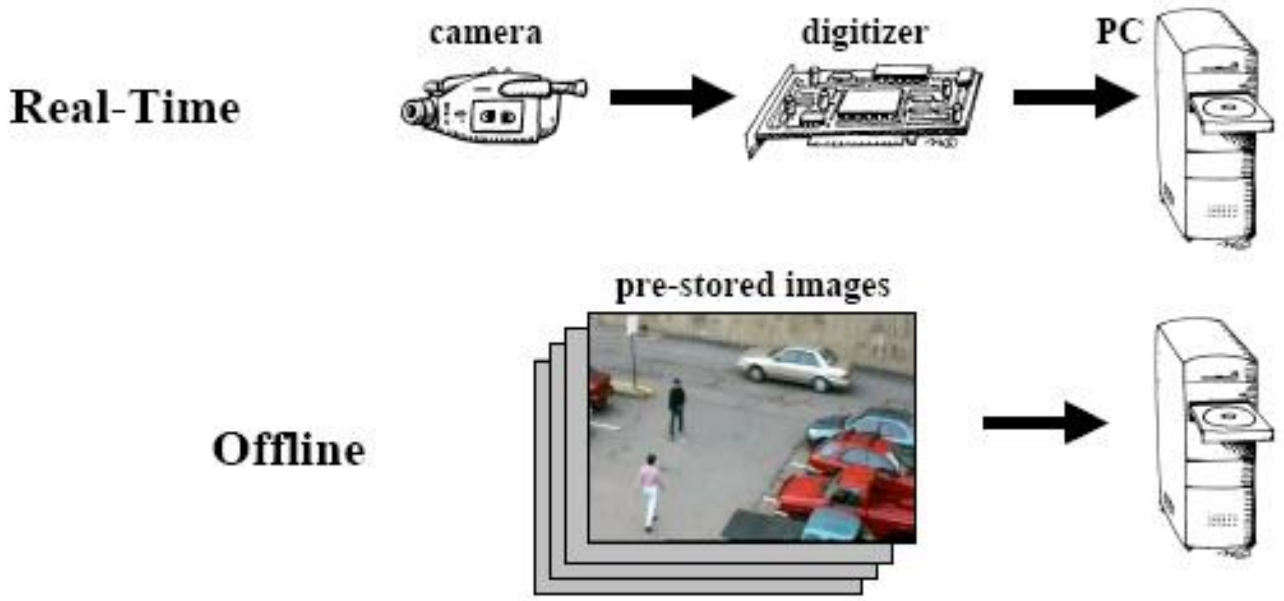
Video Processing Basics



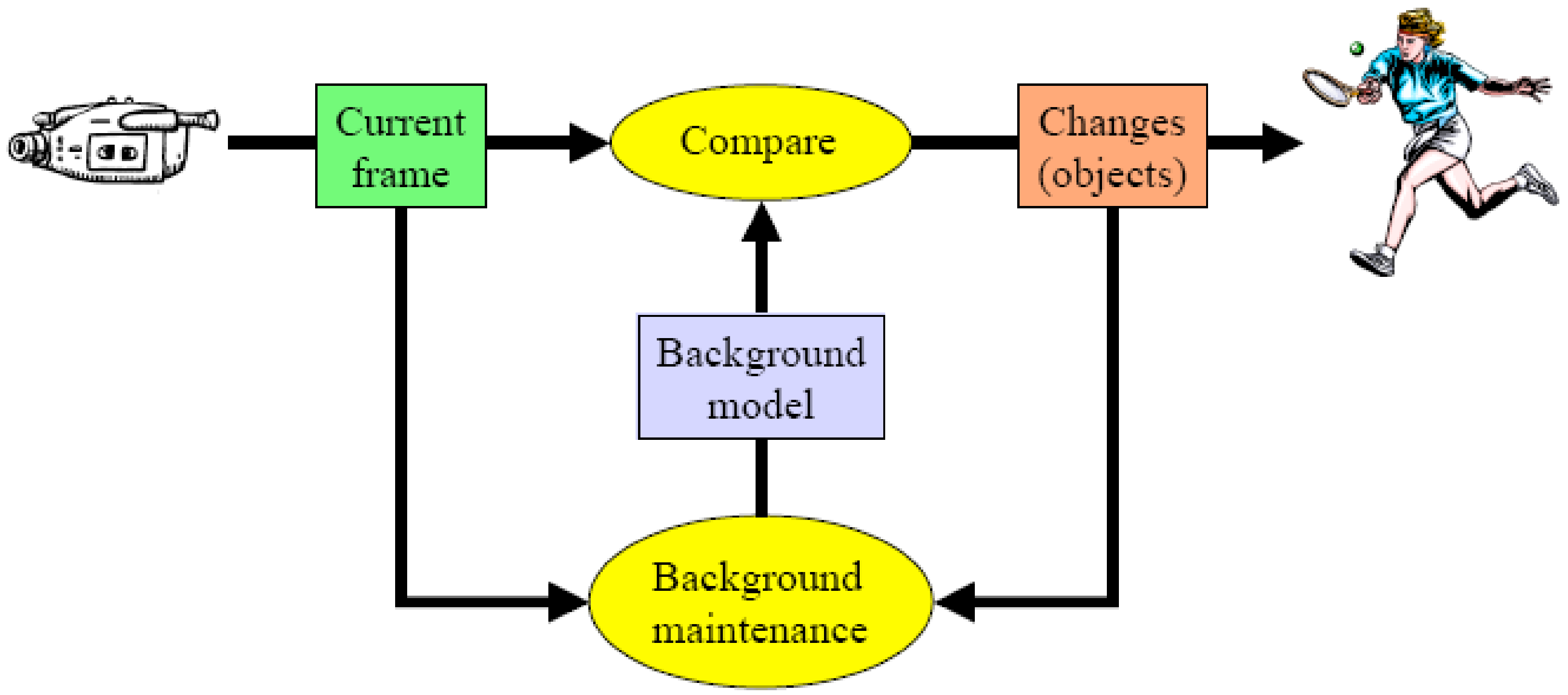
Frames come in 30 times per second. This is not much time to process each image. Real-time algorithms therefore tend to be very simple.

One of the main features of video imagery is the temporal consistency from frame to frame. Not much changes during 1/30 of a second!

Video Change Detection

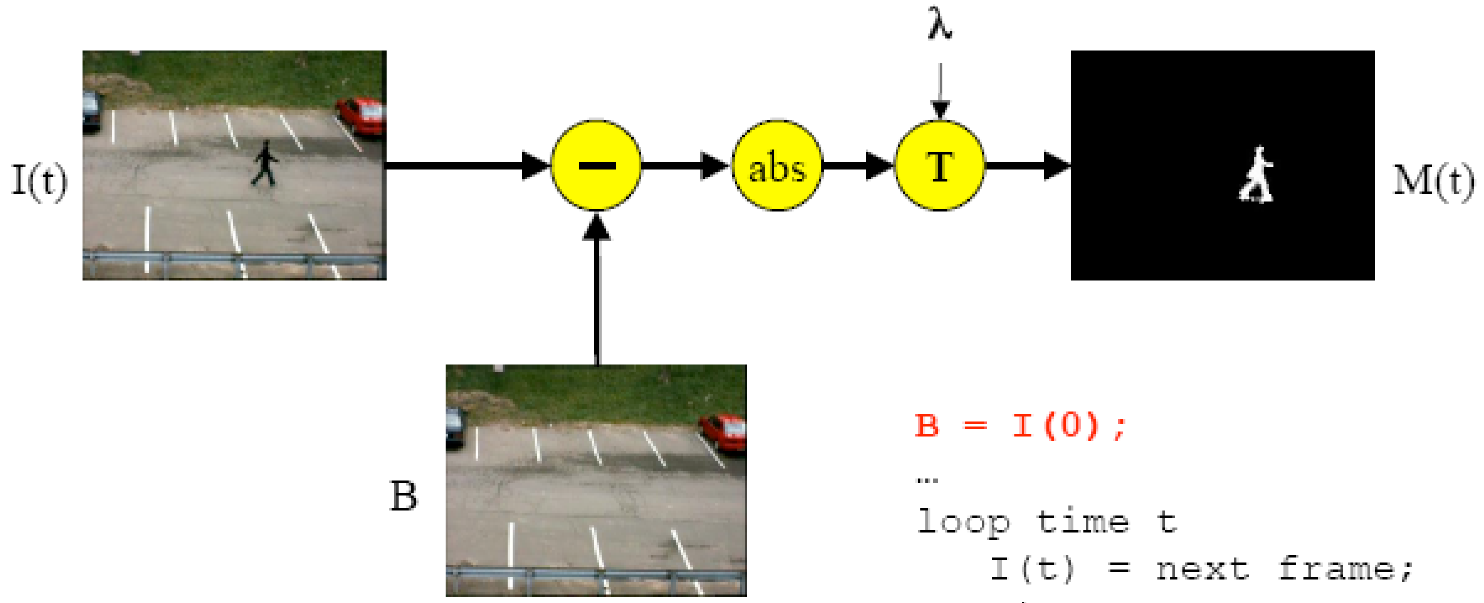
Assumption: objects that move are important (e.g. people and vehicles)

Basic approach: maintain a model of the static background. Compare the current frame with the background to locate moving foreground objects.



Simple Background Subtraction

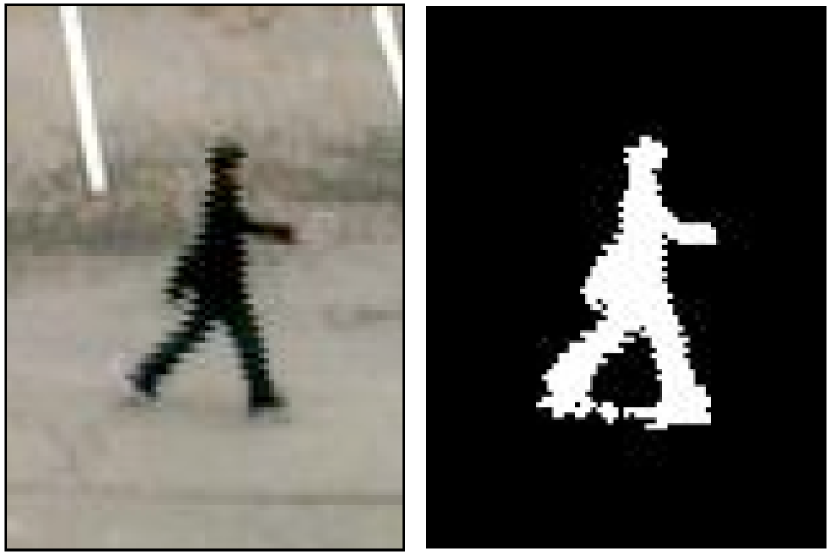
- Background model is a static image (assumed to have no objects present).
- Pixels are labeled as object (1) or not object (0) based on thresholding the absolute intensity difference between current frame and background.

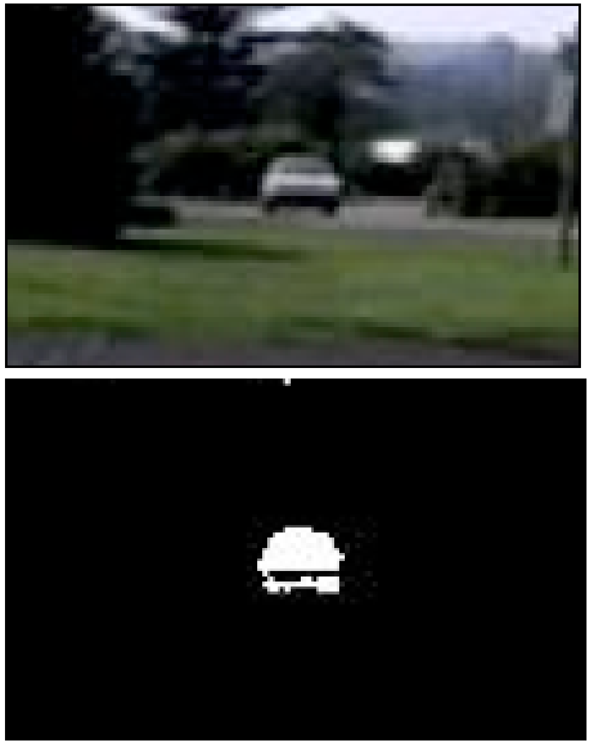


```
B = I(0);
…
loop time t
    I(t) = next frame;
    diff = abs[B - I(t)];
    M(t) = threshold(diff,λ);
    …
end
```

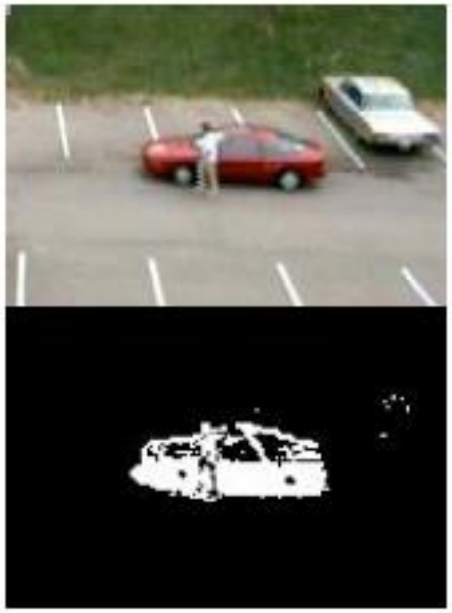
BG-Sub Samples 1

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

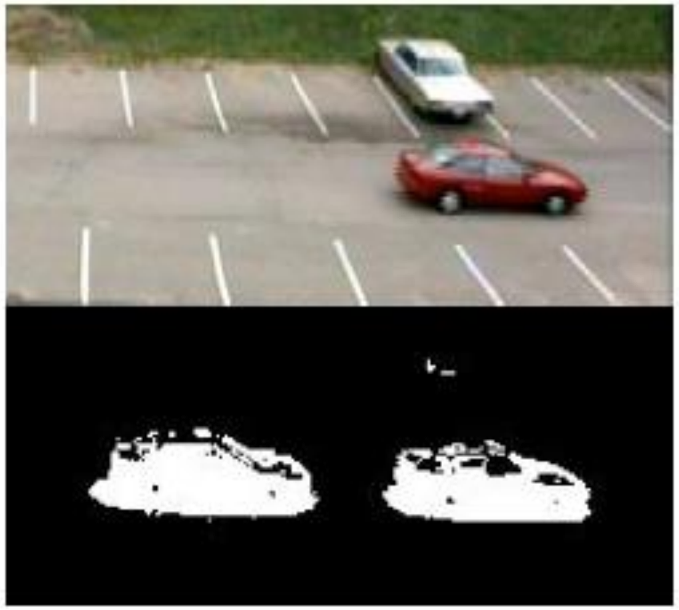




BG-Sub Samples 2

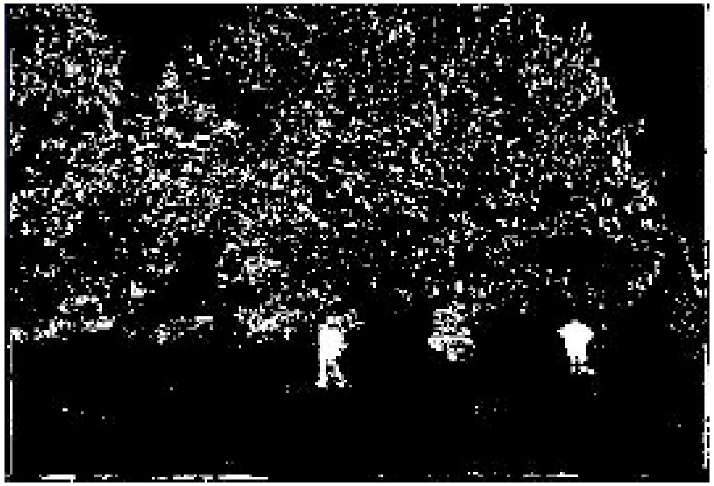


Objects that enter the scene and stop continue to be detected, making it difficult to detect new objects that pass in front of them.

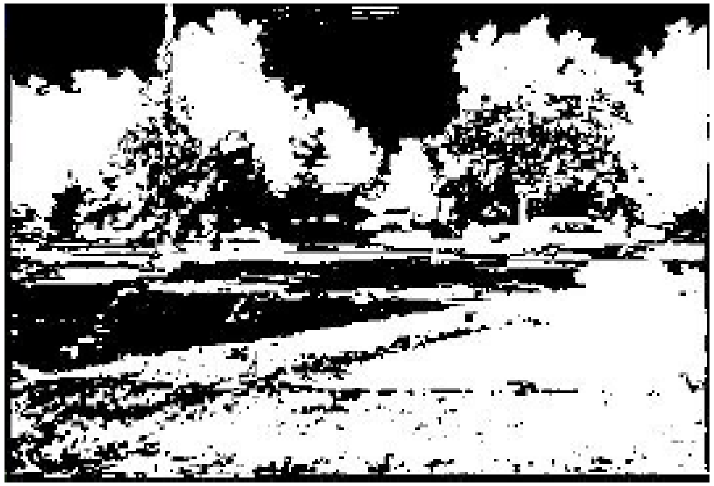


If part of the assumed static background starts moving, both the object and its negative ghost (the revealed background) are detected

BG-Sub Samples 3



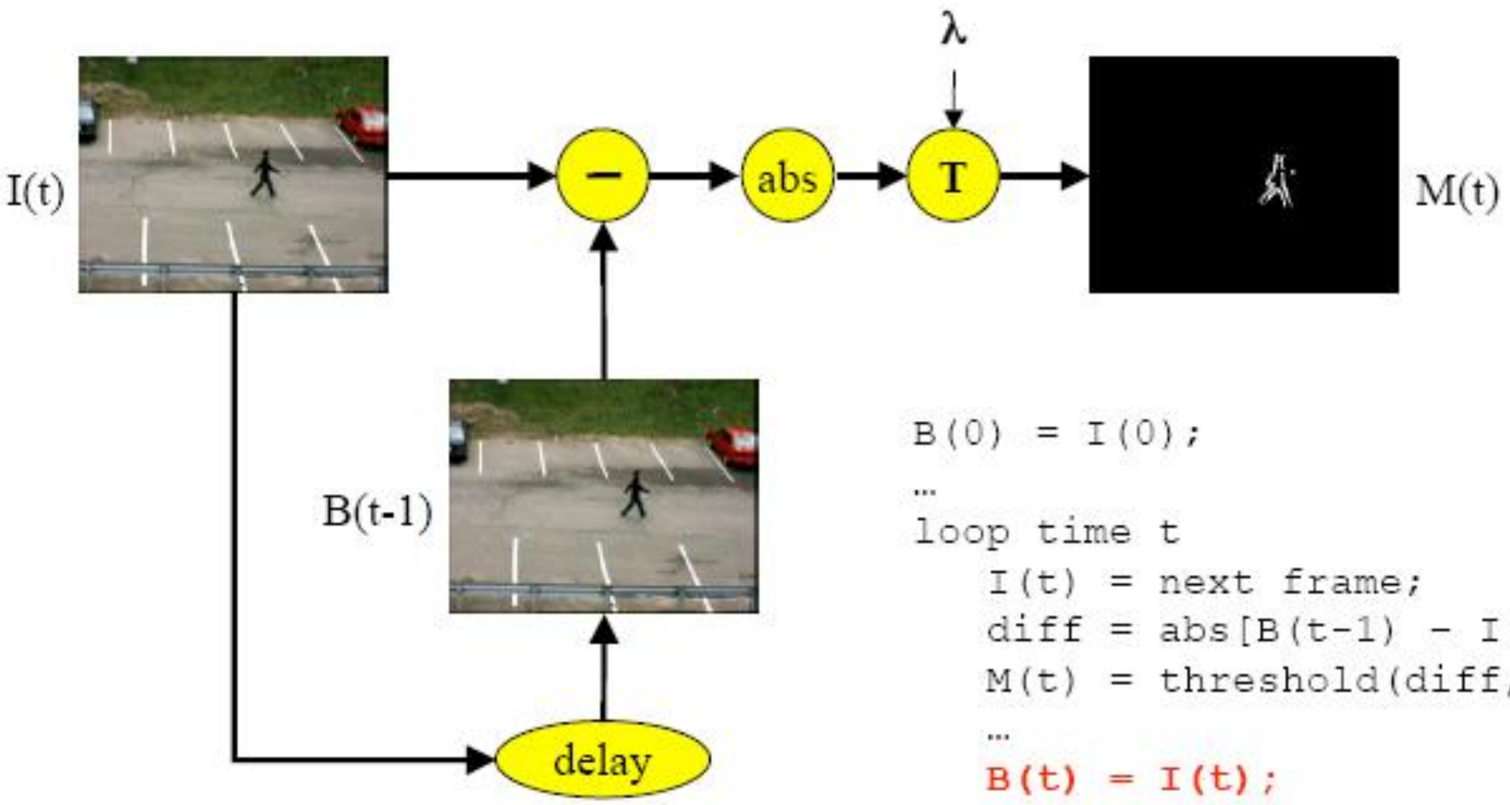
Background subtraction is sensitive to changing illumination and unimportant movement of the background (for example, trees blowing in the wind, reflections of sunlight off of cars or water).



Background subtraction cannot handle movement of the camera.

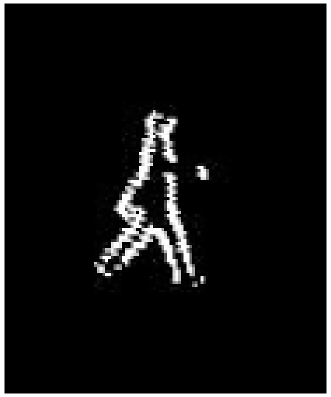
Simple Frame Differencing

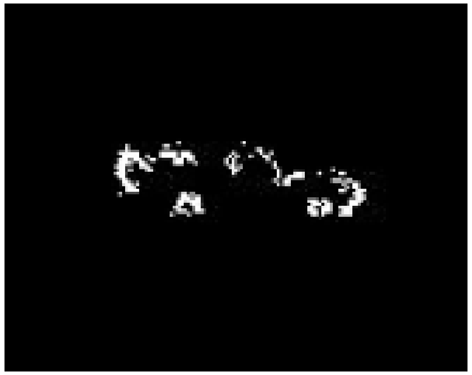
- Background model is replaced with the previous image.



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

SFD Samples





Frame differencing is very quick to adapt to changes in lighting or camera motion.

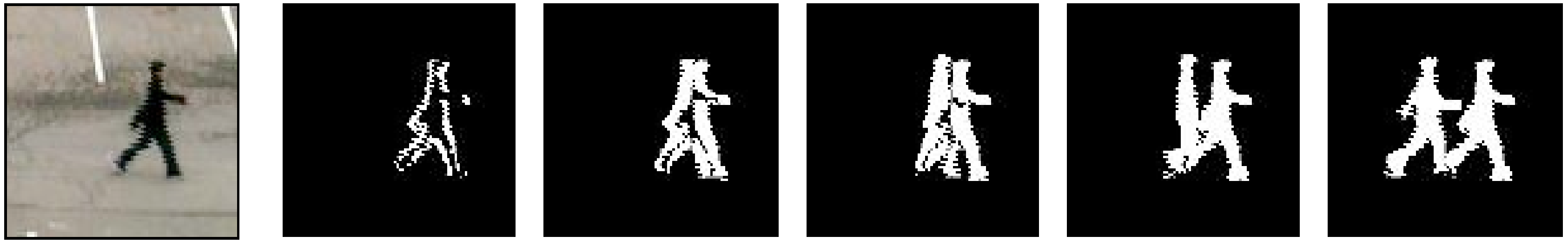
Objects that stop are no longer detected. Objects that start up do not leave behind ghosts.

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) ||



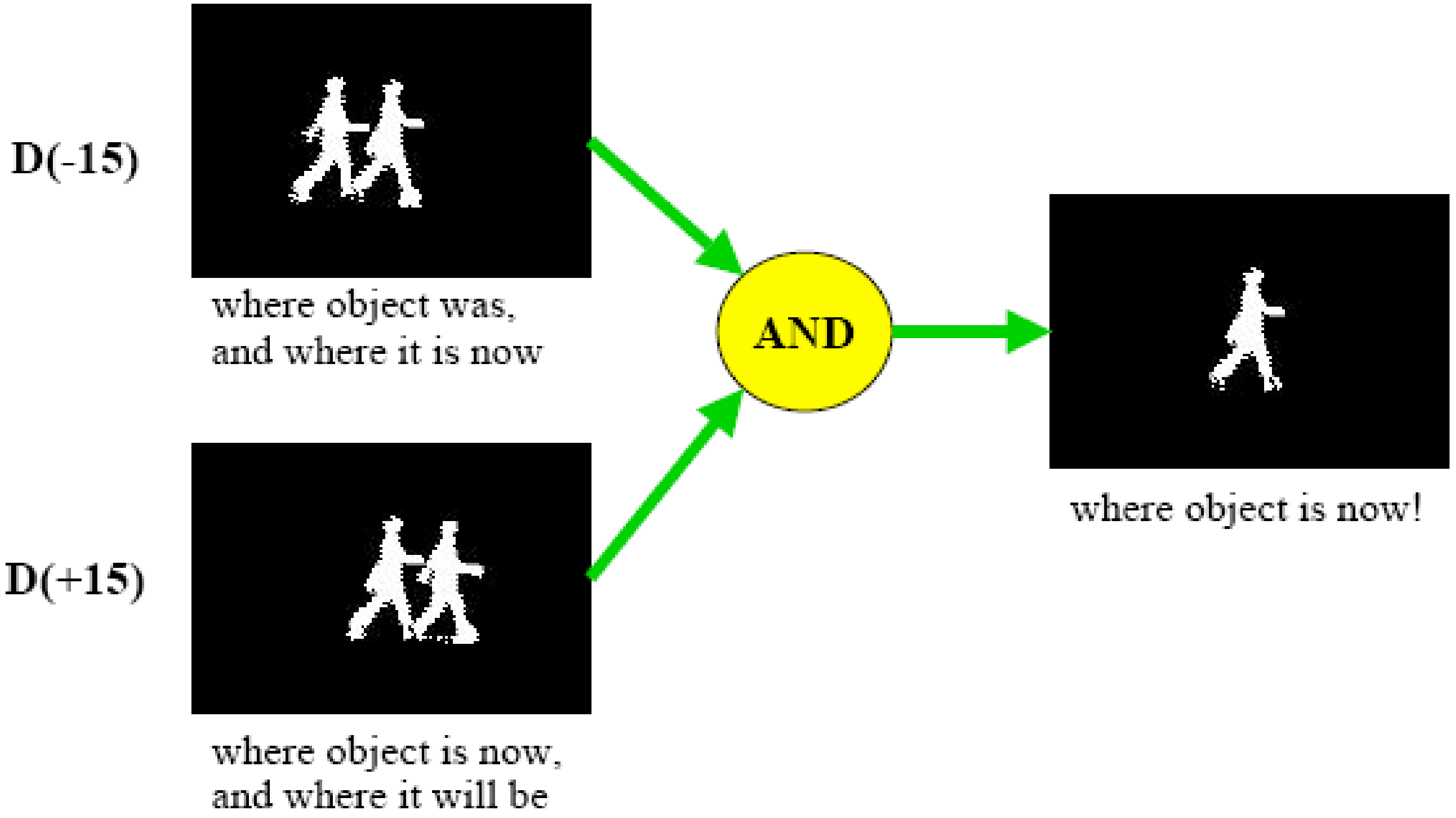
I(t) **D(-1)** **D(-3)** **D(-5)** **D(-9)** **D(-15)**

$\longrightarrow$

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

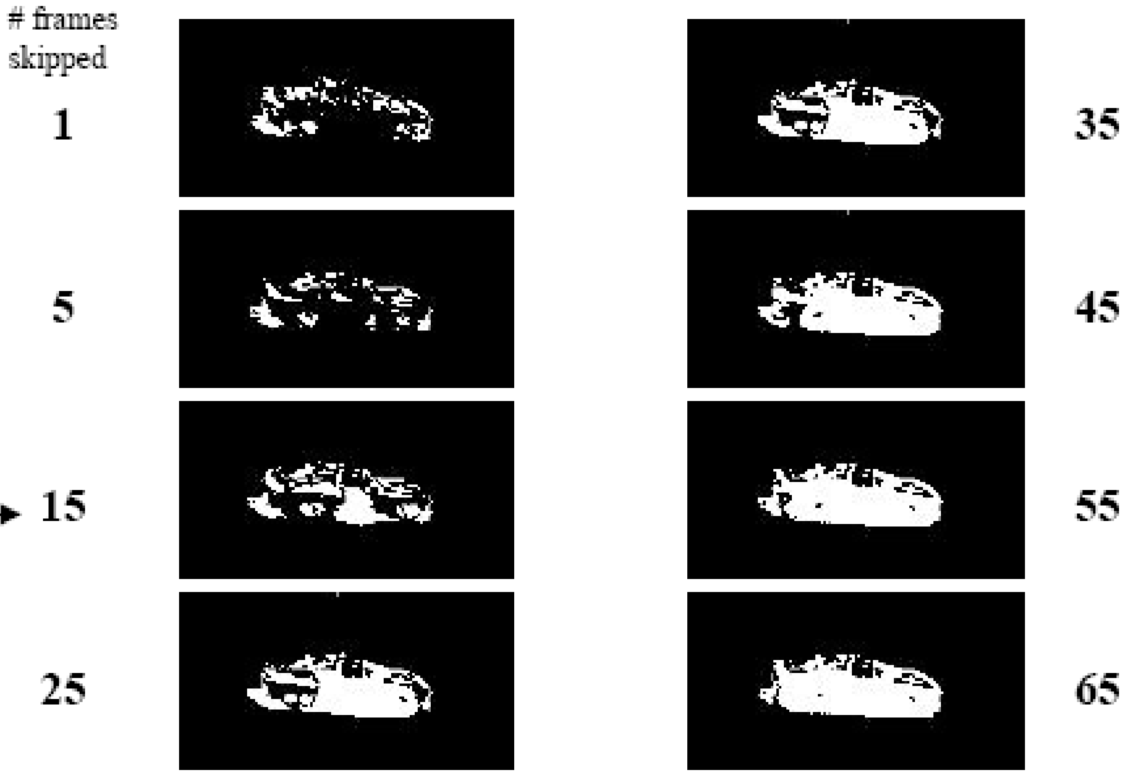
Three-Frame Differencing

The previous observation is the motivation behind three-frame differencing



Choice of Frame Rate

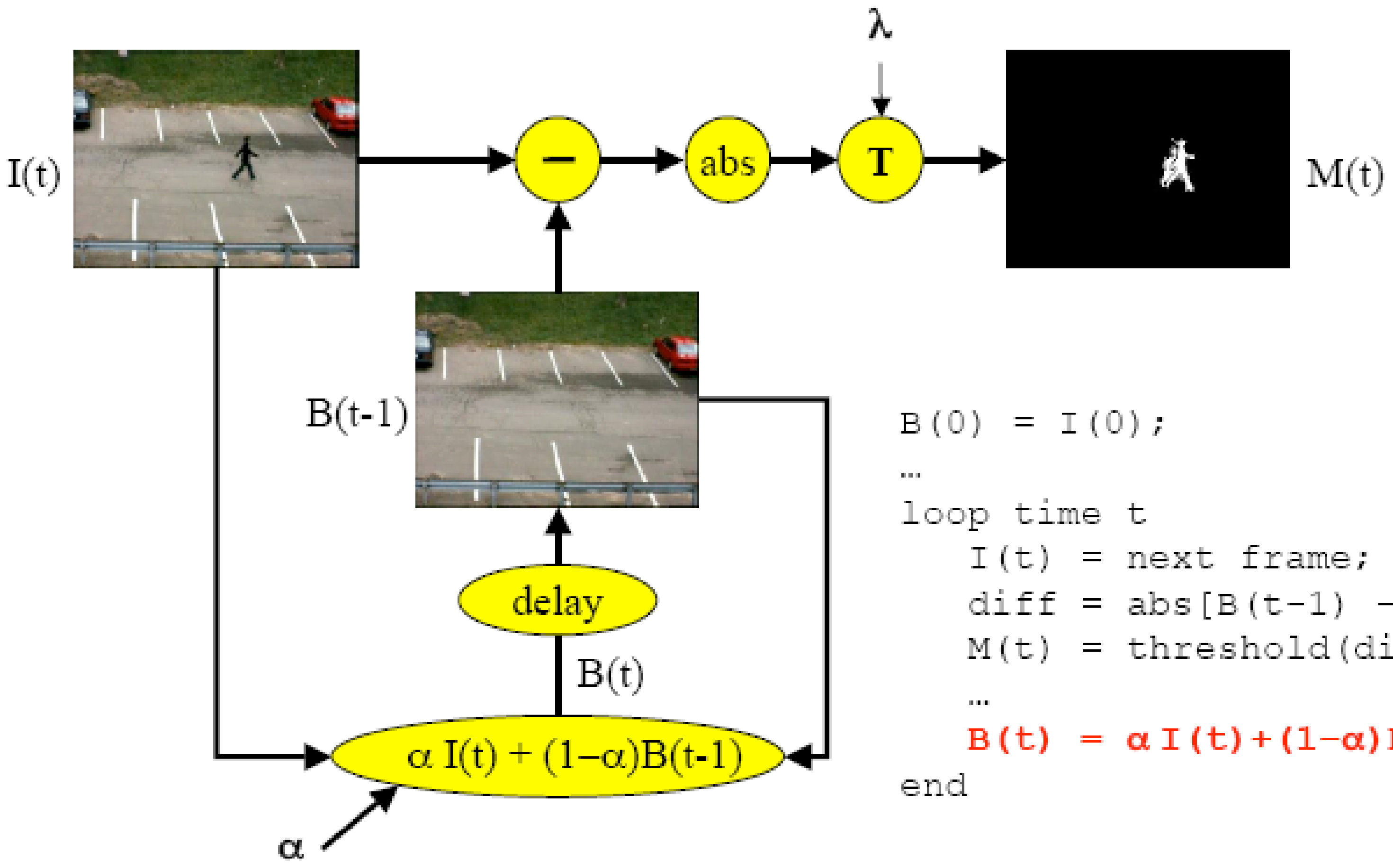
Choice of good frame-rate for three-frame differencing depends on the size and speed of the object



This worked well for the person

Adaptive Background Subtraction

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



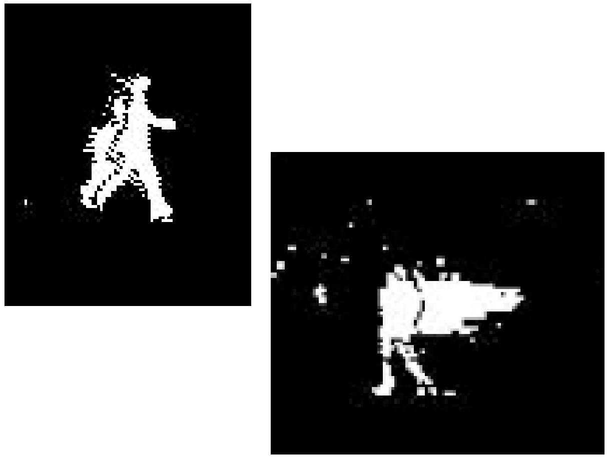
```
B(0) = I(0);
…
loop time t
    I(t) = next frame;
    diff = abs[B(t-1) - I(t)];
    M(t) = threshold(diff,λ);
    …
    B(t) = αI(t)+(1–α)B(t-1);
end
```

ABS Samples

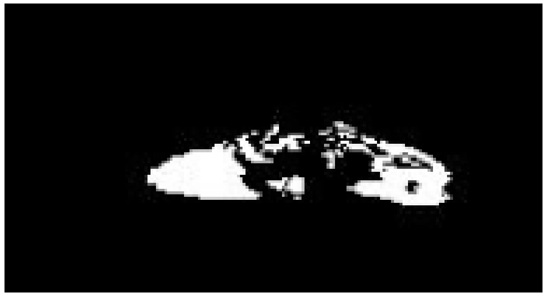
Adaptive background subtraction is more responsive to changes in illumination and camera motion.

Fast small moving objects are well segmented, but they leave behind short "trails" of pixels.

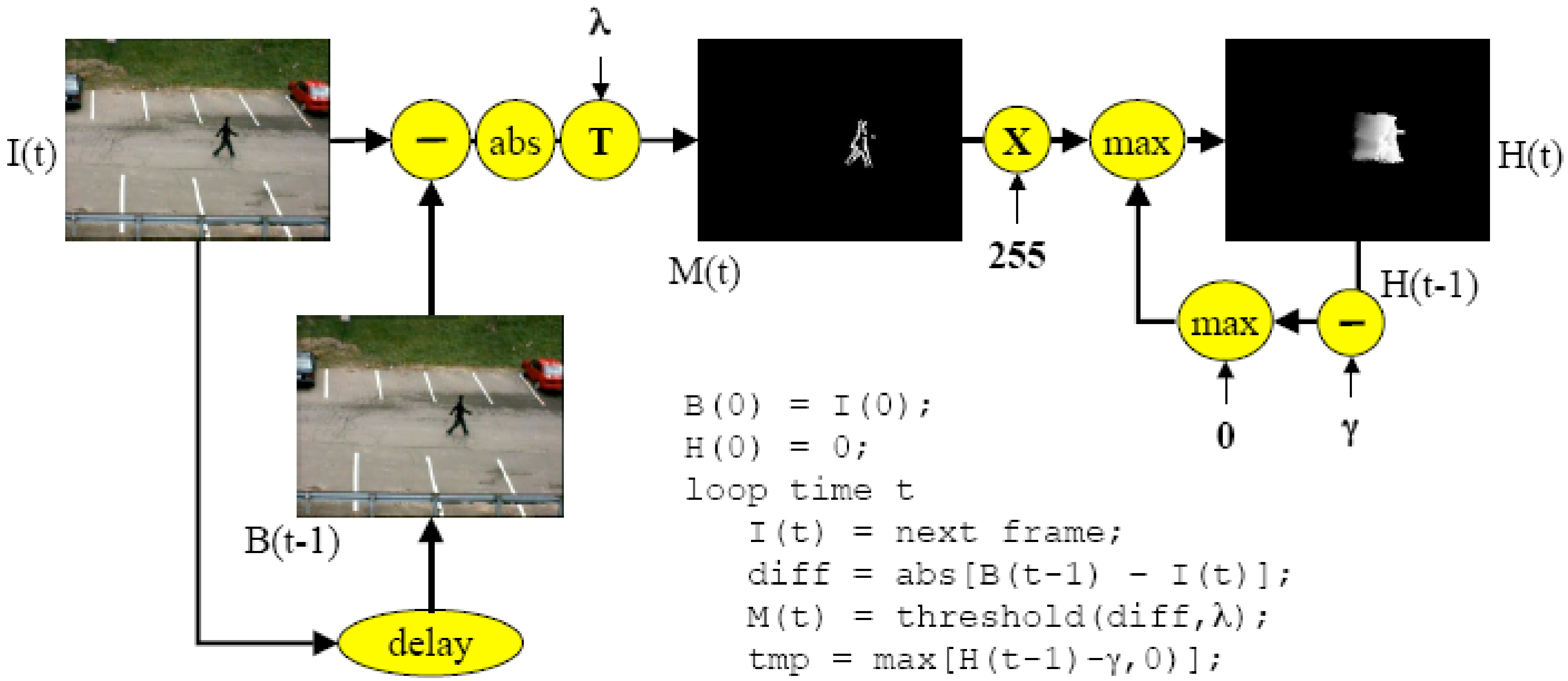
Objects that stop, and ghosts left behind by objects that start, gradually fade into the background.



The centers of large slow moving objects start to fade into the background too! This can be "fixed" by decreasing the blend parameter A, but then it takes longer for stopped/ghost objects to disappear.



Motion History Images



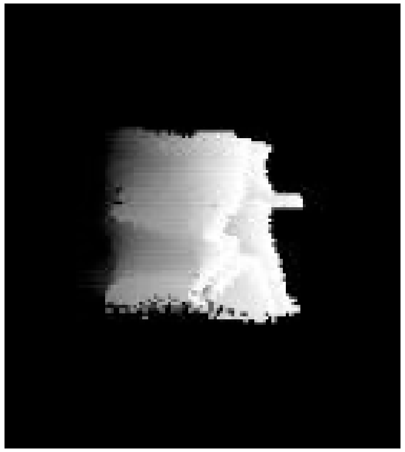
```
B(0) = I(0);
H(0) = 0;
loop time t
    I(t) = next frame;
    diff = abs[B(t-1) - I(t)];
    M(t) = threshold(diff,λ);
    tmp = max[H(t-1)-γ,0)];
    H(t) = max[255*M(t),tmp)];
    …
    B(t) = I(t);
end
```

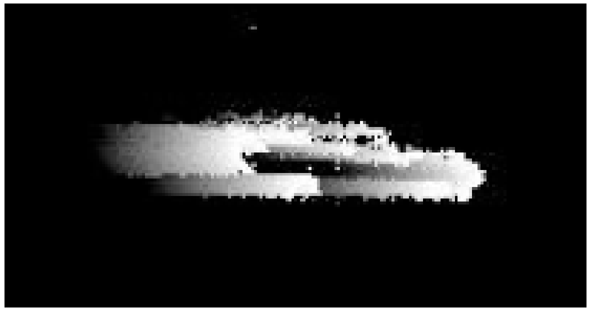
MHI Samples

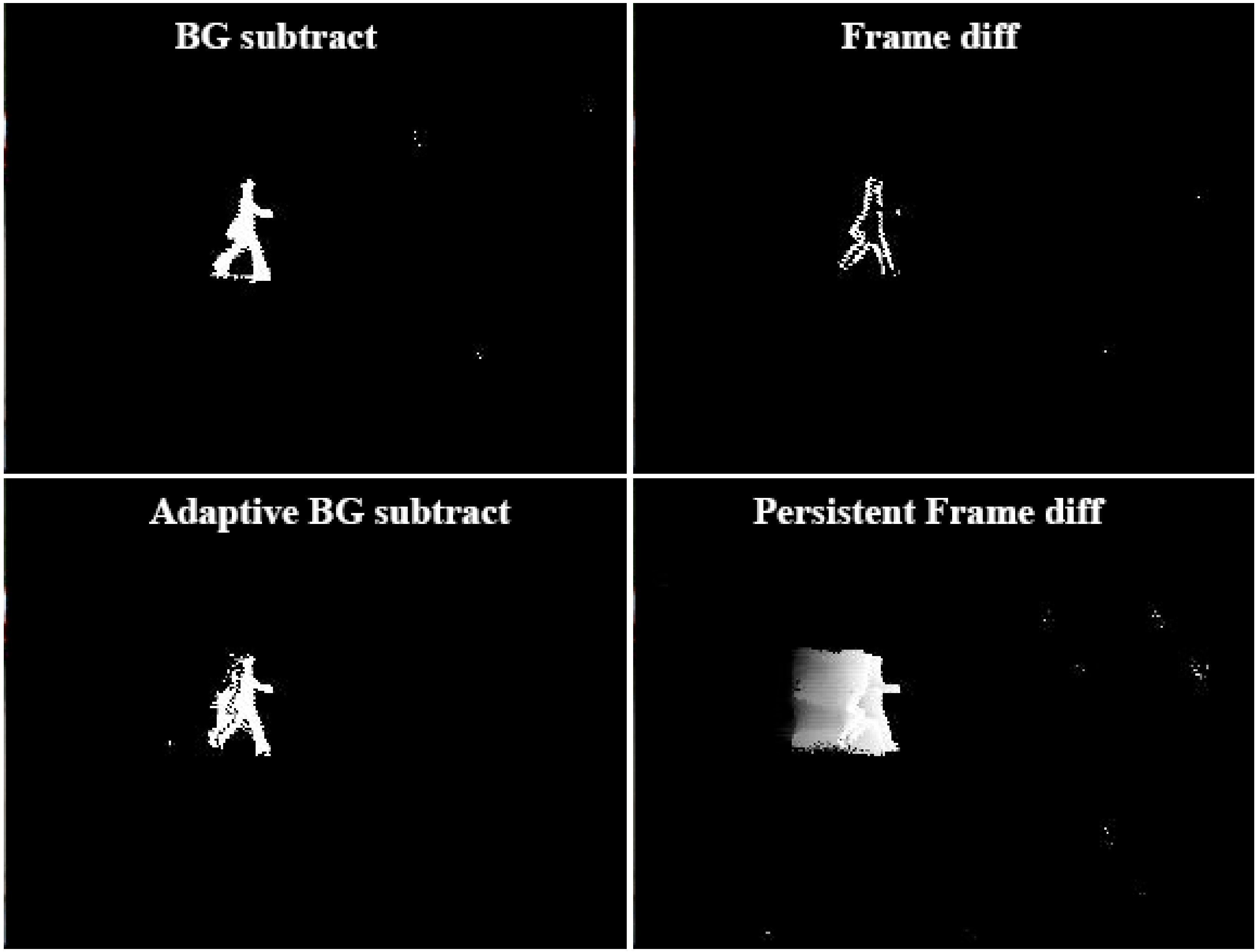
Persistant frame differencing is also responsive to changes in illumination and camera motion, and stopped objects / ghosts also fade away.

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.

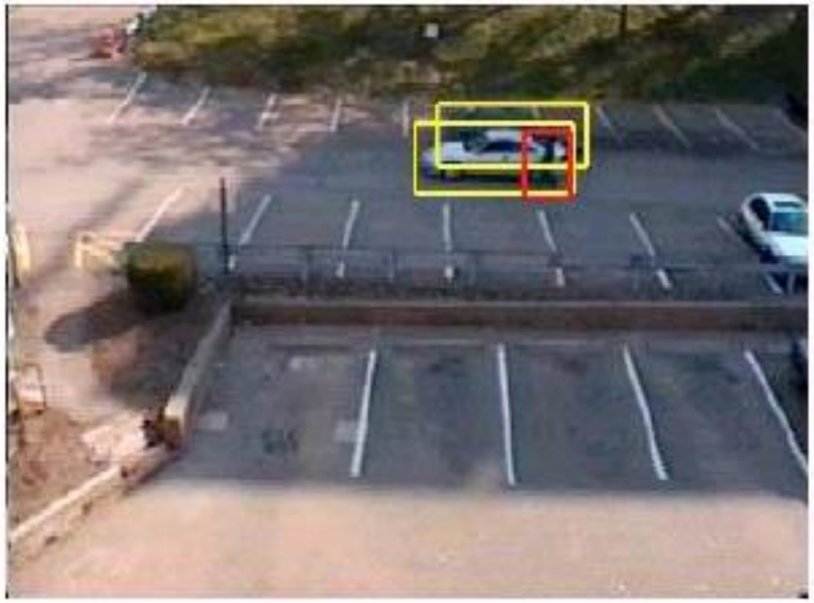


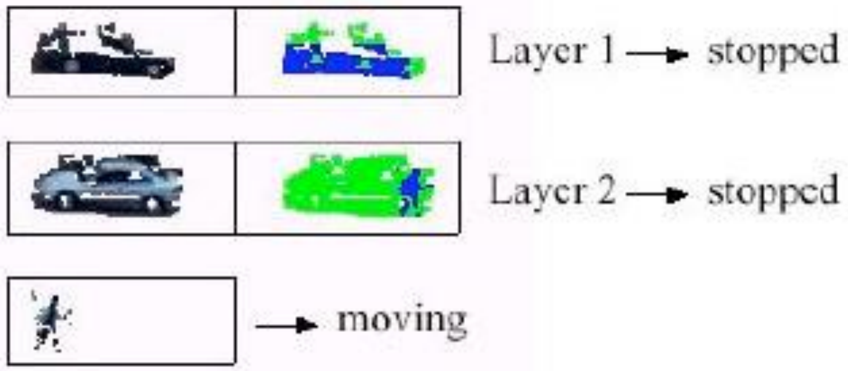



BG subtract
Frame diff
Adaptive BG subtract
Persistent Frame diff

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.



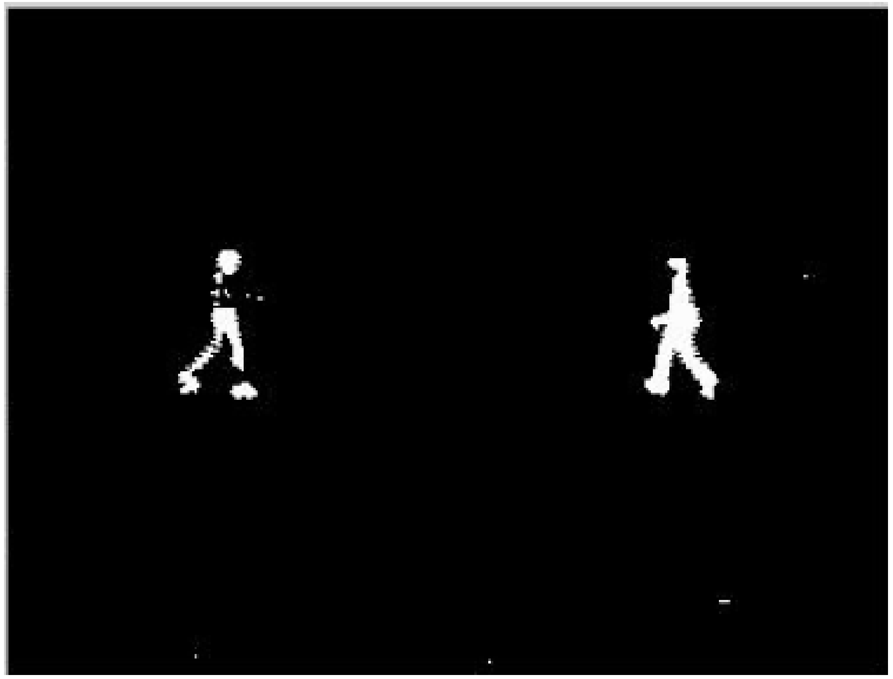


Allow blobs to be layered, so that stopped blobs can be considered part of the background for new object detection, but they will not leave behind ghosts when they start moving again.

Grouping Pixels into Blobs

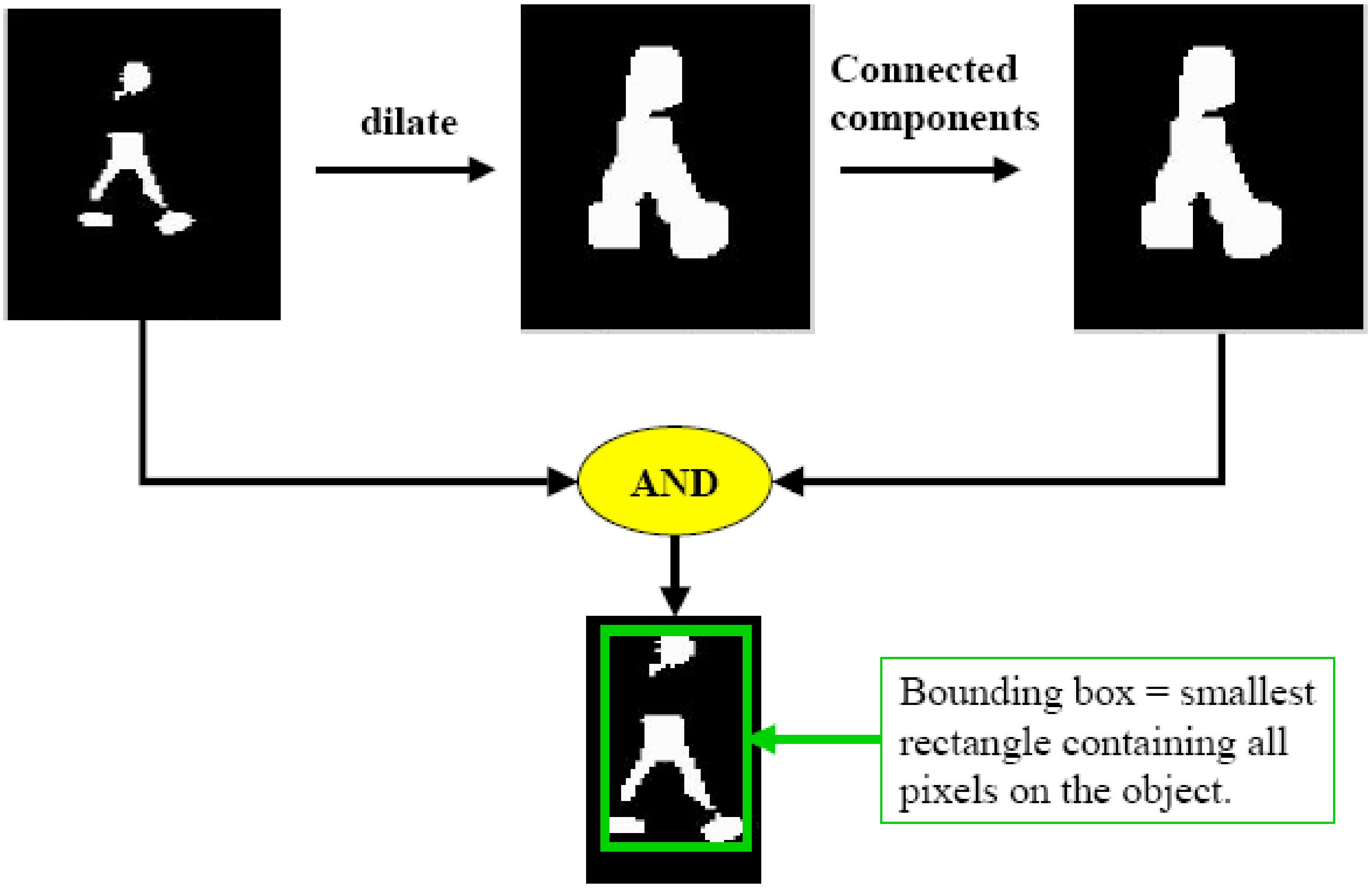
Motivation: change detection is a pixel-level process.

We want to raise our description to a higher level of abstraction.



- median filter to remove noisy pixels
- connected components (with gap spanning)
- Size filter to remove small regions

A Typical Processing



Bounding box = smallest rectangle containing all pixels on the object.

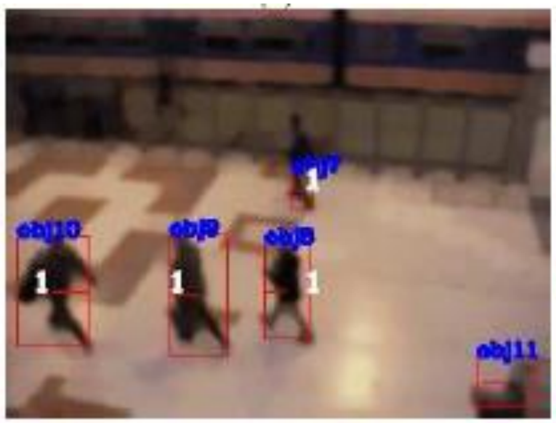
Sample for Filtering

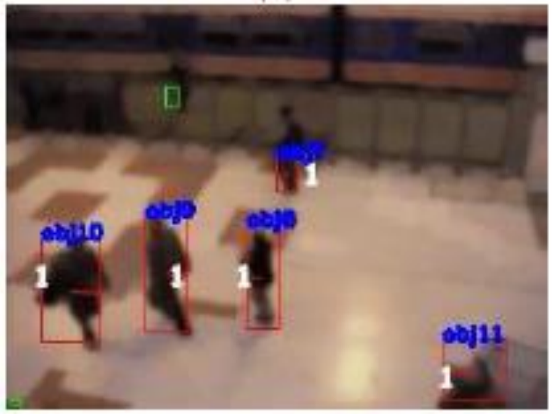
smoothing with 9 × 9 kernel

morphological operations

Minimum perimeter is 75,
and minimum area is 120

size filtering





distance filter

minimum rectangle
distance is 15 here.

temporal filter

temporal threshold duration of
0.25 seconds, 6 frames here.