Hindawi Publishing Corporation
EURASIP Journal on Image and Video Processing
Volume 2011, Article ID 972961, 12 pages
doi:10.1155/2011/972961

Research Article

Background Subtraction via Robust Dictionary Learning

Cong Zhao,[1] Xiaogang Wang,[1,2] and Wai-Kuen Cham[1]

[1] *Department of Electrical Engineering, The Chinese University of Hong Kong, Hong Kong*
[2] *Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China*

Correspondence should be addressed to Cong Zhao, czhao@ee.cuhk.edu.hk

Received 14 May 2010; Revised 29 September 2010; Accepted 18 January 2011

Academic Editor: Luigi Di Stefano

We propose a learning-based background subtraction approach based on the theory of sparse representation and dictionary learning. Our method makes the following two important assumptions: (1) the background of a scene has a sparse linear representation over a learned dictionary; (2) the foreground is "sparse" in the sense that majority pixels of the frame belong to the background. These two assumptions enable our method to handle both sudden and gradual background changes better than existing methods. As discussed in the paper, the way of learning the dictionary is critical to the success of background modeling in our method. To build a correct background model when training samples are not foreground-free, we propose a novel robust dictionary learning algorithm. It automatically prunes foreground pixels out as outliers at the learning stage. Experiments in both qualitative and quantitative comparisons with competing methods demonstrate the obtained robustness against background changes and better performance in foreground segmentation.

1. Introduction

Segmenting foreground objects from a video sequence is a fundamental and critical step in video surveillance, traffic monitoring, video conferencing, video editing, and many other applications. Background Subtraction (BGS) is used in many of these applications, where each video frame is compared against a background model, and those pixels significantly deviating from the model are considered to belong to the foreground. These "foreground" pixels are further postprocessed for object localization and tracking.

The general framework of BGS usually comprises of four steps: preprocessing, background modeling, foreground detection, and postprocessing. The preprocessing step collects training samples and removes imaging noises; The background modeling step builds a background model which is in general robust to certain background changes; the foreground detection step generates foreground candidates through calculating the deviation of a pixel from the background model; finally, the postprocessing step thresholds those candidates to form foreground masks. Among the four steps, background modeling is the most critical and challenging one to the success of a BGS method.

The difficulties in building a good background model mainly lie in the following two facts.

(a) Background Changes. In practice, the background may undergo complex changes. These changes can be at low-frequency, for example, intensity variation caused by global illumination; they can be at high-frequency, like irregular movements of the rain, tree shaking, and water waves; they can also be repetitive and sudden changes caused by background switching among different configurations, such as traffic light switching among several statuses as illustrated in Figure 2. The background pixels undergoing these complex changes are prone to be misclassified as foreground objects.

(b) Outliers in Training Samples. Another challenge in practical scenarios such as traffic monitoring is that, it is often difficult and laborious in a learning-based BGS method to build a background model with foreground-present frames. The training samples extracted directly from a video record often contain both background regions and unwanted foreground pixels. Directly employing a nonrobust learning method leads to inaccurate background

modeling and poor foreground detection performance.

In this paper, we propose a novel BGS method, which better handles background configuration changes. It exploits two sparsity assumptions for background modeling as well as foreground object detection: (1) the background has sparse linear representation with respect to a learned dictionary, each atom of which characterizes one of the background configurations. (2) The foreground is group sparse in the sense that the majority pixels in a frame belong to the background and these foreground pixels are spatially correlated. Based on these two assumptions, we formulate the background modeling step as a dictionary learning problem, and the foreground detection step as a modified sparse coding problem. Furthermore, in order for the background model to work with foreground-present training samples, we propose a robust learning approach. It simultaneously detects foreground pixels as outliers and builds a correct background model at the learning stage.

The remainder of this paper is organized as follows. Section 2 surveys the literature of background subtraction, and Section 3 gives the mathematical formulation of the proposed method. In Section 4, we show experimental results in comparison with existing methods, and in Section 5, we draw the conclusion.

2. Related Works

Existing BGS methods can be approximately classified into two categories, based on how the background and the foreground are formulated: pixel-level models and frame-level models.

Pixel-level methods typically model the distribution of each pixel in a frame locally and independently. One of the most representative examples is the frame differencing, which is fast but not able to capture interior pixels of a uniformly colored moving object. Along this direction, a more advanced method, known as Mixture of Gaussian (MoG), was proposed in [1]. It states that a static scene can be modeled reasonably well with a mixture of Gaussian distributions. Friedman used a mixture of three Gaussians (corresponding to the road, shadow, and vehicles, resp.) to model the background and foreground in traffic surveillance applications. Stauffer and Grimson [2] extended this idea using multiple Gaussians with multiple hypotheses and found it useful in modeling dynamic scenes such as waving trees, beaches, rain, and snow. The MoG method is popular and usually regarded as the basis for a large number of related techniques. When the assumptions imposed by the selected hypotheses fail, nonparametric approaches are more suitable. A popular nonparametric approach is to use kernels. In this method, a kernel is created around each of the previous samples and the density is estimated using an average over the kernels. While different kernels can be considered, the Normal kernel was proposed by Elgammal et al. [3]. The advantage of such approach is its ability in handling an arbitrary shape of the density function. Last but not the least, Kalman-filter was applied in [4, 5] to model backgrounds with dynamic textures. Kalman filters that exploit more complex state vectors often include higher-order motion characteristics such as velocity and acceleration and are be able to capture more complex dynamic behavior. These methods directly model the distribution of each pixel in the background, and for a new pixel, they calculate its probability of being a foreground or a background one. However, pixel-level BGS methods suffer from deficiencies when facing the two challenges mentioned in Section 1, because these methods often ignore the cue of spatial correlation in background changes. The innocence leads to information insufficiency in both background modeling and foreground segmentation. For example, pixel-intensity changes caused by global illumination variation are highly spatially correlated, and when considered independently they are in nature no different than those caused by the presence of foreground objects, and thus are prone to be misclassified.

Different from pixel-level methods, frame-level methods treat the background pixels of a frame as a whole image and discover the inner structure of the background variation. Owing to the introduction of higher-level information, they can better model global background changes. A representative of this line of works involves employing Principle Component Analysis and its variant versions. The basic assumption is that background changes due to illumination variation are low dimensional, and a background image can be represented by a linear combination of a set of learned basis vectors known as eigen-backgrounds [6]. Later in [7], the authors proposed an incremental PCA method to predict model states, which can be used to capture motion characteristics of backgrounds. In practical applications, there are cases like traffic monitoring that foreground-free training samples are not available. To enable the algorithm to work under these circumstances, the author in [8, 9] proposed a Robust PCA model, which was further developed in [10] to be much faster, more effective and thus more practical. The main advantage of these models is that background changes like illumination variation are treated globally and better modeled in comparison to pixel-level methods.

In addition, recent few years have witnessed successful employment of the Compressive Sensing theory [11] in solving BGS problems. The theory states that a signal can be almost perfectly recovered from only a few measurements if it is sparse [12], that is, majority of its elements are zero or close to zero. These methods make the assumption that the majority of the pixels in a frame belong to the background, and thus the foreground is sparse after background subtraction and can be nearly perfectly recovered from only a few measurements. Since the number of pixels in the foreground is significantly smaller than that in the whole frame, the foreground detection step enjoys significant power reduction on the sensor of a camera. The idea was further developed by [13], in which Markov Random Field (MRF) was employed to impose group effect on foreground pixels since they are spatially adjacent when forming an "object". Later in [14] the authors proposed an alternative approach—the Dynamic Group Sparsity (DGS).

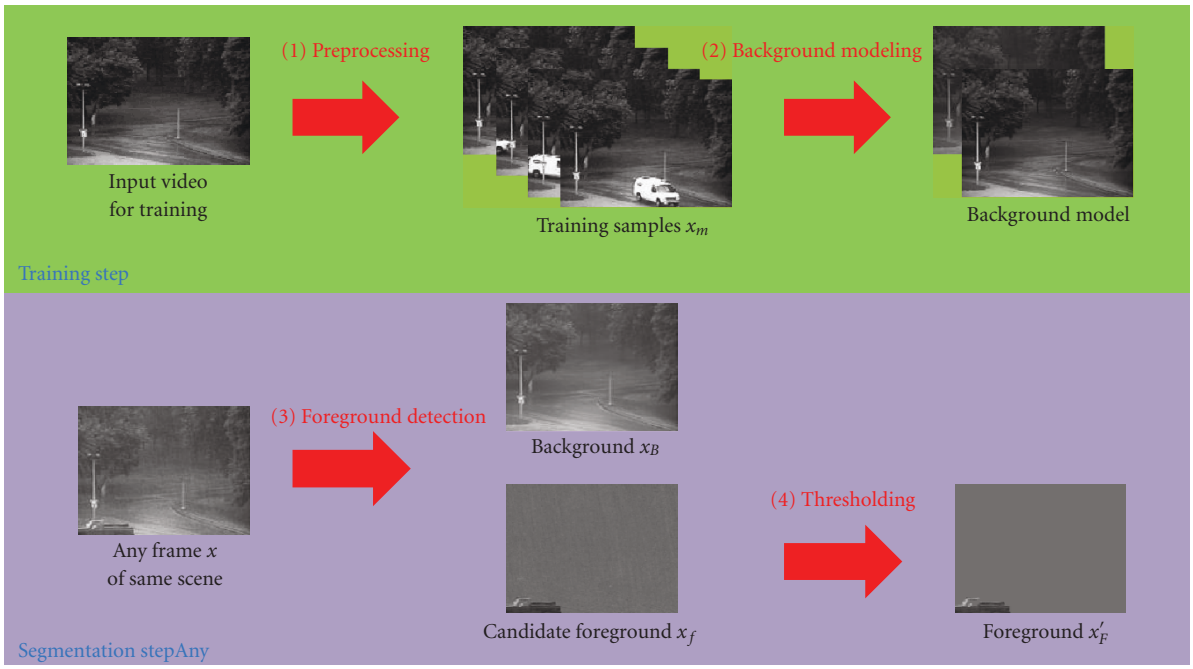


Figure 1: Framework of background subtraction.

(a) (b) (c)

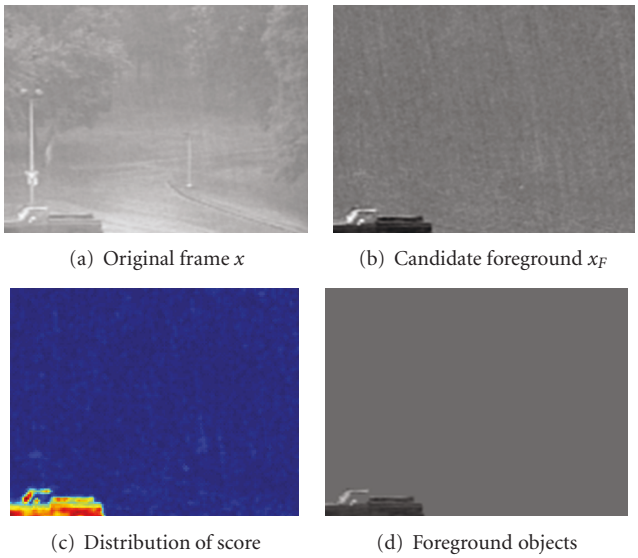
Figure 2: Background switches among several configurations controlled by the status of traffic lights.

In this paper, we propose a novel BGS approach. It is related to the eigen-background methods in the sense that a representative set of basis vectors are learned and retained for background modeling. The difference is that our method provides an automatic mechanism for the background to switch among a set of atoms for its representation without involving all of them at the same time. Our approach is also related to Compressive Sensing methods in its assumption that the pixels in the foreground are group sparse as similar as [13, 14]. However, the difference is that we also assume the background to have a sparse representation and learn a dictionary to characterize the background changes. This enables our background model to handle different configurations caused by, for example, traffic light switching among different statuses. Furthermore, the learning of the dictionary is different from conventional dictionary learning techniques such as [15, 16] in its robustness against outliers. The proposed learning method does not require foreground-free training samples, and it can build a correct background model with outlying foreground pixels automatically pruned out. This is practically important and convenient when foreground-free training samples are difficult to obtain in scenarios like traffic monitoring.

In summary, the main contributions made in this paper and the advantages obtained are the following.

(a) We use dictionary learning to model a background, so that it better handles background changes caused by switching among different configurations.

(b) In order for the learning method to work with corrupted training samples, we propose a Robust Dictionary Learning (RDL) approach, which automatically prunes unwanted foreground objects out in the learning stage and greatly reduces human labor involvement.



(a) Original frame x (b) Candidate foreground x_F

(c) Distribution of score (d) Foreground objects

Figure 3: Discovery of foreground objects.

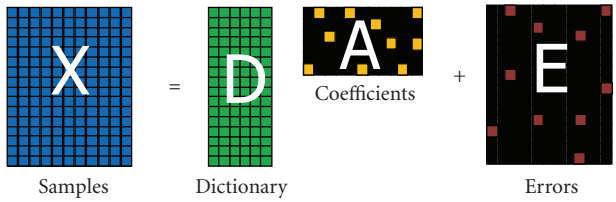


Figure 4: Robust dictionary learning.

(c) We model the foreground detection problem as an L_1-measured and L_1-regularized optimization, the global optimal solution of which can be efficiently found. Furthermore, we use the feature of group effect to segment foreground objects.

3. Methodology

The common framework of existing methods formulates the background subtraction as a linear decomposition problem: to find a background component x_B and a foreground component x_F together constituting a given frame x:

$$x = x_B + x_F, \tag{1}$$

where x_B, x_F, and x are column vectors of the size as the number of pixels. To achieve the decomposition, we rely on prior assumptions about both x_B and x_F. The key to the success is the modeling of the background x_B, which varies among different methods. For example, in [1, 2], the pixels in x_B are assumed to follow a distribution as a mixture of Gaussians. And x_F is in general regarded as the deviation of x from x_B in the sense that whenever a foreground pixel appears it occludes the collocated background pixel, and x_F reflects the confidence of a pixel in x from being a background one.

The work [6] observes that the background of a scene under varying illumination condition is a low-dimensional structure. To identify the structure, they build a set of basis vectors by performing PCA on a set of training background frames. This observation is reasonable because the dimension of illumination variation should be significantly lower than that of the image. However, this assumption is often violated in practical scenarios by (1) local and sudden changes that the background of a scene undergoes and (2) foreground objects that are present in the collected training samples used for background modeling. These scenarios may introduce inaccuracy in the background modeling step and performance degradation in the foreground detection step.

In this section, we address how to model those sudden and local changes caused by the background switching among a number of configurations. Taking Figure 2 for example, the configurations of the background are different when the traffic lights are at different statuses. In Section 3.2, we model the background as a sparse linear combination of atoms from a dictionary D, each atom of which characterizes one of the configurations. We then formulate in Section 3.3 the foreground detection as a sparse coding problem, to simultaneously recover a sparse foreground and a sparse code for the background. In Section 3.4, we address how to build a dictionary D for background modeling so that a new frame can smartly choose only a few atoms for its background representation.

3.1. Sparsity Assumptions. Suppose a scene has C configurations, we assume that each configuration of the background is low dimensional and can be characterized by a set of basis vectors. By stacking these vectors as columns of a matrix D_i, we say that the background x_B of the ith configuration has linear representation $x_B = D_i\alpha_i$, where α_i is the coefficient vector. We define a new matrix D as the concatenation of all the C matrices $D = [D_1, D_2, ..., D_C]$, and thus rewrite x_B in terms of D as

$$x_B = D\alpha, \tag{2}$$

where $\alpha = [0, ..., 0, \alpha_i^T, 0, ..., 0]^T$ is a sparse coefficient vector whose entries are ideally zeros except at those positions associated with D_i. This leads to our first sparsity assumption:

Assumption 1. Background x_B of a specific frame x has sparse representation over a dictionary D.

Furthermore, based on the observation that foreground objects usually occupy minority pixels in a frame, we make another sparsity assumption on the foreground.

Assumption 2. The candidate foreground x_F of a frame is sparse after background subtraction.

3.2. Background Subtraction. With the above two assumptions, the BGS problem can be interoperated as follows: given a frame x, to find the decomposition which has a sparse

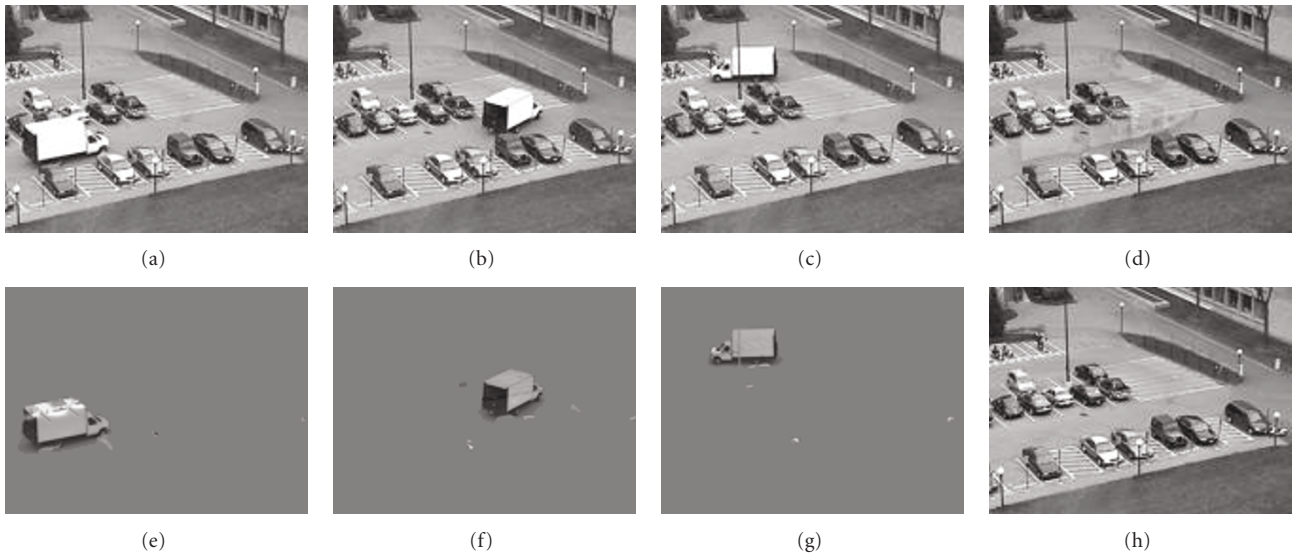


Figure 5: Robust dictionary update step. (a)–(c) A few of samples for update of an atom. (d) Updated atom by K-SVD [15]. (e)–(g) Outliers pruned out by our method. (h) Updated atom by our method.

coded background $x_B = D\alpha$ and a sparse foreground $x_F = x - D\alpha$:

$$\alpha = \arg\min_{\alpha} \|x - D\alpha\|_0 + \lambda\|\alpha\|_0. \tag{3}$$

Here $\|\alpha\|_0$ is the L_0-norm counting the number of nonzero elements, D is the dictionary capturing all the background configurations of a scene as mentioned in Section 3.1, and λ is the weighting parameter balancing between the two terms.

To find the optimal solution for (3) is NP-hard due to the nonconvexity of L_0-norm. Recent development on the theory of compressive sensing [11] advocates that a sparse signal can be recovered by either employing a greedy pursuit algorithm or replacing L_0-norm with its tightest convexation version L_1-norm. However, the problem (3) is different from the CS literature since it involves two sparse terms rather than only one: the sparse foreground $x - D\alpha$ as well as the sparse coded background α. The authors in [17] addressed this type of problem and rewrote (3) as

$$\beta = \arg\min_{\beta} ||\beta||_0 \quad s.t.\ x = D'\beta, \tag{4}$$

where β is the concatenation of α and $x - D\alpha$, that is, $\beta = [\alpha; x - D\alpha]$, and D' is the concatenation of D and the identity matrix, that is, $D' = [DI]$. Since (4) becomes a standard sparse coding problem, it can be solved without difficulty within a general CS framework.

In this paper, we make a different modification by expanding the dictionary in a different manner: we first replace L_0-norm with L_1-norm and obtain an L_1-measured and L_1-regularized convex optimization problem:

$$\alpha = \arg\min_{\alpha} \|x - D\alpha\|_1 + \lambda\|\alpha\|_1, \tag{5}$$

where $\|\alpha\|_1 = \sum_i |\alpha(i)|$. We then rewrite (5) into an equivalent L_1-approximation problem:

$$\alpha = \arg\min_{\alpha} \left\| \begin{pmatrix} x \\ 0 \end{pmatrix} - \begin{pmatrix} D \\ \lambda I \end{pmatrix} \alpha \right\|_1. \tag{6}$$

The advantage of this reformulation over [17] is that, since the number of dictionary atoms in a BGS problem is usually far less than the number of pixels leading to a tall matrix D, the dictionary of size $(K + N) \times K$ in problem (6) is dramatically smaller than that in problem (4) which is as large as $N \times (K + N)$. Therefore, the computational cost of solving (6) is lower than solving (4) in essence.

And since the set of linear equations

$$\begin{bmatrix} x \\ 0 \end{bmatrix} = \begin{bmatrix} D \\ \lambda I \end{bmatrix} \alpha \tag{7}$$

is highly overdetermined (with the number of known elements in α is far less than the number of equations), (6) gracefully satisfies the conditions posed in [18] and thus has a guaranteed global optimal solution. Thus we can reliably segment the candidate foreground x_F from the background x_B given a frame x.

It is worth mentioning that the reason we use L_1-norm instead of L_0-norm is twofold: (a) it enjoys the theoretic advantage that the global optimal solution is guaranteed. (b) It practically accommodates small errors much better than L_0-norm does. This is important since $x - D\alpha$ is usually not perfectly sparse but contain minor model errors or noises even at the locations of inliers.

3.3. Foreground Segmentation. As mentioned in Section 1, the value of a pixel in x_F is the deviation of the pixel from belonging to the background. A nonzero value can be

(a) Original (b) x_B by MoG [2] (c) x_B by DGS [14] (d) x_B by K-SVD [15] (e) x_B by ours

(f) Truth (g) x_F by MoG [2] (h) x_F by DGS [14] (i) x_F by K-SVD [15] (j) x_F by ours

Figure 6: Qualitative comparison of background subtraction results.

caused by the presence of foreground objects, high-frequency background changes, and model errors. In this section, we postprocess the candidate foreground x_F to separate foreground objects from the other two possibilities. The key idea is based on an important observation that foreground objects not only are sparse but also have grouped pixels, that is, pixels on foreground objects are spatially correlated. On the other hand, pixels of high-frequency changes or model errors are comparatively more scattered and less structured.

The authors in [13, 14] made use of this fact, and, respectively, proposed to use MRF and DGS for discovering grouped elements of a sparse signal in the framework of compressive sensing. Inspired by their work, we propose to segment foreground objects which have grouped pixels by calculating a confidence score. It measures the likelihood of a pixel in x_F belonging to a foreground object by not only taking the intensity of the pixel but also its neighborhoods':

$$\text{score}(i) = x_F^2(i) + \sum_{j \in \text{Neighbor}(i)} x_F^2(j). \tag{8}$$

Figure 3 shows an example of its distribution. As can be observed, measured by this metric, foreground objects are much more emphasized than other unstructured and high-frequency changes and model errors. The spirit is the same with [14] in the sense that grouped elements are much easier to be segmented out than other isolated elements. However, the difference is that we do not need to try different sparsity levels to determine the proper number of nonzero elements, leading to more efficient optimization.

3.4. Background Modeling via Dictionary Learning. It is obvious from previous discussion that, the key to the success of applying the approach is how to design an appropriate dictionary D, each atom of which characterizes one of the background configurations. Since the background of a scene in general occupies only a tiny part of the whole image space, those analytical dictionaries such as Over-complete DCT or Wavelets are of no interest. Furthermore, these dictionaries do not characterize background configurations, thus they cannot serve a choice.

A straightforward alternative is to manually collect a number of training samples, and then learn a basis D_i for each background configuration, and finally concatenate them into the dictionary D. This method theoretically works; however, it involves laborious manual classification of training samples, and most importantly, our concern is only the dictionary D, rather than each of its subpart D_i.

The above two facts motivate us to directly learn such a dictionary D from training samples. The idea of dictionary learning technique [15, 16, 19] is to collect a few representative background frames as training samples, and then find an optimal dictionary D satisfying the following: it tries its best at representing all the training samples with minimal error and meanwhile producing the sparsest representations.

$$D = \arg\min_{D,\{\alpha_m\}} \sum_{m=1}^{M} \|x_m - D\alpha_m\|_2^2 + \lambda \cdot \|\alpha_m\|_1, \tag{9}$$

where x_m is the mth sample within a training set of size M and α_m is a sparse coefficient vector for each sample. Notice that we choose L_1-norm instead of L_0-norm for the reason mentioned in Section 3.1.

Existing dictionary learning makes a good method for building a background model if a collection of clean background frames can be provided. However, in certain practical scenarios such as traffic monitoring, foreground-free samples are difficult or laborious to collect. When working with the above objective function, those foreground pixels may violate model assumptions (e.g., linear representation of x_B and iid-Guassin error), and the local regions which are foreground-corrupted cannot be modeled well by these methods. As shown in Figure 5(d), one drawback of these methods is its vulnerability to outliers existed in training samples. To learn a dictionary under this circumstance is a chicken-egg puzzle: a correct dictionary can be built if those outliers can be excluded, and vice versa. To solve this puzzle, we propose in Section 4 a robust learning method which achieves both the above two targets simultaneously.

Figure 7: Comparison of a few representative atoms from the dictionary learned by: (top) K-SVD [15] and (bottom) our RDL method.

4. Robust Dictionary Learning

To make the learning algorithm robust against outliers, we develop in this section a Robust Dictionary Learning (RDL) approach which simultaneously segments the outlying foreground pixels out and builds a correct dictionary (see Algorithm 1). This is achieved by optimizing a modified objective function (10): to find the optimal dictionary D which approximates the training data with minimum amount of outlying foreground pixels and produces the sparsest representations of the backgrounds.

$$D = \arg\min_{D,\{\alpha_m\}} \sum_{m=1}^{M} \|x_m - D\alpha_m\|_1 + \lambda\|\alpha_m\|_1. \quad (10)$$

The difference from conventional dictionary learning methods [15, 16, 19] lies in the measure of the reconstruction error: they employ L_2-norm while we employ L_1-norm. However, the modification is not trivial, since outlying foreground objects make the iid-Gaussian error assumption violated, and we instead assume a heavy-tailed iid-Laplacian distribution, which is known to handle outliers better.

To find an optimal solution D for (10) is not easy, since it involves the product of unknown terms D and α_m. We rewrite it into a matrix form and obtain an equivalent problem:

$$D = \arg\min_{D,A} \|X - DA\|_1 + \lambda\|A\|_1, \quad (11)$$

where X is the matrix of training data each stacked as a column, A is the matrix of coefficient vectors stacked in a similar way, and $\|A\|_1 = \sum_{i,j} |A(i,j)|$ is the "1-norm" of the matrix A defined in this paper as the sum of absolute values of its entries.

Figure 4 illustrates the objective in a matrix-factorization perspective $X = DY + E$, where E is a sparse matrix of outliers.

In this factorization problem, the known variable is only X. Previous discussions shed some lights on the unknowns D, A, and E: the dictionary D is of fixed size, coefficients A and errors E are sparse, leading to the objective function (11). Since it is not jointly convex for D and A, we use the same spirit with [15, 16, 19] which iteratively and alternately optimizes D and A with each other frozen. We name the two steps as robust sparse coding and robust dictionary update, respectively.

4.1. Robust Sparse Coding. The robust sparse coding step optimizes coefficient matrix A with dictionary D being constant:

$$A = \arg\min_{A} \|X - DA\|_1 + \lambda\|A\|_1. \quad (12)$$

Since the training samples (columns of X) are assumed to be independent from each other, we break (12) into M independent subproblems in a columnwise fashion:

$$\alpha_m = \arg\min_{\alpha_m} \|x_m - D\alpha_m\|_1 + \lambda\|\alpha_m\|_1 \quad \text{for } m = 1, \ldots, M. \quad (13)$$

Each subproblem in (13) is an L_1-measured and L_1-regularized convex optimization problem which has been addressed in Section 3.2, so we redirect the readers to that section on the solutions.

(a) Original (b) Ground truth (c) MoG [2]

(d) KDE [3] (e) DGS [14] (f) Ours

Figure 8: Results on the sequence "Rain".

4.2. Robust Dictionary Update. With the coefficient matrix A being updated and considered constant, we disregard the second term in (11) and update D:

$$D = \arg\min_{D} \|X - DA\|_1. \tag{14}$$

We assume that the atoms in D are independent from each other and thus update them each separately.

$$d_k = \arg\min_{d_k} \left\| X - d_k \alpha^k \right\|_1 \quad \text{for } k = 1, \ldots, K, \tag{15}$$

$$d_k^i = \arg\min_{d_k^i} \left\| x^i - d_k^i \alpha^k \right\|_1 \quad \text{for } i = 1, \ldots, N,\ k = 1, \ldots, K, \tag{16}$$

$$\alpha_j^k = \arg\min_{\alpha_j^k} \left\| x_j - d_k \alpha_j^k \right\|_1 \quad \text{for } j = 1, \ldots, M,\ k = 1, \ldots, K, \tag{17}$$

where d_k is kth atom of D and α^k is kth row (coefficients corresponding to d_k) of A. It is worth to mention that in (15), we only extract the columns of X whose coefficients in α_k are above a small threshold. This is because the elements in α_k may not be exactly zero but close to, and the thresholding step retains those "related" samples rather than involves all of them. It significantly speeds up updating d_k and avoids from overfitting. Besides, we normalize the atoms d_k so that they have unit Euclidean norms, otherwise (15) may run into trivial solutions with arbitrary small values.

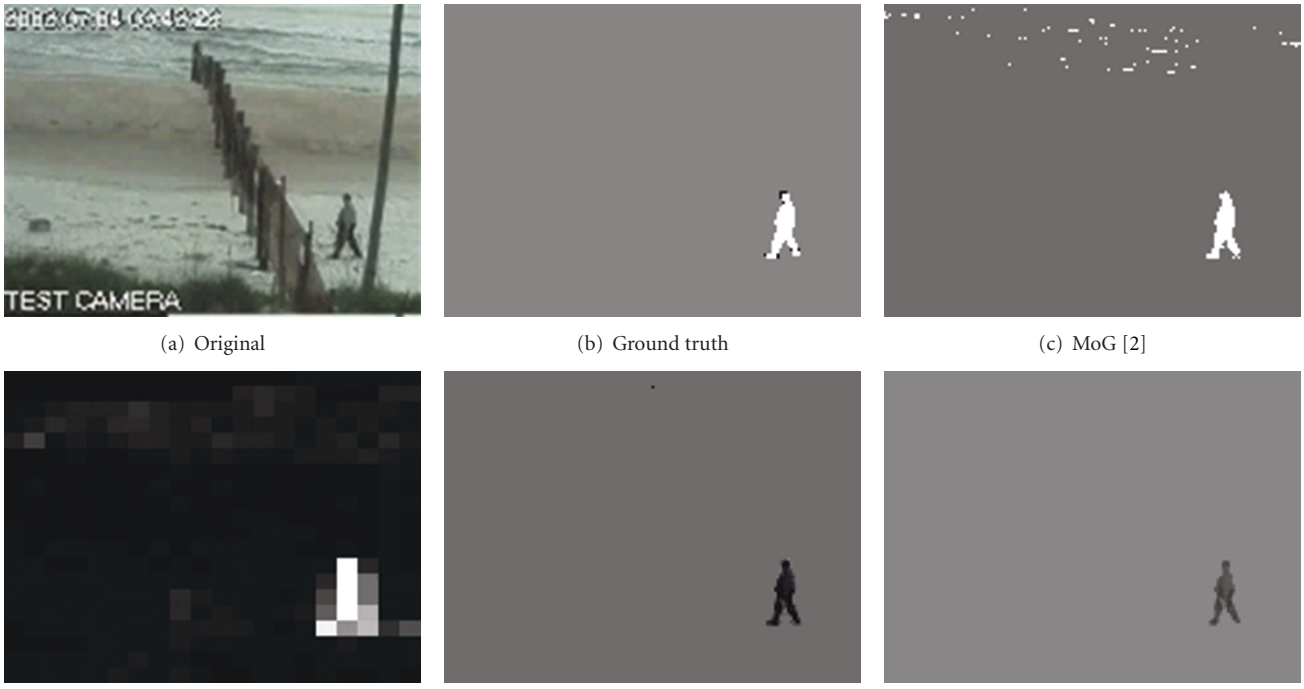
By further breaking each of (15) into a set of L_1-regression problems (16), we obtain a closed-form solution for each of them by employing the Iterative-Reweighted Least Square method [20, 21]. While this strategy works well, we find that the convergence of solving (15) requires fewer iterations if we involve (17) at the same time, which updates the coefficients α^k. In summary, we iterate between (16) and (17) until the convergence on d_k is reached.

Figure 5(h) illustrates an updated dictionary atom, with some of the involved samples shown in Figures 5(a)–5(c) and detected outliers in Figure 5(e)–5(g). For comparison with conventional learning techniques, we show the updated atom using K-SVD [15] in Figure 5(d). As can be observed, K-SVD produces inaccurate dictionary atom (ghosting effect) at regions where outliers (foreground vehicles) are present, while our method generates a correct update completely free from outliers.

5. Experimentation

To test the performance of the proposed method in handling above-mentioned background changes, in this section, we conduct two experiments: one is on the background configuration changes which are local and sudden, the other is on the nonstructured and high frequency changes.

5.1. Local and Sudden Changes. A typical example in practice is that the scene of interest contains traffic lights switching among different statuses. The background undergoing these changes can be well modeled by a dictionary each atom of which captures one of the traffic light statuses. Since this type



(a) Original (b) Ground truth (c) MoG [2]

(d) Monnet's [4] (e) DGS [14] (f) Ours

Figure 9: Results on the sequence "Ocean".

of change has not been addressed in existing works, we do not have a public dataset to experiment on. Therefore, we collect some video clips on Youtube [22] to make the data. The video of size 120×160 is shot by a surveillance camera set at a railway crossing. The traffic lights take on three different statuses: on-off, off-on, and off-off.

To model the background, we extract totally 38 frames (containing foreground vehicles) at a constant interval to form the training set. These frames cover all the three traffic light statuses. We randomly select $K = 15$ of them to initialize a dictionary, and then update the dictionary by dictionary learning techniques, such as K-SVD as well as ours. The parameters are set to be iteration number $J = 5$ and $\lambda = 3$. Notice that, while still using the name K-SVD throughout this paper, we replace the L_0-norm by L_1-norm for fair comparison. For foreground object detection, we implement both methods on the remaining part of the video. In addition, we implement the method of Dynamic Group Sparsity (DGS) [14] and Mixture of Gaussians (MOG) [2] with provided source codes.

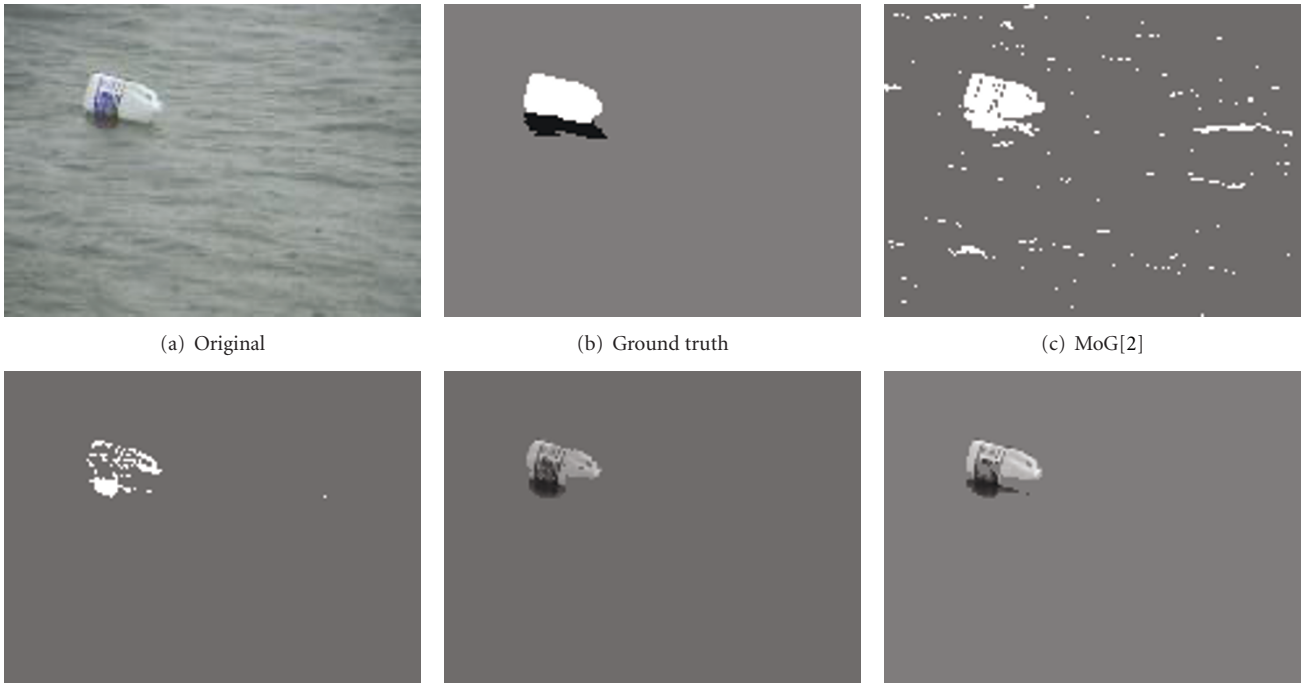
To measure the segmentation results quantitatively, we manually labeled the ground-truth masks on 10 representative frames of each sequence. As shown in Figure 6(f), white pixels indicate foreground objects, and gray ones indicate the background. Since there are ambiguous pixels, for example, on shadows or blurred edges, we label them as black and disregard them in the calculation of segmentation error. The qualitative and quantitative comparisons are given in Figure 6 and Table 1, respectively. The numbers in Table 1 indicate the permillage of misclassified pixels over the total number pixels of the frames.

Table 1: Quantitative comparisons on data collected from [22].

	MOG [2]	DGS [14]	K-SVD [15]	Ours
False negative	1.45‰	0.0‰	2.45‰	0.46‰
False positive	8.23‰	21.3‰	38.49‰	2.14‰
Total error	9.67‰	21.3‰	40.94‰	2.60‰

As can be observed, our method performs consistently better than existing BGS methods MOG [2] and DGS [14] when the traffic light switches in the background. For both MOG and DGS methods, the pixels are misclassified as foreground objects. It is worth to mention that, although the method DGS also assumes sparse representation of the background and group sparse of the foreground, it still fails to model the traffic lights because it does not successfully build a dictionary to describe these background configurations. And since a frame mostly resembles its immediate previous neighbors, their online dictionary update boils down to an extended frame-differencing approach in a certain sense. Thus the pixel changes caused by background switching are persistently regarded as foreground objects in their method. Besides, its exhaustive searching strategy for the proper amount of foreground pixels is computationally inefficient especially for large-sized foreground objects.

Also, the proposed RDL approach outperforms conventional dictionary learning methods K-SVD in the sense that it builds a much better background model when the training samples contain outliers. The resultant difference between the two methods is illustrated in Figure 7 where each atom is linearly stretched to [0, 255] for display. As can be seen,



(a) Original (b) Ground truth (c) MoG[2]

(d) Robust Kalman[5] (e) DGS [14] (f) Ours

Figure 10: Results on the sequence "Water".

Algorithm: **Robust Dictionary Learning**
Parameters: J (no. of iterations), K (number of dictionary atoms), λ (Lagrange multiplier)
Initialization: Initialize D by K randomly selected samples
Loop: Repeat J times
(i) *Robust Sparse Coding*:
Fix D, and compute coefficient vector α_m for each sample x_m by solving(11)
(ii) *Robust Dictionary Update*:
Fix all α_m, and for each dictionary atom $d_{k\cdot}$, $k \in 1, 2, \ldots, K$,
(a) Select the set of samples relevant to the atom via thresholding α^k
(b) Update d_k by iterating between (16) and (17) until convergence:
Output:
Optimized dictionary D

Algorithm 1: Description of proposed robust dictionary learning algorithm.

the dictionary atoms learned by K-SVD is corrupted at the places where outliers (foreground vehicles) are present. In comparison, our method can reliably detect and prune out the outliers as long as they are not dominant, that is, they do not persistently appear at the colocation of all the training frames.

5.2. Nonstructured High-frequency Changes. In practice, besides local and sudden background changes, there can be high-frequency changes caused by rain, tree shaking, or water waves especially for outdoor surveillance. These changes are different in nature from those caused by appearance of foreground objects, since they are much less structured. In this experiment, we perform foreground segmentation on the dataset [5] and compare its performance with [2–5, 14]. The parameters for our method are exactly the same with those used in Section 5.1. The results produced by other methods are directly drawn from http://paul.rutgers.edu/~jzhuang/. The comparison is shown in Figures 8, 9, and 10 and Table 2.

Table 2: Quantitative comparisons on dataset [5].

	Ocean			Rain			Water		
	FN	FP	Total	FN	FP	Total	FN	FP	Total
Ours	0.73	0.21	**0.94**	2.91	2.13	5.04	5.88	0.16	6.04
DGS	0.10	1.61	1.71	1.93	2.86	**4.79**	5.31	0.10	**5.41**
MOG	0.26	4.32	4.58	12.7	2.39	15.09	2.92	15.9	18.82
KDE				0.10	8.85	8.95			
KAL							15.1	0.05	15.15

As can be observed, our method performs better than conventional methods [2–5] in handling less-structured and

high-frequency background changes. It performs comparatively well as DGS [14], since the background has only one configuration for each sequence, and both DGS and our method can correctly model it. The main difference is that DGS is based on L_0-norm, while ours on L_1-norm. As mentioned in Section 5.1, our method does not involve either manually setting or exhaustively searching the proper size of foreground objects, which is practically convenient.

5.3. Discussions. While the proposed algorithm works well in background subtraction, there is a need hereby to review the key assumptions and some requirements for its successful application.

(a) The Background can be Represented as a Linear Combination of Basis Vectors (Dictionary Atoms). This constraint applies to scenarios that, as discussed in Section 3, the background undergoes global illumination variation or local sudden changes which have a limited number of configurations. It cannot work if the background changes cannot be modeled likewise. For example, when the video is captured by a moving hand-held camera, the movement of the scene makes the linear representation assumption violated.

(b) The Foreground should Cover Minority Pixels of a Frame. Our algorithm requires the foreground to be sparse so as to perfectly recover it. This constraint guarantees theoretic equivalence between L_0-norm and L_1-norm. However, we find it can be relaxed in practice. In both the background modeling step and the foreground detection step, we find the algorithm successfully detects foreground pixels and builds a correct dictionary as long as the outliers are not dominant. It seems that foreground pixels can be segmented even if it covers almost the whole frame. The empirical evidence is as similar as that reported in [17]. Notice that, even when the constraint is not perfectly satisfied, L_1-norm still enjoys the benefits mentioned in Section 3.2.

(c) Background Changes, Except those which can be Modeled as Linear Combination of Dictionary Atoms, should be at High-Frequency and Less Structured than Foreground Objects. When this constraint is met, we can employ 4-connected, 8-connected, or more sophisticated model to successfully discover the foreground objects from a sparse candidate frame.

6. Conclusion

In this paper, we proposed a learning-based method for BGS. The method exploits a novel robust dictionary learning approach for background modeling, and segments foreground objects by optimizing an L_1-measured and L_1-regularized problem. We tested the performance of the method in qualitative and quantitative comparison with existing methods. It outperforms these methods in background modeling and foreground detection when the background exhibits sudden and local changes as well as high-frequency changes.

The proposed robust dictionary learning method can also be applied to solving other problems, for example, motion segmentation [23, 24] where outliers are essentially problematic. It removes the outliers during the learning stage and generates a clean dictionary for sparse representation.

Acknowledgments

This work is supported by the General Research Fund sponsored by the Research Grants Council of Hong Kong (Ref. no. CUHK417110), National Natural Science Foundation of China (61005057), and Shenzhen Basic Research Program for Distinguished Young Scholar (JC201005270350A).

References

[1] N. Friedman and S. Russell, "Image segmentation in video sequences: a probabilistic approach," in *Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence (UAI '97)*, August 1997.

[2] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99)*, pp. 246–252, June 1999.

[3] A. M. Elgammal, D. Harwood, and L. S. Davis, "Non-parametric model for background subtraction," in *Proceedings of the 6th European Conference on Computer Vision (ECCV '00)*, pp. 751–767, 2000.

[4] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh, "Background modeling and subtraction of dynamic scenes," in *Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV '03)*, pp. 1305–1312, October 2003.

[5] J. Zhong and S. Sclaroff, "Segmenting foreground objects from a dynamic textured background via a robust Kalman filter," in *Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV '03)*, pp. 44–50, October 2003.

[6] N. M. Oliver, B. Rosario, and A. P. Pentland, "A Bayesian computer vision system for modeling human interactions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 831–843, 2000.

[7] A. Mittal, A. Monnet, and N. Paragios, "Scene modeling and change detection in dynamic scenes: a subspace approach," *Computer Vision and Image Understanding*, vol. 113, no. 1, pp. 63–79, 2009.

[8] F. de la Torre and M. J. Black, "A framework for robust subspace learning," *International Journal of Computer Vision*, vol. 54, no. 1-3, pp. 117–142, 2003.

[9] Q. Ke and T. Kanade, "Robust L1 Norm factorization in the presence of outliers and missing data by alternative convex programming," *in Proceedings of IEEE Computer Society Conference on Computer Vision and Patter Recognition (CVPR)*, vol. 1, pp. 739–746, 2005.

[10] J. Wright, A. Ganesh, S. Rao, and Y. Ma, "Robust principal component analysis?" in *Proceedings of the Conference on Neural Information Processing Systems (NIPS '09)*, Whistler, Canada, December 2009.

[11] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[12] V. Cevher, A. Sankaranarayanan, M. F. Duarte, D. Reddy, R. G. Baraniuk, and R. Chellappa, "Compressive sensing for background subtraction," in *Proceedings of the 10th European*

Conference on Computer Vision (ECCV '08), vol. 5303 of *Lecture Notes in Computer Science*, pp. 155–168, 2008.

[13] V. Cevher, M. F. Duarte, C. Hegde, and R. G. Baraniuk, "Sparse signal recovery using Markov random fields," in *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems (NIPS '08)*, pp. 257–264, Vancouver, Canada, December 2008.

[14] J. Huang, X. Huang, and D. Metaxas, "Learning with dynamic group sparsity," in *Proceedings of the 12th International Conference on Computer Vision (ICCV '09)*, pp. 64–71, October 2009.

[15] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[16] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Reseach*, vol. 11, pp. 19–60, 2010.

[17] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[18] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

[19] R. Rubinstein, A. M. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1045–1057, 2010.

[20] K. R. Gabriel and S. Zamir, "Lower rank approximation of matrices by least squares with any choice of weights," *Technometrics*, vol. 21, no. 4, pp. 489–498, 1979.

[21] G. James, *Matrix Algebra*, 6.8.1 Solutions That Minimize Other Norms of the Residuals, Springer, New York, NY, USA, 2007.

[22] http://www.youtube.com/.

[23] S. Rao, R. Tron, R. Vidal, and Y. Ma, "Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 10, pp. 1832–1845, 2010.

[24] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pp. 2790–2797, June 2009.