FINDING FAMILIAR OBJECTS AND THEIR DEPTH FROM A SINGLE IMAGE

Hwann-Tzong Chen

Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan 30013

Tyng-Luh Liu

Institute of Information Science
Academia Sinica
Taipei, Taiwan 11529

ABSTRACT

We present a classification-based method to identify objects of interest, and judge their depth in a single image. Our approach is motivated by a postulate of human depth perception that people can give a credible depth estimation for an object whose familiar size is known, even without using stereo vision. To emulate the mechanism, we categorize objects into the same class if they have similar sizes and shapes, and model the sense of discovering a familiar object by applying *multiple kernel logistic regression* to the conditional probability of feature types. The depth of a detected target can then be obtained by referencing its corresponding object category. Overall, the proposed algorithm is efficient in both the training and testing phases, and does not require a large amount of training images for good performances.

Index Terms— Machine vision, object detection, pattern classification, feature extraction

1. INTRODUCTION

We investigate the problem of finding objects and estimating their depth from a single image. Previous studies have suggested that although humans heavily rely on stereo vision to construct the 3-D structures of the real world, we can estimate very well the depths in a scene based on only monocular information [1]. This is quite evident because when people look at a picture, which contains only 2-D information, we could still "perceive" the 3-D geometric information present in the image. The ability is mostly owing to our abundant knowledge about things we have observed and have been learning. In this work, we shall specifically consider the following monocular (or pictorial) cues, including familiar size, relative size, and occlusion. Examples to illustrate the usefulness of relative size as a depth cue are shown in Figure 1.

Approaches to depth estimation mainly focus on directly analyzing stereo vision information for computing disparities and depths, or using other cues such as optical flow or blur for *depth-from-motion* or *depth-from-defocus*. Methods of this kind can derive accurate estimations, but require two or more images for computing the cues. In many vision applications, a coarse depth estimation often provides already quite useful information. It is therefore constructive to explore the possibility for having an efficient and reliable way of approximating the object depths from a single image. Torralba and Oliva [2] show that the absolute depth information can be recovered from the image structure cue. They report impressive results on computing the mean depth of a given scene. However, without resolving the difficulties of segmentation and object recognition, their method can not make the most of the typical size information of familiar objects. Michels *et al*. [3] also find depths from single monocular images, and use the depth estimations in a real-time system. Hoiem *et al*. [4] instead focus on estimating the geometric properties of a scene by learning region-based appearance models of geometric classes, including sky, ground, and vertical.

Fig. 1. Relative size between familiar objects is a useful cue to depth. For objects of similar sizes, those that appear smaller in an image should be further away from the camera.

Our method finds balance between the effectiveness and the complexity of a learning scheme. We first establish a library of image templates, corresponding to familiar objects of different categories (assuming their depth values are known in advance). To detect objects of interest and estimate their depth in a given image, we adopt the *multiple kernel logistic regression* (MKLR) [5], [6], [7] to model the multiclass probabilities of salient features, namely the SIFT descriptors, *e.g.*, [8], [9] . Kernel logistic regression has been well studied in statistics and machine learning [7], [10], and its related techniques have been shown to yield comparable performance to SVMs. Performing MKLR on an interest point would yield a hypothesis of detecting a familiar object, and its depth can be estimated from the library. The final outcome can then be derived by validating and grouping these hypotheses.

2. LEARNING OBJECTS OF MULTIPLE CLASSES

Suppose we have an image set containing C categories of object templates. Each image is cropped to roughly enclose an object of interest such that those in the same category are depicted with a similar image size. For the ease of implementation, we use the last category to include all background templates, even though they do not contain specific objects and neither do they have similar scales. We run SIFT on all templates to detect interest points $\{\mathbf{f}_i\}_{i=1}^N$ and compute the corresponding descriptors $\{\mathbf{x}_i\}_{i=1}^N$, where N is the total number of interest points extracted from all the templates. Note that we also keep a mapping $t(i)$ to indicate that interest point i comes from template t. As usual, a descriptor $\mathbf{x}_i$ is a 128-dimensional vector encoding the local gradient orientations. And each interest point $\mathbf{f}_i$ is represented by a four-dimensional vector consisting of the location, scale, and orientation. By labeling each feature descriptor with the category of the template from which the feature is extracted, we have $D = \{(\mathbf{x}_i, y_i)|i = 1, \ldots, N\}$ as training samples, where $\mathbf{x}_i$ is the feature vector and $y_i \in \{1, \ldots, C\}$ is its label.

With MKLR, we are to learn from the training samples the conditional probability $P(y|\mathbf{x})$. Thus, given a detected interest point, we can use its feature descriptor to predict the label by referencing $P(y|\mathbf{x})$. After predicting the labels of all the interest points in a test image, we combine the predictions to identify possible locations of familiar objects and their scales in the current image.

2.1. Multiple kernel logistic regression

Given the training data D, the conditional probabilities for predicting the category label are parameterized by the C latent real-valued functions $\{h^c(\mathbf{x})\}_{c=1}^C$ of MKLR based on the multiple logistic likelihood:

$$P(y = c|\mathbf{x}, \boldsymbol{\theta}) = \frac{e^{h^c(\mathbf{x})}}{\sum_{c'} e^{h^{c'}(\mathbf{x})}}, \tag{1}$$

where

$$h^c(\mathbf{x}) = \sum_i \alpha_i^c K^c(\mathbf{x}, \mathbf{x}_i) + b_c \tag{2}$$

consists of the kernel expansions, and $\boldsymbol{\theta}$ denotes the set of parameters α_i^c and b_c. One way to learn the parameters is to directly maximize the likelihood in (1). However, the scheme usually leads to overfitting such that the learned conditional probabilities may not generalize well. We consider a more robust MKLR model by adding to (2) the following regularization terms

$$\sum_{i,j} \alpha_i^c \alpha_j^c K^c(\mathbf{x}_i, \mathbf{x}_j) \tag{3}$$

to penalize kernel functions of high complexity. The task of learning $\boldsymbol{\theta}$ can now be formulated as a *maximum a posteriori* (MAP) approximation, and solved by optimization techniques [6]. Indeed MKLR can be a very efficient learning scheme. In Seeger's implementation [6], the incomplete Cholesky decomposition is used for the low-rank approximation of the kernel matrix such that the time complexity of optimization becomes linear in the number of training samples.

3. DETECTING FAMILIAR OBJECTS AND DEPTHS

In general running SIFT on a hundred templates is sufficient to give an ample amount of feature vectors for training, say $10,000$ features. We do not perform vector quantization or clustering on the feature vectors to obtain visual words, *e.g.*, [9], [11]. Instead, we use all the feature vectors to generate the training samples for MKLR. This would give rise to a large training set $D = \{(\mathbf{x}_i, y_i) \,|\, i = 1, \ldots, N\}$ with $N \simeq 10,000$. Nonetheless, since we apply low rank approximation to the kernel matrix, learning the MKLR parameters is still fast and feasible—be reminded that the time and memory requirements for training are linear to N.

3.1. Making hypotheses on familiar objects

Given a new test image, we shall generate hypotheses about all possible locations and scales of the familiar objects that may appear in the scene. This can be efficiently done by carrying out the following steps.

Step 1. Detect interest points from a given image, and then use MKLR to predict each feature vector's label. Since the remaining procedure is universal, it suffices to simply look at how an interest point is processed. Consider now some interest point $\bar{\mathbf{f}}$ and its feature vector $\bar{\mathbf{x}}$. We compute the conditional probability $P(\bar{y}|\bar{\mathbf{x}})$ and predict the label as c if $P(\bar{y} = c|\bar{\mathbf{x}})$ has the largest value.

Step 2. Use the kernel expansions to make hypotheses on the location and the relative size of an object. Based on the predicted label c, we find the index i^* such that

$$i^* = \underset{i}{\operatorname{argmax}}\ \alpha_i^c K^c(\bar{\mathbf{x}}, \mathbf{x}_i), \tag{4}$$

and then pick the corresponding interest point $\mathbf{f}_{i^*}$ in the training samples. (If the label y_{i^*} of $\mathbf{f}_{i^*}$ is not c, which is very unlikely, we drop this interest point and go on processing the remaining interest points.) The rationale behind looking into the kernel expansions for generating hypotheses is that for MKLR every feature vector in D has certain influence over the conditional probabilities of labels. We naturally choose the one exerting the strongest influence to the making of a label prediction.

Step 3. Recall that each interest point $\mathbf{f}_i$ keeps the information of its location, scale (σ_i), and orientation. We compute the scale ratio $\bar{\sigma}/\sigma_{i^*}$ between $\bar{\mathbf{f}}$ and $\mathbf{f}_{i^*}$. Because we know that $\mathbf{f}_{i^*}$ comes from the template $t(i^*)$, this ratio provides the relative size of the target to the template. According to the ratio and the relative location of $\mathbf{f}_{i^*}$ in the template $t(i^*)$, we

can determine a bounding box in the test image to enclose a possible familiar object having the interest point $\bar{\mathbf{f}}$. Therefore, the image patch within the bounding box is considered similar to the one present in $t(i^*)$, and $\bar{\mathbf{f}}$ gives us a hypothesis on the location and relative size of the object in the test image.

3.2. Speeding up MKLR with cover trees

So far we have discussed how hypotheses about familiar objects are made by performing MKLR over interest points, and realized that the kernel expansions play an important role in determining the conditional probabilities on labels. Typically, the learned coefficients α_i^c in an MKLR model are all nonzero. It follows that to make a single prediction in MKLR, one needs to compute all the terms of the kernel expansions, and therefore the complexity is linear to N. Concerning this computational cost, Zhu and Hastie [7] propose a forward selection algorithm to find a subset of training samples such that the submodel can approximate well the full model. However, we prefer not to throw away the feature vectors because they are extracted from only a small number of templates and the SIFT descriptors are generally quite informative.

Still the MKLR classification can be further speeded up. By re-examining the kernel expansions $\sum_i \alpha_i^c K^c(\mathbf{x}, \mathbf{x}_i)$, one observes that it should be harmless to skip the calculations of those $K(\mathbf{x}, \mathbf{x}_i)$ that are close to zero. In our implementation we use a fast approximation scheme suggested in Shen *et al*. [12] to accelerate the testing. Specifically, we build the *cover tree* structure [13] to store the training samples for fast finding nearest neighbors. Since the RBF kernels are used in our MKLR, we only need to find the k nearest neighbors of a test vector, and then compute the kernel expansions involving these k feature vectors. Using the cover tree approximation, the cost of making a single prediction becomes log-linear to the sample size N.

3.3. Fusing comparable hypotheses

To integrate the hypotheses for locating the objects and for obtaining the depth estimations, we consider a simple fusion estimator via *kernel smoothing* and *mode seeking*. For each object class, we apply (fixed-bandwidth) kernel smoothing to each component of the hypotheses, *i.e*., the location, width, height, and the scale ratio. We then find the significant modes as the estimations for these components. Prior knowledge and heuristics such as overlapping, impossible size, or restricted area can also be used to remove strong but biased hypotheses. Examples of fusing hypotheses about spotting familiar objects are provided in Figure 4.

4. EXPERIMENTAL RESULTS

We test our method on two collections of images: snapshots taken by a SONY AIBO and image sequences captured on the highway. In the first experiment (*AIBO*), we have 12 templates of four objects in three views, and another 12 templates used as backgrounds. Some of the templates are shown in Figure 2. The distance between AIBO and each object is kept fixed during capturing the templates. From these 24 templates we extract $1,540$ interest points, and hence a training set with $N = 1,540$ feature vectors and $C = 5$ different labels. In the second experiment (*Highway*) we have 53 templates of vehicles of various sizes. We categorize the vehicles into three types according to the typical sizes, see Figure 3 for example. We also collect 65 background templates, and in total we generate $8,293$ feature vectors as training samples from the vehicle and background templates. The proposed framework is used to train an MKLR model for each of the two experiments. We then apply the MKLR models to new images, and the results are shown in Figure 4. Note that the depths listed in the depth maps are estimated as the inverse scale ratio $(\sigma_{i^*}/\bar{\sigma})$ multiplying the constant k_i, which pertains to the distance between the camera and the object when the template is taken. About the running time (with Matlab 7), our framework is very efficient. Training the MKLR model for the *AIBO* experiment takes only 1.4 seconds, and for *Highway* 12.7 seconds. We use cover trees for finding the 20 nearest neighbors. The testing time for estimating the object depths in a single image (of size $\simeq 400 \times 300$) is about 0.12 seconds for the *AIBO* sequence and 0.6 seconds for the *Highway* sequence.

Fig. 2. *AIBO*. The images include four familiar objects (will be referred according to this left-to-right order) to AIBO, and two indoor background patches. All templates of object i are assumed to be taken at a distance k_i away from the camera.

Fig. 3. *Highway*. Three categories of vehicles being the set of familiar objects, as well as some outdoor background patches.

Acknowledgements. This work was supported in part by NSC grants 96-2218-E-007-010, 95-2221-E-001-031-MY3, and 95-2221-E-001-030.

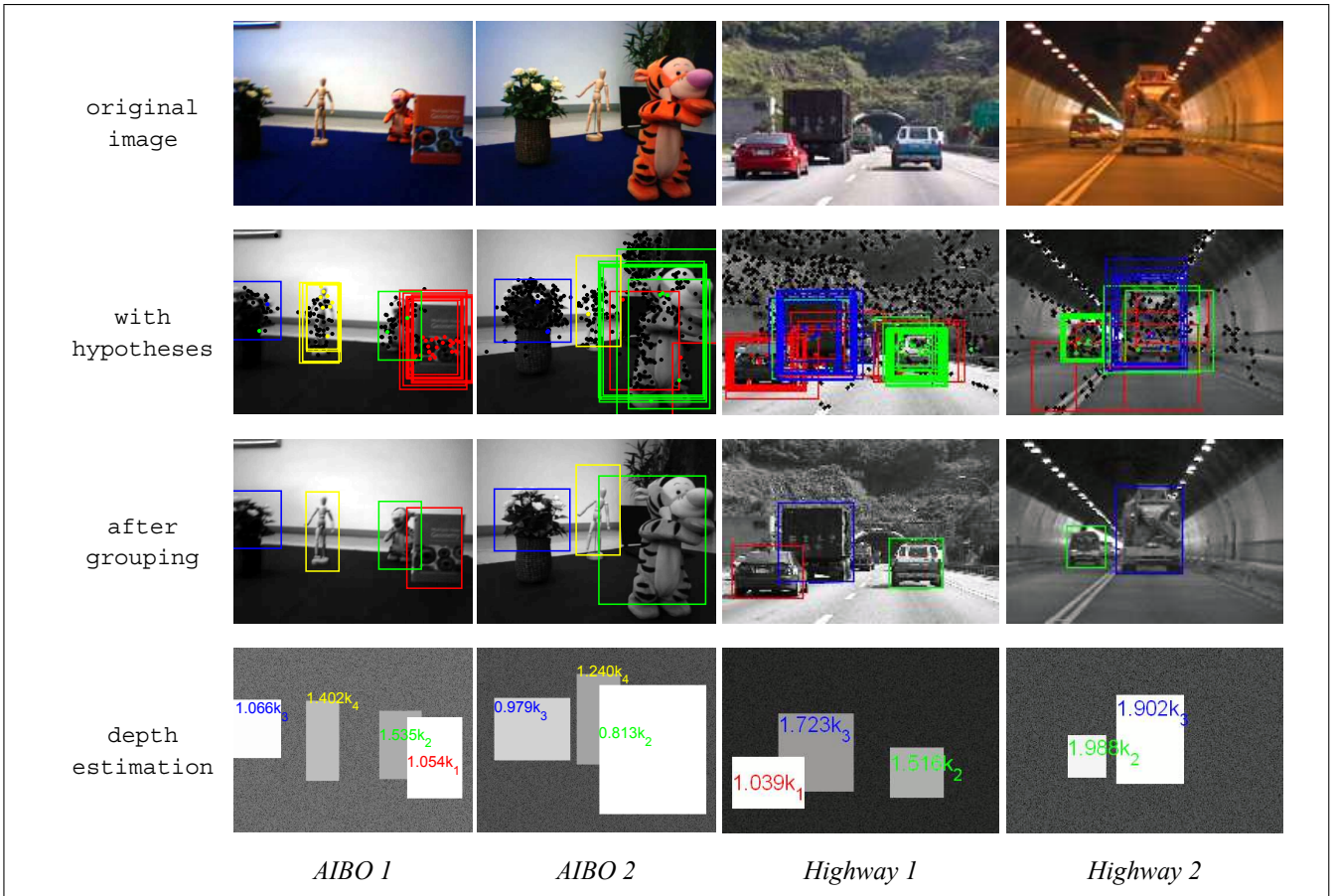

Fig. 4. Each column depicts the results of testing on a single image, including the original image, the one highlighted with hypotheses (dots of various colors are interest points), after hypothesis grouping, and with the estimated depth information. We use bounding boxes of different colors to distinguish different familiar objects. The depth value is displayed as $\beta \times k_i$ where β is the inverse scale ratio discussed in Section 3.1, and k_i is the template distance to the camera as in Figure 2. So, *e.g.*, $0.813k_2$ in *AIBO 2* implies *Tigger* is 0.813 times closer to the camera than the case for its corresponding template.

5. REFERENCES

[1] S. E. Palmer, *Vision Science: Photons to Phenomenology*, MIT Press, 1999.

[2] A. Torralba and A. Oliva, "Depth estimation from image structure," *PAMI*, vol. 24, no. 9, pp. 1226–1238, September 2002.

[3] J. Michels, A. Saxena, and A.Y. Ng, "High-speed obstacle avoidance using monocular vision and reinforcement learning," in *ICML05*, 2005, pp. 593–600.

[4] D. Hoiem, A.A. Efros, and M. Hebert, "Geometric context from a single image," in *ICCV05*, 2005, vol. 1, pp. 654–661.

[5] B. Krishnapuram, L. Carin, M.A.T. Figueiredo, and A.J. Hartemink, "Sparse multinomial logistic regression: Fast algorithms and generalization bounds," *PAMI*, vol. 27, no. 6, pp. 957–968, June 2005.

[6] M. Seeger, "Multiple kernel logistic regression," Tech. Rep., Max Planck Institute for Biological Cybernetics, 2005.

[7] J. Zhu and T. Hastie, "Kernel logistic regression and the import vector machine," in *NIPS01*, 2001, pp. 1081–1088.

[8] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman, "Learning object categories from google's image search," in *ICCV05*, 2005, vol. 2, pp. 1816–1823.

[9] J. Sivic, B.C. Russell, A.A. Efros, A. Zisserman, and W.T. Freeman, "Discovering objects and their localization in images," in *ICCV05*, 2005, vol. 1, pp. 370–377.

[10] G. Wahba, C. Gu, Y. Wang, and R. Chappell, "Soft classification, a.k.a. risk estimation, via penalized log likelihood and smoothing spline analysis of variance," in *Computational Learning Theory and Natural Learning Systems*, 1993, vol. 3, pp. 127–158.

[11] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *ICCV03*, 2003, pp. 1470–1477.

[12] Y. Shen, A. Ng, and M. Seeger, "Fast gaussian process regression using kd-trees," in *NIPS05*, 2005, pp. 1225–1232.

[13] A. Beygelzimer, S. Kakade, and J. Langford, "Cover trees for nearest neighbor," in *ICML06*, 2006, pp. 97–104.