Professor Amar Panchal teaches:

Semester 1
Computer Programming 1
(Subject for All branches)

Semester 2
Computer Programming 2
(Subject for All branches)

Semester 3
Data Structures
(Computer Branch)

Semester 4
Analysis of Algorithms and Data Structures
(Computer and IT branches)

Semester 5
Computer Networks
(Computer and IT branches)

Semester 7
Mobile Computing
(Computer and IT branches)

Semester 8
Multimedia systems
(Computer and IT branches)

Distributed Computing
(Computer and IT branches)

MAY 2005

Q.2

A) Discuss group 3 1D and 2D and then compare 3D and 4D (given in class).

B) Give 2 types of redundancy that is temporal (remove with motion compensation) and spatial (removed with baseline sequential codling's four step n give DCT formula). And for mathematical treatment notes2.

Q.3

A) Discuss RIFF with all chunks (list, sub chunk etc) and just give pseudo code for others. Example

```
RIFF 'WAVE'
{
        Subchunk 1 'fmt' < describes recording characteristics of the wave form>
        Subchunk2 'data' <Waveform data>
}

RIFF 'MIDI'
{
        Subchunk 'data' <MIDI data>
}
```

B) MIDI explain

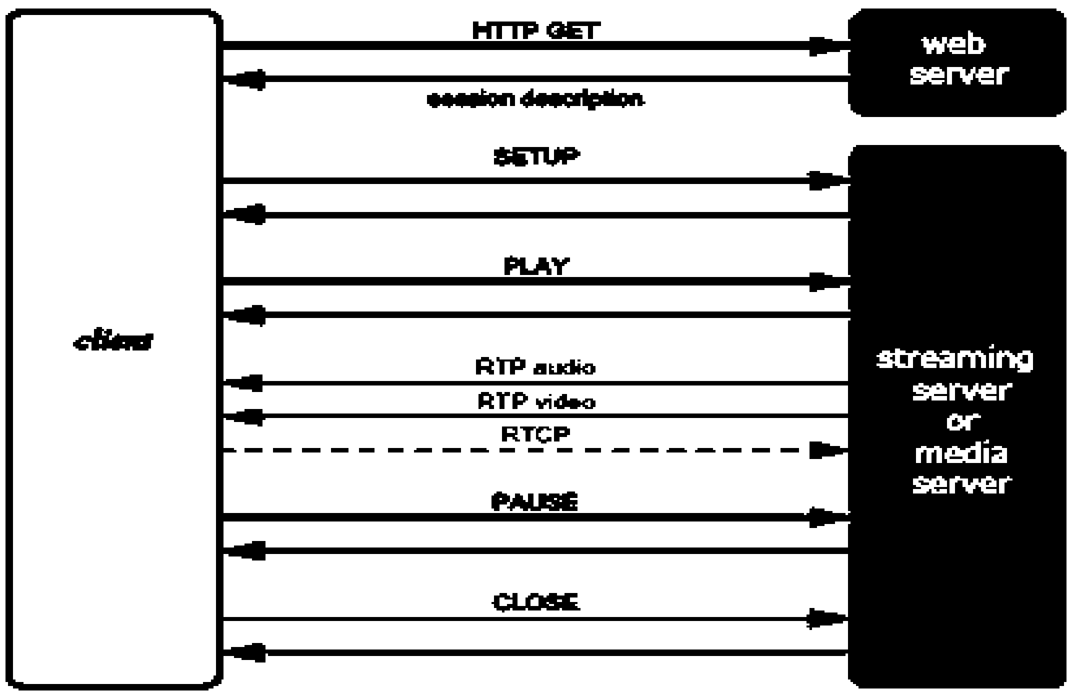
Q.5

B)explain video frame grabber architecture and full motion video architecture from notes.

Q.6

A) DONE IN CLASS

B) Explain streaming and RTP,RTCP.RSVP.
Communication between client and server.



Q.7

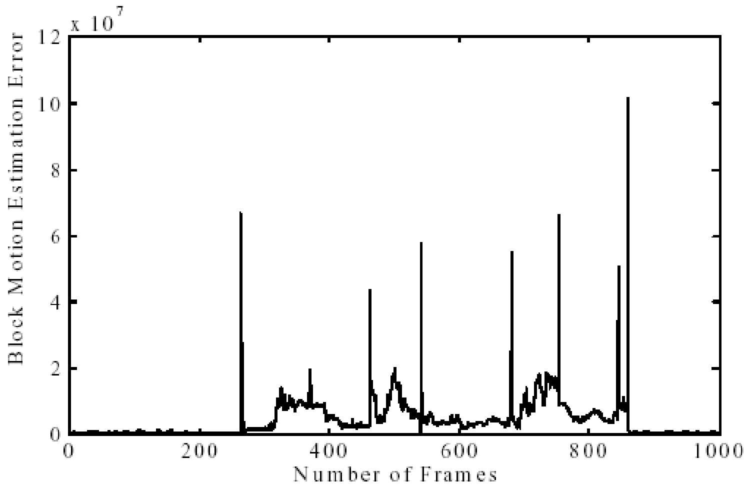
A) INDEXING AND RETRIEVAL IN VIDEO DATABASES:

The increasing development of advanced multimedia applications requires new technologies for the organization and content based retrieval of digital video databases. This objective is accomplished by extracting a collection of a small number of frames or scenes that provide sufficient information about the video sequence. This can be very useful for multimedia interactive services, e.g., browsing of digital video databases on web pages, or automatic production of video trailers. Moreover, this objective is related to searching of frames or scenes, based on certain features such as motion, luminosity, color, shape and texture, with video indexing as a potential application.

In order to achieve extraction of the most characteristic frames or scenes of a video sequence, scene cuts are first detected. A multidimensional feature vector is then generated for each frame. The representation of each frame by a feature vector, apart from reducing storage requirements, transforms the image domain to another domain, more efficient for image queries and retrieval. Based on feature vectors of all frames within a scene, a scene feature vector is computed which characterizes the respective scene. The scene vector is then fed as input to a classifier mechanism which extracts the most characteristic scenes. Finally, the most characteristic frames of a given scene are extracted, based on fluctuation of frame vectors versus time.

2. SCENE DETECTION:

The first stage of the proposed method includes the separation of a video stream into scenes, thus a scene cut detection technique is required. This can be achieved by computing the sum of the block motion estimation error over each frame and detecting frames for which this sum exceeds a certain threshold. In MPEG coded video streams scene cuts can be also detected by measuring the bit allocation in B and P frames.



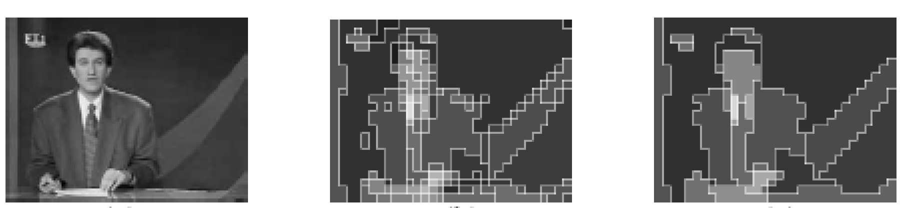
A typical graph of the block motion estimation error versus time is shown in Figure 1. The corresponding video sequence is taken from a TV news program of total duration 40 seconds (1000 frames at 25 frames/second). It consists of a main studio scene, 6 scenes of field reporting, and another main studio scene. The scene cut points, as well as the higher motion of the field reporting scenes compared to that of the main studio scenes, are obvious from the graph.

3. FEATURE EXTRACTION:

The most important task involved in the process of characterizing video scenes is the extraction of proper frame features such as motion, luminosity, color, shape and texture [3]. For this purpose, a feature vector for each frame of the video sequence is calculated. The feature vector is based on color and motion segmentation of each frame.

3.1 Color segmentation:

The color segmentation algorithm splits each frame into regions according to spatial homogeneity criteria. Having formed the segments for each frame, many features are extracted for query or retrieval, such as the number of segments, as well as their location, size, and color. Images consisting of uniform areas will be characterized by large segments, while images containing many small objects or noise, by small and distributed segments.

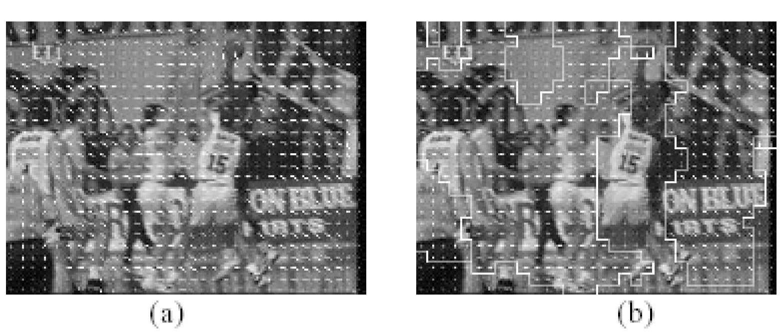


(a) (b) (c)

Figure 2. (a) Original frame from TV news program (main studio). (b) First stage of color segmentation. (c) Final stage.

3.2 Motion Segmentation:

A similar approach is carried out for the case of motion segmentation, which is an important task, since it gives the ability of indexing, query or categorization of frames according to motion characteristics. In this case, the number, size, and location of the motion segments, as well as the direction and magnitude of the corresponding motion vectors, are derived as motion features.



(a) (b)

Figure 3. (a) Motion vectors after processing, (b) Final stage of motion segmentation.

3.3 Feature Vector Formulation:
All of the above features are gathered in order to form a multidimensional feature vector which is used for collection of information content for each frame. Properties of color or motion segments cannot be used directly as elements of the feature vector, since its size will differ between frames. To overcome this problem, we classify color as well as motion segments into pre-determined categories, forming a multidimensional "histogram". To eliminate the possibility of classifying two similar segments at different categories, causing erroneous comparisons, a degree of membership is allocated to each category, resulting in a fuzzy classification . The feature vector is then constructed by calculating the sum, over all segments, of the corresponding degrees of membership, and gathering the above sums into the appropriate categories.

4. SCENE / FRAME SELECTION:
The graph of the feature vector for all frames within a scene indicates the way in which the frame properties fluctuate during a scene period. Consequently, a vector which characterizes the whole scene is constructed by calculating the mean value of feature vectors over the whole duration of a scene.

Scene Selection: In applications where extraction of the most characteristic scenes is required, the scene vector is used as input to a neural classifier. The classifier decides whether a particular vector, i.e., the corresponding scene, belongs to the set of the most characteristic scenes or not. The weights of this classifier can be either pre-determined by experts or adapted interactively according to user preferences.

Frame Selection:Other applications require the extraction of the most characteristic frames within a given scene. In this case, the decision mechanism is based on the detection of those frames whose feature vector resides in extreme locations of the graph. For this purpose, the magnitude of the second derivative of the feature vector was used as a curvature measure. Local maxima of this curve were then extracted as the locations (frame numbers) of the characteristic frames.

Video Queries: Once the feature vector is formed as a function of time, a video database can also be searched in order to retrieve frames that possess particular properties, such as dark frames, frames with a lot of motion and so on. The feature vector space is ideal for such comparisons, as it contains all essential frame properties, while its dimension is much less than that of the image space.

B) Explain what hypermedia message is, its components, linking and embedding, creating sequence (planning, gathering objects structuring them) and any one from VIM, MAPI, X-500(in brief).

C) DONE IN CLASS

DEC 2005

Q.1

A) MULTIMEDIA OBJECT SERVER.

1) Object server architecture:-

- Multimedia application.
- Common object management API.
- ORB (Object request broker): - playback, formats.
- ONS (object name server): - Directory service.
- Object Directory Manager: - update DIR.
- Object Server:- store object.(Logical)
- Object Manager: - Retrieval, Replication, Migration, Transaction and Logical management, Versioning.
- N/W Manager:-Connection n communication over n/w.
- Object Data Store:-Actual store.

B)Middleware in Distributed Workgroup Computing:

The primary role of middleware is to link back-end database servers to front-end clients in a highly flexible and loosely connected network model. A loosely connected network model implies that servers may go off-line and be unavailable without bringing the network down or significantly impacting overall operation. Similarly, clients may go off-line temporarily and continue local operations; when they connect later, they can operate as a part of the network and resynchronize their databases. Middleware provides the glue for dynamically redirecting client requests to appropriate servers that are on-line, thereby also providing a potential load-balancing function under demanding conditions. Middleware is the primary catalyst for creating a distributed network of servers and clients.

Middleware performs a number of functions in this environment:-

1) Provide the user with a local index, an object directory, for objects with which a

client is concerned.
2) Provide automatic object directory services for locating available copies of objects.
3) Provide protocol and data format conversions between the client requests and the stored formats in the server.
2) Provide unique identification throughout the enterprise-wide network for every object through time.

Note that the database architecture changes significantly when middleware is introduced in the system. The Middleware is capable of accessing multiple databases and combining information for presentation of the user.

For example, middleware can perform some or all combinations of the following functions:

- Access a document database to locate a pointer to the required multimedia object.
- Locate an object using a distributed object directory database.
- Access an object database to retrieve an object.
- Retrieve object preprocessing information from an object description database.
- Combine all of this information and preprocess the object before passing it on to a client.

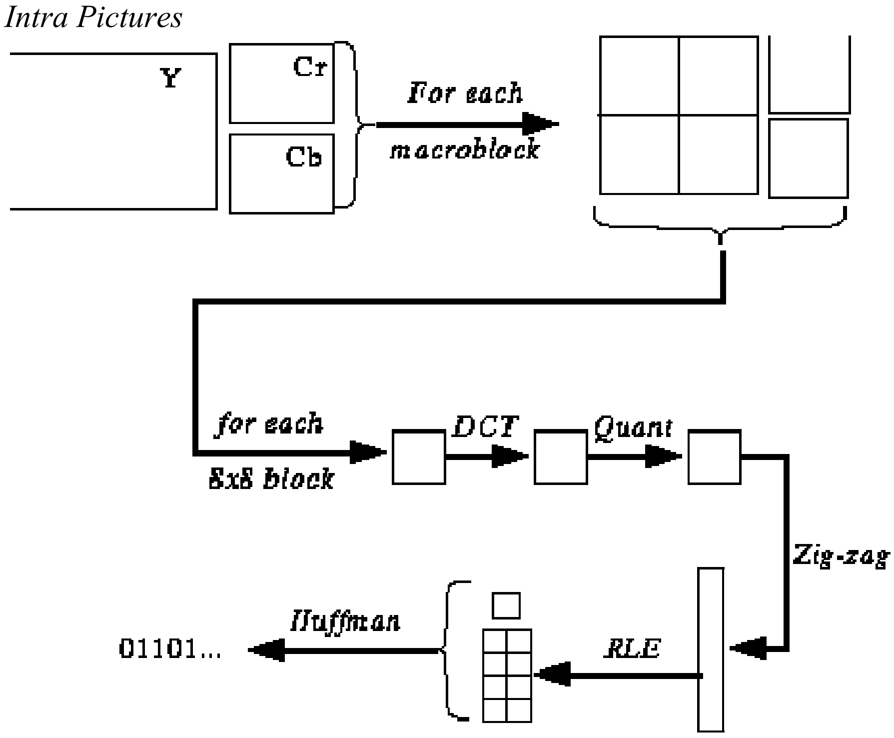
We call these actions of middleware *content-based processing*. The range and nature of such content-based processing can be changed without affecting either the servers or the clients. Content-based processing allows the middleware to address temporal characteristics of certain multimedia objects such as audio and video. Content-based processing also allows a variety of editing

Q.2

A) DONE IN CLASS.

B) MATHEMATICAL TREATMENT AND BLOCKS

Intra Pictures

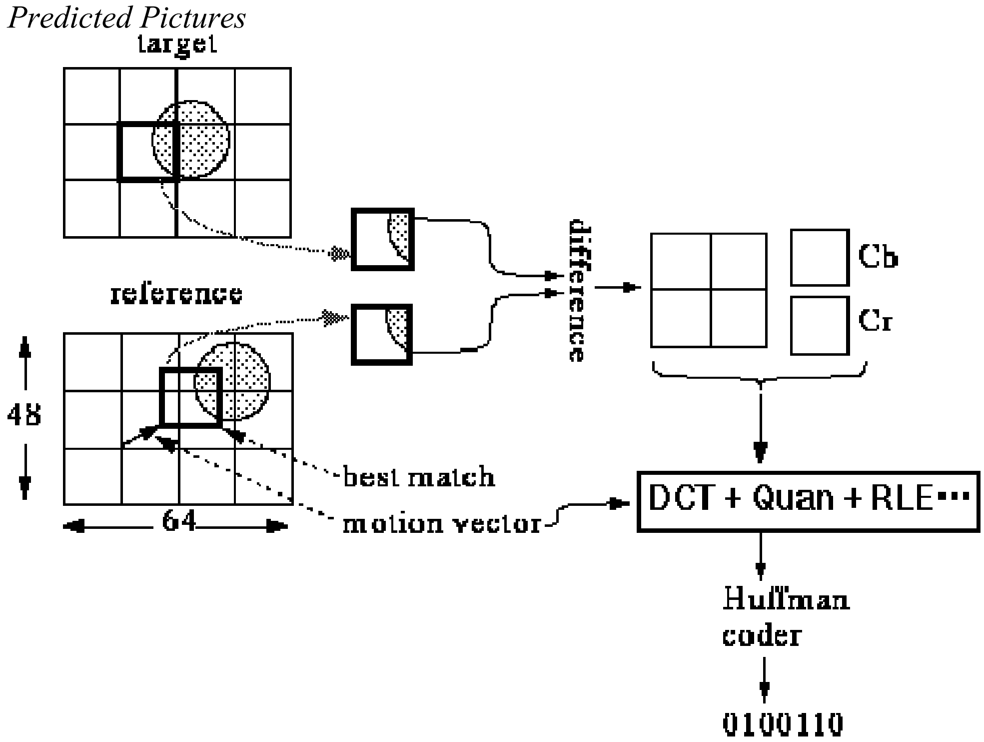


The MPEG transform coding algorithm includes the following steps:

1. Discete cosine transform (DCT)
2. Quantization
3. Run-length encoding

Both image blocks and prediction-error blocks have high spatial redundancy. To reduce this redundancy, the MPEG algorithm transforms 8x8 blocks of pixels or 8x8 blocks of error terms from the spatial domain to the frequency domain with the discrete Cosine Transform (DCT).
The combination of DCT and quantization results in many of the frequency coefficients being zero, especially the coefficients for high spatial frequencies. To take maximum advantage of this, the coefficients are organized in a zigzag order to produce long runs of zero. The coefficients are then converted to a series of run-amplitude pairs, each pair indicating a number of zero coefficeints and the amplitude of a non-zero coefficient. These run-amplitude pairs are then coded with a variable-length code(Huffman Encoding), which uses shorter codes for commonly occurring pairs and longer codes for less common pairs.
Some blocks of pixels need to be coded more accurately than others. for example, blocks with smooth intensity gradients need accurate coding to avoid visbile block boundaries. To deal with this inequality between blocks, the MPEG algorithm allows the amount of quantization to be modified for each macroblock of pixels. This mechanism can also be used to provide smooth adaptation to particular bit rate.

Predicted Pictures

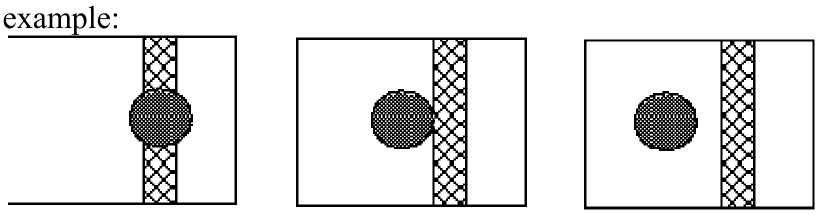


A P-picture is coded with reference to a previous image(reference image) which is an I or P Pictures. From the above figure, the highlighted block in target image(the image to be coded) is simalar to the reference image except it shifted to upper right. Since most of changes between target and reference image can be approximated as translation of small image regions. Therefore a key technique call motion compensation prediction is used.
Motion compensation based prediction exploits the temporal redundancy. Due to frames are closely related, it is possible to accurately represent or "predict" the data of one frame based on the data of a reference image, provided the translation is estimated. The process

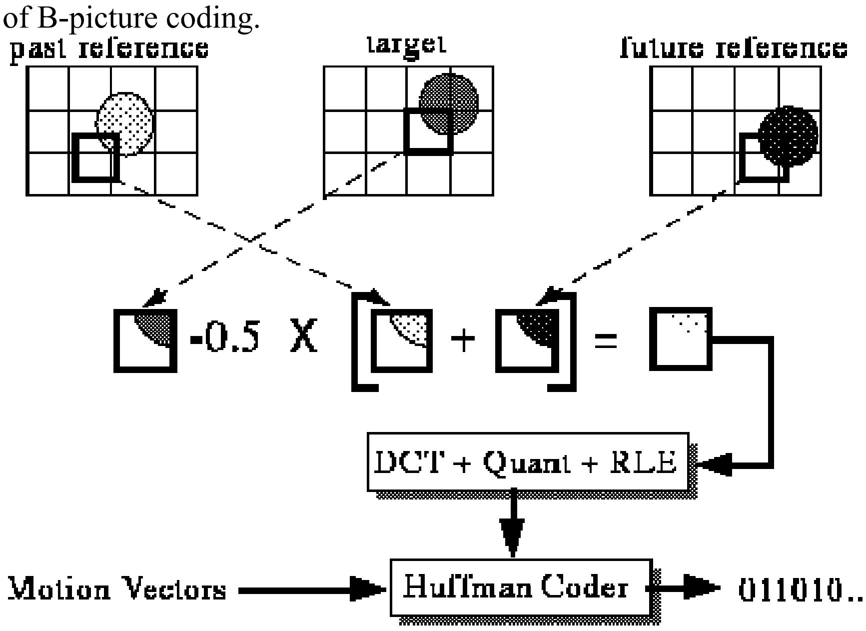
of prediction helps in the reduction of bits by a huge amont. In P-Pictures, each 16x16 sized macroblock is predicted from a macroblock of a previously encoded I picture. Sinces, frames are snapshots in time of a moving object, the macroblocks in the two frames may not be cosited, i.e. correspond to the same spatial location. Hence, a search is conducted in the I frame to find the macroblock which closely matches the macroblock under consideration in the P frame. The difference between the two macroblock is the prediction error. This error can be coded in the DCTdomain. The DCTof the errr results in few high frequency coefficients, which after the quantization process require a small number of bits for represenation. The quantization matrices for the prediction error blocks are different from those used in intra block, due to the distinct nature of their frequency spectrum. The displacements in the horizaontal and vertical directions of the best match macroblock from the cosited macroblock are called motion vectors. Differential coding is used because it reduces the total bit requirement by transmitting the difference between the motion vectors of consecutinve frames. Finally it use the run-length encoding and huffman encoding to encode the data.

Biderectional Pictures

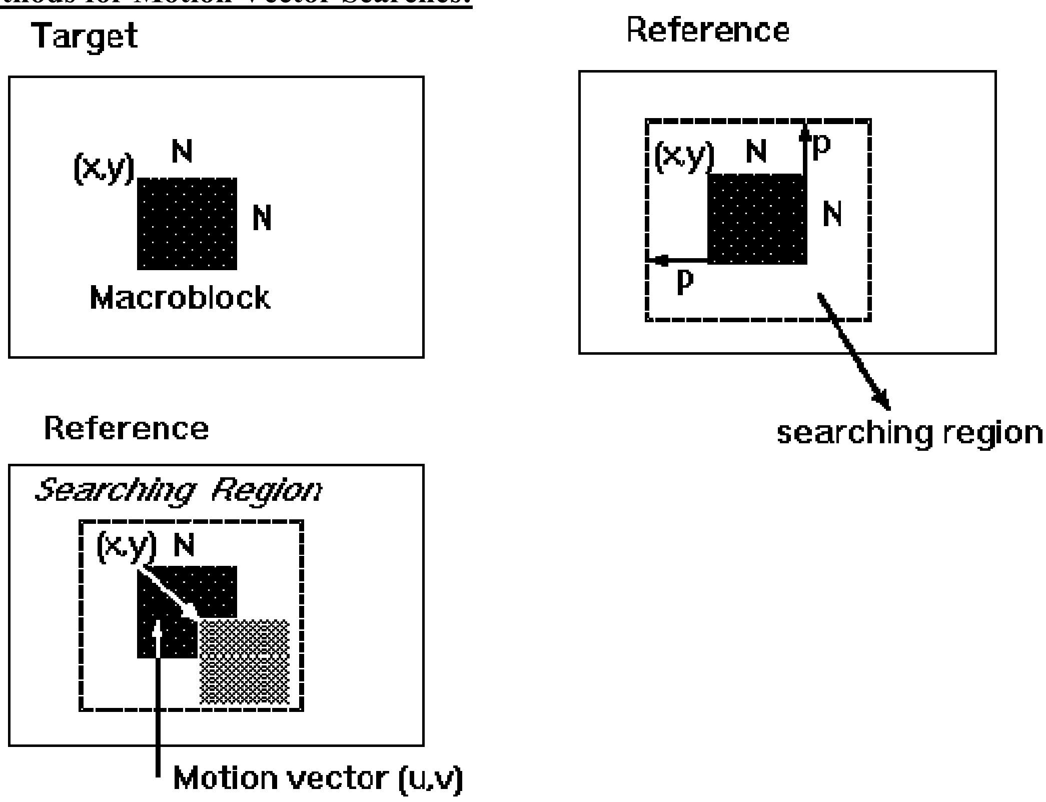
example:



From the above pictures, there are some information which is not in the reference frame. Hence B picture is coded like P-pictures except the motion vectors can reference either the previous reference picture, the next picture, or both. The following is the machanism of B-picture coding.



Methods for Motion Vector Searches:



C(x + k, y + l) -- pixels in the macro block with upper left corner (x, y) in the Target frame.
R(x + i + k, y + j + l) -- pixels in the macro block with upper left corner (x + i, y + j) in the Reference frame.
Cost function is:

$$MAE(i,j) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} |C(x+k, y+l) - R(x+i+k, y+j+l)|$$

Where *MAE* stands for *Mean Absolute Error*.
Goal is to find a vector (u, v) such that *MAE*(u, v) is minimum.

Q.3

A) DEVICE-INDEPENDENT BITMAPS

A device-independent bitmap (DIB) contains a *color table*. A color table describes how pixel values correspond to *RGB* color values, which describe colors that are produced by emitting light. Thus, a DIB can achieve the proper color scheme on any device. A DIB contains the following color and dimension information:

- The color format of the device on which the rectangular image was created.
- The resolution of the device on which the rectangular image was created.
- The palette for the device on which the image was created.
- An array of bits that maps red, green, blue (RGB) triplets to pixels in the rectangular image.
- A data-compression identifier that indicates the data compression scheme (if any) used to reduce the size of the array of bits.

The color and dimension information is stored in a BITMAPINFO structure, which consists of a BITMAPINFOHEADER structure followed by two or more RGBQUAD structures. The BITMAPINFOHEADER structure specifies the dimensions of the pixel rectangle, describes the device's color technology, and identifies the compression schemes used to reduce the size of the bitmap. The RGBQUAD structures identify the colors that appear in the pixel rectangle.

There are two varieties of DIBs:

- A bottom-up DIB, in which the origin lies at the lower-left corner.
- A top-down DIB, in which the origin lies at the upper-left corner.

If the height of a DIB, as indicated by the Height member of the bitmap information header structure, is a positive value, it is a bottom-up DIB; if the height is a negative value, it is a top-down DIB. Top-down DIBs cannot be compressed.

The color format is specified in terms of a count of color planes and color bits. The count of color planes is always 1; the count of color bits is 1 for monochrome bitmaps, 4 for VGA bitmaps, and 8, 16, 24, or 32 for bitmaps on other color devices. An application retrieves the number of color bits that a particular display (or printer) uses by calling the GetDeviceCaps function, specifying BITSPIXEL as the second argument.

The resolution of a display device is specified in pixels-per-meter. An application can retrieve the horizontal resolution for a video display, or printer, by following this three-step process.

1. Call the GetDeviceCaps function, specifying HORZRES as the second argument.
2. Call GetDeviceCaps a second time, specifying HORZSIZE as the second argument.
3. Divide the first return value by the second return value.

The application can retrieve the vertical resolution by using the same three-step process with different parameters: VERTRES in place of HORZRES, and VERTSIZE in place of HORZSIZE.

The palette is represented by an array of RGBQUAD structures that specify the red, green, and blue intensity components for each color in a display device's color palette. Each color index in the palette array maps to a specific pixel in the rectangular region associated with the bitmap. The size of this array, in bits, is equivalent to the width of the rectangle, in pixels, multiplied by the height of the rectangle, in pixels, multiplied by the count of color bits for the device. An application can retrieve the size of the device's palette by calling the GetDeviceCaps function, specifying NUMCOLORS as the second argument.

This file is taken by jpeg and compressed using dct based codding.

B) MIDI EXPLAIN

Q.6

B) Explain what hypermedia message is, its components, linking and embedding, creating sequence (all things in brief) and any one from VIM, MAPI, X-500.

Q.7

A) Give video frame grabbing architecture, types of animation difference between animation and video (given in class), last video capture board architecture.

B) Scanners with types and working and features to select scanner.

C) Done In Class.

May 2006

Q.1

A) Done In Class
B) Done In Class
C) Explain Flatbed And Its Working

Q.2

A) Entire Worm Drives
B)Done In Class

Q.3

A) Done Tourist System Only Databases.
B) Done

Q.4

A)Workflow is a term used to describe the tasks, procedural steps, organizations or people involved, required input and output information, and tools needed for each step in a business process. A workflow approach to analyzing and managing a business process can be combined with an object-oriented programming approach, which tends to focus on documents and data. In general, workflow management focuses on processes rather than documents. A number of companies make workflow automation products that allow a company to create a workflow model and components such as online forms and then to use this product as a way to manage and enforce the consistent handling of work. For example, an insurance company could use a workflow automation application to ensure that a claim was handled consistently from initial call to final settlement. The workflow application would ensure that each person handling the claim used the correct online form and successfully completed their step before allowing the process to proceed to the next person and procedural step.

Explain document imaging workflow.

B) Types on user and then types of authoring system.

Q.5

A) Ch.6

B) Done

C) Chapter 10->10.5

D) Aural User Interface:The common approach for speech-recognition-based user interfaces has been to graft the speech recognition interface into existing graphical user interfaces. This is a mix of conceptually mismatched media that makes the interface cumbersome and not very efficient. The lack of visual response makes the user uncomfortable. A true aural user interface (AUI) would allow computer system to accept speech as direct input and provide an oral response to the user actions. Speech enabling is a challenging activity for user interface designers. The real challenge in designing AUI systems is to create an aural desktop that substitutes voice and ear for keyboard and display, and be able to mix and match them. Aural cues should be able to represent icons, menus and windows of a GUI. Besides computer technology AUI designs involve human perception, cognitive science, and psycho-acoutic theory. The human response to visual and aural inputs is markedly different. Rather than duplicate the GUI functions in an AUI, this requires new thinking to adapt the AUI to the human response to aural feedback.

Q.6

A) Only explain type of messages in midi(channel messages & system messages)

B) Virtual Reality Design Considerations:
The user feedback we referred to in the discussion of modeling includes gestures. Gestures include movement of limbs (causing spatial coordinates of the participants to change relative to the environment), movement of eyes and the head (indicating change in direction of view), movement of lips (indicating user speech), and any other perceptible motion or expression the virtual reality system is capable of monitoring.

Gesture Recognition: Key design issues in gesture recognition are to determine the following:

- Start and end the gesture
- Path recognition and velocity of movement
- Combination effects and multiple related gestures
- Environmental context in which the gesture was performed

Gesture recognition is a complex programming task, especially when it involves managing multiple gesture objects such as eye, hand, arm, head and leg movements. While each individual gesture source can be managed by a dedicated gesture object for that source, the really complex gesture management arises when relationships among these gestures becomes important and has to be evaluated in real time.

Accuracy in Rendering:

In most user-computer interaction, accuracy in rendering is important but not crucial because an user can deal with an error if there is time to think and respond. In virtual reality, accuracy of rendering is much more important because the user is involved with the computer-created environment as a participant who is taken by surprise and has no time to react. Rendering errors can suddenly change the frame of reference for the participant, thereby making the interaction very confusing.

Aggregation of inputs and Rendering:

We touched on this issue briefly under the heading on gesture recognition. There are essentially two types of design issues in combining multiple gesture inputs: gesture inputs that are related and can be anticipated, and unsolicited gesture inputs that are unrelated. The former are adjust in the program logic as potential combinations (the real issue to adjust here is the sequence in which the gestures arrive at the computer) that are acceptable and has predefined responses as changes to the environment as well as other actions. The latter is a case where the participant responded in an unanticipated manner as if there was disturbing, unexpected, or incorrect feedback to some previous gesture, or the participant has lost the frame of reference and is confused. If these gestures are conflicting in terms of the current environment the handling of this by the program logic can become very complex.

Performance:

Sensory perception in the human body is very fast. When we touch a hot kettle, we instantly experience the burning sensation that causes us to rapidly remove our hand from the kettle. Similarly, touching a thorn causes instant sensation of pain. When that happens, the head automatically turns in the direction of the source of the pain and the body gets to move out of the way of danger. We react equally fast to these things we see. For example, a stone lobbed at us causes the eyes to close as we dogged the stone. Imagine what happens when the stone starts coming at us and we prepare to dogged, but when we open our eyes we see that stone has slowed down considerably and is still coming to us. Yes, this causes us to get confused and start wondering if it is a real stone or being manipulated by some means.

Virtual reality tries to stimulate real sensory perceptions. When the stimulation does not happen in the manner to which we are accustomed and the speed of the stimulation changes unexpectedly, the virtual reality experience loses its real value –that of giving the participant the feeling that the experience is real and not a computer-generated fantasy. System, network and rendering performance become much more important in the case of virtual reality than in any other application area.

Hardware Resource Demands:

Hardware resources required for any application depend on the nature of application. Virtual reality requires virtual response, a wide range of sensory inputs, high-speed rendering of multimedia objects, and a sustained performance over time. Virtual reality applications placed very intensive performance demands on hardware. While failure to perform on a sustained basis is a mere irritant in other cases, in virtual reality the impact can be devastating, making the application all but useless. A careful analysis of performance requirements is essential to ensure a responsive system.

Q.7

A) Take Example Of Work Station Architecture Chapte1 And Explain Detail Of Each Block.

B) Done In Class.

Dec 2006

Q.1

A) Chapter 10's 10.1 components of distributed multimedia systems.
Multimedia Services: Unique bundling of services: audio conferencing, telepresence and desktop video conferencing, web collaboration (eLearning) and webcasting services.
Application Performance Services: Maximizing the potential of your business applications by analyzing, modeling, simulating, optimizing and benchmarking your applications.
Web Services: Integrating communications into your applications and websites. Building on Nortel's web services platform, we enable rich web experiences

B) **RIFF DIBS (Device-Independent Bitmaps):**
Device-independent bitmap (DIB) is a Microsoft Windows standard format that defines bitmaps and color attributes for bitmaps independent of devices. This allows DIBS to be

displayed and printed on different devices; hence the name device-independent. DBIS are normally embedded in .BMP files (pain brush files). DBIS can also be embedded in Windows .WMF metafiles and clipboard or .CLP files. A DIB structure contains a bitmap information header called BITMAPINFOHEADER, a color table structure called RGBQUAD, and an array of bytes for the pixel bitmap. The structure is shown below:

BITMAPINFOHEADER	**RGBQUAD**	**PIXELS**

A DIB file contains, in addition to a DIB structure, a bitmap file header structure called BITMAPFILEHEADER specified before the DIB structure. The following shows the DIB file format (also known as .BMP format).

BITMAPFILEHEADER	**BITMAPINFO = BITMAPINFOHEADER + RGBQUAD**	**PIXELS**

Where, the BITMAPFILEHEADER structure contains the type and size of the bitmap file.The BITMAPINFOHEADER structure specifies the dimensions, compression type, and color format for the bitmap. The DIB file format contains an array of RGBQUAD structures. The number of elements in the array represents the Total number of colors in the bitmap.

- **<u>RIFF Audio Video Interleaved (AVI) File Format:</u>**

The AVI file format is designed from a movie metaphor: it contains alternate frames of audio-video objects to form an interleaved file format, as shown in figure:

Header	**First Frame Audio Video**	**Second Frame Audio Video**

AVI files can be enveloped within the RIFF format to create the RIFF AVI file format. A RIFF AVI file contains a RIFF chunk with the Form Type "AVI" and two mandatory list chunks, "hdrl" and "movi." The list chunk "hdrl" defines the format of the data, and "movi" contains the data for the audio-video streams. The third list chunk called "idx1," is optional index chunk. Both the "hdrl" and "movi" list chunks can contain optional sub chunks. The list chunk "hdrl" contains the main AVI header and sub chunks stream header "strh," stream format chunk "strf," and an optional additional data chunk "strd."

1) **<u>The "movi" Chunk:</u>**

The LIST chunk "movi" contains the actual data for audio-video streams in a sub chunk called the data chunk. The movie chunk can also contain one or more sub chunks called "rec" chunk groups the data sub chunks to form a group.

2) **<u>Index "idx1" Chunk:</u>**

An Index chunk is an optional chunk for AVI files and is placed after the LIST "movi" chunk. Each movi chunk may contain one index chunk. Index chunks are used to contain indexing information for entries of data chunks and the actual locations of these data chunks in the file.

<u>Q.2</u>

A)

<u>H/W:</u>

1> CPU speed

2> Storage and access speed of media

3> Cache(types)
3> Co-processor

S/W:
1> Compression and decompression.
2> Type of OS
3> GUI

N/W:
1> Speed
2> Type of connectivity
3> Devices used

B) Done

Q.3

A) Explain flatbed scanner from notes and then working of CCD and also variation of it.(2light source,3light source)

B) Talk about image processing from application of multimedia system and also about histrogram sliding, histrogram streaching, threshoulding from notes and i**mage processing** is any form of information processing for which the input is an image, such as photographs or frames of video; the output is not necessarily an image, but can be for instance a set of features of the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. Geometric transformations such as enlargement, reduction, and rotation **operations done in classes.**

Q.4

A) Latency is a time delay between the moment something is initiated, and the moment one of its effects begins or becomes detectable. The word derives from the fact that during the period of latency the effects of an action are latent, meaning "potential" or "not yet observed". Even within an engineering context, latency has several meanings depending on the engineering area concerned (i.e. communication, operational, simulation, mechanical, or biomedical fiber stimulation latencies).Computers run on a set of instructions called an executable. On operating systems, the execution of the executable can be postponed if other executables (a.k.a. processes) are also executing. In addition, the operating system can schedule when to perform the action that the executable is commanding. For example, suppose a process commands that a computer card's voltage output be set high-low-high-low..etc at a rate of 1000 Hz. The operating system may choose to adjust the scheduling of each transition (high-low or low-high) based on an internal clock. The latency is the delay between the executable instruction commanding the transition and the hardware actual transitioning the voltage from high to low or low to high.On Microsoft Windows, it appears that the timing of commands to hardware is not exact. Empirical data suggests that Windows (using the Windows sleep timer which accepts millisecond sleep times) will schedule on a 1024 Hz clock and will delay 24 of 1024 transitions per second to make an average of 1000 Hz for the update rate.[citation needed] This can have serious ramifications for discrete-time algorithms that rely on fairly consistent timing between updates such as those found in control theory.Linux may have the same problems with scheduling of hardware I/O. The problem in Linux is mitigated by the fact that the operating system kernel's process scheduler can be replaced by a real-time scheduler.On embedded systems, the real-time execution of instructions is expected from the low-level embedded operating system.Memory latency
In computing, memory latency is the time between initiating a request for a byte or word in memory until it is retrieved. If the data is not in the processor's cache, it takes longer to obtain them, as the processor will have to communicate with the external memory cells.

Latency is therefore a fundamental measure of the speed of memory: the less the latency, the faster the reading operation.However, memory latency should not be confused with memory bandwidth, which measures the throughput of memory. It is possible that an advance in memory technology increases bandwidth (an apparent increase in performance), and yet latency increases (an apparent decrease in performance). For example, DDR memory has been superseded by DDR2, and yet DDR2 has significantly greater latency when both DDR and DDR2 have the same clock frequency. DDR2 can be clocked faster, however, increasing its bandwidth; only when its clock is significantly greater than that of DDR will DDR2 have lower latency than DDR.

Memory latency is comprised of the following factors:

- Address decoding delay
- Word line delay
- Bit line sensing delay
- Output driving delay
- Wire RC Delay (dominant)

If the effects of chip processing improvements are considered, it has been found that DRAM latency is proportional (in some respects) to chip size, however there are many circuit, architecture, and layout techniques may be used to reduce latency. Higher speed circuits take up more area, but in general, no dominant circuit topologies have emerged as the preferred design.

Contributors to Latency
The worst case latency in DRAM is specified by the row access time (^{t}RAC). This is the time from when the RAS pin is pulled low to when valid data appears on the output. It is comprised of four main components:

1. Address decoding latency - The time required to latch the address and decide which word line needs to be activated. This delay is caused by the multiplexing of the row and column addresses and the propagation delay through the decoding logic.
2. Word line activation latency - The time required to raise the word line to a high voltage. This is fundamentally an RC delay.
3. Bit line sensing latency - The time required for the cell contents to be detected by the sense amplifiers. This is affected by bit line architecture, the RC of the sense amp drive line, cell-to-bit line capacitance ratio, and sense amp topology.
4. Output driving latency - Time needed for the data to be propagated from the sense amps to the output pads. This too is an RC delay.

Notes for second part.

B) done in class.

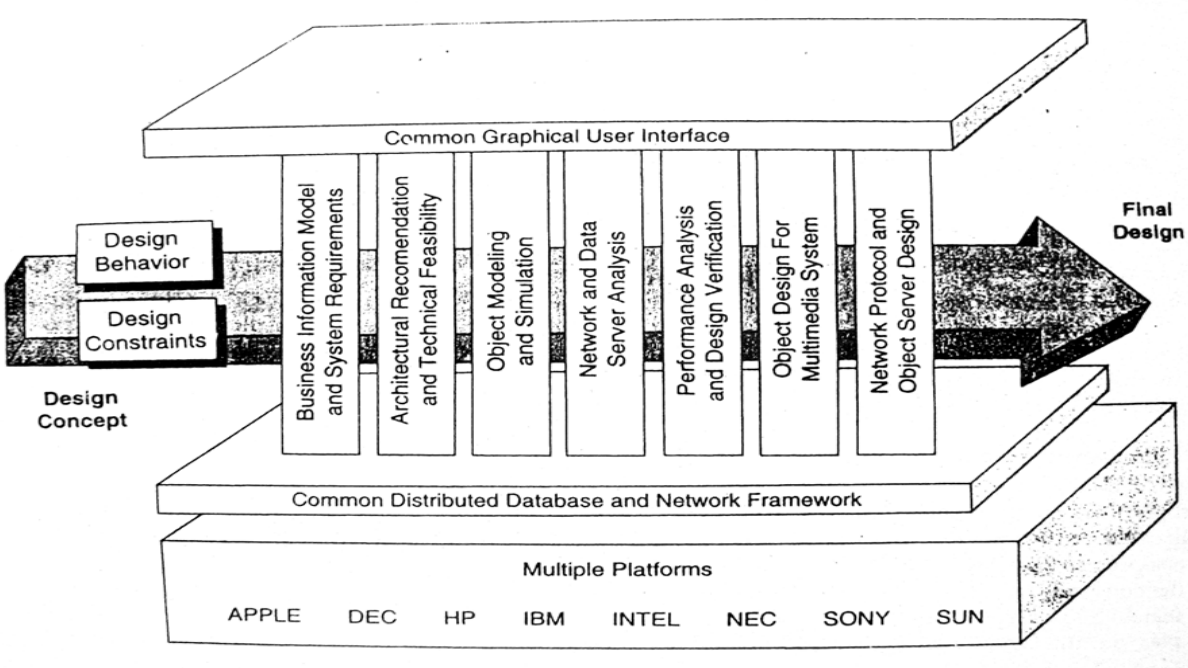
Q.5

A) done.

B) class notes. Also explain document imaging workflow.

Q.6

A) MULTIMEDIA SYSTEM DESIGN STEPS



Business model-describes the company, its market, products, locations and operations. Business information model-describes the information requirements for the corporation. These requirements, in turn, determine the functionality for the applications provided by the corporate information system. The business information model describes the following:

- Who will be using the application
- What operations each type of user will be performing
- The availability requirements
- The performance requirements
- Where data elements are created
- How widely are the data elements distributed
- The data storage requirements
- The user interface for the application

Architectural recommendation and technology feasibility report-describes the architectural approach and the selection of hardware and software technologies that make the solution feasible. The key issues that are addressed in this report include the following:
Distributed hardware: location of database servers, workstations, networks used
User access: access capability limitations and security issues
Database management: backup and archival
Database synchronization
Information system model-describes the structural and transactional components of the information system.
Object model-describes the underlying objects that support the user-visible functions and the externally visible data elements.
System design- Converts the model into a design that can be used by a programmer to code an application. Key aspects of the design effort include the following:
Describing data types and data element sizes for all attributes for every object
Describing the calling sequences for each method, including the attributes used as input parameters, return values from the methods, and error returns from the methods.

Q.6

B)

- DIGITAL AUDIO SOFTWARES:
1. *Quicktime 7:*
2. Realnetworks:

- IMAGE EDITING:
1. Adobe photoshop:
2. Coreldraw:

- VIDEO EDITING:
1. Adobe premiere elements 3.0:
2. Adobe production studio:

- ANIMATION:
1. Maya (software):
2. 3ds max:

- *FULL-FLEDGED MULTIMEDIA AUTHORING SOFTWARE:-*
1. AUTHORWARE 4
   - By macromedia inc.
   - Offers visual authoring by dragging and dropping icons
   - Permits compression and streaming of sounds, animations and graphics
   - Files run on both windows and macintosh platforms
2. ICONAUTHOR
   - By aimtech corporation
   - Allows you to integrate with a database
   - Runs java applets and activex controls
   - System requirements: 486/66 mhz, 8 mb ram, 30 mb hard disk
3. DIRECTOR 6 MULTIMEDIA STUDIO
   - By macromedia inc.
   - Steeper learning curve than most but more flexible
   - Offers a frame-by-frame, timeline-oriented approach
   - Control animation with scripts
   - System requirements: 486/66 mhz, 8 mb ram, 40 mb hard disk

Q.7

A) Video conference:

(also known as a *videoteleconference*) is a set of interactive telecommunication technologies which allow two or more locations to interact via two-way video and audio transmissions simultaneously. It has also been called visual collaboration and is a type of groupware.

The core technology used in a videoteleconference (VTC) system is digital compression of audio and video streams in real time. The hardware or software that performs compression is called a codec (coder/decoder). Compression rates of up to 1:500 can be achieved. The resulting digital stream of 1's and 0's is subdivided into labelled packets, which are then transmitted through a digital network of some kind (usually ISDN or IP). The use of audio modems in the transmission line allow for the use of POTS, or the Plain Old Telephone System, in some low-speed applications, such as videotelephony, because they convert the digital pulses to/from analog waves in the audio spectrum range.

The other components required for a VTC system include:

- Video input : video camera or webcam
- Video output: computer monitor , television or projector
- Audio input: microphones
- Audio output: usually loudspeakers associated with the display device or telephone
- Data transfer: analog or digital telephone network, LAN or Internet

There are basically two kinds of VTC systems:

1. Dedicated systems have all required components packaged into a single piece of equipment, usually a console with a high quality remote controlled video camera. These cameras can be controlled at a distance to pan left and right, tilt up and down, and zoom. They became known as PTZ cameras. The console contains all electrical interfaces, the control computer, and the software or hardware-based codec. Omnidirectional microphones are connected to the console, as well as a TV monitor with loudspeakers and/or a video projector. There are several types of dedicated VTC devices:
   1. Large group VTC are non-portable, large, more expensive devices used for large rooms and auditoriums.
   2. Small group VTC are non-portable or portable, smaller, ss expensive devices used for small meeting rooms.
   3. Individual VTC are usually portable devices, meant for single users, have fixed cameras, microphones and loudspeakers integrated into the console.
2. Desktop systems are add-ons (hardware boards, usually) to normal PC's, transforming them into VTC devices. A range of different cameras and microphones can be used with the board, which contains the necessary codec and transmission interfaces. Most of the desktops systems work with the H.323 standard. Video conferences carried out via dispersed PCs are also known as e-meetings.

- Multipoint videoconferencing:

  Simultaneous videoconferencing among three or more remote points is possible by means of a Multipoint Control Unit (MCU). This is a bridge that interconnects calls from several sources (in a similar way to the audio conference call). All parties call the MCU unit, or the MCU unit can also call the parties which are going to participate, in sequence. There are MCU bridges for IP and ISDN-based videoconferencing. There are MCUs which are pure software, and others which are a combination of hardware and software. An MCU is characterised according to the number of simultaneous calls it can handle, its ability to conduct transposing of data rates and protocols, and features such as Continuous Presence, in which multiple parties can be seen onscreen at once.

  MCUs can be stand-alone hardware devices, or they can be embedded into dedicated VTC units.

  Some systems are capable of multipoint conferencing with no MCU, stand-alone, embedded or otherwise. These use a standards-based H.323 technique known as "decentralized multipoint", where each station in a multipoint call exchanges video and audio directly with the other stations with no central "manager" or other bottleneck. The advantages of this technique are that the video and audio will generally be of higher quality because they don't have to be relayed through a central point. Also, users can make ad-hoc multipoint calls without any concern for the availability or control of an MCU. This added convenience and quality comes at the expense of some increased network bandwidth, because every station must transmit to every other station directly.

In recent years, IP based videoconferencing has emerged as a common communications interface and standard provided by VTC manufacturers in their traditional ISDN-based systems. Business, government and military organizations still predominantly use H.320 and ISDN VTC. Though, due to the price point and proliferation of the Internet, and broadband in particular, there has been a strong spurt of growth and use of H.323, IP VTC. H.323 has the advantage that it is accessible to anyone with a high speed Internet connection, such as DSL.

In addition, an attractive factor for IP VTC is that it is easier to set-up for use with a live VTC call along with web conferencing for use in data collaboration. These combined technologies enable users to have a much richer multimedia environment for live meetings, collaboration and presentations.

B) Discuss type of authoring system on basis of user and on basis of technologies)

C) Discuss jpeg and mpeg.

D) From chapter 1.

May 2007

Q-1

A)Bitmap file format

Introduction:

The .bmp file format (sometimes also saved as .dib) is the standard for a Windows 3.0 or later DIB(device independent bitmap) file. It may use compression (though I never came across a compressed .bmp-file) and is (by itself) not capable of storing animation. However, you can animate a bitmap using different methods but you have to write the code which performs the animation. There are different ways to compress a .bmp-file, but I won't explain them here because they are so rarely used. The image data itself can either contain pointers to entries in a color table or literal RGB values (this is explained later).

Basic structure:

A .bmp file contains of the following data structures:
BITMAPFILEHEADER bmfh;
BITMAPINFOHEADER bmih;
RGBQUAD aColors[];
BYTE aBitmapBits[];
bmfh contains some information about the bitmap file (about the file, not about the bitmap itself). *bmih* contains information about the bitmap such as size, colors,... The *aColors array* contains a color table. The rest is the image data, which format is specified by the *bmih* structure.

Exact structure:

The following tables give exact information about the data structures and also contain the settings for a bitmap with the following dimensions: size 100x100, 256 colors, no compression. The *start*-value is the position of the byte in the file at which the explained data element of the structure starts, the *size*-value contains the nuber of bytes used by this data element, the *name*-value is the name assigned to this data element by the Microsoft API documentation. *Stdvalue* stands for standard value. There actually is no such a thing as a standard value but this is the value Paint assigns to the data element if using the bitmap dimensions specified above (100x100x256). The *meaning*-column gives a short explanation of the purpose of this data element.

The BITMAPFILEHEADER:

start	size	name	stdvalue	purpose
1	2	bfType	19778	must always be set to 'BM' to declare that this is a .bmp-file.
3	4	bfSize	??	specifies the size of the file in bytes.
7	2	bfReserved1	0	must always be set to zero.
9	2	bfReserved2	0	must always be set to zero.
11	4	bfOffBits	1078	specifies the offset from the beginning of the file to the bitmap data.

The BITMAPINFOHEADER:

start	size	name	stdvalue	purpose
15	4	biSize	40	specifies the size of the BITMAPINFOHEADER structure, in bytes.
19	4	biWidth	100	specifies the width of the image, in pixels.
23	4	biHeight	100	specifies the height of the image, in pixels.
27	2	biPlanes	1	specifies the number of planes of the target device, must be set to zero.
29	2	biBitCount	8	specifies the number of bits per pixel.
31	4	biCompression	0	Specifies the type of compression, usually set to zero (no compression).
35	4	biSizeImage	0	specifies the size of the image data, in bytes. If there is no compression, it is valid to set this member to zero.
39	4	biXPels	0	specifies the the horizontal pixels per meter on the designated targer

		PerMeter		device, usually set to zero.
43	4	biYPelsPerMeter	0	specifies the the vertical pixels per meter on the designated targer device, usually set to zero.
47	4	biClrUsed	0	specifies the number of colors used in the bitmap, if set to zero the number of colors is calculated using the biBitCount member.
51	4	biClrImportant	0	specifies the number of color that are 'important' for the bitmap, if set to zero, all colors are important.

Note that *biBitCount* actually specifies the color resolution of the bitmap. The possible values are: 1 (black/white); 4 (16 colors); 8 (256 colors); 24 (16.7 million colors). The biBitCount data element also decides if there is a color table in the file and how it looks like. In 1-bit mode the color table has to contain 2 entries (usually white and black). If a bit in the image data is clear, it points to the first palette entry. If the bit is set, it points to the second. In 4-bit mode the color table must contain 16 colors. Every byte in the image data represents two pixels. The byte is split into the higher 4 bits and the lower 4 bits and each value of them points to a palette entry. There are also standard colors for 16 colors mode (16 out of Windows 20 reserved colors (without the entries 8, 9, 246, 247)). Note that you do not need to use this standard colors if the bitmap is to be displayed on a screen which support 256 colors or more, however (nearly) every 4-bit image uses this standard colors. In 8-bit mode every byte represents a pixel. The value points to an entry in the color table which contains 256 entries (for details see Palettes in Windows. In 24-bit mode three bytes represent one pixel. The first byte represents the red part, the second the green and the third the blue part. There is no need for a palette because every pixel contains a literal RGB-value, so the palette is omitted.

The RGBQUAD array:

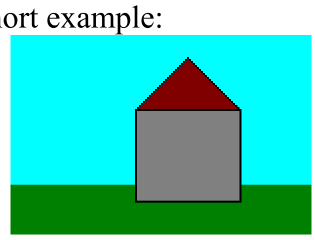
The following table shows a single RGBQUAD structure:

start	size	name	stdvalue	purpose
1	1	rgbBlue	-	specifies the blue part of the color.
2	1	rgbGreen	-	specifies the green part of the color.
3	1	rgbRed	-	specifies the red part of the color.
4	1	rgbReserved	-	must always be set to zero.

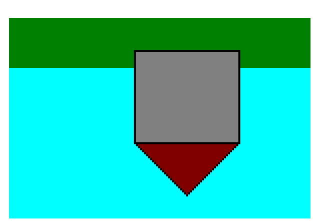
Note that the term *palette* does not refer to a RGBQUAD array, which is called *color table* instead. Also note that, in a color table (RGBQUAD), the specification for a color starts with the blue byte. In a palette a color always starts with the red byte. There is no simple way to map the whole color table into a LOGPALETTE structure, which you will need to display the bitmap. You will have to write a function that copies byte after byte.

The pixel data:

It depens on the BITMAPINFOHEADER structure how the pixel data is to be interpreted It is important to know that the rows of a DIB are stored upside down. That means that the uppest row which appears on the screen actually is the lowest row stored in the bitmap, a short example:



pixels displayed on the screen



pixels stored in .bmp-file

You do not need to turn around the rows manually. The API functions which also display the bitmap will do that for you automatically. Another important thing is that the number of bytes in one row must always be adjusted to fit into the border of a multiple of four. You simply append zero bytes until the number of bytes in a row reaches a multiple of four, an example:

```
 6 bytes that represent a row in the bitmap:        A0 37 F2 8B 31 C4
 must be saved as:                                  A0 37 F2 8B 31 C4 00 00
```

to reach the multiple of four which is the next higher after six (eight). If you keep these few rules in mind while working with .bmp files it should be easy for you, to master it.

WAVE File Format

WAVE File Format is a file format for storing digital audio (waveform) data. It supports a variety of bit resolutions, sample rates, and channels of audio. This format is very popular upon IBM PC (clone) platforms, and is widely used in professional programs that process digital audio waveforms. It takes into account some peculiarities of the Intel CPU such as little endian byte order.

This format uses Microsoft's version of the Electronic Arts Interchange File Format method for storing data in "chunks". You should read the article About Interchange File Format before proceeding.

Data Types

A C-like language will be used to describe the data structures in the file. A few extra data types that are not part of standard C, but which will be used in this document, are:

pstring	Pascal-style string, a one-byte count followed by that many text bytes. The total number of bytes in this data type should be even. A pad byte can be added to the end of the text to accomplish this. This pad byte is not reflected in the count.
ID	A chunk ID (ie, 4 ASCII bytes) as described in About Interchange File Format.

Also note that when you see an array with no size specification (e.g., char ckData[];), this indicates a variable-sized array in our C-like language. This differs from standard C arrays.

Constants

Decimal values are referred to as a string of digits, for example 123, 0, 100 are all decimal numbers. Hexadecimal values are preceded by a 0x - e.g., 0x0A, 0x1, 0x64.

Data Organization

All data is stored in 8-bit bytes, arranged in Intel 80x86 (ie, little endian) format. The bytes of multiple-byte values are stored with the low-order (ie, least significant) bytes first. Data bits are as follows (ie, shown with bit numbers on top, "lsb" stands for "least significant byte" and "msb" stands for "most significant byte"):

```
         7  6  5  4  3  2  1  0
       +-----------------------+
 char: | lsb                   |
       +-----------------------+

         7  6  5  4  3  2  1  0 15 14 13 12 11 10  9  8
       +-----------------------+-----------------------+
short: | lsb     byte 0        |       byte 1      msb |
       +-----------------------+-----------------------+

         7  6  5  4  3  2  1  0 15 14 13 12 11 10  9  8 23 22 21 20 19 18 17 16 31 30 29 28 27 26 25 24
       +-----------------------+-----------------------+-----------------------+-----------------------+
 long: | lsb     byte 0        |       byte 1          |         byte 2          |       byte 3      msb |
       +-----------------------+-----------------------+-----------------------+-----------------------+
```

File Structure

A WAVE file is a collection of a number of different types of chunks. There is a required Format ("fmt ") chunk which contains important parameters describing the waveform, such as its sample rate. The Data chunk, which contains the actual waveform data, is also required. All other chunks are optional. Among the other optional chunks are ones which define cue points, list instrument parameters, store application-specific information, etc. All of these chunks are described in detail in the following sections of this document.

All applications that use WAVE must be able to read the 2 required chunks and can choose to selectively ignore the optional chunks. A program that copies a WAVE should copy all of the chunks in the WAVE, even those it chooses not to interpret.

There are no restrictions upon the order of the chunks within a WAVE file, with the exception that the Format chunk must precede the Data chunk. Some inflexibly written programs expect the Format chunk as the first chunk (after the RIFF header) although they shouldn't because the specification doesn't require this.

Here is a graphical overview of an example, minimal WAVE file. It consists of a single WAVE containing the 2 required chunks, a Format and a Data Chunk.

```
 __________________________
| RIFF WAVE Chunk          |
|   groupID  = 'RIFF'      |
|   riffType = 'WAVE'      |
|    __________________    |
|   | Format Chunk     |   |
|   |   ckID = 'fmt '  |   |
|   |__________________|   |
|    __________________    |
|   | Sound Data Chunk |   |
|   |   ckID = 'data'  |   |
|   |__________________|   |
|__________________________|
```

A Bastardized Standard

The WAVE format is sort of a bastardized standard that was concocted by too many "cooks" who didn't properly coordinate the addition of "ingredients" to the "soup". Unlike with the AIFF standard which was mostly designed by a small, coordinated group, the WAVE format has had all manner of much-too-independent, uncoordinated aberrations inflicted upon it. The net result is that there are far too many chunks that may be found in a WAVE file -- many of them duplicating the same information found in other chunks (but in an unnecessarily different way) simply because there have been too many programmers who took too many liberties with unilaterally adding their own additions to the WAVE format without properly coming to a concensus of what everyone else needed (and therefore it encouraged an "every man for himself" attitude toward adding things to this "standard"). One example is the Instrument chunk versus the Sampler chunk. Another example is the Note versus Label chunks in an Associated Data List. I don't even want to get into the totally irresponsible proliferation of compressed formats. (ie, It seems like everyone and his pet Dachshound has come up with some compressed version of storing wave data -- like we need 100 different ways to do that). Furthermore, there are lots of inconsistencies, for example how 8-bit data is unsigned, but 16-bit data is signed.

I've attempted to document only those aspects that you're very likely to encounter in a WAVE file.

B) done.

Q2

A)DONE IN CLASS.
B)DONE IN CLASS.

Q3

A) DONE IN PREVIOUS PAPER.
B)DONE.

Q4

A) IP MULTICAST

Most high-level network protocols (such as the ISO Transport Protocols or TCP or UDP) only provide a unicast transmission service. That is, nodes of the network only have the ability to send to one other node at a time:

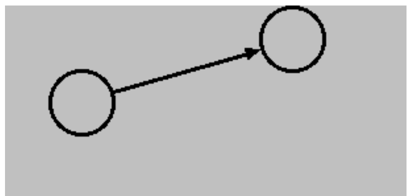


Fig. 1 - Basic Unicast Service

All transmission with a unicast service is inherently point-to-point. If a node wants to send the same information to many destinations using a unicast transport service, it must perform a replicated unicast,and send N copies of the data to each destination in turn. A better way to transmit data from one source to many destinations is to provide a multicast transport service. With a multicast transport service, a single node can send data to *many* destinations by making just a single call on the transport service:

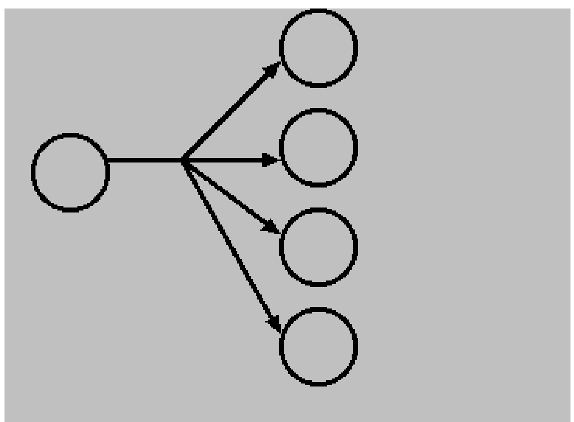


Fig. 2 - Multicast Transport Service

For those applications which involve a single node sending to many recipients, a multicast facility is clearly a more natural programming paradigm than unicast. However, the benefits of multicast are more than just logical: Many underlying transmission media (such as Ethernet) provide support for multicast and broadcast at the hardware and media-acces level. When a multicast service is implemented over such a network, there is a huge improvement in performance: If the hardware supports multicast, A packet which is destined for N recipients can be sent as just a single packet!

Multicast Applications

Multicast is is useful because it allows the construction of truly *distributed* applications, and provides important *performance* optimisations over unicast transmission. There are a number of exciting new applications for *real-time audio and video conferencing* which can make good use of a multicast service when it is available. There is currently an experimental Multicast Backbone, called the MBone, which is exploring applications of IP mulicast.

Multicast over IP networks

IP Multicast is a protocol for transmitting IP datagrams from one source to many destinations in a local or wide-area network of hosts which run the TCP/IP suite of protocols. The basic facility provided by the IP protocol is a *unicast* transmission service: That is, the current standard for IP provides only unreliable transmission of datagrams from a single source host to a single destination host. However, research has shown that only minor modifications are required to add multicast routing support to IP. The resulting IP Multicast routing protocol provides efficient delivery of datagrams from one source to an arbitrary number of destinations throughout a large, heterogenous network such as the Internet.

Q4

B) APPLICATIONS OF VR.

Health care

In the health care domain, even in the wake of improving speech recognition technologies, medical transcriptionists (MTs) have not yet become obsolete. Many experts in the field anticipate that with increased use of speech recognition technology, the services provided may be redistributed rather than replaced. Speech recognition has not yet made the skills of MTs obsolete. Speech recognition can be implemented in front-end or back-end of the medical documentation process. Front-End SR is where the provider dictates into a speech-recognition engine, the recognized words are displayed right after they are spoken, and the dictator is responsible for editing and signing off on the document. It never goes through an MT/editor.

Back-End SR or Deferred SR is where the provider dictates into a digital dictation system, and the voice is routed through a speech-recognition machine and the recognized draft document is routed along with the original voice file to the MT/editor, who edits the draft and finalizes the report. Deferred SR is being widely used in the industry currently.

Many Electronic Medical Records (EMR) applications can be more effective and may be preformed more easily when deployed in conjunction with a speech-recognition engine. Searches, queries, and form filling may all be faster to perform by voice than by using a keyboard.

Military

High-performance fighter aircraft

Substantial efforts have been devoted in the last decade to the test and evaluation of speech recognition in fighter aircraft. Of particular note are the U.S. program in speech recognition for the Advanced Fighter Technology Integration (AFTI)/F-16 aircraft (F-16 VISTA), the program in France on installing speech recognition systems on Mirage aircraft, and programs in the U.K. dealing with a variety of aircraft platforms. In these programs, speech recognizers have been operated successfully in fighter aircraft with applications including: setting radio frequencies, commanding an autopilot system, setting steer-point coordinates and weapons release parameters, and controlling flight displays. Generally, only very limited, constrained vocabularies have been used successfully, and a major effort has been devoted to integration of the speech recognizer with the avionics system.

Some important conclusions from the work are as follows: 1. Speech recognition has definite potential for reducing pilot workload, but this potential was not realized consistently. 2. Achievement of very high recognition accuracy (95% or more) was the most critical factor for making the speech recognition system useful - with lower recognition rates, pilots would not use the system. 3. More natural vocabulary and grammar, and shorter training times would be useful, but only if very high recognition rates could be maintained.Laboratory research in robust speech recognition for military environments has produced promising results which, if extendable to the cockpit, should improve the utility of speech recognition in high-performance aircraft.

Working with Swedish pilots flying in the JAS-39 Gripen cockpit, Englund (2004) found recognition deteriorated with increasing G-loads. It was also concluded that adaptation greatly improved the results in all cases and introducing models for breathing was shown to improve recognition scores significantly. Contrary to what might be expected, no effects of the broken English of the speakers were found. It was evident that spontaneous speech caused problems for the recognizer, as could be expected. A restricted vocabulary, and above all, a proper syntax, could thus be expected to improve recognition accuracy substantially.

The Eurofighter Typhoon currently in service with the UK RAF employs a speaker-dependent system, i.e. it requires each pilot to create a template. The system is not used for any safety critical or weapon critical tasks, such as weapon release or lowering of the undercarriage, but is used for a wide range of other cockpit functions. Voice commands are confirmed by visual and/or aural feedback. The system is seen as a major design feature in the reduction of pilot workload, and even allows the pilot to assign targets to himself with two simple voice commands or to any of his wingmen with only five commands.

Helicopters

The problems of achieving high recognition accuracy under stress and noise pertain strongly to the helicopter environment as well as to the fighter environment. The acoustic noise problem is actually more severe in the helicopter environment, not only because of the high noise levels but also because the helicopter pilot generally does not wear a facemask, which would reduce acoustic noise in the microphone. Substantial test and evaluation programs have been carried out in the post decade in speech recognition systems applications in helicopters, notably by the U.S. Army Avionics Research and Development Activity (AVRADA) and by the Royal Aerospace Establishment (RAE) in the UK. Work in France has included speech recognition in the Puma helicopter. There has also been much useful work in Canada. Results have been encouraging, and voice applications have included: control of communication radios; setting of navigation systems; and control of an automated target handover system.

As in fighter applications, the overriding issue for voice in helicopters is the impact on pilot effectiveness. Encouraging results are reported for the AVRADA tests, although these represent only a feasibility demonstration in a test environment. Much remains to be done both in speech recognition and in overall speech recognition technology, in order to consistently achieve performance improvements in operational settings.

Battle management

Battle management command centres generally require rapid access to and control of large, rapidly changing information databases. Commanders and system operators need to query these databases as conveniently as possible, in an eyes-busy environment where much of the information is presented in a display format. Human machine interaction by voice has the potential to be very useful in these environments. A number of efforts have been undertaken to interface commercially available isolated-word recognizers into battle management environments. In one feasibility study, speech recognition equipment was tested in conjunction with an integrated information display for naval battle management applications. Users were very optimistic about the potential of the system, although capabilities were limited.

Speech understanding programs sponsored by the Defense Advanced Research Projects Agency (DARPA) in the U.S. has focussed on this problem of natural speech interface.. Speech recognition efforts have focussed on a database of continuous speech recognition (CSR), large-vocabulary speech which is designed to be representative of the naval resource management task. Significant advances in the state-of-the-art in CSR have been achieved, and current efforts are focussed on integrating speech recognition and natural language processing to allow spoken language interaction with a naval resource management system.

Training air traffic controllers

Training for military (or civilian) air traffic controllers (ATC) represents an excellent application for speech recognition systems. Many ATC training systems currently require a person to act as a "pseudo-pilot", engaging in a voice dialog with the trainee controller, which simulates the dialog which the controller would have to conduct with pilots in a real ATC situation. Speech recognition and synthesis techniques offer the potential to eliminate the need for a person to act as pseudo-pilot, thus reducing training and support personnel. Air controller tasks are also characterized by highly structured speech as the primary output of the controller, hence reducing the difficulty of the speech recognition task.

The U.S. Naval Training Equipment Center has sponsored a number of developments of prototype ATC trainers using speech recognition. Generally, the recognition accuracy falls short of providing graceful interaction between the trainee and the system. However, the prototype training systems have demonstrated a significant potential for voice interaction in these systems, and in other training applications. The U.S. Navy has sponsored a large-scale effort in ATC training systems, where a commercial speech recognition unit was integrated with a complex training system including displays and scenario creation. Although the recognizer was constrained in vocabulary, one of the goals of the training programs was to teach the controllers to speak in a constrained language, using specific vocabulary specifically designed for the ATC task. Research in France has focussed on the application of speech recognition in ATC training systems, directed at issues both in speech recognition and in application of task-domain grammar constraints.

The USAF, USMC, US Army, and FAA are currently using ATC simulators with speech recognition provided by Adacel Systems Inc (ASI). Adacel's MaxSim software uses speech recognition and synthetic speech to enable the trainee to control aircraft and ground vehicles in the simulation without the need for pseudo pilots. Adacel's ATC In A Box Software provideds a synthetic ATC environment for flight simulators. The "real" pilot talks to a virtual controller using speech recognition and the virtual controller responds with synthetic speech.It will be a application format

Telephony and other domains

ASR in the field of telephony is now commonplace and in the field of computer gaming and simulation is becoming more widespread. Despite the high level of integration with word processing in general personal computing, however, ASR in the field of document production has not seen the expected increases in use.

Disabled people

Disabled people are another part of the population that benefit from using speech recognition programs. It is especially useful for people who are unable to use their hands, from the profoundly physically impaired to people with mild repetitive stress injuries.

Q5.

A) DONE IN CLASS. B) DONE IN CLASS.

Q6

A) DONE. B) DONE IN CLASS.

Q7.

A) DONE IN CLASS. B)DONE IN CLASS

C) TYPES OF DATABASES PAGE 121.

DEC 2007

Q-1

A) explain use of text audio and video in website. With gui as covered in class.

B) The advantage of lossy methods over lossless methods is that in some cases a lossy method can produce a much smaller compressed file than any known lossless method, while still meeting the requirements of the application.

Lossy methods are most often used for compressing sound, images or videos. The compression ratio (that is, the size of the compressed file compared to that of the uncompressed file) of lossy video codecs are nearly always far superior to those of the audio and still-image equivalents. Audio can be compressed at 10:1 with no noticeable loss of quality, video can be compressed immensely with little visible quality loss, eg 300:1. Lossily compressed still images are often compressed to 1/10th their original size, as with audio, but the quality loss is more noticeable, especially on closer inspection.

When a user acquires a lossily-compressed file, (for example, to reduce download-time) the retrieved file can be quite different from the original at the bit level while being indistinguishable to the human ear or eye for most practical purposes. Many methods focus on the idiosyncrasies of the human anatomy, taking into account, for example, that the human eye can see only certain frequencies of light. The psycho-acoustic model describes how sound can be highly compressed without degrading the perceived quality of the sound. Flaws caused by lossy compression that are noticeable to the human eye or ear are known as compression artifacts.

Lossless compression algorithms usually exploit statistical redundancy in such a way as to represent the sender's data more concisely, but nevertheless perfectly. Lossless compression is possible because most real-world data has statistical redundancy. For example, in English text, the letter 'e' is much more common than the letter 'z', and the probability that the letter 'q' will be followed by the letter 'z' is very small.

Another kind of compression, called lossy data compression, is possible if some loss of fidelity is acceptable. For example, a person viewing a picture or television video scene might not notice if some of its finest details are removed or not represented perfectly. Similarly, two clips of audio may be perceived as the same to a listener even though one is missing details found in the other. Lossy data compression algorithms introduce relatively minor differences and represent the picture, video, or audio using fewer bits.

Lossless compression schemes are reversible so that the original data can be reconstructed, while lossy schemes accept some loss of data in order to achieve higher compression. However, lossless data compression algorithms will always fail to compress some files; indeed, any compression algorithm will necessarily fail to compress any data containing no discernible patterns. Attempts to compress data that has been compressed already will therefore usually result in an expansion, as will attempts to compress encrypted data.

In practice, lossy data compression will also come to a point where compressing again does not work, although an extremely lossy algorithm, which for example always removes the last byte of a file, will always compress a file up to the point where it is empty.

Q2

A) DONE IN CLASS

B) DISCUS ALL FILE FORMATS IN BRIEF.

Q3.

A) USE NOTES.

B) DONE.

Q4.

A)ELECTRONIC LEARNING

Electronic learning or e-Learning is a general term used to refer to a form of learning in which the instructor and student are separated by space or time where the gap between the two is bridged through the use of online technologies.

E-learning is used interchangeably in so many contexts. In companies it is referred to the strategies that use the company network to deliver training courses to employees. In distance education Universities like Open University in UK or Penn State World Campus in the USA, it is defined as a planned teaching/learning experience that uses a wide spectrum of technologies mainly Internet to reach learners at a distance. Lately in most Universities, e-learning is used to define a specific mode to attend a course or programmes of study where the students rarely, if ever, attend face-to-face or for on-campus access to educational facilities, because they study on-line.

In many respects, it is commonly associated with the field of advanced learning technology (ALT), which deals with both the technologies and associated methodologies in learning using networked and/or multimedia technologies.

Technology

Many technologies can be, and are, used in e-Learning, including:

- blogs
- classroom response system
- collaborative software
- computer aided assessment
- discussion boards
- e-mail
- Educational Management System
- educational animation
- electronic performance support system
- ePortfolios
- games
- hypermedia in general
- learning management systems
- PDA's
- podcasts
- MP3 Players with multimedia capabilities
- multimedia CD-ROMs
- screencasts
- simulations
- text chat
- virtual classrooms
- web-based teaching materials
- web sites and web 2.0 communities

B) refer 7.6 application of workflow

Q5

A)DONE IN CLASS.
B)REFER DISPLAY TECHNOLOGIES FROM CHAPTER 3 .

Q6

A) DETAIN ON RAID.
B)DONE.

Q7

A)DONE.

B) VIDEO CONFERENCE DESIGN

Video-conferencing has seen rapid growth as the digital age makes the data transmission between stations more economical and widely available. Where it was once necessary to rent or install expensive facilities to communicate using broadband video and satellite or fibre links, it can now be done using video codecs capable of operating over more affordable 384kbps data lines, and those systems can connect to even less expensive rollabout units, or even a desktop PC. Increasingly, the conventional "talking head" video conference installation is being extended with interactive elements that make the room more suitable to technical and engineering applications and collaborative design. These may include electronic white boards and host PC's for shared drawings or documents, and video input devices such as document cameras, document scanners, and slide to video converters. Combining video conferencing and computer data interconnection over the Internet can turn a video conference room into an effective workroom, or a situation room.

The Elements of Video Conferencing

A video conference facility is made up of the room, the video pickup and display component, the audio pickup and delivery system, and the codec (code/decode) that is used to convert the signals into a compressed digital form to be transmitted. While the codec and the available transmission bandwidth largely determine what the best transmitted audio and video signal quality can be, it is still important to design the other components for optimum performance. It is important to remember that when there is a requirement for upgradability the easiest item to improve is the codec and the network connection, so the design of the room, and the audio and video components should support foreseeable improvements in transmission bandwidth.

The Room

The video conference room has a tremendous impact on the audio and video systems used in that room. The room acoustics affects both the microphone pickup and the audio playback in the room. Where there is full duplex voice communication, the room acoustics are a critical element in minimizing the echo-cancellation problems, and potential feedback problems. HVAC noise, reverberation and reflections that may not be noticeable to a person seated in the room can be problematic when a microphone is opened in the room. A microphone does not have the sophisticated biological processing that the human brain has for ignoring unwanted sounds in favour of the important sounds.

Ambient light affects the quality of the video display as well. Where there are reflections of light sources on the video display screen or lack of contrast caused by elevated ambient light levels the image quality will be significantly degraded.

The Video System

The modern video conference system also makes extensive use of other video sources such as document cameras to augment the communication. This can be especially valuable for technical applications where models, drawings, sketches, and broken or defective parts can be shown to the receiving station. Still images from a cmputer, DVD or a digital camera can also be effective. In addition to standard video sources, the use of active white boards and/or collaborative computing software allow material displayed on a white board in the source room to be displayed on a computer display at the receiving end. Where a similar white board and computer is located in the receiving station, true interactive design, troubleshooting and problem solving can take place in near real time.

Where the video conference room also functions as a meeting room, the video display for the video conference system can be made to double as the display for meeting support materials. The same document camera, as well as other video devices, can be used with these display systems.

The Audio System

The audio system has four major components: one or more microphones (and mixer if required); the loudspeaker system; the video codec and digital phone hybrid that handles the signal processing allowing full duplex (two way) conversation and transmission/reception of the signals; and the acoustic echo cancellation (to prevent the source station hearing itself amplified and re-transmitted from the receive location). The performance of the entire audio chain is directly tied to the room acoustics. The performance of the echo cancelling is directly tied to the decay of sound in the room, and the quality of microphone pickup is likewise affected by early reflections and reverberation. This is the reason we emphasize the importance of room acoustics as part of the system design.

Microphone placement plays an important part of the sense of distance between the talker and the receiving station. We are all familiar with the "group of people in a barrel" sound of a speaker phone. The speaker-phone sound results from a combination of an omni-directional microphone and hard surfaces in the room. The way to beat that tell-tale sound quality is by using appropriate microphones, getting the microphone closer to the talkers in the source room, and controlling the reflections from hard walls. This can help make a video conference sound like a cross-the-table meeting.

The Control System

The control of the electronics associated with an elaborate video conference system can become cumbersome as the complexity of the system increases. Where a permanently installed video conference system is used, the control of lights, drapery, video and audio equipment becomes more complicated and interconnected. An integrated control system provides ready access to controls that are located in an equipment room, or simultaneously activates several related controls in the appropriate sequence. It can control cameras, switching, routing, and even video preview or cue monitoring when a colour touch screen control device is used. A touch screen interface has a major advantage in that it can be programmed in more than one language, and can be programmed to describe the functions in plain language. This reduces the learning curve, and the intimidation presented by high tech equipment. Where the proper communication protocols are available the control of the cameras and audio at the other station is possible.

C)CHAPTER 1 DONE.

RTP SUM

- **4 users @100kbps so total bandwidth 400kbps.**
- **RTCP limits its traffic to 5% of total bandwidth.**

  So (400X5)/100=20kbps (session b/w).

- **RTCP allocates 25% bandwidth to sender and 75% to receiver.**
- **RTCP will limit its traffic for sender**

  = <u>Number of user</u>

0.25x0.5xsession b/w

= (4)/ (0.25x0.5x20)

=1.6 kbps.

- **RTCP will limit its traffic for receiver**
- **= Number of user**
- **0.75x0.5xsession b/w**
- **=(4)/(0.75x0.5x20)**
- **=0.533kbps.**

BEST OF LUCK
FROM
AMAR PANCHAL

The principal goal of education is to create men who are capable of doing new things, not simply of repeating what other generations have done.