Adaptive Color Background Modeling for Real-Time Segmentation of Video Streams*

Alexandre R.J. François and Gérard G. Medioni

Integrated Media Systems Center
Institute for Robotics and Intelligent Systems - PHE 204
University of Southern California, Los Angeles, CA, 90089-0273, USA

Abstract

We present a system to perform real-time background modeling and segmentation of video streams on a PC, in the context of video surveillance and multimedia applications.

The images, captured with a fixed camera, are modeled as a fixed or slowly changing background, which may become occluded by mobile agents. The system learns a statistical color model of the background, which is used for detecting changes produced by occluding elements. We propose to operate in the Hue-Saturation-Value (HSV) color space, instead of the traditional RGB space, and show that it provides a better use of the color information, and naturally incorporates gray-level only processing. At each instant, the system maintains an updated background model, and a list of occluding regions that can then be tracked. Other applications are video compression, enhancement and modification, such as obstacle highlight or removal.

Keywords*: Image sequence processing; Adaptive background modeling; Video stream segmentation.*

1 Introduction

We present a system to perform real-time adaptive background modeling and segmentation of video streams.

Our goal is to segment non background object silhouettes in video streams taken from a fixed camera. A number of systems performing segmentation of moving objects rely on image differencing, with various enhancements (*e.g.* [6]). A binary mask of moving pixels is obtained by thresholding the absolute difference images computed between consecutive frames. The binary mask is then segmented by connected components labeling. This method, however, is sensitive to noise in the background, slowly changing lighting, and fails to segment the non-background objects if they stop moving. The same problem occurs with methods based on spatio-temporal derivatives (e.g. [1]), which can only detect moving textured regions. This problem can be solved by estimating the background over a long image sequence, and then thresholding the absolute difference between the incoming frames and the estimated background [8]. Recent work on image sequence restoration uses a variational approach to estimate the background in a set of consecutive frames and segment the non-background regions [7]. This system requires that the entire sequence be processed before performing segmentation, and thus does not perform in real-time.

Following the Pfinder approach [10], and more recent work based on this idea [4][2], our detection engine uses an adaptive, statistical background model to detect changes in the scene. Specifically, we model the frames, taken by a fixed camera, as a fixed or slowly changing back-

*This research has been funded [or funded in part] by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center, Cooperative Agreement No. EEC-9529152, with additional support from the Annenberg Center for Communication at the University of Southern California and the California Trade and Commerce Agency.

ground, locally occluded by agents. The detection process encompasses the dynamic learning of the background, and the occluded regions segmentation. Our main contribution is the refinement of the background model dynamics using the HSV (Hue Saturation Value) color space [9]. We do not assume any specific model for the occluding elements.

The background model is pixel-based: each observed pixel value is assumed to result from an independent process. Background pixel values are modeled as multi-dimensional Gaussian distributions in HSV color space. When a new frame is processed (see figure 1), the value observed for each pixel is compared to the current corresponding distribution in order to decide whether the value is a measurement of the background or of an occluding element. The pixels marked as occlusions are then grouped into connected components. The result is a list of regions representing the segmented occluded areas in the scene. Regions whose size is below a certain threshold are rejected as false positives. The distribution for each background pixel is updated using the latest observation, in order to take into account changes in the scene background.

The segmented regions can be used for tracking, after eventual further segmentation into components based on models. Other applications are video compression, enhancement and modification, such as obstacle highlight or removal.

In the remainder of this paper, we describe the details of the background model: the pixel model and especially the choice of the color space in section 2, the occlusion detection in section 3 and the distribution update in section 4. We present results obtained with our system in section 5 and conclude this paper by a summary of our contribution and a discussion of future work in section 6.

2 Modeling the background

We detail the representation in the background model for one pixel, each pixel following the same model.

A background pixel is modeled by a Gaussian distribution, characterized by its mean value μ and its standard deviation σ. Theoretically, any available image measurement can be used in the pixel representation. If the use of color information is current in recent comparable systems, the color space used is generally the RGB space, since RGB values are readily provided by most frame grabbers. The RGB space, however, is not well behaved with respect to color perception, as a distance computed between two colors in RGB space does not reflect their perceptual similarity. In our experiments, processing each dimension independently improved the accuracy of the segmentation, but working in RGB space still does not allow modeling of camera-created artifacts that occur in high contrast areas, such as ghost shadows and hue inconsistency.

A color model which clearly separates the intensity and chromatic information of the pixel measurement allows us to consider chromaticity in a

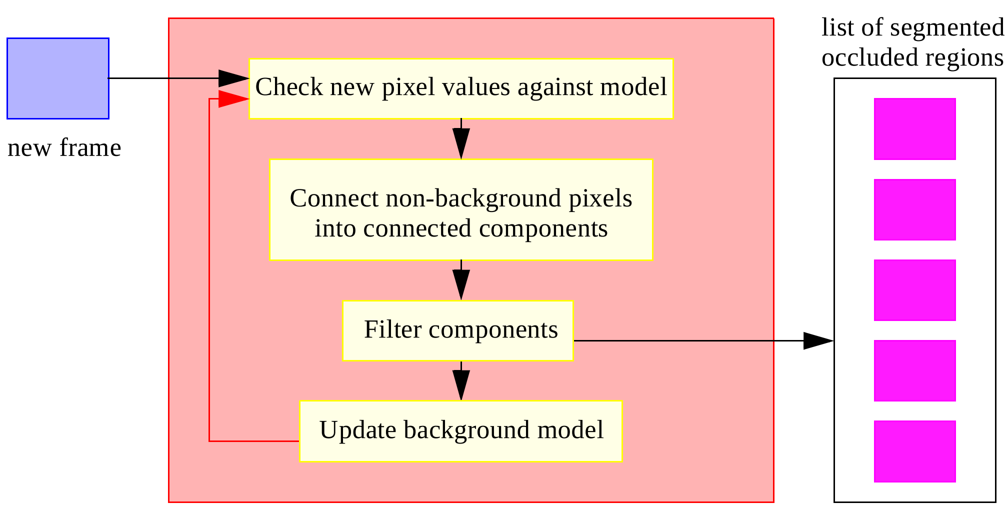


Fig. 1. One cycle of the target detection engine

framework generalizing naturally intensity only models. Pfinder uses the YUV color space, which separates intensity (Y) and chromaticity (U,V) in the pixel measurement. Similarly, the HSV model (see figure 2) separates the intensity (V) from the chromatic components (H,S). However, the UV subspace representation of chromaticity, based on linear combinations of R, G and B channels, is not as intuitive as the radial HS subspace representation (see figure 3). The geometry of the HSV space makes it more suitable to develop algorithms that rely on intensity measurements and on color information *when available and relevant.*

In our implementation, the image pixel values are HSV triplets. A background pixel distribution *B(x,y)* is a vector of 3 variables *H(x,y)*, *S(x,y)* and *V(x,y)*, each characterized by a mean μ_k and a standard deviation σ_k. The three dimensions are obviously not analogous, and cannot be processed in the same way. Using HSV values thus results in more processing (conversion of RGB measurements into HSV space and more complicated process), but, with today's computing power, it is a small price to pay for better results, in a more intuitive color processing framework.

The pixel distributions are initially unknown. In our implementation, we initialize the means using the values of the first frame, and set the standard deviations to zero. The actual distributions are learned as the incoming frames are processed.

We now describe the two steps performed for each new frame: the occluding pixels detection and the model update.

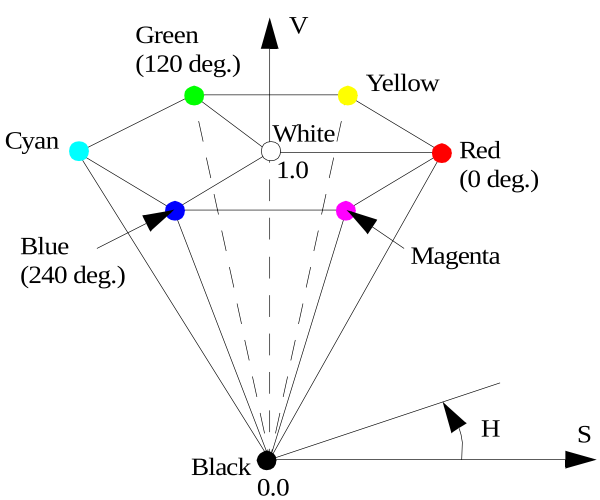


Fig. 2. The HSV color model

3 Occluding pixel detection

When a new frame is processed, each new observation (image pixel value) is checked against the current corresponding model distribution. The output is a binary mask marking the pixels interpreted as occluding the background.

The decision whether an observed value is an actual observation of the background or of an occluding agent is made based on the intensity and chromaticity information. In order for a pixel value to be interpreted as an observation of the background, each of the *meaningful* measured components H, S and V must fall within two standard deviations of the mean value of the component in the corresponding model distribution. The Value V, which represents the intensity in a gray level image, is always used. The color information, encoded in the hue H and the saturation S, is not always reliable: if the saturation, scaled by the value, is below a certain threshold, fixed according to the camera parameters and scene general lighting conditions, the measurement is said achromatic, the hue information is not relevant and thus not used. The saturation is scaled by the value to reflect the drop in color measurement accuracy when the intensity decreases. This corresponds to a dark grey subcone in HSV space. The algorithm distinguishes all combinations of chromaticity/achromaticity of the observed value and model distribution. Let *Hm*, *Sm*, *Vm* denote the current mean model values, and *Ho*, *So*, *Vo* denote the observed values for the corresponding pixel. We define the value *S'* as *S/V*. Let *St* be the chromaticity saturation threshold. The components used for the decision are as follows:

- if *Sm*<*St* and *So*<*St*, check *Vo* against *Vm*
- if *Sm*<*St* and *So*>*St*, check *Vo* against *Vm* and *So* against *Sm'*
- if *Sm*>*St* and *So*<*St*, check *Vo* against *Vm* and *So'* against *Sm*
- if *Sm>St* and *So>St*, check *Vo* against *Vm*, *So.cos(Ho,Hm)* against *Sm* and *Ho* against *Hm*. In this case the observed saturation is projected on the mean hue direction.

In this framework, the intensity only processing appears as a special case where no color information is ever available.

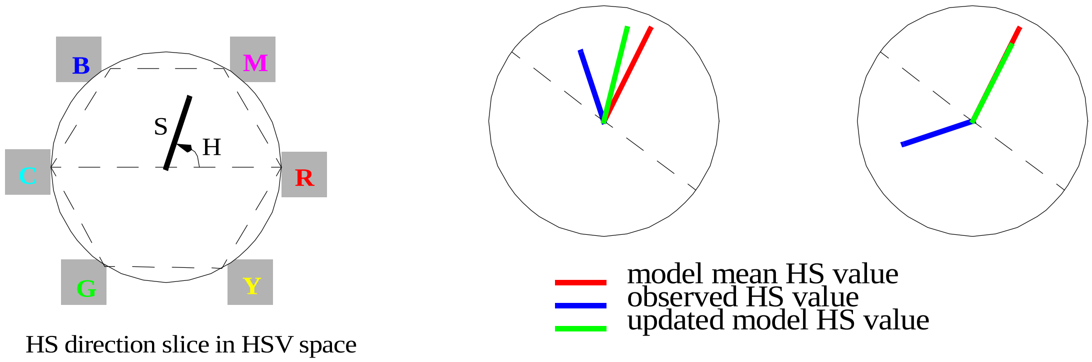


Fig. 3. Update of chromatic information

4 Background model update

After occluding pixels have been detected, the current distributions are updated to incorporate the latest information. Let x denote the current observation for a pixel. The distribution for the corresponding background pixel is updated according to the following general laws:

$$\mu \leftarrow (1-\alpha)\mu + \alpha x,$$

$$\sigma^2 \leftarrow \max(\sigma^2_{min}, (1-\alpha)\sigma^2 + \alpha(x-\mu)^2)$$

where α is the learning rate. A minimum standard deviation vector σ_{min} is introduced as a noise threshold, to prevent the standard deviation from decreasing below a minimum value should the background measurements remain strictly constant over a long period of time. These parameters are applied globally for all the individual background pixel models.

Here again, the update of the different distribution components is subject to the availability and the relevance of the chromaticity information of the observed values. The algorithm distinguishes all combinations of chromaticity/achromaticity for the observation and the model. In the case where both the observed and model values are chromatic, if the current observation and the model have very different hues, the model hue will not be updated. By using the observed saturation projected on the mean hue, we ensure that the mean saturation decreases, reflecting an increase in the uncertainty of the hue information (see figure 3).These update rules are made possible by the use of the HSV color model, and allow to average out, in the model, hue artifacts created by the imaging process. We show in figure 4 an example of background model learned with our algorithm. Chromaticity information is used only in areas where it is reliable, as shown in the chromaticity image. The saturation variance is high in dark chromatic areas, and along high contrast edges where color artifacts occur. In achromatic regions, the saturation is low and its variance is low. Naturally, the hue variance remains high in achromatic areas where the hue is by definition not reliable.

5 Results

We have implemented the described detection system on a 400 Mhz Pentium II PC, using the Intel Image Processing Library [5] to take advantage of the MMX instruction set of the Pentium II processor. The most dramatic contribution of this tool is in the speed of the RGB to HSV color space conversion. The system performs frame capture, process and display of segmentation results overlaid on frame at about 5 Hz on 160x120 pixel frames.

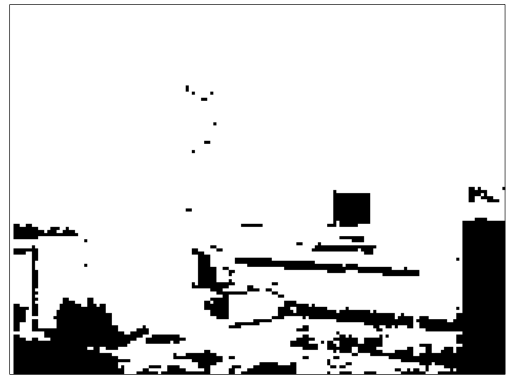
Screen shots of the application are presented figure 5, with segmentation results.

We have also run tests on gray level, noisy sequences used to demonstrate an image segmentation algorithm using a variational approach [7]. The frame size is 160x128 pixels. Our algorithm gives, in real-time, results of a quality comparable to those obtained with the variational method, which is not a real-time technique. Our results are presented in figure 6.

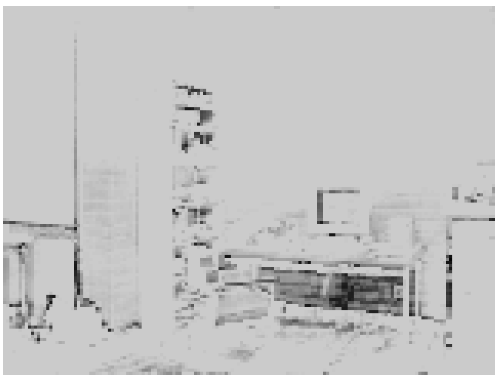
6 Conclusion

We have presented a system to perform real-time adaptive background modeling and segmentation

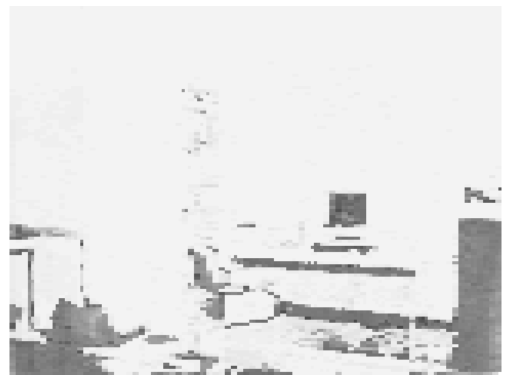
(a) mean values



(b) chromatic (white) and achromatic (black) pixel distributions



(c) variance of saturation (darker means higher)



(d) variance of hue (darker means higher)

Fig. 4. Learned background distributions

of video streams on a PC. We have described the details of the background learning and segmentation. The use of the HSV color model is central to our approach. By using color information only when relevant and in a perceptually consistent way, we can efficiently model the color background and accurately segment occluded regions. Future developments include speed optimization, and generalization to the Pan-Tilt-Zoom camera model. This system is also used as a detection engine in integrated systems under development for video surveillance and multimedia applications.

7 References

[1] Bichsel M., “Segmenting Simply Connected Moving Objects in a Static Scene”. In PAMI, vol. 16, no. 11, pp. 1138-1142, November 1994.

[2] Eveland C., Konolige K., Bolles R.C., “Background Modeling for Segmentation of Video-rate Stereo Sequences”. In Proc. CVPR, pp. 266-272, Santa Barbara, CA, June 1998.

[3] Foley J.D., van Dam A., Feiner S.K., Hughes J.F., Computer Graphics: Principles and Practice, 2nd ed.. Addison-Wesley, 1996.

[4] Grimson W.E.L., Lee L., Romano R., Stauffer C., “Using Adaptive Tracking to Classify and Monitor Activities in a Site”. In Proc. CVPR, pp. 22-31, Santa Barbara, CA, June 1998.

[5] Intel Image Processing Library, http://developer.intel.com/vtune/perflibst/ipl/index.htm

[6] Jain R.C., “Segmentation of Frame Sequences Obtained by a Moving Observer”. In PAMI, vol. 6, no. 5, pp. 624-629, September 1984.

[7] Kornprobst P., Deriche R. and Aubert G., “Image Sequence Analysis via Partial Differential Equations”. To appear in Journal of Mathematical Imaging and Vision.

[8] Shio A. and Sklansky J., “Segmentation of People in Motion”. In Proc. IEEE Workshop on Visual Motion, pp. 325-332, Princeton, NJ, 1991.

[9] Smith A.R., “Color Gamut Transform Pairs”. In Proc. SIGGRAPH, pp. 12-19, 1978.

[10] Wren C.R., Azarbayejani A., Darrell T.J., Pentland A.P., “Pfinder: Real-Time Tracking of the Human Body”. In PAMI, vol. 19, no. 7, pp. 780-785, July 1997.

Fig. 5. Screen shots of our system with segmentation results

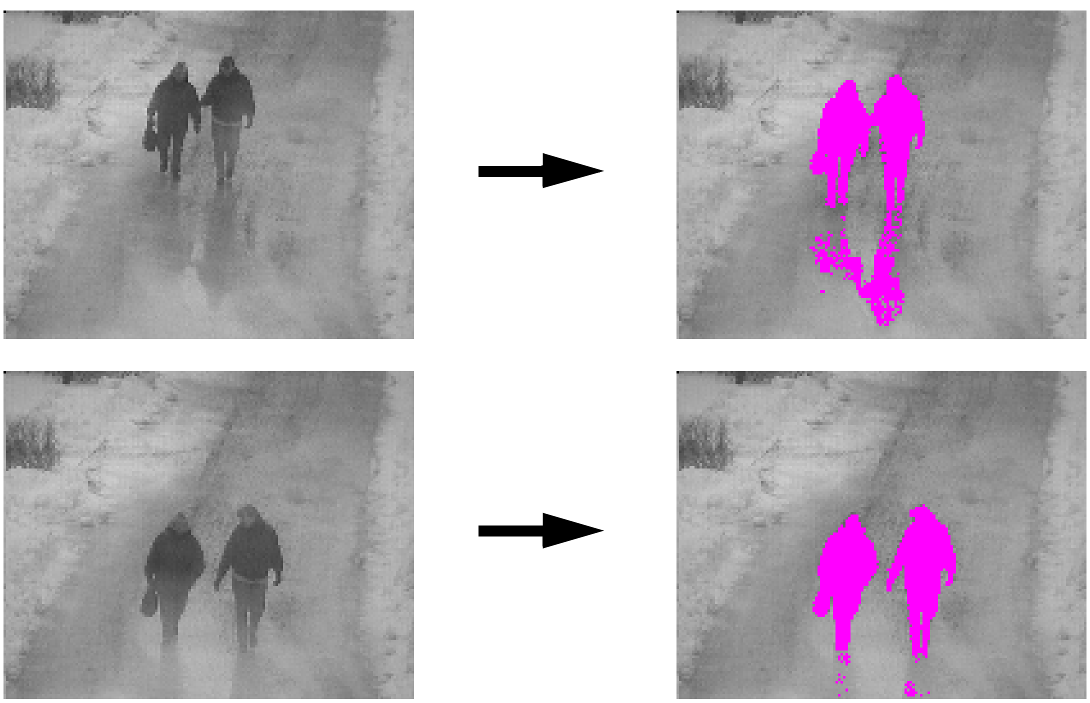


Fig. 6. Segmentation result on two frames of the Sweden sequence from [7]