INF 5300 Video Analysis

Detecting motion

Asbjørn Berge

SINTEF Technology for a better society 1

2

Outline

Introduction to the motion detection
- What is motion detection?
- Assumptions and challenges
- Applications

Algorithms
- Background subtraction
- Frame differencing
- Mixture of gaussians
- Nonparametric approaches

Improvements
- Handling shadows and ghosts
- Object association
- Rudimentary tracking by blob association

SINTEF Technology for a better society

3

Outline

Introduction to the motion detection

- What is motion detection?
- Assumptions and challenges
- Applications

Algorithms

- Background subtraction
- Frame differencing
- Mixture of gaussians
- Nonparametric approaches

Improvements

- Handling shadows and ghosts
- Object association
- Rudimentary tracking by blob association

SINTEF Technology for a better society

Problem definition

- 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

SINTEF Technology for a better society 4

The problem

- Main goal: given a frame sequence from a fixed camera, detecting all the foreground objects
- Naive description of the approach: detecting the foreground objects as the difference between the current frame and an image of the scene's static background:

$$|frame_i - background_i| > Th$$

- First consequent problem: how to automatically obtain the image of the scene's static background?

Background subtraction compares an image with an estimate of the image as if it contained no objects of interest. It extracts foreground objects from regions where there is a significant difference between the observed and the estimated image.

The problem - requirements

- The background image is not fixed but must adapt to:
  - Illumination changes
    - gradual
    - sudden (such as clouds)
  - Motion changes
    - camera oscillations
    - high-frequencies background objects (such as tree branches, sea waves, and similar)
  - Changes in the background geometry
    - parked cars
    - objects left/removed

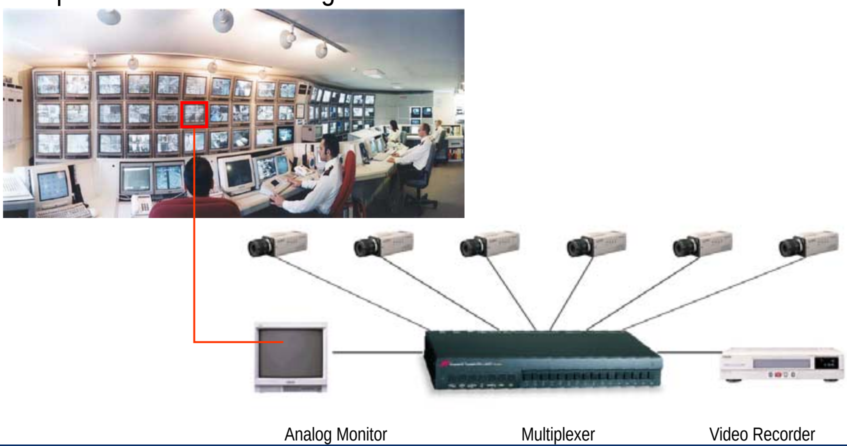
Video Analytics is everywhere

- **Traffic** (Intersections, Highways)
- **Sports Events** (many cameras)
- **Surveillance** (fixed cameras)
- Video editing (one camera, many viewpoints)
- User interaction

SINTEF Technology for a better society

First Generation (80's)

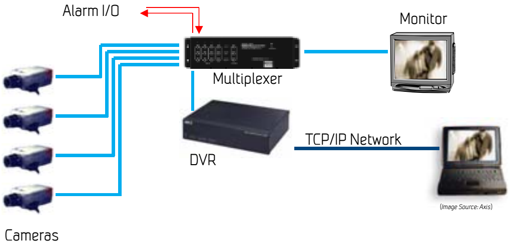
- Analog CCTV Cameras + Analog Video Tapes Recording
- Complete Human Monitoring



SINTEF Technology for a better society

Second Generation (90's)

- Analog CCTV Cameras + Digital Video Recording
- Automatic Event Detection based on Motion Detection

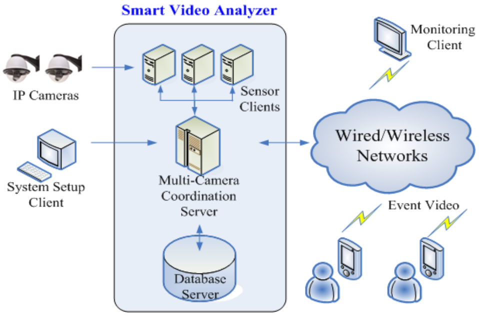


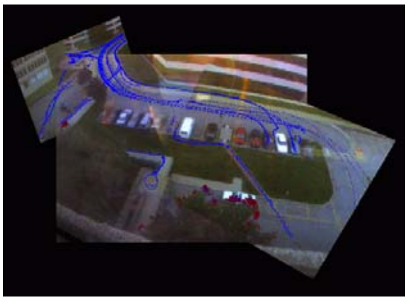
(Image Source: Mobotix)

SINTEF Technology for a better society

Third Generation (00's)

- Complete Digital Solutions: IP Cameras + Network Video Recording
- Multi-Camera Cooperated Object Detection, Tracking, and Event Analysis





(Image Source: MIT CSAIL)

SINTEF Technology for a better society

11

Outline

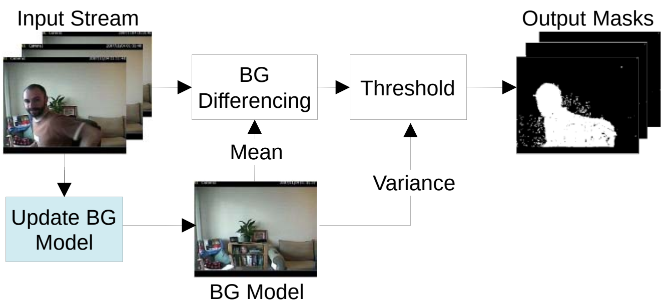
Introduction to the motion detection
- What is motion detection?
- Assumptions and challenges
- Applications

Algorithms
- Background subtraction
- Frame differencing
- Mixture of gaussians
- Nonparametric approaches

Improvements
- Handling shadows and ghosts
- Object association
- Rudimentary tracking by blob association

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



Average image

- What can we do with this?


Background Subtraction

-

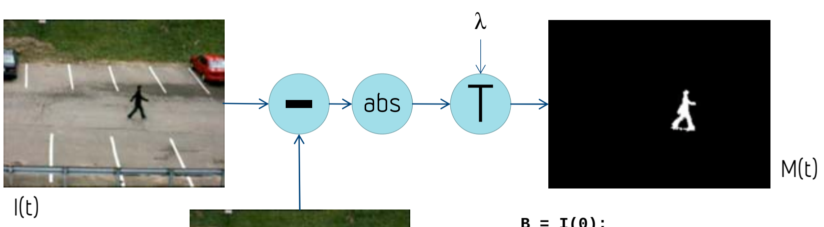
=

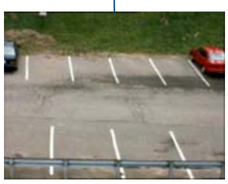
Typical uses for motion detection

- Realtime analysis of video from surveillance cameras
- Up to 30 FPS from megapixel cameras
  - Not much time for complex algorithms
    - Not much changes in the scene in 1/30th second (temporal consistency)
    - Usually static cameras – background is not moving
  - Change in background is
    - imaging noise -> Gaussian
    - "Natural" changes -> Periodical /repeated

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.





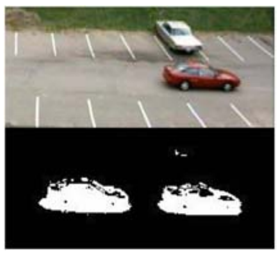
```
B = I(0);
…
loop time t
I(t) = next frame;
diff = abs[B – I(t)];
M(t) = threshold(diff,λ);
…
end
```

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.

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

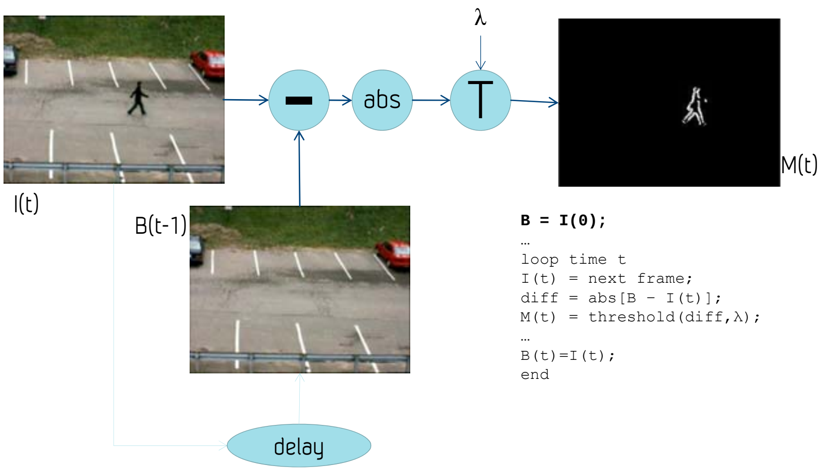


More observations

- 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.

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
```

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)

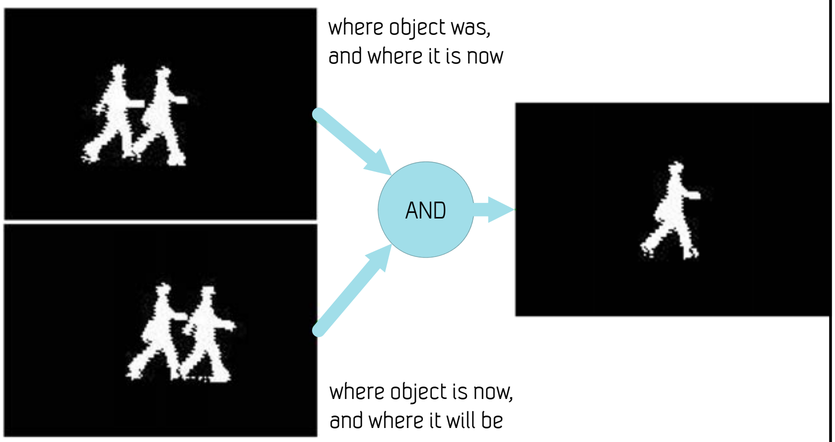
more complete object silhouette, but two copies
(one where object used to be, one where it is now).

Three frame differencing

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

D(-15)

where object was, and where it is now



D(+15)

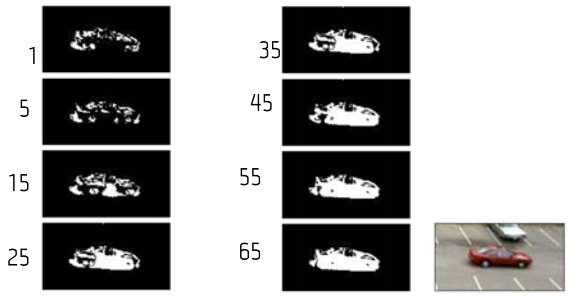
where object is now, and where it will be

Three frame differencing

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

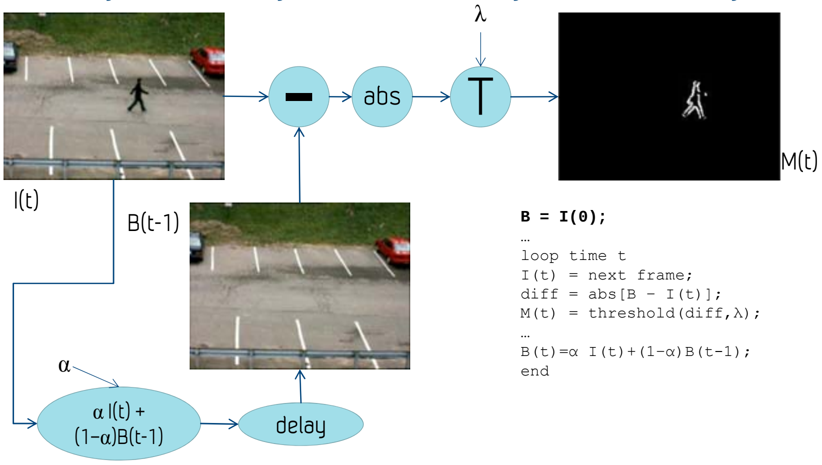
frames skipped

This worked well for the person



Adaptive background subtraction

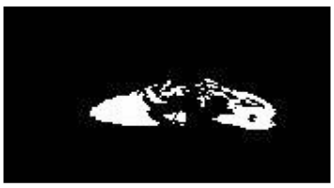
- Current image is "blended" into the background model with parameter α
- α = 0 yields simple background subtraction, α = 1 yields frame differencing




```
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
```

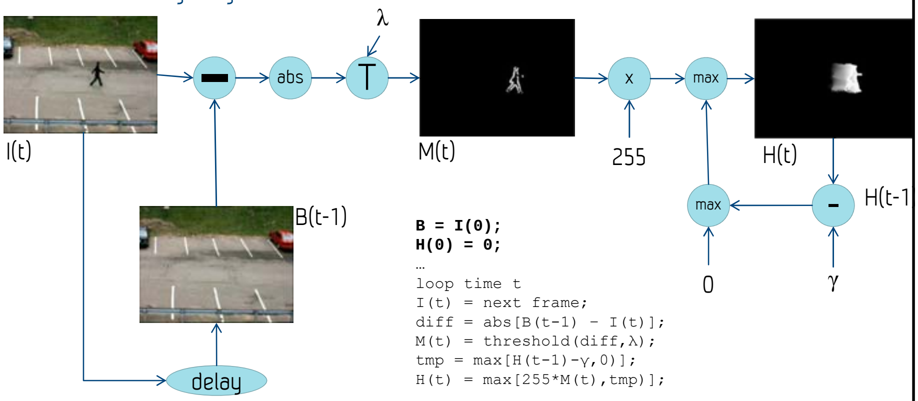

Adaptive BG observations

- 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

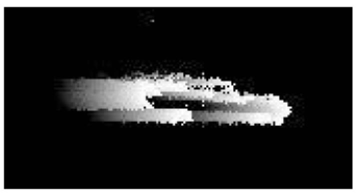
- Motion images are combined with a linear decay term also known as motion history images




```
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

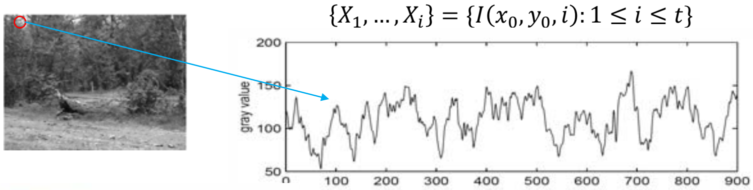
- Persistant frame differencing is also responsive to changes in illumination and camera motion, and stopped objects / ghosts also fade away.
- Objects leave behind gradually fading trails of pixels. The gradient of this trail indicates the apparent direction of object motion in the image.
- Although 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.

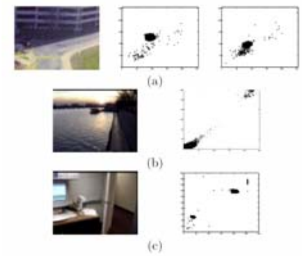


Statistical background modeling

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

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

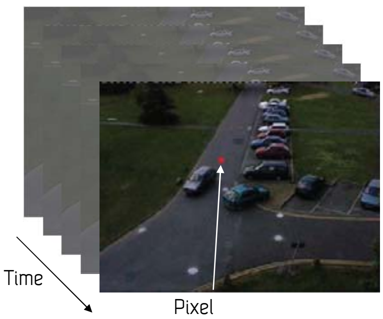


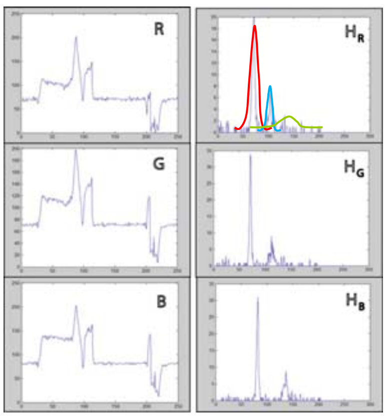


•Pixel history is likely to be multiple clusters
•Maintain an adaptive color model at each pixel based on a mixture of Gaussians (typically up to 5 components)

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.

Background subtraction – modeling by multiple gaussians





Sequence of pixel values

Histograms

X
p(

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

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

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

SINTEF Technology for a better society

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 $D^2_m(\vec{x}^{(t)}) = \vec{\delta}^T_m \vec{\delta}_m / \hat{\sigma}^2_m$

SINTEF Technology for a better society

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

SINTEF Technology for a better society

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, $\alpha = 1/T$

SINTEF Technology for a better society

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:

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

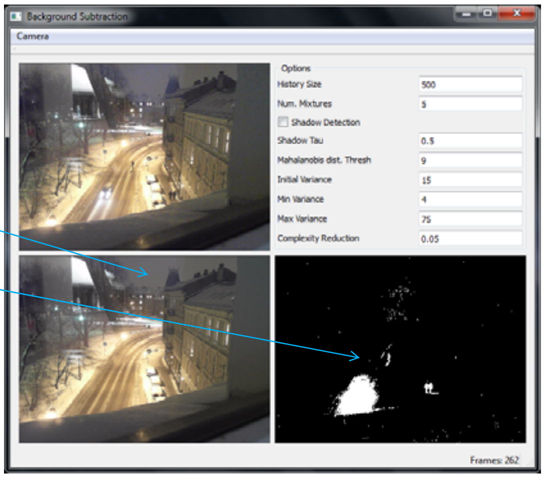
with components sorted descending by weight

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

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

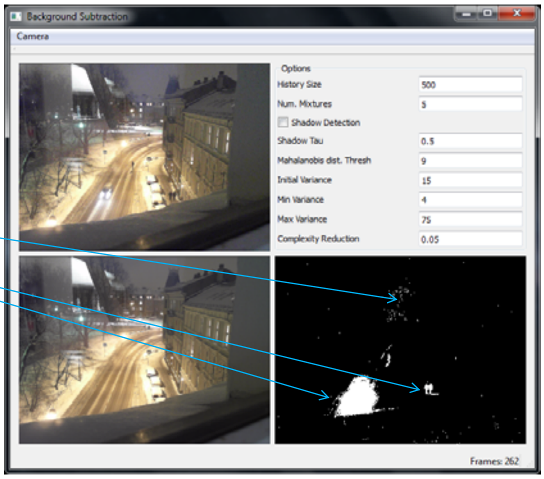
Determining the background model

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



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



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.

Statistical Background Modeling (nonparametric)

- Non-parametric color distribution, estimated via kernel density estimation
- the background model is given by the histogram of N previous values smoothed as a Kernel Density Estimator
- KDE requires only a single value, σ, to be estimated
- however:
  - subtraction requires computing *N* Gaussian values, with *N* typically *50-100*
    - much storage space
    - subtraction is time-consuming (lookup tables needed)
- updating the KDE smoothing factor is heavy
- Ahmed Elgammal, David Harwood, Larry Davis "Non-parametric Model for Background Subtraction", 6th European Conference on Computer Vision. Dublin, Ireland, June 2000.

Statistical background modeling

- Use optic flow *u,v* values at each pixel, rather than intensity/color
- R.Pless, J.Larson, S.Siebers, B.Westover, "Evaluation of Local Models of Dynamic Backgrounds," IEEE Computer Vision and Pattern Recognition (CVPR), June 2003

43

Outline

Introduction to the motion detection
- What is motion detection?
- Assumptions and challenges
- Applications

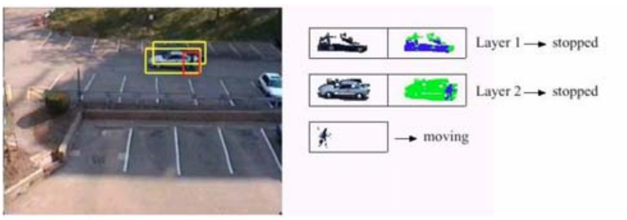
Algorithms
- Background subtraction
- Frame differencing
- Mixture of gaussians
- Nonparametric approaches

Improvements
- Handling shadows and ghosts
- Object association
- Rudimentary tracking by blob association

SINTEF Technology for a better society

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.



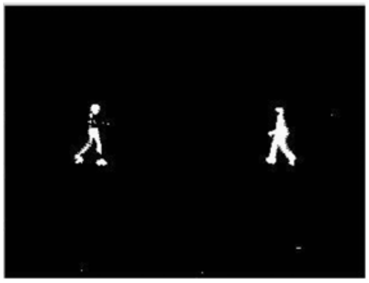
Allow blobs to be layered, so that stopped blobs can be considered part of the background for new object detection, but they will not leave behind ghosts when they start moving again.

Store stacks of background

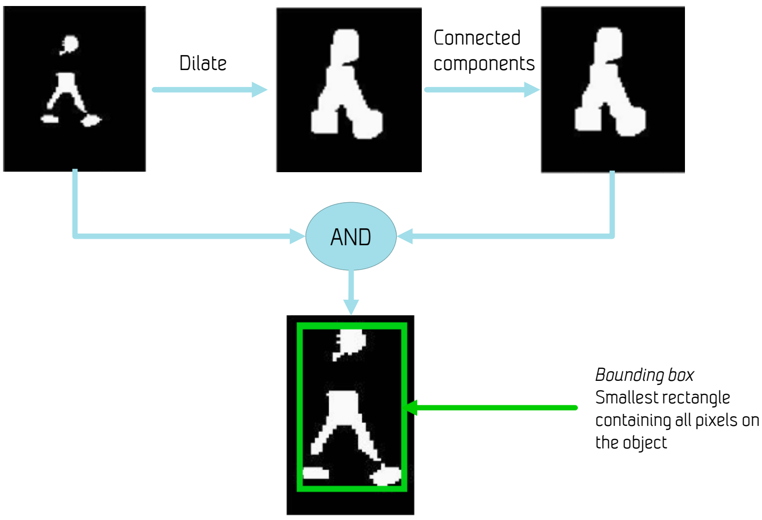
SINTEF Technology for a better society 44

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

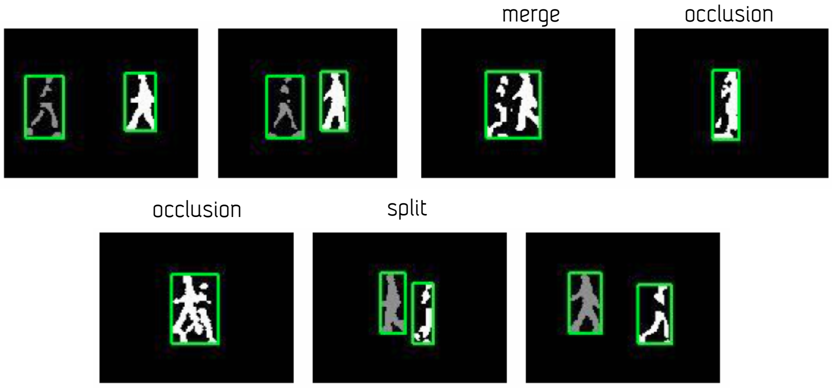


Gap Spanning Connected Components



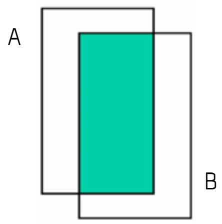
Blob merge/split

- 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:

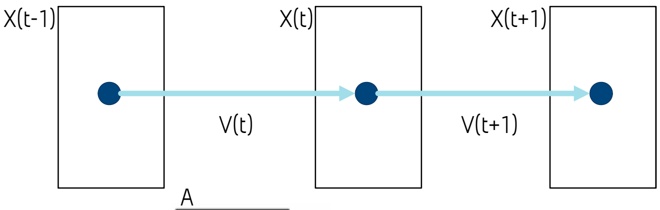


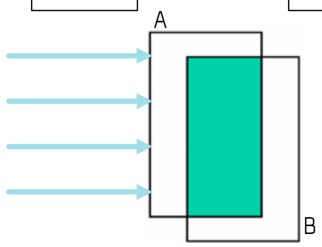
$$score = \frac{2 * area(A \text{ and } B)}{area(A) + area(B)}$$

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)





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

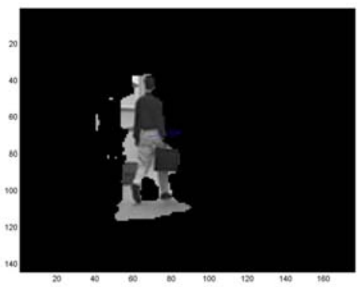
A = bounding box at time t, adjusted by velocity V(t)
B = bounding box at time t+1

SINTEF Technology for a better society 49

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.

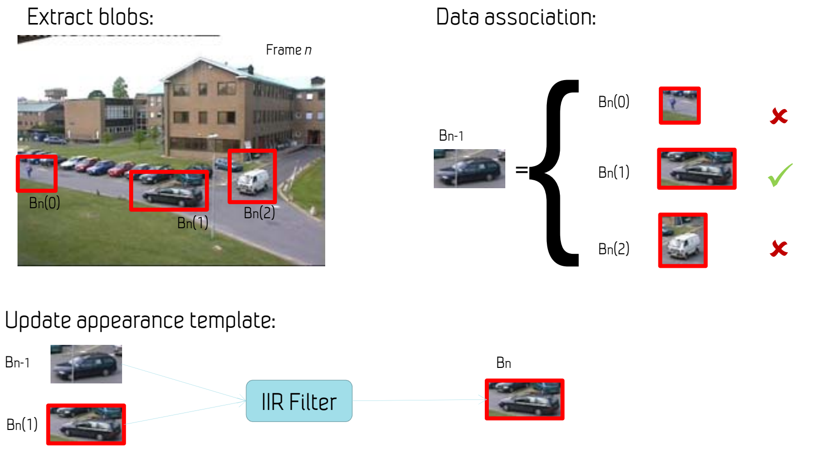
This reduces the amount of computation necessary to extract motion information. (useful in coding where layers are used)



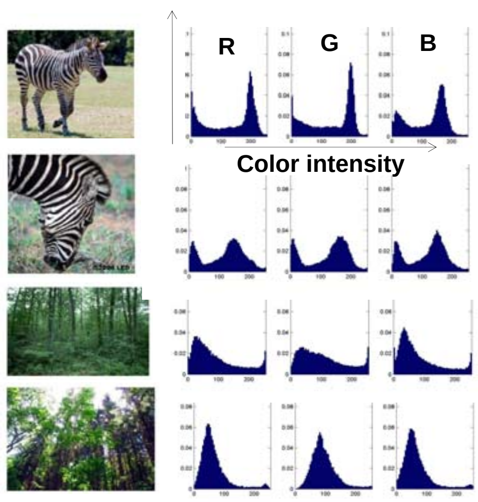
SINTEF Technology for a better society

Appearance information

- Correlation of image templates between frames is an obvious choice (moving blobs)



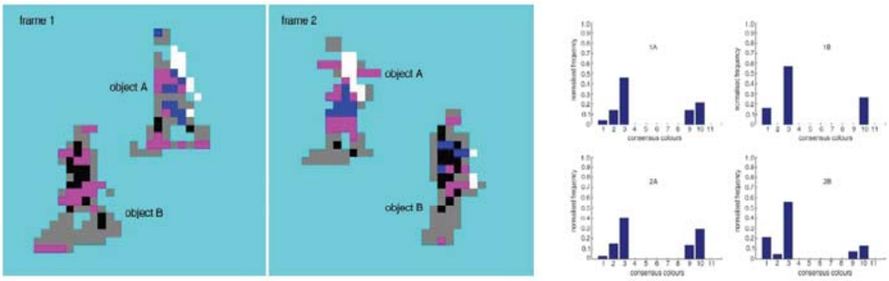
Color as a low-level cue for CBIR



- Color histogram is used to describe the color fingerprint of the object
  - Spatially invariant
  - Simple and efficient to compute
  - Through some quantization, it also provides some invariance to changes in color appearance

Color Descriptor

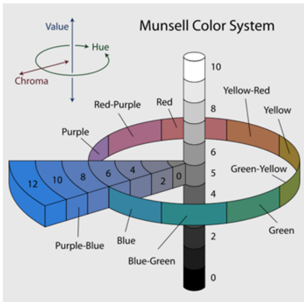
A. Gilbert, R. Bowden ; *[Incremental, Scalable Tracking of Objects Inter Camera]()* -; In Computer Vision and Image Understanding CVIU, Vol 111 Pages 43-58, 2008

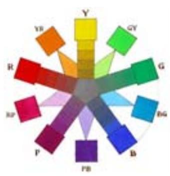


Describe colors using some kind of perceptive space. Munsell consensus colors seems like a reasonable choice

Consensus-Colour Conversion of Munsell colour space

- Breaks RGB color into 11 basic colors
- Each basic color represents a perceptual color category established through a physiological study of how humans categorize color.
- Define some threshold for chroma as *neutral,* and segment the rest of the colors into hues
- Practical size of histogram for object description (11 bins)
- Can be calculated by lookup from CIE Lab values
- Paul Centore, *Colour Theory for Painters,* http://www.99main.com/~centore/





Shadow detection

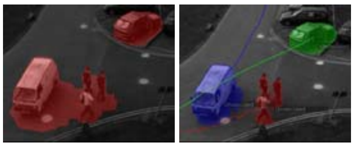
- Shadows and reflections are complex sources of noise for motion detection
  - Undersegmentation
  - Spurious detections
  - Object tracking and classification much harder
- A moving point may be a shadow for example if
  1. It is darker than the "shadowed" background

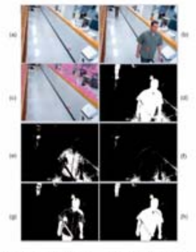
     $$\alpha < \frac{Y_p}{Y_{bg}} < \beta, 0 < \alpha < 1, 0 < \beta < 1, Y(\cdot) = \frac{R(\cdot) + G(\cdot) + B(\cdot)}{3}$$

  2. Ratios between color channelsa are approximately constant

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

  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!)



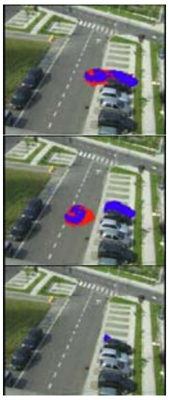


Prati, I. Mikic, M. Trivedi, R. Cucchiara, "*Detecting Moving Shadows: Algorithms and Evaluation*," IEEE Trans. on Pattern Analysis and Machine Intelligence, 2003
Nicolas Martel-Brisson and Member-Andre Zaccarin :
"*Learning and Removing Cast Shadows through a Multidistribution Approach*", IEEE Trans. Pattern Anal. Mach. Intell. 29, 7 (Jul. 2007)

SINTEF Technology for a better society 55

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

SINTEF Technology for a better society 56

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 57

REFERENCES

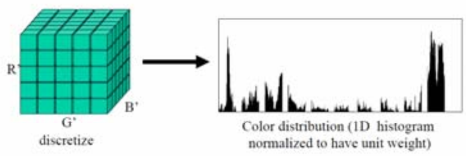
- Background subtraction
  - Z.Zivkovic, Improved adaptive Gausian mixture model for background subtraction, International Conference Pattern Recognition, UK, August, 2004, http://www.zoranz.net/Publications/zivkovic2004ICPR.pdf.
  - Radke, R.J.; Andra, S.; Al-Kofahi, O.; Roysam, B.; Image change detection 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

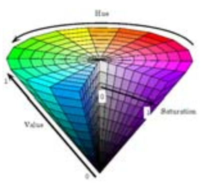
FIN.

Robert Collins
CSE486, Penn State

Appearance via Color Histograms



Color distribution (1D histogram normalized to have unit weight)



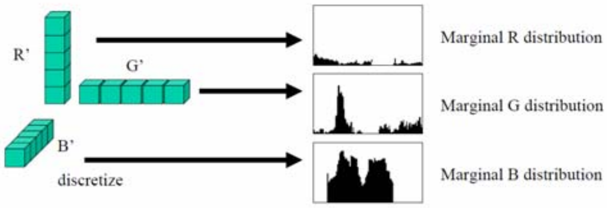
R' = R << (8 - nbits)
G' = G << (8 - nbits)
B' = B << (8 - nbits)

Total histogram size is (2^(8-nbits))^3

example, 4-bit encoding of R,G and B channels yields a histogram of size 16*16*16 = 4096.

Smaller Color Histograms

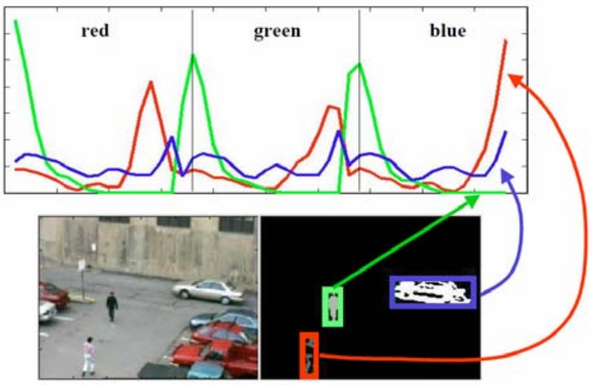
Histogram information can be much much smaller if we are willing to accept a loss in color resolvability.



R' = R << (8 - nbits)
G' = G << (8 - nbits)
B' = B << (8 - nbits)

Total histogram size is 3*(2^(8-nbits))

example, 4-bit encoding of R,G and B channels yields a histogram of size 3*16 = 48.



Adaptive statistic background models