University of Technology

Computer Science Department

4th Grade – Network Branch

Multi Media 1 – Course 1

2023-2024

أستاذ المادة: أ. د. عبد الأمير عبد الله كريم

Introduction to Multimedia

Multi media: means that computer information can be represented through audio, images, graphics, video and animation in addition to traditional media (text and graphics).

Video can be considered as an **integrated Multimedia** because it contains all the components of multimedia (images, sound and text).

Frame is any number of images in a time period (30 images per second) , those images are **similar (identical**) in characteristics.

1.1 What is Multimedia?

When different people mention the term **multimedia**, they often have quite different, or even opposing, viewpoints.

– ***A PC vendor***: a PC that has sound capability, video card, a DVD-ROM drive, and perhaps the superiority of multimedia-enabled microprocessors (e. g. Graphical Processing Unit **GPU**).

– ***A Computer Science (CS) student***: **applications** that use multiple modalities, including text, images, drawings (graphics), animation, video, sound including speech, and **interactivity**. From those applications, information hiding (**steganography**) which uses the available media as a cover to hide the secret data.

**** Multimedia and Computer Science:***

– Graphics, image processing, computer vision, data compression, networking - all have important contribution to make in multimedia at the present time.

1.2 Components of Multimedia

Multimedia involves multiple modalities of text, audio, images, drawings, animation, and video. Examples of how these modalities are **put to use**:

1. Video teleconferencing.
2. Distributed lectures for higher education (distance learning).
3. Tele-medicine.
4. Co-operative work environments – that allow business people to edit a shared document.
5. Searching in (very) large video and image databases for target visual objects (**Content Based Image or Video Retrieval**) .
6. Virtual Reality***:*** Interactive, computer-generated, three-dimensional graphics, delivered to the user through a head-mounted display.
7. Augmented Reality: placing real-appearing computer graphics and video objects into the real scenes.
8. Using **voice-recognition** to build an interactive environment application, say a **web browser**.

1.3 History of Multimedia

1. **Newspaper**: perhaps the ***first*** mass communication medium uses text, graphics, and images.
2. **Motion pictures**: conceived of in 1830's in order to observe motion too rapid for perception by the human eye.

3. **Wireless radio transmission**: Guglielmo Marconi, at Pontecchio, Italy, in 1895.
4. **Television**: the new medium for the 20th century, established video as a commonly available medium and has since changed the world of mass communications.
5. The **connection** between **computers** and ideas about **multimedia** covers what is actually only a short period:

1.4 Promising Multimedia Projects

Many exciting research projects are currently underway. Here are a few of them:

1. **Camera-based object tracking technology**: tracking of the control objects provides user control of the process.
2. **3D motion capture**: used for multiple actor capture so that multiple ***real actors*** in a ***virtual studio*** can be used to automatically produce ***realistic animated models*** with ***natural movement***.
3. **Multiple views**: from ***several cameras*** or from a single camera under ***differing lighting*** can accurately acquire data that gives both the **shape** and **surface** properties of materials, thus automatically generating synthetic **3D *graphics model***.
4. **3D capture technology**: allow **synthesis** of highly realistic facial animation from speech which is used in **fake videos**.
5. **Specific multimedia applications**: aimed at handicapped persons with **poor vision** capability and the **elderly** — a rich field of endeavor.
6. **Digital fashion**: aims to develop **smart clothing** that can communicate with other such enhanced clothing using wireless communication, so as to artificially enhance human interaction in a social setting.

7. **Electronic House call system**: an initiative for providing interactive ***health monitoring*** services to patients in their homes.
8. **Augmented Interaction applications**: used to develop **interfaces** between **real** and **virtual** humans for tasks such as augmented **storytelling (Hologram AVATAR)**.

1.5. Present Multimedia Applications

Examples of typical present multimedia applications include:

1. World Wide Web.
2. Multimedia courseware.
3. Distance learning
4. Video teleconferencing.
5. Virtual reality
6. Digital video editing and production systems.
7. Electronic newspapers/magazines.
8. On-line reference works: e.g. encyclopedias, games, etc.
9. Home shopping.
10. Interactive TV.
11. Interactive movies.
12. Video-on-demand.

In the below paragraphs, we describe some of the most important multimedia applications.

Video conferencing

Also cold teleconferencing, in which people in ***different geographical locations*** can have a meeting- can see and hear one another- using computers and communications. videoconferencing systems **rang** from ***videophones*** (is a telephone with TV-like screen and built-in camera that

allows you to see the person you're calling) to group ***conference rooms*** with cameras and multimedia equipment to desktop systems with small ***video cameras, microphones, and speakers***.

Video conferencing may eliminate the need for some ***travel*** for the purpose of meeting and allow ***people who cannot travel*** to visit "in person". Many organizations use videoconferencing to take the place of face-to-face meetings.

Distance learning

Telecommunication technology is enabling many people to ***learn outside the class room***, a process called distance learning. Distance learning can be ***point-to-point (synchronous)***, where students gathered at a specific location and the class is transmitted to them in ***real time (different place, same time)*.** The students are able to see and hear the professor, and the professor can hear the students off-site and may be able to see them as well. The off-site locations may be around the same campus or across the world.

Distance learning may also be ***asynchronous (different place, different time)***. Many courses are offered over the internet in prepackaged form.

Virtual reality : an emerging technology

There is no universal definition of virtual reality (VR). The most common ***definitions*** imply that virtual reality is interactive, computer-generated, three-dimensional graphics, delivered to the user through a head-mounted display.

Virtual reality, a computer-generated artificial reality, projects a person into a sensation of three-dimensional space. To put yourself into virtual

reality, you need ***software*** and special headgear, and then you can add gloves, and later perhaps a special suit. The headgear; which is called head-mounted display- has two small video display screens, one for each eye, for creating the sense of three-dimensionality. Headphones pipe in stereophonic sound or even 3-D sound. Three-dimensional sound makes you think you are hearing sounds not only near each ear but also in various places all around you. The glove has sensors for collecting data about your hand movements. Once you are wearing these equipments, software gives you interactive sensory feelings similar to real-world experiences.

Benefits of Virtual Reality :

1. More than one person and even a large group can share and interact in the same environment.
2. VR thus can be a powerful medium for communication, entertainment, and learning.
3. The user can grasp and move virtual objects.
4. In virtual reality, a person "believes" that he or she is doing is real, even though it is artificially created.
5. The ***entertainment*** applications are obvious, but this capability can even be utilized for gaining a competitive ***business advantage***.

Sophisticated virtual reality systems are interactive and usually simulate real world phenomena. They often simulate sight, sound, and touch and combine these senses with computer-generated input to user's eyes, ears, and skin. By using a head-mounted display, gloves, and bodysuit, or large projection images in simulator cabs, users can "enter" and interact with virtual or artificially generated environments.

With virtual reality, users can experience almost anything they want without ever leaving their chairs.

Virtual Reality Applications :

Military: training (pilots, drivers, shooting).

Medicine: training of surgeons (Surgeons can develop their skills through simulation on "digital patients).

Architecture: Simulate construction projects, create virtual objects on locations.

Entertainment Application.

Augmented Reality

Augmented Reality refers to Augmenting real world images with computer generated multimedia such as graphics, text, audio, and video.

One of the key technological challenges for creating an augmented reality is to maintain accurate **registration and tracking** between real word and computer generated object.

The main objective of AR is to increase human sense of reality by **adding some virtual information to the real scene**.

The Ultimate goal of any accurate registration method is: the computer generated virtual content (graphics, text, audio, and video), must be properly **overlaid on the real object**.

The three main **characteristics** of AR are:

1. Merge the virtual and real objects in the environment of real world.
2. Aligning (registering and Tracking) the virtual and real objects with each other in 3D environment.

3. Interactivity and in being a real time.

Virtual Reality Vs Augmented Reality

Augmented Reality	Virtual Reality
1. User maintains a sense of presence in real world. 2. System augments the real world scene with computer Generated multimedia. 3. Needs a mechanism to combine virtual and real worlds. 4. Hard to register real and virtual scene	1. The user is completely immersed in an artificial world and becomes isolated from the real environment. 2. Senses are under control of system. 3. Need a mechanism to feed virtual world to user. 4. Hard to make VR world interesting.

1.6 Internet

The internet , which is the largest computer network in the world, is actually, a network of networks. It is a collection of more than 200,000 individual computer networks owned by governments, universities, nonprofit groups, and companies. These interconnected networks exchange information seamlessly by using the ***same standards and protocols*** where the Transmission Control Protocol / Internet Protocol (TCP/IP) is the foundation protocols for the Internet. They are connected via high-speed, long-distance, backbone networks.

Thus, the internet forms a massive electronic communications network among businesses, consumers, government agencies, schools, and other organizations world-wide. Equally important, the internet has opened up exciting new possibilities that challenge traditional ways of interacting, communicating, and doing business. At the same time, the internet is raising new issues relating to culture and law, as it's about business.

1.7 World Wide Web

Are the internet and the World Wide Web the same things? Many people believe that the Web is synonymous with the internet, but that is not the case. The internet functions as the ***transport mechanism***, and the World Wide Web (WWW) is the ***application*** that uses those transport functions. Many applications run on the internet with ***e-mail, web sites and social media*** being the most widely used.

The World Wide Web is the biggest and the most widely known ***information source*** that is easily accessible and searchable. It consist of billion of ***interconnected documents*** (called ***web pages***) which are authored by millions of people.WWW is an architecture framework for accessing linked documents spread out over millions of machines all over the internet. The ***web would not be possible without the internet***, which provides the communication network for the web to function.

The World Wide Web is a system with universally accepted ***standards for storing, retrieving, formatting, and displaying information*** via client/server architecture. The ***Web handles all types of digital information, including text, hypermedia, graphics, and sound.***

Internet Vs World Wide Web

	Internet	World Wide Web
The Aim	1. Connected physical devices functions as a **transport** networks mechanism.	1. Is the **application** which use that transport mechanism.

Comprises	2. Computer networks, fiber optic cables, copper wires , wireless networks, and physical devices and services.	2. Folders, files, software and documents that stored in various computers (servers).
Nature	3. Hard ware.	3. Software.

What makes the World Wide Web so graphically inviting and ***easily navigable*** is that this international collection of servers **(1)** contains information in multimedia form and **(2)** is connected by hypertext links.

1. Multimedia form what makes the Web graphically inviting:

   Whereas e-mail messages are generally text, other web applications like ***web sites and social media*** provides information in multimedia form: graphics, video, and audio as well as text. You can see color pictures, animation, and full-motion video. You can download music. You can listen to radio broadcasts. You can have telephone conversation with others.

2. Use of hypertext and hypermedia what makes the Web easily navigable:

   Whereas with e-mail you can connect only with specific addresses you know about, with the Web you have hypertext. **Hypertext** is a system in which documents scattered across many internet sites are directly linked, so that a **word or phrase** in one document becomes a **connection to a document** in a different places.

From the most commonly terms we encounter on almost daily are:

- **Web site - the domain on the computer** : a website represent a ***centrally managed group of web pages*** , containing text , images

and all types of multimedia files presented to the internet users in an easily accessible way.

A computer with a domain name is called a ***site***. When you decide to buy books at the online site of bookseller, you would visit its Web site e.g. https://www.amazon.com/ ; the ***Web site*** is the location of a Web domain name in a computer somewhere on the internet. Another example of the web site is the university of technology / Iraq web site https://www.uotechnology.edu.iq/ .

- **Web pages - the documents on a Web site**: a Web site is composed of a web page or collection of related Web pages. A ***Web page*** is a document on the World Wide Web that can include text, pictures, sound, and video. The first page you see at a Web site is like the title page of a book. This is the ***home page***, or welcome page, which identifies the Web site and contains **links to other pages** at the site.
- **Search Engine** : is a ***software system*** that is designed to carry out web searches (Internet searches), which means to search the World Wide Web in a systematic way for particular information specified in a ***textual web search query***. The user can ***enter keywords*** or key phrases into the search engine that he wants to search for, and then the search engine will display the search results as websites, images, videos, or other online data. Common search engine are Google.com , Yahoo.com.

Goals for World Wide Web

The World Wide Web is the largest and the most commonly used hypermedia application. Its popularity is due to the ***amount of information available from web servers, the capacity to post such information, and the ease of navigating such information with a web browser***.

The W3C has listed the following ***goals*** for the WWW:

1. Universal access of web resources (by everyone everywhere).
2. Effectiveness of navigating available information.
3. Responsible use of posted material.

1.8 Hypermedia and Multimedia

- A **hypertext** system: We may think of a book as a linear medium, basically meant to be read from beginning to end, a hypertext meant to be ***read nonlinearly***, by following links that point to other parts of the document, or to other documents (Fig. 1.1)
- **Hypermedia:** not constrained to be text-based, can include other media, e.g., graphics, images, and especially the continuous media — sound and video (e.g. home pages contain a link to video, images, and sound).

Hypermedia can be considered one particular multimedia application.

*** The World Wide Web (WWW)** — the best example of a hypermedia application.

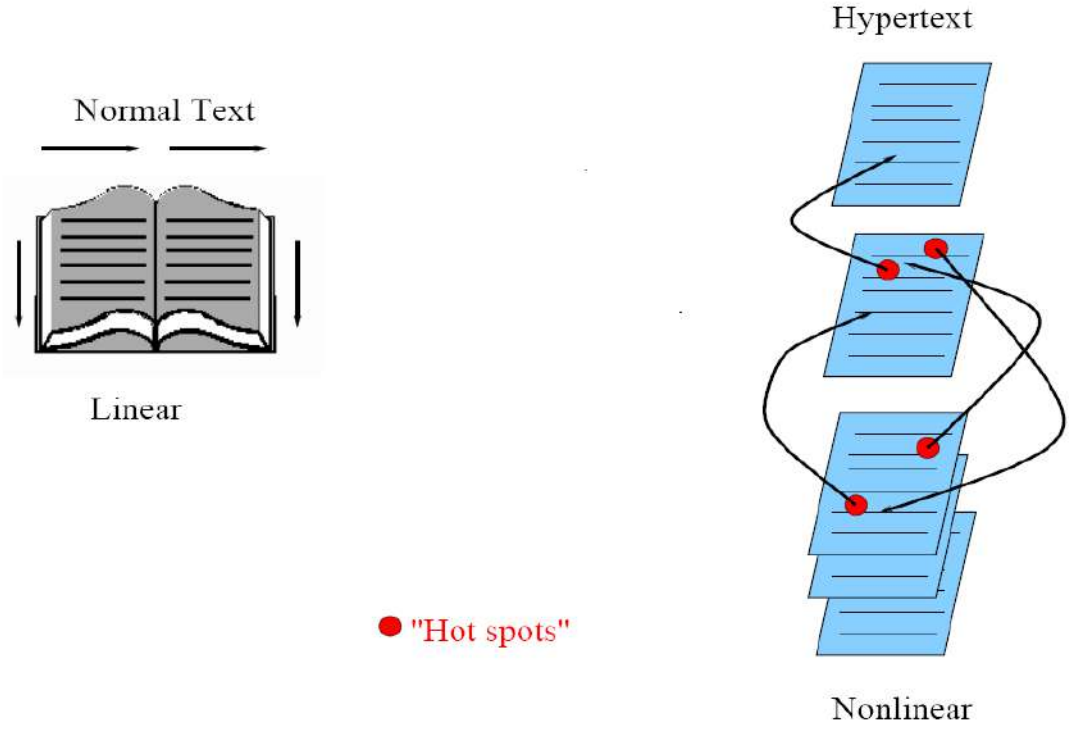


Figure 1.1 hypertext system

1.8.1 HyperText Transfer Protocol (HTTP)

HTTP (Hypertext Transfer Protocol) is the set of rules for **transferring files**, such as text, graphic images, sound, video, and other multimedia files, on the World Wide Web. As soon as a web user opens their web browser, the user is indirectly making use of HTTP. HTTP is an application-layer protocol that runs on top of the TCP/IP suite of protocols (the foundation protocols for the Internet).

The Hypertext Transfer Protocol (HTTP) is the foundation of the World Wide Web, and is used to load web pages using hypertext links. Hypertext Transfer Protocol is an application-layer protocol designed for transmitting hypermedia documents , it is an information systems that allows users to communicate data on the World Wide Web.

HTTP was invented alongside HTML to create the first interactive, text-based web browser: the original World Wide Web. Today, the protocol remains one of the primary means of using the Internet.

Information is exchanged between clients and servers in the form of Hypertext documents, from which HTTP gets its name. HTTP follows a classical client-server model, with a client opening a connection to make a request, then waiting until it receives a response. HTTP is a request-response protocol in the client-server computing model. Clients and servers communicate by exchanging individual messages. The message sent by the client, typically a Web Browser, is the request while the message sent by the server (Web server) as an answer is the response. The server will provide resources such as HTML files, which is which contain the information for formatting and displaying Web pages

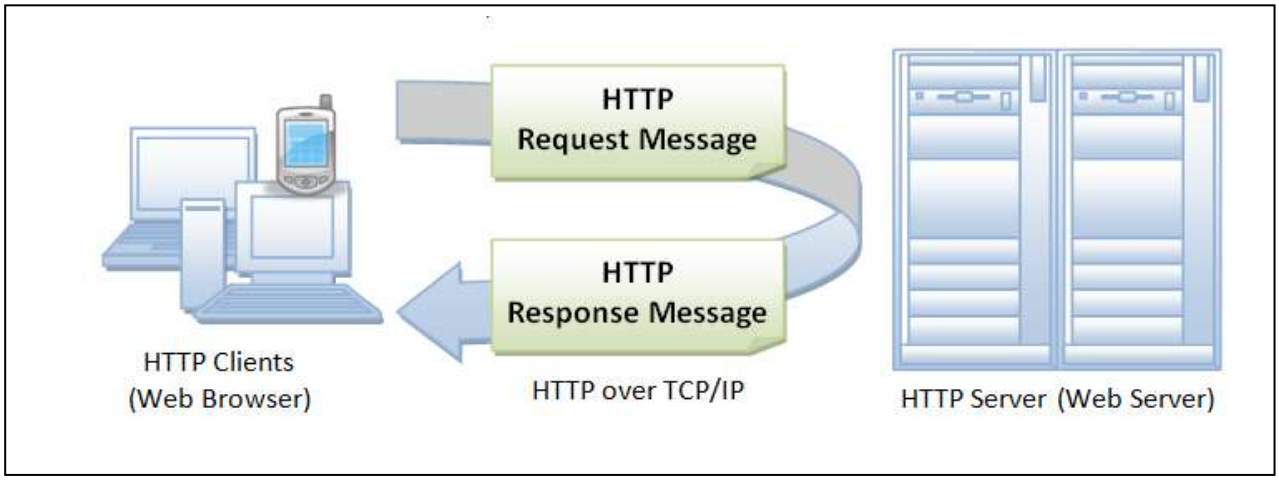


Figure 1.2 HTTP protocol

1.8.2 Hypertext Markup Language (HTML)

Hypertext is structured text that uses logical links, or hyperlinks, between nodes containing text. Hypertext documents can be manipulated using the Hypertext Markup Language (HTML). Using HTTP and HTML, ***clients*** can request different kinds of content (such as text, images, video, and application data) from web and application servers that host the content.

HTML is a language for publishing hypermedia on the World Wide Web. HTML is the standard markup language for **creating Web pages**. HTML is the basic building block of World Wide Web that **describes the structure of a Web page**. HTML consists of a series of elements. HTML elements tell the browser how to display the content. In other words, HTML is used to create electronic documents (called Web pages) that are displayed on the World Wide Web. Each page contains a series of connections to other pages called hyperlinks. Every web page you see on the Internet is written using one version of HTML code or another.

HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by scripting languages such as JavaScript. HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages.

HTML code ensures the proper formatting of text and images for your Internet browser. **Without HTML**, a browser would not know how to **display text as elements or load images** or other elements. HTML also provides a **basic structure** of the page. One could think of HTML as the bones (structure) of a web page.

Web browser receive HTML documents from a web server and render the documents into multimedia web pages. HTML describes the structure of a web page and originally included cues for the appearance of the document.

A **web browser** (commonly referred to as a browser) is a software application for accessing information on the World Wide Web.It allows the user to access websites and view web pages.

When a user requests a web page from a particular website, the web browser retrieves the necessary content from a web server and then displays the page on the user's device.

A web browser is **not the same thing** as a search engine, though the two are often confused. For a user, a search engine is just a website that

provides links to other websites. However, to connect to a website's server and display its web pages, a user must have a web browser installed.

Another difference between a browser and a search engine is that the browser is installed in the form of a local application on the user's device by the user himself, while the search engine is a software system that runs on the Internet and on all computers without prior installation of the user.

Today, the major web browsers are Google Chrome, Mozilla Firefox, Internet Explorer, Safari and Opera .

1.9 Multimedia on the Web

Many Web sites are multimedia, using a combination of text, images, sound, video or animation.

- **Text and images**: you can call up all kinds of text documents on the Web, such as newspapers, magazines, famous speeches, and works of literatures. You can also view image such as scenery, famous paintings, and photographs. Most Web pages combine both text and images.
- **Animation**: animation is a rapid sequencing of still images to create the appearance of motion. Animation is also used in online video games.
- **Video**: video can be transmitted in ***two ways***. A file, such as a movie or video clip, may have to be completely **downloaded** before you can view it. This may take several minutes in some cases. A file may be displayed as a streaming video and viewed while it is still being downloaded to your computer. **Streaming video** is the process of transferring data in continuous flow so that you can begin viewing a file even before the end is sent.

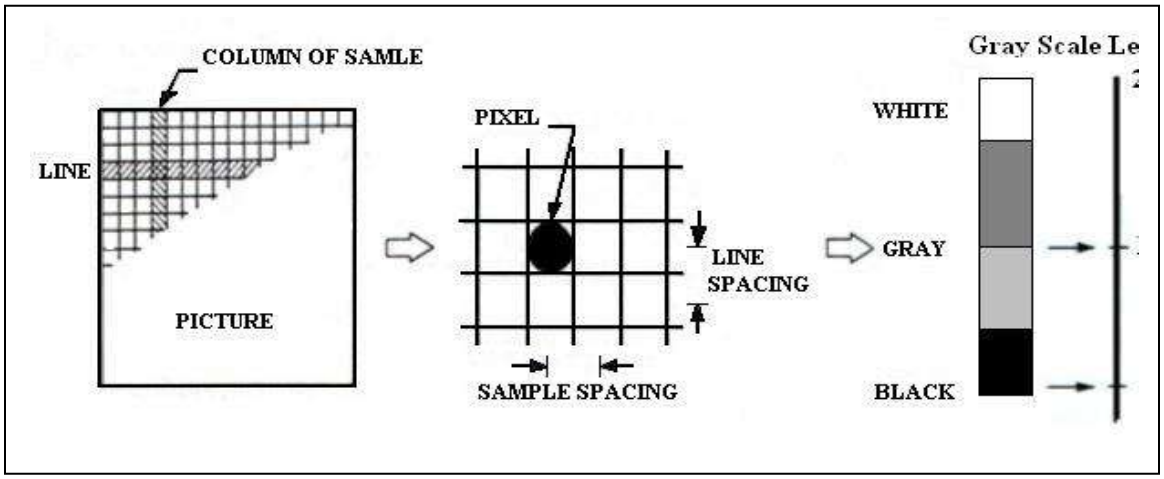
- **Audio**: audio, such as sound or music files, may also be transmitted in two ways: ***(1) downloaded completely*** before they can be played or ***(2) downloaded as a streaming audio***, allowing you to listen to the file while the data is still being downloaded to your computer.

Digital Image

1. Digital Images

An image must be converted from it analogue form to numerical form before processing. This conversion process is called ***digitization*** , and a common form is illustrated in Figure(1).The image is divided into small regions called ***picture elements***, or ***pixel*** for short . The most common subdivision scheme is the rectangular ***sampling*** grid shown in Figure(1).The image is ***divided into horizontal lines made up of adjacent pixels*** . At each pixel location, the image (samples) brightness is ***quantized***. This step ***generates an integer at each pixel representing the brightness or darkness of the image at that point***. When this has been done for all pixel, ***the image is represented by a rectangular array of integers.*** Each pixel has a location or ***address*** (***Line or row number and sample or column number***) and an integer value called ***gray level***. This array of digital data is now a candidate for computer processing.

From above we can define ***Digital Image*** as a ***sampled, quantized function of two dimensions f(x,y)*** which has been generated by optical means, sampled in an equally spaced rectangular grid pattern , and quantized in equal intervals of gray levels.



Figure(1) Digitizing an Image

Thus a digital image is now a two-dimensional rectangular array of quantized sample value.

1.1 Sampling

The process of ***creating a digital image*** from data acquired by a camera or some other kind of imaging instrument requires a ***two-dimensional*** pattern to represent the measurements (light intensity or color) that are made in the form of an image numerically.

The pattern can be described by a ***function f(x,y)***. For ***monochrome image***, the ***value*** of the function at any pair of coordinates, *x* and *y* is the ***intensity of the light*** detected at that point. In the case of ***color images*** *f(x,y)* is a ***vectored-value function***

The function *f(x,y)* must be translated into a discrete array of numerical data if it is to undergo computer processing. Translation of *f(x,y)* into an appropriate numerical form is accomplished by the process of sampling and quantization

Sampling: is the two dimensional pattern that is required to represent the image measurements (light intensity or color).

Or it is a process of measuring the value of the image function *f(x,y)* at discrete ***intervals in space***.

Each sample corresponds to a small square area of the image, known as a pixel. A digital image is a two-dimensional array of these pixels. Pixels are indexed by *x* and *y* coordinates, with *x* and *y* taking integer values.

Spatial Resolution

The spatial resolution of an image is the ***physical size*** of a pixel in that image; i.e., the ***real area*** in the scene that is represented by a ***single pixel*** in the image.

For a given field of view, ***dense sampling*** will produce a **high resolution** image in which there are many pixels, each of which represents the contribution of a very small part of the scene; ***coarse sampling***, on the other hand, will produce a **low resolution** image in which there are few pixels, each representing the contribution of a relatively large part of the scene to the image.

Spatial resolution dictates the amount of useful information that can be extracted from an image.

1.2. Quantization

If the image to be processed is analog (i.e., a ***voltage*** that changes with time) ***quantization*** is used to ***digitize it into integer numbers***.

It is usual to digitize the value of the image function *f(x,y)* in addition to its spatial coordinates. The process of quantization ***involves*** replacing a continuously varying *f(x,y)* (analog signals) with a discrete set of quantization levels (digital numbers) . The accuracy with which variations in *f(x,y)* are represented is determined by the number of ***quantization levels*** that we use : ***the more levels we use, the better the approximation.***

Quantization: is the representation of the brightness of each pixel by an integer value. Since digital computer process number, it is necessary to

reduce the continuous measurement value to discrete units and represent them by integer numbers.

Conventionally, a set of *n* quantization levels compromises the integers 0,1,2,……, *n-1*. ***0 and n-1*** are usually displayed or printed as ***black and white***. Respectively, with ***intermediate levels*** rendered in various ***shades of gray***. Quantization levels are therefore commonly referred to as ***gray levels***. The collective term for all the gray levels ranging from black to white, is a gray scale.

2. Image Representation

As we know, the human visual system receives an input image as a collection of spatially distributed ***light energy***; this form is called an ***optical image***. Optical images are the types we deal with everyday - cameras capture them, monitors display them, and we see them. We know that these ***optical images*** are represented as video information i***n the form of analog electrical signals and have seen how these are sampled to generate the digital image*** *f(x,y)*.

The digital image f(x,y) is represented as a two-dimensional array of data, where each pixel value corresponds to the brightness of the image at the point (r, c). In linear algebra terms, a two-dimensional array like our image model *f(x,y)* is referred to as a matrix, and one row (or column) is called a vector. This image model is for ***monochrome*** (one-color, this is what we traditionally refer to as black and white) image data, we also have other types of image data that require extensions or modifications to this model. Typically, these are multiband images (***color, multispectral***), and they can be modeled by a different *f(x,y)* function corresponding to each

separate band of brightness information. The image types we will consider are : 1) binary, 2) gray-scale, 3) color, and 4) multispectral.

2.1 Binary Images

Binary images are the simplest type of images and can take on two values, typically black and white, or '0' and '1' A binary image is referred to as a ***1 bit/pixel*** image because it takes only 1 binary digit to represent each pixel.

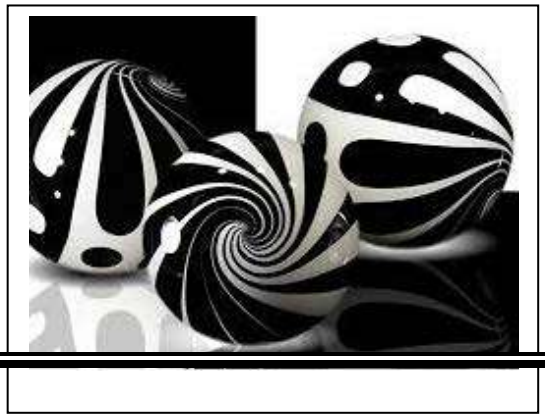
Binary image applications

These types of images are most frequently used in computer vision applications where the only information required for the task is **general shape or outline** information, For example:

1. To position a robotic gripper to grasp an object, to check a manufactured object for deformations, since robot has to work in **real time** , and working with Binary image decrease processing time.
2. In optical character recognition (OCR).
3. In text images.

Advantage of binary images

1. Reduce storage space (memory).
2. Decrease processing time.
3. Required less transmission time.



Binary images are often created from gray-scale images via a ***threshold*** operation where every pixel above the threshold value is turned white ('1'), and those below it are turned black ('0').

2.2 Gray-Scale Images

Gray-scale images are referred to as ***monochrome***, or one-color, images. They contain ***brightness information*** only, no color information. The number of bits used for each pixel determines the number of different brightness levels Available. The typical image contains ***8 bits/pixel*** data, which allows us to have ***256 (0-255) different brightness*** (gray) levels.

This representation provides more than adequate brightness resolution, in terms of the human visual system's requirements and provides a "noise margin" by allowing for approximately twice as many gray levels as required. Additionally, the ***8 bit representation*** is typical due to the fact that the byte which corresponds to 8 bit of data is the ***standard small unit*** in the world of digital computers.

2.3 Color Images

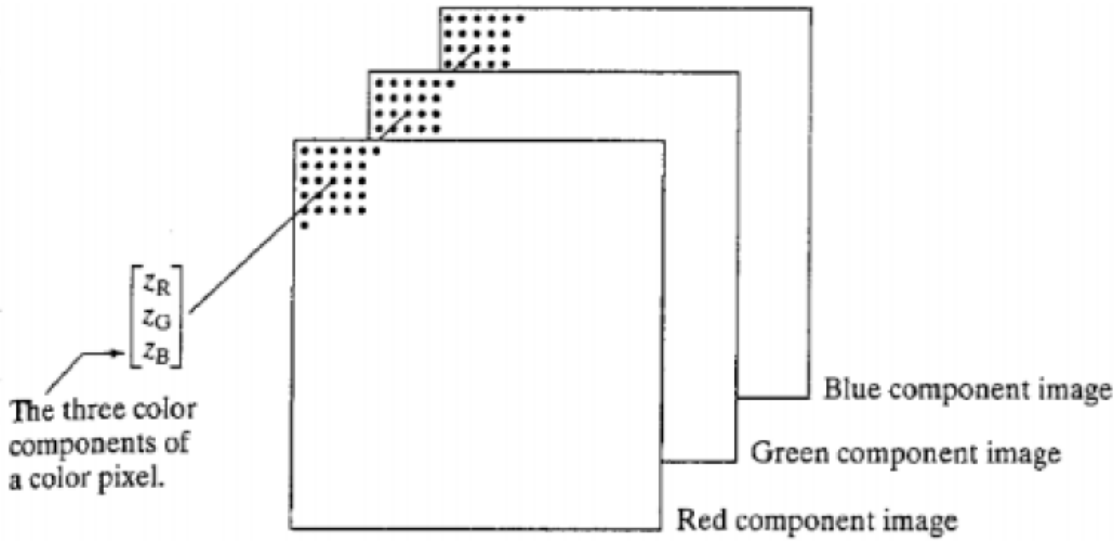
The use of color is **important** in image processing because:

1. Color is a ***powerful descriptor*** that simplifies object identification and extraction.
2. Humans can ***discern thousands of color*** shades and intensities, compared to about only two dozen shades of gray.

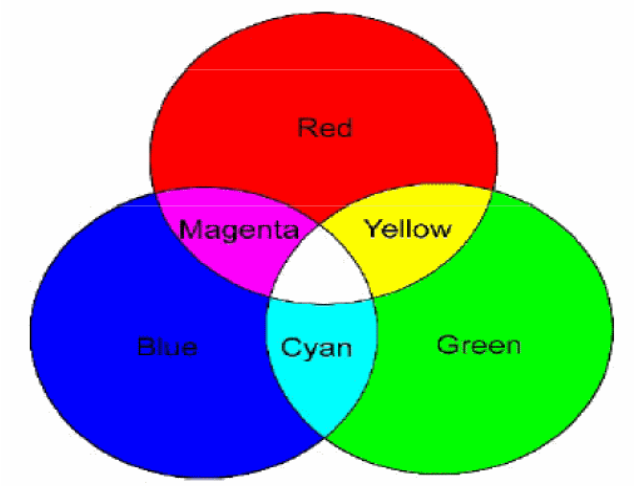
2.3.1 RGB Color Model

Color images can be modeled as three-band monochrome image data, where each band of data corresponds to a different color. The actual information stored in the digital image data is the brightness information in each spectral band. When the image is displayed, the corresponding brightness information is displayed on the screen by picture elements that emit light energy corresponding to that particular color ***which is generated from the mixture of those three bands.***

Typical color images are represented as Red, Green, and Blue, or RGB images. Using the 8-bit monochrome standard as a model, the corresponding color image would have 24 bits/pixel 8-bits for each of the three color bands (Red, Green, and Blue). In Figure (2) we see a representation of a typical RGB color image. Figure (2) illustrates that, in addition to referring to a row or column as a vector, we can refer to a single pixel's Red, Green, and Blue values as a color pixel vector (R,G,B).



(a)



(b)

Figure 2: (a) Scheme of RGB color image (b) Primary and secondary colors.

2.3.2 **HSV Color Model**

For many applications, RGB color information is transformed into a mathematical space that ***decouples*** the brightness information from the color information. After this is done, the image information consists of a ***one-***

dimensional brightness or luminance space and a ***two-dimensional color*** or chrominance space. Now the two-dimensional color space ***does not contain any brightness*** information; but it typically contains information regarding the relative amounts of the different colors. An additional ***benefit*** of modeling the color information in this manner is that it ***creates a more people-oriented way of describing the colors.***

The **H**ue/**S**aturation/**V**alue (HSV color transform allows us to describe colors in terms that became more readily understand (see Figure (3). The ***value*** is the brightness of the color, and the ***hue*** is what we normally think of as "color" (for example green, blue, or orange). The ***saturation*** is a measure of how much white is in the color (for example, pink is red with more white, so it is less saturated than a pure red).

Hue: The “true color” attributes (red, green, blue, orange, yellow, pink,…).

Saturation: The amount by which the color has been diluted with white. The ***more white*** in the color, the ***lower the saturation***. So a **deep red** has **high saturation**, and a **light red** (a pinkish color) has **low saturation**.

Value: The degree of ***brightness***, a **well-lit color** has **high intensity**; a **dark color** has **low intensity**.

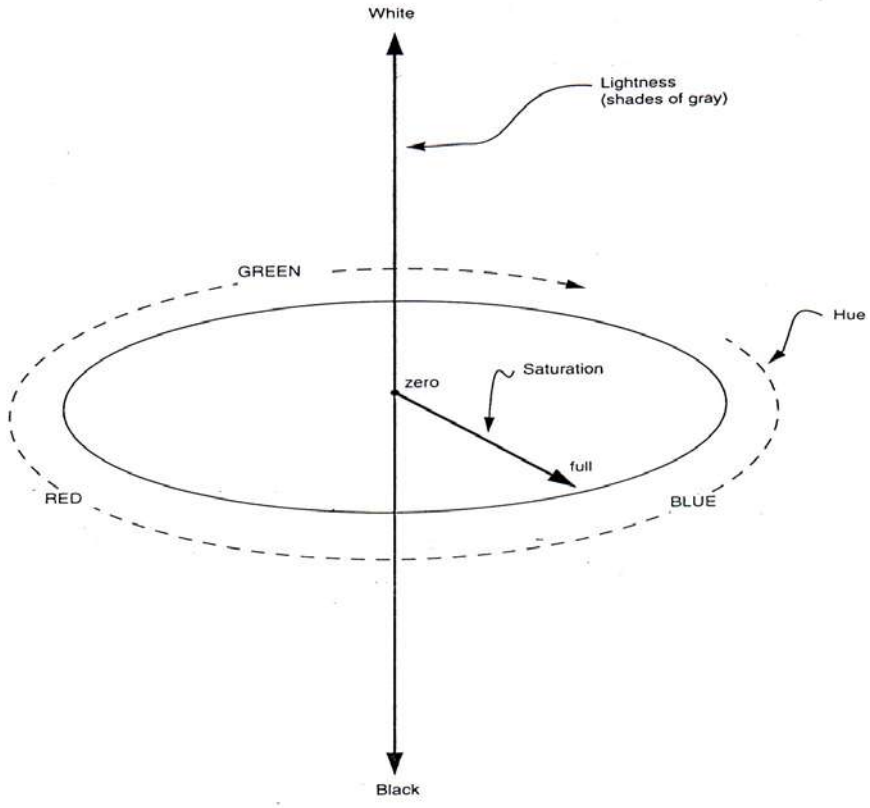
HSV color space is the ***most intuitive*** way of ***describing color***. This color system is very ***nearer*** than the RGB system to the approach in which ***humans express color sensations***. For example : "a deep, bright orange" would a have a large intensity ("bright"), a hue of "orange" and a high value of saturation ("deep"). We can picture this color in our minds, but if we defined this color in terms of its RGB component R=245, G=110, and B = 20, most people would have no idea how this color appears. Because the

HSV color space ***was developed based on heuristics relating to human perception***, various methods are available to transform RGB pixel values into the HSV color space. Most of these are algorithmic in nature and are geometric approximations to mapping the RGB color cube into some HSV-type color space.

$$H = \arccos \frac{2R - G - B}{2\sqrt{(R-G)^2 + (R-B)(G-B)}} \quad \text{..... (1)}$$

$$S = 1 - \frac{3 * \min(R,G,B)}{R+G+B} \quad \text{..... (2)}$$

$$L = \frac{R+G+B}{3} \quad \text{..... (3)}$$



Figure(3): The HSV color space

2.3.3 8-bits color images

GIF (Graphics Interchange Format) file format is commonly ***used in the World Wide Web***. GIF files are limited to a maximum of ***8 bit\pixel*** and allow for a type of compression called **LZW** (Lepel-Ziv-Welch). The 8 bit\pixel limitation does not mean that it does not support color images, it simply means that ***no more than 256 colors*** (2^8) are allowed in an image. This is typically ***implemented*** by mean of ***lookup table (LUT)***, where the ***256 colors are stored in a table***, and ***1 byte (8 bits)*** is used as an ***index (address)*** into that table for each pixel.

GIF images are indexed images where the colors used in the image are stored in a **Palette, sometimes** referred to as a color look-up-table. ***Each pixel is represented as a single byte, and the pixel data is an index to the color palette.*** The color of the palette is typically ***ordered*** from the ***most used*** color to the ***least used*** colors to ***reduce*** look-up ***time***.

Color Look-Up Tables (LUTs)

- ***Store*** only the ***index*** of the color LUT ***for each pixel***.
- ***Look up*** the table to ***find the color (RGB) for that index***.
- LUT needs to be built when converting 24-bit color images to 8-bit : ***grouping similar colors*** (each group assigned a color entry)
- Possible for palette animation by changing the color map.

8-bit index	Red	Green	Blue
0	R_0	G_0	B_0
1	R_1	G_1	B_1
2	R_2	G_2	B_2
:	:	:	:
:	:	:	:
254	R_{254}	G_{254}	B_{254}
255	R_{255}	G_{255}	B_{255}

color LUT for 8-bit Color Images

2.4 Multispectral images

Visible light is part of the ***electromagnetic*** spectrum: radiation in which the ***energy*** takes the form of waves of ***varying wavelength***. The values for the wavelengths of blue, green and red were set in 1931 by the CIE (Commission International d'Eclairage), an organization responsible for color standards. Figure 4 illustrates this.

Multispectral images typically contain information ***outside*** the normal human perceptual range. This may **include** infrared, ultraviolet, X-ray, radar data. These are ***not images in the usual sense*** because the information represented is ***not directly visible*** by the human visual system. ***However, the information is often represented in visual form by mapping the different***

spectral bands to RGB components. If more than three bands of information are in the multispectral image, the dimensionality is reduced by applying a principal component's transform.

Sources for these types of images include ***satellite systems, underwater sonar system, various types of airborne radar, infrared imaging systems and medical diagnostic imaging systems***. The number of bands into which the data are divided is strictly a function of the sensitivity of the imaging sensors used to capture the images. For example, even the visible spectrum can be divided into many more than three bands; three are used because this mimics our visual system. ***Most of the satellites currently in orbit collect image information in two to seven spectral bands typically one to three are in the visible spectrum***, ***one or more are in the infrared region***, and ***some have sensors that operate in the radar range*** (see Figure 4). ***The newest satellites have sensors that collect image information in 30 or more bands.*** As the amount of data that needs to be transmitted, stored, and processed increases, the importance of topics such as compression becomes more and more apparent.

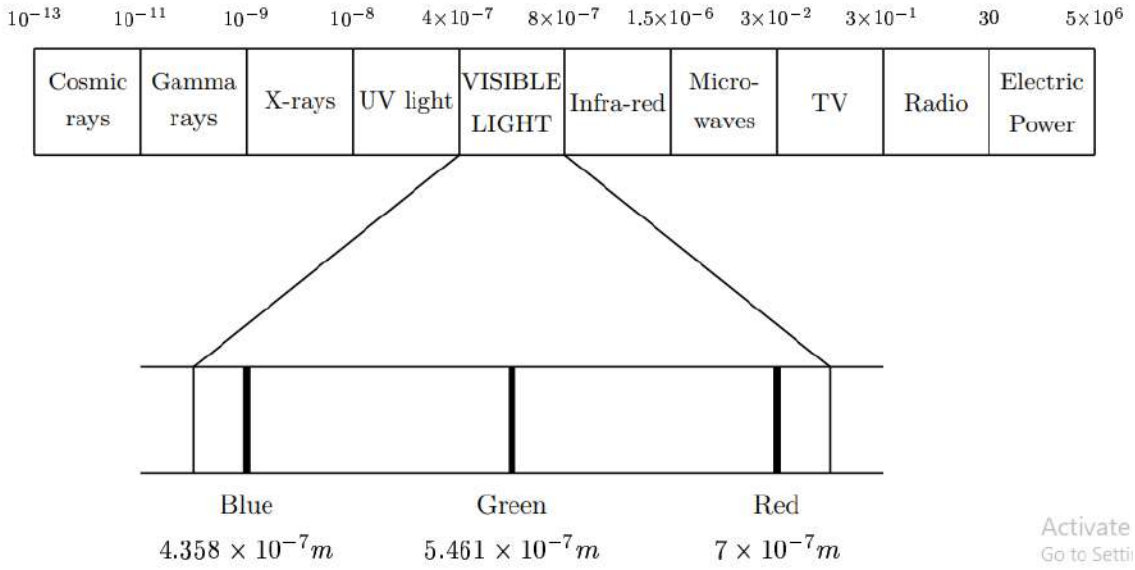
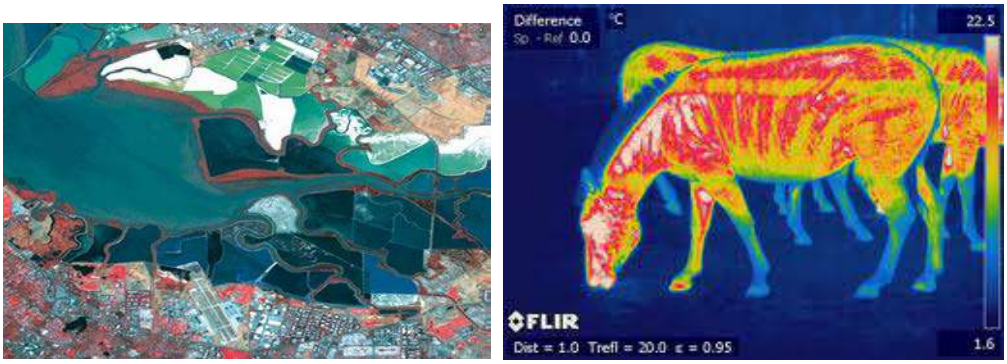


Figure (4) : The spectrum of electromagnetic radiation



A Multi spectral Image

3. Arithmetic and Logical Operations on Images (Image Algebra)

These operations are applied on ***pixel-by-pixel*** basis. So, to add two images together, we add the value at pixel (0 , 0) in image 1 to the value at pixel (0 , 0) in image 2 and ***store the result in a new image*** at pixel (0 , 0). Then we move to the next pixel and repeat the process, continuing until all pixels have been visited.

Clearly, this can work properly only if the two images have ***identical dimensions.*** If they do not, then combination is still possible, but a ***meaningful result*** can be obtained only in ***the area of overlap***. If our images have dimensions of w_1*h_1, and w_2*h_2 and we assume that their origins are aligned, then the new image will have dimensions w*h, where:

$w = \min (w_1, w_2)$

$h = \min (h_1, h_2)$

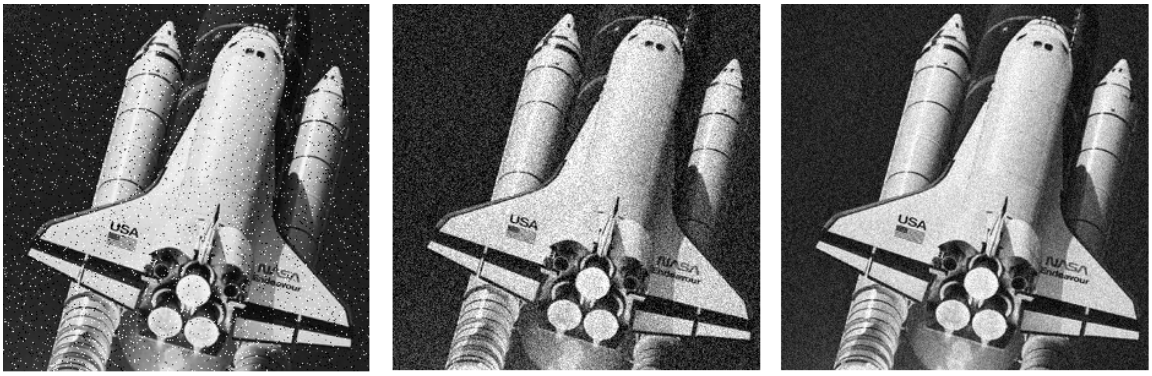
In the case of arithmetic operations, we must also ensure that the ***representation used*** for the output image is ***appropriate for the operation*** being performed. For example, ***the values produced when we add two 8-bit grey scale image, cannot , in general, be contained in an 8-bit range.***

3.1 Addition and Averaging

Adding two 8-bit grayscale images can produce pixels values in ***the range*** **0-510.** Therefore a **16-bit** ***representation*** must be chosen for the output image **or** ***divide every pixel's value by two***. If we do the later, then we are computing an ***average*** of the two images.

The main application of image averaging is ***noise removal***. Every image acquired by a real sensor is afflicted to some degree by ***random noise***. However, the level of noise represent in the image can be reduced, provided that the scene is static and unchanging, by the averaging of multiple observations of that scene. This works because the noisy distribution can be regarded as approximately symmetrical with a mean of zero. As a result, positive perturbations of a pixel's value by a given amount are just as likely

as negative perturbations by the same amount, and there will be a tendency for the perturbations to cancel out when several noisy values are added.



(a) **(b)** **(c)**

Figure (5) a) noisy image b) average of five observation c) average of ten observation

Addition can also be used to ***combine the information of two images***, such as an ***image morphing and motion pictures***.

morphing: blending between two photographs

Algorithm 1: image addition

read input-image1 into in-array1;

read input-image2 into in- array2;

for i = 1 to no-of-rows do

for j=1 to no-of-columns do

begin

out-array (i,j) = in-array1(i,j) + in-array2(i,j);

if (out-array (i,j) > 255) then out-array (i,j) = 255;

end

write out-array to out-image;

3.2 Subtraction

Subtracting two 8-bit grayscale images can produce values between **-225 and +225.** This necessitates the use of ***16-bit signed integers*** in the output image – unless ***sign is unimportant***, in which case we can simply take the ***modulus of the result*** and store it using 8-bit integers:

$$g(x,y) = |f_1(x,y) - f_2(x,y)|$$

The main application for image subtraction is in ***change detection*** (or ***motion detection***). If we make two observations of a scene and compute their difference using the above equation, then ***changes will be indicated*** by pixels in the difference image which have ***non-zero values***. Sensor noise, slight changes in illumination and various other factors can result in small differences which are of no significance so it is usual to apply a threshold to the difference image. Differences below this threshold are set to zero. Difference above the threshold can, if desired, be set to the maximum pixel value. Subtraction can also be used in ***medical imaging to remove static background information***.

Algorithm2: image subtraction

read input-image1 into in-array1;

read input-image2 into in- array2;

for i = 1 to no-of-rows do

```
for j=1 to no-of-columns do
  begin
    out-array (i,j) = in-array1(i,j) - in-array2(i,j);
    if ( out-array (i,j) < 0 )  then   out-array (i,j) = 0;
  end
```

Figure (6) a, b) two frames of video sequence c) their difference

3.3 Multiplication and Division

Multiplication and division can be used to ***adjust brightness*** of an image. **Multiplication** of pixel values by a ***number greater than one will brighten the image***, and **division** by a ***factor greater than one will darken the image***. Brightness adjustment is often used as a ***preprocessing step in image enhancement.***

One of the principle uses of image multiplication (or division) is to ***correct grey-level shading*** resulting from non uniformities in illumination or in the sensor used to acquire the image.

(a) **(b)** **(c)**

Figure (7) a) original image b) image multiplied by 2 c) image divided by 2

3.4 Logical Operation:

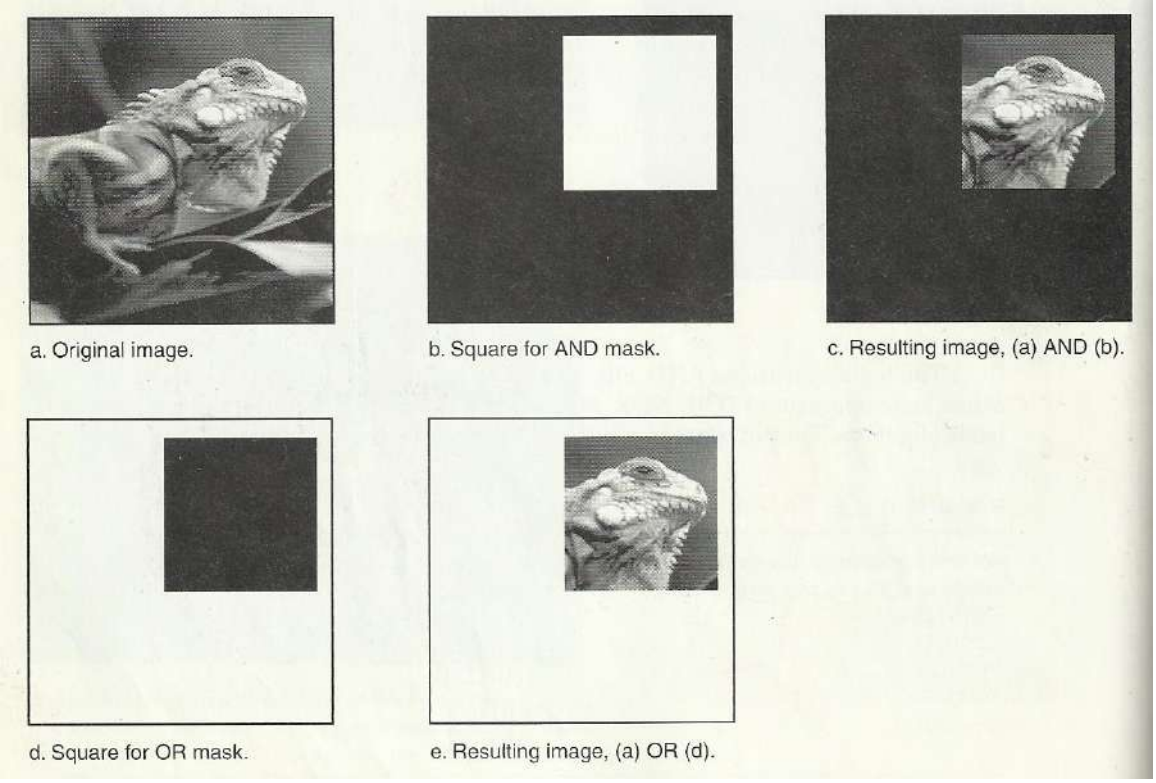
Logical operations apply ***only to binary images***, whereas arithmetic operations apply to ***multi-valued pixels***. Logical operations are basic tools in binary image processing, where they are used for tasks such as ***masking***, ***feature detection***, ***shape analysis*** and ***obtain the similarity or difference between two templates***. Logical operations on entire image are performed **pixel – by – pixel**. Because the AND operation of two binary variables is 1 only when both variables are 1, the result at any location in a resulting AND image is 1 only if the corresponding pixels in the two input images are 1. As logical operation involve only one pixel location at a time, they can be done in place, as in the case of arithmetic operations. The XOR (exclusive OR) operation yields a 1 when one or other pixel (but not both) is 1, and it yields a 0 otherwise. The operation is unlike the OR operation, which is 1, when one or the other pixel is 1, or both pixels are 1.

	AND				OR				XOR			
Input	1	0	1	0	1	0	1	0	1	0	1	0
Mask	1	1	0	0	1	1	0	0	1	1	0	0
output	1	0	0	0	1	1	1	0	0	1	1	0

Logical **AND & OR** operations are useful for the ***masking and compositing*** of images. For example, if we compute the **AND** of a binary image with some other image, then pixels for which the corresponding value in the binary image is **1** will be ***preserved***, but pixels for which the corresponding binary value is **0** will be set to 0 ***(erased)*** . Thus the binary image acts as a "***mask***" that ***removes*** information from certain parts of the image.

On the other hand, if we compute the **OR** of a binary image with some other image , the pixels for which the corresponding value in the binary image is **0** will be ***preserved***, but pixels for which the corresponding binary value is **1**, will be set to 1 (***cleared***).

So, masking is a simple method to extract a region of interest from an image.



Figure(8) : image masking

In addition to masking, logical operation can be used in feature detection. Logical operation can be used to ***compare between two images***, as shown below:

AND ^

This operation can be used to find the ***similarity between white regions*** of two different images (it required two images).

g (x,y) = a (x,y) ^ b (x,y)

Exclusive OR ⊗

This operator can be used to find the ***differences between white regions*** of two different images (it requires two images).

g (x,y) = a (x,y) ⊗b (x,y)

NOT

NOT operation can be performed on ***grey-level*** images, it's applied on only one image, and the result of this operation is the ***negative*** of the original image.

g (x,y) = 255- f (x,y)

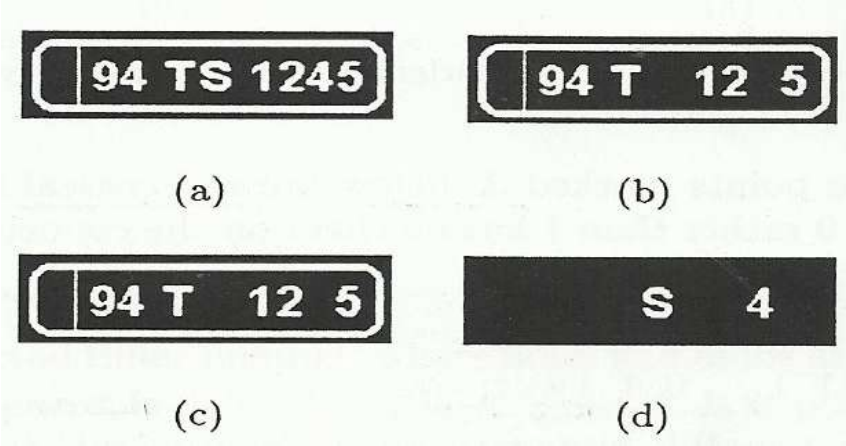


Figure (9) a) input image a(x,y) b) input image b(x,y)

c) a(x,y) ^ b(x,y) d) a(x,y) ⊗b(x,y)

4. Image Histogram

The gray level ***histogram*** is a function showing, for each gray level, the ***number of pixels*** in the image that have ***that gray level***. The abscissa is gray level and ordinate is the frequency of occurrence (number of pixels). This function summarizes the gray level counted of an image. While the histogram of any image contains considerable information. ***Certain types of images are completely specified by their histograms***.

The histogram of an image records the frequency distribution of gray levels in the image. The histogram of an ***8-bit image*** can be thought of as a ***table with 256 entries, or "bins", indexed from 0 to 255***. In bin 0 we record the number of times a gray level of 0 occurs; in bin 1 we record the number of times a gray level of 1 occurs, and so on, up to bin 255.

Algorithm 3 shows how we can accumulate in a histogram from an image. Figure 10 shows an image and its histogram computed using this algorithm.

ALGORITHM 3: Calculating of an image Histogram

Create an array histogram with 2^8 elements.

For all gray levels, I, do

Histogram [I] =0

End for

For all pixels coordinates, x and y, do

Increment histogram [f (x,y)] by 1

End for

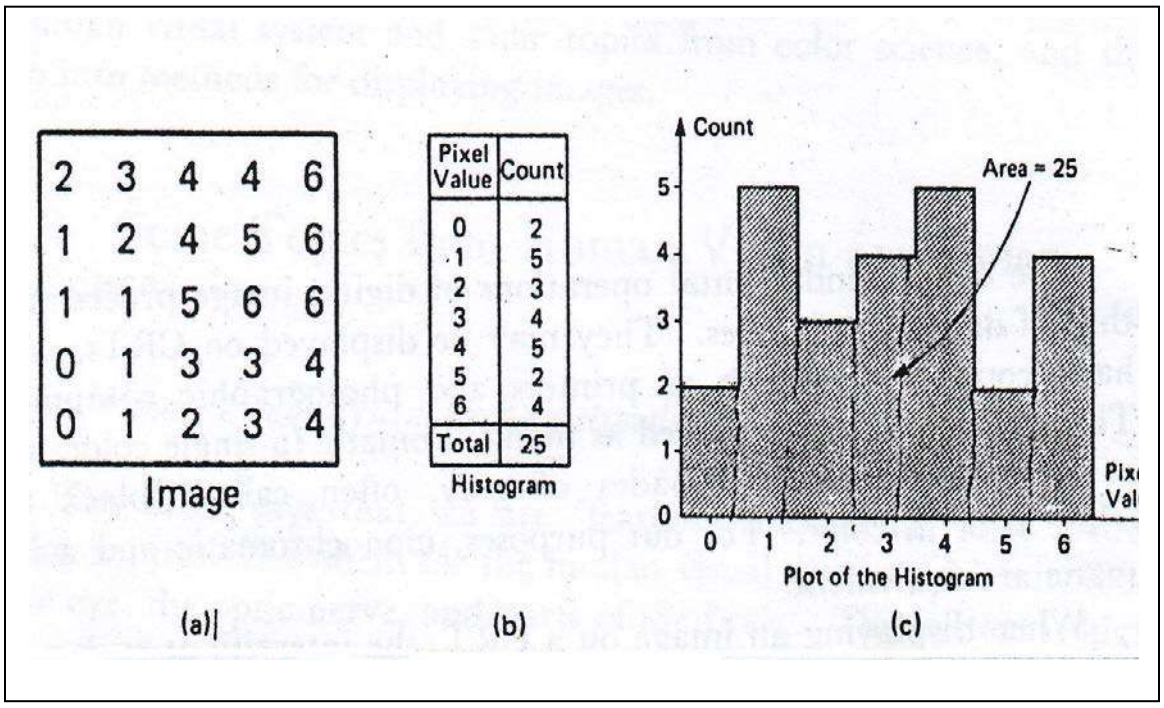
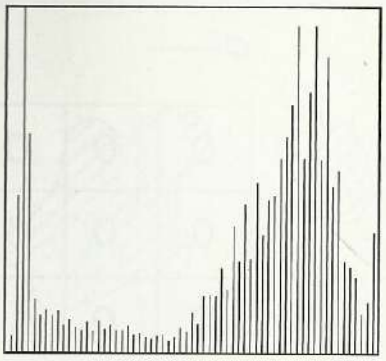


Figure 10: Subimage and its histogram

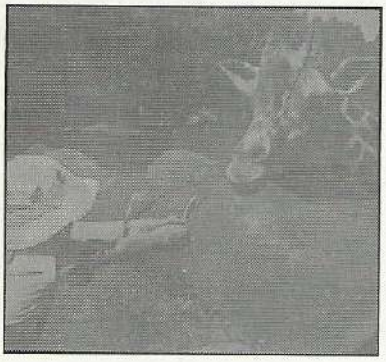
The ***shape of the histogram*** provides us with information about the nature of the image, or sub image if we are considering an object with the image. For example, a ***very wide*** histogram implies a high contrast, a ***very***

narrow histogram implies a low contrast, a histogram ***skewed toward the right*** implies a bright image, a histogram ***skewed toward the left*** implies a dark image, and a histogram with ***two major peaks***, implies an object that in contrast with the background.

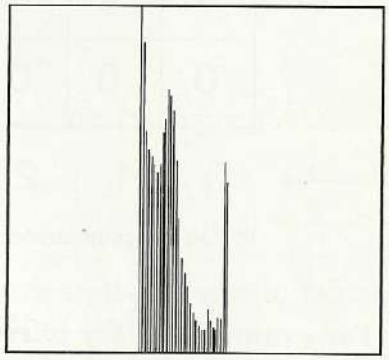
a. Object in contrast with background.



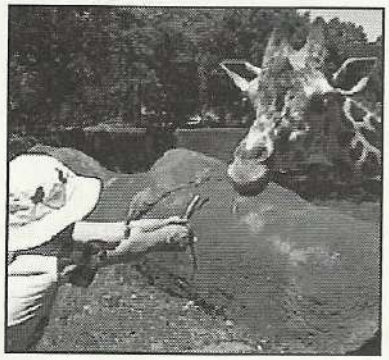
b. Histogram of (a) shows bimodal shape.



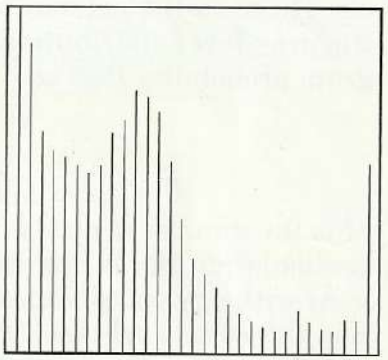
c. Low-contrast image.



d. Histogram of (c) appears clustered.



e. High-contrast image.



f. Histogram of (e) appears spread out

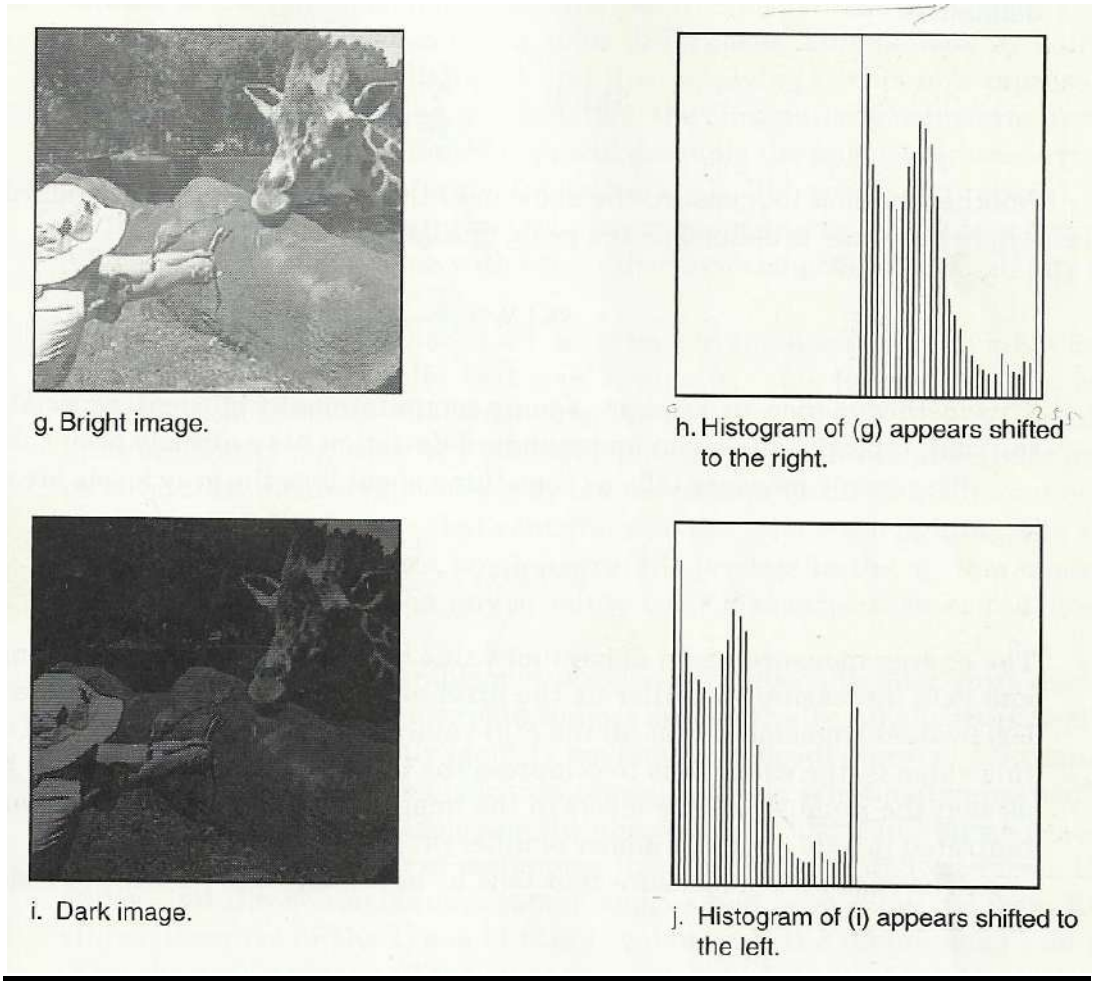


Figure 11 : a variety types of histograms

Probabilistic Histogram

We can ***normalize*** a histogram by ***dividing the counts in each bin*** by the ***total number of pixels*** in the image associated with that histogram. This gave us a table of estimated probabilities. i.e. probability density function (pdf) of the image. Hence, the entry for any gray level tells us the ***likelihood*** of finding that gray level at pixel selected randomly from the image.

However, probabilistic histogram should be used when ***comparing*** the histograms of images with ***different sizes*****.**

4.1 Properties and Usage of histogram

One of the ***principle uses*** of the histogram is in the selection of ***threshold*** parameter.

Histogram can be considered as the ***first step*** in ***image matching***, i.e. it is used as a dimensionality reduction technique.

The histogram of an image provides a ***useful indication*** of the relative importance of different gray levels in an image; indeed, it is sometimes possible to ***determine*** whether ***brightness*** or ***contrast adjustment*** is necessary merely by ***examining*** the ***histogram*** and ***not the image*** itself.

The histogram provides sufficient characteristics such as ***invariant to translation and rotation*** of the image, besides ***normalizing the histogram*** leads to achieve invariant properties ***against the scaling*** effect of the image.

When an image is condensed into a histogram, ***all spatial information is discarded*****.** The histogram specifies the number of pixels having each gray level but ***gives no hint*** as to ***where those pixels are located*** within the image. Thus, the histogram is ***unique*** for any particular image, but the ***reverse is not true***. Vastly different images could have identical histograms. Such operations as ***moving objects around within an image*** typically have ***no effect*** on the histogram.

This is evident from figure **12** which shows two very different images that have identical histograms. Although a histogram gives us the frequency distribution of gray levels in an image. It can ***tell us nothing*** about the way in which gray levels are ***distributed spatially***.

Gray level mapping operations affect the histogram of an image in predicable ways. For example, ***adding*** a constant bias to gray levels will ***shift*** a histogram along the gray level axis without changing its shape. ***Multiplication*** of gray levels by a constant gain will ***spread out*** the histogram evenly if $a > 1$, increasing the spacing between occupied bin , or c***ompress*** the histogram it if $a < 1$, which can have the effect of merging bins .

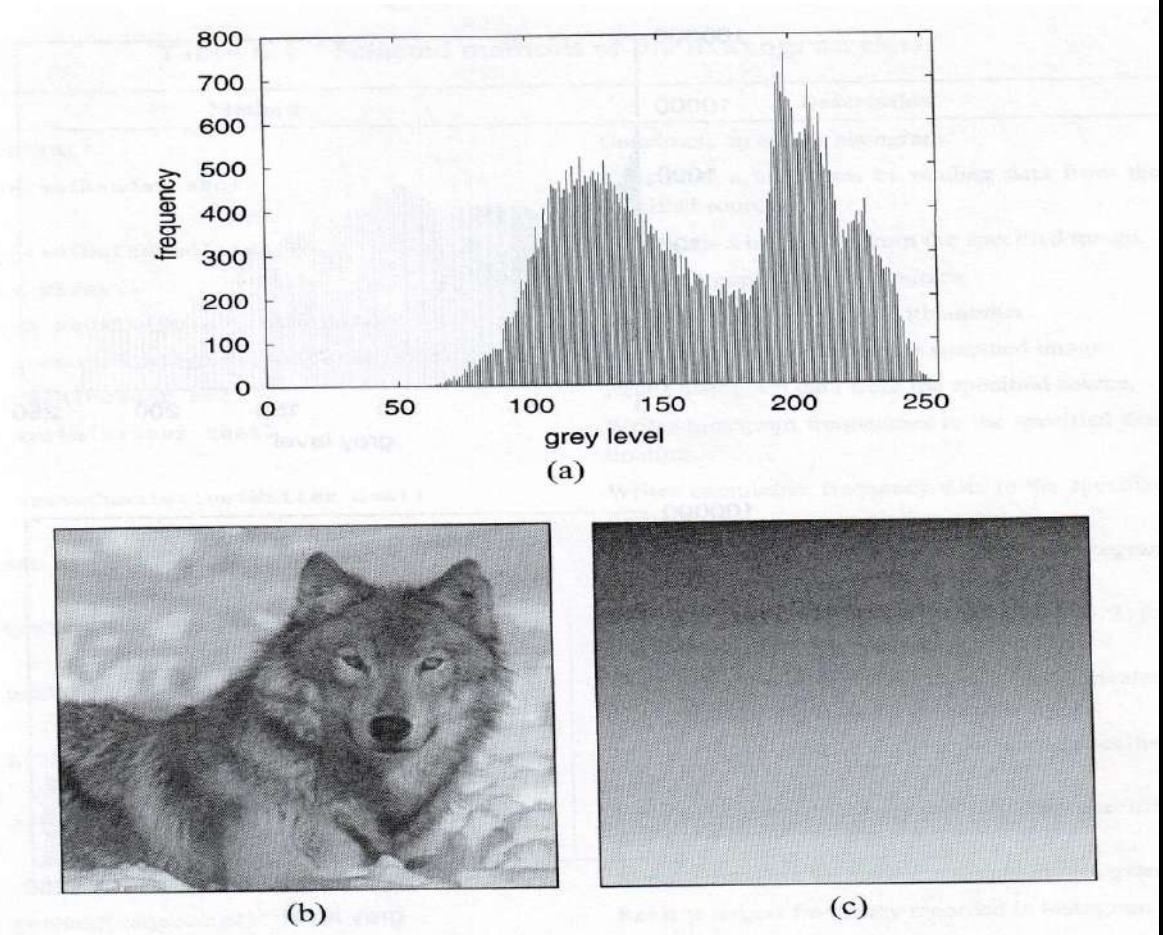


Figure 12: non-uniqueness of a histogram

(a) a histogram (b) an image with (a) as its histogram
(c) a different image with the same histogram

4.2 Histogram modification

An alternate perspective to gray-level modification that performs a similar function is referred to as histogram modification. The gray-level histogram of an image is the distribution of the gray levels in an image. In figure 11 we have see an image and its corresponding histogram. In general a histogram with a ***small-spread*** has a ***low-contrast***. And a histogram with a ***wide spread*** has a ***high contrast***, whereas an image with its histogram clustered at the ***low end*** of the range is ***dark***, and a histogram with the values clustered at the ***high end*** of the range corresponds to a ***bright*** image.

Note The ***main*** effect of the histogram modification operations is on the image histogram itself, but ***it's also effect*** on image contrast (histogram stretch, histogram shrink) and on the image brightness (histogram slide).

The histogram can also be modified by a mapping function, which will stretch, shrink (compress), or slide the histogram. Histogram stretching and histogram shrinking are forms a gray-level modification, sometimes referred to as histogram scaling. In figure **13** we see a graphical representation of histogram stretch, shrink, and slide.

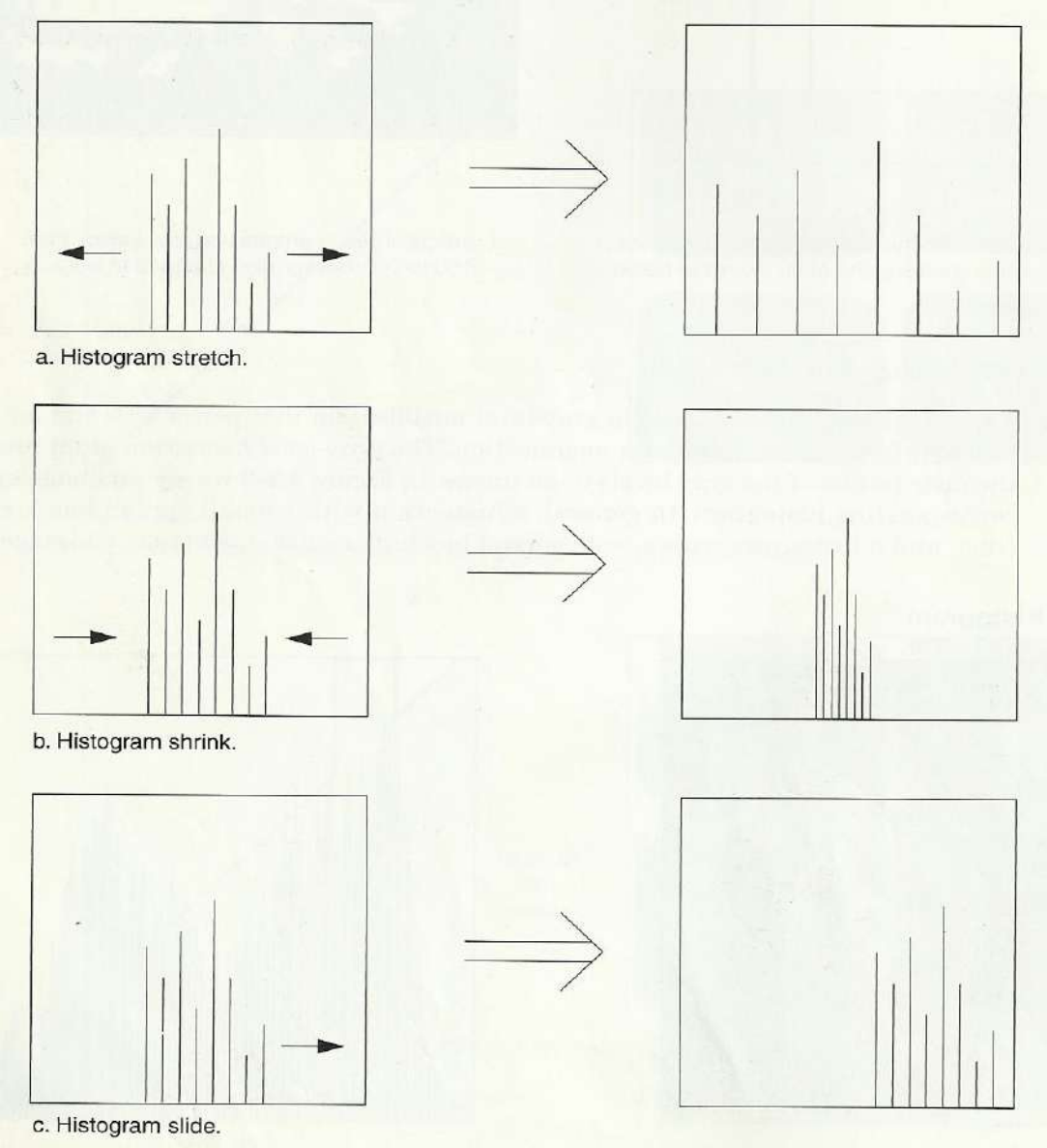


Figure 13 : histogram modification

a. Histogram Stretch

The mapping function for histogram stretch can be found by the equation:

$$\text{Stretch}(I(r,c)) = \left[\frac{I(r,c) - I(r,c)_{\text{MIN}}}{I(r,c)_{\text{MAX}} - I(r,c)_{\text{MIN}}}\right][\text{MAX} - \text{MIN}] + \text{MIN}$$

Where : $I(r,c)_{MAX}$ is the largest gray-level value in the image I(r,c)

$I(r,c)_{MIN}$ is the smallest gray-level value in the image I(r,c)

MAX and MIN correspond to the maximum and minimum gray-level values ***desired*** in the stretched histogram (for 8-bit images the typical range is between 0 and 255).

This equation will take an image and stretch the histogram across the entire gray-level range, which has ***the effect of increasing the contrast*** of a low contrast image. ***If a stretch is desired over a smaller range, different MAX and MIN values can be specified.***

In general, histogram stretch will ***increase image contrast***.

ملاحظة: في حال عدم تحديد القيمة العليا والدنيا MIN, MAX في السؤال **فتعتمد القيم الأفتراضية** صفر ، ٢٥٥ أما اذا حددت في السؤال **فيجب** لألتزام بالقيم المحددة.

Example1:

Apply histogram stretch to the below subimage :

20	90
75	30

Stretch (20) = $[\frac{20-20}{90-20}][\ 255-\ 0] + 0 = 0$

Stretch (90) = $[\frac{90-20}{90-20}][\ 255-\ 0] + 0 = 255$

Stretch (75) = $[\frac{75-20}{90-20}][\ 255-\ 0] + 0 = [\ \frac{55}{70}\ ]\ 255 = 200.35 = 200$

Stretch (30) = $[\frac{30-20}{90-20}][\ 255-\ 0] + 0 = [\ \frac{10}{70}\ ]\ 255 = 36.42 = 36$

The resulted sub image is:

0	255
200	36

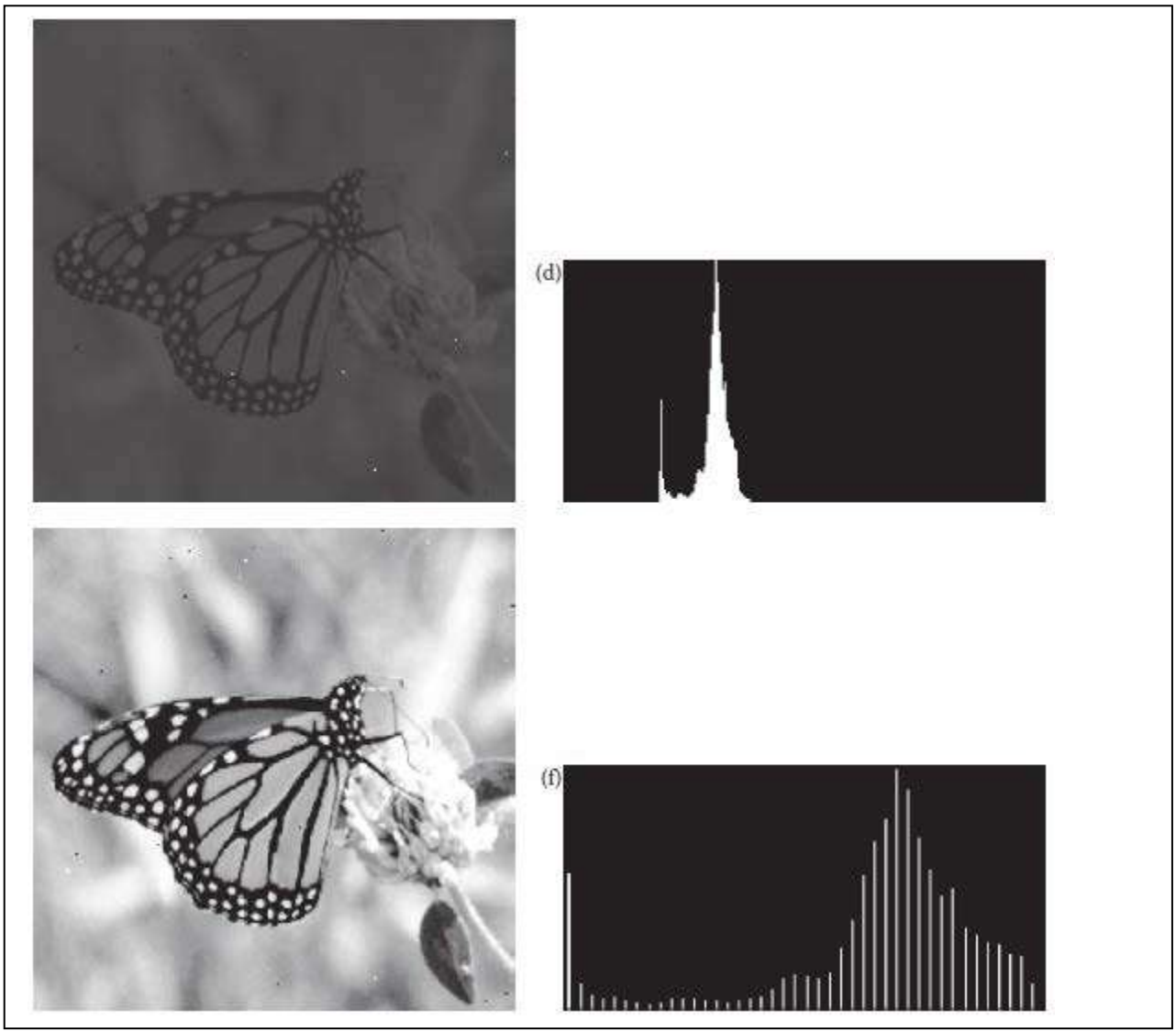


Figure 14: histogram stretch

b. Histogram Shrink

The opposite of a histogram stretch is a histogram shrink, which will decrease image contrast by compressing the gray levels. The mapping function for a histogram shrink can be found by the following equation:

$$Shrink\left[I(r,c)\right] = \left[\frac{Shrink_{MAX} - Shrink_{MIN}}{I(r,c)_{MAX} - I(r,c)_{MIN}}\right]\left[I(r,c) - I(r,c)_{MIN}\right] + Shrink_{MIN}$$

Where :

$I(r,c)_{MAX}$ is the largest gray-level value in the image I(r,c)

$I(r,c)_{MIN}$ is the smallest gray-level value in the image I(r,c)

$Shrink_{MAX}$ and $Shrink_{MIN}$ correspond to the maximum and minimum gray-level values ***desired*** in the compressed histogram.

In general, this process produces an image of ***reduced contrast*** and may not seem to be useful as an image enhancement tool.

ملاحظة : في الأسئلة التي تتطلب أجراء عملية Shrink **يجب** تحديد قيمة التقليص المطلوبة في السؤال.

Example2 :

Apply histogram shrink to the below subimage where the desired gray level values for the compressed histogram are 40, 80:

20	90
75	30

Sol

Shrink (20) = $[\frac{80-40}{90-20}]$[20- 20] + 40 = 40

Shrink (90) = $[\frac{80-40}{90-20}]$[90- 20] + 40 = $[\frac{40}{70}]$[70] + 40 = 80

Shrink (75) = $[\frac{80-40}{90-20}]$[75- 20] + 40 = $[\frac{40}{70}]$[55] + 40 = 71.42=71

Shrink (30) = $[\frac{80-40}{90-20}]$[30- 20] + 40 = $[\frac{40}{70}]$[10] + 40 = 45.71=46

The resulted sub image is :

40	80
71	46

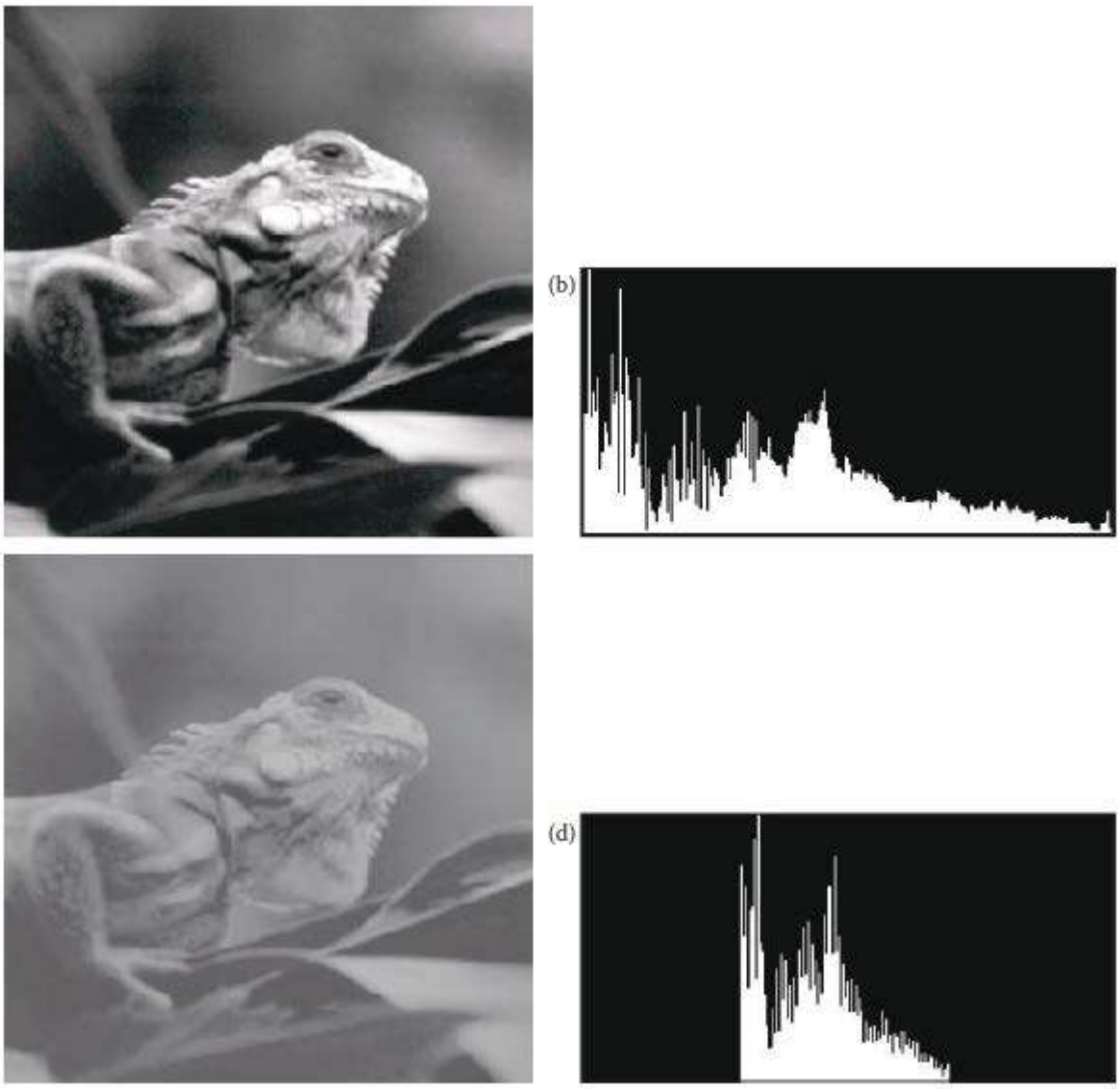


Figure 15: histogram shrink

c. Histogram Slide

The histogram slide technique can be used to make an image either ***darker*** or ***lighter*** but retain the relationship between gray-levels values. This can be accomplished by simply ***adding or subtracting a fixed number*** from all the gray level values as follow:

$$Slide(\ I(r,c)\ ) = I(r,c) + OFFSET$$

Where OFFSET value is the amount to slide the histogram**.**

In this equation we assume that any values slide past the minimum and maximum value will be clipped to the respective minimum or maximum. A **positive** OFFSET value will **increase** the overall **brightness**, whereas a **negative** OFFSET will create a **darker** image. Figure 15 shows a dark image that has been brightened by a histogram slide with a positive OFFSET value.

Example3

Make the below subimage brighter by 100 gray level value.

20	90
75	30

Sol

Slide (20) = 20+100 = 120

Slide (90) = 90+100 = 190

Slide (75) = 75+100 = 175

Slide (30) = 30+100 = 130

The resulted sub image is :

120	190
175	130

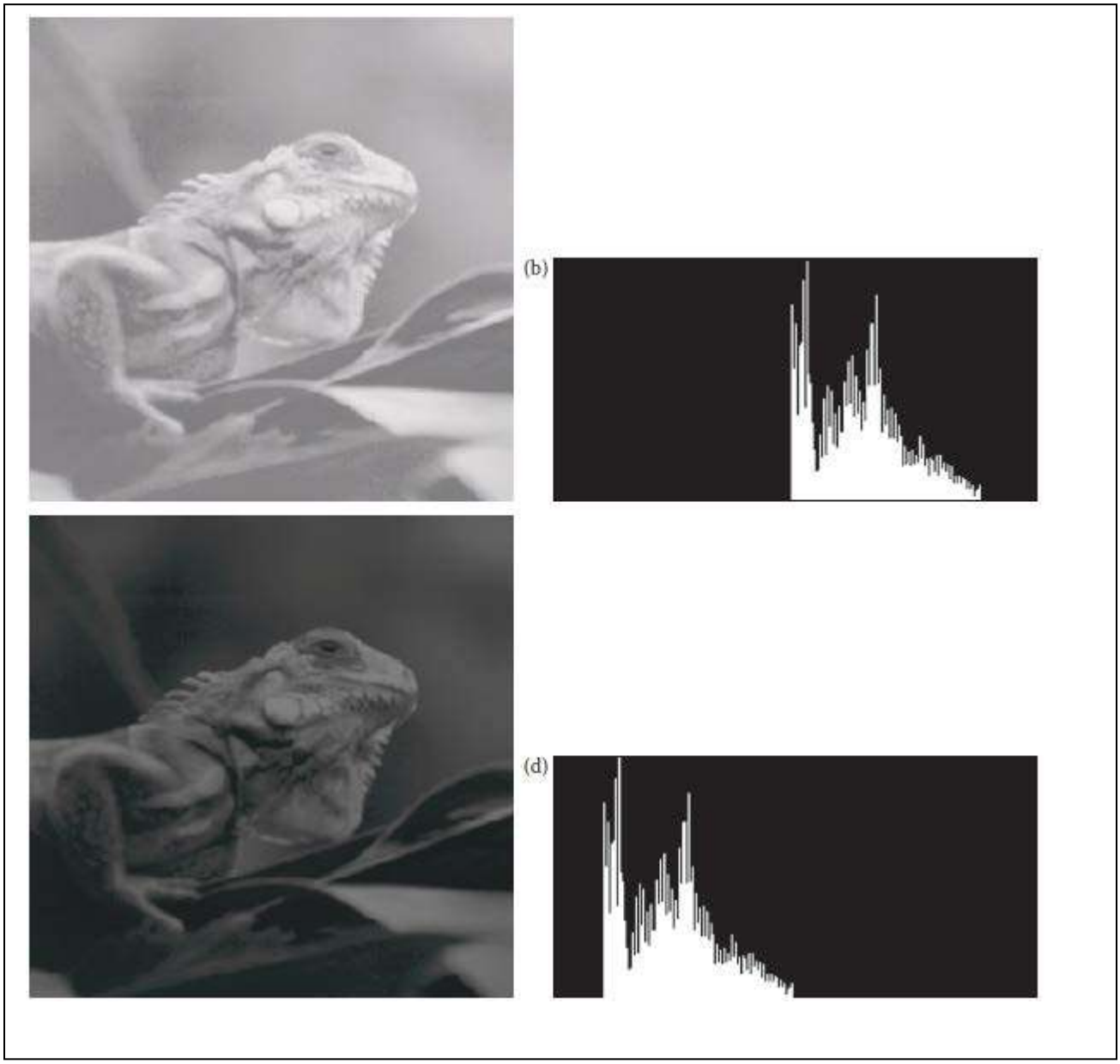


Figure 16: Histogram Slide

d. Histogram Equalization

Histogram Equalization is a popular technique for improving the appearance of a poor image. Its function is ***similar*** to that of but often provides more ***visually pleasing*** results across a wider range of images. Histogram equalization is a technique where the histogram of the resultant image is ***as flat as possible*** (with histogram stretching the overall shape of the histogram remains the same).

Histogram equalization, ***redistribute*** gray levels in an attempt to ***flatten*** the frequency distribution. ***More gray levels are allocated where there are most pixels, fewer gray levels where there are fewer pixels.*** This tends to ***increase contrast*** in the most heavily populated regions of the image.

If we are to ***increase contrast*** for the ***most frequently*** occurring gray level range and ***reduce contrast*** in the ***less popular part*** of the gray level rang, then we need a mapping function which has a steep slope ($a > 1$) at gray levels that occur frequently, and a gentle slope ($a < 1$) at unpopular grey levels. The cumulative histogram of the image has these properties.

Indeed, the mapping function we need is obtained simply by rescaling the cumulative histogram so that its values lie in the range 0-255.The algorithm below shows how this works in practice. From the histogram of the image, we determine the cumulative histogram, C, rescaling the values as we go so that they occupy an 8-bit range. In this way, C becomes a look-up table that can be subsequently applied to the image in order to carry out equalization.

Algorithm 4 : Histogram Equalization

Compute a scaling factor $\alpha = 255$ / number of pixels.

Calculate histogram using **algorithm 3**

$C[i] = \alpha * \text{histogram}[0]$

For all remaining gray levels, I , do

$C[i] = C[i-1] + \alpha * \text{histogram}[i]$

End for

For all pixel coordinate, x and y, do

g (x,y) = C[f (x,y)]

End for

Histogram Equalization is used widely in image processing, mainly because it is a ***completely automatic*** technique, with ***no parameter to set***. At times, it can improve our ability to interpret an image dramatically.

Histogram equalization may ***not*** always ***provide the desired effect*** because its goal is fixed – to distribute the grey level value as evenly as possible. However, it is difficult to predict how beneficial equalization will be for any given image; in fact, it may not be of any use at all. This is because the ***improvement in contrast is optimal statistically, rather than perceptually***. In images with narrow histograms and relatively few grey levels, a massive increase in contrast due to histogram equalization can have the adverse effect of reducing perceived image quality.

Steps of Histogram Equalization

The histogram equalization process for digital images consist of four steps :

1. Find the **running sum** of the histogram.
2. **Normalize** the values from step 1 by **dividing** by the total number of pixels.
3. **Multiply** the values from step 2 by the **maximum** gray level value and **round**.
4. **Map** the gray levels value to the results from step 3 using a one-to-one correspondence.

Example

Assume we have an image with ***3 bits/pixels***, so the possible range of the pixels values is 0 to 7. We have an image with the following histogram

Gray levels value	number of pixels (Histogram values)
0	10
1	8
2	9
3	2
4	14
5	1
6	5
7	2

STEP 1: Create a **running sum** of the histogram value. This means that the first value is 10, the second is 10+8=18, next is 10+8+9=27, and so on. Here we get 10, 18, 27, 29, 43, 44, 49, 51.

STEP 2: **Normalize** by dividing by the total number of pixels. The total number of pixels is 10 +8+9+2+14+1+5+2 = 51 (note that this is the last number from step 1), so we get 10/51, 18/51, 27/51, 29/51, 43/51, 44/51, 49/51, 51/51.

STEP 3 : **Multiply** these values by the **maximum gray-level** values, in this case 7, and then **round** the result to the closest integer. After this is done we obtain 1, 2, 4, 4, 6, 6, 7, 7.

STEP 4 : **Map** the original values to the results from step 3 by a one-to-one correspondence. This is done as follows

Original Gray level value	Histogram Equalized Gray level values	Number of pixels
0	1	10 (frequency of 0)
1	2	8 (frequency of 1)
2	4	11 (frequency of 2 + 3)
3	4	
4	6	15 (frequency of 4 + 5)
5	6	
6	7	7 (frequency of 6 + 7)
7	7	

In the below figure we see the original histogram and resulting histogram (equalized histogram). Although the result is not flat, it is closer to being flat than the original histogram.

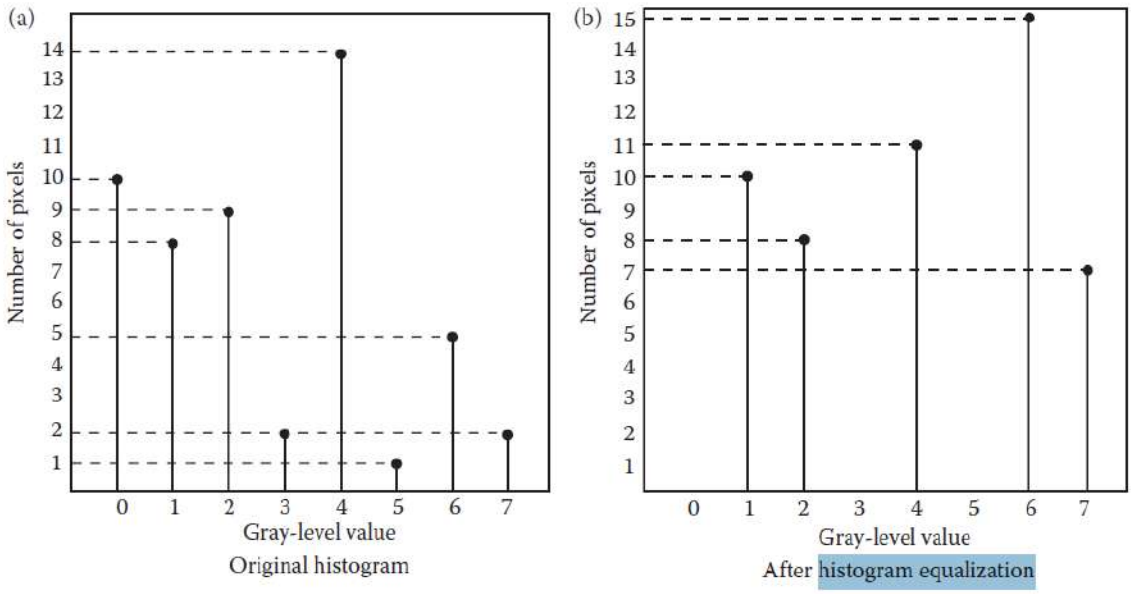


Figure 17 Histogram Equalization

Histogram equalization of a digital image will not typically provide a histogram that is perfectly flat, but will ***make it as flat as possible***.

The below figures shows the result of equalizing the histogram of two images with very poor contrast.

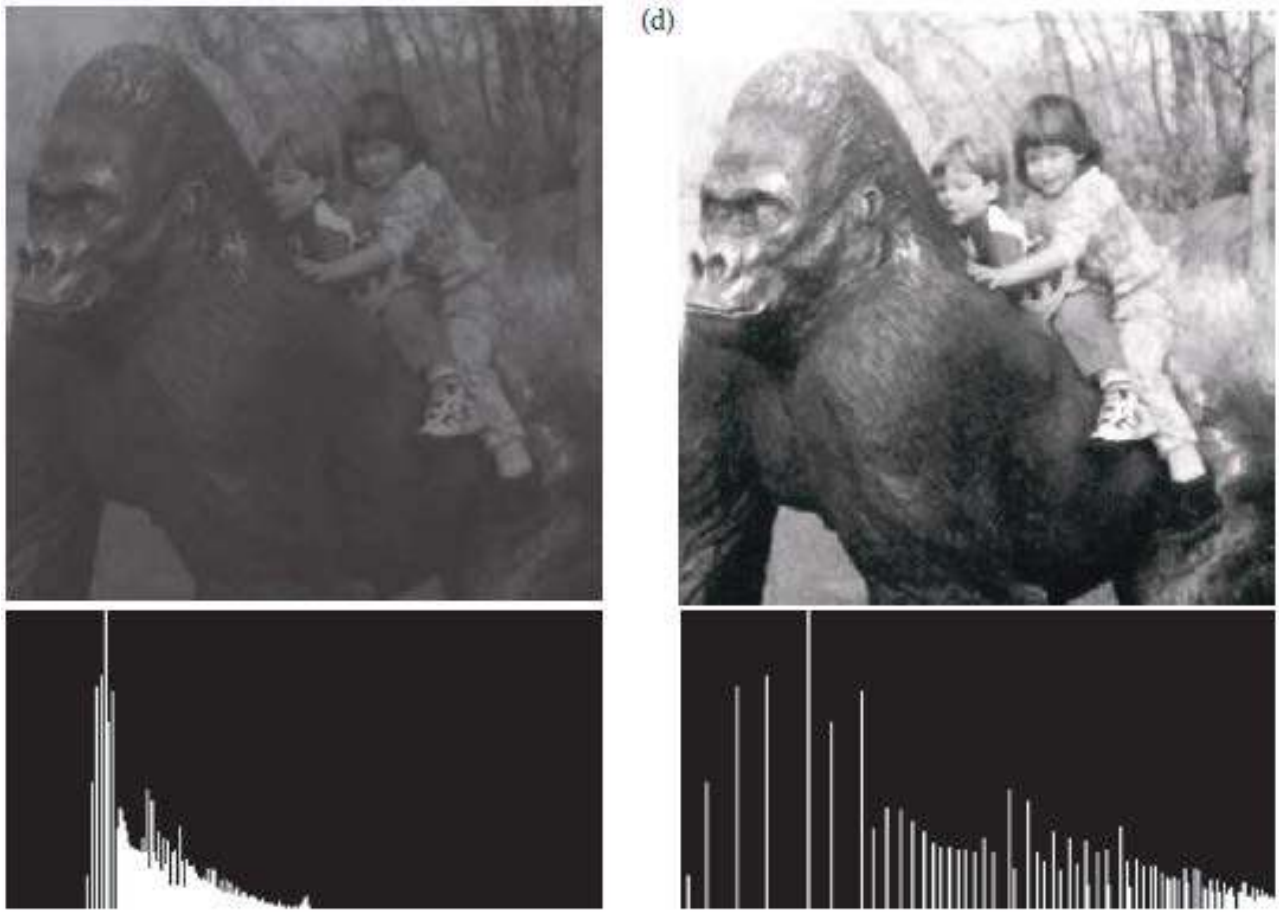


Figure 18 Histogram Equalization

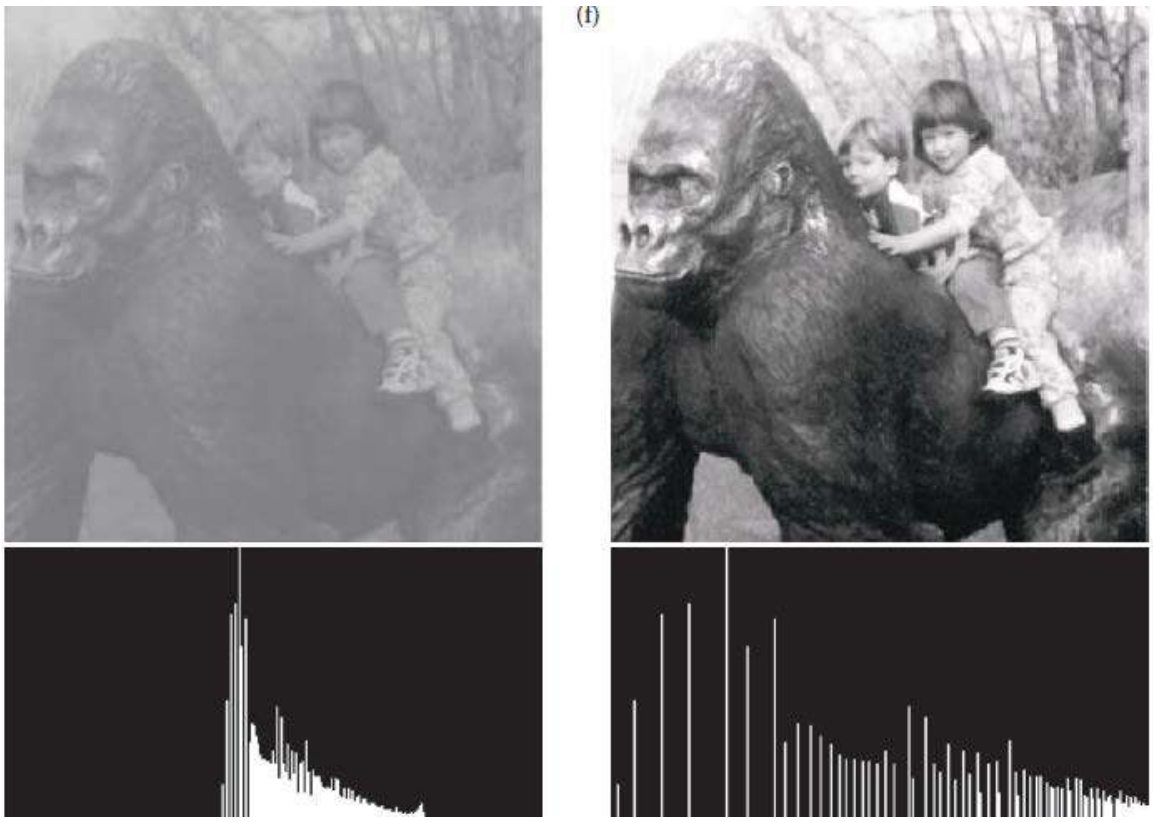


Figure 18 Histogram Equalization (continue)

5. Image Geometry

Often for image analysis, we want to investigate more closely a specific area within the image, called Region of Interest (ROI). To do this we need operations that ***modify*** the ***spatial coordinates*** of the image, and these are categorized as image geometry operations. The image geometry operations discussed here include ***crop, zoom, enlarge, shrink, translate, and rotate*****.**

Note The effect of the Image geometry operation is ***mainly on the image*** itself ***not on*** its histogram (i.e. enlarge the image, shrink the image, rotate the image,…)**.**

5.1 crop, zoom, enlarge

The image ***crop*** process is the process of ***selecting*** a small portion of the image, a sub image, and ***cutting it*** away from the rest of the image. After we have cropped a sub image from the original image, we can ***zoom*** in on it by enlarging it. This zoom process can be done in numerous ways, but typically a zero or first order hold is used.

5.1.1 Zero-order hold

A zero-order hold is performed by ***repeating*** previous pixel values, thus creating a ***blocky effect***.

$$\begin{bmatrix} 2 & 6 \\ 7 & 3 \end{bmatrix} \Rightarrow \begin{bmatrix} 2 & 2 & 6 & 6 \\ 7 & 7 & 3 & 3 \end{bmatrix} \Rightarrow \begin{bmatrix} 2 & 2 & 6 & 6 \\ 2 & 2 & 6 & 6 \\ 7 & 7 & 3 & 3 \\ 7 & 7 & 3 & 3 \end{bmatrix}$$

This method allows us to ***enlarge*** an N × N size of **2N × 2N** and can be ***repeated as desired***.

5.1.2 First order hold

To extend the image size with a first-order hold, we do ***linear interpolation*** between ***adjacent pixels***. A comparison of the images resulting from these two methods is shown in figure 19.

Although the implementation of the zero-order hold is straightforward, the first-order hold is more complicated. It can be done in ***two different ways***:

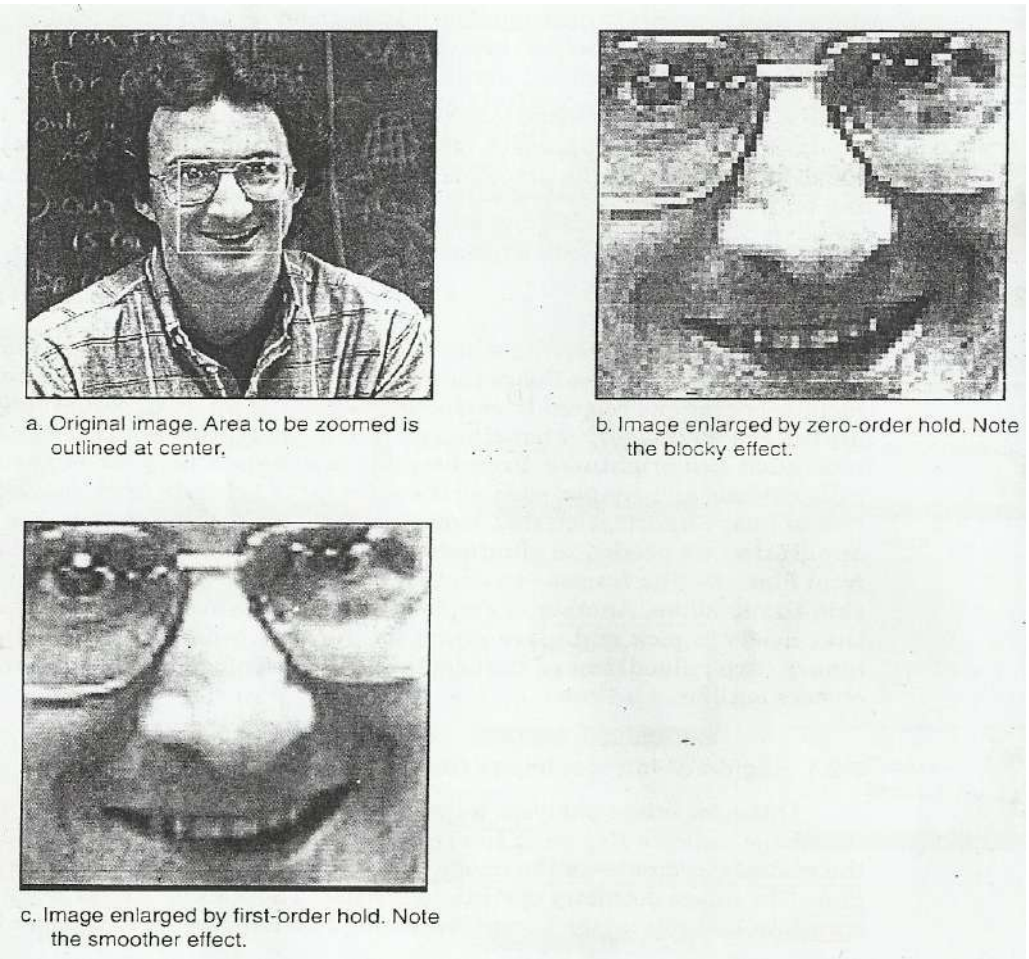


Figure 19 : zooming methods

a. Averaging method

The easiest way to do this is to find the ***average*** value between two pixels and use that as the pixel value between those two pixels; we can do this for the rows first, as follows:

ORIGINAL IMAGE ARRAY IMAGE WITH ROWS EXPANDED

$$\begin{pmatrix} 8 & 4 & 8 \\ 4 & 8 & 4 \\ 8 & 2 & 8 \end{pmatrix} \Rightarrow \begin{pmatrix} 8 & 6 & 4 & 6 & 8 \\ 4 & 6 & 8 & 6 & 4 \\ 8 & 5 & 2 & 5 & 8 \end{pmatrix}$$

The first two pixels in the first row are averaged, (8 + 4) / 2 = 6, and this number is inserted between those two pixels. This is done for every pixels pair in each row. Next, take that result and expand the columns in the same way, as follow:

IMAGE WITH ROWS AND COLUMNS EXPANDED

$$\begin{pmatrix} 8 & 6 & 4 & 6 & 8 \\ 6 & 6 & 6 & 6 & 6 \\ 4 & 6 & 8 & 6 & 4 \\ 6 & 5.5 & 5 & 5.5 & 6 \\ 8 & 5 & 2 & 5 & 8 \end{pmatrix}$$

This method allows us to ***enlarge*** an N × N size of **(2N – 1) × (2N – 1)** and can be ***repeated as desired***.

b. Convolution method

Another method that can achieve the same result requires a mathematical process called ***convolution***. With this method of image enlargement, a two-step process is required: **1)** extend the image by adding rows and columns of zeros between the existing rows and columns and **2)** perform convolution. The image is extended as follow:

ORIGINAL IMAGE ARRAY IMAGE WITH ROWS EXPANDED

$$\begin{pmatrix} 3 & 5 & 7 \\ 2 & 7 & 6 \\ 3 & 4 & 9 \end{pmatrix} \Rightarrow \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 5 & 0 & 7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 7 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 4 & 0 & 9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Next, we use convolution mask, which is slide across the extended image, and performs a simple arithmetic operation at each pixel location.

CONVOLUTIONMASK FORFIRST ORDER HOLD

$$\begin{pmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{pmatrix}$$

The convolution process requires us to ***overly*** the mask on the image, ***multiply*** the coincident values, and ***sum*** all these results. This is equivalent to finding the ***vector inner product*** of the mask with the underlying sub image. The vector inner product is found by overlying the mask on a sub

image, multiplying coincident terms, and summing the resulting products. For example, if we put the mask over the upper-left corner of the image, we obtain (from right to left, and top to bottom):

$$\frac{1}{4}(0) + \frac{1}{2}(0) + \frac{1}{4}(0) + \frac{1}{2}(0) + 1\,(3) + \frac{1}{2}(0) + \frac{1}{4}(0) + \frac{1}{2}(0) + \frac{1}{4}(0) = 3$$

Note that the ***existing image values do not change***. The next step is to slide the mask over by one pixel; and repeat the process, as follows:

$$\frac{1}{4}(0) + \frac{1}{2}(0) + \frac{1}{4}(0) + \frac{1}{2}(3) + 1\,(0) + \frac{1}{2}(5) + \frac{1}{4}(0) + \frac{1}{2}(0) + \frac{1}{4}(0) = 4$$

Note this is the average of the two existing neighbors. This process continues until we get to the end of the row, each time ***placing*** the ***result*** of the operation in the location corresponding to the ***center of the mask***.

When the end of the row is reached, the mask is ***moved down one row***, and the process is repeated row by row until this procedure has been performed on the entire image; the process of ***sliding, multiplying***, *and* ***summing*** is called ***convolution*** (see figure 20).

Not that the ***output image*** must be put in a ***separate image*** array called ***buffe***r, so that the existing values are ***not overwritten*** during the convolution process.

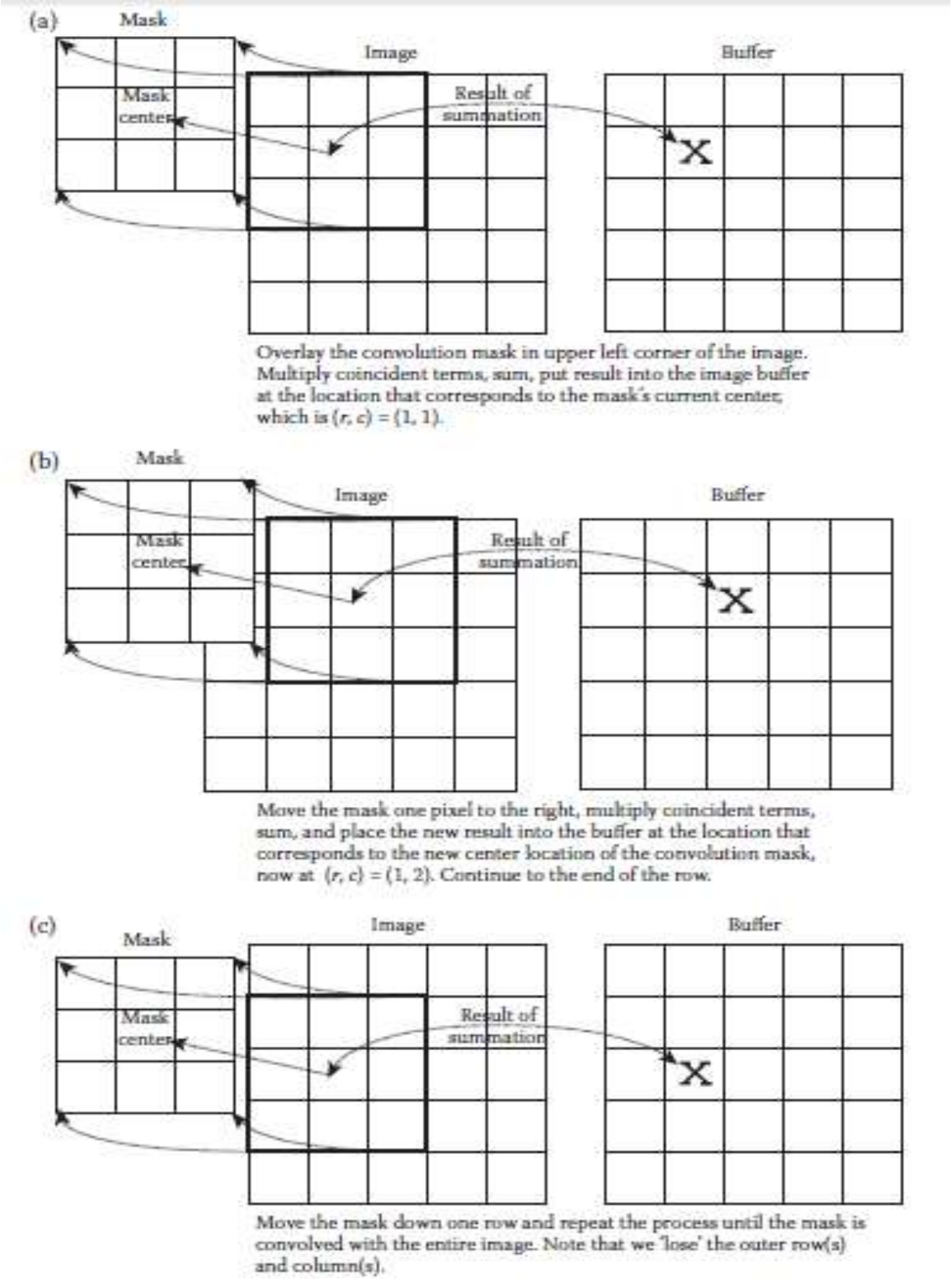


Figure 20: the convolution process

c. Linear interpolate method

The above two methods will only allow us to enlarge an image by a factor of (2N – 1) × (2N– 1), but what if we want to enlarge an image by something ***other than*** a factor of (2N – 1)? To do this we need to apply a more general method; we take two adjacent values and ***linearly interpolate*** more than one value between them. This is done by defining an enlargement number ***K*** and then following this process: **1)** subtract the two adjacent values **2)** divide the result by ***K*** **3)** add that result to the smaller value, and keep adding the result from the second step in a running total until all (***K*** – 1) intermediate pixel locations are filled.

EXAMPLE

If we want to enlarge an image to ***three times*** its original size, and we have two adjacent pixel values 125 and 140.

1. Find the difference between the two values 140 – 125 = 15.
2. The desired enlargement factor is *K* = 3, so we get 15 / 3 = 5.
3. Next determine how many intermediate pixel values we need: K – 1 = 3 – 1 = 2. The two pixel values between 125 and 140 are : 125 + 5 = 130 and 125 + 2*5 = 135.

We do this for every pair of adjacent pixels, first along the rows and then along the columns. This will allow us to enlarge the image by any factor **K (N – 1) + 1**, where K is an integer and N × N is the image size. Typically, N is large and K is small, so this is approximately equal to **KN**.

5.2 Image Shrinking

The process opposite to enlarge an image is shrinking it. This is not typically done to examine a ROI more closely but to reduce the amount of data that needs to be processed.

Image shrinking is accomplished by taking groups of pixels that are spatially adjacent (blocks of 2×2 or 3×3 pixels) and mapping them to one pixel. This can be done in one of ***three ways:*** 1) averaging 2) median 3) decimation. For averaging method, we take all the pixels in each group and find the average gray level by summing the values and dividing by the number of pixels in the group. Within the second method, i.e. median, we sort all the pixels values from lowest to highest and then select the middle value. The third approach, decimation, also known as sub sampling, entails simply eliminating some of the data. For example to reduce the image by a factor of two, we simply take every other row and column and delete them.

EXAMPLE

For the below 2 × 2 block of pixels, shrink this block using the three methods of shrinking.

$$\begin{bmatrix} 6 & 8 \\ 1 & 9 \end{bmatrix}$$

Averaging	**median**	**decimation**
6+8+1+9=24/4=6	1,6,8,9 = (6+8)/2=7	eliminate the first row and the first column and the result is 9

5.3 Translation and Rotation

Two other operation of interest for the ROI image geometry are translation and rotation. These processes may be performed for many application-specific reasons, for example to align an image with a known template in a pattern matching process or to make certain image details easier to see. The translation process can be done with the following equations :

$$r' = r + r_o$$

$$c' = c + c_o$$

where **r'** and **c'** are the new coordinates, **r** and **c** are the original coordinates, and $\mathbf{r_o}$ and $\mathbf{c_0}$ are the distance to move, or translate, the image.

The rotation process requires the use of these equations :

$$r^{\wedge} = r (\cos \Theta) + c (\sin \Theta)$$

$$c^{\wedge} = - r (\sin \Theta) + c (\cos \Theta)$$

where $r^{\wedge}$ and $c^{\wedge}$ are the new coordinate, r and c are the original coordinates, and Θ is the angle to rotate the image. Θ is defined in a ***clockwise direction*** from the horizontal axis at the image origin in the upper-left corner.

The rotation and translation process can be combined into one set of equations :

$$r^{\wedge'} = (r + r_o) (\cos \Theta) + (c + c_o) (\sin \Theta)$$

$$c^{\wedge'} = - (r + r_o) (\sin \Theta) + (c + c_o) (\cos \Theta)$$

where $r^{\wedge'}$, $c^{\wedge'}$ are the new coordinates and r, c, r_o, c_o and Θ are previously defined.

Example 1

Translate the below sub image 5 pixels toward the right.

	31	32	33
14	93	90	98
15	73	78	97
16	88	83	86

⇒

	36	37	38
14	93	90	98
15	73	78	97
16	88	83	86

Example 2

Translate the below sub image down 5 pixels.

	31	32	33
14	53	44	56
15	67	68	96
16	78	71	76

⇒

	31	32	33
19	53	44	56
20	67	68	96
21	78	71	76

6. Image Compression

Lossy / lossless compression: Certain compression methods are ***lossy.*** They achieve better compression by losing some information. When the compressed stream is decompressed, the result is not identical to the original data stream. Such a method makes sense especially in compressing images, movies, or sounds. If the loss of data is small, we may not be able to tell the difference. In contrast, text files, especially files containing computer programs, may become worthless if even one bit gets modified. Such files should be compressed only by a ***lossless*** compression method, also special purpose images like medical images, forensic images, NASA images are compressed using ***lossless*** compression methods.

RLE Image Compression

Images redundancy is illustrated by the fact that in a nonrandom image, **adjacent pixels** tend to have **similar colors**.

Compressing an image using RLE is based on the observation that if we select a pixel in the image at random, there is a good chance that its neighbors will have the same color. The compressor thus scans the bitmap row by row, looking for runs of pixels of the same color.

Run Length Encoding (RLE) is image compression methods that work by counting the number of adjacent pixels with the same gray level value. This count called the run length, is then coded and stored. Here we will explore several methods of run-length coding including basic method that are used primarily for binary images and extended version for gray scaled images.

6.1 Basic RLE

Basic RLE is used primarily for **binary images** but can be work with complex images that have been preprocessed by thresholding to reduce the number of gray level to two. To implement basic RLE, we often use horizontal RLE, counting along the rows. The compressor thus scans the bitmap row by row, looking for runs of pixels of the same color.

In basic horizontal RLE the number of bits required for the coding depends on the number of pixels in a row. If the row has 2^n pixels, then the required number of bits is **n** (at minimum) , so that a run that is the length of the entire row can be coded.

Example

For a 256 × 256 image.

Requires 8 bit to encode the count since $2^8 = 256$.

اي ان الصف الواحد يحتاج الى 8 bit كحد ادنى لترميز العداد وذلك في حال كانت كل ال Pixels في ذلك الصف من نوع واحد (1 مثلا).

For a 512 × 512 image.

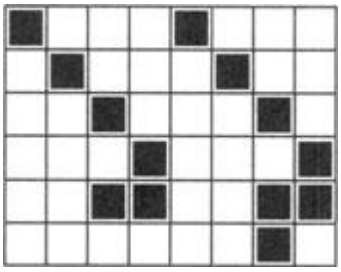
Requires 9 bit to encode the count since $2^9 = 512$.

The compressor assumes that the bitmap ***starts with white pixels***. If this is not true, then the bitmap ***starts with zero white pixels***, and the output stream should ***start with 0***. The resolution of the bitmap should also be saved at the start of the output stream.

The size of the compressed stream depends on the complexity of the image. The ***more detail***, the ***worse*** the compression.

Example

What would be the compressed file in the case of the following 6 × 8 bitmap image?



Sol

0,1,3,1,3
1,1,3,1,2
2,1,3,1,1
3,1,3,1
2,2,2,2
6,1,1

Note that in the first row, the first RLE number is 0, which means that the bitmap starts with ***zero white pixels***, since we are using the conversion that the bitmap always starts with white pixels.

6.2 Gray-Level RLE

RLE can also be used to compress grayscale images. Each run of pixels of the same intensity (gray level) is encoded as a ***pair (run length, pixel value)***. The run length usually occupies one byte, allowing for runs of up to 255 pixels. The pixel value occupies several bits, depending on the number of gray levels (typically between 4 and 8 bits).

Example:

An 8-bit deep grayscale bitmap that starts with
12,12,12,12,12,12,12,12,12,35,76,112,67,87,87,87,5,5,5,5,5,5,1, ...
Is compressed into:
9,12,35,76,112,67,**3**,87,**6**,5,1, ... , where the Red numbers indicate counts. The problem is to ***distinguish*** between a byte containing a grayscale value (such as 12) and one containing a count (such as 9). Here are some ***solutions*** (although not the only possible ones):

1. If the image is limited to just 128 grayscales, we can devote one bit in each byte (***the most significant bit***) to indicate whether the byte contains a grayscale value or a count.
2. If the number of grayscales is 256, it can be reduced to 255 with one value reserved as a flag to precede every byte with a count. If the flag is, say, 255, then the sequence above becomes:
255,**9**,12,35,76,112,67,**255**,**3**,87,**255**,**6**,5,1, ...
1. Again, an extra bit (a flag) is devoted to each byte to indicate whether the byte contains a grayscale value or a count. This time, however, these extra bits are accumulated in groups of 8, and each group is written on the output stream preceding (or following) the 8 bytes it "belongs to".

Example: the sequence **9**,12,35,76,112,67,**3**,87,**6**,5,1, ... becomes
10000010,**9**,12,35,76,112,67,**3**,87,100 .....,**6**,5,1, ...

The total size of the extra bytes is, of course, 1/8 the size of the output stream (they contain one bit for each byte of the output stream), so they increase the size of that stream by 12.5%.

Two more points should be mentioned:

1. In color images it is common to have each pixel stored as three bytes, representing the intensities of the Red, Green, and Blue components of the pixel. In such a case, runs of ***each color*** should be ***encoded***

separately. Thus the pixels (171,85,34), (172,85,35), (172,85,30), and (173,85,33) should be separated into the three sequences (171,172,172,173, ...), (85,85,85,85, ...), and (34,35,30,33, ...). Each sequence should be run-length encoded separately. This means that any method for compressing grayscale images can be applied to color images as well.

2. It is preferable to encode each row of the bitmap individually. Thus if a row ends with four pixels of intensity 87 and the following row starts with 9 such pixels, it is better to write ... ,4,87,9,87, ... on the output stream rather than ... , 13,87, ... It is even better to write the sequence ... ,4,87, eol, 9, 87, ... , where "eol" is a special end-of-line code.

RLE disadvantage

The following two points can be considers as the Disadvantage of image RLE:

1. When the image is ***modified***, the run lengths normally have to be ***completely redone***.
2. The RLE output can sometimes be bigger than pixel-by-pixel storage (i.e., an uncompressed image).

س. متى تحدث الحالة في النقطة ٢ أعلاه.