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

Anup Doshi and Mohan Trivedi
University of California, San Diego - CVRR Laboratory
9500 Gilman Drive, San Diego, CA 92093
{andoshi, mtrivedi}@ucsd.edu

Abstract

In the interest of 24-7 long-term surveillance, a truly robust, adaptive, and fast background-foreground segmentation technique is required. This paper deals with the especially difficult but extremely common problems of moving backgrounds, shadows, highlights, and illumination changes. To produce reliable foreground extraction in the face of these problems, the best practical aspects of two algorithms, Codebook Segmentation[6] and HSV Shadow Suppression[2] are combined. The main contribution of this paper is the introduction of the “Hybrid Cone-Cylinder” Codebook (HC3) model. Results show superior speed and quantitatively better performance in many different conditions and environments. Applications include people-tracking with Omni-directional cameras and vehicle-counting with rectilinear cameras.

1. Introduction and Motivation

Outdoor surveillance and security settings are likely one of the most common uses of video cameras. Yet there are still many problems to be solved to make a stable and reliable computer vision system useful for those applications. The primary problem for any system is segmenting foreground objects from the background, in the face of various environmental challenges.

These challenges include global illumination changes, as can be expected over the course of 24 hours or changing weather; moving cast shadows during the daytime; specularities, highlights, and shadows due to artificial light sources at night; and moving backgrounds such as flags or trees. Global illumination changes can occur quickly, as in a cloud suddenly blocking out the sunlight, or relatively slowly, as the sun takes a while to set. Either way, the perceived background will change, so a model which removes that should adaptively update. Moving cast shadows are those shadows which move along with a foreground object, but are not a part of the object itself[2]. These shadows can cause objects to merge, distort their shape, or occlude other objects, so they can be very problematic.

At nighttime, several more interesting problems occur. Due to the lack of ambient light, every small light source within range of a scene can cause objects to light up. As those light sources vary, the objects may get brighter or darker; these illumination fluctuations are referred to as highlights and shadows. In particular it would be useful to discard the effects of these highlights and shadows, while still detecting foreground objects as they move through the image. As an example, consider a car moving through a parking lot in the evening. When the car turns, various different parts of the image will get brighter; others will get darker. These fluctuations do not represent actual changes in the background, and thus those points should remain classified as background.

Finally, an environmental challenge which is quite common to many outdoor scenes is of moving backgrounds. A common example is of trees or grass swaying in the wind. These movements may occur frequently and with an unknown periodicity (depending on the wind), so an algorithm to adaptively learn such motions would be useful.

The ability to perform object identification and recognition depends on a reliable and accurate object detection scheme. A search for interest points or template matches can easily be narrowed down to very few locations once the foreground has been extracted. Additionally, for surveillance purposes, reliable detection of objects in the presence of such environmental noise would help in tracking and subsequent analysis of tracks.

Motivated by the desire for continuous day-and-night operations, an algorithm that works well in many different illumination conditions is essential. The applications of such an algorithm could range from people-counting to abnormal vehicle track identification, as well as countless other surveillance-related tasks. With such a system online all day, every day, powerful and useful data can be automati-
2. Related Studies in Foreground and Shadow Segmentation

Foreground-background segmentation is a relatively extensive field. In search of algorithms that would perform well in light of all the problems posed above, several main contenders appeared. Typically common models for backgrounds considered are based on single Gaussians [5, 8, 4] or a Mixture of Gaussians (MOG) [12, 7, 1]. Where single Gaussians fail to model complex backgrounds, MOGs have trouble with sensitive detections and fast variations [6, 8]. Also, while MOGs may be able to converge to a complex background with enough components, the computation required may preclude real-time operations[10].

Another method for modeling backgrounds involves non-parametric kernel density estimation [3]. This method can in many cases be prohibitively memory intensive [6, 10]. The codebook method[6], described in detail below, uses a non-statistical clustering technique to achieve a fast and efficient model for the background while allowing for moving background. There are many other techniques (Wallflower, Pfinder, W4, etc.[9]) based on texture, color spaces, and pixel-, region-, or frame-based approaches.

Shadow suppression has also been quite extensively studied. A more common implementation assumes that shadows decrease the luminance of an image, while the chrominance stays relatively unchanged [2, 10, 5, 4]. However there are many more methods for removing shadows, many of which are reviewed in [11]. Accordingly, the algorithm based on the HSV space[2] was cited in [11] as the most generally capable and robust shadow detection and suppression algorithm. This algorithm is also described in further detail below.

2.1. Codebook Model for Background - Foreground Segmentation [6]

The codebook model for background segmentation is quite simply a non-statistical clustering scheme with several important additional elements to make it robust against moving background. The motivation for having such a model is that it will be fast, because it is deterministic; efficient (requires little memory); adaptive; and able to handle complex backgrounds with sensitivity. A detailed description of the model and can be found in [6].

In the codebook model, each pixel can have a variable number of codewords representing the background. The codeword is comprised of an RGB vector, and several auxiliary components which are used in the test data comparison. A test pixel is classified as a member of the codeword’s set if it satisfies 2 conditions: 1) Brightness Constraint: the intensity (norm[RGB]) should be within some range of the lowest intensity pixel in that codeword’s set, and the highest intensity pixel in the codeword’s set. 2) Color Distance: the color or chromaticity (function of the angle between test vector and codeword rgb) should be within some constant. If both conditions are satisfied, the test pixel is added to the codeword set.

Several other auxiliary components are kept as a part of the codeword, and assist in training and pruning the codebook. Namely, these values are the first and last accesses of that codeword, along with the frequency of accesses and maximum length of time between consecutive accesses (MNRL). These values are used to determine whether a codeword should remain in the background or not. Generally, a moving background such as a tree will have a smaller MNRL the two components of the background (tree and non-tree) will get accessed relatively frequently.

Adaptive updating occurs whenever a test pixel falls into the cluster of a codeword. Additionally, when a pixel is classified as foreground, it is added as a codeword to a cache. In turn, if a pixel is repeatedly accessed from the cache, it moves into a layer of the background. These two together provide a basis for robustness under global illumination changes; however, there are additional measures in [6] which can be taken to gain more adaptability.

2.2. Shadow Detection in HSV space[2]

The algorithm presented in [2] is a deterministic non-model based solution for finding moving cast shadows. It is based on the simple idea that shadows change the brightness of the background, but do not really affect the color values. For this reason, the HSV space is chosen to distinguish luminance (V) from chrominance (H and S). Using the logic presented above, a shadow classifier for a given pixel can simply boil down to the following expression:

\[
SP_k(x, y) = \begin{cases} 
1 & \text{if } \alpha \leq \frac{I_k^V(x, y)}{B_k^V(x, y)} \leq \beta \\
& \land (I_k^B(x, y) - B_k^B(x, y)) \leq \tau_S \\
& \land |I_k^H(x, y) - B_k^H(x, y)| \leq \tau_H \\
0 & \text{otherwise}
\end{cases}
\]

(1)

where \(I_k\) and \(B_k\) are the input and background images, respectively. \(\alpha, \beta, \tau_S, \tau_H\) are all parameters to be chosen.
Parameter selection is one critical issue here and will be discussed further in Section 5. [2] does however provide an analysis of the parameter choices and implications. It is also fairly obvious that this algorithm can extend to highlight detection. A highlight is simply an increased luminance as opposed to a decreased luminance, so the equations above can be appropriately modified to find highlights as well.

3. Towards Generalization of the Codebook Model

The two algorithms above were chosen for their various properties to implement a robust foreground detection algorithm. However two pressing issues arose in the combination stage, which suggested a modified approach to these algorithms. These are presented below. First is a discussion about the choice to use the HSV color space for the entire algorithm, which greatly simplifies the number of parameters and lessens the time of calculation. Then the code-word cluster volumes and shadow and highlight volumes are considered and re-engineered to provide a more cohesive framework.

3.1. Color Space Modification

The original codebook model[6] uses a the following method, as mentioned above, to distinguish intensity and chrominance. An intensity value is simply the L2-norm of the RGB components, and the chrominance is measured as a function of the angle between the input and reference values in RGB space. These calculations are only one way of estimating intensity, since the perceived brightness of each of the primary colors is actually quite different - green (0,255,0) actually appears much brighter than blue (0,0,255).

In HSV space, which is used in [2], intensity can be approximated as the V (value) component, where the chrominance encompasses the Hue and Saturation components. It is assumed that this V will be roughly equivalent to the total power of the spectrum; hence it is a good candidate to use for intensity. Upon closer inspection, though, it is apparent that the V component is a function only of the largest of the RGB components, and so it does not encode the entire spectrum.

However the main advantage of the HSV space is that the chrominance calculations can be done separately from the luminance calculations. Also, according to [3], the HSV space more closely models human vision, where the V can be understood as roughly close to how humans perceive luminance. This implies that it is a more representational approximation of the true illumination intensity than the L2-norm. Finally according to [2], RGB space has been demonstrated to be less accurate than HSV space for detecting shadows.

Therefore to combine the codebook model and shadow suppression techniques, and in the interest of simplicity, both methods are converted into HSV space. This representation eases calculations in both cases, and allows for a smaller parameter set as well. The transformation into HSV space and corresponding distance calculations are shown to be effective and extremely quick.

3.2. Introducing the Hybrid Cone-Cylinder Volume

The main modification to the models involves the shape of the test volume around a background pixel, whose space defines the “cluster” associated with that pixel. In other words, the test pixels which fall inside this volume are associated with the corresponding background pixel, and labeled as background.

The advantages of having such a fixed volume are discussed in [6], but in essence boil down to simplicity and effectiveness. To develop a more dynamic model would involve laborious statistical calculations and potentially show little, if any, improvements.

However the choice of the volume in the original codebook model[6] does not account for certain cases. As shown in Figure 1(a), a cylinder represents the cluster space around the label. Unfortunately it is apparent that at lower intensities, more chromaticity values would have the chance of lying within the background pixel’s cylinder. This is evident when examining a background pixel of very small intensity: almost every low-intensity test pixel will be assigned to its cylinder. An unintended consequence of this is that as two similarly grouped pixels increase intensity, they have less chance of being in the same cluster, even though their respective chrominance remains the same.

As a more effective and intuitive choice of volumes, a cone corrects these problems and more precisely covers the color-space. Therefore the codebook model here is modified to use cones instead of cylinders. Tests are underway to verify the hypothesized advantages of cones over cylinders in the codebook model. However empirically, it works extremely well.

The cone has already been used in shadow suppression techniques([10]) - see Figure 1(b) . As mentioned in the shadow detection algorithm description, a shadow or highlight is simply the same chromaticity value as the original background pixel, with a lower or higher luminance. Thus the shadow and highlight “cones” would lie beyond the range of the codebook cone, adjacent on either side. However after testing, the pure conical highlight detector grabbed too many pixel values within its space and thus
pushed up the false negative rate. With a desire for sensitive detections in mind, the highlight volume was limited to a cylinder. Therefore the final Hybrid Cone-Cylinder volume used in the HC3 algorithm is shown in Figure 1(c).

4. Experimental Validation of HC3 Model

4.1. Experiments

Several experiments were performed using the methodology above, to determine its potential effectiveness on 24-7 data in various environments. First, two scenes were manually marked up as ground truth data, and a those were used to quantitatively compare the HC3 model and original codebook model. Additionally, preliminary results are presented below on several other difficult environmental contexts.

The first two experiments were conducted on two sequences, each 100 frames long, which were selected for their complexity and appearance of cast shadows. These frames were manually marked up to represent ground truth. For each experiment, both the original codebook model[6] and the HC3 model were trained on a series of 500 frames and tested on these 100 frames. For each frame, the errors were detected and a corresponding Detection versus False Alarm rate was plotted on an ROC graph. It is worth noting that in these sequences some morphology was used in both models to remove noise and fill in holes, slightly reducing the sensitivity.

Figure 2 shows results from a simple scene where two people are walking along the road with very long cast shadows. As is evident, the HC3 model does well in classifying shadows as background. Figure 3 displays a more complex scene, with moving background in the waving grass and dust storms, as well as cast shadows and dust kicked up by the car. The HC3 model correctly classifies most of the shadows as background, and the dust is classified as highlights and thereby background as well.

Evidently the ROC plots show the effectiveness of this algorithm. The errors in the HC3 model are generally well separated from the original model; this is mostly due to the classification of shadows and highlights as background. Table 1 compares the average error rates over all 100 frames for each sequence. While the detection rate of true positives remains quite similar in both sequences, the false positive rate drops relatively significantly on average using HC3.

In Figure 4 is an example of a sensitive people-detector using an omnidirectional camera. Even though the background is somewhat stable, the foreground objects are quite small and could easily be confused with noise. However in
each case the people are detected with a good deal of accuracy.

A night scene is played out in Figure 5. This is an intriguing data set for all its problems to overcome. There is in fact very little color data, so the chrominance effect is diminished. However the algorithm still performs considerably well. In the first image, the highlights due to the car’s headlamps are suppressed, as well as the person’s shadow. After all the processing, both people are detected in the foreground. Note that the car had merged into a layer of the background after sitting for a long time, so it was not detected. In the second image, the illuminations caused by the car’s lights are detected and removed, leaving a clear location for the car itself.

In terms of performance, the HC3 algorithm runs at approximately 40 frames per second on videos of size 320x240. Thus it is quite clearly viable for real-time applications. Due to pruning and adaptive updating, the sizes of the codebook and cache stay relatively constant. In fact for most pixels 1-2 codewords are sufficient on average. A select few pixels need more codewords to deal with fluctuating background scenes.

4.2. Discussion and Future Work

The algorithm presented in this paper is clearly effective and robust against many different scenes. However one drawback of the current setup is the parameter tuning. There are a considerable number of variables to adjust to find appropriate values for a certain environment. [2] has a numerical evaluation of several parameter choices, and [6] proposes sample values. One solution to allow the algorithm to run for long periods of time, would be to automatically adjust the parameters according to the environmental changes (using cues such as time-of-day, weather patterns, illumination levels, etc.). However it may also be possible to use machine learning and optimization algorithms to find the optimal parameters.

As a next step, it will also be necessary to comprehensively compare the adjusted HC3 algorithm with the original codebook model, as well as with other background subtraction methods. Such analysis will provide motivation for re-tuning and adjusting the current model to apply in more general situations.

Finally, even though HSV space empirically works quite well, other color spaces may also be appropriate. These
Figure 5. Background Removal at Night. Input (top), Segmentation: shadows as dark gray and highlights as light gray (middle), Final foreground (bottom).

include spaces such as CIE L*a*b*, which encode the estimated luminance in one component and chrominance in two other independent components. It may be the case that such a space is slightly more precise; however, careful analysis will be necessary and parameters will need to be re-tuned. At the same time, the volume of space that signifies the cluster can be adjusted as well. A thorough investigation into the effects of these changes should be performed.

5. Conclusions

A demonstrably robust algorithm to perform foreground extraction has been introduced in this paper. The Hybrid Cone-Cylinder Codebook algorithm is designed to be real-time and robust against shadows, highlights, and moving backgrounds. Additionally, it is an adaptive model and thus capable of handling global illumination changes. These properties make the algorithm an excellent candidate to run on very long video sequences, in applications such as long-term surveillance and vehicle counting.

Illustrative results are given to show the useful features of the algorithm. It clearly performs well on nighttime data, with complex backgrounds, and does so quite sensitively. Quantitatively it is also shown to remove shadows and highlights relatively well on several data sets. Thus the HC3 model combining codebook and shadow/highlight suppression in HSV space is truly a robust foreground extraction method.

Acknowledgements

The authors are grateful for the support of TSWG, and the constructive comments provided by Dr. Tarak Gandhi and AVSS reviewers. They are most appreciative of the contributions and assistance of their colleagues from the CVRR laboratory, who made the experimental phase of the research possible.

References