修士論文

Appearance-based gaze estimation: real-time implementation with accurate eye alignment (アピアランスベース視線推定：高精度な目領域位置合わせを伴うリアルタイム実装)

東京大学大学院
情報理工研究科
電子情報学専攻

48-136442 余鵬

指導教員　佐藤洋一　教授

平成 27 年 2 月

Abstract

Gaze estimation has been widely applied in human computer interaction(HCI) and academic research of cognition, e.g., gaze-based user interface and reading habit analysis. Our goal is to build a real-time gaze estimation system that is practical to consumer user without many limitations. The hardware is a single web camera and user should be able to move head freely.

[HK12] proposed a gaze system on tablet with accuracy of 4.4° ,however the they only achieved a FPS of 0.7 and head motion is not allowed. System proposed by [WB14] allow head motion and achieved a reasonable error and speed, however the eye tracking failure is still a problem.

We employed appearance-based method because the only necessary equipment is single web camera and it is robust under low resolution. However, the main limitations of appearance-based method are head pose change and eye alignment. In order to compensate the bias of gaze due to eye image distortion under head pose, we employed a compensation-based method [LOSS14]. We proposed a new method to detect eye corner and realized accurate and efficient eye alignment. The proposed eye detection method is verification based. We first calculate the locality sensitive histogram(LSH) as a illumination invariant feature. Based on the LSH feature, eye corner candidates are determined by combining the variance project function(VPF) feature and corner feature. We then verified the candidates by considering the consistency of test image and training sample. Besides we implemented some other key techniques in gaze estimation system, including image rectification, eye blink detection, Gaussian Process Regression optimization.

We evaluated accuracy and computation cost of our system and eye alignment method. Our eye alignment method achieved better error rate compared with other eye alignment methods. Our system achieved a speed of 16 fps on a consumer laptop. We achieved a gaze error of 10° under 20° head motion.

Acknowledgements

I could never have completed this work without the support and assistance of many people. First and foremost, I would like to express deepest gratitude to my advisor, Prof. Yoichi Sato, for his excellent guidance, valuable suggestions, and kind encouragement in academic. With his help, I learned how to define research problems and formalize them; how to design solutions and improve them; and how to write papers and present them. I also would like to thank to my Co-advisor Dr. Lu Feng for his extremely helpful suggestions. When I met up with difficulties, he is always patient to help me understand solve the problem in a heuristic way. Also his insight and passion for the research teaches me how to face and overcome difficulties and setbacks, and helps me grow as a researcher with a positive and optimistic view of life.

I also would like to express grateful thank to all members of Sato Laboratory, especially to Dr. Yusuke Sugano, Dr. Ryo Yonetani for their kind advices, helpful assistance, technical support and countless hours of useful discussion on the direction and many issues regarding to my research and life. I would like to thank the secretaries, Sakie Suzuki, Yoko Imagawa, Chio Usui for their support and kindness.

This thesis would not have been possible without generous financial support from Panasonic Scholarship program. Through its very kind staffs, Panasonic also provides students many chances to know japan culture and

have international communication. These supports are gratefully acknowledged.

I also would like to thank my friends. Without them, my life in Japan will not be so happy and harmonious.

Finally, all this would have not been worthwhile, but for my family. It is impossible to put into words my feelings of loves and gratitude for my parents. It is their understanding, perpetual support, and unconditional love that make me overcome all the difficulties through all the years of my study in Japan.

February 2015

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background of gaze estimation

The movement of eyes plays an important role in expressing human cognition, attention and emotion. Many researches have been conducted in the past 30 years, among them computer vision-based gaze estimation has the most technical potential since users don't need to wear special equipments such as electrodes.

Eye gaze estimation has found numerous applications in multiple fields, such as human computer interfaces, cognitive study1.1, market research and driver training. Among them, recently there are more applications on consumer-grade devices. For example, [KIUK13] developed a read habit analyzing system on ipad2 because portable devices can reveal a lot about our physical and visual context, such as enhancing reading experience and understanding reading behavior. As a result, there is more and more demand for a real-time gaze estimation system to satisfy the requirement of these applications.

The needs of eye gaze estimation system becomes our motivation and our

Figure 1.1: Application of gaze estimation in tablet UI(left) and research of reading habit(right)

goal is to develop a real-time gaze estimation system which is practical and friendly for consumer users. We aim to achieve a frame per second(FPS) to be above 12fps. In term of hardware, we use a single web camera that is available on most consumer devices. We aim to solve the limitation of head motion to make the system more friendly to ordinary users.

Gaze estimation system using camera can be realized by model-based method [SYW97], iris-based methods [WSV03] and appearance-based methods [TKA02]. Model-based methods build a 3D model of eye ball and calculate point of gaze (PoG) by a geometric approach. They use infrared LEDs to create glints on pupil to detect pupil center and corneal center. Model-based methods have advantage on accuracy, however additional hardware equipment is needed in order to detect pupil center , such as LEDs and infrared cameras, shown in Figure 1.2..

Appearance-based methods try to build a relationship between eye image and the coordinate of PoG through a regression or learning process, shown in Figure 1.3. Compared with model-based methods, appearance-based methods only require a single web camera to realize gaze tracking while keeping a reasonable accuracy. Besides, they are robust to low resolution image. [HK12] implemented a tablet system using neutral network for regression,

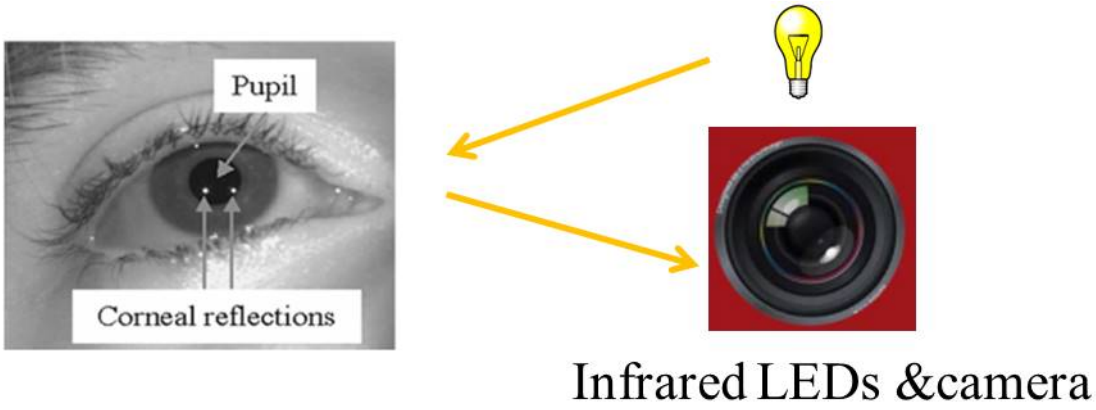

Figure 1.2: Model-based method: infrared LEDs and camera are used to create and detect glints

obtaining an average accuracy of 4.42°. However, free head motion is not allowed because head motion brings high dimensions of freedom and thus eye image varies drastically.

The main idea of iris-based method [WSV03] is to back project the iris circle to 3D space and estimate the gaze direction as a normal vector of the supporting plane, shown in Figure 1.4. Usually a circle edge operator is used to detect the iris contour. The iris contour is a circle in 3D space appears as a circle , while in 2D image plane it appears as an ellipse through a perspective projection, 3D rotation of the circle can be calculated geometrically. In [wood2014], a real-time iris-based gaze system is realized with a single web camera, which can handle head motion and have a reasonable error rate of 7°. The reason of such a high error rate is that low resolution result to eye lid localization failure and less information of iris. Besides, beyond a certain view angle, eyelid may occlude iris a lot ,which causes bad result of iris detection.

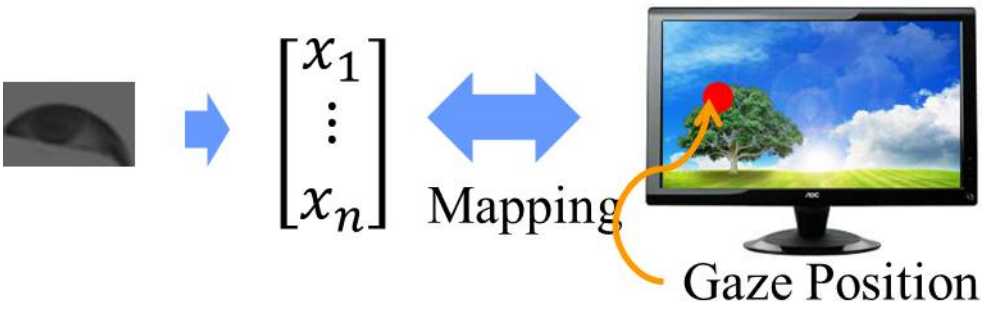

Figure 1.3: Appearance-based gaze estimation:learn gaze point from eye image pixel vector.

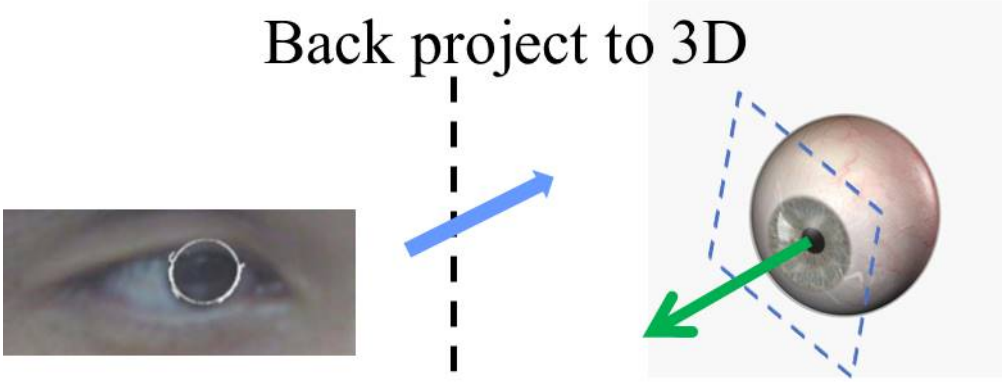

Figure 1.4: Detect iris contour(left) and back project eclipse contour to 3D circle plane(right)

Figure 1.5: Eye image appearance changes under different head pose.

1.2 Overview and Contribution

We employ appearance-based method as the basic methodology in our gaze system. The reason is that compared with model-based method extra hardware is not necessary while it is robust to low image and individual difference is small compared with iris-based method.

However appearance-based method can not handle head pose motion, shown in Figure 1.5 because distorted eye image fall out of training sample space which causes inaccurate regression. Another problem is eye alignment, which directly affects accuracy of gaze estimation. Appearance-based method learned the gaze point by pixel value of input eye image, thus eye alignment is very critical. Head motion increase difficulty of eye localization and alignment because the feature used to align eye change a lot, such as eye contour and appearance.

To solve the problem of eye image distortion, we first employed the method [LOSS14] . Their method can handle head pose variance while requiring much fewer samples than previous methods. In their approach, regression between camera viewing direction and gaze direction bias is build to compensate eye image distortion due to head pose. They claimed that

gaze error within 3° has been achieved.

To solve the problem of eye alignment. we proposed a new eye corner detection method to align eye images that can detect eye corner accurately and fast. Our eye alignment method combines variance project function(VPF) and corner features of eye corner and improve detection accuracy via verifying training-test consistency.

Our contributions are as follows:

- We implemented an appearance-based gaze estimation system in real time, which can give reasonable result under free head motion. We achieved a gaze error within 10° and the system can run under $16fps$.
- We proposed a new eye corner detection method to align eye images accurately and fast. Our eye alignment method detect eye corner by combining VPF and corner feature and improve detection accuracy via verifying training-test consistency.

1.3 Framework of Proposed System

Our system is implemented in a object-oriented way. System flow is shown in 1.6.

From the input image, we obtain the head rotation and head position from a head tracker called FaceAPI. Meanwhile we perform face alignment in order to obtain roughly localize eye. Then we do perspective transform to eye image in order to keep the eye's orientation and size uniform under head motion, which is actually aligning input feature for learning process of gaze [LOSS14]. Precise eye alignment is performed by detecting inner eye corner and cropping a rectangle with fixed size w.r.t eye corner position. For calibration phase, pixel vector of aligned eyes are used as vector for regression

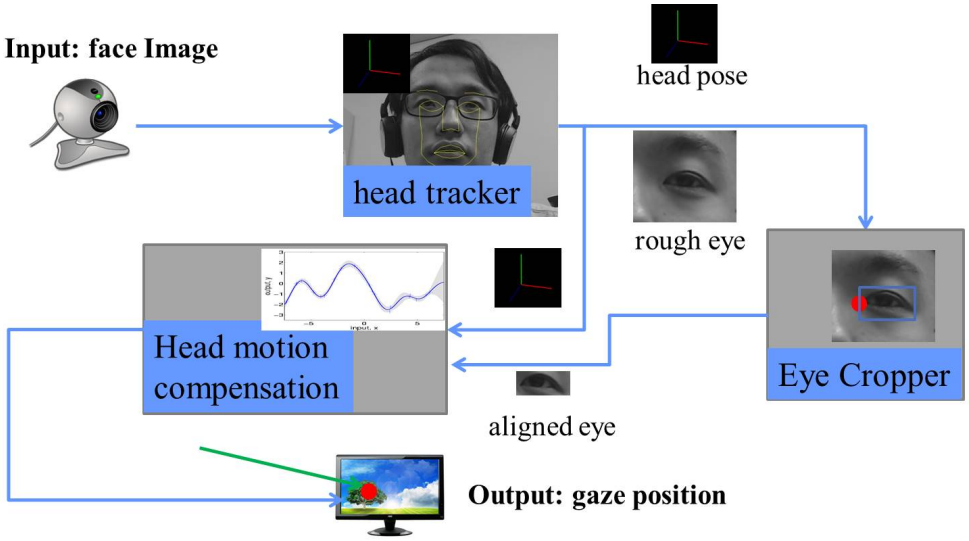

Figure 1.6: Data flow of appearance-based real-time gaze estimation system

between gaze estimation and eye image, while another regression is performed between eye image distortion due to camera taken view and gaze estimation bias. As a result of calibration, two Gaussian Process regression models are built, which are used to estimate gaze estimation under fixed pose and compensate gaze with bias of head motion and eye image distortion. Finally, geometry information is used to calculate the insect point of screen and gaze direction technology list.

Modules surrounded with read rectangle are employing methods in [LOSS14], our academic work mainly focus on eye alignment.

Contents in following chapter are as follows: in Chapter 2 we will introduce related work about eye alignment methods and appearance-based gaze estimation. In Chapter 3, we will introduce our methods to detect eye corner and align eye image and key techniques of gaze estimation. In Chapter 4, experiments on eye alignment and gaze estimation will be introduced considering accuracy and efficiency of our system. We will conclude our work and

discuss about future work in Chapter 5.

Chapter 2

Related works

Eye alignment is very important in appearance-based method because the image pixel vector decides gaze estimation directly. In this chapter, existing mainstream methods of eye alignment will be firstly introduced. Besides, eye alignment becomes difficult especially under free head motion because eye appearance and shape varies drastically, the feature of which is usually used to align eyes.

Popular approaches can be generally divided into face alignment-based methods and eye corner detection-based methods. Eye corner is a very important feature of eyes because its robustness against muscle movements. Thus eye corner has been used a reference point to do eye alignment. Instead of utilizing features of eye, eye alignment can also be done through the process of global face alignment. We will introduce them in the next two sections.

Finally appearance-based gaze estimation methods and system will be introduced.

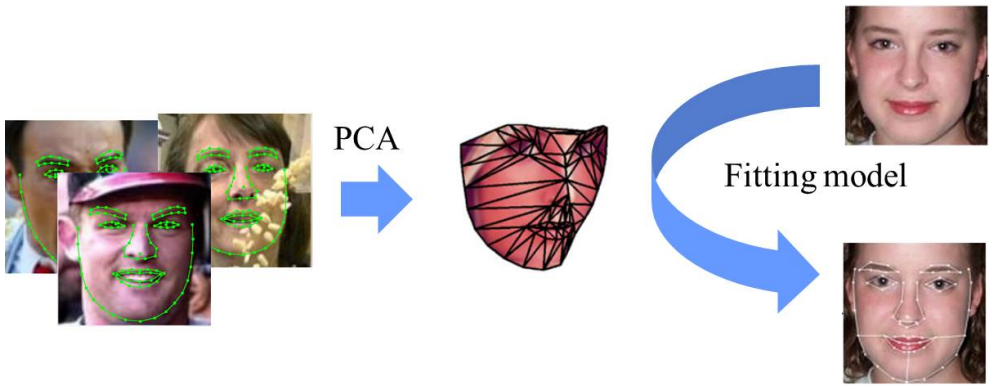

Figure 2.1: Face alignment: build a shape or appearance model from annotated training sample, face is aligned by fitting test image to this model.

2.1 Face landmark alignment methods

Face alignment method is also refered as locating facial landmarks such eyes, nose, mouth and chin. Usually a statistical model of face landmarks is built to fit local facial features, test image is aligned by fitting to the model, shown in Figure2.1 . The main face landmarks methods can be divided into optimization-based approaches and regression-based approaches. Optimization-based approaches include some popular models such as active shape model(ASM) [CTCG95], Deformable Parts Model(DPM) [FMR08]. Regression-based approaches include [XDlT13],[RCWS],[ZLLT14].

Optimization-based approaches build a model shape and appearance of face from training data, and they align test samples by designing an objective function that encodes the alignment error and try to minimum the objective function.

ASM annotated training data to build statistical models of the face's appearance and geometry. To learn a shape model, PCA is computed upon a

set of labeled data of shape or appearance and variance modes are obtained. Then a combination of shape or appearance subspace is done to achieve any shape. Although usually shape is combined by a 2D linear model, 3D shape model [BV99] is proposed to handle continuous view change. Active Appearance Model(AAM) [CET01] share a common representation of facial shape geometry. The difference is that in AAM, besides modeling of shape parameters, a appearance model is also build by performing PCA on texture. Compared with other methods, ASM/AAM show advancements in accuracy, computational complexity and generalization. However, these methods are sensitive to initial pose because some of them use gradient descent based method and may encounter local minimum under complex appearance with illumination and noise [XDlT13]. Besides, extreme view variance of face is unable to be represented by just linear combination or appearance subspace of training samples. The object function of AAM is below and alignment is achieved by finding optimal motion parameter $\boldsymbol{p}$ and appearance coefficients $\boldsymbol{c^a}$.

$$\min_{\mathbf{c^a},\mathbf{p}} ||\mathbf{d}(\mathbf{f}(\mathbf{x},\mathbf{p})) - \mathbf{U^a c^a}|| \tag{2.1}$$

DPM [FMR08] methods fuse geometry configuration into local shape or appearance model.The DPM first divide face landmark sets or appearance into a set of parts with connections between certain parts. Global geometry configuration are then established, which is usually represented with a underlying graph constituted with vertices corresponding to the landmark parts and edges corresponding to connection between different parts . Landmarks are then simultaneously detected by scoring both a local appearance model and deformation cost. Accuracy and efficiency rely heavily on structure of the graph that represents geometry configuration. Face alignment methods

based on DPM typically vary the optimization method of modeling global geometry configuration. [UFH12] learn the parameters of DPM with a structured output SVM. Instead of a binary SVM, they use a loss function to specify the classifier.

CLM [CC08] built the global geometry configuration by assuming all faces lie in a linear subspace constituted by PCA based, which is similar to AAM methods. Unlike AAM, appearance texture sampling is performed for a normalized rectangle patches around each feature, which makes the variation dimension much lower. Another difference between AAM and CLM is search procedure of how to find the fittest parameter to minimize alignment error. While AAM use a regression based method to find relationship between parameter update value and intensity difference, CLM optimized a object function of prior likelihood based on shape model and image response surfaces iteratively. Compared with AAM, CLM has advantage on computational cost and better generalization properties, however it suffers from aperture problem that the local appearance of some facial feature are inherently ambiguous. In [Sar13], a scored output version of SVM is used to score the matching quality between template and image. Unlike traditional CLM methods, a non-parametric representation is built to model the posterior likelihood. For optimization a subspace constrained mean-shifts approach is proposed and show a good balance between computational cost and reducing effects of local minimum. However, head motions are not handled well.

The original regression-based approach [CET98] learn a regression model between alignment error and the image feature, the regression is build through a iterative process. [XDlT13] build a non-parametric shape model in order to generalize better to untrained situations. For optimization, a supervised

descent method is proposed to avoid expensive computational cost of numerical approximations.The main advantage of regression methods is simplicity and efficiency focus on accurate and efficient optimization of object function of a given model through a process of regression. The main drawback is that relationship between feature location and feature value is always nonlinear, requiring high-capacity regression models that are difficult to train and often generalize poorly.

The landmark detector of [UFH12, Sar13, XDlT13] is publicly available. We compared [Sar13, XDlT13]with proposed eye alignment method in the chapter of experiment to show accuracy and efficiency of our method.

2.2 Eye corner detection

Eye alignment can be achieved by detecting eye corner position because the relative position eye corner w.r.t eye image is robust under different gaze and facial expressions. Besides, eye corner has many good features for detection. Pixels around eye corner form a corner, along which vertical and horizontal intensity variance generate as well as good corner feature. Also since eye corner is the intersection of the two eye lid curves and the end points of eyelid curves at the same time, we can use clues such as eyelid edge and eye contour to detect eye corner. Thus eye corner is a robust reference point in eye alignment. Eye corner detection methods can be roughly categorized into filter-based approaches, corner-based approaches and intensity variance-based approaches. Filter based approaches[ZY02] detect eye corners by creating a eye corner filter which is similar to the intensity distribution of eye corner, which is similar to template matching methods. Pixel that has maximum likelihood is considered as eye corner. The filter is applied

to eye image and maximum point regarded as the location of eye corner. [ZYWW05] located eye corners according to a bank of Gabor-based filter, convolved at five different scales and orientations, from which averaged outputs yielded the final detection kernel. Similar to a filter, template matching can also be used to detect eye corners, which calculates the correlation coefficient between template image and eye image. Template matching is not robust under perspective transformation of eye image due to head motion because eye corner appearance varies a lot.

Corner-based methods focus on the corner feature of eye corner. [BCVC13] detect harris corners and use local maximum as candidates of eye corner. They consider the corner nearest to nose is eye corner. According to their experiment, harris corner based method achieved the best performance compared with AAM and canny edge methods. However, their method is based on the assumption that eye area is localized relatively well. Besides, noise between area of nose and eye and low illumination could decrease robustness of this method.

Variance projection function(VPF) [FY98] utilizes the feature of intensity variations near eye corner. VPF calculates the variance of intensity on vertical and horizontal directions. The VPF can be written as:

$$\delta(x) = \sum_{y} [\boldsymbol{I}(x, y) - \boldsymbol{I}_m(x)]^2 \tag{2.2}$$

where $\boldsymbol{I}_m(x)$ is the mean intensity value of xth column. [HG09] improved VPF by adding the Harris's response function as a weight, achieving a robust result for frontal images with no significant lighting variations. They call the method Weight Variance Projection Function(WVPF).

Besides those methods utilizing feature of eye corner itself, eye contour information can also be used to detect eye corner because eye corner is the

intersection of two eye lid curves. [SP11] extract eye contour and fit a eclipse to the contour. Eye corner candidates are selected by Harris-corner method, upon which features related to the contour eclipse are calculated including internal angle, distance to eclipse center, intersection of interpolating polynomials, etc.

2.3 Appearance-based gaze estimation

Appearance-based methods regard the entire eye image as the high dimensional input instead of extracting features like glint or pupil from eye image. Mapping function is obtained by regression or learning and then directly maps the eye image to screen position.Therefore, no infrared camera or geometry calibration is needed. However one problem of appearance-based method is head motion. Head motion will bring in high dimension freedom to regression process, thus eye image distort drastically.For example, not only the eye ball, the eye lip also looks very different. Therefore, a large amount of calibration is required, which will increase the inconvenience for user. In order to allow head motion, Feng [**?**] proposed a method based on synthesis of eye image. They regard head-moving images as fixed head pose captured by multiple cameras. For computational simplicity, cameras are projected onto one camera plane, which is parallel to the image plane. They model the pixel displacement between head-moving eye images as 1D flow, and then produce such flows to synthesis new training image from original training image (four refer-ences) under fixed head pose. However, this method requires tedious computation for training and thus not suitable for real time system. [SMSK08]proposed a pose-based clustering approach that extends an appearance manifold model to handle head pose variance.

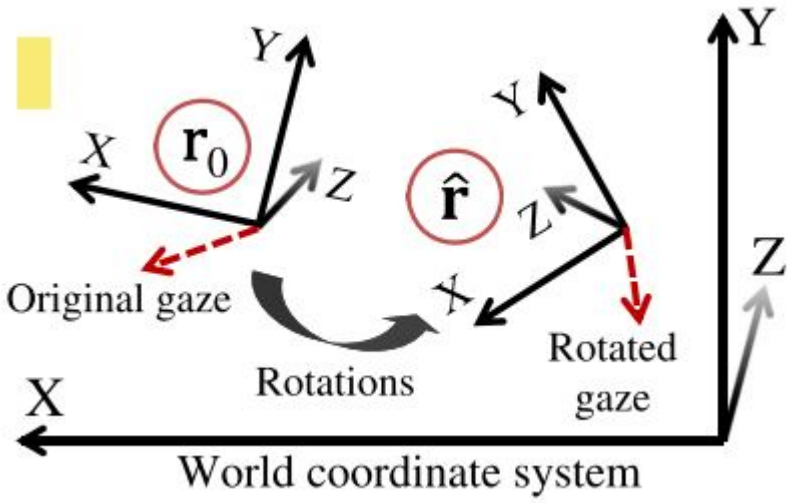

Figure 2.2: Rotate gaze vector from $\boldsymbol{r_0}$ to $\hat{\boldsymbol{r}}$. [LOSS14]

[LOSS14] proposed a learned regression between camera viewing direction and gaze direction biasto compensate eye image distortion due to head pose.They compensate the gaze bias in two steps: learning-based regression and geometry-based calculation. First they build a Gaussian Process Regression(GPR) between gaze bias $\triangle\sigma$ and head pose $\triangle v^c$, which can be written as

$$\triangle\, \sigma_i^x = f_x(\triangle v_i^c) \backsim \mathcal{GP}(0, k_w(\triangle v_i^c, \triangle v_j^c)) \tag{2.3}$$

Geometry calculation is also necessary because gaze rotates along with head rotation, shown in Figure 2.2. Assuming gaze is fixed relative to head, we rotate head from $\boldsymbol{r_0}$ to $\hat{\boldsymbol{r}}$, then gaze vector will undergo the same rotation.Their method can handle head pose variance while requiring much fewer samples than previous methods. Thus we employ the method of [LOSS14] to handle head motion problem in our system.

Chapter 3

eye corner detection-based eye alignment and system implementation

In this chapter, we first introduce how to eliminate illumination change in section3.1. Then we introduce how we detect eye corner in section 3.2. In section 3.3, we introduce how to verify detected eye corner candidates. Finally, we introduce other key component in our gaze system.

Considering eye alignment result decides the input value for the prediction, wrong eye alignment will directly decrease gaze estimation accuracy. In order to realize an efficient eye alignment method, we proposed an eye corner detection-based method. Even though face alignment corner detection method has advantage on overall accuracy. Eye corner-based method is more accurate with eye localization done because eye corner is robust for gaze and has good feature for detection. The largest limitation of eye corner detection-based methods is that rough eye localization is necessary. In our system, we use a commercial head tracker that can provide the relatively

refined eye localization. Thus we choose to use eye detection-based method.

We also implemented other key techniques in gaze estimation system including compensation method of gaze error[LOSS14], rectification of image[LOSS14].

3.1 Eliminate illumination change

It is important to eliminate the inference of light in appearance-based estimation especially under free head motion. Especially under free head motion, even if the lighting condition doesn't change, head motion will change the lights the region of eye received, which can't be ignored according to our observation. n order to eliminate illumination change, we utilized the work of [HYL+13]. We first will give a explanation why LSH is an illumination invariant feature, and then we explain more metrics that locality sensitive histogram brings in eye corner detection.

Histogram of a image is a 1D array, of which value is usually an integer indicating the frequency of occurrence of a particular intensity value. On the other hand, local histogram record statistics within a region of a pixel in an image. They are computed at each pixel location, and have been proved to be very useful for tasks like tracking [Por05]. Local sensitive histogram extended local histogram by considering weight of each pixel. In LSH, pixel far away from the target center is weighted less as they more likely contain background information or occluded objects. LSH can be written as:

$$\boldsymbol{H}_p^E(b) = \sum_{q=1}^{W} \alpha^{|p-q|} \cdot Q(\boldsymbol{I}_q, b), b = 1, \ldots, B, \tag{3.1}$$

where α is a parameter controlling the decreasing weight as a pixel moved away from the target center, W is the number of pixels and B is the total number of bins. and $Q((I)_q, b)$ is one when intensity value $(I)_q$ is b, otherwise

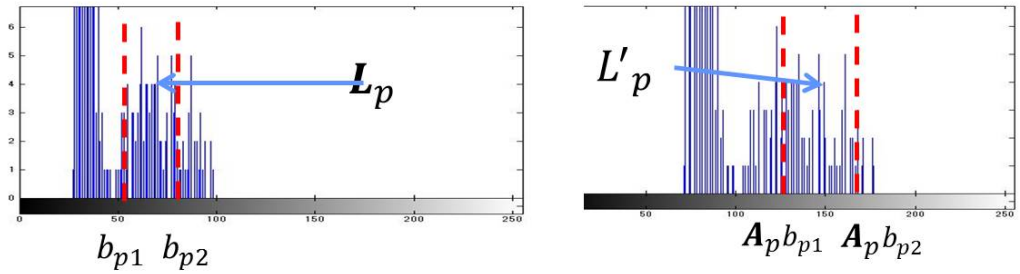

Figure 3.1: histogram interval has same affine transform as image, thus pixel number is invariant.

zero.

LSH is an illumination invariant feature, which means illumination will change not value of LSH. Assume we have a illumination change as written in:

$$\boldsymbol{I}'_p = A_p(\boldsymbol{I}_p) = a_{1,p}\boldsymbol{I}_p + a_{2,p}, \tag{3.2}$$

where $a_{1,p}$ and $a_{2,p}$ are parameters of the affine transform at pixel p. Considering a window S_p centered at pxiel p. The number of pixels in S_p residing in $[b_p - r_p, b_p + r_p]$ is

$$L_p = \sum_{b=b_p-r_p}^{b_p+r_p} \boldsymbol{H}_p^S(b), \tag{3.3}$$

where r_p control the integration interval. Under illumination transform A_p, [HYL+13] proved that the integration interval scales with the illumination affine transform.

$$r'_p = a_{1,p}r_p, \tag{3.4}$$

L'_p will corresponds to the number of pixel with intensity value that resides in $[A_p(b_p - r_p), A_p(b_p + r_p)]$., thus L'_p is equal to L_p if ignore the quantization error. Thus LSH value is invariant under illumination. The process is shown in Figure 3.1

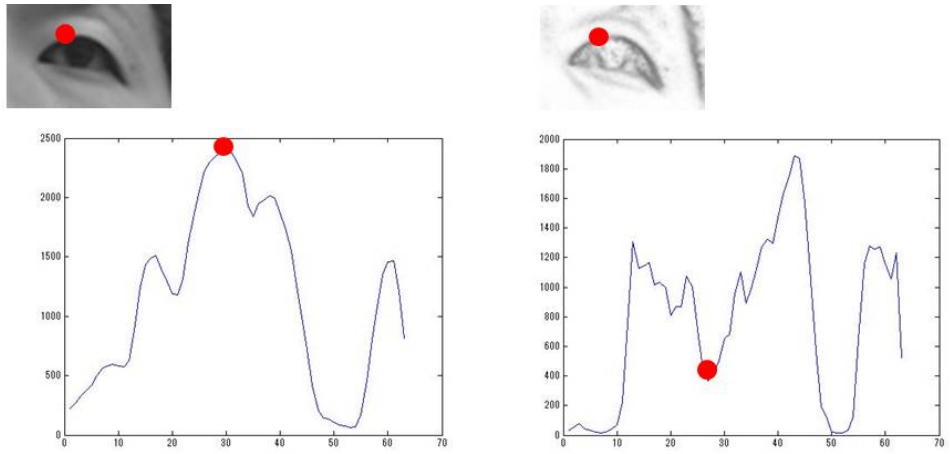

Figure 3.2: VPF value of iris region decreased drastically on LSH image.

LSH not only provide a illumination variant pixel feature, but also strengthen the VPF feature of eye corner mentioned in Chapter 2. As shown in Figure 3.2, VPF of iris region is the global maximum. While the LSH image, VPF of iris region became local minimum. The reason is that LSH computes the pixel number within a certain window that has similar intensity value with the window center. Thus if pixel values are similar in a region, all the LSH values of this region will be large. Thus the intensity variance will be small.

3.2 Detect eye corner with corner and VPF feature

We introduced several eye corner detection methods in section 2.2. Among them, we know that corner feature and VPF feature is robust even under extreme variance. However according to our experiment horizontal VPF is not applicable because head motion disordered the pattern of variance projection feature. To explain this more, in normal case VPF feature of

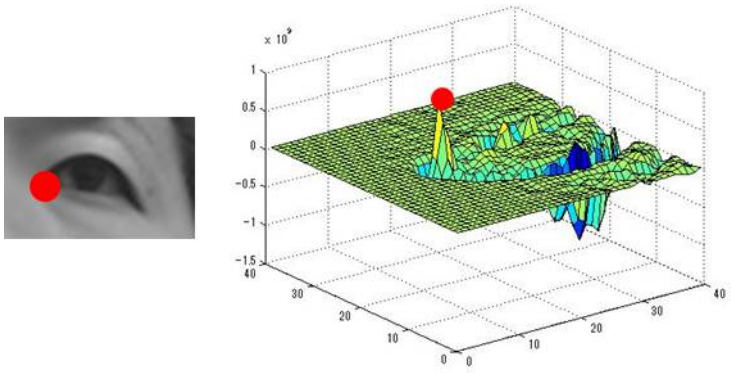

Figure 3.3: eye corner has global maximum likelihood using our feature.

eye corner has a unique pattern, by using which region of interest(ROI) is relatively less difficult to determine. However in a free head motion case, symmetry of VPF feature will be disordered.

Thus we employed these two features. Besides, in order to reduce the inference of iris area, we considered the distance to nose since inner eye corner is closed than iris to nose. The method can be formulated as below:

$$\boldsymbol{W}(x,y) = R(x,y) \cdot [\boldsymbol{I}(x,y) - \boldsymbol{I}_m(x)]^2 \cdot (w-d) \tag{3.5}$$

where $R(x,y)$ is the harris corner response function, $[\boldsymbol{I}(x,y) - \boldsymbol{I}_m(x)]^2$ is VPF feature, d is the distance to nose. As shown in Figure3.3, eye corner has global maximum likelihood using our feature which is obvious higher than other region of eye.

Finally, we can get candidates of eye corner by choosing the points with large likelihood $\boldsymbol{W(i,j)}$. In order to ensure that candidates include the correct eye corner, we split the region of interest evenly on vertical and horizontal orientation, an then certain number of points from each subregion are picked up. By smoothing the choices of candidates spatially, we can avoid

the candidates concentrate on the same region, thus ground truth point of eye corner is more likely to be included in candidates.

3.3 Eye corner candidate verification via test-training consistency

By calculating WVPF feature we described in last section, we can get the right eye corner position very possibly. However, because the ROI is difficult to determine, we have to expand the ROI to ensure eye corner is included. Thus region of iris or eye brown will also be included in some cases. Because these regions also have obvious harris corner feature and VPF feature, they may be mistook as eye corner. Besides, eye corner feature may be less obvious under dark illumination.

Because of these facts, we consider that a verification phase is necessary to ensure that we detect the correct eye corner. The idea is simple that we compare the aligned eye images with eye corner candidates with training sample images that we assume are correctly aligned. Because eye alignment is relatively robust under fixed head pose, we use the training samples as comparison targets. Comparison can be done upon two features: pixels difference and max contour size. Pixel difference comparison can be written as:

$$L = \min ||\boldsymbol{I}_{c,i} - \boldsymbol{I}_{t,j}|| \tag{3.6}$$

where $\boldsymbol{I}_{c,i}$ is the aligned image using eye corner candidate,$\boldsymbol{I}_{t,j}$is the training sample under fixed head pose. We calculate the minimum distance between a certain candidate image and training samples, and use the distance as one feature of eye corner likelihood.

We also employ max contour size of edge to calculate eye corner likelihood.

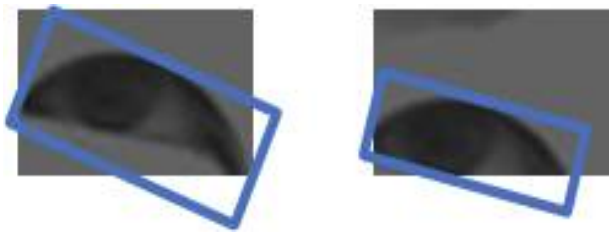

Figure 3.4: correctly aligned image has larger contour than incorrect one.

As we can see in Figure 3.4, correctly aligned eye image has larger contour than incorrect ones. We find this is a robust feature because most of the incorrect alignment is due to iris region, which align eye image incompletely. We employ algorithm in [S+85] to analyze the connected component contour. We first extract the candy edge feature and build the topological structure of the border in image. Then we can find the maximum connected component contour as the root of the topological structure.

The final likelihood function can be written as below:

$$\boldsymbol{W}(x,y) = R(x,y) \cdot [\boldsymbol{I}(x,y) - \boldsymbol{I}_m(x)]^2 \cdot (w-d) \cdot S_{max}/L \qquad (3.7)$$

where $S_max$is the max contour size and L is the min distance between candidate image and training samples.

3.4 Implementation of other key techniques

Besides eye alignment, there are several other key techniques in our gaze estimation system, including gaze bias compensation, image rectification, Gaussian Process Regression(GPR) optimization and blink detection.

Image rectification is to perform a perspective transform to image so that eye image show in similar poses. It is necessary under head motion because

eye image show in arbitrary poses Figure3.5 (left), which increase difficulty of eye alignment. To perform image rectification, we employ method in [LOSS14]. They first defined camera coordinate system and head coordinate system. The idea is to move camera hypothetically so that camera coordinate system become parallel to head coordinate system. With rectification, eye image pose look consistence(right). We implemented this technique by narrowing rectification target to roughly cropped eye image. This improved system efficiency because eye image region is less than 1/10 of the whole face image. Image rectification can be written as:

$$\boldsymbol{p'} = z\boldsymbol{K}\boldsymbol{\Omega_r}\boldsymbol{K}^{-1}\boldsymbol{p} \tag{3.8}$$

where $\boldsymbol{p}$ is the pixel coordinate in eye image $\boldsymbol{I}$ and $\boldsymbol{p'}$ is the corresponding pixel coordinate in rectified eye image. z is a distance scalar, $\boldsymbol{K}$ is intrinsic matrix of camera and $\boldsymbol{\Omega_r}$ is the rotation matrix which is the same as in [LOSS14].

Gaussian Process Regression optimization We employed Gaussian Process Regression(GPR) to learn gaze direction from input image. The predictions of a Gaussian process model will depend on the choice of covariance function. In order to achieve better accuracy and generalization, we decide to infer the parameter values from the data. We calculated hyperparameters θ by maximizing the likelihood function $p(\boldsymbol{t}|\theta)$ which is written as below:

$$lnp(\boldsymbol{t}|\theta) = \frac{-1}{2}ln|\boldsymbol{C}_N| - \frac{-1}{2}\boldsymbol{t}^T\boldsymbol{C}_N^{-1}\boldsymbol{t} - \frac{N}{2}ln(2\pi), \tag{3.9}$$

where $\boldsymbol{C}_N$ is the covariance matrix. The maximum is performed by calculating gradient of function.

Eye blink detection is performed by building a classifier based-on super vector machine(SVM). We used the RPI ISL Eye Database [WJ05], which

contained 2070 closed eye images and 1773 open eyes with different sizes and orientations.

Gaze bias compensation we collect training samples under different head poses and calculated gaze bias using the regression learned under fixed head pose. Then we build a regression between head pose and corresponding gaze bias using Gaussian Process Regression(GPR).

Figure 3.5: eye images show different pose under head motion(left). Rectified eye images has similar pose.

Chapter 4

Experiment

In this chapter, we would evaluate our gaze estimation system. The evaluation is performed from three aspects: eye alignment, gaze estimation and efficiency.

4.1 Experiment environment and setting up

Our system is measured on a laptop with 8GB memory and 1.8GHz CPU. We used a logitech920 web camera with 960*720 resolution image. The screen size is 37.5cm(height)*30cm(width) with a resolution of 1280*1024. User is sitting around 30cm from the screen. We have a two phases of training. In the first phase, they are asked to stare at 6*6 reference points that distributed evenly on screen. At that time they keep head pose fixed. The second training phase is to learn regression between gaze bias and head motion by collecting 200 sample images. We ask users to stare at one reference point in the middle of screen while rotate and move head freely. In test phase, we ask user to stare at 4*4 reference points distributed evenly on screen. Each reference point display for 4 seconds and we collect 30 test images for each

point. Gaze error is calculated by comparing ground truth and estimation result. We begin training and test phase by asking user to interact with a console user interface.

4.2 Eye alignment

We tested our data on 3200 images collected from 8 subjects with gaze and head pose on arbitrary directions. We evaluate the eye alignment by measuring inner eye corner detection error between manually labeled ground truth and detection results. We compared our method with constrained local model-based(CLM) face alignment method [Sar13] which proposed a optimization-based fitting method, a regression-based face alignment method called Intraface [XDlT13], and two feature-based method which is harris-corner [BCVC13] and gabor filter-based template matching [ZYWW05]. Figure 4.2 showed the accumulation error rate, We can know that our method achieved the best performance compared with other four methods. Intraface and harris-corner methods also have very small error rate. CLM and template-matching have relatively large error. We didn't record result of weight VPF[HG09] because the error pixel is beyond 20. However, the accumulation error rate of intraface converged fastest, which means the standard deviation is small.

To see the result in another perspective, we calculated mean value and standard validation of error, shown in Figure.4.2. We can know that appearance-based methods such as CLM and intraface have smaller standard deviation, which means their result is more stable. However our method has the least mean error.We can't definitely say that our method has the greatest performance of eye alignment because eye alignment is ambiguous and eye corner is

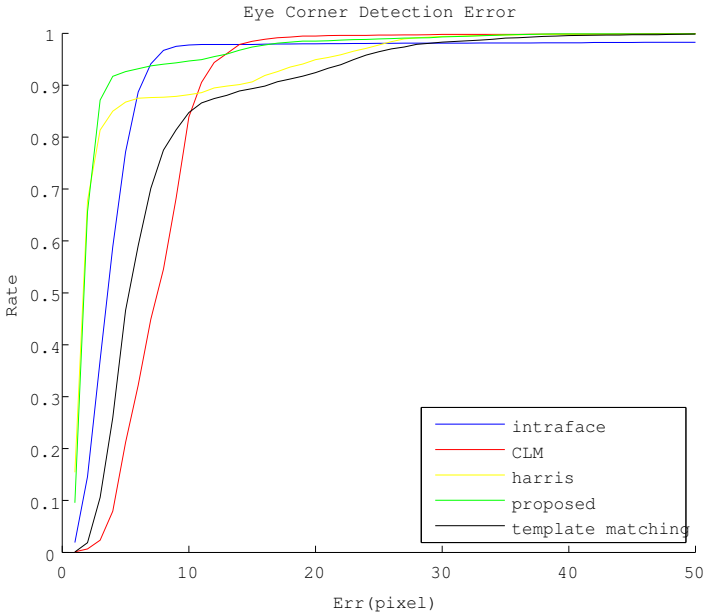

Figure 4.1: Horizontal axis showed the normalized error of inner eye corner estimation for each methods. The performance is measured as percentage of images below certain error value.

not the only criteria. Besides, the definition of eye corner itself is ambiguous. We can infer from the gaze result in Section 4.3 that eye alignment stability is also significant in appearance-based gaze estimation.

Detection result can be seen in Figure4.3. We can see that all methods achieve good result under fixed eye image except harris-corner method may be disturbed by high light area near nose. However under head motion, accuracy of template matching and CLM method both decreased due to the variance of eye image appearance. Compared with that, our method is relatively robust even under head motion.

4.3 Gaze estimation

We test our gaze estimation on the same 3200 sample images in Section 4.2. We compensated gaze error due to head motion employing the method in [LOSS14], which we believe can handle the eye distortion problem due to head motion. As we figure out that eye alignment is important in gaze esti-

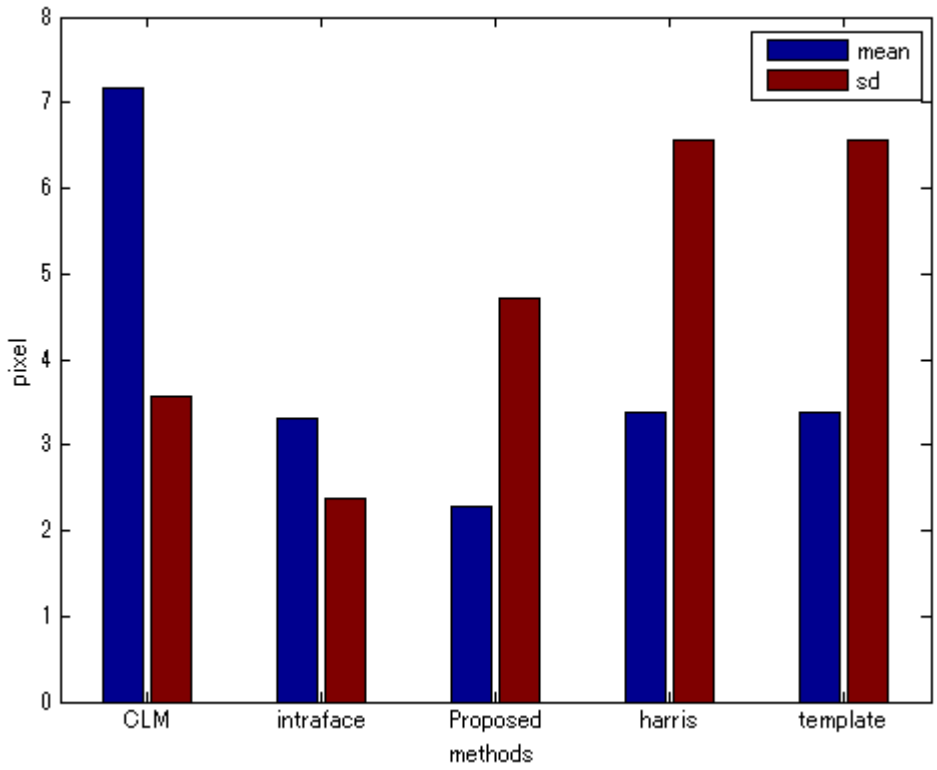

Figure 4.2: Mean and standard deviation of eye corner detection error. Our methods has the least mean error and reasonable standard deviation.

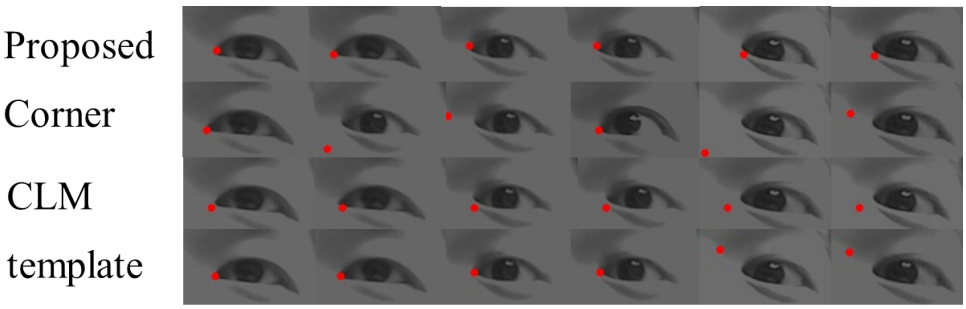

Figure 4.3: Eye corner detection result of Harris corner, template matching , proposed method and CLM.

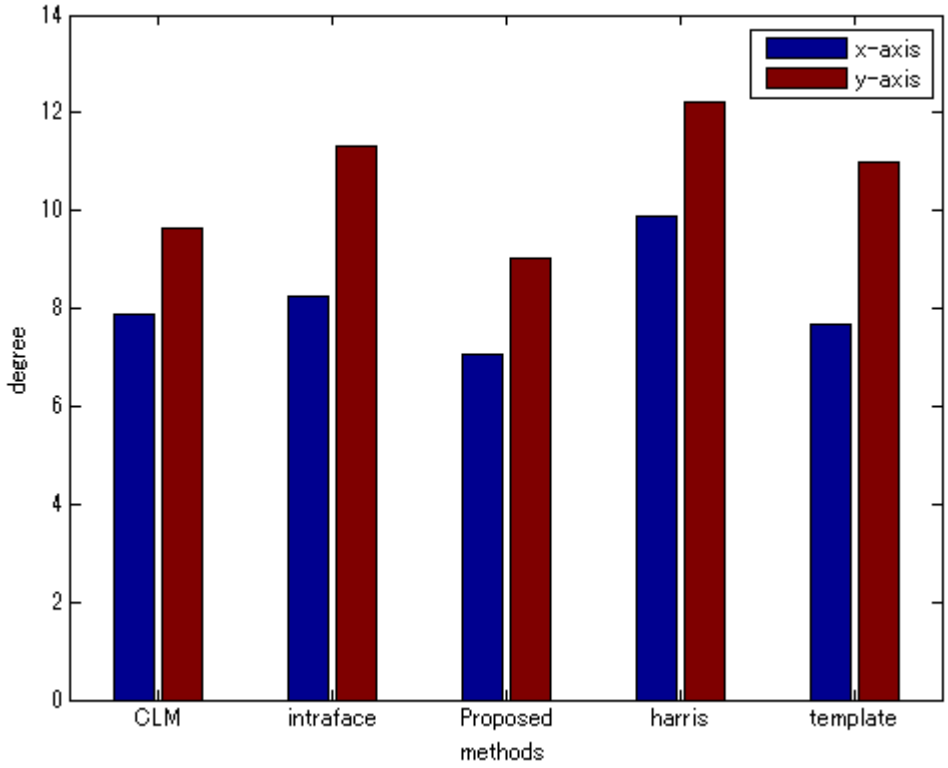

Figure 4.4: x and y axis gaze direction error. Our methods has the least mean error on both direction.

mation, we use different eye alignment methods which we already compared in previous section. We measure the gaze direction error on x axis and y axis for different alignment methods respectively.The result is shown in Figure 4.4. We can know that our method also achieved the least error on both x axis and y axis of about 11°. CLM achieved the second highest accuracy because the eye alignment of CLM is very stable. One interesting thing is that the result of template matching is good even if the eye alignment result is bad.

Gaze result on different subjects in shown in Table 4.4. We can know that

the difference between subjects is relatively small. This is the advantage of appearance-based gaze estimation compared with iris-based methods.

Table 4.1: gaze error of different subjects

subject	x axis	y axis
1	9.02°	8.92°
2	7.13°	8.68°
3	7.45°	10.35°
4	6.45°	10.34°
5	6.03°	11.89°
6	8.93°	13.29°
7	3.84°	7.10°
8	7.61°	8.23°

In order to show the affect of head pose, we measured gaze error on different head pose and compared the gaze errors before and after compensation. The result is shown in Table 4.2, we can tell that gaze error increased drastically without gaze error compensation, and the error get larger along with head rotation. Another observation is that even we performed compensation, the error is still large under head motion. The reason is that some test images are under head motion that is larger than training image, thus regression between head pose and gaze bias cannot be estimation accurately.

In order to prove that gaze result can be improved by covering large head motion cases, we exclude extreme cases from test images. First we define extreme cases, which is shown in Figure 4.5. As head pose training space is relative to head pose, we perform a perspective transform to training space $\boldsymbol{T}$ along with head motion to $\boldsymbol{T'}$. If ground truth of gaze direction fall out of

Table 4.2: gaze error under different head pose before and after gaze error compensation

head rotation	x axis	y axis	x no comp	y no comp
0°-12°	6.60°	8.64°	7.23°	9.12°
12°-24°	6.99°	9.90°	12.55°	13.67°
24°-	7.46°	11.08°	18.36°	22.13°

$\boldsymbol{T'}$, we call it an extreme case. Among 3200 images, around 1700 images are belonging to the extreme cases. We exclude the 1700 images and recalculated gaze error, the gaze error improved from 11.5° to 10°,shown in Table 4.3.

Table 4.3: gaze error of different subjects after excluding extreme cases

subject	x axis	y axis
1	7.11°	8.67°
2	8.04°	7.94°
3	7.75°	8.35°
4	5.25°	7.79°
5	6.45°	11.02°
6	6.15°	11.57°
7	4.14°	6.31°
8	4.82°	5.07°

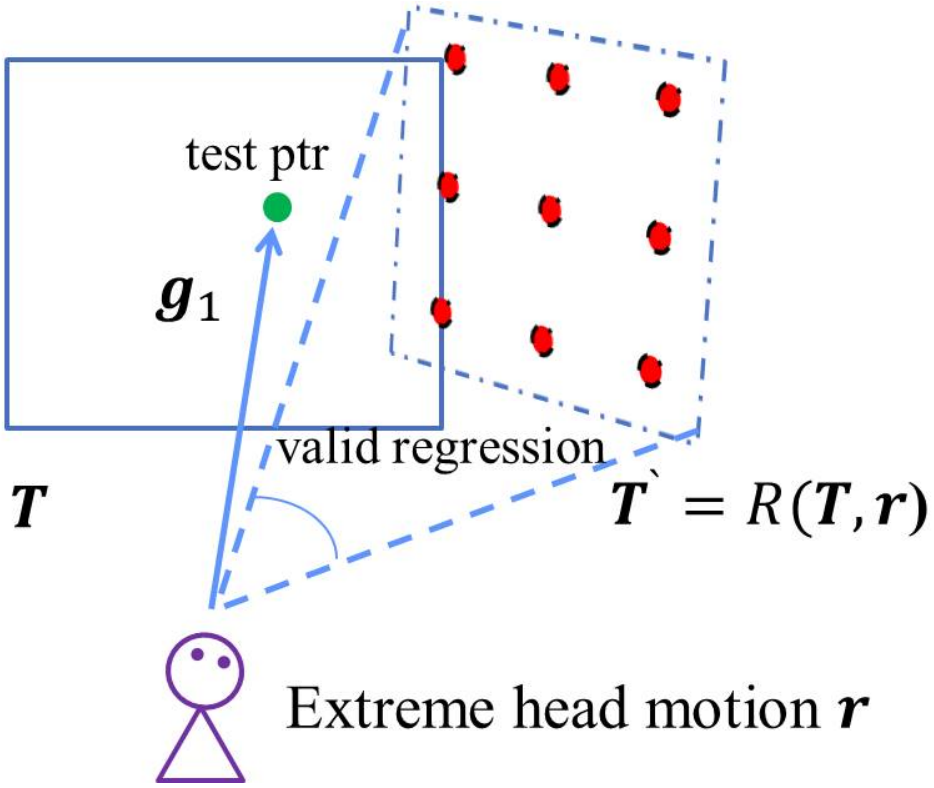

Figure 4.5: Extreme case: gaze fall out of T', thus regression is inaccurate.

4.4 Evaluation of computational cost

We performed the comparison by employing different eye alignment method, including optimization-based face alignment method CLM, regression-based face alignment method intraface,proposed method, harris corner method and optimization-based method in [LOSS14]. The result showed that our system can be run under 16 fps, faster than CLM which optimized the whole face. The fastest method is intraface. Intraface is a independent system, so we only calculated run time of eye alignment for intraface. Since eye alignment is the most time-consuming part in gaze estimation, we can still consider intraface is faster than our system. However our method is more accurate than intraface Harris corner also showed its efficiency. Optimization-based method [LOSS14] only achieved a 5 fps, the reason is that the iterative-based searching and reconstruction process is time-consuming.

Table 4.4: efficiency comparison of eye alignment method

Methods	fps	Comment	mean error
CLM	12	complex optimization for whole face	7.18
intraface	33	state of art	3.48
Proposed	17	highest accuracy	2.29
Harris Corner	24	simple feature method	3.38
Feng's	5	reported in [LOSS14]	not reported

Chapter 5

Conclusions

In this thesis, we aimed to implement a real-time gaze estimation system for ordinary user under free head motion. We employed appearance-based gaze estimation method because it needs only a single web camera and robust under relatively low resolution image. The main problems of appearance-based gaze estimation under free head motion are eye image distortion and eye alignment. We employed [LOSS14]'s method to learn gaze bias from head motion. In order to achieve accurate and fast eye alignment, we proposed a verification-based eye corner detection method. Experiment showed that our methods achieved good accuracy in eye corner detection and gaze estimation. Our system can run at a 16 fps, the gaze error is around 10° under 20° head pose motion.

The main contributions of this work are summarized as follow:

- We implemented an appearance-based gaze estimation system in real time, which can give reasonable result under free head motion.
- We proposed a weighted corner feature-based approach to detect eye corner, with a verification phase to decrease false detection.

Gaze error of our system is 10°, there are several possible reasons. Firstly, training samples have, secondly still eye tracking

For future work, we first consider to improve accuracy and robustness of our system as discussed in last section. For example, we may use different training method to obtain sufficient training sample in a user-friendly way. Besides, instead of a laptop, we consider to develop a real-time system on portable devices such as tablets or smart phones. Since user may have larger head motion more frequently, eye alignment and gaze compensation is more challenging. We are now using a commercial facetracker, in order to realize a open-source system, we consider to transfer our system to linux or ios platform, which is more accessible to mathematical library.

Bibliography

[BCVC13] Jose Javier Bengoechea, Juan J Cerrolaza, Arantxa Villanueva, and Rafael Cabeza. Evaluation of accurate eye corner detection methods for gaze estimation. In *Proc. 3rd International Workshop on Pervasive Eye Tracking and Mobile Eye-Based Interaction*, 2013.

[BV99] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999.

[CC08] David Cristinacce and Tim Cootes. Automatic feature localisation with constrained local models. *Pattern Recognition*, 41(10):3054–3067, 2008.

[CET98] Timothy F Cootes, Gareth J Edwards, and Christopher J Taylor. A comparative evaluation of active appearance model algorithms. In *BMVC*, volume 98, pages 680–689, 1998.

[CET01] Timothy F Cootes, Gareth J Edwards, and Christopher J Taylor. Active appearance models. *IEEE Transactions on pattern analysis and machine intelligence*, 23(6):681–685, 2001.

[CTCG95] Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham. Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995.

[FMR08] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[FY98] Guo-Can Feng and Pong Chi Yuen. Variance projection function and its application to eye detection for human face recognition. *Pattern Recognition Letters*, 19(9):899–906, 1998.

[HG09] Xia Haiying and Yan Guoping. A novel method for eye corner detection based on weighted variance projection function. In *Image and Signal Processing, 2009. CISP'09. 2nd International Congress on*, pages 1–4. IEEE, 2009.

[HK12] Corey Holland and Oleg Komogortsev. Eye tracking on unmodified common tablets: challenges and solutions. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 277–280. ACM, 2012.

[HYL+13] Shengfeng He, Qingxiong Yang, Rynson WH Lau, Jiang Wang, and Ming-Hsuan Yang. Visual tracking via locality sensitive histograms. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2427–2434. IEEE, 2013.

[KIUK13] Kai Kunze, Shoya Ishimaru, Yuzuko Utsumi, and Koichi Kise. My reading life: towards utilizing eyetracking on unmodified

tablets and phones. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, pages 283–286. ACM, 2013.

[LOSS14] Feng Lu, Takahiro Okabe, Yusuke Sugano, and Yoichi Sato. Learning gaze biases with head motion for head pose-free gaze estimation. *Image and Vision Computing*, 32(3):169–179, 2014.

[LSOS12] Feng Lu, Yusuke Sugano, Takahiro Okabe, and Yoichi Sato. Head pose-free appearance-based gaze sensing via eye image synthesis. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 1008–1011. IEEE, 2012.

[Por05] Fatih Porikli. Integral histogram: A fast way to extract histograms in cartesian spaces. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 829–836. IEEE, 2005.

[RCWS] Shaoqing Ren, Xudong Cao, Yichen Wei, and Jian Sun. Face alignment at 3000 fps via regressing local binary features.

[S+85] Satoshi Suzuki et al. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, 1985.

[Sar13] Jason M Saragih. Deformable face alignment via local measurements and global constraints. In *Deformation Models*, pages 187–207. Springer, 2013.

[SMSK08] Yusuke Sugano, Yasuyuki Matsushita, Yoichi Sato, and Hideki Koike. An incremental learning method for unconstrained gaze

estimation. In *Computer Vision–ECCV 2008*, pages 656–667. Springer, 2008.

[SP11] Gil Santos and Hugo Proença. A robust eye-corner detection method for real-world data. In *Biometrics (IJCB), 2011 International Joint Conference on*, pages 1–7. IEEE, 2011.

[SYW97] Rainer Stiefelhagen, Jie Yang, and Alex Waibel. A model-based gaze tracking system. *International Journal on Artificial Intelligence Tools*, 6(02):193–209, 1997.

[TKA02] Kar-Han Tan, David Kriegman, and Narendra Ahuja. Appearance-based eye gaze estimation. In *Applications of Computer Vision, 2002.(WACV 2002). Proceedings. Sixth IEEE Workshop on*, pages 191–195. IEEE, 2002.

[UFH12] Michal Uřičář, Vojtěch Franc, and Václav Hlaváč. Detector of facial landmarks learned by the structured output SVM. In Gabriela Csurka and José Braz, editors, *VISAPP '12: Proceedings of the 7th International Conference on Computer Vision Theory and Applications*, volume 1, pages 547–556, Portugal, February 2012. SciTePress — Science and Technology Publications.

[WB14] Erroll Wood and Andreas Bulling. Eyetab: model-based gaze estimation on unmodified tablet computers. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 207–210. ACM, 2014.

[WJ05] Peng Wang and Qiang Ji. Learning discriminant features for multi-view face and eye detection. In *Computer Vision and*

Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 1, pages 373–379. IEEE, 2005.

[WSV03] Jiangang Wang, Eric Sung, and Ronda Venkateswarlu. Eye gaze estimation from a single image of one eye. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 136–143. IEEE, 2003.

[XDlT13] Xuehan Xiong and Fernando De la Torre. Supervised descent method and its applications to face alignment. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 532–539. IEEE, 2013.

[ZLLT14] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In *Computer Vision–ECCV 2014*, pages 94–108. Springer, 2014.

[ZY02] Jie Zhu and Lei Yang. Subpixel eye gaze tracking. In *Automatic face and gesture recognition, 2002. proceedings. fifth ieee international conference on*, pages 124–129. IEEE, 2002.

[ZYWW05] Zhonglong Zheng, Jie Yang, Meng Wang, and Yonggang Wang. A novel method for eye features extraction. In *Computational and Information Science*, pages 1002–1007. Springer, 2005.