Introduction to Image Processing and Analysis

© 2020 Alejandro J. León Salas[1,2]

[1]Department of Software Engineering
University of Granada

[2]Virtual Reality Lab
University of Granada

15 February 2020

Outline

Motivation

- Images from medical imaging are the primary raw data in medical graphics.
- We need to understand concepts and know words from the world of digital image processing (DIP).
- Many concepts will be extended and used in other stages of the visualization pipeline.

Computer Vision

- This sequence of operations: image capture, processing, region extraction, region labelling, high-level identification, is characteristic of image understanding and computer vision (CV) problems.
- In order to simplify the task of CV understanding, two levels are usually distinguished: **Low-level image processing** and high-level image understanding.
- Low-level methods often include pre-processing methods for noise filtering, edge extraction, and image sharpening.

Digital Image Processing (DIP)

- The following sequence of processing steps is commonly seen:
  1. An image is captured by a sensor and digitized.
  2. The first stage **suppreses noise** (Image pre-processing).
  3. The second stage **enhances** some object features which are relevant to understanding the image e.g. **Edge extraction**.
- **Image segmentation** is the next step, in which the computer tries to separate objects from the image background and from each other.

Image Segmentation

- **Total and partial segmentation** may be distinguished; total segmentation is possible only for very simple tasks, an example being the recognition of dark non-touching objects from a light background.
- Low-level image processing techniques handle the partial segmentation tasks, in which only the cues which will aid further high-level processing are extracted.
- Finding parts of object boundaries is an example of low-level partial segmentation.

DIP data versus CV data

- Low-level data are comprised of original images represented by matrices composed of **brightness** values or other data but represented as **matrices** as well.
- High-level data represent knowledge about image content—for example, object size, shape, and mutual relations between objects in the image and are usually expressed in **symbolic form**.
- It is usually still a human operator who finds a sequence of relevant operations, and domain-specific knowledge and uncertainty cause much to depend on this operator's intuition and previous experience.

Basic concepts

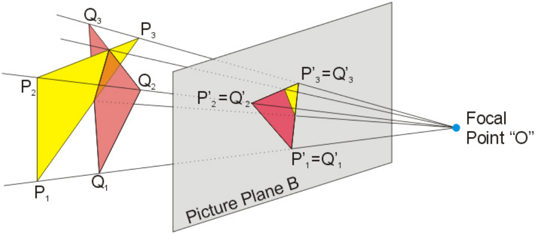
- A **signal** is a function depending on some variable with physical meaning; it can be n-dimensional $(n = 1, \ldots, n)$.
- Commonly we will be using 1- 2- 3-, and at most, 4-dimensional signals (functions).
- A **scalar function** might be sufficient to describe a monochromatic image, while a **vector function** might describe a color image.
- Functions we shall work with may be categorized as continuous, discrete, or digital.
    - A **continuous function** has continuous domain and range.
    - If the domain set is discrete but the range is continuous, then we get a **discrete function**.
    - If both domain and range are discrete, then we have a **digital function**

Image functions

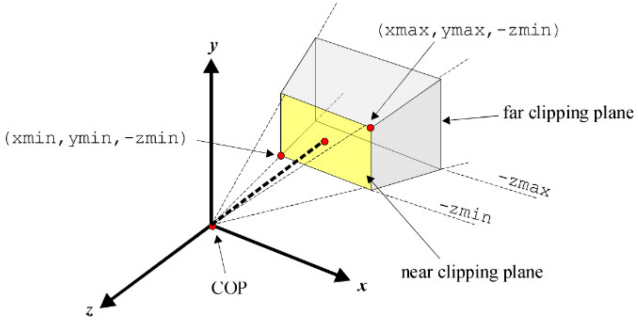
- The **image** can be modeled by a **continuous function** of two or three variables:
  - **Static images**. Function arguments are coordinates (x, y) in a plane.
  - **Dynamic images**. A third variable t might be added (x, y, t) because images change in time, e.g. video.
- The image function values (range) correspond to the brightness at image points. **Brightness** integrates different optical quantities.
- We shall call **intensity image** to the 2D image on the human eye retina (or on a camera sensor) that bears information about brightness points.
- The function values can express other physical quantities as well (temperature, pressure distribution, distance from the observer, etc.).

Intensity image

- The 2D intensity image is the result of a **perspective projection** of the 3D scene, which is modeled by the image captured by a pin-hole camera.



Perspective projection in OpenGL



Perspective projection

- The quantities $P = (x, y, z)$ are the coordinates of the point P in a 3D scene in *World Coordinates*, and f is the focal length of the lens.
- The projected point, P' has coordinates (x', y') in the 2D image plane, where

$$x' = \frac{xf}{z} \quad y' = \frac{yf}{z} \tag{1}$$

- A non-linear perspective projection is often approximated by a linear **parallel** (or **orthographic**) **projection**, where $f \mapsto \infty$. Implicitly, $z \mapsto \infty$ too.

Problems with perspective projection

- When 3D objects are mapped into the camera plane by perspective projection, a lot of information disappears because such a transformation is not one-to-one.
- First problem. **Recovering information** lost by perspective projection is a **geometric problem** aimed to obtaing an object representantion in World Coordinates from multiple views of such an object represented in multiple images.

Problems with perspective projection

- A second problem is **understanding the image brightness**.
- The only information available in an **intensity image** is the brightness of the appropriate pixel, which is dependent on a number of independent factors such as:
  - Object surface **reflectance** properties, which characterize the image by $r(x, y)$.
  - **Illumination** properties, which characterize the image by $i(x, y)$
  - **Object surface orientation** with respect to a viewer and light source.

Image as a Continuous Function

- When an image is generated from a physical process, its values are proportional to energy radiated by a physical source, then $f(x, y)$ must be nonzero and finite, i.e. $0 < f(x, y) < \infty$
- The illumination and reflectance components combine as a product to form $f(x, y)$:

$$f(x, y) = i(x, y) \cdot r(x, y) \tag{2}$$

  where $0 < i(x, y) < \infty$ and $0 < r(x, y) < 1$. Reflectance is bounded by 0 (total absorption) and 1 (total reflectance).
- For images formed via transmission of the illumination through a medium (e.g. X-ray) we would speak about a **transmissivity** instead of a reflectivity function.

Monochromatic image

- A **monochromatic static image** is represented by a **continuous image function** $f(x, y)$ where x and y are coordinates in the plane.
- Image processing uses **digital image functions** which are usually represented by matrices, so **coordinates are integer numbers**. The domain of the image function is a region R in the plane

$$R = \{(x, y), 1 \leq x \leq x_m, 1 \leq y \leq y_n\} \tag{3}$$

  where x_m, y_n represent maximal image coordinates.
- It is assumed that the image function value is zero outside the domain R.

Digital image function

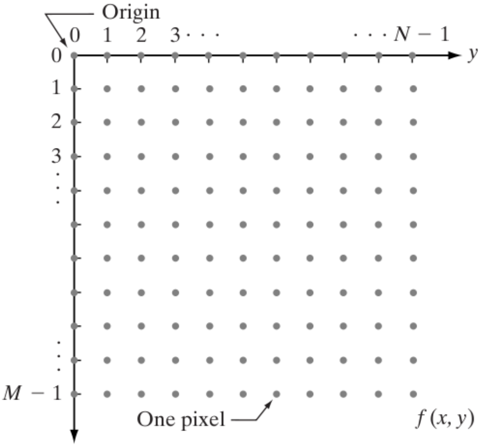
- The customary (canonical) orientation of coordinates in an image is the normal Cartesian form: horizontal x axis and vertical y axis, i.e. $(1, 0)$ and $(0, 1)$.
- The (*row*, *column*) orientation used in matrices is also often used in digital image processing, i.e.

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- The range of image function values is also limited; by convention, in monochromatic images the lowest value corresponds to black and the highest to white. Brightness values bounded by these limits are **gray-levels**.

Digital image function

- Convention for digital image function. For example, $(x, y) = (0, 1)$ represents the value of the second element $(y = 1)$ of the first row $(x = 0)$.



Digital image function

- The matrix form, $M \times N$, of a digital image.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix}$$

Digital image construction

- An image to be processed by computer must be represented using an appropriate discrete data structure. An image captured by a sensor is expressed as a continuous function $f(x, y)$ of two coordinates in the plane and also continuous in amplitude (range).
- **Image digitization** means that the function $f(x, y)$ is **sampled** into a matrix with M rows and N columns, i.e. digitizing the coordinate values.
- **Image quantization** assigns to each continuous sample an integer value. The continuous range of the image function $f(x, y)$ is split into L intervals, i.e. digitizing the amplitude values.

Continuous image function sampling

- The finer the sampling (i.e., the larger M and N) and quantization (the larger L), the better the approximation of the continuous image function $f(x, y)$ achieved.
- Two questions should be answered in connection with image function sampling:
  - The **sampling period** should be determined. The sampling period is the distance between two neighboring sampling points in the image.
  - The **geometric arrangement** of sampling points (sampling grid) should be set.

Sampling

- A continuous image function $f(x, y)$ can be sampled using a discrete grid of sampling points in the plane.
- The image is sampled at points $x = j\Delta x$, $y = k\Delta y$, for $j = 1, \ldots, M$, $k = 1, \ldots, N$. Two neighboring sampling points are separated by distance Δx along x axis and Δy along the y axis.
- Distances Δx and Δy are called the **sampling interval** (on the x or y axis), and the **matrix of samples** $f(j\Delta x, k\Delta y)$ **constitutes the discrete image**.

What sampling interval is used?

- Nyquist–Shannon sampling theorem.
- In real image digitizers, a sampling interval about ten times smaller than that indicated by the Shannon sampling theorem is used.
- The reason is that algorithms which reconstruct the continuous image on a display from the digitized image function use only a step function; i.e., a line is created from **pixels represented by individual squares**.

Aliasing

- Much of the visual degradation produced if the sampling interval is not good enough is caused by **aliasing** in the reconstruction of the continuous image function for display.
- This display can be improved by the reconstruction algorithm interpolating brightness values in neighboring pixels; this technique is called **anti-aliasing** and is often used in CG.
- If anti-aliasing is used, the sampling interval can be brought near to the theoretical value of Nyquist–Shannon's theorem.

Distribution of sampling points

- A continuous image is digitized at **sampling points**. These sampling points are ordered in the plane, and their geometric relation is called the **grid**.
- Grids used in practice are usually square or hexagonal.
- One infinitely small sampling point in the grid corresponds to one picture element (**pixel**) in the digital image. The set of pixels together cover the entire image; however, the pixel captured by a real digitizacion device has finite size.
- Why do we use square or hexagonal grids?

Quantization

- A value of the sampled image $f_s(j\Delta x, k\Delta y)$ is expressed as a digital value in image processing. The transition between continuous values of the continuous image function (brightness) and its digital equivalent is called **quantization**.
- The number of quantization levels should be high enough to permit human perception of fine shading details in the image.
- Most digital image processing devices use quantization into L equal intervals. If b bits are used to express the values of the pixel brightness, then the number of brightness levels is $L = 2^b$. L is called the number of discrete **gray levels** allowed for each pixel.

Spatial and Gray-level Resolution

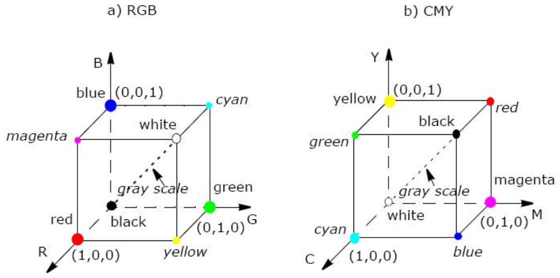
- **Spatial resolution** is the smallest discernible detail in an image.
- **Gray-level resolution** similarly refers to the smallest discernible change in gray level.
- We will refer to an L-level digital image of size $M \times N$ as having a spatial resolution of $M \times N$ pixels and a gray-level resolution of L levels.

Multi-spectral (or color) images

- Color is connected with the ability of objects to reflect electromagnetic waves of different wavelengths; the chromatic spectrum spans the electromagnetic spectrum from approximately 400 nm to 700 nm.
- Humans detect colors as combinations of the *primary colors red, green and blue*. This does not imply that all colors can be synthesized as combinations of these three.
- Display hardware generally deliver or display color via an **RGB model**; thus a particular pixel may have associated with it a 3D vector (r, g, b) which provides the respective color intensities.
- By convention, $(0, 0, 0)$ is black, (k, k, k) is white, $(k, 0, 0)$ is pure red, and so on. k here is the *quantization granularity* for each primary (256 is common).

RGB model

- The RGB model may be thought of as a 3D coordinatization of the color space.



Digital image properties

- A **digital image** consists of picture elements with finite size. These pixels carry information and we assume this hereafter that pixels are arranged into a **rectangular sampling grid**.
- We associate each pixel with a **lattice point** (grid point), i.e. a point with integer coordinates $\mathbb{Z}^2$.
- A digital image has several properties, both **metric** and **topological**.
- Another feature of difference is **human perception of images**, since judgment of image quality is also important.

Distance

- The distance between pixels p and q with coordinates (x_p, y_p) and (x_q, y_q) may be defined in several different ways:
- **Euclidean distance**. The disadvantages are costly calculation due to the square root, and its non-integer value.

$$D_E[(x_p, y_p), (x_q, y_q)] = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2} \quad (4)$$

- The pixels having a distance less than or equal to some value r from p are the points contained in a disk of radius r centered at (x_p, y_p).

Distance

- **Distance D_4 (city block).** The distance between two points can also be expressed as the minimum number of elementary steps in the digital grid which are needed to move from the starting point to the end point.

$$D_4[(x_p, y_p), (x_q, y_q)] = |x_p - x_q| + |y_p - y_q| \tag{5}$$

- The pixels having a D_4 distance less than or equal to some value r from p form a diamond centered at (x_p, y_p).

Distance

- **Distance D_8 (chessboard)**. If also moves in diagonal directions are allowed in the digitizacion grid. This distance is equal to the number of moves of the king on the chessboard from one part to another.

$$D_8[(x_p, y_p), (x_q, y_q)] = max\{|x_p - x_q|, |y_p - y_q|\} \quad (6)$$

- The pixels with D_8 distance less than or equal to some value r from p form a square centered at (x_p, y_p).

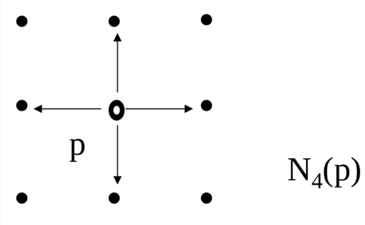
Chamfering

- Any of these metrics may be used as the basis of **chamfering**, in which the distance of pixels from some image subset (perhaps describing some feature) is generated (commonly known as **distance transform**).
- The resulting 'image' has pixel values of 0 for elements of the relevant subset, low values for close pixels, and then high values for pixels remote from it.
- Analogy with **distance fields** in 3D.

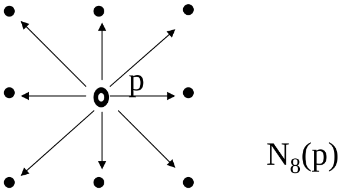
Neighbors of a pixel

- The four horizontal and vertical neighbors of a pixel p at coordinates (x, y) have coordinates $(x+1, y)$, $(x-1, y)$, $(x, y+1)$, $(x, y-1)$. This set of pixels, called the 4-neighbors of p, is denoted by $N_4(p)$. The D_4 distance from p to each of these pixels is 1.
- The four diagonal neighbors of p at (x, y) have coordinates $(x+1, y+1)$, $(x+1, y-1)$, $(x-1, y+1)$, $(x-1, y-1)$ and are denoted by $N_D(p)$. The D_E distance from p to each of these pixels is approximately 1.414.
- These set $N_4(p) \cup N_D(p)$ is called the 8-neighbors of p, and is denoted by $N_8(p)$. The D_8 distance from p to each of these pixels is 1.

Neighbors of a pixel



(a) 4-neighbors of pixel p.



(b) 8-neighbors of pixel p.

Topological adjacency

- The 4 and 8-neighbors definitions induce a binary relation between pairs of lattice points: **adjacency**. Two lattice points are adjacent if they are neighbors each other, then
- $p, q \in \mathbb{Z}^2$ are 4-adjacent iff p is 4-neighbor to q, i.e. $p \in N_4(q)$
- $p, q \in \mathbb{Z}^2$ are 8-adjacent iff p is 8-neighbor to q, i.e. $p \in N_8(q)$

Adjacency and connectivity in images

- **Adjacency**. Two pixels are adjacent if they are neighbors and if their gray levels satisfy a specified **criterion of similarity**.
- **Example 1**. Binary images.
  p adjacent q iff $q \in N_4(p)$ and $f(x_p, y_p) = f(x_q, y_q) = 1$,
  $f(x_i, y_i) \in \{0, 1\}$
- **Example 2**. Gray-level (intensity) images.
  p adjacent q iff $q \in N_8(p)$ and
  $f(x_p, y_p) \in S, f(x_q, y_q) \in S, S \subset \{0, \ldots, 255\}$

Adjacency and connectivity in images

- Let V be the set of gray levels values used to define adjacency.
- 4-adjacency. Two pixels p and q with values from V are 4-adjacent iff q is in $N_4(p)$.
- 8-adjacency. Two pixels p and q with values from V are 8-adjacent iff q is in $N_8(p)$.
- m-adjacency: Two pixels p and q with values from V are m-adjacent iff,
  - q is in $N_4(p)$.
  - q is in $N_D(p)$ and the set $N_4(p) \cap N_4(q) = \emptyset$ (has no pixels whose values are from V).

m-adjacency

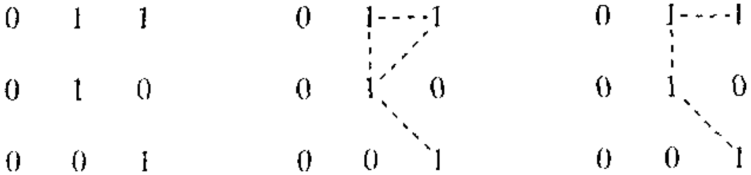


Figure: Solving 8-adjacency ambiguity with *m*-adjacency.

Adjacency and Connectivity in images

- A **path** from pixel p to pixel q as a sequence of pixels $p_1, p_2, \ldots, p_n$, where $p_1 = p$, $p_n = q$, and p_{i+1} is **adjacent to** (neighbor of) p_i, $i = 1, \ldots, n-1$. We can define 4-, 8- and m-paths based on the type of adjacency used.
- **Connectivity**. Two pixels p and q are said to be connected if there exists a path between them.
- Notice that p 4-(8-)(m-)adyacent to q iff p 4-(8-)(m-)connected to q.

Connected Components

- Let S be an image subset.
- If p and q are included in S then p **is connected to** q **in** S if there exists a path from p to q consisting entirely of pixels in S.
- For every pixel p in S, the set of pixels in S that are connected to p is called a **connected component of** S.
- If S has only one connected component then S is called **Connected Set**.

Regions and Boundaries

- **Region**. A subset R of pixels in an image is called a region of the image if R is a connected set.
- **Boundary**. The boundary (or border or contour) of a region R is the set of pixels in the region that have one or more neighbors that are not in R.

Edges

- This is a *local property* of a pixel and its immediate neighborhood. It is a vector given by a magnitude and direction.
- Images with many brightness levels are used for edge computation, and the **gradient of the image function** is used to compute edges. The edge direction is perpendicular to the gradient direction which points in the direction of image function growth.

Pixel adjacency. Considerations

- Remember taht any two pixels p and q are called 4-neighbors if they have distance $D_4 = 1$ from each other. Analogously, two pixels p and q are 8-neighbors if they have distance $D_8 = 1$.
- Remember that p 4-(8-)(m-)adyacent to q iff p 4-(8-)(m-)connected to q.
- If there is a path between two pixels in the image, these pixels could be considered **contiguous**. Alternatively, we can say that a region is the set of pixels where each pair of pixels in the set is contiguous.

The Relation *to be contiguous*

- The relation **to be contiguous** is reflexive, symmetric, and transitive and therefore defines a decomposition of the set (in our case image) into equivalence classes (regions).
- Assume that R_i are disjoint regions in the image which were created by the relation **to be contiguous**, and further assume (to avoid special cases) that these regions do not touch the image limits.
- Let region R be the union of all regions R_i. We can define a set R^C which is the set complement of region R with respect to the image.
- The subset of R^C which is contiguous with the image limits is called **background**, and the rest of the complement R^C is called **holes**.