# Attentional Object Spotting by Integrating Multimodal Input

Chen Yu, Dana H. Ballard, and Shenghuo Zhu
Department of Computer Science
University of Rochester
Rochester, NY 14627,USA
{yu,dana,zsh}@cs.rochester.edu

## Abstract

*An intelligent human-computer interface is expected to allow computers to work with users in a cooperative manner. To achieve this goal, computers need to be aware of user attention and provide assistances without explicit user request. Cognitive studies of eye movements suggest that in accomplishing well-learned tasks, the performer's focus of attention is locked with the ongoing work and more than 90% of eye movements are closely related to the objects being manipulated in the tasks. In light of this, we have developed an attentional object spotting system that integrates multimodal data consisting of eye positions, head positions and video from the "first-person" perspective. To detect the user's focus of attention, we modeled eye gaze and head movements using a hidden Markov model(HMM) representation. For each attentional point in time, the object of user interest is automatically extracted and recognized. We report the results of experiments on finding attentional objects in the natural task of "making a peanut-butter sandwich".*

## 1. Introduction

The development of multimodal human-computer interface that allows us to communicate with computers just like with other humans has recently attracted more and more interest. The aim is to increase both the communication channels between human and machine(e.g., by talking, seeing and gesturing), and at the same time the quality of interaction by being aware of the focus of user attention. For example, by monitoring users' behaviors and finding their attentions, computers can predict their needs, and provide corresponding information and services before users explicitly express their requests. In this way, computers can be seamlessly integrated into our everyday lives and work with humans as attentive human-like assistants.

We believe that the visual focus of attention plays a vital role in various multimodal human-computer interaction applications. In the scenario of making a peanut-butter sandwich, for instance, when a computer notices that the user's attentional object is a peanut butter jar, it can provide information related to peanut butter by speech, such as a set of recipes or nutritional values. Furthermore, with an actuator such as a robotic arm, the machine can grasp and deliver the peanut butter jar to the user. In this way, humans and computers can work together in a cooperative and interactive manner.

In this paper, we present an attentional object spotting system that can detect objects of user interest in real time while the user wears a head-mounted eye tracker and performs the task of making a sandwich. Eye gaze is monitored to find user's focus of attention. Input of the system consists of eye positions, head positions in concert with video captured from a head-mounted camera. We employ hidden Markov models(HMMs) to detect visual attention based on eye and head movements. Then the image at an attentional point in time is analyzed to extract the object from the visual scene. Output of the system is a sequence of object names that represent dynamic properties of user attention during the task.

The remainder of this paper is organized as follows. Section 2 describes related work in using eye gaze for human-computer interaction. Section 3 first gives an overview of cognitive studies of eye movement involved in everyday activities. We then present our approach to using eye gaze and its advantages. Section 4 describes an attentional object spotting system we implemented. In Section 5, we present the experimental setup and the results. Finally, Section 6 concludes with a discussion of our future work.

## 2. Related Work

Many techniques have been developed for tracking eye movements during the past several decades. However, most applications of eye tracking have been in psychological research for probing into subjects' perceptual or cognitive processes. This state of affairs has changed recently. Based

on a series of experiments that compare an eye gaze interaction technique for object selection with the mouse selection method, Jacob and Sibert[15, 5] argued that eye gaze can be used as the main device for controlling conventional interface components. More recently, Stiefelhagen et al.[16] have developed a system to estimate visual focus of attention of participants in a meeting from gaze and sound cues. Hyrskykari[4] has designed and is currently implementing an eye-aware application, called iDict, which is a general-purpose translation aid system. iDict monitors the user's gaze path while the user is reading texts written in a foreign language. When the reader encounters difficulties, iDict steps in and provides assistance with the translation.

## 3. The Computational Role of Eye Movement

To develop an intelligent multimodal human-computer interface, we believe that it is helpful to make use of the discoveries in cognitive studies to guide the design of our approach. This section first describes the studies of eye movements in experimental psychology. We then propose our approach to integrating eye gaze with visual information to find users' focus of attention.

Human beings continuously explore their environment by moving eyes. They look around quickly and with little conscious effort. The classic work of Yarbus[18] showed that, even when viewing the same image, subjects make different eye movements depending on the task that they are trying to solve. Subjects primarily foveate the small percentage of objects in the scene that are relevant to solving the task. Recently, two studies have addressed the nature of the involvement of vision in coordinating actions in natural tasks, such as food preparation and housework. Land et al.[7, 8] have studied the fixation patterns of humans performing the well-learned task of making tea. This work demonstrates that a subject performing a task tends to fixate on the object that is currently relevant to the ongoing task. In the other study, Hayhoe et al.[3, 8] showed similar results from the experiments in which students made peanut butter and jelly sandwiches. In both cases, the principal conclusion is that almost every action in the task sequence is guided and checked by vision, with eye gaze usually focusing on the objects being manipulated in motor actions.

Figure 1 shows a sample scene of the experiments in which a user is making a peanut-butter and jelly sandwich. We confirmed the conclusions of the previous studies by noticing that gaze rarely strays from the objects of user interest though there might be multiple eye fixations on the different parts of the object. In light of this, our hypothesis is that eye and head movements, as an integral part of the motor program of users, provide important information for building an intelligent human-computer interaction. We test this hypothesis by developing a method that can spot objects of user interest based on eye gaze and head movement.

**Figure 1.** The sample view from the first-person perspective.

The advantages of our approach are threefold. Firstly, compared with other modalities, such as gestures and voice, eye gaze has a unique property: it implicitly carries information on the focus of the user's attention at a specific point in time. Thus, we can utilize eye gaze to find objects users are interested in. Compared with our approach, the system based on a wearable camera has the same ability to "see" as the user sees from the "first-person" perspective, but it is not trivial to find the object of user interest from multiple objects in a visual scene. In our system, however, we can directly utilize eye position as a cue to segment the attentional object from the background. Secondly, compared with traditional video processing of fixed camera observations, the dynamic properties of an agent-centered view captured by a head-mounted camera provide image sequences that are more informative because the agent uses the same data for visual processing. Thirdly, the objects of an agent's interest in time can help understanding human behaviors, which is another important problem for human-computer interaction. Objects are closely related with some specific actions. For instance, a knife is related to the action of cutting and a stapler is related to the action of stapling. These object contexts would suggest specific actions for action recognition. For example, if the agent is looking at the stapler, it is mostly likely that the next action is related to "stapling" and it is unlikely that the action will be "cutting".

## 4. An Attentional Object Spotting System

As we have pointed out earlier, our goal is to build a system aware of users' focus of attention. We argue that objects focused by users are important indicators of their interests. This section presents an attentional object spotting system that first finds the eye fixations and then spots objects in the fixation durations.

### 4.1. Eye Movement Analysis

In the context of our application of eye gaze, the primary objective of eye data analysis is to determine where and

when the user looks at the objects in the visual scene. Although there are several different modes of eye movement, the two most important modes for directing cognitive works are saccades and fixation. Saccades are rapid eye movements that allow the fovea to view a different portion of the visual scene. Often a saccade is followed by one or more fixations when objects in a scene are viewed. Our goal is to find the fixations from continuous data stream of eye movement. The existing fixation finding methods[13] can be categorized into three groups: velocity-based, dispersion-based and region-based. Velocity-based methods find fixations according to the velocities between consecutive data points. Dispersion-based methods identify fixation points as the points that are grouped closely together with the assumption that the fixation points generally occur near one another. Region-based methods identify fixation points as points that fall within a fixed region called areas of interest(AOIs).

We developed a velocity-based method to model eye movements using a hidden Markov model(HMM) representation that has been widely used in speech recognition with great success[11]. A hidden Markov model consists of a set of $N$ states $S = \{s_1, s_2, s_3, ..., s_N\}$, the transition probability matrix $A = a_{ij}$, where $a_{ij}$ is the transition probability of taking the transition from state $i$ to state $j$, prior probabilities for the initial state $\pi_i$, and output probabilities of each state $b_i(O(t)) = P\{o(t)|s(t) = s_i\}$. Salvucci et al.[12] first proposed a HMM-based fixation identification method that uses probabilistic analysis to determine the most likely identifications for a given protocol. Our approach is different from his in two ways. First, we use training data to estimate the transition probabilities instead of setting predetermined values. Secondly, we notice that head movements provide valuable cues to model focus of attention. This is because when users look towards an object, they always orient their heads towards the object of interest so as to make it in the center of their visual fields. As a result of the above analysis, head positions are integrated with eye positions as the observations of HMMs.

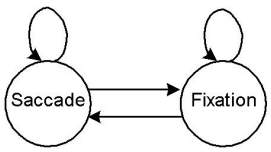


**Figure 2.** The HMM of eye movement

Figure 2 shows a 2-state HMM that is used in our system for eye fixation finding. One state corresponds to saccade and the other represents fixation. The observations of HMM are 2-dimensional vectors consisting of the magnitudes of the velocities of head rotations in three dimensions and the magnitudes of velocities of eye movements. We model the probability densities of the observations using a two-dimensional Gaussian:

$$b_j(O_t) = \frac{1}{\sqrt{(2\pi)^2 \mid \sigma_j \mid}} e^{\frac{1}{2}(O_t - \mu_j)^T \sigma_j^{-1} (O_t - \mu_j)} \quad (1)$$

The parameters of HMMs needed to be estimated comprise the observation and transition probabilities. Specifically, we need to compute the means($\mu_j$) and variances($\sigma_j$) of two-dimensional Gaussian(four parameters) for each state and the transition probabilities(2 parameters) between two states. Thus, a total of 10 parameters need to be estimated in the HMM. The estimation problem concerns how to adjust the model $\lambda$ to maximize $P(O \mid \lambda)$ given an observation sequence $O$. We can initialize the model with flat probabilities, then the forward-backward algorithm[11] allows us to evaluate this probability. Using the actual evidence from the training data, a new estimate for the respective output probability can be assigned:

$$\bar{\mu}_j = \frac{\sum_{t=1}^{T} \gamma_t(j) O_t}{\sum_{t=1}^{T} \gamma_t(j)} \quad (2)$$

and

$$\bar{\sigma}_j = \frac{\sum_{t=1}^{T} \gamma_t(j)(O_t - \bar{\mu}_j)(O_t - \bar{\mu}_j)^T}{\sum_{t=1}^{T} \gamma_t(j)} \quad (3)$$

where $\gamma_t(j)$ is defined as the posterior probability of being in state $j$ at time $t$ given the observation sequence and the model.

As a result of learning, the saccade state contains an observation distribution centered around high velocities and the fixation state represents the data whose distribution is centered around low velocities. The transition probabilities for each state represent the likelihood of remaining in that state or making a transition to another state. An example of the results of eye data analysis is shown in Figure 3.

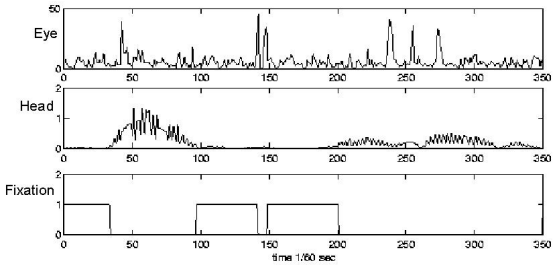


**Figure 3. Eye Fixation finding. The top plot:** Point-to-point velocities of eye positions. **The middle plot:** The velocity profile of head. **The bottom plot:** A temporal state sequence of HMM("1" indicates the fixation state and "0" represents the saccade state).

## 4.2. Object Spotting

This section describes the method of automatic object spotting by integrating visual information with eye gaze

data. For an eye fixation, the object of user interest is extracted from the snapshot of the scene. Figure 4 shows the overview of our approach composed of three steps: image segmentation, object representation and object recognition.

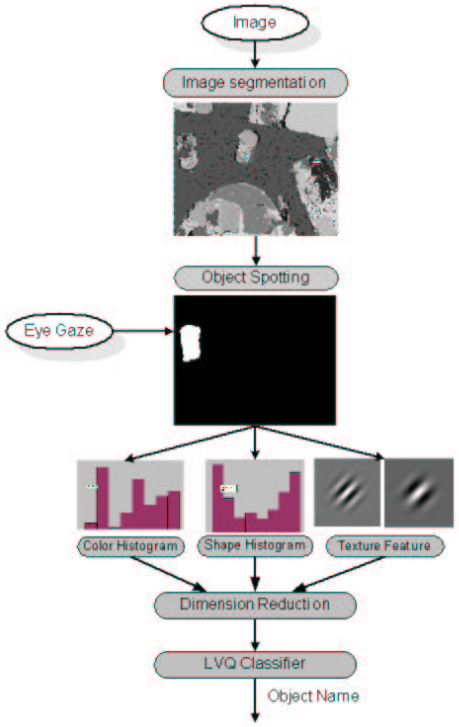


**Figure 4.** The overview of Object Spotting

#### 4.2.1 Image Segmentation

The image segmentation in our system consists of two steps. First, we apply seeded region growing(SRG) algorithms[1] to segment objects from the background. SRG is based on the conventional region growing postulate of similarity of pixels with regions, but whose mechanism is closer to that of the watershed. The method starts from an initial, incomplete segmentation and tries to aggregate the unlabeled pixels to one of the given regions. The decision whether a pixel should join a region or not is based on the fitness function that reflects the similarity between the region and the candidate pixel. The order in which the pixels are processed is determined by a global priority queue which sorts all candidate pixels by their fitness values.

Second, eye gaze is utilized as a cue to extract the object of user interest from all the objects detected. This is implemented by coordinating the eye gaze position into $(x, y)$ position in the image and choosing the region that contains this position. Figure 5 shows a scene snapshot and the segmentation result when the user is grasping a peanut butter jar.

**Figure 5. Left:** The snapshot image with eye position(black cross). **Right:** The object extracted from the left image.

#### 4.2.2 Object Representation

The extracted object is represented by a model that contains color, shape and texture features. Based on the works of [9, 14, 17], we constructed the visual features of objects that are large in number, invariant to different viewpoint, and are driven by multiple visual cues. Specifically, 64-dimensional color features are extracted by color indexing method[17], and 48-dimensional shape features are represented by calculating histograms of local shape properties[14]. The Gabor filters with three scales and five orientations are applied to the segmented image. It is assumed that the local texture regions are spatially homogeneous, and the mean and the standard deviation of the magnitude of the transform coefficients are used to represent the object in a 48-dimensional texture feature vector. The feature representations consisting of a total of 160 dimensions are formed by combining color, shape and texture features, which provide fundamental advantages for fast, inexpensive recognition.

Most classification algorithms, however, do not work efficiently in higher dimensional spaces because of the inherent sparsity of the data. This problem has been traditionally referred to as the dimensionality curse. In our system, we deduced the 160-dimensional feature vectors into the vectors with the dimensionality of 30 by principle component analysis(PCA)[2], which represents the data in a lower dimensional subspace by pruning away those dimensions that result in the least loss of information.

#### 4.2.3 Object Recognition

We employ an appearance-based object recognition method that makes our system more general and more easily trainable from visual data. The system essentially operates by comparing a feature representation of object appearance against many prototype representations stored in the memory to find the closest match. Three-dimensional objects are represented by using a view-based approach in which multiple two-dimensional images of an object are captured from multiple perspectives and grouped to collectively form a model of the object. In the training phase, the feature vec-

tors are used to train the classifier whose rule is to divide the feature space into regions that correspond to different objects. Kohonen's Learning Vector Quantilization(LVQ) algorithm [6] has been applied that allows us to build a classifier from labeled data samples. Instead of modeling the class densities, LVQ models the discrimination function defined by the set of labeled codebook vectors and the nearest neighborhood search between codebooks and data. The training algorithm involves an iterative gradient update of the winner codebook. The direction of the gradient update depends on the correctness of the classification using a nearest-neighborhood rule in Euclidean space. If a data sample is correctly classified(the labels of the winner unit and the data sample are the same), the codebook closest to the data sample is attracted toward the sample; if incorrectly classified, the data sample has a repulsive effect on the codebook. The update equation for the winner unit $m^c$ defined by the nearest-neighbor rule and a data sample $x(t)$ is

$$m^c(t+1) = m^c(t) \pm \alpha(t)[x(t) - m^c(t)] \qquad (4)$$

where the sign depends on whether the data sample is correctly classified(+) or misclassified(-). The learning rate $\alpha(t) \in [0, 1]$ must decrease monotonically in time and the training procedure is repeated iteratively until convergence. In the recognition phase, a data point $x_i$ is assigned to a class according to the class labels of the k-closest codebooks.

## 5. Experiment

The training data of the appearance-based object models are collected beforehand and utilized to determine the codebooks of LVQ. Those data are also applied to calculate the transformation matrix for PCA. In our experiments, the user wore an eye tracker mounted on the head, and was seated at a table with the items required for the task. No instructions were given except to make a peanut butter and jelly sandwich.

Monocular(left) eye position was monitored with an Applied Science Laboratories(ASL) Model 502 eye tracker(shown in Figure 6), which is a head-mounted, video-based, IR reflection eye tracker[10]. The eye position signals were sampled at 60Hz and had a real time delay of 50 msec. The accuracy of the eye-in-head signal is approximately $1^\circ$ over a central $40^\circ$ field. Both pupil and first Purkinje image centroids are recorded, and horizontal and vertical eye-in-head position is calculated based on the vector difference between the two centroids shown in Figure 6. A Polhemus 3D tracker was utilized to acquire 6-DOF head positions at $40Hz$. The headband of the ASL holds a miniature "scene-camera" to the left of the user's head that provides the video of the scene from the "first person" perspective. The video signals were sampled at the resolution of 320 columns by 240 rows of pixels at the frequency of 15Hz.

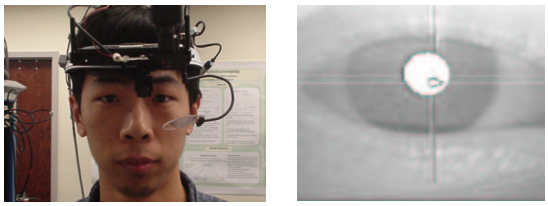

**Figure 6. Left:** A user wore a eye tracker. **Right:** The eye image with cross-hairs indicating pupil center and corneal reflection.

We collected video, eye positions and head positions from 18 users, each of whom performed the task three times. Over the 2-minute period that it took to make a sandwich, eye gaze was almost exclusively directed on the objects involved in the task, as noted in the works of Hayhoe and Land. A description of a segment of the object sequence through the task is shown in Figure 7. The user is performing the subtask of spreading peanut butter on the bread. He fixates the bread for about 300 msec to guide placement of the bread on the plate. Gaze is then transferred to the knife that guides the left hand to grasp it and move it toward a peanut butter(PB) jar. While this is in progress, gaze is transferred to the peanut butter jar in order to guide the subsequent movement of the left hand to scoop peanut butter. While the user spreads the bread with peanut butter, eye fixations alternate between the bread and the PB jar, and so on. From this example, it can be seen that eye movements are very tightly locked with the manipulations of objects. Specifically, from our analysis, fewer than $5\%$ of eye fixations were to 'irrelevant' to the on-going task. Examples are the setting down of objects without the guidance of vision, and look-around eye movements to capture the layout of visual scene before the beginning of actions.

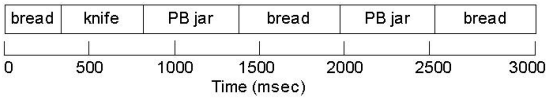


**Figure 7.** Sequence of attentional objects during the subtask of spreading the bread with peanut butter.

To evaluate performance of our system, the video records in concert with eye gaze data were analyzed and manually labeled. We then compare the results of human analysis with the results of our eye fixation finding algorithm, which is shown in Table 1. The errors in finding eye fixations are mainly caused by track loss. Although we remove the data that are out of normal range before sending them to HMM, this operation causes the discontinuity of the temporal sequence of eye positions, which leads to incorrect state tran-

sitions of HMM. The results of object spotting and recognition are also shown in Table 1 in the form of percentage compared with human analysis. A recognition accuracy of $86.5\%$ demonstrates the effectiveness of our approach. An analysis of the errors reveals that object occlusion caused by user hands lead to incorrectness in image segmentation.

**Table 1.** Results of object spotting

| Object | Eye Fixation Finding | Object Recognition |
|---|---|---|
| Overall | 93.6% | 86.5% |
| Bread | 95.2% | 91.6% |
| Jelly jar | 90.7% | 82.3% |
| Knife | 83.2% | 75.9% |
| PB jar | 89.3% | 81.5% |

## 6. Conclusion and Future Work

We have described an attentional object spotting system that finds and recognizes the objects of user interest. The approach is unique in that it analyzes both eye gaze and head position to detect the user's focus of attention. We demonstrated our approach in the domain of finding attentional objects when a user performs the task of making a sandwich.

The main goal of this work is to not only explore the use of eye gaze to detect the user's focus of attention, but also ultimately build an attention-based multimodal human-computer interface. In order to achieve the overall goal, we have developed an action recognition system to understand human actions in natural tasks[19], which addresses another fundamental problem of understanding human activities. We will integrate these two systems to obtain better understandings of user attention and behaviors in natural tasks. Based on explicitly monitoring user attention and task progress, an intelligent human-computer interface will be developed to predict next actions, suggest possible alternative actions and provide assistances for users working in natural environments.

### Acknowledgments

The authors wish to express their thanks to Mary Hayhoe for fruitful discussions. Brian Sullivan was a great help in building the experimental system.

## References

[1] R. Adams and L. Bischof. Seeded region growing. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 16(6), June 1994.

[2] C. C. Aggarwal and P. S. Yu. Finding generalized projected clusters in high dimensinal spaces. In *SIGMOD*, 2000.

[3] M. Hayhoe. Vision visual routines: A functional account of vision. *Visual Cognition*, 7:43–64, 2000.

[4] A. Hyrskykari, P. majaranta, A. Aaltonen, and K.-J. Raiha. Design issues of idict: A gaze-assisted translation aid. In *Proceedings eye tracking research and applications symposium*, 2000.

[5] R. J. Jacob. Eye tracking in advanced interface design. In W. Barœld and T. Furness, editors, *Advanced Interface Design and Virtual Environments*, pages 258–288, Oxford, UK, 1995. Oxford University Press.

[6] T. Kohonen. Improved versions of learning vector quantization. In *IJCNN*, volume 1, pages I545–I550, 1990.

[7] M. Land, N. Mennie, and J. Rusted. The roles of vision and eye movements in the control of activities of daily livng. *Perception*, 28:1311–1328, 1999.

[8] M. F. Land and M. Hayhoe. In what ways do eye movements contribute to everyday activities? *Vision Research*, 41:3559–3565, 2001.

[9] B. W. Mel. Seemore: Combining color, shape, and texture histogramming in a neurally inspired approach to visual object recognition. *Neural Computation*, 9:777–804, 1997.

[10] J. B. Pelz, M. M. Hayhoe, D. H. Ballard, A. Shrivastava, J. D. Bayliss, and M. von der Heyde. Development of a virtual laboratory for the study of complex human behavior. In *Proceedings of the SPIE*, San Jose, CA, 1999.

[11] L. R. Rabiner and B. Juang. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[12] D. D. Salvucci and J. Anderson. Tracing eye movement protocols with cognitive process models. In *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society*, pages 923–928, 1998.

[13] D. D. Salvucci and J. H. Goldberg. Indentifying fixations and saccades in eye-trakcing protocols. In *Proceedings of eye tracking research and applications symposium*, FL, November 2000.

[14] B. Schiele and J. L. Crowley. Recognition without correspondence using multidimensional receptive field histograms. *International Journal of Computer Vision*, 36(1):31–50, 2000.

[15] L. E. Sibert and R. J. K. Jacob. Evaluation of eye gaze interaction. In *CHI*, pages 281–288, 2000.

[16] R. Stiefelhagen, J. Yang, and A. Waibel. Estimating focus of attention based on gaze and sound. In *Workshop on Perceptive User Interfaces (PUI '01)*, Orlando, Florida, 2001.

[17] M. J. Swain and D. Ballard. Color indexing. *International Journal of Computer Vision*, 7:11–32, 1 1991.

[18] A. Yarbus. *Eye movements and vision*. Plenum Press, 1967.

[19] C. Yu and D. H. Ballard. Learning to recognize human action sequences. In *Proceedings of the 2nd International Conference on Development and Learning*, pages 28–34, Boston, U.S., June 2002.