# Training-free Moving Object Detection System based on Hierarchical Color-guided Motion Segmentation

Xinfeng Bao Gijs Dubbelman Svitlana Zinger Peter H. N. de With
SPS-VCA, TU/e, 5600 MB Eindhoven, the Netherlands
{xbao,g.dubbelman,s.zinger,p.h.n.de.with}@tue.nl

## Abstract

*We present a moving object detection system for surveillance based on* ***Hi****erarchical* ***Co****lor-guided* ***Mo****tion segmentation (HiCoMo). The HiCoMo system does not require training and consists of two main stages: (1) hierarchical color-guided motion segmentation, and (2) motion-based verification. The first stage is a hierarchical segmentation framework, where at each level a balance is made between static and temporal features. So that groups of pixels develop into semantic object segments. In the second stage, these object segments are further analyzed in terms of motion saliency and consistency, in order to finalize the object detection results. Our proposed system is tested on real-life surveillance videos containing various scenarios. The detection results outperform a state-of-the-art training-free moving object detection algorithm in recall (90.2% compared to 81.6%) while having a competitively promising precision (96.5% compared to 97.4%). The system has a generic nature and real-time implementation potential, which makes it applicable to various applications of computer vision.*

## 1 Introduction

Intelligent video-based surveillance systems are crucial in both public and private sites to provide safety for humans and property. Generally, video-based surveillance systems target three key features: detection of objects, tracking of objects, and analysis of the behavior of objects. Object detection, especially moving object detection, is an important task for these systems. This importance is motivated by the fact that the number and individual locations of moving objects are essential to perform further semantic analysis. The state-of-the-art approaches for moving object detection can be divided into two categories: training-based object detection and training-free object detection.

For training-based detectors, objects are located by scanning the video frames, using a trained object detector. Traditionally, the object detector is constructed by offline training on large datasets [1]. The drawback of these methods is that they need to be trained for each object of interest. For example, different training sets and descriptors are needed for vehicles on a busy road and pedestrians in a shopping area. Moreover, objects can be missed if their appearances differ from the training samples. There are also methods based on online learning [3], which adaptively update their knowledge and descriptions on the objects to be detected. However, these detectors often suffer from drifting problems, e.g., it may gradually learn to detect objects that are not of primary interest. Another popular technique, instead of learning the appearance of objects, is to learn the appearance of the background and to perform background subtraction [2]. When the background is complex and dynamic, learning the background model can be challenging, often leading to erroneous object detections.

For training-free detectors, various motion segmentation methods have been developed. Some techniques [4, 5] aim to group pixels into segments, according to the motion patterns of foreground objects. However, they cannot handle many relevant practical cases, e.g. waving branches or rippling water occurring in the background. The DECOLOR approach [6] is claimed to solve the above problems, but in our experience, the segmentation of many moving objects in close proximity of each other remains challenging.

In this paper, we present a novel two-stage training-free moving object detection system. It is specifically designed to handle challenging detection tasks, such as view-independent object detection and occlusion handling. This approach enables a broad usage of the system in various surveillance scenarios. The first stage of the system is a novel hierarchical color-guided motion segmentation algorithm. It uses a hierarchical segmentation that starts on the basis of static features (color) and increasingly incorporates temporal features to define the segments. This approach is motivated by the fact that static features are distinctive at the level of object parts, but are ambiguous at object level (objects can have multiple colors). For temporal features, the contrary is true: they are ambiguous at object-part level, but are very distinctive at the object level [5]. By combining the two features in a hierarchical approach, we can exploit features at the level where they are most discriminative. The output of the first stage consists of the locations of object candidates together with a dense motion field (optical flow). After this stage, we employ a robust motion saliency and consistency filtering [7] to finalize the object detections.

The major benefits of our moving object detection system are as follows. First, the system is training-free and has the ability to handle challenging detection tasks, such as view-independent object detection and occlusion handling. This ensures a broad usage of the system within various surveillance domains. Second, by using our novel hierarchical segmentation, it will be shown that moving object detection can be improved with respect to the state-of-the-art [6]. Third, the improved locations of objects together with their motion maps enable advanced semantic extensions of the system, such as object tracking and behavior recognition.

This paper is organized as follows. Section 2 begins with an overview of the framework. Sections 2.1 and 2.2 explain the two stages of the object detection system in detail. Section 3 illustrates the experimental results of our HiCoMo system and compares it with a state-of-the-art approach [6]. In Section 4, conclusions are drawn and future work is discussed.

## 2 Moving Object Detection

Our object detection system uses a widely adopted two-stage process: the locations of moving objects are defined in the first stage, and the detections are completed in the second stage. The system design is visualized in Figure 1. The first stage presents our novel segmentation algorithm: hierarchical color-guided motion segmentation. The system aims at segmenting a video frame into semantic regions for reliable hypothesis generation of moving objects. For doing so, it uses a bottom-up segmentation process with 6 levels of hierarchy. In the second stage, motion-based verification of the hypotheses is performed to refine the detections, based on motion saliency and consistency.

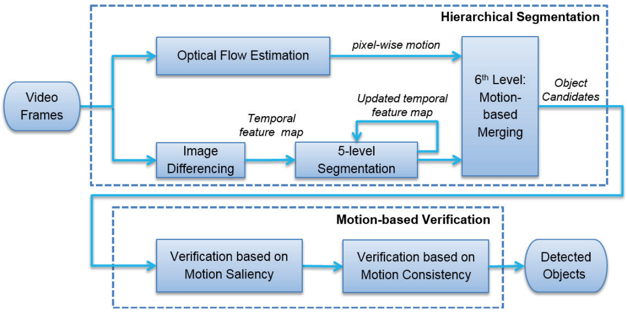

Hierarchical Segmentation
Video Frames
Optical Flow Estimation
pixel-wise motion
6th Level: Motion-based Merging
Object Candidates
Updated temporal feature map
Temporal feature map
Image Differencing
5-level Segmentation
Motion-based Verification
Verification based on Motion Saliency
Verification based on Motion Consistency
Detected Objects

Figure 1. Overview of HiCoMo system.

### 2.1 Object Hypothesis Generation

In video analysis, static features and temporal features are both important to segment the frame into semantic regions. It is obvious that static features are ambiguous at object level, while the temporal features are inaccurate at pixel level. For example, objects can contain several segments, each having their own color. This makes static features (color) unreliable descriptors of such objects. For temporal features, pixel-based motion vectors can differ between the center of the object and its contour. These observations inspire us to develop our hierarchical segmentation algorithm, in which we adjust the influence of the two types of features through a fine-to-coarse level of segmentation. It generates the hypothesis of the moving objects by means of a 6-level hierarchical color-guided motion segmentation. The algorithm also creates a binary temporal feature map and a dense optical flow map for the second stage, in order to refine the hypothesis. In the first 5 hierarchical levels of our segmentation, we use the Felzenszwalb segmentation [8] algorithm.

### 2.1.1 Felzenszwalb Algorithm

Felzenszwalb's method initially assumes that each pixel is an individual segment and an edge weight is defined as the difference of RGB values for each pair of neighboring pixels. For each segment, an intra-region weight $W_A$ is calculated as the maximum edge weight within a Segment $A$. The merging of segments is based on two values, the *minimum intra-weight* $W_{m-a}(A,B)$ of a pair of segments ($A$ and $B$) and the *minimum inter-weight* $W_{m-i}(A,B)$. The weight $W_{m-a}(A,B)$ is defined as $min(W_A+\tau_A, W_B+\tau_B)$, where $\tau$ is a threshold set to $\tau_A=\kappa/|A|$, where $|A|$ is the amount of pixels in Segment $A$ and $\kappa$ is a constant. The weight $W_{m-i}(A,B)$ is defined as the minimum edge weight in the union set $A \bigcap B$. The criterion for merging the two segments is called merge predicate $P(A,B)$:

$$P(A,B)=\begin{cases} 1 \text{ if } W_{m-i}(A,B)<W_{m-a}(A,B), \\ 0 \text{ else.} \end{cases} \tag{1}$$

### 2.1.2 Stage 1: Hierarchical Color-guided Motion Segmentation

The 6 levels in our hierarchical segmentation are organized as follows. The first level performs purely color-based over-segmentation. Then, Levels 2 to 5 perform temporal-color-based segmentation, where the influence of the temporal features is increased at each level, while Level 6 performs segment merging only using temporal features. For efficiency towards real-time operation, we define two types of temporal features: one is based on image differencing of consecutive frames and the other is based on optical flow estimation. The first one is less computational expensive, but sufficient to define the merge predicate in case the temporal features are not decisive for segmentation (Level 2 to 5). The second one is more accurate and used when we fully rely on the temporal features as in Level 6. At Level 1, we use the original Felzenszwalb's algorithm to over-segment the image using color features only. For Levels 2 to 5, we introduce a temporal-color merge predicate:

$$P(A,B)=\begin{cases} 1 \text{ if } (W_{m-i}(A,B)<W_{m-a}(A,B))\wedge \\ \quad (M(A)=M(B)), \\ 0 \text{ else.} \end{cases} \tag{2}$$

It contains the original color-only terms $W_{m-a}(A,B)$ and $W_{m-i}(A,B)$ with an additional temporal feature term $M(A)=M(B)$. This term expresses that the temporal features of Segment $A$ and $B$ must be equal in order for them to merge. Here, we use the first type of temporal features, which results in a binary temporal feature map $M$ and is computed by thresholding the absolute intensity differences of pixels from consecutive frames. A pixel in $M$ has the value unity, if the absolute pixel difference between consecutive frames is larger than 5, and 0 otherwise (see [9]). Let $M_A$ denote the number of pixels in $A$ for which their value in $M$ equals unity, then we obtain the temporal feature of Segment $A$ with:

$$M(A)=\begin{cases} 1 & \text{if } M_A/|A|\geq\delta, \\ 0 & \text{if } M_A/|A|<\delta, \end{cases} \tag{3}$$

where $|A|$ is the number of pixels in Segment $A$ and $\delta$ denotes the fraction threshold, indicating that a certain fraction of the pixels in Segment $A$ have changed. More precisely, the temporal feature of a segment expresses that at least a certain percentage of its pixels has a temporal difference larger than 5. We increase the fraction threshold $\delta$ in each of the 4 color-temporal segmentation levels. The values used for $\delta$ are $\{0.05, 0.1, 0.15, 0.2\}$. By increasing $\delta$, we raise the influence of temporal features. This can be explained: when considering $\delta=0$, the temporal features of all segments have value unity and therefore are not distinctive; by increasing $\delta$, some segments will keep the value unity and others obtain value zero, thereby increasing the distinctiveness of the temporal features.

Besides increasing the influence of temporal features, we also decrease the influence of color features. We do this by increasing $\kappa$ used to compute the weight $W_{m-a}(A,B)$. This allows larger pixel color differences to appear inside a segment, so that the influence of color features is reduced. Furthermore, we start with a minimum segment size $S_m$ and increase this value for Levels 3 and 5. To improve efficiency, we remove redundant edges from the graph prior to continuing with the next level of segmentation.

In the 6th and final level of segmentation, we fully rely on the temporal features to finalize the segmentation. Here, we use the second type of temporal features, i.e. the dense normalized optical flow. For each segment, its temporal feature is computed as the average and variance of the flow inside the segment. On the basis of these features, we apply a statistical segment merging [7]. This involves measuring the several statistical parameters for each segment and then merging segments when these parameters are sufficiently close.

The final temporal feature map is created by taking the outputs of the 6th level and computing the binary map, according to Equation (3), with a threshold $\delta = 0.3$. This temporal feature map, providing moving object hypotheses, is returned together with the optical flow field to the second stage of our HiCoMo system, which is described in the next section. The pseudocode of the above-described hierarchical segmentation is provided by Algorithm 1.

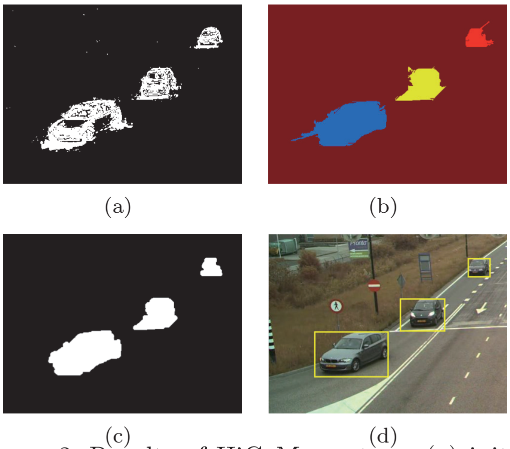
```
Input: Two consecutive frames;
Level 1: Create initial temporal feature map; perform
over-segmentation based on color features;
for Level 2 to 5 do
    Inherit graph expression of the image; update $\kappa$ and $\delta$;
    update temporal feature map;
    for each segment do
        Extract the pair of segments $A$ and $B$;
        re-calculate the $W_{m-a}(A,B)$ using new $\kappa$
        if $P(A,B)=1$ according to Eqn. (2) then
            Join A and B as a new segment; Update the
            weights of the new segment and delete the
            corresponding edges;
        end
    end
end
Level 6: Perform statistical segment merging based on dense
optical flow.
Output: Temporal feature map (hypotheses of moving
objects) with dense optical flow map.
```

**Algorithm 1:** Our hierarchical color-guided motion segmentation in pseudo-code.

### 2.2 Stage 2: Object Hypothesis Verification

At this stage, the previously generated moving object hypotheses are verified in order to remove the false alarms. The verification is achieved through a cascaded motion-based hypothesis verification, which employs motion saliency and consistency.

*A. Verification based on Motion Saliency.* In video surveillance, we aim at detecting moving objects, which implies that they have salient motion compared to their surroundings. We remove the false positives from our candidate detections by checking motion saliency. We first extract the Region of Interest (ROI), including the region outer part $R_{obj}$ of a candidate and its local background $R_{bg}$, as defined in [7]. Then, we calculate the region-level motion of $R_{obj}$ as $\mathbf{v}_{obj}$ and of $R_{bg}$ as $\mathbf{v}_{bg}$, based on the optical flow calculated in the previous stage. Last, motion saliency is defined by two criteria to verify the hypothesis: $|\mathbf{v}_{obj}| - |\mathbf{v}_{bg}| > T_1$ and $\frac{|\mathbf{v}_{obj} - \mathbf{v}_{bg}|}{|\mathbf{v}_{obj}|} > T_2$. The two criteria are imposed to remove false positives whose motion is intrinsically small (Criterion 1) or relatively small (Criterion 2) compared to the surroundings (e.g. waving trees).


(a) (b)

(c) (d)

Figure 2. Results of HiCoMo system: (a) initial temporal feature map; (b) hierarchical segmentation; (c) final temporal feature map; (d) detections after object hypothesis verification.

*B. Verification based on Motion Consistency.* Motion consistency is defined to further improve the detections based on the assumption that a moving object cannot suddenly disappear unless it leaves the monitored area. For each detection in the previous frame, we search for a pre-defined neighboring area in the current frame. If there is no detection in the search area, the previous detection is propagated to the current frame. The recovered detection is re-verified through the motion-saliency verification described in the previous paragraph. Complex detection-by-tracking algorithms are thus avoided by re-using the flow values computed before. Figure 2 illustrates the outputs of the different stages of our system.

## 3 Experiments

We validate our moving object detection system using three videos captured from crossroads ($S1$), two videos from parking lots ($S2$) and one video from "PETS 2009" ($S3$) [10]. The crossroads and parking lots videos have a resolution of 1280 × 960 pixels and their length varies between 180 and 260 frames. "PETS 2009" video has a resolution of 768 × 576 pixels and 230 frames. Two categories of moving objects are captured: vehicles and pedestrians. The vehicles are of different types (passenger car, van, truck, etc.) and have a variable motion pattern (cruising, turning and parking). Furthermore, the vehicles and pedestrians present are recorded from different viewing angles and distances from the camera. For parameters used in our system: $T_1$ and $T_2$ are set to 0.1; the minimum segment sizes $S_m$ used for each of the 6 segmentation levels are $\{100, 100, 200, 200, 400, 400\}$ for Levels 1 to 6, respectively; for segmentation Levels 1 to 5, $\kappa$ equals $\{50, 150, 250, 450, 850\}$, respectively. These parameters were set experimentally and applied to all our test videos. To evaluate the performance of our HiCoMo system, we have compared it with a state-of-the-art

Table 1. Detection results of HiCoMo and DECOLOR. “Num” indicates the number of moving objects in the videos. “P” and “R” stands for Precision and Recall respectively.

| Videos | Num | Methods | P (%) | R (%) |
|---|---|---|---|---|
| $S1$ | 994 | HiCoMo | 99.0 | 91.2 |
| | | DECOLOR | 98.3 | 87.6 |
| $S2$ | 605 | HiCoMo | 95.2 | 90.9 |
| | | DECOLOR | 96.3 | 85.1 |
| $S3$ | 979 | HiCoMo | 94.7 | 88.9 |
| | | DECOLOR | 97.2 | 73.3 |

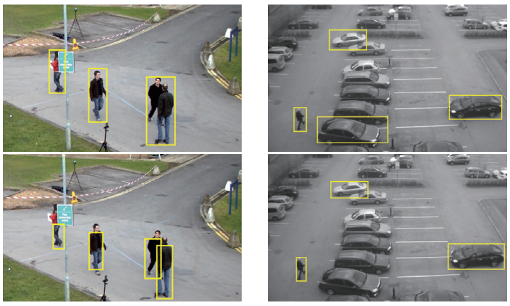

(a) (b)

Figure 3. Detection results: upper shows results of HiCoMo and bottom are results of DECOLOR. (a) Pedestrians are crowded and occluded; (b) a vehicle is parking, a vehicle is partly occluded and a pedestrian is walking.

moving object detection algorithm DECOLOR [6].

Table 1 shows the detection results of the two approaches, in terms of recall and precision. The average recall over all 6 videos of our approach is 90.2%, compared to 81.6% of DECOLOR. The improved result of our HiCoMo system can be explained by its better segmentation capability, when multiple objects are close to each other. Our approach achieves this segmentation, by adjusting the influence of static features and temporal features at different levels of segmentation. This can be seen from the adaptation of the $\delta$ fraction parameter and $\kappa$ parameter adjustment for each segmentation level. This mechanism ensures more reliable semantic segments. In $S2$ and $S3$, the precision of our approach is lower because the parameters in our segmentation are chosen to allow aggressive separation of occluded objects. It results in a slight decrease in precision, which is compensated by large improvement in recall. This design is motivated by our aim to minimize miss detections while keeping high precision. Figure 3 shows a visual comparison of HiCoMo and DECOLOR. In Figure 3(a), DECOLOR erroneously detects two crowded pedestrians as a single object while HiCoMo successfully separates them from each other. In Figure 3(b), both approaches locate the walking pedestrian and the vehicle partly occluded with one parked vehicle. However, DECOLOR misses one car which is turning for parking, because the local change in the object is small, which indicates an example failure of DECOLOR. In contrast, our algorithm exploits static features to guide the segmentation and successfully separates the vehicle from the background.

## 4 Conclusions and Future work

In this paper, we have presented our HiCoMo system: a training-free system for moving object detection by introducing the hierarchical color-guided motion segmentation algorithm. The hierarchical segmentation functions as the first stage of the system to locate moving object candidates. This segmentation aims at clustering the changes in the scene so that moving objects are found. The hierarchy ensures that temporal features obtain gradually a higher weight than static features. Motion saliency and consistency are analyzed in the second stage, to obtain a higher accuracy and improved robustness in moving object detection.

We have evaluated our HiCoMo system for various surveillance scenes with different moving objects. The system is compared against a state-of-the-art training-free detection approach. The experiments show that the HiCoMo system has a competitive precision of 96.5% (compared to 97.4%) and higher recall of 90.2% (compared to 81.6%). Furthermore, our system demonstrates a better segmentation ability when objects are close to each other. Therefore, we can conclude that our training-free approach is attractive to many video surveillance scenarios and applications.

In future work, extensive tests will be performed to validate the generic applicability of the system. For this, more datasets will be recorded, each focusing on different surveillance applications, i.e., harbor surveillance, traffic surveillance, and perimeter surveillance.

## References

[1] P. Dollar, C. Wojek, B. Schiele and P. Perona. “Pedestrian detection: an evaluation of the state of the art,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34 (4), pp. 743-761, 2012.

[2] T. Huang, J. Qiu, T. Sakayori and T. Ikenaga. “Robust background segmentation using background models for surveillance application,” *Machine Vision Applications (MVA)*, Tokyo, 2009.

[3] S. Sivaraman, M. M. Trivedi. “Active learning for on-road vehicle detection: a comparative study,” *Machine Vision and Applications*, vol. 25 (3), pp. 599-611, 2014.

[4] D. Cremers and S. Soatto. “Motion competition: a variational approach to piecewise parametric motion segmentation,” *Int. J. Comput. Vision.*, vol. 62 (3), pp. 249-265, 2005.

[5] T. Brox and J. Malik. “Object segmentation by long term analysis of point trajectories,” *European Conference on Computer Vision*, Crete, 2010.

[6] X. Zhou, C. Yang and W. Yu. “Moving object detection by detecting contiguous outliers in the low-rank representation,” *Trans. Pattern Anal. Mach. Intell.*, vol. 35 (3), pp. 597-610, 2013.

[7] X. Bao, S. Javanbakhti, S. Zinger, R. Wijnhoven and P. H. N. de With. “Context modeling combined with motion analysis for moving ship detection in port surveillance,” *J. of Electronic Imaging.*, vol. 32 (4), 2013.

[8] P. Felzenszwalb and D. Huttenlocher. “Efficient graph-based image segmentation,” *Int. J. Comput. Vision.*, vol. 59 (2), pp. 597-610, 2004.

[9] P.L. Rosin. “Thresholding for change detection,” *Int. Conf. on Computer Vision*, Bombay, 1998.

[10] http://www.cvg.reading.ac.uk/PETS2009/a.html