Report Day 2 and 3

Chapter 1: Representation

Introduction

Chapter 1 in this book grounds your understanding by providing insight into how the computer vision and image processing systems represent, store, and process digital images. It covers the underlying structural contents of images, such as pixel values and how they represent colours and image types and colour spaces that the encoding system may use to store image data. That is why the chapter also sets an introduction as to how the images are compressed and transformed for analysis.

1.1 What is an Image?

An image is essentially a spatially organized set of numerical values, where each value corresponds to a specific point (pixel) in space. Images are stored in 2D or 3D arrays depending on their type:

- - Binary Images: 1-bit per pixel; pixel value is either 0 or 1.
- - Grey-scale Images: 8-bit typically; intensity ranges from 0 (black) to 255 (white).
- - Colour Images: Consist of multiple channels (e.g., RGB) and are stored as 3D arrays

1.2 Types of Images

The chapter classifies digital images into several categories based on how they encode information:

- Binary Images: Simplest form, used in segmentation and shape analysis.
- Grey-scale Images: Intensity-based, used in medical and scientific imaging.
- Colour Images: RGB format with 3 channels.
- Indexed Images: Use a colour map for efficient storage.
- Multispectral and Hyperspectral Images: Capture multiple spectral bands, used in satellite imaging.

1.3 Image Compression

Compression reduces the size of image files for storage and transmission:

- - Lossless Compression: No data loss. Formats: PNG, GIF.
- - Lossy Compression: Some data loss for higher compression. Formats: JPEG.

1.4 Colour Spaces

Colour spaces define how colours are encoded numerically.

1.4.1 RGB Colour Space

RGB is the most common format, using three values (Red, Green, Blue) per pixel. It can be visualized as a 3D cube. It's device-based but perceptually nonlinear.

1.4.1.1 RGB to Grey-Scale Conversion

Grey-scale conversion uses a weighted sum of RGB values based on human visual perception:

$$I_{grey}(n, m) = 0.2989 * R + 0.5870 * G + 0.1140 * B$$

This is a non-invertible transformation.

1.4.2 Perceptual Colour Space – HSV

HSV stands for Hue, Saturation, and Value. It is more intuitive and suitable for segmentation and analysis:

- - Hue: Dominant wavelength (e.g., red, green).
- - Saturation: Purity of colour.
- - Value: Brightness.

Chapter 2 Formation

Introduction

Chapter 2 introduces the basic concepts about how digital images are derived from real-world scenes and the engineering factors. This chapter is essential to know about the origins of data from which every image processing and computer vision method refers. This chapter discusses the role of physical/mathematical models of image forming mechanisms, engineering requirements of image capturing, and the noise always present that degrades image appearance.

1. Formation of Images

1.1 Real-World Scene to Image

- A real-world scene is a three-dimensional (3D) object/space containing objects, light sources, and spatial relations.
- An image is the 2D equivalent of this 3D, scene captured by a camera or imaging device.
- The image formation process involves optics funneling light (from the scene) to some sensor which map physical light intensities into electronic signals.

1.2 Optical and Sensor Principles

- Optics (Lenses): Light rays emitted from various parts of the scene are bent and focused by the camera lens, and the sensor plane holds the processed information as an image.
- Sensor: The sensor (CCD or CMOS) responds to discrete levels of intensity of light at discrete locations (known as pixels or picture elements).

- Projection: The camera projects 3D points onto a 2D plane through perspective or orthographic projection methods, reducing the spatial information.

1.3 Sampling and Quantization

- The sensor effectively samples the continuous intensity distribution of the light into discrete pixels.
- Quantization assigns each sampled value to finite levels of intensity such as a range of 0 to 255, resulting in an 8-bit image.
- Each pixel location in the image is represented by the assigned levels within the image as intensity values that are represented as a matrix of intensity values.
- Limitations in sampling frequency, quantization depth, and the association of light intensity can drastically inhibit the ability of camera images to faithfully deliver what was originally scene in space during recording, with reduced detail, and some distortion to the projected or recorded image.

2. The Mathematics of Image Formation

2.1 Mathematical Modeling

- The image formation can be described by a function $I(x,y) = f(L(x,y))$, where L is scene radiance and I is the intensity of the image recorded.
- Transitioning from a continuous to a discrete form involves sampling (demonstrated with the operations on functions), convolution with the lens point spread function, and then the functions correspondence to discrete values through

2.2 Sampling Theory and Aliasing

- The Sampling Theorem: For the purpose of sampling without aliasing, you must sample at a frequency with at least double the highest frequency found in the scene.
- Not sampling at least double the frequency means we will experience aliasing, where high frequency detail in the scene is confused into distortions and false patterns in the sampled, or the record image.
- No camera in practical use today can follow these conditions perfectly, this is why aliasing is important to consider.

2.3 Digital Image Representation

- A digital image will be stored as a 2D matrix $I[m,n]$, where each value in that matrix will correspond to pixel intensity.
- For example, in a color image, there are several 2D matrices that correspond with the respective color channels (RGB).
- Each pixel value will also be discrete values based on quantization.

3. The Engineering of Image Formation

3.1 Imaging System Components

- Lens system which focuses the scene onto the image sensor- the quality of lens affects sharpness, distortion, and field of view.

- Image Sensor which takes in light and processes onto electrical signals. The image sensor's sensitivity and noise characteristics are often defined by the variety of sensor design and manufacturing technologies.
- Signal Processing electronics, to convert the sensor analog signals into digital values, to perform first step processing which can include gain controlling, white balancing, and compression process.

3.2 Limitations and Challenges in Engineering

- Sensor saturation where there is too much incoming light which saturates pixels and loses detail during bright image processing
- Underexposure where there is not enough incoming light which results in low intensity noisy images.
- Lens aberrations where imperfections in optics cause blur, unintentional distortion, and unwanted vignetting.
- Limitations in focus depth, and the physical mechanism of focusing that limits depth of sharpness across the scene to a very small area.
- Mechanical factors such as vibration, dust, and the mechanical and electrical environment can also have an impact on the image.

3.3 Image Processing Pipeline

- The pipeline starts with capturing the incoming light energy through image intensity from the sensor, followed by following each phase of the pipeline through the sampling of the image sensor, the A/D conversion stage, the DSP phase of signal processing, and the image encoding or compression & storage stage.
- At each stage, signal and noise imperfections can be introduced.

4. Noise in Image Formation

The presence of noise is one of the most challenging obstacles to the attainment of accurate and clear images. Noise indicates pixel value variations that arise due to random variables associated with uncontrollable environmental factors inherent in the imaging pipeline.

4.1 Sources of Noise

4.1.1 Capture Noise

- Caused by environmental and sensor factors during image acquisition:
  * Variations in lighting conditions.
  * Temperature-induced sensor noise.
  * Electrical noise within sensor circuits.
  * Sensor nonuniformity (unequal pixel responses).
  * Dust particles or dirt on lens or sensor.
  * Vibration during capture.
  * Lens distortion and focus inaccuracies.
  * Sensor saturation due to excessive light.
  * Underexposure due to insufficient light.

4.1.2 Sampling Noise

- Results from limited sampling resolution and intensity quantization.
- A representative aliasing is the digital image as a representation of a continuous analog image scene.

4.1.3 Processing Noise

- Occurs when processing steps are required in the digital sense of the image.
- Causes include inherent limits in numerical precision (e.g., floating-point roundoff errors), internal integer overflow, or intrinsically approximate mathematics. (e.g., using $\pi \approx 3.142$).

4.1.4 Image-Encoding Noise

- Given that lossy compression algorithms (e.g., JPEG) cause a loss of image detail that is so small as to be unrecoverable by human perception.
- Lossy compression will incur some image artifacts, which are distortions that are discernable visually.
- Lossless formats (e.g., PNG) add digital noise by preserving information.

4.1.5 Scene Occlusion

- The objects in the scene are either partially blocked by other objects or too far away to identify, resulting in incomplete information.
- This occlusion complicates recognizing object boundaries and event explanation in the framed image, that results in limits the performance based on the features that were available.

4.2 Noise Types and Characteristics

4.2.1 Salt and Pepper Noise (Impulse Noise)

- Random occurrences of bright white or dark black pixels.
- Often caused by sensor faults like hot pixels or dead pixels.
- Less common with modern sensors but still important in some applications.

4.2.2 Gaussian Noise (Additive Noise)

- Noise values distributed according to a normal (Gaussian) distribution centered around the true pixel value.
- Most common noise model in image processing.
- Represents the cumulative effect of many small, independent noise sources.

4.3 Impact of Noise on Image Processing

- Noise significantly degrades image quality and can cause failure of image processing algorithms.
- Noise reduction and removal are critical steps in any image processing pipeline.
- Robust image processing systems must be designed to cope with noise at all stages.
- The presence of noise necessitates advanced filtering, denoising algorithms, and sometimes additional hardware design improvements.

Chapter 3: Pixels

Introduction

Chapter 3 focuses on the pixel, the building block of digital images, and the various operations that may be performed on pixels. The chapter primarily focused on point operations that alter image pixels independently, and how we can use histograms -- which describe the distribution of pixel intensities in an image -- to guide and provide choices for the various contrast enhancement techniques. In this sense, this chapter represents fundamental concepts that are useful for understanding digital images and working with digital images, at a low level.

1. What is a Pixel?

• The term pixel, short for "picture element," refers to the smallest discrete element of an image grid.
• Each pixel represents a spatial location and contains data about the light intensity (in greyscale images) or color (in color images).
• Usually, pixels are arranged in a 2D rectangular area defined by a number of rows and a number of columns.
• In the case of greyscale images, each pixel's value represents intensity and is typically quantized into levels, for example, 0 to 255 for 8 bit depth.
• Each pixel in a color image has more than one channel (e.g., Red, Green, Blue) each with its own intensity; the various channels are combined to define the perceived color of the pixel.
• The number of pixels that are defined horizontally and vertically create the image resolution.
• Understanding pixels is essential because they are the basis of all subsequent image processing.

2. Operations Upon Pixels

• Image processing can manipulate pixels in several ways:
- Point operations can modify single pixels without regard to their neighbors.
- Neighborhood operations can involve pixels and their neighbors.
- Geometric operations can change the location of pixels (e.g., rotation).
• In this chapter, we focus exclusively on point operations, which change the value of the pixels without consideration of neighboring pixels.
• Point operations are computationally cheap and basic operations are applicable for basic

enhancements which include brightness adjustment, inversion, and thresholding.
• Formally, an operation is described by a mapping function $s = T(r)$, where r is an input pixel intensity and s is the output pixel intensity.

3. Point-Based Operations on Images

Definition and Concept

• A point-based operation applies a function to each pixel's intensity value independently.
• Because pixels are processed independently, these operations are fast and can be implemented via lookup tables (LUTs) for real-time applications.

Common Point Transformations
• Identity Transformation:
-This is the simplest operation. The output pixel equals the input pixel, or $s = r$. As with all operations of this kind, no change occurs.
• Negative Transformation:
This transformation results in the inversion of pixel intensities in the image, creating a photographic negative of the image. The expression for this transformation is as follows: $s = L - 1 - r$, where L is the total number of grey levels in the image (for example, 256). Consequently, lower pixel values become higher pixel values and vice versa.
• Contrast Stretching:
This transformation stretches the full range of all distant grey levels to make use of the full scale of grey levels (for example, stretched to 0 to 255) in effect improving the contrast of the image. This usually means that the transformation is some form of linear mapping of the distant grey levels in the image to the range of grey levels available. In general, the maximum and minimum pixel intensities in the image should be the maximum and minimum pixel values in the available range.
• Thresholding:
Thresholding is a transformation that will produce a binary image (0-1) compared to original image values. This is done by setting values greater than some threshold value to one value (e.g. 255) and values less than the threshold to another value (e.g. 0). This transformation serves a useful purpose in the segmentation of images, and feature extraction.
• Gamma Correction: Gamma correction is a nonlinear mapping applied to images to comply with how humans perceive brightness and for adjustments in brightness of display outputs. The transformation expression is as follows: $s = c r^\gamma$, where parameter γ controls the degree of adjustment of the brightness. Values of γ lesser than 1 are used brightens an image where values of γ greater than 1 are used to darken it.

Lookup Tables for measurement of Operational Code
• One way to accomplish this is to pre-compute the transformation function T(r) into a lookup table of the required size, i.e., L.
• For each possible input intensity of r, compute T(r) and put it into the same index for the lookup table, so we can map the pixel intensity values using a lookup table by the appropriate index.

• This echnique can be useful for increasing performance in a real-time system, or a large images.

4. Pixel Distributions: Histograms

Histogram Definition

• A histogram is a visual display of the frequency of occurrence of each intensity level of pixels in an image.

• An 8-bit image has a histogram that contains 256 bins, with each bin equaling an intensity value (0 to 255).

• The height of the bin corresponds to the number of pixels in the image that are at that intensity value.

Interpreting Histograms

• Histograms are useful in extracting information on an image's brightness, contrast and dynamic range.

• General Characteristics:

• A histogram that is narrow and concentrated and all of the pixel intensities are in a narrow range, indicates low contrast.

• A histogram that is skewed to the left indicates that the image is dark overall (underexposed).

• A histogram that is skewed to the right indicates that the image is bright (overexposed).

• A histogram that has a broad and evenly distributed spread indicates good contrast with a good amount of dynamic range.

Uses of Histograms

• Histograms are useful for determining the appropriate image enhancement operations.

• Manipulating a histogram is a powerful method to enhance an image's contrast and visual quality.

5. Histogram-Based Enhancement Operations

Global histogram equalization

• Purpose: To enhance image contrast by redistributing pixel intensities so that the histogram of the output image is nearly uniform.

• It involves:

- Computing the cumulative distribution function (CDF) of the histogram of the image.
- Using the CDF of the histogram as a transformation function to relate input intensities to output intensities.

• This "spreads out" commonly occurring intensity values across the entire intensity range, providing greater visibility.

• The main drawbacks are:

- It can over-enhance noise.
- It may create unnatural effects by over-flattening the histogram.

Histogram matching (specification)
• This operation maps the histogram of a source (input) image to a defined, specified or target histogram.
• It is useful for enforcing consistency between images and/or making an image look a specific way, e.g., matching the visual style of a reference image.
• Representing histogram matching (specification) involves computing the transformation from the input histogram to the target histogram using their respective CDFs, and fusing the two transformations.

6. Adaptive Histogram Equalization (AHE)

• Global histogram equalization functions poorly if an image has regions with significantly different intensity distributions.
• For instance, some regions may have brighter pixels, while others may have darker pixels. This means that a single global histogram equalization will not properly enhance all of these areas.

Tile-Based Adaptive Histogram Equalization
• The image is divided into contiguous small tile/regions; for instance, in a 100 x 100 pixel image, one might tile it with small 16 x 16 pixel tiles.
• A local histogram equalization will be calculated for each tile, separately.
• This local histogram equalization, adjusts local mapping and gives a better local contrast enhancement and reveal details in different regions of the image.

Problems with Simple Tile-Based AHE
• A big problem with processing tiles separately is the creation of block artifacts due to discontinuities at tile boundaries.
• This can create visible artifacts at tile boundaries. The resulting image looks like a patchwork of blocks, with varying degree contrast enhancements.

Overlapping Window Method (Pizer Method)
• To limit block artifacts, the images used overlapping "neighborhoods".
• An overlaying grid is placed on top the image. At each grid point, the "neighborhood" includes a surrounding window twice that grid spacing.
• Thus, neighboring windows overlap by 50% to include the same pixel in four adjacent neighborhoods.
• The output pixel value is computed as the sum of the weighted bilinear interpolation of the four histogram equalizations calculated from these surrounding overlapping windows.
The formula:
$I = (1 - a)(1 - b) I_1 + a(1 - b) I_2 + (1 - a) b I_3 + a b I_4$
where I_1, I_2, I_3, I_4 are equalized values from the four windows and a,b are distance weights (between 0 and 1).
• This smooth blending removes abrupt changes and reduces blockiness.

Contrast-Limited Adaptive Histogram Equalization (CLAHE)

• AHE can over-enhance noise in smooth areas, through excessive contrast stretching.
• CLAHE uses a contrast limit parameter l to limit maximum contrast amplification.
• Limits the histogram redistribution on a tile by tile basis, limiting noise and artefacts.
• CLAHE has become commonly used in applications and is also integrated into software tools like MATLAB.

Practical Considerations
• Tile size, overlap, and contrast limit will impact the final image quality.
• These parameters usually need to be individually set and adapted based on the image characteristics and desired result.
• Adaptive Histogram Equalization is flexible but somewhat empirical, with no global rule for settings that will always yield optimal results.

7. Histogram Operations on Colour Images

Challenges with Colour Images

• Colour images are comprised of multiple channels (e.g., RGB).
• If histogram equalization is applied independently to every channel, the colour information (chromaticity: colour hue and saturation) can be distorted, resulting in colours that look unrealistic.

Solution: Transform to Perceptual Colour Spaces
• The RGB image is transformed into a colour space that separates intensity (luminance) from colour (chromaticity) to maintain natural colours.
• The most common choice is the HSV (Hue, Saturation, Value) colour space:
 - Hue (H): Type of colour (red, green, blue etc.).
 - Saturation (S): Purity of the colour.
 - Value (V): Brightness/intensity of the colour.
• Histogram equalization will only be applied to the Value component, and the Hue and Saturation will not be altered.

Alternative colour models
• Other colour spaces such as Lab also separate luminance from chromaticity, and they can be used in the same way.

Conversion process
• Convert the RGB image to HSV.
• Perform histogram operations (equalization, matching, etc.) on the V channel.
• Convert the updated HSV image back into RGB.

Result

• The contrast of the image is enhanced without changing the natural appearance of the colours.
• This approach avoids colour distortions caused by independently applying histogram equalization to each channel.