“Hybrid Cone-Cylinder” Codebook Model for Foreground Detection with Shadow and Highlight Suppression

Anup Doshi and Mohan Trivedi
University of California, San Diego

Presented by: Shaurya Agarwal

Motivation

Challenges in segmenting foreground objects from the background include:

- Illumination changes due to
  - course of 24 hours (direction and intensity of sunlight)
  - changing weather (clouds etc)
  - moving cast shadows during the daytime
  - highlights and shadows due to artificial light sources at night
- Moving backgrounds such as flags or trees

Related Studies and their shortcomings

- Single Gaussians: Fail to model complex backgrounds
- Mixture of Gaussians: Have trouble with sensitive detections and fast variations. MOGs require large components for a complex background increasing computational requirement
- Non-parametric kernel density estimation: In many cases this method can be memory intensive

Background

Proposed hybrid model in this paper is based on the following three studies:

1. "Improving Shadow Suppression in Moving Object Detection with HSV Color Information".

   Cucchiara, R., Grana, C., Piccardi, M., Prati, A., & Sirotti, S

   Key Takeaway: Use of HSV space insted of RGB

2. "Real-time foreground–background segmentation using codebook model".

   Kim, K., Chalidabhongse, T. H., Harwood, D., & Davis, L.

   Key Takeaway: Mainly based on this one. The codebook model and the cylinder shaped test volume.

3. "Bayesian background modeling for foreground detection".

   Porikli, F., & Tuzel, O.

   Key Takeaway: The concept of cone shaped test volume.

Shadow Detection in HSV Space [1]

- This algorithm is based on the concept that shadows change the brightness of the background, but do not really affect the color values.
- Hence, HSV space is chosen to distinguish luminance (V) from chrominance (H and S)
- A shadow classifier for a given pixel can be expressed as:

$$SP_k(x,y) = \begin{cases} 1 & if\, \alpha \leq \frac{I_k^V(x,y)}{B_k^V(x,y)} \leq \beta \\ & \wedge (I_k^S(x,y) - B_k^S(x,y)) \leq \tau_S \\ & \wedge \left| I_k^H(x,y) - B_k^H(x,y) \right| \leq \tau_H \\ 0 & otherwise \end{cases}$$

- Where I_k and B_k are the input and background images

Color Space Modification

- The original codebook model uses L2-norm of the RGB components as intensity values, and the chrominance is measured as a function of the angle between the input and reference values in RGB space.
- In HC3 (Hybrid Cone-Cylinder Codebook)model, HSV space is used, where intensity is approximated as the V (value) component, and chrominance encompasses the Hue and Saturation components.
- This simplifies the process and helps in combining the codebook model and shadow suppression techniques
- HSV space eases calculations in both cases, and also reduces parameter set.

Codebook model for foreground background segmentation [2]

- The codebook algorithm applies a clustering technique to model the background. It maintains a codebook for each pixel consisting of codewords obtained from the pixel parameters.
- Codeword is comprised of an RGB vector, and several auxiliary components used in the test data comparison.
- RGB vector $\mathbf{v_i}$ containing RGB values
- 6-tuple $\mathbf{aux_i}$ containing parameters like:

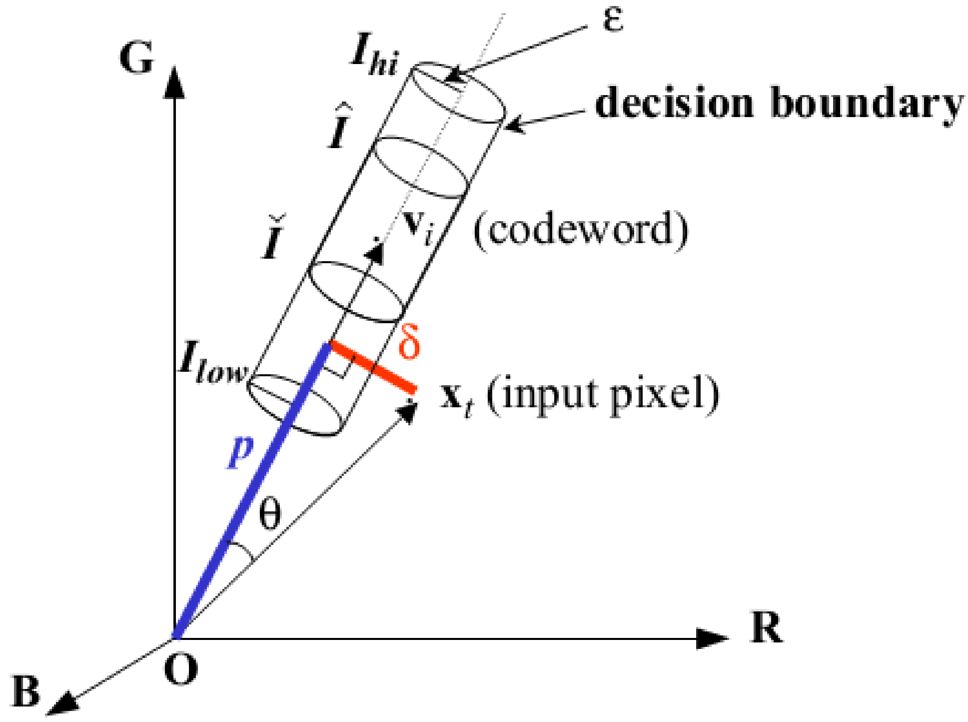
$\check{I}$ and $\hat{I}$	the min and max brightness, respectively, of all pixels assigned to this codeword
f	the frequency with which the codeword has occurred
λ	The maximum negative run-length(MNRL) defined as the longest interval during the training period that the codeword has NOT recurred
p and q	The first and last access times, respectively, that the codeword has occurred

Codebook model for foreground background segmentation

A test pixel is classified as a member of the codeword's set if it satisfies 2 conditions:

- Brightness Constraint: the intensity (norm[RGB]) should be within some range of the lowest intensity and the highest intensity pixel in the codeword's set
- Color Distortion: the color or chromaticity (function of the angle between test vector and codeword RGB) should be within some range.

Codebook model for foreground background segmentation



Algorithm for Codebook Construction

Algorithm for Codebook construction

I. $L \leftarrow 0^1$, $\mathscr{C} \leftarrow \emptyset$ (empty set)

II. **for** $t = 1$ to N **do**

(i) $\mathbf{x}_t = (R, G, B)$, $I \leftarrow \sqrt{R^2 + G^2 + B^2}$

(ii) Find the codeword $\mathbf{c}_m$ in $\mathscr{C} = \{\mathbf{c}_i | 1 \leqslant i \leqslant L\}$ matching to $\mathbf{x}_t$ based on two conditions (a) and (b).

(a) $colordist(\mathbf{x}_t, \mathbf{v}_m) \leqslant \varepsilon_1$

(b) $brightness(I, \langle \check{I}_m, \hat{I}_m \rangle) = \mathbf{true}$

(iii) If $\mathscr{C} = \emptyset$ or there is no match, then $L \leftarrow L + 1$. Create a new codeword $\mathbf{c}_L$ by setting

- $\mathbf{v}_L \leftarrow (R, G, B)$
- $\mathbf{aux}_L \leftarrow \langle I, I, 1, t-1, t, t \rangle$.

(iv) Otherwise, update the matched codeword $\mathbf{c}_m$, consisting of

$\mathbf{v}_m = (\bar{R}_m, \bar{G}_m, \bar{B}_m)$ and $\mathbf{aux}_m = \langle \check{I}_m, \hat{I}_m, f_m, \lambda_m, p_m, q_m \rangle$, by setting

- $\mathbf{v}_m \leftarrow \left(\frac{f_m \bar{R}_m + R}{f_m + 1}, \frac{f_m \bar{G}_m + G}{f_m + 1}, \frac{f_m \bar{B}_m + B}{f_m + 1} \right)$
- $\mathbf{aux}_m \leftarrow \langle \min\{I, \check{I}_m\}, \max\{I, \hat{I}_m\}, f_m + 1, \max\{\lambda_m, t - q_m\}, p_m, t \rangle$.

end for

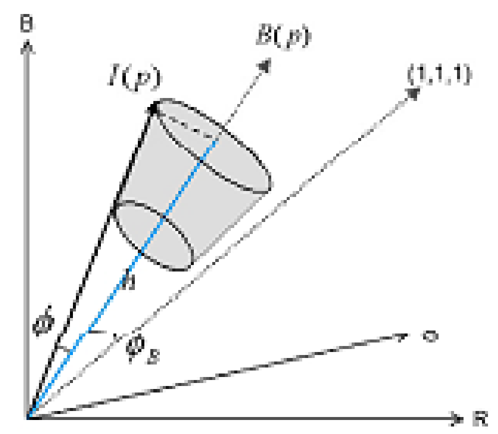
III. For each codeword $\mathbf{c}_i$, $i = 1, \ldots, L$, wrap around λ_i by setting $\lambda_i \leftarrow \max\{\lambda_i, (N - q_i + p_i - 1)\}$.

Shortcomings of the cylinder shaped test volume

- Although it is simple and effective, it has some shortcomings
- It is apparent that at lower intensities, the cylinder can contain larger set of chromaticity values.
- With a background pixel of low intensity, almost every low-intensity test pixel will be assigned to this cylinder.
- Hence as two similarly grouped pixels increase intensity (highlights), they have less chance of being in the same cluster

Shadow and Highlight Supression using Conical shaped Test Volume [3]

- A shadow or highlight is simply the same chromaticity value, with a lower or higher luminance.
- Hence the conical volume makes more sense than the cylindrical one to model highlights and shadows.
- Cone corrects the problems of cylinder based volume and more precisely covers the color-space.



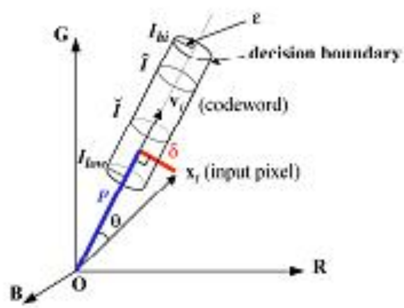
Shortcomings of the conical shaped test volume

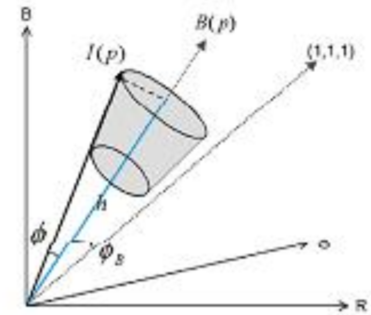
- Shadow and highlight ‘cones’ lie beyond the range of codebook cone, adjacent on either side.
- But the pure conical highlight detector model grabs too many pixel values in its space, pushing up the false negative rate.

Hybrid Cone-Cylinder Volume

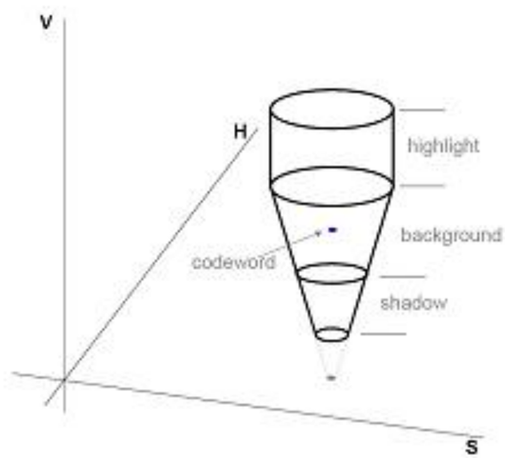
- Based on the pros and cons of both cylinder and cone shaped test volume, a hybrid Cone-Cylinder volume is used in HC3.
- In HC3, while cone volume is used to model background and shadow, cylinder volume is used for highlights.
- This provides the desired sensitivity in the detections.

Hybrid Cone-Cylinder Volume





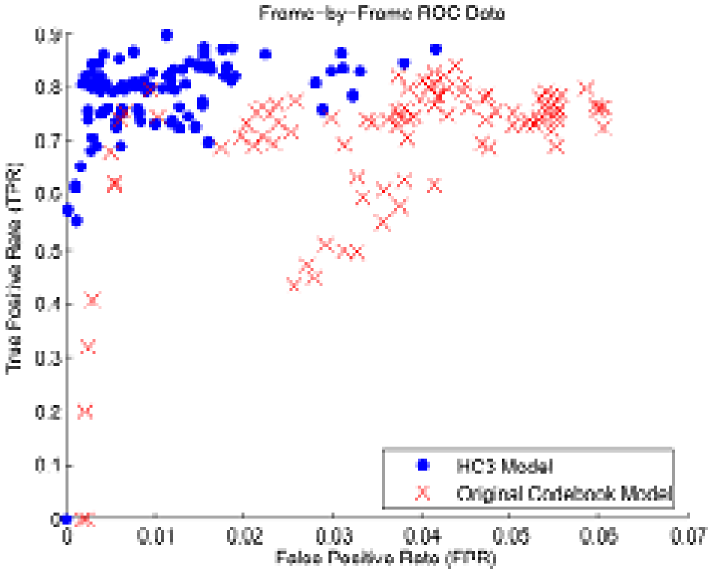
(a) Codebook 'codeword' model[6] (b) Shadow suppression model[10]



(c) Hybrid Cone-Cylinder model

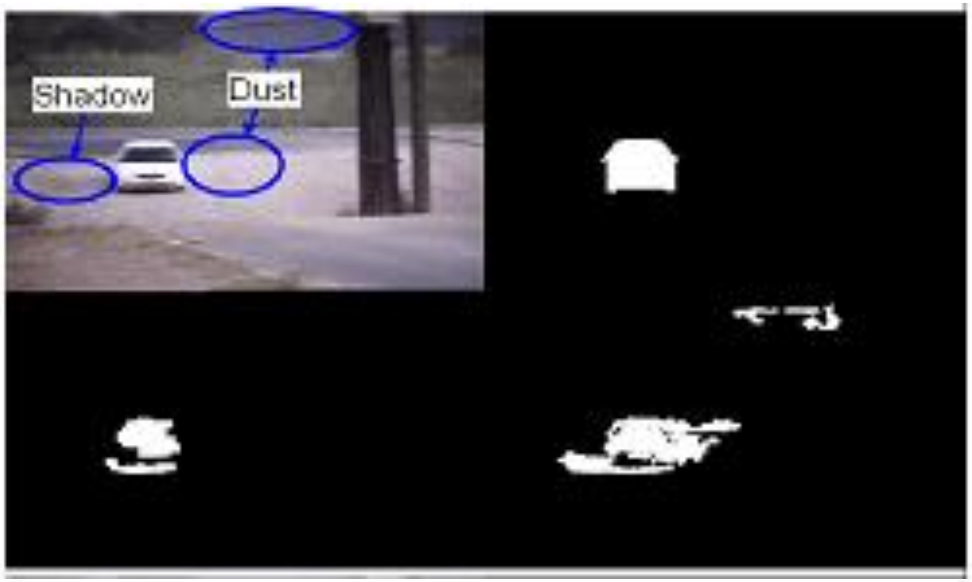
Results

(a) Input frame (top left), ground truth (top right), HC3 model (bottom left), original model (bottom right)

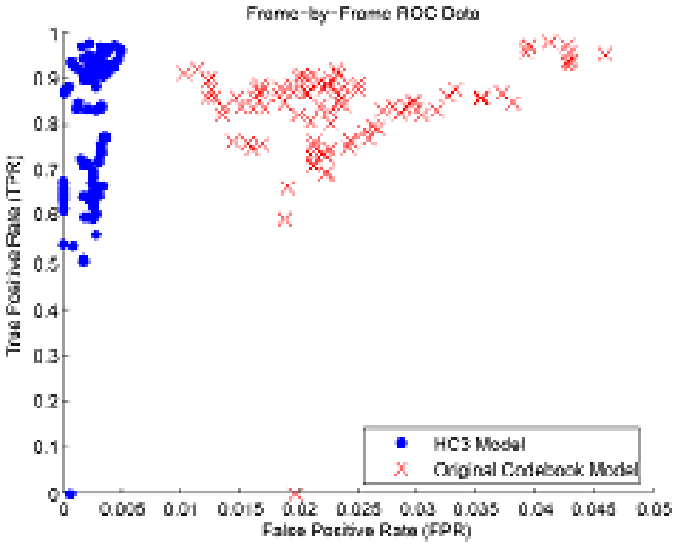


(b) Errors for each of 100 frames in the sequence

Results



(a) Input frame (top left), ground truth (top right), HC3 model (bottom left), original model (bottom right)



(b) Errors for each of 100 frames in the sequence

Results

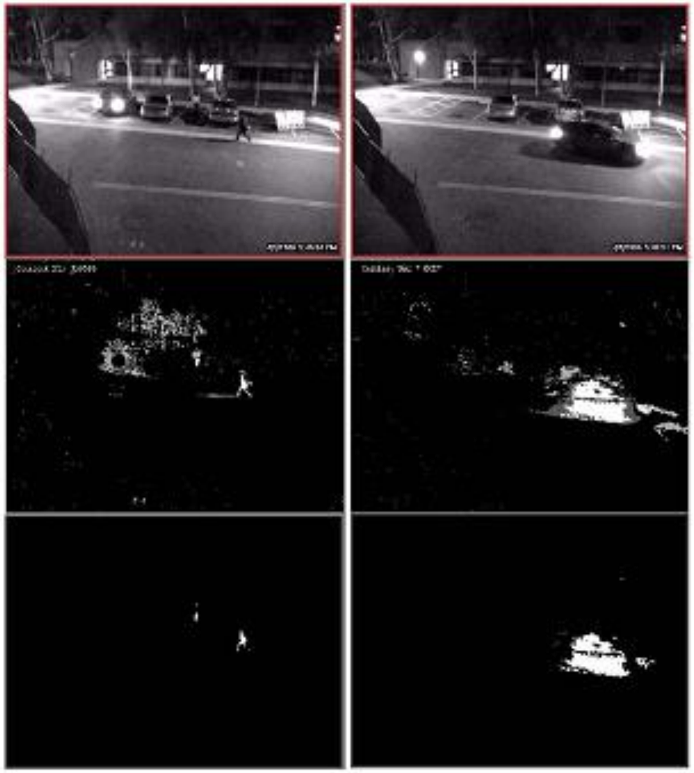


Figure 5. Background Removal at Night. Input (top), Segmentation: shadows as dark gray and highlights as light gray (middle), Final foreground (bottom).

Performance

- HC3 algorithm runs at approximately 40 frames per second on videos of size 320x240.
- Hence it is suitable for real-time applications.
- Due to trimming and adaptive updating, the sizes of the codebook and cache stay almost constant.
- For most pixels, 1-2 codewords are sufficient on average.
- ROC curves show the effectiveness of this algorithm and it out performs the original method.
- Detection rate of true positives are quite similar, but the false positive rate drops significantly using HC3.
- This is mostly due to the classification of shadows and highlights as background by HC3 algorithm.

Conclusion

- The HC3 algorithm presented in this paper is clearly effective and robust against many different scenes.
- One drawback of the current setup is the required parameter tuning.
- There are a considerable number of variables to adjust to find appropriate values for a certain environment.
- One solution is to allow the algorithm to run for long amount of time, so that it automatically adjusts the parameters according to the environmental changes.
- It may also be possible to use machine learning and optimization algorithms to find the optimal parameters.

Questions?