Multimedia Theory (Finals) - Jaish Khan

Multimedia is computer information which can be represented through audio, video, animations, text, graphics or images. A *Multimedia Application* is then a collection of multiple media sources.

A **Multimedia System** is then a system capable of processing, storing, generating, manipulating and rendering multimedia. They have 4 characteristics (*computer controlled*, *integrated*, *digital*, *interactive*).

- Tooling required for Multimedia Systems
  - **Hardware** → Capture Devices, Storage Devices, Networks, Display Devices and Computer Systems.
  - **Software** → Software Tools and Applications.

It might also need **Synchronization** as different sources like audio, video and text can get out of sync.

Hypertext ⟶ A system of organizing and presenting text-based information with links that allow users to navigate between related topics or documents.
Hypermedia ⟶ An extension of hypertext that incorporates multimedia elements such as images, audio, video, and animations along with text.

Feature	Hypertext	Hypermedia
Focus	Text and links	Multimedia (text, audio, video, etc.)
UX	Simple, text-based navigation	Rich, interactive, multimedia experience
Examples	Wikipedia, text-based HTML pages	YouTube, multimedia-rich websites
Application	Documentation, articles, reference	E-learning, entertainment, interactive apps
Key Features	Linear and non-linear navigation through text.	Non-linear navigation with rich media.

Animation Techniques ⟶ *Cel animation*, *Path animation*, *Morphing*.

Steganography → The practice of hiding/concealing information within other non-suspicious data, making the hidden message undetectable to the human eye or conventional analysis.

1. Data Formats

Different types of data requires differen types of inputs and ways to store it.

Data Format	Input	Storage
Text	Keyboard, OCR, Voice	1B per ASCII >1B per Unicode
Images	Digital Cameras and Scanners	1b/pixel (Black/White) 8b/pixel (Grayscale) 24b/pixel (Truecolor) 32b/pixel (TC with Alpha)
Graphics	Generated by programs	Not much as its based on code
Audio	Microphones	1min, Mono → 5MB 1min, Stereo → 10MB
Video	Video Cameras	1s, HD → 150MB 1min, HD → 9GB

- **Graphics Standards:** OpenGL, PHIGS, GKS.

Data can be **Static** (Discrete) → Text, Images, Graphics or **Dynamic** (Continuous) → Audio, Video, Animations. Data can also be digital or analog and can be converted using **ADC/DAC**

OCR (*Optical Character Recognition*) → A method used to scan text data from physical objects.

1.1. Multimedia Data Compression

Because the multimedia can reach enormous sizes (especially video) so compression becomes necessary. There are two ways of doing it:

- **Lossless** → Preserves all data.
  - Examples: Zip (Text), PNG (Images), FLAC (Audio) etc...
- **Lossy** → Discards perceptually less relevant data to achieve higher compression ratios.
  - Examples: JPEG (Images), MP3 (Audio), AVC (Video) etc...

A **codec** (Coder-Decoder) is used to compress and decompress data.

2. Text

A **Typeface** is a family of characters that include many sizes and styles like Times, Arial, Helvetica etc.

- The size of a font is measured in **Points** and is measured from the top of the ascender to the bottom of the descender. $1p = \frac{1}{72}\ inch = 0.0138\ inch$.
- **Font** → Collection of characters of a single point-size and style belonging to a typeface family like Times 12-point italic. A *Font Family* is a collection of different styles of the same font.
- **Styles** → Bold, Italic etc...
- **Serif** → Small lines attached to the ends of letters.
  - Fonts which have them are called *Serif* Fonts (more readable on printed media) and those without them are called *Sans-Serif* Fonts (more readable on screens).
- **Leading** → Vertical space between lines of text (measured from baseline to baseline).
- **Tracking** → Horizontal space between characters in a block of text (applies as a whole).
- **Kerning** → Horizontal space between individual characters to make them more readable (applies to pairs of characters).
- **X-height** → Height of the lowercase 'x' in a typeface.
- **Other Character Metrics**
  - Baseline → The line on which most characters sit.
  - Ascender → The part of lowercase letters that goes above x-height such as h, b and f.
  - Descender → The part of lowercase letters that goes below baseline suchas p, q and y.
  - Counter → The whitespace inside letters such as O, A and P.
  - Set width → The horizontal space a character takes.
  - Cap height → The height of capital letters.
  - Height → The total height of a character including ascenders/descenders.
  - Cap line → The line which marks the top of uppercase characters.
  - Mean line → The line which marks the top of lowercase characters (ignoring ascenders).

Fonts are bundled in different file types. *Bitmap* fonts can not be altered while *TrueType* and *PostScript* fonts can be. A letter on the screen using dots/pixels.

2.1. Legibility

It refers to how easily characters can be identified/distinguished from each other in a text. It is influenced by *Font Size*, *Background and Foreground Color*, *Font Style* and *Spacing*.

Case can also affect legibility in that it is easier to read words with mixture of upper and lower case letters VS all upper case letters. Capitalization Schemes → UPPER CASE, lower case, Title Case (**Intercap**), Sentence case etc.

2.2. Design Tips

A designer should use the most legible font available and not use as multiple different types faces. It is called **ransom-note** typography.

- Normal line length is `10` words or `70` characters. On mobiles its even less.
- Use **bold** and *italics* to convey meaning. Also experiment with other styles like underlines, outlines and shadows.
- All caps or All italics makes reading difficult so avoid that. All caps denotes Shouting.
- Adjust spacing between lines (leading) and the spacing between letters in headings to remove gaps (tracking). Also adjust the spacing between individual letters (kerning).
- Use background and foreground colors to increase legibility. **Reverse Type** is light text on dark background.
- Use anti-aliased text as it blends the colors along the edges of letter.
- Use **whitespace** {space: ASCII 32, tab: ASCII 9 and newlines}. A nonbreaking space entity ` ` is used to force spaces into lines of text.
- *Portrait* (Taller-than-Wide) vs *Landscape* (Wider-than-Tall) screen orientation.
- Keep font sizing in your mind. Fonts smaller than 12-points are not very legible on a monitor.

> People blink 3-5 times/minute, using a computer and 20-25 times/minute reading a book. This causes fatigue, dryness and hence makes reading much slower comparatively to reading a book.

- **ASCII** → 128 characters, only supports English-like languages. Every character takes 1 byte.
- **Unicode** → 65536 characters, supports every character from every single language. Every characters takes 1-4 bytes (depending on the language).

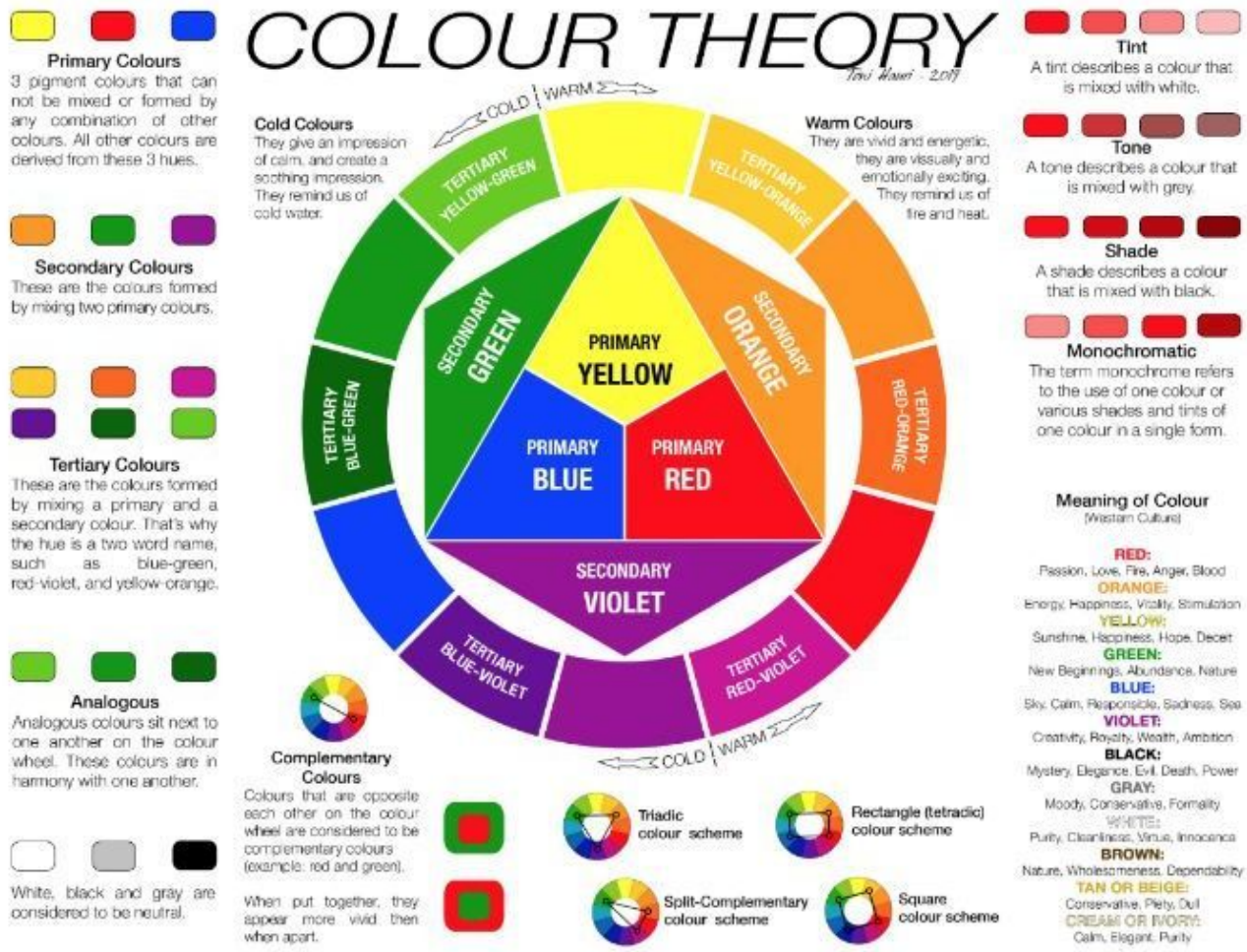
---

3. Color

Color conveys meaning which is different depending on the culture. It is affected by light, context and environment.

3.1. Color Wheel

When it comes to Art and Design, the RBY (Red, Blue, Yellow) has been used.



Primary Colors → [Red, Blue, Yellow].
Mixing them gives us 3 more:
Secondary Colors → [Orange, Green, Violet].
Mixing them gives us 6 more colors:
Tertiary Colors → [Red-Orange, Red-Violet, Blue-Violet, Blue-Green, Yellow-Green, Yellow-Orange].

White, Black and Gray are considered neutral colors.

3.1.1. Color Strategies

Also called Color Harmony Schemes.

It is the process of choosing colors to be used such that they follow one of these schemes and are visually appealing.

1. *Monochromatic* → Uses variations in lightness and saturation of a single color.
2. *Grayscale* → Uses only black, white and grays.
3. *Analogous* → Uses colors that are adjacent/next to each other.
    1. Warm vs Cool Colors
4. *Complimentary* → Uses colors that are opposite to each other.
5. *Triadic* → Uses 3 colors that are 120° apart.
6. *Split-Complimentary* → Uses a base color and then two colors adjacent to its complimentary.
7. *Tetradic* → Uses 4 colors that are in pairs of two-complementaries.
8. *Discord* → Uses mismatched colors for an eye-catching effect.

Some Terms:

- **Color Contrast** → The difference between the darkest darks and the brightest brights.
- **Color Gamut** → It is the set of all colors that can be produced or recorded.
- **Dithering** → Process of simulating colors that are not in the palette by mixing pixels of different colors.
- **Color Palettes** → The colors that are at our disposal (we can use).

- **Palette Flashing:** Occurs when a series of images, each with its own color palette, are displayed, causing a flashing effect. Solutions include using a single palette for all images or fading images to black or white before displaying the next.

Hex Codes → Used for defining color on the computer. Starts with `#` then 3 pairs (RGB) of 6 hexadecimal numbers.

Color	Hex Code	RGB Values
Red	`#FF0000`	$(255, 0, 0)$
Green	`#00FF00`	$(0, 255, 0)$
Blue	`#0000FF`	$(0, 0, 255)$
Yellow	`#FFFF00`	$(255, 255, 0)$
Magenta	`#FF00FF`	$(255, 0, 255)$
Cyan	`#00FFFF`	$(0, 0, 255)$
White	`#FFFFFF`	$(255, 255, 255)$
Black	`#000000`	$(0, 0, 0)$

3.2 Physics of Color

Our human eye has a wall at the back of it called the "retina". This retina has millions of photosensitive cells called *rods* and *cones*. Rods are sensitive to Light while cones come in three variants: Red cone, Blue cone, Green cone. Cones are sensitive to their respective Color.

Our eyes have around 20-30 times more Rods compared to Cones which makes our eyes more sensitive to changes in lightness compared to changes in color.

3.3. Color Representation

The most common way to represent color is the **Additive** RGB (Red, Green, Blue) color model, used for displays, and the **Subtractive** CMYK (Cyan, Magenta, Yellow, Key/Black) color model, used for printing.

Color can also be defined using the HSL/HSV model where

- **Hue** → what color it is.
- **Saturation** → how much color there is.
- **Value**/**Lightness** → how much lightness/darkness there is.
- **Tint** → Color mixed with *white*.
- **Tone** → Color mixed with *gray*.
- **Shade** → Color mixed with *black*.

Then there're also these color models (not important)

1. YCrCb → Luma, Red-difference Chroma, Blue-difference Chroma
2. YIQ → Luma, Orange-Blue Chroma, Purple-Green Chroma
3. YUV → Luma, Blue-luminance difference, Red-luminance difference

and CIE Lab → Used for color management and is device independent.

---

4. Images

Image Resolution is the number of pixels in an image. It is written like `horizontal pixels x vertical pixels`.

Frame Buffer → Used to store bitmaps.
Color Histograms → A graph that show color distribution where whites/light are on the left, blacks/dark on the right and grays in the middle.

Storage

1. Bitmap → 1 bit per pixel.
2. 8-bit Grayscale → 8 bits per pixel.
3. 8-bit Color → 8 bits per pixel. Done using a CLUT (Color Lookup Table).
4. True Color → 8 bit per color (8 for Red, 8 for Green, 8 for Blue)
5. True Color with Alpha → Extra 8 bits for Transparency.

Image File Formats

1. JPEG - Joint Photographic Experts Group
2. PNG - Portable Network Graphics
3. GIF - Graphics Interchange Format → LZW Encoding
4. DCT - Discrete Cosine Transform
5. TIFF - Tagged Image File Format
6. EXIF - Exchange Image File Format
7. Animation Formats (FLC, GL, Apple QuickTime etc)
8. Postscript and PDF
9. BMP, WMP, PSD etc

- **Spatial Resolution:** The density of pixels per inch. Measured in ppi (pixels per inch) for monitors and dpi (dots per inch) for printers.
- **Color Resolution:** The number of colors each pixel can display, determined by bit depth.
- **Device Dependence:** Image dimensions depend on the resolution of the output device. Bitmapped images are device-dependent.
- **Vector Graphics:** Created from mathematically defined shapes, allowing for smooth scaling without distortion.

- **Image Size:** A 512×512 grayscale image takes up 1/4 MB, a 512×512 24-bit image takes 3/4 MB with no compression. Overhead increases with image size. Modern high digital cameras (10+ Megapixels) produce approximately 29MB uncompressed images. Compression is commonly applied.

4.1. Image Artifacts

These are unwanted abnormalities/distortions that appear due to issues.

1. Compression Artifacts → Occurs due to lossy compression.
2. Motion Blur → Bluriness that shows when the subject/camera moves.
3. Chromatic Aberration → Issues with the lens of the camera causes color fringing.
4. Color Banding → Less bit-depth causes bands of color to appear in gradients.
5. Aliasing → Low-resolution causes jagged/stair-stepped edges.

4.2. Camera Attributes

Camera Resolution is measured in Megapixels.

1. **ISO** → sensitivity of the camera's sensor to light.
2. **Aperture** → opening inside the lens that controls how much light enters the camera.
3. **Shutter Speed** → how long the camera's shutter stays open, controlling the amount of time light hits the sensor.

---

5. Audio

5.1. Sound

Sound Pressure is measured in **decibels** (dB) and Sound Frequency in **hertz** (Hz).
It is used for either *content sounds* (for information) or *ambient sounds* (for mood).

CD Quality Audio: Requires 16-bit sampling at 44.1 kHz. Higher rates exist (e.g., 24-bit, 96 kHz) for audiophiles. Audio can be Mono (Single Channel) vs Stereo (Double Channel) and even Surround (5+1 or 7+1).

5.2. Physics of Audio

5.2.1. Human Hearing

Sound begins as air pressure waves that our ears transform into perceived sound through a three-stage process

1. The Outer Ear channels sound waves to the eardrum, which vibrates in response.
2. The Middle Ear contains three small bones (malleus, incus, and stapes) that amplify these vibrations.
3. The Inner Ear transforms these mechanical vibrations into nerve signals through two key components:
    - The cochlea: A fluid-filled chamber where pressure waves travel. Its basilar membrane contains over 20,000 stereocilia (hair-like nerve cells) that respond to different frequencies based on their position and properties.
    - The auditory nerve: Carries electrical signals from stimulated stereocilia to the brain for interpretation.

5.2.2. Hearing Capabilities

The human auditory system has specific ranges and sensitivities:

- Frequency Range: 20 Hz to 20 kHz (20,000 Hz), with peak sensitivity at 2-4 kHz
- Speech Range: 500 Hz to 2 kHz (vowels and bass in lower range, consonants in higher range)
- Volume Range: About 96 dB from quietest to loudest perceivable sounds
- Common Sound Levels:
    - Normal conversation: 60-70 dB
    - Classroom background: 20-30 dB
    - Pain threshold: 120 dB (can cause permanent damage)

The Fletcher-Munson curves demonstrate that our ears are most sensitive to frequencies around 3-4 kHz, which aligns with the frequency range most important for

speech perception.

5.3. Psycho Acoustic Phenomena

> Lossy audio compression algorithms, like MP3, achieve significant file size reductions by exploiting psychoacoustic phenomena – characteristics of human hearing that allow certain information to be discarded without a noticeable loss in perceived quality.

5.3.1. Frequency Masking

> Occurs when a louder sound at one frequency makes it difficult to hear a quieter sound at a nearby frequency.

A lower tone can effectively mask a higher tone played simultaneously, but the reverse is not true. The stronger the masking tone, the wider the range of frequencies it masks.

- **Critical Bands**: Ranges of frequencies within which masking occurs. It varies with frequency: Constant 100 Hz for frequencies below 500 Hz. Increases linearly by about 100 Hz for each additional 500 Hz.
- **Cause**: When one frequency excites a group of stereocilia, it becomes difficult for a weaker, similar frequency to further excite those same cells.

5.3.2. Temporal Masking

> Occurs when a loud sound makes it temporarily difficult to hear quieter sounds that occur shortly before or after it.

- **Cause**: Loud sounds saturate the hearing receptors in the inner ear, requiring time to recover. Strong stimuli fatigue the stereocilia, leading to temporary hearing loss or ringing in the ears.

5.3.3. Reverb

> The persistence of sound in a space after the original sound has stopped.

It's created by multiple reflections of sound waves off surfaces within an enclosed space.When these waves encounter surfaces, they reflect, and the combination of these reflections creates the effect we perceive as reverb.

5.3.4. Critical Band

> A range of frequencies within which sounds can mask each other. Sounds within a critical band are more likely to mask each other than sounds outside that band.

The width of the critical band varies with frequency, remaining relatively constant at lower frequencies (below 500 Hz) and increasing at higher frequencies. This means that the human ear is more sensitive to frequency differences at lower frequencies and less sensitive at higher frequencies.

Lossy audio compression algorithms exploit this by allocating more bits to represent sounds in critical bands where the ear is more sensitive and fewer bits to represent sounds outside of critical bands, where masking is likely to occur.

5.3.5. Echo

> A distinct, delayed repetition of a sound, typically heard when the sound wave reflects off a distant surface and returns to the listener.

It's a special case of "reverb" where the reflected sound is perceived separately from the original sound due to the longer delay.

5.4. Digitizing Sound

Digital Sampling → A microphone receives sound and converts it to an analog signal. Computers work with discrete entities, therefore analog signals need to be converted to digital using dedicated hardware like a sound card.

Bit Size (Quantisation) → Determines how each sample value is stored.

- *8-bit value*: 0 to 255.
- *16-bit value*: 0 to 65535.

5.4. Nyquist Sampling Theorem

> The **sampling rate must be at least twice the highest frequency component** of the signal, a value known as the **Nyquist rate**.

Sampling below this rate causes **aliasing**, where high-frequency components are misrepresented as lower frequencies, distorting the signal and preventing accurate reconstruction.

Preventing Aliasing → *Low-pass filters* remove frequencies above the Nyquist limit before sampling, ensuring accurate digital representation of the original signal.

Trade-Offs in Sampling Rate → Higher sampling rates improve audio quality but increase file size. Choosing the right rate involves balancing quality with storage and bandwidth limitations.

5.5. Audio File Types

1. **Uncompressed**

    1. *WAV* (Waveform Audio File Format) → Default for professionals.
    2. *AIFF* (Audio Interchange File Format) → Apple's alternative to WAV.
2. **Lossless**
    1. *FLAC* (Free Lossless Audio Codec) → Open Source .
    2. *ALAC* (Apple Lossless Audio Codec) → Apple's alternative to FLAC.
3. **Lossy**
    1. *MP3* (MPEG-1 Audio Layer 3) → Most Popular.
    2. *AAC* (Advanced Audio Codec) → Better than MP3.
    3. *OGG* (Ogg Vorbis) → Open Source.
    4. *WMA* (Windows Media Audio) → Microsoft's alternative to MP3.

5.6. Synthetic Sounds

Sounds are synthesized using hardware or software. Instead of sending the entire sound, only control parameters are sent to the client to produce the sound (using formats like MIDI/MP4/HTML5).

Synthesis Methods → FM (Frequency Modulation) Synthesis, Wavetable synthesis, Additive synthesis, Subtractive synthesis, Granular Synthesis, Physical Modelling, Sample-based synthesis.

---

6. Video

6.1. Video Concepts

6.1.1. PAL (Phase Alternate Line)

> A color encoding system used for analog TV and one of the main broadcast video standards, alongside NTSC (National Television Standards Committee) and SECAM (Sequential Color and Memory).

- **Scan lines:** PAL uses 625 scan lines to create a television picture, compared to 525 lines in NTSC.
- **Frame rate:** PAL has a frame rate of 25 frames per second.
- **Interlacing:** Like NTSC, PAL uses interlaced scanning, meaning it displays odd-numbered lines in one pass (field) and even-numbered lines in the next.
- **Modulation:** PAL employs amplitude modulation to encode color information.

6.1.2. Progressive Scan

> A method of displaying video where all the lines of each frame are drawn in sequence.

This is different from **interlaced scan**, where only half the lines are displayed at a time, creating the "combing" artifacts often seen in older television broadcasts.

Progressive Scan gives *Smoother motion* with *Reduced artifacts*.

6.2. Video Formats and Codecs

1. **Video Format** → The container that holds the video data.
    1. *MP4* (MPEG-4 Part 14) → Most Popular.
    2. *MKV* → Most Flexible, Supports multiple audio, video, and subtitle tracks.
    3. *AVI* (Audio Video Interleave) → Older format with less efficient compression.
    4. *MOV* (Quicktime File Format) → Used by Apple for QuickTime videos.
    5. *WEBM* → Open Source container optimized for web streaming (uses VP9 or AV1).
2. **Codec** (compressor-decompressor) → Dictates how the video is compressed and decompressed.
    1. *H.264* (AVC) → Most Popular with high compression efficiency.
    2. *H.265* (HEVC) → Better (but slower) compression when compared to H.264.
    3. *VP9* → Open Source, Used by YouTube.
    4. *AV1* → Open Source, High compression efficiency.
    5. *MPEG-2* → Older codec used in DVDs.

3. **File Type** → The extension at the end of a video file. Examples: `.mp4`, `.mkv`, `.avi` etc.

6.2.1. H.261

Made for video telecommunication applications and uses CIF and QCIF resolutions with 4:2:0 chroma subsampling. It has two frame types:

- **Intraframes (I-frames)**: Coded independently (like JPEG) and provide refresh points.
- **Interframes (P-frames)**: Coded using differences from the previous frame (predicted).

Feature	Intra-frame Coding	Inter-frame Coding
How it works	Compresses each frame individually, like a standalone image.	Compresses by comparing a frame with the previous one.
Method	Similar to JPEG, using color separation (YUV) and DCT blocks.	Uses motion prediction to find changes between frames.
Focus	Deals with one frame at a time.	Focuses on differences between consecutive frames.
What it stores	Entire frame information.	Only stores changes (motion and residual errors).
Key Idea	Compressing the image itself.	Predicting and encoding motion between frames.
Use Case	Useful for keyframes (starting points for decoding).	Efficient for saving space by reusing data from earlier frames.

6.3. Motion

6.3.1. Motion Vector Search

Identifies how blocks of pixels (macroblocks) move between frames. If no good match is found, the macroblock is encoded as an **intra macroblock**.

Block Matching → Finds the best match for a macroblock within a search area in the previous frame.

- **Search Methods** → *Full Search* (Search everything), *Logarithmic Search* (Less calculations), *Hierarchical Motion Estimation* (Starts with lower resolution frames).
- Matching metrics like **Sum of Absolute Differences (SAD)** or **Mean Absolute Error (MAE)** measure block similarity.

6.3.2. Motion Estimation and Compensation

Motion Estimation → Calculates the movement of blocks between frames, represented as *motion vectors*.
Motion Compensation → Predicts the current frame's content using *motion vectors* and previous (or future) frames.

> The main idea behind motion compensation is to **predict the content of a frame based on a previous or future frame, encoding only the difference (residual) between them**.

This is a core technique in standards like MPEG-2 where techniques like Block Matching and Motion Vector Search are used.

---

7. Sampling

How Sampling Affects Quality and Causes Artifacts in Different Media Types

1. Audio Sampling
    1. **Resolution/Bit-depth** → Number of bits used to represent each sound sample. The higher the bit depth, the more accurately the sound wave can be quantized, resulting in better sound quality and a wider dynamic range.
    2. **Sampling Rate** → Determines how many times per second the sound wave is measured. A higher sampling rate captures more details of the sound wave, leading to better audio quality and a wider frequency range that can be accurately represented.
    3. **Sampling Artifacts** → When the sampling rate is too low, **aliasing** occurs. Aliasing introduces unwanted frequencies that were not present in the original sound. This can manifest as distorted or "metallic" sounds in the audio recording.
2. Graphics Sampling
    1. **Resolution:** For vector graphics, resolution isn't as critical as it is for bitmaps. This is because vector graphics are defined mathematically by shapes and paths, not by pixels. As a result, vector graphics can be scaled without losing quality.
    2. **Sampling Artifacts:** While vector graphics themselves don't suffer from traditional sampling artifacts like aliasing, artifacts can be introduced during the **rasterization** process.
        1. Rasterization converts the vector image into a bitmap for display or printing. If the resolution of the rasterized image is too low, jagged edges or pixelation may become apparent, particularly for curved lines or complex shapes.
3. Image Sampling
    1. **Resolution** → Number of pixels in an image, typically expressed as width x height (e.g., 1920 × 1080). Higher resolution images have more pixels, capturing finer details and resulting in sharper images. However, higher resolution also leads to larger file sizes.
    2. **Bit Depth** → Number of bits used to represent each pixel's color information. A higher bit depth allows for a wider range of colors, resulting in smoother color transitions and reduced **color banding**, which appears as noticeable steps in color gradients.
4. Video Sampling
    1. **Frame Rate** → Video frame rate refers to the number of individual images (frames) displayed per second, measured in frames per second (fps). Higher frame rates result in smoother motion but also increase the file size.

2. **Resolution** → Video resolution, like images, dictates the number of pixels in each frame. A higher resolution video displays more detail but requires more storage and bandwidth.
3. **Bit Depth** → Similar to images, video bit depth determines the number of colors each pixel can display. Higher bit depths lead to better color accuracy and smoother gradients but increase file sizes.

Sampling Artifacts:

- **Motion Jerkiness:** A low frame rate can make motion appear jerky, especially for fast-moving objects.
- **Aliasing:** Low resolution can cause aliasing, leading to jagged edges or "blockiness" in the video, especially around moving objects.
- **Color Banding:** Insufficient bit depth can cause color banding, where noticeable steps in color appear instead of smooth gradients.
- **Compression Artifacts:** Additionally, lossy video compression techniques, such as those used in MPEG, can introduce artifacts like blockiness, blurring, or "mosquito noise" around edges, depending on the compression level and codec used.

---

8. Compression

8.1. Text Compression

Text is that one medium where Lossy compression doesn't work. It **must** always be Lossless otherwise we might lose important information.

8.1.1. Data Compression

Zip, **Rar** and **7zip** are three of the most well known file compression formats. *Zip* is universal in nature and is supported on every platform natively.

Linux has the *gzip*, *bzip2*, *xz* and *tar* formats.

8.1.1.1. Huffman Encoding

A lossless compression, entropy coding algorithm which is used in JPEG, MP3 and ZIP.

It uses "variable-length" codes to represent input symbols based on their frequency which means that symbols with higher frequency get shorter codes, while rare symbols get longer codes.

Example → We have this string: "AABBBCCCCDDDD"

1. Calculate frequency of each character → A: 2, B: 3, C: 4, D: 4.
2. Create a frequency queue → (A,2), (B,3), (C,4), (D,4).
3. Build the Huffman Tree
    1. Take the two nodes with lowest frequency → (A,2), (B,3) and Create a new node with their sum (5), with A and B as its children.
    2. Now we have → (5), (C,4), (D,4)
    3. Take the two lowest again → (C,4), (D,4) and Create a new node with their sum (8), with C and D as its children.
    4. Finally, combine (5) and (8) → (13).
4. Now, Assign codes by traversing the tree, assigning 0 for left branches and 1 for right branches → A: `00`, B: `01`, C: `10`, D: `11`.
5. Encode the string
   "AABBBCCCCDDDD" becomes `00 00 01 01 01 10 10 10 10 11 11 11 11`

The encoded bitstring is 26 bits long, compared to the original 96 bits.

8.1.1.2. LZW Encoding

LZW(Lempel-Ziv-Welch) is a dictionary-based encoding algorithm that builds a dictionary of sequences as it processes the input.

Example → We have the string: "AABABBABCABABBA"

1. Initialize the dictionary, Start with single characters → 1: A, 2: B, 3: C
2. Encode the string

Step	Current	Next	Output	Add to Dictionary
1	A	A	1	4: AA
2	A	B	1	5: AB
3	B	A	2	6: BA
4	AB	B	5	7: ABB
5	B	A	2	(BA already in)
6	BA	B	6	8: BAB
7	C	A	3	9: CA
8	AB	A	5	10: ABA
9	AB	B	5	(ABB already in)
10	BA	-	6	-

Final encoded output → `1, 1, 2, 5, 2, 6, 3, 5, 5, 6`

This sequence of numbers represents the compressed version of the original string. To decompress, we would use these numbers to rebuild the dictionary and reconstruct the original string.

8.2. Image Compression

8.2.1. JPEG Compression

The JPEG file format uses a lossy compression technique which removes data and details from an image, that is unnoticeable to the human eyes.

JPEG compression reduces file size by:

1. Converting colors to **YCbCr** format.
2. Using **chroma subsampling** to save less color data.
3. Transforming pixel values into **frequencies** (DCT).
4. **Quantizing** the data by rounding off unimportant values.
5. Compressing with **entropy encoding**.

1. Color Space Conversion (RGB to YCbCr)

- **Why?**: Human eyes are more sensitive to brightness (**luminance, Y**) than color differences (**chrominance, Cb and Cr**).
- **How?**: The image's RGB colors are transformed into **YCbCr** format:
  - **Y** = Luminance (brightness).
  - **Cb** = Blue-difference chrominance.
  - **Cr** = Red-difference chrominance.

2. Chroma Subsampling (Reducing Color Data)

- **Why?**: Our eyes don't notice small changes in color as much as they notice changes in brightness. JPEG takes advantage of this by **storing fewer color details** to save space.
- **How?**: Chroma subsampling reduces the resolution of Cb and Cr channels (by creating 2×2 blocks and taking the average):
  - **4:4:4**: No subsampling (full quality for Y, Cb, and Cr).
  - **4:2:2**: Cb and Cr are halved horizontally.
  - **4:2:0**: Cb and Cr are halved both **horizontally and vertically**.

This means **fewer color values** are saved, while the brightness stays sharp.

3. Discrete Cosine Transform (DCT)

The image is split into **8×8 pixel blocks** for easier processing with values from 0-255. They are then shifted by subtracting 128 from each value making the range from -127 to 128. Each block will undergo compression separately.

- **How?**: converts the pixel values in each 8×8 block into **frequency values** (using the DCT formula):
  - **Low frequencies** = General shapes and smooth gradients.
  - **High frequencies** = Fine details (like edges or noise).

4. Quantization (Simplifying Data)

Less important frequencies (especially high ones) are rounded off to reduce precision.

- **How?**: A Quantization Table is used and every value in it divides a corresponding value in the block.
  - **Lower numbers** (more rounding) = **Smaller file, more quality loss**.
  - **Higher numbers** (less rounding) = **Better quality, larger file**.

5. Entropy Encoding (Further Compression)

The quantized data is compressed using:

- **Zig-Zag Scan** → The 8×8 block is scanned in a zig-zag pattern as this makes it more likely for larger strings of 0s to occur.
- **Huffman Encoding** to assign shorter codes to frequently used values. MSD (Mids) - Jaish Khan > Huffman Encoding

Run-Length Encoding (RLE)

Used for AC Coefficients. To reduce repeating patterns, Instead of listing every 0 we just list how many there are in a sequence.

Example → We got coefficients (after zig-zag scan) in this order

```
[12, 5, 0, 0, 0, -2, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Applying RLE converts it into:

```
(0, 12), (0, 5), (3, -2), (1, 1), (15, 3), EOB
```

- (0, 12): No zeros before 12.
- (0, 5): No zeros before 5.
- (3, -2): Three zeros before -2.
- (1, 1): One zero before 1.
- (15, 3): Fifteen zeros before the last non-zero value 3.
- EOB: End of Block. Denotes that the remaining values are zeroes.

Discrete Pulse Code Modulation (DPCM)

Used for DC Coefficients. DPCM is a signal encoding technique for data compression.

Instead of encoding absolute values of each sample (as in PCM), DPCM encodes the difference between the current sample and a predicted value.

Example → We have a simple audio signal: [100, 105, 111, 114, 116]

1. First number (100) is encoded as is.
2. For next numbers, we encode the difference:
    - 105 - 100 = 5
    - 111 - 105 = 6
    - 114 - 111 = 3
    - 116 - 114 = 2

So our DPCM encoding would be: [100, 5, 6, 3, 2]

Image Reconstruction

When you open the JPEG file, the reverse process occurs:

- Decompressed frequency values are used to reconstruct the image.
- Some details (like colors and fine edges) might be lost, especially if the file was highly compressed.

8.2.2. PNG Compression (Not Important)

PNG is a lossless image compression format that uses a combination of filtering and DEFLATE compression.

It supports true color, grayscale, and palette-based images as well as alpha channel support for transparency and gamma correction for cross-platform color consistency.

1. Pixels to Index Stream

Pixels are mapped to an index stream. Each row is called a **scanline**.

2. Filtering

PNG applies a filter to each scanline of the image. This process is called "filtering" and it helps to make the image data more compressible.

There are five filter types:
0. None: No filtering

1. Sub: Subtract the value of the pixel to the left
2. Up: Subtract the value of the pixel above
3. Average: Use the average of the left and upper pixel
4. Paeth: A special adaptive filter

The filter type that produces the smallest output is chosen for each scanline.

Example Original: [20, 30, 40, 30, 20] → Filtered: [20, 10, 10, -10, -10]

The filtered data often has smaller values and more repetition, making it more compressible.

3. DEFLATE Compression

After filtering, PNG uses DEFLATE compression, which is a combination of LZ77 and Huffman coding.

1. LZ77:
    - Looks for repeated sequences in the data

- Replaces repeated sequences with references to previous occurrences

2. Huffman coding:
   - Assigns shorter codes to more frequent symbols
   - Further compresses the output from LZ77

8.3. Audio Compression

Simple Audio Compression Methods

- **Silence compression**: Detects and encodes silence, a form of run-length encoding.
- **Differential Pulse Code Modulation (DPCM)**: Encodes the small differences in amplitude between successive samples using fewer bits.
- **Adaptive Differential Pulse Code Modulation (ADPCM)**: Encodes the difference between samples using adaptive quantization.

8.3.1. MPEG Audio Compression Steps

> MPEG audio compression removes irrelevant parts of the audio signal by taking advantage of masking. It exploits the fact that the human auditory system cannot perceive quantization noise under masking conditions. Frequency masking is always used in MPEG, while more complex forms also utilize temporal masking.

1. **PCM Sampling and Quantization**
   The algorithm begins with raw audio input, typically in PCM (Pulse Code Modulation) format. The audio signal is divided into overlapping time-domain segments.
2. **Filter Bank Analysis**

- *Sub-Band Filtering* → The audio signal is passed through a polyphase filter bank to split it into multiple frequency sub-bands (32 sub-bands in MPEG Layer I/II).
- *Frequency Domain Representation* → This divides the signal into components, enabling separate analysis of high- and low-frequency content.

3. **Psychoacoustic Modeling**
   A psychoacoustic model estimates which parts of the audio signal are perceptible to the human ear.

- *Signal-to-Mask Ratio* (SMR): Determines the level of quantization noise that can be introduced without being perceptible.

4. **Quantization**
   Bit Allocation → Based on psychoacoustic analysis, more bits are allocated to critical audio components (important sub-bands) and fewer to less critical ones. The amplitude of the signal is quantized to reduce the amount of data.

5. **Encoding**
   The encoded output consists of:

- Header (sample rate, bitrate, and other coding parameters)
- Subband Sample Data (quantized scaling factors and frequency component values for each subband).

6. **Decoding**
   The decoder reverses the encoding process.

- Demultiplexes the bitstream into subbands.
- Dequantizes the subband samples using the information provided in the bitstream.
- Synthesizes the audio signal using inverse filtering and channel multiplexing.

7. **Stereo Redundancy Coding** (Optional)
   For stereo audio, additional compression can be achieved by exploiting the redundancy between the two channels.

8.3.2. MPEG Layers

MPEG defines three layers of audio processing:

- **Layer 1**: Basic mode, suitable for bitrates above 128 kbits/sec per channel, uses frequency masking.
- **Layer 2**: Enhanced version of Layer 1, uses temporal masking to some extent, targets bitrates around 128 kbits/sec per channel.
- **Layer 3**: Most common form for web audio files (MP3), includes temporal masking, utilizes modified DCT (MDCT), and Huffman coding for greater compression, targets bitrates around 64 kbits/sec per channel.

8.3.3. Dolby Audio Compression

A proprietary audio compression algorithm.

- **Differences from MPEG** → MPEG controls quantization accuracy by computing the number of bits per sample (forward adaptive bit allocation). Dolby uses fixed bit rate allocation per subband based on the ear's characteristics (backward adaptive bit allocation).
- **Versions** → Dolby AC-1, Dolby AC-2, Dolby AC-3.

8.4. Video Compression

Video compression is essential due to the massive size of uncompressed video data.

- Lossy compression methods are necessary because lossless methods do not achieve sufficient compression ratios.

MPEG Standards

- *MPEG-1* (1991): Targeted VHS quality video on CD-ROM (320×240 resolution at 1.5 Mbits/sec).
- *MPEG-2* (1994): Designed for television broadcasting and DVD.
- *MPEG-3*: Initially intended for HDTV but merged into MPEG-2.
- *MPEG-4* (1998): Focused on very low bitrate coding and later included H.264 (MPEG-4 Part 10 or AVC) for a wider range of bitrates and better compression.
- *MPEG-7* (2001): Defines a "Multimedia Content Description Interface" for metadata.
- *MPEG-21* (2002): Aims to create a "Multimedia Framework."

8.4.1. Frame Types

1. **I-Frames** (Intra-coded frames) → They are coded independently of other frames, and only rely solely on spatial compression within the frame itself, similar to JPEG compression.
    - They act as reference points for other frame types, providing random access points within the video stream.
2. **P-Frames** (Predictive-coded frames) → They utilize temporal redundancy by encoding the difference between the current frame and the previous frame.
    - They rely on *motion estimation*, which identifies blocks of pixels that have moved between frames, represented by motion vectors.
3. **B-Frames** (Bi-directionally Predictive-coded frames) → They take a step further in exploiting temporal redundancy by referencing both previous and next frames for prediction.
    - They use *bidirectional prediction*, analyzing motion in both directions to achieve higher compression efficiency compared to P-frames.
    - Due to their dependence on future frames, B-frames need to be encoded out of order, requiring additional buffering and processing.
    - **Advantages** → More efficient compression, increased quality for moving objects with less error.
    - **Disadvantages** → Requires more memory, Encoding is more complex with increased delay.

8.4.1. MPEG-1 Video Compression

The MPEG-1 algorithm achieves compression by exploiting the inherent redundancies present in video data

- **Spatial Redundancy** → Within a single frame, neighboring pixels often have similar values. This similarity is exploited using techniques like Discrete Cosine Transform (DCT).
- **Temporal Redundancy** → Consecutive frames in a video sequence tend to be very similar. MPEG-1 uses this temporal correlation to predict future frames based on past frames.

Typical Frame Sequence → A common frame sequence in MPEG-1 is: IBBPBBPBB IBBPBBPBB... This pattern interleaves I-frames, P-frames, and B-frames, striking a balance between compression efficiency, random access, and error resilience.

Steps in the MPEG-1 Video Encoding Process

1. Color Space Conversion.
2. Chroma Subsampling.
3. Macroblock Partitioning.
4. Motion Estimation and Compensation.
5. Discrete Cosine Transform (DCT).
6. Quantization.
7. Zigzag Scanning and Run-Length Encoding.
8. Entropy Coding.

> **MPEG-1 Decoding** → The decoding process essentially reverses the encoding steps, reconstructing the video frame from the compressed data.

8.4.2. MPEG-2 Video Compression

Enhancements of MPEG-2

- Search on fields instead of just frames.
- Support for 4:2:2 and 4:4:4 chroma subsampling.
- Larger frame sizes.
- Scalable modes (temporal, progressive, etc.).
- Non-linear macroblock quantization factor.

---

9. Immersive Reality

Immersive reality is an umbrella term that includes technologies designed to merge the digital and physical worlds, creating interactive and engaging experiences.

Two primary forms of immersive reality are **Virtual Reality (VR)** and **Augmented Reality (AR)**, which differ in how they interact with the user's environment.

9.1. Virtual Reality (VR): Entering a Digital World

VR completely replaces the real world with a digital environment (**Complete Immersion**), immersing users in a 3D, interactive experience (**Interactive Elements**). Users can look around the virtual environment by moving their heads (**Head-tracking**).

- **Applications** → Gaming (Beat Saber), Simulations, Virtual tours, Therapy, Social Interactions.

9.2. Augmented Reality (AR): Enhancing the Real World

AR overlays digital elements onto the real world (**Partial Immersion**), blending physical and digital realities. Digital content is superimposed onto the user's view of the real world (**Real World Integration**). AR can provide relevant information about the user's surroundings, such as product details, directions, or historical facts (**Contextual Information**).

- **Applications** → Gaming (Pokémon GO), Navigation tools (AR maps), Retail, Education (Anatomy).

9.3 Mixed Reality

An environment that merges the real and virtual worlds, allowing physical and digital objects to coexist and interact in real time. Unlike Virtual Reality (VR), which creates completely immersive virtual environments, Mixed Reality blends virtual elements into the user's perception of the real world.

9.4. VRML (Virtual Reality Modeling Language)

A platform-independent language for creating 3D environments, initially designed for web display. Its objective is to enable the placement of colored objects in a 3D space.

---

10. Multimedia Skills, Authoring and Tools

> The process of creating multimedia productions, often interactive applications.

10.1. Multimedia Skills

A diverse team is usually needed to develop multimedia projects. Members may have specialized roles, or they may wear multiple hats, contributing to various aspects of the project.

1. **Project Manager** → The main person of a multimedia project, overseeing the entire process from conception to completion. They are responsible for the smooth execution of the project and ensuring that all elements come together cohesively. Their duties include Managing budgets and schedules, Facilitating creative sessions, Handling unexpected situations and Fostering team dynamics.
2. **Multimedia Programmer** → Plays a pivotal role in bringing all the multimedia elements together into a functional and engaging product.
   They use authoring systems or programming languages to integrate text, graphics, audio, video, and interactive components into a seamless whole.
3. **Video Specialist** → Responsible for all aspects of video production, ensuring that the visual elements of the multimedia project are high-quality and engaging.
   Duties:
   - *Shooting video footage* → Capturing high-quality visuals using professional video cameras.
   - *Transferring footage to a computer* → Importing the recorded video into a digital editing environment.
   - *Editing footage* → Using non-linear editing software to refine the video, incorporating transitions, special effects, and other enhancements.
4. **Audio Specialist** → Responsible for designing and producing audio elements that enhance the user experience. Their duties include:
   - *Creating and selecting music* → Composing original music or finding suitable existing tracks to set the tone and atmosphere.
   - *Producing voice-over narrations* → Recording and editing professional voice-overs to provide commentary or guide the user.
   - *Designing sound effects* → Incorporating realistic or stylized sound effects to create a more immersive and engaging experience.
   - *Digitizing and editing audio* → Converting analog audio sources to digital formats and refining them using audio editing software.
5. **Interface Designers** → They focus on creating intuitive and user-friendly navigation pathways, ensuring that users can easily interact with the multimedia content.

6. **Writers** → They craft compelling narratives, script voice-overs, and write concise and informative text screens, ensuring the clarity and effectiveness of the project's message.

10.2. Multimedia Authoring

> The creation of multimedia productions, often referred to as "movies" or "presentations."

Key Considerations in Multimedia Authoring → **Graphics Styles**, **Sprite Animation**, **Video Transitions** and **Technical Design Issues**.

Automatic Authoring → Capture of media, Authoring, Publication.

10.2.2. Multimedia Authoring Metaphors

1. **Scripting Language Metaphor**: Uses specialized languages for interactivity and control flow.
2. **Slide Show Metaphor**: Sequentially presents media in a linear format.
3. **Hierarchical Metaphor**: Organizes elements into a tree structure for user navigation.
4. **Iconic/Flow-Control Metaphor**: Uses graphical icons and flowcharts to depict actions and control.
5. **Frames Metaphor**: Links conceptual elements within a flow-control structure.
6. **Card/Scripting Metaphor**: Utilizes index cards for media representation and hyperlinking, common in education.
7. **Cast/Score/Scripting Metaphor**: Combines timelines and event-driven scripts for interactive, dynamic content (e.g., Macromedia Director).

10.3. Multimedia Tools

1. **Adobe Suite**
    1. Adobe Photoshop (Photo Editing and Manipulation)
    2. Adobe Lightroom (Photo Editing and Color Correction)
    3. Adobe Illustrator (Vector Graphics)
    4. Adobe InDesign (Book Design)
    5. Adobe Premiere Pro (Video Editing)
    6. Adobe After Effects (Motion Graphics)
    7. Adobe Animate (Animations)
    8. Adobe Audition (Audio Editing)
    9. Adobe Dreamweaver (Websites)
2. **Open-Source Replacements**
    1. Gimp (Photoshop)

2. Darktable (Lightroom)
3. Inkscape (Illustrator)
4. Kden Live (Premiere)
5. Lightworks (Animate)
6. Audacity (Audition)

3. **Others**
    1. Figma (Designing)
    2. Canva (Multimedia Creation)
    3. Webflow/Framer (Websites)

---

11. Multimedia Delivery

11.1. Challenges of Multimedia Delivery

1. **Large Data Requirements and Bandwidth Constraints**
   Multimedia data tends to be very large. Uncompressed high-definition video can easily exceed 1 Gbps in bitrate, posing significant challenges for storage and transmission over networks.

This immense data size requires the use of compression to reduce the amount of data that needs to be stored and transmitted. *Lossy Compression* (JPEG, MPEG etc.) and *Lossless Compression* (PNG, LZW etc.) are two ways of doing that.

2. **Real-Time Delivery and the Need for Processing Power**
   Often time "real-time delivery of content" is required, especially for streaming where users expect smooth playback of audio and video without interruptions.

This real-time constraint requires high processing power to handle the large data volumes and perform tasks like decompression, synchronization, and rendering. For that *Specialized Hardware* (like GPUs) are used.

3. **Storage and Memory Considerations**
   The large data sizes require "large storage units", potentially on the order of hundreds of terabytes or more. *Sufficient memory* and *Large caches* are also very important for efficient multimedia processing, ensuring that the system can readily access the data it needs.
4. **Synchronization**
   Ensuring that audio and video streams are synchronized is important for a proper viewing experience. Lip synchronization in video, for example, is highly sensitive to timing errors, and even slight errors can be distracting.
5. **Data Formats and Compatibility**
   The amount of different multimedia file formats can lead to compatibility issues, requiring format conversions or the use of specific codecs to ensure that content can be played back on different devices and platforms.

11.2. Content Distribution Networks (CDN)

> ⓘ **Example**
>
> Imagine you're watching a live stream of a major sporting event. Millions of viewers around the globe are trying to access the same video stream from a single origin server. This can put a tremendous strain on the server and the network, leading to buffering, lag, and a poor viewing experience.

> CDNs are made to solve this challenge by distributing copies of the content to multiple servers located in various geographic locations. When a user requests the content, the CDN directs them to the server closest to their location, reducing latency and improving the streaming quality.

Key Benefits of CDNs → **Reduced Latency**, **Improved Scalability**, **Enhanced Reliability**.

10.3. Quality of Service (QoS)

The overall performance of a network in terms of its ability to deliver a satisfactory user experience. For multimedia streaming, QoS is crucial because it directly impacts factors like video playback smoothness, audio fidelity, and interactive responsiveness.

Important QoS Parameters:

1. **Latency** → Time it takes for data to travel from the source to the destination.
2. **Jitter** → Variation in delay.
3. **Packet Loss** → Occurs when data packets get lost during transmission.
4. **Bandwidth** → Capacity of the network connection.

Strategies for Managing QoS → **Prioritization**, **Traffic Shaping**, **Buffering**.

10.4. Multimedia Delivery Concepts

Net Neutrality → The principle that internet service providers should treat all data on the internet equally, without discriminating or charging differentially by user, content, website, platform, application, type of attached equipment, or mode of communication. It ensures a free and open internet where users can access content without interference based on source or type.

RSVP (Resource ReSerVation Protocol) → A network protocol that enables applications to request and reserve resources like bandwidth along a network path. This is often used for real-time applications like streaming media to ensure sufficient network capacity for smooth playback.

RSTP (Rapid Spanning Tree Protocol) → A network protocol used to prevent loops in Ethernet networks, ensuring that data can flow efficiently.