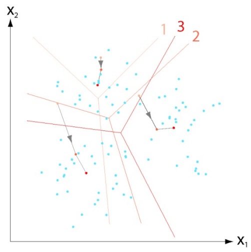
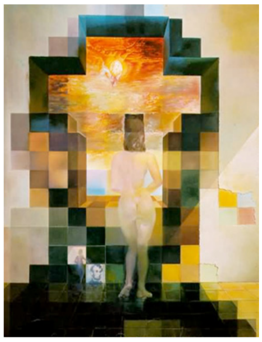
INF 5300 Advanced Topic: Video Content Analysis

Classification and clustering in video

Asbjørn Berge





Salvador Dali (1974)
Gala Contemplating the Mediterranean Sea Which at Twenty Meters Becomes the Portrait of Abraham Lincoln – Homage to Rothko (first version)

Outline

Clustering for motion detection

- Video and motion
- Frame differences
- Cluster algorithms
- Gaussian mixtures
- Example/code

Classification for face detection

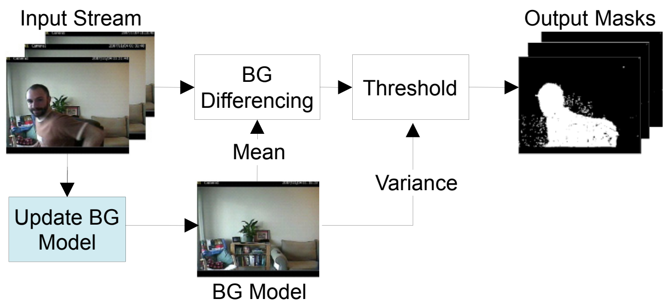
- Object detection in general
- The Adaboost algorithm
- Face detection using rectangular features
- Cascading classifiers
- Extensions

Clustering for foreground detection

- Automatically estimate which parts of the image are not part of the background
- Build a model of the background
- What is not background, must be foreground (an object of some sort)
- The background image model must include:
  - Illumination changes (gradual and sudden)
  - Distractions (leaves and trees swaying, shadows, weather)
  - Semi-permanent changes (parked cars)
  - Camera noise
- Foreground (implicitly handled by background model) contaminated by
  - Camouflage / similar colors
  - Fragmentation

Detecting moving objects

- **Assumption**: objects that move are important (e.g., people, vehicles)
- **Basic approach**: maintain a model of the static background. Compare the current frame with the background to locate moving foreground objects



5

Average image

- What can we do with this?

SINTEF Technology for a better society

6

Background Subtraction

-

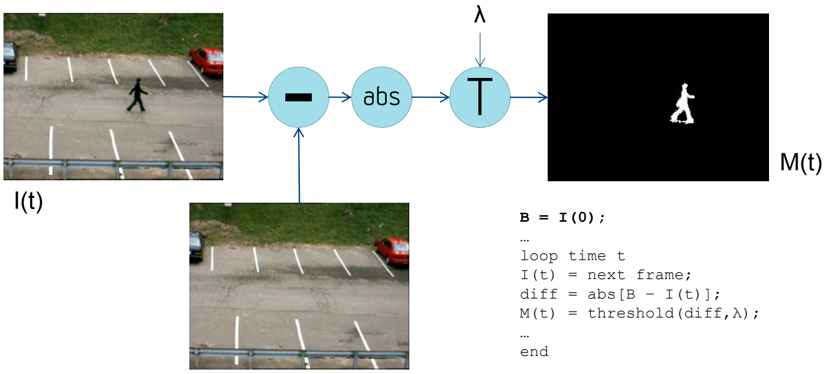
=

SINTEF Technology for a better society

7

Simple background subtraction

- Background model is a static image (assumed to have no objects present).
- Pixels are labeled as object (1) or not object (0) based on thresholding the absolute intensity difference between current frame and background.

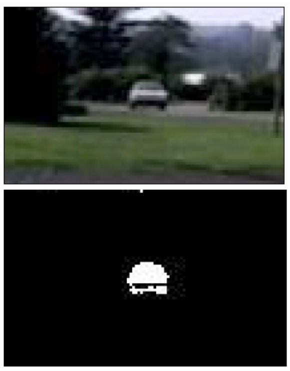



```
B = I(0);
…
loop time t
I(t) = next frame;
diff = abs[B – I(t)];
M(t) = threshold(diff,λ);
…
end
```


8

BG Observations

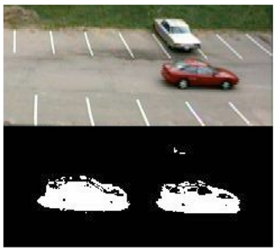
- Background subtraction does a reasonable job of extracting
- the shape of an object, provided the object intensity/color is
- sufficiently different from the background.



9

BG observations 2

- Objects that enter the scene and stop continue to be detected, making it difficult to detect new objects that pass in front of them.
- If part of the assumed static background starts moving, both the object and its negative *ghost* (the revealed background) are detected



10

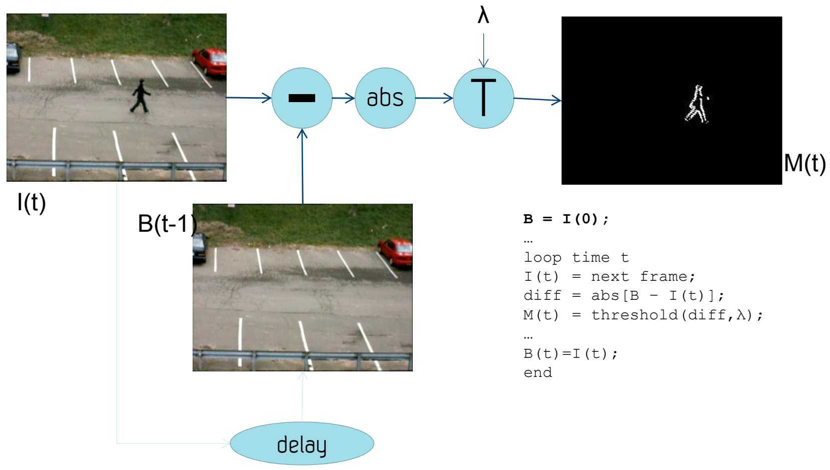
More observations

- Background subtraction is sensitive to changing illumination and unimportant movement of the background (for example, trees blowing in the wind, reflections of sunlight off of cars or water).
- Background subtraction cannot handle movement of the camera.

11

Simple frame differencing

- Background model is replaced with the previous image.




```
B = I(0);
…
loop time t
I(t) = next frame;
diff = abs[B – I(t)];
M(t) = threshold(diff,λ);
…
B(t)=I(t);
end
```


12

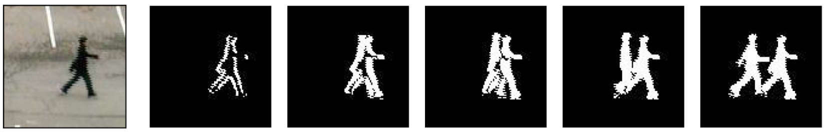
Frame differencing observations

- Frame differencing is very quick to adapt to changes in lighting or camera motion.
- Objects that stop are no longer detected. Objects that start up do not leave behind ghosts.
- However, frame differencing only detects the leading and trailing edge of a uniformly colored object. As a result very few pixels on the object are labeled, and it is very hard to detect an object moving towards or away from the camera

Differencing and temporal scale

- Note what happens when we adjust the temporal scale (frame rate) at which we perform two-frame differencing …

Define D(N) = || I(t) - I(t+N) ||



I(t) D(-1) D(-3) D(-5) D(-9) D(-15)

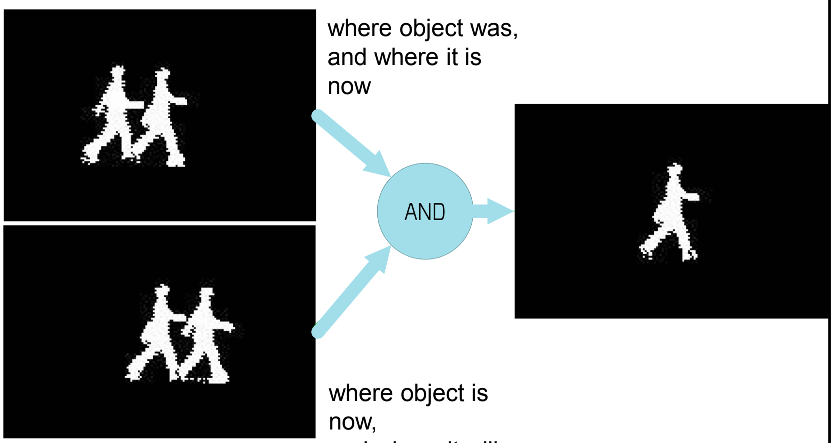
more complete object silhouette, but two copies
(one where object used to be, one where it is now).

Three frame differencing

- The previous observation is the motivation behind three-frame differencing

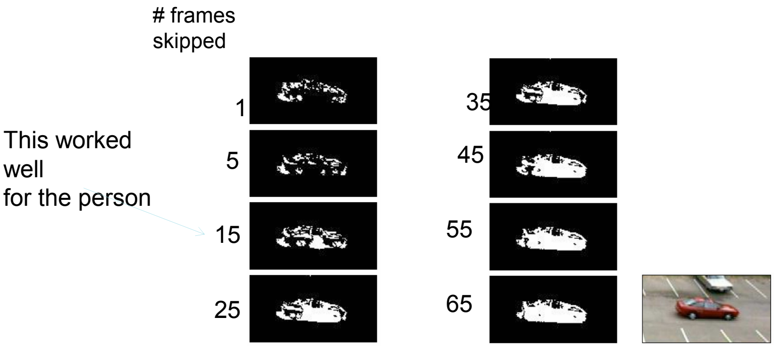
D(-15)

D(+15)



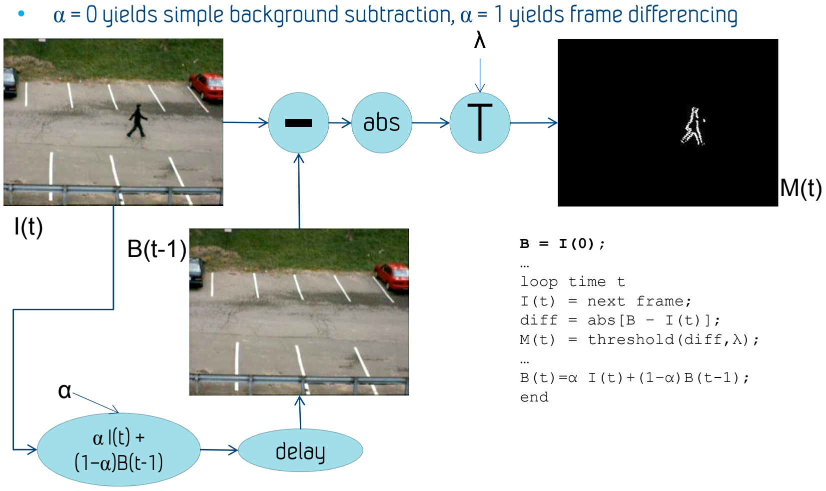
Three frame differencing

- Choice of good frame-rate for three-frame differencing depends on the size and speed of the object



Adaptive background subtraction

- Current image is "blended" into the background model with parameter α
- α = 0 yields simple background subtraction, α = 1 yields frame differencing



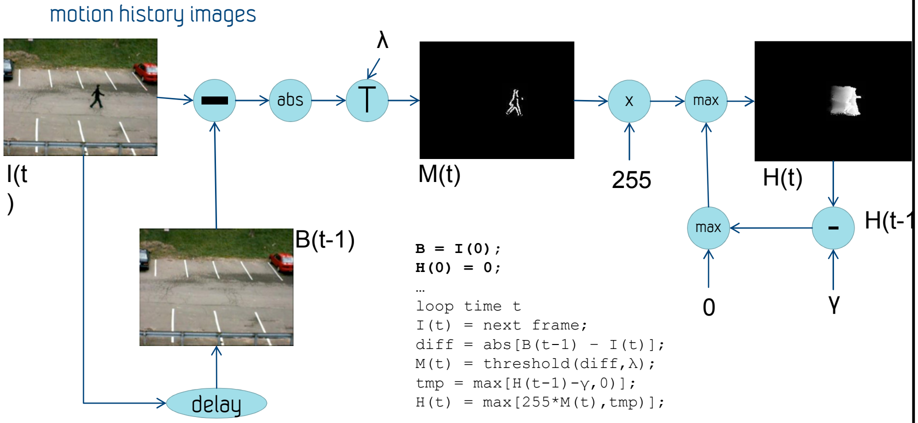

```
B = I(0);
…
loop time t
I(t) = next frame;
diff = abs[B – I(t)];
M(t) = threshold(diff,λ);
…
B(t)=α I(t)+(1–α)B(t-1);
end
```

Adaptive BG observations

- Adaptive background subtraction is more responsive to changes in illumination and camera motion.
- Fast small moving objects are well segmented, but they leave behind short "trails" of pixels.
- Objects that stop, and ghosts left behind by objects that start, gradually fade into the background.
- The centers of large slow moving objects start to fade into the background too! This can be "fixed" by decreasing the blend parameter α, but then it takes longer for stopped/ghost objects to disappear

Persistent Frame Differencing

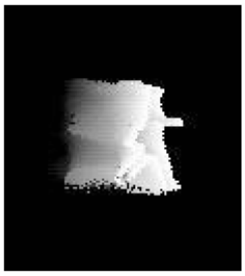
- Motion images are combined with a linear decay term also known as motion history images

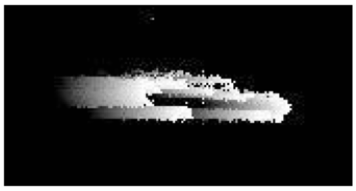



```
B = I(0);
H(0) = 0;
…
loop time t
I(t) = next frame;
diff = abs[B(t-1) - I(t)];
M(t) = threshold(diff,λ);
tmp = max[H(t-1)-γ,0)];
H(t) = max[255*M(t),tmp)];
…
B(t)=I(t);
end
```

Persistant FD Observations

- Persistant frame differencing is also responsive to changes in illumination and camera motion, and stopped objects / ghosts also fade away.
- Objects leave behind gradually fading trails of pixels. The gradient of this trail indicates the apparent direction of object motion in the image.
- Although the centers of uniformly colored objects are still not detected, the leading and trailing edges are make wider by the linear decay, so that perceptually (to a person) it is easier to see the whole object.

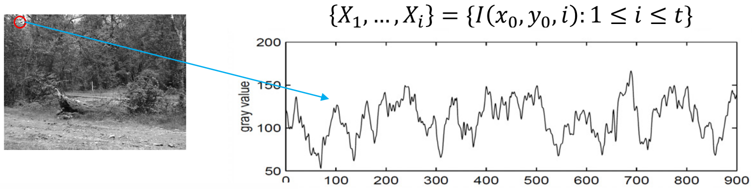


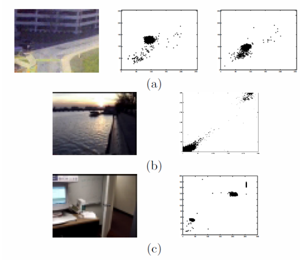


Statistical background modeling

At any time, t, what is known about a particular pixel, {x0; y0}, is its history

$$\{X_1, \ldots, X_i\} = \{I(x_0, y_0, i) : 1 \le i \le t\}$$





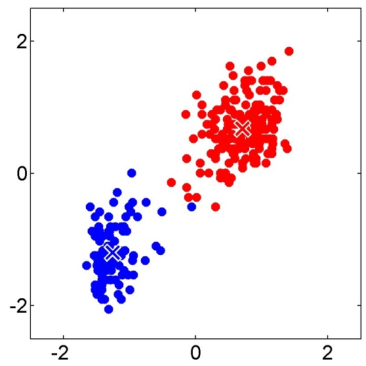
•Pixel history is likely to be multiple clusters
•Maintain an adaptive color model at each pixel based on a mixture of Gaussians (typically up to 5 components)

Chris Stauffer and Eric Grimson, "*Adaptive Background Mixture Models for Real-time Tracking*," IEEE Computer Vision and Pattern Recognition (CVPR), June 1999, pp.246-252.

Reminder: K-means Algorithm

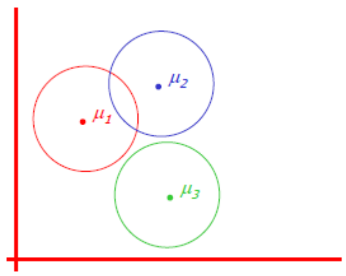
Step 4:

If the clusters don't change; $\mu_k^{(i+1)} \approx \mu_k^{(i)}$ (or prespecified number of iterations i reached), *terminate*, else reassign - increase iteration i and goto step 2.



Reminder: Mixture of Gaussians

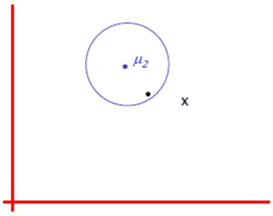
- A Probabilistic Clustering algorithm whose steps are as follows:
  1) Assume there are k components where ω_i represents the i'th component and has a mean vector μ_i with a covariance matrix $\sigma^2 I$



Reminder: Mixture of Gaussians

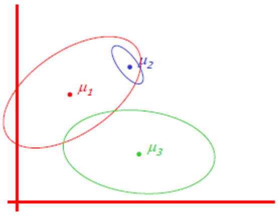
2) Select a random component i with probability $P(\omega_i)$

3) A datapoint can then be generated as $N(\mu_i, \sigma^2 I)$



Reminder: Mixture of Gaussians

4) We can define the general datapoint as $N(\mu_i, \Sigma_i)$ where each component generates data from a Gaussian with mean μ_i and covariance matrix Σ_i.



5) Next, we are interested in calculating the probability that an observation from class ω_i, would have the data x given means $\mu_i, \ldots, \mu_x$:

$P(x|\omega_i, \mu_i, \ldots, \mu_k)$

Reminder: Mixture of Gaussians

6) Goal: maximize the *probability* of a datum given the centers of the Gaussians

$$P(x|\mu_i) \ = \ \Sigma_i \, P(\omega_i)\, P(x|\omega_i, \mu_1, \mu_2, \dots, \mu_k)$$

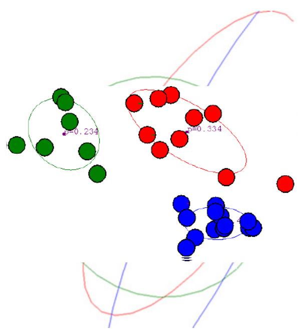
$$\downarrow$$

$$P(data|\mu_i) \ = \ \prod_{i=1}^{N} \Sigma_i \, P(\omega_i)\, P(x|\omega_i, \mu_1, \mu_2, \dots, \mu_k)$$

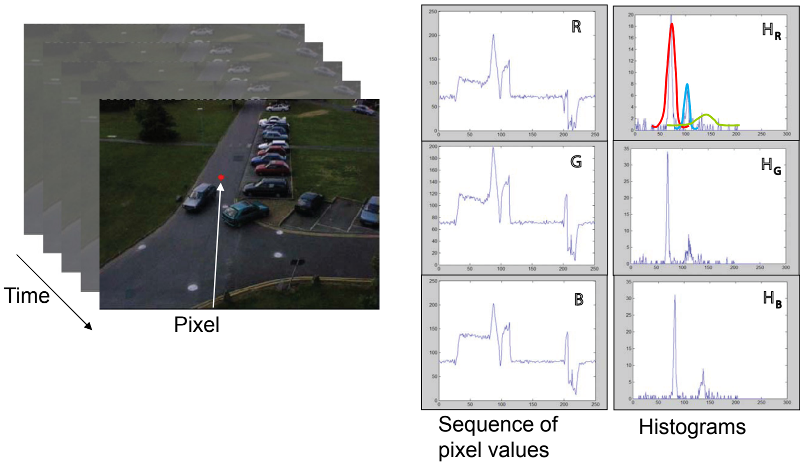
The most popular and simple algorithm that is used is the *Expectation-Maximization* (EM) algorithm

Reminder: Mixture of Gaussians

- An iterative algorithm for finding maximum likelihood estimates of parameters in probabilistic models whose steps are as follows:
  1) Initialize the distribution parameters
  2) Estimate the *Expected* value of the unknown variables
  3) Re-estimate the distribution parameters to Maximize the likelihood of the data
  4) Repeat steps 2 and 3 until convergence



Background subtraction – modeling by multiple gaussians



Modeling pixel values using mixtures of Gaussians (GMM)

- For some reasonable choice of *T* we have a set of pixel vectors (R,G,B)

$$\mathcal{X}_T = \{x^{(t)}, ..., x^{(t-T)}\}$$

- Probability of observing current pixel value is

$$\hat{p}(\vec{x}|\mathcal{X}_T, BG{+}FG) = \sum_{m=1}^{M} \hat{\pi}_m \mathcal{N}(\vec{x}; \widehat{\vec{\mu}}_m, \widehat{\sigma}_m^2 I)$$

- *M* is typically 2-5, and the covariance in the Gaussian mixtures is assumed diagonal for simplicity

Estimating/Updating the parameters of the model

- Each new observation is integrated into the model using standard learning rules (using the EM algorithm for every pixel would be costly).
  - Every pixel value, x_t, is checked against the existing M Gaussian distributions to find the one that represents it most
  - A match is defined as a pixel value within 3σ of a distribution (i.e., **each pixel has essentially its own threshold**), measured with Mahalanobis distance $D_m^2(\vec{x}^{(t)}) = \vec{\delta}_m^T \vec{\delta}_m / \hat{\sigma}_m^2$

Estimating/Updating the parameters of the model (cont'd)

- If a **match** is found, the parameters of each mixture model are updated as follows:

$$\hat{\pi}_m \leftarrow \hat{\pi}_m + \alpha(o_m^{(t)} - \hat{\pi}_m)$$

exponential decay

where the ownership for class m is 1 for the best match and 0 for the others

- Renormalization after update to sum to 1
- Also, we define *close* components, and generate a new component if none match

31

Estimating/Updating the parameters of the model (cont'd)

- The parameters of the matched Gaussian *i* are updated as follows:

$$\widehat{\vec{\mu}}_m \leftarrow \widehat{\vec{\mu}}_m + o_m^{(t)}(\alpha/\hat{\pi}_m)\vec{\delta}_m$$
$$\widehat{\sigma}_m^2 \leftarrow \widehat{\sigma}_m^2 + o_m^{(t)}(\alpha/\hat{\pi}_m)(\vec{\delta}_m^T\vec{\delta}_m - \widehat{\sigma}_m^2)$$

using the distance to mean for each component

$$\vec{\delta}_m = \vec{x}^{(t)} - \widehat{\vec{\mu}}_m$$

with a learn rate dependent on the number of frames used for learning, $\alpha = 1/T$

32

Estimating/Updating the parameters of the model (cont'd)

- If a match is **not** found, the least probable distribution is replaced with a distribution with the current pixel value as its mean value, an initial high variance, and a low prior weight
  $\hat{\pi}_{M+1} = \alpha,\ \widehat{\vec{\mu}}_{M+1} = \vec{x}^{(t)},\ \hat{\sigma}_{M+1} = \sigma_0$
- If there is max amount of components, discard the one with smallest prior π

Determining the background Gaussians

- Determine which Gaussians from the mixture represent the "background processes".
- The following heuristic is used to determine the "background" Gaussians: "Choose the Gaussians which have most supporting evidence and the least variance"
- Observations:
  - Moving objects are expected to produce more variance than a "static" (background) object - VARIANCE
  - There should be more data supporting the background distributions because they are repeated, whereas pixel values from different objects are often not the same color – PERSISTANCE

Determining the background model

- The presented algorithm presents an on-line clustering algorithm. Usually, the intruding foreground objects will be represented by some additional clusters with small weights π,
- We can approximate the background model by the first B largest clusters:

$$p(\vec{x}|\mathcal{X}_T, BG) \sim \sum_{m=1}^{B} \hat{\pi}_m \mathcal{N}(\vec{x}; \widehat{\vec{\mu}}_m, \sigma_m^2 I)$$

with components sorted descending by weight

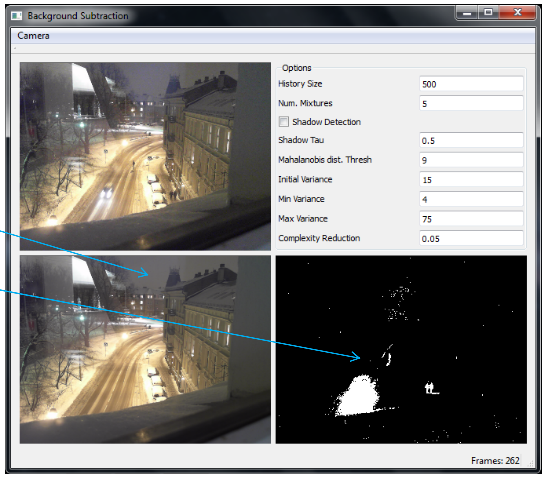
$$B = \arg\min_b \left(\sum_{m=1}^{b} \hat{\pi}_m > (1 - c_f) \right)$$

where the amount c_f is the maximum amount of the data that can belong to the foreground without influencing the background.

35

Determining the background model

- Summing the mean values corresponding to background components give continously updated background model
- Thresholding the deviations from the background model gives the foreground

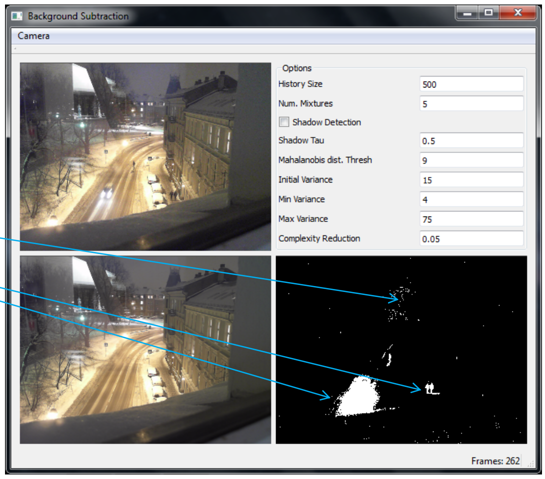


SINTEF Technology for a better society 35

36

Background subtraction example

- Enhanced GMM
  - OpenCV 2.3/Zivkovic 2006
- HD webcam (720p)
- Note the
  - Tree moving
  - Shadows
  - Headlights blooming
- Live demo
  - Should run on most platforms
  - Win 32 Binaries + Qt project available



SINTEF Technology for a better society 36

37

Updating the number of components in the model

- Approximate Dirichlet prior (penalty for increase in number parameters in model) , similar to Minimum Description Length.
- Cancel component *m* when prior becomes negative

$$\hat{\pi}_m \leftarrow \hat{\pi}_m + \alpha(o_m^{(t)} - \hat{\pi}_m) - \alpha c_T$$

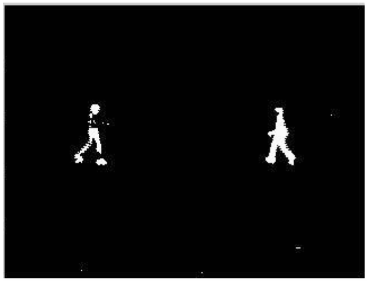
- Complexity penalty c_T is dependent on T (history window) and number of classes
- Proves useful, even if it is a very coarse approximation
- So, two mechanisms (heuristics) regulate the number of components, increasing count when statistical deviations occurs, and canceling when there is too little evidence.

SINTEF Technology for a better society 37

38

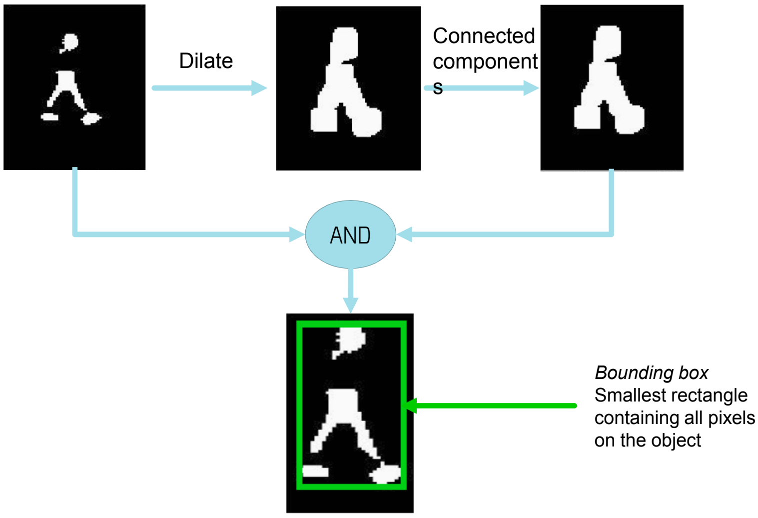
Blobbing

- Motivation: change detection is a pixel-level process.
- We want to raise our description to a higher level of abstraction
- Standard tools from image analysis is used
  - median filter to remove noisy pixels
  - connected components (with gap spanning)
  - size filter to remove small regions



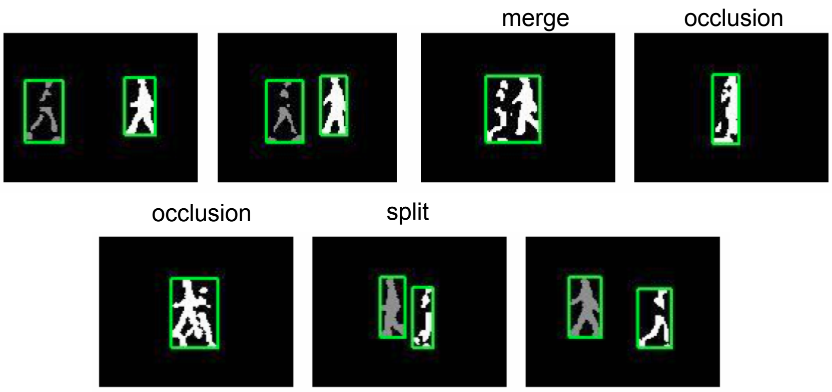
SINTEF Technology for a better society 38

Gap Spanning Connected Components



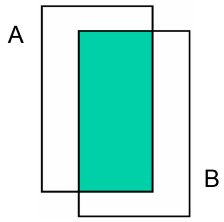
Blob merge/split

- When two objects pass close to each other, they are detected as a single blob.
- Often, one object will become occluded by the other one. One of the challenging problems is to maintain correct labeling of each object after they split again.



Data association

- Determining the correspondence of blobs across frames is based on feature similarity between blobs.
- Commonly used features: location , size / shape, velocity, appearance (eg colors)
- For example: location, size and shape similarity can be measured based on bounding box overlap:

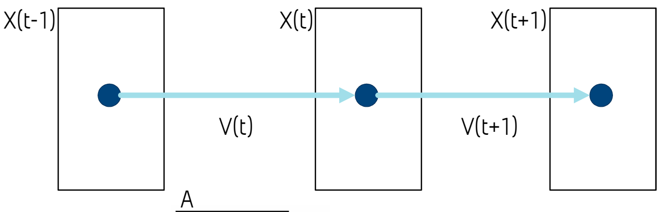


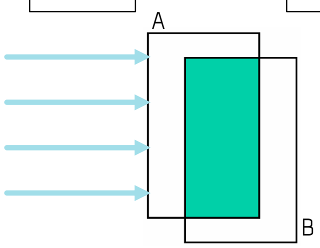
$$score = \frac{2 * area(A\ and\ B)}{area(A) + area(B)}$$

A = bounding box at time t
B = bounding box at time t+1

Data association

- It is common to assume that objects move with constant velocity : V(t)=V(t+1)





$$score = \frac{2 * area(A\ and\ B)}{area(A) + area(B)}$$

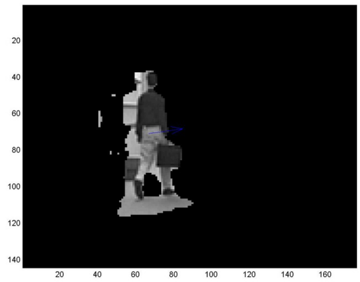
A = bounding box at time t, adjusted by velocity V(t)
B = bounding box at time t+1

43

Motion trajectory for data association

The direction of motion for each blob object can be estimated by only dealing with motion vectors computed from within foreground sections.

This reduces the amount of computation necessary to extract motion information. (useful in coding where layers are used)

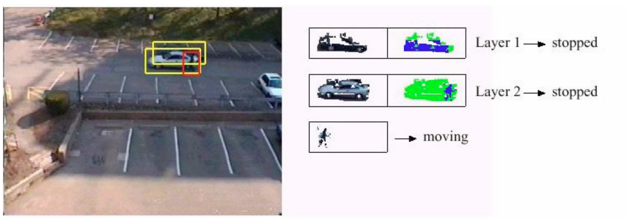


SINTEF Technology for a better society

44

Layered detection

- R.Collins, A.Lipton, H.Fujiyoshi and T.Kanade, "*Algorithms for Cooperative Multi-Sensor Surveillance,*" Proceedings of the IEEE, Vol 89(10), October 2001, pp.1456-1477.

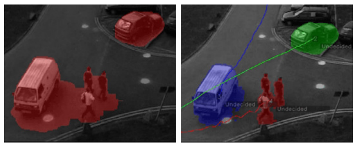


Allow blobs to be layered, so that stopped blobs can be considered part of the background for new object detection, but they will not leave behind ghosts when they start moving again.

Store stacks of background

SINTEF Technology for a better society

Shadow detection



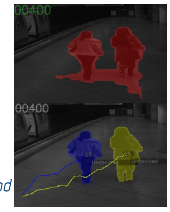
- Shadows and reflections are complex sources of noise for motion detection
  - Undersegmentation
  - Spurious detections
  - Object tracking and classification much harder
- A moving point may be a shadow for example if
  1. It is darker than the "shadowed" background

     $$\alpha < \frac{Y_p}{Y_{bg}} < \beta, 0 < \alpha < 1, 0 < \beta < 1, Y(\cdot) = \frac{R(\cdot) + G(\cdot) + B(\cdot)}{3}$$

  2. Ratios between color channelsa are approximately constant

     $$\frac{R_p}{R_{bg}} \cong k_R, \frac{B_p}{B_{bg}} \cong k_B, \frac{G_p}{G_{bg}} \cong k_G$$

  3. If it matches shadow-model from GMM of scene pixels (Martel Brisson, 2007)
- How do you detect a refletction without desensitizing the detection system? (Good luck!)



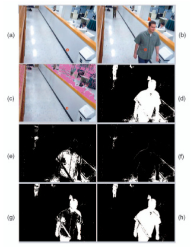


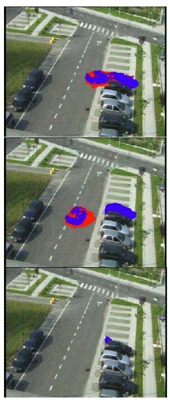
Fig. 10. Hallway 1. (a) Mean value of first background state of GMM. (b) Frame in the sequence. (c) Mean value of the first GMSM state. (d) Foreground detection from GMM. (e) Shadow detection from the YUV description. (f) Shadow detection from the GMSM. (g) Foreground detection: Img (d) - Img (e). (h) Foreground detection: Img (d) - Img (f).

Prati, I. Mikic, M. Trivedi, R. Cucchiara, "*Detecting Moving Shadows: Algorithms and Evaluation*," IEEE Trans. on Pattern Analysis and Machine Intelligence, 2003

Nicolas Martel-Brisson and Member-Andre Zaccarin : *"Learning and Removing Cast Shadows through a Multidistribution Approach"*, IEEE Trans. Pattern Anal. Mach. Intell. 29, 7 (Jul. 2007)

Ghost detection

- Effect of the continous update of background models
  - Apparent object – disappears slowly when background adapts
  - Observe: Some ghosts are *useful* (e.g., objects removed / left luggage)
- Can be detected by analyzing the motion (optical flow) within blobs
  - If the average magnitude of the flow is low/near zero, it is probably a ghost
  - When a blob that has recently detached from another blob has zero/significantly lower flow -> classify as ghost immediately and relearn the background
- Risk: if the ghost area is immediately occupied by another real object, the ghost will not be detected



Cucchiara, R.; Grana, C.; Piccardi, M.; Prati, A.; , "*Detecting moving objects, ghosts, and shadows in video streams*," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* , vol.25, no.10, pp. 1337- 1342, Oct. 2003 DOI: 10.1109/TPAMI.2003.1233909

47

Example implementation in OpenCV

- Implemented in the class BackgroundSubtractor
  http://opencv.itseez.com/modules/video/doc/motion_analysis_and_object_tracking.html#backgroundsubtractor

- Straightforward algorithm, no point in reimplementing for our purposes?
- QT project and binaries available
- Exercise (choose one or more):
  - Write your own background subtraction code using whatever programming language
  - Implement the simple blob post processing and association discussed shortly here
  - Make your own shadow detection

SINTEF Technology for a better society 47

48

REFERENCES

- Background subtraction
  - Z.Zivkovic, Improved adaptive Gausian mixture model for background subtraction, International Conference Pattern Recognition, UK, August, 2004, http://www.zoranz.net/Publications/zivkovic2004ICPR.pdf.
  - Radke, R.J.; Andra, S.; Al-Kofahi, O.; Roysam, B.; Image change detection algorithms: a systematic survey, Image Processing, IEEE Transactions on 14(3), 2005, Digital Object Identifier (DOI): 10.1109/TIP.2004.838698
  - C. Stauffer and E. Grimson, "Adaptive background mixture models for real-time tracking", IEEE Computer Vision and Pattern Recognition Conference, Vol.2, pp. 246-252, 1998
  - C. Stauffer and W. Grimson , "Learning Patterns of Activity Using Real-Time Tracking", IEEE TPAMI, 22(8):747–757, 2000.
  - Ahmed Elgammal, David Harwood, Larry Davis "Non-parametric Model for Background Subtraction", 6th European Conference on Computer Vision. Dublin, Ireland, June 2000.
  - R.Pless, J.Larson, S.Siebers, B.Westover, "Evaluation of Local Models of Dynamic Backgrounds," IEEE Computer Vision and Pattern Recognition (CVPR), June 2003
- Color and tracking features
  - A. Gilbert, R. Bowden ; Incremental, Scalable Tracking of Objects Inter Camera -; In Computer Vision and Image Understanding CVIU, Vol 111 Pages 43-58, 2008
  - R.Collins, A.Lipton, H.Fujiyoshi and T.Kanade, "Algorithms for Cooperative Multi-Sensor Surveillance," Proceedings of the IEEE, Vol 89(10), October 2001, pp.1456-1477.
  - Paul Centore, Colour Theory for Painters, http://www.99main.com/~centore/
- Shadows
  - Cucchiara, R.; Grana, C.; Piccardi, M.; Prati, A.; , "Detecting moving objects, ghosts, and shadows in video streams," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.25, no.10, pp. 1337- 1342, Oct. 2003 DOI: 10.1109/TPAMI.2003.1233909
  - Martel-Brisson, Nicolas and Zaccarin, Andre, Learning and Removing Cast Shadows through a Multidistribution Approach, IEEE Trans. Pattern Anal. Mach. Intell., 29(7), 2007
  - T. Horprasert, D. Harwood, and L. Davis, "A statistical approach for real-time robust background subtraction and shadow detection", In Proceedings IEEE ICCV'99, pages 1–19, 1999.

SINTEF Technology for a better society

49

Outline

Clustering for motion detection

- Video and motion
- Frame differences
- Cluster algorithms
- Gaussian mixtures
- Example/code

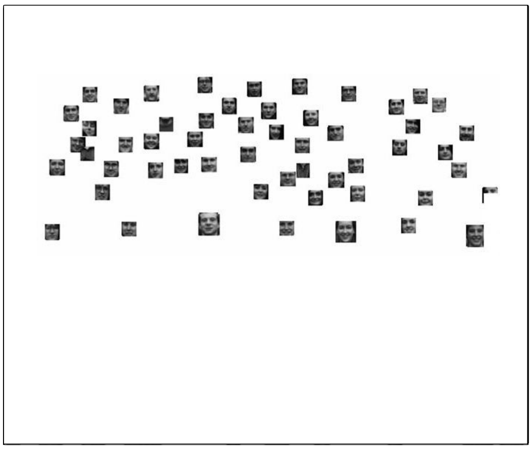
Classification for face detection

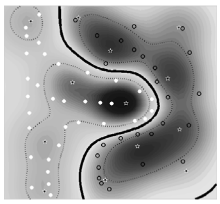
- Object detection in general
- The Adaboost algorithm
- Face detection using rectangular features
- Cascading classifiers
- Extensions

SINTEF Technology for a better society

50

Object Detection by classification: Motivation





Pictures from Romdhani et al. ICCV01

SINTEF Technology for a better society

Detection via classification: Main idea

Basic component: a binary classifier

Car/non-car
Classifier

No, not a car.
Yes, car.

Detection via classification: Main idea

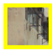
If object may be in a cluttered scene, slide a window around looking for it.

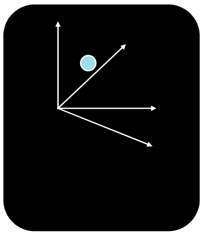
Car/non-car
Classifier

Detection via classification: Main idea

Fleshing out this pipeline a bit more, we need to:

1. Obtain training data
2. Define features
3. Define classifier





Car/non-car Classifier

Using rectangular features as weak classifiers

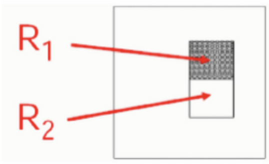
Image feature f is obtained as

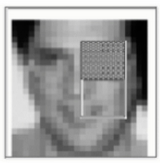
$$f(I) = \int_{R_2} I(x,y)\,dxdy - \int_{R_1} I(x,y)\,dxdy$$

and thus a weak classfier for a face is

$$h(x) = \begin{cases} 1 & \text{if } f(x) > \theta \\ -1 & \text{otherwise} \end{cases}$$

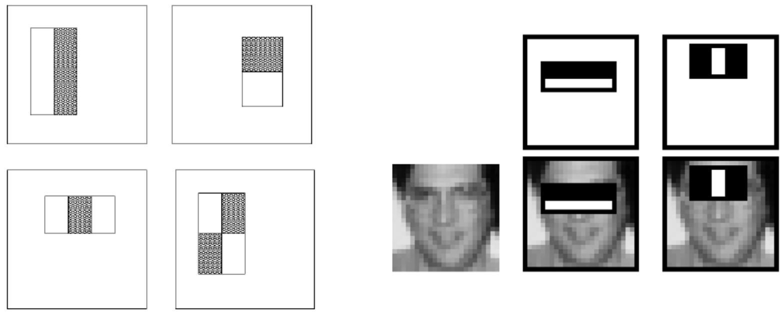
Need to choose threshold θ and which features to use





55

Gradient-based representations: Rectangular features



Compute differences between sums of pixels in rectangles

Captures contrast in adjacent spatial regions

Similar to Haar wavelets, efficient to compute

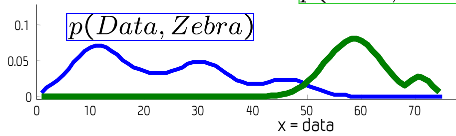
Viola & Jones, CVPR 2001

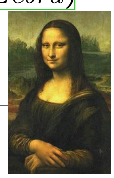
56

Discriminative vs. generative

- Generative model
  separately model class-conditional and prior densities

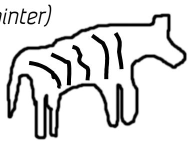
(*The artist*)

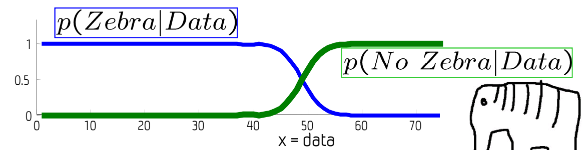




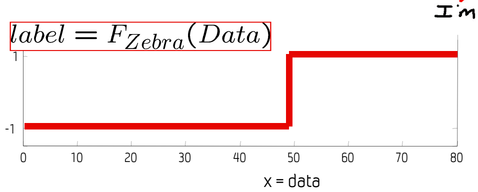
- Discriminative model
  directly model posterior

(*The lousy painter*)





- Classification function

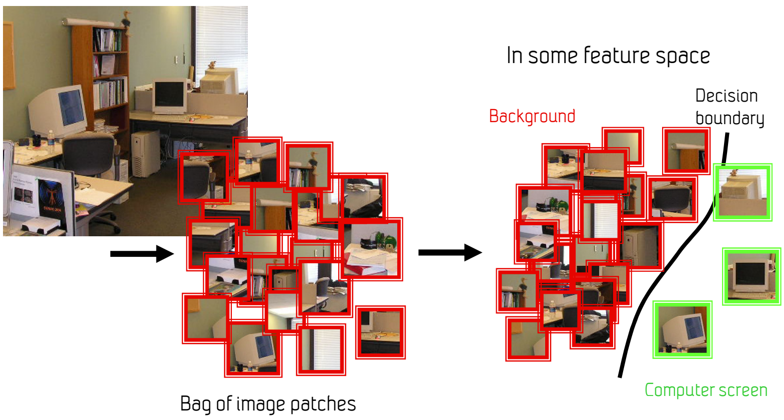


Discriminative vs. generative models

- Generative:
  - + possibly interpretable
  - + can draw samples
  - - models variability unimportant to classification task
  - - often hard to build good model with few parameters
- Discriminative:
  - + appealing when infeasible to model data itself
  - + excel in practice
  - - often can't provide uncertainty in predictions
  - - non-interpretable

Discriminative methods

Object detection and recognition is formulated as a classification problem.
The image is partitioned into a set of overlapping windows
… and a decision is taken at each window about if it contains a target object or not.



Binary classification

Features $x =$ x_1 x_2 x_3 … x_N x_{N+1} x_{N+2} … x_{N+M}

Labels $y =$ -1 +1 -1 -1 ? ? ?

Training data: each image patch is labeled as containing the object or background

Test data

- Classification function

$$\hat{y} = F(x)$$ Where $F(x)$ belongs to some family of functions

- Minimize misclassification error

(Not that simple: we need some guarantees that there will be generalization)

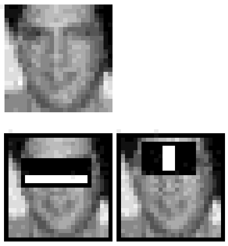
Example: Face detection

- Frontal faces are a good example of a class where global appearance models + a sliding window detection approach fit well:
  - Regular 2D structure
  - Center of face almost shaped like a "patch"/window

- Now we'll take AdaBoost and see how the Viola-Jones face detector works

61

Features



- Can a simple feature (i.e. a value) indicate the existence of a face?
- All faces share some similar properties
  - The eyes region is darker than the upper-cheeks.
  - The nose bridge region is brighter than the eyes.
  - **That is useful domain knowledge**
- Need for encoding of Domain Knowledge:
  - ***Location - Size:*** eyes & nose bridge region
  - ***Value:*** darker / brighter

62

Boxlet As Weak Classifier [Viola & Jones 01]

Boxlet: compute the difference between the sums of pixels within two rectangular regions

Compute boxlets all over a pattern

Harr-like wavelets

Over-complete representation: lots of boxlet features

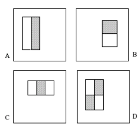
For each boxlet j, compute $f_j(x)$ where x is a positive or negative example

Each feature is used as a weak classifier

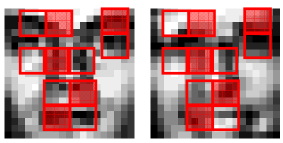
Set threshold θ_j so that most samples are classified correctly:

$$h_j(x, f, p, \theta) = 1 \; if \; f_j(x) < \theta_j \; (or \; f_j(x) > \theta_j)$$

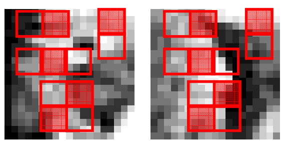
Sequentially select the boxlets



Boxlet:
2-rectangle,
3-rectangle,
4-rectangle



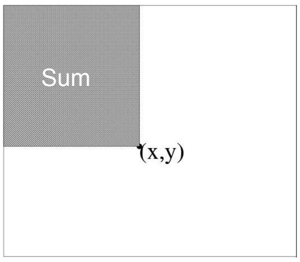
face samples



non-face samples

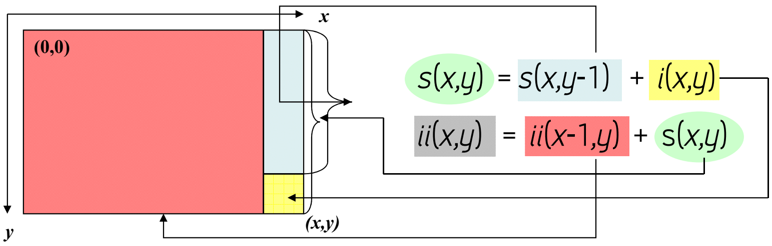
63

Define an "Integral image"



Def: The *integral image* at location (*x,y*), is the sum of the pixel values above and to the left of (*x,y*), inclusive.

Using the following two recurrences, where *i*(*x,y*) is the pixel value of original image at the given location and *s*(*x,y*) is the cumulative column sum, we can calculate the integral image representation of the image in a single pass.



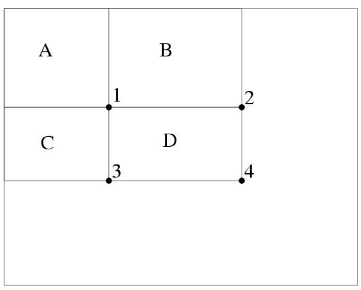
$$s(x,y) = s(x,y-1) + i(x,y)$$

$$ii(x,y) = ii(x-1,y) + s(x,y)$$

Paul Viola and Michael Jones www.cs.ucsd.edu/classes/fa01/cse291/ViolaJones.ppt
ICCV 2001 Workshop on Statistical and Computation Theories of Vision

64

Allows rapid evaluation of rectangular features



Using the integral image representation one can compute the value of any rectangular sum in constant time.

For example the integral sum inside rectangle D we can compute as:

$$ii(4) + ii(1) - ii(2) - ii(3)$$

As a result: two-, three-, and four-rectangular features can be computed with 6, 8 and 9 array references respectively.

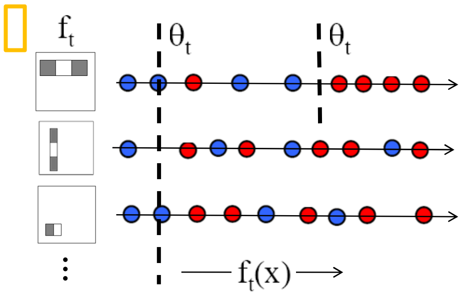
Paul Viola and Michael Jones www.cs.ucsd.edu/classes/fa01/cse291/ViolaJones.ppt
ICCV 2001 Workshop on Statistical and Computation Theories of Vision

Three goals for a face detector

1. *Feature Computation*: features must be computed as quickly as possible
2. *Feature Selection*: select the most discriminating features
3. *Real-timeliness*: must focus on potentially positive image areas (that contain faces)

Viola-Jones detector: AdaBoost

- **Want to select the single rectangle feature and threshold that best separates positive (faces) and negative (non-faces) training examples, in terms of *weighted* error.**



Outputs of a possible rectangle feature on faces and non-faces.

Resulting weak classifier:

$$h_t(x) = \begin{cases} +1 & \text{if } f_t(x) > \theta_t \\ -1 & \text{otherwise} \end{cases}$$

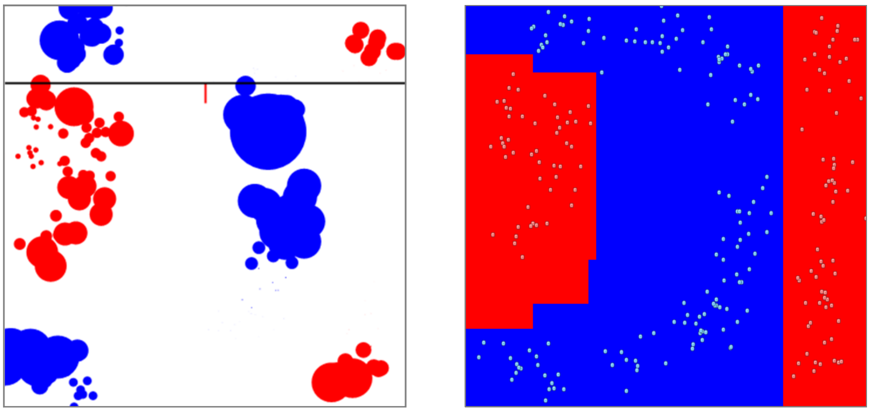
For next round, reweight the examples according to errors, choose another filter/threshold combo.

67

Boosting

Iteratively reweighting training samples.

Higher weights to previously misclassified samples.



50 rounds

68

AdaBoost

- Stands for "**Ada**ptive" **boost**
- Constructs a "strong" classifier as a linear combination of weighted simple "weak" classifiers

$$F(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \alpha_3 f_3(x) + \ldots$$

Weak classifier

Strong classifier

Image

Weight

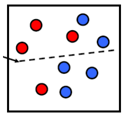
69

- Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \ldots, T$:
  1. Normalize the weights,
  $$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$
  so that w_t is a probability distribution.
  2. For each feature, j, train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to w_t, $\epsilon_j = \sum_i w_i \, |h_j(x_i) - y_i|$.
  3. Choose the classifier, h_t, with the lowest error ϵ_t.
  4. Update the weights:
  $$w_{t+1,i} = w_{t,i}\beta_t^{1-e_i}$$
  where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.
- The final strong classifier is:
$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$
where $\alpha_t = \log \frac{1}{\beta_t}$

AdaBoost Algorithm

Start with uniform weights on training examples

For T rounds



$\{x_1, \ldots x_n\}$

Evaluate *weighted* error for each feature, pick best.

Re-weight the examples:
Incorrectly classified -> more weight
Correctly classified -> less weight

Final classifier is combination of the weak ones, weighted according to error they had.

[**Freund & Schapire, 1995**]

SINTEF Technology for a better society

70

AdaBoost - *Characteristics*

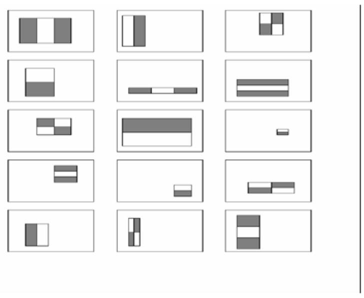
- Features as weak classifiers
  - Each single rectangle feature may be regarded as a simple weak classifier
- An iterative algorithm
  - AdaBoost performs a series of trials, each time selecting a new weak classifier
- Weights are being applied over the set of the example images
  - During each iteration, each example/image receives a weight determining its importance

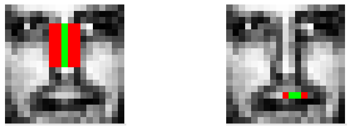
SINTEF Technology for a better society

71

Feature selection

- Problem: Too many features
  - In a sub-window (24x24) there are ~160,000 features (all possible combinations of orientation, location and scale of these feature types)
  - impractical to compute all of them (computationally expensive)
- We have to select a subset of relevant features – which are informative - to model a face
  - **Hypothesis**: "A very small subset of features can be combined to form an effective classifier"
  - How?
    - AdaBoost algorithm





Relevant feature Irrelevant feature

SINTEF Technology for a better society

72

AdaBoost – *Feature Selection*

Problem

- On each round, large set of possible weak classifiers (each simple classifier consists of a single **feature**) – Which one to choose?
  - choose the most efficient (the one that best separates the examples – the lowest error)
  - choice of a classifier corresponds to choice of a feature
- At the end, the 'strong' classifier consists of T features

Conclusion

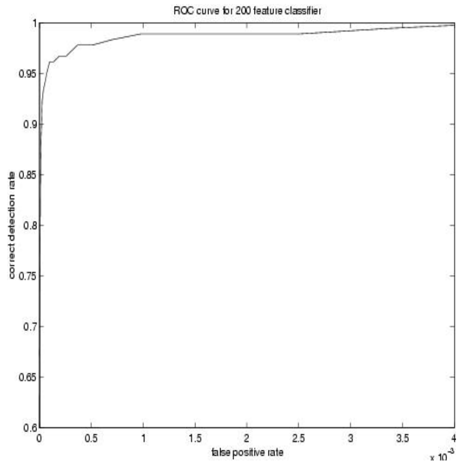
- AdaBoost searches for a small number of good classifiers – features (feature selection)
- adaptively constructs a final strong classifier taking into account the failures of each one of the chosen weak classifiers (weight appliance)
- AdaBoost is used to **both** select a small set of features and train a strong classifier

SINTEF Technology for a better society

73

Now we have a good face detector

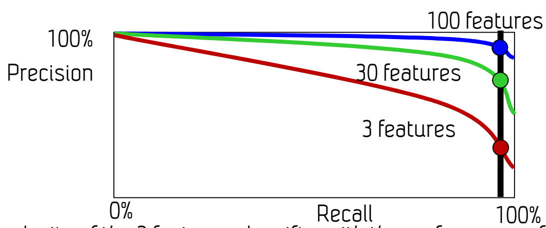
- We can build a 200-feature classifier!
- Experiments in original paper showed that a 200-feature classifier achieves:
  - 95% detection rate
  - 0.14×10^{-3} FP rate (1 in 14084)
  - Scanned all sub-windows of a 384x288 pixel image in 0.7 seconds (on Intel PIII 700MHz)
- The more the better (?)
  - Gain in classifier performance
  - Lose in CPU time
- Verdict: good & fast, but not enough
  - Competitors achieve close to **1 in a 1.000.000 FP rate!**
  - 0.7 sec / frame **IS NOT** real-time.



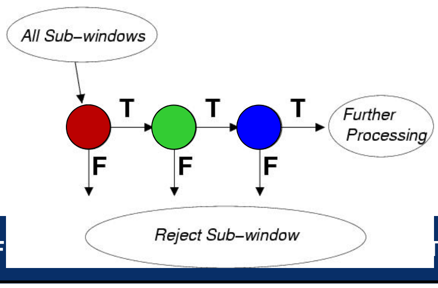
74

Cascade of classifiers

Fleuret and Geman 2001, Viola and Jones 2001



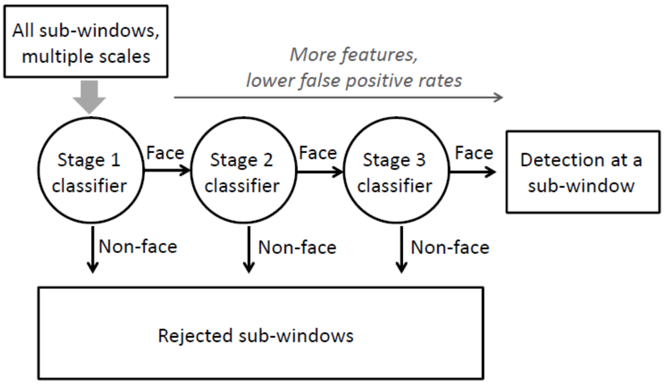
We want the complexity of the 3 features classifier with the performance of the 100 features classifier:



Select a threshold with high recall for each stage.

We increase precision using the cascade

Cascading classifiers for detection



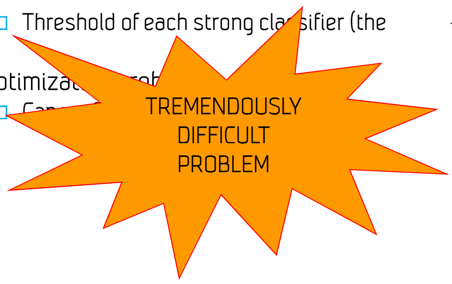
- Form a *cascade* with low false negative rates early on
- Apply less accurate but faster classifiers first to immediately discard windows that clearly appear to be negative

Training a cascade of classifiers

Keep in mind:
- Competitors achieved 95% TP rate, 10^{-6} FP rate
- These are the goals. Final cascade must do better!

Given the goals, to design a cascade we must choose:
- Number of layers in cascade (strong classifiers)
- Number of features of each strong classifier (the 'T' in definition)
- Threshold of each strong cla sifier (the $\frac{1}{2}\sum_{t=1}^{T}\alpha_t$ in definition)

Optimiza rob
- Can



Strong classifier definition:

$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T}\alpha_t h_t(x) \ge \frac{1}{2}\sum_{t=1}^{T}\alpha_t \\ 0 & otherwise \end{cases},$$

where $\alpha_t = \log(\frac{1}{\beta_t})$, $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$

A simple framework for cascade training

Do not despair. Viola & Jones suggested a heuristic algorithm for the cascade training:

- does not guarantee optimality
- but produces a "effective" cascade that meets previous goals

Manual Tweaking:

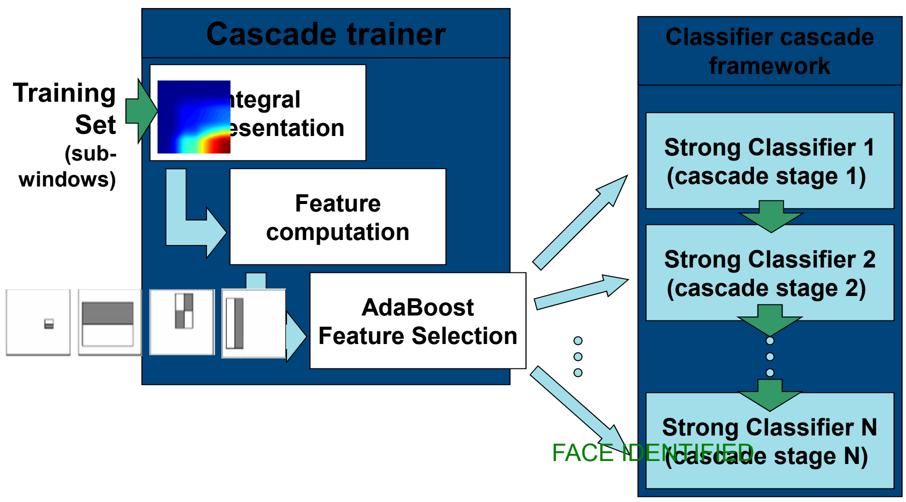
- overall training outcome is highly depended on user's choices
- select f_i (Maximum Acceptable False Positive rate / layer)
- select d_i (Minimum Acceptable True Positive rate / layer)
- select F_{target} (Target Overall FP rate)
- possible repeat trial & error process for a given training set

Until F_{target} is met:

- Add new layer:
  - Until f_i, d_i rates are met for this layer
    - **Increase** feature number & train new strong classifier with **AdaBoost**
    - Determine rates of layer on validation set

Testing phase

Training phase



pros …

Extremely fast feature computation
Efficient feature selection
Scale and location invariant detector

- Instead of scaling the image itself (e.g. pyramid-filters), we scale the features.

Such a generic detection scheme can be trained for detection of other types of objects (e.g. cars, hands)

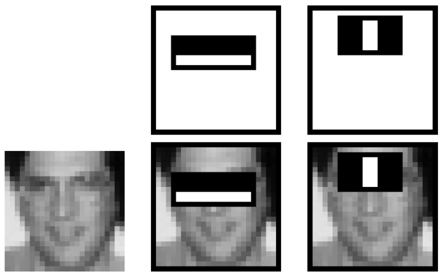
… and cons

Detector is most effective only on frontal images of faces

- can hardly cope with 45º face rotation

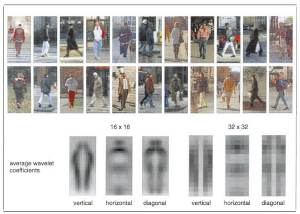
Sensitive to lighting conditions
We might get multiple detections of the same face, due to overlapping sub-windows.

Viola-Jones Face Detector Live demo



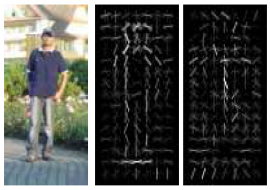
Pedestrian detection

- Detecting upright, walking humans also possible using sliding window's appearance/texture; e.g.,



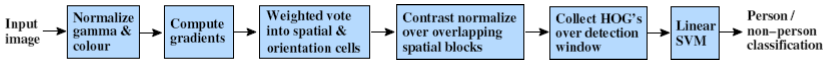
SVM with Haar wavelets [Papageorgiou & Poggio, IJCV 2000]

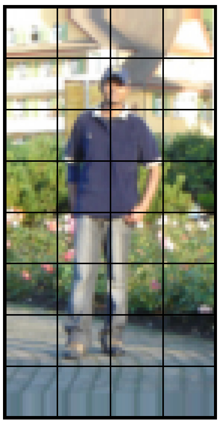
Space-time rectangle features [Viola, Jones & Snow, ICCV 2003]



SVM with HoGs [Dalal & Triggs, CVPR 2005]

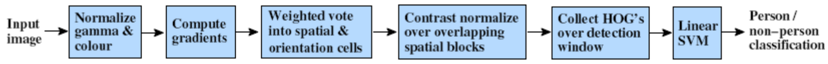
A short detour: HOG features for pedestrian detection





Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05

A short detour: HOG features for pedestrian detection



-1	0	1

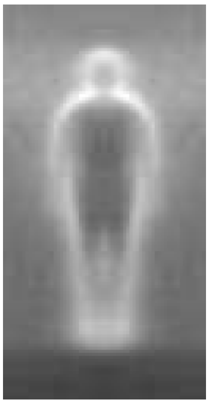
centered

-1	1

uncentered

1	-8	0	8	-1

cubic-corrected



0	1
-1	0

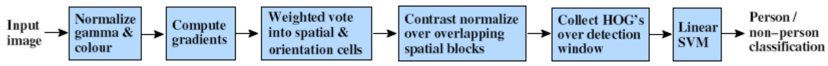
diagonal

-1	0	1
-2	0	2
-1	0	1

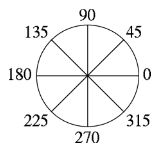
Sobel

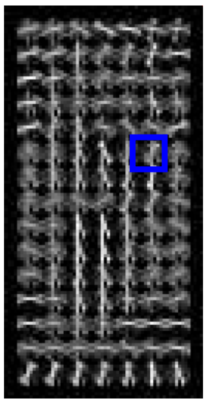
Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05

A short detour: HOG features for pedestrian detection



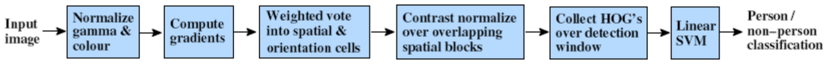
- Histogram of gradient orientations
  - -Orientation



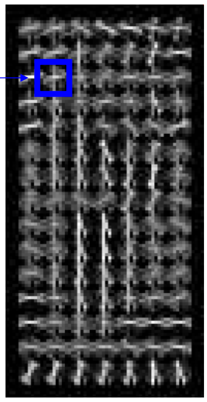


Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05

A short detour: HOG features for pedestrian detection



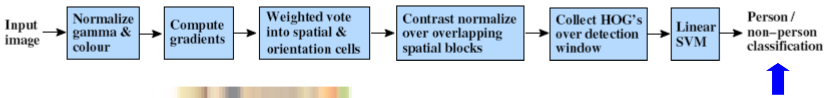
8 orientations

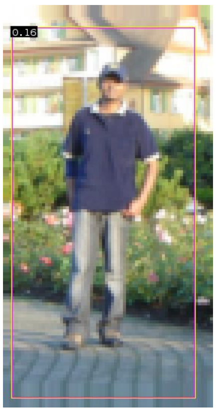
X=  $\in R^{840}$

15x7 cells

Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05

A short detour: HOG features for pedestrian detection





$$0.16 = w^T x - b$$

$$sign(0.16) = 1$$

$=>$ pedestrian

Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05

87

HOG Detector

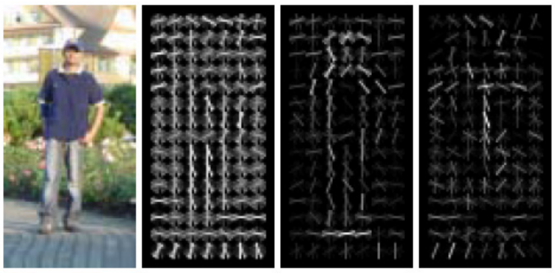


Image | HOG descriptor | HOG descriptor weighted by pos. SVM weights | neg. SVM weights

SINTEF Technology for a better society

88

Window-based detection: strengths

- Sliding window detection and global appearance descriptors:
  - Simple detection protocol to implement
  - Good feature choices critical
  - Past successes for certain classes

SINTEF Technology for a better society
Kristen Grauman

89

Window-based detection: Limitations

- High computational complexity
  - For example: 250,000 locations x 30 orientations x 4 scales = 30,000,000 evaluations!
  - If training binary detectors independently, means cost increases linearly with number of classes
- With so many windows, false positive rate better be low

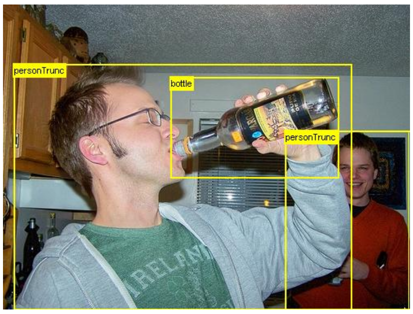
SINTEF Technology for a better society

Kristen Grauman

90

Limitations (continued)

- Not all objects are "box" shaped

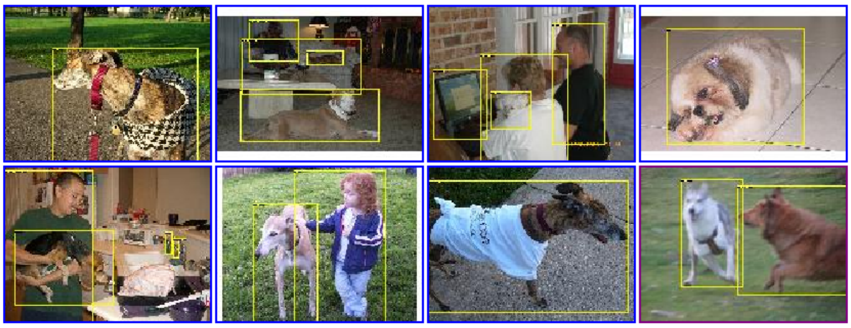


SINTEF Technology for a better society

91

Limitations (continued)

- Non-rigid, deformable objects not captured well with representations assuming a fixed 2d structure; or must assume fixed viewpoint
- Objects with less-regular textures not captured well with holistic appearance-based descriptions



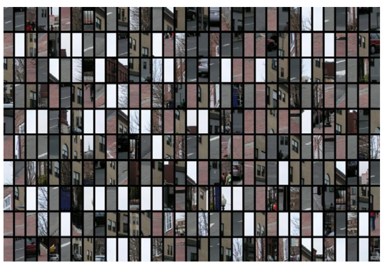
SINTEF Technology for a better society

92

Limitations (continued)

- If considering windows in isolation, context is lost

Sliding window



Detector's view

SINTEF Technology for a better society

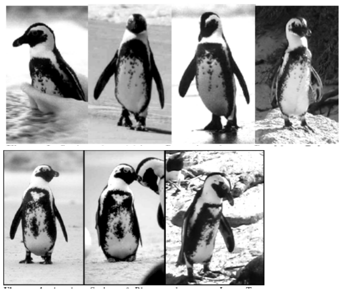
Limitations (continued)

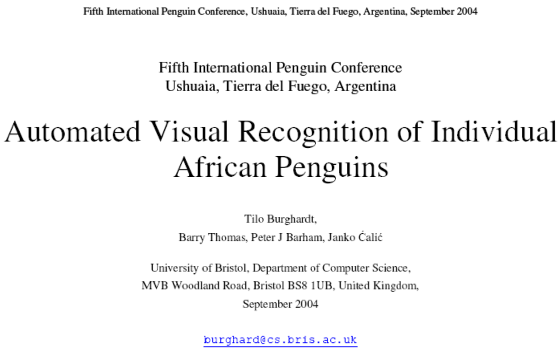
- In practice, often entails large, cropped training set (expensive)
- Requiring good match to a global appearance description can lead to sensitivity to partial occlusions

Summary and reading materials

- Basic pipeline for window-based detection
  - Model/representation/classifier choice
  - Sliding window and classifier scoring
- Boosting classifiers: general idea
- Viola-Jones face detector
  - key ideas: rectangular features, Adaboost for feature selection, cascade
- Some reading suggestions:
  - Richard Szeliski: Computer Vision: Algorithms and Applications, Chap 14.1 http://szeliski.org/Book/
  - Friedman, Hastie, Tibshirani. "Additive Logistic Regression: a Statistical View of Boosting" (1998)
  - Paul Viola and Michael J. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. IEEE CVPR, 2001. The paper is available online at http://www.ai.mit.edu/people/viola/
  - OpenCV documentation: http://opencv.itseez.com/modules/objdetect/doc/cascade_classification.html
  - Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection., In IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR'2005), pp. 886–893, San Diego, CA.

A cool (pun intended) example





Fifth International Penguin Conference, Ushuaia, Tierra del Fuego, Argentina, September 2004

Fifth International Penguin Conference
Ushuaia, Tierra del Fuego, Argentina

Automated Visual Recognition of Individual African Penguins

Tilo Burghardt,
Barry Thomas, Peter J Barham, Janko Ćalić

University of Bristol, Department of Computer Science,
MVB Woodland Road, Bristol BS8 1UB, United Kingdom,
September 2004

burghard@cs.bris.ac.uk

This project uses the Viola-Jones Adaboost face detection algorithm to detect penguin chests, and then matches the pattern of spots to identify a particular penguin.

Adaboost for chest stripe detection

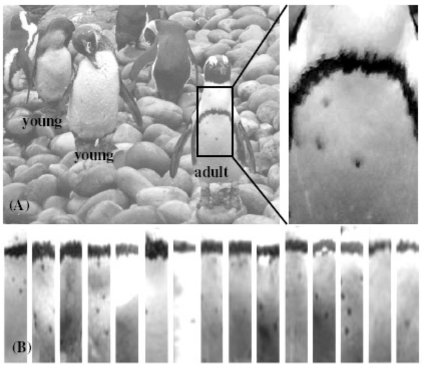
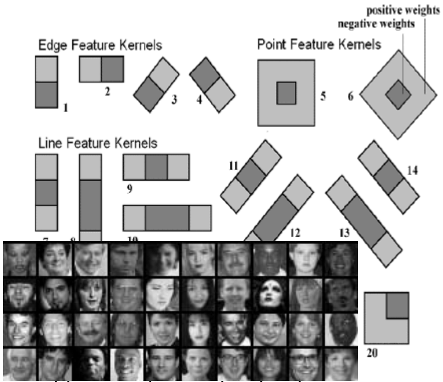


Figure 6. *Distinctive Chest Stripe of Adult African Penguins:* **(A)** Adult African penguins carry a distinctive and stable black stripe on their chest whilst young members of the species still change the colour of their chest feathers. **(B)** Various chests of adult African penguins under different lighting conditions. (figure source [19])



Use rectangular features, select good features to distinguish the chest from non-chests with Adaboost

Burghart, Thomas, Barham, and Calic. Automated Visual Recognition of Individual African Penguins , 2004.

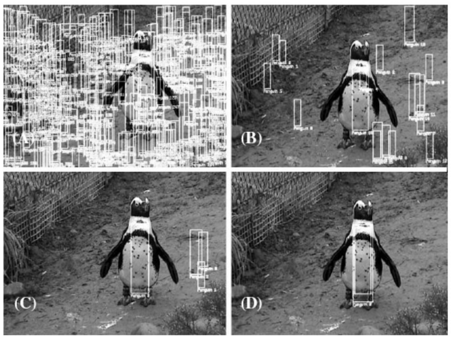


Figure 12. *Application of Attentional Cascades on Chests:* **(A)** Image areas that are accepted as likely to represent a chest after one stage are marked as white rectangles. **(B)** After three stages… **(C)** After five stages… **(D)** …and after seven stages with final result. (figure source [18], [19])

Attentional cascade

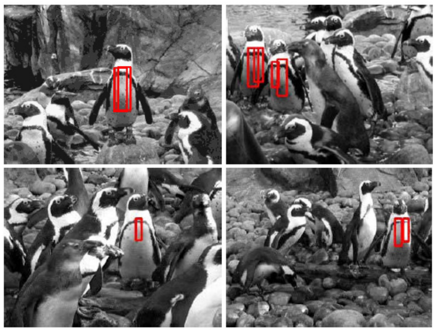


Figure 10. *AoI Detector Spotting Frontal Penguin Chests:* The detector was tested on a series of black and white still images and footage. Some result images are shown above. The detector might fire several times on one and the same chest instance. (figure

Penguin chest detections

Burghart, Thomas, Barham, and Calic. Automated Visual Recognition of Individual African Penguins , 2004.

Given a detected chest, try to extract the whole chest for this particular penguin.

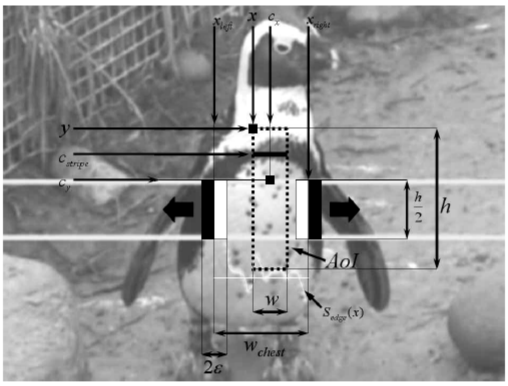
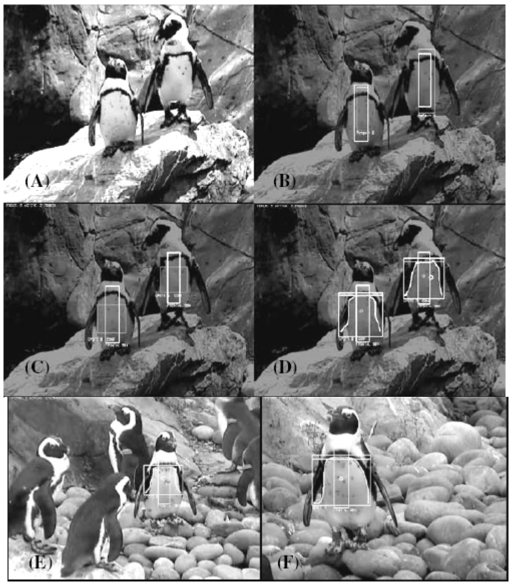
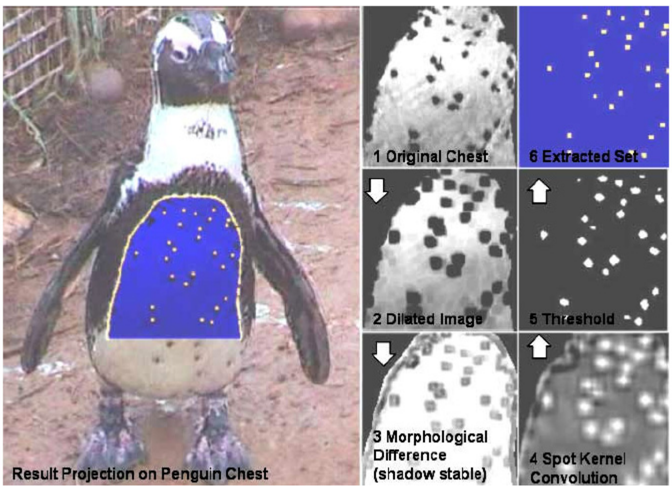


Figure 14. *Visual Description of the Chest Width Measurement:* Starting from an upper central point of the chest AoI two locally operating edge detectors moving apart search for the left and right boundary of the assumed chest. (figure source [17])

Burghart, Thomas, Barham, and Calic. Automated Visual Recognition of Individual African Penguins , 2004.



Example detections



Perform **identification** by matching the pattern of spots to a database of known penguins.

Penguin detection & identification

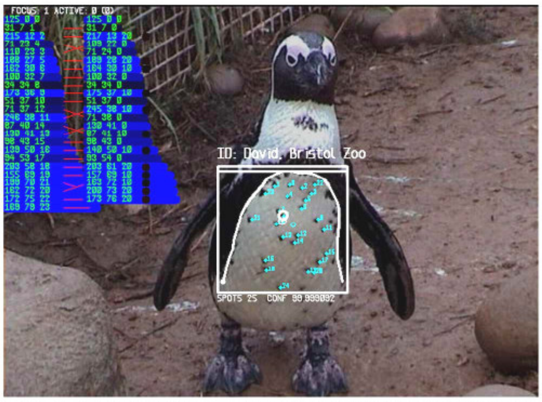


Figure 1. *Identification of an African Penguin by its Chest Pattern:* Screenshot of Software Prototype; African penguins carry a unique pattern of black spots on their chest. The detection of the chest location and the decomposition of the spot pattern allow checking a photographed individual (here penguin 'David' from Bristol Zoo) against a population database. (figure source [18], [19])