Discovering Class Specific Composite Features through Discriminative Sampling with Swendsen-Wang Cut

Feng Han[1] Ying Shan[2] Harpreet S. Sawhney[1] Rakesh Kumar[1]
[1]Sarnoff Corporation
201 Washington Road
Princeton, NJ 08540, USA
{fhan,hsawhney,rkumar}@sarnoff.com
[2]Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
{Ying.Shan}@microsoft.com

Abstract

This paper proposes a novel approach to discover a set of class specific "composite features" as the feature pool for the detection and classification of complex objects using AdaBoost. Each composite feature is constructed from the combination of multiple individual features. Unlike previous works that design features manually or with certain restrictions, the class specific features are selected from the space of all combinations of a set of individual features. To achieve this, we first establish an analogue between the problem of discriminative feature selection and generative image segmentation, and then draw discriminative samples from the combinatory space with a novel algorithm called Discriminative Generalized Swendsen-Wang Cut. *These samples form the initial pool of features, where AdaBoost is applied to learn a strong classifier combining the most discriminative composite features. We demonstrate the efficacy of our approach by comparing with existing detection algorithms for finding people in general pose.*

1. Introduction

Recent trend in computer vision research sees proliferation of algorithms detecting complex objects such as people with multiple aspects and poses. Although automatic object detection is not yet a solved problem, two methodologies have prevailed over time. One methodology, originated from the AdaBoost face detection algorithm [20], detects object in a top-down fashion by quickly searching through the whole image with a "strong" classifier, represented as a boosted collection of "weak" classifiers. The other methodology, rooted from the Generalized Hough Transformation, detects object in a bottom-up fashion by piecing together the voting scores of a collection of location/pose encoded visual words [12]. Recently, these two methodologies have found a common ground [15] by designing weak classifiers based on voting units using bottom-up features, and combining these units with AdaBoost. Therefore the general philosophy for object detection seems to converge to a unified framework of first generating a pool of features, and then selecting discriminative features to form the final classifier. Under this framework, the quality of the features inside the pool is therefore critical to the performance of the final classifier.

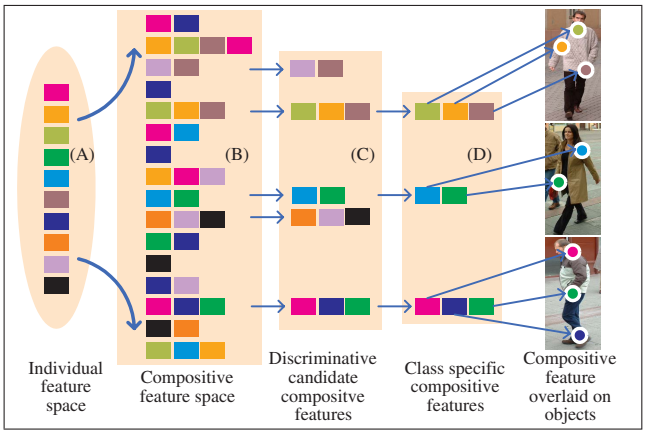

Figure 1. Learning class specific composite features for the detection of complex objects. (A) is a set of individual features, whose identities are color coded. (B) is the composite feature space that consists of all the combinations of the features in (A), where each row denotes one composite feature. The key idea is to first upgrade the feature space from (A) to the more discriminative (B), and then select discriminative features from (B) for detection. To accomplish the seemingly impossible task, we proposed a discriminative sampling algorithm that reduces (B) to (C), which is a manageable-sized list of candidate composite features. AdaBoost is then used to select from (C) a set (D) of class specific composite features for real-time detection. The images shown on the right most column exemplify the selected composite features, when overlaid on top of the objects being detected.

The original AdaBoost-based face detection algorithm [20] generates a pool of individual features by vary-

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

ing the size and position of several types of Haar features. Since these Haar features are designed mostly for frontal face detection, the feature pool does not cover features that are critical to the detection of other complex objects such as people. One approach to remedy this is to replace Haar feature with newly designed stronger feature, such as HOG in [21]. The other approach, which is becoming increasingly popular, is to strengthen existing features by "composite features" as in [21, 5, 14, 1]. A composite feature usually consists of multiple individual features with specific spatial relationships, as shown in the right most column in Fig. 1. Unlike individual features, composite features are not only more discriminative, but also more flexible, which means they are configurable by varying both the number of the constituting features and their spatial configurations. It is therefore possible to build a composite feature set that provides a continuum of discriminative features to model complex objects with large structural variations.

However, because of the limited search capability of AdaBoost, and the combinatorially explosive nature of composite features, the computational cost of selecting features from a large composite feature set is prohibitively expensive. One straightforward way of avoiding this problem is to limit the constituting individual features to a small number, as in [15]. The other interesting approach is to constrain the composite features with some heuristics. For instance, in [14], Gestalt laws (e.g. proximity) are first applied to reduce the possible candidate compositions among individual features. These candidates are further refined with an entropy based discriminative relevance measure before being used to build the final classifier. While promising, these methods leave the full power of the composite features unexploited.

As illustrated in Fig. 1, the objective of this paper is to first upgrade a set (A) of individual features to a *composite feature space* (B), which contains all the combinations of the individual features, and then select discriminative features from (B). Since the upgrade greatly extends both the dimensionality and the number of feature configurations, composite feature space provides superior discriminating power for the detection of complex objects. The key technical contribution of this paper is the algorithm that allows us to go from (B) to (C) in Fig. 1. We achieve this by establishing an analogue between the problem of discriminative feature sampling and image segmentation, and solving it with a novel algorithm called *Discriminative Generalized Swendsen-Wang Cut*. More specifically, we define a discriminative proposal function and a posterior distribution according to a discriminative objective function, and use them to guide the sampling from the composite feature space. The algorithm enjoys the nice properties of the Generalized Swendsen-Wang Cut, which ensures that samples drawn from the posterior are discriminative or *class specific* with high probabilities. The final classifier for detection is constructed with AdaBoost by selecting from (C) a set (D) of class specific composite features, as shown in Fig. 1. We demonstrate the efficacy of our approach by comparing with the existing AdaBoost-based algorithms for detecting people in general pose.

2. Related Work

In addition to the prior works mentioned in Sec. 1, our approach is related to a class of object classification algorithms using part-based representation. The Constellation model [7] and the pictorial structure [6] represent object class using a collection of parts with spatial relationships. A layered pictorial structure [11] extends the pictorial structure by assigning layer numbers to individual parts. Instead of learning multiple composite features with various configurations as in our approach, the focus of these works is to learn a statistical shape model with fixed topology and number of parts. While the statistical nature offers a certain level of flexibility, these models are not expected to handle complex objects with drastic topological variations. There is a large body of class-specific segmentation works [3, 12], where part-based object models are used to guide the search for reliable object boundaries. OBJCUT [10] proposes an efficient approach to integrate a layered pictorial model, learned in advance, with the segmentation process. Levin et. al. [13] proposes an algorithm that combines the learning process with the segmentation process, using object features as a bias term in the Conditional Random Field formulation. The algorithm maintains a collection of features, from which a subset is selected as the object model during segmentation. This algorithm is not designed for efficient object detection, and also seems to have limitation on the number of features used for modeling.

Our work is also related to the classic feature selection approaches such as SVM, PCA, LDA, and pLSA [9]. For real time object detection, which is the targeted application domain of this paper, these approaches do not share the same computational advantages with the AdaBoost framework, though algorithms such as SVM have comparable performance in terms of classification accuracy. Generalized Swendsen-Wang Cut [2] is an efficient sampling algorithm primary designed for generative region segmentation. In this paper, we have adapted it to solve a discriminative sampling problem.

3. Problem Statement and Notations

3.1. The Fundamental Problem

Let $\mathcal{B} = \{(I_1, y_1), \cdots, (I_m, y_m)\}$ be a training set with both positive and negative samples, where I_i is the ith sample image, and $y_i = \{-1, 1\}$ is the class label of the image.

We want to find an optimal classifier by minimizing an exponential loss function L over the training set $\mathcal{B}$,

$$\min_{\{\Theta_s|s\in\mathcal{S}\}} \sum_{i=1}^{m} L\left(y_i, F(I_i;\{\Theta_s\})\right), \tag{1}$$

where s is a feature in the feature space[1] $\mathcal{S}$, and $\Theta_s \equiv (\tau_s, \alpha_s, \Lambda_s)$ is the corresponding parameter set of s in the additive model

$$F(I;\{\Theta_s\}) \equiv \sum_{s\in\mathcal{S}} \tau_s \alpha_s f(I;\Lambda_s), \tag{2}$$

which is used to map an image I to a class label. In the parameters set Θ_s, $\tau_s = \{0,1\}$ indicates whether the feature is selected to be included in the additive model, α_s is the weight of each basis function f, and Λ_s is the parameter set of the basis function. The loss function L is defined as

$$L(y, F(I)) = \exp(-y\ F(I)).$$

The rationale of using an exponential loss function and an additive model for classification is well-established in the AdaBoost literatures by Schapire [17] and Freund et. al. [8]. As compared with other state-of-the-art classification algorithms such as SVM, Viola et. al.'s work in [20] demonstrated additional advantages of using AdaBoost for object detection, i.e., effective feature selection and real time classification. The basis function f in the context of object detection is a "weak classifier" that corresponds to an object feature. If we partition the feature space into two subsets $\mathcal{S}_0 = \{s|\tau_s = 0, \forall s \in \mathcal{S}\}$ and $\mathcal{S}_1 = \{s|\tau_s = 1, \forall s \in \mathcal{S}\}$, (2) becomes

$$F(I;\{\Theta_s\}) \equiv \sum_{s\in\mathcal{S}_1} \alpha_s f(I;\Lambda_s), \tag{3}$$

which means that the additive model contains only the basis functions whose corresponding features are selected. In brief, the objective of (1) is to minimize the training error by finding an optimal subset $\mathcal{S}_1^* \subseteq \mathcal{S}$, building weak classifiers f from the selected features $s \in \mathcal{S}_1^*$, and searching for the optimal parameters (α_s, Λ_s) for the weak classifiers. While many prior works exist to improve the solution of (1) from different aspects, the focus of this paper is to study the fundamental problem of *how to choose a good feature space $\mathcal{S}$ to begin with*.

3.2. Composite Feature Space

Given a set of basic features $\mathcal{V} = \{v_1, v_2, \cdots, v_N\}$, we define a *composite feature space*, denoted as $\mathcal{S}^{\mathcal{V}}$, as the set of all combinations of the features in $\mathcal{V}$. The key idea is to select discriminative features from $\mathcal{S}^{\mathcal{V}}$ instead of $\mathcal{V}$. We propose to solve this problem using a new algorithm called Discriminative Generalized Swendsen-Wang Cut.

[1] Feature space in this context is equivalent to feature set or feature pool.

4. Sampling Class Specific Composite Features with Generalized Swendsen-Wang Cut

Instead of exhaustively searching through all the composite features in $\mathcal{S}^{\mathcal{V}}$, which is an intricate problem, we focus the search in a much smaller subspace where class specific composite features occur with high probabilities. This falls into the MCMC sampling framework, in which Generalized Swendsen-Wang Cut (GSWC) [2] is a recent development applied to image segmentation and provides an efficient way of drawing samples from a general form distribution – usually a posterior. Following this direction, we formulate the problem of sampling class specific composite feature in a way that is similar to the image segmentation problem, and solve it with a discriminative version of Generalized Swendsen-Wang Cut. The rationale of using GSWC will become clear later in this section.

4.1. Background: Image Segmentation with GSWC

In [2], image segmentation is formulated as a graph partitioning problem, where the objective is to partition an adjacency graph $\mathcal{G} =< \mathcal{V}, \mathcal{E} >$ so that the subgraphs represent coherent regions. The set $\mathcal{V}$ contains nodes of the graph, which correspond to the *atomic regions* on an oversegmented image. The spatial adjacency relationships between the atom regions form the edges inside the set $\mathcal{E}$. A *link probability* is defined on $\mathcal{E}$ to reflect the feature dissimilarity between each connected pair of atomic regions. The partitioning algorithm searches the solution space by iteratively drawing samples under the guidance of an objective function – the *generative segmentation posterior*, which is computed by fitting hypothetical regions with region models and then combining the posteriors of individual regions. The solution space contains both the segmentation labels and the *region model parameters*. Searching in this huge solution space is a non-trivial problem, which is not feasible for optimization algorithms based on the gradient of the objective function. A practical solution is to use Markov Chain-based sampling algorithms, among which Gibbs Sampling is an early attempt. The success of Gibbs Sampling is nevertheless limited by the search strategy that flips/changes the label of only one node at each iteration. Swendsen-Wang method [18] improves the efficiency by simultaneously changing a subgraph with multiple coupled nodes. However, both approaches can deal with only Potts model, which excludes the use of data term to guide the search. DDMCMC [19] addresses this problem by allowing general form posteriors including the data term, but the search is still based on changing individual nodes. GSWC generalizes the Swendsen-Wang method to allow general form posteriors. As a result, it can change a subgraph at a single step with the guidance from the bottom-up

data term, and achieve dramatic speedup over Gibbs sampler (400 times) and DDMCMC (40 times). Figure 2 illustrates the process of a single GSWC step, where the labels of a group of nodes (the atomic regions) are changed from black to white when transiting between two partitions. Given the current Partition A, GSWC traverses through the graph and randomly turns on and off edges according to their link probabilities. After this stage, previously connected regions such as $\mathcal{V}_1$ can be broken into sub-regions. In this case, the sub-graph $\mathcal{V}_0$ highlighted inside the polygonal region becomes an independent connected component. From a number of regions like $\mathcal{V}_0$, GSWC uniformly selects one and determines the posterior probability of assigning the region with another label. As exemplified in Fig. 2, $\mathcal{V}_0$ is decided to merge with $\mathcal{V}_2$, and assigned with the white label. *Swendsen-Wang Cut* refers to the set of edges between $\mathcal{V}_1$ and $\mathcal{V}_0$, highlighted with the little crosses.

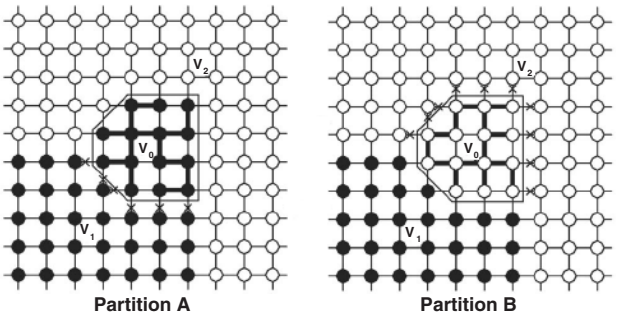

Figure 2. Flipping the labels of a component V_0 at a single step from Partition A to Partition B. (Graphics courtesy of A. Barbu and S. Zhu)

4.2. Analogue with Image Segmentation Problem

We now consider the adjacency graph $\mathcal{G}_0 =< \mathcal{V}, \mathcal{E}_0 >$ for the discriminative feature sampling problem. A major difference is that $\mathcal{V}$ is the set of basic features that construct the composite features, and $\mathcal{E}_0$ is a set of edges connecting the basic features. An n-partition of the graph is defined as

$$\pi_n = (\mathcal{V}_1, \ldots, \mathcal{V}_n); \cup_{i=1}^{n} \mathcal{V}_i = \mathcal{V}; \mathcal{V}_i \cap \mathcal{V}_j = \emptyset, \forall i \neq j. \quad (4)$$

Since $\mathcal{V}$ is the basic feature set, each subset $\mathcal{V}_j$ of a n-partition is essentially a composite feature in $\mathcal{S}^{\mathcal{V}}$ as defined in Sec. 3.2. As a result, if we can generate n-partitions under the guidance of the objective function in (1), we will be able to draw class specific composite features with high probabilities. Note here a single base feature can contribute to multiple composite features from different n-partitions. To achieve this goal, we form a new set of parameters

$$\mathcal{W} = (n, \pi_n, \Theta_1, \ldots, \Theta_n), \quad (5)$$

where each subset $\mathcal{V}_j$ of the partition π_n is a composite feature, and Θ_j is the parameters of the corresponding weak classifier. Given the training data $\mathcal{B}$, we define the posterior of $\mathcal{W}$ based on (1) as the following

$$p(\mathcal{W}|\mathcal{B}) \equiv \frac{1}{Z} \exp\left\{ -\sum_{i=1}^{m} L\left(y_i, F(I_i; \mathcal{W})\right) \right\}, \quad (6)$$

where Z is a normalization constant. The formulation in (6) has established an analogue between our problem and the image segmentation problem, for which GSWC has proved to be an efficient way of sampling around the peaks of the posterior. The correspondences between some key elements of the two problems include *basic feature* vs. *atomic region*, *composite feature* vs. *segment*, $\{\Theta_j\}$ vs. *region model parameters*, and discriminative $p(\mathcal{W}|\mathcal{B})$ vs. *generative segmentation posterior*, respectively. To complete the analogue, we still need to define the edge set $\mathcal{E}_0$ of the adjacency graph $\mathcal{G}_0$ – a problem discussed later in Sec. 4.4.

4.3. Discriminative GSWC

While there is no significant difference among the three GSWC algorithms proposed in [2], we choose SWC-3 for the simplicity and the theoretical guarantee that all proposals are accepted with probability one. The algorithm starts with an initial partition that in theory does not affect the sampling results in later stages. At each iteration it draws a random connected component $\mathcal{V}_0$ from a segment[2] $\mathcal{V}_l$ of the current partition $\pi = (\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_n)$. $\mathcal{V}_0$ is then merged with one of the following sets,

$$\mathcal{U}_1 = \mathcal{V}_1, \ldots, \mathcal{U}_l = \mathcal{V}_l \backslash \mathcal{V}_0, \ldots, \mathcal{U}_n = \mathcal{V}_n, \mathcal{U}_{n+1} = \emptyset. \quad (7)$$

The probability of merging $\mathcal{V}_0$ with $\mathcal{U}_l$ in (7) is given by

$$q(l|\mathcal{V}_0, \pi) = \frac{\omega_l p(\mathcal{W}_l|\mathcal{B})}{\sum_{j=1}^{n+1} \omega_j p(\mathcal{W}_j|\mathcal{B})}, \quad (8)$$

where $\mathcal{W}_j$ is the parameter set of a hypothetical partition π_j after merging $\mathcal{V}_0$ with $\mathcal{U}_j$, and ω_j denotes the weight for π_j, which is given by

$$\omega_j = \prod_{e \in \mathcal{C}_j} (1 - q_e), j = 1, 2, \ldots, m, \quad (9)$$

where $\mathcal{C}_j$ is the Swendsen-Wang Cut between $\mathcal{V}_0$ and $\mathcal{U}_j$, and q_e is the discriminative link probability (defined later in Sec. 4.4) between the nodes linked by the edge e. Once the merge is accomplished with the probability given in (8), a new sample of $\mathcal{W}$ is successfully drawn from the posterior.

Since the above algorithm is based on a posterior which is related to the discriminative function in (1), we call this new algorithm *Discriminative GSWC*, or DGSWC. Recall that $\mathcal{W}$ includes an n-partition entity, which in turn contains a set of composite features in our context. Since $\mathcal{W}$ is

[2]Segment and composite feature are interchangeable in this context.

drawn from a discriminative posterior function, the corresponding composite features have high probabilities of being class specific. We have therefore constructed an algorithm that draws class specific composite features with high probabilities. However, it still remains to specify q_e and the procedure to compute $p(\mathcal{W}_j|\mathcal{B})$.

4.4. Discriminative Link Probability and Neighborhood

The link probability q_e plays an important role in drawing the proposal segment $\mathcal{V}_0$ from the current partition. Suppose that $e =< v_i, v_j >$ is the edge that links two nodes in the graph $\mathcal{G}_0$, q_e is defined in [2] as the KL divergence of the histogram features of v_i and v_j. This definition is good for the segmentation purpose, but not applicable to our problem, in which the link probability should reflect the gain of discriminating power by combining two individual features as oppose to use them separately. More specifically, we define

$$q_e(on|v_i, v_j) = C \exp\left(\frac{-\beta(v_i, v_j)}{\min(\beta(v_i), \beta(v_j)) + \beta(v_i, v_j)}\right), \tag{10}$$

where $C = 0.632$ is a constant, and $q_e(on|v_i, v_j)$ denotes the probability of turning on e given v_i and v_j, $\beta(\{v_l\})$ represents the error rate when using features in $\{v_l\}$ for classification. The error rate is computed by applying linear SVM as the classifier.

We also use q_e to construct the adjacency graph $\mathcal{G}_0 =< \mathcal{V}, \mathcal{E}_0 >$, where two nodes are connected only if their q_e is larger than a threshold ϵ. Unlike the notion of "spatial neighborhood" used in segmentation, this essentially defines a concept of "discriminative neighborhood", which declares two nodes to be neighbor only if combining them together provides sufficiently stronger discriminating power than using them separately. Note that the graph topology is fixed after $\mathcal{G}_0$ is constructed.

4.5. Posterior Computation

Computing the posterior $p(\mathcal{W}_j|\mathcal{B})$ requires knowing the values of the parameters in $\mathcal{W}_j$, i.e., the partition π_j and the parameters $\{\Theta_i\}$. At each iteration of the Discriminative GWSC, when $\mathcal{V}_0$ is hypothetically merged with $\mathcal{U}_j$ in (7), we obtain a hypothetical partition π_j. Given this partition, we compute $\{\Theta_i\}$ using the AdaBoost algorithm, which iterates through all the composite features $\mathcal{V}_i$ in π_j, and builds weak classifiers f_i by computing α_i and Λ_i from the training data. It selects the most discriminative weak classifiers, and hence the corresponding features, through iterative reweighting of the training samples. As a result, the indicator parameter τ_i is set to one for the selected features, and to zero otherwise. Once all the parameters $\{(\tau_i, \alpha_i, \Lambda_i)\}$ are known, the additive model is evaluated according to (3), followed by the posterior computation as in (6). Note that the constant Z does not have to be computed because of the cancellation in (8).

5. Algorithm

Putting everything together, we provide the outline of the proposed algorithm learning an AdaBoost classifier with the class specific composite features sampled by the Discriminative GSWC. To simplify the discussion and focus on the big picture, we will delay the descriptions of some components to Sec. 6 dedicated to the implementation details. Noted that Algorithm 1 describes only a single layer of the cascaded classifier we actually use for object detection. The process of cascading such single layer classifiers follows the standard approach [20], which is not elaborated here for the simplicity of the discussion. AdaBoost algorithm is not detailed here for the same reason. To prevent confusion, also note that the AdaBoost algorithm used here serves different purpose from that of the AdaBoost used in Sec. 4.5, where the objective is to support the sampling of composite features, not to build a classifier for detection.

Algorithm 1 Learning AdaBoost Classifier with Discriminative GSWC

Input: Training samples $\mathcal{B} = \{(I_i, y_i)\}$, and a preference of the basic feature types.

1: Generate a set $\mathcal{V}$ of individual features with the basic feature types.
2: $\mathcal{L}$ = DGSWC_Sampler($\mathcal{B}$, $\mathcal{V}$), where $\mathcal{L}$ is a list of composite feature samples.
3: ($\mathcal{S}_1^*$, $\{\Theta_j^*\}$) = AdaBoost($\mathcal{B}$, $\mathcal{L}$), where $\mathcal{S}_1^*$ is the set of class specific composite features, selected from $\mathcal{L}$.

Output: An AdaBoost classifier that minimizes (1), with $\mathcal{S}_1^*$ and the optimal parameters set $\{\Theta_j^*\}$.

Subroutine: $\mathcal{L}$ = DGSWC_Sampler($\mathcal{B}$, $\mathcal{V}$)

4: $\mathcal{L} \leftarrow \emptyset$.
5: Compute q_e using (10) for all pairs of $< v_i, v_j >$ in $\mathcal{V}$, and construct an adjacency matrix M.
6: Remove entries with $q_e < \epsilon$ in M to build the graph $\mathcal{G}_0$.
7: **repeat**
8: Randomly flip all the edges according to q_e.
9: Draw a connect component $\mathcal{V}_0$ from a random $\mathcal{V}_k$ in the current partition π.
10: Merge $\mathcal{V}_0$ to $\mathcal{U}_l$ according to $q(l|\mathcal{V}_0, \pi)$ in (8).
11: Add $\mathcal{V}_k \backslash \mathcal{V}_0$ and $\mathcal{U}_l \cup \mathcal{V}_0$ to $\mathcal{L}$.
12: **until** Stop criterion met.

Return: The list $\mathcal{L}$.

As shown in Algorithm 1, other than taking a training data set as input, the algorithm also allows the choice of feature types. As a matter of fact, because the proposed framework is independent of specific feature types, the same algorithm can also work with a set of features with hetero-

geneous types, providing the potential of another level of feature composition. The steps from 1 to 3 are the major components of the algorithm, which use DGSWC_Sampler to generate samples from the composite feature space, induced from $\mathcal{V}$, and select the most discriminative features using standard AdaBoost. The steps from 4 to 12 lay out the DGSWC_Sampler subroutine, which corresponds to the sampling algorithm described in Sec. 4.3. The stop criterion in Step 12 is the same as in [2]. While the algorithm description seems rather simple, implementing it efficiently requires careful treatment for some of the key components.

6. Implementation Details

In the implementation of Algorithm 1, we choose Haar feature and HOG feature because of their popularity and the wide availability of published results using these two features. The individual feature set $\mathcal{V}$ in Step 1 is set to be the features selected by running a standard AdaBoost with large number of iterations. This helps removing the features that are not discriminative or highly overlapped. We use Intel's OpenCV implementation for selecting Haar features, and our own implementation for selecting HOG features.

To construct the adjacency matrix M in Step 5 and 6, we set the threshold ϵ to be 0.11, below which two nodes are not connected. Since most edge weights are below this threshold, we usually get a very sparse matrix. Storage memory is therefore not a problem. To avoid unnecessary computations, we also use some heuristics such as skipping a pair of nodes if they are spatially too close. Note here linear SVM is used to compute the edge weights. In Step 9, we simply use uniform distribution to select $\mathcal{V}_k$, from which $\mathcal{V}_0$ is generated. Choosing other proposal distributions is also possible, but does not have significant impact on the results.

In Step 10, the merge probability $q(l|\mathcal{V}_0, \pi)$ has to be computed for $n+1$ hypothetical partitions as described in (7) and (8) of Sec. 4.3. Fortunately, this seemingly expensive process can be performed efficiently because of the incremental nature of the AdaBoost algorithm. More specifically, each time $\mathcal{V}_0$ is merged with a hypothetical segment $\mathcal{U}_l$, only two segments $\mathcal{V}_k$ and $\mathcal{U}_l$ in the current partition need to be changed. Therefore we can reuse most parameters of the AdaBoost, used to compute the posterior for the previous hypothetical merge, to compute the current merge probability. Suppose the previous AdaBoost has converged at the tth step, the incremental AdaBoost process is as simple as an additional iteration at $t+1$ with all the previous parameters, and adding the newly changed segments as two weak classifiers. The AdaBoost parameters are then updated accordingly and the probability for the current hypothetical merge is computed, as described in Sec. 4.5.

Figure 3. Samples from the database of people in general poses and viewing aspects.

7. Experiments

We test our algorithm on a public people database provided by [4]. The database contains both training and testing images with people in general poses and viewing aspects, of which some samples are shown in Fig. 3. It provides 2416 64x128 people images, which we downscale to the size of 24x48 to form the positive training set. The database provides 1218 negatives images, from which random patches are sampled during the training process. The final detection results are reported based on the testing images used in [4] and [16], except that we downscale the images to the size of 320x240. For each test image, we scan through 4800 windows, which takes Haar feature-based AdaBoost on average 71 ms, and HOG-based AdaBoost 223ms. These numbers are comparable with the numbers reported in [21]. Since [21] represents the current state-of-art, we use their results as our reference point for all the performance comparisons. In the experiments, DGSWC generates $\sim$ 1000 candidate composite features for AdaBoost to learn the strong classifier.

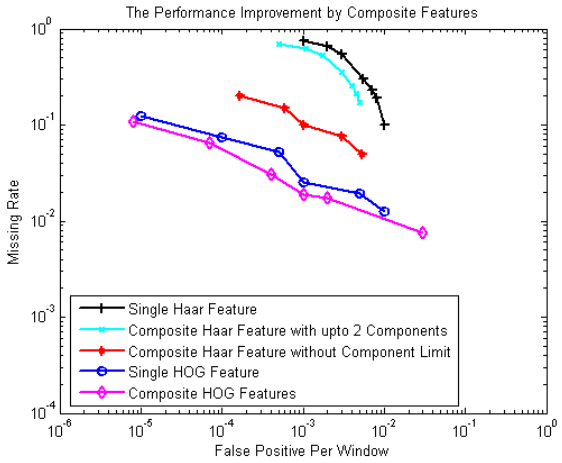

Figure 4. Performance comparison.

In the first experiment, we evaluate how the composite feature idea can improve the Haar feature results. We first run our algorithm by limiting the number of constituting

features in the composite feature to be 2, and then remove this limit in the second experiment. The performance curves for these two experiments are shown in Fig. 4, together with the result using single Haar features from [21]. It can be seen from the figure that using composite features with two components improves the performance, but not as dramatic as the full strength composite features.

In the second experiment, we evaluate how the composite feature idea can improve the HOG feature result. The performance of our algorithm with composite HOG features is shown in Fig. 4, together with the result of using single HOG features from [21]. The improvement in this case is less significant than in the previous experiments. This is expected, because HOG is a much stronger feature that already encodes a certain level of spatial relationships in its representation. However, as compared with the improvement of [21] against Dalal and Triggs' results (see [21] for details), our improvement is on both the low-end and the high-end of false alarm rates, and is therefore more consistent.

Fig. 5, Fig. 6, Fig. 7, and Fig. 8 exemplify some typical detection results under different poses and various environmental conditions. These results are generated with our HOG-based implementation.

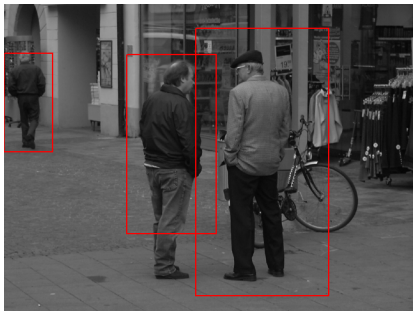

Figure 5. Detection of three back view people in various poses.

Figure 6. Detection of one side view walking people and one frontal view people at far distance.

Fig. 9 shows an example of the adjacency graph. It can be seen that features far apart from each others have better chances of being connected. This is expected since these feature pairs are likely to be more discriminative than nearby features.

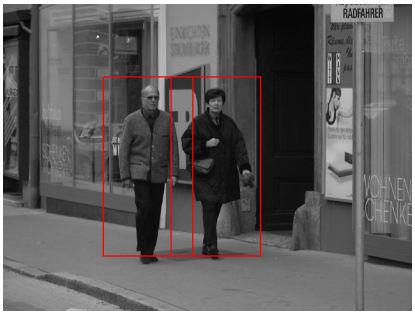

Figure 7. Detection of two frontal view walking people.

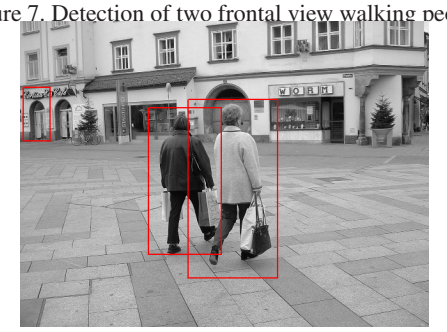

Figure 8. Detection of two back view walking people, with one false alarm of a door at far distance. The door looks somewhat like a person.

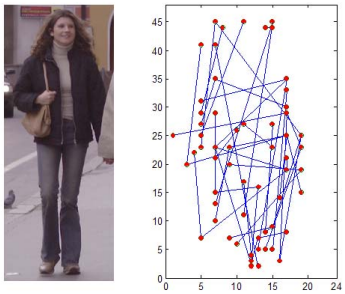

Figure 9. Example adjacency graph. Only a portion of the full adjacency graph is shown here for clarity.

Fig. 10 shows the top three composite Haar features, where the top feature looks especially meaningful.

Figure 10. Top three composite Haar features automatically discovered. The top one composite feature, shown in the second left image, indeed looks like a person with two legs. For the second composite feature, one component lands on the foot, and the other lands on a leg. These are all important areas for people detection. Note that only the bounding boxes of the Haar features are displayed for clarity.

8. Conclusion and Future Work

We have developed a DGSWC algorithm that allows AdaBoost searching through a much larger and more discriminative feature space, making the detection of complex objects a more solvable problem. DGSWC algorithm exemplifies the adoption of generative sampling algorithms for the purpose of discriminative sampling. The same methodology can be applied to other applications such as class specific segmentation, where discriminative sampling can potentially play a significant role.

Upgrading to composite feature space and choosing discriminative features using DGSWC is not only a feature selection process, but also a process of new feature discovery. With exactly the same mechanism, but much lower level individual features such as a simple patch, we can automatically design class specific Haar or HOG features as the composite features of these simple features. It replaces the traditional time-consuming and error-prone processes that require certain levels of human involvement for feature designing.

Our approach works with any types of features as the basic features, and provides a unified platform for integrating heterogeneous features. However, it does not solve the problem of how to choose the types of the features. Another problem of the current approach is that it does not have a prior term that controls the complexity of the composite features. This is one of our future directions to improve the performance.

References

[1] A. Agarwal and B. Triggs. Hyperfeature - multilevel local coding for visual recognition. In *Proceedings European Conference on Computer Vision (ECCV06)*, 2006.

[2] A. Barbu and S. Zhu. Generalizing swendsen-wang to sampling arbitrary posterior probabilities. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 27(8):1239–12531, 2005.

[3] E. Borenstein and S. Ullman. Class-specific, top-down segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV02)*, 2002.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR05)*, 2005.

[5] B. Epshtein and S. Ullman. Feature hierarchies for object classification. In *Proceedings International Conference on Computer Vision (ICCV05)*, 2005.

[6] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision (IJCV)*, 61(1):55–79, 2005.

[7] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR03)*, 2003.

[8] Y. Freund and R. E. Schapire. A decision-theoretic generalization of online learning and an application to boosting. *Journal of Computer and System Science*, 55, 1997.

[9] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 43:177–196, 2001.

[10] M. P. Kumar, P. Torr, and A. Zisserman. OBJCUT. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR04)*, 2004.

[11] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Learning layered pictorial structures from video. *ICVGIP*, pages 148–153, 2004.

[12] B. Leibe and B. Schiele. Interleaved object categorization and segmentation. In *BMVC*, 2003.

[13] A. Levin and Y. Weiss. Learning to combine bottom-up and top-down segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV06)*, 2006.

[14] B. Ommer and J. Buhmann. Learning compositional categorization models. In *Proceedings European Conference on Computer Vision (ECCV06)*, 2006.

[15] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment-model for object detection. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR06)*, 2006.

[16] C. Papageorgiou and T. Poggio. A trainable system for object detection. *International Journal of Computer Vision (IJCV)*, 38(1):15–33, 2000.

[17] R. E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

[18] R. Swendsen and J. Wang. Nonuniversal critical dynamics in monte carlo simulations. *Physical Rev. Letters*, 58(2):86–88, 1987.

[19] Z. Tu and S. Zhu. Image segmentation by data-driven markov chain monte carlo. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 24(5):657–673, 2002.

[20] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR01)*, 2001.

[21] Q. Zhu, S. Avidan, M.-C. Yeh, and K.-T. Cheng. Fast human detection using a cascade of histograms of oriented gradients. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR06)*, 2006.