Adaptive Background Mixture Models for Real-Time Tracking[†]

Eric Thul

March 30, 2007

[†]By Chris Stauffer and W.E.L Grimson [4, 1]

Outline

Problem Statement

Topic

The authors are working to develop an adaptive mixture of Gaussians that can model a changing scene.

Question

How can such a probabilistic model be used to distinguish between background and foreground pixels in the scene?

Rationale

To understand how effective their probabilistic model performs when applied to real-time tracking.

Significance

Improvement to the stability and reliability for video surveillance systems compared to other approaches.

Problem Statement

Other Approaches

The authors compare their technique to:

- Averaging images over time to threshold-out foreground pixels.
  - Not robust to scenes with many (slow) moving objects
  - Single threshold for the entire scene.
- Ridder et al. [3] improvement of modeling each pixel with a Kalman filter.
  - Not robust to backgrounds with repetitive change.
  - Takes significant time to re-establish the background.
- Wren et al. [5] improvement of modeling each pixel with a single Gaussian.
  - Good indoor performance.
  - Not tested for out-door scenes (for repetitive change).

Hypothesis

If we model each pixel as a mixture of Gaussians to determine whether or not a pixel is part of the background, then we will arrive at an effective approach to separate the background and foreground, which can be used for real-time tracking.

Idealized Example

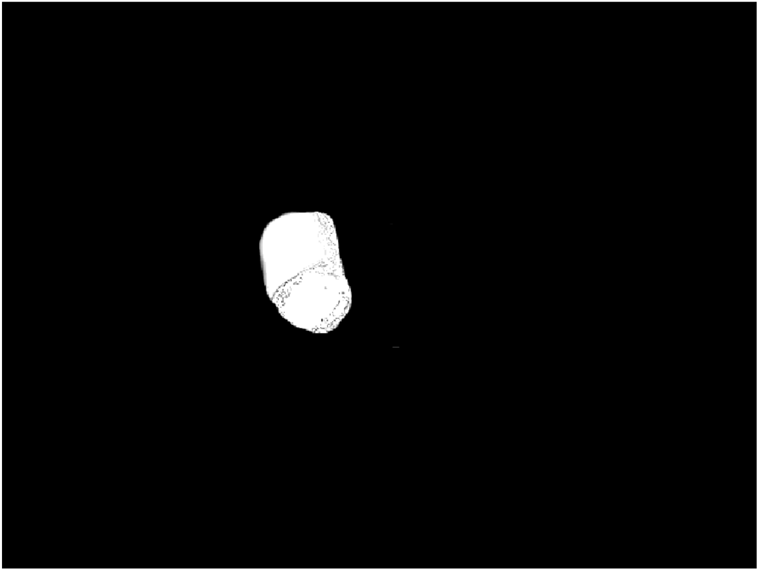
Original Scene

Idealized Example

Background of the Scene

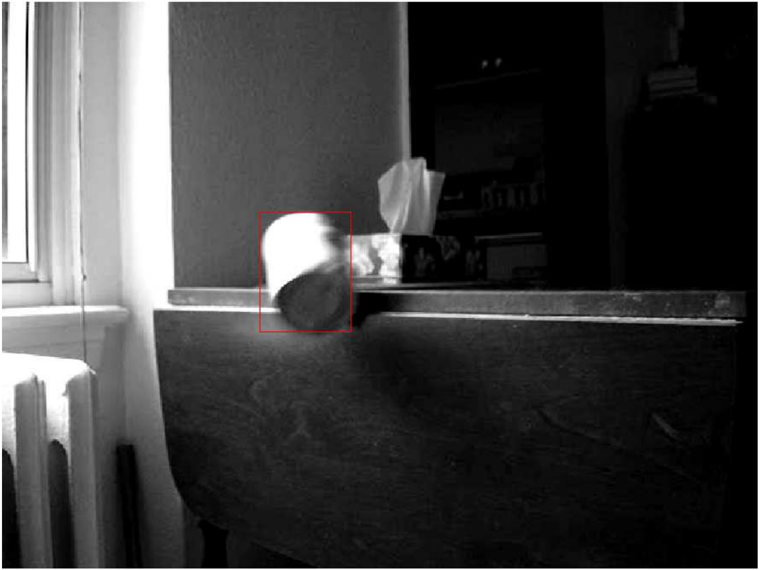
Idealized Example

Foreground of the Scene



Idealized Example

Original Scene with Tracking



Methodology

Overview

Two main parts:

1. Probabilistic model for separating the background and foreground.
   - Adaptive mixture of multi-modal Gaussians per pixel.
   - Method for updating the Gaussian parameters.
   - Heuristic for determining the background.
2. Technique for tracking objects in the foreground.
   - Algorithm to label connected components.
   - Method for tracking each connected component.

Definition

We define the pixel process of a pixel X as the history of it's value from frame 1 to frame t. This can be represented as follows.

$$\{X_1, \ldots, X_i, \ldots, X_t\} \text{ where } 1 \leq i \leq t$$

But what does this tell us?

Methodology

The Pixel Process Examples 1 and 2

Figure: Example 1 of gradual change over time.

Figure: Example 2 two of repetitive change.

Methodology

The Pixel Process Example 1

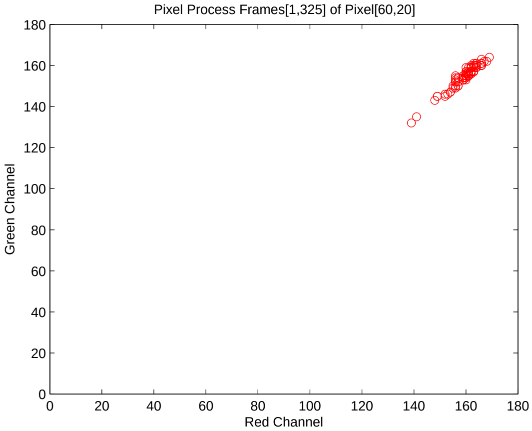


Figure: Plot of [red,green] values of a pixel for frames 1 to 325.

Methodology

The Pixel Process Example 1

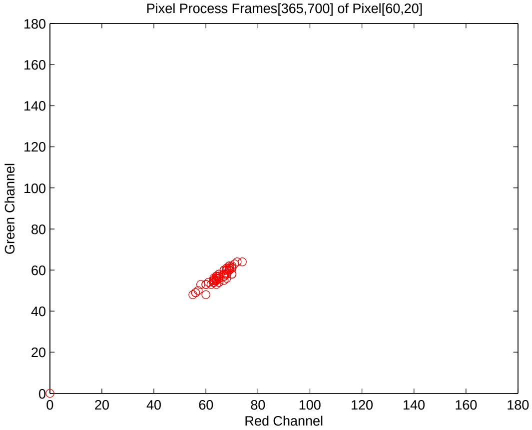


Figure: Plot of [red,green] values of the same pixel for frames 365 to 700.

Methodology

The Pixel Process Example 1

Tells us there is a need for an adaptive model.

Methodology

The Pixel Process Example 2

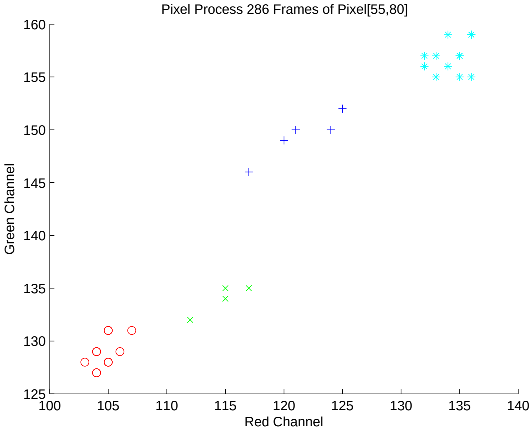


Figure: Plot of [red,green] values of a pixel during monitor flicker.

Methodology

The Pixel Process Example 2

Tells us there is a need for a multi-modal model.

Part 1: Probabilistic model for separating the background and foreground.

Methodology

Probabilistic Model: Mixture of Gaussians

Let $\{X_1, \ldots, X_t\}$ be a pixel process for X. The probability of observing a X at frame t is as follows, where K is the number of Gaussians in the mixture.

$$P(X_t) = \sum_{i=1}^{K} weight_{i,t} * GaussianPDF(X_t, Means_{i,t}, Covariance_{i,t})$$

Assume the *Covariance* matrix is $Variance * Identity$ matrix.

Methodology

Probabilistic Model: Update Parameters

On the next frame, we have a new pixel X_t where $t = t + 1$. We must update our probabilistic model as follows.

Look at each Gaussian in the mixture of pixel X_t.

- If $X_t \leq 2.5$ standard deviations from the mean then label matched and stop.
- Any Gaussian not matched is labeled unmatched.

Methodology

Probabilistic Model: Update Parameters

From the labeling of each Gaussian we have the following cases.

1. If $Gaussian_{i,t}$ is marked as matched.
   - Increase the weight.
   - Adjust the mean closer to X_t.
   - Decrease the variance.
2. If $Gaussian_{i,t}$ is marked as unmatched.
   - Decrease the weight.
3. If all the Gaussians in the mixture for pixel X_t are unmatched.
   - Mark X_t as a foreground pixel.
   - Find least probable Gaussian in the mixture and set:
     - $Mean = X_t$
     - *Variance* as a high value
     - *weight* as a low value

Methodology

Probabilistic Model: Update Parameters

Equations for the update:

- $\omega_{i,t} = (1-\alpha) * \omega_{i,t-1} + \alpha * M_{i,t}$
- $\mu_t = (1-\rho) * \mu_{t-1} + \rho * X_t$
- $\sigma_t^2 = (1-\rho) * \sigma_{t-1}^2 + \rho * (X_t - \mu_t)^T * (X_t - \mu_t)$
- $\rho = \alpha * \eta(X_t, \mu_{t-1}, \Sigma_{t-1})$

α is the learning parameter.

Methodology

Probabilistic Model: Background Heuristic

Since we already have our foreground pixels, we need to determine which Gaussians from the mixtures represent the background pixels. We want distributions which have a high weight and low variance.

Methodology

Probabilistic Model: Background Heuristic

To find such distributions the following heuristic is proposed.

1. Order the Gaussians of each mixture by *weight*/*standard devation*.
2. Sum the weights in this ordering until the sum is greater than a pre-set threshold T.
   - Larger T means more pixel values in the background.
3. Use the means from each Gaussian in the sum to represent the background.

Methodology

Tracking Technique

Part 2: Technique for tracking objects in the foreground.

Methodology
Tracking Technique: Connected Components

We now have the following:

- Model representing the background.
- Marked set of pixels representing the foreground.

The next step is to segment these foreground pixels into connected components. An algorithm for this is described by Horn [2].

Methodology

Tracking Technique: Horn's Algorithm

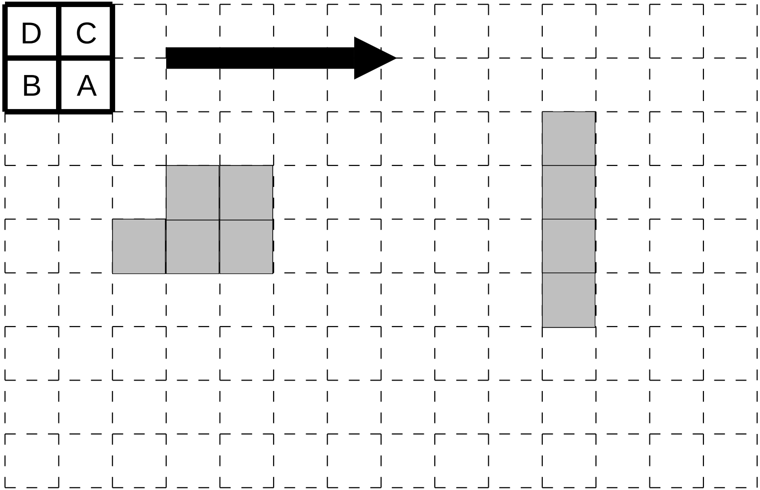


Figure: A is not marked so continue

Methodology

Tracking Technique: Horn's Algorithm Cases

Given A marked:

- Case 1: D is labeled so copy label to A
- Case 2: B or C are labeled exclusively so copy label to A
- Case 3: B and C are not labeled so pick new label for A
- Case 4: B and C are both labeled
  - Case 4a: B and C are labeled differently so copy label and note equality
  - Case 4b: B and C have the same label so copy label to A

Methodology

Tracking Technique: Horn's Algorithm

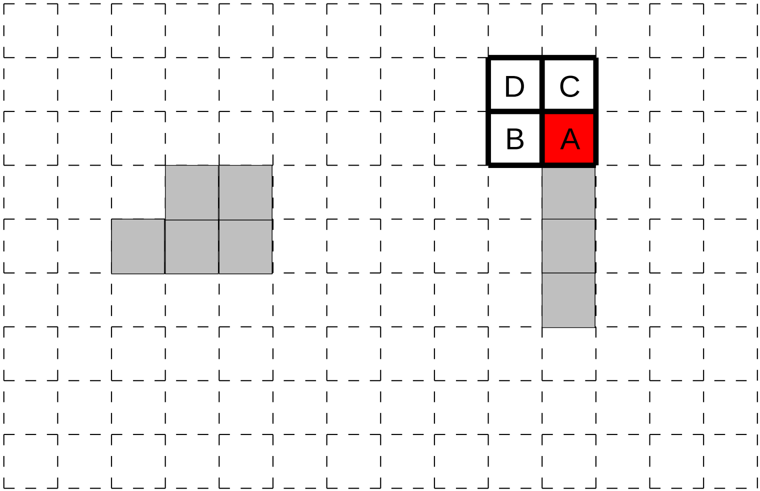


Figure: Case 3: B and C are not labeled so pick new label for A

Methodology

Tracking Technique: Horn's Algorithm

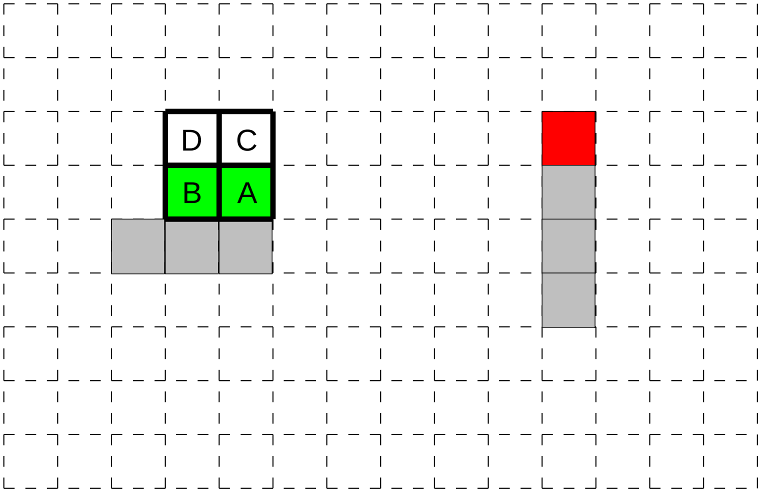


Figure: Case 2: B or C are labeled exclusively so copy label to A

Methodology

Tracking Technique: Horn's Algorithm

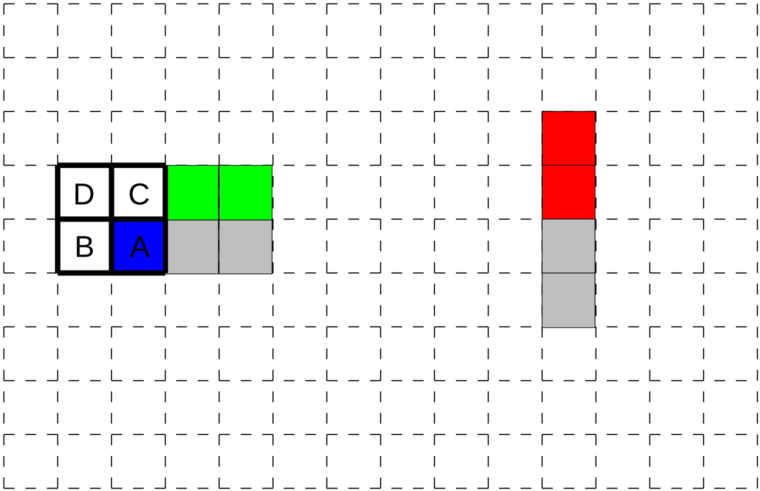


Figure: Case 3: B and C are not labeled so pick new label for A

Methodology

Tracking Technique: Horn's Algorithm

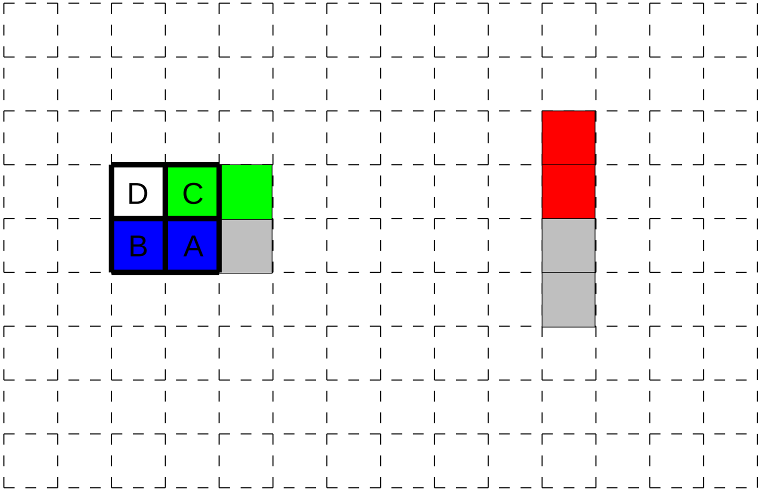


Figure: Case 4b: B and C are labeled differently so copy label and note equal

Methodology

Tracking Technique: Horn's Algorithm

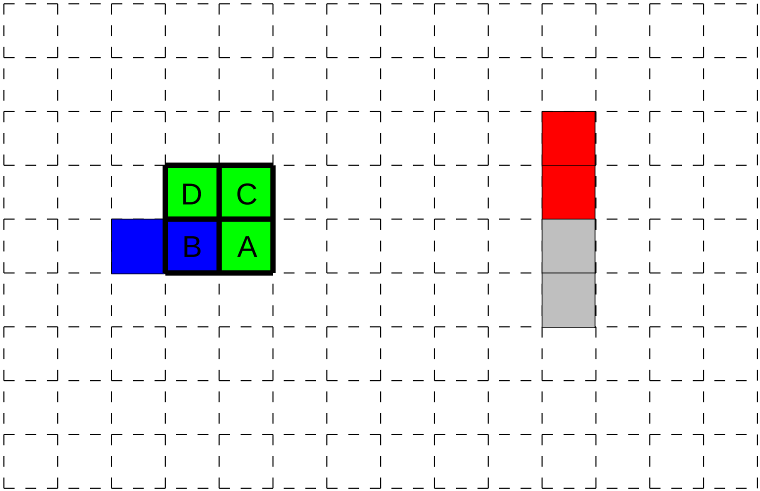


Figure: Case 1: D is labeled so copy label to A

Methodology

Tracking Technique: Horn's Algorithm

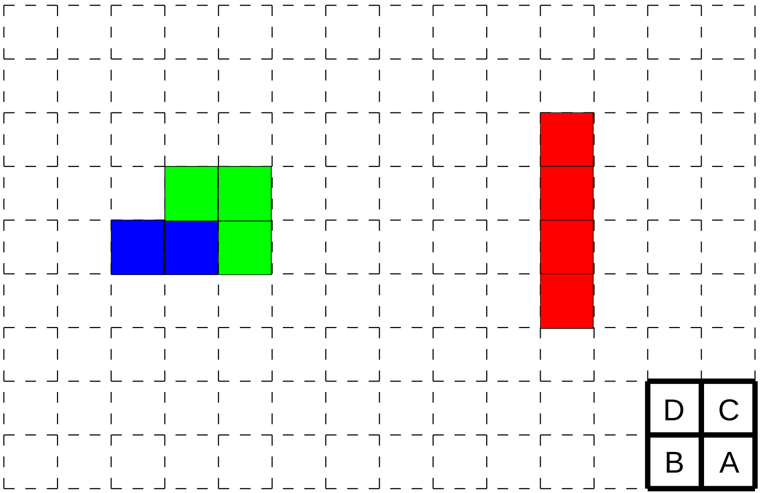


Figure: A is not labeled so continue

Methodology

Tracking Technique: Multiple Hypothesis Tracker

The result of Horn's algorithm is a set of connected components we can track across each frame. The essential tool for this is a set of Kalman filters.

Three main steps at each frame:

1. Find the Kalman filter which fits a connected component.
2. Create a new Kalman filter if no existing ones fit.
3. Remove any Kalman filters which are no longer relevant.

Experimentation

The method described has been applied to the following.

- Five outdoor scenes from stationary cameras.
  - Technology Square, Cambridge Massachusetts
  - Used for over 16 months
  - Sun, Overcast, Snow, Rain
  - Tracked people and cars
- Other experiments.
  - Birds at a feeder, mice at night, fish in a tank, people entering a lab.

Results

Overcast Day[†]

Figure: Original image during an overcast day.

Figure: Foreground detection of people in the scene.

[†] http://www.ai.mit.edu/projects/vsam/Tracking/index.html

Results

Snowy Day[†]

Figure: Original image during a snowy day.

Figure: Foreground detection of cars and snow in the scene.

[†] http://www.ai.mit.edu/projects/vsam/Tracking/index.html

Results

Sunny Day[†]

Figure: Original image during an sunny day.

Figure: Foreground detection of cars and people in the scene.

[†] http://www.ai.mit.edu/projects/vsam/Tracking/index.html

Results

Results Overall

- The Good.
  - Robust against rain, snow, sleet, hail, overcast.
  - Handles motion from swaying branches, rippling water, noise.
  - Works under day and night cycles.
- The Bad.
  - Difficulty in full sun (long shadows).
  - Very windy and partly cloudy days were also challenges.
  - Difficulty with objects overlapping.
  - Fast lighting changes were an issue.

Results

My Results

Let's go to the video tape.

Discussion

Drawbacks/Problems

Problems during implementation:

- As time increases, the variance decreases for stable pixels. If the variance gets too small, then noise from the camera is marked as foreground pixels.
- The mixtures may adapt too quickly to slow moving objects that are uniform in color. This will incorporate a foreground object into the background.
- The world may not be Gaussian.

Discussion

Future Work

Authors':

- More Gaussians in the mixture.
- Full covariance matrix.
- Adding prediction for each Gaussian may improve tracking during lighting changes.

Me:

- If more than one Gaussian in a mixture can be matched to the new pixel, pick the closest.
- Factor in the weight when finding the least probable distribution to replace.
- Fish! (If you have pet fish, I'm very interested)

Feedback

Comments / Questions / Suggestions

References

[1] W.E.L Grimson, C. Stauffer, R. Romano, and L. Lee. Using adaptive tracking to classify and monitor activities in a site. In *Computer Vision and Pattern Recognition*, pages 22–29, 1998.

[2] B.K.P. Horn. *Robot Vision*. The MIT Press McGraw-Hill Book Company, 1986.

[3] Christof Ridder, Olaf Munkelt, and Harald Kirchner. Adaptive background estimation and foreground detection using kalman-filtering. In *International Conference on recent Advances in Mechatronics*, pages 193–199, 1995.

[4] Chris Stauffer and W.E.L Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition*, volume 2, pages 252–258, 1999.

[5] Christopher R. Wren, Ali Azarbayejani, Trevor Darrell, and Alex Pentland. Pfinder: Real-time tracking of the human body. In *Pattern Analysis and Machine Intelligence*, volume 19, pages 780–785, 1997.