Computer Vision

Diego Oniarti

Anno 2024-2025

Contents

1 24-02-2025

Main topics of the course

1. acquisition
2. motion detection
3. motion analysis
4. stereo/multi -view
5. 3D point cloud
6. feature extraction / classification

Evaluation

The written exam will be 40% of the vote.
You can choose between a project and an oral exam. If you're not satisfied with the result of the written you can take an oral later.

Reading groups are also a thing.

- 24 mar: teams + project ideas
- 31 mar project titles assignment for those who haven't chosen one

For the project we'll use python, openCV, and ffmpeg.

You can deliver the project and written exam in different sessions. But the written exam expires in 1 year.

2 28-02-2025

Bayer Pattern

is a pattern used in camera sensors to optimize the distribution of colors. Since the human eye is more sensitive to green light, the pattern is composed of a checkerboard pattern where one color is green and the other is divided between red and blue.

Quantization

Usually we use 8bpp (bits per pixel) because it is byte aligned and because it's plenty enough for the human eye. At lower bpp, contouring appears.

Video

Static images loose the temporal and movement information, so we need videos. The frame rate of an image must be compliant with the thing that is being captured. With an high enough rate we can ensure a smooth transition between frames without loosing information.

This is the reason video-cameras usually have a lower resolution that photo-cameras. With too high of a resolution, there is too much information that needs to be processed and it can't be done at a fast enough rate.

Humans also focus less on image quality while watching a video, as they're more captivated by the evolution of events than the single frames.

Relevant features

The relevant features in an image are color, edges, and contrast. In a video the features are the same but also their progression through time.

Image compression

Image take up a lot of space and videos take up even more. Compression standards exist to reduce the amount of data required.
Compression requires there to be an **encoder** and a **decoder**. Some examples are JPEG, MPEG, and DIVX. Both visualization and processing are executed on the uncompressed image, since humans can't visualize raw compressed data, and filters can't work on the compressed image.

Some compression algorithms are lossy while some other are lossless.

Histogram

is a simple way to describe the color distribution of a picture by approximating a probability function.

$$hist(p) = \frac{\#pixels : I(x,y) = p}{N \cdot M} \approx f(p)$$

Where N, M are the size of the picture in pixels.

Various filters can be applied to an image by manipulating the histogram with operations like stretching and thresholding.

We can equalize an histogram defining a partial sum $CHist_I(p) = \sum_{k=0}^{p} hist(k)$ e assegnando $hist_{eq}(p) = \frac{CHist(p) - CHist_{min}}{M \cdot N - 1} \cdot 255$.
Even equalizing we can not get to a flat histogram, but we can do our best to get to that point.

Edge extraction

Usual Sobel su X, Y, thresholding, etc. Convolution in 1D and its natural translation in two dimensions.
A convolution in the space domain is equivalent to a product in the frequency domain and vice versa.

Low-pass filtering

The easiest way to implement a discrete low pass filter is to design a kernel that takes the average of the values surrounding a pixel.
A better visual result is given by a Gaussian filter. Funnily enough, the Fourier transform of a gaussian curve is still a gaussian curve.

Low Pass vs Median

Low pass filtering can reduce noise in an image, but it also spreads the noise over the image. In some cases this may be undesirable. The common approach would be to threshold the filtered image, but finding the threshold value can be cumbersome.

Some other filters to denoise is the **median filter**. It's not *isotropic* and it doesn't work with a normal convolution, but it requires a *sorting* operator.

Gaussian and averaging filters introduce in the image values that were not in the original image. The median filter, instead, only "selects" values from the image, not inventing new ones.

3 Morphology

A form of non linear filtering that refers to the shape of a region.

Goals:

- check whether a certain shape fits into another
- check whether a picture has holes of a certain size
- remove areas smaller than a threshold

Binary morphology

We need a **binary image**[1] and **structuring elements** and implement four main operations:

- erosion
- dilation
- opening
- closing

Erosion and dilation are intuitive, enlarging or reducing the size of a region. Opening and closing are combinations of erosion and dilation in sequence.

Structuring elements can be squares, circles, other primitives, or custom shapes. For every structuring element we need to define a "center". It is usually the geometric center of the image but it doesn't have to be.

3.1 Dilation

Dilation performs an $\oplus$ (or) operation between the image and the element. More specifically:

- sweep the element over the image
- if the origin of the element touches the image (a 1 in the image).
  - perform the or, "stamping" the element onto the image

It is important to note that the output of the filter has to be stored in a separate image, to avoid it recursively dilating a pixel across the whole image.

3.2 Erosion

Erosion works in a similar way by scanning the element over the image: We don't check with the center of the element anymore but we "activate" the filter

[1] A binary image is not grayscale but an image composed only of true and false

when every 1 in the filter overlaps a 1 in the image.

Question: In the output image, do we only put the center of the element or the whole element?

3.3 Closing and Opening

Closing: dilate and then erode
Opening: erode and then dilate.

Closing fills the holes in the image with the dilation, and then removes the excess added by the first operation with erosion.
Similar but inverse result is gotten by opening. The holes are enlarged, eating away at the shape. Then the remaining bits are consolidated.

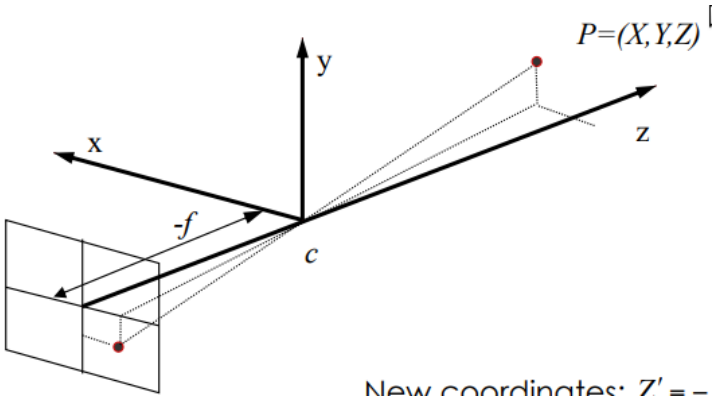
4 Models

4.1 Pinhole camera model

It consists of a box with a small hole on one of the walls. The light that passes through that hole projects a reflected image of the outside world onto the opposite wall.
One clear problem is that only a small amount of light can pass through the hole, making the projection very dim. We can fix this by making the hole bigger but this also makes the image blurry.

If the *image plane* is the plane opposite to the camera, the *virtual image plane* is an imaginary plane parallel to the image plane and equally spaced with the pinhole in the other direction.

Model to reality How do we map the pixel location to a location in space? Following the pinhole camera model we would also need the focal length f of the camera.
Then we know the point in space is somewhere on the line that passes through the pixel and the origin shifted by the focal length.



new coordinates From the camera

$$\begin{cases} Z' = -f \\ X' = -f\frac{X}{Z} \\ Y' = -f\frac{Y}{Z} \end{cases} \qquad (X, Y, Z) \to (x, y, f) = (f\frac{X}{Z}, f\frac{Y}{Z}, f)$$

It's easy to see that we loose some information, since a point in the camera plane is mapped to a whole line in the real world, loosing the distance.
We can approximate the distance with context clues, knowing additional information about the space etc. but these are not means of **measuring** the distance, only approximating it.

Multiple cameras To solve the problem of the loss of information we can use *two cameras*, or even more, to measure depth.

Properties of the pinhole model

- Parallel lines converge to a single vanishing point
- Parallel lines on the same plane lead to collinear vanishing points
- The line is called the horizon for a plane
- Vertical lines are perpendicular to the horizon

4.2 Orthographic projection model

The *orthographic projection model* assumes that all rays originated from the 3D object and from the scene are parallel among each other. The image plane is parallel to (X, Y)

Mathematically we're just taking the x and y of the point we're capturing, completely disregarding the depth component. This works since the model

assumes object behave the same way regardless of distance.

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

5 Illumination models

Illumination is an important component in understanding the content of an image. Different lighting can change the colors perceived in the image, change the shape of perceived edges, etc.

Some materials and surfaces respond to light in different manners, depending how much they **absorb**, **reflect**, and **transmit** it.

Reflections can be

- Specular: more energy is concentrated in the light source direction
- Diffuse: constant in all directions

Surfaces vary in *specularity*, going from matte to glossy. Glossy materials are harder to work with, because they introduce things in the image that are not "real".

Illumination from one light source Problem: determine how the surface is irradiated by the light source assumption: light is far, we can assume all rays can be represented by a single unit vector s (ortho projection)

For each surface element the light is irradiated considering the cosine of the angle between the surface normal and the light direction

5.1 Lambertian surface

Model for diffuse reflection, so the specular reflections are ignored. It is possible to make this assumption when the surface is rough enough.

The luminance of a surface in this model is the same regardless of the viewing angle.

This model assumes every surface has a property ρ (*albedo*) that describes how much of the light is reflected by the object.