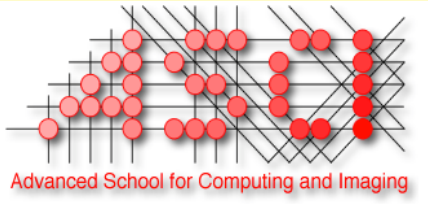
Computer Vision by Learning

Cees Snoek
Laurens van der Maaten
Arnold W.M. Smeulders
with Shih-fu Chang, Columbia University




UNIVERSITY OF AMSTERDAM

TU Delft

Abstract

Computer vision has been revolutionized since the year 2000. Learning from examples is now leading. None of the methods for learning in computer vision is older than 15 years. In the course we will discuss methods of computing, learning to distinguish objects in general and in very specific classes, both from a computer vision as well as from a learning point of view. The course is supplemented with practical work and is completed with an assignment.

Administration

Tuesday to Friday

Lectures	0930-1215	D1.116 (1.115)
Lunch	1215-1330	on your own
Lab	1330-1700	D1.111

Monday

Lecture **Shih-Fu Chang**	0930-1215	G2.10
Lunch	1215-1400	on your own
Lab	1400-1700	G2.02

Lab

Lab 1	Measuring invariance
Lab 2	Pedestrian detection
Lab 3	Learning object and scene detectors
Lab 4	Fine-grained categorization using attributes
Lab 5	Your own research problem

Each team of 2 persons hands in a 10-page report using CVPR style sheet, 2 pages per lab.

Deadline: **Monday April 21, 2014.**

Email to: cgmsnoek@uva.nl

Overview

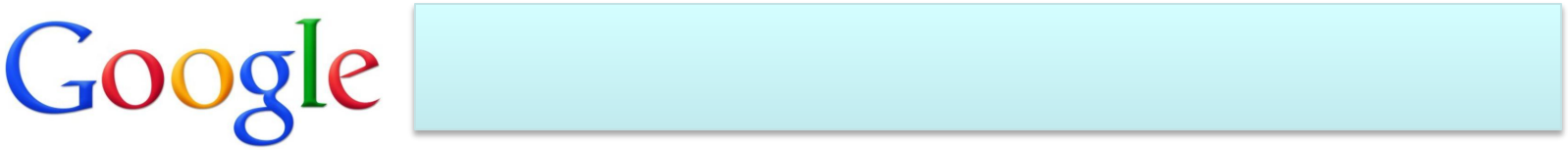
1. Introduction, types of concepts, relation to tasks, invariance
2. Observables, color, space, time, texture, Gaussian family
3. Invariance, the need, invariants, color, SIFT, Harris, HOG
4. BoW overview, what matters
5. On words and codebooks, internal and local structure, soft assignment, synonyms, convex reduction, Fisher & VLAD

1. Introduction

When you type a question in Google search or any other main stream search engine, it goes through its records to find out what other people have found important about the topic. How is a computer capable of converting digital camera recordings to a notion what is on the picture? How does it know how a boat looks like a boat, when there are so many different types and so many different views? And what is harder to recognize, a fork or a knife? A glass or a plate? We discuss concepts and aspects and how a computer can learn to recognize these things in an image.

What does Google know?

When you type in a question in Google, it goes through its records to see what other people have used about the topic.



Brilliant move. They know what other people know.

What do other people know?

We start from the digitized light

How do we convert numbers into meaning?

How does your eye learn this?

Mankind’s great invention

Once upon a time, someone pointed at

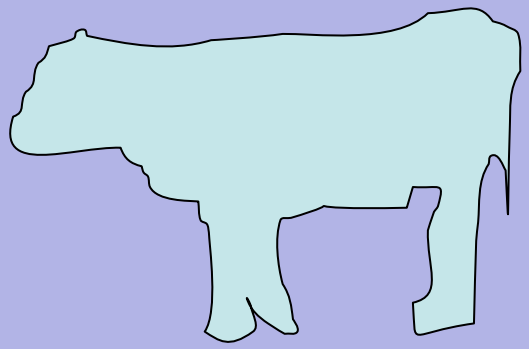
and said “look”...

The invention of visual recognition and of language:

Vision and words

Later, “look” became “cow”. Things got their names.

It is widely assumed: we first outline then recognize.

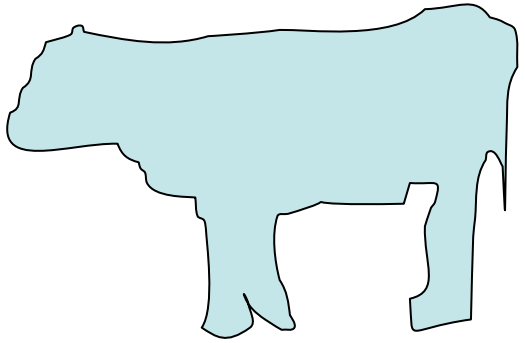


Learn to recognize objects

What knowledge is needed for proper segmentation?

When is segmentation actually needed?

Quiz: What actions require a. segment object b. recognize c. both?



Learning in Computer Vision

What is *tea*?

This is the semantic gap. Learn, not model.

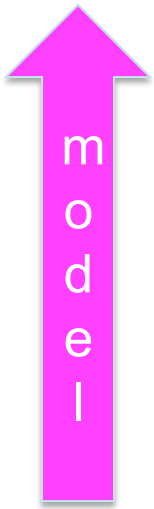
Smeulders PAMI 2000

Things learned in Computer Vision

A concept is a named entity *bicycle*, *2 persons* or
a mass-good *grass*, *ants*, *tea*, *measles* or
a labeled scene *mountainous*, *birthday*

An aspect is a state of *happy*, *dried*, *atherosclerosis*

Tasks in learned Computer Vision



Learn concepts given labeled examples.

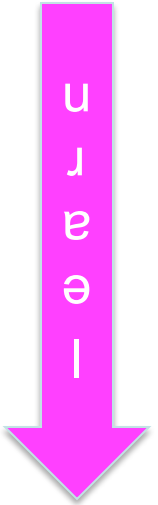
Localize concepts given localized examples.

Localize concepts given labeled examples.

Search for examples similar to this.

Explain evidence for concepts.

Estimate variance intrinsic or extrinsic to concept.

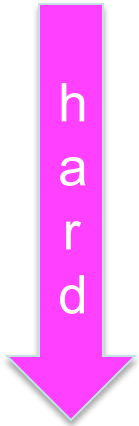


Counting aspects

Generalize aspects over concepts (write a paper).

Numbers of examples

Number of examples per concept



50	gives a good insight in variations
1000	covers even rarities
1	if this is the only example, search for look-alikes
0	other sources of knowledge
5 – 10	hard to separate out accidental variation
-1	(write a paper)

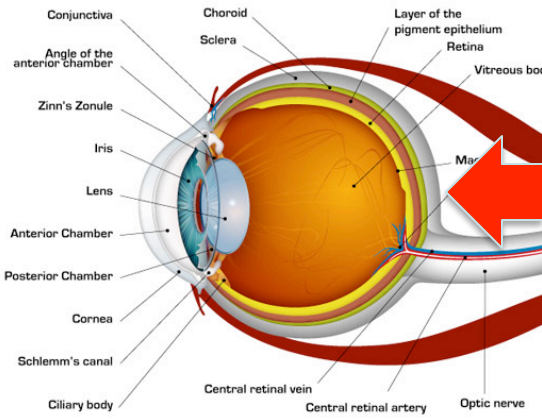
Quiz 1: Name concept subtypes.
2: Visual tasks per subtype?

Invariance

Human knowledge is captured in invariances. Statements which are always true, regardless of accidental conditions.

Learning starts from an analysis of variances.

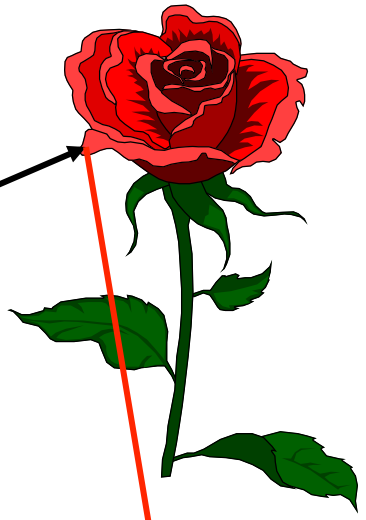
In the eye, the first level invariants are here.

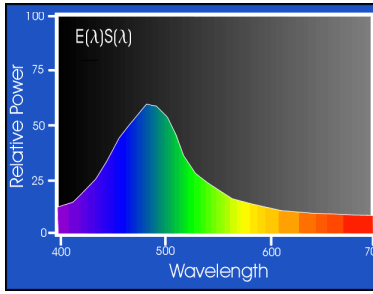


2. Observables

We recapitulate color, spatial, temporal and texture observables. Color is powerful as pixel color observations may lead to object intrinsic information separated from image accidental information. Spatial observables form the Taylor expansion of the image. We discuss the Gaussian implementation. The temporal equivalent of Gauss is given and demonstrated. Finally, texture is defined by Gabor filters. The observables are brought together as the dimensionally separable Gaussian family.

The formation of the image pixels



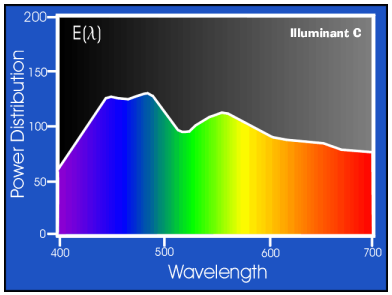


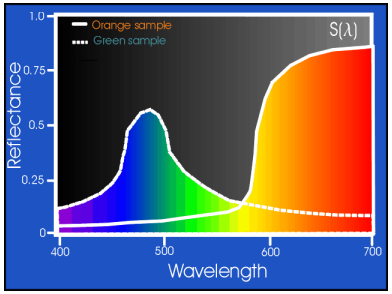
$S(\lambda)$

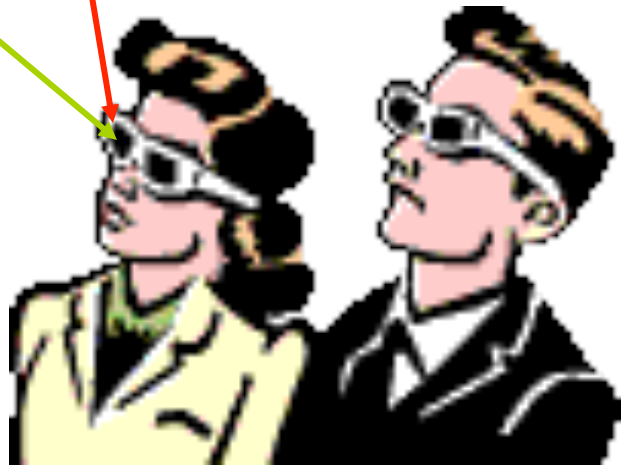
$r(\lambda)S(\lambda)$

object reflection

Source

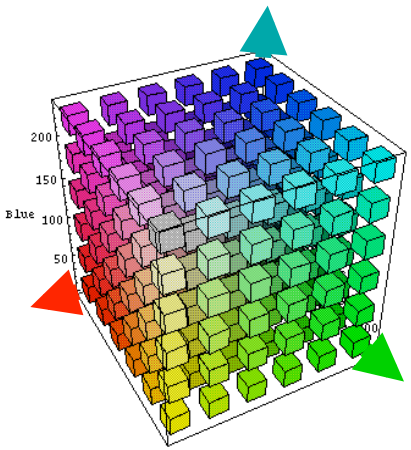


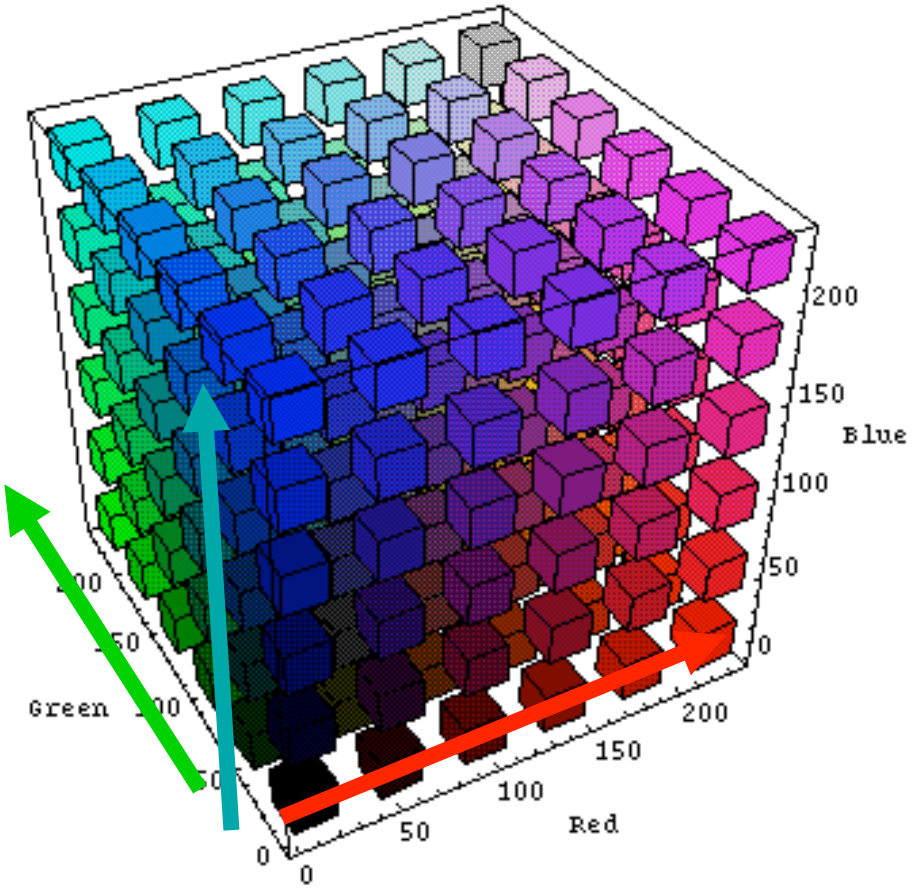




E = (R,G,B)

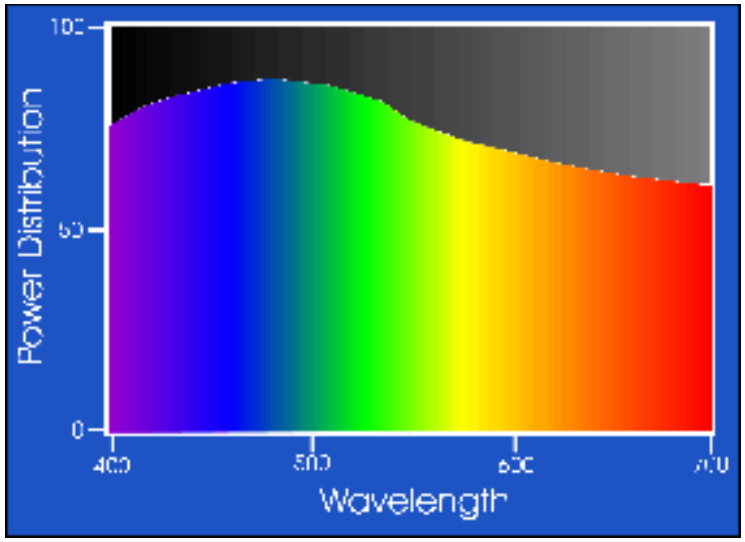
$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} \int_\lambda e(\lambda)\rho(\lambda)f_R(\lambda)d\lambda \\ \int_\lambda e(\lambda)\rho(\lambda)f_G(\lambda)d\lambda \\ \int_\lambda e(\lambda)\rho(\lambda)f_B(\lambda)d\lambda \end{pmatrix}$$





HSI = Human sensation of light

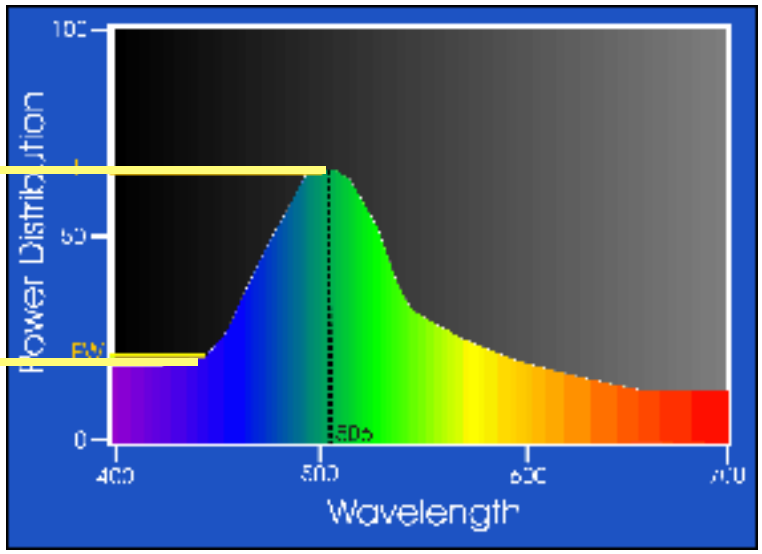
Hue:	dominant wavelength	$\lambda(E_H)$
Saturation:	purity of the colour	$(E_H - E_W)/E_H$
Intensity:	brightness of the colour	E_W



white light

E_H

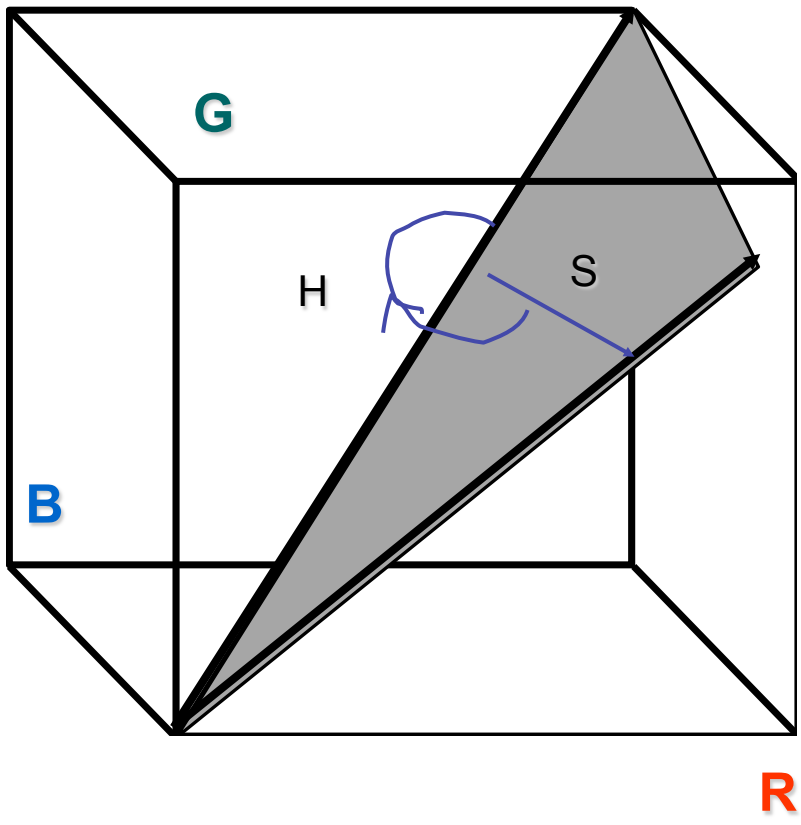
E_W



green light

HSI in **E**

$$\begin{pmatrix} H \\ S \\ I \end{pmatrix} = \begin{pmatrix} arctg\left(\dfrac{\sqrt{3}(G-B)}{(R-G)+(R-B)}\right) \\ 1-\dfrac{\min(R,G,B)}{R+G+B} \\ R+G+B \end{pmatrix}$$

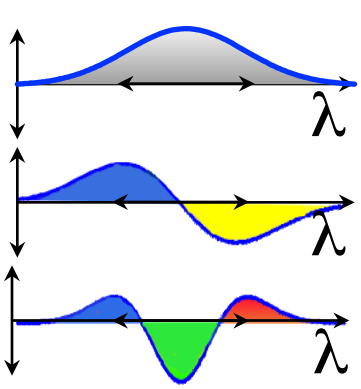


H & S are illumination invariant body properties, separating the accidental recording conditions from the intrinsic body condition.

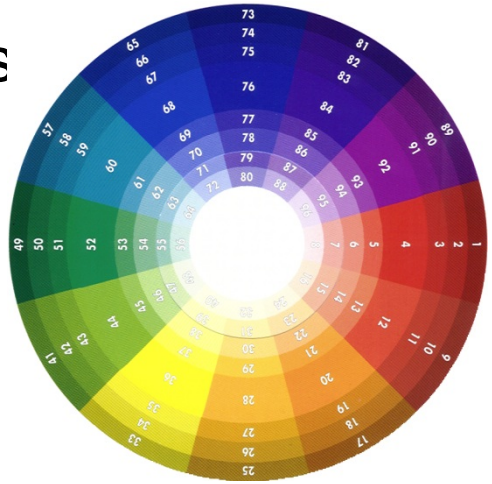
Opponent colors

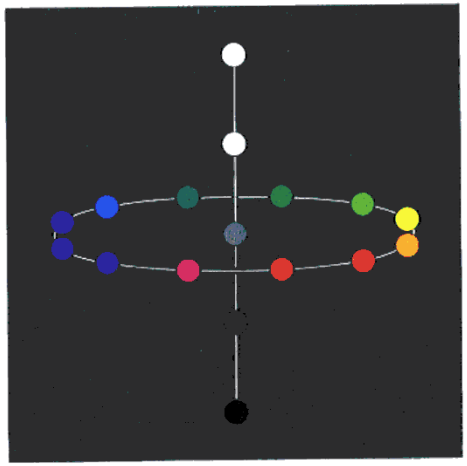
Human perception combines (R,G,B) respons
of the eye in opponent colors

$$\begin{pmatrix} \text{Luminance} \\ \text{BlueYellow} \\ \text{PuperGreen} \end{pmatrix} = \begin{pmatrix} R+G+B \\ \frac{1}{2}(R-G) \\ \frac{1}{4}(2B-R-G) \end{pmatrix}$$



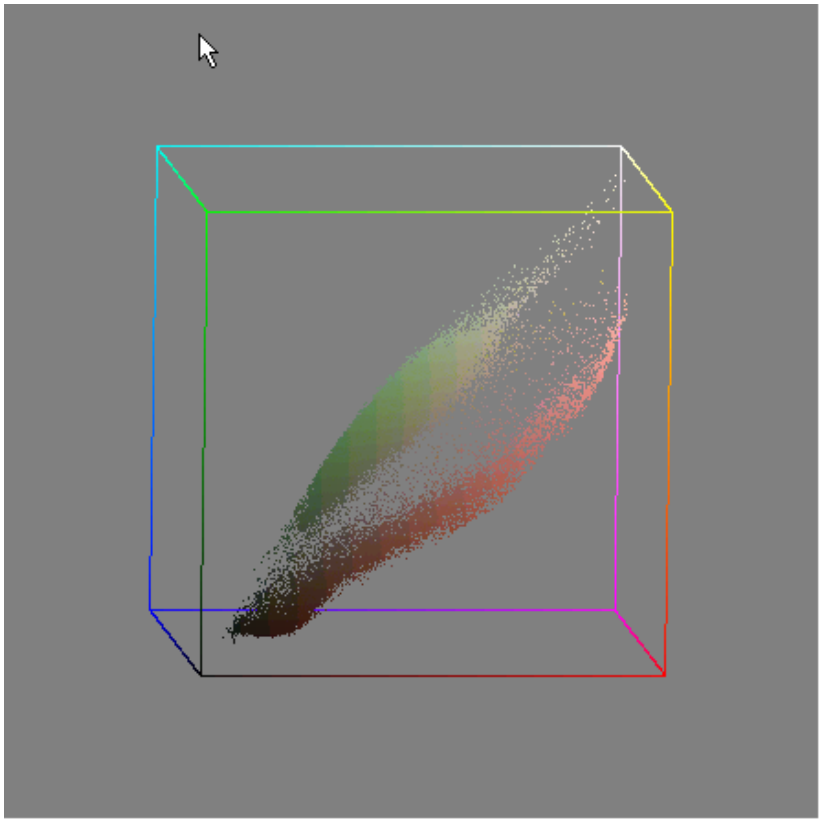
as it maximizes the contrast.





Pixel-level variation

Pixel values have accidental variation due to body reflection.

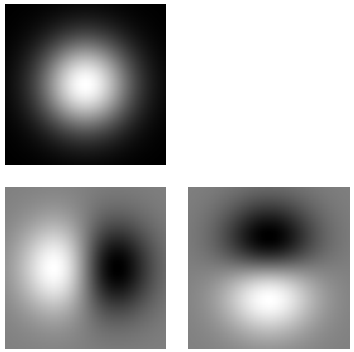


Cancel them out, first.

Spatial observables

$$\hat{E}(x) \approx \sum_{j=0}^{n} \frac{1}{j!} (x \nabla_x)^j \hat{E}(x)$$

Taylor expansion of the image around **x**



Differentials are important to understand the local structure of the image.

Gaussian filters

$$\hat{E}(x) \approx \sum_{j=0}^{n} \frac{1}{j!} (x \nabla_x)^j \hat{E}(x)$$

For discretely sampled signal use the Gaussians

$$\nabla_x{}^j \hat{E} = \frac{\partial^j \hat{E}}{\partial^j x} = E * G_{x^j}^{x_0, \sigma_x}$$

Among the class of linear filters, Gauss is separable by dimension, rotationally uniform, sloping to zero, adds no maxima and so on. It is the preferred brand of local filters.

Fast approximate recursive implementation:

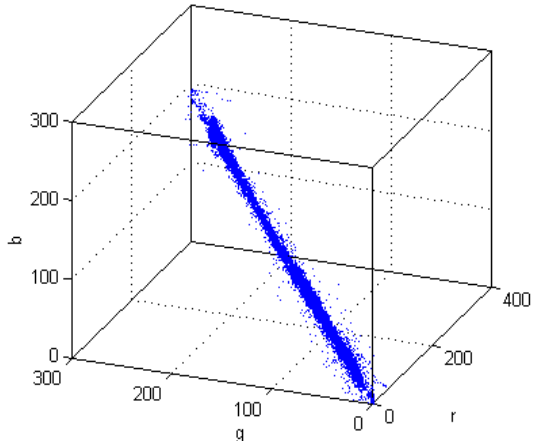
Geusebroek, Van de Weijer & Smeulders 2002

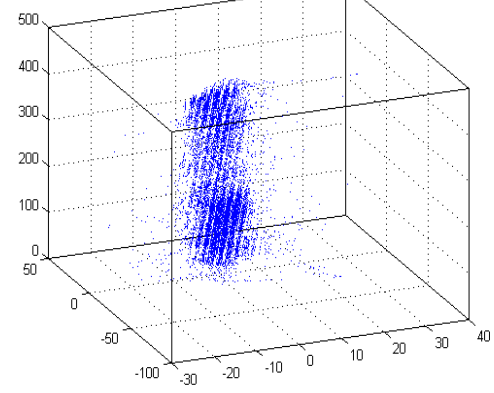
Color Gaussian space

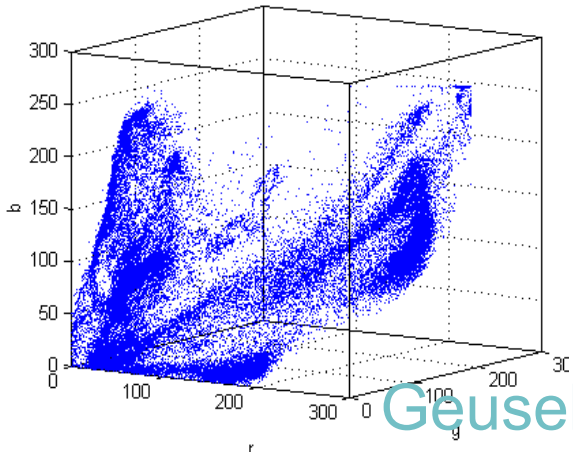
$$\begin{pmatrix} E \\ E_\lambda \\ E_{\lambda\lambda} \end{pmatrix} = \begin{pmatrix} 0.06 & 0.63 & 0.27 \\ 0.30 & 0.04 & -0.35 \\ 0.34 & -0.60 & 0.17 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

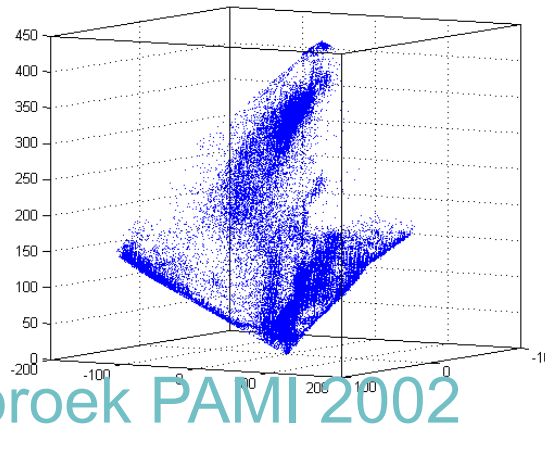
(R,G,B)-pdf

(E_0,E_λ,$E_{\lambda\lambda}$)-pdf





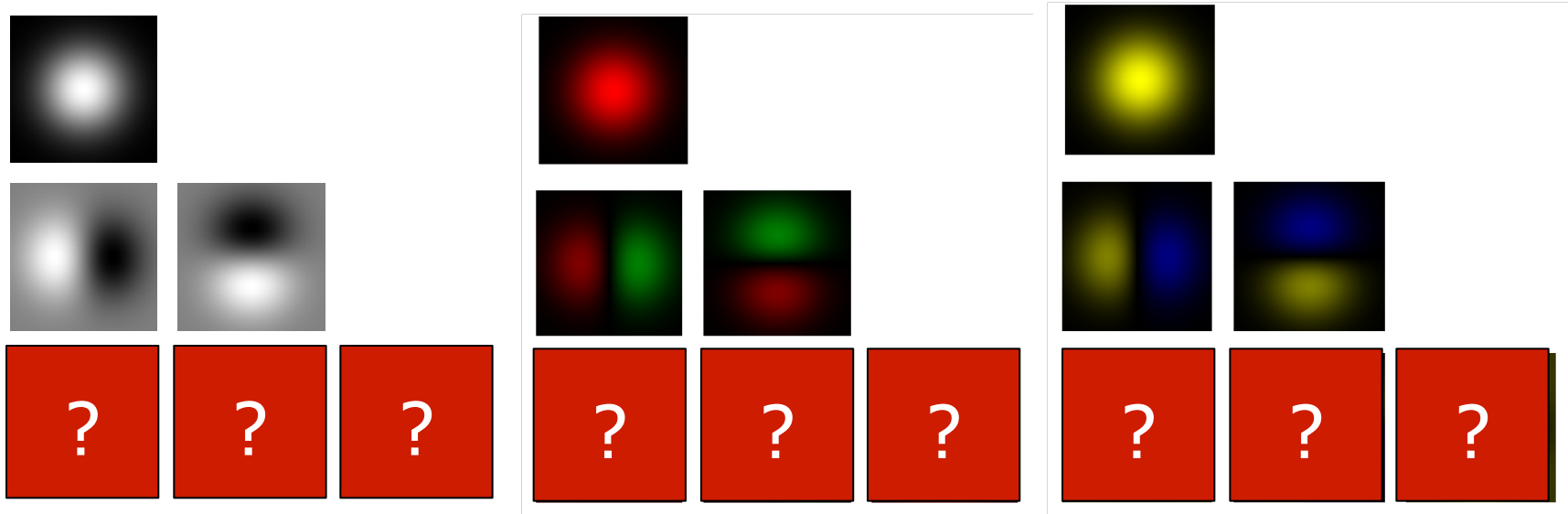




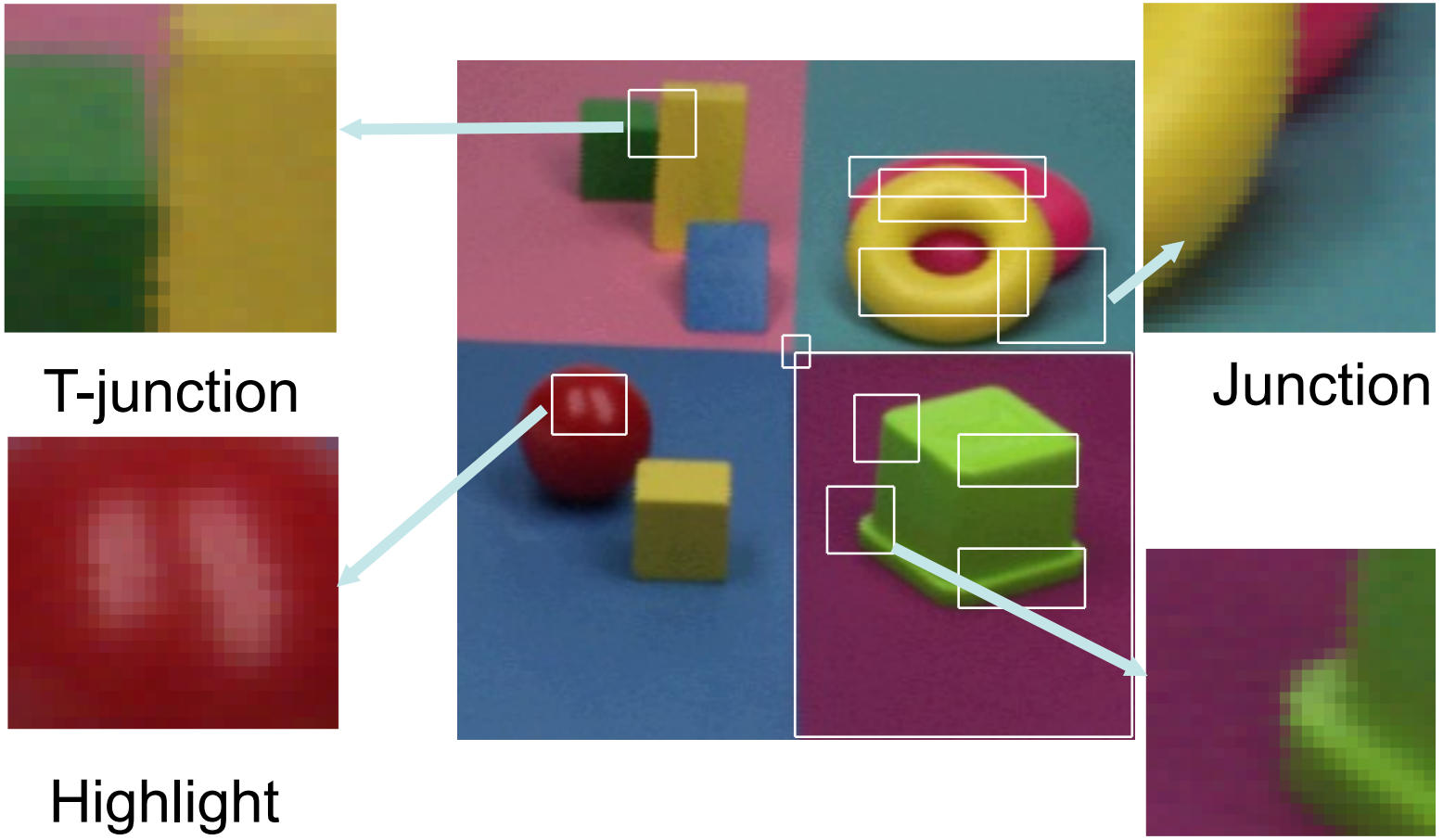
Geusebroek PAMI 2002

Quiz: Draw 2nd order

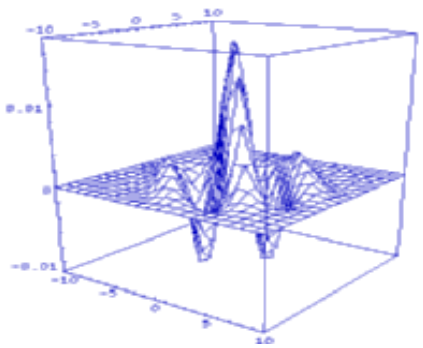
Gaussian in zero and first order and their receptive fields:



Taxonomy of diff-image structure



Gabor texture



The 2D Gabor function with parameters: *u, v, σ*:

$$h(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\delta^2}} e^{2\pi j(ux+vy)}$$

Gabor for texture in Fourier-space

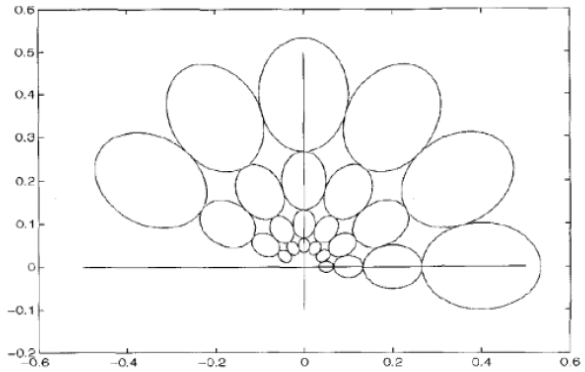


Fig. 1. The contours indicate the half-peak magnitude of the filter responses in the Gabor filter dictionary. The filter parameters used are $U_h = 0.4$, $U_l = 0.05$, $K = 6$, and $S = 4$.

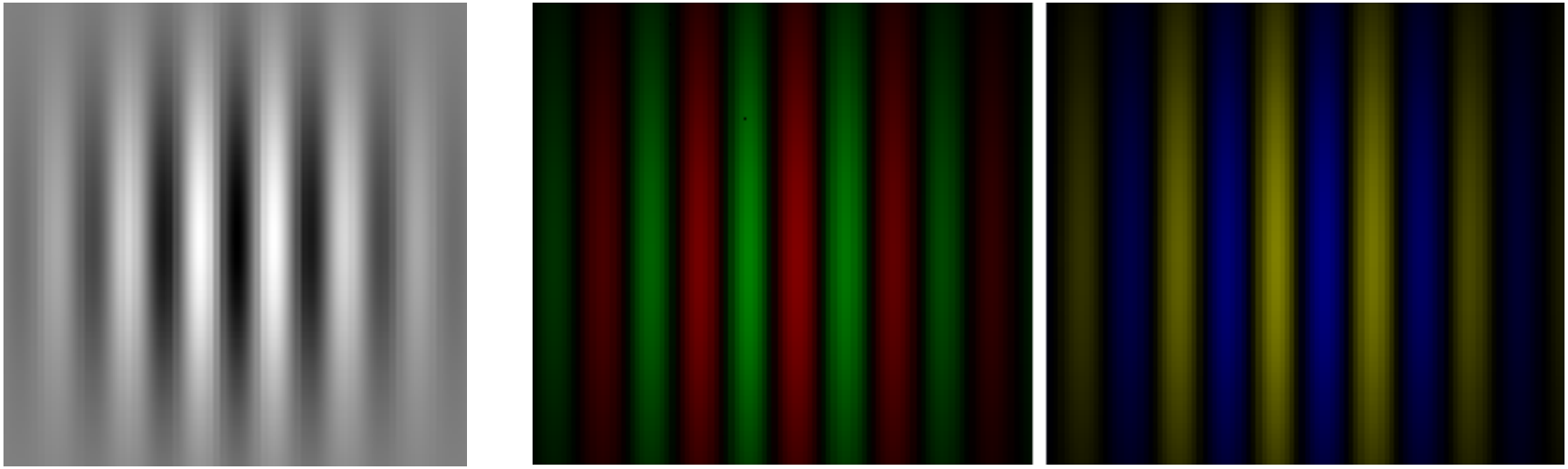
$$a = (U_h/U_l)^{-\frac{1}{S-1}}, \quad \sigma_u = \frac{(a-1)U_h}{(a+1)\sqrt{2\ln 2}},$$

$$\sigma_v = \tan\left(\frac{\pi}{2k}\right)\left[U_h - 2\ln\left(\frac{\sigma_u^2}{U_h}\right)\right]\left[2\ln 2 - \frac{(2\ln 2)^2\sigma_u^2}{U_h^2}\right]^{-\frac{1}{2}}, \quad (4)$$

Manjunath & Ma

Gabor texture

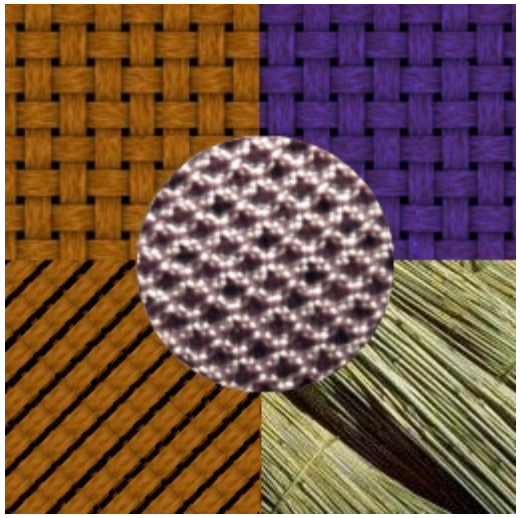
The receptive fields for (u, v) measured locally

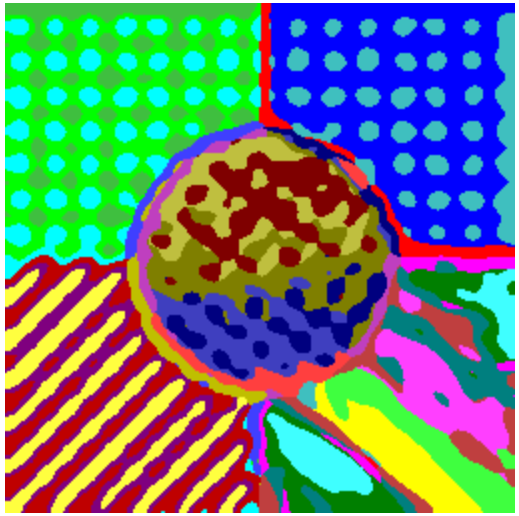


Grey and opponent color feature sets.

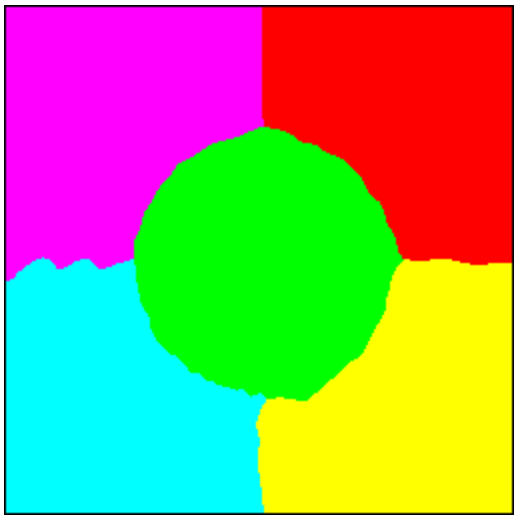
Hoang SP 2003

Gabor texture





K-means cluster of RGB

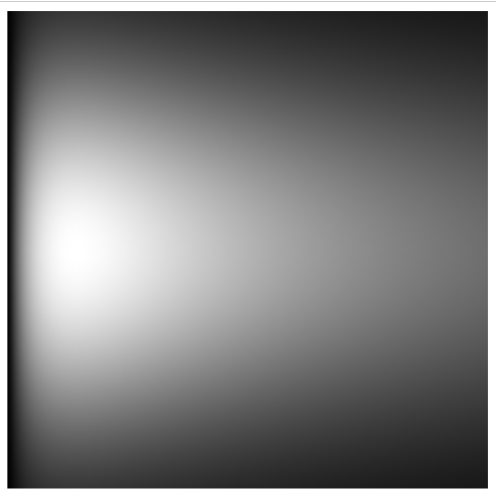


K-means cluster Gabor opponent

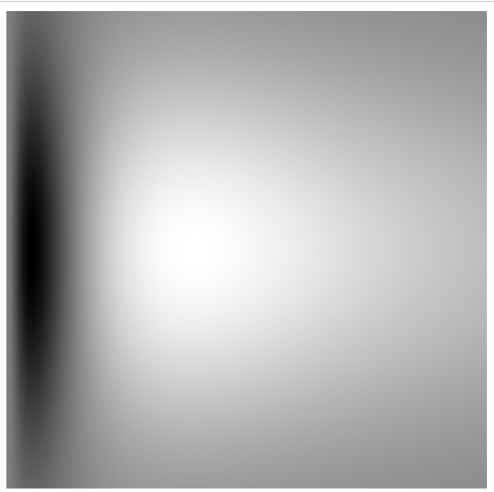
Hoang ECCV 2002

Gaussian temporal

Gaussian equivalent over x and t:



zero order

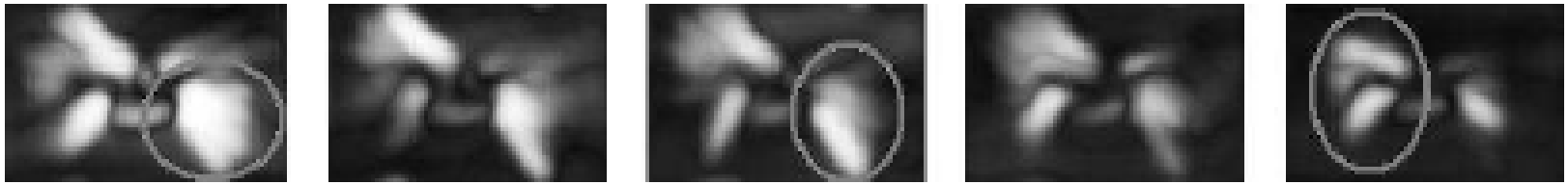


first order over t

Good observables ≅ easy algorithm

Periodicity:

Detect periodic motion by one steered filter:



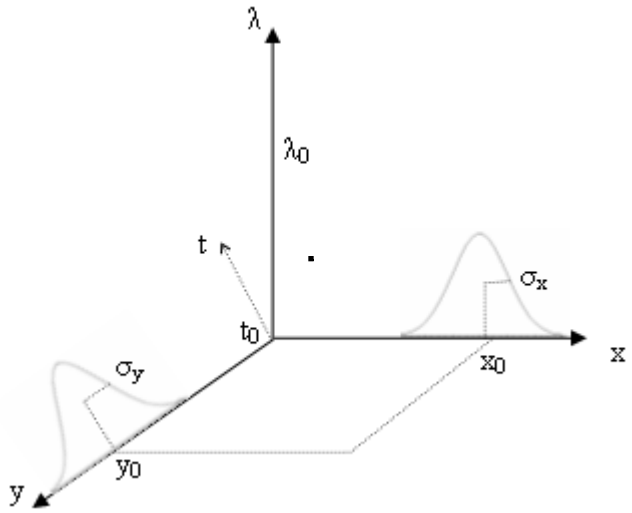
Deadly simple algorithm…

Burghouts TIP 2006

Gaussian: the complete family

The basic visual observables are:

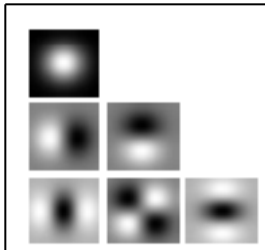
$$G(x, y, \lambda, t)$$

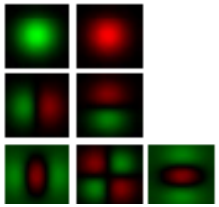


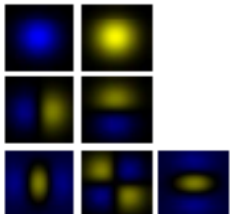
$$G^{x_0;\sigma_x} * G^{y_0;\sigma_y} * G^{\lambda_0;\sigma_\lambda} * G^{t_0;\sigma_t}$$

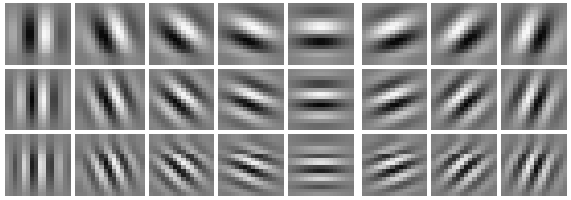
Gaussian: the complete family

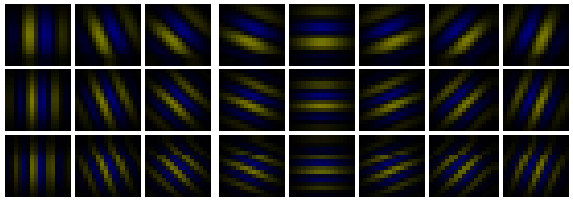
Among the class of linear filters, only Gauss is separable by dimension when rotationally uniform, and adds no maxima.

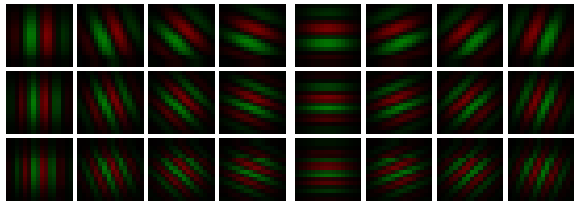


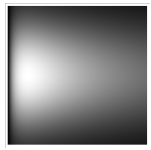


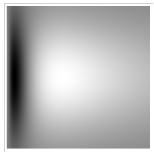












All observables up to first order color, second order spatial scales, eight frequency bands & first order in t.

Appendix: Quality of observations

At all levels of processing:

1. Good features capable of capturing all *relevant* variations.
2. Good samples capture all *present* variations.

    Sampling: stratified, selective, complete?

3. Dimension is the unit.

    In general ease of invariance achieved by normalization.

Appendix: Errors of Observation

In the previous chapter, we have studied observables.

Beware of their error when handling them.

Errors: systematic & stochastic.

All acceptable until they propagate during your computations.

Propagation rules:

For a function: $\Delta f(x) \approx \Delta x \,.\, |f'(x)|$

For the sum of two observables: $\Delta x = \Delta x_1 + \Delta x_2$

Product of two measurements: $\frac{\Delta x}{x} = \frac{\Delta x_1}{x_1} + \frac{\Delta x_2}{x_2}$

Warning: beware of division by 0!

3. Invariants

We recapitulate properties of images which are caught by invariant features. Invariance aims to exclude all irrelevant variations. Color invariants are very powerful for everything related to illumination, and in the end simple. We discuss the quality of invariant features: invariance and discrimination. We cannot go into ways to design invariants. Compositions of observables and invariants lead to the powerful SIFT and Harris patch descriptors.

The need for invariance

There are a million appearances to one object

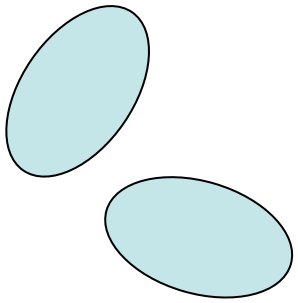
The same part of the same shoe does not have the same appearance in the image. This is the sensory gap.

Remove unwanted variance as early as you can.

The need for invariance

A feature g is invariant under condition (transform) W caused by accidental conditions at the time of recording, iff g observed on equal objects t_1 and t_2 is constant:

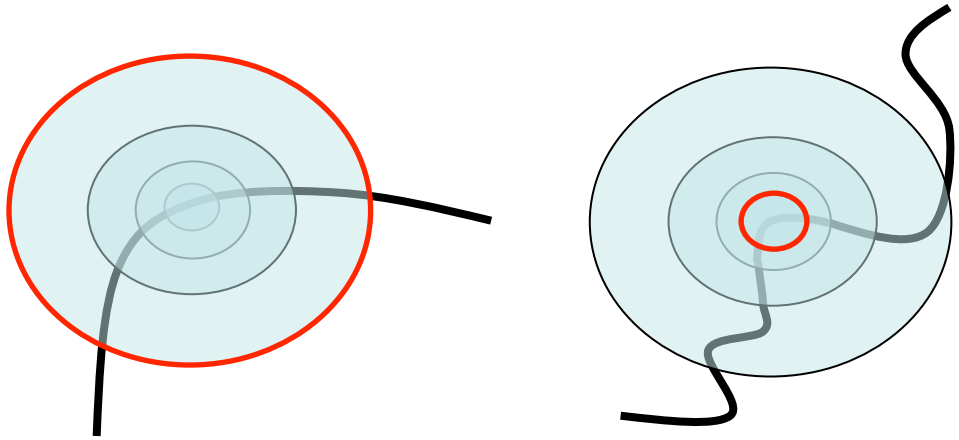
$$t_1 \overset{W}{\sim} t_2 \Rightarrow f_{t_1} = f_{t_2}$$



Length of long axis / short axis independent of scale and rotation.

Quiz: scale invariant detection

What properties are invariant to observation scale?



Scale invariants are at least dimensionless numbers:

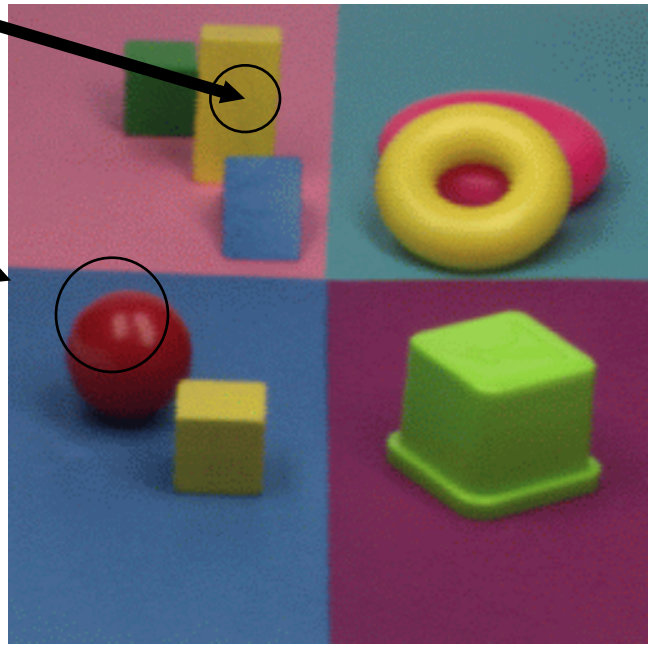
+ The ratio of the long and short axes.
+ Corners.
+ Length squared divided by area.

Color invariance: reflectance

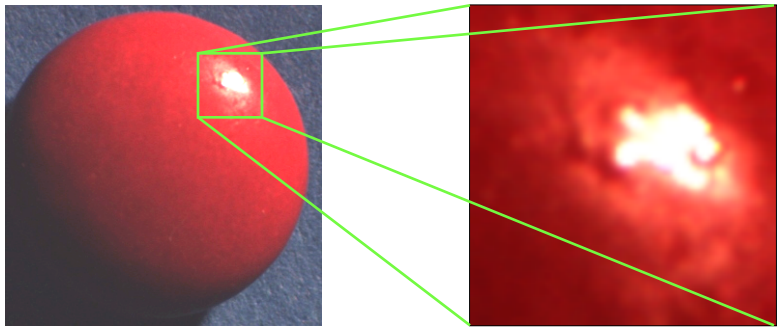
Reflectance model [Shafer]

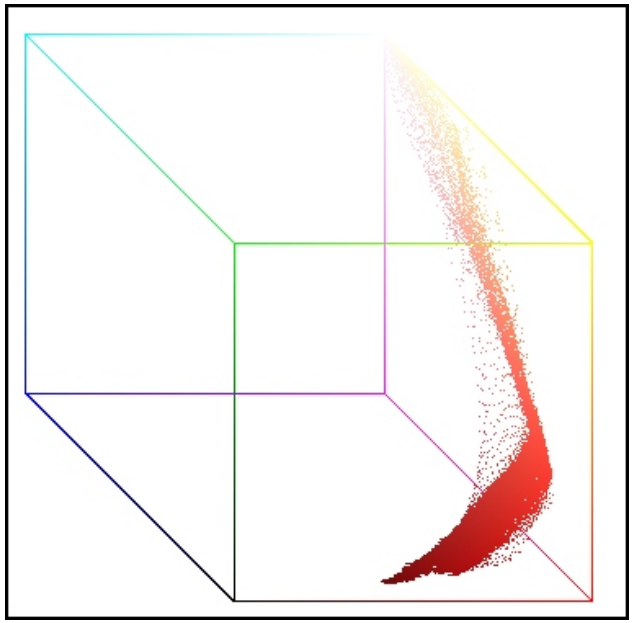
body = $m_b(\vec{n},\vec{s})\int_\lambda e(\lambda)c_b(\lambda)f_C(\lambda)d\lambda$

surface = $m_s(\vec{n},\vec{s},\vec{v})\int_\lambda e(\lambda)c_s(\lambda)f_C(\lambda)d\lambda$



E is viewpoint variant





The surface reflection term:

$$m_s(\vec{n},\vec{s},\vec{v})\int_{\lambda} e(\lambda)c_s(\lambda)f_C(\lambda)d\lambda$$

reduces to simpler form of Phong:

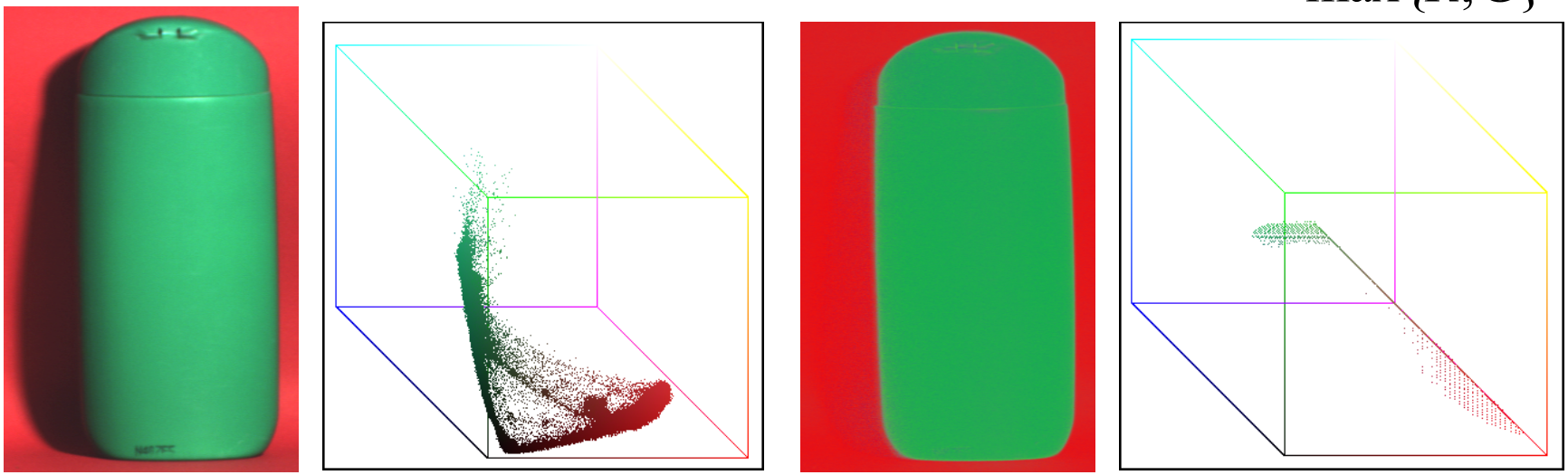
$$m_s(\vec{n},\vec{s},\vec{v}) \propto \cos^n(\alpha(\vec{n},\vec{s},\vec{v}))$$

C is viewpoint invariant

$$c_1(R,G,B) = \arctan\frac{R}{\max\{G,B\}}$$

$$c_2(R,G,B) = \arctan\frac{G}{\max\{R,B\}}$$

$$c_3(R,G,B) = \arctan\frac{B}{\max\{R,G\}}$$



E space

C space

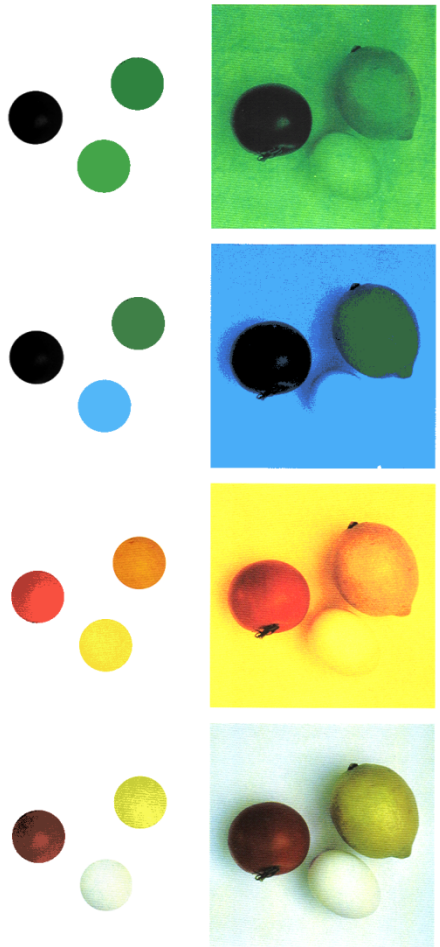
Gevers TIP 2000

M is illumination color invariant

$$\begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} = \begin{pmatrix} \frac{R_{x_1} G_{x_2}}{R_{x_2} G_{x_1}} \\ \frac{G_{x_1} B_{x_2}}{G_{x_2} B_{x_1}} \\ \frac{B_{x_1} R_{x_2}}{B_{x_2} R_{x_1}} \end{pmatrix}$$

Taking the natural log

$$\begin{pmatrix} \ln m_1 \\ \ln m_2 \\ \ln m_3 \end{pmatrix} = \begin{pmatrix} \ln \frac{R_{x_1}}{G_{x_1}} - \ln \frac{R_{x_2}}{G_{x_2}} \\ \ln \frac{G_{x_1}}{B_{x_1}} - \ln \frac{G_{x_2}}{B_{x_2}} \\ \ln \frac{B_{x_1}}{R_{x_1}} - \ln \frac{B_{x_2}}{R_{x_2}} \end{pmatrix}$$



Differential invariants **C'**, **W'**, **M'**

C' is for matte objects and uneven white light:

$$C_\lambda = \frac{E_\lambda}{E} \qquad C_{\lambda\lambda} = \frac{E_{\lambda\lambda}}{E} \qquad C_{\lambda x} = \frac{E_{\lambda x}E - E_\lambda E_x}{E^2}$$

W' is for matte planar objects and even white light:

$$W_x = \frac{E_x}{E} \qquad W_{\lambda x} = \frac{E_{\lambda x}}{E}$$

M' is for matte objects and monochromatic light:

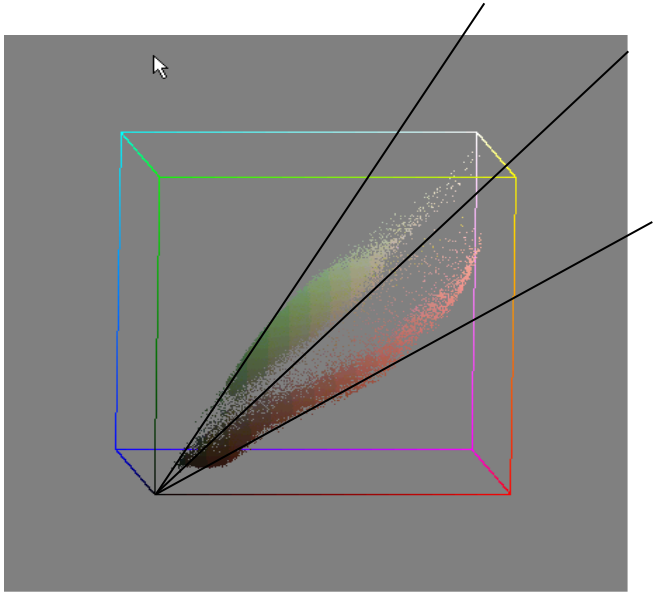
$$N_{\lambda x} = \frac{E_{\lambda x}E - E_\lambda E_x}{E^2}$$

Geusebroek PAMI 2002

Invariance & Discrimination

The most invariant feature is the value “42”. Not *useful*!

Desired invariance ↔ undesired loss of discriminative power.



Retained discrimination

	shadows	shading	highlights	ill. intensity	ill. color
E	-	-	-	-	-
H	+	+	+	+	-
W & W'	-	+	-	+	-
C & C'	+	+	-	+	-
M & M'	+	+	-	+	+
L	+	+	+	+	-

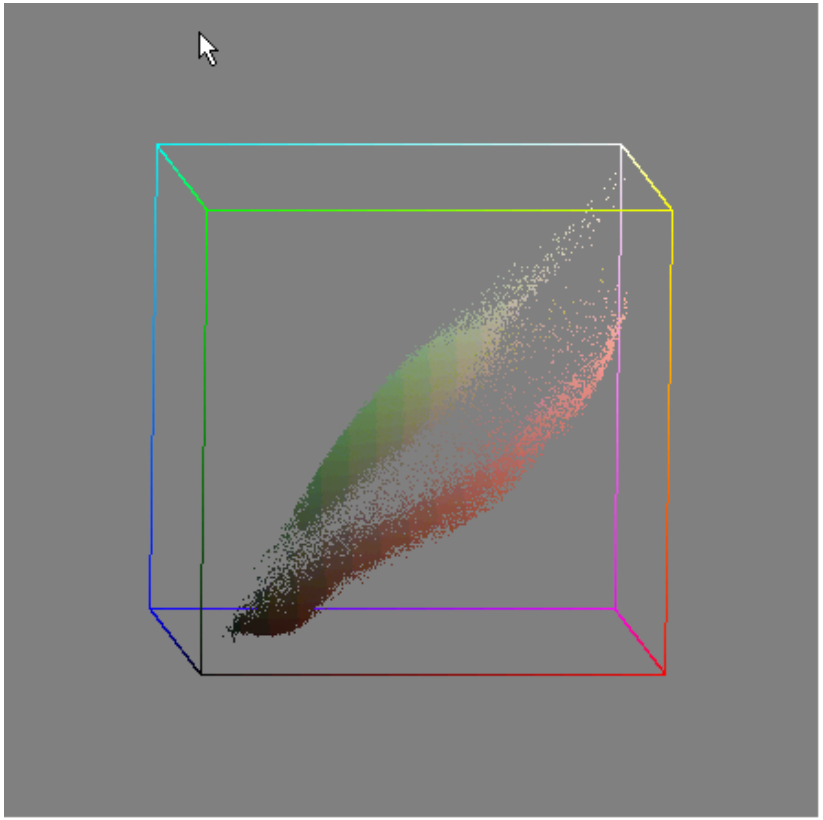
Retained from 1000 colors $\sigma = 3$:

E	990
H	315
W'	995
C'	850
M'	900

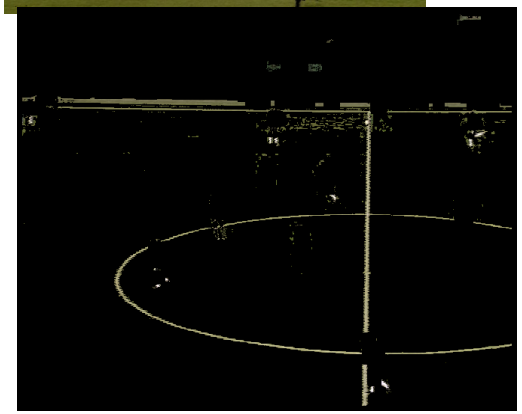
Geusebroek PAMI 2003

Quiz: a. What sources of variance b. Find ways to get invariance

This is an orbit.



Color invariant detection



Tracking invariant appearance

Nguyen PAMI 2004

SIFT

For 4x4 patches, find local gradient directions over t.

Count the directions per patch, 128D SIFT histogram.

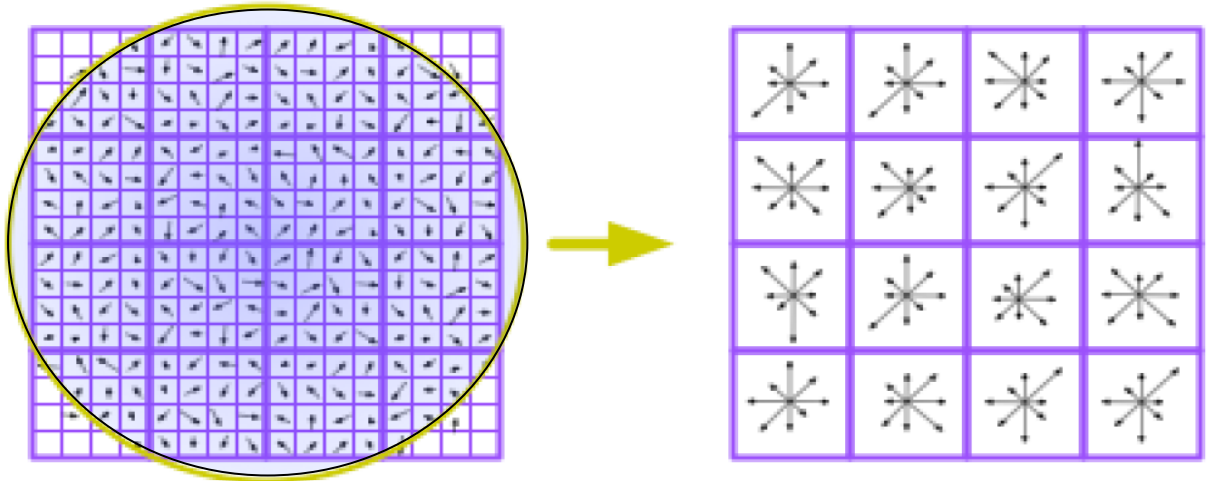
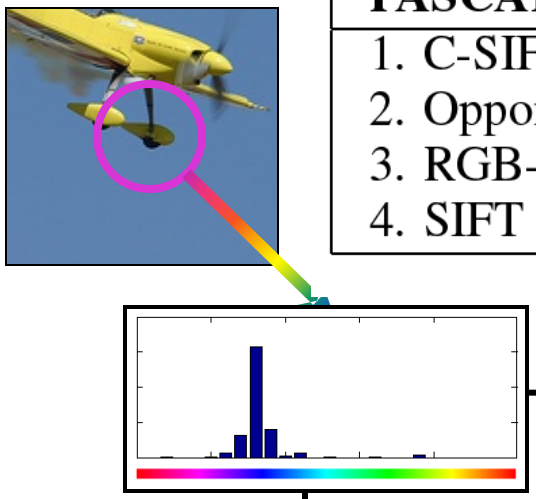


Image gradient directions

histogram = SIFT

Lowe IJCV 2004

Color Descriptors



Recommended Color Descriptors Per Dataset

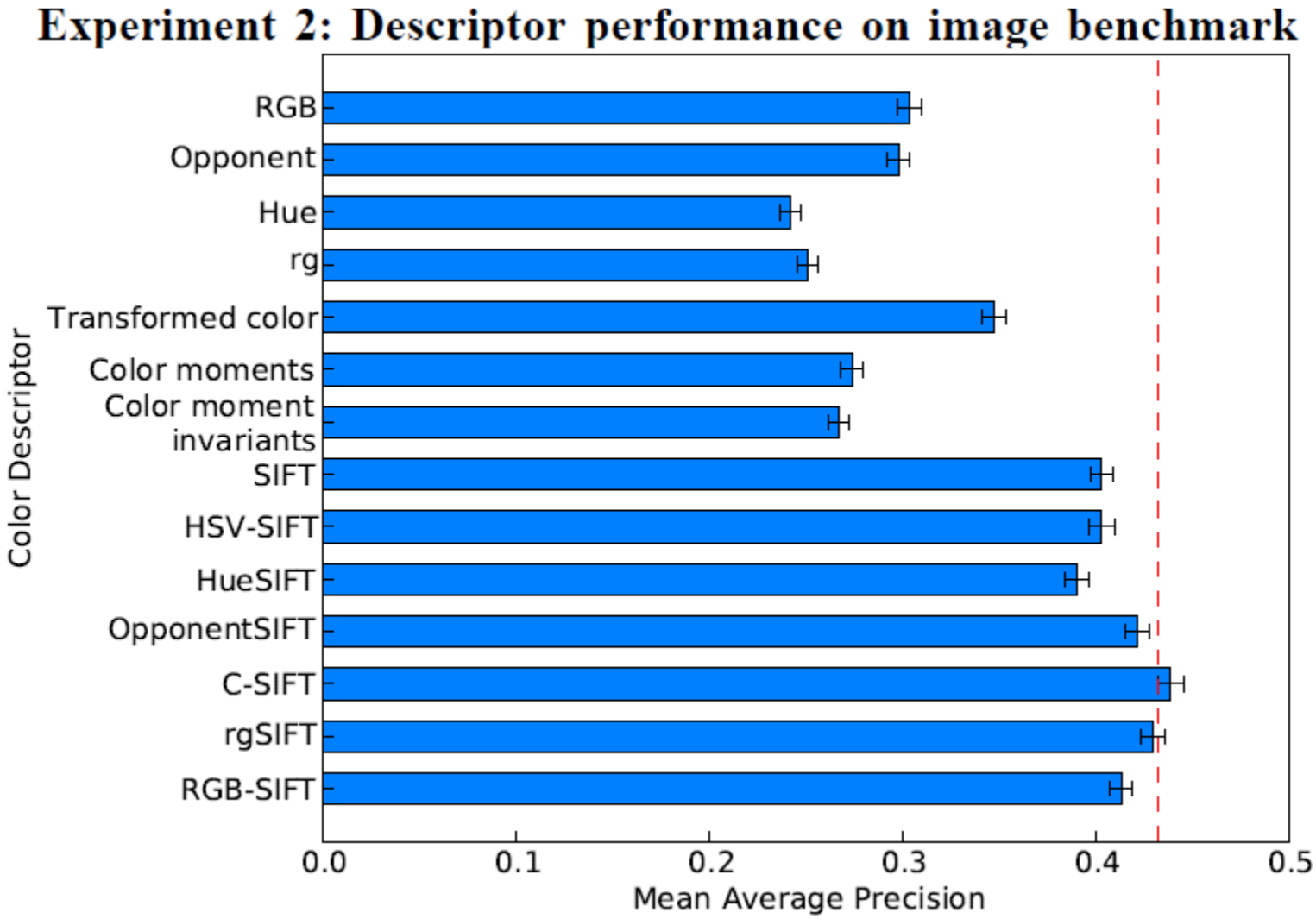
PASCAL VOC 2007	Mediamill Challenge	Unknown Data
1. C-SIFT	1. OpponentSIFT	1. OpponentSIFT
2. OpponentSIFT	2. RGB-SIFT	2. C-SIFT
3. RGB-SIFT	3. C-SIFT	3. RGB-SIFT
4. SIFT	4. SIFT	4. SIFT

Invariance properties per descriptor

	Light intensity change	Light intensity shift	Light intensity change and shift	Light color change	Light color change and shift
SIFT	+	+	+	-	-
OpponentSIFT	+	+	+	-	-
C-SIFT	+	-	-	-	-
RGB-SIFT	+	+	+	+	+

van de Sande PAMI 2010

Color SIFT


Experiment 2: Descriptor performance on image benchmark
RGB
Opponent
Hue
rg
Transformed color
Color moments
Color moment invariants
SIFT
HSV-SIFT
HueSIFT
OpponentSIFT
C-SIFT
rgSIFT
RGB-SIFT
Color Descriptor
0.0
0.1
0.2
0.3
0.4
0.5
Mean Average Precision

Color SIFT

Ranking Positions for Bird: From Baseline to Two-Channel




553 → 83 176 → 12 868 → 34 51 → 10

Ranking Positions for Train: From Baseline to Two-Channel




856 → 63 167 → 66 104 → 67 225 → 99

HOG

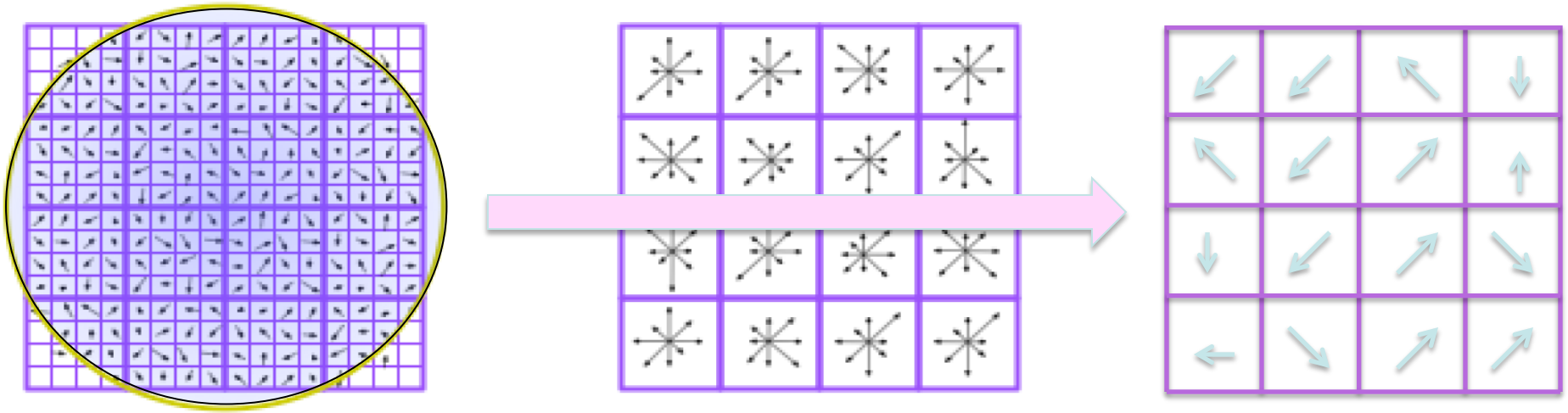
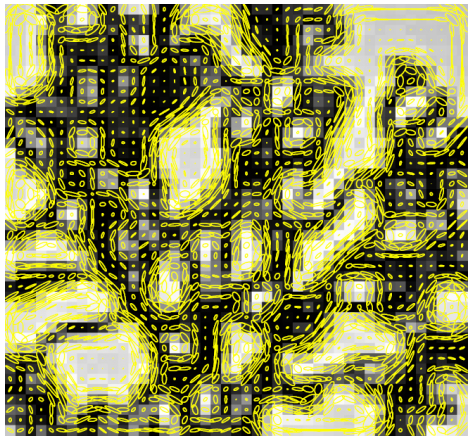


Image gradient directions,

dense smoothed locally normalized
= HOG

Picture: University Texas

Harris' Corner selector



The change energy at **x** over a small vector **u**:

$$E_{xy}(u,v) \approx [u \;\; v]\; M \begin{bmatrix} u \\ v \end{bmatrix}, \quad M = \begin{bmatrix} f_x f_x & f_y f_x \\ f_x f_y & f_y f_y \end{bmatrix}$$

Since M is symmetric, we have

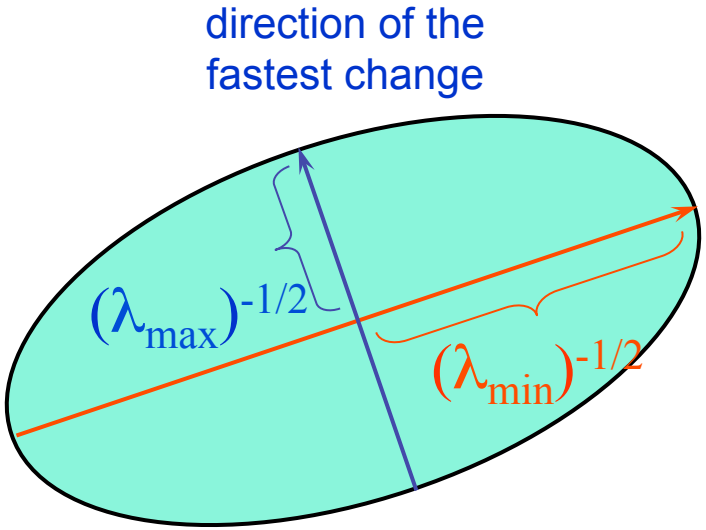
$$M = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

For a corner both should be large.

$$R = \det M - k\left(\text{trace}\, M\right)^2$$

$$\det M = \lambda_1 \lambda_2 = I_x^2 I_y^2 - (I_x I_y)^2$$

$$traceM = \lambda_1 + \lambda_2 = I_x^2 + I_y^2$$



Quiz: What are invariants Harris?

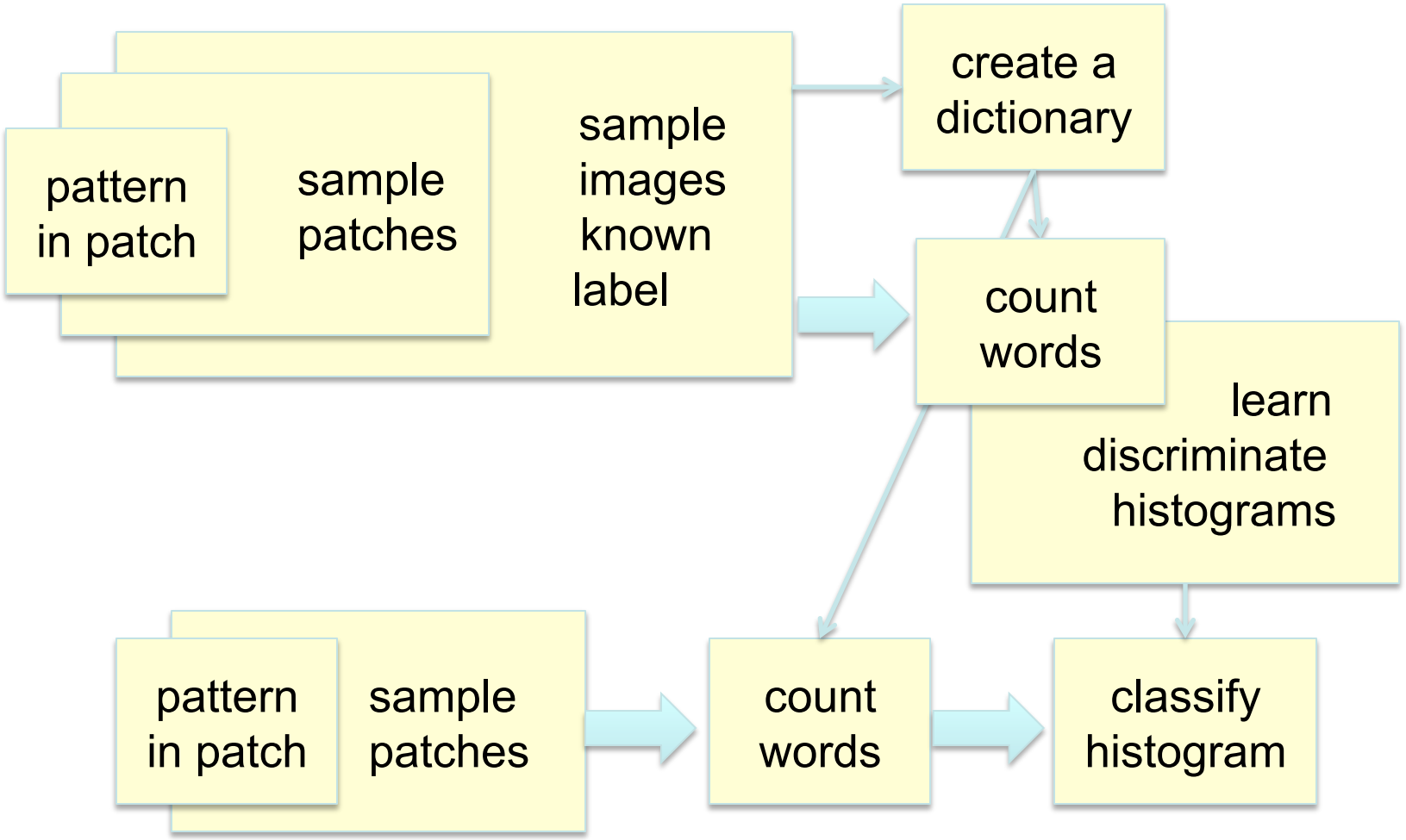
4. Bags of Words

Bag of words is a 2-stage approximation of Bayes' rule which handles the large dimensionality of the feature space and remote correlations.

Before there were words

N images . M patches . D feature dimensions.
M = 1000 patches and D = 128 features minimum.
1000 images ~ 7 minutes / 100 Mbyte storage

Word processing chain



Capture the pattern in patch

Measure the pattern in a patch with abundant features.

SIFT is good,

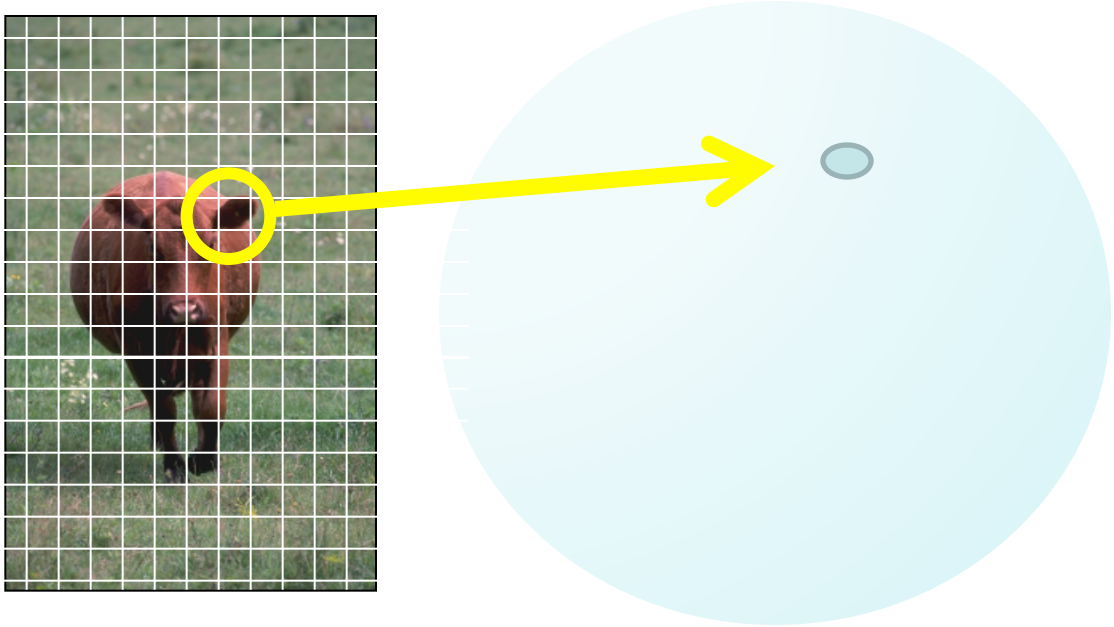
cSIFT is better.

More is better.

Normalized is better.

Other features for

fine-grain.



Sample many patches

Sample the patches in the image. Densely 256 K 2**18.

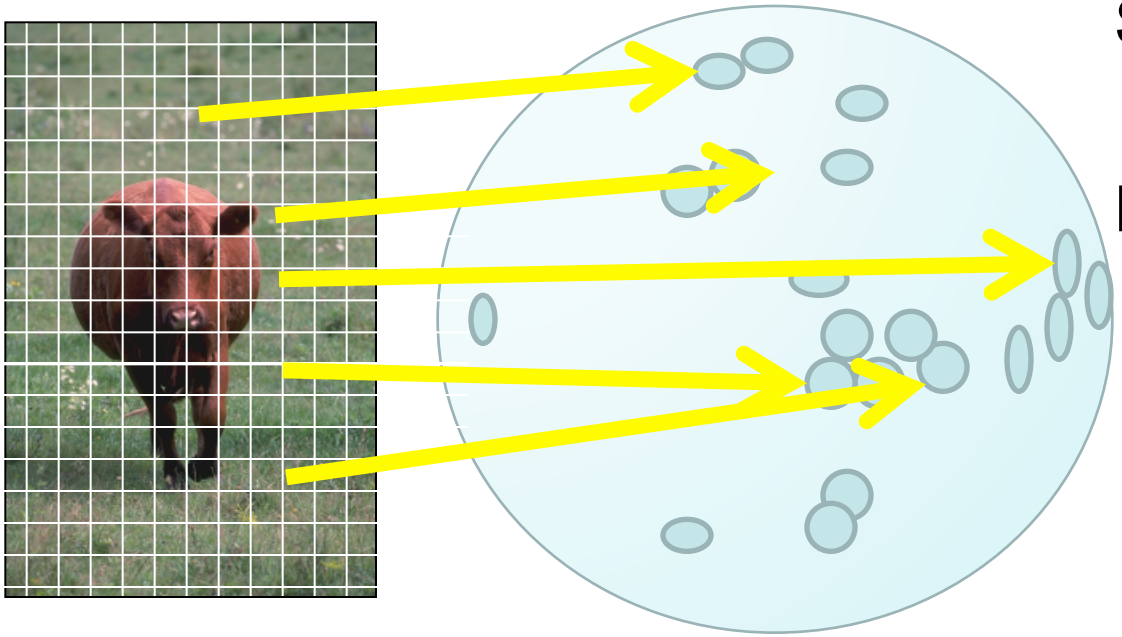
Salient-sampling 1 K 2**10.

Salience is good, dense better.

Salient is specific

and memory.

Dense is compute efficient.

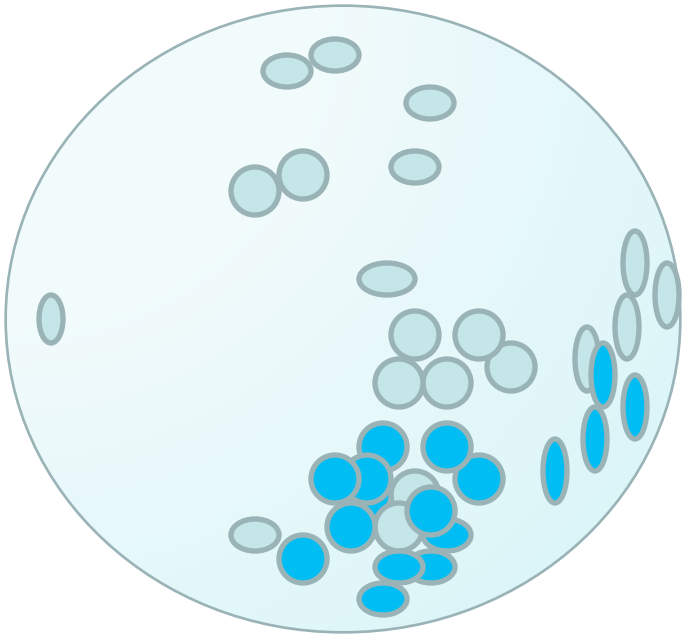


Tinne Tuytelaers discusses dense saliency.

Sample many images

Sample the images in the world: the learning set.

Learn all relevant variety and irrelevant variations in patches not covered by the invariance of features.



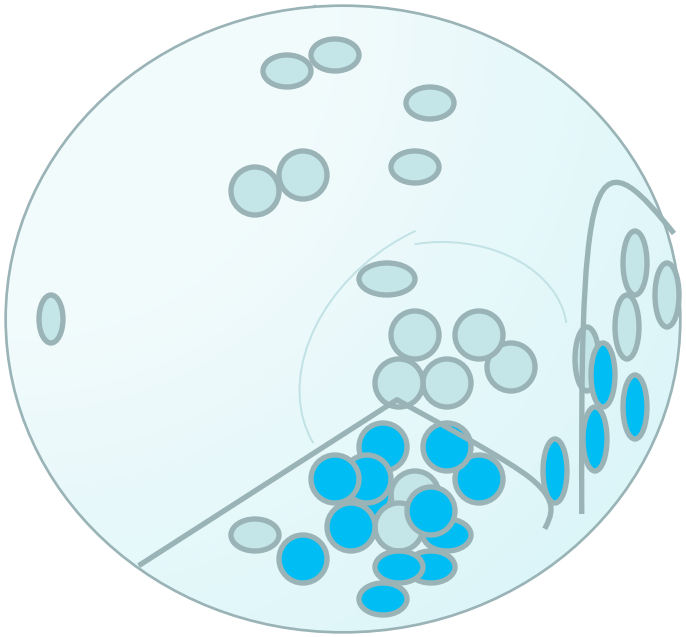
Form a dictionary of words

Form regions in feature space.

400 – 4,000 – 400,000 words, more words record more detail.

Words may cover several types of patches.

Similar patches may emerge distant.



Count words per image

Retain the word boundaries.

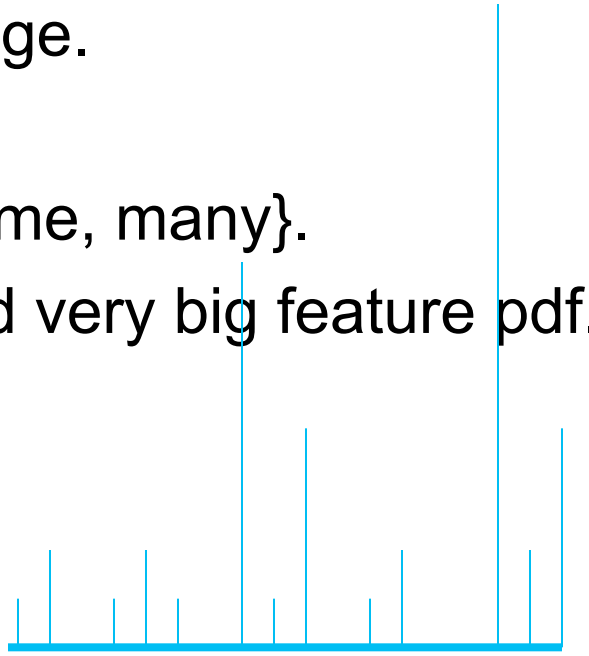
Fill the histogram of words per training image.

Counting words = the area in the image.

A log-scale would do? {none, rare, few, some, many}.

The word-histogram = importance-sampled very big feature pdf.

Patch counts are highly correlated.



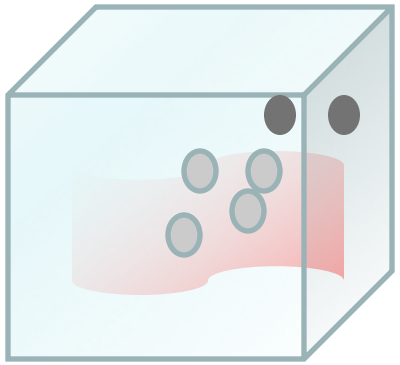
Learn histogram similarity

Learn the word-count histogram **V**j = {t1, t2, … tN} distinction sorted per type in the image learning set.

Similarity between **V**q and each vector **V**j in the dataset by:

intersection Sq = min (**V**q .and. **V**j)

normalized scalar product = **V**q . **V**j / ||**V**q||.||**V**t||

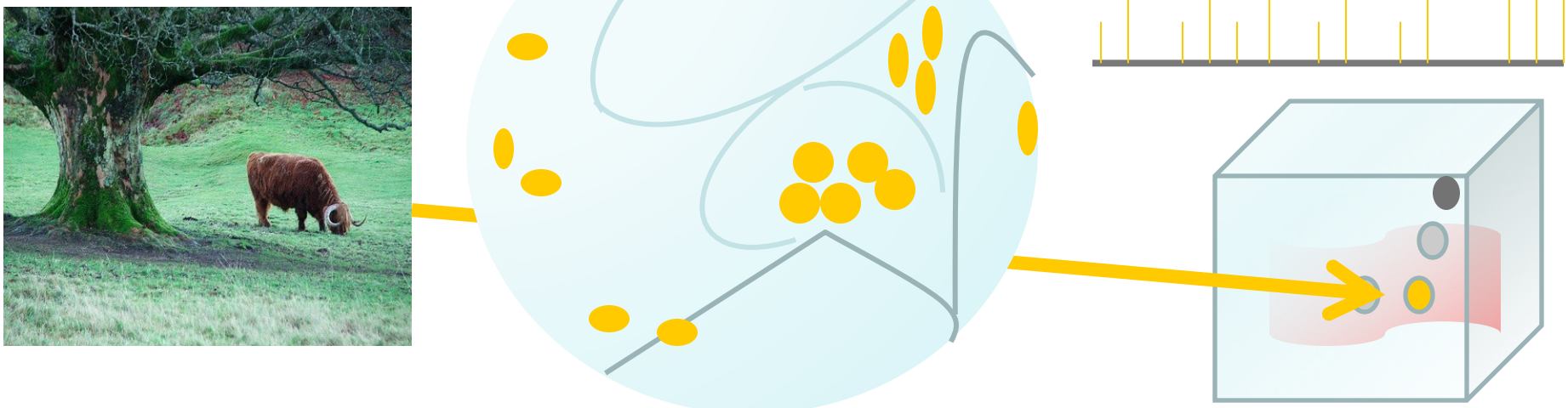


Classify unknown image

Retain the word count discrimination + support vectors

Go from patch to patch > words > counts > discriminate

Then assign to most similar class.



5. On words and codebooks

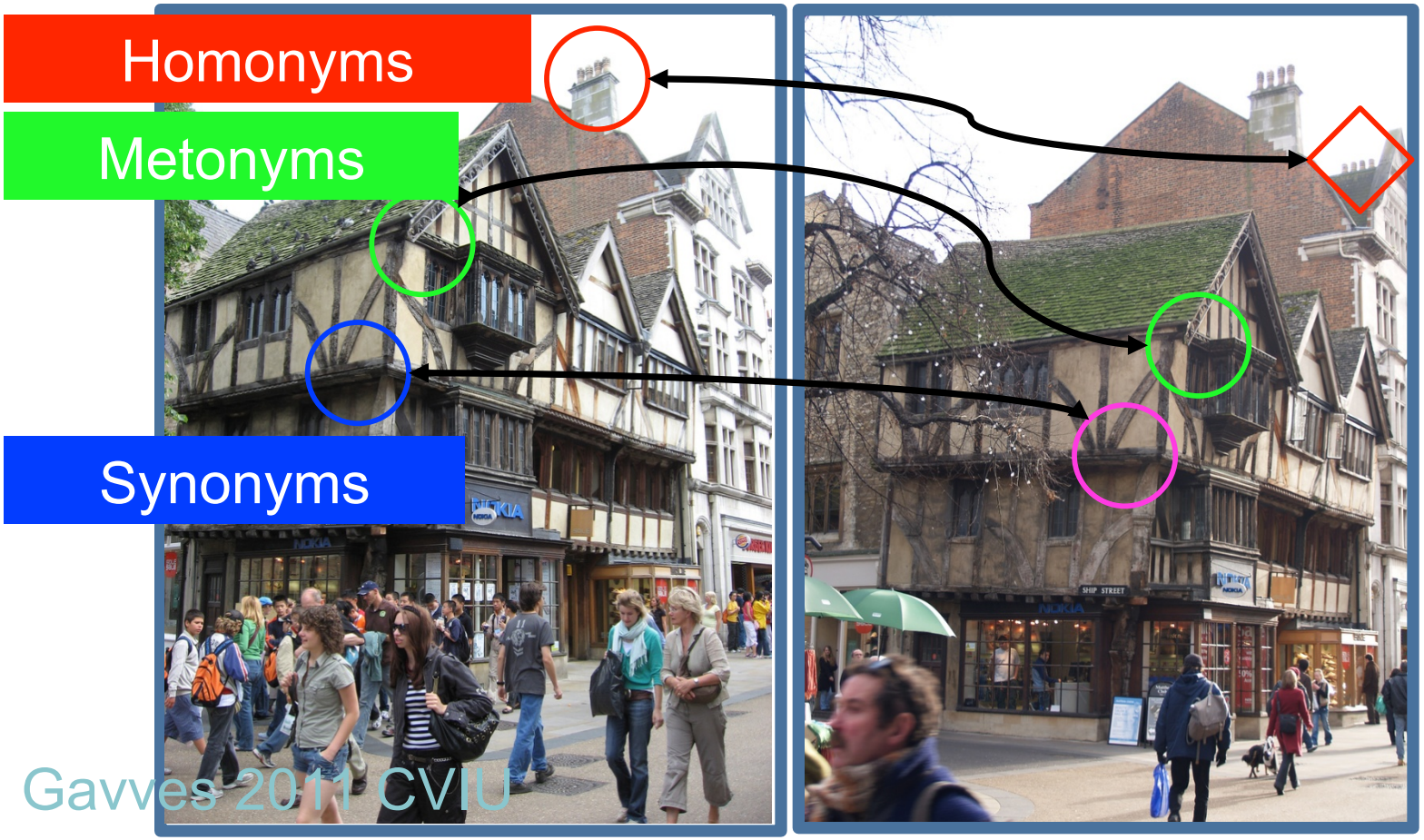
We consider the internal structure of a word, as well as the local structure around the word in feature space.

Where are words in feature space? What is the repertoire of a visual word? Convex reduced codebooks.

Knowledge of the distribution within the words delivers two new representations: Fisher and VLAD.

Synonyms in codebooks

For a better understanding of feature-space:
where are similar words in SIFT-space?



Gavves 2011 CVIU

Synonyms in codebooks

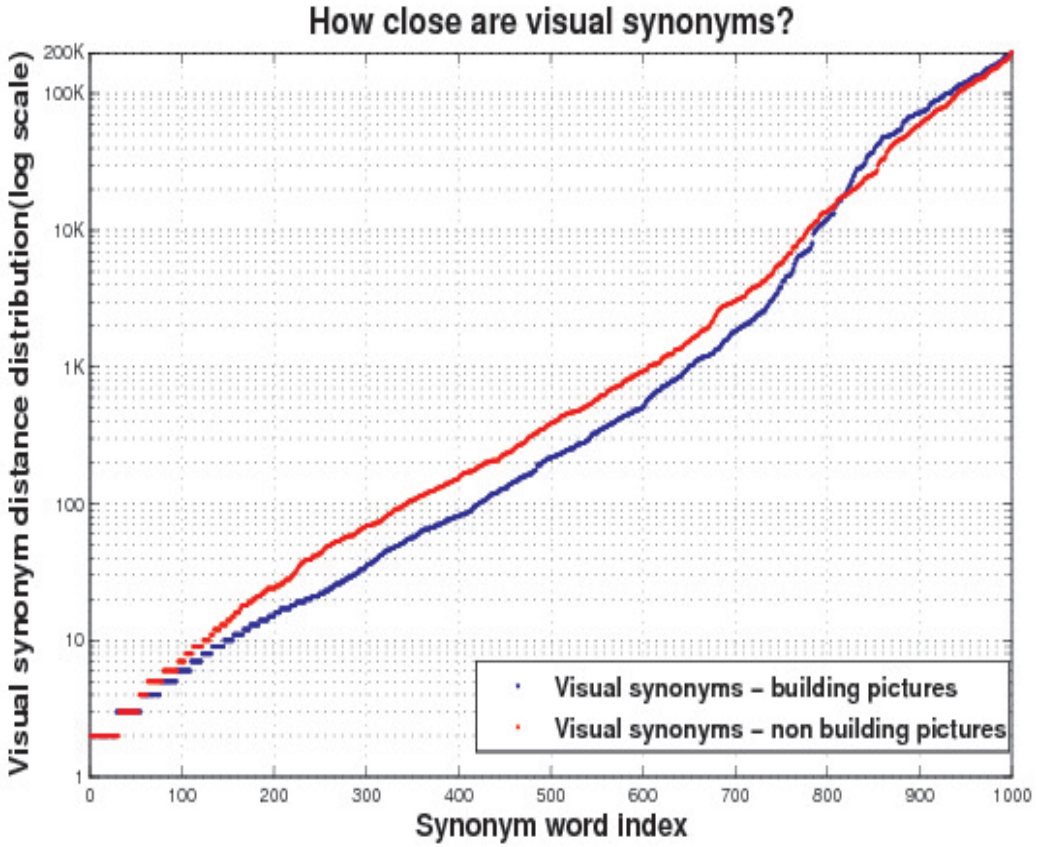
Word 1

Word 2

Gavves 2011 CVIU

Synonyms in codebooks

How close are synonyms?



Visual synonyms are not close.

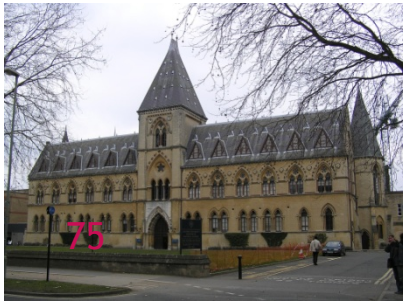
(Compare words in language as observed by De Sausure.)

Study the redundancy, external & internal structure of words!

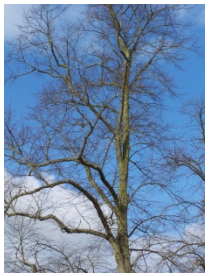
Gavves 2011 CVIU

Redundancy in codebooks

Queries



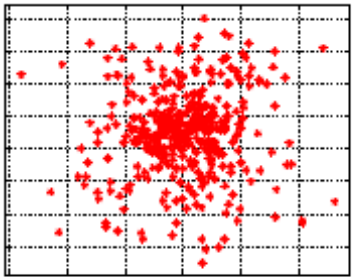
90% Convex reduced, same result.



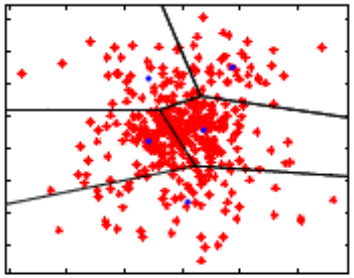
External & Internal Structure

External structure

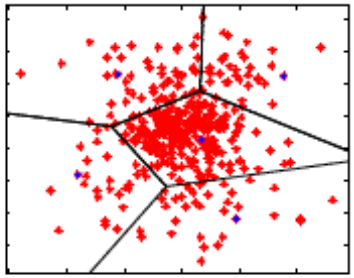
1. Actual words boundaries pretty arbitrary.
2. Use multiple neighbor codes: soft assignment.



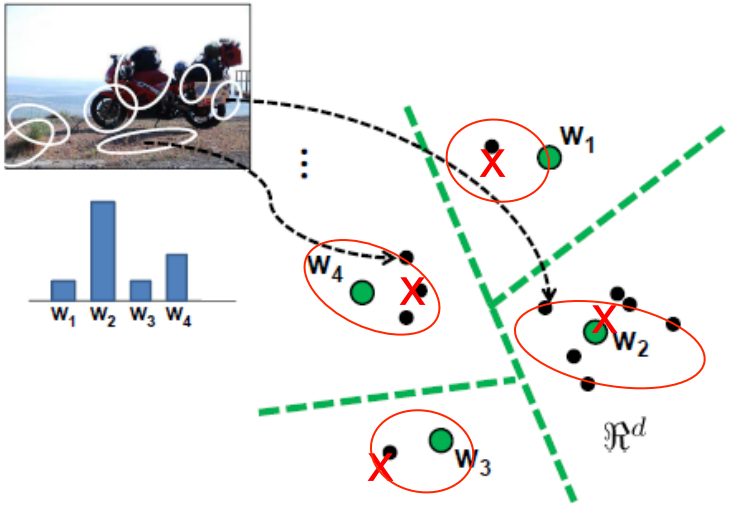
(a) Histogram



(b) K-means



(c) Radius-based



Internal structure

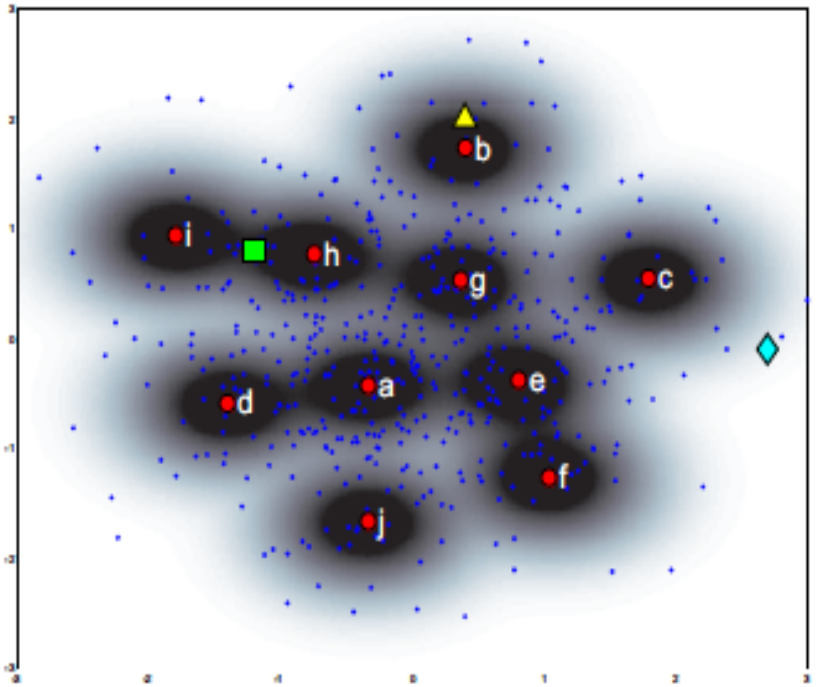
1. Internal structure of words ignored by (hard or soft) counting.
2. Use other statistics > Mean subtract: VLAD / GMM: Fisher

Figure credit: Van Gemert & Grauman

External: soft word assignment

Assign to more than 1 word by weights

Code word uncertainty:

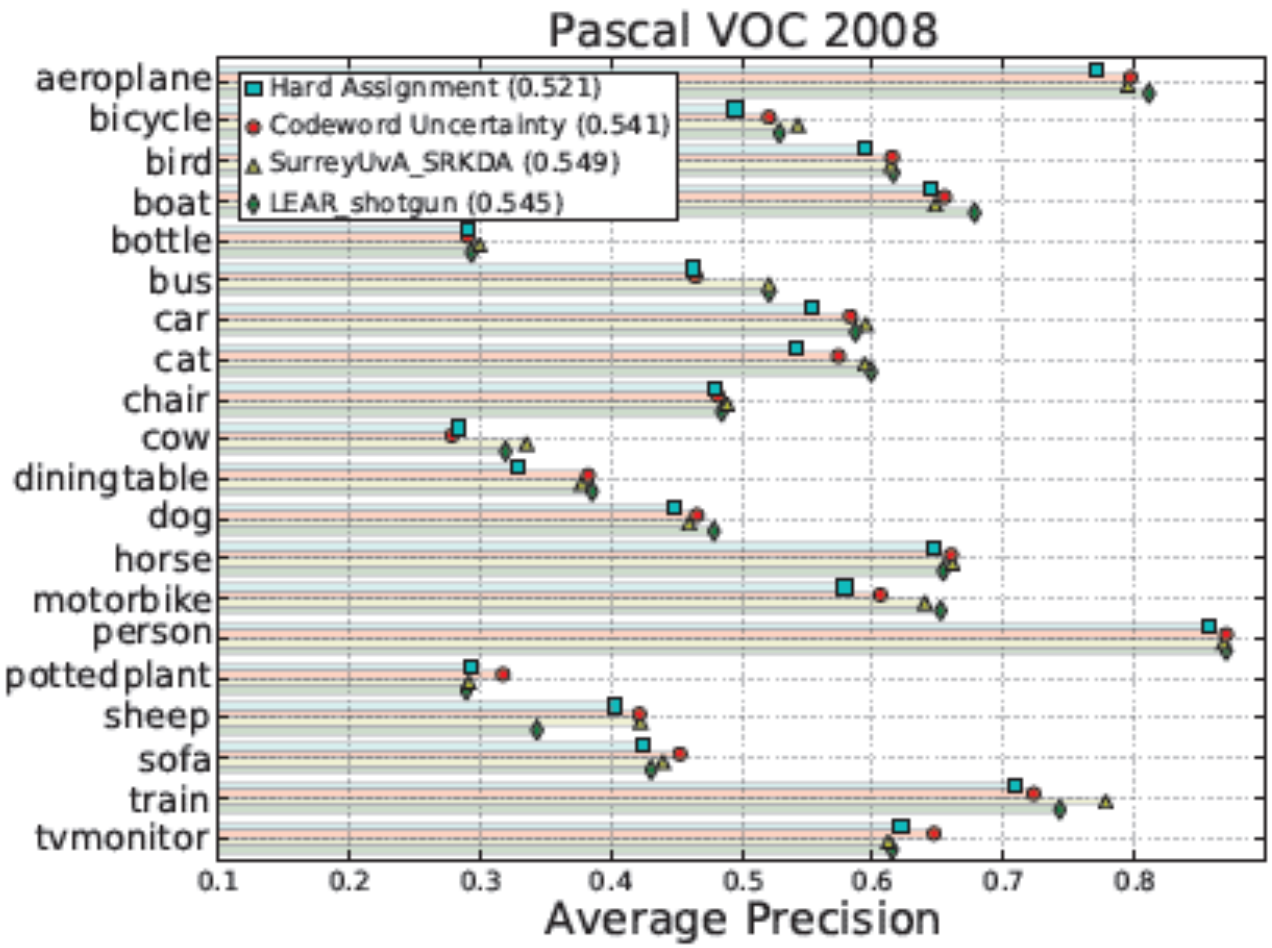


$$\mathrm{UNC}(w) = \frac{1}{n}\sum_{i=1}^{n}\frac{K_\sigma\left(D\left(w,r_i\right)\right)}{\sum_{j=1}^{|V|}K_\sigma\left(D(v_j,r_i)\right)},$$

Van Gemert PAMI 2010

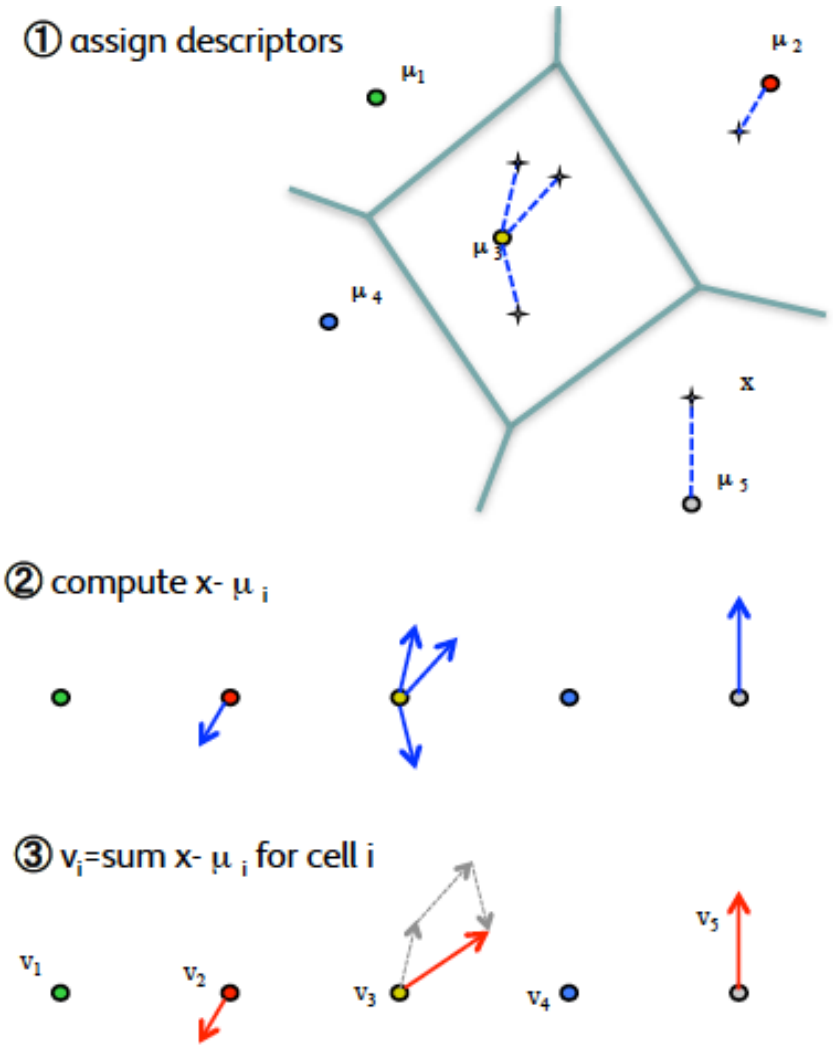
External: soft word assignment

Smaller codebooks & improve performance a few percentage.



Van Gemert PAMI 2010

Internal: VLAD



Jegou CVPR 2010

Hard assignment to nearest word.

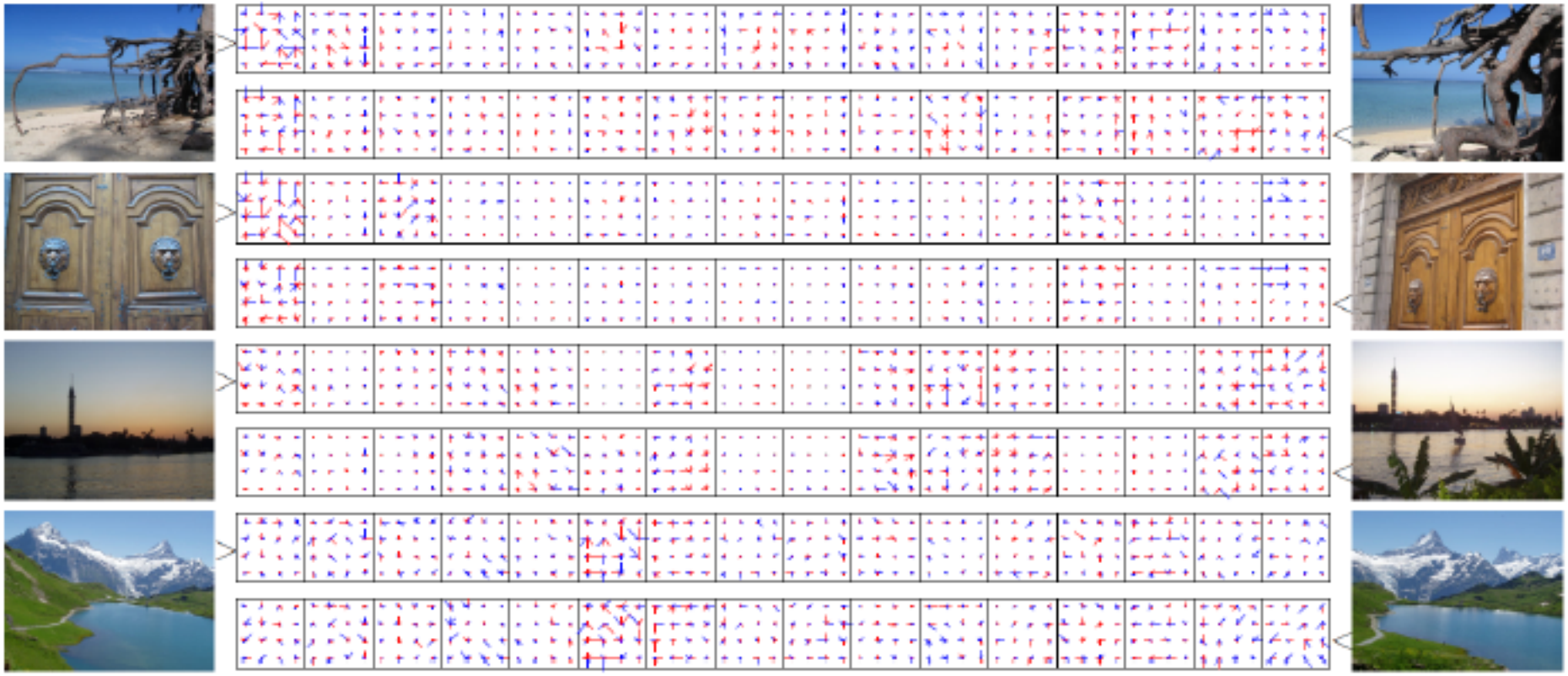
Subtract learned mean for differential coding efficiency.

Rather than count, sum all code residues with in a word.

Concatenate all word sums and l2-normalize.

Memory efficient coding.

Internal: VLAD



Red = residual of **v** for 16 words of 128 dimensions each.

Jegou CVPR 2010

Internal: Fisher vector

Score G of sample X given GMM-likelihood u with param λ:

$$G_\lambda^X = \nabla_\lambda \log u_\lambda(X)$$

Hessian reveals differential GMM-parameters for the sample.

Form the Fisher matrix to measure …

$$F_\lambda = E_{x \sim u_\lambda}\left[\nabla_\lambda \log u_\lambda(x) \nabla_\lambda \log u_\lambda(x)'\right]$$

… similarity between samples by the (inversed) Fisher kernel:

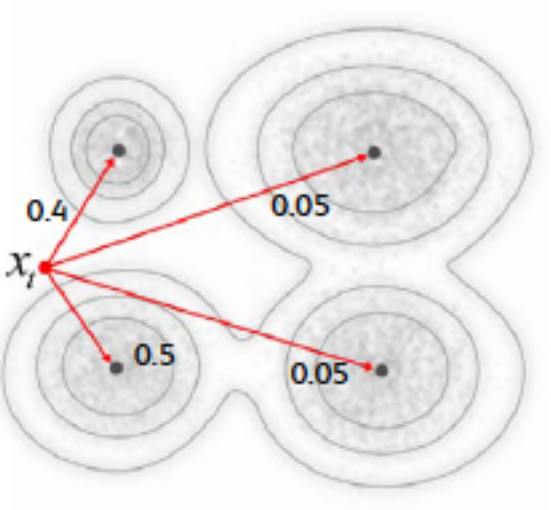
$$K(X,Y) = G_\lambda^{X\prime} F_\lambda^{-1} G_\lambda^Y$$

$$\mathcal{G}_\lambda^X = L_\lambda G_\lambda^X \qquad G_\lambda^X = \frac{1}{T}\sum_{t=1}^{T} \nabla_\lambda \log u_\lambda(x_t)$$

This transform spreads out all encodings for better distinction.

Perronnin CVPR 2007, ECCV 2010

Internal: Fisher vector



Compare N words to 2DN for D feature dimensions.

Perronnin CVPR 2007, ECCV 2010

Internal: Fisher vector

Fisher vector with diagonal covariance matrix:

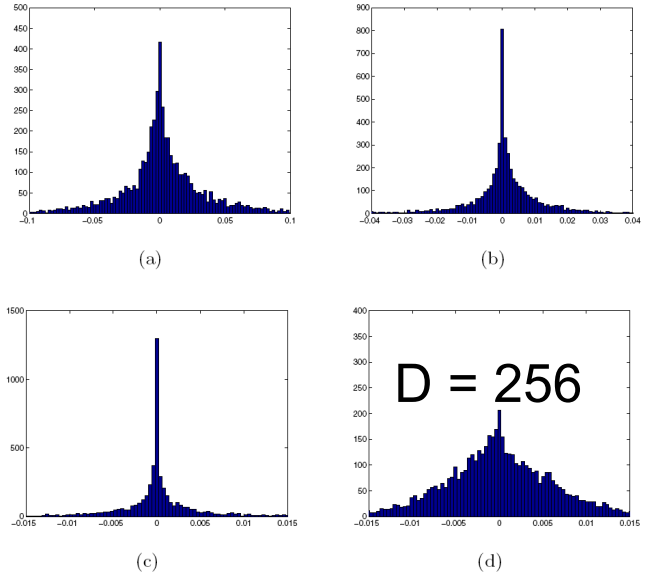
5% better than BoW, smaller D of codebooks needed.

Improved by using the L2 – norm.

Improved by using a power – norm.

Improved by spatial pyramids.

Perronnin CVPR 2007, ECCV 2010



BoW, the story continues

1. It starts with features
   - large dimensionality for omni-potency.
   - powerful normalization & invariance.
2. Two stage classifier to reduce complexity
   - the one faces world complexity.
   - the other one face class confusion.
3. Patterns in feature space produce subtle methods
   - task-specific computational methods
   - more examples > more subtle methods