CONTEXTUAL DISAMBIGUATION FOR MULTI-CLASS OBJECT DETECTION

Xiaodong Fan

Dept. of Electrical and Computer Engineering
The Johns Hopkins University
Baltimore, MD 21218
xdfan@cis.jhu.edu

ABSTRACT

We consider the problem of detecting and localizing instances from multiple object classes. Suppose an overcomplete index - an initial list with extra detections but none missed - is provided. We and others have previously shown how this can be done efficiently with coarse-to-fine search. How would one prune such a list to a final interpretation? We propose a method based on contextual disambiguation: First, a Viterbi algorithm is utilized to extract N candidate interpretations by using the global context to provide constraints among object classes or poses. Then, the extracted candidates are compared in a pairwise fashion to resolve remaining ambiguities, and the final interpretation is constructed. The whole procedure is illustrated by experiments in reading license plates.

1. INTRODUCTION

Our objective is to detect and localize instances from multiple object classes in cluttered grey level images. This is more general than detecting instances from a generic object class such as faces, cars and pedestrians.

In our recent work [1, 2] we divide the solution to the whole problem into two steps, where indexing, in the sense of non-contextual detection, primes global scene interpretation. First compile a list – an index – of candidate instances (object identities and poses) under a zero false negative constraint, but at the expense of numerous false positives. Then exploit contextual information, e.g., constraints among object instances, to resolve ambiguities and extract a final interpretation from the index. Such a division is motivated by computational efficiency since the intensive processing is limited to a reduced portion of the scene.

This paper focuses on the second step – a systematic method to prune the initial overcomplete index to the final interpretation. The method presented here integrates contextual analysis and ambiguity resolution: First, an objective function is defined over all possible subsets of the index, accounting for the confidence of individual detections in the subset as well as their agreement with global contextual constraints among object classes and poses. High scoring subsets correspond to globally consistent interpretations extracted from the index. A generalized Viterbi algorithm is utilized to search for the N highest scoring subsets as candidate interpretations. Such a process is called "contextual analysis". Then, the extracted candidates are compared in a pairwise fashion to resolve remaining ambiguities, and the final interpretation is constructed.

We start with a formulation of scene interpretation and summarize hierarchical object indexing in §2. Section 3 mentions some related work in object indexing and contextual disambiguation. Contextual analysis and disambiguation by pairwise competition are described in sections §4 and §5 respectively. Finally, experiments on reading license plates and conclusions are given in §6 and §7.

2. SCENE INTERPRETATION

The goal of multi-class object detection is to compile a list of object instances (c, θ), where $c \in \mathcal{C}$ indicates the *class* or *category* of the object and $\theta \in \Theta$ represents its *pose* or *presentation* in the scene.

Let $\mathcal{Z} = \mathcal{C} \times \Theta$ be the set of all possible object instances and let $Y \subset \mathcal{Z}$ denote the true list of instances, which is our goal of scene interpretation. Indexing means identifying a subset $\mathcal{D} = \mathcal{D}(I) \subset \mathcal{Z}$ where I is the image, under the constraint that Y belongs to $\mathcal{D}$. Pruning the index $\mathcal{D}$ by contextual disambiguation simply means reducing it to Y.

Recently, we proposed an efficient indexing algorithm to search $\mathcal{Z}$ in a coarse-to-fine manner based on its hierarchical decomposition; see [2] for details. Starting with $\mathcal{Z}$ itself at the root, the subset or "cell" of $\mathcal{Z}$ at each internal node t is denoted as $\mathcal{Z}_t$. The cells at the leaves are pure in class but may contain poses within some range. There is also a binary classifier $X_t \in \{-1, 1\}$ at each node with $X_t = 1$ indicating $Y \cap \mathcal{Z}_t \neq \emptyset$. We assume that the image I is represented by a fixed family of *binary* features $\{x_j\}$; in our experiment we use the local operators as in [3] to extract

The author thanks Prof. Donald Geman for many helpful suggestions.

0-7803-8554-3/04/$20.00 ©2004 IEEE.

$\{x_j\}$. X_t is a linear test relative to a distinguished subset $\mathcal{J}_t$ of binary image features:

$$X_t = sign\left(\sum_{j\in\mathcal{J}_t} \lambda_t(j)x_j - \tau_t\right) \quad (1)$$

X_t is applied *if and only if* all the tests at the ancestors of t are positive. The surviving leaves are the ones whose tests are applied with positive responses. Each surviving leaf contributes a candidate object instance to the index $\mathcal{D}$ with a unique class whose pose is determined up to the resolution of its cell.

This strategy is especially efficient since the explanation "background" is usually statistically dominant; hence the search terminates quickly along most paths with high probability. But some paths survive due to false positive error, and contextual disambiguation must be invoked to filter these false alarms to reduce $\mathcal{D}$ to Y.

3. RELATED WORK

Various methods exist for object indexing [4, 6, 11, 12]. Our hierarchical algorithm, as well as the two-step strategy for object detection, was first proposed in [1]. In constrast with [1], the tree hierarchy used here to search for $\mathcal{D}$ is generated *automatically* (forthcoming paper), and the binary classifier X_t at each node is defined and sequentially refined as in [2]. For contextual disambiguation, there is no general method in [1] to utilize global context as priori knowledge about constraints among object instances; only a specific example is shown in an application to reading license plates. Such a problem is addressed in this paper.

Our contextual analysis process is different from the contextual priming in [10], in which the holistic image features are used as global context. Instead, we explore the intercorrelations among object instances. The spatial context model in [8] and the partial context model in [9], which model the constrains among object classes and their geometric layout, are related to ours. Our method differs from them in the way of modeling these constrains, of searching for a globally optimal interpretation consistent with these constrains and of resolving ambiguities that arise from multiple classifiers.

4. CONTEXTUAL ANALYSIS

Contextual analysis is to extract globally consistent interpretations (subsets of $\mathcal{D}$). Therefore, we first introduce an objective function as a consistency measurement, and then propose an algorithm to search for high scoring subsets of $\mathcal{D}$ as candidate interpretations.

4.1. The objective function

Obviously, were Bayesian inference to be used, the natural objective function would be the a posteriori probability $P(y|I)$, where y denotes a subset of Z representing a hypothesized interpretation. Such a method involves a probabilistic image model. However, the model we used in [2] to derive the test X_t is very crude and is based on the conditional independence assumption of the image features (hence somewhat unrealistic).

Therefore we propose a heuristic objective function:

$$F(y;I) = \frac{1}{k}\sum_{i=1}^{k} s(c_i,\theta_i;I) + \log\pi(y)$$

where $(c_1,\theta_1),\ldots,(c_k,\theta_k)$ are object instances in y, and k is the number of instances in the scene, a random variable. $s(c,\theta;I)$ is the *detection score* measuring the confidence of detecting an instance (c,θ) and $\pi(y)$ is the priori probability of the joint appearance of all object instances in y, which models the global constraints among object classes or poses. (Also incorporated is a hard-wired constraint that all object instances in any *valid* interpretation are non-overlapping.)

The detection score measures how well a detected instance fits the local image data. For this purpose, at each node t in the tree hierarchy for object indexing, we are interested in estimating the probability of $Y \cap \mathcal{Z}_t \neq \emptyset$ rather than merely making a hard decision. We achieve this by assuming $P(Y \cap \mathcal{Z}_t \neq \emptyset)$ is proportional to the following logistic function:

$$\phi\left(\sum_{j\in\mathcal{J}_t} \lambda_t(j)x_j - \tau_t\right)$$

where $\phi(z) = (1+e^{-z})^{-1}$. To calculate the detection score, we take the product of such probability estimates at *all* nodes along the branch leading to the corresponding leaf in the hierarchy.

Modeling $\pi(y)$ is more complicated because the number of object instances is data-dependent. In this paper, only a much simplified case is considered, based on the following assumptions:

Geometric Layout: We assume that object instances are approximately located on a line (denoted as L; see Figure 1(b)) whose orientation can be coarsely estimated. Such a geometric layout is satisfied in applications involving license plates or other OCR problems such as recognizing a text line.

Markovity: Examining the object instances in $\mathcal{D}$ according to their projections on L, we assume *any* two instances are *conditionally independent* given a third instance whose projection on L lies between them. (As we only consider the projections on L, only the orientation, not the position, of L is needed. In Figure 1(b) for instance, v_1 and v_3 are conditionally independent given v_2.) Equivalently, the sequence $\{(c_1,\theta_1),\ldots,(c_k,\theta_k)\}$ forms a Markov chain under π conditional on k, assuming their labels are ordered properly based on their projections on L and a pre-defined direction to scan L (e.g., from left to right were L to be horizontal). (For instance, in Figure 1(b), the detections labeled

as $v_1 \sim v_4$ form a Markov chain.) Hence,

$$\pi(y|k) = P((c_1, \theta_1)) \prod_{i=2}^{k} p((c_i, \theta_i)|(c_{i-1}, \theta_{i-1})) \quad (2)$$

where $p((c_i, \theta_i)|(c_{i-1}, \theta_{i-1}))$ is the transition probability. (See §6 for an example.) The full prior model is $\pi(y) = \pi(y|k)\nu(k)$, where $\nu(k)$ reflects the domain specific knowledge of the number of instances in the scene. Here, for simplicity we assume $\nu(k)$ is uniform, and therefore $\pi(y) \propto \pi(y|k)$.

4.2. N-Best search

As we use a heuristic objective function, the highest scoring interpretation may not be Y (the truth). Therefore, we first search for the N highest scoring interpretations with a fairly large N (set to 20 in our experiment) so that true object instances in Y are likely to be included.

To achieve this, we introduce a trellis (directed graph) representation of candidate object instances in $\mathcal{D}$, and formulate extracting interpretations as searching for connected paths on the trellis.

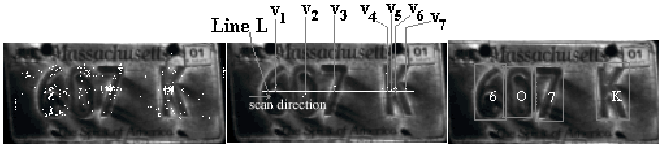
To construct the trellis, each instance in $\mathcal{D}$ is represented as a vertex in a graph. Two vertices are connected *if and only if* their corresponding object instances satisfy: (i) They are *non-overlapping*; (ii) They are *adjacent* in the sense that there is no object instance non-overlapping with each of them, whose projection on L lies between them. There is a directed arc between two connected vertices, whose direction follows the pre-defined direction to scan L. Furthermore, we define a virtual starting vertex v_s which has an out-going arc to each vertex with no in-coming arcs. Similarly, a virtual ending vertex v_e is also defined. Each connected path from v_s to v_e represents a valid interpretation extracted from D and contains no overlapping instances. (See Figure 2(a) for an illustration of a trellis for a reduced index containing detections $v_1 \sim v_7$ in Figure 1(b).)

The arc connecting two non-virtual vertices v_i and v_j (from v_i to v_j) is associated with a weight: $\log(p(c_j, \theta_j)|(c_i, \theta_i))$, where $(c_{i(j)}, \theta_{i(j)})$ is the corresponding object instance. Using the decomposition in (2), calculating $\log \pi(y)$ for a valid interpretation corresponds to summing all the arc weights along the path, so the objective function $F(y; I)$ can be easily computed (given the detection scores at all non-virtual vertices).

N-Best search refers to extracting N highest scoring paths on the trellis from v_s to $v_e$via a generalized Viterbi algorithm. The only difference from the standard Viterbi algorithm is that, for each vertex in the trellis, the N highest scoring paths leading into it are recorded. (See [5] for details.)

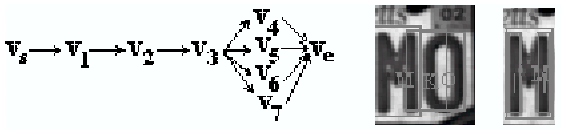
5. PAIRWISE COMPETITION

Pairwise competition is to compare the candidate interpretations selected above, and construct the final interpretation.



(a) initial detections (b) after post-processing (c) final interpretation

Fig. 1. Selected detections, represented by the white dots.



(a) (b) (c)

Fig. 2. (a): Illustration of a trellis; (b), (c): Examples of pairwise competitions. In (b), besides "M" and "0", a "K" is also detected that straddles over them ($u = \{M, 0\}$ and $u' = \{K\}$). In (c), Two vertical bars ("|") are detected at the left and right sides of "M" ($u = \{M\}$ and $u' = \{|, |\}$).

Constructing Competitions: We proceed pairwise competitions sequentially. To begin with, let y and y' be the highest and second highest scoring interpretations among N candidates respectively. After comparing y with y', y is modified to a "better" interpretation. Then at the next step, let y' represent the third highest scoring candidate. Generally, at the i-th step, set y' to the $(i+1)$-th highest scoring interpretation and continue modifying y by comparing it with y'. The process continues until the lowest scoring candidate interpretation is compared with, and y is the final interpretation of the scene.

To compare y with y', pairs of different but spatially overlapping subsequences of object instances (u, u') ($u \subset y$ and $u' \subset y'$) are extracted from y and y'. Each pair (u, u') represents a pair of conflicting sub-interpretations. In most cases, both u and u' contain only one object instance, so that actually two individual detections are compared. (Referring to the trellis in Figure 2(a), in the comparison of two interpretations $\{v_1, v_2, v_3, v_4\}$ and $\{v_1, v_2, v_3, v_5\}$, $u = \{v_4\}$ and $u' = \{v_5\}$.) In other cases, the subsequences may contain more than one object instances. But this is rare and only occurs when detections are found that straddle two real object instances (Figure 2(b)) or split one real object instance into two or more objects (Figure 2(c)); such cases can often be reduced by exploiting some simple pruning mechanisms [2] and contextual analysis as in §4.

For each pair of subsequences (u, u'), a binary classifier $X_{u,u'} \in \{-1, 1\}$ learned *online* is applied to resolve the ambiguity. The subsequence u in y will be replaced by u' if $X_{u,u'} = -1$. (By doing this, we assume that such a replacement will not violate the global constraints among object instances in y.)

Likelihood Ratio Test: We fix a pair (u, u') and learn the binary classifier $X_{u,u'}$. The one we use is a linear test rel-

ative to a distinguished subset $\mathcal{J}$ of features. Actually, it is a likelihood ratio test between u and u', were features in $\mathcal{J}$ to be independent conditional on two hypotheses. The test has the same form as (1), where

$$\lambda(j) = \log \frac{p(j)(1-q(j))}{q(j)(1-p(j))}$$

$p(j) = P(x_j = 1|u)$ and $q(j) = P(x_j = 1|u')$ are estimated from $\mathcal{L}^+$ and $\mathcal{L}^-$ respectively, where $\mathcal{L}^+$ and $\mathcal{L}^-$ are two sets of training samples belonging to u and u'.

Learning $X_{u,u'}$ involves constructing $\mathcal{J}$ and estimating τ. We fix $|\mathcal{J}|$ (set to be 20 in our experiment) and add features to $\mathcal{J}$ sequentially. At each step, the one that minimizes the empirical risk or maximizes the empirical margin between two hypotheses (in cases of zero empirical risk) is selected. The empirical risk is the error of $X_{u,u'}$ estimated from training samples. And the empirical margin is defined as:

$$\min_{I\in\mathcal{L}^+}\{\sum_{j\in\mathcal{J}} \lambda(j)x_j(I)\} - \max_{I\in\mathcal{L}^-}\{\sum_{j\in\mathcal{J}} \lambda(j)x_j(I)\}$$

τ is chosen to minimize the empirical risk or simply the mid point of the empirical margin in cases of zero empirical risk.

6. READING LICENSE PLATES

Our objective is to identify the characters on the license plate from the photograph of the rear of a car, as shown in Figure 3. The plates displayed in Figure 3 demonstrate the challenges due to variations in stroke widths, variable illumination, background clutters and other effects.

Fig. 3. A typical photograph and samples of extracted plates.

There are 37 object classes – 26 letters, 10 digits and one special symbol, a vertical bar. The pose parameters specify the location, scale and orientation of character instances.

Figure 1(a) shows the initial detections in $\mathcal{D}$ by hierarchical non-contextual indexing. A simple pruning mechanism reduces it to 1(b); on average there are around 40 detections per plate in $\mathcal{D}$. (See [2] for details.)

Because characters on the plates have approximately the same scales and tilts, and are located on a horizontal line (assuming the tilts are small), the transition probability between two adjacent object instances (c_i, θ_i) and (c_j, θ_j) for contextual analysis is defined as the product of three zero-mean Gaussians. Taking the log, it then becomes:

$$C - \frac{(u_j-u_i)^2}{2s_1^2} - \frac{(\sigma_j-\sigma_i)^2}{2s_2^2} - \frac{(\rho_j-\rho_i)^2}{2s_3^2}$$

, where C is a constant and u, σ and ρ are the vertical component of the location, the scale and tilt of the pose parameter θ. s_1, s_2 and s_3 are three empirically set variances. (We take $s_1^2 = 4, s_2^2 = 0.2, s_3^2 = 2$.)

The final interpretation is shown in Figure 1(c). We test our algorithm on 380 plates and the classification rate per symbol is 99.3%.

7. CONCLUSION

We apply contextual disambiguation to prune an initial index to the final scene interpretation. To search for globally consistent interpretations, a much simplified case is considered so that the 1D Markov model and Viterbi algorithm can be used. We believe this method can be generalized to a 2D Markov model for more complex spatial configurations.

8. REFERENCES

[1] Y. Amit, D. Geman and X. Fan, "Computational strategies for model-based scene interpretation," Technical report, The Johns Hopkins University, 2003. http://www.cis.jhu.edu /∼xdfan/strategies.pdf

[2] X. Fan and D. Geman, "Hierarchical object indexing and sequential learning," *Proc. ICPR'04*

[3] F. Fleuret and D. Geman, "Coarse-to-fine face detection," *Inter. Journal of Computer Vision*, vol. 41, pp. 85-107, 2001.

[4] D.M. Gavrila, "Multi-feature hierarchical template matching using distance transforms," *Proc. ICPR'98*, 1998.

[5] F. Jelinek, *Statistical methods for speech recognition*, The MIT Press, Cambridge, Massachusetts, 1998.

[6] B. Lamiroy and P. Gros, "Rapid object indexing and recognition using enhanced geometric hashing," *Proc. ECCV'96*, pp. 59-70, 1996.

[7] S. Mahamud and M. Hebert, "The optimal distance measure for object detection," *Proc. CVPR'03*, pp. I: 248-255, 2003.

[8] A. Singhal, J. Luo and W. Zhu, "Probabilistic spatial context models for scene content understanding," *Proc. CVPR'03*, pp. I: 235-241, 2003.

[9] X. Song, Y. Abu-Mostafa, J. Sill and H. Kasdan, "Image recognition in context: application to microscopic urinalysis", *Proc. NIPS'99*, 1999.

[10] A. Torralba and P. Sinha, "Statistical context priming for object detection," *Proc. ICCV'01*, pp. I: 763 -770, 2001.

[11] V.C. de Verdi'ere and J.L. Crowley, "Object indexing using local appearance," *Proc. ECCV'98*, 1998.

[12] O. Yamaguchi and K. Fukui, "Pattern hashing: distributed appearance model using local images and invariant indexing," *IPSJ Transactions on Computer Vision and Image Media*, vol. 44, no. SIG05, 2003