3D AUGMENTED MARKOV RANDOM FIELD FOR OBJECT RECOGNITION

Wei Yu[1], *Ahmed Bilal Ashraf*[1], *Yao-Jen Chang*[2], *Congcong Li*[2], *Tsuhan Chen*[2] *

[1] Carnegie Mellon University, [2] Cornell University

ABSTRACT

In this paper, we propose to use 3D information to augment the Markov random field (MRF) model for object recognition. Conventional MRF for image-based object recognition usually uses appearance and 2D location as features in the model. The problem is solved by finding the globally optimal assignment that minimizes an energy defined in MRF. We estimate rough 3D information from stereo image pairs, and incorporate such information into node and edge potential models in the conventional MRF. Introducing 3D location into the node potential can take advantage of the 3D location distribution statistics of different classes. Considering 3D distance in the edge potential can help distinguish "true" neighbors from "fake" neighbors in 2D. Experiments show improved recognition results by using the proposed technique.

Index Terms— object recognition, Markov random field, 3D, stereo

1. INTRODUCTION

MRF is widely used in many vision applications like segmentation and object recognition. It offers a mathematical framework to model contextual relationship of pieces of local information. A typical application of MRF is to recognize object categories of each pixel(or superpixel) in an image. In training procedure, feature distribution statistics of each object category is collected based on training samples; in testing procedure, assignment of class labels depends on each pixel's similarity to training samples and interaction among neighboring pixels. Optimal class assignment is solved by minimizing a target energy function.

Related work. MRF is used for interactive segmentation to partition an image into two classes: foreground and background [1]. In their papaer, 2D knowledge (texture, edges, object boundary) is explored, but no 3D information. 3D object recognition has also been studied. Brostow et al. show motion and structure can boost recognition performance [2]. They use randomized decision forests classifier, which does not model contextual relationship of pixels. Kushal et al. investigate 3D object recognition [3], focusing on recognizing one 3D object from arbitrary view point and 3D geometry modeling. Our work don't rely on precise 3D geometry reconstruction, and shape variation within one class can be large.

*Thanks to ITRI of Taiwan for funding.

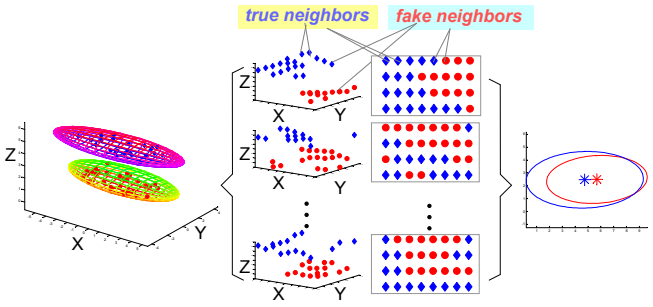


Fig. 1. A synthesized example showing the advantage of using 3D location. First, 2D location distribution of two classes may overlap with each other (2D ellipses), while additional depth information may help distinguish two classes (3D ellipses). Second, neighboring pixels in 2D may be far apart in 3D (so-called "fake" neighbors), which can be distinguished from "true" neighbors in 3D by measuring 3D distance.

Proposed method. The primary contribution of our paper is to incorporate 3D information into the MRF framework for object classification. How 3D can help improving the classification performance in MRF? Fig. 1 uses a simple synthesized example to illustrate the motivation. Assuming we have a lot of images taken at different places at different time in similar environment (e.g. driving environment). We want to classify every pixel in the image to be one of the two classes. Assuming objects from class 1 (●) tend to be closer to camera than objects in class 0 (♦) (e.g. when class 0 is building along the sidewalk, and class 1 is vehicle). If we use a Gaussian model to capture the 2D location distribution of each class, Gaussian distributions of two classes (2D ellipses) may well overlap with each other, making it difficult to distinguish class label from 2D location. However, Gaussian distributions of 3D location of two classes (3D ellipses) are well separated, thus informative of the class label. MRF can easily takes advantage of 3D location distribution by incorporating the distribution statistics into the *node* potential. Another way MRF can utilize 3D information is to estimate real distance of neighboring pixels. If two pixels are neighbors in 2D image, they are more likely to be assigned the same class label in MRF. However,

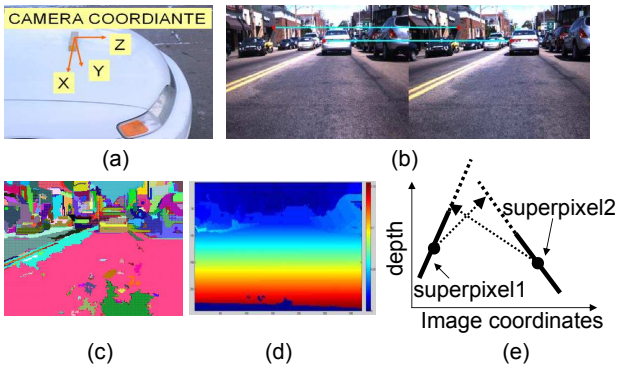


Fig. 2. (a) camera setup and camera coordinates (b) correspondences in rectified stereo image pair (c) segmentation of image into superpixels (d) disparity map (e) computing 3D distance of two superpixels.

neighbors in 2D may come from different classes and be far apart in 3D space. We call those neighbors as "fake" neighbors, compared to "true" neighbors in 3D space, as shown in Fig. 1. By incorporating 3D distance into the *edge* potential, MRF is able to discern "fake" and "true" neighbors.

Synopsis. In Section 2, we explain details of the proposed method, including estimation of 3D from stereo, a brief review of basic MRF and 3D extensions. Section 3 shows qualitative and quantitative results. Section 4 concludes.

2. 3D EXTENSION OF MRF BASED ALGORITHM

2.1. Estimating 3D from Stereo

We use stereo image pairs captured by Point Grey Bumblebee BBX3-13S2C camera to estimate the disparity map. Images are rectified and scaled to size 320×240 before processing. Correspondences lie on the same row of rectified images (Fig. 2(b)). Disparity is the horizontal displacement of correspondence. We use the camera coordinates system, where the origin is the camera center and XY plane is fixed to the image plane as shown in Fig. 2(a).

3D coordinates (X, Y, Z) can be computed from 2D image coordinates (x, y) and disparity d using $[X, Y, Z] = [xB/d, yBf_x/(df_y), Bf_x/d]$. f_x, f_y (focal length) and B (baseline) are parameters specified by the BBX3 camera.

We adopt a similar approach to [4] to design the stereo algorithm. It gives satisfying disparity estimates for the outdoor driving dataset. The right view is divided into segments showing local appearance homogeneity using mean shift segmentation (Fig. 2(c)) [5]. These segments are usually referred to as superpixels. Assuming all pixels in each superpixel come from the same plane Π in 3D ($\Pi_1 X + \Pi_2 Y + \Pi_3 Z = 1$), thus the disparity satisfy $d = ax + by + c$ [4]. The task is to find the best (a, b, c) (termed as warping parameters) for all superpixels. This is done in two steps. First, the best warping parameter for each superpixel is estimated using optical flow algorithm. Second, all found warping parameters are shared among all superpixels, and MRF is applied to find globally optimal solution. Fig. 2(d) shows the final disparity map. Any other stereo algorithm producing good disparity estimate can be used to replace this algorithm.

The 3D plane parameter $\Pi = [\Pi_1, \Pi_2, \Pi_3]$ is given by $\Pi = [a/B, bf_y/(Bf_x), c/(Bf_x)]$. We can compute 3D location of any pixel using its 2D coordinates and 3D plane parameter Π. In section 2.4, we will compute 3D distance of neighboring superpixels. The distance is defined as the average distance of projecting the center of one superpixel to the plane of the other superpixel, as illustrated in Fig. 2(e).

2.2. Baseline: MRF based Algorithm

In the MRF based algorithm, the image is modeled as a Markov random field. Each node x_i represents the class of a pixel (or superpixel), and neighboring pixels (or superpixels) are connected via edges. Assuming there are C object classes, and our goal is to inference the class label $c \in \{1, 2, ..., C\}$ for each pixel. We choose the superpixel representation, because its computational complexity is much less than that of the pixel representation. Each node emits an observation y_i, which is the appearance feature represented by mean Luv color of the superpixel. The model is illustrated in Fig. 3.

The joint probability of superpixel class and appearance over the entire image is given in Eq.(1):

$$\mathcal{P}(\mathcal{X}, \mathcal{Y}) \propto \prod_i \phi_i(x_i) \prod_{i,j \in E} \psi_{ij}(x_i, x_j) \quad (1)$$

$$\phi_i(x_i = c) = \mathcal{P}(y_i | \mathcal{GMM}_{color}(c)) \quad (2)$$

$$\psi(x_i, x_j) = e^{-\beta \mathcal{I}(x_i \neq x_j)} \quad (3)$$

In Eq.(1), ϕ represents the node-potential and ψ represents the edge potential. $\phi_i(x_i)$ captures the correlation between the appearance and class label, indicating the likelihood of x_i coming from class c based on color of x_i and color distribution of class c. Node potentials can be learned from the training data. In Eq.(2), $\mathcal{GMM}_{color}(c)$ is a Gaussian mixture model (GMM) learnt for class c. In Eq.(3), ψ is simply the Potts model, which biases neighboring nodes to have the same class label. $\mathcal{I}(x_i \neq x_j)$ is the identity indication function, which equals 1 when $x_i \neq x_j$ and equals 0 otherwise. β controls the strength of such bias: $\beta = 0$ means no interaction among neighbors and class label is decided by ϕ_i locally; large β puts high penalty when neighbors are assigned different class labels, thus encouraging homogenous label assignments in the image as a whole. Approximate MAP (maximum a posteriori) solution to the MRF can be inferred by Loopy belief propagation (LoopyBP) [6], which maximizes the joint probability $\mathcal{P}(\mathcal{X}, \mathcal{Y})$. The inference engine will output for each node a vector of size $C \times 1$, representing the belief of

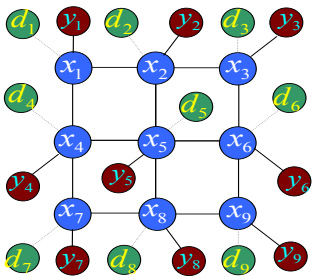


Fig. 3. MRF model: each node x_i represents a superpixel, each edge corresponds to two neighboring superpixels. The node potential $\phi_i(x_i)$ captures the correlation between a superpixel's class label and its appearance. The edge potential $\psi_{ij}(x_i, x_j)$ is simply the Potts model, penalizing different label assignments to neighboring nodes. y_i nodes represent appearance feature (color). d_i nodes represent 3D location, which don't exist in conventional MRF.

this node coming from each class. Optimal class label x_i is simply the class c with the largest belief.

2.3. Incorporating 3D in MRF Node Potential

Now we extend a node to emit two observations, y_i and d_i, representing appearance and 3D location (Fig. 3). This leads to the addition of another node potential, ζ, which captures the correlation between the class label and 3D location of the superpixel center. ζ can be learned in a manner analogous to the one used for learning ϕ in the previous section (Eq.(4)). Here we assume color feature y_i and location feature d_i are independent features given class label x_i. This assumption of independent features is similar to Naive Bayes classifier. It reduces model complexity and makes the model robust in case of limited training samples. Eq.(5) shows corresponding change in the joint probability $\mathcal{P}$ (the change is highlighted in red).

$$\zeta_i(x_i = c) = \mathcal{P}(d_i|\mathcal{GMM}_{3DLoc}(c)) \quad (4)$$

$$\mathcal{P}(\mathcal{X}, \mathcal{Y}, \mathcal{D}) \propto \prod_i \phi_i(x_i) \prod_i \zeta_i(x_i) \prod_{i,j \in E} \psi_{ij}(x_i, x_j) \quad (5)$$

In Eq.(5), ϕ and ζ are required to be learnt only once, and thus can be lumped together into a single node potential, thus allowing to reuse the machinery of the original MRF framework. Adding ζ does not add to complexity of inference, because node potentials are pre-computed once before feeding into the inference engine.

2.4. Incorporating 3D in MRF Edge Potential

Now we modify the edge potential ψ to consider 3D distance of two neighboring nodes. In Potts model, the same penalty β is exerted if two neighboring superpixels in 2D are assigned different class labels. To justify the fact that "fake" neighbors in 2D which are actually far apart in 3D are less likely to come from the same class than "true" neighbors, the edge potential is modified to reflect the 3D distance of neighboring superpixels, as shown in Eq.(6) (the change is highlighted in red).

$$\psi_{ij}(x_i, x_j) = e^{-(\beta \cdot closeness(x_i, x_j)) \cdot \mathcal{I}(x_i \neq x_j)} \quad (6)$$

$$closeness(x_i, x_j) = e^{-Dist(x_i, x_j)} \quad (7)$$

The term $closeness(x_i, x_j)$ computed from Eq.(7) is in between 0 and 1. Effective β (penalty of assigning different class labels to x_i, x_j) in the edge potential ψ_{ij} becomes $\beta \cdot closeness(x_i, x_j)$, putting higher penalty when x_i and x_j are close and lower penalty when they are far apart. $Dist(x_i, x_j)$ in Eq.(7) is the 3D distance of superpixel x_i and x_j, computed as described in section 2.1.

Changing representation of edge potential ψ does not add to the complexity of solving the energy-minimization problem in LoopyBP, because $\psi_{ij}(x_i, x_j)$ of all edges are precomputed before feeding into the inference engine.

3. EXPERIMENTAL RESULTS

Dataset. Our testing dataset includes randomly sampled 50 images from a 60 minutes' stereo video, filmed from a moving car when driving in typical residential and commercial area. The dataset is very challenging, with a lot of occlusions, cluttered background, shadows, illumination change and intra class variation. The training dataset includes 50 images randomly selected from another video captured in a different area. Both training and testing dataset share typical 3D layout of street scenes viewed from a driving vehicle.

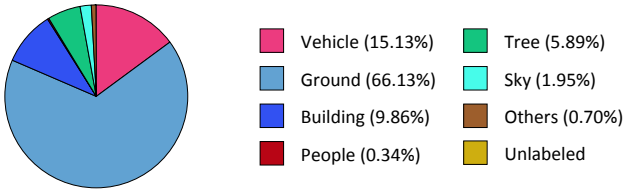
For completeness, we conduct comparison experiments of five models: Nw_EP, Nw2D_EP, Nw3D_EP, Nw_E3D, Nw3D_E3D. The naming rule is that first part N* specifies what features are used in node potential, 'w' means using appearance only, 'w2D' uses both appearance and 2D location, 'w3D' uses both appearance and 3D location. The second part E* specifies what edge potential model is used. 'EP' is the Potts model. 'E3D' uses 3D into the edge potential as in Eq.(6). For whatever feature ('w' or '2D' or '3D'), it's modeled by GMM of 10 centers for each class.

Each image is segmented into superpixels [5], and we manually label each superpixel into one of the seven categories: vehicle, ground, building, people, tree, sky, others. The last category includes all superpixels not belonging one of the first 6 categories such as trash box, poles, traffic lights, etc. A few superpixels are not labeled either because they are too small or the class label is ambiguous (composed of pixels from more than one category). The unlabeled pixels count for a very small portion of all pixels and they are not used in the evaluation. Fig. 4 shows breakdown by category of the proportion of pixels of all images in the testing dataset.

For each of the five models, four β values $\{0.5, 1.0, 1.5, 2.0\}$ are tested. We report per-class accuracy in Fig. 4, which is diagonal of the normalized pixel-wise confusion matrix, i.e. $\frac{\text{\# of pixels correctly labeled in class } c}{\text{\# of pixels with ground truth label } c}$. The last column shows the overall accuracy, i.e. $\frac{\text{\# of correctly labeled pixels}}{\text{total \# of pixels}}$. Samples of qualitative results are shown in Fig. 5. For each model, the reported results use the β that gives the best overall accuracy performance. From the result, we can see that

- Nw3D_EP outperforms Nw2D_EP, showing benefit of using 3D location as a feature in the node potential.
- Nw_E3D outperforms Nw_EP, showing benefit of using 3D distance in the edge potential.
- Nw3D_E3D and Nw3D_EP have comparative performance. The reason could be for this dataset, 3D location is already very informative of class labels.
- Performance improvement from using 3D depends on the quality of 3D estimation. 3D estimation is a hard problem itself, especially for object categories with a small portion of pixels (like people) or little texture (like sky). Ground is probably the easiest category for 3D estimation, and we can see obvious performance boost by using 3D for this category.

Processing time per stereo pair is about 4 minutes (Matlab code) on a desktop. Most of time is spent on estimating 3D from stereo. Real time stereo algorithms such as [7] can be used when processing time is a constraint.



	Vehicle	Ground	Building	People	Tree	Sky	Others	Overall
Nw_EP	0.68	0.77	0.76	0.45	0.88	0.82	0.12	0.76
Nw_E3D	0.72	0.80	0.67	**0.65**	0.89	0.75	0.28	0.78
Nw2D_EP	0.84	0.90	**0.84**	0.28	0.89	0.94	**0.29**	0.88
Nw3D_EP	**0.85**	0.94	**0.84**	0.50	**0.90**	0.94	0.09	**0.90**
Nw3D_E3D	0.81	**0.95**	0.78	0.59	0.85	**0.95**	0.09	**0.90**

Fig. 4. Top: breakdown by category of proportion of pixels in testing dataset. Bottom: per-class accuracy and overall accuracy of five models.

4. DISCUSSION AND CONCLUSION

In this paper, we propose to augment MRF model for object classification by incorporating 3D information into the node and edge potential. 3D location distribution statistics is helpful for applications where depth is informative for different object classes. 3D distance can help discern "fake" neighbors from "true" neighbors in 2D, which is generally applicable to many situations. We believe advances in 3D reconstruction will lead to more active research in using 3D for high level vision tasks.

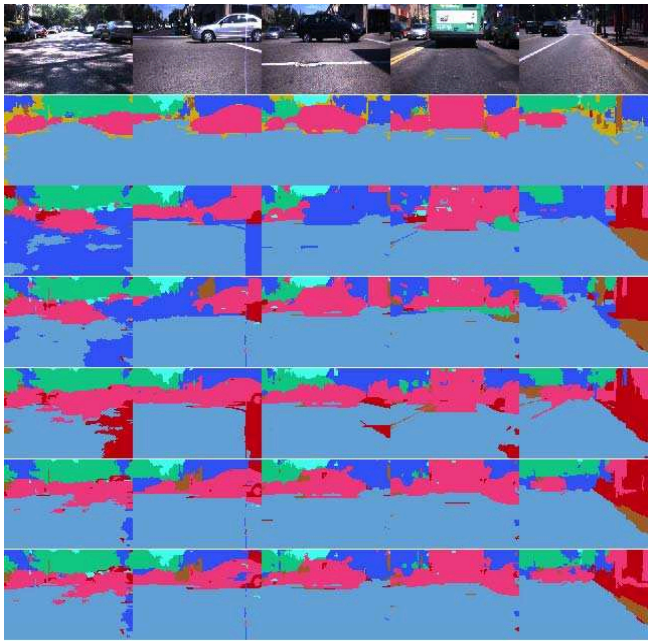


Fig. 5. Qualitative results. From top to bottom: input image, ground truth, Nw_EP, Nw2D_EP, Nw3D_EP, Nw_E3D, Nw3D_E3D. Color of each category is the same as in Fig. 4.

5. REFERENCES

[1] Y Boycov and M Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images," in *ICCV*, 2001.

[2] G. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *ECCV*, 2008.

[3] A. Kushal and J. Ponce, "Modeling 3d objects from stereo views and recognizing them in photographs," in *ECCV*, 2006.

[4] M. Bleyer and M. Gelautz, "Graph-cut-based stereo matching using image segmentation with symmetrical treatment of occlusions," in *Image Communication*, 2007.

[5] D. Comanicu and P. Meer, "Mean shift: A robust approach toward feature space analysis," in *IEEE Trans. PAMI*, 2002.

[6] Talya Meltzer, ," http://www.cs.huji.ac.il/ talyam/inference.html.

[7] W. Yu, T. Chen, and J. C. Hoe, "Real time stereo vision using exponential step cost aggregation on gpu," in *ICIP*, 2009.