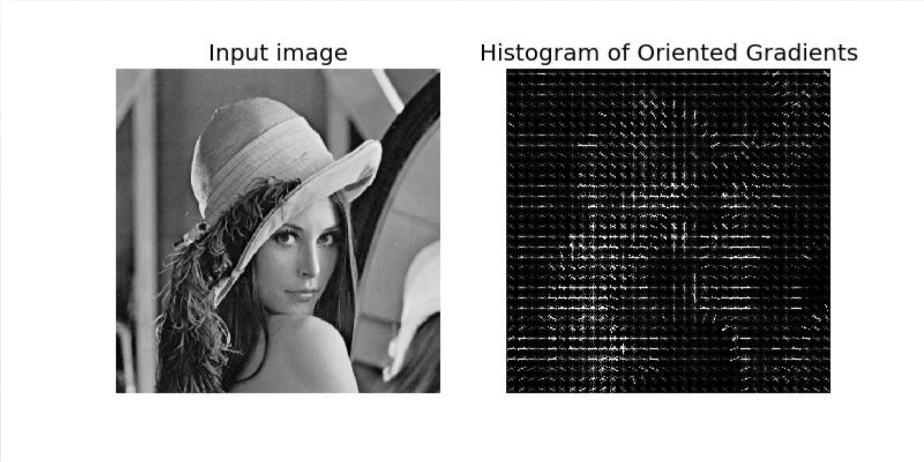
Computer Vision Course

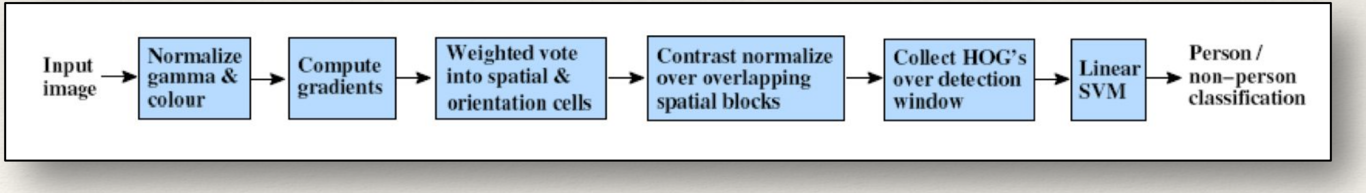
Lab 4: Features detection & Classification

Nicola Garau
nicola.garau@unitn.it

Histogram of Gradients (HOG)



- We want to perform binary classification



Input image → Normalize gamma & colour → Compute gradients → Weighted vote into spatial & orientation cells → Contrast normalize over overlapping spatial blocks → Collect HOG's over detection window → Linear SVM → Person / non−person classification

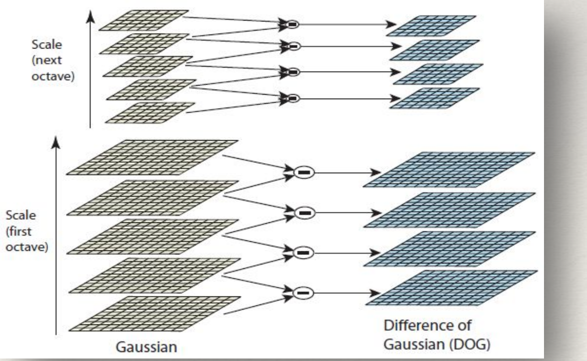
Exercise

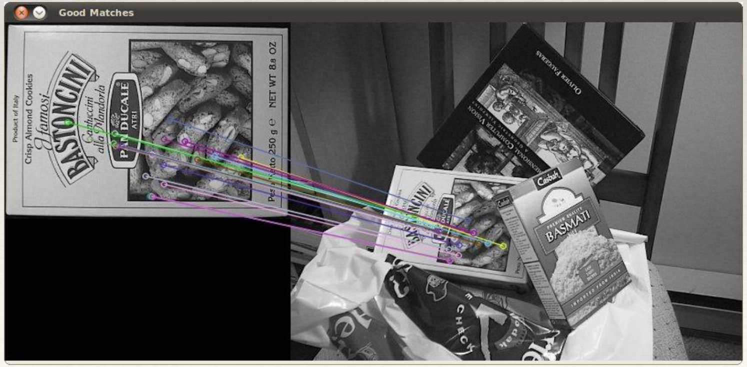
- Plot HOG features using skimage library
- Try google it (solution in the last slide)

SIFT

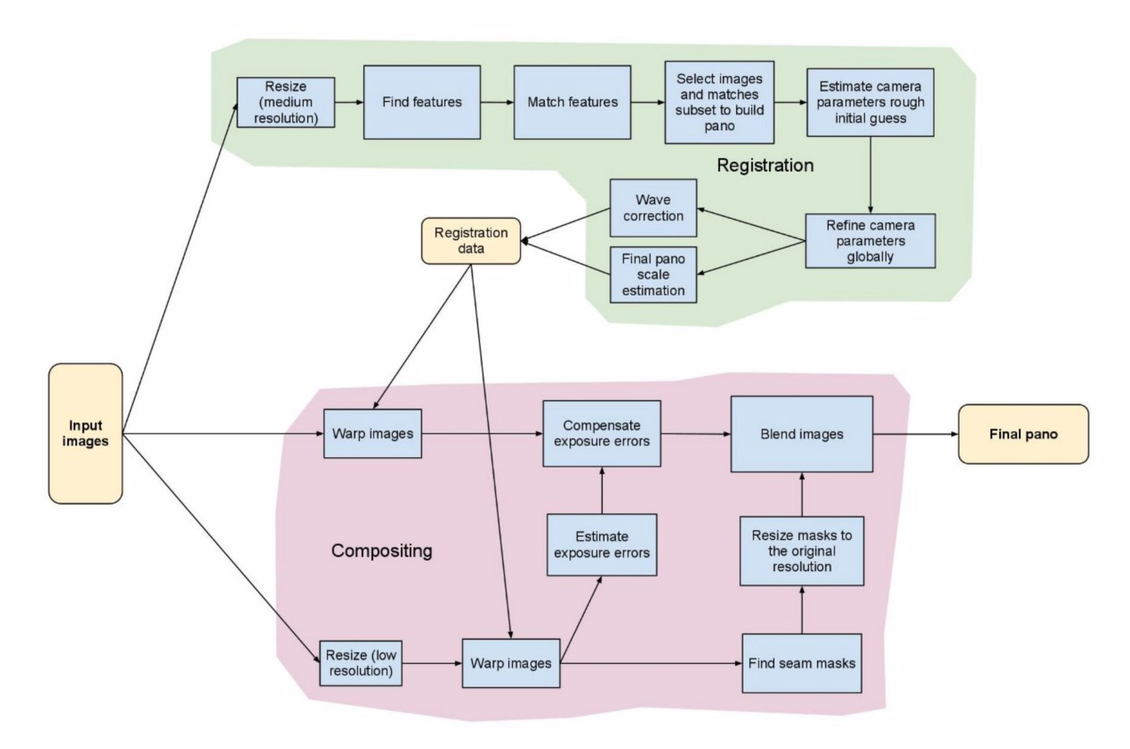
The idea is to make scale-invariant the image of concern

1. Construct a subspace representation of the image and progressively apply a Gaussian smoothing filter
2. At every iteration, each image becomes a blurred version of the previous one.

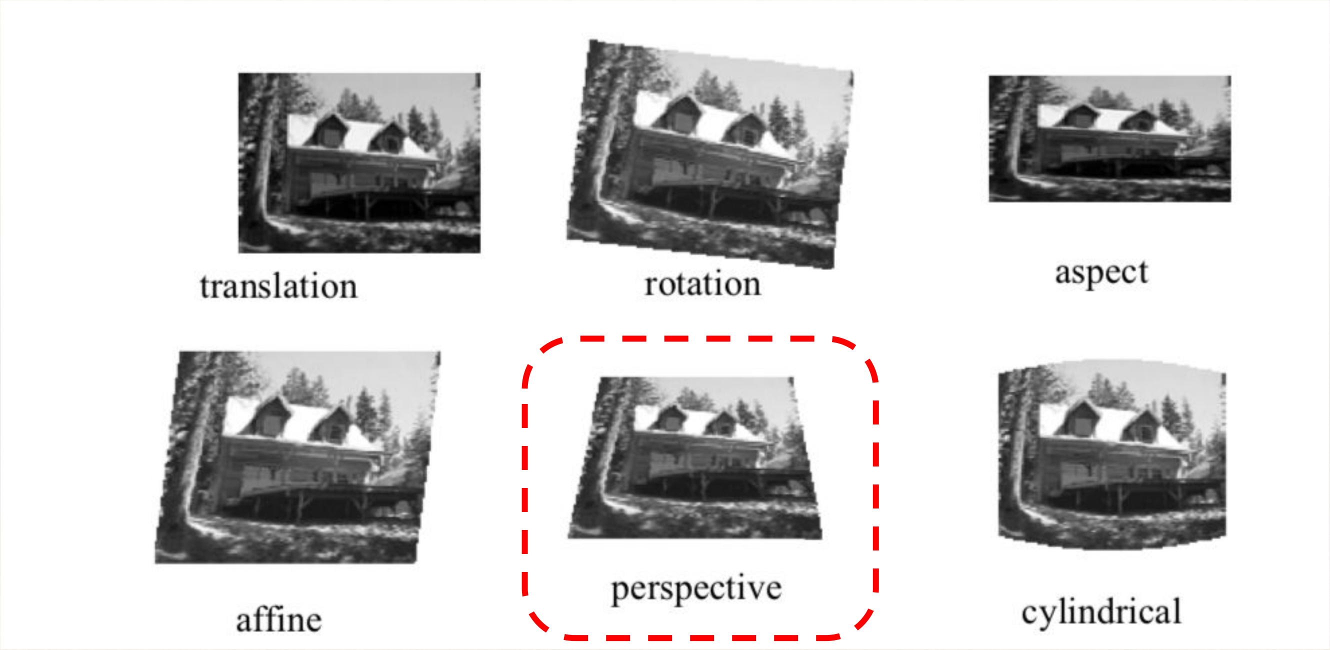




Stitching


Registration
Resize (medium resolution)
Find features
Match features
Select images and matches subset to build pano
Estimate camera parameters rough initial guess
Refine camera parameters globally
Wave correction
Final pano scale estimation
Registration data
Input images
Compositing
Warp images
Compensate exposure errors
Blend images
Final pano
Estimate exposure errors
Resize masks to the original resolution
Resize (low resolution)
Warp images
Find seam masks

Stitching



translation

rotation

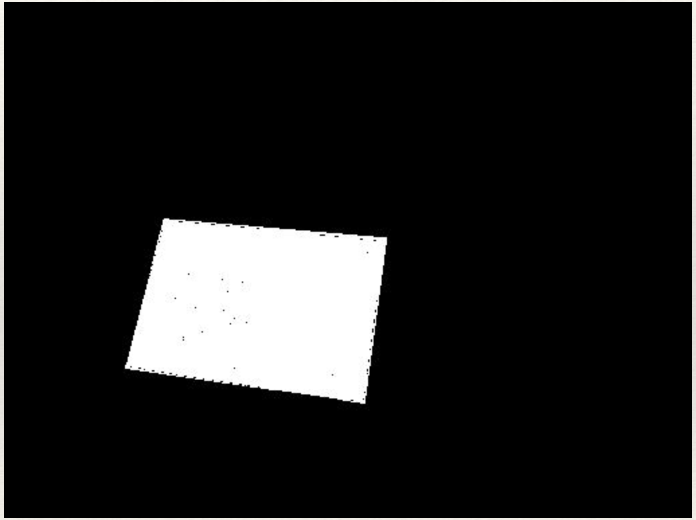
aspect

affine

perspective

cylindrical

Stitching



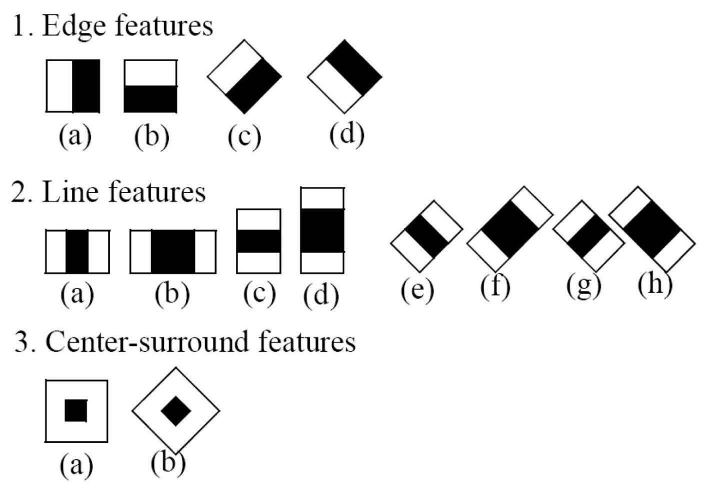
Stitching

Exercise

- Test with image ‘book.png’
- What’s the difference?

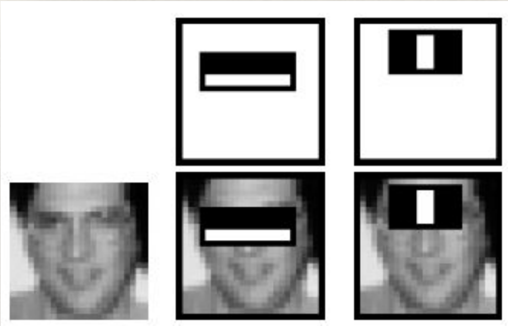
Viola-Jones Algorithm

- The Viola-Jones algorithm is one the most widespread face detectors
- Goal: Implement a robust classifier using simple features, based on binary features

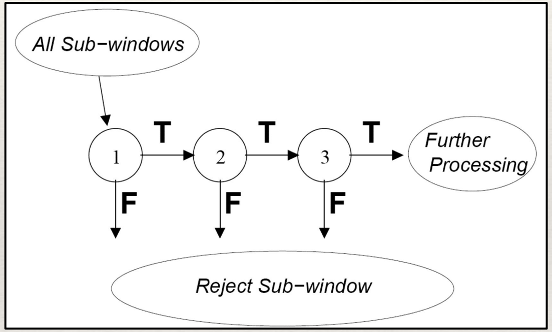


Face Detection

AdaBoost



Cascade of simple Classifiers



Exercise

- Improve the algorithm using the haarcascade_profileface.xml , in parallel to the frontal face detector
- Plot in different colors when a face is detected as frontal or as profile

Practice: face anonymization

- Try to anomymize the detected faces with different techniques
  - BLUR
  - SHUFFLING PIXELS
  - NEGATIVE
  - EMOJI (replace face)
  - Try your own technique!

HOG display- Solution

```
from skimage import exposure
from skimage import feature

# Compute HOG features for visualisation
 (H, hogImage) = feature.hog(hogImage, orientations=8,
        pixels_per_cell=(16, 16), cells_per_block=(2, 2),
        transform_sqrt=True, block_norm="L1",visualize=True)
 hogImage = exposure.rescale_intensity(hogImage,
        out_range=(50, 255))
   hogImage = hogImage.astype("uint8")
```