Efficient projection pursuit density estimation for background subtraction

T. V. Pham
Arnold W. M. Smeulders

Intelligent Systems Lab Amsterdam (ISLA) - University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam, the Netherlands
{vietp, smeulders}@science.uva.nl

Abstract
This paper presents a method for background subtraction in surveillance videos captured from static cameras. In contrast to previous methods using pixel-based features, we propose to use local image descriptor to better eliminate unwanted imaging conditions. To facilitate the shift from low-dimensional feature spaces to a higher dimensional space, we employ the projection pursuit density estimation technique. Like kernel density estimation, this technique does not assume any parametric model of the unknown distribution. The advantage in comparison to kernel density estimation is that it is much faster in the run-time. We also derive a fixed point method to improve the traditional implementation of this technique computationally. Our experiment shows the improvement gained by the combination of complex features and projection pursuit density estimation.

1. Introduction
Background subtraction is an important task in most automatic video surveillance systems [6]. This is a difficult problem because of the continuously varying imaging conditions. We mention two common examples: lighting condition due to the sun and cloud, and the weather factors such as wind and rain. To handle these sources of variation, it is necessary to incorporate both the prior knowledge about the problem as well as statistical modeling from a large training set into a robust system.

Probability density estimation has been the basic tool in recent advances in background modeling [12, 4, 10]. Let \( x \in \mathbb{R}^d \) denote a random variable describing the image variation at a particular location \( I_t \). Furthermore, let \( \{x_1, x_2, \ldots, x_N\} \) denote the training examples obtained from multiple frames (of the background scene) at location \( I_t \). The goal is to estimate the unknown probability density \( P(x) \) from the training examples. The background subtraction procedure is merely applying the density estimate of a new point \( x_0 \)

\[
P(x_0) > t \quad (1)
\]

where \( t \) is a threshold value.

Most previous approaches have employed a pixel based representation [12, 4], namely \( x = [R, G, B]^T \) where \( R, G \) and \( B \) denote the three color channels, or in some normalized color space to achieve certain degree of invariance to shadow. However, this representation is rather sensitive to most imaging conditions.

In this paper we propose the use of complex descriptor for background modeling. In particular, we employ a variant of the SIFT descriptor introduced in [2] for human detection. This type of descriptor has been shown to be effective for generic object matching across various imaging conditions [8].

The use of complex features, however, introduces a challenge in statistical modeling because features are no longer low dimensional. In particular, the descriptor employed here results in a 36-dimensional vector, \( x \in \mathbb{R}^{36} \). This rules out existing statistical methods for background modeling. For instance, let us consider the specific case of kernel density estimation [4] with the Gaussian kernel

\[
p(x) = \frac{1}{Z} \sum_{i=1}^{N} \prod_{j=1}^{d} e^{-\frac{(x_{(j)} - z_{(j)})^2}{2\sigma_j^2}} \quad (2)
\]

where \( Z \) is a normalizing constant, \( z_{(j)} \) denotes the \( j \)th component of the vector \( z \), and \( \sigma_j \) is the scale parameters. This method has been popular in background modeling because it does not assume any particular model and in theory can approximate any probability density distribution. Here the run-time complexity is \( O(Nd) \) which is significant as we have to evaluate at all point in a dense image grid. Fast evaluation method in [4] is effective for low dimension only.

We employ another non-parametric method called projection pursuit density estimation [5]. Here, the density \( p(x) \) is approximated by a product density

\[
p(x) = p_0(x) \prod_{m=1}^{M} f_m(\alpha_m^T x) \quad (3)
\]

where \( \|\alpha_m\| = 1 \). \( p_0(x) \) is some prior probability density, \( f_m(\alpha_m^T x) \) are one-dimensional augmenting functions, and
$M$ is the number of augmenting functions. Like kernel density estimation, this method does not assume any particular parametric model, and can approximate arbitrary density as $M \to \infty$. The result in [7] shows that projection pursuit density estimation compares favorably to kernel density estimation. Also, in contrast to (2), (3) can be evaluated very efficiently by $M$ table look-ups.

The implementation of projection pursuit density estimation is rather cumbersome [13]. One approach is to assume a parametric model for each one-dimensional density [13]. This, however, restricts the flexibility of the method. Here we will provide a new, efficient fixed point optimization based on one-dimensional kernel density estimation.

The paper is organized as follows. In Section 2 we present the complex features. Projection pursuit density estimation is reviewed in Section 3, and our implementation is given in Section 4. The experiment is presented in Section 7. Finally, Section 8 concludes the paper.

2. Complex features

This section describes the histogram of oriented gradients in [2] for human detection. It is derived from the SIFT descriptors which has been shown to perform well for image matching [8].

The image (frame) is divided into a grid of overlapping cells and blocks. The advantage of using features derived from an image block is that one can perform local normalization which is essential for good performance. In this paper, we use a block of $2 \times 2$ cells, and cell of $8 \times 8$ pixels (see Figure 1).

Each descriptor is computed as follows. First, the gradient vector is computed at each pixel. For color images, the gradients are computed separately for each channel, and the one with maximal norm is selected as the pixel’s gradient vector. The gradients are computed with a simple $[-1 \ 0 \ 1]$ kernel.

In the next step, the gradient vectors within each $8 \times 8$ cell are accumulated in a histogram of orientation. Here, we used nine orientations, resulting in a 9-vector $u$ for each cell.

Finally, the descriptor $v$ at each location is formed by concatenate the vectors in each $2 \times 2$ block, $v = [u_1, u_2, u_3, u_4]$ and normalized

$$v = \frac{v}{||v||} \quad (4)$$

In short, each descriptor is a vector with 36 components. Thus, in the current setting $d = 36$. The reader is referred to [2] for a discussion on the various parameter values used here.

Figure 1: Histogram of oriented gradients [2]. The descriptors are computed on a dense grid rather than at separate interest points. The gradients are accumulated in each $8 \times 8$ pixel cell, and then normalized within a $2 \times 2$ cell block.

3. Projection pursuit density estimation

From (3) projection pursuit density estimation (PPDE) approximates the multivariate density $p(x)$ by an initial density $p_0(x)$, and multiplied by a product of univariate function $f_m(\alpha^T_m x)$ where $\alpha^T_m x$ is the projection of the data point $x$ onto the direction $\alpha_m$. The approximation is incremental. At each round, PPDE finds the direction $\alpha_m$ and construct the function $f_m(\cdot)$.

Let $p_{m-1}(x)$ be the current approximation to the unknown $p(x)$. Then

$$p_m(x) = p_{m-1}(x)f_m(\alpha^T_m x) \quad (5)$$

The goal is to get closer to $p(x)$ in each step, measured by the empirical cross-entropy

$$W = \frac{1}{N} \sum_{n=1}^{N} \log p_m(x_n) \quad (6)$$

It is shown that given a direction $\alpha$ and known $p(x)$, the optimal $f_m(\cdot)$ is

$$f_m(\alpha^T x) = \frac{p^{(\alpha)}(\alpha^T x)}{p^{(\alpha-1)}(\alpha^T x)} \quad (7)$$
where \( p^{(\alpha)}(\cdot) \) and \( p^{(\alpha)}_{m-1}(\cdot) \) are respectively the data and current model marginal density along the one-dimensional subspace spanned by \( \alpha \).

Thus, the main problem is to find the best direction \( \alpha_m \) for the current round of approximation. This is achieved by optimizing an objective function

\[
\alpha_m = \arg \min_{\alpha} \frac{1}{N} \sum_{n=1}^{N} \log f_m(\alpha^T x_n) \tag{8}
\]

where \( f_m(\cdot) \) is an estimate for the ratio of data and model marginal along \( \alpha \).

The complete algorithm is straightforward

- Start with an initial estimation \( p_0(x) \), reflecting prior knowledge about the problem.
- For \( m = 1 \) to \( M \):
  1. Choose direction \( \alpha_m \) by solving (8).
  2. Update \( p_m(x) \) by estimating (7).

In fact in solving (8) it is certain that one has to estimate (7). In [5], (8) is optimized by numerical method, while the ratio estimate (7) is done by histogram estimation.

The estimation is involved with two univariate densities: \( p^{(\alpha)}(\alpha^T x) \) and \( p^{(\alpha)}_{m-1}(\alpha^T x) \). For a direction \( \alpha \) the former can be estimated simply by projecting the given data \( \{x_n\} \) on to \( \alpha \), and the employing histogram estimation on the one-dimensional line. For the latter, one resorts to the Monte Carlo method to sample \( \{s_\ell\} \) from the current estimate \( p_{m-1}(x) \) (this is possible because we can evaluate \( p_{m-1}(x) \)), and then using \( \{\alpha^T s_\ell\} \) for estimation as in the other case.

4. An efficient implementation

The most computationally expensive step is finding the optimal direction \( \alpha_m \). The use of histogram estimation as in [5] prevents us from employing continuous optimization. Hence, we propose using kernel density estimation for the one-dimensional density.

Recall that we have at our disposal two sets \( \{x_n\} \) and \( \{s_\ell\} \). Without loss of generality, assume that both sets are of size \( N \).

Employing kernel density estimation with the Gaussian kernel we have

\[
p^{(\alpha)}(\alpha^T x) = \frac{1}{N \sqrt{2\pi\sigma}} \sum_{n=1}^{N} e^{-\frac{(\alpha^T(x_n-x))^2}{2\sigma^2}} \tag{9}
\]

\[
p^{(\alpha)}_{m-1}(\alpha^T x) = \frac{1}{N \sqrt{2\pi\sigma}} \sum_{\ell=1}^{N} e^{-\frac{(\alpha^T(x_s-s))}{2\sigma^2}} \tag{10}
\]

where \( \sigma \) is the scale parameter.

From (8), (7), (9) and (10) we have the objective function \( J(\alpha) \) to find the optimal direction \( \alpha_m \) (without the constant factor)

\[
J(\alpha) = \sum_{k=1}^{N} \left\{ \log \sum_{n=1}^{N} e^{-\frac{(\alpha^T(x_k-x_n))^2}{2\sigma^2}} \right\}
\]

To minimize \( J(\alpha) \) subject to \( \alpha^T \alpha = 1 \), we formulate the Lagrange \( L(\alpha, \lambda) \)

\[
L(\alpha, \lambda) = \sum_{k=1}^{N} \left\{ \log \sum_{n=1}^{N} e^{-\frac{(\alpha^T(x_k-x_n))^2}{2\sigma^2}} \right\} - \frac{\lambda}{2} (\alpha^T \alpha - 1) \tag{12}
\]

We have

\[
\frac{\partial L(\alpha, \lambda)}{\partial \alpha} = A(\alpha) \alpha - B(\alpha) \alpha - \lambda \alpha \tag{13}
\]

where

\[
A(\alpha) = \sum_{k,n} e^{-\frac{(\alpha^T(x_k-x_n))^2}{2\sigma^2}} (x_k - x_n)(x_k - x_n)^T \sum_{i=1}^{N} e^{-\frac{(\alpha^T(x_k-x_i))^2}{2\sigma^2}} \tag{14}
\]

\[
B(\alpha) = \sum_{k,\ell} e^{-\frac{(\alpha^T(x_k-x_\ell))^2}{2\sigma^2}} (x_k - s_\ell)(x_k - s_\ell)^T \sum_{i=1}^{N} e^{-\frac{(\alpha^T(x_k-x_i))^2}{2\sigma^2}} \tag{15}
\]

By setting (13) to zero, we have

\[
\lambda \alpha = A(\alpha) \alpha - B(\alpha) \alpha \tag{16}
\]

Furthermore, at the optimal solution \( \alpha^T \alpha = 1 \). From this we design an iterative procedure

\[
\lambda^{(t)} = \alpha^{(t)T} \left( A(\alpha^{(t)}) - B(\alpha^{(t)}) \right) \alpha^{(t)} \tag{17}
\]

\[
\alpha^{(t+1)} = A(\alpha^{(t)})^{-1} \left( B(\alpha^{(t)}) \alpha^{(t)} + \lambda^{(t)} \alpha^{(t)} \right) \tag{18}
\]

\[
\alpha^{(t+1)} = \frac{\alpha^{(t+1)}}{\|\alpha^{(t+1)}\|} \tag{19}
\]
5. Examples

This section illustrates the PPDE algorithm and our optimization method on a synthetic two-dimensional dataset. We generated a mixture of two Gaussian distributions as depicted in Figure 2. We start from a uniform \( p_0(x) \). Figures 3, 4, and 5 show the approximated densities after one, two and three round, respectively.

It can be seen that after only two rounds the original density has already been approximated well. And, after the third round the elliptical shape of the Gaussians is further refined.

In this experiment we observed that the fixed point iteration consistently converges with different starting values.

6. Encoding prior information

This section illustrates the advantage of exploiting prior information. In Section 2 we have that all feature values are non-negative \((x \geq 0)\), and due to the normalization in (4) \( x^T x = 1 \). Hence, we wish to reflect this information in \( p_0(x) \).

Again, we generate a 2D example. Figure 6(a) shows the data \( x = [x_1 \; x_2]^T \) living in the positive quadrant of the unit circle \( Q = \{ x | x_1 \geq 0, x_2 \geq 0, x_1^2 + x_2^2 = 1 \} \).
The data is distributed according to a mixture of two Gaussians. Figure 6(b) presents two approximations, both use two fixed projection onto the axis, namely \( \alpha_1 = [0 \, 1]^T \) and \( \alpha_2 = [1 \, 0]^T \). For one approximation \( p_0(x) \) is uniform in \([0 \, 1]^2\), and for the other \( p_0(x) \) is uniform in \( Q \).

The difference between the two approximations is clear. The one with proper constraint of \( x \) approximates the true distribution more accurately than without exploiting prior knowledge.

Note that in the PPDE algorithm, we have to sample from the current approximation, which consisting of sampling from \( p_0(x) \). To generate random samples in \( Q \) uniformly, one generates \( x_1 \) and \( x_2 \) from a Gaussian distribution with zeros mean and unit variance, and subsequently takes the absolute values and normalizes \( x \). This method can be generalized to higher dimensions [9], which we will use in our background subtraction experiments.

7. Experiments

We performed experiments on two datasets. The first one is a public dataset the CAVIAR shopping sequences [1]. The second one is a set of city surveillance videos.

In this initial experiment we examine the results by visual inspection. The results are shown without any spatial or temporal smoothing. There are two reasons for the lack of quantitative evaluation. Firstly, we are not aware of a standard dataset with ground-truth for this problem. Secondly, background subtraction is in most cases used as a preprocessing step, for example for target detection. The evaluation in [3], for instance, geared towards this objective using connected component analysis. In our system, we use background subtraction differently. Human detection is carried out by applying complex classifiers such as [2] on image regions with a certain amount of foreground (determined by a threshold value). This approach is more robust than connected component analysis with regard to background noise and holes in object foreground. Since the detection step is different, evaluation with the target detection objective is not meaningful.

For the CAVIAR shopping sequences, we used the WalkByShop1cor sequence consisting of 2360 frames as the training set. We ran the training algorithm without outlier rejection. Figure 7 contrasts the result of our algorithm and that of single Gaussian model [3]. The single Gaussian model has been shown to perform well for this particular dataset [3]. We observed that in general because the imaging condition is quite stable in this dataset the advantages of complex features are not clear. We also observe that the new method often leaves holes in the object appearance. This is probably due to the use of orientation histogram, which for homogeneous region in the foreground the discrimination against background vanishes. This is, however, not a serious problem when one uses background subtraction to guide post-processing such as performing object classification rather than connected component analysis.

For the second set of surveillance videos, we used the complete set as the training set. This is useful for off-line analysis. It can be seen also as outlier rejection procedure. Our purpose here is too show that complex features help in this challenging outdoor scenarios. We use only the simple naive classifier for computational reason as it requires only one-pass through the dataset. Figure 8 compares our result and histogram estimate on gray value. Here we observed that the new algorithm produces much less false alarm. In both cases, the target object is detected by the background subtraction algorithm.
Figure 7: Result on the CAVIAR dataset [1]. The first column is the original frame. The second is the result of the single Gaussian method. The last column is the result of the new algorithm.

8. Discussion

We have proposed the use of complex image descriptors for background subtraction. For efficiency in high-dimensional space, we employ the projection pursuit density estimation method as opposed to the popular kernel density estimation. This results in a more efficient method in run-time. We have made an improvement to the traditional projection density estimation method in the optimization procedure. By employing one-dimensional kernel density estimate, we have derived a fixed point iteration to quickly reduce the objective function. Finally, our future work includes quantitative evaluation of the background subtraction method and theoretical analysis of the fixed point algorithm.

References


Figure 8: Result on city surveillance cameras. Again, the first column is the original frame. The second column shows the result of histogram estimation on gray values. The third column shows the result of the new algorithm. The two pedestrians are detected in both cases. The new algorithm gives no false alarm in the first case, and only three false alarms in the second case.


