Background subtraction based on phase feature and distance transform

Gengjian Xue, Jun Sun, Li Song

Institute of Image Communication and Information Processing, Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
Shanghai Key Laboratory of Digital Media Processing and Transmissions, Shanghai Jiao Tong University, Shanghai 200240, China

Abstract

A novel background subtraction method that can work under complex environments is presented in this paper. The proposed method consists of two stages: coarse foreground detection through the phase based background model we present, and foreground refinement using the distance transform. We first propose a phase feature which is suitable for background modeling. The background model is then built where each pixel is modeled as a group of adaptive phase features. Although the foreground detection result produced by the background model only contains some sparse pixels, the basic structure of the foreground has been captured as a whole. In the next stage, we adopt the distance transform to aggregate the pixels surrounding the foreground so that the final result is more clear and integrated. Our method can handle many complex situations including dynamic background and illumination.
variations, especially for sudden illumination change. Besides, it has no bootstrapping limitations, which means our method is without background initialization constraints. Experiments on real data sets and comparison with the existing techniques show that the proposed method is effective and robust.

*Keywords:* 
Background subtraction; Phase feature; Phase based background model; Distance transform;

### 1. Introduction

Background subtraction is often the first task in vision-based applications, such as security and surveillance. The output of background subtraction is usually an input to higher level processes, making it a critical part of the system. Background subtraction consists of two phases: building a statistical representation of the background scene, and detecting the foreground by "subtracting" the background from the scene. The performance of background subtraction depends mainly on the background modeling technique it uses. Natural environments make background subtraction a challenging task since they usually contain complex scenes including rippling water, waving trees, illumination variations, etc.

In the last decade, many kinds of approaches have been proposed for moving object detection. These techniques have used pixel intensity, texture or other effective information for background modeling. However, background modeling techniques utilizing phase information are rarely seen. Perhaps the
phase wrapping property and the narrow value range restrict its applications.

In this paper, we propose an efficient background subtraction technique based on a phase feature and the distance transform. Our method consists of two stages: modeling the phase based background for coarse foreground detection, and foreground refinement by using the distance transform. We choose the phase for background modeling because it has the property of being insensitive to illumination variations. In order to overcome the inherent limitations of phase wrapping and its narrow value range, a new phase feature which is suitable for background modeling is proposed. After the image patch is convolved with local Gabor filters, the phase feature is constructed by adding up the Gabor phases corresponding to the first largest amplitudes. Assuming the feature value of a particular pixel over time as a pixel process, we model its current value using a mixture of Gaussian distributions. In addition, the adaptive updating scheme ensures that the model has no bootstrapping limitations. The proposed model can detect the basic structure of the foreground but with a sparse representation, the distance transform is then applied to aggregate the pixels surrounding the foreground in order to get more integrated result. We will justify our method by experiments.

The rest of this paper is organized as follows: Section 2 provides a brief review of existing works. A new phase feature for background modeling is proposed in Section 3. In Section 4, our phase based background model is described in detail. The distance transform for foreground refinement is given in Section 5. Experimental results and evaluations are given in Section 6. Conclusions are finally drawn in Section 7.
2. Related work

One of the most common methods of background description is based on a Gaussian distribution. Wren et al. (1997) represented the intensity distribution of each background pixel with one Gaussian distribution. In order to describe more complicated scenes, a Gaussian mixture model (GMM) was proposed (Stauffer and Grimson, 1999). The model for each pixel intensity consisted of a few Gaussians, and an online K-means approximation technique instead of the exact EM algorithm was adopted for updating. The GMM technique was then modified by several researchers. For example, Zivkovic and van der Heijden (2006) extended the model by constantly selecting the appropriate number of Gaussian components for each pixel while updating the model parameters. Lee (2005) presented an adaptive learning rate calculated for each Gaussian at every frame to improve the model convergence speed.

Another popular technique is the nonparametric statistical approach. Elgammal et al. (2002) utilized a kernel density estimation (KDE) technique for background modeling, where the probability density function (PDF) of the pixel intensity was estimated directly from the data without any distribution assumptions. In Mittal and Paragios (2004), an estimation method with an adaptive kernel size for each data point was used. Based on the assumption that ergodicity in time often holds spatially, Jodoin et al. (2007) performed pixel kernel density estimation with only one background frame. This method had a lower memory requirement.
Some authors have proposed region-based techniques for background modeling. Heikkinä and Pietikäinen (2006) modeled the background using local binary pattern (LBP) histograms calculated over a circular region around the pixel. In Zhang et al. (2008), the LBP feature was computed considering both spatial and temporal information. Mason and Duric (2001) adopted edge and color histograms calculated over the block area as the features to describe the block. Since a region can capture more global information than a single pixel, region-based approaches are more robust under dynamic background scenes.

Several other effective models and methods have also been used for background subtraction. Zhong and Sclaroff (2003) cast the dynamic background region in time as an autoregressive moving process, and they used a robust Kalman filter to estimate the region intrinsic appearance. In Stenger et al. (2001), a Hidden Markov Model (HMM) with an online parameter estimation scheme was proposed to model the background. Patwardhan et al. (2008) proposed to detect the foreground using pixel layers. Inspired by the biological mechanisms of motion-based perceptual grouping, Mahadevan and Vasconcelos (2010) treated background subtraction as a saliency detection problem, and they proposed a spatiotemporal saliency algorithm that worked well with dynamic background scenes. Assuming background and foreground were two mutual independent signals, Tsai and Lai (2009) adopted the independent component analysis (ICA) technique to extract the foreground. Maddalena and Petrosino (2008) proposed a neural network architecture to model the background, but this technique needed more memory space. Cast-
ing background subtraction as a sparse error recovery problem, Dikmen and Huang (2008) presented a sparse representation framework for foreground detection, then they further discussed the different base selection methods (Dikmen et al., 2009).

3. New phase feature for background modeling

Phase contains a wealth of information, and its great importance has been introduced in detail by Oppenheim and Lim (1981). In recent years, phase as a feature has been successfully applied to several fields, such as palmprint identification (Zhang et al., 2003), face recognition (Zhang et al., 2007), etc. However, little work has utilized phase information for background modeling.

In this section, we propose a new phase feature for background modeling. The input image is first convolved with local Gabor filters so that each pixel has a group of features containing multiple amplitudes and corresponding phase values. For each pixel, we select the effective phase information according to the criteria that higher amplitude value in the feature group means more accurate local structure information has been captured, and its corresponding phase information is more representative. The new phase feature is then defined as the sum of the selected phase values.

Due to the properties of spatial localization, orientation selectivity, and spatial-frequency selectivity, Gabor filters have been widely used to extract pixel amplitude and phase information. A two-dimensional Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave. The Gabor
wavelets can be defined as follows (Zhang et al., 2007):

$$\psi_{\varphi,v}(z) = \frac{||k_{\varphi,v}||^2}{\sigma^2} e^{(-||k_{\varphi,v}||^2/2\sigma^2)} \left[ e^{ik_{\varphi,v}z} - e^{-\sigma^2/2} \right]$$  \hspace{1cm} (1)

where $\overrightarrow{k_{\varphi,v}} = \begin{pmatrix} k_{jx} \\ k_{jy} \end{pmatrix} = \begin{pmatrix} k_v \cos \phi_v \\ k_v \sin \phi_v \end{pmatrix}$, $k_v = f_{max}/2^{v/2}$, $\phi_v = \varphi(\pi/\varphi_{max})$, $v = 0, \ldots, v_{max} - 1, \varphi = 0, \ldots, \varphi_{max} - 1$, $v$ is the frequency and $\varphi$ is the orientation. $v_{max}$ and $\varphi_{max}$ represent the number of frequencies and orientations respectively. The first term in the square brackets in (1) determines the oscillatory part of the kernel, and the second term compensates for the DC value. $\sigma$ determines the ratio of the Gaussian window width to wavelength.

The Gabor transformation of a given image is defined as its convolution with the Gabor functions:

$$G_{\varphi,v}(z) = I(z) * \Psi_{\varphi,v}(z)$$  \hspace{1cm} (2)

where the symbol "\(*\)" represents the convolution operator, $z = (x, y)$ denotes the image position, and $G_{\varphi,v}(z)$ is the convolution result corresponding to the Gabor kernel at frequency $v$ and orientation $\varphi$. $G_{\varphi,v}(z)$ is a complex value which is composed of one amplitude item $A_{\varphi,v}(z)$ and one phase item $\theta_{\varphi,v}(z) \in [0, 2\pi)$. It can be written as:

$$G_{\varphi,v}(z) = A_{\varphi,v}(z) \cdot exp(i\theta_{\varphi,v}(z))$$  \hspace{1cm} (3)

Since the phase value varies quickly with image locations, applying the Gabor filter to the entire image would produce results too coarse to effectively represent the phase feature. In order to more precisely extract the phase information for each pixel, we divide the image into non-overlapping partitions.
and execute patch based convolutions. Based on our experience, the Gabor filters are designed with four frequencies and six orientations, which means $v_{\text{max}} = 4$, $\varphi_{\text{max}} = 6$. Figure 1 shows the amplitude responses of Gabor function with different frequencies. The frequency of Gabor wavelet is computed according to the formula: $k_v = f_{\text{max}}/2^{v/2}, v = 0, \ldots, v_{\text{max}} - 1$.

Figure 1: The amplitude responses of Gabor function with four different frequencies.

It can be seen that the frequency value is higher and the Gabor wavelength is shorter when $v$ has a lower value. While the value $v$ is increasing, the frequency value is decreasing and the Gabor wavelength is becoming longer. We select four different frequencies in our experiments because the wavelength of the central envelope is about eight pixels when $v$ equals to 3, which is capable of capturing image local structures; and more frequencies would contribute very little to our problem. The number of orientations is set to 6 to maintain the discriminability of the Gabor filters.

The patch size is determined by considering the Gabor wavelength. Larger patch size may result in a coarse information representation, while smaller patch size may not capture image local structures. Varying the patch size from 3 to 6, we observe the results so as to choose the appropriate value. Tak-
ing a public sequence which features a sudden illumination change (Toyama et al., 1999) as an example, Figure 2 shows the convolution results at a randomly selected point from the 1865th frame using different patch sizes.

Figure 2: Convolution results at a randomly selected point using different patch sizes. (a)~(d) represent the results using the patch size from 3 to 6 respectively. Top row: the amplitude responses. Bottom row: the corresponding phase responses.

It can seen that the phase values at different frequencies and orientations are almost the same when the patch size equals to $3 \times 3$, which may result in the phase feature to be less discriminative. If the patch size is chosen to $6 \times 6$, the difference between the amplitudes is not significant but their corresponding phase values change rapidly, so the phases to be selected according to the amplitudes may not be very stable in this case. In our method, we set the appropriate patch size to $4 \times 4$ by experience.

To select the number of effective Gabor phases for each pixel, we analyze its $4 \times 6$ convolution results. Considering the amplitude response as an energy measure, larger amplitude means higher energy and more accurate
local structure information has been captured by the filter at this frequency and orientation, therefore the phase value corresponding to this amplitude is more representative and discriminative. Based on this criterion, we propose a novel phase feature, which is constructed by selecting some representative Gabor phases and adding them together. The new feature is defined as:

\[ \Phi(z) = \sum_{l=1}^{L} \theta_l(z) \]  \hspace{1cm} (4)

where \( z \) denotes the image position, \( \Phi(z) \) denotes the feature value at position \( z \), \( \theta_l(z) \) is the Gabor phase value satisfying the condition that its corresponding amplitude is the \( l_{th} \) largest, and \( L \) denotes the number of Gabor phases to be selected.

![Amplitude and Phase Value Distributions](image)

Figure 3: Convolution results at two random points used for the analysis of selecting the number of effective Gabor phases.

We also give the public sequence above as an example. Figure 3 shows the convolution results at two randomly selected points from the 1865th frame. Theoretically, the index value can range from 1 to 24. If we just select the
Gabor phase corresponding to the largest amplitude, the new phase feature is not very stable and its value range is too narrow, which is not suitable for background modeling. If 10 or more Gabor phases are selected, although the new feature is stable, it may be less discriminative. Considering the discriminability and the stability of the new feature, we select five Gabor phases, which means $L = 5$.

Our phase feature is similar to phase congruency, but in fact they are different. Phase congruency (Kovesi, 1999) as a measure of phase has been widely used for edge detection, image fusion, etc. However, this measure is sensitive to noise. Besides, its value only ranges from 0 to 1, which is too narrow for background modeling and updating.

To demonstrate the effectiveness of the new feature, we select two points from a sequence which features light variation. From about the 213th frame in this sequence, the light begins to change and the foreground starts to appear. One of the selected points represents the background, and the other one is a foreground point. The value variations of these features are shown in Figure 4. It shows that the new feature is not only discriminating for foreground from background, but also immune to the variations of ambient light.

The proposed phase feature has some important properties for background modeling. First, it is insensitive to illumination changes; this is an inherent property of phase information. Next, the feature is relatively stable. We can see that although some noise exists in real settings, the feature value changes very little for the background point. In addition, when com-
pared to phase congruency, the wide value range makes it more suitable for background modeling. Finally, the feature is discriminative. When the true foreground really appears, its value changes rapidly and drastically.

4. Phase based background model

Based on the new feature, we propose a novel approach for background modeling. The procedure is identical for each pixel in this approach. We consider the phase feature of a particular pixel over time as a pixel process. The probability of observing the current phase feature is calculated as:

$$P(\Phi) = \sum_{i=1}^{K} \omega_i \mathcal{N}(\Phi_i, \mu_i, \sigma_i)$$

where $K$ is the number of Gaussians, $\mathcal{N}(.)$ is a Gaussian distribution, $\mu_i$ and $\sigma_i$ stand for the mean value and the variance of $i_{th}$ Gaussian respectively, and $\omega_i$
is the weight of $i_{th}$ Gaussian in the mixture and $\sum_{1}^{K} \omega_i = 1$, $\Phi$ represents the value of the feature.

For every new pixel, we compute the feature $\Phi$ and compare it with the current set of $K$ models to check whether a matching model exists. By analyzing the property of the new feature, we find that the feature has a narrow value distribution and changes rapidly compared to the gray value features. Consequently, the matching condition is defined as the feature value within the standard deviation of the distribution.

Due to the phase wrapping property, there exists a singular region near the lowest and the highest values. As shown in Figure 5, the singular region is near 0 and $2\pi$ for a given Gabor phase; and for the new feature, the region is near 0 and $2L\pi$.

![Figure 5: Feature wrapping demonstration.](image)

When the new feature and the mean value of the background model both fall into this area, the distance between them is computed as $2L\pi - Dist$, where $Dist$ is the absolute value between two values. The complete matching conditions are defined as follows:
where $\epsilon$ denotes the range of the singular region. If such models exist, we select the first matching one; then its associated parameters are updated in three different ways:

if $\mu_i < \epsilon$ and $2L\pi - \Phi < \epsilon$

\[
\begin{align*}
\mu_i &= \mu_i + \alpha_2(\Phi - \mu_i - 2L\pi) \\
\sigma_i^2 &= (1 - \alpha_2)\sigma_i^2 + \alpha_2(\Phi - \mu_i - 2L\pi)^2
\end{align*}
\]  

if $\Phi < \epsilon$ and $2L\pi - \mu_i < \epsilon$

\[
\begin{align*}
\mu_i &= \mu_i + \alpha_2(\Phi - \mu_i + 2L\pi) \\
\sigma_i^2 &= (1 - \alpha_2)\sigma_i^2 + \alpha_2(\Phi - \mu_i + 2L\pi)^2
\end{align*}
\]  

others:

\[
\begin{align*}
\mu_i &= \mu_i + \alpha_2(\Phi - \mu_i) \\
\sigma_i^2 &= (1 - \alpha_2)\sigma_i^2 + \alpha_2(\Phi - \mu_i)^2
\end{align*}
\]  

the model weights are updated as:

\[
\omega_i = (1 - \alpha_1)\omega_i + \alpha_1 M_k
\]  

where $\alpha_1 \in (0, 1)$ is the first learning rate and $\alpha_2 = \alpha_1/\omega_i$ is the second learning rate, $M_k$ is 1 for the first matching Gaussian and 0 for the others. When the value $\mu_i$ is beyond the scope $[0, 2L\pi)$, we have:

\[
\begin{align*}
\mu_i &= \mu_i - 2L\pi \quad \text{if } \mu_i \geq 2L\pi \\
\mu_i &= \mu_i + 2L\pi \quad \text{if } \mu_i < 0
\end{align*}
\]
If no model satisfies the matching conditions, the Gaussian with the lowest \( \omega \) is replaced by a new Gaussian with \( u = \Phi, \sigma = \sigma_{\text{init}} \) and an initial weight.

In the foreground detection stage, the Gaussians are sorted in descending order according to the values \( \omega/\sigma \). The first \( B \) distributions are chosen as the background distributions,

\[
B = \arg\min (\sum_{i=1}^{b} \omega_i > T)
\]  

(12)

where \( T \) is a threshold value representing the minimum prior probability that the background should be in the scene. If the new feature matches at least one of the \( B \) distributions, the corresponding pixel is classified as background, otherwise the pixel is marked as foreground.

In our method, finding proper values for this model depends mainly on experiments. The parameter \( K \) relates to the scene being modeled, \( K \in (3 \sim 5) \) is a good value. \( T \in (0.7 \sim 0.8) \) is an experimented value in most cases. Higher \( T \) means a higher probability that the data should be accounted as background. We adopt \( K = 3, T = 0.8 \) in this paper. Considering the narrow value distribution of the features, we set \( \sigma_{\text{init}} = 3 \).

One of the detection results produced by our model is shown in Figure 6(a). It can be seen that the foreground pixels are sparse but the basic structure of the foreground has been captured, which demonstrates that the proposed model is valid for foreground detection. In the next section, we will adopt the distance transform for blobs aggregation to get more integrated result.
5. Blob aggregation using distance transform

The distance transform, which was first introduced by Rosenfeld and Pfaltz, is a general operation constituting the basis of many methods in computer vision and geometry. It converts a binary image of black and white pixels into a distance map. The distance map is a representation where each pixel has a value indicating its distance to the nearest white pixel. If the pixel value is 1, its distance value is 0. We represent a $M \times M$ binary image by $A = \{(i, j) : a(i, j) = 0 \text{ or } 1\}$, for $i, j = 1, \ldots, M$, and $W = \{(x, y) : w(x, y) = 1\}$ represents the coordinates of white pixels in the image. The Euclidean distance transform of pixel $a(i, j)$ is defined as:

$$d(i, j) = \min_{(x, y) \in W} \sqrt{(i - x)^2 + (j - y)^2}$$

(13)

Since most white pixels concentrate on the foreground regions in our binary detection result, the pixels having lower values are more compact in the distance image and they basically correspond to the foreground. We segment the distance image with a threshold value in order to separate these regions from the whole. Choosing a proper value for the threshold should consider preserving the foreground structure while suppressing the noise as much as possible, a good value is set by experience to be between 1.2 and 1.5. After using morphological operations for noise removal, we obtain the final result.

The blob aggregation mainly includes these steps: (a) Before the distance transform, we execute a dilation operation, using a disk-shaped structuring element with an adaptive radius, on the initial detection result. This opera-
tion is helpful to the subsequent processing. Considering connecting nearby pixels while reducing shape distortion as much as possible, we set the radius value of each frame according to the ratio value, which is represented by the number of the initial foreground detected pixels in the image to the total number of the image pixels. The radius equals to 6 when the ratio value is lower than 3%; and it equals to 5, 4, 3, and 2, corresponding to the ratio value is in the range of 3%-6%, 6%-9%, 9%-12%, and 12%-21%, respectively; otherwise, it equals to 1. (b) The distance transform is applied to the image, and we segment the distance map to get the binary result. The threshold value is set to 1.2 in our experiments. (c) Since the segment result may contain scattered small areas and points, we remove them in order to get a cleaner result. The connected areas to be removed are those having fewer than 200 pixels. (d) We fill the holes of the binary image so that the foreground mask is more complete. It should be noted that for getting more closed boundaries, the same dilation operation mentioned above is executed before filling the holes. (e) After executing an erosion operation using a disk-shaped structuring element with the doubled adaptive radius to thin the detected mask, we get the final result. An example of the results before and after the blob aggregation is shown in Figure 6.

We emphasize that blob aggregation using the distance transform differs from traditional post processing techniques. Since the foreground pixels detected by our model are sparse, the blobs would be fragmentary without the aggregation; therefore the blob aggregation is an integral part of our method. In contrast, the more traditional post processing techniques such
as noise removal, boundary smoothing, etc. are supplementary operations in most cases, and the result being processed by these techniques is basically complete. Based on the above analysis, we provide an overview of our method in Figure 7.

6. Experiments

6.1. Tests on various environments

To validate the effectiveness of the proposed method, some video sequences representing complex scenes are adopted for testing. Both visual and numerical methods are used for performance evaluation. In addition, we also compare our method with three widely used techniques: GMM, KDE, and LBP methods. For fair comparison, the operation for blobs aggregation are applied to all methods, and their parameters remain unchanged in all experiments. The experiments on illumination variation are an exception. Since for the three widely used techniques, these operations have produced very bad results, we adopt a noise removal operation instead. In the following
construct the Gabor filter bank, set the image patch size.
initialize the phase based background model

For each new frame

do the patch based Gabor filter convolution

For(each pixel in the frame)

cmpare its phase feature \( \phi \) with existing models

| If (matched model is found)
| update the matched model parameters
| Else
| the model with lowest weight is replaced
| End

sort the background models according to \( \alpha/\sigma \)
select the first \( B \) distributions as background model

| If (feature \( \phi \) fits at least one background model)
| \( \text{Cur\_Pixel} = \text{background} \)
| Else
| \( \text{Cur\_Pixel} = \text{foreground} \)
| End

End

blobs aggregation using distance transform and
morphological operations

End

Figure 7: Overview of the proposed framework
Moving object detection in dynamic scenes is a difficult problem. Two
typical sequences named *wavering curtains* and *rippling water* are selected
for this test, both drawn from Li et al. (2004). In the *wavering curtains*
sequence, the curtains flutters in the wind when a person enters the office and
speaks. The *rippling water* sequence features a moving background provided
by ripples in the water. Figure 8 and Figure 9 show some of the detection results on these sequences. It can be seen that before the blob aggregation is used, the GMM and KDE methods have identified some background areas. In contrast, the LBP and our methods can suppress the background noise, but they provide a coarse localization. After the distance transform and post processing operations, the four methods produce comparable results.

![Figure 9: Experimental results on the rippling water sequence. (a) Input frame. (b) Sample of test frames. (c) Ground truth. (d) GMM method. (e) KDE method. (f) LBP method. (g) Our method. Top row of (d)~(g): The initial detection results. Bottom row of (d)~(g): The corresponding final results.](image)

The second test case is about the bootstrapping problem. Currently,
Figure 10: Experimental results on the restaurant sequence. (a) Input frame. (b) Sample of test frames. (c) Ground truth. (d) GMM method. (e) KDE method. (f) LBP method. (g) Our method. Top row of (d)–(g): The initial detection results. Bottom row of (d)–(g): The corresponding final results.
many background subtraction approaches can work provided that there exist some clean background frames during the training period. However, clean background frames for training may not be always possible in real settings. We test our method on two typical sequences which lack clean background frames. The first sequence, named restaurant, is from Toyama et al. (1999), and the second sequence, named shopping center, is from Li et al. (2004). Figure 10 and Figure 11 show some of the detection results on the two sequences respectively. We consider that our method produces comparable results to the other techniques.

Illumination variation often challenges background subtraction significantly. To test the robustness of our method under illumination variation, two sequences characterized by rapid illumination change are selected. We choose them because foreground detection under sudden illumination change is one of the most difficult problems in background subtraction, which often has to utilize higher level processing. The first sequence, named light switch on, is from Toyama et al. (1999), and it is usually used as a benchmark for fast illumination variation. In this sequence, the room starts with the light off. After a while, a person enters the room, turns on the light, and moves the chair. The second sequence, named local light switch, is from Cristani et al. (2010). The sequence shows an indoor scenario, where a dark corridor is portrayed. A person moves between two rooms, opening and closing the related doors. The light in the rooms is on, so the illumination spreads out over the corridor, locally changing the visual layout. For both sequences, we select test frames shortly after the light changes so that many adapta-
tion mechanisms have inadequate time to adapt. In these experiments, the initial detection results produced by the three traditional methods contain many background pixels, and the blob aggregation would yield very bad results. Thus for these methods, we adopt a noise removal operation as the post processing technique instead. Figure 12 and Figure 13 show some of the detection results produced by the four different methods. We find that the GMM, KDE, and LBP methods can not adapt to the sudden illumination change and they produce incorrect results. In contrast, our method yields much better results.

To evaluate these methods systematically, quantitative evaluations were executed on all the above datasets. The measures Precision and Recall were adopted as the metrics,

\[
\text{Precision} = \frac{\text{Number of true positives detected}}{\text{Total number of positives detected}} \quad (14)
\]

and

\[
\text{Recall} = \frac{\text{Number of true positives detected}}{\text{Total number of true positives}} \quad (15)
\]

For each sequence, we randomly selected 10 samples and computed their averaged precision and recall rates as the final evaluation of the sequence. The ground truth images were all marked manually in the experiments. The evaluation results for all the datasets are shown in Table 1, which reveals that our method has a comparable performance to the three widely used techniques under dynamic background and bootstrapping situations. For the illumination variation scene, our method is greatly superior to the others.

Since the GMM and KDE techniques are both pixel-based techniques,
Figure 11: Experimental results on the shopping center sequence. (a) Input frame. (b) Sample of test frames. (c) Ground truth. (d) GMM method. (e) KDE method. (f) LBP method. (g) Our method. Top row of (d)–(g): The initial detection results. Bottom row of (d)–(g): The corresponding final results.
Figure 12: Experimental results on the *light switch on* sequence. (a) Input frame. (b) Sample of test frames. (c) Ground truth. (d) GMM method. (e) KDE method. (f) LBP method. (g) Our method. Top row of (d)~(g): The initial detection results. Bottom row of (d)~(g): The corresponding final results.
Figure 13: Experimental results on the local light switch sequence. (a) Input frame. (b) Sample of test frames. (c) Ground truth. (d) GMM method. (e) KDE method. (f) LBP method. (g) Our method. Top row of (d)\textasciitilde(g): The initial detection results. Bottom row of (d)\textasciitilde(g): The corresponding final results.
Table 1: Quantitative evaluation using the precision and recall measures

<table>
<thead>
<tr>
<th>Sequence</th>
<th>GMM</th>
<th>KDE</th>
<th>LBP</th>
<th>Our Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>wavering curtains</td>
<td>Precision</td>
<td>85.29</td>
<td>74.23</td>
<td>70.68</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>83.24</td>
<td>94.84</td>
<td>87.50</td>
</tr>
<tr>
<td>rippling water</td>
<td>Precision</td>
<td>80.41</td>
<td>81.11</td>
<td>63.38</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>76.02</td>
<td>76.05</td>
<td>88.12</td>
</tr>
<tr>
<td>restaurant</td>
<td>Precision</td>
<td>62.82</td>
<td>55.50</td>
<td>35.82</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>56.63</td>
<td>69.18</td>
<td>68.12</td>
</tr>
<tr>
<td>shopping center</td>
<td>Precision</td>
<td>66.55</td>
<td>63.55</td>
<td>46.39</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>66.21</td>
<td>77.91</td>
<td>76.31</td>
</tr>
<tr>
<td>light switch on</td>
<td>Precision</td>
<td>11.88</td>
<td>9.82</td>
<td>13.02</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>75.82</td>
<td>63.70</td>
<td>78.04</td>
</tr>
<tr>
<td>local light switch</td>
<td>Precision</td>
<td>6.71</td>
<td>14.95</td>
<td>12.34</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>74.30</td>
<td>66.02</td>
<td>80.39</td>
</tr>
</tbody>
</table>
they can provide a precise localization but are highly sensitive to noise. On the dynamic background scene, they can produce fine foreground detection results, but meanwhile many moving background pixels are misclassified. After the operation of blobs aggregation, the results become much better. The LBP method uses texture features for background modeling, which can be considered as a region-based technique. This method is more robust to noise, and it can be seen that the detection results for dynamic background are almost without background noise. However, region-based methods often produce coarse results. With regard to our method, we assume the phase feature is independent while modeling the background, so our background model is sensitive to noise.

In addition, due to the fact that the feature distribution is being approximated by the mixture of Gaussians, the detection results contain sparse pixel points. However, as the basic structure of the foreground can be captured by our model, we then adopt the distance transform for blobs aggregation.

Our method and the other three techniques all have adaptive learning mechanisms, consequently they have no bootstrapping limitations. On the illumination variation scene, the GMM and KDE methods both adopt pixel intensity as features; therefore they are sensitive to illumination changes. Although the LBP texture has certain robustness to illumination variations, the LBP method cannot handle the rapid global illumination change. In contrast, our method is effective in solving this challenging problem; the reason is that the proposed phase feature has the inherent property of being insensitive to illumination changes.
To verify the applicability of our method, we further test our method under natural circumstances. Some of the sequences taken from the PETS database are adopted for this experiment. Figure 14 shows some of the detection results, which demonstrates that our method has a good generality.

![Figure 14: The detection results of the 318th, 640th, 871th and 1830th frames in an outdoor sequence taken from the PETS database.](image)

6.2. Tests on different parameters

In the previous sections, we have carried on the analysis of parameter determination for our method. In this subsection, we will test its performance when the values of some key parameters are varied. Two sequences named light switch on and shopping center are used for our test, and the evaluation technique remains the same as above. In every experiment, we only alter the value of one parameter while keeping the others unchanged.

First, we consider the changing patch size, varying from $3 \times 3$ to $6 \times 6$. The detection results on the two sequences are shown in Figure 15 and Figure 16. It can be seen that the initial detection results all contain more noise.
when the patch size is $3 \times 3$ or $6 \times 6$, and the initial detection results are the best when the patch size equals to $4 \times 4$. After the blob aggregation, we find that the final results across different patch sizes are comparable. The quantitative evaluation results shown in Table 2 further illustrate that our method is not very sensitive to patch size variation. In general, a patch size equal to $4 \times 4$ or $5 \times 5$ is a good choice.

![Image](image_url)

Figure 15: Experimental results on the light switch on sequence using different patch sizes. (a) $3 \times 3$ patch size. (b) $4 \times 4$ patch size. (c) $5 \times 5$ patch size. (d) $6 \times 6$ patch size. Top row of (a)~(d): The initial detection results. Bottom row of (a)~(d): The corresponding final results.

The second parameter considered is the number of Gabor phases to be selected for background modeling. Since if the number of selected Gabor phases is varied, the value range of the new phase feature consequently alters, we change the parameter value $\sigma_{init}$ correspondingly. The test results on the two sequences are shown in Figure 17 and Figure 18. It can be seen that if we just select one Gabor phase for background modeling, the initial detection
Figure 16: Experimental results on the shopping center sequence using different patch sizes. (a) $3 \times 3$ patch size. (b) $4 \times 4$ patch size. (c) $5 \times 5$ patch size. (d) $6 \times 6$ patch size. Top row of (a)–(d): The initial detection results. Bottom row of (a)–(d): The corresponding final results.

Table 2: Quantitative evaluation on different patch sizes

<table>
<thead>
<tr>
<th>Sequence</th>
<th>$3 \times 3$</th>
<th>$4 \times 4$</th>
<th>$5 \times 5$</th>
<th>$6 \times 6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>light switch on</td>
<td>Precision</td>
<td>70.97</td>
<td>66.16</td>
<td>55.59</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>95.64</td>
<td>96.77</td>
<td>97.83</td>
</tr>
<tr>
<td>shopping center</td>
<td>Precision</td>
<td>85.72</td>
<td>77.53</td>
<td>72.29</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>33.00</td>
<td>40.36</td>
<td>54.42</td>
</tr>
</tbody>
</table>
results are noisy which may not be suitable for blobs aggregation. Meanwhile, we find the background noise and the detected true object pixels are both decreasing while the selected number is increasing. If the initial detected pixels are too sparse, the final results may produce a partial representation of the foreground objects. Analyzing the numerical evaluation results shown in Table 3, we consider that the optimal selected number should be between 5~10, which verifies that our choice to extract 5 phases is reasonable.

![Figure 17: Experimental results on the light switch on sequence using different number of Gabor phases for background modeling. (a) 1 phase. (b) 5 phases. (c) 10 phases. (d) 15 phases. Top row of (a)~(d): The initial detection results. Bottom row of (a)~(d): The corresponding final results.](image)

6.3. Discussions

The experimental results shown above demonstrate that our method can work well in many complex environments. However, we notice that in some cases such as bootstrapping, the results are not always the best. There may
Figure 18: Experimental results on the *shopping center* sequence using different number of Gabor phases for background modeling. (a) 1 phase. (b) 5 phases. (c) 10 phases. (d) 15 phases. Top row of (a)~(d): The initial detection results. Bottom row of (a)~(d): The corresponding final results.

Table 3: Quantitative evaluation on different number of Gabor phases selected for background modeling

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Precision 1 phase</th>
<th>Precision 5 phases</th>
<th>Precision 10 phases</th>
<th>Precision 15 phases</th>
</tr>
</thead>
<tbody>
<tr>
<td>light switch on</td>
<td>59.79</td>
<td>66.16</td>
<td>78.19</td>
<td>83.71</td>
</tr>
<tr>
<td></td>
<td>96.40</td>
<td>96.77</td>
<td>91.49</td>
<td>86.40</td>
</tr>
<tr>
<td>shopping center</td>
<td>76.51</td>
<td>77.53</td>
<td>82.07</td>
<td>81.13</td>
</tr>
<tr>
<td></td>
<td>29.46</td>
<td>40.36</td>
<td>24.58</td>
<td>16.94</td>
</tr>
</tbody>
</table>
be two reasons for this problem. The first one is that in our morphological operations, we consider small areas after the distance transform as noise to be removed. However, in some applications, where the true foreground is relatively small, it may sometimes be erroneously removed. The second reason is because we use both the dilation and the erosion operations in post processing, which may cause some shape distortion, especially for small objects. For example, the foreground objects are relatively small in the shopping center and restaurant test sequences; as a result, they are either distorted or removed entirely from the foreground image during post processing.

Regarding the complexity of our method, we adopt time complexity as a measure and compare our technique with the other methods. Since the GMM and KDE methods model the background directly using pixel intensities, they have the lowest complexity. The LBP method has to compute the LBP histograms for each pixel, hence its complexity is higher than the previous two. As our method has to do the patch based convolutions and blobs aggregation, it has the highest complexity. We implement all these algorithms using Matlab7.1 on a PC computer with 2.4GHz Intel CPU, 2G RAM. Processing a 128 × 160 frame, the GMM technique takes about 0.48s, the KDE technique needs approximately 0.13s, the LBP method uses about 7.10s, and our method needs about 25.46s.

7. Conclusions

In this paper, a novel background subtraction method has been presented. We first proposed a new phase feature for background modeling, then we
discussed the background modeling and updating schemes in detail. As the results produced by the background model contained sparse pixels in addition to the basic structure of the foreground, we used the distance transform to determine which pixels to be aggregated into the foreground.

Our method works with a high degree of accuracy and precision in practical applications, such as natural conditions and other complex environments. Compared to the other approaches mentioned above, our method can produce comparable results on dynamic background and bootstrapping scenes. On the illumination variation scene, our method significantly outperforms the others; this is because the phase feature is inherently insensitive to illumination changes. Unlike many other techniques, we deal with the illumination variation problem at the pixel level.

The blob aggregation using the distance transform is an integral part of our method, although it may cause some shape distortion. In computer vision applications, the goal of background subtraction is to provide blob detection which provides an input to higher level processes, such as object recognition and tracking. In our method, the detected blobs effectively represent the foreground shapes and regions, hence it can be considered that our method has achieved the goal of foreground detection. Although other better techniques for blobs aggregation may exist, the distance transform has the advantage of having a low computational cost, which is also an important factor for performance evaluation.
Acknowledgements

This work was supported in part by NSFC (60702044, 60902073, 61102098), STCSM(10511500802), and 973 program (2010CB731401, 2010CB731406).


