Gaze Guided Object Recognition Using a Head-Mounted Eye Tracker

Takumi Toyama* Thomas Kieninger† Faisal Shafait‡
Andreas Dengel§
German Research Center for Artificial Intelligence (DFKI) GmbH, Kaiserslautern, Germany

Abstract

Wearable eye trackers open up a large number of opportunities to cater for the information needs of users in today's dynamic society. Users no longer have to sit in front of a traditional desk-mounted eye tracker to benefit from the direct feedback given by the eye tracker about users' interest. Instead, eye tracking can be used as a ubiquitous interface in a real-world environment to provide users with supporting information that they need. This paper presents a novel application of intelligent interaction with the environment by combining eye tracking technology with real-time object recognition. In this context we present i) algorithms for guiding object recognition by using fixation points ii) algorithms for generating evidence of users' gaze on particular objects iii) building a next generation museum guide called Museum Guide 2.0 as a prototype application of gaze-based information provision in a real-world environment. We performed several experiments to evaluate our gaze-based object recognition methods. Furthermore, we conducted a user study in the context of Museum Guide 2.0 to evaluate the usability of the new gaze-based interface for information provision. These results show that an enormous amount of potential exists for using a wearable eye tracker as a human-environment interface.

CR Categories: H.5.2 [INFORMATION INTERFACES AND PRESENTATION]: User Interfaces—Prototyping;

Keywords: object recognition, eye tracking, augmented reality

Links: DL PDF

1 Introduction

Research over the last century has contributed to understanding the nature of human attention by analyzing eye movements using eye tracking [Buswell 1935; Henderson 2003; Yarbus 1967; Kovic et al. 2009]. As a result, eye tracking itself has emerged as a new technology to interact with computers. Since people generally prefer simple and intuitive interaction mechanisms to complicated or incomprehensible ones, any kind of interface available today could possibly be replaced by a simpler and more intuitive one. From this viewpoint eye tracking is a highly remarkable technology due to its immediate connection to human intuition.

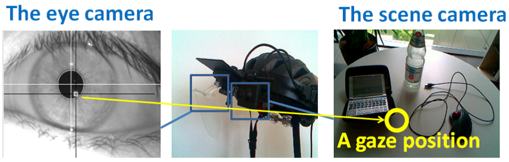


Figure 1: *The SMI head-mounted eye tracker and sample views of both cameras.*

In recent years, gaze on a computer display is employed as an interactive interface in a wide range of applications such as reading of text [Biedert et al. 2010], communicating with a virtual character [Bee et al. 2010], typing [Majaranta et al. 2009] and so on.

Wearable eye trackers available today provide a lot of opportunities to interact with the environment around us in an intelligent way, for instance by using eye tracking with Augmented Reality (AR). AR presents a view of the real world whose elements are "augmented" by computers in several ways (such as embedding signs, sounds, etc.). Recent smartphone applications like Wikitude[1] or Google Goggles[2] present a platform to overlay information about things in the real world onto a mobile phone display.

These advances in technology are due to research in image based object recognition - which is also one of the most rapidly developing research fields in recent years. The objective of image based object recognition is to recognize the objects present in an image or in a video stream in the same way as humans do. Early studies in object recognition started to employ global features such as color or texture histograms [Haralick et al. 1973]. However, since such global features are not robust enough to illumination or perspective changes and occlusions, methods based on local features took over [Zhang et al. 2007]. Local features, which are extracted from small patches of an image are widely utilized nowadays [Li and Allinson 2008]. In particular SIFT (Scale-invariant feature transform) [Lowe 2004] is widely accepted due to its invariance to scale, orientation, and affine distortion. Based on these methods, recognition systems can be developed that have excellent robustness against lighting and position variations, background changes, and partial occlusion [Roth 2008; Ponce et al. 2007].

In this paper, we investigate how human gaze can be used as a new interface for AR applications. First, we develop algorithms for guiding object recognition by using fixation points. Then, we present how to detect users' gaze in the context of an AR application, given raw eye tracking data and the corresponding object recognition results. Finally, we develop a novel AR application named *Museum Guide 2.0* that utilizes eye tracking as an interactive interface and recognizes objects in a real environment to demonstrate the application of our algorithms in practice. For this purpose, we utilize a head-mounted eye tracker to capture a view of the real world. One camera captures images of the user's eye, while the other captures the scene in front of him as shown in Figure 1.

There are some related works that also integrate the object

[*] e-mail: Takumi.Toyama@dfki.de
[†] Thomas.Kieninger@dfki.de
[‡] Faisal.Shafait@dfki.de
[§] Andreas.Dengel@dfki.de

[1] http://www.wikitude.org
[2] http://www.google.com/mobile/goggles/

recognition system with an eye tracking application, such as [Ishiguro et al. 2010; Bonino et al. 2009]. However, the evaluation of the benefits of the integration is not discussed deeply in these previous works. Here, we present a new approach for triggering the infomation provision and the evaluation of the approach including the user study in a practical use-case.

The basic idea of *Museum Guide 2.0* is that visitors of a museum would wear a head mounted eye tracker while strolling through the exhibition. Whenever the user watches any of the exhibits for a certain duration, the system automatically presents corresponding AR meta-information in a certain way (just like a personal human guide might do). The main considerations that inspired us to this application are:

- All exhibits are known and sufficient training data is available.
- Exhibits are well illuminated and the backgrounds are static and not cluttered.
- Users watch exhibits from typical perspectives only.

Considering these aspects, we can start from a restricted scenario that still contains a lot of challenges.

Museum Guide 2.0 works as follows: When a fixation is detected, the image from the scene camera is piped to the object recognition framework, which returns the name of the object the fixation points to. Our gaze detection algorithm judges if the user's gaze is on this very object. In this case, Museum Guide 2.0 presents AR to the user. In this paper, we only use pre-recorded voice data containing additional information about the respective object for informaiton provision. However, the way of providing information to the user can also be alternated by another AR form, such as information overlay using a smart phone display or so on. Once the voice data starts playing, the user would usually concentrate and listen to it. Consequently, the user's gaze position may remain on the same object or unconsciously move away from the object. Note that these actions must not trigger another presentation. Until gaze on another object is detected, the system keeps playing the audio data.

Although there are many different types of exhibits in a real museum, in this paper we only deal with *3D*, *small-sized* objects. This is because of the following two reasons. First, the shape of a 3D object is generally more complicated than that of a 2D object, thus it is a more general environment to test the overall performance of the application. Second, the entire shape of small-sized objects can mostly be captured in one frame whereas large-sized objects typically require several frames to be captured completely when working with a fixed focal length camera. To deal with such large-sized objects, we need a sort of compensation algorithm which adds the required functionality to the application.

In Section 2 we describe the method we used in Museum Guide 2.0. The experiments we conducted to evaluate the system are explained in Section 3. In Section 4 we present our conclusion.

2 Method

2.1 Eye tracking

The goal of eye tracking is to find out what a person is looking at by monitoring the eyes of that person. We used the iView X^{TM} HED[3] as a head-mounted monocular eye tracker in this work and therefore briefly describe its working principle in the following paragraph.

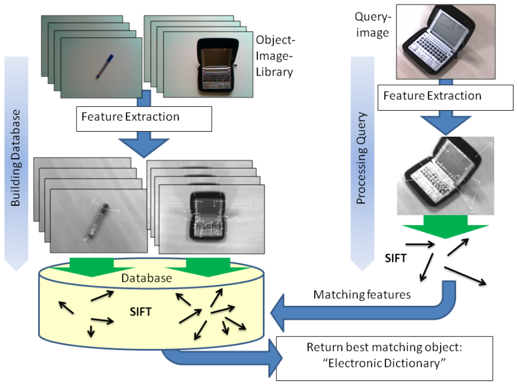


Figure 2: *Object recognition process. SIFT features are extracted from images and matched between the database and the query.*

The iView X^{TM} system employs the dark pupil system for eye tracking. In a dark pupil system the eye is illuminated by an infrared (IR) LED. An IR-sensitive camera watches this eye from a fixed position (relative to the head) from a close distance. To avoid visual disturbance for the user, LED and camera are mirrored at a transparent glass in front of the eye and are mounted outside the field of vision above the eyes. The eye and the face reflect this illumination but the pupil absorbs most IR light and appears as a high contrast dark ellipse. The image-analysis software provided by the iView X^{TM} system determines where the center of the pupil is located and this is mapped to gaze position via an eye-tracking algorithm. Dark pupil systems are versatile and easier to set up though they also require head movement compensation.

The iView X^{TM} requires a calibration process for each user in order to establish the relationship between the position of the eye in the camera view and a gaze point in space. We use a five-point calibration which indicates 5 respective points which the user has to watch owing to its quickness and accuracy. Once this process is done appropriately, we can obtain an accurate estimate of the gaze position.

The sampling rate of the iView X^{TM} HED is 50 Hz and its gaze position accuracy is $0.5°$ - $1.0°$ (typ.).

2.2 Basic Object Recognition Method

In the object recognition framework, we adopt SIFT as the feature extraction method. These features are used to find the best matched object to a query from the database. To acquire a fast computation, we also use Approximate Nearest Neighbour (ANN)[4] for matching features. A brief model of this recognition process is shown in Figure 2. First of all, we build a database consisting of features from images of all objects to be displayed in the museum. Object recognition is processed by finding the most similar feature from the database when a feature from the query is given. The name of the object which has the majority of matched features is returned as the result.

To detect interest points of SIFT, a given image is filtered by Gaussian Kernel with multiple scales of Gaussian parameter σ. Consequently, interest points are detected as the maxima/minima of the Difference of Gaussian (DoG). For further information on the SIFT features, we like to refer to [Lowe 2004].

Since we assume the museum scenario, pictures for building up the

[3] http://www.smivision.com/en/gaze-and-eye-tracking-systems/products/iview-x-hed.html

[4] http://www.cs.umd.edu/ mount/ANN/

database can be taken under the same conditions that are later given for the runtime system as described in Section 1, i.e. with the same illumination and spatial arrangement.

The following step describes the process of building a database. For all objects,

1. Place the object on a table in the museum.
2. A person wearing the iView X^{TM} HED walks around the table, thereby directing the scene camera to the object.
3. Record the video data from the scene camera.
4. Select images taken of different views from the video stream and extract SIFT features from the images.
5. Label images with the name of the object.

The number of extracted SIFT features varies for each object. To avoid disproportion in the number of indexed SIFT features for each object, we use more images from the objects having fewer SIFT features.

In the recognition phase, the image of an unknown object is given as the query to the retrieval system. After extracting SIFT features from the query image, our object recognition method matches them to the closest features from the database in the feature space. For the measurement of the distance, we use Euclidian metric. The identity of the closest match for each feature is retrieved from the database and a histogram is built representing how frequently a particular identity was returned. The histogram is normalized to unit length in order to remove disproportion of the number of features. In case the highest value in the histogram exceeds a threshold, the identity of the corresponding object is returned as the recognition result. If none of the entries in the histogram exceeds the threshold, no recognition result is returned.

The computational cost of the presented object recognition method using local descriptors not only depends on the number of features to be matched for each image, but also on the number of stored/indexed features in the database. The larger the number of indexed features, the longer the processing time. To demonstrate in a real-time environment, it is necessary to reduce the computational cost of matching. We use ANN to lower computational costs for matching features. In ANN, the nearest feature to a query feature in the database is returned with a certain error bound ϵ [Indyk and Motwani 1998]. As the value of ϵ increases, the retrieval becomes faster but probability of error also gets higher. Thus, it is required to find a suitable value for ϵ depending on the size of the database and the number of queried features per image.

2.3 Real-Time Gaze-Based Object Recognition

In this section, we describe *real-time gaze-based object recognition*. The main objective here is to develop a computational model which detects the existence of the user's gaze on particular objects using fixation information and images from the eye tracker and to meet real-time requirements.

Although Museum Guide 2.0 is a simple and uncomplicated scenario, it contains many challenges:

First, a significant difference to an ordinary camera or image based object recognition framework is that we can obtain the user's fixation point which is directly connected to the user's interest point in the image. By taking advantage of this fact, we extend the basic object recognition method to *fixation guided object recognition* in order to improve the performance of the recognition.

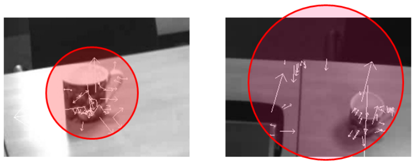


Figure 3: *An illustration showing regions around the fixation point containing a fixed number of feature points. The circles in the images represent the approximate region in each image. Note that the distribution varies for each object depending on the complexity of the object.*

Secondly, the most primitive way to evaluate the recognition system is to judge if the system outputs the correct name of the object indicated by the fixation for *each frame*. However, this kind of evaluation does not consider a well-known problem, the so-called Midas Touch problem [Jacob 1993]. Since the eyes are one of our key perceptual organs, they provide a large amount of information. Besides, the movements of the eye (fixations and saccades) strongly reflect the mental process of viewing. Many saccades and fixations are not caused by the user on purpose but are rather subconscious. If the application responds for each frame individually, the overflow of the user with irrelevant information would not lead to any acceptance towards the application. Therefore, we need to define another criteria to evaluate the system based on the user's gaze which can be observed as a sequence of fixations on a particular object rather than a fixation for each frame. Furthermore, to satisfy such a gaze-based evaluation method, we propose the methods to detect the existence of gaze on a particular object by using the object recognition result of consecutive frames.

Lastly, since image processing generally requires high computational cost, we need to reduce it when we apply the system in a real-time environment, where system reactions to user behavior should be triggered with minimal delay. The majority of processing time however is occupied by the SIFT feature extraction and matching. While the processing for one fixated image area is done by the system, other fixations might occur in the meanwhile. Queuing these events for later processing is not suitable for a real-time system. To catch up with real-time, we propose a compensation approach.

2.3.1 Fixation Guided Object Recognition

A quite distinct point of our object recognition system compared to an ordinary camera or image based one is that not only images from the scene camera but also fixation points in the images are given from the eye tracker. A typical object recognition system has to deal with complete images and has to perform image analysis to locate where the object of interest is. Hence, for example, when an image is highly cluttered, the recognition task becomes quite hard. Unlike such a system, we can take an advantage of fixation information which indicates the location of the object of interest.

Ideally, we would like to extract only the image of the interested object. Although the estimation of the boundary of an object is one of the active topics in computer vision [Pantofaru 2008], the technology is not mature enough to be utilized in a real-time situation. Therefore, we simply crop a rectangular region from the image centered on the fixation point and extract SIFT features from that region. The region is chosen to be large enough to contain sufficient interest points for reliable object recognition. Besides, performing SIFT feature extraction on regions near the fixation point also speeds up the feature extraction process.

Generally, the performance of object recognition relies on the number of the features extracted from the query [Kise et al. 2010]. Therefore, we select n features closest to the fixation point for use in object recognition. For example, when 50 is a given number for features, the 50 closest features to the centroid are used for object recognition as shown in Figure 3. Limiting the number of features not only enables the object recognition module to work "locally" on the object of interest, it also speeds up the recognition process for complex objects. Assuming that k features were originally extracted from the rectangular region around the fixation point, the number of features actually used for object recognition would be $\min(n, k)$.

In addition, by expanding this *non-weighting fixation guided recognition method* described above, we propose another method called *SIFT feature weighting fixation guided recognition method* that reasonably utilizes the geometrical configuration of features.

The eye position is considered as the point where the user is most interested at that moment. In other words, the attention of the viewer decreases as its distance from the gaze position increases. This insight gives us the idea to weight SIFT features according to the distance from the fixation point. Hence when building the histogram (see Section 2.2), more weight is given to the features close to the fixation point as compared to those far away. In this *SIFT weighting method*, we employ a Gaussian function to weight the vote in the histogram, i.e. the weight w of the feature that has an Euclidean distance d from the fixation point is given by

$$w(d) = \exp(-\frac{d^2}{a}),$$

where a is a given parameter.

Each weight is added to the corresponding identity in the histogram. Finally, the histogram is normalized to unit length as before.

2.3.2 Gaze-Based Ground Truth Generation

In order to apply any kind of benchmarking or evaluation to the system results, one needs to define the so called ground truth - a manually created result that represents the ideal system output. We need to model the time intervals, in which the user likes to get additional information (Augmented Reality, AR) about a specific object presented. The primitive manual tagging however, which is made on the basis of frames, needs to deal with noise which occurs through unconscious eye-movements and respective fixations. As the data that we manually tag with labels are the individual frames, the frames representing noise will also be labeled. In order to judge, whether a fixation to a specific object can be considered as noise or as within gaze, we need to define where the border of gaze and non-gaze fixations are.

To identify the event of a user gazing one specific object, we analyze the stream of fixations based on the following observations:

- When we gaze at an object, the *duration* is usually longer than that for any unintentional fixation or glance.
- Gaze position usually does not stay at a fixed point while we are gazing at an object. Instead, it moves around that object of interest. This may be considered as *noise*.

Hence, in this context we define **gaze** as a sequence of fixations on one specific object X. The number of frames on that object X must be longer than the *duration threshold* T_{dur}[5] but may also contain a certain amount of *noise*, which we consider as fixations on objects other than X. We explain how gaze is detected from manually labeled video frames in detail in the following.

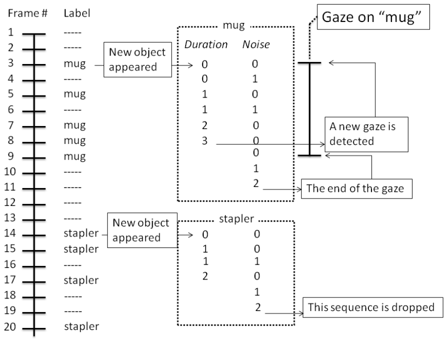


Figure 4: *An example of gaze detection with noise threshold $T_{noise} = 2$ and duration threshold $T_{dur} = 3$.*

Suppose we have a video stream and each frame is manually labeled as the identity of the object being indicated by the corresponding fixation. If there is no fixation for that frame nor an occurence of the object in the database, the frame is labeled as "undefined". From frame number zero, successively the labels of the frames are counted. When an inspected label is a defined object X, the algorithm starts to count the number (duration) of the frames F_X that have the label X. While counting up the F_X, if the number of consecutive frames that are *not* labeled as X (considered to be "noise") F_{noise} exceeds the noise threshold T_{noise}, the sequence is dropped (F_X is set to zero). As soon as the duration F_X exceeds the duration threshold T_{dur}, the sequence starting at the first frame with label X (where the recent counting started) is recognized as gaze on object X. This gaze ends at the last frame with the label X when the noise F_{noise} exceeds the noise threshold T_{noise}.

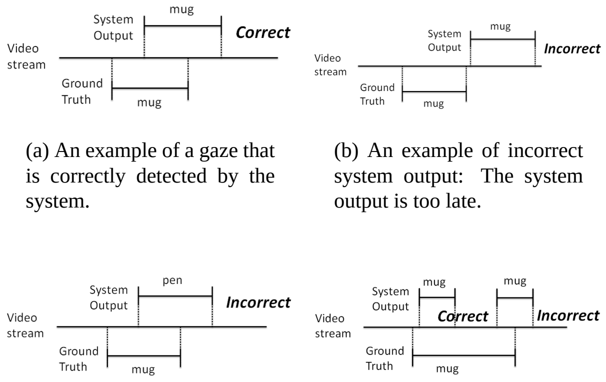
Figure 4 shows an example of gaze detection given a sequence of labeled video frames. In this example, we set noise threshold $T_{noise} = 2$ and duration threshold $T_{dur} = 3$. At frame number 3, the label *"mug"* appeared for the first time and we thus start counting up F_{mug}. Until frame number 8 (where the duration reaches T_{dur}), it does not contain any consecutive noise frames more than 2, thus it is recognized as gaze on the object *"mug"*. But for the next sequence of *stapler*-labels, the noise F_{noise} exceeds T_{noise} before the duration $F_{stapler}$ reaches T_{dur}. Consequently, the sequence of the frames is dropped, $F_{stapler}$ is set to zero.

We investigate video and eye tracking data with varying T_{noise} and T_{dur} thresholds to evaluate how much detected gaze by this algorithm matches expressed consciousness within our ground truth data of attention. These experiments and their evaluation allowed us to find suitable values of these thresholds.

2.3.3 Gaze Detection Methods Based on Recognition Results

As a criterion for evaluation, *gaze-based ground truth* is generated by the process stated in the previous section. Now we need to discuss how to detect the existence of gaze from results of fixation guided object recognition framework. Let us for now disregard the real-time requirements and assume, that our object recognition process is done for each frame that has a fixation. As a result every

[5] To find an optimal value of this threshold T_{dur}, we conducted experiments in which the user had to give explicit verbal feedback when he was looking at some object with consciousness. We evaluated, which threshold values yielded best result w.r.t. this spoken ground truth.



(a) An example of a gaze that is correctly detected by the system.

(b) An example of incorrect system output: The system output is too late.

(c) An example of incorrect system output: The name of object is incorrect.

(d) An example of incorrect system output: The gaze event is split up.

Figure 5: *Examples of correct and incorrect system output.*

such frame then contains a respective machine generated (recognized) label denoting the object in focus. In the following we need to verify whether the user's gaze is really focusing on that object or whether it can be considered as an unconscious glance or noise. Therefore, we propose one plain method and two different sophisticated methods to compute existence of gaze from a sequence of fixation guided recognition results (respective labels).

1. **Plain Method**: This method directly returns the results from fixation based object recognition.

2. **Accumulation of Last n Frames Method**: In this method, we directly accumulate the normalized histograms (Section 2.2) of best matches of SIFT features from each frame. The result is returned as the identity of the object that has the highest value in the accumulated histogram, but only if it exceeds a given threshold (otherwise as "undefined").

3. **Pseudo Ground Truth Generative Method**: In this method, the same process is applied to the recognition results that was used to post-process manually labeled ground truth data as described in the Section 2.3.2 (see Figure 4). The algorithm counts the number of frames that have the same label X. When the number F_X exceeds the *duration threshold* T_{dur} before noise F_{noise} exceeds *noise threshold* T_{noise}, gaze is detected for object X.

In Museum Guide 2.0, once the user's gaze is detected by one of these methods, the system starts to present AR. The presentation of AR is not stopped unless new gaze on another object X' (with $X' \neq X$) is detected, i.e. as long as these methods return either the name of the same object X or "undefined", the display of AR information remains active.

We evaluate the system by comparing the output from each of these methods with gaze-based ground truth that was generated from manual labels on each video stream. Generated ground truth represents a time interval, in which the user likes to get information about a specific object. Therefore, as shown in Figure 5(a), if there is a chronological overlap between the detected gaze and the ground truth, it is considered as a correct output. On the other hand, if there is no overlap, the name of object is not the same, or the event of gaze on the object is split up, as shown in Figure 5(b), 5(c) and 5(d) respectively, they are considered as incorrect outputs.

2.3.4 Compensation Approach for Real-time Processing

Our intended application is characterized by strong real-time requirements: The user wants to get AR presentations right at the time he is looking at an object. Ideally, the processing time of object recognition is required to be short enough so that the entire process catches up the real-time frame rate. However, the processing of a given query-image (fixated part of a frame) by the SIFT based retrieval system takes too long to process all frames (at 25 frames per second) that are delivered by the eye tracker. Consequently, not all fixation events can be processed and this system cannot detect gaze correctly.

To resolve this problem, we propose a *compensation approach for real-time processing*. In this approach, we prepare a standby image to catch up to the real-time environment and to minimize the loss of information. When a new fixation is detected, the corresponding image is stored as the current standby image. If the object recognition framework becomes idle after a recognition process, the image is piped to it to start over immediately. Simultaneously, the number of frames having the same fixation is counted and the recognition result is multiplied by it when the recognition process on that fixation has ended. This way, the recognition unit of our system will be kept as busy as possible and thus produces as many labels for fixated images as possible, while on the other hand the system always analyzes the newest fixation image. Thus, if gaze is recognized and AR is presented, it is based on the newest possible data.

3 Experiments and Results

To thoroughly evaluate different aspects of our real-time gaze-based object recognition framework, we conducted a series of experiments[6].

1. We proposed the *gaze-based ground truth generation* algorithm in Section 2.3.2. Thus, we conducted real-world experiments with different users to evaluate suitable threshold values for our generation algorithm (Subsection 3.1).

2. Using the suitable threshold values obtained in Experiment (1), we generated *gaze-based ground truths* which are aimed to be detected by our methods proposed in Section 2.3.3. We evaluated each of the *gaze detection methods based on recognition results* using the evaluation method stated in the section. All methods and parameters were optimized for Museum Guide 2.0 based on this evaluation (Subsection 3.2).

3. We evaluated the performance of the system in a real-time environment using the *compensation approach for real-time processing* introduced in Section 2.3.4 (Subsection 3.3).

4. Finally, we conducted a user study to test the overall performance and usability of the presented real-time gaze-based object recognition framework (Subsection 3.4).

In Experiment (1) to (2), we ignore the constraints of a real-time environment, i.e. there is sufficient time to process for *each frame* that we call *off-line analysis*. The parameters and methods in Experiment (3) and (4), which are processed in real-time, are optimized based on the results from the *off-line analysis*.

Before conducting the experiments, we designed our museum for the entire experiments. As stated in Section 1, we focused on 3D and small-sized objects. The objects we used are shown in Figure 6. These objects were placed well spaced-out on a long table

[6] In addition to the experiments described here, we would also like to refer [Toyama 2011] which presents how the object recognition performance is improved by fixation (POR) guided object recognition method.

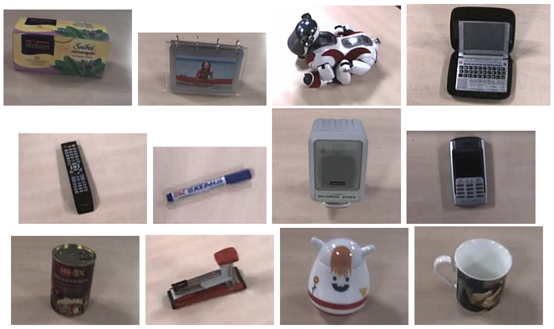


Figure 6: *Example images of objects in our museum: The first row (a tea packet, a photo frame, a robot pet and an electronic dictionary), the second row (a remote control, a whiteboard marker, a speaker and a cell phone) and the third row (a tin, a stapler, a pot and a mug).*

and all recordings and experiments were done under the same light setting. The objects are relatively less complex and less fascinating compared to real exhibits in a museum, therefore, the tasks are considered to be easier in a real scenario since a viewer in a museum tends to look longer and objects have more features.

For building the database, we captured 438 images in total. As stated in the previous section, we used more images from the objects that had less SIFT features within one image. For example, we captured 21 images of an *electronic distionary* which approximately has 200 features in one image, whereas 55 images were captured from a *tin* which approximately has 40 features in one image.

3.1 Validation of Gaze-Based Ground Truth Generation

In this experiment we aimed to find suitable threshold values for our gaze-based ground truth generation algorithm by analyzing video and gaze data.

In this analysis, five objects (*a tin, a pen, a cell phone, a PC speaker* and *a tea packet*) were placed on a table and we asked the test persons to give spoken feedback (e.g. *"Now, I am looking at a pen."*) when they were looking at objects *consciously*. This explicit verbal feedback represents the ideals of generated gaze-based ground truths. Six test persons took part in this experiment and they were asked to look at objects at least 20 times in total. We also asked the test persons to act as if they were browsing around a real museum, where some objects are only looked at for short time but also not consciously. We evaluated with which threshold values the algorithm generates the best overlapping result with respect to the spoken ground truth.

We applied our gaze-based ground truth generation algorithm with changing duration threshold T_{dur} and noise threshold T_{noise} to the recorded and manually labeled data. If the number of generated ground truths by a particular combination of threshold values is close to the number of verbal feedback, these ground truths are considered to be correctly reflecting our intentional behavior. Thus, we compared the number of ground truths generated by the algorithm to the amount of the verbal feedback from the test people.

We would like to identify a general tendency rather than the variation between individuals. Therefore, we average the number of the generated ground truths for each test person. Figure 7 shows the average number of generated ground truths by changing T_{dur} for $T_{noise} = 3$, $T_{noise} = 18$, and $T_{noise} = 30$, respectively. The average amount of verbal feedback for each test person is also shown in the figure as the horizontal dotted line, with a value of 24.5.

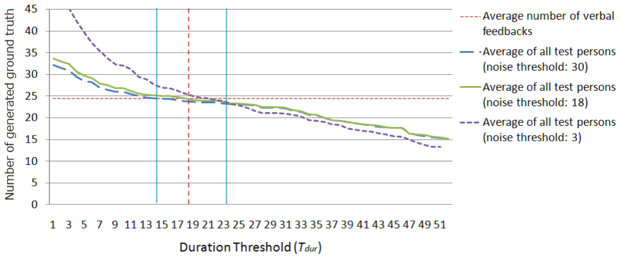


Figure 7: *Gaze-based ground truth generated by the algorithm with changing T_{dur}. The dotted vertical line is drawn at duration threshold 18. In the area between the two vertical solid lines (on 14 and 23, respectively), the graphs (noise threshold: 18 and 30) reach almost flat lines.*

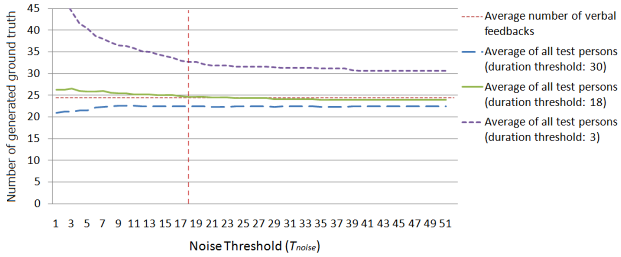


Figure 8: *Gaze-based ground truth generated by the algorithm with changing T_{noise}. The dotted vertical line is drawn at duration threshold 18. All graphs converge as T_{noise} increases.*

While T_{noise} is low (noise threshold: 3), the slope on each point in the graph is steep. Then, as T_{noise} becomes larger (noise threshold: 18 or 30), the graph reaches an almost flat line between respective thresholds around T_{dur} 14 and 23. Since the number of generated ground truths on the flat area is close to the average number of verbal feedbacks, the T_{dur}s in that range are considered as candidates for the optimal T_{dur} with respect to the ground truth.

Figure 8 shows the average number of generated ground truths with changing T_{noise} for $T_{dur} = 3$, $T_{dur} = 18$ and $T_{dur} = 30$, respectively. All graphs converge as T_{noise} increases. From these graphs, we can infer that 18 is a reasonable value for T_{noise} as the number of generated ground truths remain constant from this point on.

From these observations, we can conclude that the algorithm reliably generates quite similar results to the spoken ground truth with the proper setting of threshold values. We select 18 as the optimal threshold values for both noise and duration in a general case because this combination reflected verbal feedback well within this experimental framework.

3.2 Evaluation of Methods for Detection of Gaze

In the previous subsection, we confirmed that the gaze-based ground truth generated by our algorithm reasonably reflects the verbally expressed consciousness. By using the ground truths generated by this algorithm, we evaluated the methods for detection of gaze(*plain method*, *accumulation of last n frames method* and *pseudo ground truth generative method*) using the fixation guided object recognition method.

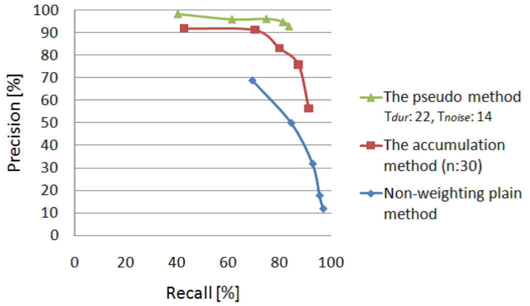


Figure 9: *The best results of each gaze detection method. The pseudo method outperformed the other two methods.*

This experiment is processed off-line so the processing time is not considered critical here. The test video files and gaze data were recorded from six test persons while they were strolling around our museum and looked at objects according to their interests. Ten video files were recorded from them and each frame was labeled manually as the name of the object being indicated by the fixation point. By applying the generating algorithm, 183 ground truths were generated from them in total. We compared the generated ground truths and detected gaze on particular objects by the system as described in Section 2.3.3. To evaluate the methods, we use recall R and precision P. Since each gaze has a label (the identity of the object being looked at), evaluation is done on a per class basis and then averaged over all classes.

We compare all the detection methods in Figure 9. In this figure, the best results from a number of combinations of each parameter are shown for each method. We can observe that the two sophisticated methods outperformed the plain method. The accumulation method worked well compared to the simple method, however it was inferior to the pseudo method. The reason for this was that this method was highly depending on the features from each frame. For example, even if only one frame captured object X in a sequence of frames and the other frames did not capture any objects, the features from X affected the entire recognition process in this method. In this case, X is returned even though this is not considered as gaze.

Based on these experiments, we selected *pseudo ground truth generative method* and *SIFT weighting method* as our gaze-based object recognition system for Museum Guide 2.0. And the parameters were set to $T_{dur} = 22$ and $T_{noise} = 14$ (for the recognition method).

3.3 Evaluation of the Approach for Real-time Processing

We evaluate our *compensation approach for real-time processing* introduced in Section 2.3.4. In this experiment, we used the same video and gaze data as in the previous experiments but sending 25 frames per second (the same frame rate as the iViewX$^{\mathrm{TM}}$) to the gaze-based object recognition system optimized in the last subsection to simulate a real-time environment.

Figure 10 shows the results obtained by the method with and without the compensation approach and the result from the off-line experiment (obtained in the previous subsection). The threshold values (T_{dur} and T_{noise}) for the gaze detection method with compensation approach were the same as in the off-line experiment. However, we needed to use different values for the no compensation approach. Due to its long processing time for object recognition, the method optimized in the last section could not detect any gaze in the no compensation approach. Generally, the optimized T_{dur} or T_{noise} for the off-line system are too large as the recognition system cannot catch up the real-time speed. Therefore, we dropped $T_{dur} = 6$ and $T_{noise} = 4$ for the no compensation approach to obtain a similar precision score as with the other approach. We can observe an enormous drop of recall as compared to the results in the off-line experiments. However, compared to the no compensation approach, the method with compensation approach worked significantly better.

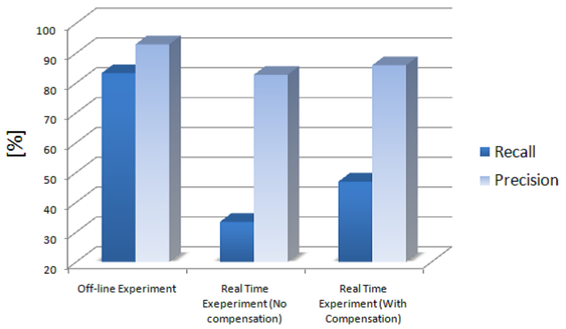


Figure 10: *Results of real-time simulation. Although recall drops significantly, precision still remains at an acceptable level.*

We adopted the compensation approach for real-time processing and it was used in the following user study.

3.4 User Study

To evaluate the usability of the complete system, we conducted a user study with 23 users. The users were asked to stroll in our museum with two different guide system. One is our Museum Guide 2.0 and the other is an audio player based guide system. Audio player based museum guides are currently used in most of the museums and therefore provide a good basis of comparison with existing technology. Usually exhibits have a tag number in front of them and the users have to select the corresponding audio track from the audio guide to get more information about that exhibit. The same setup was used in our experiment by assigning a tag to each of the twelve objects in our museum and storing the corresponding audio information with the same tag in the audio player. The users were asked to freely move in the museum and get information about the object they are interested in with the help of the audio player.

After the users finished their round with the audio guide, they were introduced to the eye tracker and the eye tracker was calibrated for each user using the five point calibration algorithm mentioned in Section 2.1. Then, the users were asked to go around the museum again wearing the eye tracker. Whenever the users gazed at an exhibit and gaze on exhibits was detected, Museum Guide 2.0 played a pre-recorded audio file to provide the same information as the audio player about the gazed upon exhibit.

When the users finished their round with Museum Guide 2.0, they were given a questionnaire to assess different aspects of the system. A summary of user responses to the questions comparing the gaze-based interface with the traditional audio player interface is shown in Figures 11. Since the eye tracker used in the study has several hardware constraints (such as uncomfortable helmet, chin rest, etc.), we referred only to a “gaze-based interface (device)” in the questionnaire to judge the real potential of gaze-based information provision. The results show that most of the users would prefer to use a gaze-based device as compared to an audio player when they go to a museum. Another interesting result was that al-

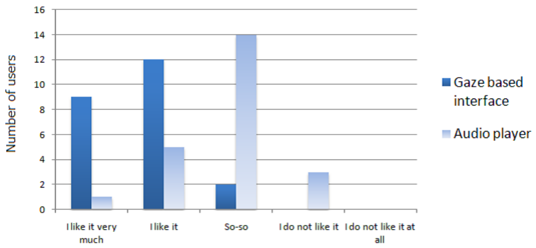


Figure 11: *Responses in the user study for the question: How much do you like a gaze-based interface (or a traditional audio player) for getting information?*

though many users were satisfied with the traditional audio player, the mean opinion score (MOS) for Museum Guide 2.0 was 4.3 as compared to 3.2 for an audio player.

The goal of this user study was to evaluate the effectiveness of gaze based interface for information provision. We are planning to conduct a study in a real museum to more thoroughly evaluate the Museum Guide application.

4 Conclusion

This paper introduced a new interface for AR application that effectively utilizes human gaze and technologies of object recognition. First, we showed that the object recognition framework can successfully be guided by using fixation points. Then, by detecting users' gaze on particular objects, the system could reasonably trigger the presentation of AR. By testing Museum Guide 2.0 in a real-world environment with our proposed compensation approach for real-time processing, we demonstrated the feasibility of our approach. The results from user study showed that the usability of this interface is superior to the traditional audio guide. Future work would focus on developing this application not only for a museum but in a wider environment. We are confident that this application has great potential and would contribute to the development of this technology.

Acknowledgements

This work was partially funded by the BMBF (German Federal Ministry of Education and Research), project Perspecting (01 IW 08002). We also wish to thank SensoMotoric Instruments (SMI) GmbH for providing us with an iView X$^{\mathrm{TM}}$ HED mobile eye tracker for building the application and conducting all the experiments.

References

Bee, N., Wagner, J., Andre, E., Charles, F., Pizzi, D., and Cavazza, M. 2010. Interacting with a gaze-aware virtual character. In *Proceedings of the International workshop on eye gaze in intelligent human machine interaction*.

Biedert, R., Buscher, G., Schwarz, S., Hees, J., and Dengel, A. 2010. Text 2.0. In *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*, 4003–4008.

Bonino, D., Castellina, E., Corno, F., Gale, A., Garbo, A., Purdy, K., and Shi, F. 2009. A blueprint for integrated eye-controlled environments. *Universal Access in the Information Society 8*, 4, 311–321.

Buswell, G. T. 1935. *How people look at pictures*. The University of Chicago press, Chicago.

Haralick, R. M., Shanmugam, K., and Dinstein, I. 1973. Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics 3*, 6, 610–621.

Henderson, J. M. 2003. Human gaze control during real-world scene perception. *Trends in Cognitive Sciences 7*, 11, 498–504.

Indyk, P., and Motwani, R. 1998. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing*, 604–613.

Ishiguro, Y., Mujibiya, A., Miyaki, T., and Rekimoto, J. 2010. Aided eyes: eye activity sensing for daily life. In *The 1st Augmented Human International Conference (AH2010)*, ACM, H. Saito, J.-M. Seigneur, G. Moreau, and P. Mistry, Eds., 25.

Jacob, R. J. K. 1993. Eye movement-based human-computer interaction techniques: Toward non-command interfaces. In *Advances in Human-Computer Interaction*, Ablex Publishing Co, vol. 4, 151–190.

Kise, K., Chikano, M., Iwata, K., Iwamura, M., Uchida, S., and Omachi, S. 2010. Expansion of queries and databases for improving the retrieval accuracy of document portions - an application to a camera-pen system. 309–316.

Kovic, V., Plunkett, K., and Westermann, G. 2009. Eye-tracking study of animate objects. *Psihologija 42*, 3, 307–327.

Li, J., and Allinson, N. M. 2008. A comprehensive review of current local features for computer vision. *Neurocomputing 71*, 10-12, 1771–1787.

Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60*, 2, 91–110.

Majaranta, P., Ahola, U.-K., and Špakov, O. 2009. Fast gaze typing with an adjustable dwell time. In *Proceedings of the 27th international conference on Human factors in computing systems*, ACM, Boston, MA, USA, CHI '09, 357–360.

Pantofaru, C. 2008. *Studies in Using Image Segmentation to Improve Object Recognition*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

Ponce, J., Hebert, M., Schmid, C., and Zisserman, A. 2007. *Toward Category-Level Object Recognition*. Springer-Verlag New York, Inc.

Roth, P. 2008. Survey of Appearance-based Methods for Object Recognition. Tech. Rep. ICG-TR-01/08, Computer Graphics & Vision, TU Graz.

Toyama, T. 2011. Object recognition system guided by gaze of the user with a wearable eye tracker. In *Proceedings of the 33rd international conference on Pattern recognition*, Springer-Verlag, Berlin, Heidelberg, DAGM'11, 444–449.

Yarbus, A. L. 1967. Eye movements and vision. *Plenum Press*.

Zhang, J., Lazebnik, S., and Schmid, C. 2007. Local features and kernels for classification of texture and object categories: a comprehensive study. *International Journal of Computer Vision 73*, 213–238.