J. KSIAM Vol.13, No.4, 315–321, 2009

SIMULTANEOUS FOREGROUND AND BACKGROUND SEGMENTATION WITH LEVEL SET FUNCTION

SUK-HO LEE[1]

[1] DEPT. MULTIMEDIA ENGINEERING, DONGSEO UNIVERSITY, SOUTH KOREA
E-mail address: petrasuk@gmail.com

ABSTRACT. In this paper, a level set based energy functional is proposed, the minimization of which results in simultaneous reference background image modeling and foreground segmentation. Due to the mutual constraint of the two processes, a good estimate of the background can be obtained with a small number of frames, and due to the use of the level set, an Euler-Lagrange equation that directly solves the problem can be derived.

1. INTRODUCTION

Background subtraction refers to segmentation techniques that are applied to video sequences to segment the moving object out by comparing the current frame against an estimate of the reference background image. The regions that show significant differences in the brightness value of the observed and the estimated reference background image indicate the regions in which motion has occurred. The reference background image is usually estimated from a given sequence taken of the same scene for a certain time interval, and the success of the background subtraction technique relies on how well the background image is estimated. Ideally, the reference background image should contain no moving regions in it. However, the difficulty in estimating the reference background image lies in the fact that all the frames in the given image sequence may contain moving objects which have to be excluded from the background reference image.

Background subtraction methods vary according to how the background image is modeled and how the moving regions, i.e., the foreground is removed from the background image. Several background subtraction techniques update the reference background image by blending the current background image with the current frame, where the degree of blending is determined by a certain blending parameter [1]-[3]. In [4][5], the background image is modeled by modeling the color of each pixel by a single or a mixture of Gaussians.

All the background subtraction techniques confront the trade-off problem that either a large

Received by the editors July 16 2009; Accepted October 12 2009.
2000 *Mathematics Subject Classification.* 68U10.
Key words and phrases. Partial differential equation, Level Set, Background modeling, Object Detection.
This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD, Basic Research Promotion Fund) (KRF-2008-331-D00566).

number of frames are required to obtain a "clean" reference background image, or moving objects may leave some "tails" in the background image. If several moving objects are present in the scene all the time, then a sufficient large number of frames are required, which again require for a large storage memory. Since the foreground segmentation, i.e., motion segmentation process and the background image modeling process have a mutual influence on each other, the required frame number can be reduced, if those two processes are performed in coupled way such that the two processes have an positive effect on each other. An energy functional has been proposed in [6] which deals the problem of motion segmentation and background image formation in a coupled way. However, a direct Euler-Lagrange equation cannot be derived from the energy functional, and the partitioning function that performs the foreground-background segmentation has to be obtained as the limit of the minimizing sequence, which is difficult to work in real-time.

In this paper, we propose a level set based energy functional that deals the problem of foreground segmentation and background image formation in a coupled way such that the two processes mutually constrain each other, and from which a level set based Euler-Lagrange equation can be derived. Due to the mutual constraining, a good estimation of the static background can be obtained with a relatively small number of frames. Furthermore, the algorithm works in real-time due to the direct implementation of the Euler-Lagrange equation, which can be solved with one iteration.

2. PROPOSED MODEL

We introduce the following energy functional that formulates the problem of simultaneous background modeling and foreground segmentation into a level set based minimization problem:

$$E(B,\phi) = \int_{\Delta t}\int_{\Omega} \phi(\mathbf{r})H(\phi(\mathbf{r}))\left[(B(\mathbf{r}) - I(\mathbf{r},t))^2 - \alpha\right] d\mathbf{r}dt \tag{2.1}$$

where B is the background image, α is a positive constant, ϕ is the level set function, $I(t)$ is the frame at time t, Ω is the domain of the image, Δt is a certain time interval along the sequential axis, and $H(\phi)$ is the Heaviside function which is defined as follows:

$$H(\phi) = \begin{cases} 1 & \text{if } \phi \geq 0 \\ 0 & \text{if } \phi < 0. \end{cases}$$

The integration over the time interval Δt can be the integration of all contiguous frames in the time interval Δt, or can be the integration of a sampled version of the frames, e.g., the integration of every fourth frame in the time interval. The energy functional is minimized with respect to the background image B and the level set function ϕ, which are the solutions being sought. The level set function plays two different roles in the scheme. First, it acts as a partitioning operator which segments the image region into the foreground and the background region, where $\{\mathbf{r} \mid \phi(\mathbf{r}) \geq 0\}$ represents the background region, and $\{\mathbf{r} \mid \phi(\mathbf{r}) < 0\}$, the foreground region. Second, the value of the level set function acts as a weighting function which is used in the construction of the background image.

The minimization of the energy functional with respect to ϕ, when the background image B is kept as a constant, makes the $\phi(\mathbf{r})$ value at the pixel $\mathbf{r}$, where $(B(\mathbf{r}) - I(\mathbf{r},t))^2 < \alpha$ holds, increase to ∞. This is due to the fact that $\left[(B(\mathbf{r}) - I(\mathbf{r},t))^2 - \alpha\right]$ is negative, and the larger $\phi(\mathbf{r})$ becomes, the more the energy decreases. If we interrupt the minimization process after a prescribed number of iterations, then we get positive $\phi(\mathbf{r})$ values at these pixels. On the other hand, if $(B(\mathbf{r}) - I(\mathbf{r},t))^2 > \alpha$, then $\phi(\mathbf{r})$ value becomes negative, which makes the integrand zero at this point. Therefore, the minimization with respect to ϕ can be seen as a thresholding based segmentation process, where α acts as a threshold value, and every pixel $\mathbf{r}$ where $(B(\mathbf{r}) - I(\mathbf{r},t))^2 < \alpha$ becomes classified in the region $\{\mathbf{r} \mid \phi(\mathbf{r}) \geq 0\}$, while pixels at which $(B(\mathbf{r}) - I(\mathbf{r},t))^2 > \alpha$ hold become classified in the region $\{\mathbf{r} \mid \phi(\mathbf{r}) < 0\}$. As a result, the image domain becomes partitioned into the foreground and the background region.

Actually, the minimizer ϕ is a three dimensional function in the domain $[\Omega \times \Delta t]$, and has to be computed using all the image frames in the time interval Δt. However, since B is being kept fixed, and the image frames in Δt are independent of each other, we find the 2 dimensional minimizer ϕ_{2D} function for each frame instead of the 3 dimensional minimizer ϕ function, and assume that the 3 dimensional minimizer is the stack of all ϕ_{2D} slices in Δt. Therefore, we reformulate the problem as:

$$arg \min_{\phi_{2D}} \int_\Omega \phi_{2D}(\mathbf{r}) H(\phi_{2D}(\mathbf{r})) \left[(B - I(t))^2 - \alpha\right] d\mathbf{r} \tag{2.2}$$

for each frame in Δt. The minimizer of (2.2) can be obtained by letting the gradient of (2.2) to be zero.

$$\left(1 + H'(\phi_{2D})\right) \left[\alpha - (B - I(t))^2\right] = 0. \tag{2.3}$$

This minimizer can be obtained by solving the following Euler-Lagrange equation for ϕ_{2D} to the steady state:

$$\frac{\partial \phi_{2D}}{\partial t} = \left(1 + H'(\phi_{2D})\right) \left[\alpha - (B - I(t))^2\right]. \tag{2.4}$$

Now, keeping ϕ fixed, $E(\phi, B)$ is minimized with respect to B, to obtain the estimated background image. The minimizing with respect to the background image is done directly by letting the gradient of the functional with respect to the background image B be zero:

$$\begin{aligned}
\int_{\Delta t} \int_\Omega 2\phi H(\phi)(B - I(t)) d\mathbf{r} dt &= 0 \\
\Leftrightarrow \int_{\Delta t} \int_\Omega \phi H(\phi) B d\mathbf{r} dt &= \int_{\Delta t} \int_\Omega \phi H(\phi) I(t) d\mathbf{r} dt \\
\Leftrightarrow \int_{\Delta t} \phi H(\phi) dt \int_\Omega B d\mathbf{r} &= \int_{\Delta t} \int_\Omega \phi H(\phi) I(t) d\mathbf{r} dt \\
\Leftrightarrow \int_\Omega B d\mathbf{r} &= \frac{\int_{\Delta t} \int_\Omega \phi H(\phi) I(t) d\mathbf{r} dt}{\int_{\Delta t} \phi H(\phi) dt} \\
\Leftrightarrow \int_\Omega B d\mathbf{r} &= \frac{\int_\Omega \int_{\Delta t} \phi H(\phi) I(t) dt d\mathbf{r}}{\int_{\Delta t} \phi H(\phi) dt}
\end{aligned}$$

One solution that satisfies the above equation is:

$$B(\mathbf{r}) = \frac{\int_{\Delta t} \phi(\mathbf{r})H(\phi)I(t)dt}{\int_{\Delta t} \phi(\mathbf{r})H(\phi)dt}, \tag{2.5}$$

which we use for the construction of the reference background image. It should be noticed that the integration in (2.5) is along the sequential axis, and not of the image domain. The brightness value of $B(\mathbf{r})$ for each pixel $\mathbf{r}$ is a weighted average of the brightness values $I(\mathbf{r}, t)$ at the same position $\mathbf{r}$ and different t in , and $\phi(\mathbf{r})$ acts as the weighting function. The Heaviside function $H(\phi)$ ensures that only the intensity values of pixels that have positive $\phi(\mathbf{r})$ values, i.e., pixels that are in the background region, are included in the construction of the background image. Therefore, a good estimation of the static background can be obtained with relatively small number of frames. The pixels in the background region are those which differ within α in the intensity value with the current reference background image. Intensity change with magnitude less than α become reflected in the construction of the next reference background image, e.g., gradual intensity change due to slow illumination change become reflected in the next reference background image, while intensity change larger than α become not reflected. In the case that $\phi(\mathbf{r}) > 0$, the brightness values $I(\mathbf{r}, t)$ are reflected in the construction in proportion to the $\phi(\mathbf{r})$ value at t. The $\phi(\mathbf{r})(> 0)$ value has a large value if the value $\alpha - (B - I(t))^2$ is large, as can be seen in (2.4), i.e., if the intensity difference of the current frame and the reference background image is small. This means that intensity values at $\mathbf{r}$ for different t that are close to the current reference background image are weighted more in the construction of the next reference background image than intensity values that are not. Thus, the ϕ function act as a weighting function in the case that $\phi(\mathbf{r}) > 0$.

3. Numerical Implementation of the Algorithm

Even though the algorithm uses several frames for the computation of the current reference background image and implements a partial differential equation, it can be executed in real-time. Assume that the image is of size $N_{width} \times N_{height}$, and there are N_{frame} frames in Δt. Then the computation of the reference background image requires at most $2 \times N_{width} \times N_{height} \times N_{frame}$ multiplication and $N_{width} \times N_{height} \times N_{frame}$ addition operations. There are N_{frame} numbers of ϕ_{2D} functions in Δt which all have to be computed by (2.4). Equation (2.4) is implemented by a forward iteration scheme. However, actually one iteration is enough, since the relative magnitude ratios of $\phi_{2D}(\mathbf{r})$ for different t are the same regardless of the number of iterations. Likewise, we can omit $H'(\phi_{2D}$ from (2.4), and the relative magnitude ratios still remain the same. Therefore, only $2 \times N_{width} \times N_{height}$ addition and $N_{width} \times N_{height}$ multiplication operations are required for the computation of a single ϕ_{2D} function. To conclude, a total of $3 \times N_{width} \times N_{height} \times N_{frame}$ number of multiplication and addition operations are required.

The computational cost depends largely on the number of frames used in Δt. As explained in section 2, the proposed algorithm can obtain a reliable static background image with a relatively small number of frames. The number of frames can be further reduced if the frame-rate decreases, that is, if a sampled version of the frames are taken from the video sequence. In the experiments, we usually used $5 \sim 10$ frames. The fundamental steps of the proposed algorithm is presented below.

Principle Steps of the Algorithm

(1) At the initial step, an initial reference background image is constructed, e.g., by taking the average image of contiguous frames.
(2) Using the initial reference background image computed in step 1, The ϕ_{2D} functions for every frame in the time interval Δt are computed using (2.4).
(3) Compute the reference background image according to (2.5) using all the ϕ_{2D} functions in Δt.
(4) Update Δt such that all the frames in Δt are shifted by one frame along the sequential axis.
(5) Compute ϕ_{2D} functions for every frame in the time interval Δt using (2.4).
(6) Repeat step 3–5.

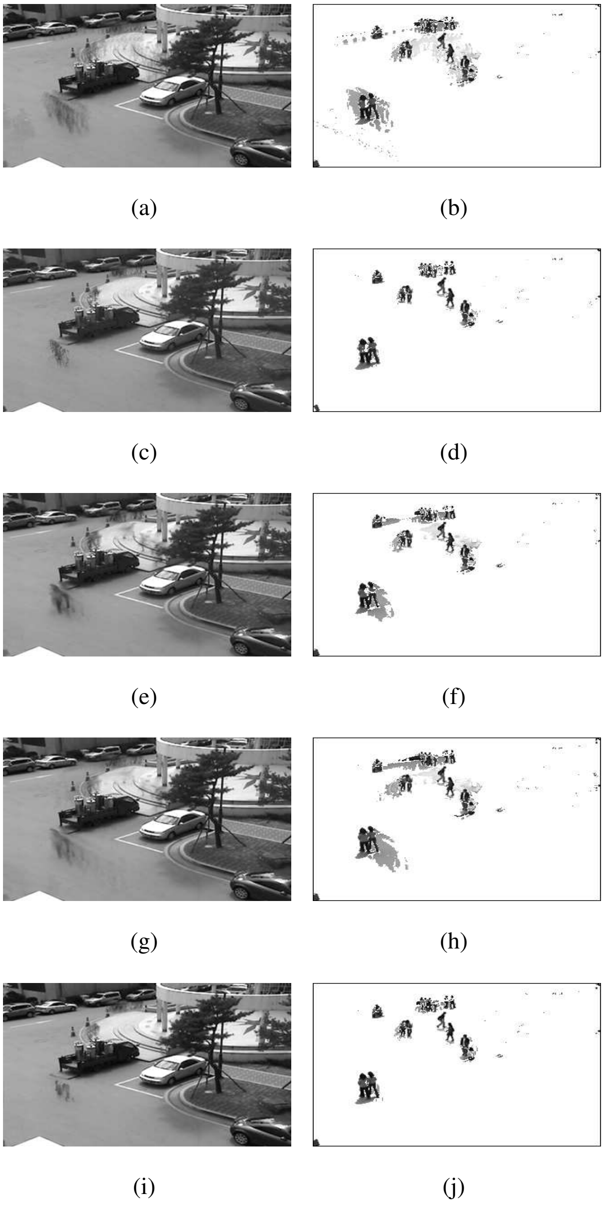
4. Experimental Results

Figure 1 shows the video sequence which we used in the experiments. All the frames in the sequence contain moving objects.

(a) (b) (c)

FIGURE 1. Shows the 60, 80 and 100 frame of the video sequence used in the experiments. The video sequence was obtained at a 30 fps frame rate.

Figure 2 compares the segmented foreground images and the modeled background images obtained by different methods. The simple averaging, median, and the proposed method all use 10 sampled frames, which are sampled every second frame, i.e., have a frame rate of 15 fps. For Gaussian modeling based methods, 10 frames are too few and have been excluded from the comparison. The foreground has been obtained by thresholding the difference image of the current frame and the background image with a threshold value of 30 with other methods, and with the proposed method by (2.4) with $\alpha = 30$. As can be observed from the first, the third, and the fourth row in Fig. 2, the averaging method and the blending method leave large "tails" of the moving objects in the background image, which again affects the foreground



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

FIGURE 2. Comparison of various background modeling methods. The first column shows the modeled reference background image and the second row shows the segmented foreground. First row: Background modeling by simple averaging. Second row: Background modeling by median method. Third and Fourth row: Background modeling by blending method, where in the third row the blending parameter is 0.1, and in the fourth row it is 0.05. Bottom row: Background modeling with proposed model.

segmentation. The median method shows better results, but the computational cost is large due to the inherent sorting process.

Table 1 compares the computational time of the median, the averaging, and the proposed method. The proposed method shows results as good as the median method and is faster. Furthermore, if the sequence includes noise, it can be removed with the proposed method due to the inherent averaging process.

TABLE 1. Comparison on computational cost

Algorithm	Computation time for one frame (ms)
Median	23.4
Averaging	1.2
Proposed	12.2

ACKNOWLEDGEMENT

This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD, Basic Research Promotion Fund) (KRF-2008-331-D00566).

REFERENCES

[1] J. Heikkila and O. Silven: A real-time system for monitoring of cyclists and pedestrians. Second IEEE Workshop on Visual Surveillance Fort Collins, Colorado (Jun. 1999) 74–81

[2] C. Wren, A. Azabayejani, T. Darrell and A. Pentland: Pfinder: Real-time tracking of the human body. IEEE Trans. on Pattern Analysis and Machine Intelligence. **19** (1997) 780–785

[3] G. Halevy and D.Weinshall: Motion of disturbances: detection and tracking of multibody non-rigid motion Machine Vision and Applications. **11** (1999) 122–137

[4] C. Stauffer and W.E.L. Grimson: Learning patterns of activity using real time tracking. IEEE Trans. on Pattern Analysis and Machine Intelligence. **22** (2000) 747–767

[5] X. Gao, T.E. Boult, F. Coetzee, and V. Ramesh: Error Analysis of Background Subtraction. In IEEE Int. Conf. on Computer Vision (2000)

[6] P. Kornprobst and R. Deriche: Image Sequence Analysis via Partial Differential Equations. Journal of Mathematical Imaging and Vision. **11** (1999) 5–26