Markov Random Field Segmentation For Traffic Sign Detection

Team 10

I. Introduction

Traffic sign detection and recognition have received an increasing interest in the last few years. This is due to the wide range of applications: Highway Maintenance, Sign Inventory, Driver Support Systems, and Intelligent Autonomous Vehicles. Sign detection involves locating road signs within images, and sign recognition involves extracting the sign type. In this paper, we focus on the sign detection task for the development of an intelligent sign inventory for the Georgia Department of Transportation (GDOT). The goal of detection is to extract a region of interest (RoI) for each sign candidate. A RoI is a part, usually a rectangle, of the original image in which there is high probability to find a traffic sign. A 3-step system is often used: segmentation, verification, rectification. Image segmentation plays an important role in detection. As traffic signs have very characteristics features – they are composed of few solid colors and have a regular shape – a segmentation step, usually based on the shape, the color, or both, is performed to detect RoI. During verification, each segmented regions have to fulfill a set of rules. Basically, the size, the shape and the color percentages of each regions are checked. The last step, rectification, extract the sign from the background in order to prepare the sign to be recognized. In this paper, we focus on the segmentation step.

II. Related work

We chose to focus on segmentation methods that are color-based. Color-based segmentation algorithms first classify pixels in different colors corresponding to the usual sign colors (e.g. red for stop signs). The simplest systems use thresholds [4] on one or two color components to classify pixels in one of the sign colors. More elaborated systems use learning methods. For example, in [7] a RBF neural network is learned to differentiate between signs pixel and background pixels. Another possibility is to learn the conditional-class probability density from a training set of pictures. In [6], a Gaussian Mixture Model for each color is separately learned by Expectation-Maximization. In [8], an Artificial Neural Network is used as a regressor to learn the class-conditional density. In [5], they use a Bayesian classifier, based on a color modeling space in which the illumination condition is considered. Then pixels are assigned to the color that has the maximum likelihood. After classification, pixels are aggregated in blobs, by connected component analysis [7, 8] or a regions growing algorithm [6]. Our work on segmentation can be compared to [2], which use Gabor filters and K-means to segment the image.

All these systems have some difficulties with these challenging issues: (1) Change in lighting due to the time of day, the weather, or shadows, (2) Complex backgrounds: road signs can be confused with man-made object patterns, (3) Condition Change: the paint color fades after a long exposure to the sun and rain. Two reasons explain why actual methods fail. First, the spatial locality of pixels is not taken into account during pixel classification but only after (i.e. when pixels are aggregated in blobs). Second, the flat characteristic texture of road signs opposed to the complex background texture is not used. To solve these issues we proposed to use a recent work [1] that, combines the use of color, texture (described by Gabor filters) and spatial relationship between pixels for segmentation, using Markov Random Field (MRF).

III. Implementation

A. *Feature extraction using a gabor filter*

We used a gabor filter to segment color regions of our images. This was done following the implementation in [2]. First, the original image was transformed from RGB to the CIE L* a* b* color space. This creates the layers: lighting, red-green, and blue-yellow. We discard the lighting layer and focus on the distribution of the a* and b* layers.

Using the values from [2], we get two gabor filters that are even and odd symmetric filters. These are the real (eqn (1)) and imaginary (eqn (2)) components, respectively, of the complex gabor filter.

$$G_\theta^e(x,y) = e^{-\frac{x^2+y^2}{18}}\left(\cos(0.2\pi(x\cos\theta + y\sin\theta)\right) \quad (1)$$

$$G_\theta^o(x,y) = e^{-\frac{x^2+y^2}{18}}\left(j\sin(0.2\pi(x\cos\theta + y\sin\theta)\right) \quad (2)$$

At each point, θ is determined by a gradient function to find the dominant direction of the neighborhood surrounding that pixel(eqn 3. This gradient function is comprised of both the vertical (∇_y) and horizontal (∇_x) gradients of the given layers.

$$\theta(x,y) = \frac{\pi}{2} + \frac{1}{2}\arctan(\frac{2g_{xy}}{g_{xx} - g_{yy}}) \tag{3}$$

$$g_{xy} = \sum_{p=x-(k-1)}^{x+(k-1)} \sum_{q=y-(k-1)}^{y+(k-1)} \nabla_x(p,q)\nabla_y(p,q) \tag{4}$$

$$g_{xx} = \sum_{p=x-(k-1)}^{x+(k-1)} \sum_{q=y-(k-1)}^{y+(k-1)} \nabla_x^2(p,q) \tag{5}$$

$$g_{yy} = \sum_{p=x-(k-1)}^{x+(k-1)} \sum_{q=y-(k-1)}^{y+(k-1)} \nabla_y^2(p,q) \tag{6}$$

With the orientation computed and the symmetric gabor filters, we can get the local energy for each point in the image (eqn 7). It is described as the square root of the convolution of a pixel with its corresponding even gabor filter squared, plus the convolution of the odd gabor filter on the pixel squared. This is done for each layer using the same formula. It is from these local energies that the segmentation is performed.

$$LE^{a^*}(x,y) = \sqrt{(a^*(x,y) * *G_\theta^e(x,y))^2 + (a^*(x,y) * *G_\theta^o(x,y))^2} \tag{7}$$

B. Markov Random Fields

Markov Random Field generalizes graphical model (like Bayesian network) to the undirected case. For the segmentation purpose, pixels of an image represent a lattice; each pixel is connected to its direct neighbors. Let $x_{i,j}$ the random hidden variable of pixel (i,j). $x_{i,j}$ can take discrete values corresponding to the class label, $x_{i,j} \in= [1;L]$ where L is the number of classes. In our experiments, the segmentation is performed with five classes ($L = 5$). Let $\vec{y}_{i,j} \in \mathcal{F}$, the observed data, in other words the features associated to the pixel (i,j). We note n the number of features per pixel. Then we assume that observed data follows a Gaussian distribution depending on the class of the hidden variable:

$$p(\vec{y}_{i,j}|x_{i,j} = \lambda) = \frac{1}{\sqrt{(2\pi)^n \left|\Sigma_\lambda\right|}} e^{-\frac{1}{2}(\vec{y}_{i,j} - \vec{\mu}_\lambda)\Sigma_\lambda{}^{-1}(\vec{y}_{i,j} - \vec{\mu}_\lambda)^T} \tag{8}$$

Where $\vec{\mu}_\lambda$ and Σ_λ are the mean and the covariance of the Gaussian distribution associated to the label $x_{i,j} = \lambda$. In MRF segmentation, a Gibbs potential is often assumed for the joint distribution of $x_{i,j}$'s.

$$P(X = x) = \frac{1}{Z}\exp\left(\sum_{i,j}\beta \sum_{(l,m)\in\mathcal{N}_{i,j}} \delta(x_{i,j}, x_{l,m})\right) \tag{9}$$

Where Z is a normalization constant, $\mathcal{N}_{i,j} = \{(i+1,j),(i,j+1),(i-1,j),(i,j-1)\}$ is the set of neighboring indexes of (i,j), β is the smooth parameter and $\delta(.,.)$ equals 1 if the two terms are different, 0 otherwise. The Gibbs potential gives a higher probability for smooth segmentations. Thus MRF segmentation use the spatial correlation between pixels to segment the image. In our experiments, we set $\beta = 4$.

1) Supervised segmentation: In supervised segmentation, we assume that $\vec{\mu}_\lambda$ and Σ_λ are known. Thus, the segmentation is found by maximizing a posteriori probability ($P(x|y)$). Unfortunately, exact methods are intractable. So, we used simulated annealing (SA) with Gibbs sampler to maximized this probability.

2) Unsupervised segmentation: Now, assume that $\vec{\mu}_\lambda$ and Σ_λ are unknown, we want to estimate the Gaussian parameters simultaneously to the segmentation. For this purpose, we used Expectation-Maximization:

1) Start from a random labeling.
2) Estimate Gaussian parameters.
3) Maximize the a posteriori using SA with Gibbs sampler.
4) Return in 2. until the stop criterion is reached.

In step 2., Gaussian parameters are estimated with the following equations:

$$\vec{\mu}_\lambda = \frac{1}{N}\sum_{\substack{i,j \\ x_{i,j}=\lambda}} \vec{y}_{i,j} \tag{10}$$

$$\Sigma_\lambda = \frac{1}{N-1}\sum_{\substack{i,j \\ x_{i,j}=\lambda}} (\vec{y}_{i,j} - \vec{\mu}_\lambda)(\vec{y}_{i,j} - \vec{\mu}_\lambda)^T \tag{11}$$

Where N is the number of pixels. At each step the log likelihood increases. The algorithm stops when the difference in the log likelihood function between two steps is less than $10^{-}7$.

IV. Evaluation

Our initial goal was to perform measurements over approximately 37,000 images by the Louisiana Department of Transportation and Development. However, the CPU time required for the undertaking was too significant. With the addition of the Gabor filter described in [2], a Gabor filter had to be computed for each pixel, in each color channel, of every image. This proved to be lengthy process that limited us to a reduced set of test data and modified evaluation procedures. Additionally, time constraints did not allow us to incorporate our algorithm into an existing traffic sign detection system [8] as we had planned.

Our approach towards evaluating our implementation is to measure the uniformity of our segments on a synthetic image (img IV). Given our knowledge correct boundaries in this image, we can determine the accuracy of the segments produced by our algorithm. Pixels within these boundaries will increment a counter in a corresponding array index. The maximum array element indicates the highest number of uniform pixels that exist within the segment.

In addition to the quantitative approach, we will be visually evaluating the results of image segmentation on real world images taken from [3]. What we are primarily looking for is how well the signs in the images are segmented from their backgrounds. More distinct edges would allow for more accurate image recognition of these signs, as the shape of a sign holds important information as to its meaning.

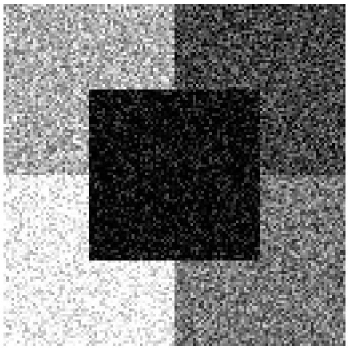

Figure 1. The synthetic image used to test uniformity in the segmented images

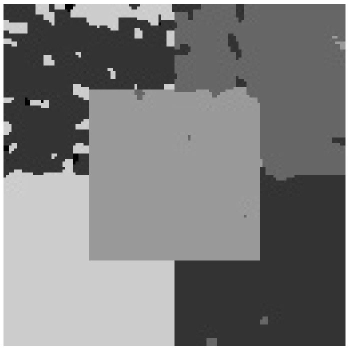

Figure 2. An example result at iteration 8.

Iteration	Average Uniform Pixels
0	2869
1	2867
2	2857
3	2869
4	2867
5	2422
6	1765
7	1901
8	1890
9	1765
10	1745

Table I
Measures how many uniform pixels exist within each of the 5 segments on average.

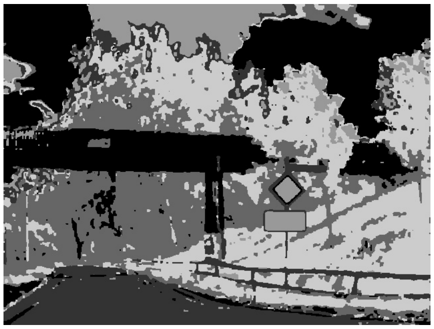

Figure 4. The results of running img IV through our MRF implementation.

V. Discussion

One of the main obstacles we faced while implementing the gabor filter specifically was the CPU time. Because θ dynamically changed with each pixel, the filter had to be computed each time. Moreover the Expectation-Maximization procedure takes a lot of time to converge. With this significant overhead even smaller images took a great deal of time to complete. We tested on images under 400x400 pixels and most of the actual traffic sign images were well over 1000 pixels in width. This is a scale that we could not compute in a timely manner. One thing we would like to do in the future is parallelize this in an effort to process larger images.

On the other hand, MRF allows to take into account the spatial information during the segmentation, producing smoother results and well defined contours which are essential in the sign detection chain. Although unsupervised segmentation was an attractive approach due to its robustness to change in illumination conditions, the high computation time makes this approach unsuitable in a production system. In the future, we would like to design a supervised MRF segmentation algorithm.

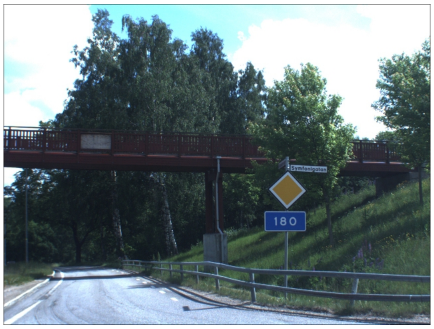

Figure 3. The initial image of the traffic sign to segment.

References

[1] Z. Kato and T.C. Pong. “A Markov random field image segmentation model for color textured images”. In: *Image and Vision Computing* 24.10 (2006), pp. 1103–1114.

[2] J.F. Khan, S.M.A. Bhuiyan, and R.R. Adhami. “Image Segmentation and Shape Analysis for Road-Sign Detection”. In: *Intelligent Transportation Systems, IEEE Transactions on* 12.1 (2011), pp. 83 –96.

[3] Fredrik Larsson, Michael Felsberg, and Per-Erik Forssen. “Correlating Fourier descriptors of local patches for road sign recognition”. In: *IET Computer Vision* 5.4 (2011), pp. 244–254.

[4] S. Maldonado-Bascon and al. “Road-Sign Detection and Recognition Based on Support Vector Machines”. In: *Intelligent Transportation Systems, IEEE Transactions on* 8.2 (2007), pp. 264 –278.

[5] J. Marinas et al. “Detection and tracking of traffic signs using a recursive Bayesian decision framework”. In: *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. 2011, pp. 1942 –1947.

[6] Andrzej Ruta, Yongmin Li, and Xiaohui Liu. “Real-time traffic sign recognition from video by class-specific discriminative features”. In: *Pattern Recognition* 43.1 (2010), pp. 416 –430.

[7] Luo-Wei Tsai et al. “Road sign detection using eigen color”. In: *Proceedings of the 8th Asian conference on Computer vision - Volume Part I*. ACCV’07. Tokyo, Japan: Springer-Verlag, 2007, pp. 169–179.

[8] Yichang (James) Tsai, Pilho Kim, and Zhaohua Wang. “Generalized Traffic Sign Detection Model for Developing a Sign Inventory”. In: *Journal of Computing in Civil Engineering* 23.5 (2009), pp. 266–276.