The effectiveness of eye contour features in gaze estimation

September, 2023

1 Abstract

We proposed a naive gaze estimation method and showed the effectiveness of the eye shape feature in this article. It first extracted feature vectors based on the eye shape segmentation results and the histogram of oriented gradients (HOG). And then a multiple-output Support Vector Regression (SVR) model was trained with the features. The experiment in the part of Multi-view Gaze Dataset (CVPR' 14) showed that the combination of the 2 types of features outperformed each of them alone. It was indicated that the segmentation result could be used to improve the feature descriptor-based prediction. Therefore, a further exploration of the segmentation could be made (e.g., Deep Learning based segmentation) for the purpose of more accurate gaze estimation.

2 Method

2.1 Feature Extraction

The segmentation result of the eye shape is used as a prediction feature. The method generates a mask (a binary image) of the eye, resizes the mask and flattens it into the feature vector. We use Chan-Vese segmentation [1] for the segmentation task. [1] assumes that the average intensities of the object region and non-object region should be different. So it is based on the level sets that are evolved iteratively to minimize energy, which is defined by:

$$E = \int_{inside(C)} |u_0 - c_1|^2 dxdy + \int_{outside(C)} |u_0 - c_2|^2 dxdy$$

where C is the object contour, u_0 is the image intensity, c_1 and c_2 are the average pixel intensity values inside/outside the contour respectively. More regularizing terms are added, like the length of C and the area inside C, for a better performance:

$$E = \mu length(C) + varea(inside(C)) + \lambda_1 \int_{inside(C)} |u_0 - c_1|^2 dxdy + \lambda_2 \int_{outside(C)} |u_0 - c_2|^2 dxdy$$

where μ, υ, λ_1 and λ_2 are fixed parameters.

Thus, the segmentation is converted to a minimizing problem. A binary function, Heaviside function $H(x) = sgn(x)$, is used to indicate the length and inside area of the object.And the energy could be expressed in a more effective way:

$$E = \mu \int_\Omega |\nabla H(\phi)| dxdy + \upsilon \int_\Omega H(\phi) dxdy + \lambda_1 \int_\Omega |u_0 - c_1|^2 H(\phi) dxdy + \lambda_2 \int_\Omega |u_0 - c_2|^2 (1 - H(\phi)) dxdy$$

where Ω is the domain of the whole image, and ϕ is a function on Ω, $\phi(x, y)$, and the object contour can be the set of points, $(x, y) \in \Omega : \phi(x, y) = 0$.

Therefore, the gradient descent of the energy E should be:

$$\frac{\partial \phi}{\partial t} = \frac{d}{d\phi} H(\phi)(\mu div \frac{\nabla \phi}{|\nabla \phi|} - \upsilon - \lambda_1 (u_0 - c_1)^2 + \lambda_2 (u_0 - c_2)^2)$$

It iterates the contour ϕ from the initialization status to a locally optimal solution, which is the object contour. In practice, a continuous function is used instead of $H(\phi) = sgn(\phi)$, to be differentiable at $\phi(x, y) = 0$.

Besides, the Histogram of Oriented Gradient (HOG) [3] is an object detection feature. It calculates the gradient distribution in the local window and keeps information that is not contained in the eye contour. The HOG descriptor is also flattened into a feature vector and combined with the contour vector.

2.2 Gaze direction prediction

The features are used in SVR [4] model training. We define a multiple output regressor that fits one regressor per dimension of the output. In the gaze estimation tasks, a 3-dimensional vector represents the gaze direction in the real world coordinate. Therefore, the multi-ouput regressor consists of 3 separate sub-regressors.

3 Experiments

This method is implemented on Multi-view Gaze Dataset (CVPR' 14) [5] database. It contains 2 types of samples, including synthesized and real test cases. All cases are 60x36 pixel eye images. The SVR models are cross-subject trained on every 3 pieces of synthesis data and predict the results of the corresponding real cases. Subject 00 Subject 09 are used in the experiment. Chan-Vese algorithm parameters, $\mu = 0.25$, $\lambda_1 = 1.0$, $\lambda_2 = 1.0$, the number of iterations 500 and the step of iterations 0.5 are applied to the code implementation. All the contours are initialized as $radius = 18$ circles at the center of the images. The segmentation results are resized to 1/8 of the original images. HOG orientation, window size and cell(s) per block are fixed to 9, 16 and 1. We use RBF kernel [2] and $\epsilon = 0.1$ in the epsilon-SVR model.

The differences (in degree) between the prediction and ground truth results are shown in Figure 1. 3 kinds of features are extracted for the prediction: eye contour feature only, HOG feature only and the combination. It is suggested that the combination of the 2 features has a lower error (15.244 ± 11.287) than using each of them (contour feature 17.706 ± 13.112, HOG feature 11.283 ± 11.283) separately.

It is worth noting that the contour feature has significantly higher variance than the other 2 feature sets. The error rises sharply when the contour feature is inaccurate. The experiment results in Figure 2 show that Chan-Vese fails

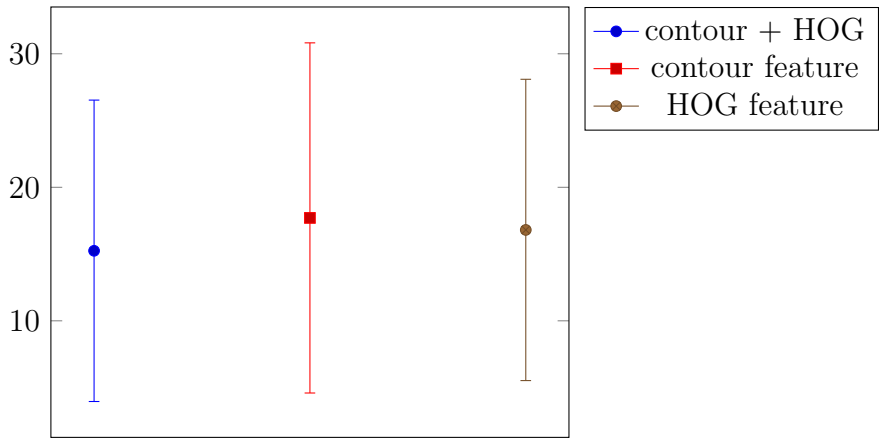

Figure 1: Prediction errors by the feature type.

to segment some cases.

4 Conclusion

A gaze estimation model, based on eye contour and HOG features, has been developed. The result shows that eye contour information improves the prediction effectiveness. The method still has drawbacks, especially in segmentation accuracy. So it is a feasible avenue of research to enhance the segmentation (e.g. studying the Deep Learning based segmentation/prediction model), for better performance.

References

[1] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.

[2] Yin-Wen Chang, Cho-Jui Hsieh, Kai-Wei Chang, Michael Ringgaard, and Chih-Jen Lin. Training and testing low-degree polynomial data mappings via linear svm. *Journal of Machine Learning Research*, 11(48):1471–1490, 2010.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer*

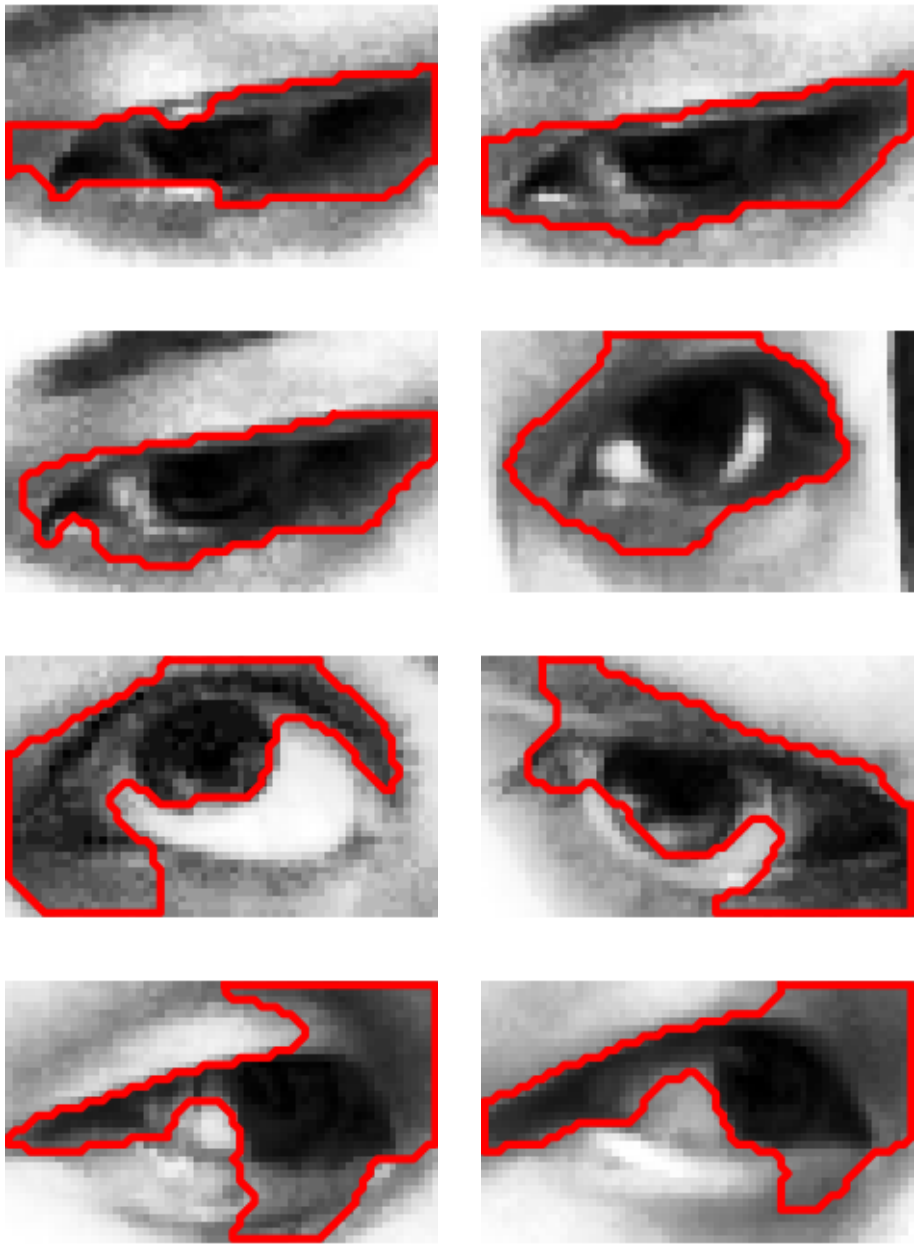

Figure 2: Segmentation results of Chan-Vese algorithm. The accuracy of the results varies in different images. The first 4 images show some precise results, while the others are failure cases.

Vision and Pattern Recognition (CVPR'05), volume 1, pages 886–893 vol. 1, 2005.

[4] John Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Adv. Large Margin Classif.*, 10, 06 2000.

[5] Yusuke Sugano, Yasuyuki Matsushita, and Yoichi Sato. Learning-by-synthesis for appearance-based 3d gaze estimation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1821–1828, 2014.