FREIE UNIVERSITÄT BERLIN

Computer Vision

Lecture 4: Histograms

13.11.2024

Manuel Heurich, Tim Landgraf

List of Topics

1.	Introduction (Color)	Computer Vision Fundamentals
2.	Edge Detectors	
3.	Hough Transform	Conventional Computer Vision
4.	**Histogram-based Detection and Tracking**	
5.	Optic flow	
6.	SIFT / SURF	
7.	Introduction to Neural Information Processing	Deep Learning (Supervised)
8.	Convolutional Neural Networks	
9.	Image Classification, Object Detection	
10.	Semantic Segmentation	
11.	Pose Estimation	
12.	Style Transfer	
13.	Recurrent Neural Networks, Image Captioning	
14.	Generative Models	Unsupervised Learning
15.	Unsupervised feature extraction	

(Bradski, 1998)

Cited by 2052

Computer Vision Face Tracking For Use in a Perceptual User Interface

Gary R. Bradski, Microcomputer Research Lab, Santa Clara, CA, Intel Corporation

Index words: computer vision, face tracking, mean shift algorithm, perceptual user interface, 3D graphics interface

Abstract

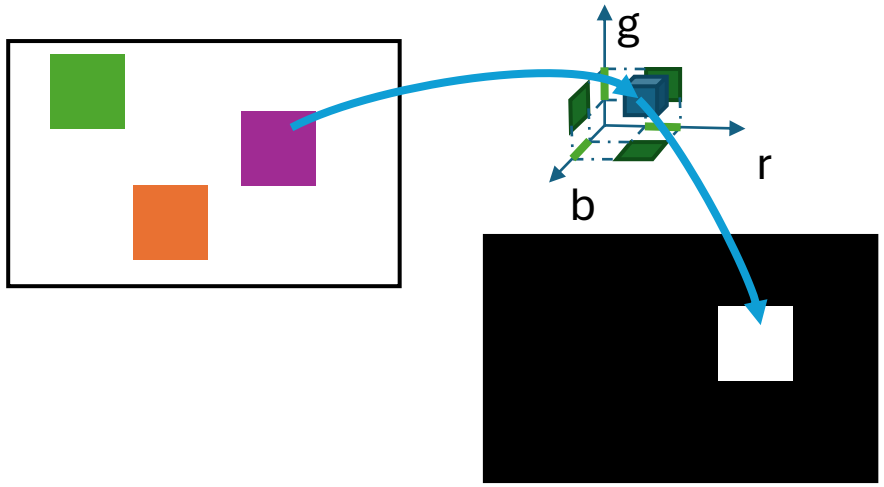
As a first step towards a perceptual user interface, a computer vision color tracking algorithm is developed and applied towards tracking human faces. Computer vision algorithms that are intended to form part of a perceptual user interface must be fast and efficient. They must be able to track in real time yet not absorb a major share of computational resources: other tasks must be able to run

this effort: a 4-degree of freedom color object tracker and its application to flesh-tone-based face tracking.

Computer vision face tracking is an active and developing field, yet the face trackers that have been developed are not sufficient for our needs. Elaborate methods such as tracking contours with snakes [[10][12][13]], using Eigenspace matching techniques [14], maintaining large sets of statistical hypotheses [15], or convolving images

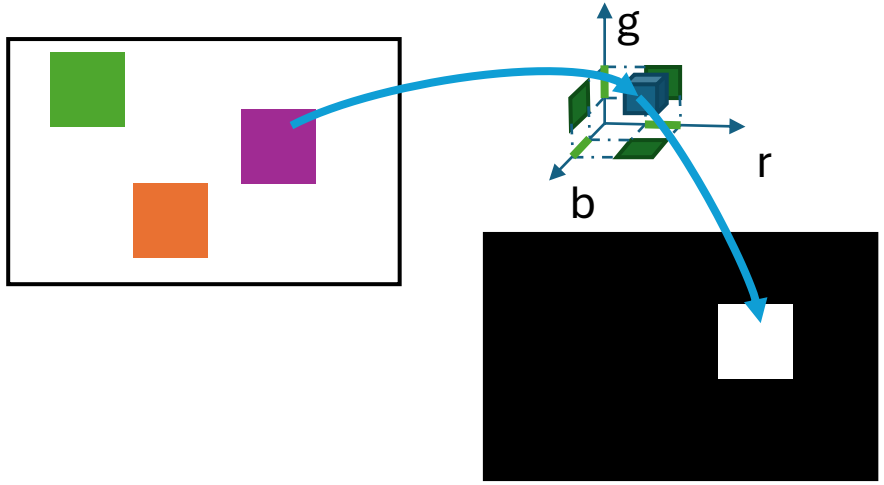
https://www.youtube.com/watch?v=Qcj4e0SGxLI

Today: Histograms as Object Features

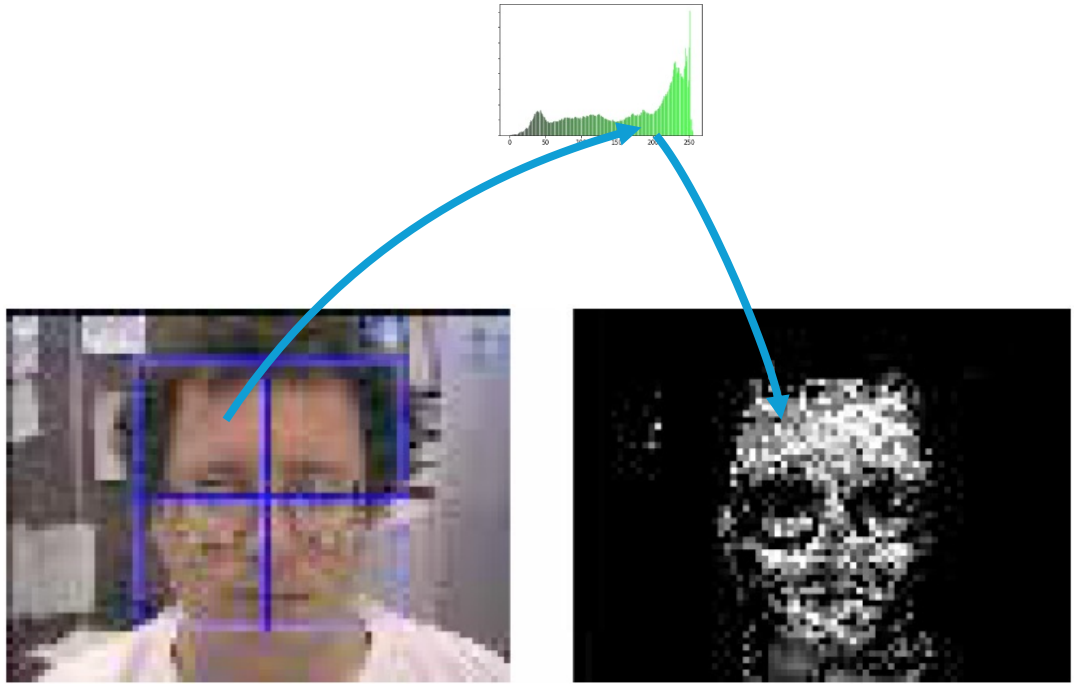


Lecture #1: color sub-space defines all colors that belong to an object

Today: Histograms as Object Features

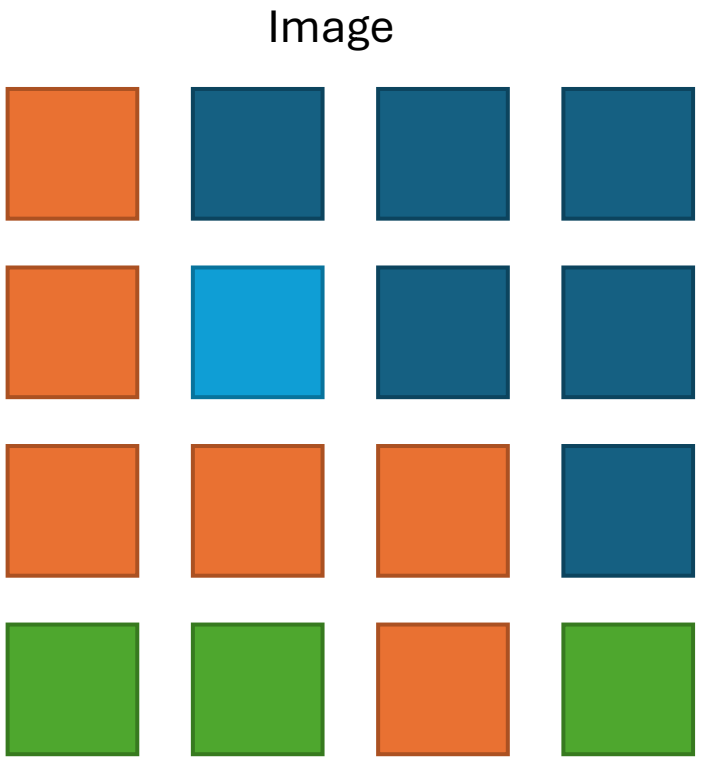


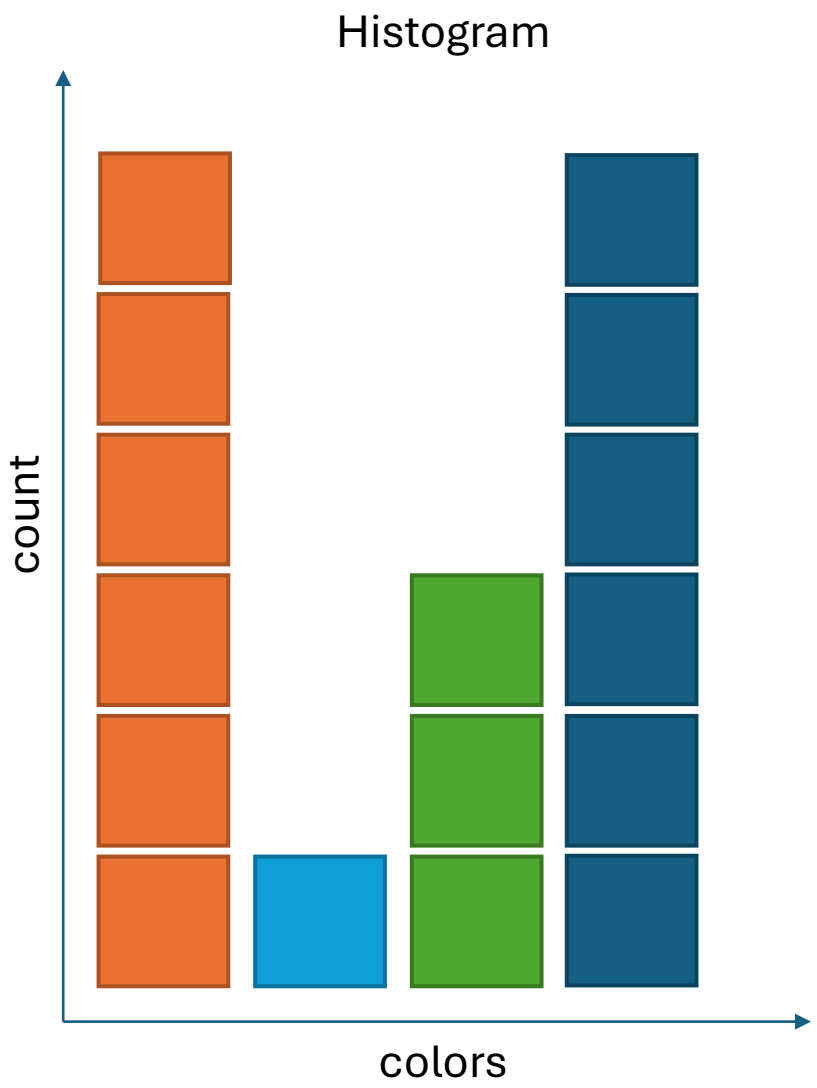
Lecture #1: color sub-space defines all colors that belong to an object

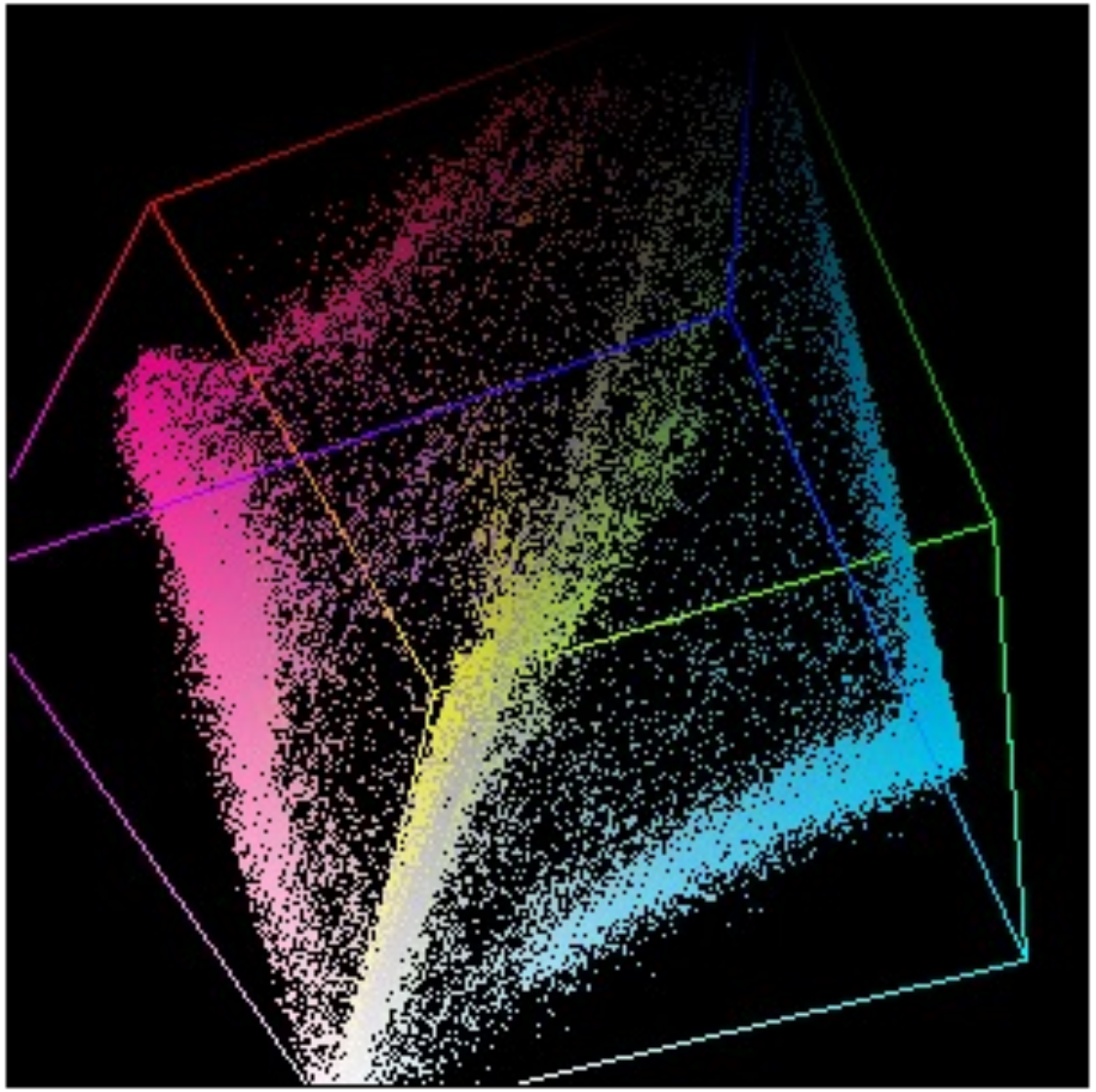


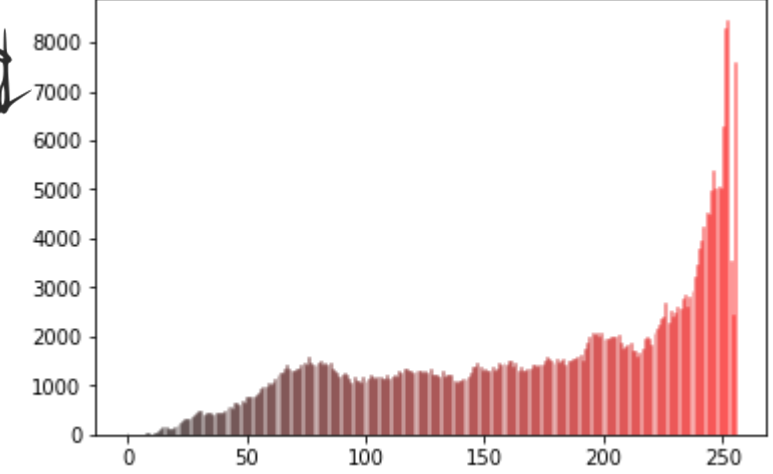
Today: color histogram defines likelihood of colors to belong to an object

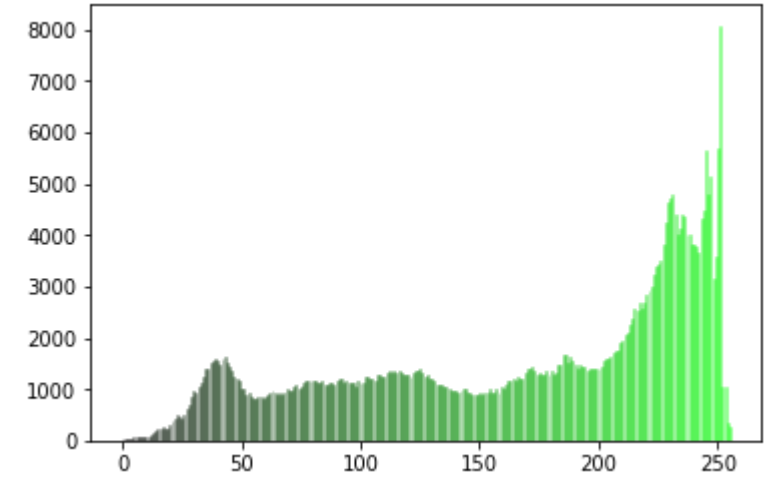
Color Histograms

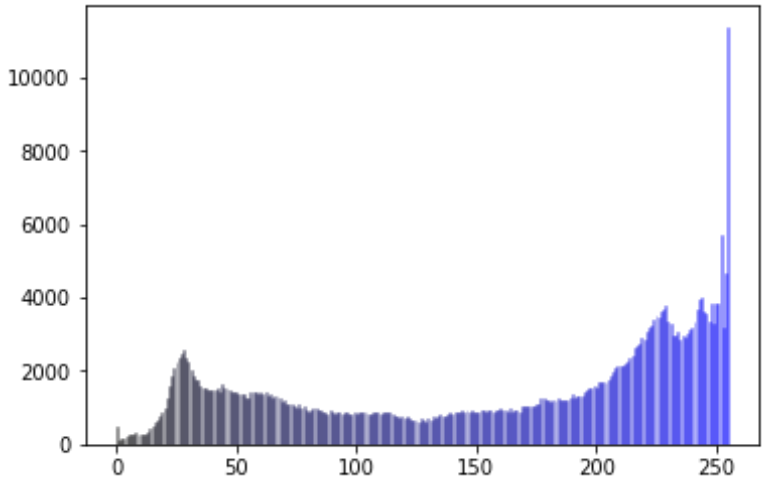












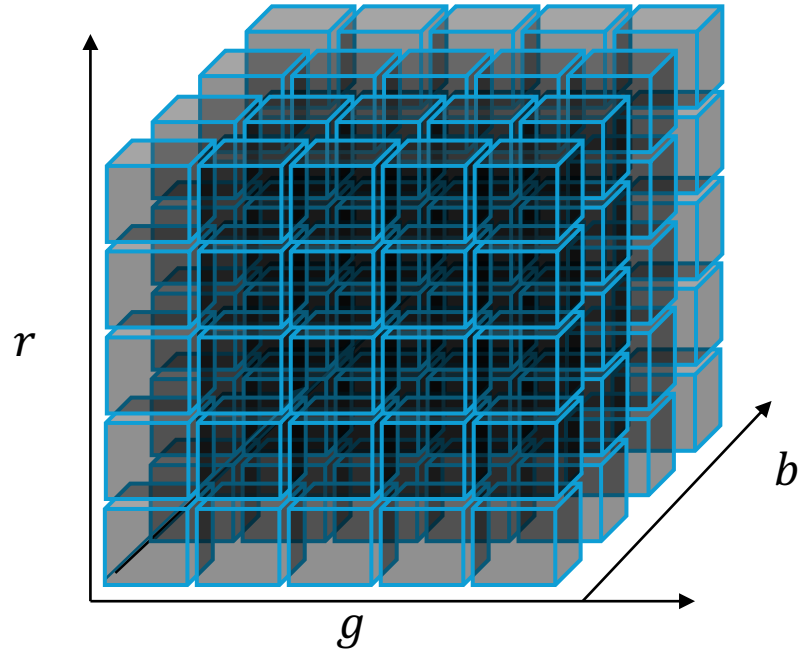
3D-Histogram ≠ 3 Histograms!

3D-Histogram, actual

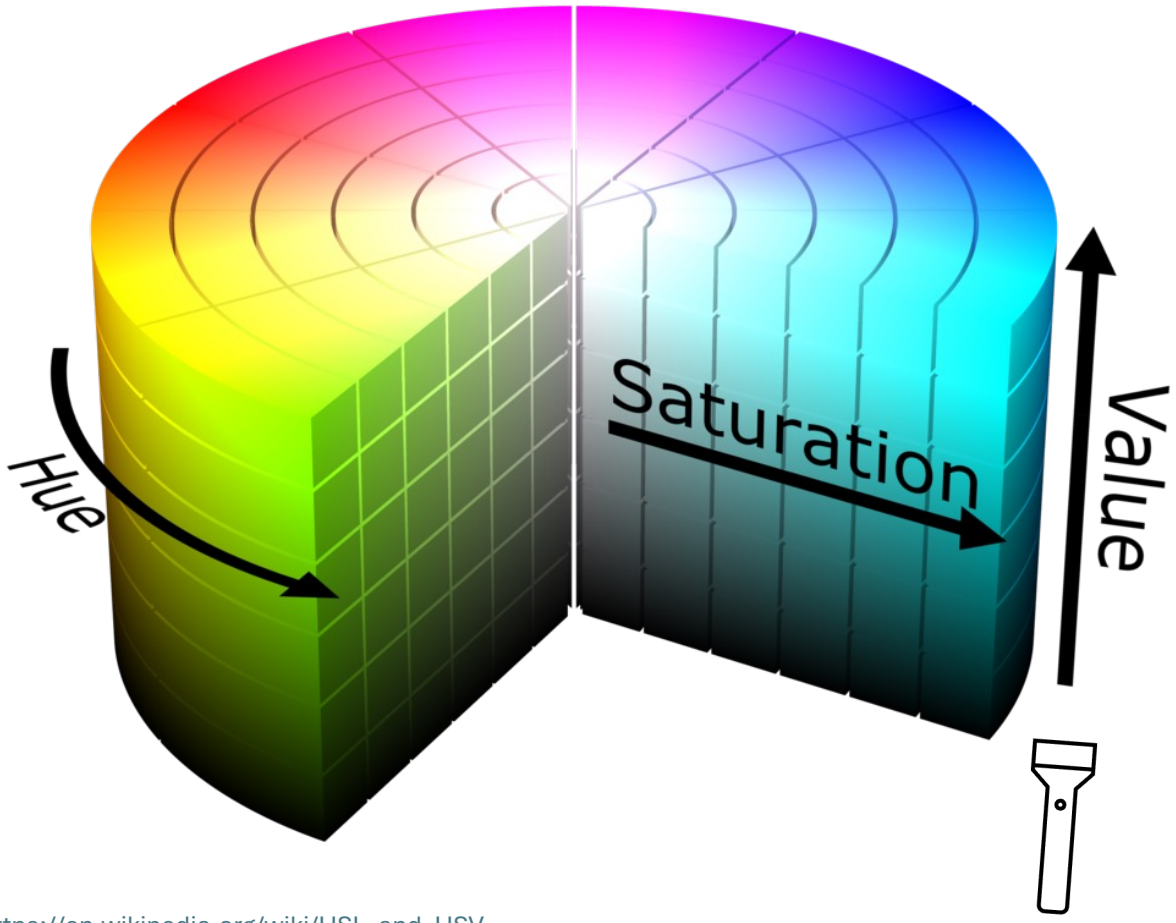
Index into histogram with rgb-triplet

Every cell contains count of that specific triplet

Q: What is the problem with a 3D histogram?



HSV, a better color space?

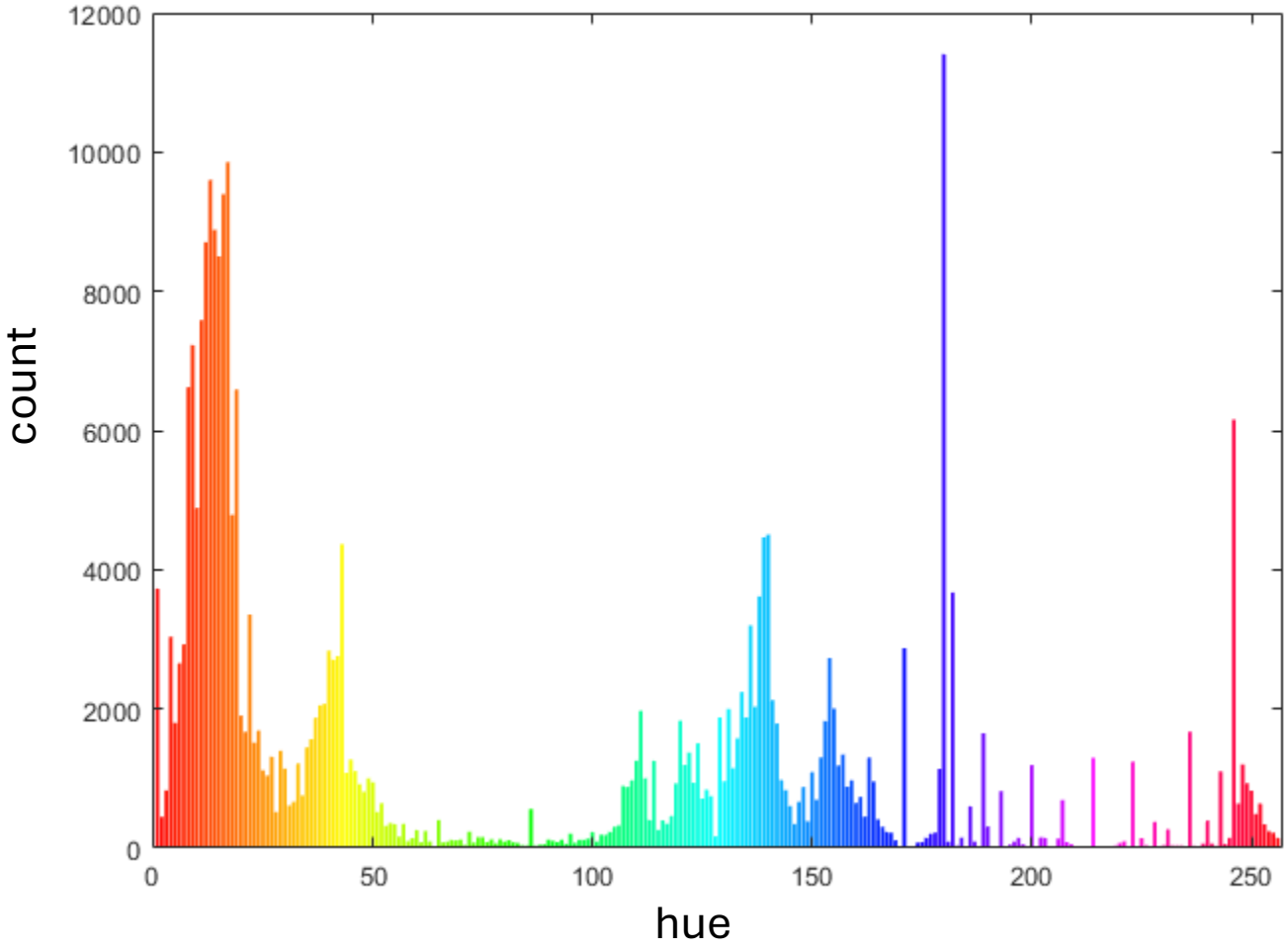


https://en.wikipedia.org/wiki/HSL_and_HSV

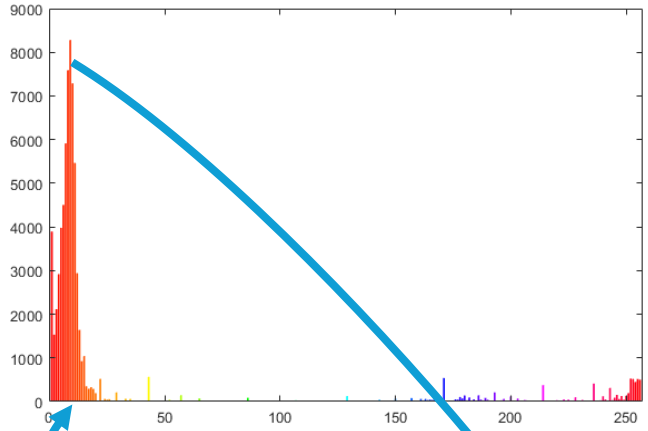
HSV representation models how colors appear under light

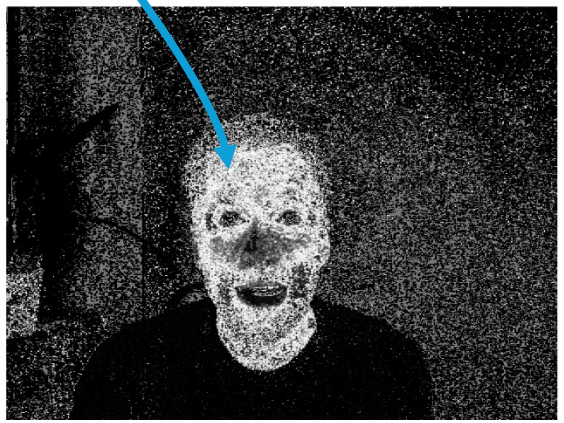
- ***Hue***: angular dimension, represents color tone from red (0°), green (120°) to blue (240°)
- ***Saturation***: $S_{\max}$ is purest color, mix with white → decrease S
- ***Value***: shine light on color, $V_{\max}$yields most intense color

Hue histogram



Histogram as probability map

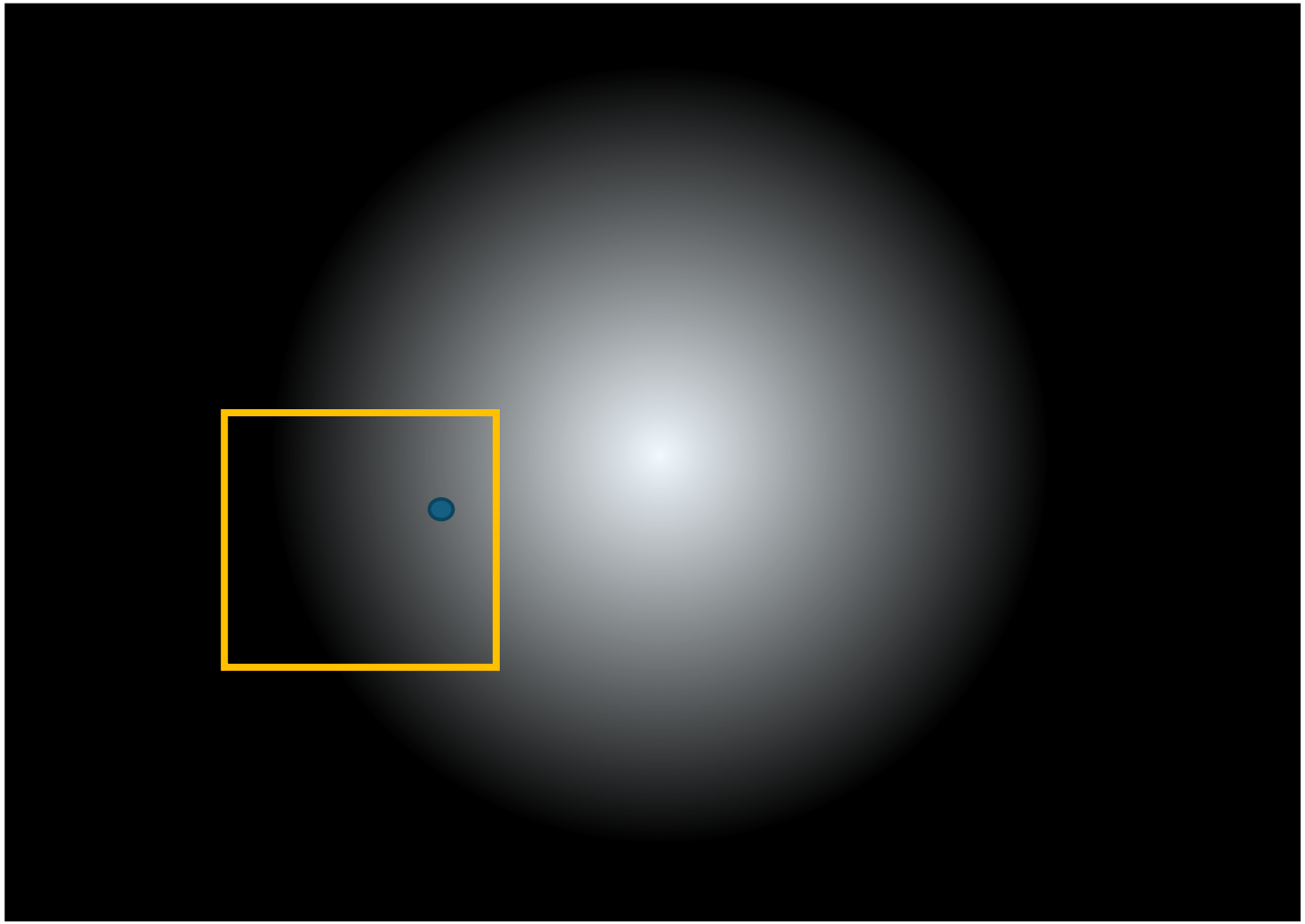




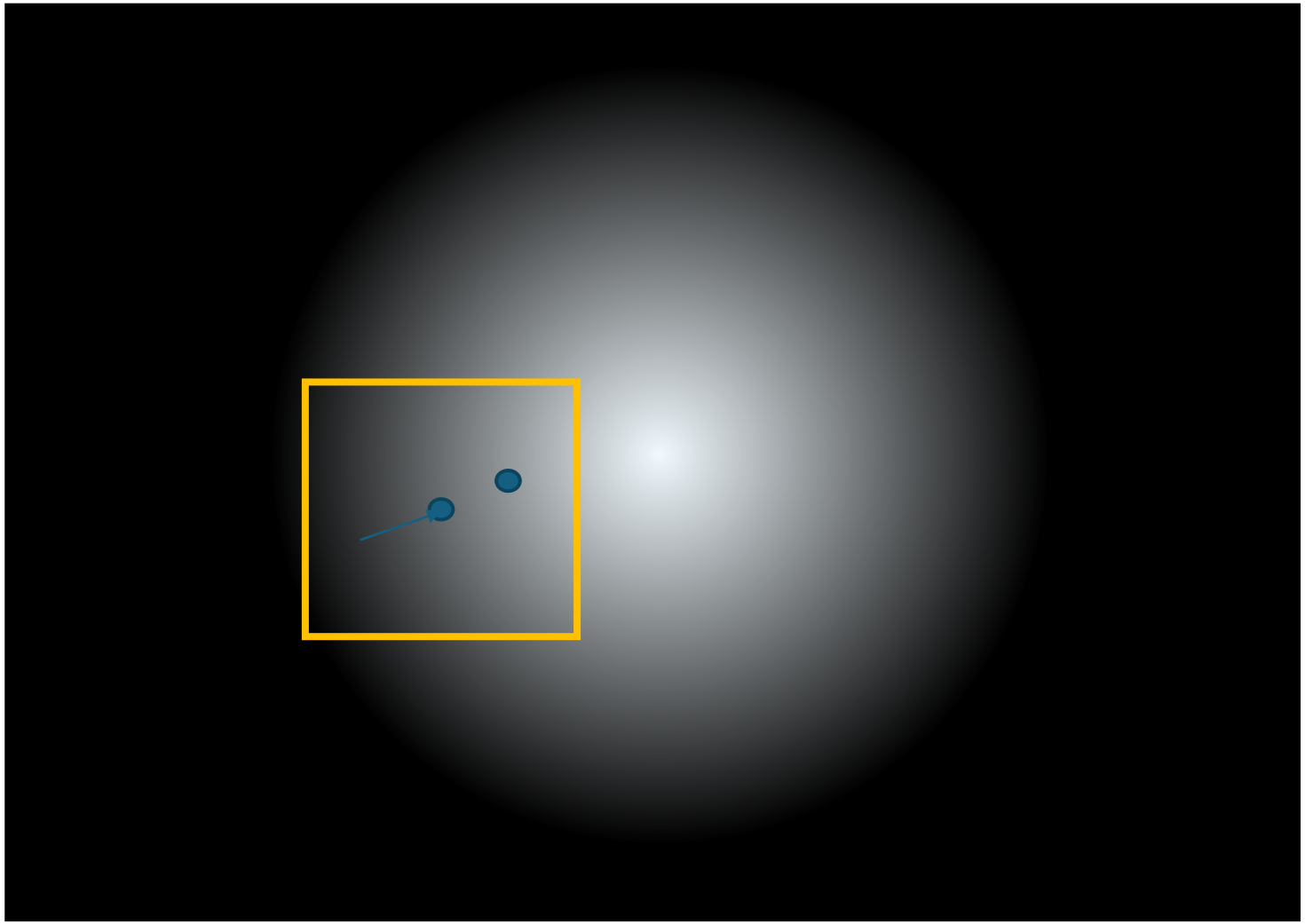
The distribution of color hues

- reflects how likely a specific hue is within the object boundaries
- is an **object descriptor**
- given a pixel p with hue h, how likely belongs p to the object?

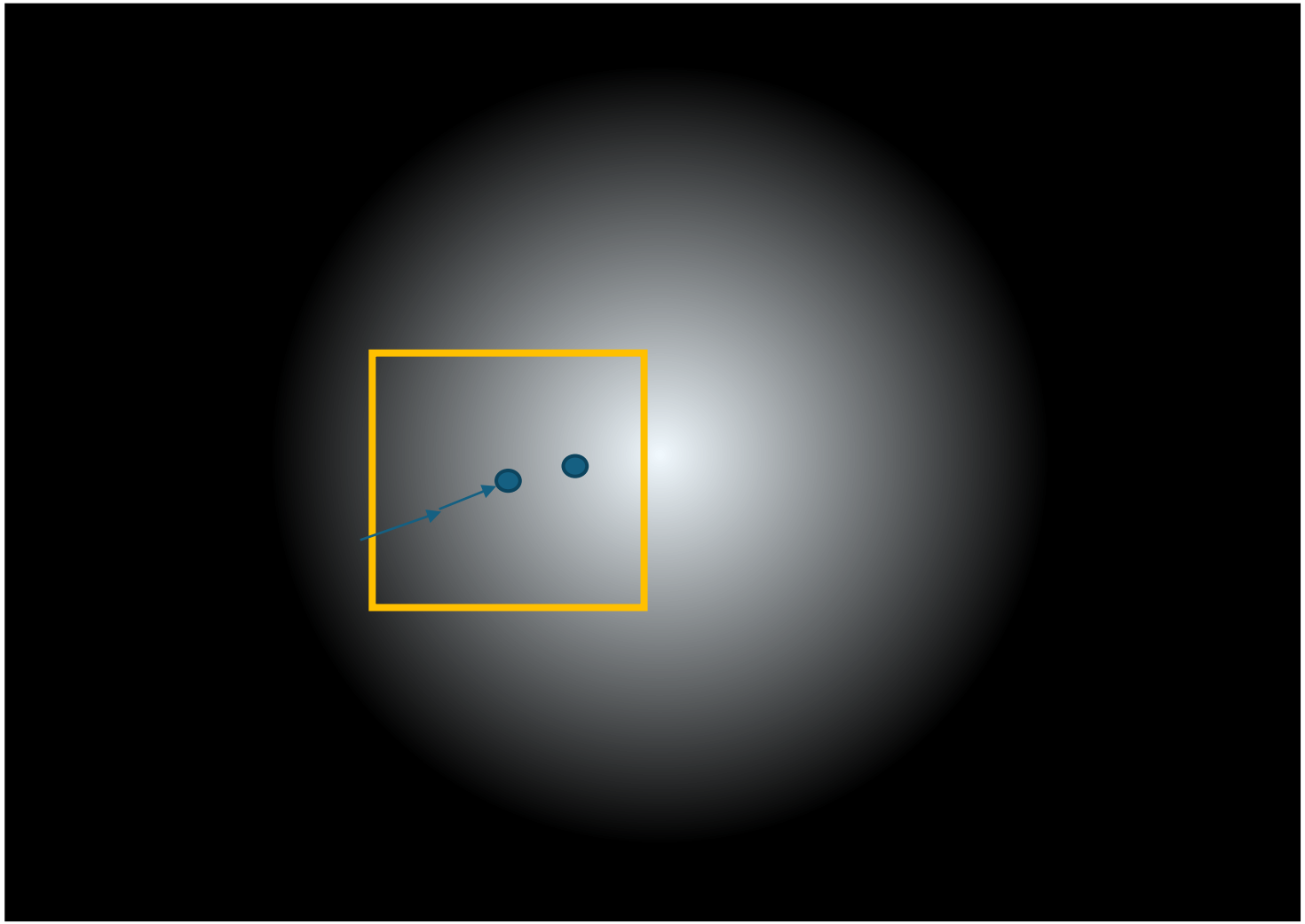
Track the position of the distribution: Mean Shift



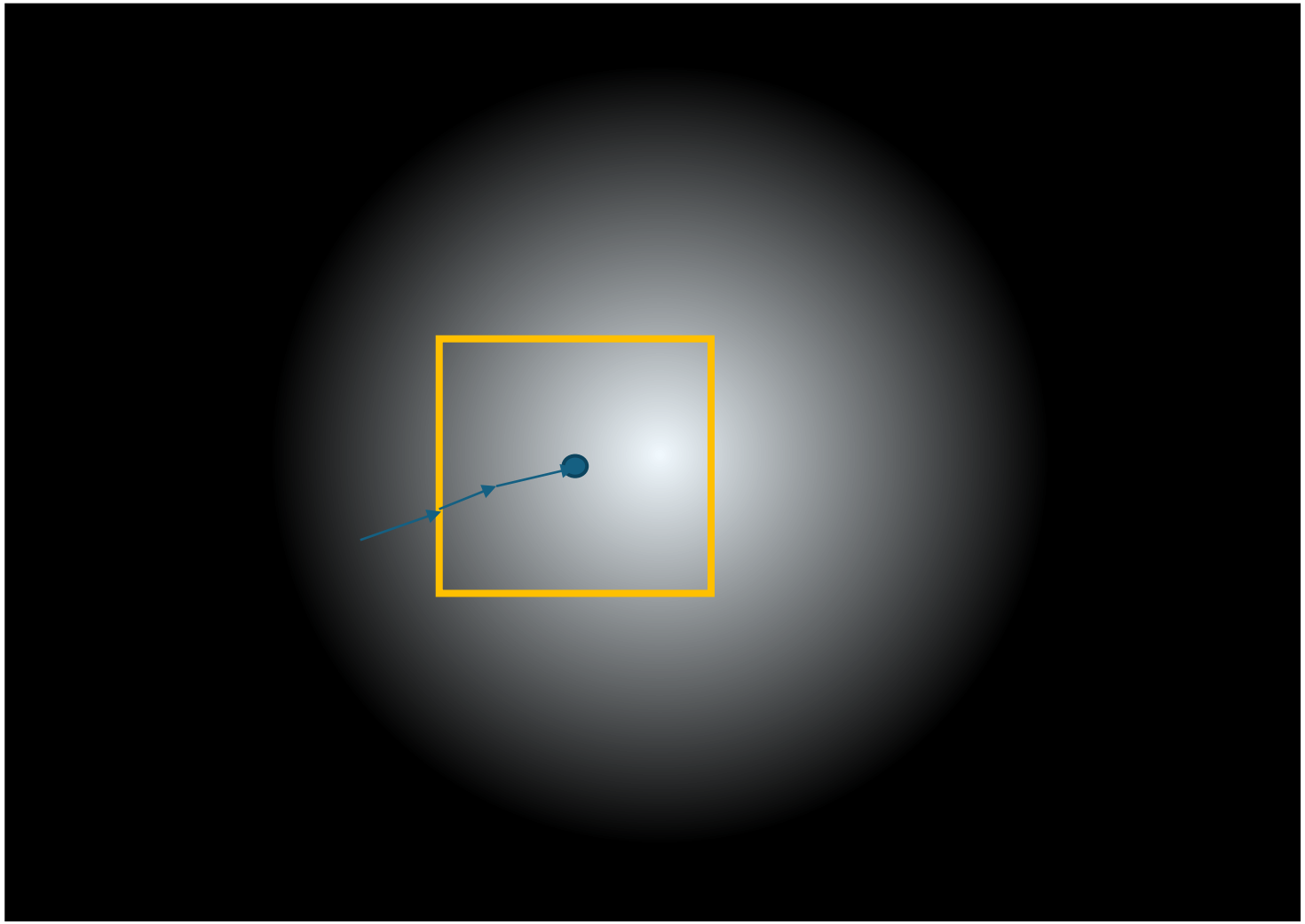
Track the position of the distribution: Mean Shift



Track the position of the distribution: Mean Shift



Track the position of the distribution: Mean Shift

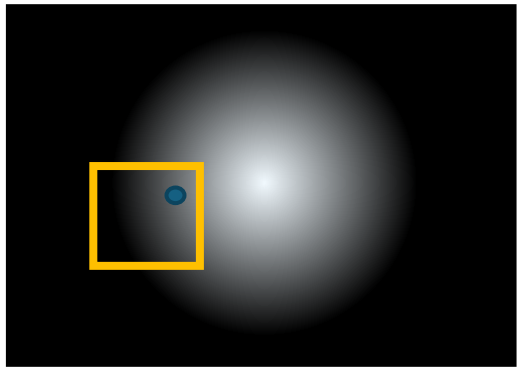


Mean Shift: using image moments

$$M_{ij} = \sum_x \sum_y x^i y^j I(x,y)$$

$$M_{00} = \sum_x \sum_y I(x,y)$$ zeroth moment

$$M_{10} = \sum_x \sum_y x I(x,y) \quad M_{01} = \sum_x \sum_y y I(x,y)$$ first moments



$$x_c = \frac{M_{10}}{M_{00}} \qquad y_c = \frac{M_{01}}{M_{00}}$$

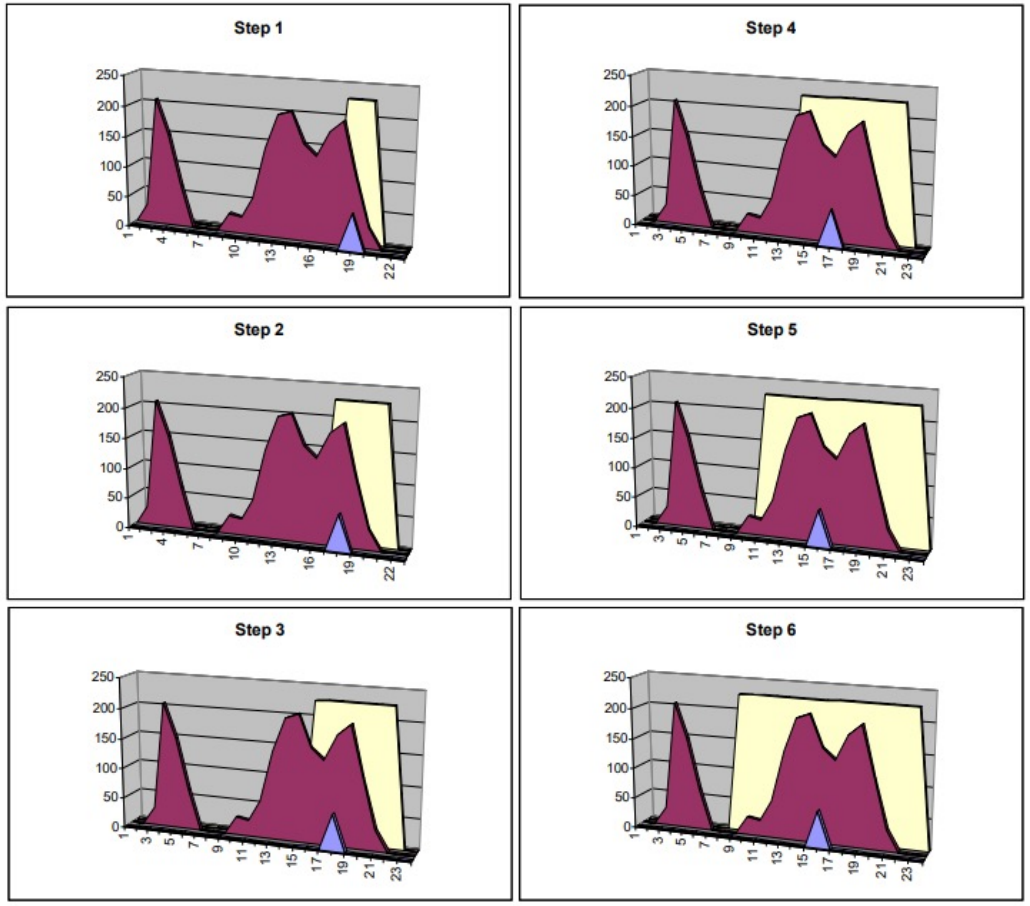
center of mass

Ex. 3.1: track colored objects with mean shift

- read the CAMSHIFT paper
- use an initial search window to extract an object's hue histogram
- implement the probability image given the histogram descriptor
- track the mode of the color distribution with mean shift

What if we lose track of the object?

CAMSHIFT: adaptive window size

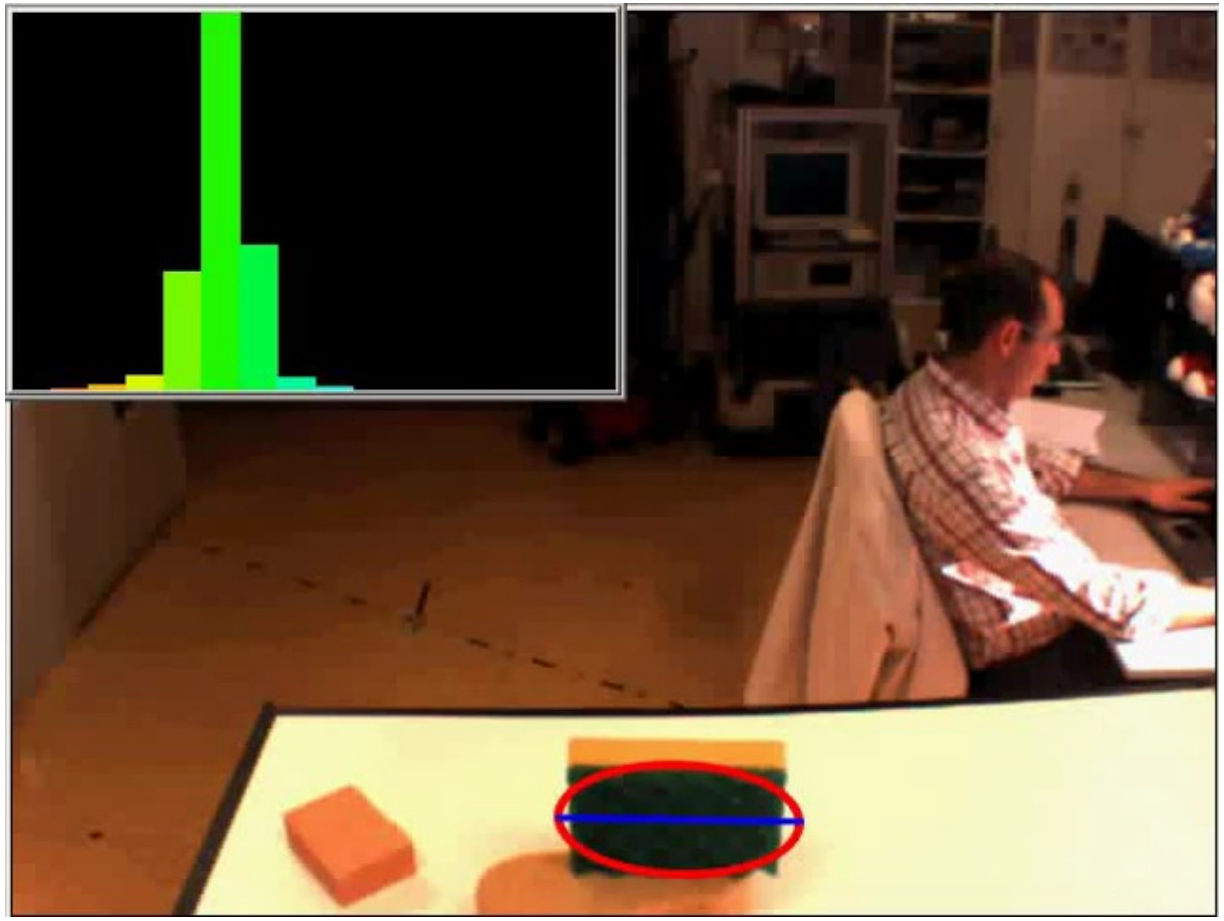


Update search window size depending on probability mass in window:

$$s = 2\sqrt{\frac{M_{00}}{256}}$$

Q: What happens with non-zero noise levels?

CAMSHIFT: Track the orientation of the object

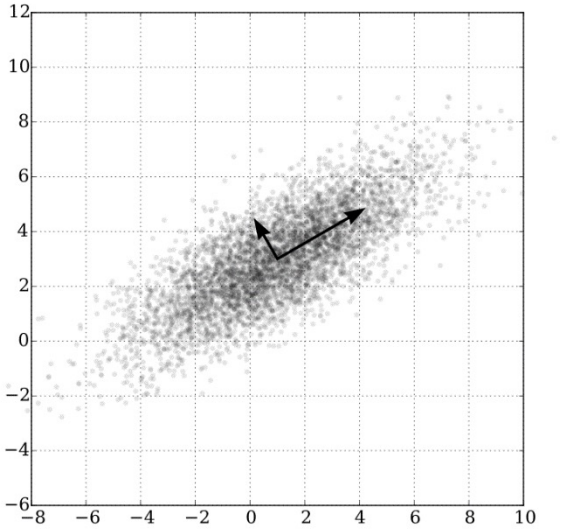


https://www.youtube.com/watch?v=iBOlbs8i7Og

CAMSHIFT: object orientation

$$\theta = \frac{\operatorname{atan}\left(\frac{2\left(\frac{M_{11}}{M_{00}} - x_c y_c\right)}{\left(\frac{M_{20}}{M_{00}} - x_c^2\right) - \left(\frac{M_{02}}{M_{00}} - y_c^2\right)}\right)}{2}$$

$$M_{ij} = \sum_x \sum_y x^i y^j I(x, y)$$



https://en.wikipedia.org/wiki/Principal_component_analysis

θ is the angle of the first principal component of the probability distribution!

Read more here:
https://en.wikipedia.org/wiki/Image_moment

Q: Why can't we just use the eigenvector of the covariance matrix?

Ex. 3.2: track colored objects with CAMSHIFT

- implement the automatic determination of the search window size
- implement the extraction of object orientation and ellipse axes
- draw a respective ellipse on the image(s)

FREIE UNIVERSITÄT BERLIN

Summary: Color-based object detection

- color *can be* a stable feature, when using the right color space (HSV rather than RGB)
- color does not encode *structural* image features
- color histograms can serve as **object descriptor** and **predictor** of a pixels object membership (see probability image in CAMSHIFT)
- even in HSV color may vary, depending on various factors

FREIE UNIVERSITÄT BERLIN

HOG: Histograms of Oriented Gradients

(Dalal & Triggs, 2005)

Cited by 22193

Histograms of Oriented Gradients for Human Detection

Navneet Dalal and Bill Triggs
INRIA Rhône-Alps, 655 avenue de l'Europe, Montbonnot 38334, France
{Navneet.Dalal,Bill.Triggs}@inrialpes.fr, http://lear.inrialpes.fr

Abstract

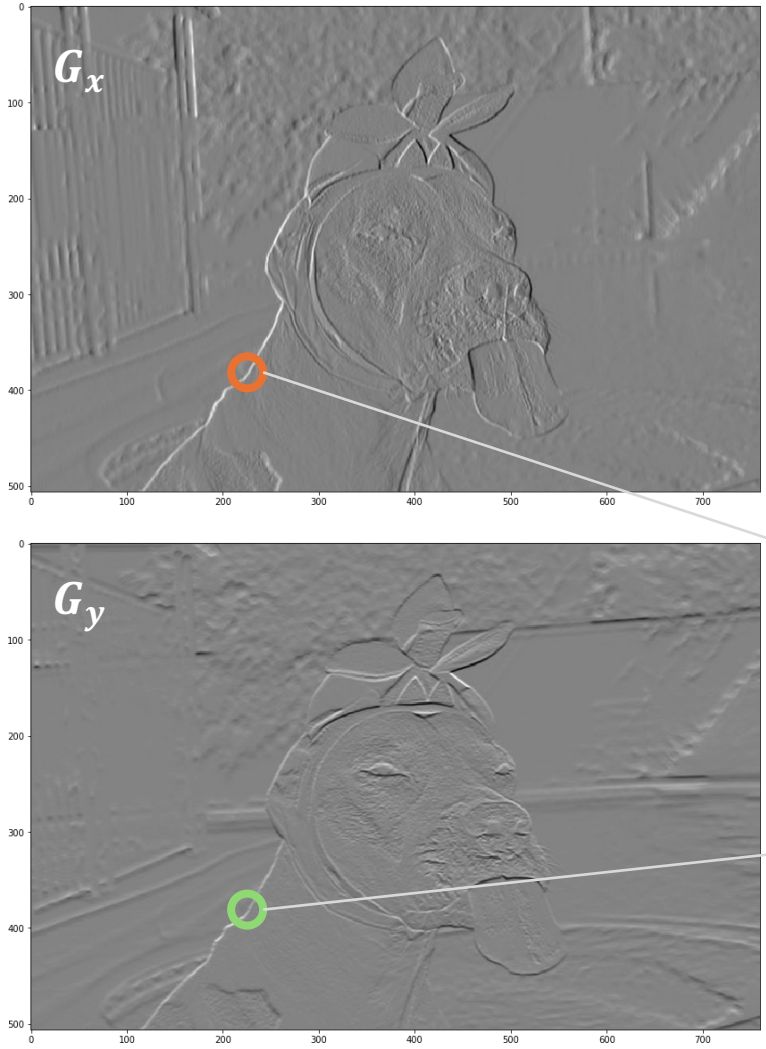
We study the question of feature sets for robust visual object recognition, adopting linear SVM based human detection as a test case. After reviewing existing edge and gradient based descriptors, we show experimentally that grids of Histograms of Oriented Gradient (HOG) descriptors significantly outperform existing feature sets for human detection. We study the influence of each stage of the computation on performance, concluding that fine-scale gradients, fine orientation binning, relatively coarse spatial binning, and high-quality local contrast normalization in overlapping de-

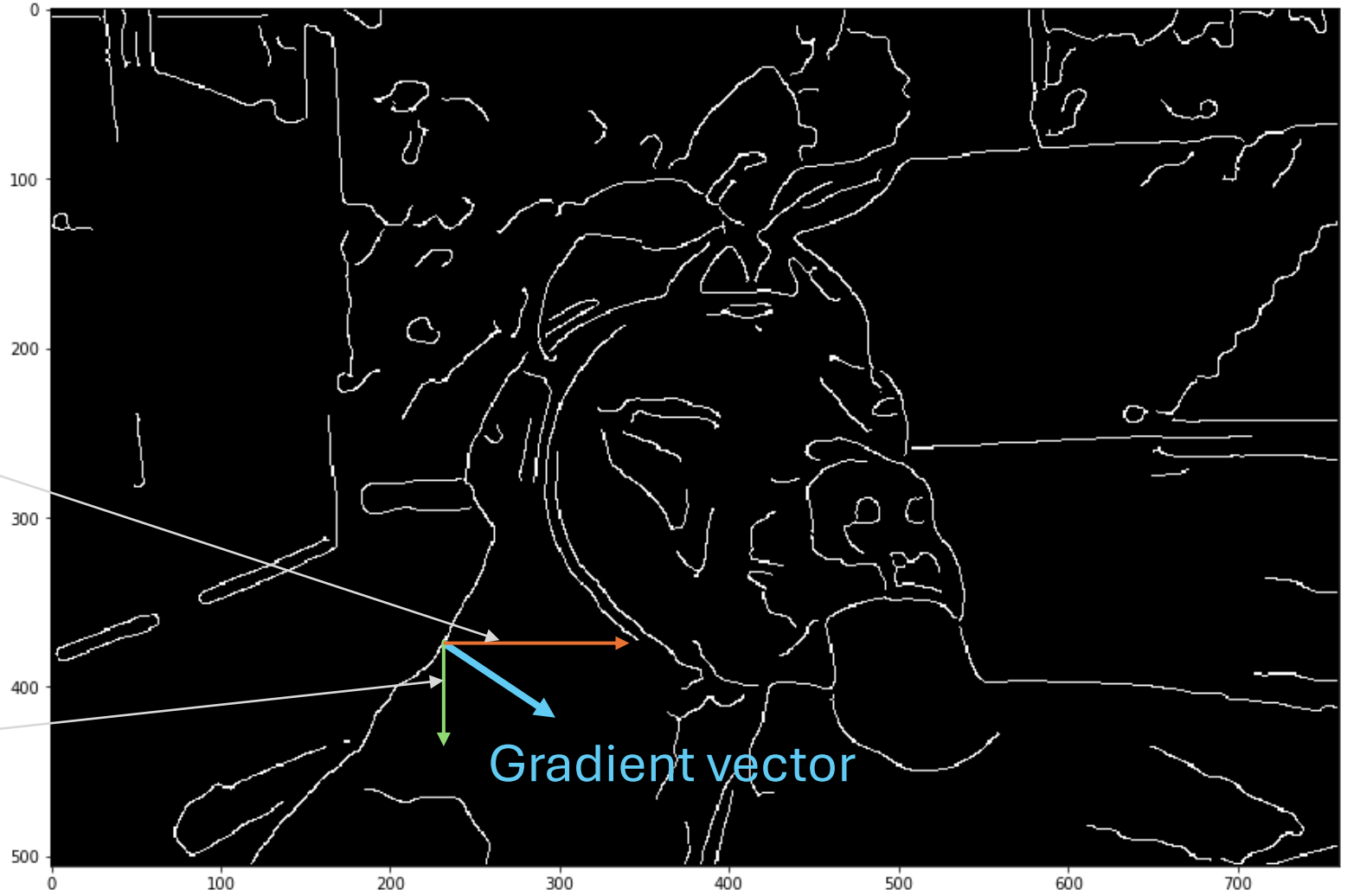
We briefly discuss previous work on human detection in §2, give an overview of our method §3, describe our data sets in §4 and give a detailed description and experimental evaluation of each stage of the process in §5–6. The main conclusions are summarized in §7.

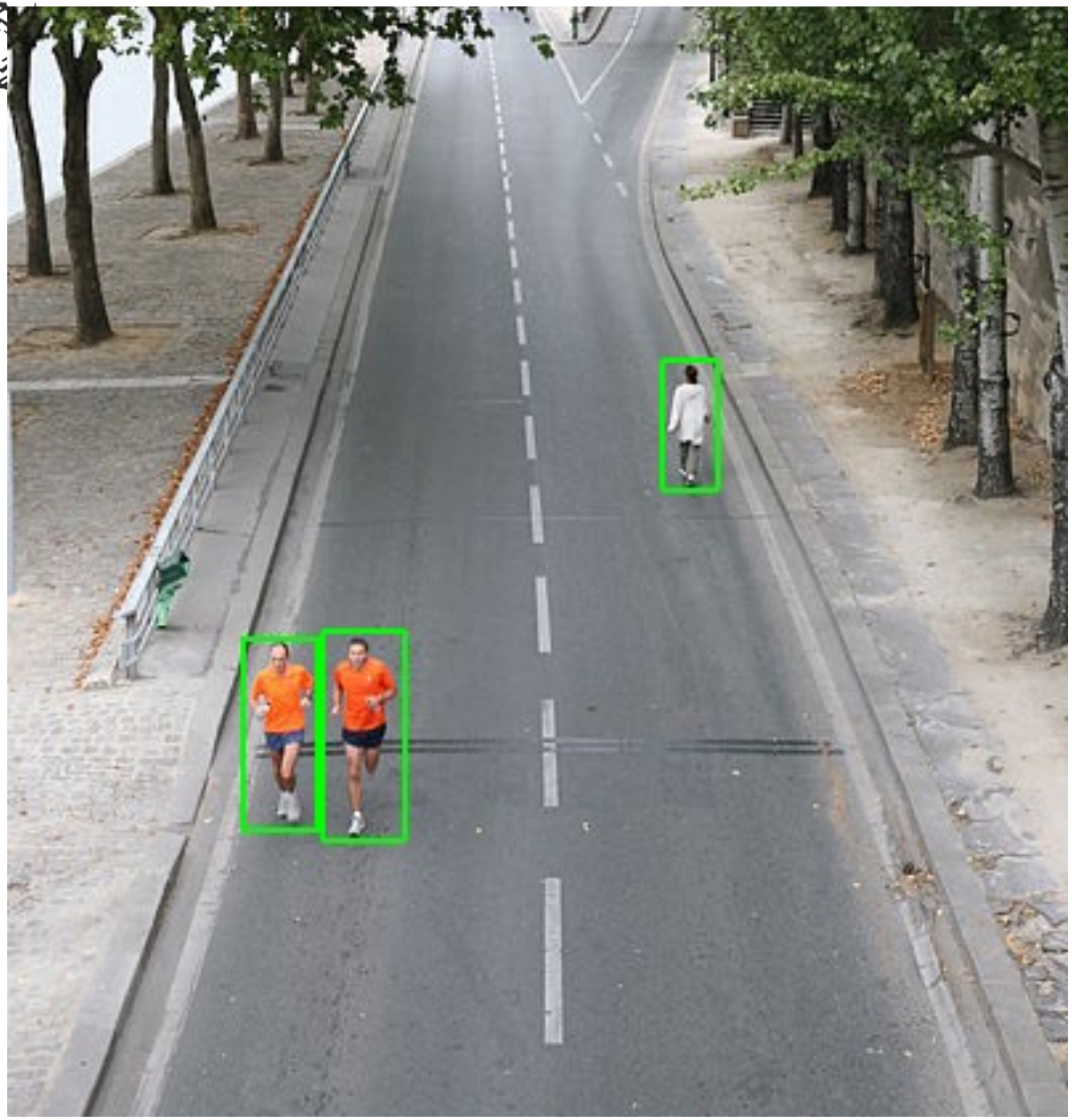
2 Previous Work

There is an extensive literature on object detection, but here we mention just a few relevant papers on human detection [18, 17, 22, 16, 20]. See [6] for a survey. Papageorgiou *et al* [18] describe a pedestrian detector based on a polynomial

Remember lecture 2: Convolution, Gradients and Edge Detection

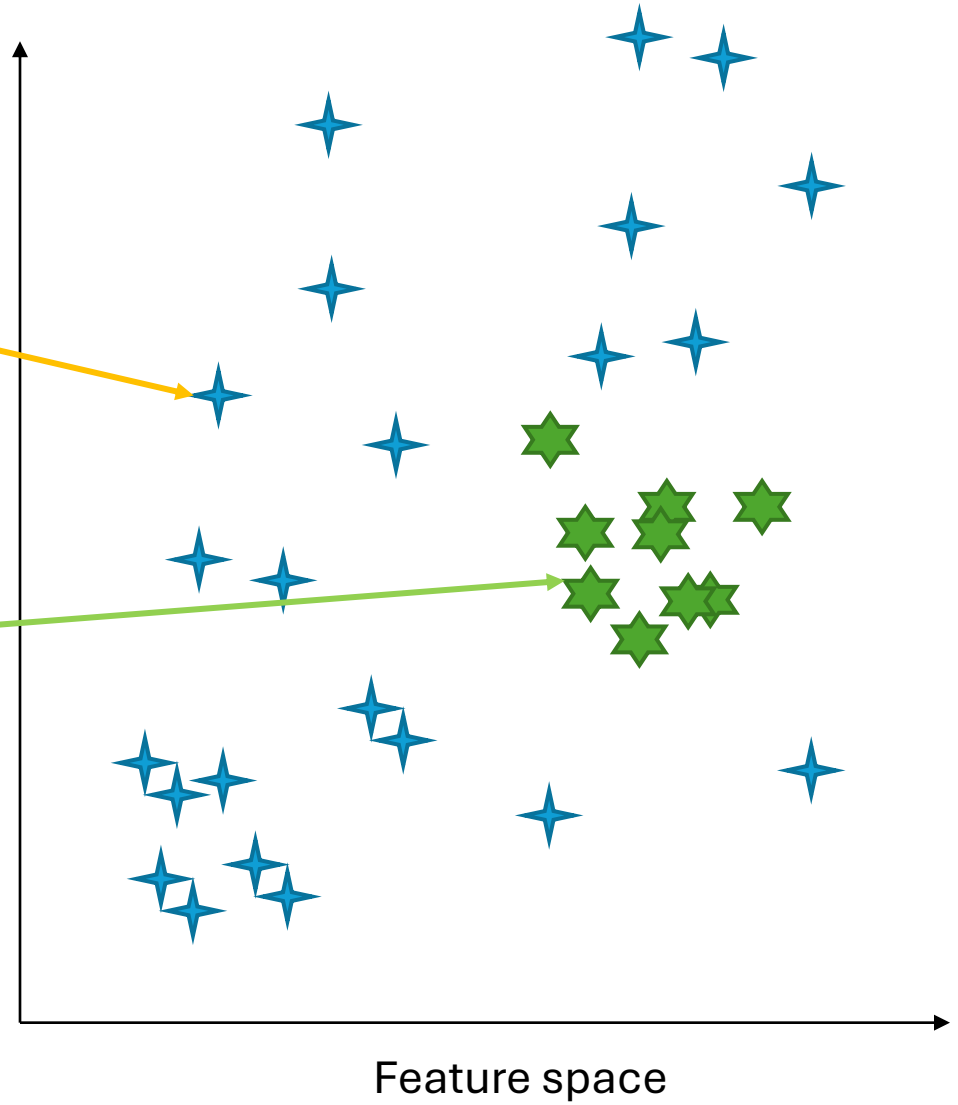




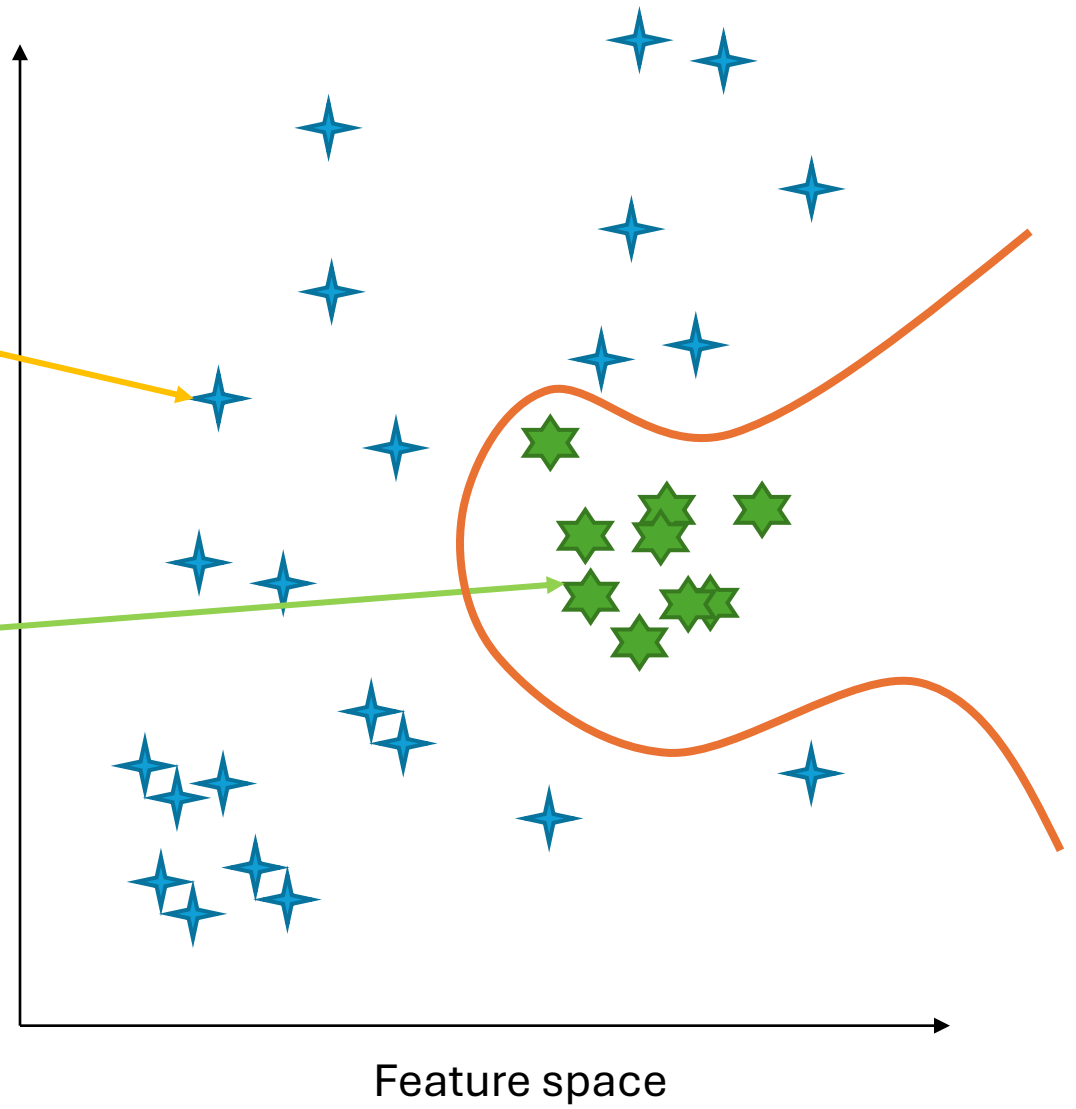


HOG example: pedestrian detection (and localization)

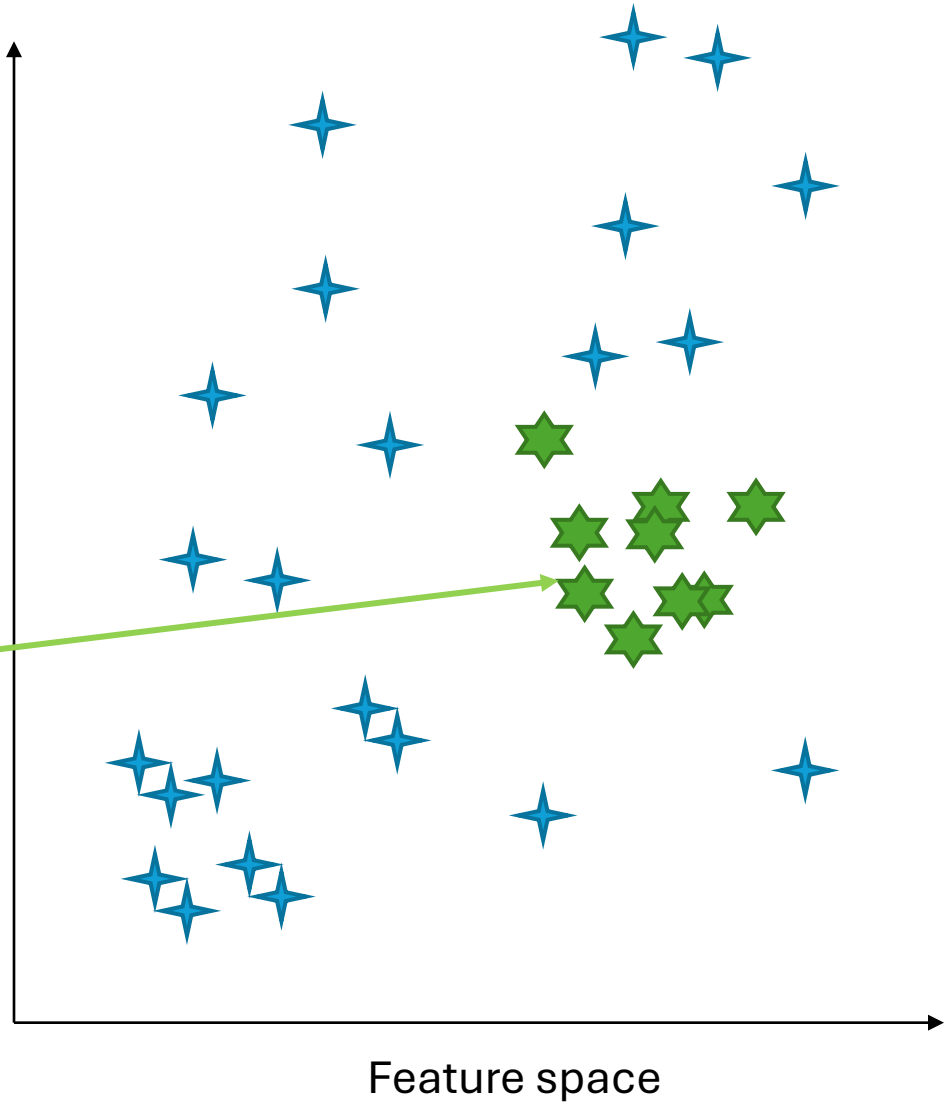
Detection (of pedestrians) = classification of ROIs



Detection (of pedestrians) = classification of ROIs



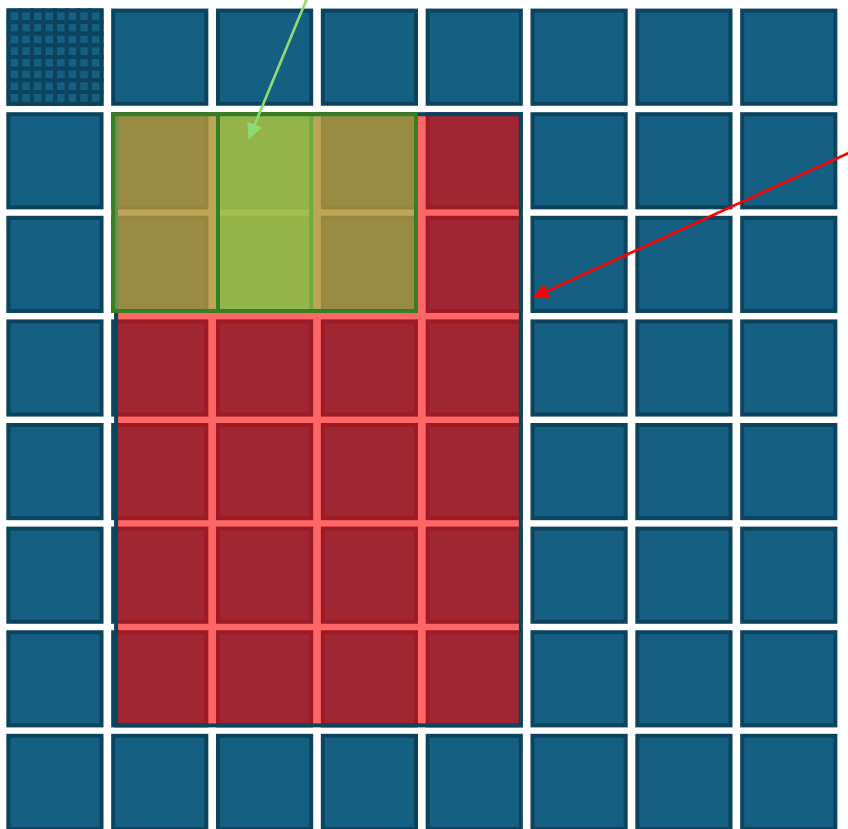
Detection (of pedestrians) =
classification of ROIs



ROI from Blocks, Blocks from Cells

Cells (8 x 8 pixels)

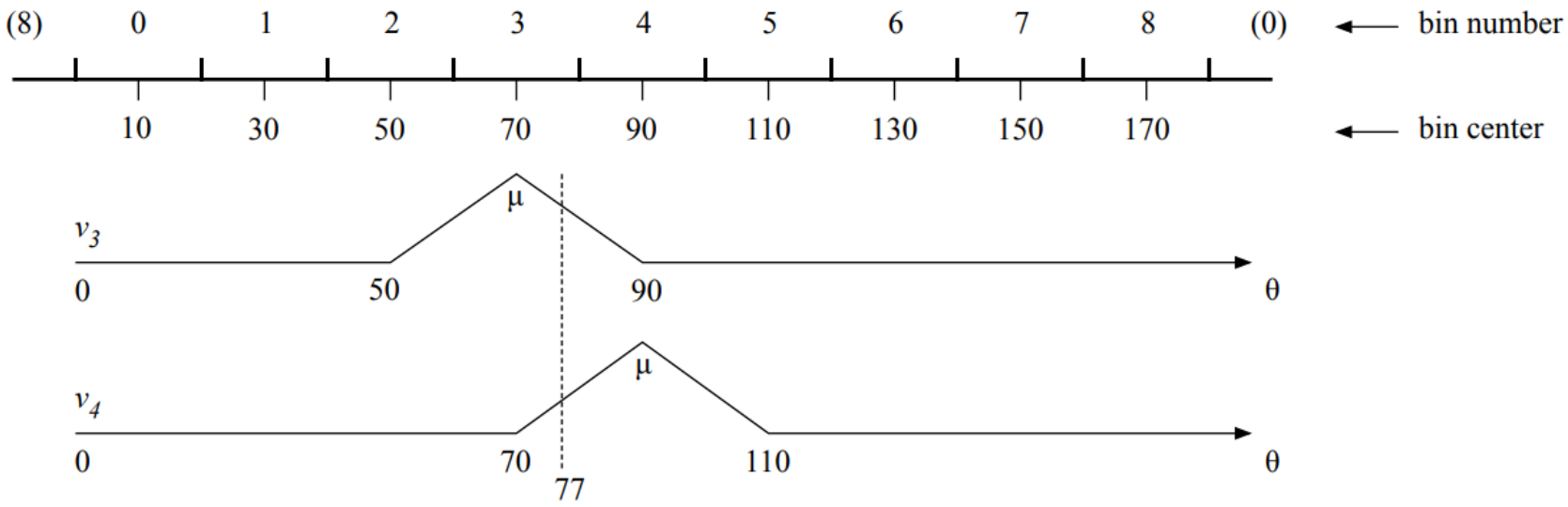
Blocks (2 x 2 cells, stride : 1 cell)



Region of interest (fixed size)

Creating Gradient Histograms

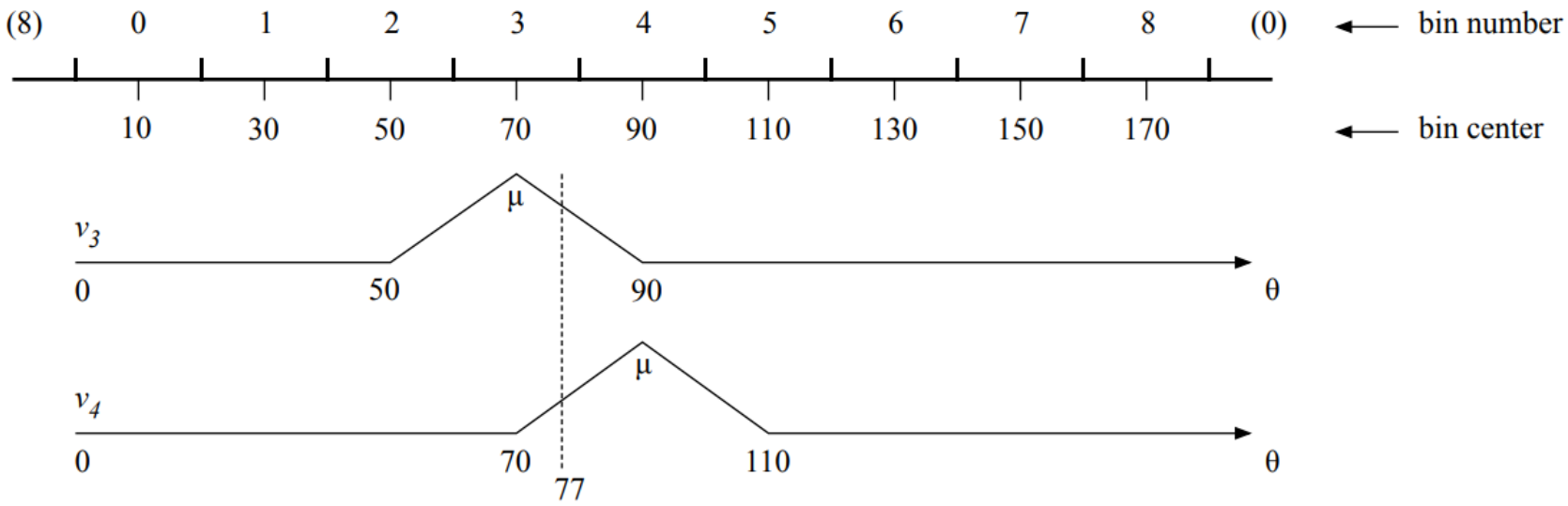
- 9 bins for 180° (**Q:** why not 360° ?)



https://courses.cs.duke.edu/fall17/compsci527/notes/hog.pdf

Creating Gradient Histograms

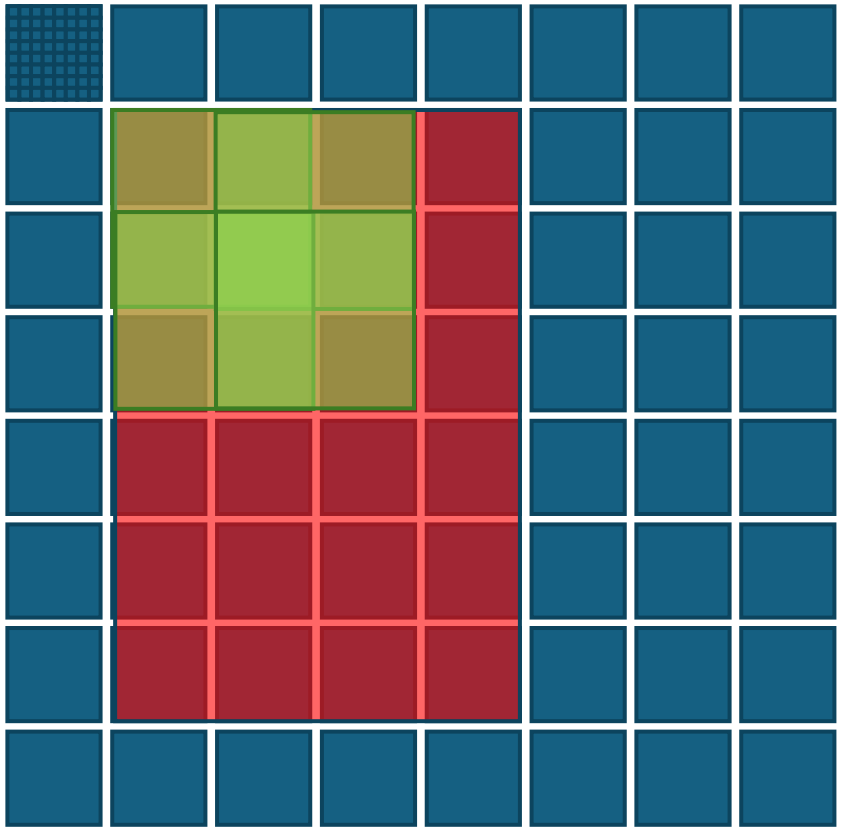
- 9 bins for 180° (**Q:** why not 360° ?)
- For each pixel in cell C_i compute (8x8=64) gradient vectors
- *Don't count*, **add** gradient magnitudes - "voting by bilinear interpolation"



https://courses.cs.duke.edu/fall17/compsci527/notes/hog.pdf

Block descriptor

- For each block of 2 x 2 cells, concatenate the 4 histograms to a block descriptor **b**
- Normalize b: $b \leftarrow \frac{b}{\sqrt{\|b\|^2+\epsilon}}$
- Since each cell is covered by four blocks, each histogram is represented four times with four different normalizations (except for cells at the border of the ROI)



HOG descriptor

- All block descriptors in a ROI are concatenated into a HOG feature vector h
- Normalize h : $h \leftarrow \frac{h}{\sqrt{\|h\|^2+\epsilon}}$
- Clip every entry : $h_n \leftarrow \min(h_n, \tau)$
- Normalize again : $h \leftarrow \frac{h}{\sqrt{\|h\|^2+\epsilon}}$

- With ROI 128 x 64, we have 16 x 8 cells and 15 x 7 blocks, resulting in a 3780-dimensional feature vector

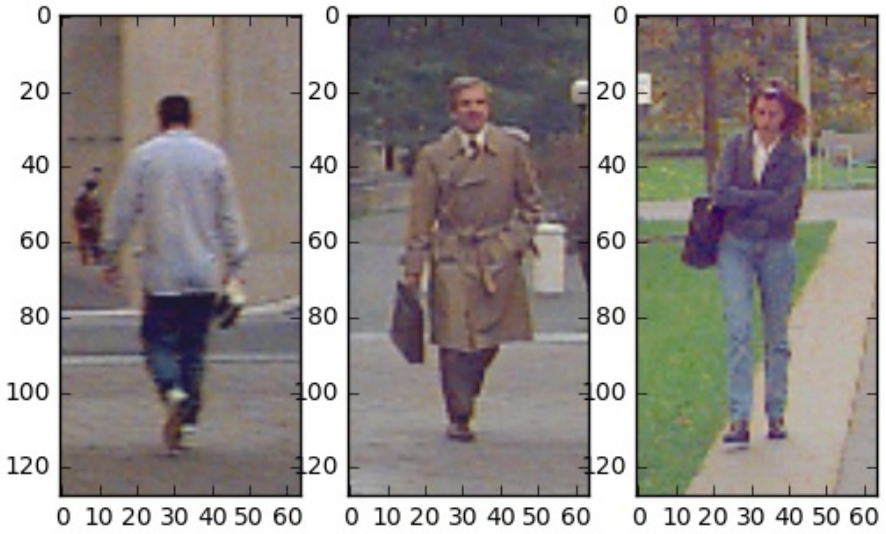
h now needs to be classified

- Binary classification (pedestrian vs. Rest)
- The original work used a SVM
- Others have used random forests, gaussian mixtures, etc.
- The classifier has to be trained with labeled data

https://www.youtube.com/watch?v=QaeJQTVnB-Y

Ex. 4.3: track objects with HOG

- read the paper
- implement and test gradient histograms
- implement HOG descriptor
- detect pedestrians (using kNN, SVM, Logistic Regression)



http://cbcl.mit.edu/cbcl/software-datasets/PedestrianData.html

FREIE UNIVERSITÄT BERLIN

Summary

- HOG features are based on histograms of gradient orientations (of so-called cells), concatenated over blocks to create a block descriptor, further concatenated to form the feature vector
- Special type of histogram: we sum up the gradient magnitudes rather than count the gradients that fall in a histogram bin
- HOG can be used to detect not just pedestrians (given a respective dataset)
- HOG is the feature, the classifier can be anything you like