# Color in Image & Video Processing Applications


**Theo Gevers**
**Joost van de Weijer**

# Overview

## *PART I (low-level) – Joost van de Weijer*

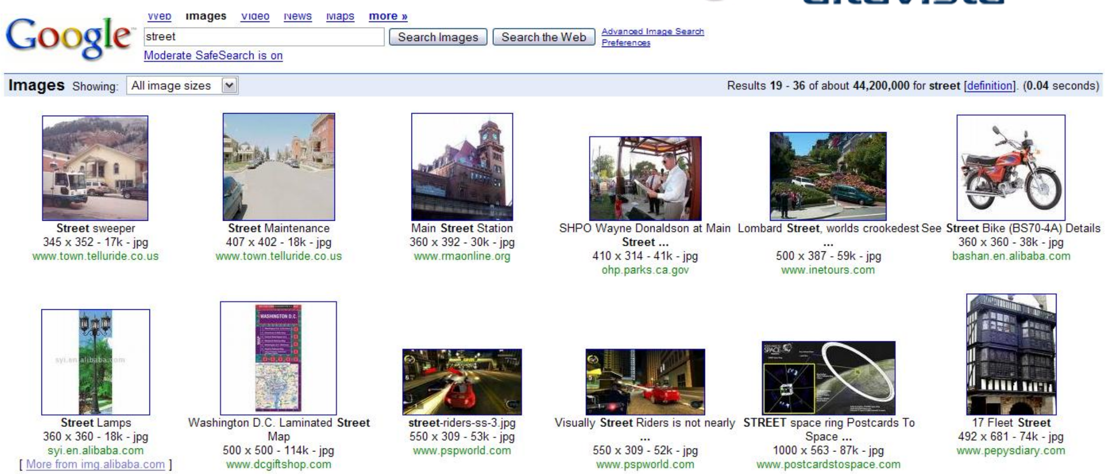
1. ***Reflection Models***
   - Dichromatic reflection model
2. ***Photometric/Color Invariance***
   - At the pixel
   - Instability handling
   - Color differential structure
3. ***Color Constancy***
   - Low-level
   - High-level
4. ***Saliency and Color Boosting***
   - Itti and Koch model
   - Color boosted

## *PART II (higher Level) – Theo Gevers*

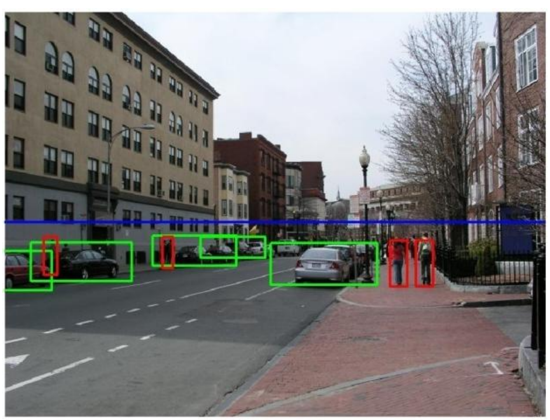
1. ***Interest point detection***
   - Harris Laplace
   - Color boosted
2. ***Descriptors***
   - SIFT
   - Extension to color
3. ***Object recognition (VOC/TRECVid)***
   - Dense and point sampling
   - Code book generation
   - Results
4. ***Applications***
   - Tracking in video
   - Object replacement
   - Emotion recognition
   - Head pose estimation

# Image and Video Retrieval

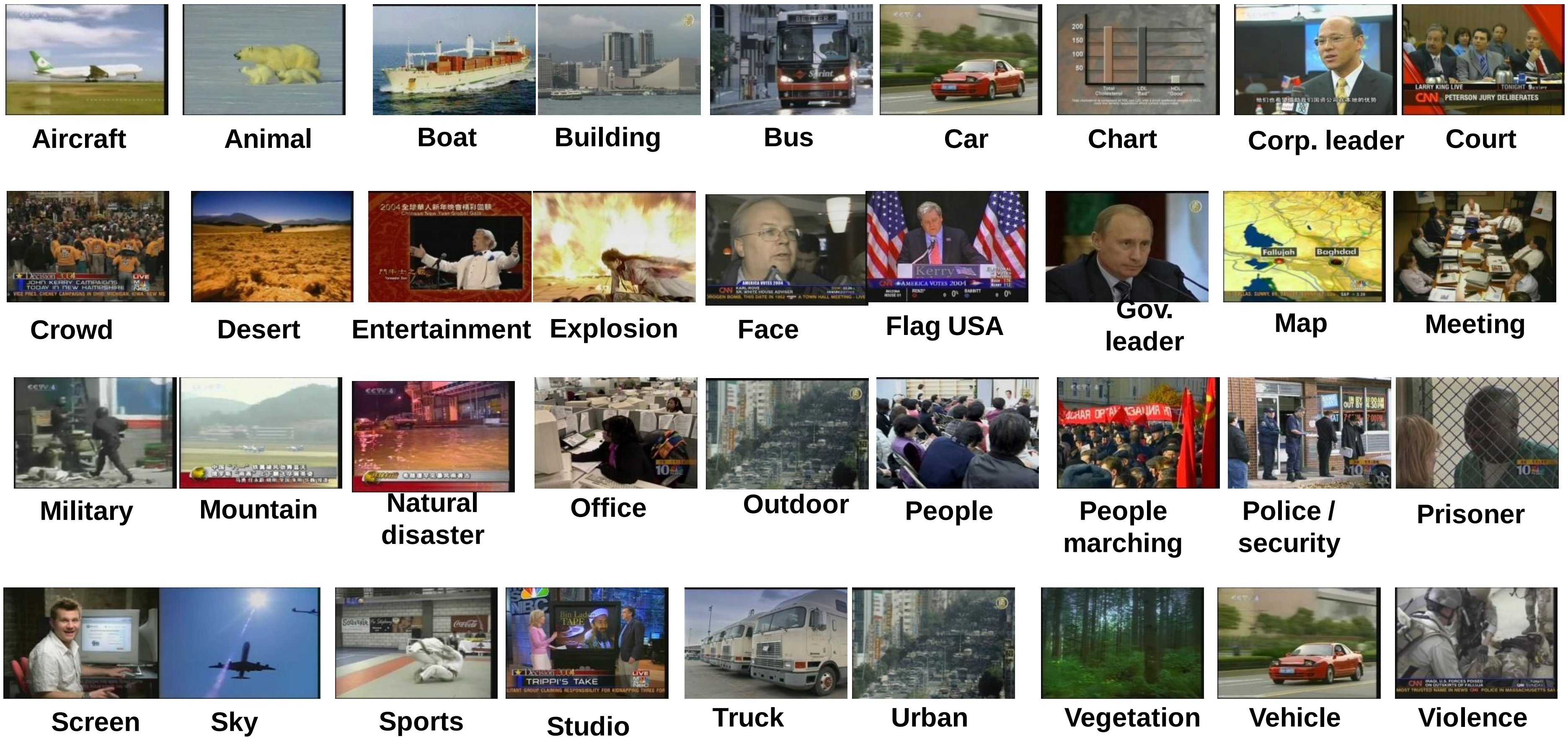
- Online photo and video Retrieval




- Surveillance



# Object/Scene Categories



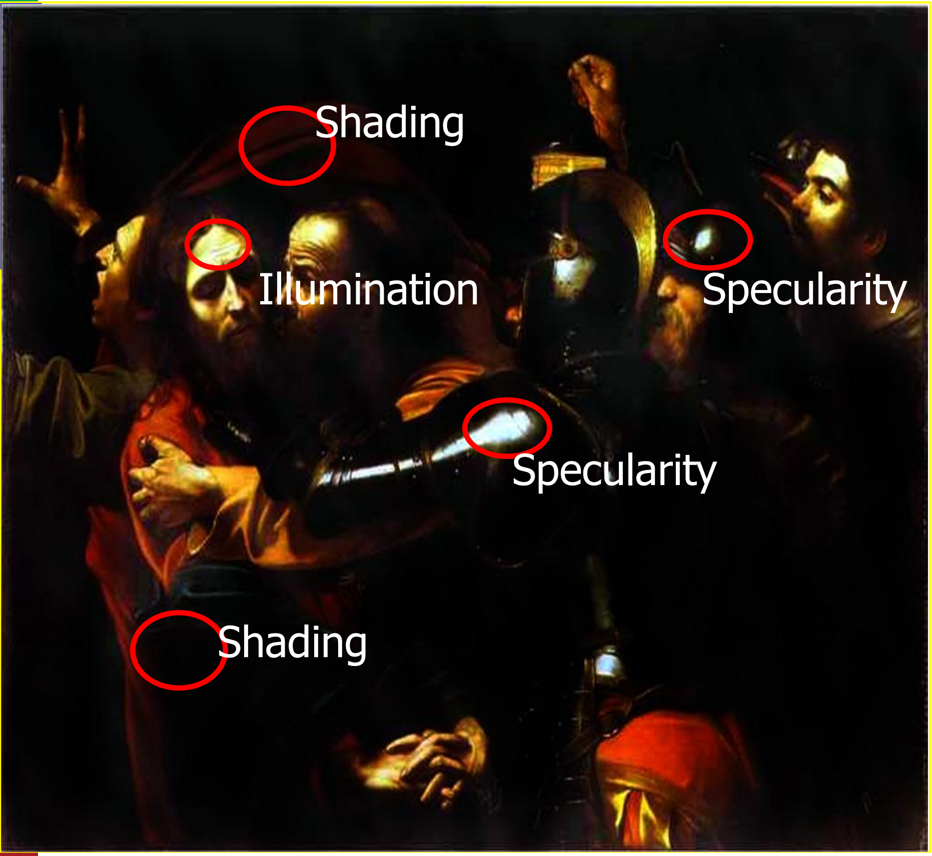
# Video Retrieval

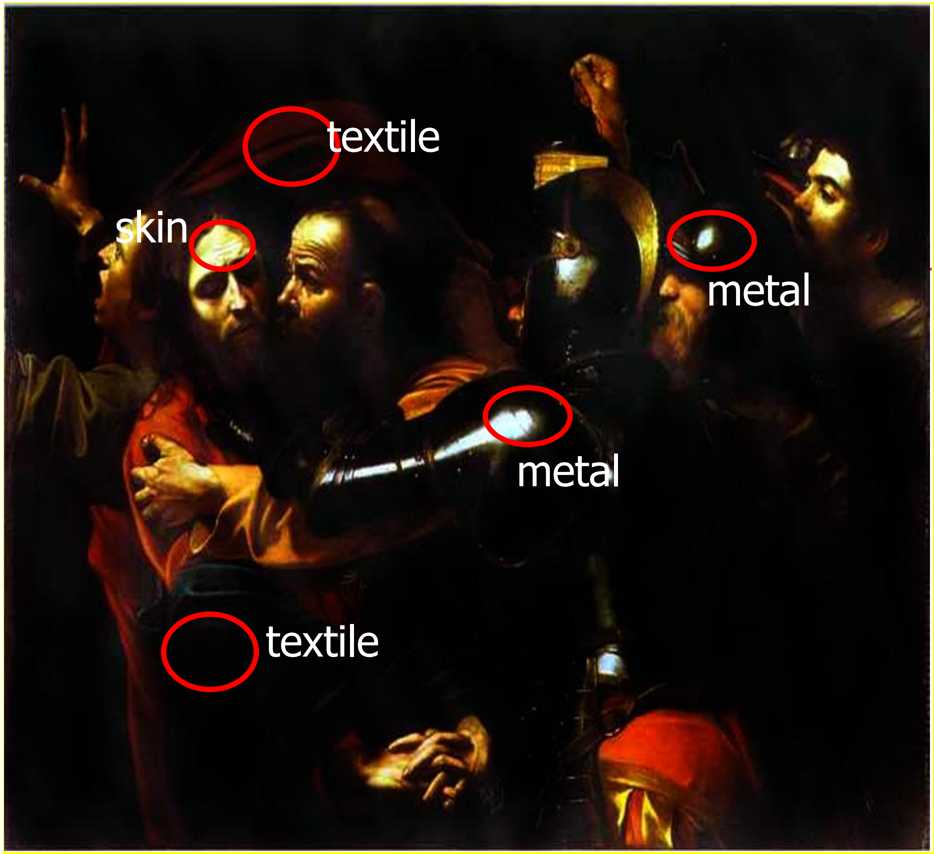
Given a shot from a video…
… is some semantic *concept* present in that shot?

Example concepts:

- Airplane
- Building
- **Car**
- Crowd
- **Desert**
- **Explosion**
- **Outdoor**
- People
- **Vehicle**
- **Violence**

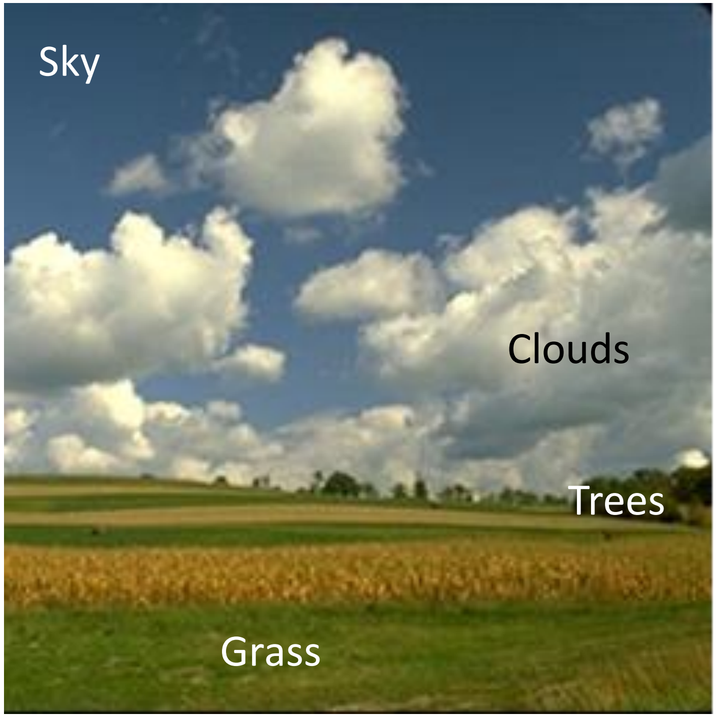
# Image semantics (low-level)



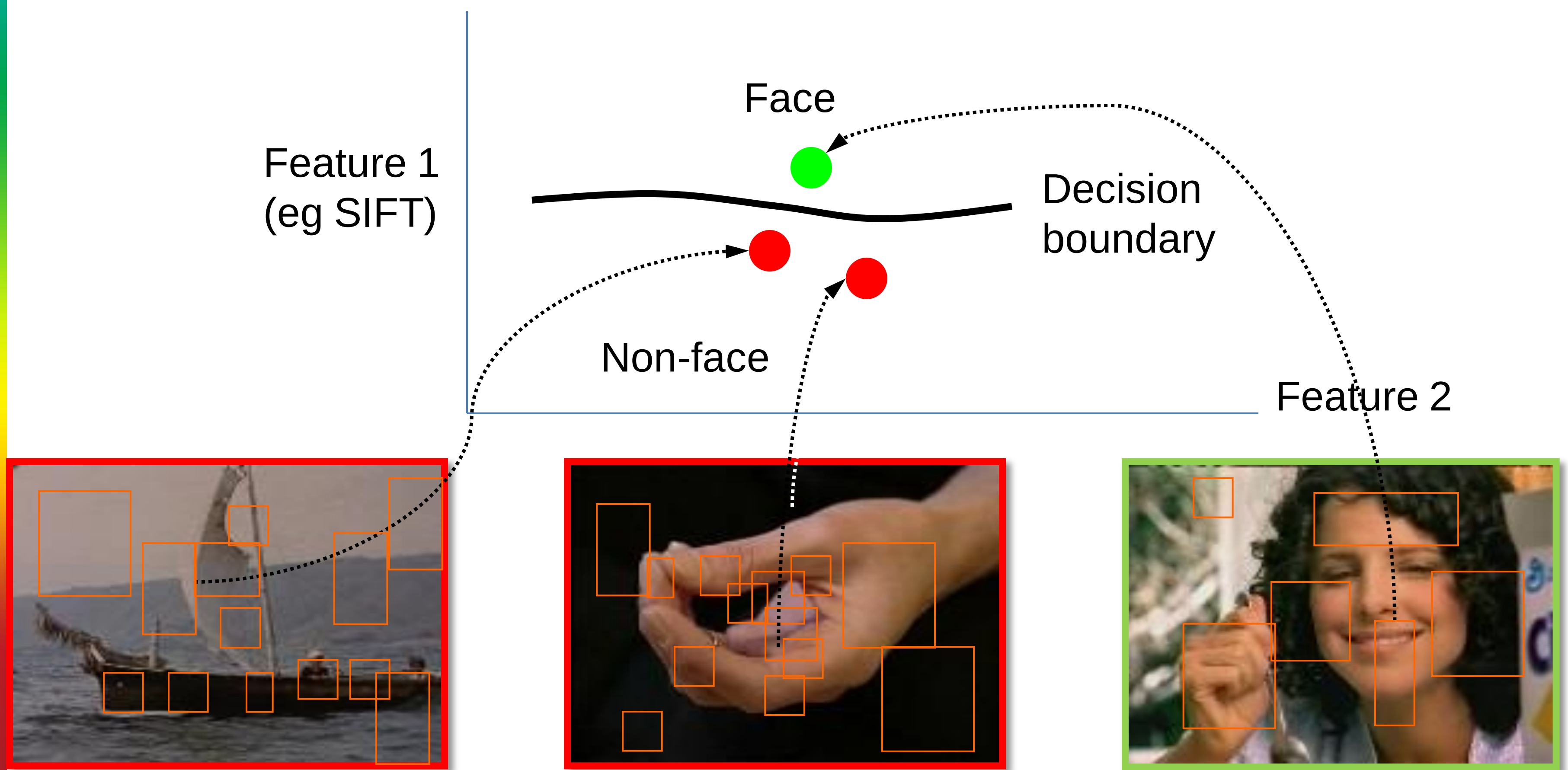
# categorization (high-level)

Outdoor/Landscape/vegatation…

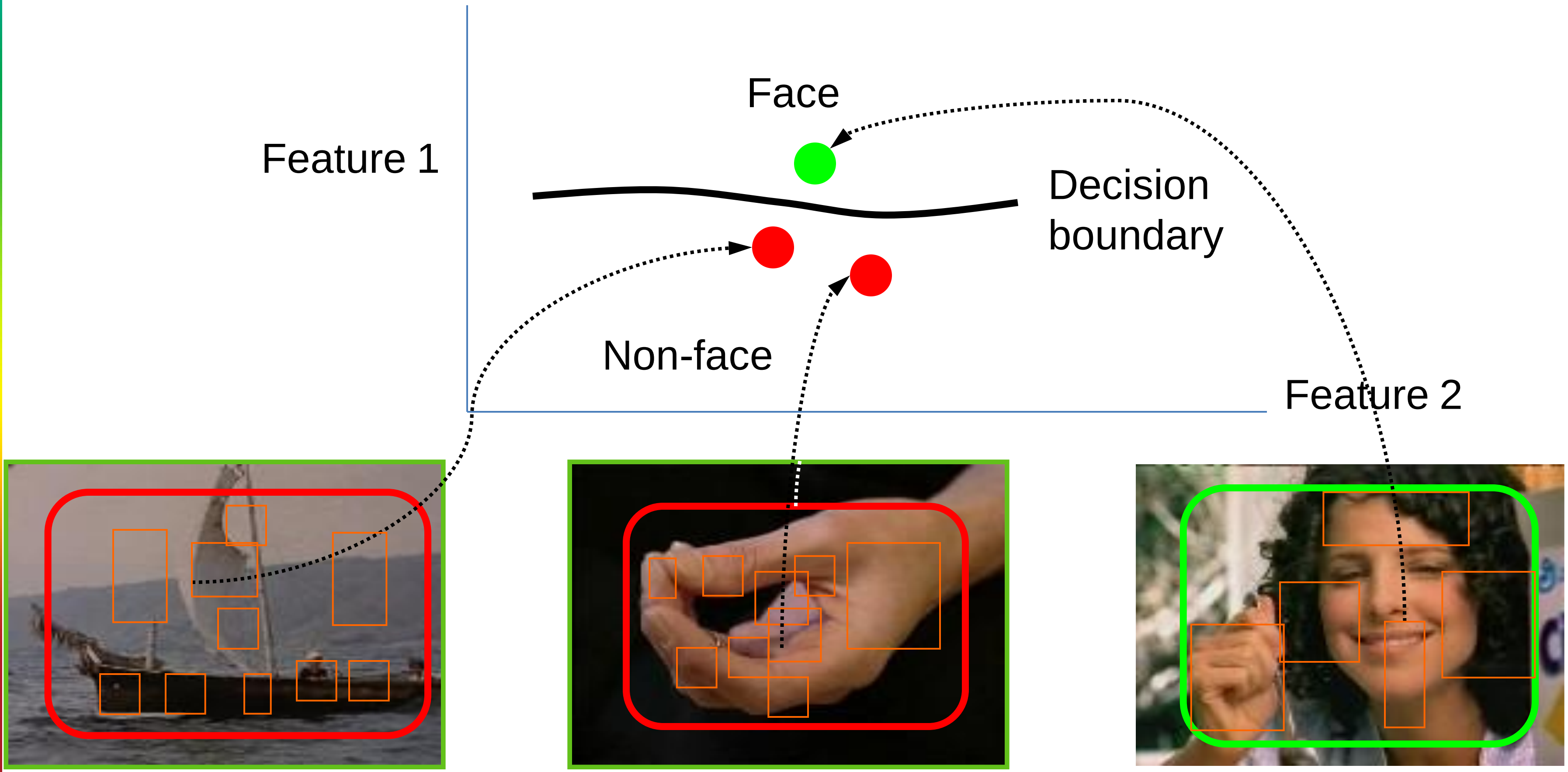



Outdoor/city/street…

# Machine learning (K-NN, SVM)

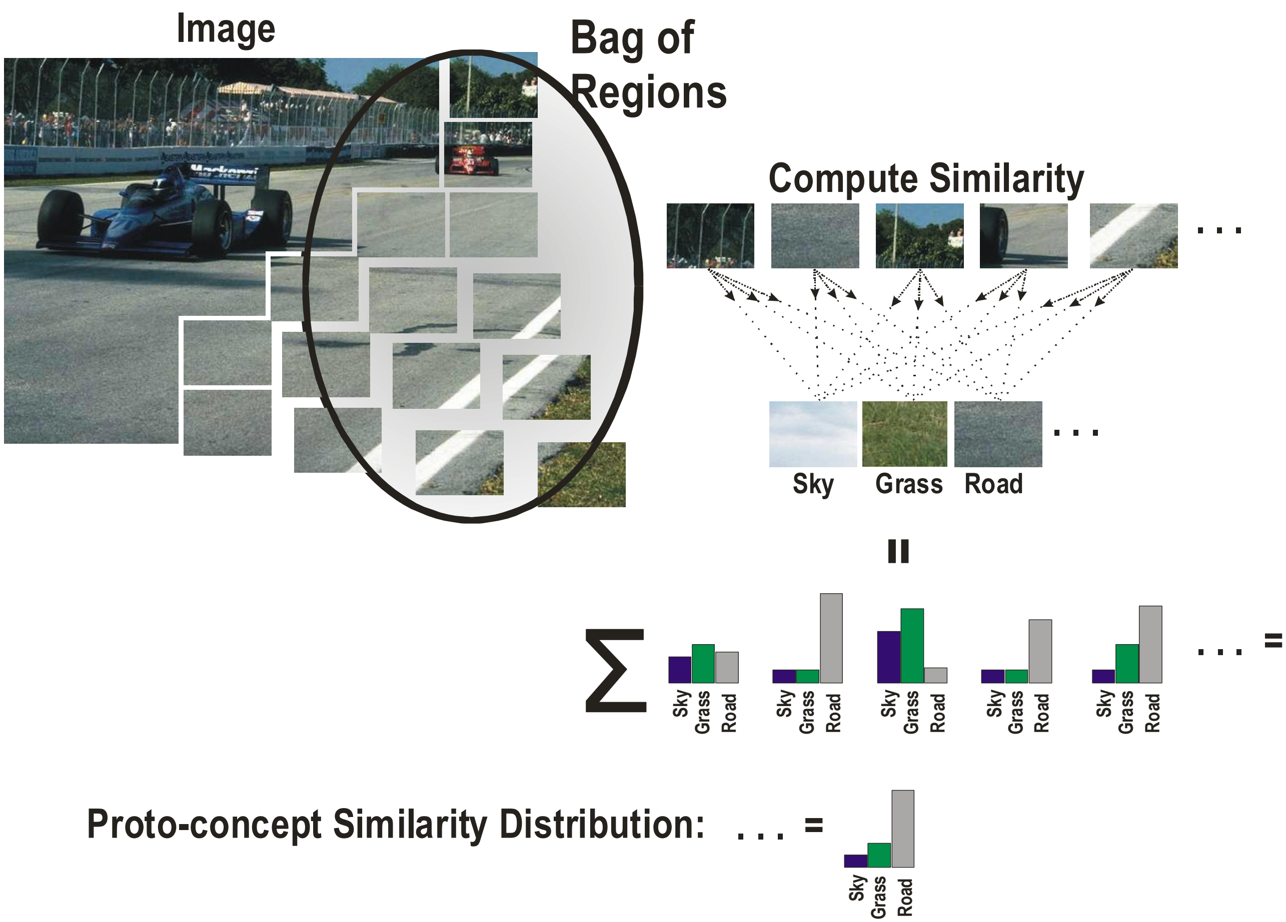


# Machine learning (K-NN, SVM)

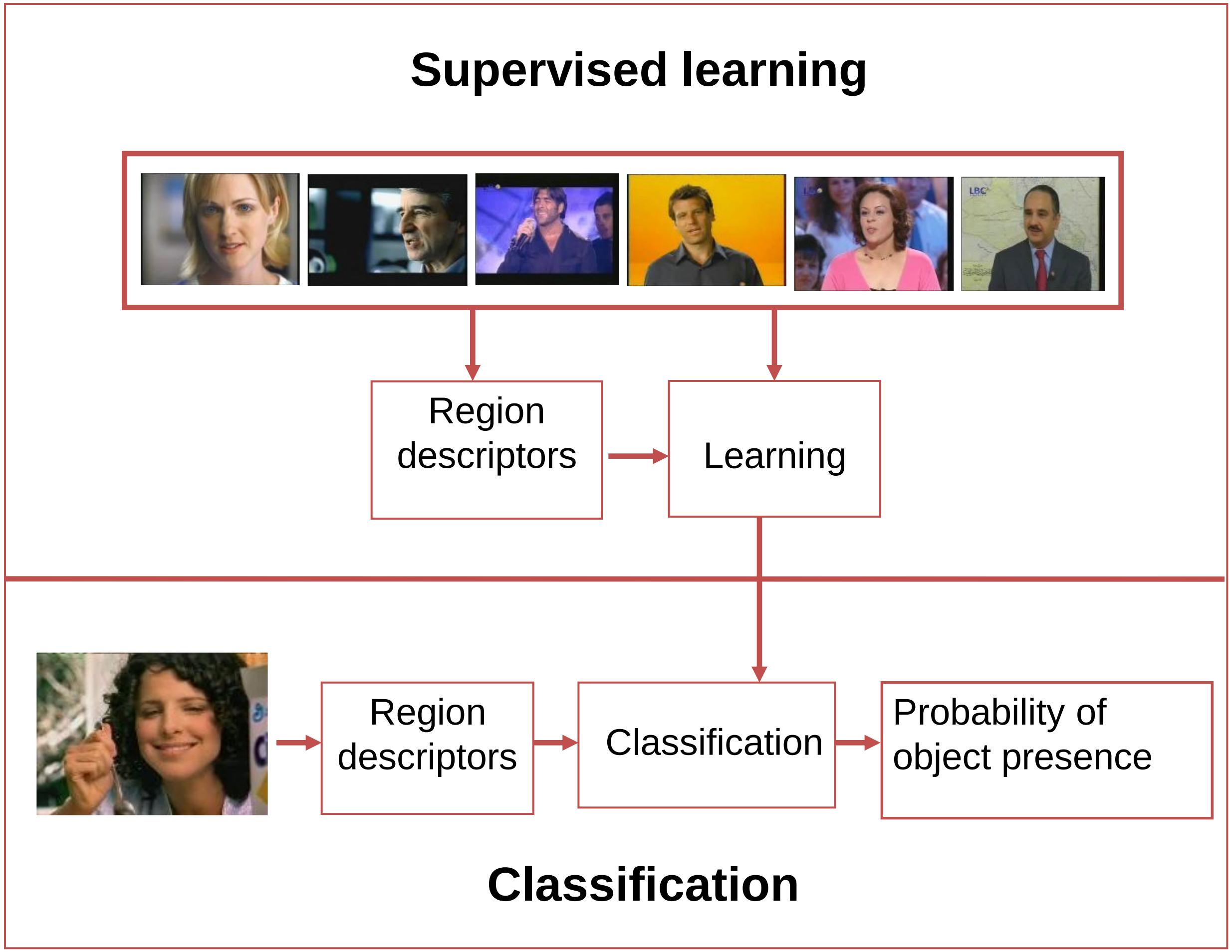


[van Gemert, ECCV08]

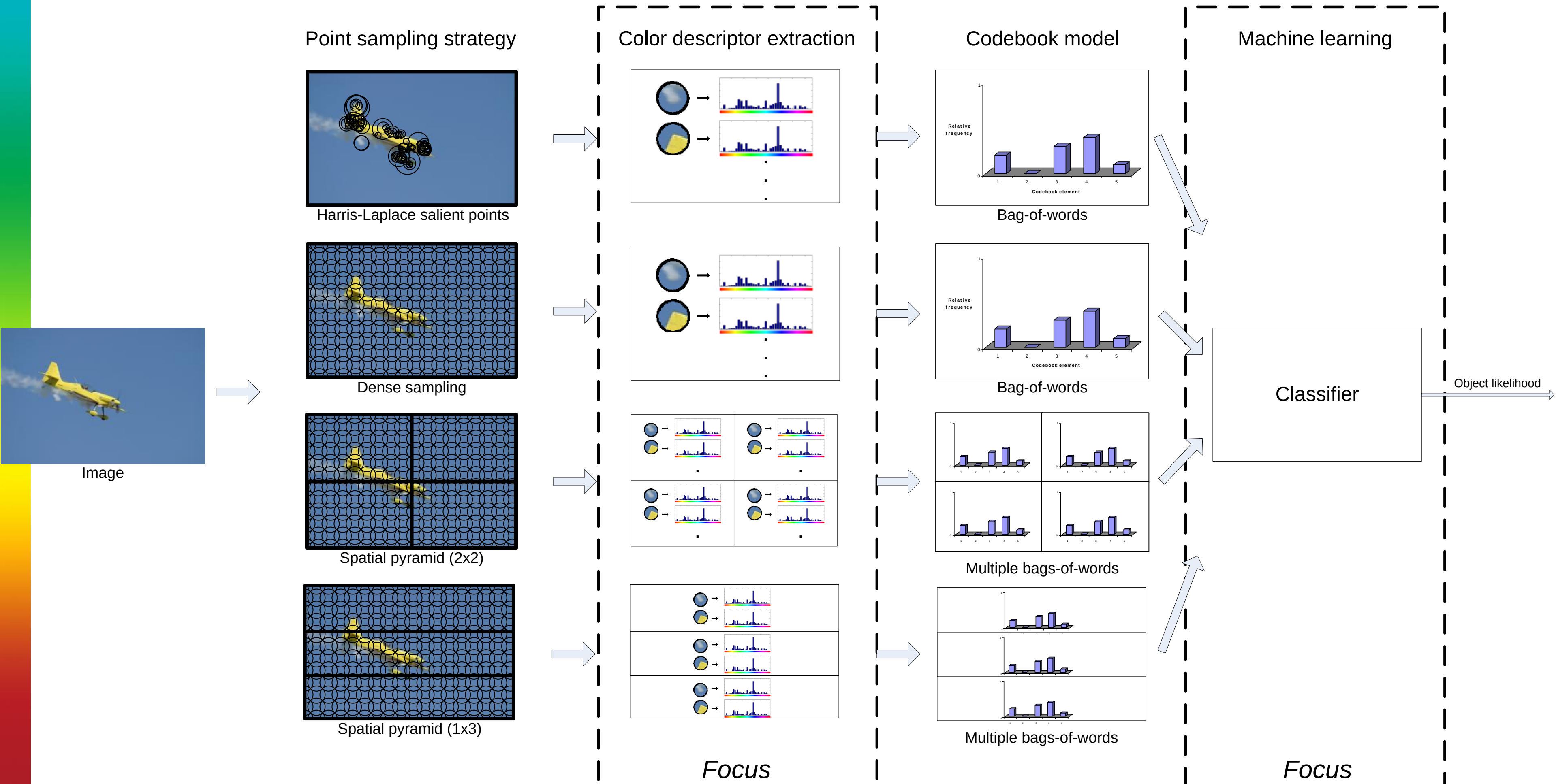
# Video Retrieval – Scene



# Recognition scheme



# Pipeline Overview



# Dataset: image and video retrieval

## TRECVID and PASCAL VOC competition

- 86 hours of video from TRECVID 2005
- Shot segmentation available: 43.907 shots
- Ground truth available from Mediamill Challenge


- The goal of VOC challenge is to recognize objects from a number of visual object classes in realistic scenes

- The twenty object classes are:
  - *Person:* person
  - *Animal:* bird, cat, cow, dog, horse, sheep
  - *Vehicle:* aeroplane, bicycle, boat, bus, car, motorbike, train
  - *Indoor:* bottle, chair, dining table, potted plant, sofa, tv/monitor.

# Overview

## *PART I (low-level)*

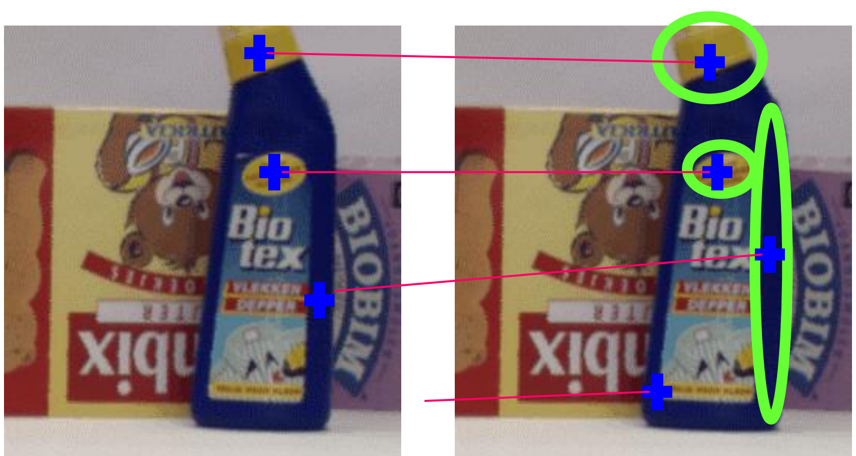
1. ***Reflection Models***
   - Dichromatic reflection model
2. ***Photometric/Color Invariance***
   - At the pixel
   - Instability handling
   - Color differential structure
3. ***Color Constancy***
   - Low-level
   - High-level
4. ***Saliency and Color Boosting***
   - Itti and Koch model
   - Color boosted

## *PART II (higher Level)*

1. ***Interest point detection***
   - Harris Laplace
   - Color boosted
2. ***Descriptors***
   - SIFT
   - Extension to color
3. ***Object recognition (VOC/TRECVid)***
   - Dense and point sampling
   - Code book generation
   - Results
4. ***Applications***
   - Tracking in video
   - Object replacement
   - Emotion recognition
   - Head pose estimation

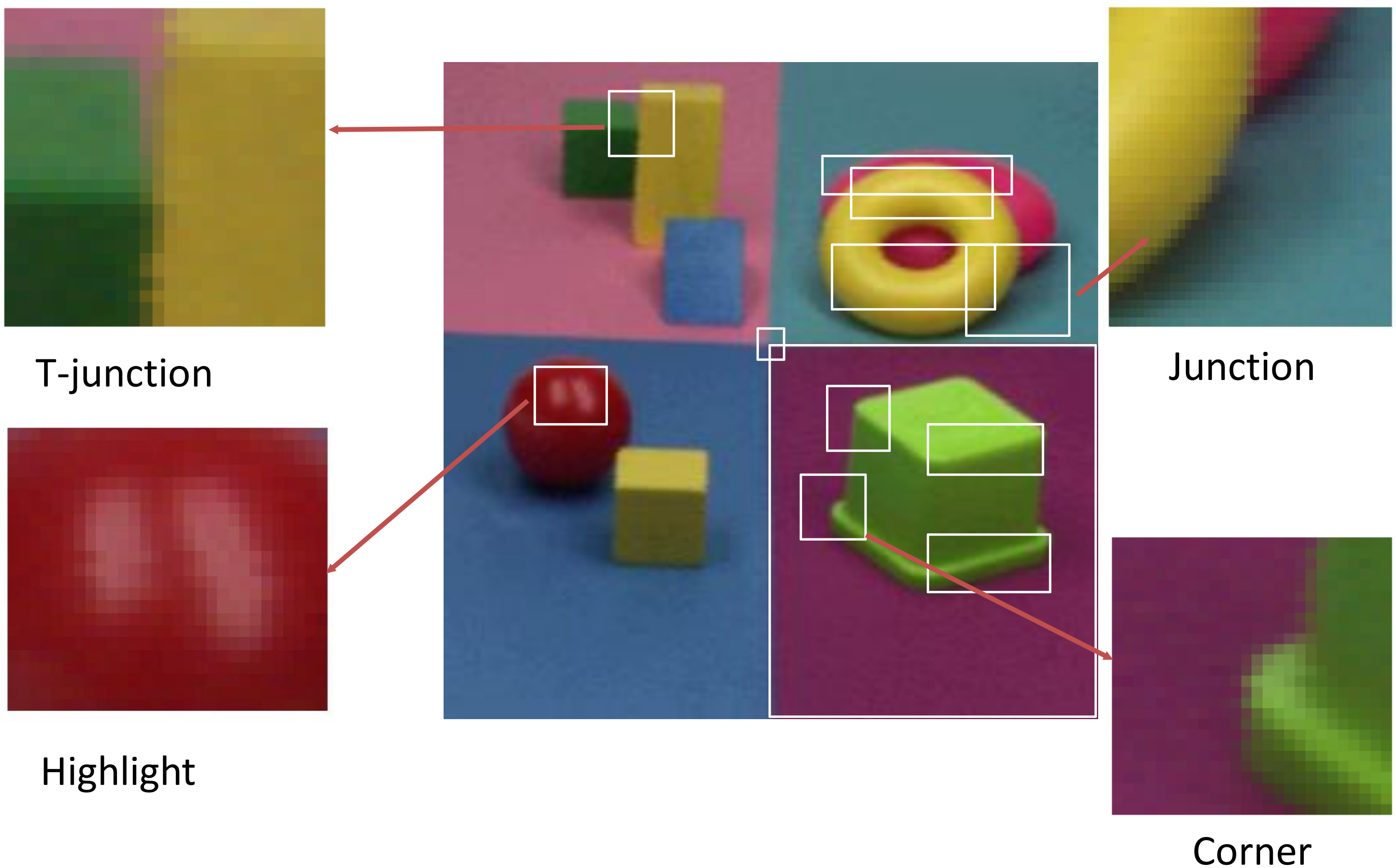
# Local Image Structures: Matching

salient point matching
A. Zissermann, Oxford



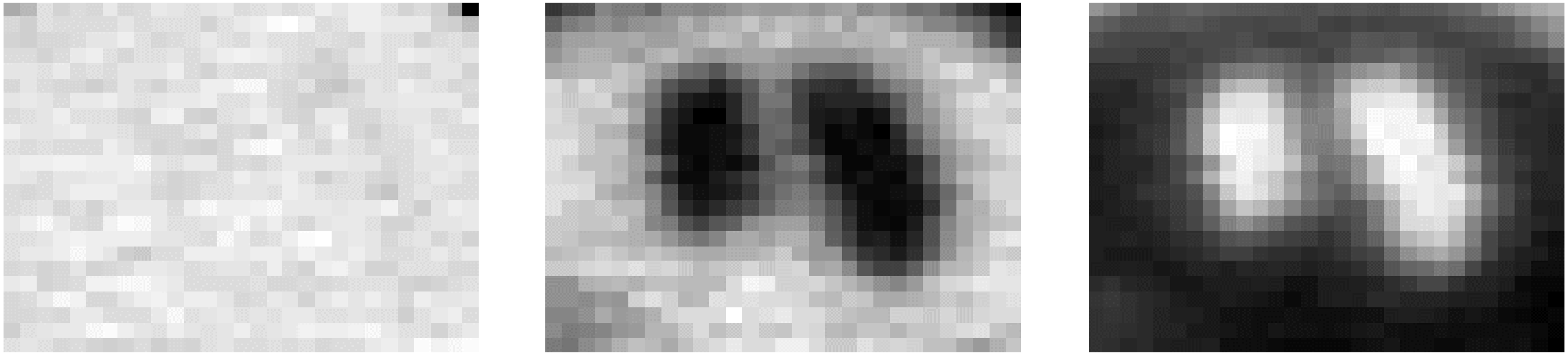

affine local region
L. van Gool, Leuven

# Taxonomy of Image Structures

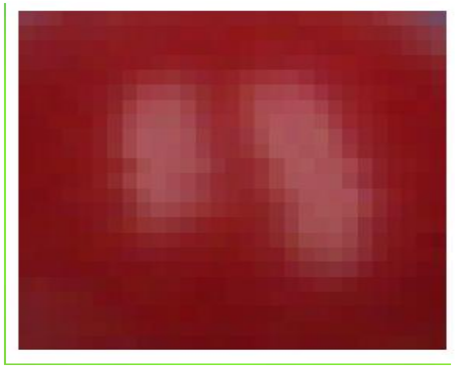


# Local Image Structures: Classification of Highlights

- Properties
  - Hue remains the same
  - Saturation will decrease
  - Brightness will increase



# Local Image Structures: Classification of Highlights



$f_x(\mathbf{p}_X, \mathbf{p}_Y) = 0$ and $f_y(\mathbf{p}_X, \mathbf{p}_Y) = 0$

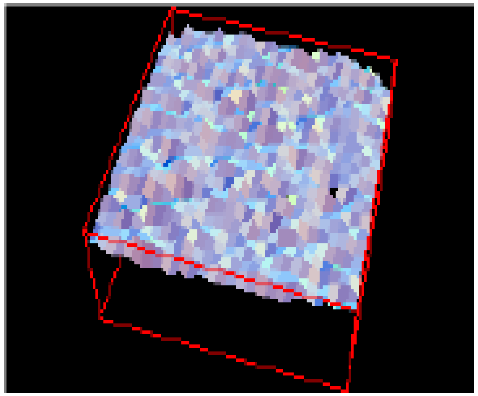
where $d = f_{xx}(\mathbf{p}_X, \mathbf{p}_Y) f_{yy}(\mathbf{p}_X, \mathbf{p}_Y) - [f_{xy}(\mathbf{p}_X, \mathbf{p}_Y)]^2$

then

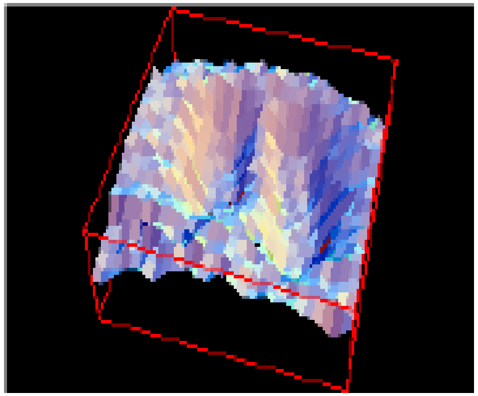
i. $f(\mathbf{p}_X, \mathbf{p}_Y)$ is a relative minimum if $d > 0$ and $f_{xx}(\mathbf{p}_X, \mathbf{p}_Y) > 0$

ii. $f(\mathbf{p}_X, \mathbf{p}_Y)$ is a relative maximum if $d > 0$ and $f_{xx}(\mathbf{p}_X, \mathbf{p}_Y) < 0$

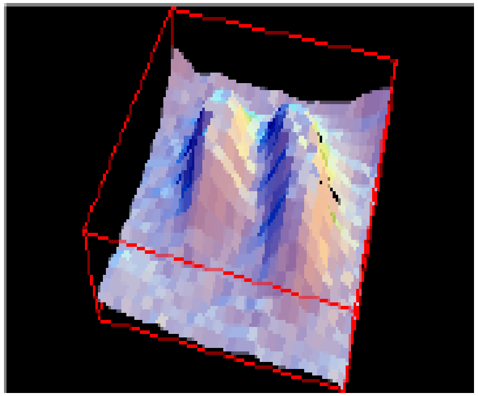
iii. $f(\mathbf{p}_X, \mathbf{p}_Y)$ is a saddle point if $d < 0$



H



S



I

# N-jet: Description of Local Image Patches

Taylor series(second order):

$$f(\mathbf{p}_X + x, \mathbf{p}_Y + y) \approx f(\mathbf{p}_X, \mathbf{p}_Y) + xf_x(\mathbf{p}_X, \mathbf{p}_Y) + yf_y(\mathbf{p}_X, \mathbf{p}_Y) +$$

$$\frac{1}{2}x^2 f_{xx}(\mathbf{p}_X, \mathbf{p}_Y) + xyf_{xy}(\mathbf{p}_X, \mathbf{p}_Y) + \frac{1}{2}y^2 f_{yy}(\mathbf{p}_X, \mathbf{p}_Y)$$

Taylor series(second order) gauge coordinates:

$$\begin{pmatrix} f_{ww} & f_{vw} \\ f_{vw} & f_{ww} \end{pmatrix} = \frac{1}{f_x^2 + f_y^2} \begin{pmatrix} f_x^2 f_{yy} - 2 f_x f_y f_{xy} + f_x^2 f_{xx} & (f_y^2 - f_x^2) f_{xy} + f_x f_y (f_{xx} + f_{yy}) \\ (f_y^2 - f_x^2) f_{xy} + f_x f_y (f_{xx} - f_{yy}) & f_x^2 f_{xx} - 2 f_x f_y f_{xy} + f_y^2 f_{xx} \end{pmatrix}$$

# Hessian and Curvature Gauge

Hessian :

$$H = \begin{pmatrix} f_{xx} & f_{yy} \\ f_{xy} & f_{yx} \end{pmatrix}$$

eigenvalues are $f_{xx} - f_{yy} \pm \sqrt{(f_{xx} + f_{yy})^2 + 4 f_{xy}^2}$

eigenvalues in gauge coordinates are

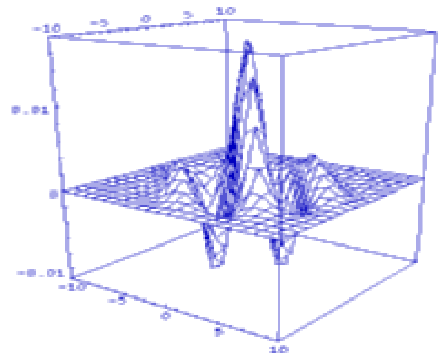
$$\kappa_1 = f_{xx} + f_{yy} - \sqrt{(f_{xx} + f_{yy})^2 + 4 f_{xy}^2}$$

$$\kappa_2 = f_{xx} + f_{yy} - \sqrt{(f_{xx} + f_{yy})^2 + 4 f_{xy}^2}$$

# Higher-order N-Jet: Blobs, Bars, Ridges, Saddle Points etc

- Dark blob on bright background : $f_w \approx 0, \kappa_1 \approx \kappa_2 > 0$
- Dark bar on bright background: $f_w \approx 0, \kappa_1 \approx 0, \kappa_2 > 0$
- Bright blob on dark background: $f_w \approx 0, \kappa_1 \approx \kappa_2 < 0$
- Bright bar on dark background: $f_w \approx 0, \kappa_1 < 0, \kappa_2 \approx 0$
- Constant patch: $f_w \approx 0, \kappa_1 \approx 0, \kappa_2 \approx 0$
- Saddle point : $f_w \approx 0, \kappa_1 < 0, \kappa_2 > 0$

# Texture: Gabor Filters

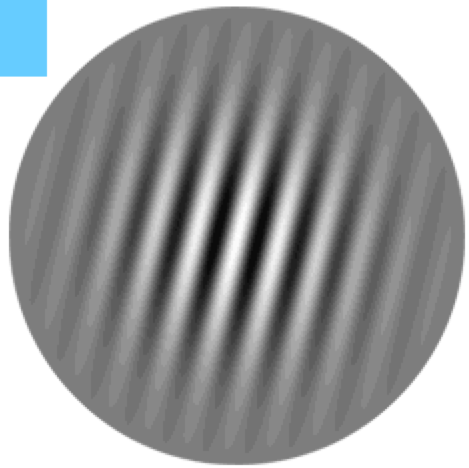


The 2D Gabor function is:

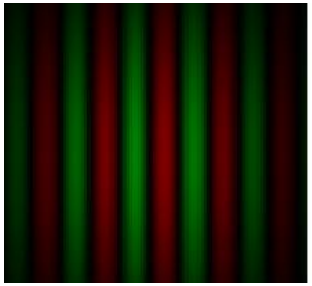
$$h(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\delta^2}} e^{2\pi j(ux+vy)}$$

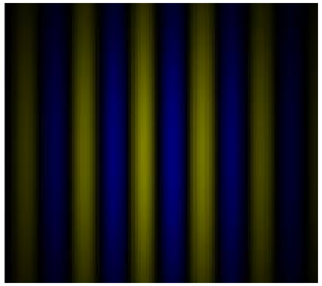
Tuning parameters: $u$, $v$, □

+ usual invariants by combination

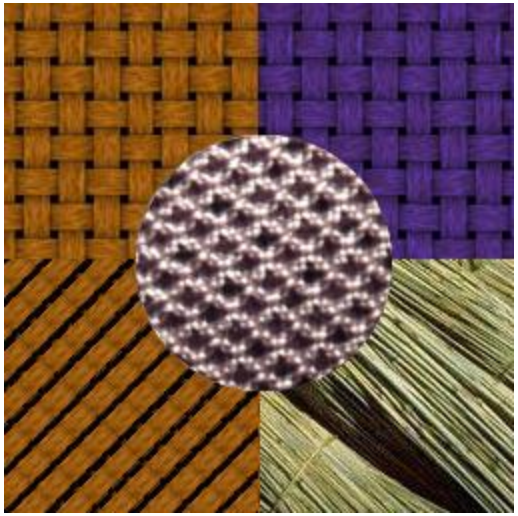


Minh SP 2005

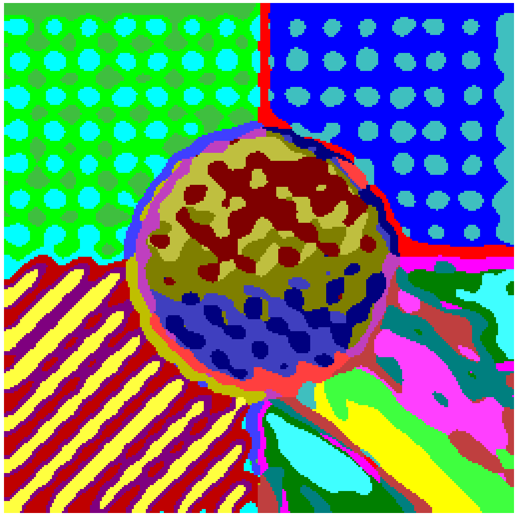




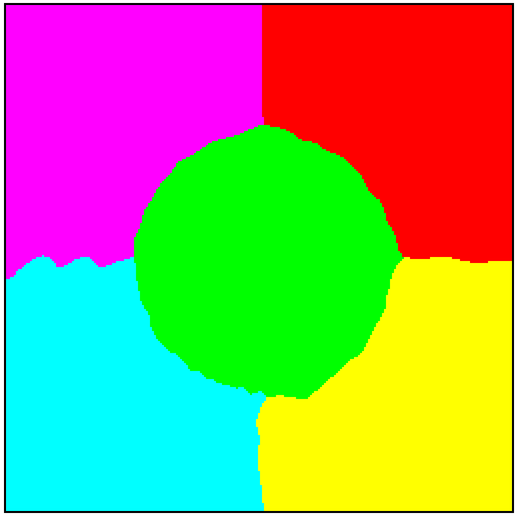
# Texture: Gabor Filters



Original image



K-means clustering



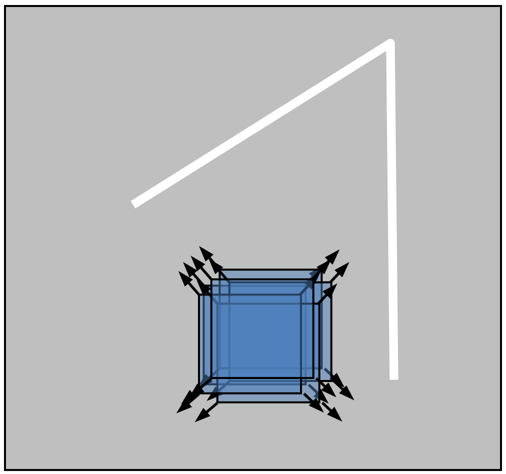
Segmentation

[Minh, ECCV, 2002]

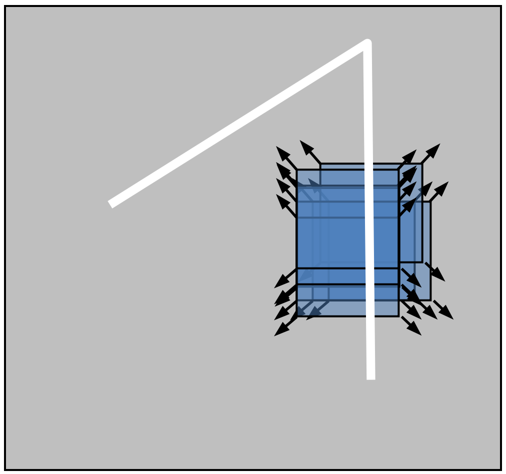
# Texture: Gabor Filters

# Interest point detection
# - Harris corner detector -

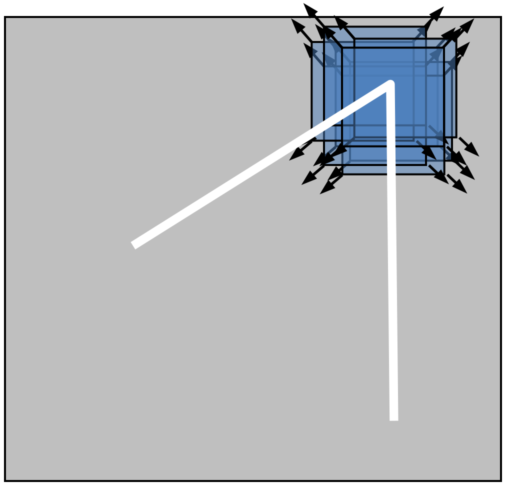
# Basic idea



"flat" region:
no change in intensity



"edge":
no change along the intensity edge direction



"corner":
change in all intensity edge directions

# Intensity change

Change of intensity for a shift [$u$,$v$]:

$$E(u,v) = \sum_{x,y} w(x,y)[I(x+u, y+v) - I(x,y)]^2$$

Window function

Shifted intensity

Intensity

# Harris Detector: Mathematics

For small shifts $[u,v]$ we have a *bilinear* approximation:

$$E(u,v) \cong [u,v] \; M \begin{bmatrix} u \\ v \end{bmatrix}$$

where $M$ is a 2×2 matrix computed from image derivatives:

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

From luminance to color:

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} R_x^2 + G_x^2 + B_x^2 & R_x R_y + G_x G_y + B_x B_y \\ R_x R_y + G_x G_y + B_x B_y & R_y^2 + G_y^2 + B_y^2 \end{bmatrix}$$

# Measure of corner response

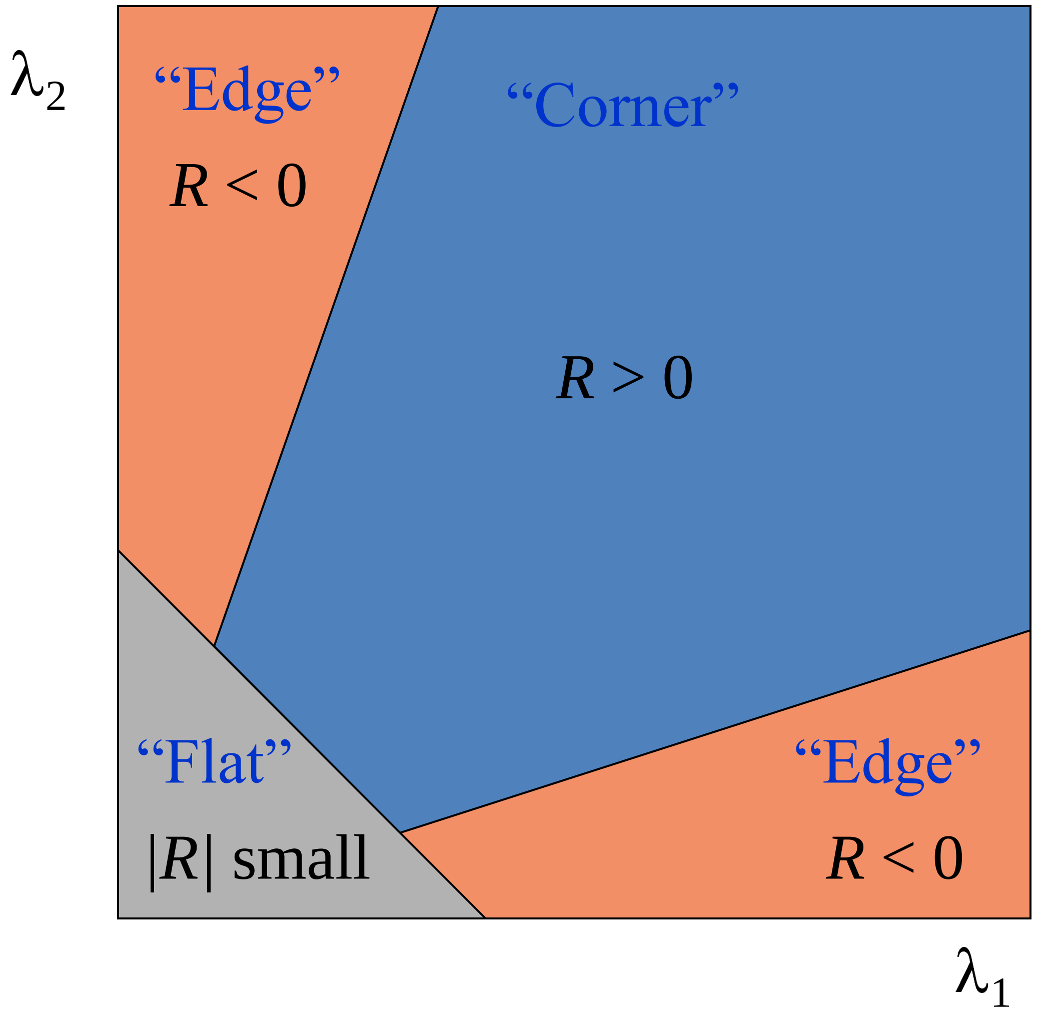
$$R = \det M - k(\operatorname{trace} M)^2$$

$$\det M = \lambda_1 \lambda_2$$

$$\operatorname{trace} M = \lambda_1 + \lambda_2$$

($k$ – empirical constant, $k$ = 0.04-0.06)

# Measure of corner response

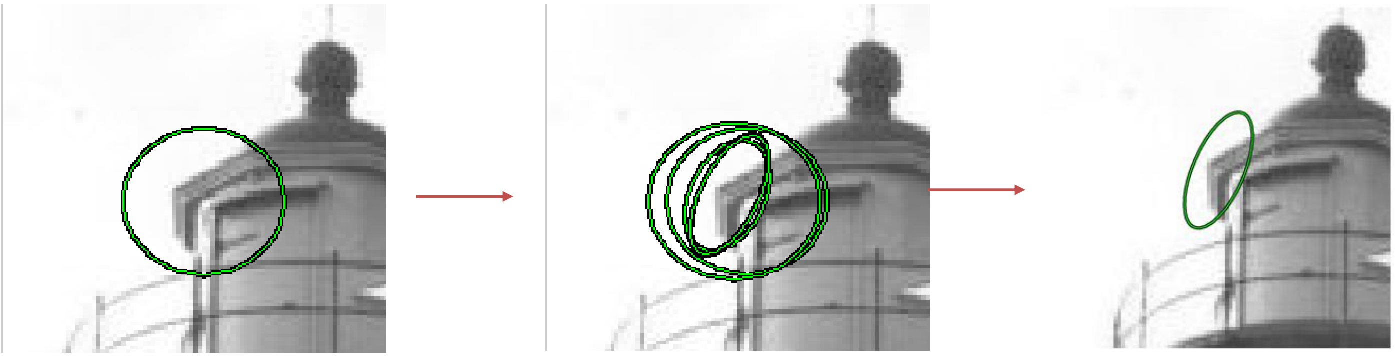


# Harris Detector

- The Algorithm:
  - Find points with large corner response function $R$ ($R$ > threshold)
  - Take the points of local maxima of $R$

# Object and Concept Detection: Find the Proper Scale

- Existing method by Mikolajczyk
  - Iterative affine invariant point detector
    - Multi-scale Harris corner detector
    - Laplacian characteristic scale selection
    - Second moment matrix shape determination



Initial region based on initial scale and location

Iteratively adjust scale, position and shape of region

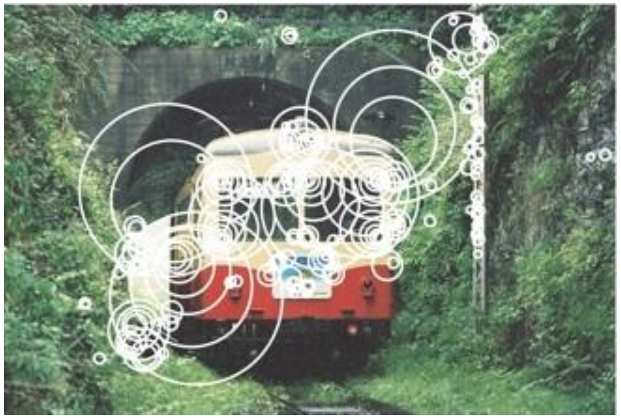
final region

Original Image

Harris Laplacian
impl. by Mikolajczyk (e.g. CVPR06)

Shape adapted Harris Laplacian
impl. by Mikolajczyk (ICCV07)

Color salient points
Quasi invariant HSI



Color salient points
Color boosted OCS

Most of the time, both color approaches agree on the most salient parts of an image.

Original Image

Harris Laplacian
impl. by Mikolajczyk (e.g. CVPR06)

Shape adapted Harris Laplacian
impl. by Mikolajczyk (ICCV07)

Color salient points
Quasi invariant HSI

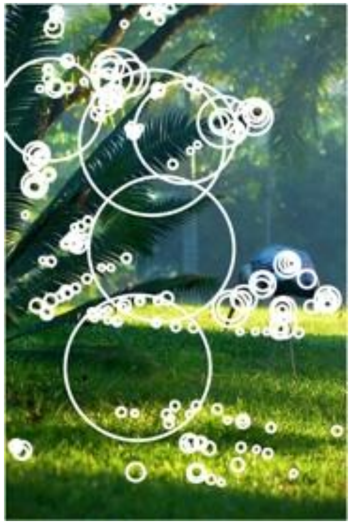
Color salient points
Color boosted OCS

Structured backgrounds of same color tones and shadowing effects are discarded effectively.

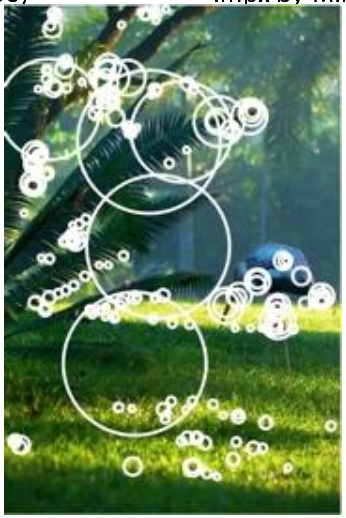
Original Image

Harris Laplacian
impl. by Mikolajczyk (e.g. CVPR06)

Shape adapted Harris Laplacian
impl. by Mikolajczyk (ICCV07)



Color salient points
Quasi invariant HSI



Color salient points
Color boosted OCS

Illumination invariance shifts the features to real color differences - shadows are less salient.

Original Image

Harris Laplacian
impl. by Mikolajczyk (e.g. CVPR06)

Shape adapted Harris Laplacian
impl. by Mikolajczyk (ICCV07)

Color salient points
Quasi invariant HSI

Color salient points
Color boosted OCS


Original Image

Harris Laplacian
impl. by Mikolajczyk (e.g. CVPR06)

Shape adapted Harris Laplacian
impl. by Mikolajczyk (ICCV07)

Color salient points
Quasi invariant HSI

Color salient points
Color boosted OCS

Original Image

Harris Laplacian
impl. by Mikolajczyk (e.g. CVPR06)

Shape adapted Harris Laplacian
impl. by Mikolajczyk (ICCV07)

Color salient points
Quasi invariant HSI

Color salient points
Color boosted OCS

# Overview

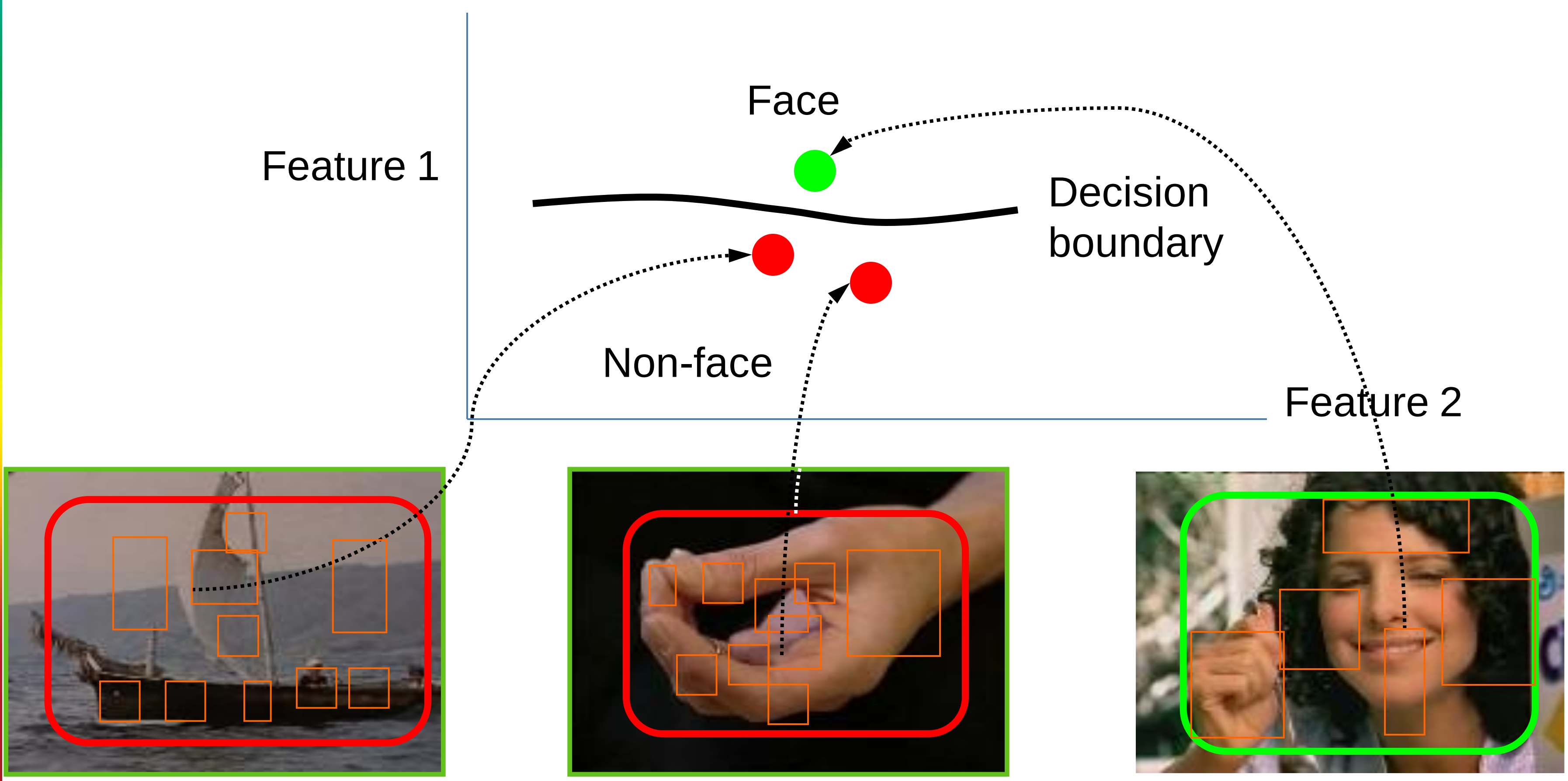
**_PART I (low-level)_**

1. ***Reflection Models***
   - Dichromatic reflection model
2. ***Photometric/Color Invariance***
   - At the pixel
   - Instability handling
   - Color differential structure
3. ***Color Constancy***
   - Low-level
   - High-level
4. ***Saliency and Color Boosting***
   - Itti and Koch model
   - Color boosted

**_PART II (higher Level)_**

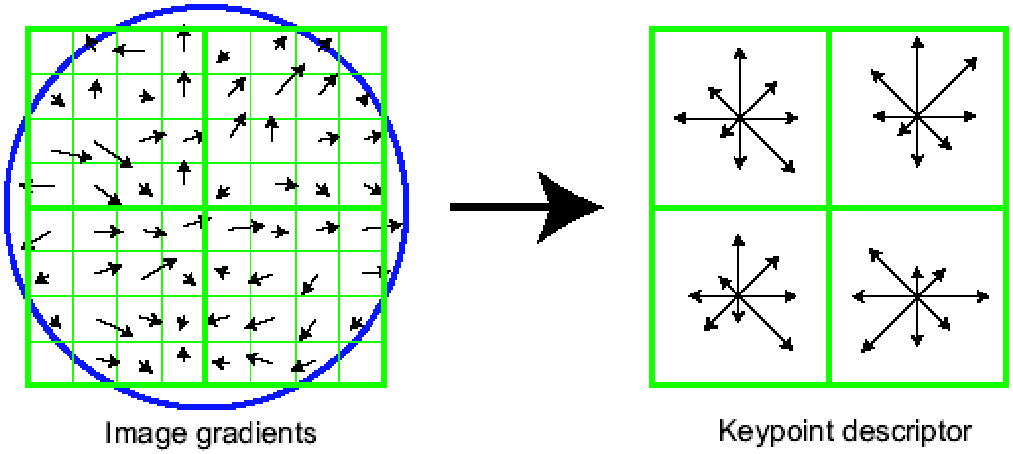
1. ***Interest point detection***
   - Harris Laplace
   - Color boosted
2. ***Descriptors***
   - SIFT
   - Extension to color
3. ***Object recognition (VOC/TRECVid)***
   - Dense and point sampling
   - Code book generation
   - Results
4. ***Applications***
   - Tracking in video
   - Object replacement
   - Emotion recognition
   - Head pose estimation

# Machine learning (K-NN, SVM)

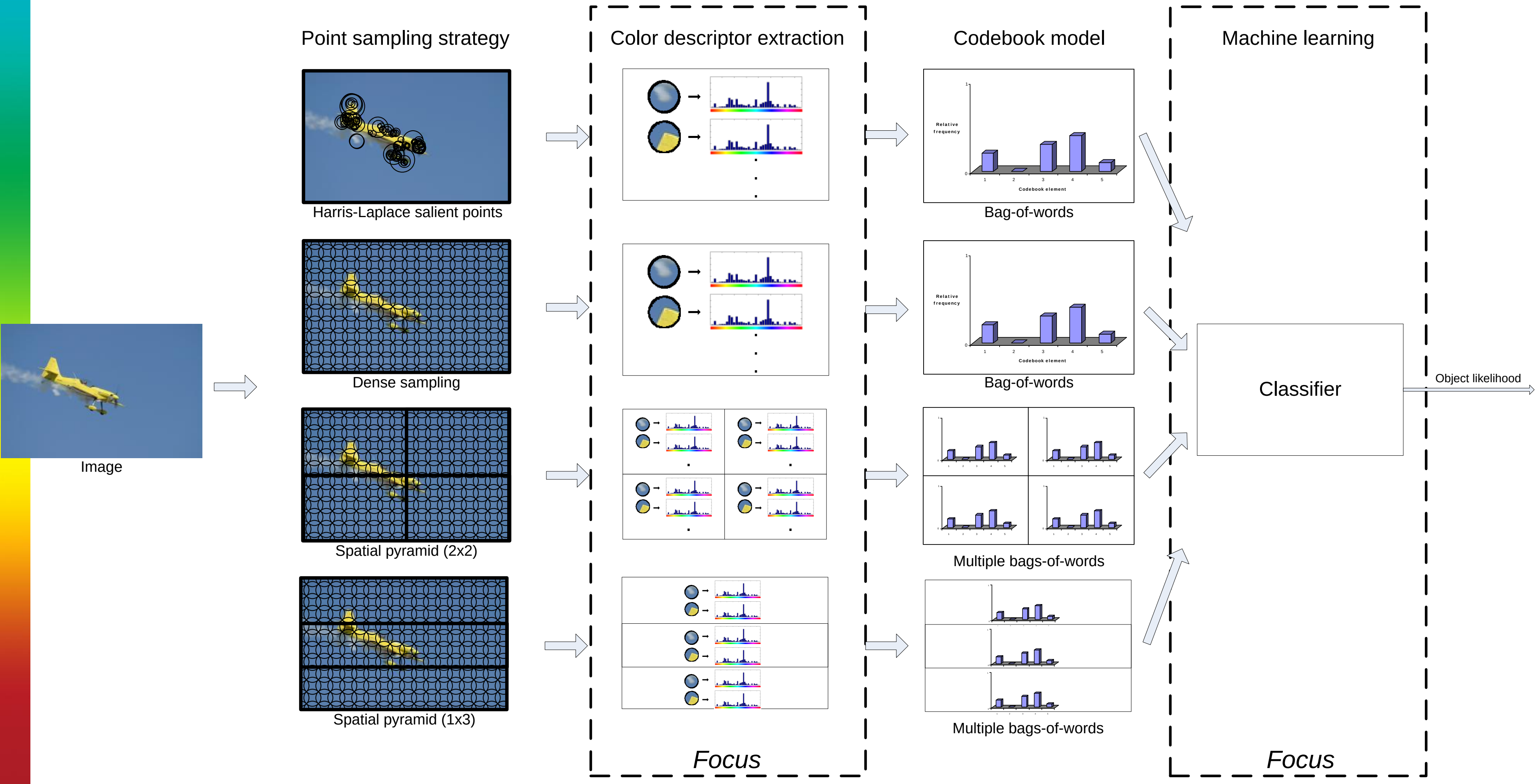


# SIFT – Scale Invariant Feature Transform

- Descriptor overview:
  - Determine scale (by maximizing DoG in scale and in space), local orientation as the dominant gradient direction.
    Use this scale and orientation to make all further computations invariant to scale and rotation.
  - Compute gradient orientation histograms of several small windows (128 values for each point)
  - Normalize the descriptor to make it invariant to intensity change



# Pipeline Overview



Von Kries 1970
Finlayson ICIP 2005

# Invariance properties: Diagonal model

Lambertian reflectance model

$$\mathbf{f}(\mathbf{x}) = \int_\omega e(\lambda)\rho_k(\lambda)s(\mathbf{x},\lambda)d\lambda + \int_\omega a(\lambda)\rho_k(\lambda)$$

Corresponds to diagonal-offset model of illumination change

$$\boxed{\begin{pmatrix} R^c \\ G^c \\ B^c \end{pmatrix}} = \boxed{\begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{pmatrix}} \boxed{\begin{pmatrix} R^u \\ G^u \\ B^u \end{pmatrix}} + \boxed{\begin{pmatrix} o_1 \\ o_2 \\ o_3 \end{pmatrix}}$$

Illuminant parameters

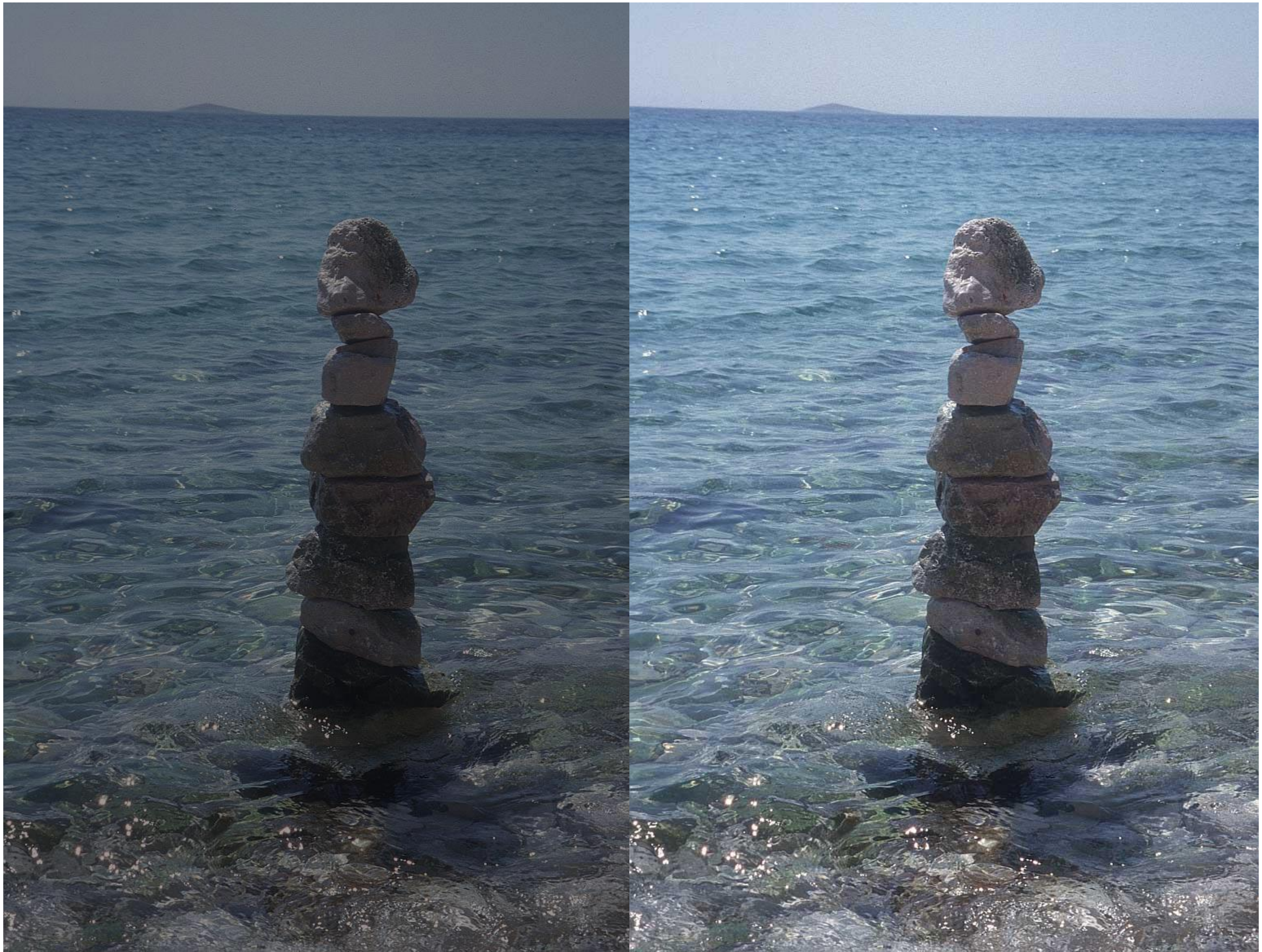
Canonical illuminant

Unknown illuminant

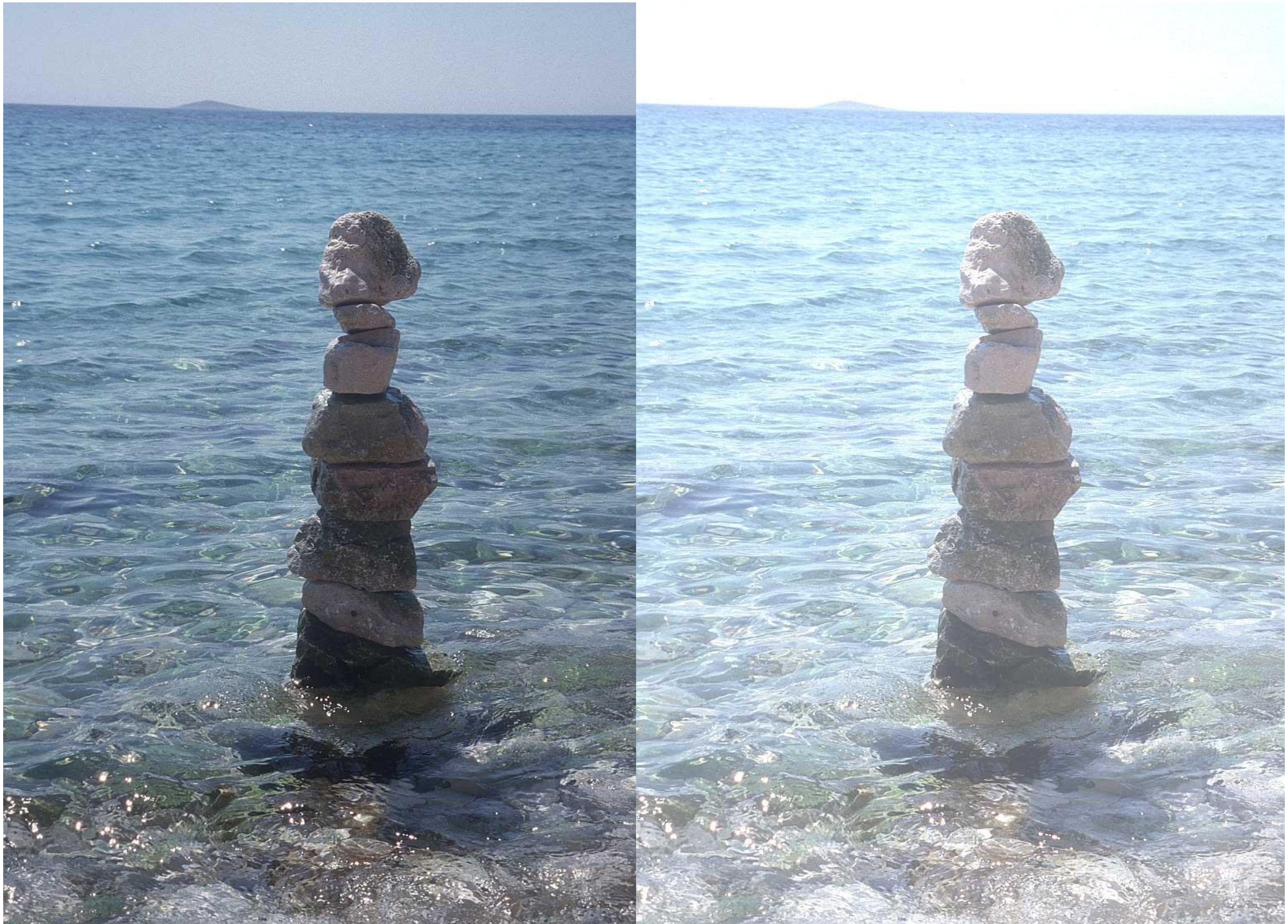
Unified framework for modeling:

- Shadows
- Shading
- Light color changes
- Highlights
- Scattering

# Photometric Analysis



# Photometric Analysis (2)



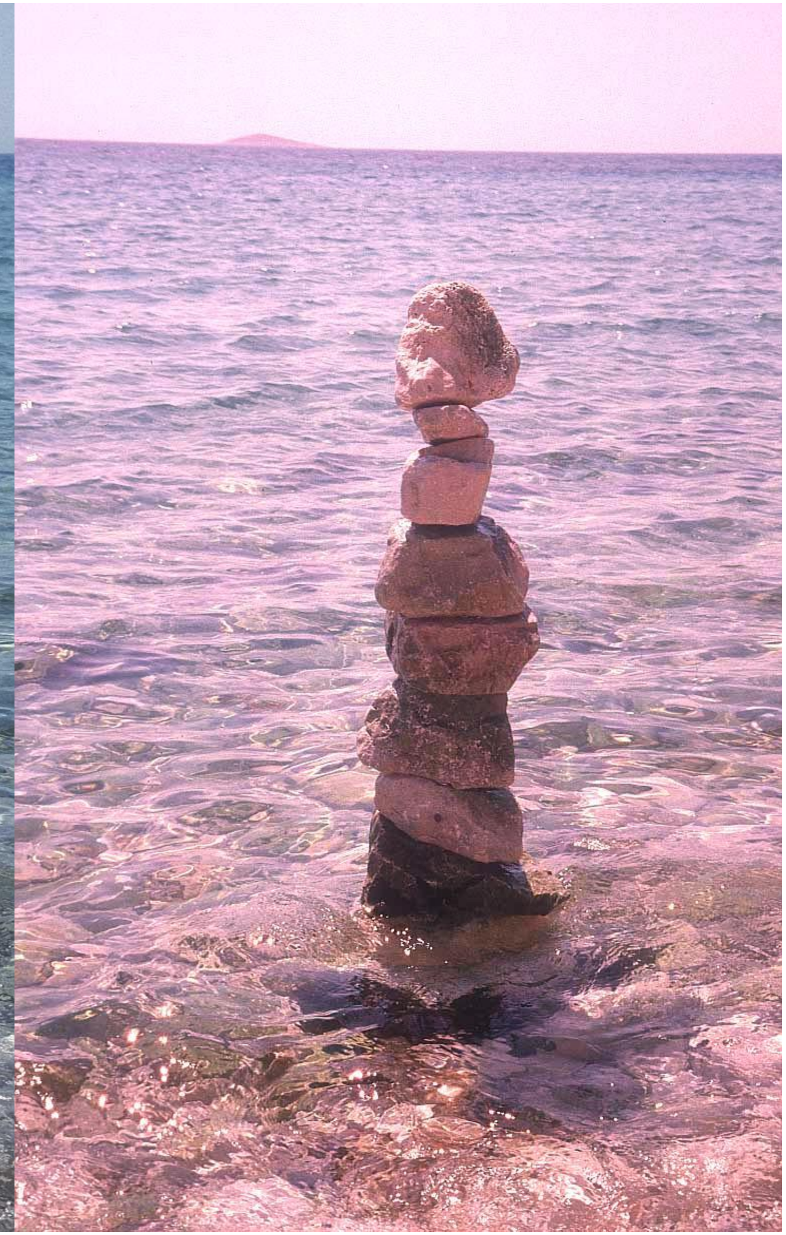
# Photometric Analysis (3)

3. Light intensity change *and* shift ($a = b = c$; $o_1 = o_2 = o_3$)

$$\begin{pmatrix} R^c \\ G^c \\ B^c \end{pmatrix} = \begin{pmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & a \end{pmatrix} \begin{pmatrix} R^u \\ G^u \\ B^u \end{pmatrix} + \begin{pmatrix} o_1 \\ o_1 \\ o_1 \end{pmatrix}$$

➔ *scale-invariant* and *shift-invariant*

$I^c = a\, I^u + o_1$



**(1,1,1)** **(1.8,1.2,1.2)**

# Color Descriptor Taxonomy

[van de Sande, IEEE PAMI, 09]

| | Light intensity change $\begin{pmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & a \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$ | Light intensity shift $\begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} o_1 \\ o_1 \\ o_1 \end{pmatrix}$ | Light intensity change and shift $\begin{pmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & a \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} o_1 \\ o_1 \\ o_1 \end{pmatrix}$ | Light color change $\begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$ | Light color change and shift $\begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} o_1 \\ o_2 \\ o_3 \end{pmatrix}$ |
|---|---|---|---|---|---|
| RGB Histogram | - | - | - | - | - |
| $O_1, O_2$ | - | + | - | - | - |
| $O_3$, Intensity | - | - | - | - | - |
| Hue | + | + | + | - | - |
| Saturation | + | + | + | - | - |
| $r, g$ | + | - | - | - | - |
| Transformed color | + | + | + | + | + |
| Color moments | - | + | - | - | - |
| Moment invariants | + | + | + | + | + |
| SIFT ($\nabla I$) | + | + | + | + | + |
| HSV-SIFT | + | + | + | +/- | +/- |
| HueSIFT | + | + | + | +/- | +/- |
| OpponentSIFT | +/- | + | +/- | +/- | +/- |
| W-SIFT | + | + | + | +/- | +/- |
| $rg$SIFT | + | + | + | +/- | +/- |
| Transf. color SIFT | + | + | + | + | + |

# Overview

### *PART I (low-level)*

1. ***Reflection Models***
   - Dichromatic reflection model
2. ***Photometric/Color Invariance***
   - At the pixel
   - Instability handling
   - Color differential structure
3. ***Color Constancy***
   - Low-level
   - High-level
4. ***Saliency and Color Boosting***
   - Itti and Koch model
   - Color boosted

### *PART II (higher Level)*

1. ***Interest point detection***
   - Harris Laplace
   - Color boosted
2. ***Descriptors***
   - SIFT
   - Extension to color
3. ***Object recognition (VOC/TRECVid)***
   - Dense and point sampling
   - Code book generation
   - Results
4. ***Applications***
   - Tracking in video
   - Object replacement
   - Emotion recognition
   - Head pose estimation

# Evaluating Color Descriptors

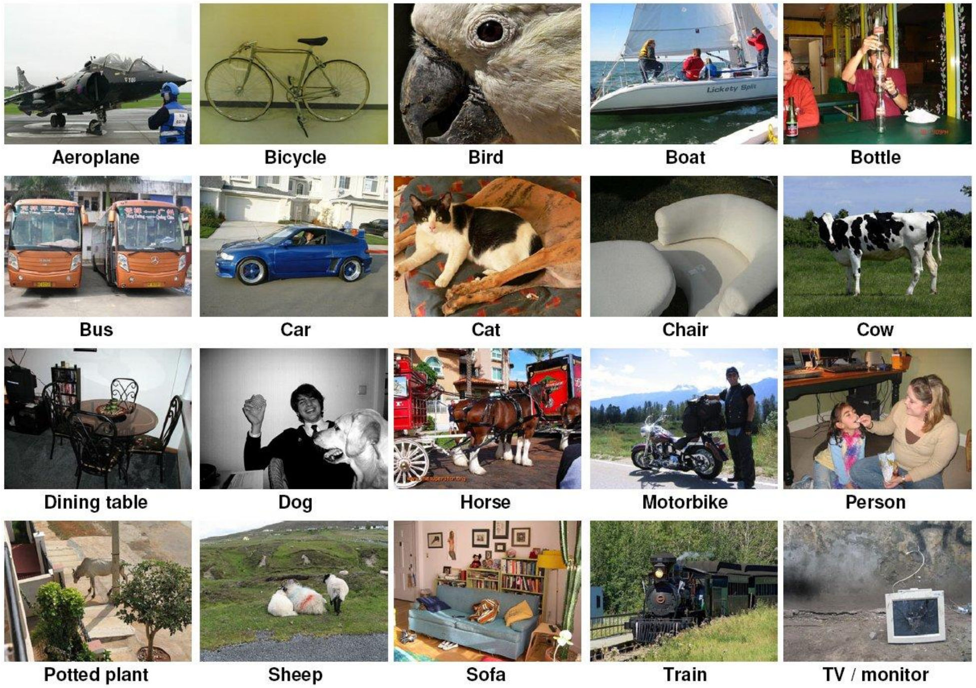
Look at:

1. *Repeatability*
   Analytically: taxonomy of invariant properties within the diagonal model of illumination change
2. *Distinctiveness*
   Experimentally: using image and video benchmarks

# Distinctiveness

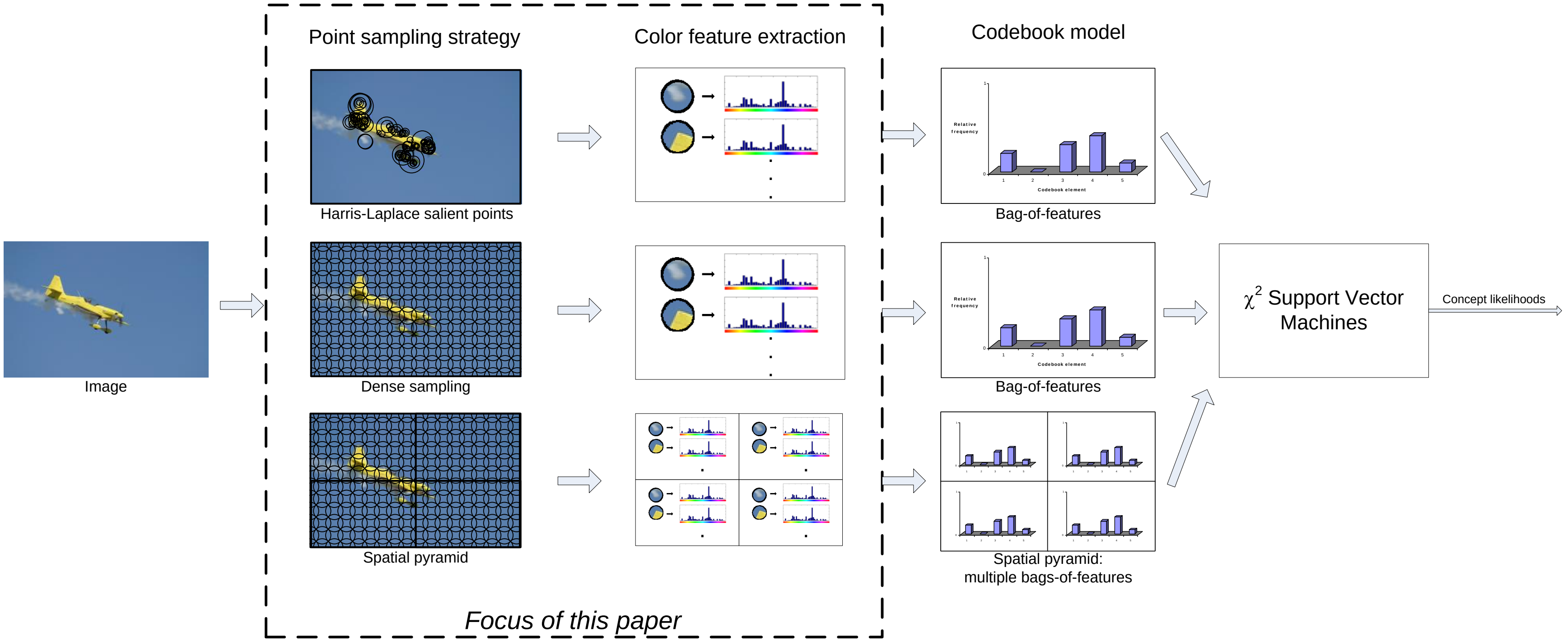
Distinctiveness studied experimentally:

- Image benchmark: PASCAL Visual Object Classes Challenge 2007
- 9963 photos from Flickr
- 20 object types
- Earth Movers Distance (EMD) between cluster sets of different images, used in EMD kernel function for SVM [ZhangIJCV2007]



# PASCAL VOC 2007/2008

Codebook size=4000




**Point sampling**

Harris-Laplace

Dense sampling

**Spatial Pyramid**

1x1

2x2

1x3

**Color Descriptor**

SIFT

OpponentSIFT

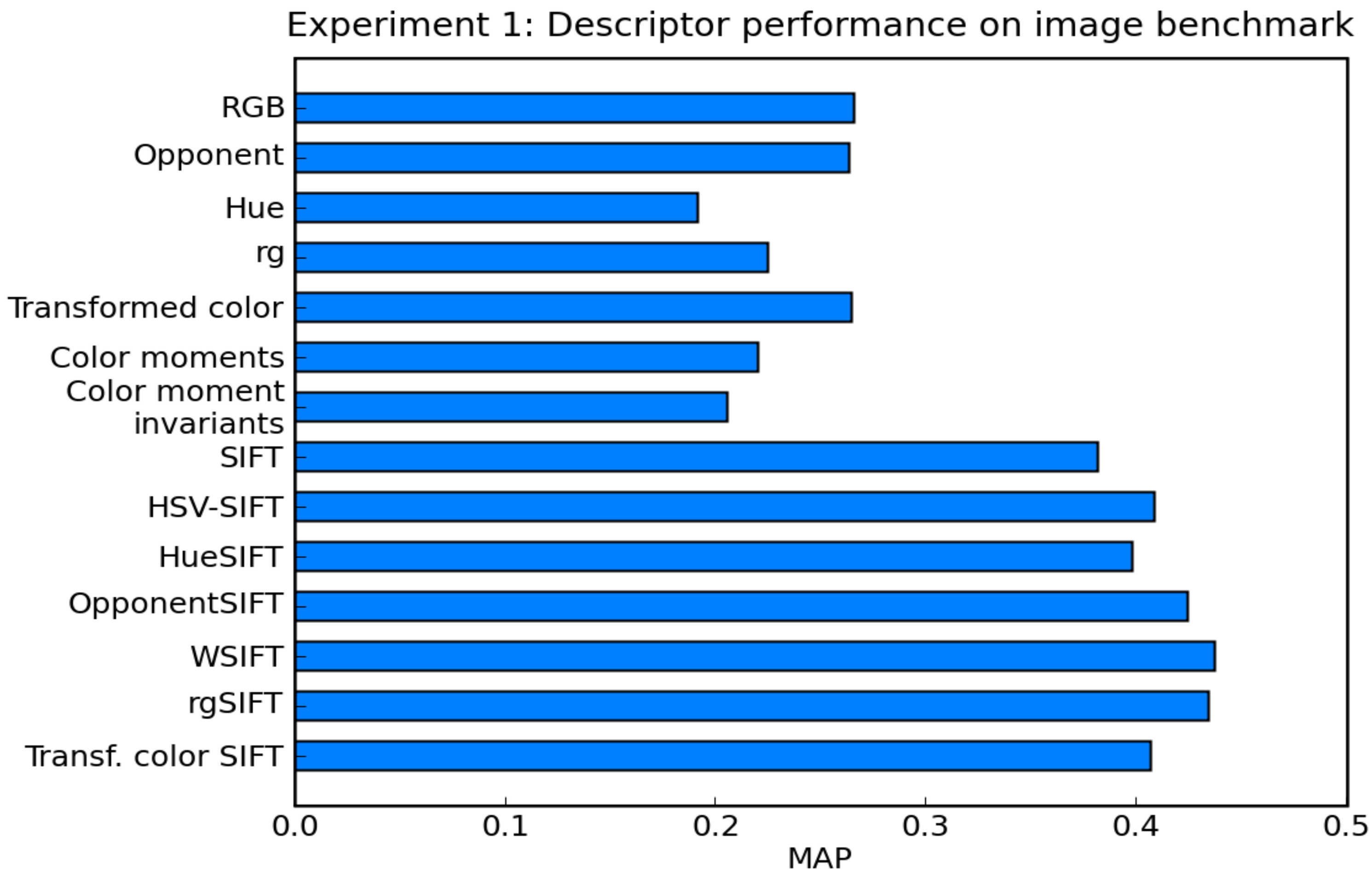
WSIFT

rgSIFT

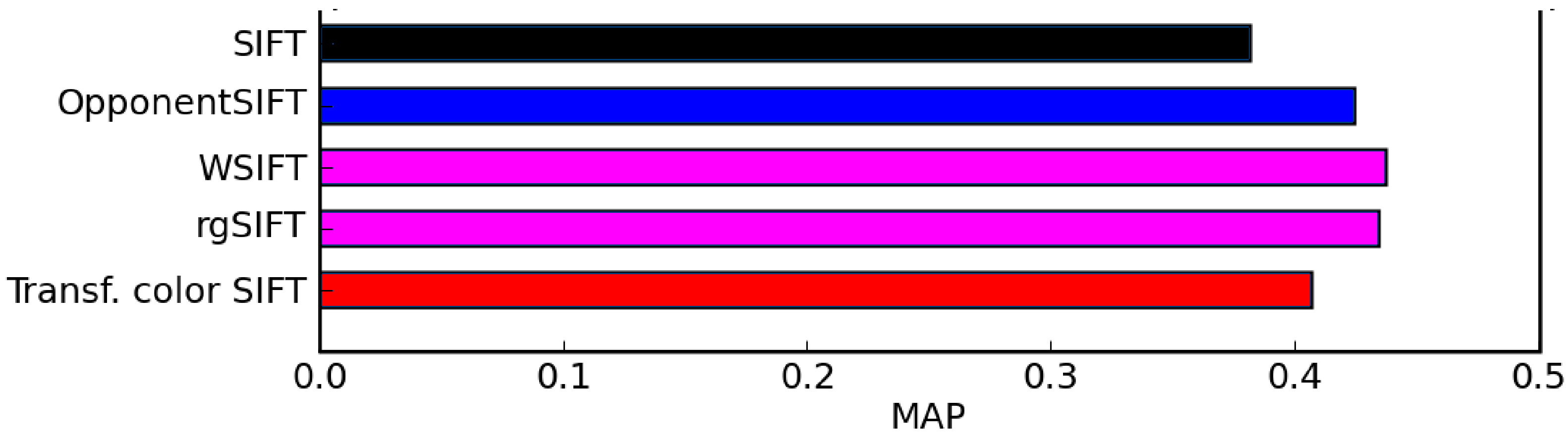
Transformed color SIFT

[van de Sande, IEEE PAMI, 09]

# Results on PASCAL VOC 2007

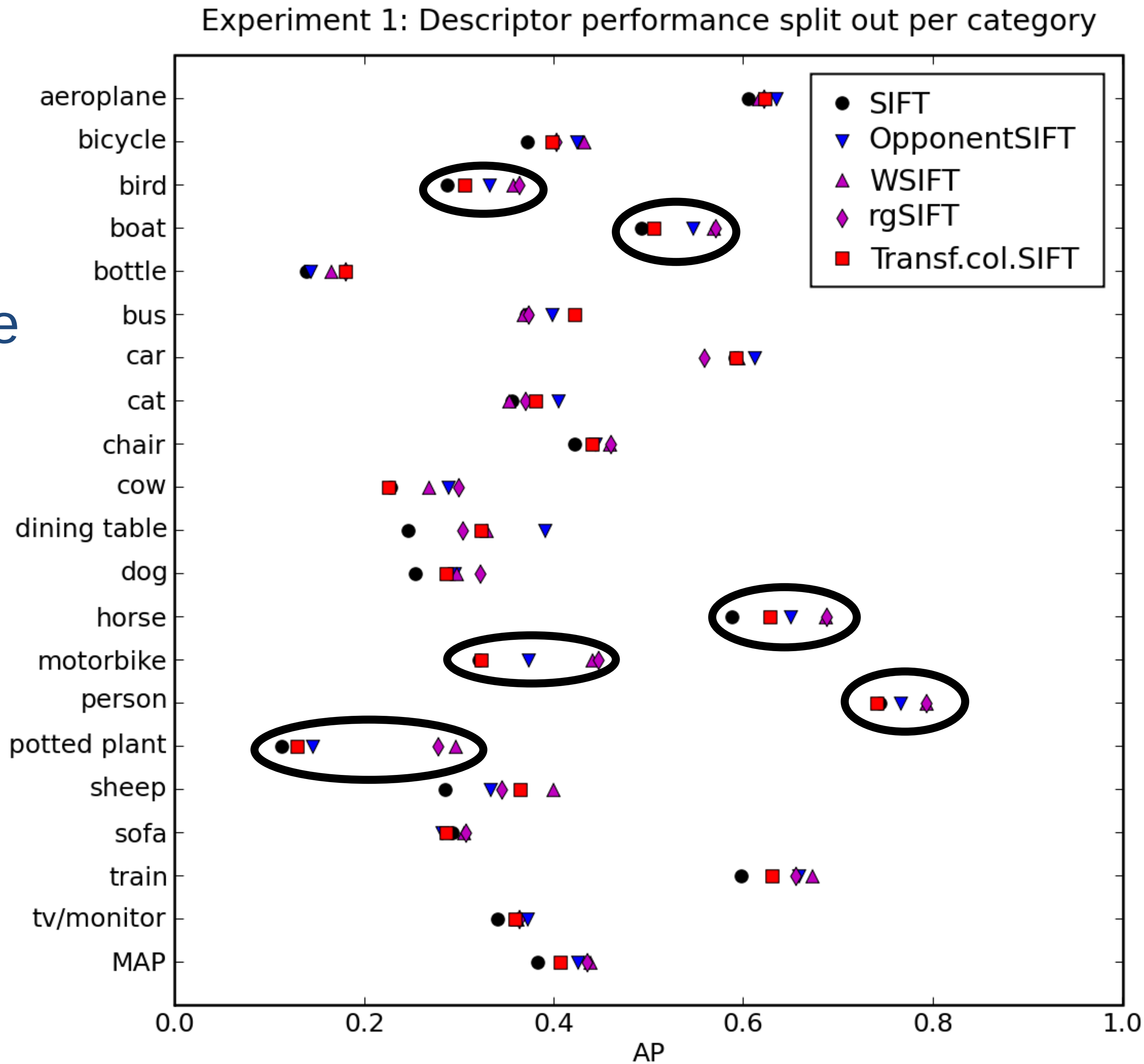


# Results on PASCAL VOC 2007 (2)




| | Light intensity change $\begin{pmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & a \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$ | Light intensity shift $\begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} o_1 \\ o_1 \\ o_1 \end{pmatrix}$ | Light intensity change and shift $\begin{pmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & a \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} o_1 \\ o_1 \\ o_1 \end{pmatrix}$ | Light color change $\begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$ | Light color change and shift $\begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} o_1 \\ o_2 \\ o_3 \end{pmatrix}$ |
|---|---|---|---|---|---|
| SIFT | + | + | + | + | - |
| OpponentSIFT | + | + | + | - | - |
| WSIFT | + | + | + | - | - |
| rgSIFT | + | + | + | - | - |
| **Transf. SIFT** | + | + | + | + | + |

Scale-invariance w.r.t. light intensity important



# Conclusion

- #1 model for the VOC:

**Light intensity change and shift**

$$\begin{pmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & a \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} o_1 \\ o_1 \\ o_1 \end{pmatrix}$$

- You need scale-invariance and shift-invariance w.r.t. light intensity
- Invariance to light color is not needed and decreases the discriminative power

# VOC2008 results

| | LEAR_shotgun | SurreyUvA_SRKDA | UvA_Soft5ColorSift | UvA_TreeSFS |
|---|---|---|---|---|
| aeroplane | 81.1 | 79.5 | 79.7 | 80.8 |
| bicycle | 52.9 | 54.3 | 52.1 | 53.2 |
| bird | 61.6 | 61.4 | 61.5 | 61.6 |
| boat | 67.8 | 64.8 | 65.5 | 65.6 |
| bottle | 29.4 | 30.0 | 29.1 | 29.4 |
| bus | 52.1 | 52.1 | 46.5 | 49.9 |
| car | 58.7 | 59.5 | 58.3 | 58.5 |
| cat | 59.9 | 59.4 | 57.4 | 59.4 |
| chair | 48.5 | 48.9 | 48.2 | 48.0 |
| cow | 32.0 | 33.6 | 27.9 | 30.1 |
| dining table | 38.6 | 37.8 | 38.3 | 39.6 |
| dog | 47.9 | 46.0 | 46.6 | 45.0 |
| horse | 65.4 | 66.1 | 66.0 | 67.3 |
| motor bike | 65.2 | 64.0 | 60.6 | 60.4 |
| person | 87.0 | 86.8 | 87.0 | 87.1 |
| potted plant | 29.0 | 29.2 | 31.8 | 30.1 |
| sheep | 34.4 | 42.3 | 42.2 | 41.5 |
| sofa | 43.1 | 44.0 | 45.3 | 45.4 |
| train | 74.3 | 77.8 | 72.3 | 74.3 |
| tv/monitor | 61.5 | 61.2 | 64.7 | 59.8 |
| **MAP** | **54.5** | **54.9** | **54.1** | **54.4** |

# Color Descriptors on VOC08

- Invariance properties of the descriptors used

| | Light intensity change $\begin{pmatrix}a&0&0\\0&a&0\\0&0&a\end{pmatrix}\begin{pmatrix}R\\G\\B\end{pmatrix}$ | Light intensity shift $\begin{pmatrix}R\\G\\B\end{pmatrix}+\begin{pmatrix}o_1\\o_1\\o_1\end{pmatrix}$ | Light intensity change and shift $\begin{pmatrix}a&0&0\\0&a&0\\0&0&a\end{pmatrix}\begin{pmatrix}R\\G\\B\end{pmatrix}+\begin{pmatrix}o_1\\o_1\\o_1\end{pmatrix}$ | Light color change $\begin{pmatrix}a&0&0\\0&b&0\\0&0&c\end{pmatrix}\begin{pmatrix}R\\G\\B\end{pmatrix}$ | Light color change and shift $\begin{pmatrix}a&0&0\\0&b&0\\0&0&c\end{pmatrix}\begin{pmatrix}R\\G\\B\end{pmatrix}+\begin{pmatrix}o_1\\o_2\\o_3\end{pmatrix}$ |
|---|---|---|---|---|---|
| SIFT | + | + | + | + | + |
| OpponentSIFT | + | + | + | - | - |
| WSIFT | + | + | + | - | - |
| *rg*SIFT | + | + | + | - | - |
| RGBSIFT | + | + | + | + | + |

| Descriptors | MAP on VOC2008val |
|---|---|
| Intensity SIFT | 42,3 |
| All five (=Soft5ColorSIFT) | 45,5 |

By adding color:
+8%

# TRECVid

Koen van de Sande
Cees Snoek
Jan van Gemert
Jasper Uijlings
Jan-Mark Geusebroek
Theo Gevers
Arnold Smeulders

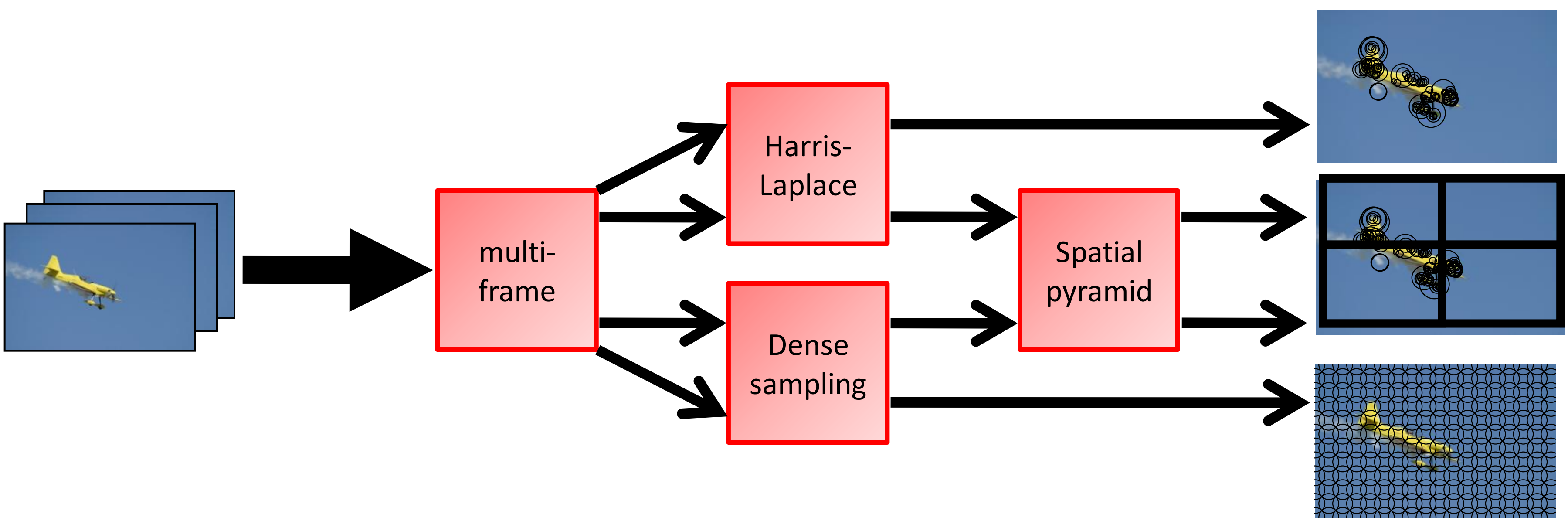
**University of Amsterdam**

[Lazebnik, CVPR06]
[Marszalek, VOC07]

# Spatio-Temporal Sampling

- Spatial pyramid
  - 1x1 whole image
  - 2x2 image quarters
  - 1x3 horizontal bars

- Temporal analysis of up to 5 frames per shot



# Invariant Visual Descriptors

Color SIFT:

- Intensity-based SIFT
- OpponentSIFT
- C-SIFT
- *rg*SIFT
- Transformed color SIFT


Add color, but also keep intensity information

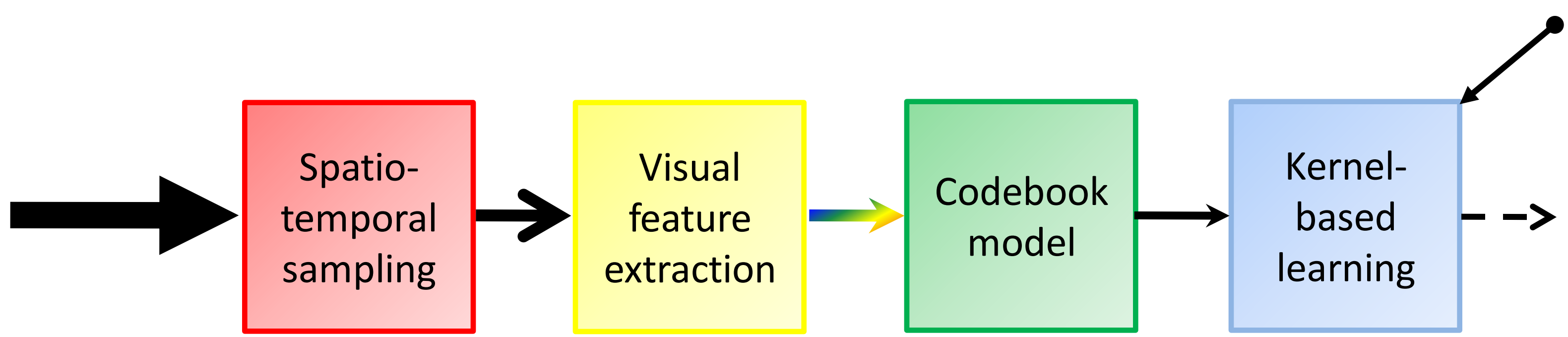
| Visual Descriptors | MAP on TV2007test |
|---|---:|
| Intensity SIFT | 0,144 |
| 5x Color SIFT | 0,155 |

**relative +8%**

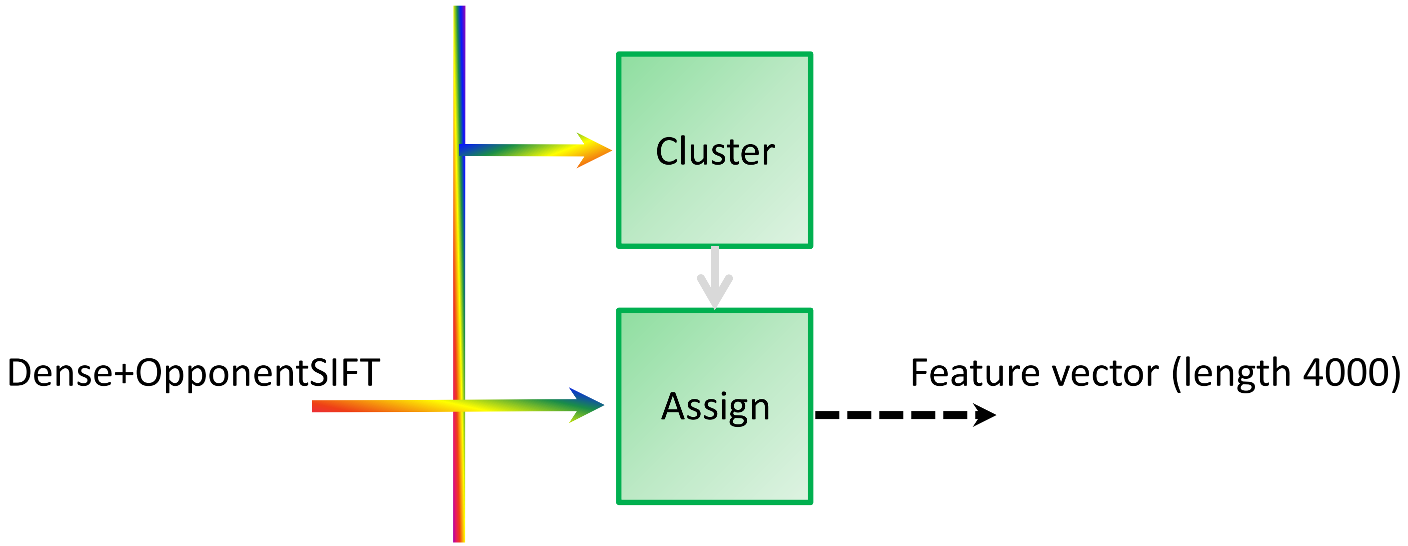
TV2007test results:

- Trained on TRECVID2007 development set
- Evaluated on TRECVID2007 test set
- TRECVID2007 development + test = 2008 development

# Concept Detection Stages



# Visual Codebook Model

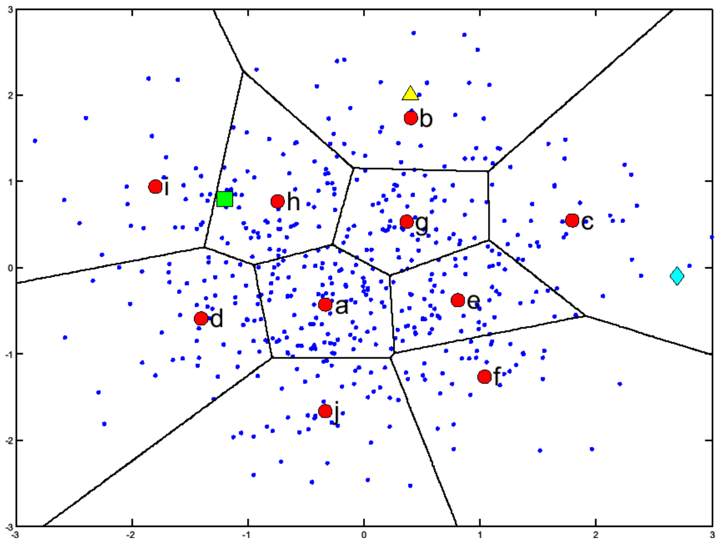



- Codebook consists of codewords
- Constructed with k-means clustering on descriptors
- We use 4,000 codewords per codebook

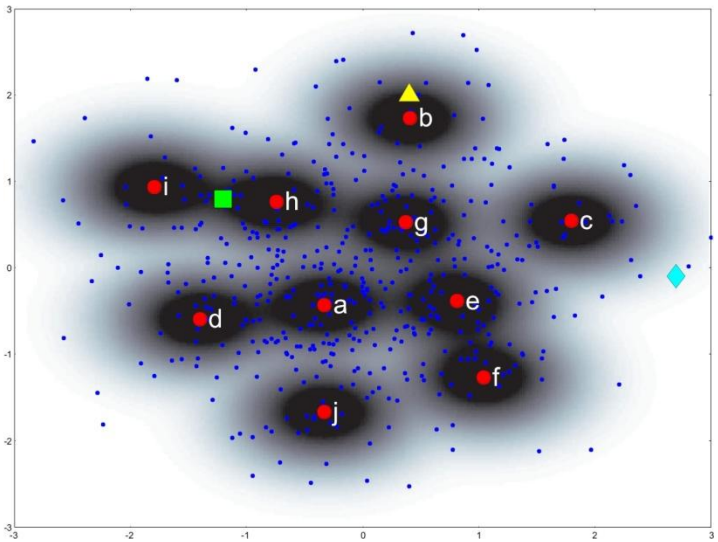
Codeword

# Codebook Assignment

## Soft assignment using Gaussian kernel




Hard assignment

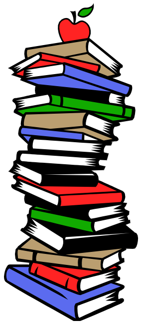



Soft assignment

| Assignment | MAP on TV2007test |
|---|---|
| Hard | 0,155 |
| Soft | 0,166 |

relative +7%

# Codebook Library



| Codebook | Sampling method | Descriptor | Construction | Assignment |
|---|---|---|---|---|
| #1 | Dense | OpponentSIFT | K-means | Soft |
| #2 | Harris-Laplace | SIFT | Radius-based | Soft |
| #3 | Dense | *rg*SIFT | K-means | Hard |
| … | Dense | C-SIFT | K-means | Hard |

Single codebook depends on

- Sampling method
- Descriptor
- Codebook construction method
- Codebook assignment

Codebook library is…

- a configuration of several codebooks

[Snoek, ICME05]

# Robust Temporal Approach

- No cloud computing yet: need to be efficient ☺
- Process 5 frames per shot in test set
- Linear increase in computation: x5

| Codebook library | Frames/shot | MiAP on TV2008test |
|---|---|---|
| 3x Color SIFT | 1 | 0,152 |
| 3x Color SIFT | 5 | 0,184 |

relative +20%

- In 2005 paper 7.5% to 38% improvement noted for multi-frame (worst-case vs. best-case using oracle)
- **Robust color SIFT *with* temporal = ~20% improvement**

# The Good

- Close-up of hands

- Boats and ships

- Cityscape

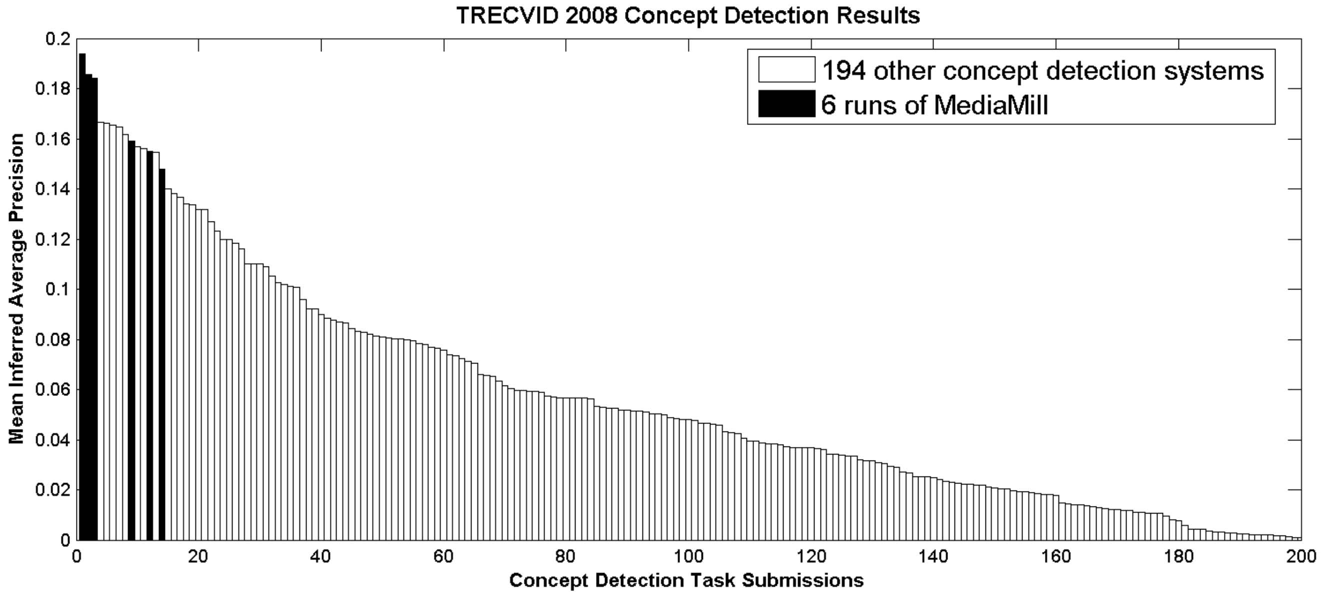
# The Bad

- Emergency Vehicle (only 46 examples, many at night)

- Bus (only 64 examples)

# Conclusions

- Illumination conditions affect concept detection
- SIFT+colorSIFT improves ~8%
- Soft codebook assignment improves ~7%
- Robust colorSIFT with simple multi-frame improves ~20%:
- Precomputed kernel matrix reduces SVM computation time
- Near-duplicates from trailers hamper progress:
  - We suggest to exclude them, or count only once

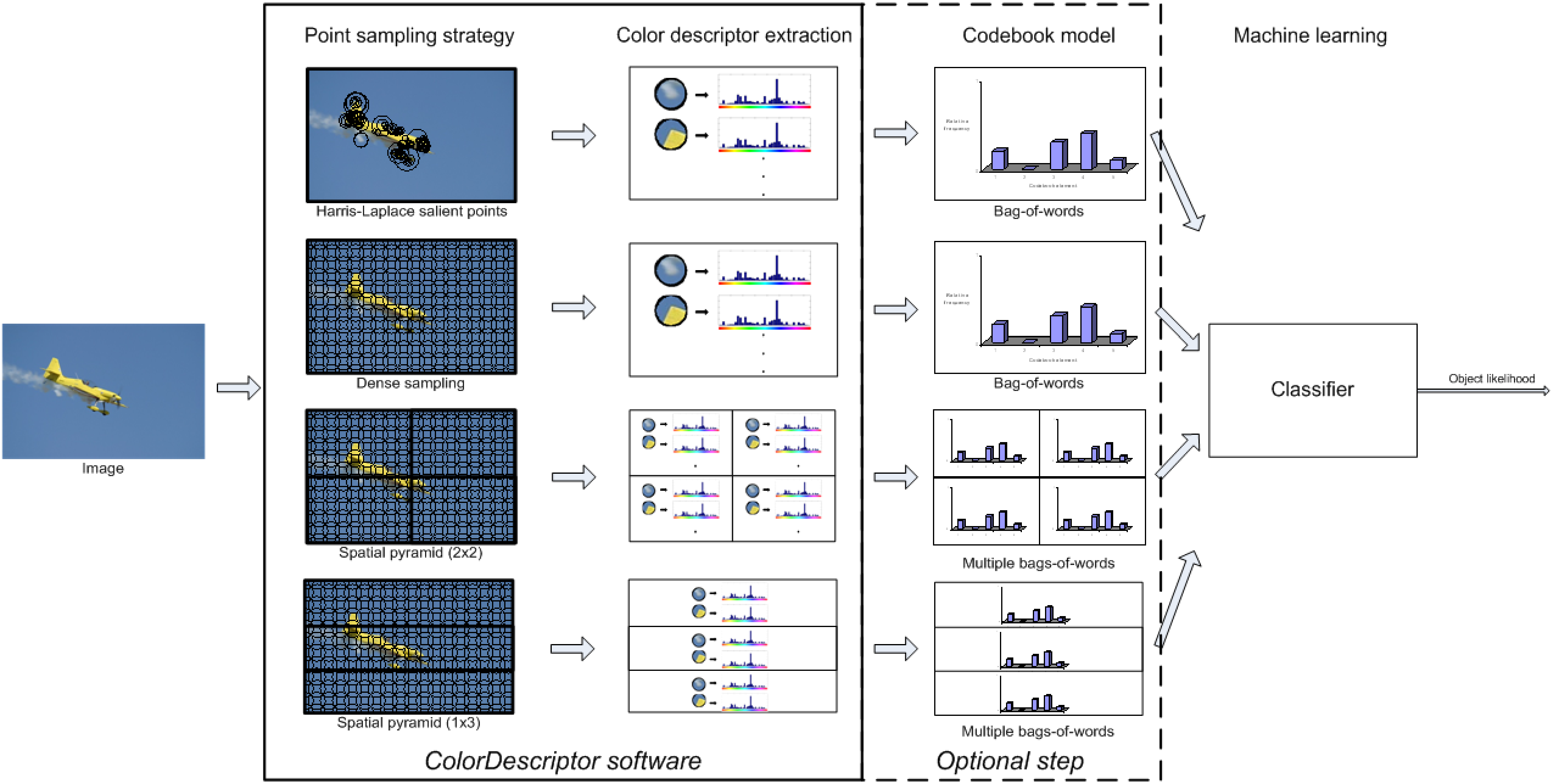


# References


- K. E. A. van de Sande, T. Gevers and C. G. M. Snoek, "*Evaluation of Color Descriptors for Object and Scene Recognition*", CVPR 2008
- M. Marszalek, C. Schmid, H. Harzallah and J. van de Weijer, "*Learning Object Representations for Visual Object Class Recognition*", Visual Recognition Workshop in conjunction with ICCV 2007
- J.C. van Gemert, J.M. Geusebroek, C.J. Veenman, A.W.M. Smeulders, "*Kernel Codebooks for Scene Categorization*", ECCV 2008
- K. Mikolajczyk and C. Schmid, "*A Performance Evaluation of Local Descriptors*", PAMI 2005
- D. G. Lowe, "*Distinctive Image Features from Scale-Invariant Keypoints*", IJCV 2004
- J. Zhang, M. Marszalek, S. Lazebnik and C. Schmid, "*Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study*", IJCV 2007
- C. G. M. Snoek et al, "*The MediaMill TRECVID 2008 Semantic Video Search Engine*", TRECVID Workshop 2008

# ColorDescriptor software

## for object and scene categorization




**Created by Koen van de Sande**



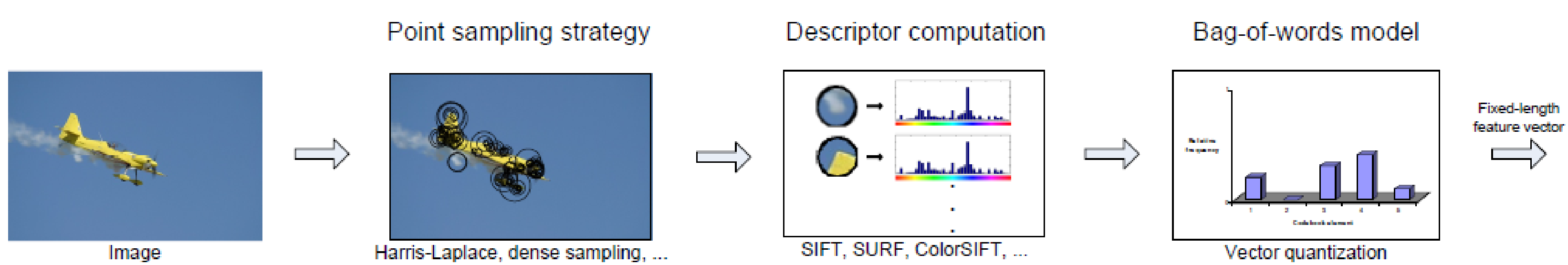
**Visit http://colordescriptors.com for color descriptor software**

# Conclusion

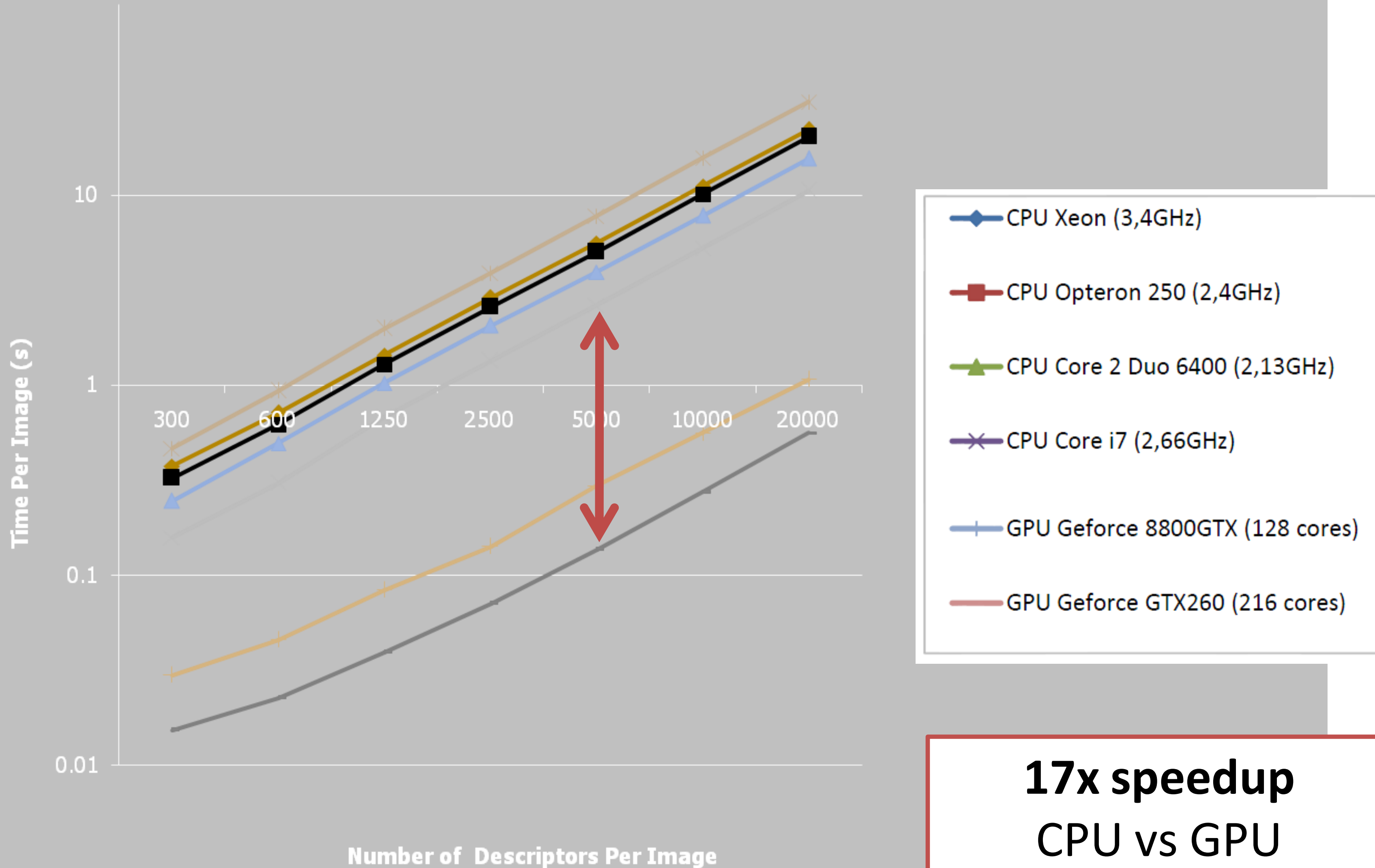
- Bag-of-word approach works
- Good local descriptors: SIFT, OpponentSIFT, rgSIFT/WSIFT, RGB SIFT
- Combining these color features gives state-of-the-art performance:
  - PASCAL VOC: 1$^{st}$ position in VOC08 and shared 1$^{st}$ in VOC09.
  - TRECVid: 1$^{st}$ position in TRECVid08 and TRECVid09.
- Drawback: computational costs of bag-of-word approach

# GPU-Accelerated Feature Extraction

- Single bag-of-words feature up to 15s/frame (CPU-time)
- TRECVid 2008 / PASCAL VOC 2008 UvA entries used 10 of these features
- More than 80% of time spent in vector quantization



# Vector Quantization Timings for ColorSIFT

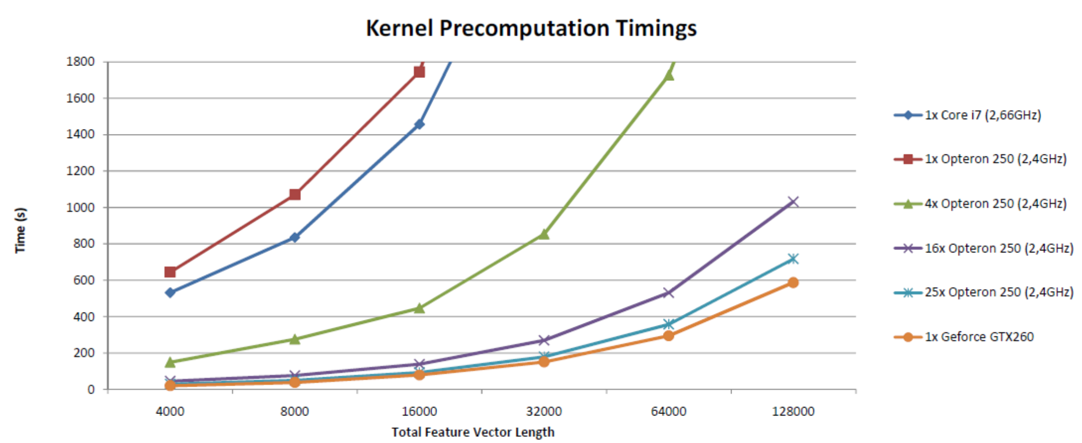



**17x speedup**

CPU vs GPU

# Kernel Value Precomputation

- Step from image feature vectors to kernel-based classifiers from WP5 (SVM/SR-KDA)
- Computes $\chi^2$ distance between pairs of images
- Suitable for GPU implementation: **22x speedup**
- TRECVid 2008 processing time: 800 CPU hours vs. 37 GPU hours




$\Rightarrow$ Process datasets order of magnitude larger

***or***

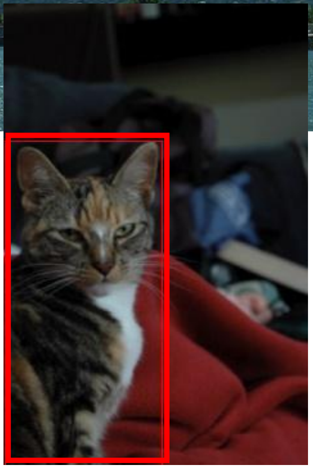
$\Rightarrow$ Single GPU replaces medium-sized cluster

# Object Localization in Images using Sliding Windows

Henco Visser

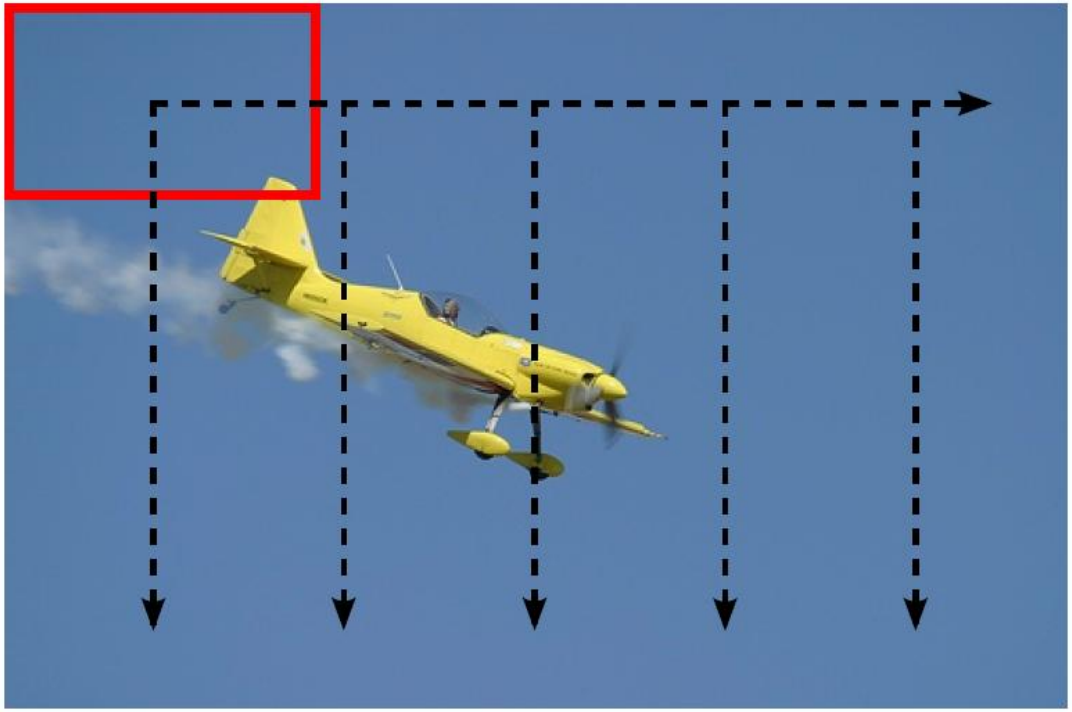
# Introduction

- Object localization
  - Where is the object located?

# General Approach

- Sliding window approach



# Sliding Window Approach

- Current state of the art
  - Slide window over an image
  - Classify each window
  - Disadvantage: slow
- Approaches to increase speed
  - Skip pixels/positions
  - Efficient Subwindow Search

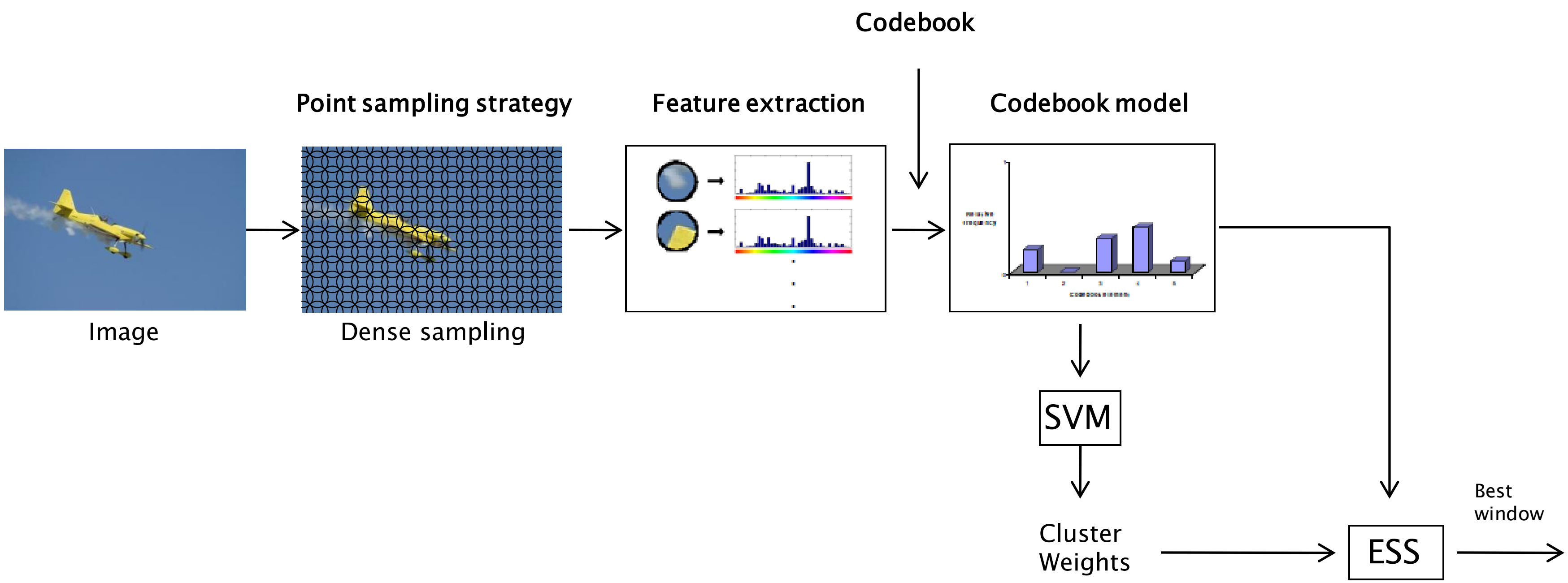
# Efficient Subwindow Search

- Developed by Lampert et al.
- Relies on branch-and-bound scheme
- Parameter space
  - Set of all possible rectangles in an image
  - Represented through [T, B, L, R]
  - T = [t_low, t_high] etc.
- Bounding function
  - Bounds the output of the classifier
- Search is stopped when most promising set contains only one rectangle

# Dataset

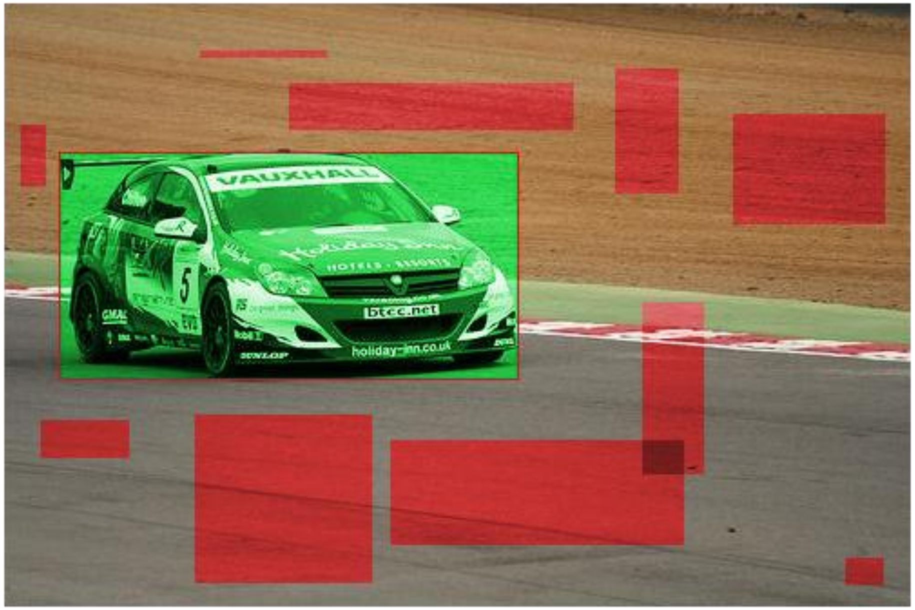
- PASCAL Visual Object Classes 2007
  - 9963 images, 20 object categories
  - Clutter and occlusion

# Bag-of-words-ESS System

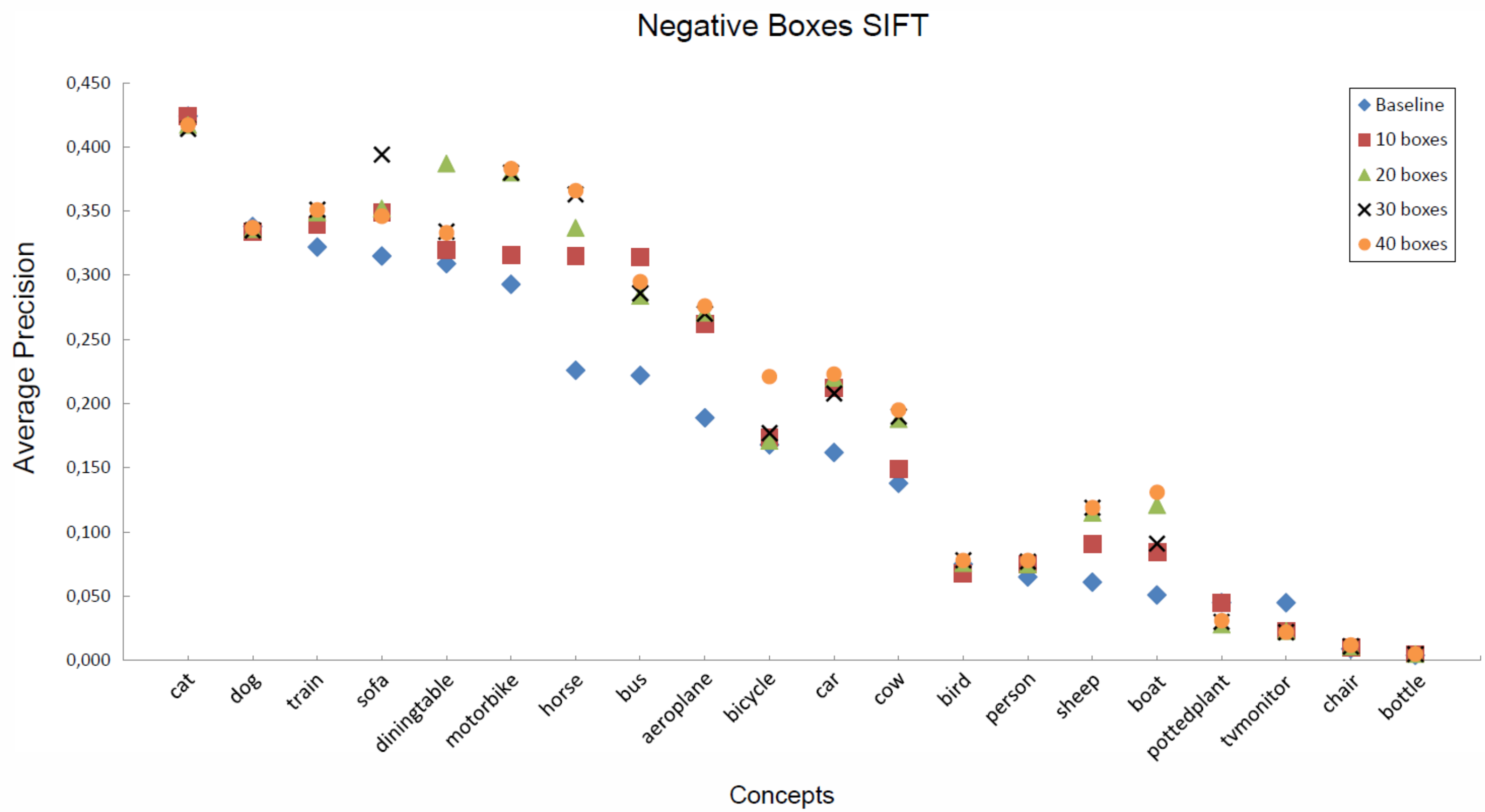


# Experiments

- Negative outside boxes (Boxbags5)
  - Train on inside box
  - Train on n random generated outside boxes
  - Used: n=10, n=20, n=30, n=40



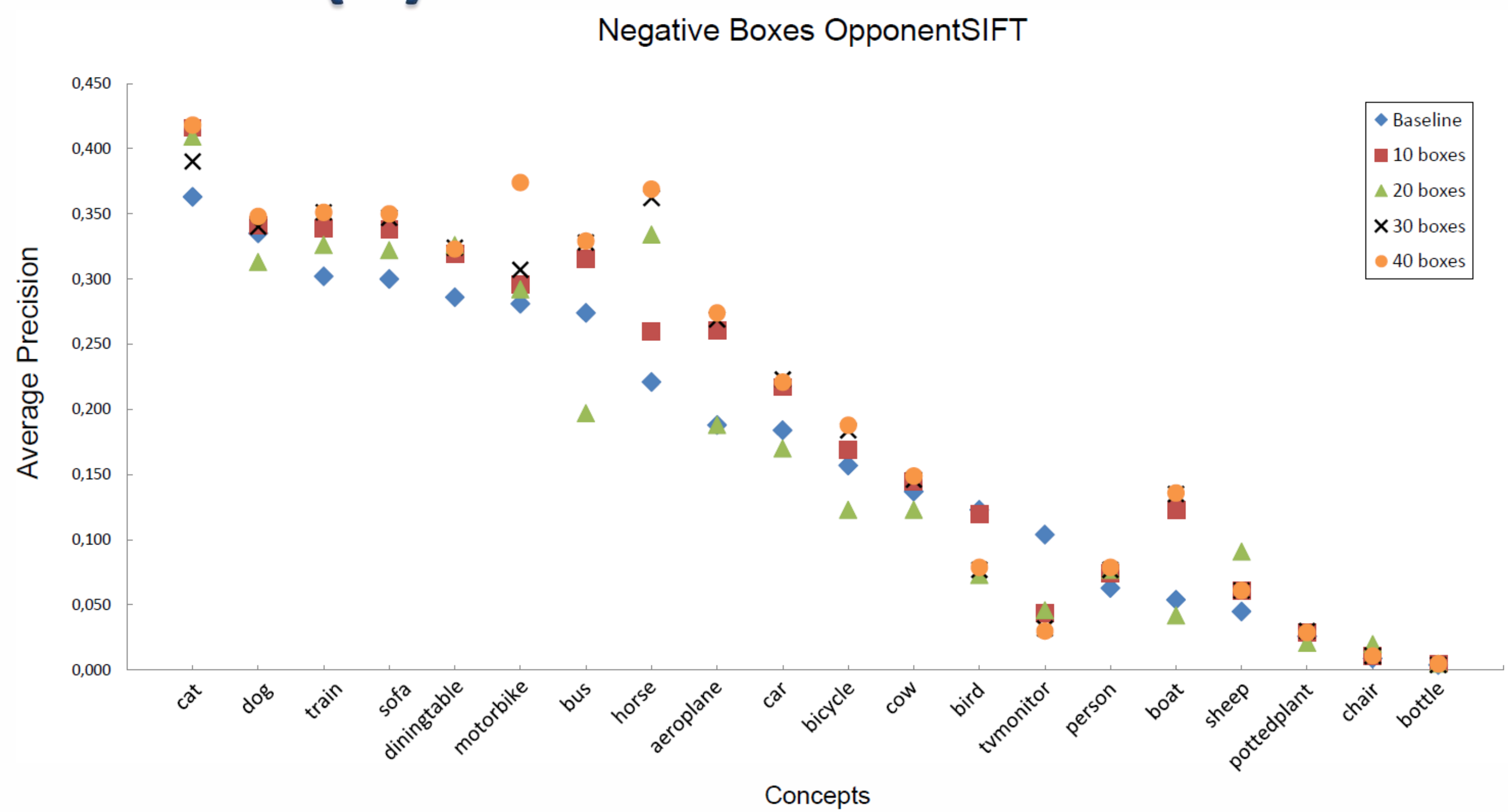
# Results




-Significant increase when compared to the baseline
-Cat: More variance in environment
-Aeroplane: Environment has negative influence on localization (sky)

| Mean Average Precision | |
|---|---|
| SIFT Baseline | 0.173 |
| SIFT 10 Negatives | 0.195 |
| SIFT 20 Negatives | 0.207 |
| SIFT 30 Negatives | 0.207 |
| SIFT 40 Negatives | 0.211 |

# Results (2)




-Significant increase when compared to baseline
-Boat: Environment has negative influence on localization (water, sky)
-Same can be observed for the horse concept

| Mean Average Precision | |
|---|---|
| OpponentSIFT Baseline | 0.173 |
| OpponentSIFT 10 Negatives | 0.194 |
| OpponentSIFT 20 Negatives | 0.175 |
| OpponentSIFT 30 Negatives | 0.200 |
| OpponentSIFT 40 Negatives | 0.206 |

# Conclusion

- Color information can increase localization performance
  - Depends on point sampling methods
- Fusion of systems does not seem to improve localization performance
  - Depends on scaling methods

# Overview

### *PART I (low-level)*

1. ***Reflection Models***
   - Dichromatic reflection model
2. ***Photometric/Color Invariance***
   - At the pixel
   - Instability handling
   - Color differential structure
3. ***Color Constancy***
   - Low-level
   - High-level
4. ***Saliency and Color Boosting***
   - Itti and Koch model
   - Color boosted

### *PART II (higher Level)*

1. ***Interest point detection***
   - Harris Laplace
   - Color boosted
2. ***Descriptors***
   - SIFT
   - Extension to color
3. ***Object recognition (VOC/TRECVid)***
   - Dense and point sampling
   - Code book generation
   - Results
4. ***Applications***
   - Tracking in video
   - Object replacement
   - Emotion recognition
   - Head pose estimation

# Object tracking

# Tracking

- **Background clutter**: the presence of other objects or non-informative patterns in the image complicates the detection of the right object.
- A **dynamic background**: moving camera.
- **Illumination change**: change in direction or intensity of light source, shadow…
- **Viewpoint change**: change of object pose or camera position.
- **Occlusion**: the target disappears partially or completely behind another object for a while.

# Standard tracking algorithms

- **Background subtraction**.
- **Template tracking**:
  - SSD matching.
  - Correlation matching.
- **Mean-shift tracking**

# Standard tracking algorithms

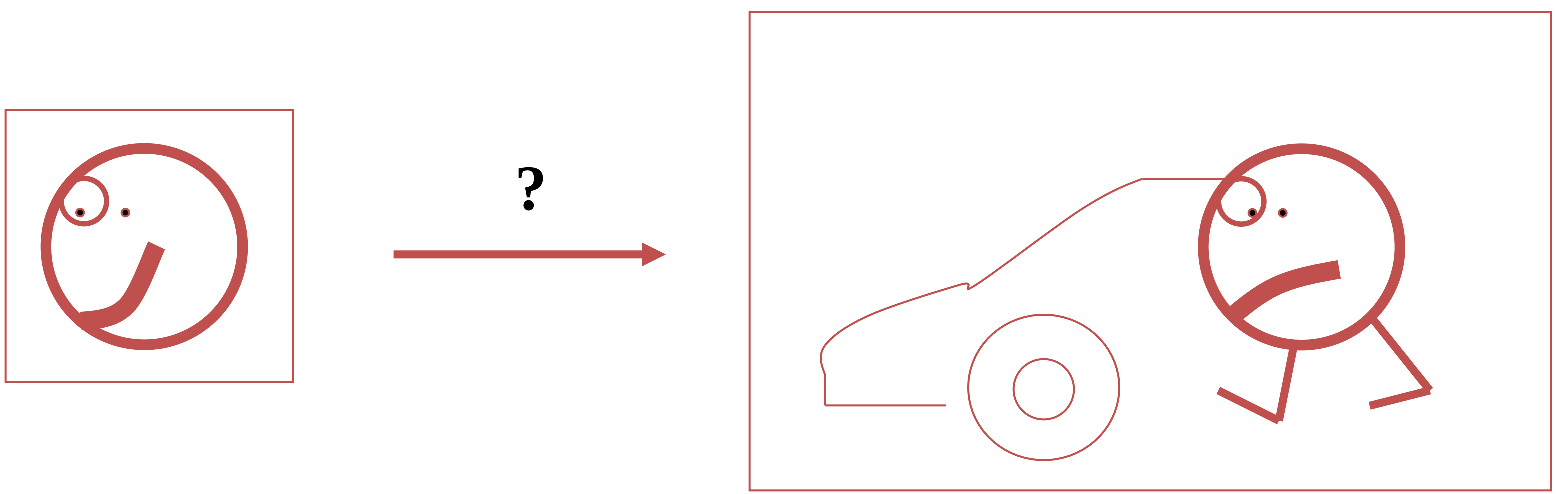
Template tracking

Mean-shift tracking

Tracking Objects based on Foreground-Background Separation

# Template-based Tracking

- Tracking consists in searching for the target object in a frame by comparing with a **template** image.
- We assume that the template is fixed and given in advance.




Template image
$T(\mathbf{x})$

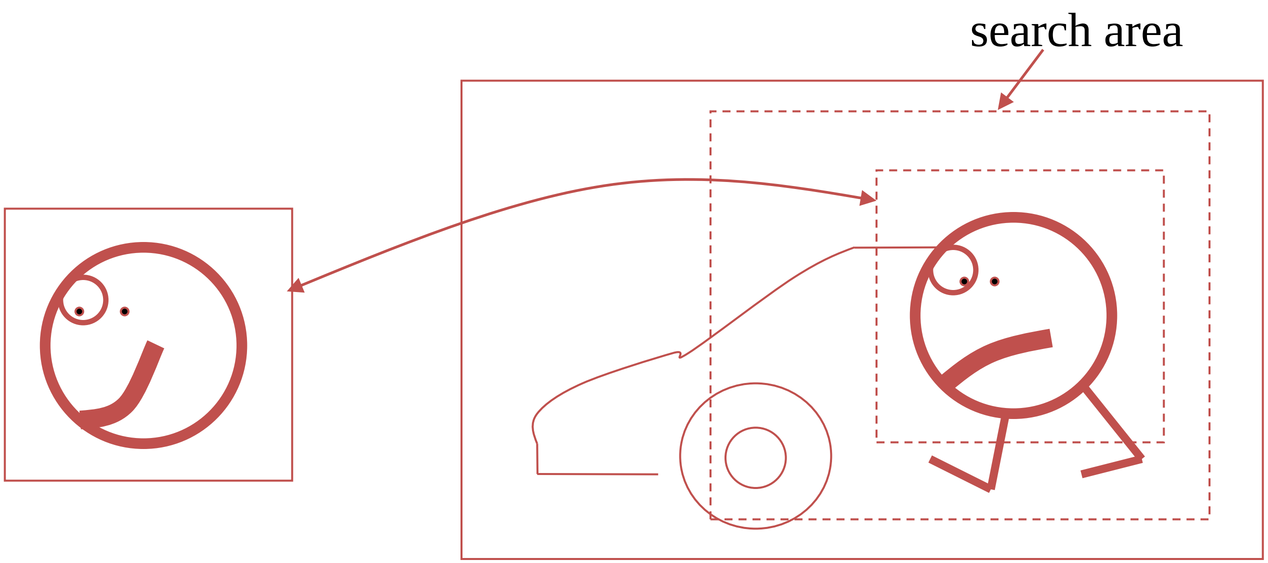
$I(\mathbf{x},\mathbf{t})$

# Motion Models

- The type of transformation φ specifies the type of object motion that the tracker is able to deal with.
  - Translation: $\varphi(\mathbf{x};\, \mathbf{y}) = \mathbf{x} + \mathbf{y}$
  - Rotation:
$$\varphi_1 = x_1 \cos y - x_2 \sin y$$
$$\varphi_2 = x_1 \sin y + x_2 \cos y$$
  - Scaling:
$$\varphi_1 = y x_1$$
$$\varphi_2 = y x_2$$
  - Affine:
$$\varphi_1 = y_1 + y_2 x_1 + y_3 x_2$$
$$\varphi_2 = y_4 + y_5 x_1 + y_6 x_2$$

# Search

- Align the template with every possible candidate region in the image, and find the most similar candidate according to a **similarity measure**.
- We search the target only in an area around the previous position exploiting general knowledge that the object won't have moved far.



# SSD and correlation

- SSD is short for sum-of-squared-difference:

$$D(\mathbf{y}) = \sum_{\mathbf{x}\in\Omega} [I(\mathbf{x}+\mathbf{y}) - T(\mathbf{x})]^2 \rightarrow \min_{\mathbf{y}}$$

- A simpler similarity measure is the (unnormalized) cross-correlation:

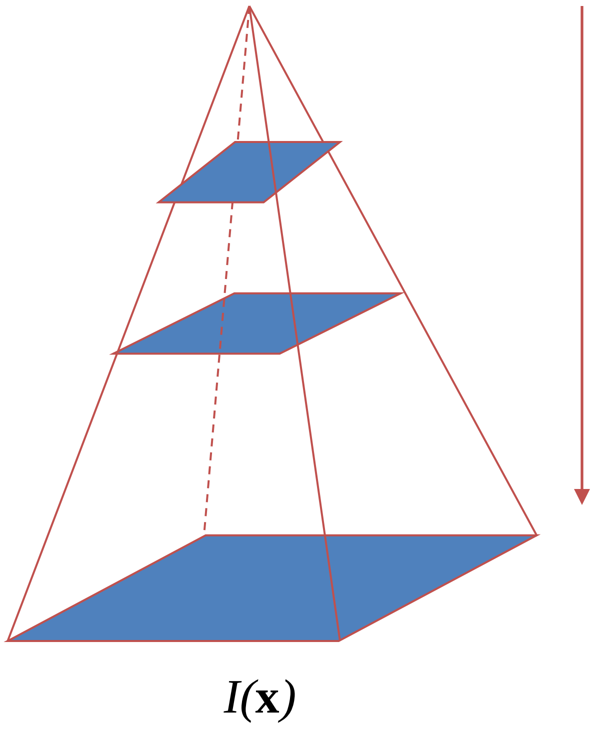
$$C(\mathbf{y}) = \sum_{\mathbf{x}\in\Omega} I(\mathbf{x}+\mathbf{y})T(\mathbf{x}) \rightarrow \max_{\mathbf{y}}$$

# Exhaustive search

- Calculate SSD for every $\mathbf{y}$ in a search window and choose the position with the least SSD.
- Strengths: robustness and simplicity in implementation.
- Weaknesses:
  - Computations could be time-consuming in case of a large search window.
  - Only suitable for translation.

# Coarse-to-fine strategy

- Propagate the search results through different resolution levels using image pyramids.
- First search for the target in a low resolution and then use the result as initial point for the higher resolution.
- Able to overcome the issues of complexity and local minima:
  - Reduce complexity since images at low resolution have small sizes
  - At low resolution local minima are smoothed over.



Template tracking

# Mean-shift tracking

Tracking Objects based on Foreground-Background Separation

# Mean-shift tracking

- Features:
  - Target detection is performed by matching **weighted histograms**.
  - Very fast in comparison with SSD or correlation trackers,.

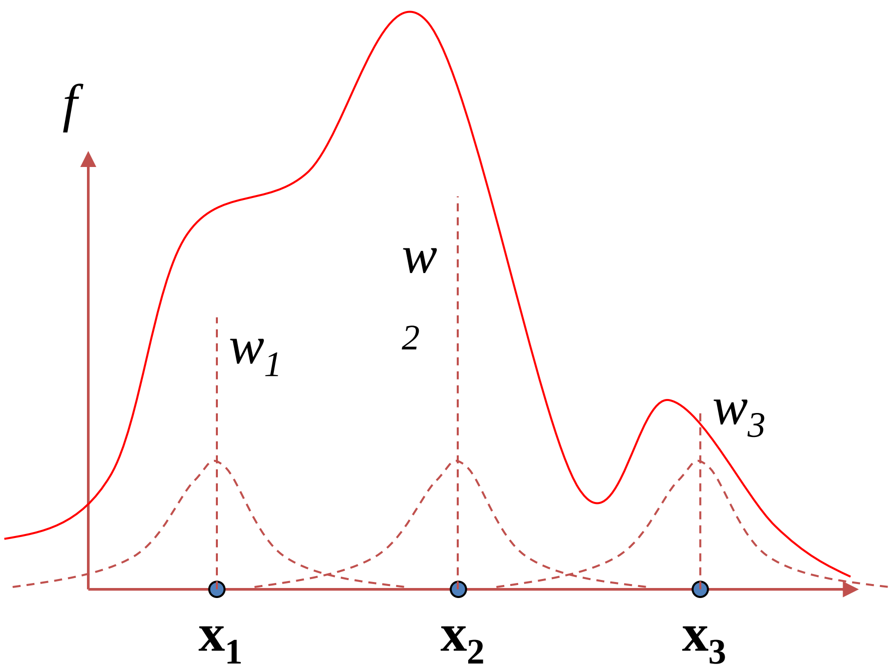
- Reference: Comaniciu et al. *Real time tracking of Non-Rigid Objects using Mean Shift*, In CVPR 2000.

# Mean-shift algorithm

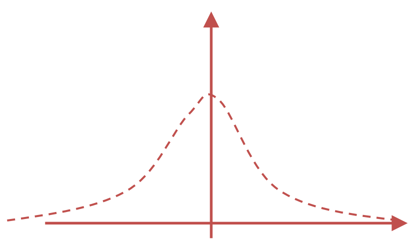
- The mean-shift algorithm finds a local maximum of a density function of the form:

$$f(\mathbf{y}) = \sum_i w_i K\left(\frac{|\mathbf{y}-\mathbf{x}_i|^2}{\sigma}\right)$$

- where $K$ is the local kernel.



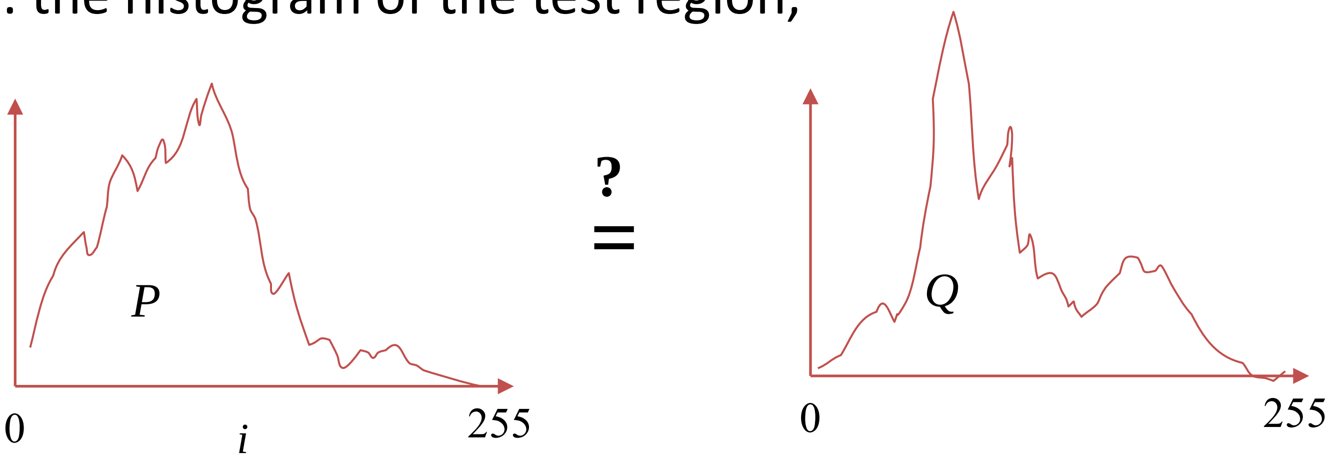

Gaussian kernel:



$$K(|\mathbf{x}|^2) = (2\pi)^{-d/2}\exp\left(-|\mathbf{x}|^2/2\right)$$

# Similarity measure

- *P(i)*: the template histogram,
- *Q(i;$\mathbf{y}$)*: the histogram of the test region,




- *The Bhattacharyya coefficient* can measure the similarity between two distributions:

$$r(\mathbf{y}) = r(P, Q(\mathbf{y})) = \sum_{i=0}^{255} \sqrt{P(i)Q(i;\mathbf{y})} \rightarrow \max_{\mathbf{y}}$$

# Color-based object tracking

# Player tracking

# Player tracking with occlusion

# Player tracking with occlusion

Template tracking

Mean-shift tracking

# Tracking Objects based on Foreground-Background Separation *(Jette Bunders)*

# Basic steps

- *"On-Line Selection of Discriminative Tracking Features"*

  ***Robert Collins & Yanxi Liu, ICCV 2003***

- System consists of three phases:
  - Constructing a feature space.
  - Classification: selection of the features.
  - Tracking: tracking of objects.

# Online feature selection

- A histogram is computed of the foreground and background window.

  $H_{obj}$ and $H_{bg}$

- Probability density function is generated from the histograms:

  $$p(i) = H_{obj} / n_{obj} \quad \text{and} \quad q(i) = H_{bg} / n_{bg}$$

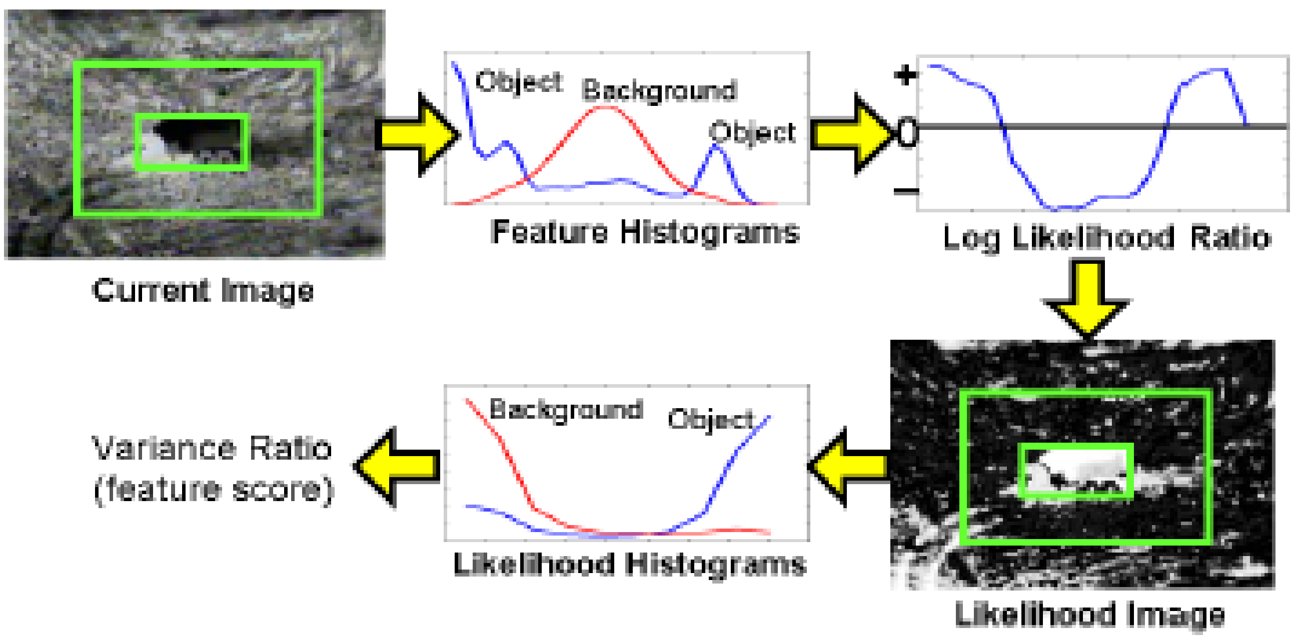
- A log likelihood histogram is computed of the pdf's of the foreground and the background according to the ratio:

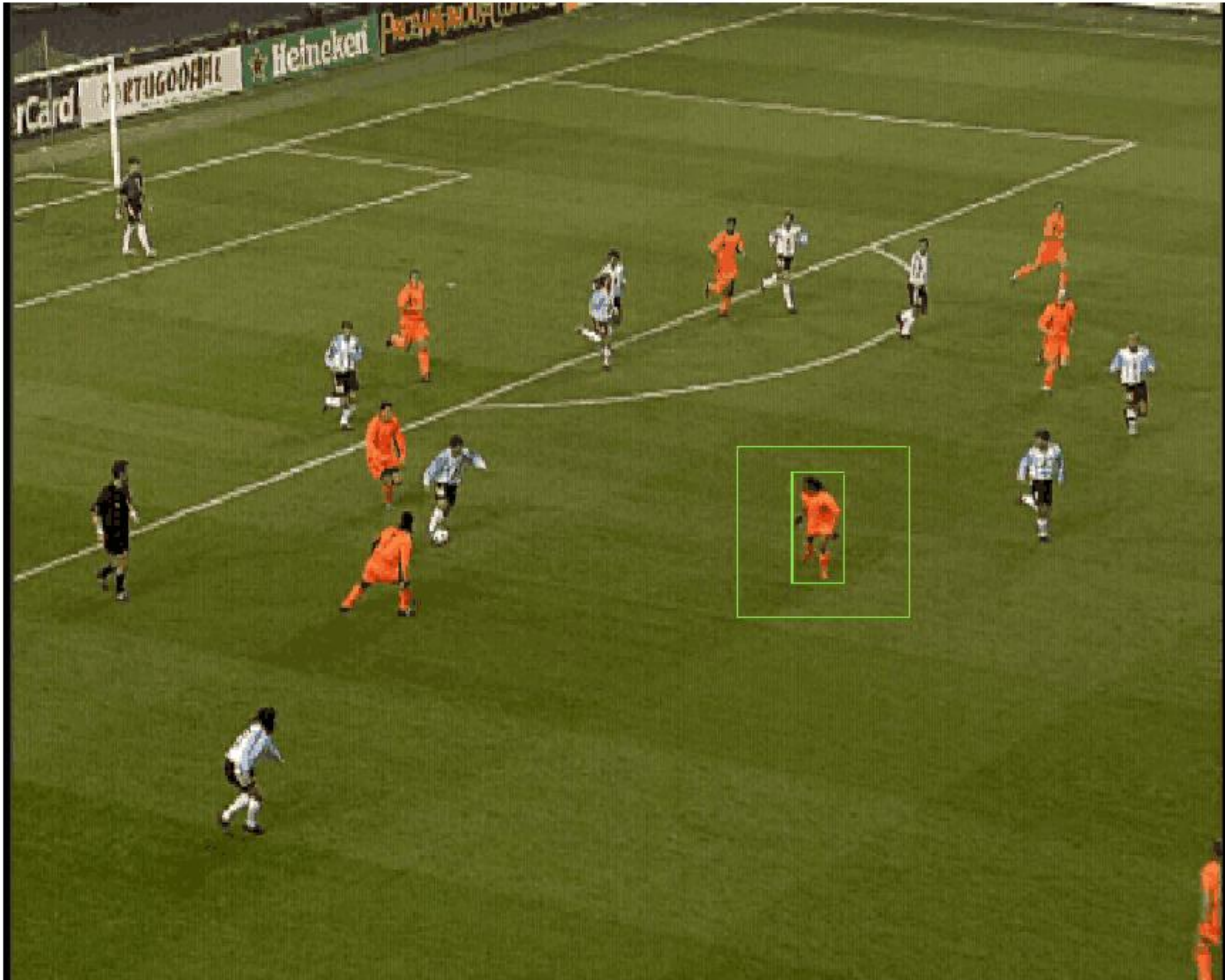
  $$L(i) = \max((\log p(i), \delta)) - \max((\log q(i), \delta))$$

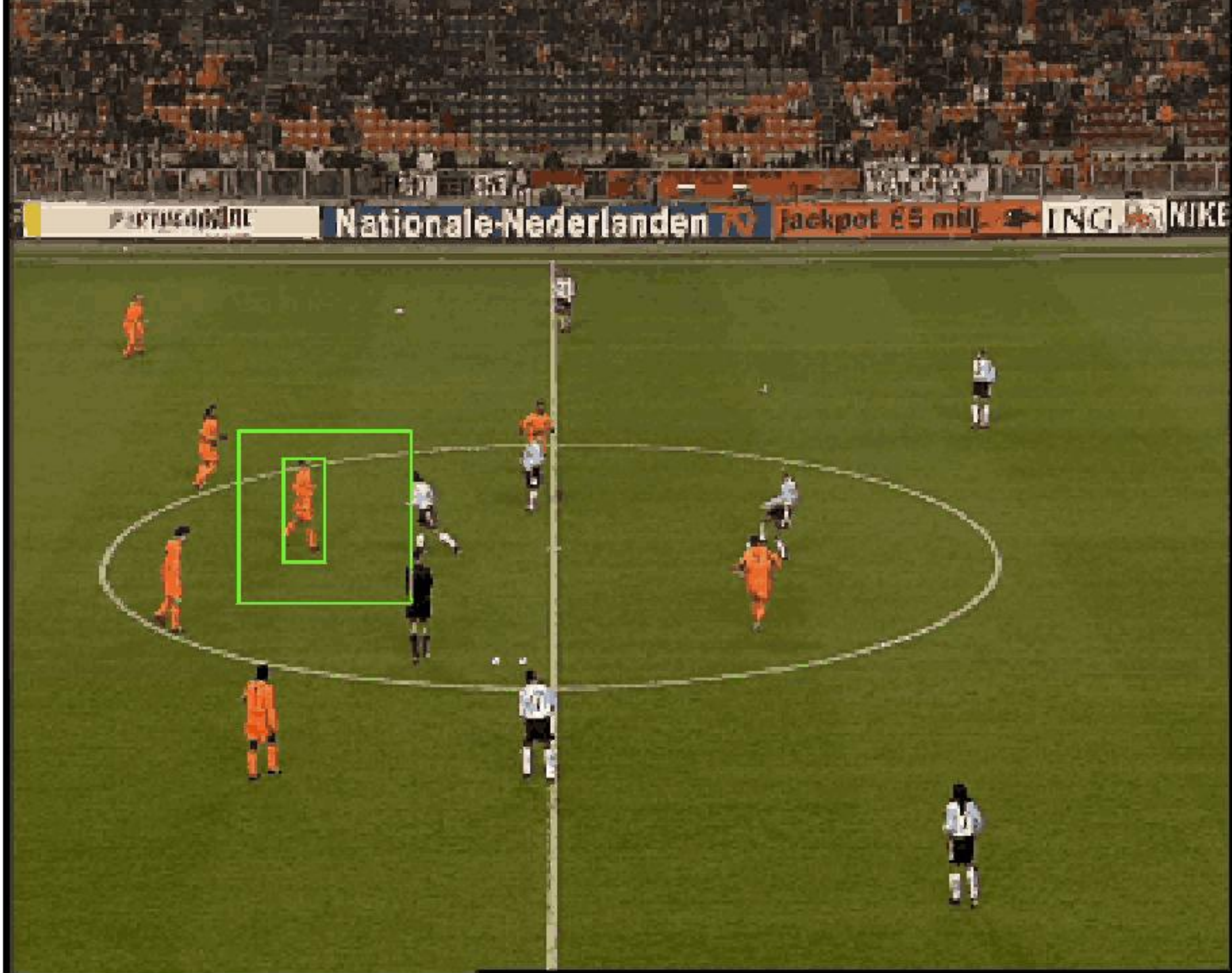
- The log likelihood contains positive values for regions corresponding to the object and negative values for the regions corresponding to the background

# Algorithm

***Figure taken from "Online Selection of Discriminating features" Collins and Liu***




Heineken

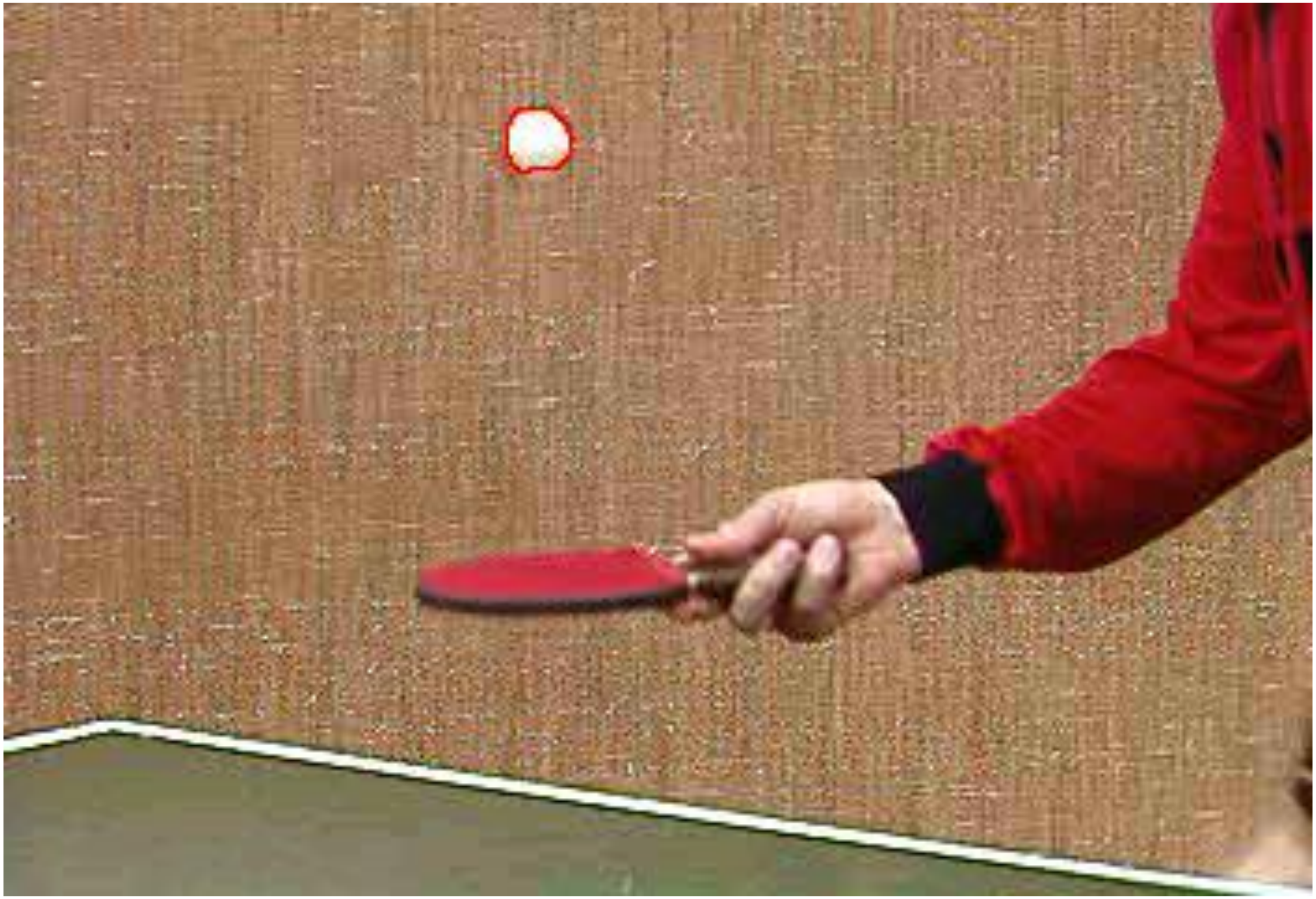

Nationale-Nederlanden
jackpot 65 milj.
ING
NIKE

# Deformable contours

# Deformable contours

[Hieu, IEEE PAMI, 2003]

# Deformable contours



# *Object replacement*

# Visual Tracking



# Visual Tracking

# Mosaics

# Visual Tracking

Feike Winkelman

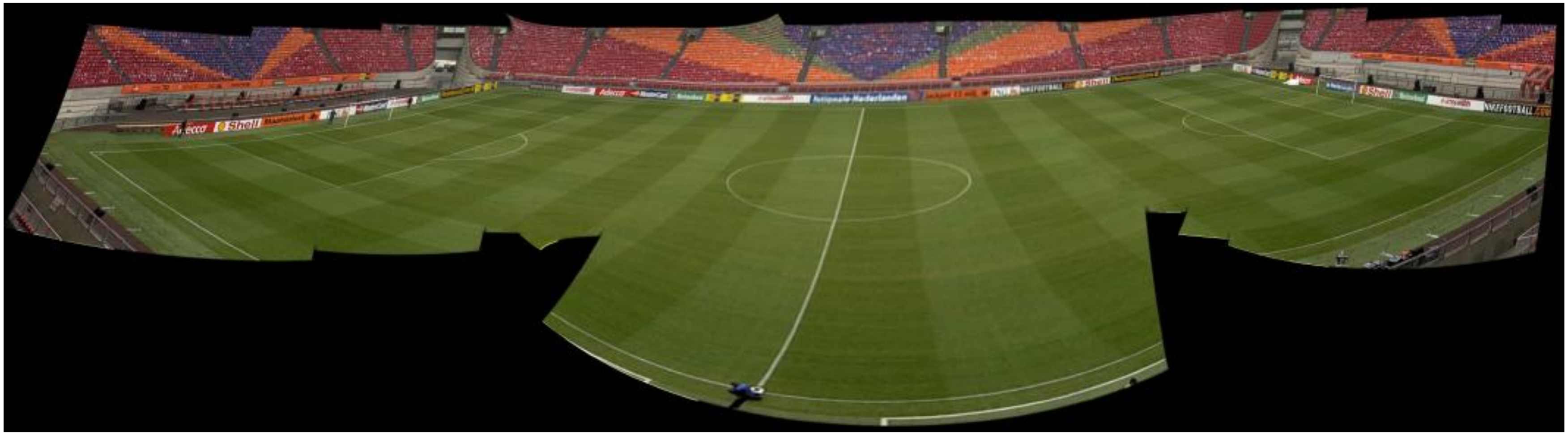
# Visual Tracking


Techniques:
- Mosaics.
- Shot and key-frame detection.
- Analysis of camera-motion.

# Motion and Visual Tracking

# Motion and Visual Tracking

Mosaic created from video



# Motion and Visual Tracking

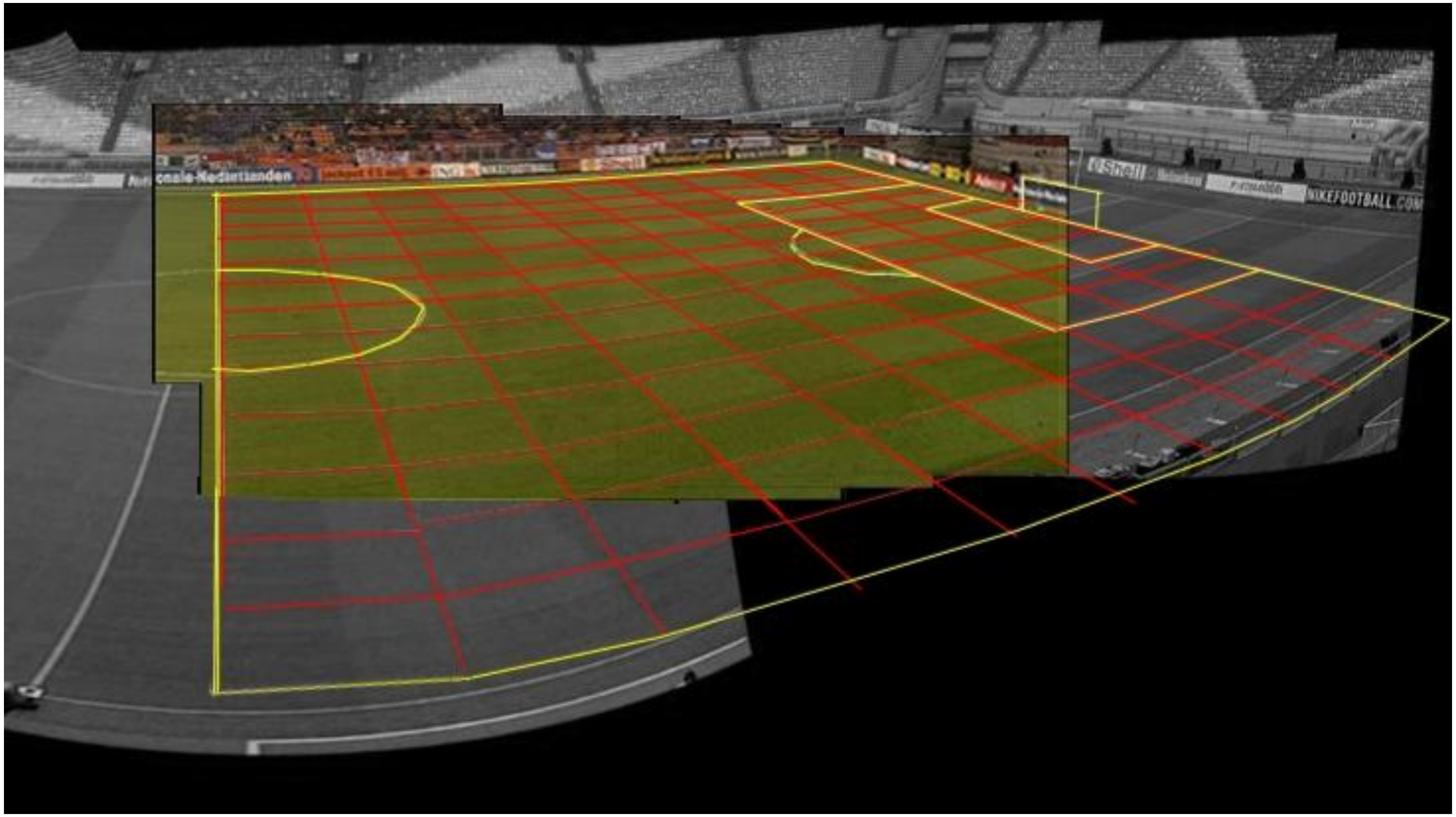
Using model for matching

# Motion and Visual Tracking

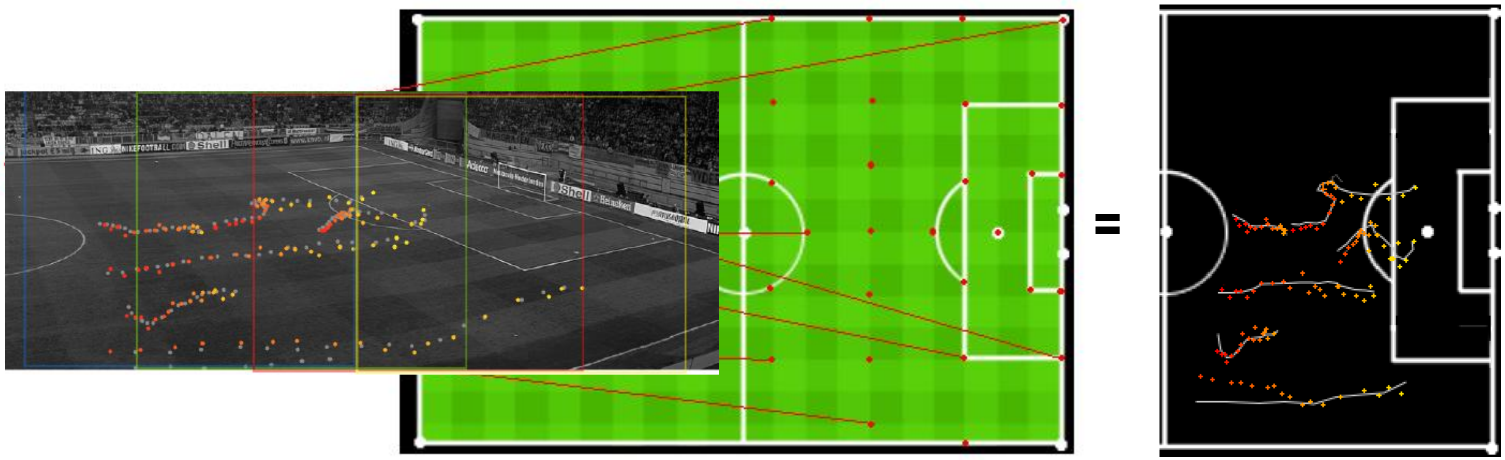

Several frames projected on the mosaic, according to their recovered registration parameters.

Showing ‘ghosts’ of players is very illustrative

# Motion and Visual Tracking

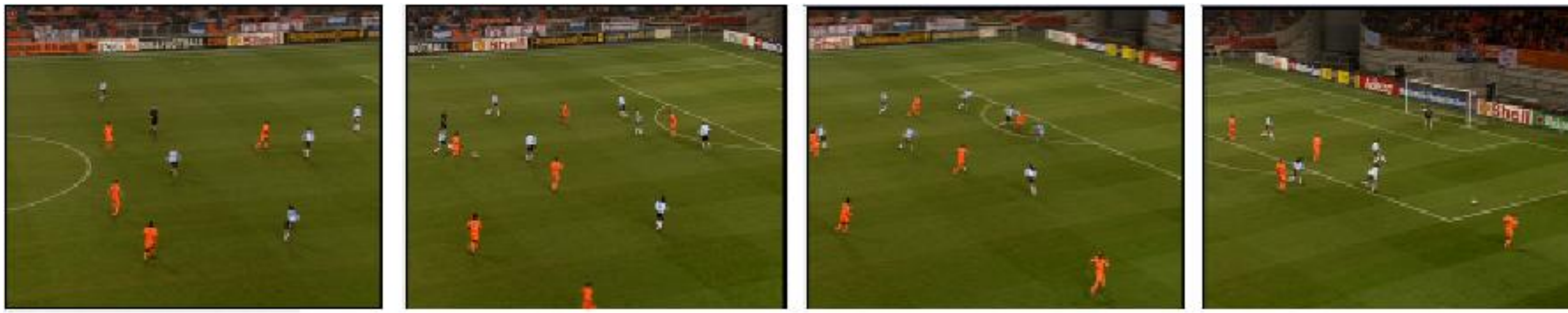


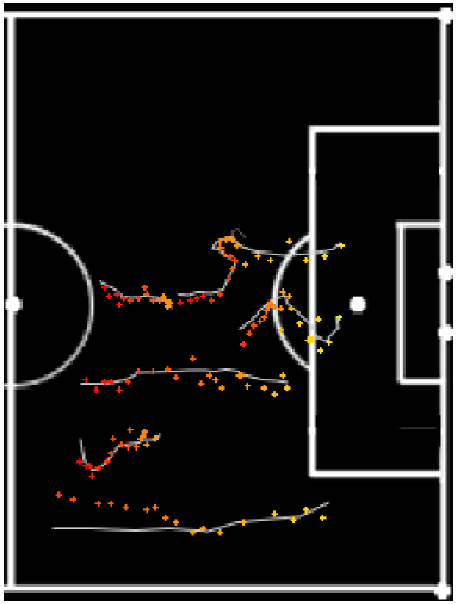
# Homography Transform Phase



- After iteratively plotting the foot-positions of each frame a trajectory plot is constructed. Distinctive or salient features are selected and mapped to the geometrically correct line-model. Finally, conversion to an orthogonal perspective using a homography is performed.

# Motion and Visual Tracking





# Motion and Visual Tracking

# Overview

## *PART I (low-level)*

1. ***Reflection Models***
   - Dichromatic reflection model
2. ***Photometric/Color Invariance***
   - At the pixel
   - Instability handling
   - Color differential structure
3. ***Color Constancy***
   - Low-level
   - High-level
4. ***Saliency and Color Boosting***
   - Itti and Koch model
   - Color boosted

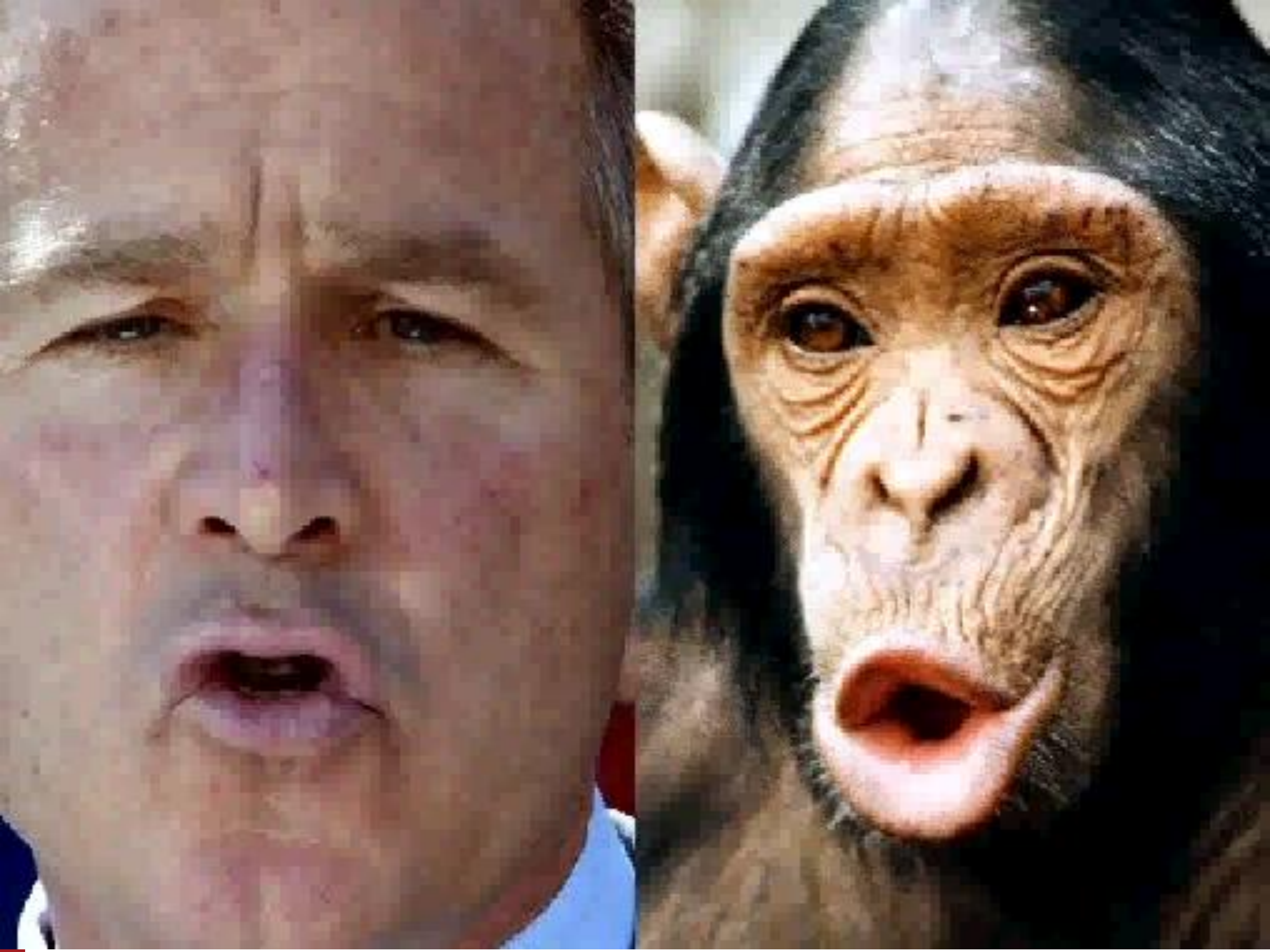
## *PART II (higher Level)*

1. ***Interest point detection***
   - Harris Laplace
   - Color boosted
2. ***Descriptors***
   - SIFT
   - Extension to color
3. ***Object recognition (VOC/TRECVid)***
   - Dense and point sampling
   - Code book generation
   - Results
4. ***Applications***
   - Tracking in video
   - Object replacement
   - Emotion recognition
   - Head pose estimation

# Facial Expression Recognition

With Nicu Sebe, Intelligent Systems Laboratory Amsterdam (ISLA), Faculty of Science, University of Amsterdam, The Netherlands

Beckman Institute at the University of Illinois, Urbana-Champaign, USA



## Angst, woede, walging, blijdschap, verrassing, verdriet en minachting: over de hele wereld vertonen gezichten dezelfde uitdrukking bij deze zeven basisemoties. Hieraan herkent u ze.

**PSYCHOLOGIEMAGAZINE.NL**

Verder oefenen? Plusabonnees maken kans op een van de twintig exemplaren van *Gegrepen door emoties*, het standaardwerk van Ekman.

**1. Walging:** rimpelingen langs de rand van de neus. De neusgaten worden smaller, de bovenlip gaat omhoog, de onderlip puilt uit, de wangen gaan omhoog. De wenkbrauwen zakken iets, waardoor soms kraalenpootjes rond de ogen te zien zijn.

**2. Verdriet:** de mondhoeken gaan naar beneden, net als de oogleden. De ogen staan minder scherp.

**3. Boosheid:** gefronste wenkbrauwen, samengeperste lippen en opeengeklemde kaken maken dit de makkelijkst herkenbare emotie.

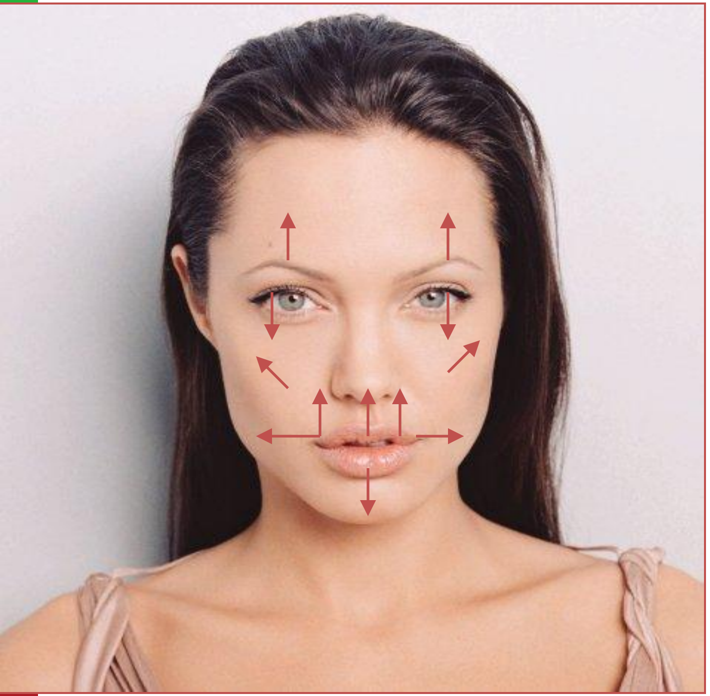
**4. Angst:** de ogen worden groter, de mondhoeken gaan naar achteren, de mond is een klein beetje geopend. Er zijn een paar rimpelingen in het voorhoofd te zien; de wenkbrauwen gaan omhoog en naar elkaar toe.

**5. Verrassing:** de wenkbrauwen gaan omhoog zonder naar elkaar toe te trekken, zoals bij angst het geval is. De ogen worden groter en ronder. In extreme gevallen valt de mond open van verbazing.

**6. Minachting:** één mondhoek is aangespannen en gaat een beetje omhoog, net als de kin, maar de neus is niet opgetrokken zoals bij walging.

**7. Blijdschap:** beide mondhoeken gaan omhoog. Een echte glimlach is te onderscheiden van een nepglimlach doordat de ogen ook meedoen: de wenkbrauwen gaan naar beneden terwijl de wangen omhoog gaan, evenals de huid vlak onder de ogen.

# Facial Expression Recognition



12 facial motion measurements

vertical movement of the lips

horizontal movement of the mouth corners

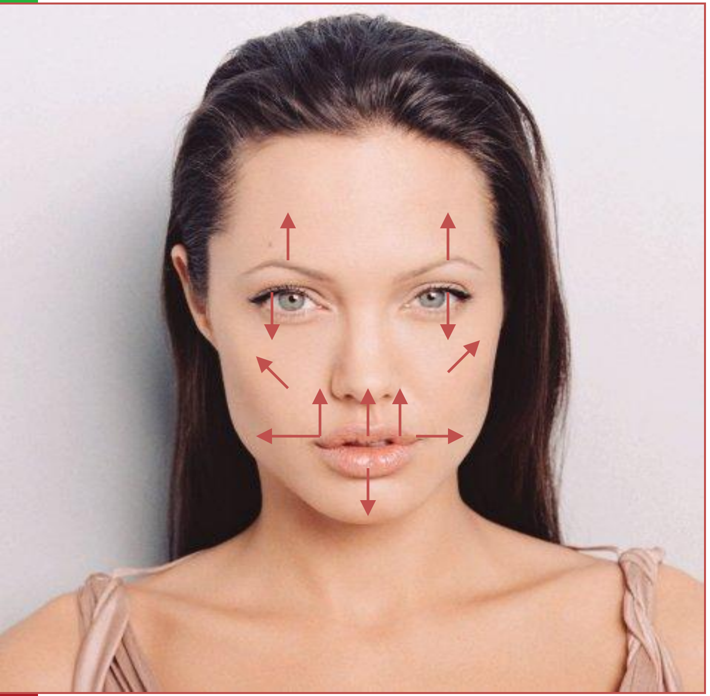
vertical movement of the mouth corners

vertical movement of the eye brows

lifting of the cheeks

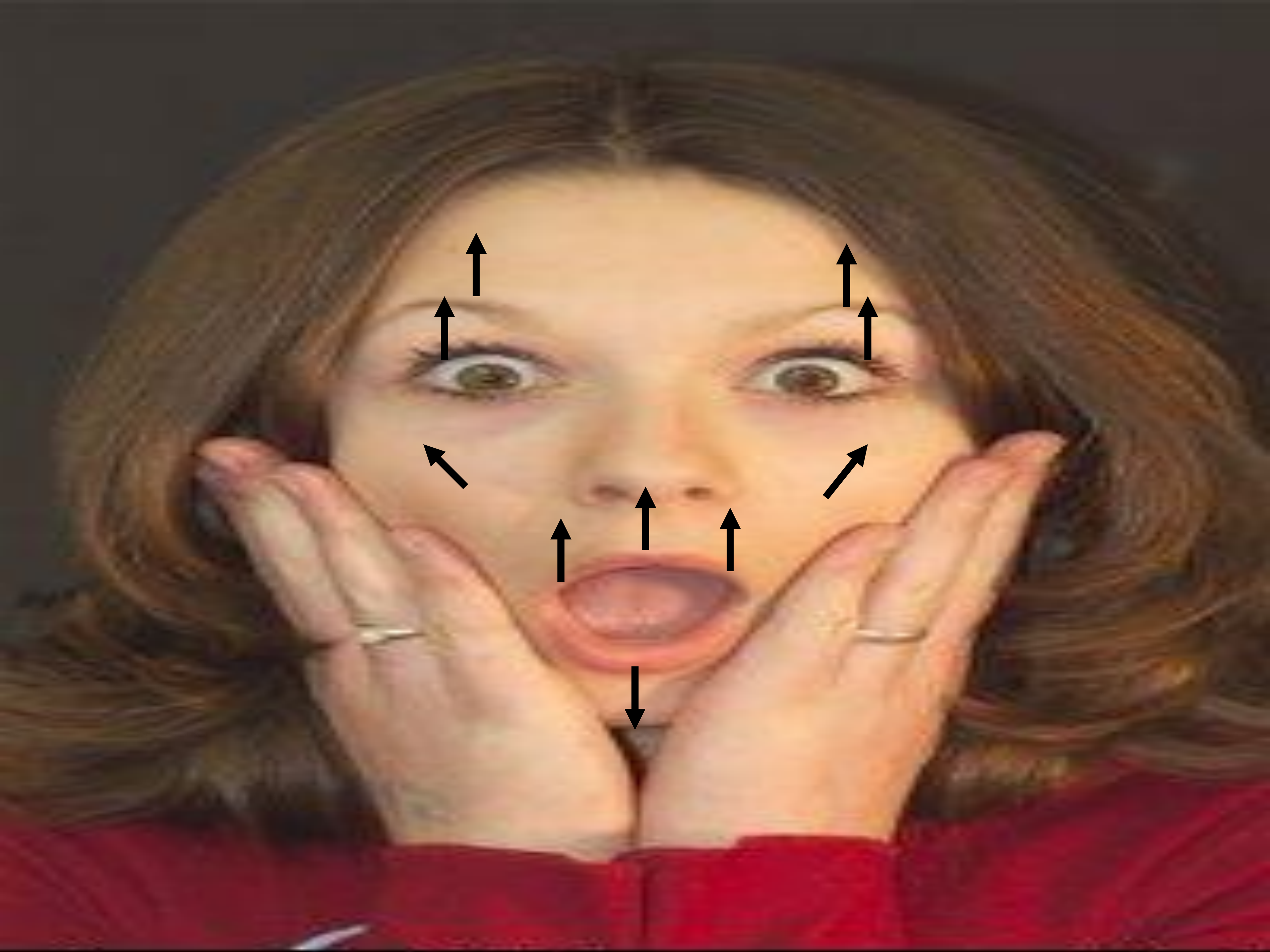
blinking of the eyes

# Facial Expression Recognition

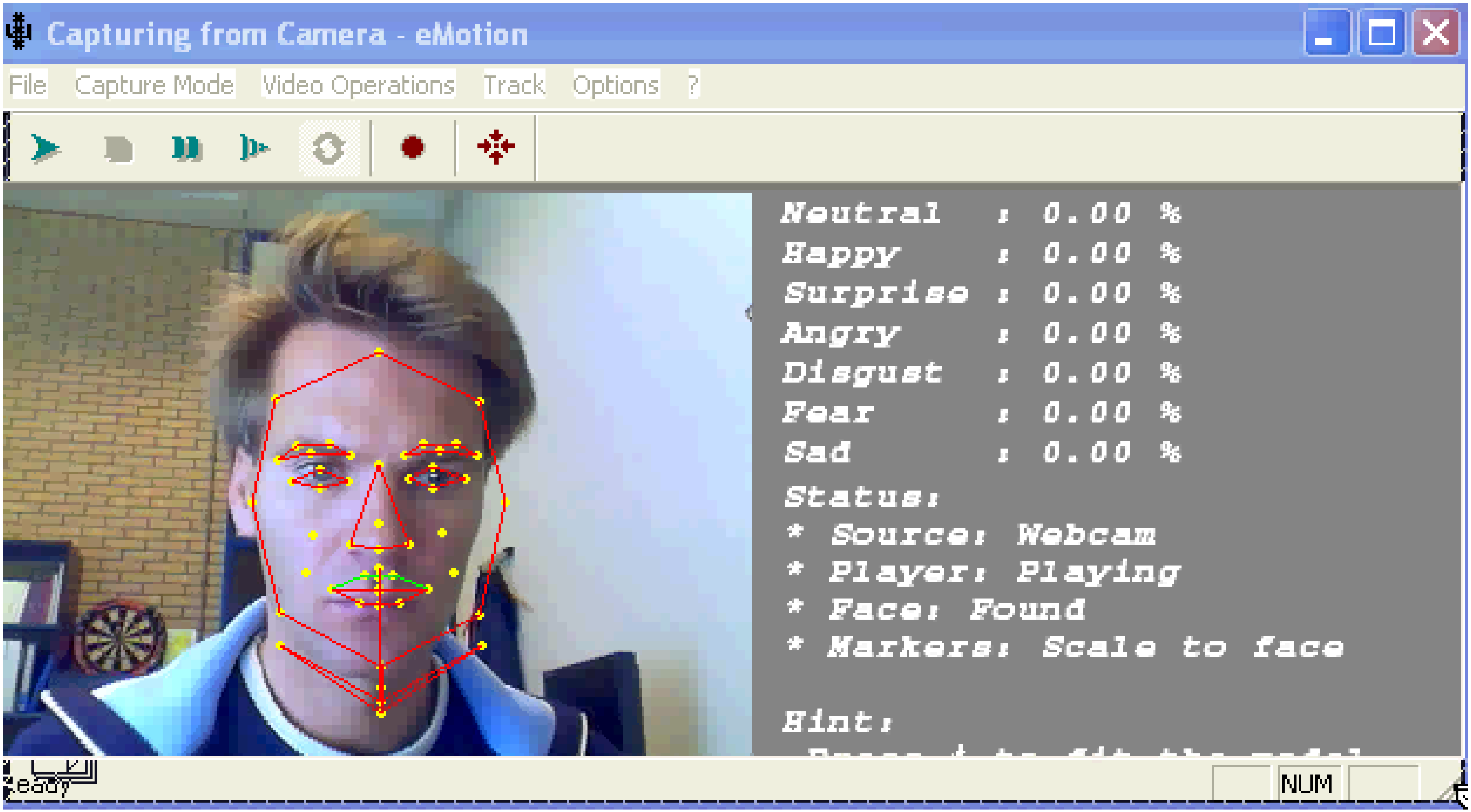


We use 12 facial features = 12 facial motion measurements

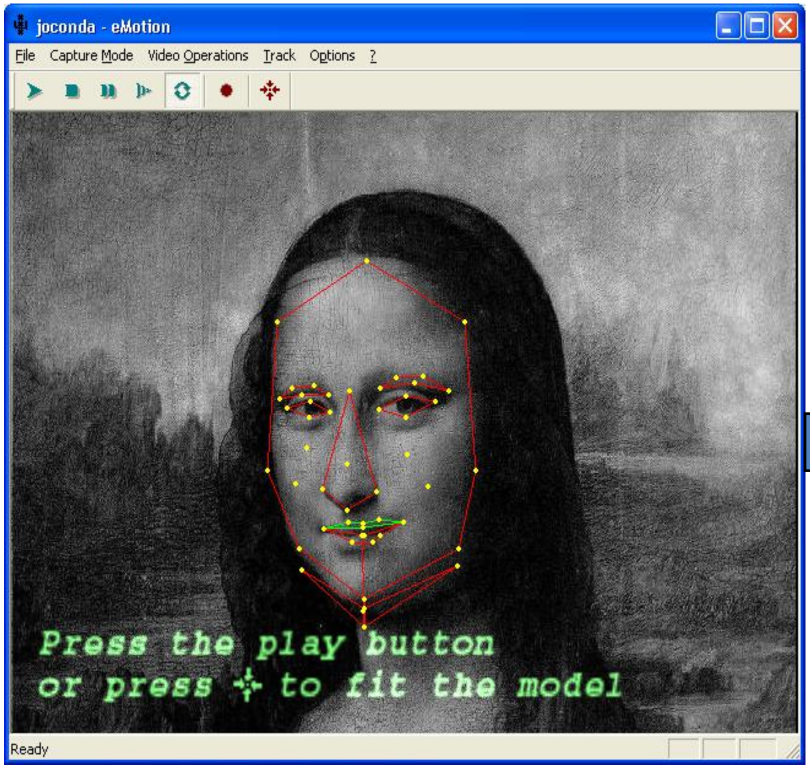
The combination of these features define the 7 basic classes of facial expression we want to classify: ***Neutral, Happy, Anger, Disgust, Fear, Sad, Surprise***

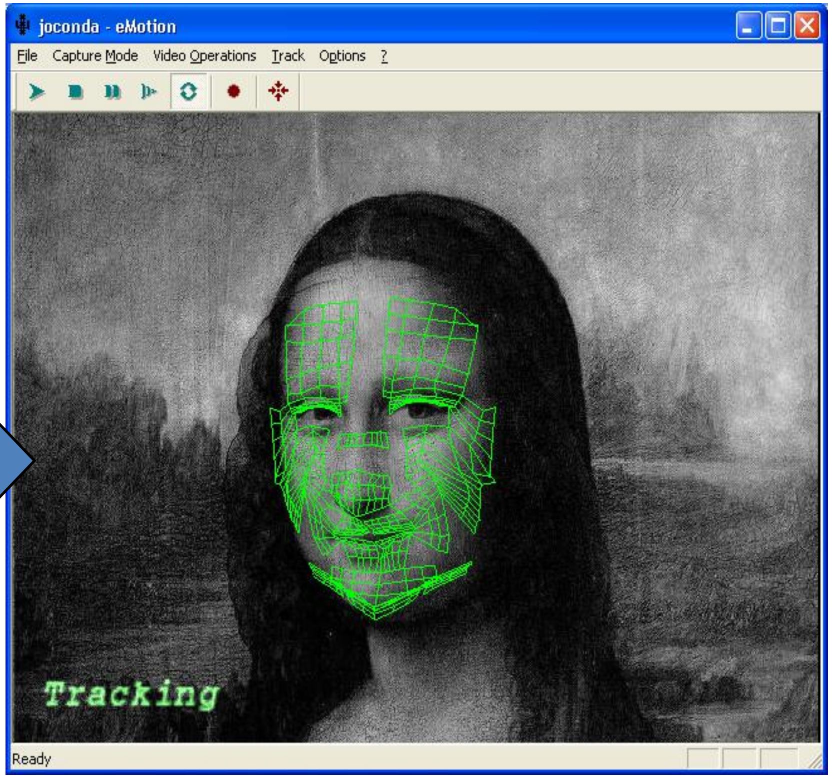


# Facial Expression Recognition

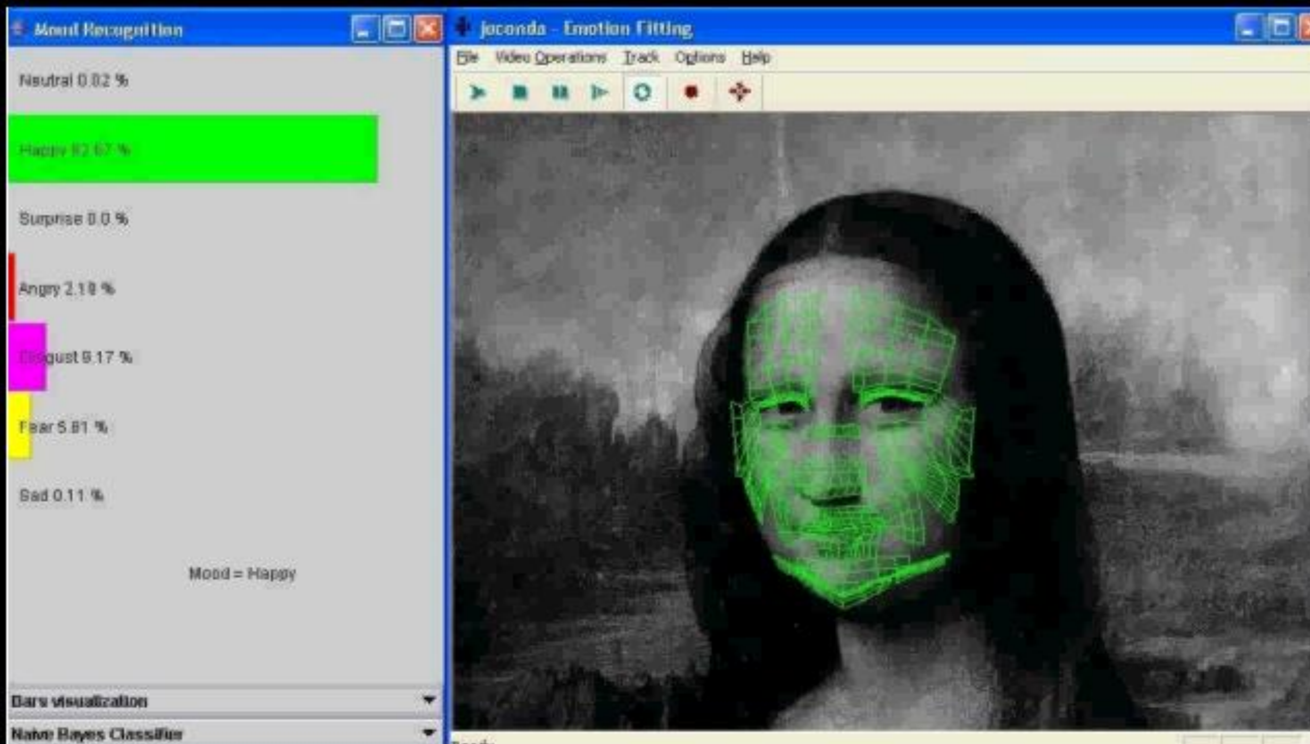


# Facial Expression Recognition

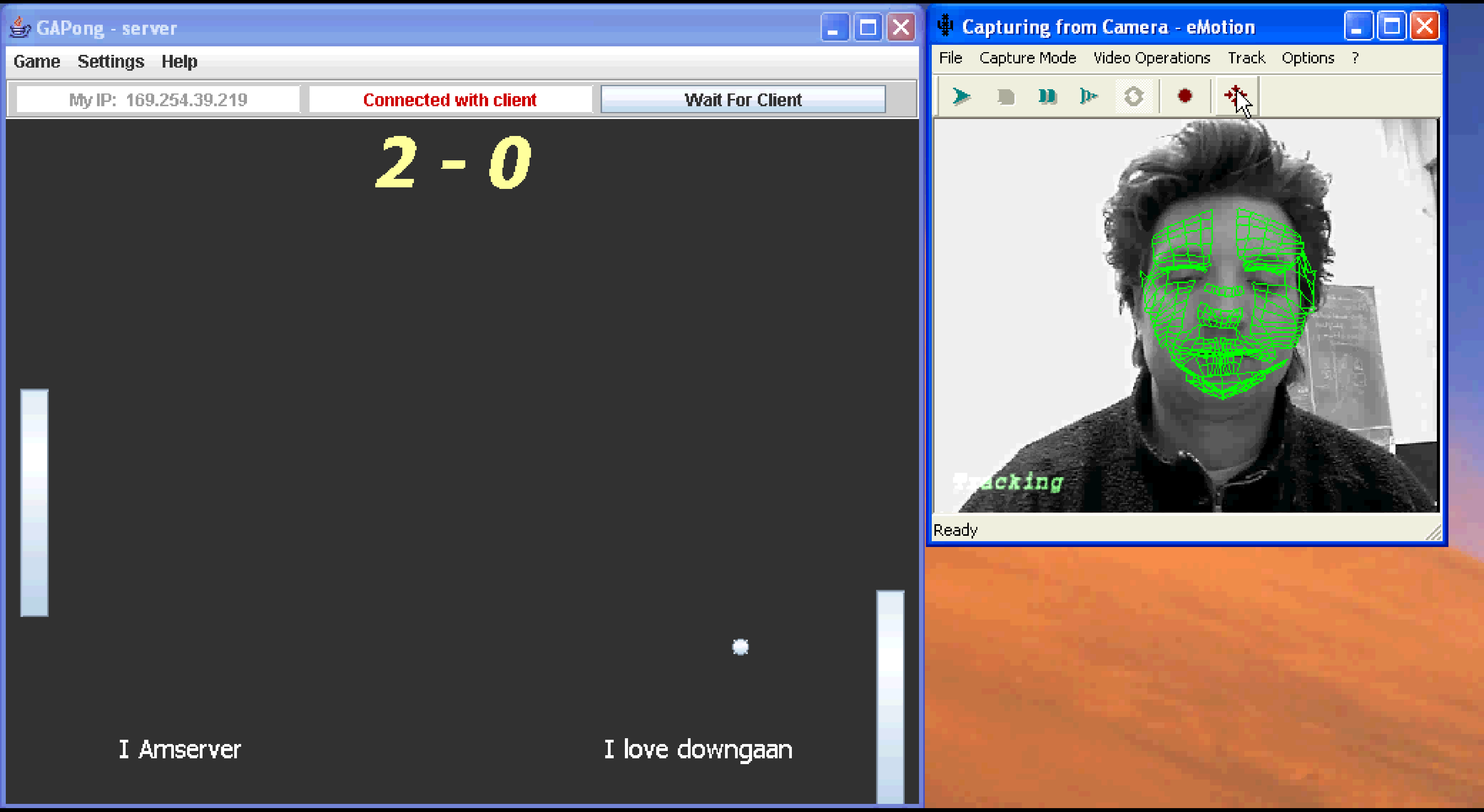


# Facial Expression Recognition: CNN

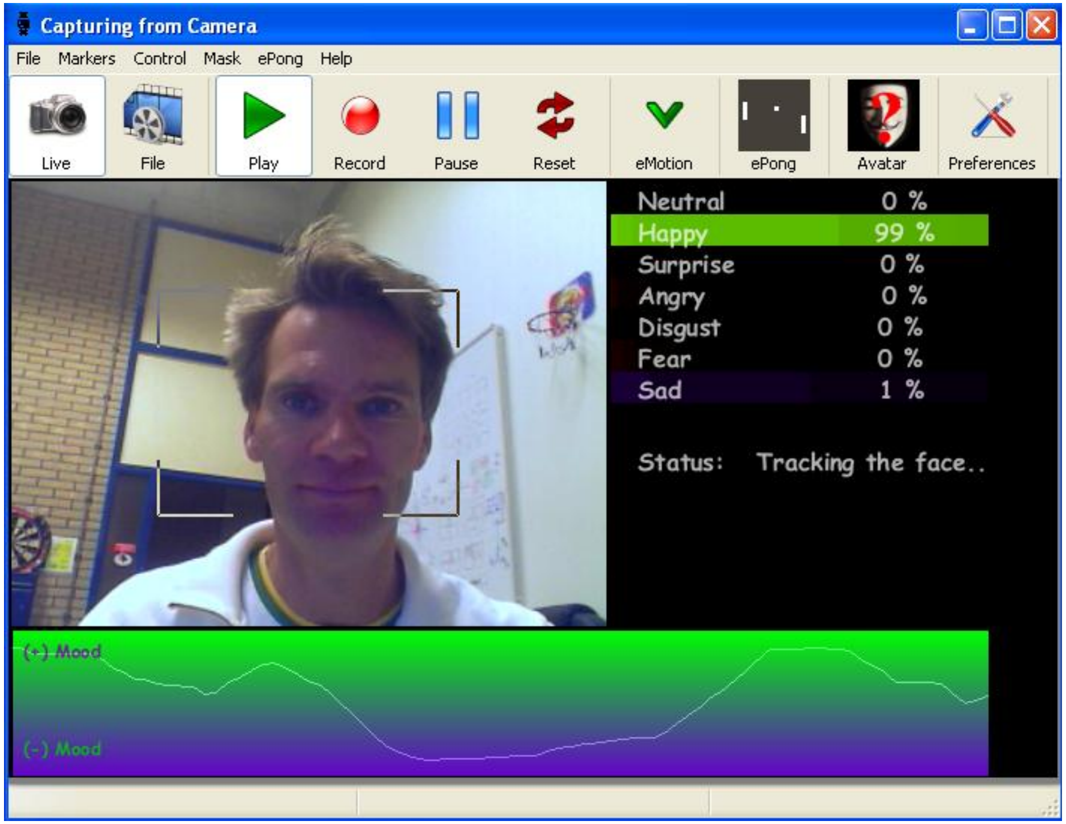

Mood Recognition
Neutral 0.02 %
Surprise 0.0 %
Angry 2.18 %
Fear 5.81 %
Sad 0.11 %
Mood = Happy
Bars visualization
Naive Bayes Classifier
Joconda - Emotion Fitting
File
Video Operations
Track
Options
Help

Glad or Sad
[ Emoties aantoonbaar gemaakt ]
• Home
• Upload foto
• Stem op foto
• Foto galerij
• Over
• Contact
Glad or Sad is een samenwerking tussen
ilse media
Universiteit van Amsterdam
Stap 1
Upload een foto.
Stap 2
Laat zoveel mogelijk mensen stemmen op jouw foto.
Stem
Aantal stemmen: 15%
Stap 3
Bekijk de resultaten in de foto galerij.
Analyse resultaten
Vrolijk 91%
Verrast 8%
Boos 0%
Walging 0%
Angstig 1%
Droevig 0%
Verfijnen / Sorteren
Selecteer een categorie
• Politici
• Sporters
• Schoonmoeders
• Overige
Sorteer op
• Datum
• Meest vrolijk
• Meest verrast
• Meest boos
• Meeste walging
• Meest angstig
• Meest droevig
Selecteer een tag
Acda Balkenende Bangebroek Duh Gezicht Glimlach Huilen Lachen Man Smile Voetbal Vrouw
Change language to
[ English ]


GAPong - server
Game Settings Help
My IP: 169.254.39.219
Connected with client
Wait For Client
2 - 0
I Amserver
I love downgaan
Capturing from Camera - eMotion
File Capture Mode Video Operations Track Options ?
Ready

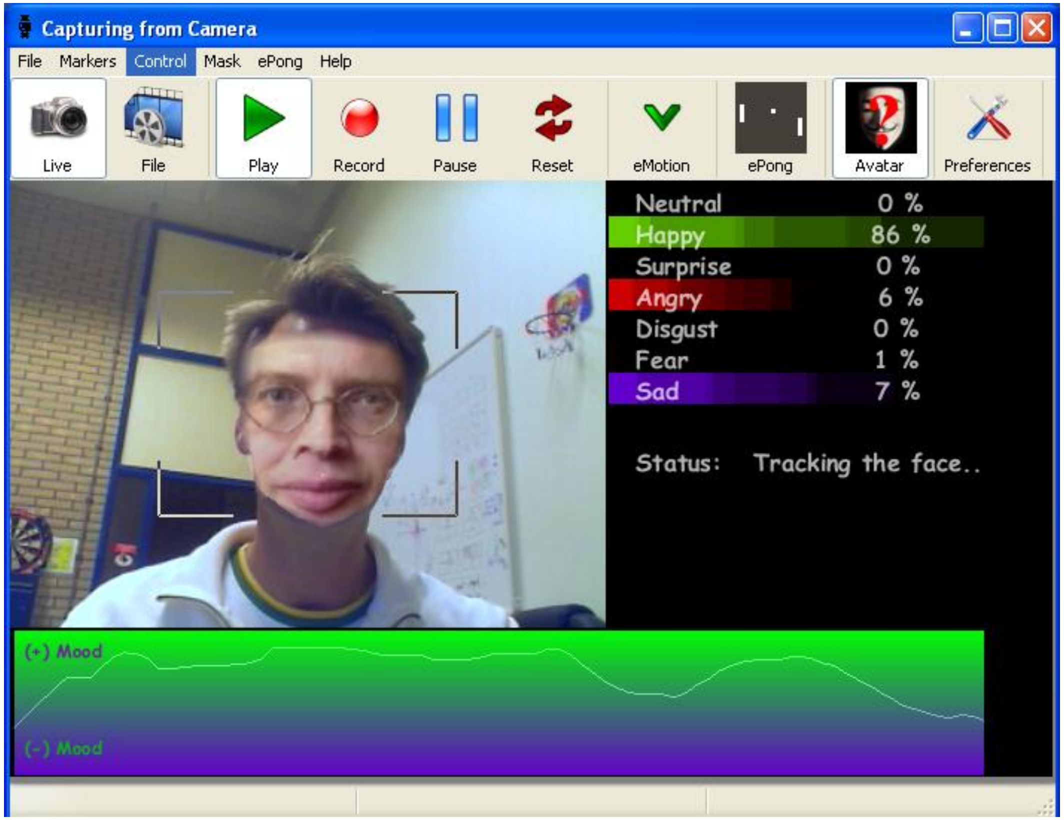
# Face off

# The mask

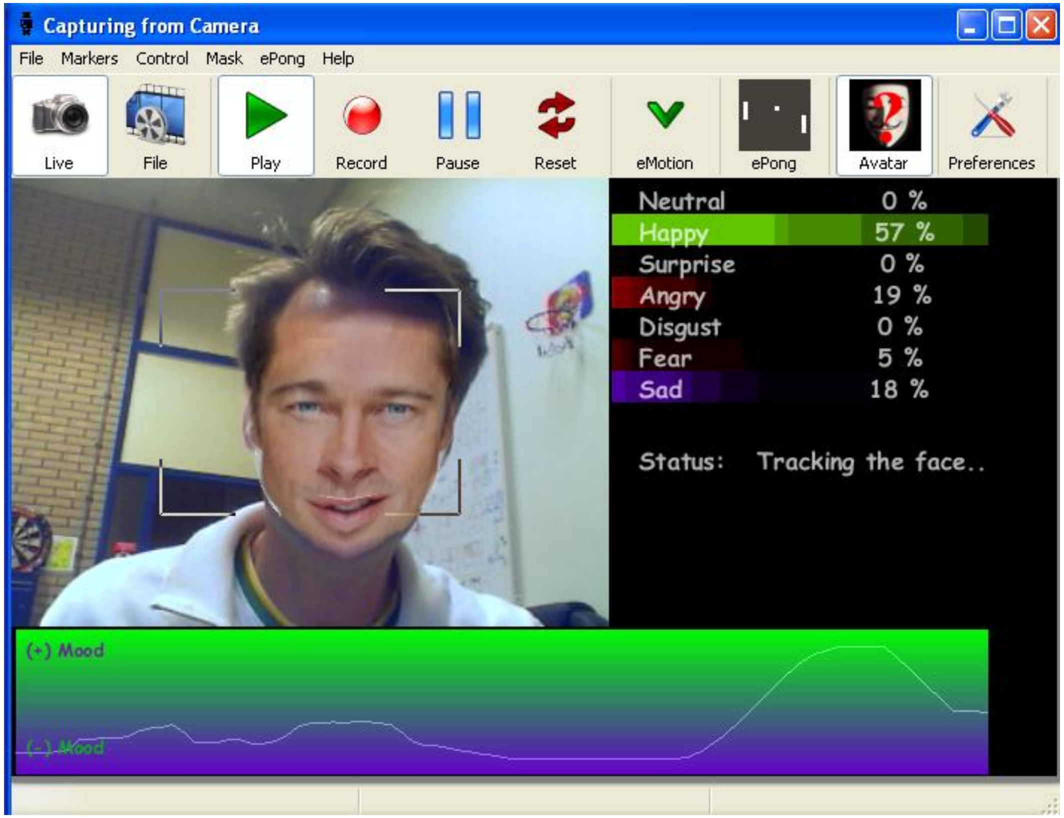


Capturing from Camera
File Markers Control Mask ePong Help
Live
File
Play
Record
Pause
Reset
eMotion
ePong
Avatar
Preferences
Neutral 0 %
Happy 99 %
Surprise 0 %
Angry 0 %
Disgust 0 %
Fear 0 %
Sad 1 %
Status: Tracking the face..
(+) Mood
(-) Mood

# The mask


Capturing from Camera
File
Markers
Control
Mask
ePong
Help
Live
File
Play
Record
Pause
Reset
eMotion
ePong
Avatar
Preferences
Neutral 0 %
Happy 86 %
Surprise 0 %
Angry 6 %
Disgust 0 %
Fear 1 %
Sad 7 %
Status: Tracking the face..
(+) Mood
(-) Mood

# The mask

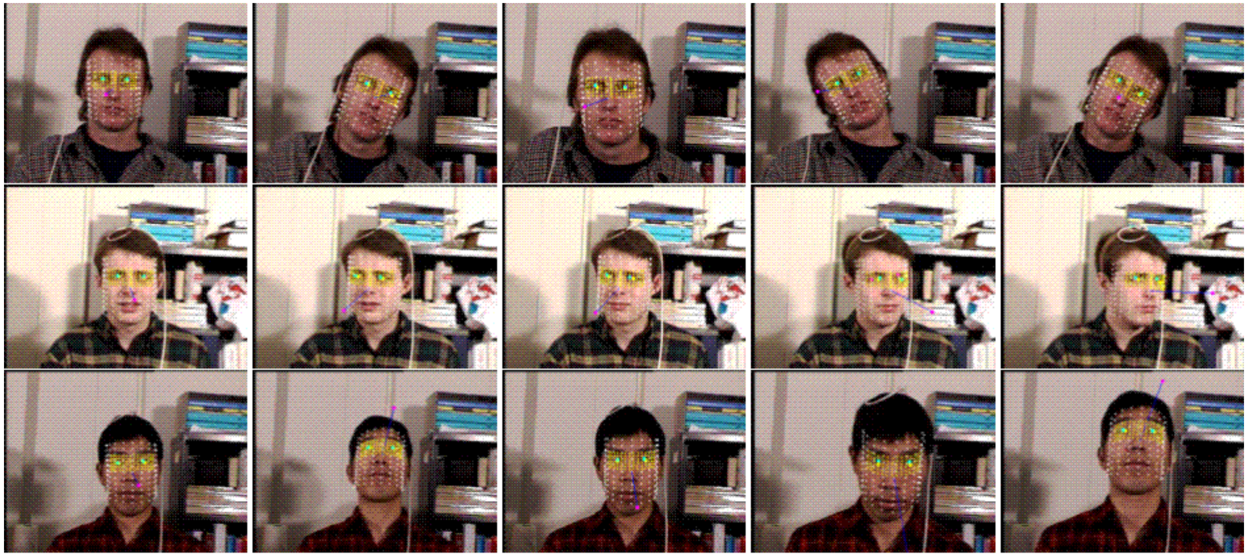


Capturing from Camera
File Markers Control Mask ePong Help
Live
File
Play
Record
Pause
Reset
eMotion
ePong
Avatar
Preferences
Neutral 0 %
Happy 57 %
Surprise 0 %
Angry 19 %
Disgust 0 %
Fear 5 %
Sad 18 %
Status: Tracking the face..
(+) Mood
(-) Mood

[Valenti, cvpr08]

# Human behaviour understanding

- Facial expression
- Head pose
- Eye Tracking
- Voice



# Motivation - The big Picture


**Roberto Valenti**
***Intelligent Systems Lab Amsterdam***
**University of Amsterdam**

# Motivation – The big picture

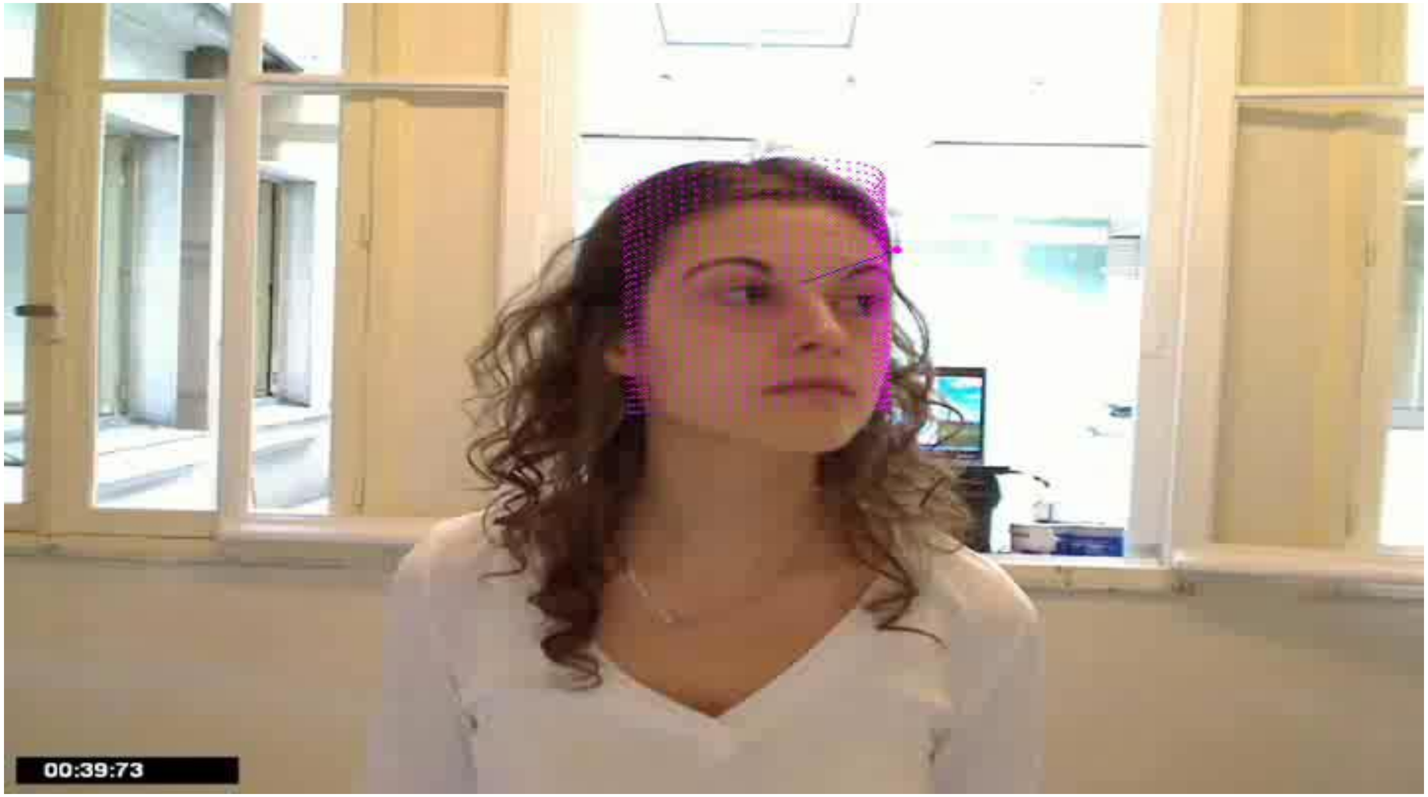
# Dataset


Special
Special
Fitness
& Fruits

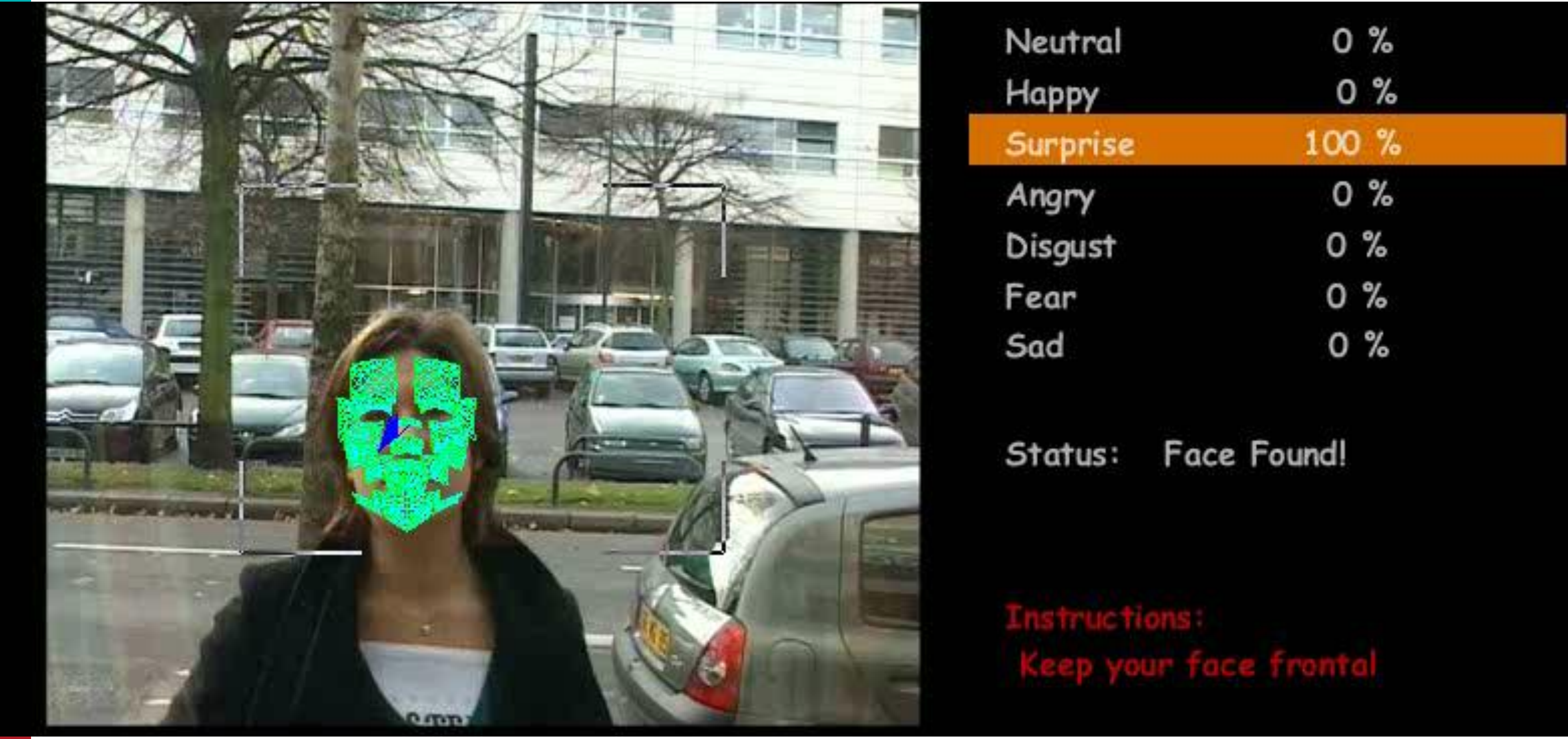


GRANY
Special
Special
Nestlé
Fitness
& Fruits

# Head pose estimation



# Pose Tracking



# Conclusions

- Color invariance needed
- Balance between discriminative power and invariance
- Color add information to classification achieving best performance in VOC08/VOC09 and TRECVid08/TRECVid09.
- Speed up is required (e.g. GPU)
- Higher semantics like aggression, emotions etc.