ALLAMA IQBAL OPEN UNIVERSITY

AIOU LIBRARY
aioulibrary.com

Name:

Registration No.:

Course Code: 3477

Program: Bachelor

Semester: Autumn 2026

WhatsApp Orders: +92 324 5836626

ALLAMA IQBAL OPEN UNIVERSITY, ISLAMABAD

(Department of Computer Science)

Course: Design Fundamentals (3477) • Level: Bachelor

Semester: Autumn 2026

Assignment No. 1

Total Marks: 100 • Pass Marks: 50

SOLVED ASSIGNMENT GUIDE — For Study & Reference

Note: This guide is prepared for study and reference purposes only. Students are advised to read the relevant units of the course book, understand the concepts, and then write the assignment in their own words. Copying this guide word for word is not allowed.

AIOU LIBRARY aioulibrary.com

Question 1: Hypertext — Definition, Early Development, Navigation and Information Linking (20 Marks)

Question as printed: a) Define hypertext and explain its early development in multimedia systems. b) How does hypertext support navigation and information linking in modern systems?

Introduction

Few ideas have shaped modern computing as quietly and completely as hypertext. Every time a reader taps a highlighted word on a phone, follows a link from one web page to another, clicks a chapter title in an electronic book or jumps from a footnote to a source, that reader is using hypertext. Before hypertext, almost all writing was linear: a book was read from the first page

to the last, a file was scrolled from top to bottom, and information stored in a computer had to be found in the order in which it had been saved. Hypertext broke that single fixed order. It allowed a document to contain many possible paths, so that different readers could follow different routes through the same material according to their interest, their previous knowledge and their immediate need. This question defines hypertext, traces how the idea developed from early proposals of the 1940s and 1960s into working multimedia systems, and then explains how hypertext supports navigation and information linking in the systems we use today, from the World Wide Web to help systems, electronic textbooks and mobile applications.

What Is Hypertext? A Definition

Hypertext is a way of organising and presenting information in which pieces of text — and, in extended systems, pictures, sound and video — are connected by links, so that the reader can move directly from one piece to a related piece by selecting the link, instead of reading everything in a fixed sequence. The word was coined by Ted Nelson in the 1960s, combining 'hyper', meaning beyond or above the ordinary, with 'text'. A single unit of hypertext is often called a node: a page, a paragraph, a definition, an image or a short clip. Nodes are joined by links, and the starting point of a link — a highlighted word, phrase, button or image — is called an anchor. Selecting the anchor takes the reader to the destination node. A collection of linked nodes is sometimes called a hyperdocument, and when the nodes contain media other than text, the result is called hypermedia. Three features distinguish hypertext from ordinary text. First, it is non-linear: there is no single correct reading order. Second, it is interactive: the reader actively chooses the path, rather than passively receiving a sequence chosen by the

author. Third, it is associative: links express relationships — definition, example, evidence, contrast, next step — in the same way that human memory connects ideas by association rather than by page number.

Table 1: Ordinary (linear) text compared with hypertext

Feature	Linear text	Hypertext
Reading order	Fixed by the author, first page to last	Chosen by the reader at each link
Access to information	Sequential; earlier pages must be passed	Direct; any linked node can be reached at once
Structure	One chain of pages or paragraphs	A network (graph) of nodes and links
Reader's role	Passive receiver of a sequence	Active navigator making choices
Cross-references	Printed notes such as 'see page 40'	Selectable links that act immediately
Updating	Whole document must be reprinted or reissued	A single node or link can be changed

AIOU LIBRARY aioulibrary.com

Early Development of Hypertext

The first clear proposal came in 1945, when Vannevar Bush, a senior American science administrator, published an essay describing a hypothetical machine called the Memex. Bush imagined a desk containing a library of microfilmed documents in which a user could build 'trails' of association between items, jumping from one document to a related one and saving

the trail for later use. The Memex was never built, and it used mechanical rather than electronic ideas, but it introduced the central insight of hypertext: information becomes more useful when it is organised the way the mind works, by association, and not only by alphabetical or numerical filing. Bush's essay influenced a generation of researchers who later built real systems once computers became interactive.

In the 1960s the idea received both its name and its first working demonstrations. Douglas Engelbart's team at Stanford Research Institute built the oN-Line System (NLS), demonstrated publicly in 1968 in what is now called 'the mother of all demos'. NLS allowed documents with linked references, a mouse-driven pointer and collaborative editing — remarkably early versions of features that became ordinary decades later. In the same period Ted Nelson coined the words 'hypertext' and 'hypermedia' and began Project Xanadu, an ambitious design for a universal linked library in which every quotation would remain linked to its original source, versions would be preserved, and authors would be credited automatically. Xanadu was never completed in its full form, but its ideas — permanent links, versioning and two-way connections — still appear in discussions of how linking ought to work.

Working hypertext systems multiplied in the 1970s and 1980s as personal computers arrived. The Aspen Movie Map (1978) linked photographic images of streets so a user could take a virtual drive through a town — an early multimedia hypertext. Guide (1986) and Apple's HyperCard (1987) brought hypertext to ordinary users: HyperCard stacks of cards, joined by buttons, let teachers, students and businesses build linked information systems without being programmers, mixing text, graphics and sound. In 1989–1991 Tim Berners-Lee at CERN combined hypertext with the Internet, creating the World Wide Web: HTML for marking up

linked documents, HTTP for transferring them, and the URL for addressing any node anywhere in the world. The Web succeeded where earlier systems had not because it was simple, open and global, and hypertext moved from research laboratories into everyday life.

How Hypertext Supports Navigation in Modern Systems

Navigation means finding your way around an information space: knowing where you are, where you can go, and how to get back. Hypertext supports navigation in several practical ways. Links themselves are signposts: a highlighted word tells the reader, before clicking, roughly what lies at the other end, especially when link text is written descriptively ('admission schedule 2026' rather than 'click here'). Browsers and apps add standard navigation aids built on top of links — the Back and Forward buttons, a history list, bookmarks or favourites, tabs, and 'breadcrumb' trails that show the path Home → Section → Page. Menus, sidebars, tables of contents and site maps are all organised sets of links that give the reader an overview of the whole structure, while a search box with linked results lets a reader jump straight to a node without walking the structure at all. In electronic textbooks and help systems, 'next' and 'previous' links restore a guided path for beginners, while cross-reference links let advanced readers skip. Good hypertext design also fights the classic problem of becoming 'lost in hyperspace' by keeping link labels honest, keeping pages focused on one topic, and always providing a way home.

How Hypertext Supports Information Linking in Modern Systems

Linking is the other half of the story: it is how separate pieces of information become one connected body of knowledge. First, links join a claim to its evidence: a news article links to the

report it summarises, a course page links to the assignment file, and an encyclopaedia entry links to its sources, so the reader can verify rather than merely trust. Second, links join general material to detail on demand: a short definition in the text links to a full explanation, keeping the main page light for experts and complete for beginners – a technique called progressive disclosure. Third, links connect different media: in hypermedia, a paragraph about a historical event can link to its photograph, a map, an audio recording and a video clip, so understanding is built from several senses at once. Fourth, links connect organisations and people: academic citation networks, Wikipedia's internal links, product pages linking to manuals and spare parts, and government portals linking a service to its form, fee and office all show linking as public infrastructure. Search engines themselves are built on links: a page linked by many trusted pages is treated as important, so the link structure of the Web is also its map of authority. Modern additions include two-way links and 'linked data', where a link carries a labelled meaning (for example 'was born in') that software can process, and in-page anchor links that jump to an exact section rather than merely to a page.

- Definition remembered: hypertext = nodes of information joined by selectable links, read non-linearly.
- Historical line to quote: Bush's Memex (1945) proposed associative trails; Engelbart's NLS (1968) demonstrated linking; Nelson named hypertext and designed Xanadu; HyperCard (1987) popularised it; Berners-Lee's Web (1989–91) made it global.
- Navigation aids built on links: descriptive link text, menus, breadcrumbs, history, bookmarks, search and site maps.

- Linking functions: claim to evidence, summary to detail, text to other media, and page to related page across organisations.
- Main risk of hypertext is disorientation ('lost in hyperspace'), controlled by clear labels, focused pages and a visible home path.

Conclusion

Hypertext is therefore much more than blue underlined words. It is an organising principle — information stored as a network of nodes and links — proposed by Bush in 1945, named and developed by Nelson and Engelbart in the 1960s, spread by systems such as HyperCard, and made universal by the World Wide Web. In modern systems it supports navigation by turning every document into signposted, searchable space with a history and a way back, and it supports information linking by joining claims to evidence, summaries to detail and text to images, sound and video. A student who understands hypertext understands the basic grammar of almost every digital document, course, shop and public service in use today.

Question 2: Applications of Audio in Computer Systems; Converting Analogue Sound to Digital Form (20 Marks)

Question as printed: a) List common applications of audio in computer systems. b) Briefly explain how analogue sound is converted into digital form.

Introduction

Sound was one of the last senses to enter everyday computing — early personal computers could barely beep — yet audio is now built into almost everything a computer does, from the

lecture a student records on a phone to the spoken warning in a car, the music streamed by a shop and the voice note sent to a family group. Audio matters because hearing is fast, works without looking at a screen, and carries tone and emotion that text cannot. This answer first lists the common applications of audio in computer systems, grouped so that each use is easy to remember, and then explains, step by step, how a continuous analogue sound — a voice, a musical instrument, the noise of a street — is converted into the numbers a computer can store, copy, edit and send. That conversion, called analogue-to-digital conversion, rests on three operations: sampling, quantisation and coding, governed mainly by the sampling rate and the bit depth.

Part (a): Common Applications of Audio in Computer Systems

Communication is the most familiar application. Computers carry voice in video and audio calls, voice notes, internet telephony (VoIP) and conference systems used by offices, teachers and doctors; the same microphones and speakers support voice messages in social applications. Closely related is speech interaction: voice assistants answer questions, speech-to-text software types dictated documents and examination answers for users who cannot type comfortably, and text-to-speech software reads screens aloud for visually impaired users — audio here is an accessibility lifeline, not a luxury. Entertainment is the largest consumer use: music recording, editing and streaming, film and video soundtracks, podcasts, audiobooks and computer games, where sound effects and background music create presence and give feedback on events the player may not see.

Education and training use audio constantly: recorded lectures and language-learning material, pronunciation practice that compares a learner's voice with a model, oral examinations conducted online, and multimedia courseware in which narration explains a diagram while the student's eyes stay on the picture. Business and public systems use audio for customer-service call recording, announcements in stations and hospitals, alarms and warning sounds, hold music and voice menus ('press 1 for admissions'). Computers also work with audio behind the scenes: sound analysis identifies songs, detects a baby crying or glass breaking in security systems, transcribes court and meeting recordings, and helps doctors listen to recorded heart and lung sounds. Finally, audio is a feedback channel of the interface itself — the click, chime and error tone that confirm an action without demanding visual attention.

AIOU LIBRARY aioulibrary.com

- Communication: voice and video calls, VoIP, voice notes, conferencing.
- Speech interaction and accessibility: voice assistants, dictation (speech-to-text), screen readers (text-to-speech).
- Entertainment: music production and streaming, film sound, podcasts, audiobooks, game sound.
- Education: recorded lectures, language and pronunciation practice, narrated courseware, online oral assessment.
- Business and public systems: call recording, announcements, alarms, voice menus, meeting transcription.
- Analysis and recognition: song identification, sound-event detection in security, medical and industrial listening.

- Interface feedback: clicks, chimes and warning tones confirming actions.

Part (b): How Analogue Sound Is Converted into Digital Form

Sound in the air is analogue: air pressure rises and falls smoothly and continuously, and a microphone turns those changes into a continuously varying electrical voltage that mirrors the wave. A computer cannot store a smooth curve directly; it stores numbers. The conversion is done by an analogue-to-digital converter (ADC), usually on the sound card or built into the phone or computer, and it happens in three steps. The reverse journey, from numbers back to sound in a loudspeaker, is done by a digital-to-analogue converter (DAC) using the same measurements in reverse.

Step 1 — Sampling. The converter measures (samples) the height of the analogue wave at regular intervals and records each measurement. The number of measurements per second is the sampling rate, measured in hertz. CD audio uses 44,100 samples per second (44.1 kHz); speech in telephony may use 8,000. The Nyquist rule states that the sampling rate must be at least twice the highest frequency we wish to capture; human hearing reaches about 20 kHz, which is why CD rate is a little above 40 kHz. Sampling too slowly loses high notes and produces a false low sound called aliasing, so real systems filter out frequencies above half the sampling rate before converting.

Step 2 — Quantisation, and Step 3 — Coding. Each sample's height must be written as one of a fixed set of levels; choosing the nearest level is quantisation, and writing that level in binary is coding. The number of levels is set by the bit depth: 8 bits give 256 levels, while CD's 16 bits give 65,536 levels, so each sample is recorded much more accurately and the quiet sounds and

fine detail survive. The small difference between the true height and the nearest level is quantisation error, heard as background noise if the bit depth is too low. The result of the three steps is a stream of binary numbers — for uncompressed CD stereo, about 1.4 million bits per second — which can then be stored as WAV, or compressed into formats such as MP3 or AAC by discarding sounds listeners barely notice. The diagram below shows the smooth wave, the moments at which it is sampled, and the staircase of quantised levels that the computer actually keeps.

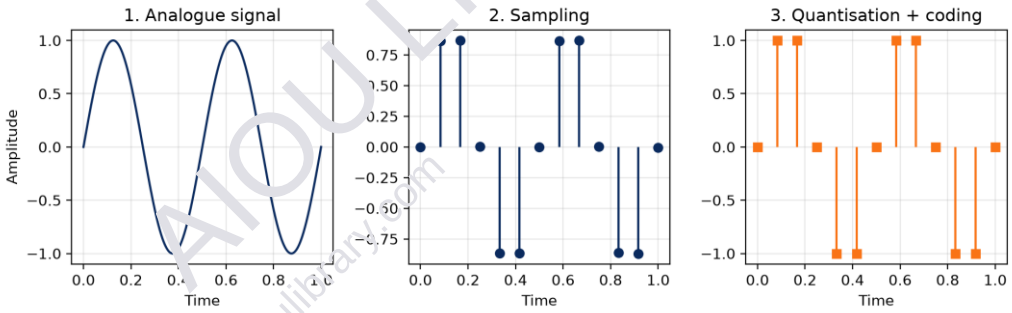

Figure 1: Analogue-to-digital conversion — the continuous wave is sampled at regular intervals, and each sample is quantised to the nearest level and coded in binary.

Table 1: Two quality settings compared

Setting	Sampling rate	Bit depth	Typical use
Telephone speech	8 kHz	8 bits (companded)	Voice calls — speech remains understandable, music sounds thin
CD audio	44.1 kHz	16 bits	Music recording and playback with full

			audible range
Studio / high-resolution	96 kHz	24 bits	Professional recording and editing headroom

A practical example ties the steps together. A student records a two-minute lecture on a phone. The microphone produces an analogue voltage; the phone's ADC samples it — commonly 48,000 times per second at 16 or 24 bits; the numbers are compressed, usually into AAC or Opus, so two minutes fit in a few megabytes instead of tens of megabytes; and the file can now be copied without any loss of quality, edited to remove a cough, slowed for revision, transcribed automatically and shared with the whole class — none of which an analogue tape could do without degrading at every copy. Digital form is what makes audio editable, searchable and permanent, which is why conversion quality (rate and depth chosen for the purpose) matters in every application listed in part (a).

Conclusion

Audio in computer systems serves communication, speech interaction and accessibility, entertainment, education, business systems, automatic analysis and simple interface feedback. All of these depend on one foundational process: an ADC samples the analogue wave at a rate at least twice the highest frequency present, quantises each sample to one of the levels allowed by the bit depth, and codes the result in binary for storage, processing and transmission — with a DAC reversing the process at the loudspeaker. Choosing the sampling rate and bit depth is choosing the quality and size of every digital sound a system will ever handle.

Question 3: Raster Scanning — How Images Are Displayed Using Scan Lines and Frame Buffers (20 Marks)

Question as printed: Explain raster scanning and describe how images are displayed on a screen using scan lines and frame buffers.

Introduction

Every photograph, film and computer screen a student looks at today is a raster image: a picture built from a rectangular grid of tiny coloured dots called pixels (picture elements). Raster scanning is the orderly process by which that grid is produced on a display, one row at a time, and the frame buffer is the memory that holds the grid while it is being shown. The two ideas belong together: the frame buffer stores what the picture is, and raster scanning is how the stored picture is read out, line by line, fast enough that the eye sees a steady, complete image instead of a moving point of light. This answer explains the raster idea, describes the scanning process step by step including the scan line and the retrace, explains the frame buffer's structure and size, and shows how the processor, the display controller and the screen cooperate — the same scheme used, with modern variations, in monitors, televisions, phones and projectors.

The Raster and the Scan Line

A raster is the complete grid of pixel positions covering the screen: for a Full HD display, 1,920 pixels across each row and 1,080 rows down the screen, giving just over two million pixels. A scan line is one horizontal row of that grid. In the original cathode-ray tube (CRT) displays, a

beam of electrons swept across the inside of the screen, exciting phosphor dots whose brightness was set by the signal, then switched off and flew back — the horizontal retrace or flyback — to begin the next line slightly lower down. After the last line at the bottom, the beam returned to the top (vertical retrace) and the whole picture, called a frame, began again. Modern LCD and OLED screens have no moving beam, but they keep exactly the same logical order: pixels are still addressed and refreshed row by row, top to bottom, because cameras, video signals, graphics hardware and software were all standardised around that order. Scanning the full frame typically repeats 60 times per second (60 Hz) or more; at that speed the eye fuses the successive frames into a stable image and any change appears as smooth motion.

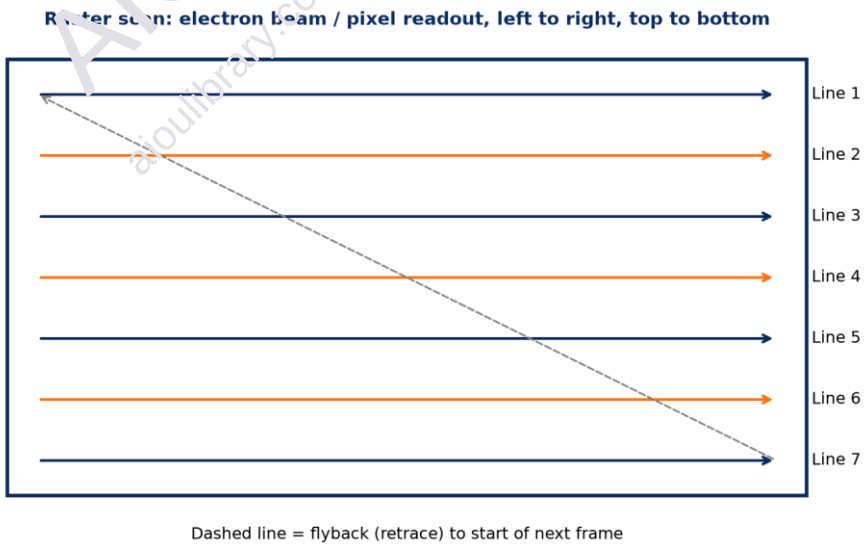

Figure 1: Raster scanning — the image is written one scan line at a time, left to right and top to bottom, with flyback between lines and between frames.

The Frame Buffer: Memory That Holds the Picture

The frame buffer (also called video memory or VRAM in a graphics card) is a block of fast memory that stores one value for every pixel on the screen: its colour. If each pixel is stored in 24 bits — 8 bits each for red, green and blue — a Full HD frame buffer holds 1,920 × 1,080 × 3 bytes, about 6.2 megabytes, and allows about 16.7 million different colours per pixel. Simpler systems used fewer bits: 1 bit per pixel gives a black-and-white image, 8 bits allow 256 colours, often chosen through a colour lookup table (palette) so the stored number is an index into a table of full colours. The essential point is that the frame buffer decouples the picture from the processor: the CPU or GPU writes new pixel values into the buffer whenever the picture changes, while separate display hardware reads the buffer out continuously, in raster order, whether or not the picture is changing. A still desktop is therefore being redrawn from the buffer dozens of times per second, exactly like a film frame.

The readout path works as follows. A display controller (video controller) steps through the buffer addresses in order — pixel (0,0), (1,0), and so on along the first scan line, then the second line, until the last pixel of the last line. For each pixel it fetches the stored value, converts the digital colour into the signal the panel needs (in CRT days, through digital-to-analogue converters for the red, green and blue guns; in modern panels, as digital data over HDMI or DisplayPort), and advances. Timing signals travel with the data: a horizontal sync marks the end of each line, and a vertical sync marks the end of each frame, so the screen and the controller never drift apart. Many systems use double buffering: two frame buffers are kept, one being displayed while the other is being drawn, and they are swapped at the vertical

sync. This removes tearing — the visible split when a half-drawn frame is shown — and is standard in games and video playback.

Table 1: Frame buffer size for common resolutions (24 bits per pixel)

Resolution	Pixels	Frame buffer size (approx.)
HD — 1280 × 720	921,600	2.8 MB
Full HD — 1920 × 1080	2,073,600	6.2 MB
QHD — 2560 × 1440	3,686,400	11.1 MB
4K UHD — 3840 × 2160	8,294,400	24.9 MB

AIOU LIBRARY aioulibrary.com

Raster Display Compared with Random-Scan (Vector) Display

It is worth contrasting raster scanning with the older random-scan or vector approach, in which the beam was steered directly along the lines of the drawing, like a pen following a path, and only the objects on the screen were stored as drawing commands. Vector displays produced beautifully sharp lines and needed little memory, which is why they were used in early CAD and oscilloscope-style graphics. But they could not fill areas with colour or show photographs at all, and the drawing had to be re-traced within the phosphor's glow time, so complex scenes flickered. Raster scanning won because memory became cheap: once a frame buffer can hold millions of coloured pixels, any image — text, photographs, film, shaded 3D scenes — can be displayed by the same simple, regular process, and the cost of the picture no longer depends on its complexity in the same way. Modern GPUs still think in this raster model first, converting 3D scenes into pixels ('rasterisation') before writing them to the frame buffer.

- Raster = the full rectangular grid of pixels; scan line = one horizontal row of that grid.

- Scanning order is always left to right along a line, line by line to the bottom, then back to the top — repeated at the refresh rate (commonly 60 Hz or more).
- Frame buffer = fast memory holding one colour value per pixel; its size = width × height × bits per pixel ÷ 8.
- The display controller reads the buffer in raster order continuously, with horizontal and vertical sync signals keeping screen and memory in step.
- Double buffering (display one buffer, draw into the other, swap on vertical sync) prevents tearing and flicker.

Conclusion

Raster scanning is the discipline that turns a stored grid of numbers into a visible picture: the frame buffer holds the colour of every pixel, the display controller reads that memory out one scan line at a time — left to right, top to bottom, frame after frame at the refresh rate — and the screen lights each pixel accordingly. Because the eye is slower than the scan, the moving point of activity is perceived as a complete, steady image. Understanding scan lines and frame buffers explains screen resolution, refresh rate, video memory size and artefacts such as tearing, and it is the foundation on which all later topics in this course — image formats, video and compression — are built.

Question 4: Video Compression — Spatial and Temporal Compression Techniques (20 Marks)

Question as printed: What is video compression? Explain the difference between spatial and temporal compression techniques.

Introduction

Uncompressed video is astonishingly large. A single second of ordinary high-definition video — 24 or 30 still pictures, each of about two million pixels stored at three bytes per pixel — needs roughly 150 to 190 megabytes, so a two-hour film would run into hundreds of gigabytes and could never be streamed over a home connection or stored on a phone. Yet films are streamed daily in a few gigabytes or less. The technology that makes the difference is video compression: the set of methods by which a video is represented in far fewer bits for storage and transmission, and reconstructed for viewing, ideally with no loss of quality that the viewer can notice. Compression works because video contains enormous redundancy — information repeated or predictable that need not be stored again — and because the human eye is forgiving of certain kinds of small error. This answer defines video compression, identifies the main kinds of redundancy, and then explains the two great families of technique, spatial compression within a single frame and temporal compression between frames, and how real systems combine them.

AIOU LIBRARY aioulibrary.com

What Is Video Compression?

Video compression is the process of encoding a sequence of video frames into a smaller number of bits (the encoder's job) in such a way that a decoder can reconstruct a viewable version of the same sequence. Compression is measured by the compression ratio (original size divided by compressed size) or by the resulting bitrate in bits per second. Methods divide first into lossless compression, where the decoded picture is bit-for-bit identical to the original — used in archives and professional editing, achieving modest ratios of about 2:1 to 3:1 on video — and lossy compression, where some information is permanently discarded in exchange for much higher ratios, often 50:1 to 200:1, with the discarding carefully aimed at detail the eye notices least. Almost all delivered video — broadcast, streaming, video calls, DVDs and Blu-ray — is lossy, using standards such as MPEG-2, H.264/AVC, H.265/HEVC, VP9 and AV1. A compressor hunts for three kinds of waste: spatial redundancy (similarity within one picture), temporal redundancy (similarity between pictures), and statistical/perceptual redundancy (patterns in the data, and detail the eye cannot see). Spatial and temporal compression attack the first two, and they are the subject of the rest of this answer.

Spatial Compression: Removing Redundancy Within a Frame

Spatial compression, also called intra-frame compression, treats each frame as an individual still picture and removes the repetition inside it, without reference to any other frame. Neighbouring pixels in real images are usually similar: a blue sky, a wall, a face — large areas change slowly, and edges occupy a small part of the picture. A spatial compressor exploits this by transforming blocks of pixels (commonly 8 × 8) with the Discrete Cosine Transform (DCT)

into frequency components, so that most of the block's energy is concentrated in a few low-frequency coefficients; the high-frequency coefficients, representing fine detail the eye is less sensitive to, are quantised coarsely or discarded, and the remaining numbers are packed efficiently with run-length and Huffman (entropy) coding. This is exactly the JPEG method applied to a picture. A frame compressed only spatially is an intra-coded frame, or I-frame (keyframe): it is completely self-contained and can be decoded on its own, which is why video players can jump to I-frames when a user seeks. Spatial compression alone, as in the MJPEG format, typically achieves about 10:1 to 20:1 — useful, but far short of what streaming needs, because it ignores the biggest redundancy of all: the similarity between successive frames.

Temporal Compression: Removing Redundancy Between Frames

Temporal compression, also called inter-frame compression, exploits the fact that consecutive frames of a video are usually almost identical. In a news broadcast, the background, the desk and most of the presenter stay unchanged for seconds at a time; only the lips, eyes and hands move. Storing every frame in full would store that unchanged background thirty times a second. Instead, temporal compression stores a full reference frame occasionally and, for the frames between, stores only the differences. The key technique is motion-compensated prediction: the encoder divides the current frame into blocks, searches the previous frame for where each block has moved to, and records a motion vector ('this block is the same as that one, shifted four pixels left') plus a small residual — the difference between the predicted block and the actual block, which is itself spatially compressed. Predicted frames are called P-frames (predicted from a previous frame); B-frames (bidirectional) are predicted from both a previous and a following frame, filling in areas that become uncovered, and are the most compact of all.

A typical pattern — a group of pictures (GOP) — might be I B B P B B P …, with one self-contained I-frame every half-second or so and the cheap predicted frames between, lifting total compression to 50:1–200:1.

Table 1: Spatial and temporal compression compared

Aspect	Spatial (intra-frame)	Temporal (inter-frame)
Redundancy removed	Within one frame — similar neighbouring pixels	Between frames — unchanged or merely moved content
Reference needed	None; the frame is self-contained	Previous (and for B-frames, following) reference frames
Main techniques	Block transform (DCT), quantisation, entropy coding	Motion estimation and compensation, difference (residual) coding
Frame types produced	I-frames (keyframes)	P-frames and B-frames
Typical contribution	About 10:1–20:1 on its own	Raises total to roughly 50:1–200:1 in a full system
Strengths	Random access, editing, robustness — one damaged frame does not spread	By far the largest savings in most real video
Weaknesses	Limited savings; ignores frame-to-frame repetition	Seeking and editing are harder; errors can propagate until the next I-frame; fast motion and scene changes compress poorly

AIOU LIBRARY aioulibrary.com

The difference between the two techniques can be held in one sentence each. Spatial compression asks, 'Within this picture, what need not be repeated?' Temporal compression asks, 'Compared with the pictures before and after, what has actually changed?' Their weaknesses are mirror images: a sudden scene change defeats temporal prediction, so the encoder spends bits on a fresh I-frame — which is why action films and sports, full of fast movement and cuts, need higher bitrates than a lecture or a video call with a still background. Conversely, spatial-only systems waste bits redescribing an unchanging wall every frame. Error behaviour also differs: damage to an I-frame affects only that frame, whereas damage to a reference frame can corrupt every P- and B-frame predicted from it until the next keyframe arrives, one reason streaming services insert I-frames regularly and live broadcasters balance GOP length against channel-change time.

How Real Systems Combine Both

Every mainstream video codec — MPEG-2, H.264, HEVC, AV1 — is a hybrid: temporal prediction first produces a predicted frame and a residual, and spatial techniques then compress that residual and the periodic I-frames. Around this core sit supporting tools: variable bitrate control that spends more bits on complex seconds and fewer on simple ones, entropy coding (CABAC or Huffman-style methods) for the final packing, and deblocking filters that hide the edges of blocks after coarse quantisation. When a student chooses a quality setting in a video app or watches a streaming picture soften when the connection slows, they are seeing this hybrid machinery trading spatial detail and temporal smoothness against the bits available.

Conclusion

Video compression makes digital video practical by removing redundancy and imperceptible detail so that a sequence can be stored and transmitted in a small fraction of its raw size. Spatial compression works inside a single frame, using transforms and quantisation to exploit the similarity of neighbouring pixels, and produces self-contained I-frames. Temporal compression works between frames, using motion compensated prediction to store only changes, and produces compact P- and B-frames. Neither alone is sufficient for modern streaming; combined in a hybrid codec with a regular pattern of keyframes, they routinely deliver 50:1 to 200:1 compression with quality that viewers accept as the picture itself.

Question 5: JPEG Compression and MPEG Video Compression (20 Marks)

AIOU LIBRARY aioulibrary.com

Question as printed: a) What is JPEG compression? Briefly describe its main steps. b) What is MPEG? Explain its basic approach to video compression.

Introduction

Two names dominate the world of compressed pictures. JPEG is the standard that made digital photography and web images practical: almost every photograph on a phone, a website or a social feed is a JPEG file. MPEG is the family of standards that did the same for moving pictures and sound: DVDs, digital television, and the ancestors of today's streaming formats all descend from MPEG work. The two are closely related — MPEG video begins, in effect, by compressing individual frames in a JPEG-like way and then adds a second, powerful idea: it also compresses

the differences between frames. This answer explains what JPEG compression is and walks through its main steps one by one, then introduces MPEG, the group behind it and its best-known standards, and explains its basic approach — frame types, motion compensation and the group of pictures — with a comparison that shows exactly where the two methods meet and where they part.

Part (a): What Is JPEG Compression?

JPEG (Joint Photographic Experts Group, the committee that created it in 1992) is a lossy compression standard for still images, especially photographs and other continuous-tone pictures with gradual changes in colour. It typically shrinks a photograph to about one-tenth to one-twentieth of its original size with little visible loss, by discarding fine detail that the human eye is poor at seeing — particularly small colour changes and very fine texture — while keeping brightness structure, to which the eye is most sensitive, more carefully. The user-visible 'quality' setting in cameras and editors is a JPEG setting: it controls how aggressively detail is discarded. JPEG is a poor choice for sharp-edged material such as text screenshots and line diagrams, where its block method creates visible smudging ('artefacts') around edges; PNG or WebP serve those better. But for photographs its balance of size, quality, speed and universal support has kept it dominant for three decades.

The Main Steps of JPEG Compression

Step 1 — Colour conversion and (often) subsampling. The image is converted from RGB into a brightness-plus-colour representation (YCbCr): one luminance channel Y, which carries the detail the eye sees best, and two chrominance channels carrying colour differences. Because

the eye notices colour detail less than brightness detail, the colour channels are commonly subsampled — stored at half resolution across, or half in both directions (4:2:0) — immediately discarding a large share of the data with almost no visible effect. Step 2 — Division into blocks and the DCT. Each channel is divided into 8 × 8 pixel blocks, and each block is transformed by the Discrete Cosine Transform into 64 frequency coefficients: one DC coefficient representing the block's average, and 63 AC coefficients representing change at increasing fineness. In natural photographs most of the block's information concentrates in a few low-frequency coefficients near the top-left corner.

Step 3 — Quantisation (the lossy step). Each coefficient is divided by a number from a quantisation table and rounded. High-frequency coefficients are divided by large numbers and mostly become zero — this rounding is the one irreversible step where information is genuinely thrown away, and the quality setting works by scaling these tables. Step 4 — Zigzag scan and run-length coding. The quantised block is read in a zigzag order from low to high frequency, which groups the long run of trailing zeros together; the sequence is then run-length coded ('twelve zeros, then a 3'). Step 5 — Entropy coding. Finally the numbers are Huffman-coded (arithmetic coding is an allowed alternative), giving frequent values short codes. The compressed stream plus header information — dimensions, tables and settings — forms the .jpg file. Decoding reverses the steps: entropy decoding, dequantisation (approximately, since the discarded detail cannot return), inverse DCT, and conversion back to RGB.

AIOU LIBRARY aioulibrary.com



Figure 1: The main steps of JPEG compression, from 8 × 8 blocks to the finished file.

Table 1: JPEG steps at a glance

Step	What happens	Lossy?
Colour conversion + subsampling	RGB → YCbCr; colour channels stored at lower resolution	Subsampling discards colour detail
Block DCT	Each 8 × 8 block → 64 frequency coefficients	No (reversible in principle)
Quantisation	Coefficients divided and rounded using tables; most high frequencies become zero	Yes — the main loss
Zigzag + run-length	Coefficients ordered low→high frequency; runs of zeros coded compactly	No
Entropy (Huffman) coding	Frequent values given short binary codes	No

AIOU LIBRARY aioulibrary.com

Part (b): What Is MPEG?

MPEG (Moving Picture Experts Group) is the international working group, established in 1988 under ISO/IEC, that standardises compression of moving pictures and associated audio — and, by extension, the name of the family of standards it produced. The best known are MPEG-1 (1993), which made the Video CD and carried the famous MP3 audio layer; MPEG-2 (1995), the standard of DVD and digital television broadcasting; MPEG-4 (1999 onwards), a broad multimedia framework whose video part AVC/H.264 became the dominant format of the streaming and smartphone era; and the later HEVC/H.265 line that continued under joint MPEG and ITU work. An MPEG standard specifies the format of the compressed bitstream and how a decoder must reconstruct the pictures — not how the encoder must think — so manufacturers compete on encoder quality while every compliant player can decode every compliant file.

MPEG's Basic Approach to Video Compression

MPEG's approach is a hybrid of the spatial and temporal compression explained in Question 4. It organises the video into a repeating group of pictures (GOP) containing three kinds of frame. I-frames (intra) are complete pictures compressed spatially, very much like JPEG — block DCT, quantisation and entropy coding — and need no other frame to decode. P-frames (predicted) store a compressed description of the change from the previous I- or P-frame. B-frames (bidirectional) are predicted from both neighbouring reference frames and are the smallest. The engine of P- and B-frames is motion estimation and compensation: the encoder divides the picture into macroblocks (traditionally 16 × 16 pixels), searches reference frames for the best matching position for each block, and transmits a motion vector plus the residual difference

WhatsApp Orders: +92 324 5836626

between the real block and its prediction, with the residual compressed by the same DCT-and-quantisation machinery as a JPEG image. Because in most scenes most blocks barely change, the residuals are small and full of zeros, and MPEG routinely reaches overall compression of 50:1 to 200:1 for delivered video, with the audio (AAC or MP3-family) multiplexed into the same stream with timestamps that keep sound and picture in step.

Two consequences follow from this approach and are worth stating in an examination answer. First, access and editing: because only I-frames stand alone, players seek to the nearest I-frame, editors prefer I-frame-heavy material, and a damaged reference frame can spoil the frames predicted from it until the next I-frame — the price paid for temporal savings. Second, adaptivity: the encoder can trade quality against bitrate continuously by changing quantisation and GOP structure, which is exactly what allows the same MPEG-family technology to serve a shaky mobile connection and a cinema-sized television from the same master. JPEG and MPEG are therefore best understood as parent and child: JPEG compresses one picture by removing spatial redundancy; MPEG keeps that machinery and adds motion-compensated removal of temporal redundancy between pictures.

Table 2: JPEG and MPEG compared

Aspect	JPEG	MPEG (family)
Media	Single still image	Video sequence with associated audio
Redundancy attacked	Spatial (within the picture)	Spatial and temporal (within and between pictures)
Core tools	DCT, quantisation, zigzag,	Same tools plus motion

	Huffman coding	estimation/compensation, I/P/B frames, GOP structure
Typical result	About 10:1–20:1 on photographs	About 50:1–200:1 on delivered video
Typical uses	Photographs, web images	DVD, digital TV, streaming (via its successors)

Conclusion

JPEG is the standard lossy method for still photographs: convert to brightness-and-colour form (with colour subsampling), split into 8 × 8 blocks, transform with the DCT, quantise — the step where quality is traded for size — then order by zigzag, run-length and Huffman coding into a compact file. MPEG is both the expert group and the family of standards (MPEG-1, MPEG-2, MPEG-4/AVC and successors) that extended the same spatial machinery to moving pictures, adding groups of pictures made of I-, P- and B-frames and motion-compensated prediction so that only changes between frames are stored in detail. Together they are the reason photographs and films fit on the devices and networks students use every day.

AIOU LIBRARY aioulibrary.com