File option or the small_ data_ compress option , the user should give the following inputs.

When you compress the file the following details are the inputs to the system.

1. File name to be compressed
2. Output File name with path to save the compressed file
3. confirmation of the deletion of original file

When you decompress the file the following details are the inputs to the system.

1. File name to be decompressed
2. Output File name with path to save the decompressed file
3. confirmation of the deletion of original file

If it is Directory or Drive option user user should give the following inputs. When you compress the Directory or Drive the following details are the inputs to the system.

1. Directory or Drive name to be compressed
2. confirmation of the deletion of original Directory od Drive

When you decompress the Directory or Drive the following details are the inputs to the system.

1. Directory or Drive name to be decompressed
2. confirmation of the deletion of original Directory or Drive

If it is Networking option user user should give the following inputs in the client system form . When you compress or decompress file on the neworking option the following details are the inputs to the system.

1. The path of the File name to be sent to the server which is to be compressed or decompressed.
2. Remote host name of the server.
3. The path of the output file to be saved.

4.2 Output Design

An inevitable activity in the system is the proper design of input and output in a form acceptable to the user. Outputs from the system are required primarily to communicate the result of processing to user.

An output also provides a permanent copy of the results for later consultation. An intelligible output design will improve system relationships with the user and help in the decision making process.

The approach to output design is very dependent on the input and the nature of data. Special attention has to be made to data editing. The choice of appropriate output medium is also an important task. The output design must be specified and documented; data items have to be defined and arranged for clarity and easy comprehension.

If it is file option, the output will be a compressed file, when we compress the file and the output will be a original file, when we decompress the file.

If it is Directory or Drive option, the output will be a compressed Directory or Drive, when we compress the Directory or Drive and the output will be a original Directory or Drive, when we decompress the Directory or Drive.

4.3 Process Design

A computer procedure is a series of operations designed to manipulate data to produce output from a computer system. A procedure may be a single program or set of programs. Simple definition of process is program under execution.

The process design is the heart of the system design. The system specification is designed here. In this software, compressing the data and decompressing the data are main processing activities. Here in this software compression algorithms are written for

both maximum and minimum size data. In the algorithm for maximum size data compression ,there are two main modules are used in this software. That are

4.3.1 Phases of Compression

4.3.2 Phases of Expand

4.3.1. Phases of Compression

This module is having some important member functions. That member functions are described below. The member functions are

a. Count Bytes
b. Scale Counts
c. Build Tree
d. Convert Tree To Code
e. Output Counts
f. Compress File

a. Count Bytes

The function of this routine is to count the occurrences of each ASCII character in the input file and stored them in an array.

b. Scale Counts

The function of this routine is to scale the counts of each ASCII character so they will fit in an integer when they are added together to obtain the relative node weight in the tree. These weights become the saved weight after building the binary tree and are saved with the compressed file.

c. Build Tree

This one is used is to build the encoding tree for compression. Huffman trees are built from the bottom up using the relative weights (counts) of the nodes. As the minimum node pairs are found, the SavedWeight contains the relative Weight and the Weight is driven to 0 (zero) and will not be used again. The SavedWeight is used for debugging and OutputCounts routine. The Nodes array contains the ASCII characters decimal value in nodes 0 - 255. The EndOfStream node is 256 and is the last Symbol

output in the compressed file to tell the decoder where to stop decoding. The bottom of the tree will start with node 257 and progress upward until the Root node is stored. The function then returns the Root node.

d. Convert Tree To Code

Searching the tree each time to create the Symbol that represents an input character makes little sense. It makes more sense to convert the tree to the code symbols and look them up in an array. The purpose of this routine is to do just that, by recursively walking the tree in preorder and saving Leaf information in the Codes array.

e. Output Counts

This one is to save the weights in the compressed file so the decoder can build an identical tree. It would be inefficient to save all weights with a zero value. If there are more than 3 zero weights, a start, stop, and save weights takes place.

Data saved to file is in the form:

FirstNode, LastNode, Weights,..... &HFFF

The &HFFF tells the decoder it has reached the end of the weight stream.

f. Compress File

This member function of this routine is to re-read the input file and replace the ASCII character with its coded Symbol. It calls the another sub function BitHandler to save thesymbols. When the input reaches EOF, the BitHandler is called to output the EndOfStream symbol. The EndOfStream symbol tells the decoder to quit processing symbols.

4.3.2. Phases of Expand

This module is also having some important member functions. That member functions are described below. The member functions are

a. Input Counts
b. Build Tree
c. Expand Data

a. Input Counts

This one is used to read the NodeWeights from the compressed file. EndWeightStream indicates all Weights have been read in.

b. Build Tree

The function of this routine is to build the decoding tree based on the InputCounts of the compressed file.

c. Expand Data

The function of this routine is to walk the tree to a Leaf Node and then save the Leaf to the output File. It used the another sub function Input bit in it that is used to is to read the Huffman code bits and send them a bit at a time to the ExpandData routine.

In the algorithm for minimum size data compression , the following steps have to be followed to compress the files.

- The idea behind this encryption technique is to build a dictionary of 160 common letter pairs.

- As the program scans the message to compress, it tries to find pairs of letters in the dictionary. If it succeeds, the program stores the index of the pair in the compressed message.

- If the pair isn't in the dictionary, the program stores the first letter in the compressed message. The program continues processing the text until it has all been compressed.

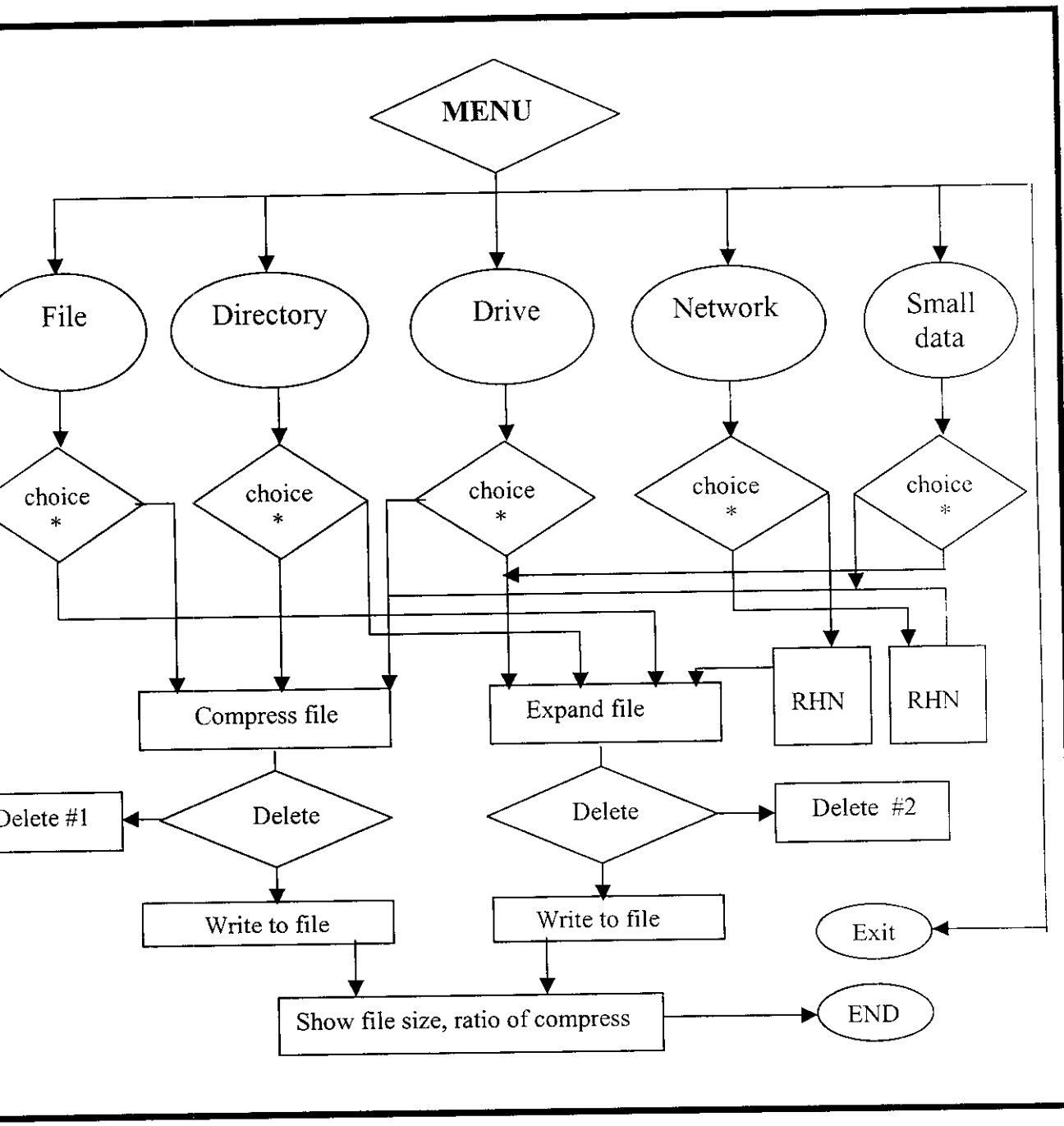
- If the program simply stored a character's ASCII code or a pairs index in the dictionary, there would be some overlap. To prevent that, the program maps these values into different values within the range 2 to 255. ASCII

codes are mapped to the beginning of the range, and pair codes are mapped to the end.

- To decompress the data, the program examines a byte value. If the value is 95 or smaller, it represents a character, so the program adds 32 to convert it back into an ASCII code. If the byte value is greater than 95, it's a pair code. The program uses it to look the pair up in the dictionary.

SYSTEM FLOWCHART

5. SYSTEM FLOWCHART



RHN - Remote Host Name

*** - compress or Expand #1 – Original file #2 – Compressed file**

SYSTEM TESTING AND IMPLEMENTATION

6. SYSTEM TESTING & IMPLEMENTATION

6.1 System Testing

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. Testing is a process of executing a program with the intent of finding errors. The user tests the developed system and changes are made according to their needs. The testing phase involves the testing of developed system using various kinds of data.

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. System testing is the state of implementation that is aimed at assuring that the system works accurately and efficiently before live operations commence.

Testing is vital to the success of the system. System testing makes the logical assumption that if all the parts of the system are correct, the goal will be successfully achieved. The candidate system is subject to variety of tests. A series of testing is performed for the proposed system before the system is ready for user acceptance test. The system is tested on all types of networks and the problem created by the system is identified and it is removed from the system after testing. The testing steps involved in system testing are,

6.1.1 Unit testing

6.1.2 Integration testing

6.1.3 Validation testing

6.1.4 Output testing

6.1.1. Unit testing

Unit testing focuses on the smallest unit of the software design module. This is known as module testing. The modules of the project are tested separately. The testing was carried out during programming stage itself. In this testing step each module was found to be working satisfactorily with regard to the expected output from the module.

In this software the modules file compression, Directory compression, Drive compression, Networking, Small data compress are tested separately. Each module was found to be working satisfactory with regard to the expected output from the module.

6.1.2. Integration testing

Strategies for integrating software components into a functioning product include the bottom-up strategy, the top-down strategy, and the sandwich strategy. Careful planning and scheduling are required to ensure that modules will be available for integration into the evolving software product when needed. The integration strategy dictates the order in which modules must be available, and thus exerts a strong influence on the order in which modules are written, debugged, and unit tested. All the modules are combined and tested as a whole. Thus in the integration testing step, all the errors uncovered are corrected for the next testing steps.

Here in this software there are five modules are available. All the modules are combined with the menu editor and tested as a whole.

6.1.3. Validation testing

Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in a manner that can be reasonably expected by the client. Here the uncompressed file has been given as an input . And the output file has been came with the expected compressed form by the user.

After validation test has been conducted, one of the two possible conditions exists. The function or performance characteristics conform to specification and are accepted. A deviation from specification is uncovered and a deficiency list is created.

Proposed system under consideration has been tested by using validation tests and was found to be working satisfactorily.

6.1.4. Output testing

After performing the validation tests, the next step is the output testing of the proposed system. No system is useful, if it does not produce the required output in a specified format. Considering the format required by the users tests the output generated or displayed by the system under consideration. Here, the output format is considered on the monitor only.

The output format on the screen is found to be correct as the format was designed in the system design phase according to the user needs. The hardcopy output also comes out as specified requirements by the user. Hence, output testing does not result in any correction in the system. The system is tested in many networks.

Here the outputs are the compressed file, compressed directory and compressed drive. All of them are correct as the format designed in the system design phase.

6.2 Implementation

Implementation is the process that includes all those activities that take place to convert from the existing system to new system. The new system should be totally user-friendly, replacing an existing manual and automated system. Proper implementation is essential to provide a reliable system to meet the organization requirements.

The system is at present implemented and checked by the authority person on a parallel basis and is found to be working more satisfactory.

CONCLUSION

7. CONCLUSION

The system has been tested using test data cases and it has been found to work successfully under test conditions.

After successful user testing, it has been found that the new system overcomes most of the limitations of the existing system and works accordingly to the design specifications.

All the procedures and assumptions made in designing this project were strictly followed.

Reviews have been collected from the users about this system and are found to be satisfactory Maintenance of the system have to be undertaken to the new requirements that may pop-up during passage of time.

SCOPE FOR FUTURE DEVELOPMENT

8. SCOPE FOR FURTHER DEVELOPMENT

The seach for even more efficient compression algorithm will continue in to the next century. Martyn believes that this search will move toward mathematical methods.

It is expected that newer mathematical transforms that compress data with comparatively less degration in quality, will be developed in the near future.

The statistical analysis can be elaborated by adding some more routines to have a complete data compression package.

When ftp protocol were used for the file transfer, tranfering speed can be still improved and the execution time can be reduced.

BIBLIOGRAPHY

8. BIBLIOGRAPHY

REFERENCED BOOKS

- Binstock, Andrew and Rex, John, '***Practical Algorithms for Programmers***', Addison-Wesley, ISBN 0-201-63208-X, 1998 .
- Nelson, Mark and Gailly, Jean-Loup, '***The Data Compression Book***', M&T, Second edition, ISBN 1-55851-434-1, 1996 .
- Stephens, Rod, '***Visual Basic Algorithms***', John Wiley & Sons, Inc., ISBN 0-471-13418-X, 1996 .
- E.Vanglos Perroutsos '**Matering VB 6.0**' BPB Publications, 1997.
- Ellis.M.Awad '**System Analysis and Design**' Galogotia Publications Pvt Ltd., 1993.
- Roger.S.Pressman. '**Software Engineering**' McGrawHill Internation Edition, 1997

WEBSITES

- www.cis.ohio-state.edu/hypertext/faq/usenet/compression-faq/top.html
- www.rasip.fer.hr/research/compress/index.html
- http://www.ics.uci.edu/~dan/topic.html#dcl

APPENDIX

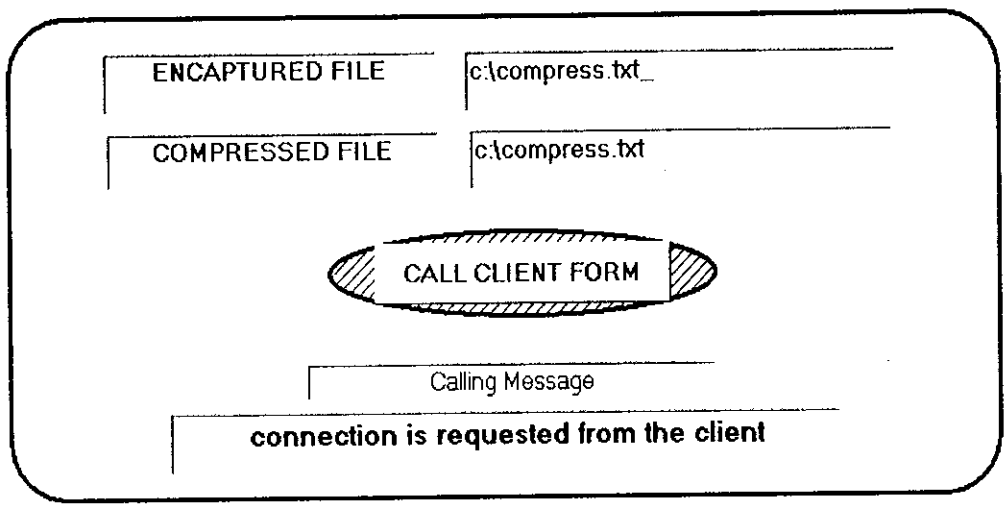
A PROJECT ON COMPRESSION OF DATA

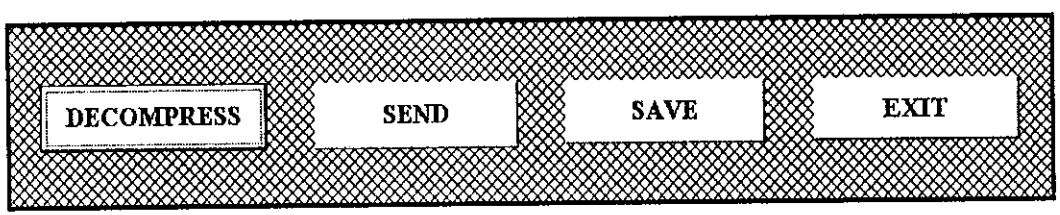
Developed

at

SAKTHISRI INFOTECH (P) Ltd.

Encapturing the file from the client for decompression

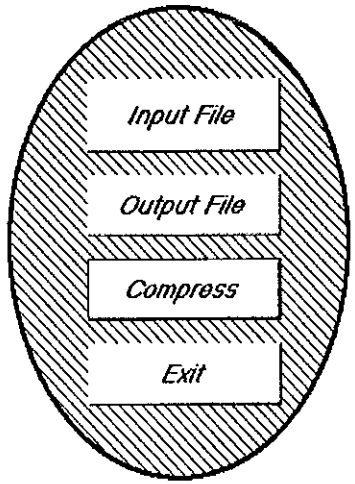




Small data compression

compression of the small size of datas

Flie to compress	C:\Program Files\Huffman.txt
path of the compressed file	C:\Program Files\Huffman.txt_
Original Length	650 Bytes
Compressed Length	374 Bytes
% Compression	42.46%



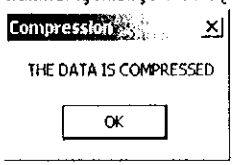
☑ Confirmation of deletion

ORIGINAL DATA

Huffman algorithm uses a frequency table enddata structure called binary tree for constructing avariable length code. Here , the binary tree is referred to as code tree. In a code tree each node consists of two child nodes or none. The nodes with no child nodes are called leaves of the tree and they represent the symbols. These nodes are linked to one another based on the probabilities. i.e., frequencies of the symbols .The Huffman tree is so formed that the bits allocated for the most probable symbol is the least and is maximum for the least probable symbol. This results in shorter code for the message or the image to be coded .

COMPRESSED DATA

ÿ`````(UFFÎqIGr'HÛImbIIQUICIIBIbtDIIIRøTIa½l€Ie~ITIaFrfpI RøIIbV~'tABI¹IGIIO*{{haIfaBe~ITIa}²EFhloxbcI*dlè§qII*dIaÃÀ· O*Ip IcOITiI©Lob*cr pë`i·O*cÆf·wA»ob*c~a½l€¹Ãlc0IfaTIakof EII¼mIjfaSYMBOLâ´il·O*c~a²NK€dwpakÐinB~€sqfa¼OBAB»Ý âtlèIFIQUICÝcOIfaSYMBOLctÀa(UFFÎqTIa}IwFr*oIIdil´clÔ'½lo FrdilñjÝ¼OBABIIYMBOø}di¹Ãtbltc×IMUÛFrdi¹ÃII¹BABIIYMBOL {À}²mÎÔtqS'ÜhIO*IrdiImSAÇsnfaÎÇdw¯IO*o{



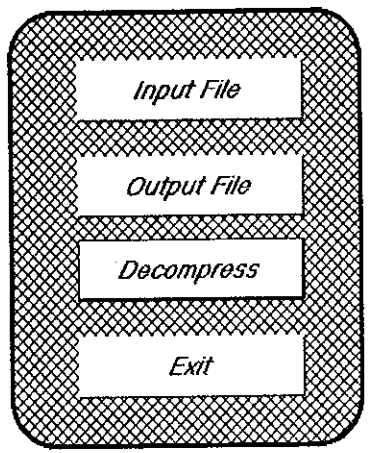
Decompression of small data

Decompression of the small size of datas

Flie to decompress — C:\Program Files\Huffman.txt_

path of the decompressed file — C:\Program Files\Huffman.txt

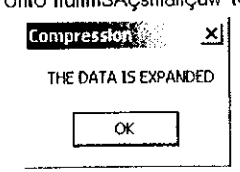
Original Length	374 Bytes
Decompressed Length	650 Bytes
% of Decompression	42.46%



☑ Confirmation of deletion

COMPRESSED DATA

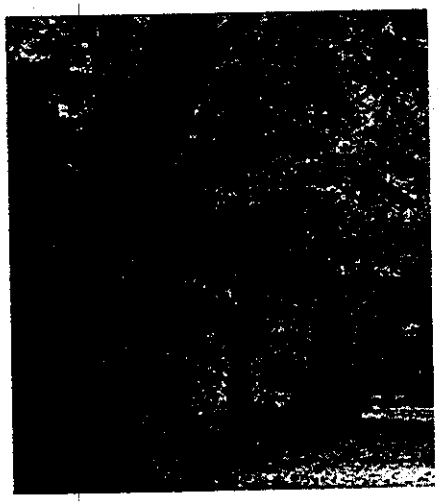
ÿ`````(UFFÍqIGrˆHÛImbIIQUICIIBIbIDIIIRøTIa½I€Ie˜ITIaFrIpI
RøIIbV˜IABI²IGfIO'{(halfaBe˜ITIa}²EFhloxbcI'dIè§qII'dIaÂA·
O'Ip IcOITiI©LoÞ'cr·pè'i·O'cÆf·wÁ›oÞ'c˜a'½I€²ÃIcOIfaTIakof
EII¼mIjfaSYMBOLâ´iI·O'c˜a²NK€dwpakÐinB~€sqIa¼OBAB»´
Ýâ‡èIFIQUICÝcOIfaSYMBOLcIÀa(UFFÍqTIa}IwFr×ofIdiI´cIÔ½
IoFrdiIñj¼OBABIIYMBO¢}di²ÃfbItcÌXIMUÛFrdi²ÃII²BABIIYMB
OL{À}²mÍÒtqS'ÙhIO'IrdiImSAÇsnfaÍÇdw¯tO'o{



DECOMPRESSED DATA

Huffman algorithm uses a frequency table anddata structure called binary tree for constructing avariable length code. Here , the binary tree is referred to as code tree. In a code tree each node consists of two child nodes or none. The nodes with no child nodes are called leaves of the tree and they represent the symbols. These nodes are linked to one another based on the probabilities. i.e., frequencies of the symbols .The Huffman tree is so formed that the bits allocated for the most probable symbol is the least and is maximum for the least probable symbol. This results in shorter code for the message or the image to be coded .

About software



Developed by

P.Rajendhiran M.C.A
Kumaraguru college of technology.
Coimbatore.

Developed for

Sakthsri infotech Pvt Limited.
Coimbatore.

File decompression

Decompression of the single file

FILE INFORMATION

Compressed File

Path of input file: C:\Games\B2Demo\Screenshot1.bmp_

Length of input file: 115505 Bytes

Expanded File

Path of out file: C:\Games\B2Demo\Screenshot1.bmp

Length of output file: 921655 Bytes

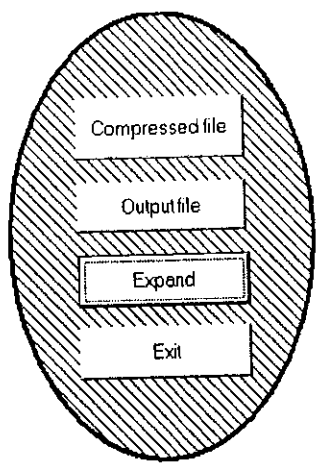
PROGRESS INFORMATION

Progress Messages

Expansion Complete...

Expanding file 497% Complete

MENU



☑ **Confirmation of deletion**

P-748