Using Video in a Graphical User Interface for Remote Control of a Mobile Robot

Title: Using Video in a Graphical User Interface for Remote Control of a Mobile Robot
Video i ett grafiskt användargränssnitt för fjärrkontroll av en mobil robot

Publication type: Master’s Thesis in Computer Science (20 credits)
Department: Department of Numerical Analysis and Computer Science
Author: Julia Narmack
Supervisor: Henrik I. Christensen, CAS, NADA
Examiner: Henrik I. Christensen, CAS, NADA
Using Video in a Graphical User Interface for Remote Control of a Mobile Robot

Abstract

Rapidly developing Internet communication technologies provide today wide opportunities to create different applications for interacting with various robotic systems. This Master’s thesis presents a study of how to supply a graphic user interface for remote control of an autonomous mobile robot through the Internet with video information.

The goal of this project is to analyze different techniques for video compression and the transmission of video information through the Internet, and to find which one gives the best video quality and the best performance. The study also analyzes how to create a robust and secure video transmission system. Another aim of this project is to implement all conclusions achieved in the analysis, to choose the best solution and build a simple prototype of such a video transmitting system. The construction of this prototype is based on an existing mobile robot – the domestic cleaner Trilobite.

Video i ett grafiskt användargränssnitt för fjärrkontroll av en mobil robot

Referat

Den snabba utvecklingen av Internets kommunikationsteknologi ger idag stora möjligheter för att skapa olika applikationer för interaktion med robotiksystem. Denna examensarbetsrapport presenterar en studie av att använda videoinformation i ett grafiskt användargränssnitt för fjärrkontroll av en självgående mobil robot genom Internet.

Contents

1 Background, problems and aims
   1.1 The importance of providing a remote user with video information 1
   1.2 The existing system 1
   1.3 The system available for prototype building 2
   1.4 The goals of this project 2

2 Related works – Control of autonomous robots through the Internet
   using video feedback
   2.1 The development of the robots control through the Internet 4
   2.2 MAX-dog robot 4
   2.3 Mobile Robots through on the Web 5
   2.4 Khep On The Web 5
   2.5 Xavier: an autonomous mobile robot on the Web 6
   2.6 Lessons learned 6

3 Possible problems and solutions
   3.1 Design of the system architecture 8
   3.2 Software structure 10
   3.3 Choosing the appropriate hardware 11
   3.4 Video streaming technologies
      3.4.1 Video compression technologies 11
      3.4.2 MPEG 15
      3.4.3 JPEG 24
      3.4.4 H.261/H.263 Video Coding 26
      3.4.5 QuickTime 27
      3.4.6 Wavelet based image compression 28
      3.4.7 AVI 29
      3.4.8 Summary 29
      3.4.9 Conclusions 32
   3.5 Video Server technology is the answer
      3.5.1 Video streaming server 33
      3.5.2 Different streaming techniques 33
      3.5.3 Hardware based video servers 33
      3.5.4 Conclusion 34
   3.6 Providing security of video information
      3.6.1 MySQL 35
      3.6.2 PHP 36
      3.6.3 Comparing ASP with PHP 37
3.7 Creating a reliable robust system ..................................................38
  3.7.1 Compiled-time program errors ..................................................38
  3.7.2 Run-time program errors .......................................................38
  3.7.3 Run-time exceptions .............................................................39
  3.7.4 Run-time user errors .............................................................40
4 Implementation ..................................................................................42
  4.1 Chosen video server ....................................................................42
  4.2 Database ......................................................................................42
  4.3 Page configuration ......................................................................44
  4.4 Implementation of video security and robustness illustrated
    by means of use cases and MSCs .....................................................45
  Exceptional cases .............................................................................62
4.5 Conclusions ...................................................................................65
5 Result, conclusions and future work ..................................................66
  5.1 Conclusions – project achievements ............................................66
  5.2 Future work ................................................................................66
    5.2.1 Providing an upper view over the robot ....................................67
    5.2.2 Increasing robustness of the system ........................................67
    5.2.3 Remote portable controller ....................................................67
References .............................................................................................69
Appendix 1: Profiles and levels of MPEG-2 ........................................73
Appendix 2: Acronyms .........................................................................74
1 Background, problems and aims

Different kinds of robots are coming more and more in human everyday life. Hence nowadays technology is developing fast, the problem of how to let a user without any technical experience to easily use advanced technique has become more obvious.

1.1 The importance of providing a remote user with video information

There are many cases when there is a need to control a mobile robot from a remote computer. In these cases interaction between the user and the robot happens through a graphical user interface. So, an appropriate built user interface would help to shorten the distance between an inexperienced user and the advanced technology. A feedback to user commands in such an interface can be given in a number of different ways, e.g. some text about what the robot has performed last. The most effective feedback appears to be a video stream received as an output from a video device installed on the robot. The video information provides the user with a possibility to be present on the other side of the Internet connection and to see what the robot is doing and in which way it responds to the user’s commands. If a robot is static there is only the robot’s state that can be interesting to the user. When it is mobile the user also wants to know where the robot is located at the moment.

The possibility to see the robot responding to user commands has some benefits:
- It increases the user’s feeling of being in charge of the whole mobile robot control system.
- It improves interaction between the user and the robot.
- It provides a clear feedback to the user’s commands.
- The system’s feedback is easier to interpret.
- It allows inexperienced people, without any sense for technology, to handle the system of remote robot control.

All these considerations become more important when the remote robot is a domestic cleaner. This means that the whole system of remote control has to be suited for a large group of users that often do not have any technical experience. That is why the provision of video information about the robot is the logical solution to enhance the system of remote robot control.

1.2 The existing system

Electrolux has developed an autonomous mobile robot “Trilobite”. This is a fully automatic vacuum cleaner. It scans the room and starts operating autonomously after the user has chosen a cleaning program. The scanning of the room allows the
robot to compute the time needed for the cleaning of the whole room. The built-in sensors allow the robot to avoid obstacles such as furniture or other objects. The Trilobite is also equipped with a system for automatic battery charging. When its batteries need charging, the Trilobite automatically finds the charger and then continues working until the job is done. There are four buttons to control the Trilobite. By using them, the user can choose one of the special programmes for cleaning. A display beside the buttons informs the user about the robot’s state, such as the battery capacity, the status of the dust container, the time left for cleaning or any other problems.

1.3 The system available for prototype building

The following units were available for current prototype building:
- A Trilobite.
- A mobile scout with Apache server on Linux OS.
- A USB Phillips video camera mounted on the scout.

There is no possibility to place a video camera on the Trilobite, so instead there is a temporary camera on the mobile unit huey. This allows the demonstration of the connection between the camera and a remote user.

1.4 The goals of this project

The aim of this project is to build a video transmission system as part of a remote control system for the autonomous domestic cleaner Trilobite. The Trilobite is supposed to travel around in the environment and transmit video pictures or a video-stream to the remote user.

Giving the user the possibility to control the robot from a remote computer through the Internet would considerably enhance the Trilobite’s autonomous behaviour. Because the user is allowed to control a Trilobite when actually being far from the robot he has no possibility to see where the robot is at the moment and whether everything goes well. A remote control system also includes the possibility for the user to guide the robot through the Internet manually to a special place where he wishes the robot to begin cleaning. This feature makes the video feedback a factor of great importance.

So, the goal of the Master’s project is to build a prototype of the system that enables the user to receive a video-stream from the Trilobite through an interactive user interface.

Building such a video control system involves solving the following problems:
- The design of the system architecture.
  - Both hardware design and software configuration has to be analysed and the best solution implemented.
• The choice of appropriate hardware.
  o It is relatively easy to get small cameras (size 1 cm in diameter, 10mm high). Thus a camera could easily be integrated into a vacuum cleaner. The camera can either be digital or analog. The tradeoffs are changing very fast, but a digital option is likely to be a preferable solution.

• The analysis and choice of video coding technique.
  o Modern technology provides a number of different techniques for video coding and for transmitting compressed video information. To analyze them and choose an appropriate method is one of the challenges of this work.

• To ensure the security of video information.
  o The system should support multiple accesses to the robot. Security of the image transmitting demands a login system with user authentication. Only one person at any time can control the robot, the other users can just watch the robot performing its tasks. A database is required to store the information about the users in order to authenticate a user.

• The creation of a robust system.
  o Any system has to be robust to be usable. Robustness enhances the reliability of the system. Here all robustness principles have to be analyzed and implemented in the prototype.

All the above means that the work includes the analysis of the system needs and different solutions. A prototype of the system has to be based on this analysis and on the chosen solution.
2 Related works – control of autonomous robots through the Internet using video feedback

The analysis of some projects done about the same subject can help to get the best answers to the questions mentioned above.

2.1 The development of the robots control through the Internet

Nowadays there are a number of mobile robot systems available on the Internet. The history of autonomous robots controlled through the Internet takes its beginning long before 1992 (about 10 years earlier), the year when WWW was introduced.

One of the first such devices were the “coke” machines. A remote user was able to interact with a remote coke or coffee machine, checking its status or reserving a can [1].

One of the first truly teleoperated devices accessible through the Internet was the Australian Telerobot – a remotely operated 6 degrees of freedom manipulator developed by Ken Taylor at the University of Western Australia [2]. This six-axis robot has a fixed observing camera, continuously sending a video stream to the Web page.

There are some more recent interesting scientific projects about the control of mobile robots through the Internet. They are done from different aspects and are helpful in different ways. They give an overview of the different techniques for video service.

2.2 MAX-dog robot

One project is a MAX robot dog project developed by the N-CART team at the Ryerson Polytechnic University, Canada [3]. The MAX dog is a wireless microcontroller-based mobile robot prototype. It allows teleoperated control via a web browser communicating over an IP network.

The robot streams video images continuously via an analog radio link from an on-board camera to a web server. Anyone having a connection via a Java enabled Netscape web browser can control MAX through an intuitive interface. Two Java applet windows present the system’s interface. One of them provides a configurable “dog’s eye view” of the environment. This is very similar to the
current Trilobite project. There, the user too gets a view of the environment through the camera mounted on the robot.

### 2.3 Mobile Robots through on the Web

“Mobile Robots through on the Web” [4] is a project that intends to establish a modular framework for mobile robots on the Internet. It should allow guiding mobile robots through the Internet by using an appropriate type of interface. There are three mobile robots presented. They are supposed to be easily connected for operation in various environments. We are going to look at the system architecture and design of the web interface. The web interface in this project consists of five independent modules for custom service. Each of them includes a server-side program and a client-side applet. These five modules are: a login service, a video feedback service, a robot guidance service, a virtual representation service and a chat service. It is the video feedback service and the login service which are of interest considering the Master’s project. The module called Image Grabber continuously grabs images from the camera and transmits them in JPEG format on request to the Java server. Further, the image is sent by a Java (server) program to a Java applet, via a socket, and is interpreted by the applet (in either GIF or JPEG format). This system allows feedback from any number of sources. A local network handles images of medium quality, from 10 to 15 200x150 frames per second.

### 2.4 Khep On The Web

Another interesting example of a mobile robot on the Internet is the project “Khep On The Web”[5]. The goal of the project is to provide the access through network communication facilities to a complex mobile robotic setup. In this experiment the mobile robot Khepera is used. It is equipped with an on-board camera, which is similar to the one used in the Master’s project. This CCD-camera is connected to a PC via a wired link. The camera sends the video signal to a frame grabber placed on the same PC. The robot is placed in a special room and can move within it. There is an external camera too, mounted on the ceiling above the robot in order to give the user a global view. This is an interesting helping device for planning the robot movements. Considering the current project, there is no such possibility to place an external camera because the Trilobite is going to travel without hinder from one room to another. In this case there is the need to put a camera in every possible room. This problem is beyond the current project goal, but can be seen as a possible extension.

As a solution to video client-server communication, the following method is used: One of the program processes grabs the images and puts them in shared memory in JPEG format (160x120). When a client performs an access to the control page, the server starts a program that continuously sends the images stored in the memory. On the client side transmitted images are placed into a Java applet. The user has access to a control page made in plain HTML and using clickable images.
A major disadvantage of the technique described above is that it is supported by Netscape but not by Internet Explorer. There are only completed images that are sent to the client. Other techniques are discussed in the article, such as to use animated GIF and to send only the part that changes in the images. This can result in “the caterpillar effect” [5].

2.5 Xavier: an autonomous mobile robot on the Web

One of the first web-based robots was Xavier [6]. Its difference from other web-based robots is that it was mobile and autonomous. Xavier was created for operating in a peopled environment, such as an office. The similarity with the Trilobite is that Xavier had to be also controlled by a remote user (office worker) when performing some tasks. As a way for interacting with the Xavier, a web-based user interface was constructed. This interface is a kind of a command user interface presented as a web page. It provides a list of possible destinations to send the robot to, a list of simple tasks to perform at that location and it shows Xavier’s current status, updated every 5 to 10 seconds. There is an additional monitoring web page that includes the robot’s current status, a zoomable map of the floor Xavier is currently on and a color picture of the robot’s view. This picture, taken by a camera mounted on the Xavier’s top, is sent to the web page in gif format and is updated also only each 5 to 10 seconds. The same updating rate is used for a zoomable map that shows the area around the robot and its most likely location based on predictions that the robot maintains. All operating tasks on Xavier are carried out by two 200 Mhz Pentium computers, running Linux. Communication to the outside world is provided via a WaveLAN wireless Ethernet system.

Xavier on the Web was a successful project that achieved the main goals the science group had pursued. It means that their experience about hardware configuration can be useful for the current Trilobite project, as it has likely similar purposes. The only weakness of Xavier is the lack of efficiency in the video feedback that cannot be considered as video at all. Single images are sent at such a low rate that a user is not able to associate them with the video. The low image rate also contributes to a time delay in the robot’s movements on the Web page. This can be confusing for the user because the robot would do a command faster than the user could get the information about it. So, the user can ask the robot to perform a task that is already done.

2.6 Lessons learned

All considered projects are similar to the current project in some way. They show a number of possible solutions for problems emerging in connection with the current project. The solutions presented in these projects have different efficiency. Most of them, such as system architecture, video compression and video streaming techniques, should be analyzed and evaluated. The others, such as gif image format in Xavier project, teach us never to use them.
This brief analysis of similar projects gives overview of existing technologies in the current project area and shows directions for further researches. In the next sections possible problems solutions will be analysed and there will be discussed why a particular solution was chosen.
3 Possible problems and solutions

All problems and questions defined in section 1.4 will be considered and analyzed in this chapter. Different possible solutions are going to be discussed and the most suitable solution for this project will be chosen.

3.1 Design of the system architecture

To provide a user with the ability to interact with a remote robot, the whole system should combine the network technology with a mobile robot (Trilobite).

We have the Trilobite with its on-board camera on the one side of the Internet and the remote user on the other – see figure 1. An on-board camera will be mounted on the front of the Trilobite in order to give the user a view of the environment in front of the robot. It is connected to the video server through a video transceiver.

![Figure 1. System configuration.](image)

On the robot’s side there has to be a video server and a command server. There has to be a HTTP server in the system also to execute requests from clients and deliver them to the robot side. By means of this Web server the central server architecture can be implemented. All clients and servers can be connected to this central Web server and communicate with each other through it. There are two possibilities to manage all system servers with such a central architecture:

The first solution is to place the Web server, the video server and the command server on the same computer (PC). This host computer will communicate with the Trilobite by a radio modem, i.e. Wireless LAN (see figure 2) connected to a serial port. The video signal will be captured by the frame grabber, which is placed on
the host computer. This computer will be connected to the Internet by a standard Ethernet card.

Another solution is to have both video and command servers inside the Trilobite. In this case an additional workstation is needed anyway to place
- the HTTP server,
- the dynamic script,
- the database,
- other applications.

All those applications cannot be located inside the Trilobite because of the space needed for a more powerful computer that has to be on the Trilobite.

Even if there is an additional base station there remain some difficulties combined with this solution too. Both the video server and the commando server mounted on the Trilobite can take considerable space influencing the form of the robot. The Trilobite is a portable and mobile robot, so its weight is of great importance.
However, this version does not suit the current project state, which is focused on the construction of a simple prototype of the system and does not have possibilities to implement such a hardware architecture. This last version can anyway be considered as a direction for future work.

So, the current project follows the hardware configuration presented in figure 2.

3.2 Software structure

Figure 3. Software configuration.

Figure 3 illustrates the software structure of the video transmission system. An onboard video camera placed on the Trilobite is connected to a video server through a wireless video link. The Web server program is an Apache HTTP Web server running on the Linux platform. The whole system consists of several independent modules, which are located on the server side and are interacting with each other. One module is the dynamic script in the HTTP server. It handles the communication between the client and the server and interacts with two other modules: the database and the video server. The video server is a visual feedback module. It sends the continuous image stream to the client through the virtual representation module-dynamic script. This dynamic script generates a Web page where the video information is presented to the client.
3.3 Choosing the appropriate hardware

Capturing device
A camera can capture two types of video:
- analog or
- digital

If the captured video is analog it has to be represented in digital form.
The choosing of a video camera relies on several demands. It should be
- compatible with the robots OS
- very little
- cheap
- preferably digital.

Compared to traditional analogue video, digital video does not have to be
digitised, which means that it provides the following advantages:
- There is no copy.
- The picture does not get fuzzy
- The signal-to-Noise ratio goes down slowly.
- Editing, storage and retrieval is simpler, quicker and cheaper.

3.4 Video streaming technologies

Successful delivery of video over the network has to involve processing the video
using the following methods:
- Compressing the video using some compression technique. This is
generally needed for almost all networks because of the high bandwidth
requirements of the video.
- Streaming the video using data packets over the network. [7]

3.4.1 Video compression technologies

Digital representation of video
A video stream can be represented in the following ways:
- Moving images
- By a sequence of frames (pictures)
  - each frame being a digital image
  - delay constant between appearance of two successive frames
  - frame rate: number of frames displayed per second
  - an impression of smooth motion starts at about 16 frames per
second.e.g.
  - movies: fps = 24
  - American TV: fps = 30 (NTSC)
  - European TV: fps = 25 (PAL) (appendix 2).
Digital representation of images

Bitmaps:
- two dimensional matrix made up of picture elements called pixels
- each pixel has a value called pixel amplitude
- the number of bits available to code a pixel is called pixel depth e.g.
  - pixel depth = 1 black and white images pixel depth > 1
docoloured or grey-scaled images

The need for video compression

Video information represented in digital form enables us to process and transmit it using digital computers, processors and networks. However, the very high bit-rates of digital video signals means that the information must be compressed before it can be processed and transmitted:

- Bit rates required for uncompressed NTSC TV (appendix 2)
  - 858 x 525 = 45450 pixels in each frame
  - frame rate = 30 fps
  - bit rate > 160 Mbps
- Bit rates required for uncompressed high definition TV (HDTV)
  - 1920 x 1080 = 2073600 pixels in each frame
  - frame rate = 60
  - bit rate > 2 Gbps
- Bit rates required for uncompressed audio data streams
  - telephone quality: 64 Kbps
  - CD quality: 1.4 Mbps

Frame Rate

How many frames are displayed per second, also the method of frame display: progressive – each line of video is shown one after the other; interlaced – odd lines (fields) are shown first, then – the even lines.

Modes of compression

The digital video data contains a considerable amount of redundancy. [8] Compression based on statistical redundancy is a reversible process (lossless compression). Higher levels of compression are achieved by removing subjectively redundant information but this destroys some of the original image information (lossy compression).

- Lossless compression
  - Redundancies removed only.
  - Data not altered or lost in the process.
  - E.g. computer file compression.
- Lossy compression
  - Redundancies as well as some unessential information in the data removed.
E.g. image and video compression.

Many people are scared off by the term “lossy compression”, but when it comes to representing real-word scenes there is no digital image format that can retain the information that impinges on the human eyeball. The technical meaning of “lossy” refers to loss of information over repeated compression cycles. [8]

**Compression efficiency**
Different picture and video compression techniques have been derived to reduce high bit rates. Their ability to perform this task is quantified by the compression ratio:
- Compression ratio = (size of original data) / (size of compressed data)

The higher the compression ratio is, the smaller the bandwidth consumption. On the other side, increasing compression causes an increasing degradation of the image. These are called artifacts. [9]

The dilemma is however that: the more sophisticated the employed compression technique is, the more complex and expensive the system. This generally makes sophisticated compression restrictive in terms of maintaining low system costs.
- Figure of merit = (size of compressed data) / (size of original data)
  - reciprocal of compression ratio

**Two types of compression**
There are two types of compression: entropy encoding and source encoding.
- Entropy encoding
  - does not take into account the nature of information to be compressed
  - treats all data as sequences of bits
  - is lossless
- Source encoding
  - makes use of transformations dependent on the type of original data
    - e.g. suppression of silence for audio, exploring motion redundancies for video
  - is either lossless or lossy

**Real-Time / Non-Real-Time**
This refers to capturing, compressing, decompressing and playing back the video in real time with no delays. The requirement is to have sufficient frame rate (frames per second) to make sure that there is no jerky motion.

**Color Resolution**
This refers to the number of colours displayed at any one time. There are also various color formats: RGB and YUV (Appendix 2) are two common formats. Color depth is the maximum number of colours displayed.

**Coding of still images**
The coding of still images can be performed according to the scheme in figure 4.
Entropy model: model image in some way to exploit properties/redundancy.

Entropy encoder: code, based on statistics of model output, e.g. Huffman encoding.

**Standards for video encoding**
Many different techniques for compressing digital video have been developed in recent years. All of these exploit the spatial and temporal redundancy of a video sequence in order to achieve compression. Different applications handle compression with different success for each compression type. Some preserve better the static details (but there are unaccepted jumps between frames), while others maintain continual motion (but only the rude details could be recognized). Several International Standards have been issued, which provide a means of encoding still and moving video information. Here are some of them:

- MPEG standards
  - have been developed to support a wide range of digital video applications
- JPEG – the standard designed for still image coding
- H.261 – the standard is part of the H.320 videoconferencing standards and is optimised for two-way videoconferencing
  - targeted at low resolution videoconferencing
- H.263
  - improved H.261 standard
- Quicktime
  - a format for video streaming developed especially for Macintosh computers.
- Wavelet Transform
  - the standard based on still image compression.
- AVI
  - a format for streaming of video.

Video coding is still an important area for research and development. Low bit rate applications are not supported by the existing standards. The H.263 standard [10], currently at draft stage, should provide higher quality video communications at bit
rates below 64 Kbps. The MPEG4 standard, which is currently under development is likely to support more advanced video communication applications.

### 3.4.2 MPEG

MPEG ([Appendix 2](#)) is an Audio/Video compression, storage, and transmission standard. It is commonly used for audio and video on web pages, embedded videos in software, video CD, DVD, and digital television. The major advantage of MPEG compared to other video and audio coding formats is that MPEG files are much smaller for the same quality. This is because MPEG uses very sophisticated compression techniques. While MPEG players are commonly available and bundled with many operating systems and off the shelf software, there is much less choice for encoders to produce these MPEGs. There are several commercial encoders available, and few freeware versions. [11]

MPEG compression works on the assumption that much of the image from one frame to the next in film or video remains the same. Except for key frames, no single MPEG frame is complete and each has meaning only with reference to the frame that preceded it. That is why if a single frame is lost while transmitting video sequence, all other frames that have reference to this lost frame will be lost too. [12]

Besides the fact that MPEG works well in a wide variety of applications, a large part of its popularity is that it is defined in following finalized international standards:

- **MPEG-1**
  - the standard, on which such products as Video CD and MP3 are based
  - designed for low bitrate audio and video playback applications
- **MPEG-2**
  - the standard, on which such products as Digital Television set top boxes and DVD are based
- **MPEG-4**
  - the standard for multimedia for the fixed and mobile Web.
- **MPEG-7**
  - the standard for description and search of audio and visual content.
- **MPEG-21**
  - aims to provide a truly interoperable multimedia framework.
  - the standard is in development, that is why it is not discussed here.

All MPEG standards are back compatible. This means that an MPEG-1 video sequence also can be packetized as MPEG-2 or MPEG-4 video. Similarly, MPEG-2 can be packetized as an MPEG-4 video sequence. The difference between a true MPEG-4 video and an MPEG-4-packetized MPEG-1 video sequence is that the lower standard does not make use of the enhanced or new features of the higher standard.
The essence of all MPEG efforts is interoperability – interoperability for the consumer. Interoperability means that consumers can be sure to be able to use the content and not be bugged by incompatible formats, codecs, metadata, and so forth. [13]

Software-based and hardware-based are the two types of decoding methods for MPEG files. The disadvantage of software-based MPEG video decoding is that it requires a lot of computing power. In order to get a real-time performance, 30 frames per second are required. Because of the extremely CPU-intensive requirement of software-based decoding, the quality of MPEG files viewed on software-based decoders is usually poorer than that on hardware based decoders.

Even though hardware based decoders offer better performance over software based decoders, they are expensive. Because MPEG boards can cost up to thousands of dollars, average users usually stay away from these kinds of products. MPEG file offers the same features as the Quicktime movie files. These kinds of files offer sequential image and audio that may greatly help people to understand and visualize a process. [14]

**MPEG video encoding**

- Reducing spatial redundancies
  - for individual images (cf. JPEG)
  - intracoded (I – Intra frame) pictures
- Reducing temporal redundancies
  - with motion compensation
  - predicted (P – Predicted frame) pictures and bidirectional (B) pictures
- Generating a sequence of I, B, and B pictures
  - e.g. IBBPBBPBBI...
  - reference pictures I for P; I and P for B
  - (M, N) notation for MPEG encoding pattern

**Motion compensation: Coding of P and B pictures**

Motion compensation means that only changes in the image in a sequence have to be compressed and transmitted. This problem is usually addressed by dividing the image into blocks (makroblocks). Each block is examined for motion. If a block is found to contain no motion, a code is sent to the decompressor to leave the block the same as the previous one.

If enough processing power is available, still more powerful techniques may be applied. For example, blocks may be compared to previous blocks to see if there is a difference between the two. Only this difference (motion vector) is sent. [15]

- Macroblock
  - 16 x 16 pixel area (YUV color model used), or 8 x 8
  - finding best matching makroblocks in reference pictures
  - computing difference between actual and best matching makroblocks
- using DCT (discrete cosine transform) to code the difference
- Motion vector
  - indicating spatial translation of a macroblock between two pictures

**Advantages:**
- Efficient compression enabling high quality at low bit-rates.
- Cross platform support MPEG is a well-documented standard enabling different MPEG systems to use the same data files (although how the data is stored in the file may need to be converted).
- MPEG files can be transferred as compressed data in MTS (MPEG Transport Streams) or with SDTI-CP.
- No licensing issues

**Disadvantages:** Users must encode at I-frame only at 30/40 or 50 Mb/s eliminating the advantages of temporal compression. Long GOP MPEG is difficult to edit, trim and manipulate by some systems because each frame does not contain all the data needed to reconstruct that frame. MPEG I-frame only encoding is often used when editing, thus requiring much higher bit rates (similar to M-JPEG). [15]

Another disadvantage of MPEG is the cost of getting it digitized and playing it back. It does require special hardware to digitize video into MPEG, and specific hardware is needed to play back MPEG with any sort of quality. The hardware to digitize MPEG is extremely expensive, starting around $4,000 for a lower-level digitizing board. Another drawback is that even though many IBM-compatible machines come out of the box with MPEG playback hardware, the quality of the playback often varies. Videos that might look great on a Compaq might look horrible on an Acer. Also, MPEG is not compatible with the Macintosh platform, so those users will be excluded from seeing the MPEG video. [15]

The MPEG algorithm is based on the division of the image into small blocks (typically of the size of 8x8 pixels) and the use of the DCT inside these blocks to exclude the spatial redundancy presented in the original image. These methods suffer from the following main disadvantages:
- Only spatial correlation of pixels inside one block is used, and the correlation of the pixels from neighbouring blocks is neglected.
- Under large compression ratios, the blocking structure of image becomes visible; this is called as “blocking artefact” in DCT based image compression.
- The entropy coding of spectrum coefficients from different blocks is independent of each other, which influences the “blocking artefact” in MPEG compression.

**MPEG-1**
Main features, that define MPEG-1 standard are:
- Target bit-rate about 1.5 Mbit/s.
- Typical image format CIF (appendix2), no interlace.
- Frame rate 24 – 30 fps.
- Main application: video storage for multimedia (e.g. on CD-ROM). [16]

First off, the MPEG starts with a relatively low-resolution video sequence (possibly decimated from the original) of about 352 by 240 by 30 frames/sec, but original high quality audio (CD). The images are in colour, but converted to YUV space, and the two chrominance channels (U and V) are decimated further to 176 by 120 pixels. It turns out that you can get away with a lot less resolution in those channels and not notice it, at least in “natural” (not computer generated) images.

A number of techniques are used to achieve a high compression ratio. The first is to select an appropriate spatial resolution for the signal. The algorithm then uses block-based motion compensation to reduce the temporal redundancy. Motion compensation is used for causal prediction of the current picture from a previous picture, for non-causal prediction of the current picture from a future picture, or for interpolative prediction from past and future pictures. The difference signal, the prediction error, is further compressed using the discrete cosine transform (DCT) to remove spatial correlation and is then quantised. Finally, the motion vectors are combined with the DCT information, and coded using variable length codes. [16]

The basic scheme is to predict motion from frame to frame in the temporal direction, and then to use DCT's to organize the redundancy in the spatial directions. The DCT's are done on 8x8 blocks, and the motion prediction is done in the luminance (Y) channel on 16x16 blocks. In other words, given the 16x16 block in the current frame that you are trying to code, you look for a close match to that block in a previous or future frame (there are backward prediction modes where later frames are sent first to allow interpolating between frames). The DCT coefficients (of either the actual data, or the difference between this block and the close match) are quantized, which means that you divide them by some value to drop bits off the bottom end. Hopefully, many of the coefficients will then end up being zero. The quantization can change for every “macroblock” (a macroblock is 16x16 of Y and the corresponding 8x8’s in both U and V). The results of all of this, which include the DCT coefficients, the motion vectors, and the quantization parameters (and other stuff) is Huffman coded using fixed tables. The DCT coefficients have a special Huffman table that is two-dimensional in that one code specifies a run-length of zeros and the non-zero value that ended the run. Also, the motion vectors and the DC DCT components are DPCM (subtracted from the last one) coded.

Sample rates
The source input format for MPEG I, called SIF, is CCIR-610 decimated by 2:1 in the horizontal direction, 2:1 in the time direction, and an additional 2:1 in the chrominance vertical direction. And some lines are cut off to make sure things divide by 8 or 16 where needed. For 50 Hz display standards (PAL, SECAM)
change the number of lines in a field from 243 or 240 to 288, and change the display rate to 50 fields/s or 25 frames/s. Similarly, change the 120 lines in the decimated chrominance channels to 144 lines. Since 288x50 is exactly equal to 240x60, the two formats have the same source data rate. The MPEG-1 syntax permits sampling dimensions as high as 4095 x 4095 x 60 frames per second. In fact MPEG-1 can use any frame size, including CCIR-601 resolutions.

**Compression quality of MPEG-1**
Audio CD data rates are about 1.5 Mbits/s. It is possible to compress the same stereo program down to 256 Kbits/s with no loss in discernible quality. (So they say. For the most part it is true, but every once in a while a weird thing might happen that the loss of stereo information will be noticeable. However the effect is very small, and it takes a trained listener to notice these particular types of effects.) That's about 6:1 compression. So, a CD MPEG I stream would have about 1.25 MBits/s left for video. The video compression ratio from the numbers here can be appreciated to be about 26:1. Of course, there is a loss in compression, but it can be pretty hard sometimes to see the loss.

**Advantages:** Can play in almost any Pentium-class PC or Macintosh with regular CD,CD-RW or DVD drive. No additional software required. Simply click on CD icon to play. Mass distribution on CD-R disc is low.

**Disadvantages:** Picture not as good as MPEG-2 or DVD but very acceptable if played less than full-screen. In order to get a good picture, it does require more bits/second than other codecs. [16]

**MPEG-2**
MPEG is an encoding and compression system for digital multimedia content defined by the Motion Pictures Expert Group (MPEG). MPEG-2 extends the basic
MPEG system to provide compression support for TV quality transmission of digital video.

The MPEG-2 video compression algorithm achieves very high rates of compression by exploiting the redundancy in video information. MPEG-2 removes both the temporal redundancy and spatial redundancy, which are present in motion video.
Main features, that define MPEG-2 standard are:
• Video compression, which is backwards compatible with MPEG-1.
• Full-screen interlaced and/or progressive video (for TV and Computer displays).
• Enhanced audio coding (high quality, mono, stereo, and other audio features).
• Transport multiplexing (combining different MPEG streams in a single transmission stream).
• Other services (GUI, interaction, encryption, data transmission, etc)

The MPEG-2 concept is similar to MPEG-1, but includes some extensions to cover a wider range of applications. However, the MPEG-2 syntax has been found to be efficient for other applications such as those at higher bit rates (e.g. HDTV). The most significant enhancement over MPEG-1 is the addition of syntax for efficient coding of interlaced video (e.g. 16 x 8 block size motion compensation).

Interaction
One of the potential new services, which may be provided by MPEG-2, is the ability to use a return channel to allow the user to control the content or scheduling of the transmitted video/audio/data. This is known as interaction, and is seen by many as the key discriminator between traditional video and MPEG-2. MPEG-2 defines an interaction channel using DSM-CC.

Interaction channels may be used for diverse services including:
• Display and control of small video clips to promote products/future programming.
• Ability to select and pay for Video on Demand (VoD).
• Access to remote information servers.
• Access to remote databases / systems providing home shopping, banking, etc.
• Internet Access.

Profiles and levels
MPEG-2 builds on the powerful video compression capabilities of the MPEG-1 standard to offer a wide range of coding tools. These have been grouped in profiles to offer different functionalities.

There are 4 profiles and 4 levels. The 4 profiles define the colorspace resolution and scalability of the bitstream. The levels define the maximum and minimum for image resolution, and Y (Luminance) samples per second, the number of video and audio layers supported for scalable profiles and the maximum bit rate per profile. In other words:
• Profile = quality of the video
• Level = resolution of the video

The combination of a profile and a level produces an architecture, which defines
the ability of an encoder to handle a particular bitstream (see Appendix 1). The 4:2:2 profile is now an integral part of MPEG-2 Video.

**Typical decoder specifications are:**
- 720 x 576 x 25 fps (PAL CCIR 601) (Appendix 2)
- 352 x 576 x 25 fps (PAL Half-D1)
- 720 x 480 x 30 fps (NTSC CCIR 601)
- 352 x 480 x 30 fps (NTSC Half-D1)

Most decoders will also support MPEG-1:
- 352 x 288 x 25 fps (PAL SIF)
- 352 x 240 x 30 fps (NTSC SIF)

**Advantages:** Very good picture quality, good compression. MPEG-2 gives better picture quality than MPEG-1 at full CCIR 601 resolution (720x480) and at comparable bitrate. Basically, MPEG-2 scales, while MPEG1 does not.

**Disadvantages:** An MPEG-2 codec (decoder software) needs to be installed in the computer for it to play. However, if one already has a DVD drive and software player (WinDVD, PowerDVD etc.) installed in the computer, then the MPEG-2 codec is already present and the video can simply be played through the player.

MPEG-2 takes more CPU power to decompress, and basically has no advantage over MPEG-1 at lower resolutions. For distributing video for the Web, there is no reason to use MPEG-2 compression instead of MPEG-1 because the benefits of MPEG2 aren't realized until the video is in full TV resolution, which means big file sizes.

**MPEG-4**
MPEG-4 is based upon the same technique as earlier MPEG versions. The most important new features of MPEG-4, ISO/IEC 14496, concerning video compression are the support of even lower bandwidth consuming applications, e.g. mobile units, and on the other hand applications with extremely high quality and almost unlimited bandwidth. The making of studio movies is one such an example.

MPEG-4 builds on the proven success of three fields:
- Digital television.
- Interactive graphics applications (synthetic content).
- Interactive multimedia (World Wide Web, distribution of and access to content).

MPEG-4 provides the standardized technological elements enabling the integration of the production, distribution and content access paradigms of the three fields. [18]

**Major functionalities in MPEG-4**
In principle, MPEG-4 does not define transport layers. In a number of cases, adaptation to a specific existing transport layer has been defined:
• Transport over MPEG-2 Transport Stream (this is an amendment to MPEG-2 Systems).
• Transport over IP (In cooperation with IETF, the Internet Engineering Task Force).

**Formats Supported**
The following formats and bitrates are supported by MPEG-4 Visual:
- Bitrates: typically between 5 Kbit/s and more than 1 Gbit/s
- Formats: progressive as well as interlaced video
- Resolutions: typically from sub-QCIF to “Studio” resolutions (4K x 4K pixels)

**Compression Efficiency**
- For all bit rates addressed, the algorithms are very efficient. This includes the compact coding of textures with a quality adjustable between "acceptable" for very high compression ratios up to “near lossless”.
- Efficient compression of textures for texture mapping on 2-D and 3-D meshes.
- Random access of video to allow functionalities such as pause, fast forward and fast reverse of stored video.

**Advantages:** MPEG4 offers improved video quality, support for lower bandwidth, recognizable video in extremely low data rate environments (5–64 Kbps), very efficient compression algorithm for all bit rates.

**Disadvantages:** difficult to get MPEG-4 functioning in real time, demands additional CPU power. [18]

**MPEG-7**
MPEG-7 will complement MPEG-4, not replace it. MPEG-4 defines how to *represent* content; MPEG-7 specifies how to *describe* it.

MPEG-7 defines an interoperable framework for content descriptions way beyond the traditional “metadata”. MPEG-7 has descriptive elements that range from very “low-level” signal features like colours, shapes and sound characteristics, to high level structural information about content collections. MPEG-7 is also unique in its tools for structuring information about content. MPEG-7 and MPEG-4 form a great couple, especially when MPEG-4 objects are used. With MPEG-7, it is now possible to exchange information about multimedia content in interoperable ways, making it easier to find content and identify just what you wanted to use. MPEG-7 information will be added to broadcasts; personal video recorders and search engines can use it, and it greatly facilitates managing multimedia content in often large content repositories. Audiovisual archives are currently hard to search from outside the organizations that own them, because they all employ their own metadata schemes. MPEG-7 will lift that barrier. [19]
**Advantages:** MPEG-7 is a flexible tool, as it is not aimed at any one application in particular; rather, the elements that MPEG-7 standardizes support as broad a range of applications as possible.

**Disadvantages:** MPEG-7 uses as additional tool to MPEG-4 standard for describing the multimedia content data. It does not improve compression efficiency and other features of MPEG-4, just adds to it new functionality. MPEG-7 functions best together with MPEG-4. [19]

**MPEG comparison**

The comparison of the MPEGs in Table 1 contains the MPEG-1 with its most often used limitation (Constrained Parameters Bitstream, CPB), MPEG-2 with its Main Profile at Main Level (MP@ML), and MPEG-4 Main Profile at L3 Level.

*Table 1. Comparison of MPEG standards.*

<table>
<thead>
<tr>
<th>MPEG</th>
<th>1</th>
<th>2</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max bit rate ($Mbps$)</td>
<td>1,86</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>Picture width ($pixels$)</td>
<td>352</td>
<td>720</td>
<td>720</td>
</tr>
<tr>
<td>Picture height ($pixels$)</td>
<td>288</td>
<td>576</td>
<td>576</td>
</tr>
<tr>
<td>Picture rate ($fps$)</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
</tbody>
</table>

**3.4.3 JPEG**

JPEG (appendix 2) is a standard for storing and compressing digital images. JPEG is designed for compressing either full-colour or grey-scale images of natural, real-word scenes. It works well on photographs, naturalistic artwork and the similar; not so well on lettering, simple cartoons or line drawings. JPEG is a *lossy compression* technique for colour images. Although it can reduce files sizes to about 5% of their normal size, some detail is lost in the compression.

JPEG standard defines four **coding modes:**

- **Sequential encoding:** Each component is encoded in a single scan with the scanning order left-to-right and top-to-bottom. The encoding process is based on the DCT.
- **Progressive encoding:** Each scan contains a partially coded version of the image. A “rough” image is quickly decoded and then this is built up using further scans.
- **Hierarchical coding:** Each component is encoded at multiple resolutions. The image can be decoded at a low resolution without decompressing the full-resolution image.
• **Lossless encoding:** Is based on a differential prediction system. This mode provides compression without any loss of quality, at the expense of a considerable reduction in compression efficiency.

**Image compression with JPEG (algorithm description)** [20]

- Discrete Cosine Transform (DCT)
  - Each 8x8 block of pixels treated as 64-point discrete signal, and decomposed into 64 orthogonal basis signals transformation from spatial domain to frequency domain.
  - DCT coefficients (amplitudes of basis signals) obtained.
- Quantization
  - DCT coefficients quantized using a quantization table some coefficients more important than others.
  - Most coefficients becoming “0”.
- Huffman coding
  - zigzag sequence to increase the length of repeated “0”s.

**JPEG compression quality**

For full-colour images, the uncompressed data is normally 24 bits/pixel. The best-known lossless compression methods can compress such data about 2:1 on average. JPEG can typically achieve 10:1 to 20:1 compression without visible loss, bringing the effective storage requirement down to 1 to 2 bits/pixel. 30:1 to 50:1 compression is possible with small to moderate defects, while for very-low-quality purposes such as previews or archive indexes, 100:1 compression is quite feasible. An image, compressed 100:1 with JPEG takes up the same space as a full-colour one-tenth-scale thumbnail image, yet it retains much more detail than such a thumbnail.

**JPEG encoding**

- Supports progressive coding and hierarchical coding.
- Efficient for images that are not too complex (e.g. captured images).
- Performance
  - Lossless mode: compression ratio around 2 – which is not so significant to use in image encoding.
  - Lossy: compression ratio in the order of 20–25. A JPEG image is not perceptually significantly different from the original if it has > 1.5 bits per pixel on average.

**Motion JPEG**

JPEG handles only still images, but there is an extension, Motion-JPEG – a variant of the ISO JPEG specification for use with digital video streams. In motion-JPEG, each frame in the video is stored with the JPEG format.

Unlike MPEG video compression, JPEG video stream does not depend on inter-frame differences and so is suitable for applications where multiple video sources are interleaved (for example multiplexed security cameras).
Instead of compressing an entire video sequences into a single bitstream as MPEG does, Motion-JPEG compresses each video field separately, returning the resulting JPEG bitstreams consecutively in a single frame. This makes direct access to every picture available when editing.

There is no standard for Motion JPEG. Various vendors have applied JPEG to individual frames of a video sequence, and have called the result “M-JPEG”. Unfortunately, in the absence of any recognized standard, they have each done it differently. The resulting files are usually not compatible across different vendors. So if you buy a product identified as “M-JPEG”, be aware that you are probably locking yourself into that one vendor. [21]

Advantages:
- A digital video sequence can be represented as a series of JPEG pictures. The advantages are the same as with single still JPEG pictures – flexibility both in terms of quality and compression ratio.
- M-JPEG is field based, which makes it easy to edit, add transitions, perform off-speed play and manipulate. It was the standard format for several years and still widely used in both on-air play-out and production.
- There is a free library. [22]

Disadvantages:
- The main disadvantage of Motion JPEG (a.k.a. MJPEG) is that since it uses only a series of still pictures it makes no use of video compression techniques. The result is a slightly lower compression ratio for video sequences compared to “real” video compression techniques.
- No standard. Since there were several different codecs available and no published standards for encoding the compressed data stream, each server or editing system had its own unique version of M-JPEG thus you could not easily interchange video files. Also newer compression formats are much more efficient and can produce higher quality video at much lower bit rates. Motion JPEG does not specify maximal picture size or rate or bandwidth usage.

3.4.4 H.261/H.263 Video Coding

The H.261 and H.263 are not International Standards but only Recommendations of the ITU. They are both based on the same technique as the MPEG standards and can be seen as simplified versions of MPEG video compression.

They were originally designed for video-conferencing over telephone lines, i.e. low bandwidth. However, it is a bit contradictory that they lack some of the more advanced MPEG techniques to really provide efficient bandwidth use.
The conclusion is therefore that the H.261/H.263 are not suitable for usage in general digital video coding.

H.261

- Only two image resolutions are supported: Common Intermediate Format (CIF – 352 x 288 Y samples) and Quarter CIF (QCIF – 176*144 Y samples).
- In both cases, the chrominance components have half the horizontal and vertical resolution of the luminance component.
- Frame rate: 7.5–30 fps.
- Bit-rate: multiple of 64 kbps (= ISDN-channel), typically 128 Kbps including audio.
- Picture quality: for 128 kbps acceptable with limited motion in the scene, over a 64 or 128 Kbps ISDN connection substantial data compression is required. QCIF is normally chosen in this case and the source frame rate is restricted to around 10 frames per second.
- Stand-alone videoconferencing system, integrated with PC. [23]

H.263

- Image format: CIF, QCIF or Sub-QCOF.
- Frame rate: usually below 10 fps.
- Bit-rate: arbitrary, typically 20 Kbps for PSTN.
- Picture quality: with new options as good as H.261 (at half rate).
- Software-only PC video phone or TV set-top box.
- Widely used as compression engine for Internet video streaming.
- H.263 is also the compression core of the MPEG-4 standard. [24]

3.4.5 QuickTime

QuickTime is a software video standard developed by Apple that is currently available for playback on both PC and Macintosh. Digitising video into the QuickTime format requires hardware that is extremely inexpensive compared to MPEG. Four thousand dollars can purchase a higher-end QuickTime digitising hardware board, and there are several Macintosh models available that come with built-in A/V boards that will produce acceptable results. Since it is software-based, editing and manipulating QuickTime video is much cheaper than using the MPEG format.

It has a relatively low compression ratio, so it requires relatively high amounts of disk space.

QuickTime movies are normally encoded at very high bitrates. Rates that are too high for dial up modems, and even some 'broadband' connections. That is why it is best suited for short length (60 seconds or less) videos.
Although QuickTime is cross-platform compatible, it requires special playback software on non-native machines.

Advantages
- Good compression (worse than MPEG, better than AVI)
- Strong software support. (Existing Mac apps quickly moving to PCs)
- Default Plugin support by Netscape.

Disadvantages
- Licensing is free, but still an issue.
- Unix support is not as strong as MPEG.
- Dependency on future of Apple. [25]

3.4.6 Wavelet based image compression

Image and video coding methods that use Wavelet transform have been successful in providing high rates of compression while maintaining good image quality. Wavelet based compression competes with DCT based technologies.

The JPEG compression process compress the image as a series of blocks. This is because this technique is memory based, so it needs to limit the size of each block. Because of this, JPEG and related formats suffer from significant and visible degradations when higher compression ratios are used. Recent breakthroughs in Wavelet processing have removed memory limits from image compression. This means that large imagery can be compressed efficiently, without introducing visible artifacts into the compressed image.

The primary steps in wavelet compression are performing
- a Discrete Wavelet Transformation (DWT),
- quantization of the wavelet-space image subbands, and then
- encoding these subbands.

Wavelet images by and of themselves are not compressed images, rather it is the quantization and encoding stages that do the image compression. Image decompression, or reconstruction, is achieved by carrying out the above steps in reverse and inverse order. Thus, to restore the original image, the compressed image is decoded, dequantized, and then an inverse-DWT is performed. [26]

Because Wavelet compression inherently results in a set of multi-resolution images, it is well suited to work with large imagery, which needs to be selectively viewed at different resolutions, as only the levels containing the required level of detail need to be decompressed. Wavelet mathematics embraces an entire range of methods each offering different properties and advantages. For example, it is possible to compress 3 or more dimensional imagery using Wavelets.
The Wavelet Transform (WT) is widely used for image compression, but can be extended naturally to video sequences. There is no standard for video compression based on the Wavelet Transform. Different companies have developed different techniques for Wavelet based video compression.

The Department of Computation, UMIST, UK has for example achieved Motion JPEG alike technology with use of the Wavelet transform. Each image in video sequence is compressed by the Wavelet codec separately without motion compensation. [27] Such video compression algorithm has shown to outperform Motion JPEG in term of compression ratio at the same level of quality.

Another company, IDM, has defined the Motion Wavelets video codec in a different way. This video codec is based on the Discrete Wavelet Transform combined with motion compensation for inter-frame compression. This is the same technique as MPEG uses with the difference that instead of DCT, as in MPEG, the more efficient DWT is used. [28] This video compression algorithm provides higher image quality than MPEG-2 or MPEG-4 techniques at the same compression ratio.

**Advantages:**
One of the most attractive aspects of the Wavelet compression is the ability to make a degraded improvement. The same compression creates different “layers” of details, with different quality, so the quality can be improved by combining more and more layers in the frame, when the bandwidth allows it. [29]

**Disadvantages:**
Wavelet compression has not been widely used because
- DWT operation takes a lot of compute power,
- Historic techniques perform the DWT operation in memory or by storing intermediate results on hard disk. This limits either the size of the image that can be compressed, or the speed at which it can be compressed.
- No standard for video compression.

### 3.4.7 AVI

AVI (Audio Video Interleave) Video format was a production of Microsoft’s for Intel. Currently Microsoft holds the rights to the format. Specifically designed for optimization on the Intel processor chips, the AVI format is most typically a Windows/PC based solution.

**Advantages**
- Native support on PC's means fast rendering.
- Default plug-in on Netscape for Windows.

**Disadvantages**
- Support on Macs and Unix is limited.
• Possible Licensing Issues.
• Mediocre compression.

3.4.8 Summary

The essential properties of video encoding technologies, which were considered above, are listed in Table 2.
### Table 2. Comparison of different video encoding technologies.

<table>
<thead>
<tr>
<th>Format</th>
<th>Frame resolution, rate &amp; compression ratio</th>
<th>Bit-rate</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPEG-1</td>
<td>352x240x30 fps 352x 288x25 fps Up to 4095x4095x60 fps any frame size 26 : 1 – compr. ratio</td>
<td>1.5 Mbit/sec</td>
<td>Higher resolution and frame rates. Better scaling. CD Quality audio. Low data rates.</td>
<td>May require extra hardware. High cost encoding. Non-interlaced (progressive) video.</td>
</tr>
<tr>
<td>MPEG-2</td>
<td>Frame size must be x 16 Interlaced mode: 1440x960x30 Hz Progressive mode: 12080x720x60 Hz Builds on MPEG-1 supporting interlaced video format</td>
<td>4.00 Mbit/sec</td>
<td>Currently the highest resolution and frame rate available. Supports MPEG-1 format.</td>
<td>Higher data rates. Large storage requirements. Requires extra hardware for decoding. Very high cost encoding.</td>
</tr>
<tr>
<td>JPEG</td>
<td>720x485 or 720x 575 Up to 30 fps 20 : 1 – compr. ratio (up to 100:1)</td>
<td>1.5 Mbit/sec</td>
<td>Flexibility both in terms of quality and compression ratio. Free library.</td>
<td>Lower compression ratio for video sequences compared to MPEG No removal of interframe redundancy. Higher compression ratio give lower image quality.</td>
</tr>
<tr>
<td>QuickTime</td>
<td>320x240 fps</td>
<td>high</td>
<td>Inexpensive to compress analogue video. Video can be compressed at low frame rates and picture sizes. Supports low data rates.</td>
<td>Requires Apple platform for encoding and editing. Requires QuickTime plug-in to be installed on the user side to watch the video. Low compression ratio, high bitrates.</td>
</tr>
<tr>
<td>Wavelet</td>
<td>512x512 up to 60 fps 5:1 to 200:1 compr. ratio</td>
<td>16 Mbit/sec</td>
<td>Higher image quality than MPEG or M-JPEG. Compatible with Macintosh, Windows, Unix/Linux.</td>
<td>Requires high computer power.</td>
</tr>
<tr>
<td>H.261</td>
<td>Two picture formats: 352x288x10 fps 352x288x30 fps 25 : 1 – compr. ratio</td>
<td>12.2 Mbit/sec 36.5 Mbit/sec</td>
<td></td>
<td>Low accuracy. Motion vector search is difficult. Propagation of errors Bit rate control.</td>
</tr>
<tr>
<td>H.263</td>
<td>5 picture formats: 352x288x10 Hz 352x288x30 Hz 27 : 1 – compr. ratio</td>
<td>12.2 Mbit/s 36.5 Mbit/s</td>
<td>Improved performance and error recovery.</td>
<td></td>
</tr>
</tbody>
</table>

According to this table, it is obvious that all the formats have their problems.
3.4.9 Conclusions

Hence the H.261/H.263 recommendations are neither international standard nor offer any compression enhancements compared to MPEG, they are not of any real interest.

Considering QuickTime technology, due to poor support by Unix/Linux and lower compression quality then MPEG, this software can not compete with MPEG.

MPEG remains the best compression technology in many cases considering image quality and compression ratio. Hence comparable video quality can be achieved with MPEG-2 at 1/2 to 1/3 the bit rate of Motion JPEG, it has some weaknesses. This standard is build so that all frames, excluding a key frame, depend on information in the previous frame. It can result in a loss of a considerable part of video sequence. Meanwhile, if the Motion JPEG is used this cannot happen. JPEG compresses each image separately, so if one of them is lost it does not influence the others.

Wavelet technology appears interesting since it proposes a more efficient way to compress a single image and the possibility to choose video encoding like Motion JPEG. On the other hand there is no free software available, which uses Wavelet technology.

All these considerations show that the right choice for this project is the Motion JPEG video encoding technology.

3.5 Video Server technology is the answer

After video compression technology is chosen, another problem emerges: how to transmit the compressed video information.

Video server technology offers a solution for both compression of the video information and transmission.

There are two primary possibilities in choosing a video server:
- hardware-based MPEG encoders or
- software-based encoders combined with streaming servers.

It may seem natural to prefer the streaming servers, since those servers are often free. Furthermore, streaming servers can conserve bandwidth, since it is often possible to send decent quality images even over a dial-up connection – a trick hardware encoders can not perform.

On the other hand, hardware encoders provide the ultimate in quality. While the MPEG encoders have a higher entry cost and greater cost per seat than do the streaming servers, they have the advantage of being one-box solutions that need
no additional hardware outside of the video source and a network connection. To deliver video to every desktop within an organization, MPEG is probably not the solution. But conversely, streaming servers may not be the best solution for providing a high-quality video signal to boardrooms or sales meetings [30].

3.5.1 Video streaming server

Video streaming is based on three components:
- Digital video compression. It eliminates repetitive scenes and excessive detail.
- Transmission of data packages that are read by the user as they arrive.
- The use of buffers or memories to minimize general information network delays.

3.5.2 Different streaming techniques

Streaming Video is a technology that allows watching video almost instantly on the Internet without having to download the whole file to computer first. This is the opposite to downloading, which means that the whole video file is downloaded from the Internet to the remote computer before it starts to play back.

Streaming works in following way: the remote computer downloads the first few seconds of video (called a buffer) to the disk; then the video starts playing as it continues to download in the background.

There are some different techniques that are used in streaming video. Sometimes, only streaming of content in real-time through UDP is called a “real streaming”. In another context the term is used to refer to any video that plays on the Web.

The biggest difference in using these two transmitting protocols is that true streaming through UPD only works when there is enough bandwidth to play the video in real-time. Progressive download transfers, that uses http, at the available speed download as much of the file to the receiver disk as necessary to act as a buffer before playback begins. Progressive download can ensure high-quality playback at any bandwidth, but potentially with a very long delay. [31]

3.5.3 Hardware based video servers

A hardware-based server is a powerful computer that can carry out multiple operations simultaneously. It uses software that quickly responds to customer requests and the appropriate files are subsequently delivered.

A video server includes one or more analog video inputs, image digitiser, image compressor and Web server with network/phone modem interface. Video servers digitise analog video sources and distribute digital images over a computer network, turning an analog camera into a network camera. It is ideal for integration with existing analog CCTV (closed circuit television) system.
Using built-in serial ports, the video server can control equipment such as Pan/Tilt/Zoom cameras. Inputs can be used to trigger the server to start transmitting images. Servers equipped with image buffers can send pre-alarm images.

A video server can also be connected to special cameras such as super sensitive black/white cameras, miniature cameras or microscope cameras.

### 3.5.4 Conclusion

Although the hardware-based video servers provide ultimate quality and one-box solution there are some advantages in streaming video servers that make them a more preferable choice. A streaming video server is software, often free, and most important, more flexible solution than a hardware video server. So the choice will be a streaming video server. The next challenge is to choose a right free video server, which suits the system and will be the best solution.

### 3.6 Providing security of video information

User interaction with domestic cleaner Trilobite involves transmission of video information through the Internet. With all the benefits the video can provide, there are some problems that arise at the same time. The Internet is accessible for a wide public all over the world. So, the reasons for security over the Internet and the Web are obvious. As far as a person gets the information he needs through the Internet, anyone else can get it as well. The video information that a remote user is going to view is confidential and extremely sensitive. There is no doubt that the user does not want to allow other people to view his apartments and his private environment. This means that there is a need to secure the Web pages, which present this video information.
To protect information completely, there are numbers of technologies to use, i.e. VPN, SSL, S/WAN (Appendix 2). Many of them are expensive and unfortunately, they do not always provide security of the explored information. The current work does not have as a goal to provide complete protection of the video information. Although a video of one’s own home is sensitive information, a simple solution may satisfy.

In this case, a login system can be a simple solution for preventing unauthorized access to some Web pages. For such system, a HTTP Authentication can be used. In Basic HTTP Authentication, the password that is passed over the network is not encrypted but is not in plain text (data over the internet that can be understood without any special effort is called a plain text) – it is “uuencoded”. Anyone watching packet traffic on the network will not see the password in the clear, but the password will be easily decoded by anyone who happens to catch the right network packet. So basically this method of authentication is roughly as safe as telnet-style username and password security.

For an authentication system to function there are some modules required to be build:
- A database with valid users.
- A dialog between the user and the system, where the database is used to authenticate the user.

One solution can be using MySQL as a database and PHP as a CGI script that handles the dialog between the user and the system using MySQL.

### 3.6.1 MySQL

MySQL is a very fast and very functional Database System. There are a number of other Web based databases:
- Access 2000
- Adabas 10.01.00
- SAP DB 7.3.0.29
- SAP DB 7.3.0.29 (ODBC)
- IBM DB2 5
- EMPRESS Version 6.10
- FrontBase Version 2.1
- Informix 7.30C1
- Interbase 6.0Beta
- Microsoft SQL server 7.00.842
- MIMER 8.2.0C
- mSQL server Version 2.0.10
- Oracle 8.1.6.0.0
- PostgreSQL 7.1.1
- SOLID Server -v.02.30.0026 (Linux ix86)
- Sybase enterprise 11.5 NT
Although they have some good properties, which are preferable in some circumstances, neither of them can compete with MySQL concerning speed and functionality.

MySQL offers following advantages over most of the databases mentioned above:
- faster
- more efficient
- more robust
- has a much larger user base
- the code is more tested
- more stable
- works on more platforms
- has simpler commands
- is supported by more programs
and many others. [32]

There is a research that shows MySQL features and compares it to all other available databases showing advantages of MySQL over other databases [33]. At the same time, MySQL suits perfectly the Linux platform.

The major disadvantage of using MySQL is its lack of ability to cope with a high traffic on a webpage. If the number of users connecting to MySQL builds up, the system seems to have a lot of problems. In the current project this disadvantage cannot be of great importance because the number of users of the system will be limited to a number of family members, that probably do not increase a number of 10–15 persons, and the probability that there are more than 2–3 persons using the system at the same time is little.

### 3.6.2 PHP

With continuously expanding of the Internet there are numbers of communications technologies available in a network environment to execute different requests. The most widely used Web browser is currently the Hyper Text Transfer Protocol (HTTP). Different requests can be launched by some dynamic script, such as PHP, which is equivalent to Communication Getway Interface (CGI) language. Through the Hyper Text Makeup Language (HTML) form, a request can be passed from a client to a server to execute a process and perform some predefined actions on the server side. A dynamically generated HTML page with new information will be returned to the client.

PHP is designed to be fast and was written to do database work. [34] PHP is a server side HTML embedded scripting language, in other words, PHP is embedded in the HTML files and the server does the work of translating the PHP script. This means that it is totally platform and browser independent. PHP is free for download and works very well on all of the Unix-like, Linux and Windows platforms.
PHP is an established server-side scripting language for creating dynamic Web pages. As a language that has been designed expressly for the Web, it brings many features that a software developer can appreciate:

- Quick development time.
- Very high performance.

In addition, PHP supports all major platforms (UNIX, Windows and even mainframes), and features native support for most popular databases. In a nutshell, PHP is fast, free, portable, and tightly integrated with MySQL. Apache webserver is usually shipped with a built-in interpreter for PHP (a module called mod_php). [35]

Comparing PHP with other server-side languages as JSP shows clear advantages of PHP in performance. A recent survey in ZDnet's *eWeek* online publication found that PHP is as much as 3.5 times faster than JSP. In the Web environment, this makes JSP a significantly worse solution – because it can perform less transactions per second, and features considerably slower response times. [36]

Another server-side language is Perl. It is a CGI script, that was used as base for creating PHP. Perl is more mature language than PHP and can be used to perform nearly any task. For current project purposes there would be no difference in functionality or performance depending on whether PHP or Perl is used. [37]

To conclude, both languages are useful. When using mail forms, feedback forms, or anything that needs one time processing, then Perl CGI is a good choice. Meanwhile, if the goal is to build dynamic HTML to make a complex site integrated with a database, then PHP is to prefer. [38]

### 3.6.3 Comparing ASP with PHP

Active Server Pages (ASP) is Microsoft's technology for displaying dynamic Web pages. ASP supports multiple programming languages; the most commonly used is VBScript. PHP is the open source alternative to ASP that runs on multiple operating systems, including Linux and Windows.

PHP overcomes ASP in speed. ASP is built on a COM-based architecture. When an ASP programmer uses VBScript, he is running a COM object. When he writes to the client, he is calling the Response COM object's Write method. When he accesses a database, he uses another COM object to do so. When he accesses the file system, another COM object is called. All this COM overhead adds up and slows things down.

In PHP modules, everything runs in PHP’s memory space. This means that PHP code will run faster because there is no overhead of communicating with different COM objects in different processes.
ASP is integrated with other databases as Microsoft Access, SQL Server 7 and Oracle instead of MySQL. MySQL’s properties comparing with these databases have been considered above.

So, PHP integration with MySQL becomes one of the most important properties. There are more PHP advantages over ASP mentioned in research “PHP versus ASP” that are more or less the concern of this project. [39] All these PHP features and PHP’s compatibility with Linux OS and the Apache server show PHP to be one of the best existing solutions.

3.7 Creating a reliable robust system

A system can be called robust if it prevents all abnormal program termination. A robust program should terminate either when it is successfully completed or with a meaningful error message. A robust program should detect any possible error and gracefully recover from it. A robust program should not depend on the robustness of the internal subprograms.

There are some basic errors that usually cause abnormal program termination:
- Compile-time program errors.
- Run-time program errors.
- Run-time exceptions.
- Run-time user errors.

All these can in their turn be divided into
- Recoverable error conditions and
- Unrecoverable error conditions.

If there occurs an unrecoverable error a program should issue an error message and cleanly shut down. [40]

3.7.1 Compiled-time program errors

Compile-time program errors can be all possible syntax errors. There can be colons missing, parse errors, wrong spelling and all types of using wrong syntax. Compile program errors can be detected quite easily by testing the program, due to the chosen script language. PHP has a strong error detecting support. Because PHP is not a compile language, compile-time errors can only be detected when a program runs. So they belong to run-time program errors in this case.

3.7.2 Run-time program errors

In case of using PHP as a script language both syntax errors and semantic errors become run-time program errors. Syntax errors are easily detected due to PHP error support. If a colon, for example, is missing an error message “Parse error at
line …” is issued. Once fixed, syntax error cannot appear again depending on which situation occurs.

It is worse with semantic errors. A program with semantic errors can behave differently in different situations. An example of a semantic error can be the following function:

```c
function divide (int n, int m)
    return n/m;
```

This function works well in many cases until an exceptional case occurs when the sent input value m = 0. Then the program will collapse. It was a simple example of a program that is not robust due to containing a semantic error. Other semantic errors can be infinite loops, wrong conditions, wrong expected values in internal program calculations, etc. Multiple testing of the program and a thorough tracing of emerging errors can prevent those semantic errors.

### 3.7.3 Run-time exceptions

Run-time exceptions occur when an error happens during program processing by e.g. overflow. Following errors belong to run-time exceptions:

- network failures,
- missing configuration files,
- not enough memory on the disk,

etc.

If the disk, for example, becomes full or fails, it can cause losing data. Writing to a file when the disk is full may cause the file to become truncated and all its contents lost.

In the current project there are specific run-time exceptions that have to be considered:

- Internal network failure – connection between video camera and video server (PC) is broken.
- External network failure – connection between video server (PC) and the user client broken.
- The video server cannot start.
- The video server cannot transmit video data.

If the run-time exceptional error is unrecoverable, the program has to terminate with an error message, which informs what kind of error has happened. In the current project the first and second errors are unrecoverable. So in the case they occur the program should issue a suitable message and cleanly shut down. In two other cases the program does not need to terminate. The appropriate solution in those situations is to inform the user about the error and give him the possibility to choose whether to continue without receiving the video information or to
logout and leave the system. All these cases are worked out and represented in MSC (Message Sequence Charts), which describe different use cases. [41], [42]

3.7.4 Run-time user errors

If we call well-defined set of input values as program’s standard domain, then input values from the complement of the standard domain, will be called the program's exceptional domain. So in a robust program the output from both standard and exceptional domain should be well defined. Most programs are concerned about only the output of the standard domain. Therefore violation of the output of the exceptional domain yields in incorrect result when an erroneous situation has occurred. In worst case incorrect results are not recognized, for instance, because they are sent immediately as input into other procedures or they approximate correct values in some way. The figure below illustrates how a robust program functions.

To prevent abnormal program termination caused by wrong user input or inappropriate user action it’s necessary to consider all possible user actions in all possible situations that can appear. For this purpose the construction of use cases and implementing them into the Message Sequence Charts (MSC) is a good solution. The program has to check all user performed actions or inputs. In case of a recoverable erroneous situation the program should emit a warning error message and give the user a possibility to correct the error and continue to run the program. If an unrecoverable error situation has occurred, the program should emit an error message and gracefully terminate. The MSCs that represent user cases illustrate how robustness is implemented in the current project.
Figure 6. A robust program handling user input.
4 Implementation

This section will present how all principles considered above was implemented.

4.1 Chosen video server

Considering all moments discussed above, the chosen video server must have the following features:

- A software streaming video server.
- Use of JPEG compression technology adapted for video stream.
- Free to download.
- Compatible to Linux OS.

In the number of existing streaming video servers available on the Internet the choice has fallen on the software Camserv, last version from Sun Sep 15 15:00:00 PST 2002.

The Camserv streaming video server has following features:

- Transmission protocol – HTTP_IMG protocol for transmitting images via the Internet.
- Driver for Video4Linux.
- Input file format – all file formats defined for Video4Linux (both analog video and digital video).
- Compression algorithm – JPEG.

This streaming video server is implemented in C programming language. It creates a video from a stream of still images encoded into JPEG format. Then the Camserv sends the video stream via a socket to a Web client at a fixed interval. The C program streams this video to all clients that are connected to this Web server.

Camserv software was successfully downloaded, compiled and configured. Afterwards the Camserv could directly run.

4.2 Database

To provide the possibility to authenticate a user, a storing module – a database – should be constructed. As it was considered in chapter 3.6, MySQL server is the most suitable program to build such a database for current project. So, MySQL software was installed on Apache HTTP server and the database named “users” for storing information about users was successfully constructed.
The following information about a user is needed:
- User name.
- Password.

The database “users” consists of a single table with two attributes: “user_name” and “password”.

The user name is a primary key and should have a unique value for every table. This means that two users cannot have the same user name. Both user name and password have to be “not null” attributes to be inserted into the database.
4.3 Page configuration

In Figure 7 the State Chart Diagram shows all main Web pages that build up the site and their interaction with each other in this project.

Figure 7. Page configuration – State Chart Diagram.

Square boxes represent Web pages, rombs represent conditions that can be filled or not. Double frames show the start state and the final state of the system.
4.4 Implementation of video security and robustness illustrated by means of use cases and MSCs

Describing user cases and building MSC – Message Sequence Chart – are excellent tools to work through a program, different user inputs and prevent possible run-time user errors and run-time exceptions. Figure 8 illustrate all possible user cases. The MSCs show user actions in detail and the system response to them.

The basic use case: Using remote robot control system
It contains all other use cases. Figure 8 shows the MSC in this use case.

Preconditions: a user is on the login page.
Post conditions: the user had performed all actions she intended to do and successfully logs out. The system displays the logout page.

The login page displays a form with two fields in which to put the user name and the password, two buttons and a link to the registering page (see figure 9). There are two alternatives on the login page:

- First: the user can login by her user name and password if she is registered in the system database.
- Second: the user can register herself as a new user and get a valid user name and password.

Figure 9. Login window.

Use case: Login

Figure 10 illustrates a MSC to this use case.

Preconditions: the login window is active (see figure 9).

Post conditions: the login is successful.
This use case illustrates how security for video information is implemented in the current project. Only the authorised user has permission to enter the system. The authentication of users is performed by the system through comparing the user name and password, which the user inputs, to user names and passwords registered in the database.
There is a field for user name and a field for password that have to be filled. By clicking on the button “Login” the user sends his password and user name to the dynamic script located on the HTTP server to check their validity.

Robustness principles are implemented in the following way: if any of the fields is not filled or the user inputs user name and password, which are not registered in the database, the program returns the user to the login page. The error message “Wrong user name or password” is issued and the user has a possibility to input user name and password again and continue. This can be repeated infinite times until the user input will be correct. When login is successful, the system updates the number of users, establishes a session that identifies the user and the user can go to next use case “Choosing the action”.

**Use case: Choosing the action**

*Preconditions:* Login is successful.

*Post conditions:* Either the window for watching, the robot, the window for controlling the robot or the window for logout is displayed.

The figure 13 shows the MSC that illustrates use case “Choosing the action”. In this use case the user is supposed to choose what he is going to do. There are two possibilities: to control the robot or merely to see what it is doing. In both cases the video server has to be on, so the dynamic script checks if the video server is started, and if it is not, starts it through the system command.

Here a run-time exception can occur in case the video server would not start for some reason (see figure 13). Handling this exception adds robustness to the video system because it prevents the system from crashing and gives the user possibility to continue. It implements another principle of robustness: the whole program should not depend on the robustness of the inner programs. So even if the video server does not work properly, the main program does not crash, but recovers with an error message to the user and the possibility to continue.

If no exception occurs, the program continues. It welcomes the user by his user name, informs that the user is logged in and how many users are in the system at the moment. The program checks if there is a user who is already controlling the robot. If there is, the window for choosing the action is displayed and the user is informed that he can only watch the working robot or logout (see figure 11). If there is no one who controls the robot at the moment, the user gets the possibilities either to control the robot, only to watch the robot or to logout (see figure 12).

The user inputs in this use case are limited and do not influence the program running. That is why they cannot cause any run-time user error.
Welcome, Julia! You are logged in.

There are 2 users attached.
There is already a user, who controls the robot.
You can only watch the robot.

Logout

Figure 11. Window for choosing an action (only watch possibility).
Welcome, Julia! You are logged in.

There are 2 users attached.
Here you can choose
to control the robot or to watch it.

Figure 12. Window for choosing an action (control and watch possibilities).
Figure 13 (part I), MSC – Choosing the action.
Figure 13 (part II). MSC – Choosing the action.
**Use case: Control the robot**

*Preconditions:* The user is logged in into the system and has chosen to control the robot.

*Post conditions:* The user is successfully controlling the robot and getting video feedback.

Figure 14 represents the use case “Control the robot”.

After the user has chosen to control the robot the system displays the window for remote robot controlling (see figure 15). Condition to continue is no network connection failure, which means that the window for robot controlling is displayed. Otherwise the system goes into an exceptional case “Network connection failure” (see Exception: Network connection failure). If no exception occurs the system notes that this user is controlling the robot. The script notifies too that this user controls the robot. At the same time, as soon as the controlling
window is displayed, the video server transmits the video and dynamic script presents it on the page. This process continues until the user chooses to logout. Here the only user inputs are choosing to watch and to logout. These inputs cannot be done at wrong time, so user run-time error cannot occur in this use case.

![Manual cleaning](image)

**Figure 15.** Window for manual control of the robot.
**Use case: Watch the robot**

*Figure 16. MSC – Watch the robot.*

**Preconditions:** The user is logged in into the system and has chosen to watch the robot.

**Post conditions:** The user is watching the robot and getting video stream.

Figure 16 illustrates by MSC a use case “Watch the robot”. When the user has chosen to watch the robot the system displays the window for watching (see figure 17). A run-time exception “Network connection failure” can occur while the system is trying to display the page (see Exception: Network connection failure). If it does not, the page is displayed as a condition for the program to continue to run. The video server transmits the video, which is represented by dynamic script in the webpage for remote user. The user can interrupt viewing at any time he likes by logging out.

Here the only user inputs are choosing to watch and to logout. These inputs cannot be done at the wrong time, so a user run-time error cannot occur in this use case.
Figure 17. Window for watching the robot.
Use case: Logout

The user is logged into the system

logout

Update number of users

Is there other users left?

no

Close the video server

This user is controlling the robot

Update controlling user

Display logout page

Logout page is displayed

Login

Figure 18. MSC – Logout.
Preconditions: The user is logged in into the system and has chosen to logout.
Post conditions: The user is successfully logged out and the logout page is displayed.

Figure 18 illustrates how a user can logout from the system. When a user chooses to logout by clicking an appropriate link on some webpage, the system logs him out by altering number of users who are in the system and closing a session that identifies this user.

An important part of the video system security in this project is implemented here. When the last user has left the system, the video server shuts down. This is a necessary measure to achieve improved security for exploring transmitted video information. The login system provides a user with the control of access to sensitive video information. When a user leaves the system there is no possibility to go back without logging in again. But in some Web browsers it is possible to go back to previous page and view it. So closing the video server is crucial in this case. It means that when someone who is not authorised tries to view a webpage with video, he cannot get any video stream if the video server is down.

To close the video server the system checks first if there are other users left in the system. If there are not, the video server shuts down, otherwise the video server continues to run. Next, the system checks if the current user had a control over the robot. In case he had the system updates the information about controlling the user by cancelling him from control. At last, a logout page is displayed (see figure 19) in case that no exception “Network connection failure” (see Exception: Network connection failure) has occurred. The logout page issues a farewell message and a link to log in again.
Use case: Register new user

Preconditions: The login window is displayed.
Post conditions: Registration is successful and user can login.

Figure 21 shows MSC that illustrates the use case “Register a new user”. After the user has chosen to register him, a window for registering (see figure 22) is shown if no exception “Network connection failure” occurs. To register him, the user has to input a user name, a password and a repeated password in a form. Wrong user input does not cause any program error but handles in the exceptional case “Wrong input” (see figure 23). Any possible user input is considered there and the possibility to correct errors and continue the program is provided for the user.

Program robustness is achieved here by predicting any possible user input and handling its consequences. For instance, a primary key in the database tuple that contains information about users is user-name. If the program tried to put in a new
user into the database with the user name that already existed this would not have any effect. That is why the dynamic script checks first if the user name is already in the database and only then puts it there. The same principle concerns passwords.

When the system had got all right values of the user input, it inserts a new user in the database with all valid users. After the user is successfully registered, he can choose to login.

Figure 21. MSC – Register a new user.
Figure 22. Window for registering a new user.
Exceptional cases

Exception: Wrong input
Preconditions: The window for registering a new user is displayed and the user input wrong.
Post conditions: The window for registering a new user is displayed.

Figure 23 illustrates the exceptional case “Wrong input”. There can be some different user errors when registering a new user. For instance, if the user name is absent, the user is sent to the same registration page with the error message “You have to input a user name”. This process can continue for an infinite number of times until the user types the correct user name in the user name field. The same process occurs when password fields remain empty. If two passwords do not match each other, the user is also sent back to the same page with an appropriate error message. It is repeated as long as the user does not fill in two passwords that are equal. The last user error can be that the given user name already exists in the database. In this case the system informs the user that the chosen user name is in use and asks the user to choose another until the chose is successful and the new user name does not exist in the database.

Exception: The video server could not start
Preconditions: The video server could not start.
Post conditions: Either the logout window or the window for controlling the robot is displayed.

When the video server does not start, the system displays a window with an error message that the server is down and there is no possibility to get video stream from the robot. The system checks if there is a user who controls the robot. In case there is, there is nothing more to do, because it has to be only one user who controls the robot, so the user is invited to logout. If there is no one who is controlling the robot at the moment, the user is invited either to control the robot without video feedback or to logout (see figure 24).

Exception: Network connection failure
Preconditions: There is no connection between a remote user and the HTTP server.
Post conditions: En error message is displayed.

This exceptional case can occur any time when a user is connected to the robot through the Internet. Especially clear network connection failure can be detected when user is trying to download a new page. In this case the browser issues an error message that informs the user about network connection failure.
Figure 23. MSC – Exception: Wrong input.

- **No user name or no password**
  - You have to put in a user name/password
  - Submit new input

- **Two passwords do not match**
  - You have put in different passwords
  - Submit new input

- **Is there the same user name in the database?**
  - **yes**
    - This name is in use, please choose another one
    - Submit new input
  - **no**
    - Loop back to previous step
Figure 24. MSC – Exception: The video server cannot start.

User

Script

System

The video server could not start

Displays error window

The error window is displayed

Sorry, the video server could not start, you cannot get video

Is there a user who controls the robot?

opt

The robot control device is busy with another user

yes

Logout

no

alt

Control the robot

Logout
4.5 Conclusions

The use cases illustrated by MSCs are effective methods to improve the robustness of the system. With their help, all user actions and all system responses on them become well documented, which is one of the robustness principles. In the current project the MSCs show that many of the robustness principles are implemented. Considering security the basic principles are implemented, and the more complicated security system can be developed in future works.
5 Result, conclusions and future work

This project was focused on finding the best technology for building a system for transmitting a video stream from a mobile autonomous robot (Trilobite) to a remote user. For achieving the best result many different techniques were analyzed.

5.1 Conclusions – project achievements

The most suitable video compression technology considering the current project is JPEG compression. It provides sufficient quality of the video stream compared to other compression techniques. Even if it does not overcome all of them (i.e. MPEG-2) in video performance it helps to achieve a more robust video system due to its flexibility, which is not presented to the same degree in other technologies.

To transmit video the video streaming technology was chosen. The streaming video server Camserv was installed and thoroughly program testing showed that it worked perfectly. The properly functioning system shows that all parts were chosen and combined in the right way.

The chosen dynamic script is PHP that is functioning properly with MySQL database. Together they build a login module that provides the video transmitting system with security.

The robustness of the whole system is achieved by going through all robustness principles by means of use cases and related MSCs. Most of them were implemented in the project creating a robust video transmitting system.

On the whole, the project can be considered as successful because the main goal has been achieved and user interface for controlling the Trilobite has been provided with a video, presenting a robot view. A user has got “an eye” on the other side of the Web when controlling his domestic cleaner Trilobite through the user interface on the Internet. This “eye” gives the user a feeling of presence on the other side of the Internet, in the room where the robot works. By providing a user with video information the user interface becomes more usable and flexible that considerably enhances the Trilobite’s autonomous behaviour.

5.2 Future work

Although the project has achieved good results, there are many moments that can be improved and developed. These moments could not be included into the current project due to limitations of time.
5.2.1 Providing an upper view over the robot

According to the tasks of the current project, a video feedback from the Trilobite is managed in the way that the user can see what the Trilobite sees but cannot see the robot itself. This can often be very disorienting. That is why one of the directions for this project development in the future can be to get an upper view of the Trilobite. This could be achieved by placing an overhead video camera in each room in the apartment. In this way the user would get an overview that includes the Trilobite itself. Such an upper view would solve some problems that can appear if a controlling interface provided only with a “robot’s eye” view. In the case the robot is standing in the corner of a room, it may not be possible to recognise which room or which corner it is if the user interface is supplied by only the robot’s view. An upper view would sufficiently increase the users feeling of being in charge of the system and having control of the situation. On the other hand this way of development demands external resources (a camera in each room). That could be avoided by using another method.

This method is the simulation of an upper view over the robot and can present robot movements in apartments. Instead of having cameras in each room, a map of the apartments could be constructed. A picture of the robot could be placed on this map and each time the user gives the robot a command, the simulated robot would move in the predicted direction. In this way the user would also have an upper view with all its benefits.

5.2.2 Increasing robustness of the system

Another direction of the project development is creating a system which is more robust. For instance, the situation when a client who is logged in into the system disappears because of network connection failure or some other reason was not considered in this work. Handling this problem can be a future development of the system.

According to the current project state, the first user who wishes to control the system gets the control; all others will have to wait until the controlling system is free again. Who ever gets the control next depends on which one is faster and more updated, because there is no queue for users waiting to control the Trilobite. Creating such a queue may be considered as one more development of this project.

5.2.3 Remote portable controller

During the development of this project, some other ideas came up. One of them is the future security enhancement. This can be provided in quite a different way than it was done in this project. The whole system configuration can remain the same excluding the remote client. The difference is that the client does not have to be a PC, as it is supposed to be in the current project. The client can instead be a little portable remote controller with a screen for presenting the video information transmitted from the robot (see figure 25). A benefit of this solution is that the
user does not have to explore the sensitive information about her private environment by calling the Web page in her Web browser on the PC or any other computer. This solution also provides the user with the possibility to be independent from any PC. The benefits of such a portable remote control are obvious thinking of mobile telephones. There are stable phones everywhere: in every apartment, office and even on the streets. Nevertheless, the comfort of having a mobile phone cannot be doubted. It is the same about having a mobile remote controller for a domestic cleaner: the user is not dependent on having the Internet in the neighbourhood. Sitting on the train or in the car, it will still be easy to control the Trilobite just by picking up a mobile controller, that does not have to be bigger than a Palm organiser, from the pocket.

![Remote controller with video screen as a remote client for the video transmitting system.](image)

*Figure 25. Remote controller with video screen as a remote client for the video transmitting system.*

The remote portable controller can consist of a display for presenting a view from the domestic cleaner and a set of controller buttons.
References

All web references last visited 20 December 2002.


From MPEG-1 to MPEG-21: Creating an Interoperable Multimedia Infrastructure. 
*International Organization for Standardization (ISO/IEC JTC1/SC29/WG11 N4518)* 
Pattaya, 2001

[13] K. Sam
*Report on Making Graphics and Interactive Demos, MPEG.*
University of British Columbia, 1996
http://www.geog.ubc.ca/vgd/resources/ksam/project1/mpeg.htm

*Digital Video, MPEG and Associated Artifacts.*
Departments of Computing & Electrical Engineering, 1996

[15] J.L. Mitchell (Editor), W.B. Pennebaker(Editor)
*Mpeg Video: Compression Standard (Digital Multimedia Standards Series).*

[16] International Organization for Standardization, L. Chiariglione
http://mpeg.telecomitalialab.com/standards/mpeg-1/mpeg-1.htm

[17] International Organization for Standardization, L. Chiariglione
*Short MPEG-2 description, ISO/IEC JTC1/SC29/WG11 N MPEG 00/, 2000*

[18] International Organization for Standardization, Editor: R. Koenen
*MPEG-4 Overview - (V.21 – Jeju Version), ISO/IEC JTC1/SC29/WG11 N4668, 2002*

[19] International Organization for Standardization, 
Editor: J. M. Martínez (UPM-GTI, ES)
*MPEG-7 Overview (version 8), ISO/IEC JTC1/SC29/WG11*
Klangenfurt, July 2002

[20] Steinmetz, Nahrstedt
*Fundamentals of Multimedia Computing and Communications,* 
(JPEG encoding algorithm description), *IEEE Computer,* 1995
http://www.cs.sfu.ca/undergrad/CourseMaterials/CMPT479/material/notes/Chap4/Chap4.2.html

[21] Moving Picture Experts Group web site at:
http://www.faqs.org/faqs/jpeg-faq/part1/
[22] Media Technique
Motion JPEG video CODEC Library, 2002

[23] P. Cherriman, L. Hanzo and R. Lucas
ARQ-assisted H261 and H263-based Programmable Video Transceivers, 1996

[24] L.J. Riley
H.263 video coding, 2002

[25] R. Lainhart
Using Final Cut Pro and the Sorenson Video Codec to Prepare QuickTime Video for
Web Distribution in Video Compression, 2002

[26] S. Murthy
ECW (Wavelet compression for Digital Imagery and Image Web server from Earth
http://www.gisdevelopment.net/technology/ip/techip0003a.htm

[27] Mitsuharu Tokunaga, Vu Tuong Thuy
Wavelet-Based Clustering Method to Detect Building in Urban Area from Airborne
Laser Scanner Data
http://www.co.umist.ac.uk/~dtm/VIPROM2000.pdf

[28] Dr.T.Morris, D.Britch
Intra-frame Wavelet video coding
International Workshop on Video Processing and Multimedia
Communications, 2000

[29] R. Serber, E. Brosh, S. Chaikin
Video over the InterNet, 1996.

http://www.networkcomputing.com/1306/1306f1.html

[31] Ben Waggoner
Choosing a Streaming Video Technology


[33] Michael Kofler
[34] R. Lerdorf, K. Tatroe

[35] L. Welling, L. Thomson

[36] H.E. Williams, D. Lane


[38] P. Lambert, *Perl versus PHP for Web Design*


[40] P. Fradet, M. Sudholt
*An aspect language for robust programming*, ECOOP Workshops, 1999

[41] M. Bishop
*Robust Programming*, Department of Computer Science University of California at Davis, 1998

[42] M. Bishop
Appendix 1
Profiles and levels of MPEG-2

Level description

<table>
<thead>
<tr>
<th>Level</th>
<th>Max sampling dimensions fps</th>
<th>Pixels/sec</th>
<th>Max bitrate</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>352 x 240 x 30</td>
<td>3.05 M</td>
<td>4 Mb/s</td>
<td>CIF, Consumer tape equivalent</td>
</tr>
<tr>
<td>Main</td>
<td>720 x 480 x 30</td>
<td>10.40 M</td>
<td>15 Mb/s</td>
<td>CCIR 601, studio TV</td>
</tr>
<tr>
<td>High 1440</td>
<td>1440 x 1152 x 30</td>
<td>47.00 M</td>
<td>60 Mb/s</td>
<td>4 x 601, consumer HDTV</td>
</tr>
<tr>
<td>High</td>
<td>1920 x 1080 x 30</td>
<td>62.70 M</td>
<td>80 Mb/s</td>
<td>Production SMPTE 240M std</td>
</tr>
</tbody>
</table>

Profile description

<table>
<thead>
<tr>
<th>Profile</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
<td>4:2:0 sampling, I/P pictures only, no scalable coding</td>
</tr>
<tr>
<td>Main</td>
<td>As above, plus B pictures</td>
</tr>
<tr>
<td>SNR</td>
<td>As above, plus SNR scalability</td>
</tr>
<tr>
<td>Spatial</td>
<td>As above, plus spatial scalability</td>
</tr>
<tr>
<td>High</td>
<td>As above, plus 4:2:2 sampling</td>
</tr>
</tbody>
</table>

Only a limited number of profile/level combinations are recommended in the standard. These are summarised in the following table:

PROFILE/LEVEL

<table>
<thead>
<tr>
<th>PROFILE/LEVEL</th>
<th>Low</th>
<th>Main</th>
<th>High-1440</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Main</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>SNR</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spatial</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
Appendix 2

Acronyms

The following is a description of the acronyms used in this document.

**CCIR 601** – A standard for digital video for picture size of 720 × 485 at 60 interlaced pictures per second or 720 × 576 at 50 interlaced pictures per second.

**CIF** – Common Intermediate Format. Video of picture size 352 × 288 at 30 pictures per second.

**DVD** – Digital Versatile Disc. A standard to store digital audio and/or video on a CD-sized disc.

**HDTV** – High-Definition Television. A standard for television of picture size 1920 × 1080 at 30 pictures per second.

**HTTP** – Hyper Text Transfer Protocol

**ISO** – International Standards Organization. A worldwide federation of national standards bodies from some 140 countries. Homepage at: www.iso.ch


**ITU** – International Telecommunications Union. An international organization within the United Nations System where governments and the private sector coordinate global telecom networks and services. Homepage at: www.itu.int

**JPEG** – Joint Photographic Experts Group. The committee responsible for developing the JPEG and JPEG 2000 standards. Homepage at: www.jpeg.org

**MPEG** – Motion Picture Experts Group. The committee responsible for developing the MPEG standards. Homepage at: www.mpeg.telecomitalialab.com

**NTSC** – National Television Standards Committee. This is the standard for the analog television format used in the US with 525 lines at near 60 pictures per second, i.e. *Interlaced* video.

**PAL** – Phase Alternating Line. This is the standard for the analog television format used in Europe with 625 lines at 50 half-pictures per second, i.e. *Interlaced* video.
PHP – PHP is a language for creating interactive web sites and is used on over 3.3 million web sites around the world. It was originally called “Personal Home Page Tools” when it was created in 1994 by Rasmus Lerdorf to keep track of who was looking at his online resume.

QCIF – Quarter CIF. Video of picture size 176 × 144 at 30 pictures per second.

RGB – video color system that separates the signal into three components: Red, Green and Blue.

SSL – Secure Sockets Layer, digital data encrypting technology, the industry-standard method for protecting web communications developed by Netscape Communications Corporation. The SSL security protocol provides data encryption, server authentication, message integrity, and optional client authentication for a TCP/IP connection.

S/WAN – The Secure Wide Area Network, intends to establish security ground rules based on the Internet Engineering Task Force’s (IETF) proposed Ipsec standard.

VPN – a Virtual Private Network, one way to provide secure access to databases over the Internet.

YUV – an approach to component video that separates the signal into three channels: Y – Luminance, U – Hue and V – Saturation.