An Incremental Learning Method for Unconstrained Gaze Estimation

Yusuke Sugano[1,⋆], Yasuyuki Matsushita[2], Yoichi Sato[1], and Hideki Koike[3]

[1] The University of Tokyo
Tokyo, Japan
{sugano,ysato}@iis.u-tokyo.ac.jp
[2] Microsoft Research Asia
Beijing, China
yasumat@microsoft.com
[3] The University of Electro-Communications
Tokyo, Japan
koike@is.uec.ac.jp

Abstract. This paper presents an online learning algorithm for appearance-based gaze estimation that allows free head movement in a casual desktop environment. Our method avoids the lengthy calibration stage using an incremental learning approach. Our system keeps running as a background process on the desktop PC and continuously updates the estimation parameters by taking user's operations on the PC monitor as input. To handle free head movement of a user, we propose a pose-based clustering approach that efficiently extends an appearance manifold model to handle the large variations of the head pose. The effectiveness of the proposed method is validated by quantitative performance evaluation with three users.

1 Introduction

Gaze estimation is a process of detecting the position the eyes are looking at. It has been an active research topic in computer vision because of its usefulness for a wide range of applications, including human computer interaction, marketing studies and human behavior research. However, despite considerable advances in recent research, current gaze estimation techniques still suffer from many limitations. Creating an accurate gaze estimator that uses simple and low-cost equipment with allowing users to move their heads freely is still an open challenge.

Prior approaches are either model-based or appearance-based. Model-based approaches use an explicit geometric model of the eye, and estimate its gaze direction using geometric eye features. For example, one typical feature is the pupil-glint vector [1,2], the relative position of the pupil center and the specular reflection of a light source. While model-based approaches can be very accurate,

⋆ This work was done while the first author was visiting Mirosoft Research Asia.

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part III, LNCS 5304, pp. 656–667, 2008.

they typically need to precisely locate small features on the eye using a high-resolution image and often require additional light sources. This often results in large systems with special equipment that are difficult to implement in casual, desktop environments.

Appearance-based approaches directly treat an eye image as a high dimensional feature. Baluja and Pomerleau use a neural network to learn a mapping function between eye images and gaze points (display coordinates) using 2,000 training samples [3]. Xu *et al*. proposed a similar neural network-based method that uses more (3,000) training samples [4]. Tan *et al*. take a local interpolation approach to estimate unknown gaze point from 252 relatively sparse samples [5]. Recently, Williams *et al*. proposed a novel regression method called S^3GP (Sparse, Semi-Supervised Gaussian Process), and applied it to the gaze estimation task with partially labeled (16 of 80) training samples [6]. Appearance-based approaches can make the system less restrictive, and can also be very robust even when used with relatively low-resolution cameras.

Among model-based methods, one popular approach that handles head movements is to use multiple light sources and camera(s) to accurately locate 3D eye features. Shih and Liu used both multiple cameras and multiple lights for 3D gaze estimation [7]. Zhu *et al*. use a stereo camera setup with one light source to locate the 3D eye position and estimate 2D gaze positions by considering a generalized gaze mapping which is a function of the pupil-glint vector and the eye position [8,9]. Morimoto *et al*. propose a single camera method with at least two lights, but show only simulated results [10]. Hennessey *et al*. develop a similar system with multiple light sources to locate the 3D cornea center by triangulation, and compute the gaze point as the 3D intersection of the monitor surface and the optical axis of the eye [11]. Yoo and Chung use a structured rectangular light pattern and estimate the gaze point from the pupil's position relative to the rectangle [12]. Coutinho and Morimoto later extended this method with a more precise eye model [13].

In addition to 3D eye features, some methods also use 3D head pose information. Beymer and Flickner, for example, use a pair of stereo systems [14]. The first stereo system computes the 3D head pose, which is then used to guide a second stereo system that tracks the eye region. Matsumoto *et al*.'s method uses a single stereo system to compute the 3D head pose and estimate the 3D position of the eyeball [15]. A similar approach is also taken by Wang and Sung [16]. These approaches all require special equipment, preventing their use in casual environments.

Among the appearance-based approaches, little study has been dedicated to dealing with changes in head pose. Baluja *et al*.'s method allows some head movement by using training samples from different head poses, but the range of movement is limited. They describe two major difficulties. One is that the appearance of an eye gazing at the same point varies drastically with head motion. Additional information about the head pose is needed to solve this problem. The second difficulty is that the training samples must be collected across the pose space to handle the head movement. This results in a large number of training samples and an unrealistically long calibration period.

Our goal is to make a completely passive, non-contact, single-camera gaze estimation system that has no calibration stage yet still allows changes in head pose. To achieve this goal, we develop a new appearance-based gaze estimation system based on an online learning approach. We assume a desktop environment with a PC camera mounted on the monitor, and observe the fact that the user can be assumed to look at the mouse when he or she clicks. Our system incorporates recent advances in robust single-camera 3D head pose estimation to capture the user's head pose and the eye image continuously. By using the clicked coordinates as gaze labels, the system acquire learning samples in the background while users use the PC. Thus, it can learn the mapping between the eye and the gaze adaptively during operation, without a long preliminary calibration.

This work has the following major contributions:

- **An incremental learning framework.** To eliminate the lengthy calibration stage required for appearance-based gaze estimators with free head movement, we employ an incremental learning framework using the user's operations on the PC monitor.
- **A pose-based clustering approach.** We take a local interpolation-based approach to estimate the unknown gaze point. To use the gaze distance to constrain the appearance manifold in the pose-variant sample space, we propose a method using sample clusters with similar head poses and their local appearance manifold for the interpolation. The details are described in Section 3.

The outline of the rest of the paper is as follows. In Section 2 we describe the architecture of our system. Section 3 explains our incremental learning algorithm with local clusters. Section 4 provides experimental results, and Section 5 closes with a discussion of the potential of our method and future research directions.

2 Overview

Our gaze estimation system operates in a desktop environment with a user seated in front of a PC monitor, and with a PC camera mounted on the monitor. We assume that the user's gaze is directed at the mouse arrow on the monitor when he or she clicks the mouse, so we collect learning samples by capturing eye images and mouse arrow positions for all mouse clicks. The architecture of the system is summarized in Fig. 1. The input to the system is a continuous video stream from the camera. The 3D model-based head tracker [17] continuously computes the head pose, $\boldsymbol{p}$, and crops the eye image, $\boldsymbol{x}$, as shown in Fig. 2. At each mouse click, we create a training sample by using the mouse screen coordinate as the gaze label $\boldsymbol{g}$ associated with the features (head pose $\boldsymbol{p}$ and eye image $\boldsymbol{x}$). Using this labeled sample, our system incrementally updates the mapping function between the features and the gaze. This incremental learning is performed in a reduced PCA subspace, by considering sample clusters and the local appearance manifold. The details are described in Section 3. When the user is not using the mouse, the system runs in a prediction loop, and the gaze is estimated using the updated mapping function.

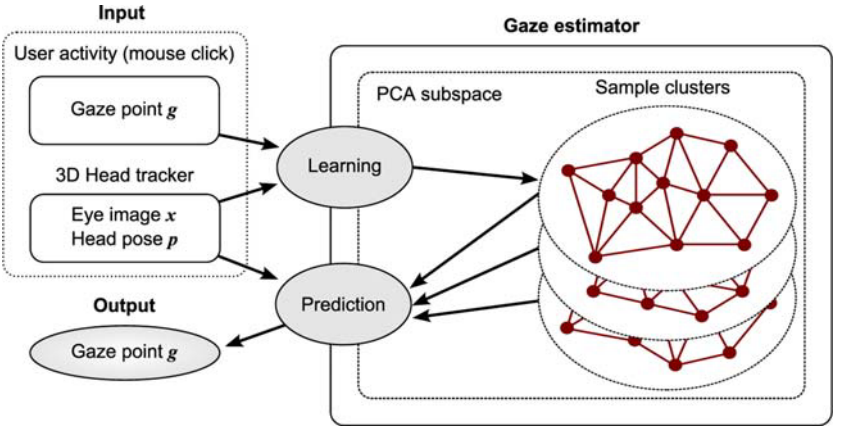

Fig. 1. Learning and prediction flow of the proposed framework

2.1 Head Tracking and Eye Capturing

Here we describe in detail how we capture input features. As stated above, our framework uses the head tracking method of Wang *et al.* [17]. The tracker estimates the head pose of a person from a single camera, using a 3D rigid facial mesh. The tracker outputs the user's 7-dimensional head pose $\boldsymbol{p} = (\boldsymbol{t}^T, \boldsymbol{r}^T)^T$, where $\boldsymbol{t}$ is a 3-dimensional translation and $\boldsymbol{r}$ is a 4-dimensional rotation vector defined by four quaternions. The facial mesh in Fig. 2(a) shows an estimated head pose.

The system converts the input image to gray-scale, then crops the eye region as follows. First, it extracts an eye region (the rectangle in Fig. 2(a)) that is predefined on the facial mesh. It then applies a perspective warp to crop the region as a fixed size rectangle. Fig. 2(b) shows the warped result, $\boldsymbol{I}_{src}$. The offset of the eye is still too large for this image to be a useful feature.

We reduce this offset error using a two-stage alignment process. For the initial alignment, we apply a vertical Sobel filter to the cropped image, then threshold to create a binary image. We then crop a $W_1 \times H_1$ image region $\boldsymbol{I}_1$ from $\boldsymbol{I}_{src}$ such that the cropped image center corresponds to the average coordinate of the edge points. At this time, we also do some preliminary image processing. We apply histogram equalization to the image, then truncate higher intensities to eliminate the effects of illumination changes. We also apply a bilateral filter to reduce image noise while preserving edges. After this, we improve the alignment using image subspaces. Specifically, we choose the $W_2 \times H_2$ ($W_2 < W_1$ and $H_2 < H_1$) image region $\boldsymbol{I}_2$ that minimizes the reconstruction error

$$\mathcal{E} = ||\boldsymbol{I}_2 - \acute{\boldsymbol{I}}_2||^2. \tag{1}$$

Here $\acute{\boldsymbol{I}}_2$ is the approximated version of $\boldsymbol{I}_2$ using the PCA subspace. As described later, this subspace is updated incrementally using labeled samples.

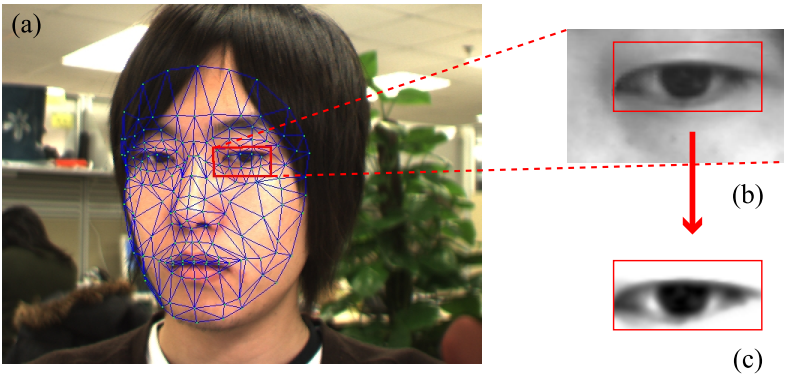

Fig. 2. Capturing results. (a) Head pose estimation result. (b) Cropping result around predefined eye region on the facial mesh (the rectangle in (a)). (c) Eye alignment and image preprocessing result (the image feature used in our gaze estimator.)

Fig. 2(c) shows the final result of the cropping process, raster-scanned to create an image vector $\boldsymbol{x}$. In our experiment, the size of the final image is set to $W_2 = 75 \times H_2 = 35$ pixels, so $\boldsymbol{x}$ is 2625-dimensional. Finally, we compose the feature vector $\boldsymbol{f} = (\boldsymbol{x}^T, \boldsymbol{p}^T)^T$, which consists of the eye image $\boldsymbol{x}$ and the head pose $\boldsymbol{p}$.

3 Gaze Estimation

The goal of our gaze estimator is to learn the mapping between the feature vector $\boldsymbol{f}$ and the gaze $\boldsymbol{g}$. We use a local linear interpolation method similar to [5,18]. Given an unlabeled feature $\dot{\boldsymbol{f}}$, we predict the unknown label $\dot{\boldsymbol{g}}$ by choosing k nearest neighbors from the labeled samples and interpolating their labels using distance-based weights. For our application, it is critical to choose neighbors from a manifold according to the gaze variation. Tan *et al.* [5] use 2D topological information about the gaze points as constraints. Two points are assumed to be neighbors on the manifold if they are also neighbors in the gaze space. However, this assumption is not always satisfied in our case, because there can be many samples which have different head poses but the same gaze point. To overcome this problem, we construct sample clusters with similar head poses and consider a local manifold for each sample cluster. The architecture of the clusters is partially inspired by Vijayakumar *et al.*'s work [19].

The distance measure of the cluster, i.e., how close the head pose and the cluster are, is defined as a Gaussian function:

$$g(\boldsymbol{p}) = \prod_i \frac{1}{\sqrt{2\pi\kappa_g\sigma_{p,i}^2}} \exp\left(-\frac{(p_i - \bar{p}_i)^2}{2\kappa_g\sigma_{p,i}^2}\right), \tag{2}$$

Algorithm 1. Cluster-based gaze estimation.

Learning: given tth labeled sample $\boldsymbol{f}_t = (\boldsymbol{x}_t^T, \boldsymbol{p}_t^T)^T$ and $\boldsymbol{g}_t$
  Update image subspace using incremental PCA: mean $\bar{\boldsymbol{x}}^{(t)}$, eigenvectors $\boldsymbol{U}^{(t)}$, eigenvalues $\boldsymbol{\lambda}^{(t)}$, coefficients $\boldsymbol{A}^{(t)}$. $\boldsymbol{x}_t \approx \bar{\boldsymbol{x}}^{(t)} + \boldsymbol{U}^{(t)}\boldsymbol{a}_t$.
  for $k = 1$ to K **do**
    if $g_k(\boldsymbol{p}_t) > \tau_g$ **then**
      Add sample to the cluster
    end if
  end for
  if No $g_k(\boldsymbol{p}_t)$ is above threshold **then**
    Create new $K + 1$th cluster and add sample
  end if

Prediction: given unlabeled feature $\dot{\boldsymbol{f}} = (\dot{\boldsymbol{x}}^T, \dot{\boldsymbol{p}}^T)^T$.
  Project image $\dot{\boldsymbol{x}}$ into current subspace: $\dot{\boldsymbol{a}} = \boldsymbol{U}^{(t)T}(\dot{\boldsymbol{x}} - \bar{\boldsymbol{x}}^{(t)})$
  for $k = 1$ to K **do**
    Calculate interpolated gaze $\dot{\boldsymbol{g}}_k$ and a prediction confidence c_k
  end for
  Get final prediction as a weighted sum: $\dot{\boldsymbol{g}} = \sum_k c_k \dot{\boldsymbol{g}}_k / \sum_k c_k$.

where p_i is the ith element of the pose $\boldsymbol{p}$, and $\bar{p}_i$ and $\sigma_{p,i}^2$ are the corresponding average and variance calculated from the samples contained in the cluster. The constant weight κ_g is empirically set. The framework is outlined in Algorithm 1. Given a labeled sample, the image $\boldsymbol{x}_t$ is first used to update the PCA subspace. We use the algorithm of Skocaj *et al.*[20] to update all stored coefficients $\boldsymbol{a}_1 \ldots \boldsymbol{a}_t$. After updating the subspace, the sample is added to all clusters whose weight $g_k(\boldsymbol{p}_t)$ is higher than the predefined constant threshold τ_g. In Algorithm 1, K is the total number of clusters at the time. If no suitable clusters are found, we create a new cluster containing only the new sample. Given an unlabeled feature, the output gaze $\dot{\boldsymbol{g}}$ is calculated as a weighted sum of predictions from each cluster. The following sections describe the processes executed in each cluster.

3.1 Learning

Here, we describe the learning process in detail. As stated above, the labeled sample $s_t = \{\boldsymbol{a}_t, \boldsymbol{p}_t, \boldsymbol{g}_t\}$ is added to a cluster only when its pose $\boldsymbol{p}_t$ is sufficiently close to the cluster average. However, this rule cannot reject outliers (e.g., a mouse click without user's attention). Moreover, the sample density can increase too much if all samples are stored in the cluster. For interpolation, the sample distribution in the gaze space does not have to be too dense. For these reasons, we introduce another constraint on the local linearity between the image distance $d_a^{(i,j)} = ||\boldsymbol{a}_i - \boldsymbol{a}_j||$ and the gaze distance $d_g^{(i,j)} = ||\boldsymbol{g}_i - \boldsymbol{g}_j||$. We define a linearity measure $l(s_i)$ for the sample s_i as the correlation between $d_a^{(i,j)}$ and $d_g^{(i,j)}$ among $\{s_j | d_g^{(i,j)} < r_1\}$. Here, r_1 is the distance threshold. The sample selection rule

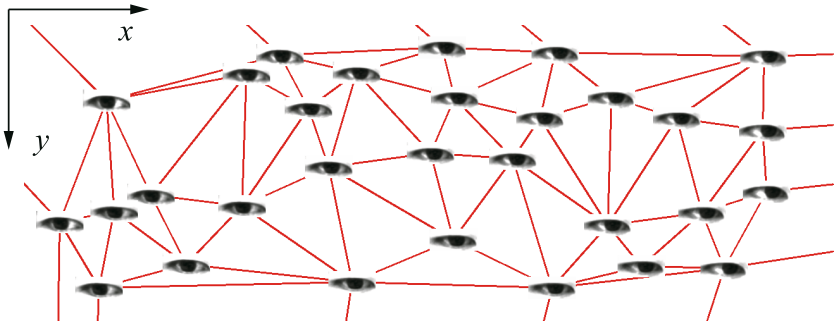

Fig. 3. Gaze triangulation example shown in the screen coordinates. Each eye image (flipped horizontally for presentation clarity) is located at the corresponding gaze point, and the lines indicate Delaunay edges between these gaze points.

is as follows. If there is more than one sample around the new sample s_t, i.e., $\{s_j | d_g^{(t,j)} < r_2\} \neq \emptyset$ $(r_2 < r_1)$, keep the sample with the highest $l(s)$ and reject the others. The threshold r_2 controls the sample density and should be chosen according to the size of the target display area and the memory capacity.

Next, we update cluster mean $\bar{\boldsymbol{p}}_k$ and the variance $\boldsymbol{\sigma}_k^2$ (in Eq.(2)) to fit the current sample distribution. Furthermore, we compute a Delaunay triangulation of the gaze point for the current point set. Fig. 3 shows an example triangulation. Each point corresponds to the 2D coordinate of the gaze point. This topological information is used in the prediction process.

3.2 Prediction

When the unlabeled data $\dot{\boldsymbol{a}}_t$ and $\dot{\boldsymbol{p}}_t$ are given to the cluster, the system predicts the unknown gaze $\dot{\boldsymbol{g}}_k$ by interpolation.

The neighbors to be used for interpolation are chosen on the manifold. The system selects a closest triangle (concerning an average distance d_a between three vertices) to $\dot{\boldsymbol{a}}_t$ to the local triangulation. Points adjacent to this triangle are also chosen as neighbors.

Using the chosen set $\mathcal{N}$, interpolation weights $\boldsymbol{w}$ are calculated to minimize a reconstruction error:

$$\boldsymbol{w} = \underset{\boldsymbol{w}}{\operatorname{argmin}} ||\dot{\boldsymbol{a}} - \sum_{i \in \mathcal{N}} w_i \boldsymbol{a}_i||^2, \tag{3}$$

subject to

$$\sum_{i \in \mathcal{N}} w_i = 1. \tag{4}$$

w_i denotes the weight corresponds to the ith neighbor. Finally, under the assumption of local linearity, the gaze $\dot{\boldsymbol{g}}_k$ is interpolated as

$$\dot{\boldsymbol{g}}_k = \sum_{i \in \mathcal{N}} w_i \boldsymbol{g}_i. \tag{5}$$

To reject the outliers from clusters that do not contain sufficient samples, we define an interpolation reliability measure that represents how well the input $\dot{\boldsymbol{a}}$ is described by the selected neighbors:

$$r(\dot{\boldsymbol{a}}) = \exp\left(-\sum_i \frac{(\dot{a}_i - \bar{a}_i)^2}{2\kappa_r \sigma_{a,i}^2}\right), \tag{6}$$

where $\dot{a}_i$ is the ith element of $\dot{\boldsymbol{a}}$, and $\bar{a}_i$ and $\sigma_{a,i}^2$ are average and variance of the corresponding element among the neighbors $\mathcal{N}$. The factor κ_r is empirically set. The prediction confidence c_k of the cluster is defined as a product of the reliability $r(\dot{\boldsymbol{a}})$ and the distance $g(\dot{\boldsymbol{p}})$:

$$c_k = r(\dot{\boldsymbol{a}}) \cdot g(\dot{\boldsymbol{p}}). \tag{7}$$

The final prediction result $\dot{\boldsymbol{g}}$ is calculated as a weighted sum of $\dot{\boldsymbol{g}}_k$, based on c_k. The value $r(\dot{\boldsymbol{a}})$ is useful for measuring the reliability of the current estimate, so we also output the cluster-weighted average of $r(\dot{\boldsymbol{a}})$:

$$\bar{r}(\dot{\boldsymbol{p}}, \dot{\boldsymbol{a}}) = \sum_k g_k(\dot{\boldsymbol{p}}) r_k(\dot{\boldsymbol{a}}) / \sum_k g_k(\dot{\boldsymbol{p}}). \tag{8}$$

4 Experiments

We have conducted some experiments to evaluate our system and the effect of the proposed cluster-based learning method. Our system consists of VGA resolution color camera (a PointGrey Dragonfly) and a Windows PC with a 3.00GHz CPU and 1GB of RAM. In current implementation, the whole process runs at about 10 fps.

In our experiments, no special pattern is used to indicate the learning positions. Users are simply asked to randomly click on the desktop region without any unusual attention, but looking at the mouse pointer. This means the experimental situation is reasonably close to real behavior. During the 10-minute experiment, users are allowed to freely move their heads. Table 1 shows the actual range of head motion for each user during the experiments. The estimation error is evaluated at each time t when a new labeled sample is acquired. Before

Table 1. Actual range of head motion for each target user. The rotation is defined as a quaternion $q_w + q_x\boldsymbol{i} + q_y\boldsymbol{j} + q_z\boldsymbol{k}$.

	Translation [mm]			Rotation			
	x	y	z	q_w	q_x	q_y	q_z
Person A	170	47	169	0.134	0.011	0.058	0.202
Person B	220	54	203	0.211	0.027	0.342	0.351
Person C	142	32	134	0.242	0.019	0.126	0.277

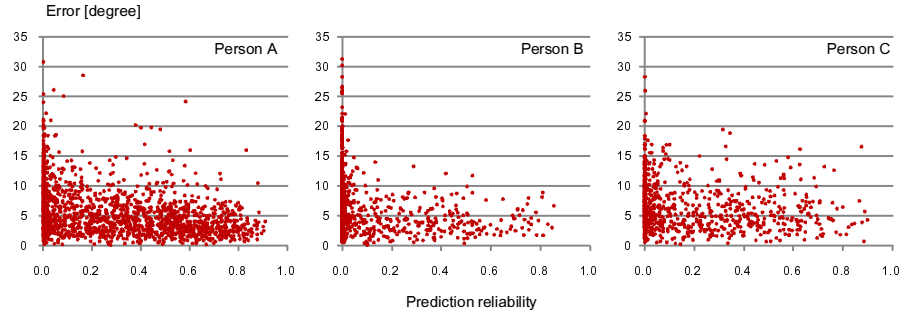

Fig. 4. Angular error against prediction reliability. Each graph shows the scatter plot of the estimation error versus the reliability we defined in Eq.(6) and Eq.(8).

adding it to the sample clusters, the angular error θ_t between the true (clicked) gaze position $\boldsymbol{g}_t$ and the estimated position $\dot{\boldsymbol{g}}_t$ (interpolation based on the past labeled samples) is calculated as:

$$\theta_t = \tan^{-1}\left(\frac{D_m(\boldsymbol{g}_t, \dot{\boldsymbol{g}}_t)}{p_{z,t} - d_{\mathrm{cam}}}\right), \tag{9}$$

where D_m indicates the distance between two points in the metric unit, $p_{z,t}$ is the depth element of the estimated head pose, and d_{cam} is the pre-calculated distance between the camera and the display.

First, Fig. 4 shows the angular error θ_t plotted against the reliability $\bar{r}_t$. We see that the estimation accuracy increases as our reliability measure increases, and using this measure we can reject the outliers that have large error. Low reliability estimates are caused by the absence of labeled samples around the estimated gaze position. It can be improved if new labeled samples are added around there, and results with low reliability can be ignored by the system when a reliable prediction is needed.

Fig. 5 shows the cumulative weighted average (C.W.A.) error $\sum_{i=i}^{t} \bar{r}_i\theta_i / \sum_{i=i}^{t} \bar{r}_i$ over time. We also show the result of another experiment (the left lower graph) to validate our cluster-based approach. For this experiment, the system did not create pose clusters, which is equivalent to normal appearance-based estimation with pose-varying input. Naturally, the C.W.A. error gradually increases. By contrast, even if the user moves his/her head, the estimation error of our method (shown in other graphs) does not increase and converges to a certain range. This shows the effectiveness of our cluster-based solution.

Table 2 shows the average estimation error for three users. The left column is the normal average error, and the center column is the weighted average error throughout experiment. The right column indicates the number of points clicked during the experiments. The angular error of our gaze estimation is roughly $4 \sim 5$ degrees. This accuracy may not be sufficient to replace a mouse with our method as a user input, however, it is helpful to achieve our goal, i.e., to estimate the approximate region where the user is looking at.

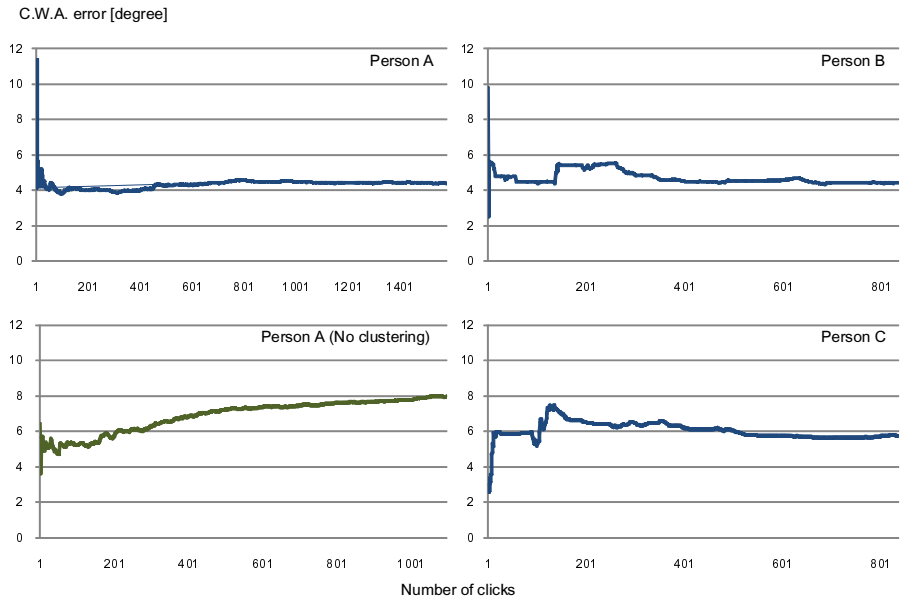

Fig. 5. Time variation of the cumulative weighted average (C.W.A.) estimation error. The lower left graph is the result without the cluster-based solution. The other graphs show results using our method for three different users.

Table 2. Estimation error. The left column is the normal average error, and the center column is the weighted average error throughout experiment. The right column indicates the number of points clicked during the experiments.

	Average error [deg]		Number of
	Normal	Weighted	clicked points
Person A	5.6	4.4	1796
Person B	7.1	4.4	1748
Person C	6.6	5.8	1095

5 Conclusions

We have proposed a gaze estimation system that learns incrementally as the user clicks the mouse. When the user clicks somewhere on the display, our system uses the captured eye image and the head pose to learn the mapping function, with the clicked coordinate as learning label. Moreover, we extended an appearance interpolation-based method to the case with free head movement, by clustering learning samples with similar head poses and constructing their local manifold model. We showed the efficiency and reasonable estimation accuracy of our method through the experiments in actual environment.

Because our method wholly relies on the image distance between samples, the estimation accuracy mostly depends on the accuracy of the distance measurement. We employed PCA-based distance measure for the sake of computational efficiency and implementation simplicity. However, it can be too sensitive to the appearance variation not related to the gaze, such as cropping shift and rotation. To diminish this effect, we conducted subspace-based eye alignment after cropping. Even so, there can be slight jitter and drift of the result. Thus, the estimation accuracy of our method can be improved by using more precise alignment, or shift-invariant distance measure method.

Also, we should mention about the memory efficiency of our method. Since there are no scheme to adjust the number of the sample clusters, memory usage and computational cost can increase unlimitedly in theory. We verified that it does not become a major issue in the case of usual desktop environment, but some kind of reformation will be needed when it is applied to more general situation with a wide range of head pose. This is partially achieved by wider cluster kernel (κ_g in Eq.(2)) or higher threshold to create clusters (τ_g in Algorithm 1).

In future work, we plan to extend this framework to higher level regression in the pose space. Our system is less accurate compared to state-of-the-art gaze estimation methods with an accuracy of less than 1 degree, however it has great advantage that it allows for free head movement and works with minimal equipment; single camera without additional light source. It has considerable potential for developing practically ideal gaze estimator, with further investigations.

Acknowledgement

This research was supported by the Microsoft Research IJARC Core Project.

References

1. Hutchinson, T.E., White Jr., K.P., Martin, W.N., Reichert, K.C., Frey, L.A.: Human-computer interaction using eye-gaze input. IEEE Transactions on Systems, Man and Cybernetics 19(6), 1527–1534 (1989)
2. Jacob, R.J.: What you look at is what you get: eye movement-based interaction techniques. In: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 11–18 (1990)
3. Baluja, S., Pomerleau, D.: Non-intrusive gaze tracking using artificial neural networks. Advances in Neural Information Processing Systems (NIPS) 6, 753–760 (1994)
4. Xu, L.Q., Machin, D., Sheppard, P.: A novel approach to real-time non-intrusive gaze finding. In: Proceedings of the British Machine Vision Conference, pp. 428–437 (1998)
5. Tan, K.H., Kriegman, D.J., Ahuja, N.: Appearance-based eye gaze estimation. In: Proceedings of the Sixth IEEE Workshop on Applications of Computer Vision (WACV 2002), pp. 191–195 (2002)
6. Williams, O., Blake, A., Cipolla, R.: Sparse and semi-supervised visual mapping with the S^3GP. In: Proceedings of the 2006 IEEE Conference on Computer Vision and Pattern Recognition, pp. 230–237 (2006)

7. Shih, S.W., Liu, J.: A novel approach to 3-d gaze tracking using stereo cameras. IEEE Transactions on Systems, Man and Cybernetics, Part B 34(1), 234–245 (2004)
8. Zhu, Z., Ji, Q.: Eye gaze tracking under natural head movements. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2005), vol. 1, pp. 918–923 (2005)
9. Zhu, Z., Ji, Q., Bennett, K.P.: Nonlinear eye gaze mapping function estimation via support vector regression. In: Proceedings of the 18th International Conference on Pattern Recognition (ICPR 2006), vol. 1, pp. 1132–1135 (2006)
10. Morimoto, C., Amir, A., Flickner, M.: Detecting eye position and gaze from a single camera and 2 light sources. In: Proceedings of the 16th International Conference on Pattern Recognition (ICPR 2002), pp. 314–317 (2002)
11. Hennessey, C., Noureddin, B., Lawrence, P.: A single camera eye-gaze tracking system with free head motion. In: Proceedings of the 2006 symposium on Eye tracking research & applications, pp. 87–94 (2006)
12. Yoo, D.H., Chung, M.J.: A novel non-intrusive eye gaze estimation using cross-ratio under large head motion. Computer Vision and Image Understanding 98(1), 25–51 (2005)
13. Coutinho, F.L., Morimoto, C.H.: Free head motion eye gaze tracking using a single camera and multiple light sources. In: Proceedings of the Brazilian Symposium on Computer Graphics and Image Processing, pp. 171–178 (2006)
14. Beymer, D., Flickner, M.: Eye gaze tracking using an active stereo head. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2003), vol. 2, pp. 451–458 (2003)
15. Matsumoto, Y., Ogasawara, T., Zelinsky, A.: Behavior recognition based on head pose and gaze direction measurement. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000), vol. 3, pp. 2127–2132 (2000)
16. Wang, J.G., Sung, E.: Study on eye gaze estimation. IEEE Transactions on Systems, Man and Cybernetics, Part B 32(3), 332–350 (2002)
17. Wang, Q., Zhang, W., Tang, X., Shum, H.Y.: Real-time bayesian 3-d pose tracking. IEEE Transactions on Circuits and Systems for Video Technology 16(12), 1533–1541 (2006)
18. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500), 2323–2326 (2000)
19. Vijayakumar, S., D'Souza, A., Schaal, S.: Incremental online learning in high dimensions. Neural Computation 17(12), 2602–2634 (2005)
20. Skocaj, D., Leonardis, A.: Weighted and robust incremental method for subspace learning. In: Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003), pp. 1494–1501 (2003)