#### **IMAGE COMPRESSION**

#### Image compression is a critical process in digital image processing that aims to reduce the size of an image file without significantly degrading its quality. The compression system comprises two primary structural components: the encoder and the decoder. These components work together to efficiently store, transmit, and reconstruct images.

#### The compression system consists of an encoder that converts the input image into a set of symbols, and a decoder that reconstructs the image from these symbols. The input image, denoted as f(x,y) is fed into the encoder, which processes it to create a compressed version. After transmission over a channel, this encoded data is fed into the decoder, which generates the output image, denoted as f^(x,y). The output image may or may not be an exact replica of the original image. If the reconstructed image f^(x,y) is identical to f(x,y) the compression is considered error-free or lossless. If there are differences, the compression is lossy, meaning some level of distortion is introduced.

#### **The Compression Process**

The encoding process in an image compression system involves three main steps:

1. **Mapping (Mapper):**Mapper transformed the input image f(x,y) into a format designed to reduce spatial and temporal redundancy. This mapping process is generally reversible and may or may not directly reduce the data required to represent the image. For example, run-length coding is a type of mapping that can achieve compression in this step. In video applications, this step may also involve using previous or future frames to reduce temporal redundancy.
2. **Quantization (Quantizer):**The quantizer reduces the accuracy of the mapper's output according to a pre-established fidelity criterion, essentially filtering out irrelevant information. This step is irreversible, meaning it cannot be undone during decompression. In video compression, the bit rate (measured in bits per second) often controls the quantizer to maintain a predetermined average output rate, which can lead to variations in visual quality from frame to frame depending on the content.
3. **Symbol Coding (Symbol Coder):**The final step involves generating a fixed-length or variable-length code to represent the quantizer's output. Variable-length coding is often used to minimize coding redundancy by assigning shorter codes to more frequent quantizer outputs. This step is reversible, meaning it can be undone during decompression.

#### **The Decompression Process**

The decompression process is simpler, involving two main steps:

1. **Symbol Decoding (Symbol Decoder):**This step reverses the symbol coding process, converting the coded data back into a format similar to the quantizer output.
2. **Inverse Mapping (Inverse Mapper):**The inverse mapper then converts the decoded data back into a spatial image representation.
3. Since quantization is irreversible, the decoder cannot fully reconstruct the original image when lossy compression is used. Thus, there is no inverse quantizer block in the source decoder.

### 

### 

### 

### 

### 

### 

### **Fundamental Steps in Digital Image Processing**

Digital Image Processing (DIP) involves a series of systematic steps to analyze, manipulate, and transform images. The fundamental steps in DIP are as follows:

1. **Image Acquisition:**
   * **Overview:** Image acquisition is the first step in digital image processing, where an image is captured and converted into a digital form. This stage might involve preprocessing tasks such as scaling or adjusting the resolution.
   * **Process:** This can be as simple as obtaining an image that is already in digital form, or it can involve more complex tasks like using cameras or sensors to capture an image and convert it into a digital signal.
2. **Image Enhancement:**
   * **Overview:** Image enhancement focuses on improving the visual appearance of an image or highlighting certain features to make them more distinguishable. It is a subjective process, meaning that the methods used are often chosen based on what "looks better" to human observers.
   * **Examples:** Common enhancement techniques include adjusting the contrast, brightness, or sharpness of an image.
3. **Image Restoration:**
   * **Overview:** Image restoration aims to improve the appearance of an image, similar to enhancement, but it is based on objective criteria. It uses mathematical models to reverse known degradation processes, such as noise or blurring, to recover the original image. Unlike enhancement, which is subjective, restoration is grounded in mathematical and probabilistic models, making it more objective.
4. **Color Image Processing:**
   * **Overview:** This step involves processing images in color. Color image processing has gained importance due to the increased use of digital images on the internet. It includes tasks like color correction, color balancing, and processing images in different color spaces (e.g., RGB, CMYK).
   * **Applications:** It is used in various applications such as digital photography, medical imaging, and remote sensing.
5. **Wavelets and Multiresolution Processing:**
   * **Overview:** Wavelets are mathematical functions used to represent images at multiple levels of resolution. Multiresolution processing allows for analyzing an image at various scales, which is useful for tasks like image compression and feature extraction.
   * **Applications:** Wavelet transforms are widely used in image compression techniques, such as JPEG2000.
6. **Image Compression:**
   * **Overview:** Image compression reduces the amount of data required to represent an image, making it easier to store and transmit. Compression techniques can be lossless or lossy, depending on whether they preserve all image data or allow some data to be discarded for higher compression.
   * **Examples:** Common image compression formats include JPEG (lossy) and PNG (lossless).
7. **Morphological Processing:**
   * **Overview:** Morphological processing involves the analysis and manipulation of image structures based on their shapes. This step is crucial for tasks that require understanding the geometry of objects within an image.
   * **Applications:** It is used for tasks like edge detection, noise removal, and object extraction.
8. **Segmentation:**
   * **Overview:** Segmentation is the process of partitioning an image into distinct regions or objects. It is one of the most critical and challenging tasks in image processing because accurate segmentation is essential for subsequent analysis, such as object recognition.
   * **Challenges:** A robust segmentation algorithm is necessary to ensure that objects are accurately identified and separated from the background.
9. **Representation and Description:**
   * **Overview:** After segmentation, the image data is typically raw and needs to be represented in a form suitable for further processing. This stage involves choosing a method to represent the segmented regions, either by their boundaries or by their entire regions.
   * **Description:** The description (or feature selection) step involves extracting attributes or features from the image that can be used for further analysis, such as object recognition.
10. **Recognition:**
    * **Overview:** Recognition is the final step in digital image processing, where the processed image is analyzed to identify and classify objects within the image. This step assigns labels to objects based on their features.
    * **Applications:** Recognition is used in various applications, such as facial recognition, medical imaging, and automated vehicle identification.

### **Components of an Image Processing System**

Image processing systems have evolved significantly over the past few decades, reflecting advancements in both hardware and software technologies. The evolution has seen a shift from large, bulky peripheral devices attached to substantial host computers to more compact and integrated systems. This transformation has been driven by the need for cost-effectiveness, increased processing power, and the miniaturization of hardware components. Let us delve into the various components that comprise a typical general-purpose image processing system, each of which plays a crucial role in the overall functionality of the system.

#### **1. Image Sensing and Acquisition**

The process of image processing begins with image sensing and acquisition. To acquire digital images, two primary components are required:

* **Physical Sensing Device:** This device is sensitive to the energy radiated by the object being imaged. For instance, in a digital video camera, this would be the sensor that detects light intensity and converts it into an electrical signal. The sensor's role is to capture the scene or object in question, transforming the physical phenomenon (such as light) into a form that can be digitized.
* **Digitizer:** The digitizer converts the analog output of the sensing device into a digital form. For example, in a digital video camera, the analog signals generated by the sensor are converted into digital data by the digitizer. This digital data is what forms the basis for further processing. The process of digitization involves sampling and quantization, where the continuous analog signal is sampled at discrete intervals and then quantized into a finite number of levels.

#### **2. Specialized Image Processing Hardware**

Once the image is digitized, specialized hardware often takes over for initial processing tasks. This hardware is designed to perform operations that require high-speed processing, such as:

* **Arithmetic Logic Unit (ALU):** The ALU performs arithmetic and logical operations on entire images in parallel. An example of its use is in the averaging of images to reduce noise. The speed of the ALU is crucial because it allows for real-time processing, which is particularly important in applications like video processing where images need to be processed at rates of 30 frames per second or higher.
* **Front-End Subsystem:** This hardware unit, sometimes referred to as the front-end subsystem, handles high-throughput data operations that the main computer might not be able to process efficiently. Its distinguishing characteristic is speed, enabling it to digitize and perform operations like averaging video images rapidly.

#### **3. General-Purpose Computer**

The core of any image processing system is the computer itself, which can range from a personal computer (PC) to a supercomputer, depending on the application. For general-purpose image processing systems, a well-equipped PC is often sufficient for offline tasks. However, in dedicated applications, specially designed computers may be used to achieve the necessary performance levels.

The computer performs several critical functions:

* **Processing:** It executes the software that applies various image processing algorithms to the digitized images. These algorithms might include filtering, enhancement, segmentation, and feature extraction, among others.
* **Storage Management:** The computer manages the storage of images and processed data, ensuring that images can be accessed, processed, and stored efficiently.
* **User Interface:** It provides the interface through which users interact with the system, allowing them to control the image processing tasks, visualize results, and make decisions based on the processed images.

#### **4. Image Processing Software**

Software is the backbone of any image processing system, providing the algorithms and user interface necessary to process images. Image processing software consists of specialized modules designed to perform specific tasks, such as image enhancement, filtering, and segmentation.

A well-designed image processing software package includes:

* **Modular Structure:** This allows users to perform specific image processing tasks by selecting and applying the relevant modules.
* **Customization and Integration:** Advanced software packages enable users to write custom code that integrates with the specialized modules, offering flexibility and extending the software’s capabilities.
* **Support for Programming Languages:** The software may support integration with general-purpose programming languages, allowing for more sophisticated and customized processing routines.

#### **5. Mass Storage Capabilities**

Given the large amount of data generated in image processing, especially when dealing with high-resolution images or large volumes of data (such as in satellite imaging), robust storage solutions are essential. Storage is generally categorized into three types:

* **Short-Term Storage:** This is used for temporary storage during processing. Computer memory or specialized boards called frame buffers are typically used for this purpose. Frame buffers allow for rapid access and manipulation of images, which is essential for real-time applications like video processing.
* **On-Line Storage:** This storage is used for data that needs to be accessed frequently and quickly. Magnetic disks or optical media are commonly used for online storage. The key characteristic is the speed of data retrieval, which is crucial for efficient processing and analysis.
* **Archival Storage:** This type of storage is used for long-term storage of data that is not accessed frequently but needs to be preserved for future reference. Archival storage often uses magnetic tapes or optical disks, with "jukeboxes" being used to manage large volumes of media.

#### **6. Image Display**

The ability to display images is fundamental in an image processing system. The display device, typically a color monitor, is used to visualize the processed images. Modern image displays include:

* **TV Monitors:** These are commonly used for displaying images in real-time. They are driven by graphics display cards that are part of the computer system.
* **Specialized Displays:** In certain applications, such as virtual reality or 3D imaging, stereo displays may be used. These are often implemented as headgear containing small displays embedded in goggles worn by the user.
* **Display Cards:** The display cards in the computer are responsible for rendering images on the screen. These cards handle the necessary computations to ensure that images are displayed accurately and at high speed.

#### **7. Hardcopy Devices**

In many cases, there is a need to produce a physical record of the processed images. Hardcopy devices, including laser printers, film cameras, inkjet printers, and digital storage devices like CD-ROMs, are used to create physical copies of the images. Film, for instance, provides the highest resolution, but paper is more practical for most printed materials. Digital formats are increasingly popular for presentations and long-term storage.

#### **8. Networking**

Networking is an essential component in modern image processing systems, particularly when images need to be shared across different locations or systems. The primary concern with networking in image processing is bandwidth, given the large size of image data files. Advances in broadband technologies, such as optical fiber, have significantly improved the efficiency of image transmission over networks

### **Visual Perception**

Understanding visual perception is crucial in the field of digital image processing, where techniques are often selected based on subjective visual judgments. Here’s a breakdown of the key elements of visual perception:

#### **1. Structure of the Human Eye**

The human eye is a complex organ responsible for capturing visual information. A simplified cross-section of the eye reveals the following key components:

* **Cornea and Sclera**: The **cornea** is the transparent, dome-shaped front part of the eye. It acts as the eye’s initial lens, bending light to help focus it onto the retina. The **sclera** is the white, opaque part of the eye that surrounds the cornea and provides structural support and protection to the inner components.
* **Choroid**: Positioned between the sclera and retina, the **choroid** is a layer rich in blood vessels. These vessels supply nutrients and oxygen to the eye tissues. The choroid also contains pigment that absorbs excess light, preventing internal reflection and contributing to clearer vision. Damage to the choroid can severely impact vision due to inflammation and reduced blood supply.
* **Ciliary Body and Iris**: The **ciliary body** contains muscles that control the shape of the lens, enabling the eye to focus on objects at varying distances. The **iris** is the colored part of the eye and regulates the amount of light entering the eye by adjusting the size of the **pupil**, the central opening.
* **Lens**: The lens is a transparent, flexible structure made of concentric layers of fibrous cells. It changes shape to focus light on the retina. Composed of about 60-70% water and 6% fat, the lens also contains proteins that absorb a portion of the visible light spectrum. As people age, the lens can become clouded, leading to conditions like cataracts that affect color discrimination and visual clarity.
* **Retina**: The **retina** is the innermost layer of the eye that lines the back of the eye. It contains photoreceptors that detect light and convert it into electrical signals. The retina’s structure includes two types of photoreceptors: **cones** and **rods**. Cones are concentrated in the fovea, the central part of the retina, and are responsible for detailed and color vision under bright light conditions. Rods are more numerous and distributed around the retina, providing vision in low-light conditions and helping with peripheral vision.

#### **2. Image Formation in the Eye**

The human eye functions similarly to a camera, where the lens focuses light onto the retina to form images. The lens adjusts its curvature to focus on objects at varying distances, changing the focal length between 14 mm to 17 mm. This flexibility allows the eye to focus on both near and distant objects.

For example, if a person is looking at a 15-meter-tall tree 100 meters away, the size of the tree's image on the retina can be calculated using the lens' focal length. The perception of this image occurs through the excitation of light receptors in the retina, which convert light into electrical impulses that the brain decodes.

### 

### 

### 

### **Brightness Adaptation and Discrimination**

* **Brightness Adaptation**:

The human eye is quite good at adjusting to various levels of light. Whether you’re stepping from a bright room into a dark theater or vice versa, your eyes can quickly adapt to help you see clearly. However, the eye can’t handle extremely bright and very dim light at the same time. Instead, it adjusts to the current lighting condition. For instance, if you move from a sunny spot into a dark room, your eyes will gradually adjust to the lower light level. This ability to change sensitivity based on lighting conditions is called **brightness adaptation**.

* **Discrimination**:

Even though your eyes can adapt to different light levels, they have a limit to how well they can tell apart various brightness levels within that adapted range. For example, if you’re in a dimly lit room, you might have trouble distinguishing between very similar shades of gray. This is because the range of brightness you can clearly perceive is relatively narrow around your current lighting condition. The **brightness adaptation curve** shows how the eye’s ability to distinguish between different brightness levels changes based on the current light level. Essentially, the eye can only differentiate between light intensities effectively within a small range around the level it is adapted to.

**Sampling and Quantization**

When we capture an image using a sensor, such as a camera, the sensor typically generates a continuous voltage waveform. This waveform varies based on the physical characteristics of the scene being captured, like brightness, colors, and other details. However, to store and process this image on a computer, which deals with digital data, we need to convert this continuous analog information into digital form. This conversion is accomplished through two fundamental processes: **sampling** and **quantization**.

### **1. Sampling:**

Sampling is the process of selecting specific points from the continuous image to represent the entire image. It involves breaking down the image into small, discrete units, typically pixels.Consider a continuous image, such as a photograph, which can be represented by a function f(x, y). In this function, 'x' and 'y' represent the coordinates of a point on the image, and f(x, y) gives the amplitude (intensity or brightness) at that point.

* **Sampling in the Spatial Domain**: In sampling, we choose a grid of points on the continuous image and record the intensity values at these points. For example, if you draw a line across the image and measure the brightness at regular intervals along this line, you are performing a one-dimensional sampling. This line is often referred to as a "scan line."
* **Equally Spaced Samples**: In the example, imagine that the line across the image is divided into equal segments, and at each segment's boundary, we measure the intensity. The locations where these measurements are taken are called sampling points. The spacing between these points is crucial; if the points are too far apart, we might miss important details, and if they are too close together, it might be unnecessary or too data-heavy.
* **Digital Representation of Coordinates**: The coordinates of these sampling points are also digitized, meaning each point on the image is now represented by a discrete set of x and y coordinates, corresponding to the rows and columns of a grid.

### **2. Quantization:**

Quantization is the process of converting the continuous intensity values of the sampled points into discrete levels. This step is necessary because digital systems can only handle a finite number of levels.

* **Continuous Intensity Values**: After sampling, the intensity values at each sampled point are still continuous, meaning they can take any value within a certain range. However, to store these values digitally, we must map them to a limited set of discrete levels.
* **Dividing the Intensity Range**: Imagine the range of possible intensity values as a continuous scale. We divide this scale into a set number of levels. For instance, if we choose to have 8 levels, we divide the entire intensity range into 8 equal parts. Each part represents a discrete level.
* **Assigning Discrete Values**: During quantization, each sampled intensity value is assigned to the nearest discrete level on this scale. For example, if a sampled point has an intensity that falls between the third and fourth levels, it will be assigned to either the third or fourth level, depending on which one it is closer to.
* **Resulting Digital Image**: After both sampling and quantization, the continuous image is now represented as a grid of pixels, each with a specific x and y coordinate and a discrete intensity value. This grid forms the digital image that can be stored, displayed, and processed by digital devices.

**Color models**

Color models are mathematical models that describe how colors can be represented as tuples of numbers, typically as three or four values or color components. These models are essential in various fields, such as image processing, computer graphics, and photography. Here’s a detailed explanation of the RGB, CMYK, HSV, and YIQ color models:

### **1. RGB Color Model**

* **Description**: The RGB (Red, Green, Blue) color model is an additive color model in which colors are created by combining different intensities of red, green, and blue light. It is one of the most widely used color models, especially in electronic displays like monitors, TVs, and digital cameras.
* **How It Works**: Each color in the RGB model is represented by three components: red, green, and blue. The intensity of each component can range from 0 to 255 (in 8-bit color), with 0 meaning no contribution and 255 meaning full intensity. By varying the intensities of these three primary colors, a wide range of colors can be created. For example:
  + **Black**: (0, 0, 0)
  + **White**: (255, 255, 255)
  + **Red**: (255, 0, 0)
  + **Green**: (0, 255, 0)
  + **Blue**: (0, 0, 255)
  + **Yellow**: (255, 255, 0)
* **Applications**: The RGB model is used in devices that emit light, such as screens and digital cameras.

### **2. CMYK Color Model**

* **Description**: The CMYK (Cyan, Magenta, Yellow, Black) color model is a subtractive color model used in color printing. Unlike RGB, which deals with light, CMYK is based on the absorption of light by inks.
* **How It Works**: In the CMYK model, colors are created by subtracting varying percentages of cyan, magenta, and yellow from white light. The black (K) component is added to enhance the depth and contrast of the color. The percentages range from 0% (no ink) to 100% (full ink coverage). For example:
  + **White**: (0%, 0%, 0%, 0%)
  + **Black**: (0%, 0%, 0%, 100%)
  + **Red**: (0%, 100%, 100%, 0%)
  + **Green**: (100%, 0%, 100%, 0%)
  + **Blue**: (100%, 100%, 0%, 0%)
* **Applications**: The CMYK model is widely used in color printing, including in printers and in the production of printed materials like brochures, magazines, and posters.

### **3. HSV Color Model**

* **Description**: The HSV (Hue, Saturation, Value) color model, also known as HSB (Hue, Saturation, Brightness), is a cylindrical coordinate representation of colors. It is often used in image editing software and in contexts where color perception by humans is important.
* **How It Works**:
  + **Hue**: Represents the type of color and is measured in degrees from 0° to 360° on the color wheel (0° is red, 120° is green, and 240° is blue).
  + **Saturation**: Represents the vibrancy of the color, ranging from 0% (gray) to 100% (fully saturated color).
  + **Value/Brightness**: Represents the brightness of the color, ranging from 0% (black) to 100% (full brightness).
  + **Examples**:
    - **Red**: (0°, 100%, 100%)
    - **Pink**: (0°, 50%, 100%)
    - **Gray**: (0°, 0%, 50%)
    - **White**: (0°, 0%, 100%)
* **Applications**: The HSV model is used in graphics design, computer vision, and image editing because it aligns more closely with how humans perceive and interpret colors.

**HSI COLOR MODEL**

* **Description**: The HSI color model is specifically designed to represent how humans perceive colors. It separates the image intensity from color information (hue and saturation), making it particularly useful for tasks like image analysis, computer vision, and image processing.
* **Components**:
  + **Hue (H)**: Represents the type or shade of the color. It is measured in degrees from 0° to 360° on a color wheel, where 0° is red, 120° is green, and 240° is blue.
  + **Saturation (S)**: Represents the purity or vibrancy of the color, ranging from 0 to 1. A saturation of 0 corresponds to a shade of gray, and a saturation of 1 corresponds to a pure color.
  + **Intensity (I)**: Represents the brightness or lightness of the color, ranging from 0 to 1. A value of 0 represents black, and a value of 1 represents white.

### **2. How It Works**

* **Hue (H)**:
  + Hue is derived from the angle around the color wheel.
  + Pure red has a hue angle of 0°, green has 120°, and blue has 240°.
  + The hue component is independent of the color’s brightness or saturation.
* **Saturation (S)**:
  + Saturation describes how much the color is diluted with white light.
  + A fully saturated color contains no white light and appears vivid, while a less saturated color appears more washed out or pastel.
* **Intensity (I)**:
  + Intensity is the average of the red, green, and blue components.
  + It represents the amount of light in the color and is calculated as the average value of the RGB components.
  + This makes it possible to separate the color information from the brightness, which is useful in various image processing applications.

### **Inverse Filter**

#### **Overview**

The inverse filter is a technique used in image restoration to recover an original image that has been degraded by some known distortion. It is a straightforward method that assumes the degradation process can be modeled by a linear system, and the goal is to reverse the effect of this degradation.

#### **Mathematical Formulation**

Let H(u,v) be the frequency response of the degradation function, and G(u,v) be the degraded image in the frequency domain. The relationship between the original image F(u,v) and the degraded image is given by:

G(u,v)=H(u,v)⋅F(u,v)G(u,v)

The inverse filter seeks to recover the original image by applying the inverse of the degradation function:

F^(u,v)=G(u,v)H(u,v)\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)}F^(u,v)=H(u,v)G(u,v)​

Where F^(u,v)\hat{F}(u,v)F^(u,v) is the estimated original image.

#### **Challenges**

1. **Noise Amplification**: If H(u,v)H(u,v)H(u,v) has very small values, the division by H(u,v)H(u,v)H(u,v) can lead to significant amplification of noise in the image.
2. **Ill-Conditioning**: In cases where H(u,v)H(u,v)H(u,v) approaches zero, the inverse filtering operation can become unstable or even impossible, making the reconstruction highly sensitive to errors.

### **Wiener Filter**

#### **Overview**

The Wiener filter is a more advanced image restoration technique that takes into account both the degradation function and the presence of noise. It aims to minimize the mean square error between the estimated image and the original image, making it more robust compared to the inverse filter.

#### **Mathematical Formulation**

The Wiener filter estimates the original image F(u,v)F(u,v)F(u,v) by considering both the degradation function H(u,v)H(u,v)H(u,v) and the power spectra of the noise and the original image. The Wiener filter is given by:

F^(u,v)=[H∗(u,v)∣H(u,v)∣2+SN(u,v)SF(u,v)]G(u,v)\hat{F}(u,v) = \left[\frac{H^\*(u,v)}{|H(u,v)|^2 + \frac{S\_N(u,v)}{S\_F(u,v)}}\right] G(u,v)F^(u,v)=​∣H(u,v)∣2+SF​(u,v)SN​(u,v)​H∗(u,v)​​G(u,v)

Where:

* F^(u,v)\hat{F}(u,v)F^(u,v) is the estimated original image.
* H∗(u,v)H^\*(u,v)H∗(u,v) is the complex conjugate of the degradation function.
* SN(u,v)S\_N(u,v)SN​(u,v) is the power spectral density of the noise.
* SF(u,v)S\_F(u,v)SF​(u,v) is the power spectral density of the original image.

#### **Advantages of Wiener Filter over Inverse Filter**

1. **Noise Consideration**: The Wiener filter explicitly accounts for the noise in the image, which helps to prevent the amplification of noise that often occurs with the inverse filter.
2. **Robustness**: The Wiener filter is more robust in cases where the degradation function H(u,v)H(u,v)H(u,v) has small values or is close to zero, as it does not rely solely on the inverse of H(u,v)H(u,v)H(u,v) but rather balances it with noise considerations.
3. **Minimized Mean Square Error**: The Wiener filter is designed to minimize the mean square error between the restored image and the original image, leading to a better overall restoration quality.

JPEG (Joint Photographic Experts Group) compression is one of the most widely used image compression techniques. It is particularly effective for photographic images, balancing compression ratio and image quality. JPEG compression works by reducing the redundancy and irrelevant data in images, allowing them to be stored and transmitted efficiently. Below is a detailed explanation of the parts of the JPEG compression block diagram:

### **1. Color Space Transformation**

* **Input:** The input image is usually in RGB color space.
* **Process:** The RGB image is first converted to the YCbCr color space. This color space separates the image into one luminance component (Y) and two chrominance components (Cb and Cr).
* **Reason:** Human eyes are more sensitive to changes in brightness (luminance) than color (chrominance), so the YCbCr space allows for more efficient compression by reducing the amount of data in the chrominance channels without significantly affecting perceived image quality.

### **2. Downsampling**

* **Input:** The YCbCr components obtained from the color space transformation.
* **Process:** The chrominance components (Cb and Cr) are typically downsampled by a factor of 2 in both the horizontal and vertical directions. Common downsampling ratios include 4:2:2 and 4:2:0.
* **Reason:** Since the human eye is less sensitive to color information, downsampling these components reduces the data size significantly while maintaining acceptable visual quality.

### **3. Block Splitting**

* **Input:** The Y, Cb, and Cr components after downsampling.
* **Process:** The image is divided into 8x8 pixel blocks. Each 8x8 block is processed independently in the subsequent steps.
* **Reason:** Processing the image in blocks helps to simplify the subsequent compression steps and allows for more efficient handling of the image data.

### **4. Discrete Cosine Transform (DCT)**

* **Input:** Each 8x8 block of the image.
* **Process:** The DCT converts the spatial domain data (pixel values) of each block into the frequency domain. The result is an 8x8 matrix of coefficients, where the top-left coefficient represents the DC component (average intensity), and the remaining coefficients represent the AC components (spatial frequencies).
* **Reason:** The DCT helps to concentrate most of the image’s energy into the low-frequency components, allowing for more effective compression in the subsequent quantization step.

### **5. Quantization**

* **Input:** The DCT coefficients of each 8x8 block.
* **Process:** Each DCT coefficient is divided by a corresponding value in a quantization matrix and then rounded to the nearest integer. The quantization matrix is designed so that higher-frequency coefficients, which typically contain less visually significant information, are more heavily quantized (i.e., more aggressively reduced).
* **Reason:** Quantization is the primary source of compression in JPEG. It reduces the precision of the DCT coefficients, leading to a loss of information that is usually not noticeable to the human eye. The extent of quantization can be adjusted, affecting the trade-off between compression ratio and image quality.

### **6. Zigzag Scanning**

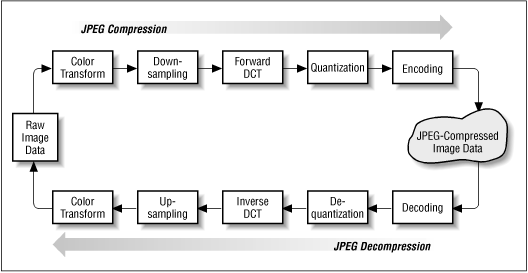
* **Input:** The quantized DCT coefficients.
* **Process:** The 8x8 block of quantized coefficients is scanned in a zigzag order, which arranges the coefficients from the lowest frequency to the highest frequency.
* **Reason:** Zigzag scanning helps to group the low-frequency (and often non-zero) coefficients together, which is advantageous for the next step, entropy coding. It maximizes the run lengths of zeros, which can be efficiently encoded.

### **7. Entropy Coding**

* **Input:** The zigzag-ordered quantized coefficients.
* **Process:** JPEG typically uses a combination of **Run-Length Encoding (RLE)** and **Huffman coding** for entropy coding.
  + **RLE:** Compresses sequences of zeros by recording the number of consecutive zeros before a non-zero coefficient.
  + **Huffman Coding:** Assigns shorter codes to more frequent values and longer codes to less frequent values, reducing the overall number of bits needed to represent the data.
* **Reason:** Entropy coding further reduces the data size by exploiting statistical redundancy in the quantized coefficients.

### **8. Bitstream Formation**

* **Input:** The entropy-coded data.
* **Process:** The entropy-coded data from each block, along with any necessary header information (e.g., quantization tables, Huffman tables), is assembled into a compressed JPEG file.
* **Reason:** The final step ensures that the data is in the correct format for storage or transmission as a JPEG file.



### **Thresholding in Image Segmentation**

Thresholding is a fundamental technique in image segmentation used to separate objects from the background in an image. It is a simple and effective method for binary image segmentation, where the goal is to classify each pixel into one of two categories: object or background.

#### **Concept of Thresholding**

Thresholding involves converting a grayscale or color image into a binary image based on pixel intensity values. The idea is to choose a threshold value such that all pixel values greater than or equal to this threshold are classified into one category (e.g., foreground or object), and all pixel values less than the threshold are classified into the other category (e.g., background).

#### **Types of Thresholding**

1. **Global Thresholding**
   * **Description**:
     + In global thresholding, a single threshold value is used to segment the entire image. This threshold is typically chosen based on the histogram of pixel intensities.
   * **Process**:
     + Convert the grayscale image into a binary image using the global threshold value TTT.
     + For each pixel I(x,y)I(x, y)I(x,y) in the image: Binary(x,y)={1if I(x,y)≥T0if I(x,y)<T\text{Binary}(x, y) = \begin{cases} 1 & \text{if } I(x, y) \geq T \\ 0 & \text{if } I(x, y) < T \end{cases}Binary(x,y)={10​if I(x,y)≥Tif I(x,y)<T​
   * **Applications**:
     + Suitable for images with a clear distinction between foreground and background intensities.
   * **Example**:
     + If the threshold TTT is set to 128 in an 8-bit grayscale image (0-255), pixels with values greater than or equal to 128 are classified as foreground, and those with values less than 128 are classified as background.
2. **Adaptive Thresholding**
   * **Description**:
     + Adaptive thresholding adjusts the threshold value based on local image properties, such as the pixel intensity values in a local neighborhood. This method is useful for images with varying illumination or uneven background.
   * **Process**:
     + For each pixel (x,y)(x, y)(x,y), compute the threshold value based on the local neighborhood around (x,y)(x, y)(x,y).
     + Apply the local threshold to segment the image.
   * **Applications**:
     + Useful for images with non-uniform lighting or varying contrast across different regions.
   * **Methods**:
     + **Mean or Median-Based Adaptive Thresholding**: The threshold is computed as the mean or median of the pixel values in the local neighborhood.
     + **Gaussian Adaptive Thresholding**: The threshold is computed as the weighted sum of pixel values in the local neighborhood using a Gaussian filter.
   * **Example**:
     + For a pixel at position (x,y)(x, y)(x,y), the local threshold might be computed as the average of pixel values in a surrounding window. Pixels are then classified based on this local threshold value.
3. **Otsu’s Method**
   * **Description**:
     + Otsu’s method is an automatic thresholding technique that chooses the threshold value to minimize the intra-class variance (or equivalently, maximize the inter-class variance) between the foreground and background.
   * **Process**:
     + Compute the histogram of pixel intensities.
     + Calculate the cumulative distribution function and the class probabilities.
     + Find the threshold value TTT that minimizes the weighted within-class variance.
   * **Applications**:
     + Suitable for images with bimodal histograms (two distinct peaks representing foreground and background).
   * **Example**:
     + For an image with a bimodal histogram, Otsu’s method will compute the threshold that best separates the two peaks, thus optimizing the segmentation.
4. **Multi-Level Thresholding**
   * **Description**:
     + Multi-level thresholding extends the concept of binary thresholding to segment the image into more than two regions. Multiple thresholds are used to classify pixels into multiple categories.
   * **Process**:
     + Compute several thresholds to create multiple binary images or segments.
     + Assign each pixel to one of the multiple categories based on its intensity value.
   * **Applications**:
     + Useful for images with multiple distinct regions or objects.
   * **Example**:
     + For an image with different gray levels representing different objects, multiple thresholds can be used to segment the image into several regions, each corresponding to a different object or feature.

#### **Role in Separating Objects from Background**

1. **Binary Segmentation**:
   * Thresholding converts a grayscale image into a binary image where the foreground (objects) and background are distinctly separated. This simplification facilitates further analysis, such as object counting, shape analysis, and feature extraction.
2. **Handling Variations**:
   * Adaptive and multi-level thresholding techniques address variations in illumination, contrast, and object sizes. By considering local pixel neighborhoods or using multiple thresholds, these methods improve segmentation accuracy in challenging conditions.
3. **Image Analysis and Processing**:
   * Once objects are separated from the background, various image processing tasks can be performed, including object recognition, tracking, and classification. For instance, thresholding can help isolate objects of interest in medical imaging, automated inspection systems, and video surveillance.
4. **Feature Extraction**:
   * By segmenting objects from the background, thresholding allows for the extraction of features such as contours, shapes, and sizes. This information is essential for subsequent analysis and interpretation of the image content.