

# Computer Vision - Notes

Marco Mustacchi

## 1 OPENCV

Preprocessor: source file  $\rightarrow$  code edited

Compiler: code edited  $\rightarrow$  object file

Linker: object file  $\rightarrow$  executable

Dynamic libraries

- $\rightarrow$  save disk space (one installation serves all)
- $\rightarrow$  can be recompiled without touching the executables
- $\rightarrow$  SO (Shared Object) under Linux, DLL (Dynamic Linking Library) under Windows

Static libraries

- $\rightarrow$  generate executables that can't be broken at a later stage
- $\rightarrow$  self-contained

standard library

Standard Template Library (STL)

- $\rightarrow$  a special section of the standard library
- $\rightarrow$  based on the concept of template

OpenCV library

OpenCV depends on other libraries  $\rightarrow$  not self-contained

Header of OpenCV files  $\rightarrow$  .hpp extension

`cv::Mat`  $\rightarrow$  elements accessed using `img.at<_T>(i,j)`

`cv::Vec`

- $\rightarrow$  elements accessed using `[]`
- $\rightarrow$  class used for dealing with tuples
- $\rightarrow$  used to describe the triplet of color pixels `img.at<Vec3b>(i,j)[0]`

## 2 IMAGE PROCESSING

### 2.1 IMAGE CODING

coordinate system (c,r)

Spatial resolution  $\rightarrow$  # of pixels per unit distance  $\rightarrow$  smallest detectable detail in the image

Depth  $\rightarrow$  # bits per pixel

Gray-level/intensity resolution  $\rightarrow$  # bits per pixel  $\rightarrow$  smallest detectable change in gray level

Contrast  $\rightarrow$  difference between highest and lowest intensity value in the image

### 2.2 TRANSFORMING AN IMAGE

#### GEOMETRIC TRANSFORMATIONS

Two steps:

1. coordinate transform  $\rightarrow$  works on geometrical points
2. image resampling  $\rightarrow$  works on pixels

difference between forward and backward mapping

Affine transform

Preserves  $\rightarrow$  point collinearity  $\rightarrow$  distance ratios along a line

## SINGLE PIXEL OPERATIONS

functions that change the gray levels of an image

examples:

Negative

Logarithm  $\rightarrow$  if underexposed

Gamma (tunable):  $\rightarrow$  gamma<sub>1</sub> if underexposed,  $\rightarrow$  gamma<sub>2</sub> if overexposed

Measurements for contrast: RMS/RMSC

Slicing

$\rightarrow$  enables to consider only a subset of the given image intensity slicing

$\rightarrow$  Logarithm  $\rightarrow$  intensity slicing

$\rightarrow$  bit plane slicing

[see graphs for: contrast stretching, thresholding, intensity slicing x2]

## 2.3 HISTOGRAM

The CDF (cumulative distribution function) is the T(r) transformation used to equalize the histogram  
Output not perfectly flat (caused by the discrete nature of data)

1. Histogram equalization
2. Local histogram equalization
3. Histogram specification
  - $\rightarrow$  define the desired probability density function
  - $\rightarrow$  evaluate

## 3 FILTERING

In general, the bigger the filter, the higher the smoothing (or in general the effect)

Image restoration  $\rightarrow$  based on  $\rightarrow$  averaging  $\rightarrow$  Gaussian filter  $\rightarrow$  Median filter [slides 101: Vedere differenza tra sobel mask e laplacian-derived mask, both cases non isotropic (4 directions available)] (in teoria laplacian isotropic + double line effect, in pratica implementando con mask e' anisotropic)

The filter weights can change the image brightness, if  $\sum(w_i)=1$  unchanged, if  $\sum(w_i)<1$  brightness decreased, if  $\sum(w_i)>1$  brightness increased

In general: Average filter  $\rightarrow$  smoothing Derivative filter  $\rightarrow$  sharpening (affilamento)  
+ Derivative filters highlights edges in the image

Gaussian filter

- $\rightarrow$  spatial filter (the weight assigned to each pixel depends on how far it is from the central one)
- $\rightarrow$  takes only spatial distance into account
- $\rightarrow$  Kernel is everywhere the same
- $\rightarrow$  losing edge information

Bilateral filter

- $\rightarrow$  consider not only spatial distance, but also color distance
- $\rightarrow$  non-linear filter
- $\rightarrow$  Kernel is not everywhere the same
- $\rightarrow$  edge preserving

## 4 IMAGES IN FREQUENCY

Aliasing caused by Undersampling

- $\rightarrow$  artifacts
- $\rightarrow$   $\rightarrow$  can be compensated by means of a Low Pass Filter

The DFT of an image can be decomposed into

- $\rightarrow$  Spectrum

→ Phase

Translation → does not affect the spectrum

Rotation → affects the spectrum Filtering in frequency

Low pass filter → Smoothing (ringing effect is present)

→ Ideal → cut-off frequency → ringing effect

→ Butterworth

→ Gaussian

High pass filter → Sharpening (ringing is present as in lowpass filtering)

→ Ideal

→ Butterworth

→ Gaussian

Band pass filter

Band reject filter → Notch filter

## 5 EDGE DETECTION

Gradient edge detector

→ gradient is a vector pointing towards the fastest varying direction ( $\perp$  to the edge direction)

→ magnitude and phase are scalars

## 6 SEGMENTATION

Thresholding → not connected regions → discontinuity

Region Growing

→ connected regions → similarity

→ 3 elements: image, seed array, predicate

Watershed

→ connected regions → similarity

→ flood the surface from the minima

Watershed: Questi bacini vengono allagati e vengono identificate le aree in cui le acque di piena dei diversi bacini si incontrano. (Morphological interpretation → a set of subsequent dilations)

In queste aree vengono costruite delle barriere sotto forma di pixel.

Di conseguenza, queste barriere agiscono come partizioni nell'immagine e l'immagine viene considerata segmentata.

→ Infatti se considero solo grayscale, le watershed lines possono essere molto spesse, in questo modo ottengo solo delle lines

Watershed on Gradient → catchment basins correspond to homogeneous graylevel regions → extract uniform regions (colore costante)

Watershed with Markers

→ the number of markers determines the number of regions that will be created by the algorithm (evitare overfitting)

→ Internal markers:

Mean shift: Kernel density estimation

→ Kernel density gradient estimation

→ steepest ascend method

Markov Random Fields (MRF)

Segmentation as a per-pixel labelling task

Assign a label to each pixel in order to minimize an energy function

Energy function composed of two terms

→ Data term → depends on specific feature at the pixel location

→ Smooth term → depends on the labels in the neighboring pixels (commonly 4 neighbors considered)

Active Contours: The primary use of active contours in image processing is to define smooth shapes in images and to construct closed contours for regions.

The curvature of the models is determined using several contour techniques that employ external and internal forces. External energy is described as the sum of forces caused by the picture that is specifically used to control the location of the contour onto the image, and internal energy, which is used to govern deformable changes.

Active snake models, often known as snakes, are generally configured by the use of spline

## ACTIVE CONTOUR MODEL

Contour is moving,  
scissors → user input  
snakes

- without any interaction with the user
- energy balance
  - Internal energy → keeps the snake in shape with the contour
  - External energy → what inside the image attracts the snake, for instance strong edges

Internal energy: where  $\alpha(s)$  and  $\beta(s)$  are user-defined weights; these control the internal energy function's sensitivity to the amount of stretch in the snake and the amount of curvature in the snake, respectively, and thereby control the number of constraints on the shape of the snake. In practice, a large weight  $\alpha(s)$  for the continuity term penalizes changes in distances between points in the contour. A large weight  $\beta(s)$  for the smoothness term penalizes oscillations in the contour and will cause the contour to act as a thin plate.

## 7 FEATURES

Feature detection (keypoint detection) → find points easily recognizable

Feature description → describe a region around the keypoint to make matching easy

Feature matching → find similar descriptors in two images

Feature tracking → find similar descriptors in a neighborhood in the same image

similarity is applied to the descriptor (not to the keypoint) using a distance!!

## HARRIS CORNER

Consider a patch and a shifted version of the patch

Similarity is measured by means of the autocorrelation (a weighted sum function of the displacement)

Approximate this autocorrelation using the Taylor series (use of derivatives)

Autocorrelation function is weighted, two main choices

- Box (1 inside the patch, 0 elsewhere)
- Gaussian

Study of the eigenvectors and eigenvalues

- two small eigenvalues → flat region
- one large and one small → edge
- two large eigenvalues → corner

!! Not invariant to scaling !!

## USAN/SUSAN

MSER (Maximally Stable Extremal Regions) → Blob detector → apply a series of threshold

## SIFT

features are local → robust to occlusion

approximating LoG using DoG

Steps:

Keypoint localization

- Interpolation of nearby data for accurate position → The interpolation is done using the quadratic Taylor expansion  $D(x)$  of the Difference-of-Gaussian scale-space function
- Discarding low contrast (low gradient) keypoints → using the value of the second-order Taylor expansion  $D(x)$
- Discard edge points → eigenvalues

Keypoints orientation

Each keypoint comes with its scale

The gaussian smoothed image closest to that scale is selected, called L

Descriptor calculation

In the image L corresponding to the keypoint scale

Considering coordinates that are rotated based on the measure keypoint orientation

## SIFT PCA

maximizes variance

minimizes mean squared distance

from 3042 to 36

## SURF

speed up computations by fast approximation of Hessian matrix → Integral image → Box filters

Orientation → Evaluate gradients in x and y using the Haar wavelets (again type of box filter) → quantize bins of 60° Descriptor → considero 16 regioni, ma 4 elementi per region

## SIFT

Descriptor → considero 16 regioni, ma 8 histogram value per region → values weighted gaussian function

## FEATURE MATCHING

- Matching strategy → what features to compare?
  - Brute force (BF) matcher (all)
  - threshold on maximum distance (more than one)
  - Nearest Neighbor (NN) (in general one)
  - Nearest Neighbor Distance Ratio (NNDR)
- Match evaluation → assign a value to the match → Euclidean distance or Hamming distance
- Match performance → Precision and Recall → use of: True positives, True negatives, False positives, False negatives

## DESCRIPTOR ALGORITHMS, OTHER APPROACHES:

- GLOH → log polar location grid
- Shape context
- Locally Binary Pattern (LBP)
- Brief
  - not rotation invariance
  - consider random point in the neighborhood → compared using the Hamming distance (XOR)

## DETECTOR ALGORITHMS

FAST → no orientation component and no multiscale features

ORB (Oriented FAST and Rotated BRIEF)

- rotated BRIEF introduces rotation invariance → orientation assignment based on the centroid
- intensity wrt the central pixel (in general they don't coincide)

## 8 IMAGE RESAMPLING

Image resampling

- change the image resolution
- Decimation: smaller image
- Interpolation: larger image
  - Nearest neighbor interpolation → closest pixel
  - Bilinear interpolation → closest 4 samples
  - Bicubic interpolation → 16 closest samples

## 9 OBJECT DETECTION

Viola Jones algorithm → Haar cascade classifier (in particular detection on faces)