Computer Vision

By

Alessia Pivotto

Lorenzo Orsingher

Handouts of the course 2023/2024

Contents

Chapter 1

Images and Videos

1.1 Definitions

1.1.1 Computer Vision

Is the science and technology of machines that see, where "see" in this case means that the machine is able to extract information from an image that is necessary to solve some task. The image data can take many forms, such as video sequences, views from multiple cameras, or multidimensional data from a medical scanner.

Some examples of applications of computer vision include systems for controlling processes, like an industrial robot or an autonomous vehicle, detecting events as for visual surveillance or people counting, or again organizing information for example for indexing databases of images and image sequences, even modeling objects or environments like in industrial inspection, medical image analysis or topographical modeling and interaction as the input to a device for computer-human interaction.

1.1.2 Acquisition

It refers to the process of transferring a portion of the real 3D world onto a 2D surface bringing a continuous-parameter real world into a discrete-parameter one. The acquisition process is the first step in the processing chain and it is the transformation of
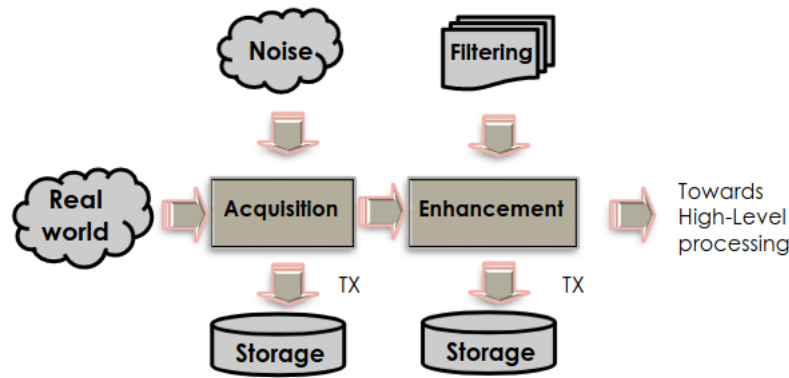
Figure 1.1: The processing chain(The first part is more about signal and video, in this course we'll focus mainly on the second half)

a physical signal into an electrical one, by means of a sensor. Where the sensor is a device that responds to a physical stimulus (light, heat, pressure, etc.) and produces an electrical signal. *PS: The representation is in a standard format.*

1.1.3 Digital Images

A digital image is a representation of a two-dimensional image as a finite set of digital values, called picture elements or pixels. The digital image contains a fixed number of rows and columns of pixels and so is a collection of coordinates. Pixels are the smallest individual element in an image, a single pixel represent a projection of a portion of the real world, holding quantized values that represent the brightness of a given color at any specific point. Pixels can be **Grayscale** or **Color**. Grayscale images are represented by a single component, typically 8bit, while color images are represented by three components, typically 24bits.

1.1.4 Sampling

Is the process of converting a continuous signal into a discrete signal. This is because the "real world" is a continuous function, and computers are digital. Analog video is a 1-D continuous function where one spatial dimension is mapped onto time by means of a scanning process, while digital video is instead sampled in a three dimensions (2D spatial and 1D temporal).

$$s_c(x_1, x_2, t) \quad\boxed{\text{Spatio-Temporal Sampling}}\quad s(n_1, n_2, k)$$
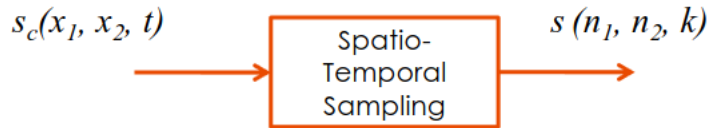
Figure 1.2: Sampling

In a Nutshell:

- Continuous signal

$$s_c(x_1, x_2)$$

- Spatial rectangular sampling

$$x_1 = n_1 \triangle x_1, x_2 = n_2 \triangle x_2$$

- So

$$s(n_1, n_2) = s_c(n_1 \triangle x_1, n_2 \triangle x_2)$$

- Once we have the digital format, we can manipulate data and apply filters, change colors, store and transmit.

For what concerns handling images we need only to know that pixels are numbered starting from the top left corner (so the top left corner is the origin, arrow down for the rows and right for the columns). The value of a pixel in a certain position is defined as $I(r, c)$, where r is the row index and c the column one. (Start at 0 or 1, depends on you

:D) Monochrome images have values normalized in the range $[0, 1]$, where 0 is black, 1 is white and the intensity is called *grey level*. While color pictures have 3 channels (RGB) and the values are normalized in the same way.

1.2 Color

But what is color? It's the attribute the human visual system associates to objects, more scientifically it's a mathematical relationship that combines different wavelengths. It's important to check whether something we see is what we expect, to recognize objects or to distinguish similar things. For example, a white car, that for us is obviously white, for a computer is a combination of red, green and blue, moreover it has some black points for the wires, some point gray due to the street, etc. For this reason let's talk about color perception. The human eye is like a camera with a focal length of about 20mm, where the iris controls the amount of light by adjusting the size of the pupil. The perception of color is possible through cones in the fovea that has around 100M receptors. Cones have peak responses on three main wavelengths, red (700nm), green (546.1nm) and blue (435.8nm).

But going back to the main topic, data can be processed locally but even transmitted remotely or archived on a storage unit. The problem is that images and videos require a lot of bandwidth, so we need to compress them with a codec. A codec is a device or computer program for encoding or decoding a digital data stream or signal in the compressed domain. It stands for coder-decoder, and it's a way to reduce the dimension of the file (e.g JPEG, MPEG, DIVX).

NB: Compression is lossy, so reduces quality (we lose some information) and can even introduces visual artifacts (such as blocking, blurring, chromatic aberrations(plus some noise from the sensor)). However, the loss is not a problem because the human visual system is not perfect. NB1: Exist even lossless compression, but it's not used much. NB2: Processing is typically done in the uncompressed domain. NB3: Raw images are vector of pixels.

1.2.1 Additive color model

The additive color model is a method to create color by mixing the primary colors.
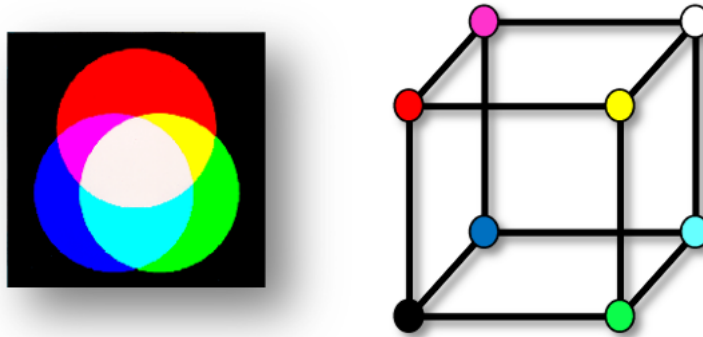
Figure 1.3: Additive Color Model

Colored beams are projected onto a black surface, then overlap so the human eye receives the stimula without generating interference, mixing the components and perceiving the resulting color. Starting from the primary colors RGB we can obtain:

- R+G = Yellow

- R+B = Magenta

- B+G = Cyan

- R+G+B = White

NB:Subtractive color is the inverse process.

About the way colors combined, we can have:

- Black: RGB(0,0,0)

- Green: RGB(0,1,0)

- Yellow: RGB(1,1,0)

- White: RGB(1,1,1)

- Gray: RGB(0.5,0.5,0.5)

Indeed, looking at the image above we can identify the gray scale along the diagonal connecting the black corner with the white corner.

NB: If we separate the three components and generate single images we notice that components are correlated, this means that the three version in grey-scale of the starting image carry almost the same amount of information.

In RGB we have a major response in the green component, while red and blue are less relevant. In addition, human eye is more sensitive to luminance and contrast variations than color so maybe a different representation would be more effective. Indeed, more effective is the YCbCr color space, where Y, the luminance is separated from Cb and Cr that are the chrominance. *PS: YCbCr is a generalization of YUV, just a matter of conversion matrices. (TODO: Downsampling)*

Another color space is the HSV, where H is the hue (color), S is the saturation (brightness) and V is the value (intensity).

Figure 1.4: HSV Color Space

Summing up just a little bit:

- RGB is used in general for visualization, in displays each pixel in composed by three phosphors (CRT) or LEDs (LCD);

- YUV is suitable for compression since we are less sensitive to chrominance variations and U and V can be downsampled;

- HSV is robust for computer graphics and image analysis.

1.2.2 Bayer Pattern

In the acquisition phase, light is captured by the CCD (Charge Coupled Device) that is an array of cells. Each cell is a photosensitive element that converts light into an electrical signal. The Bayer pattern is a color filter array for arranging RGB color filters on a square grid of photosensors. The best solution would be to have devices with 3 different CCDs and to correctly exploit the human eye response there would be three types of photosensors of color filters 50% green, 25% red and 25% blue. *NB: Green sensors are defined as luminance-sensitive elements, while the red and blue ones are defined as chrominance-sensitive.*

1.2.3 Quantization

Like in the mono-dimensional case, signals need to be quantized. That implies the definition of a number of levels to define our signal. Typically, the range 0-1 is quantized using 8bpp but even other representations with 10-12 bpp are available. But why 8bpp? well, 8bpp represent 256 levels, which is fine for the human eye, that can distinguish 100 levels of grey. Indeed, if we quantize with less that 6bpp (64 levels, minimum to ensure smooth pictures), then false contours will appear $\Rightarrow$ contouring.

1.2.4 Limits of the 2D

Still images provide a reliable information about static scenes. We lose motion information like temporal evolution of the scene, rapid changes, dynamics of motion (qualitative and quantitative) and even how subjects/objects relate one to each other. Moreover, analyzing a video provides a more consistent representation of the scene. It's closer to what humans do every day.

1.2.5 Video

A video is a sequence of 2D images that represent a projection of moving 3D scene onto the video camera image plane. PS: It's expected that adjacent frames are strongly correlated. For what concerns the resolution, the images are up to 50mp, while in video is typically lower, 4k is 3840x2160=8.3mp, 8k is 7680x4320=33.2mp and full HD is around 2mp. The reasons are that video can last hours, so we need to store a lot of data, and the human eye is less sensitive to resolution in motion. Just move on a little bit, once the image is acquired, what are the most relevant features we could be interested in? Maybe color and its distribution or presence of edges and contours. . . And again, if the analysis concerns a video, instead of an image, what is the added value? Consistency of the features over time, evolution of the scene and objects displacement and, of course, objects that may enter or exit the scene.

So in the end, what makes a scene complex to analyze and process?

- The presence of multiple objects;

- The presence of occlusions;

- The presence of shadows;

- The presence of noise;

- The presence of motion;

- The presence of illumination changes;

- The presence of clutter;

- The presence of non-rigid objects.

TODO: vedi se aggiungere gli esempi.

1.2.6 Histogram

The histogram is a simple way to describe the color distribution of a picture, indeed, it can be seen as a probability density function and it represents the occurrence of all

colors on a graph. In other words it's a statistical representation of the pixel values. For an MxN image

$$hist(p) = \frac{I(x,y)}{MxN}$$

Where $I(x,y)$ is the intensity of the pixel at position (x,y) and MxN is the total number of pixels in the image.

The equation holds for one component, so in case of more components, a histogram can be obtained from each of them. So, what kind of information can we obtain from the
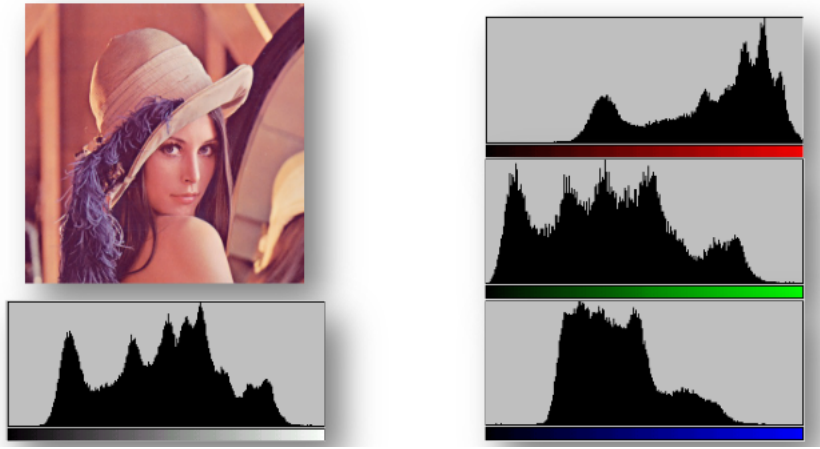

Figure 1.5: Histograms of Lana

histogram?

- Is the image dark or bright?

- Are colors distributed equally?

- Are there dominant colors?

- Are there colors missing?

Some applications:

- Is the illumination of a scene correct for environmental monitoring?

- How has the background changed in the past 2 hours?

- Can I use the histogram to determine whether the moving object I'm observing is A or B?

In general it can be seen as a "signature" to be applied in different domains.

1.2.7 Operations

Stretching:

A simple operation to change the dynamic range of the image by stretching the histogram and increasing the contrast. Usually apply a piecewise linear function.
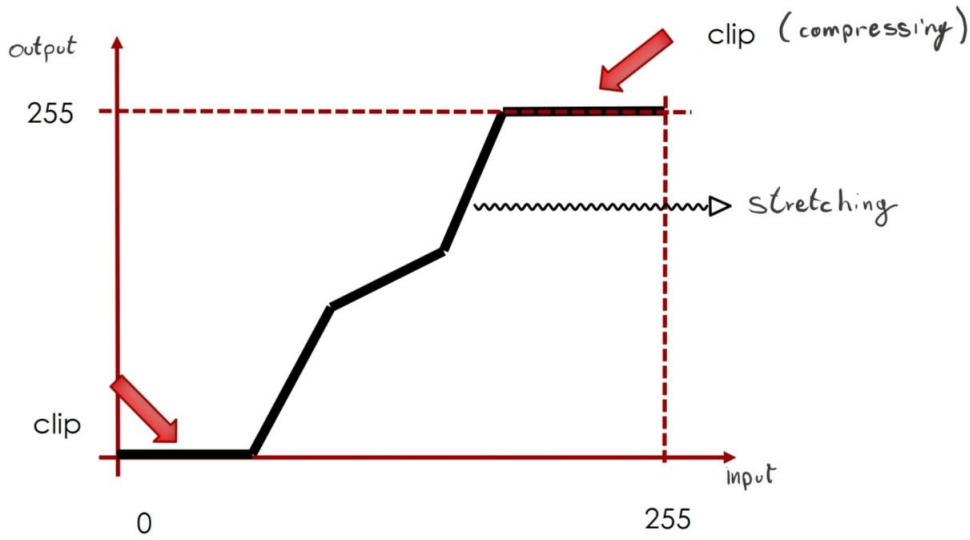

Figure 1.6: Stretching

Equalization:

Working on the statistics of the pixels it is possible to improve the quality of an image. Ideally we'd like to obtain a FLAT histogram, and to do so we have to compute the cumulative histogram (equivalent to CDF).

$$CHist_I(p) = \sum_{k=0}^{p} hist(k)$$

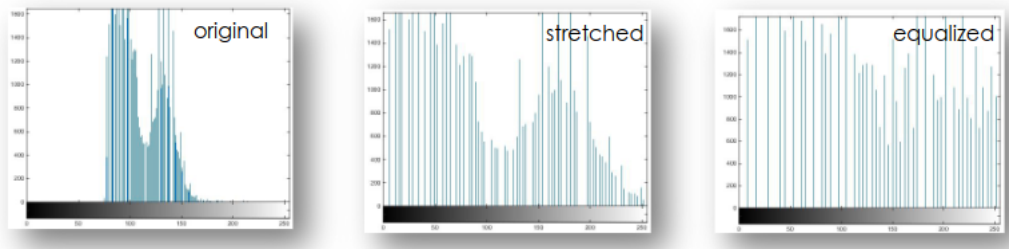$$hist_eq(p) = \frac{CHist(p) - CHist_{min}}{M \times N - 1} \times 255$$

Figure 1.7: Different operations on the histogram

1.3 Edge extraction

We perceive objects through color (appearance model) and shape. Shape is the boundary between the object and the rest of the picture, and it helps us recognizing things. A sharp change in image brightness can be seen as an edge.

PS: Environmental conditions, such as cast shadows or surface orientation, can generate changes. Moreover edges are not always meaningful, for example in textured areas or in the presence of noise.

Steps:

- Determine intensity, and possibly the direction of an edge for each pixel location through gradient or Laplacian;

- Find a threshold and binarize.

NB: The first step is the most difficult one. Different tools are available but we'll focus here on the gradient-based algorithms.

Edge extraction by gradient analysis

As for mono-dimensional signals, the goal is to find maxima and minima but in this case, we also have a direction. This means finding a gradient along a line r oriented in the direction θ.

$$\frac{\partial f}{\partial r} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial r} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial r} = f_x \cos(\theta) + f_y \sin(\theta)$$

The edge extraction process consists of computing the 1st order derivative in two orthogonal directions, f_1 and f_2. To each of them we associate an amplitude. Going a

little more into the concrete, for edge detection typically we use FIR filters, that are linear and shift-invariant. The most common are the Prewitt, Roberts and Sobel operators.

For the first two we have to choose a convolution mask and apply it to the picture. The convolution is computed for both masks and a threshold is chosen to highlight only the strongest edges.

Roberts operator: $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ $\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ **Prewitt operator:** $\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$ $\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$

NB: The Prewitt operator is preferred because it is isotropic $\Rightarrow$ you know where is the center.

For the Sobel operator, we have to apply two masks, one for each orthogonal direction. The mask is a 3x3 matrix and the gradient is computed for each point. Then we have to threshold the result.

Sobel operator: $D_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$ $D_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$

The convolution with the FIR masks is performed similarly to the 1D convolution. Take the mask, rotate, slide from left to right and associate to the central point the value of the convolution. (formule ancora da mettere, non avevo voglia sorry)
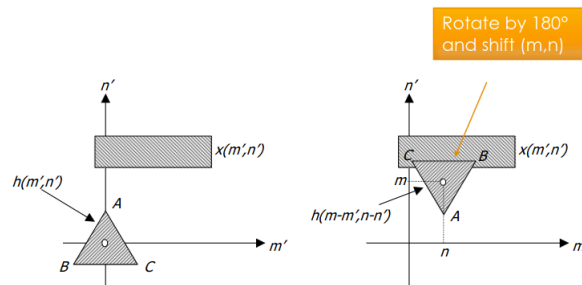

Figure 1.8: How to (convolution edition)

1.4 Filters

1.4.1 Low-pass filtering

The easiest way is to average the values in the sliding window. The window is a matrix of size $m \times n$ and the result is the average of the values in the window.

$$I_{LP}(x, y) = \frac{1}{mn} \sum_{x=-a}^{a} \sum_{y=-b}^{b} I(x, y)$$

Preserve low-frequency components and remove high-frequency ones. It's a weighted sum of low high frequency functions. Cut the high frequency components above the threshold and smooth the image. Average filtering $\Rightarrow$ average of the pixels in the window. Bigger the filter, more the smoothing (the difference is smaller). It helps denoising the signal.

1.4.2 Gaussian filtering

Gaussian filtering is a low-pass filter that is used to remove the high-frequency components of the image. It is a 2D convolution filter that is used to blur the image and remove noise. Gaussian try to reconstruct a gaussian function, so it's isotropic and the mask is not flipped. The mask is a 2D Gaussian usually centered in the center of the filter mask, set big enough to make sure values in the mask are integer. Using a Gaussian as a mask is convenient as in the Fourier domain it is still a Gaussian, it is isotropic (simmetric) and no need to flip the mask (no rotation in convolution).

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

the bigger the better. The central pixels are more considered, far pixels have a lower impact.

1.4.3 Median filtering

Median filtering is a nonlinear method used to remove noise from an image. It is a spatial domain method that replaces the pixel value with the median value of the

noise   low-pass   median
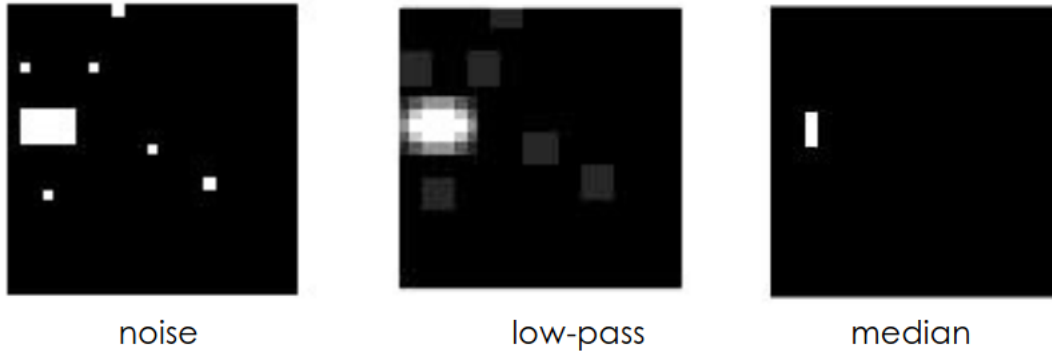
neighborhood pixels. It is particularly effective in removing salt and pepper noise. It is a good method to use when the image is corrupted by impulse noise. In case the noise is zero-mean, smoothing is fine. But when you have spikes, LP filtering blurs also the noise. Indeed, LP spread the dot, so it generates artifact. The noise is less visible but larger. Median avoid this issue, remove noise, no linear function, no convolution, sorting operation for every shift, filter ranks the values of the filter from bigger to smaller and store the median value in the central pixel. Bigger could introduce artifacts, so it's better to use a small filter. The noise that remains is because the size of the mask is not big enough to cover the noise.

1.5   Morphology

Morphology refers to the shape of a region. The goal is to check whether a certain shape fits into another, check whether a picture has holes of a certain size, remove areas smaller than a threshold and with certain shape, etc. *PS:It refers to non-linear filtering.* It uses a structuring element, a known arbitrary shape, and the operations are performed by sliding the structuring element over the image. The structuring element has an origin (typically central point, but not necessarily) and the operation is performed by comparing the structuring element with the image. There are four main operations: Erosion, Dilation, Opening and Closing.

Erosion and Dilation are self-explanatory, the first one reduces the area of a shape, the second one enlarges it. Instead, opening and closing are a combination of erosion and

dilation. Opening gets rid of small portions of the image close to the boundaries of relevant areas, while closing fills holes and makes region boundaries smoother. *Concerning structuring elements, depending on the type of shape we want to edit, the right element must be chosen.*

1.5.1   Dilation

Dilation is a morphological operation that is used to enlarge the boundaries of regions of foreground pixels in an image. It is used to merge the objects in the image. The structuring element is a binary mask that defines the neighborhood of the pixel. The dilation of the image is obtained by sliding the structuring element over the image and placing the origin of the structuring element at the center of the pixel. More formally, the dilation of an image B by a structuring element S is given by:

$$B \oplus S = \cup_{b \in B} S_b$$

Sweep the structuring element on the whole image, as the origin of the structuring element touches a "1" of the image all pixels of the structuring element are OR'ed to the output image. *PS: OR/union $\Rightarrow$ every time that the structuring element touches a 1, the output is 1, so it enlarges the shape.*

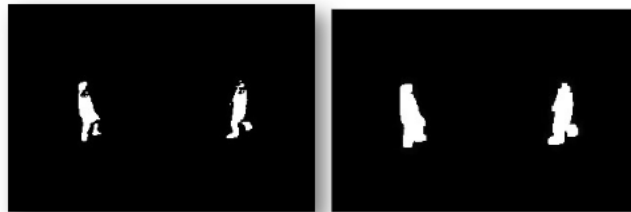

Figure 1.9: Dilation

1.5.2   Erosion

Erosion is a morphological operation that is used to reduce the boundaries of regions of foreground pixels in an image. It is used to separate the objects in an image. Sweep

the structuring element on the whole image, at each position where every 1-pixel of the structuring element covers a 1-pixel of the binary image the binary image corresponding to the origin is OR'ed with the output image. More formally, the erosion of an image B by a structuring element S is given by:

$$B \ominus S = \{b | b + S \in B \quad \forall s \in S\}$$

Erosion of A by B can be understood as the locus of points reached by the center of B when B moves inside A.

Figure 1.10: Erosion

PS: se prima bastava che il centro toccasse un 1, ora TUTTI devono toccare un 1.

1.5.3 Closing and Opening

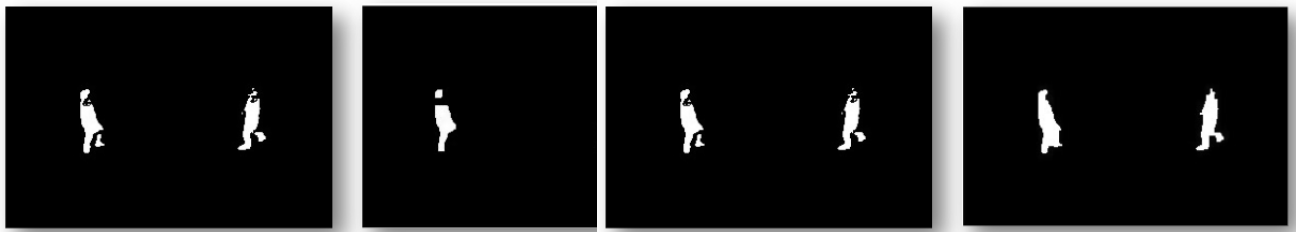- Closing $\Rightarrow$ first dilatate and then erode;
  Non-contiguous regions are firs merged, then the boundaries are refined.

- Opening $\Rightarrow$ first erode and then dilatate;
  First small areas are removed, then residual information is refined.

(a) Opening   (b) Closing

This operations are in general non-reversible, so the result depends on the order of the operations.
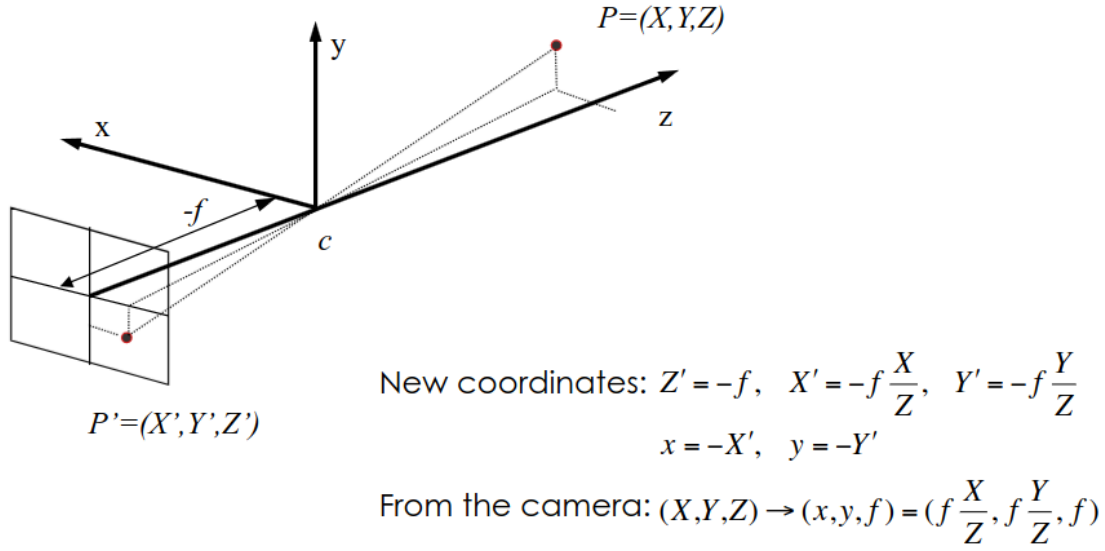
Chapter 2

Models

Let's begin the chapter with a question. Why do we need models? well, they represent a good approximation of the real world and allow describing events through a parametric representation (parameters can be extracted and used for processing). In the context of computer vision, models are used to describe the geometry of the world, the appearance of objects, and the imaging process. In this chapter, we will discuss the most common models used in computer vision. We will start with the pinhole camera model,then we will see the perspective projection model and finally, we will discuss the appearance model.

2.1   The Pinhole Camera Model

The pinhole camera model is the simplest model used to describe the imaging process. It is based on the principle that light travels in straight lines. The model is composed of a pinhole, a plane, and an image plane. The pinhole is a small hole in the plane, and the image plane is placed behind the pinhole. The light rays coming from the scene pass through the pinhole and project the scene onto the image plane.

PS: If the object is far, it appears small in the image. If the object is close, it appears large in the image.

New coordinates: $Z' = -f, \quad X' = -f\dfrac{X}{Z}, \quad Y' = -f\dfrac{Y}{Z}$

$$x = -X', \quad y = -Y'$$

From the camera: $(X,Y,Z) \rightarrow (x,y,f) = (f\dfrac{X}{Z}, f\dfrac{Y}{Z}, f)$

2.1.1   Features of the Pinhole Camera Model

- Parallel lines in the world converge to a single point in the image;

- Parallel lines on the same plane lead to *collinear vanishing* points;

- The line is called the *horizon* for that plane;

- Vertical lines are perpendicular to the horizon.

2.2   Projection

The projection is the process of mapping 3D points (+time) to 2D points. The projection of a 3D point onto the image plane is defined by a line that passes through the point and the center of projection. The intersection of this line with the image plane gives the 2D projection of the 3D point.

$$f : \mathbb{R}^4 \rightarrow \mathbb{R}^3$$

$$f(X, Y, Z, t) = (x, y, t)$$

NB: contiuous variables.

Exist two types of projection:

- Perspective projection;

- Orthographic projection.

2.2.1   The Perspective Projection Model

The perspective projection model is an extension of the pinhole camera model. It is used to describe the imaging process in a more realistic way. The model is based on the principle that light travels in straight lines and that the image is formed by the intersection of these lines with the image plane. For simplicity we usually consider the image plane on the same side of the "real world", to avoid the picture flip. The center
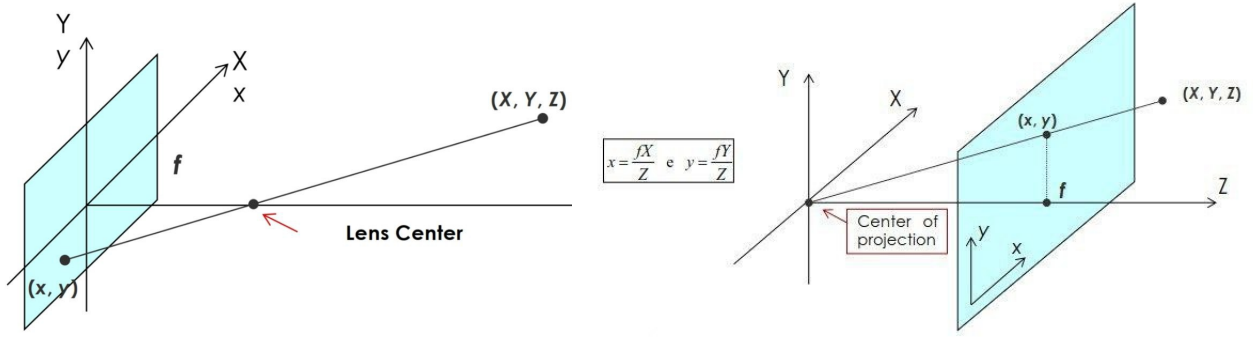
of projection corresponds to the origin of the 3D space. The plane (X,Y) is parallel to (x,y).

PS: This approximation is allowed when Z>>f.

From a computational point of view nothing changes, we simply shift the image plane from back to front in order to have the image straight. For what concerns the projection, we can use the same formulas as before but we remove the minus f from the denominator(too small).

$$\begin{cases} \frac{x}{f} = \frac{X}{Z-f} \\ \frac{y}{f} = \frac{Y}{Z-f} \end{cases} \Rightarrow \begin{cases} x = \frac{fX}{Z-f} \\ y = \frac{fY}{Z-f} \end{cases} \Rightarrow \begin{cases} x = \frac{fX}{Z} \\ y = \frac{fY}{Z} \end{cases}$$

2.2.2 The Orthographic Projection Model

The orthographic projection model is a simplified version of the perspective projection model. It is assumed that all rays originated from the 3D object, and from the scene in general, are parallel among each other. In the drawing, the image plane is parallel to (X,Y).
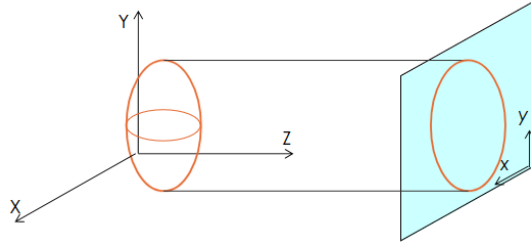
Assuming that the image plane is parallel to (X,Y), the orthographic projection can be simply described in Cartesian coordinates as:

$$\begin{cases} x = X \\ y = Y \end{cases}$$

Or in form of a matrix:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

NB: The distance of the object from the camera does not affect the intensity of the image projected onto the 2D plane. It is a good approximation when the distance of the object is much bigger than the depth of the object itself.

2.3 Illumination Models

Illumination is a key element, it models the appearance/perception of objects. When a light source hits an object, light can be absorbed, reflected, or transmitted. It's very complex to model, we perceive objects because they reflect light in specific wavelengths. Reflection can be specular or diffuse. Specular means that more energy is concentrated

in the light source direction. While in diffuse the energy is constant in all directions, and the position of the observer is irrelevant. Different surfaces vary in specularity, some of them (matte) reflect light uniformly in all directions, while glossy objects reflect light in specific directions. It also depends on the distance and the inclination of the light source. The problem is to determine how the surface is irradiated by the light source.We can make an assumption that the light source is far away, so we can assume that all rays can be represented by a single unit vector s (orthographic projection). For each surface element (btw dashed lines) the light is irradiated considering the cosine of the angle between the surface normal and the light direction.
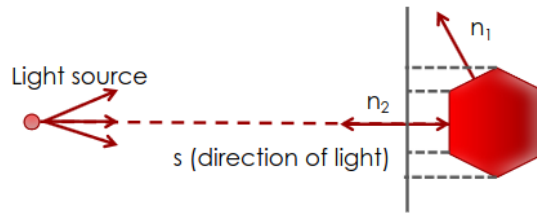
NB: intensity we perceive is equal to dot product of the normal of the surface and the light direction. Dot product $\Rightarrow$ cosine $\Rightarrow$ more close to 90 degrees, less we see.

2.3.1 Lambertian Reflectance

Is a model for diffuse reflection, it is based on the assumption that the surface is rough enough w.r.t. the light wavelength. This means that each surface element reflects light evenly in all directions. The luminance of the surface is the same regardless of the viewing angle. The specular component is neglected.
Assumes that the surface we're dealing with is ideal, so looking at it from any angle we'll see the same intensity.

$$I = \rho n \cdot s$$

Where:

- I is the intensity of the light;

- ρ is the albedo, the ratio of the reflected illumination to the total illumination,

intrinsic property of the surface (not true hihih, some surfaces may reflect light differently depending on the view angle);

- n is the normal to the surface;

- s is the direction of the light.

An element is not visible if the angle between the normal and the light direction is greater than 90 degrees. The pixel in image

$$I(r, c)$$

depends on the light source direction and the normal of the element direction.

2.4 Remarks on cameras and lenses

Cameras are equipped with lenses (no pinhole in real life!). This means that the object is on focus if the distance from the center of the camera and the image plane obeys to the thin lens equation. If not, we have aberrations.We typically assume the object is on focus, but let's see what this means first.

2.4.1 Typical issues with lenses

Spherical aberration:
The lens does not focus light rays that strike the lens far from the center. This means that the image is not sharp (Causes blur). In simple terms the lens does not focus all the rays in the same point, so the light is reflected wrongly because the lens is not properly manufactured.

Chromatic aberration:
The lens does not focus all the colors in the same point, it's not able anymore to convey correctly the ray (at different locations). es tv a tubo catodico.

Vignetting:
The lens does not focus all the rays in the same point, so the image is darker in the

corners. The spreading of light is not uniform due to the impurity of glass.

Barrel distortion:

The focal length is too short or the lens is too wide, so the image is distorted and the lines are not straight anymore. In simple terms, the lens is able to grasp a wide area of view. To compensate we can use a software to correct the distortion. (Linee storte possono dare problemi nell'analisi del percorso tipo... problema della distorsione-¿ecco perchè i cell moderni hanno più lenti)

2.4.2 Focus

Pinhole is an abstract model, in real life we have lenses. The lens is able to focus the light rays in a single point. The distance from the center of the camera and the image plane obeys to the thin lens equation.
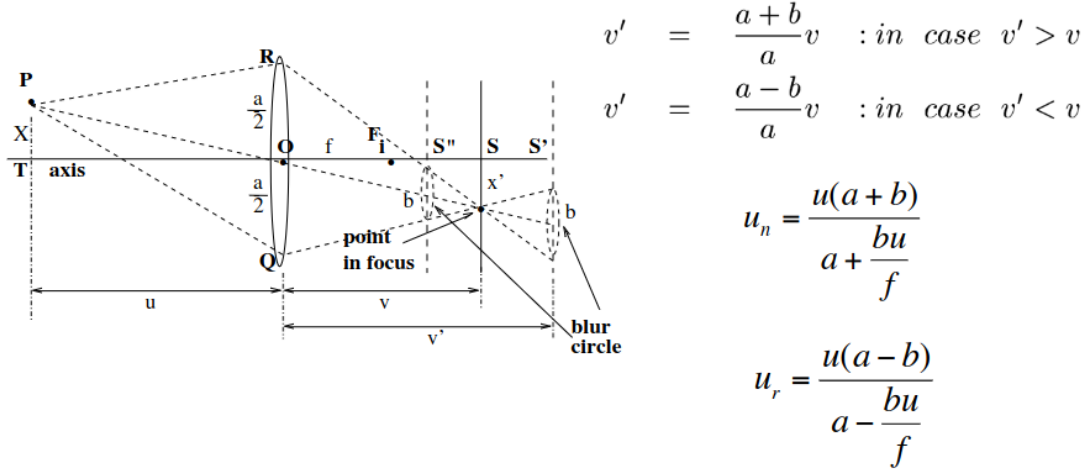
$$\frac{1}{f} = \frac{1}{u} + \frac{1}{v}$$
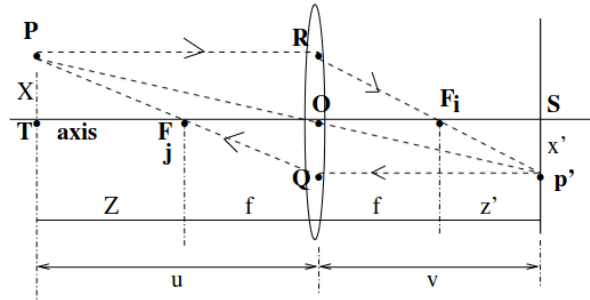
If we move the image plane, point p' is out of focus $\Rightarrow v$ changes $\Rightarrow v'$. If P is moved u changes $\Rightarrow u'$. The result is that the image is blurred on the image plane (instead of a point I see a circle, the rays of light don't converge).

NB: If the object is too far you can try to set the focal length but it won't work.

We want to configure the camera in order to have the quantity b as small as possible. Notice that:

- In general $u > f = u_n < u$;

$$v' = \frac{a+b}{a}v \quad : in \ \ case \ \ v' > v$$

$$v' = \frac{a-b}{a}v \quad : in \ \ case \ \ v' < v$$

$$u_n = \frac{u(a+b)}{a + \dfrac{bu}{f}}$$

$$u_r = \frac{u(a-b)}{a - \dfrac{bu}{f}}$$

- If f becomes smaller, u_n is closer to the camera;

- If f becomes smaller, u_r is farther to the camera;

- $u_r > u$;

- If $u->\infty$ rays are parallel and converge to the camera center;

- The difference between the far and near planes limiting b is called *depth of field.*

Autofocus

Is the capability of focusing a specific portion of the image, it can be active, passive or a combination of both.

Active:

The camera emits a signal and measures the time it takes to return. The time is used to compute the distance of the object from the camera. It is mainly used for point-and-shoot cameras. The issues are related to obstacles and glossy and bright surfaces.

Passive:

The camera uses the contrast of the image to focus. It is mainly used for professional cameras. It is more expensive SLR (single-lens reflex) cameras, it uses the phase detection method. Distance computed using image analysis: take a strip of pixels and analyze the distribution, then if values are too similar the object is out of focus, if

contrast is high the object is on focus. The problems are related to flat surfaces, to low contrast and low light. Good cameras compute the metric on the vertical and horizontal axes.

Chapter 3

Motion Detection

Chapter 4

Motion Tracking

Chapter 5

Geometry

Chapter 6

Local Feature Extraction

Chapter 7

Classification

Chapter 8

Motion Capture

Bibliography