Detecting Eye Fixations by Projection Clustering

Thierry Urruty
Laboratory of Lille 1
LIFL-UMR CNRS 8022
Université de Lille 1, France
urruty@lifl.Fr

Stanislas Lew
Laboratory of Lille 1
LIFL-UMR CNRS 8022
Université de Lille 1, France
lew@lifl.Fr

Chabane Djeraba
Laboratory of Lille 1
LIFL-UMR CNRS 8022
Université de Lille 1, France
djeraba@lifl.Fr

Dan A. Simovici
University of Massachusetts Boston
Department of Computer Science
Boston, MA, USA
dsim@cs.umb.edu

Abstract

The identification of the components of eye movements (fixations and saccades) is an essential part in the analysis of visual behavior because these types of movements provide the basic elements used by further investigations of human vision. However, many of the algorithms that detect fixations present some problems (consistency, robustness, many input parameters). In this article we present a new eye fixation identification technique that is based on clustering of eye positions using projections and projection aggregation.

1 Introduction

The eye movements are certainly the most natural and repetitive movement of a human being. The most mundane activity, watching television, or reading a newspaper involves this automatic activity which consists of shifting our gaze from one point to another.The eyes need movement since the physiology of the retina requires the eyes to move constantly in order to obtain a clean image. This frequent movement of the eye is known as the *saccadic movement*. During the exploration of a scene or an image, the eye has a tendency to remain fixated during a few milliseconds on the most significant areas; after that, the eye moves towards a new zone of interest. The trajectory of the eye consists of fixation periods interrupted by saccades that shift the eyes from an area of interest to another. The fixation periods, when the eye is almost stationary, allow the brain to process information. Their duration θ ranges between 70ms and several hundreds of milliseconds [9]. During saccades, the brain does not have the time to process the images transmitted by the visual system and the quality of the image formed on the retina is deteriorated. Thus, the brain assigns a semantics to the image only during the eye fixations. This implies that the study of fixations is essential for the understanding and interpretation of the eyes move-

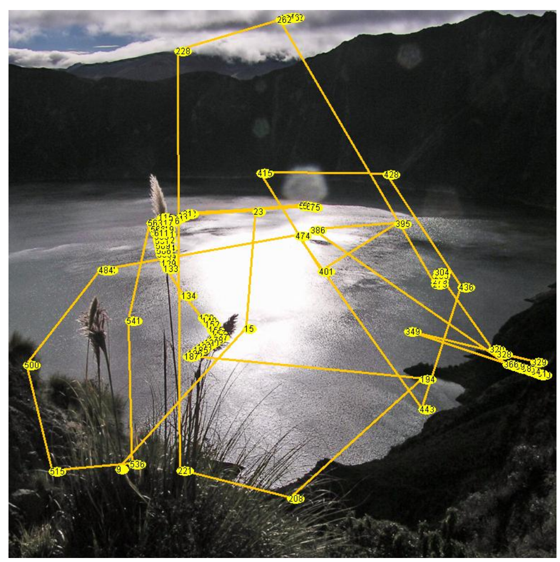

Figure 1. Example of Eye-tracking on a Fixed Image

14th International Conference of Image Analysis and Processing - Workshops (ICIAPW 2007)
0-7695-2921-6/07 $25.00 © 2007 IEEE

ments.

Most commercial systems provide horizontal and vertical coordinates of the point of regard (POR) relative to the screen of a monitor. Thus, one obtains a sequence of points corresponding to the positions of the eyes of an individual. These PORs correspond to triplets of the form $\langle x, y, t \rangle$ and reflect the eye path. The eye path consists in eye fixations (having an high density of points) separated by large spaces where only a few isolated points (saccades) are present (see figure 1).

Automatic identification of fixation and saccades is an essential part in the analysis of visual behavior. Indeed, the saccades and fixations are often used as basic knowledge for the many metrics that are used for interpreting eye movements (number of fixations, saccades, duration of the first fixation, average amplitude of saccades, etc). A good survey on this topic is [6].

The most widespread identification technique is by computing the velocity of each point (defined as the angular speed of the eye in degrees/sec). The velocity of a point correspond to the distance that separates it from its predecessor or successor [4]. Separation of points into fixations and saccades is achieved by using a velocity threshold. The disadvantage of this approach is its lack of robustness and its behavior with respect to slow eye movements. When the eye moves slowly the algorithm has the tendency of grouping together a large number of PORs. Also, the choice of parameters is often difficult and requires experimental work for determining the most appropriate values.

Other algorithms use the fact that fixation points have the tendency of being close to each other. For example, in [12] an *dispersion* algorithm identifies as fixations groups of points having dispersion lower than a certain threshold. In [5] a purely spatial approach is proposed using a *minimum spanning tree* (MST) of the set of points. This method allows the maximization of the inter-cluster separation by eliminating the edges of the MST that exceed a certain length. This technique ignores the temporal aspect of the eye movements and confuses points that are closed spatially but distant temporarily. Its use is recommended in the identification of the areas of interest. In [10], the reader will find a comprehensive, comparative study of identification algorithms for identification of fixations and saccades. Recently a new method was proposed by [11] that uses a procedure called *Mean shift* to group data. This is an iterative process that searches a local maximum in a d-dimensional space by shifting each point of the space towards higher-density areas (the direction of the gradient) in order to improve cluster separation until such movements involve a small number of points.

Application of several algorithms for detecting eye fixation can lead to totally divergent result interpretations. Also, the choice of parameters on which algorithms are based influences considerably the quality of identification of eye fixations. Santella [11], has identified the three criteria for a good clustering algorithm for fixation identification: *consistency*, *robustness*, *unsupervised*. Independent of the precision, flexibility and robustness of the identification algorithm, the identification of an eye fixation remains a relatively subjective problem. Thus, it is rather difficult to evaluate the quality of an algorithm, and, in most situations, this evaluation is based on comparing the results of an algorithm with those provided by a "certified" observer.

The algorithm that we propose is based on aggregating clusters formed by uni-dimensional projections of eye-tracking data. It deals efficiently with consistency, robustness and input parameters, with asymptotic cost $O(N \log N)$, where N is the number of fixation points. The algorithm is based on mathematic foundations that reduce subjective interpretations. Finally, the algorithm is a pioneer study exclusively destined to digital images. We begin by presenting the main ideas that underlie our algorithm, describe its implementation, and present experimental results.

2 Projection Clustering

Periodically, surveys or monographs (see [7, 8]) are published that summarize progress made in developing new clustering methods and new applications of clustering. Clustering in spaces with low dimensionality can be applied with rather high computational efficiency. The notion of projected clustering was introduced by Agrawal et al. in [3], who made the crucial observations that points may cluster better in subspaces of lower dimensionality than in the entire space $\mathbb{R}^n$. They developed the CLIQUE algorithm that works starting with low dimensional subspaces towards higher dimensional subspaces. In [2] Aggarwal et al. focus on a technique to discover clusters in small dimensional subspaces, which is the focus of their PROCLUS algorithm. The number of clusters is a given parameter in PROCLUS and the algorithm identifies these clusters and a set of dimensions associated with each cluster such that the points of the cluster are correlated with these dimensions. Another contribution to projective clustering is [1], where an objective function is introduced that takes into account a trade-off between the dimension of a subspace and the clustering error; an extension of k-means to projective clustering in arbitrary subspaces is introduced.

We develop a specialized clustering algorithm for detecting eye fixations in data that results from recording the position of the eyes of a person who examines fixed images. Our approach is similar to the approach adopted in [3] in that we construct clusters in one dimensional spaces and, then gather information from each dimension to identify clusters in the original data set. Our clustering algorithm combines ideas from projection techniques and density-based clustering. The distance between points in $\mathbb{R}^n$ is the Euclidean distance. The proposed algorithm is applicable to numeric data, that is, to data in $\mathbb{R}^n$ and involves projecting the data on a randomly chosen base. Then, histograms of the uni-dimensional projections are combined to yield the locations of clusters in $\mathbb{R}^n$.

Let S be a finite subset of $\mathbb{R}^n$ and let δ, γ be two positive real numbers. If C is a Borel subset of $\mathbb{R}^n$, let $\mathsf{vol}(C)$ the value of its Lebesgue measure which we regard as its volume.

A (δ, γ)-*clustering of* S is a family $\kappa = \{C_1, \ldots, C_p\}$ of non-empty subsets of $\mathbb{R}^n$ (the clusters of κ) which satisfy the following conditions: the subsets of κ are pairwise disjoint; for every i, $1 \leq i \leq p$, the density of the points of S in each of the sets C_i is larger than δ; the fraction of points located outside the sets C_i is no larger than γ, that is, $\frac{|\mathsf{UNC}(\kappa)|}{|S|} \leq \gamma$, where $\mathsf{UNC}(\kappa) = S - \bigcup_{i=1}^{p} C_i$ is the *set of unclustered points of* S. The *classes* of the clustering κ are the sets $S \cap C_i$ for $1 \leq i \leq p$.

Identifying the clusters in a uni-dimensional is a process that can be accomplished in linear time once the points of this space have been sorted (which, of course requires $O(n \log n)$ time, where n is the number of points). To this end we use an algorithm which constructs the histogram of the distribution of the number of projected points.

If the projection of a bi-dimensional set K on say, the x axis, is a cluster it is not possible to conclude that K is a bi-dimensional cluster since the points of K can be widely dispersed relative to the y coordinate. However, if both projections of K are uni-dimensional clusters, then we can conclude that K itself is a cluster.

Another difficulty may be created that two distinct bi-dimensional clusters may have uni-dimensional projections on the x-axis or the y-axis that overlap and create a single uni-dimensional cluster. We refer to this phenomenon as *x-occultation*, or *y-occultation*, respectively. If this occurs only with respect to one axis, then the projection on the other axis can serve to separate the clusters. We will see that, under some simplifying hypotheses, if the size of the clusters is relatively small compared to the size of the whole image, then the probability of having occultations on both axes is quite small.

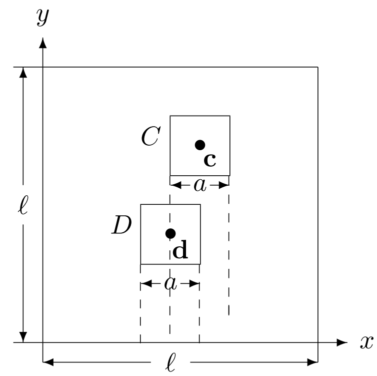

Figure 2. Occurrence of an x-occultation O_x

Let C, D be two bi-dimensional clusters. We make the following simplifying assumptions: the centers of these clusters $\mathbf{c} = (x_1, y_1)$ and $\mathbf{d} = (x_2, y_2)$, respectively are bi-dimensional independent random variables uniformly distributed in the square $[0, \ell]$ and each cluster is a square of side a. An x-occultation O_x occurs if $|x_1 - x_2| < a$; similarly, an y-occultation O_y occurs if $|y_1 - y_2| < a$.

In turn, the components x_1, x_2 of $\mathbf{c}, \mathbf{d}$ are independent random variables uniformly distributed in the $[0, \ell]$ interval. An x-occultation O_x occurs when $|x_1 - x_2| < a$; thus, the probability of an x-occultation can be easily see to be

$$P(O_i) = \frac{\ell^2 - (\ell - a)^2}{\ell^2} = \frac{2a\ell - a^2}{\ell^2}.$$

Consequently, assuming that O_x and O_y are independent, the probability that there is at least one projection without occultation is

$$P(\bar{O}_x \cup \bar{O}_y) = 1 - P(O_x \cap O_y) = 1 - P(O_x)P(O_y)$$

$$P(\bar{O}_x \cup \bar{O}_y) = 1 - \left(\frac{a}{\ell}\right)^2 \left(2 - \frac{a}{\ell}\right)^2.$$

If a is less than 10% of ℓ, a frequent situation in the image data we have analyzed, the probability of having at least one occultation-free projection is at least 96%.

We propose an algorithm for clustering eye-tracking data that is based on clustering uni-dimensional projections and on aggregating these projections. The purpose of the algorithm is to detect eye fixations and saccades starting from data acquired during viewings of single fixed images. Data obtained from tracking eye movements has a three-dimensional structure: two spatial dimensions x_1, x_2 that reflect the position of the regarded point in the image and one spatial coordinate t.

The first phase of the algorithm consists of building the uni-dimensional clusters by projecting the data set on the axes of coordinates. Thus, we obtain for each axis a density histogram (see Figure 3). In a second phase we combine the histograms to retrieve the bi-dimensional clusters (see Figure 4). The second part of the algorithm improves the clustering by using two post-processing procedures: the ϵ-expansion and the temporal slicing of the clusters based on the t-component of the data points.

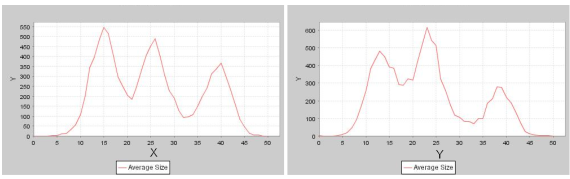

Figure 3. Density Histograms of Projections

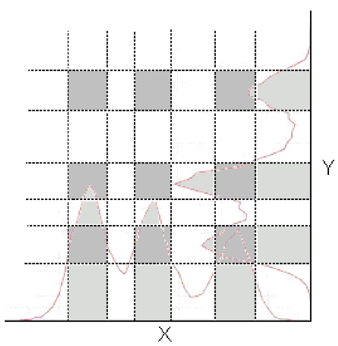

Figure 4. Combination of Histograms

In the example presented in Figures 3 and 4, the projections on the x and y axes have three peaks. The combination of these histograms indicates that we need to explore nine regions (in dark gray in the figure) that may have high densities of points.

At the end of this clustering phase, our algorithm yields a number of k of clusters that correspond to areas where point densities are important. The asymptotic cost of our algorithm is $O(n \log n)$, where n is the number of points. This time is determined by the need of sorting the projections of the points on the two axes. A second phase of the algorithm involves two post-processing techniques, the ϵ-*expansion* and the *time-slicing* process, which allow the improvement of the quality of the clustering. The post-processing techniques do not affect the asymptotic cost of the algorithm.

During the first part of the algorithm our data set is divided in k classes that are automatically determined. The combination of several intervals allows us to determine areas of high density; however, outside points located near these areas of high densities are not captured by the clustering process. It is quite possible that some of these points belong effectively to some of the clusters but are located at the outside limit of the intervals obtained in the first phase. This suggests that a secondary process that attempts to capture these points may help improve the quality of the clustering and this is indeed the case.

The first post-processing algorithm called ϵ-*expansion* attempts to expand by a factor ϵ the minimal rectangle $MBR(C)$ that includes a cluster C. Suppose that:

$$\mathsf{MBH}(C) = [a_1, b_1] \times [a_2, b_2].$$

The *density of* C is defined as the number:

$$\mathsf{dens}(C) = \frac{|C|}{\mathsf{vol}(\mathsf{MBH}(C))}.$$

An ϵ-*expansion of* C is the set $C^\epsilon = C \cup L^\epsilon$, where

$$L^\epsilon = \mathsf{UNC}(\kappa) \cap ([a_1 - |a_1|\epsilon, b_1 + |b_1|\epsilon] \times [a_2 - |a_2|\epsilon, b_2 + |b_2|\epsilon]) .$$

If $C_i^\epsilon \cap C_j^\epsilon \neq \emptyset$, then we assign the points of K_ϵ to the cluster that has the larger density among the clusters C_i^ϵ or to C_j^ϵ.

This algorithm increases the minimum bounding rectangle of a cluster by a fraction ϵ and determines if the unclassified points located in the expanded region belong to the cluster. If new points are added to the cluster in sufficient number, the minimum bounding rectangle is recomputed and a new extension is attempted. Points left unaffiliated with a cluster after this phase are considered as "noise".

The temporal dimension is important for detecting eye fixations. Recall the definition given in Section 1 of the minimal time θ of an eye fixation. Choosing $\theta = 150ms$ plays an important role in the second, "time-slicing" phase of the algorithm. This second phase consists mainly in the following steps: first, the points that belong to the same cluster may not be separated in time by more than θ, or the cluster will be split. Then, the new classes must last more than θ to avoid being considered as "noise".

It is possible to use a three-dimensional variant of our algorithm by including the temporal dimension in the clustering process and combining the histograms that correspond to the spatial projections with the histogram of the temporal projection. However, the distribution of the temporal projection is rather homogeneous. The regions of low density correspond to the loss of eye contact with the tracking device during the viewing session; these regions are rather short and contain little information.

3 Detection of Eye Fixations by Projection Clustering

We present in this section the results obtained by our projection technique for detection of saccades and eye fixations of a viewer of a fixed image. The data acquisition is achieved through a high-definition camera. In average, an user regards a fixed image for about 10 seconds, and the camera records data at a rate of 60Hz (every 17ms) for a normal usage. This yields between 600 and 700 viewing points per image. The Figure 5 shows the viewing points of an user in chronological order (yellow points).

To be able to compare various methods of detection of saccades and eye fixations we have added saccade-like data with a noise-generating algorithm. This algorithm entails several phases: first, it stores chronologically consecutive points covering the longest trajectories; then, it creates randomly noise points between consecutive points and maintaining a minimal distance to the first type of points. Thus, we generate about noise equivalent to about 10% of the number of real eye positions which we include in the data set.The Figure 5 shows the set of viewing points in yellow with the noisy data added by our algorithm shown in red on the same image. The use of an algorithm for synthesizing saccades allows us to use several customary measures that are popular in evaluation of classifiers. We define fixations well identified as *True Positive* (TP), saccades misidentified as *False Positives* (FP), saccades well identified as *True Negative* (TN), and finally fixations misidentified as *False Negative* (FN)

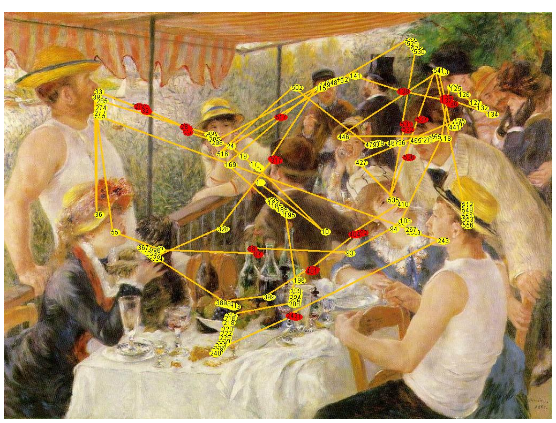

Figure 5. Example of Eye-tracking on a Fixed Image with Noise Added (red points)

In the Table 1 and 2 we show the averages of the precision, recall and F1-measure values obtained

Algorithm	TP	FP	TN	FN
MST	0.867	0.038	0.008	0.086
Dispersion	0.803	0.046	0.0	0.150
Velocity	0.846	0.046	0.0002	0.106
Our method	0.873	0.038	0.007	0.079

Table 1. Contingency Tables Comparing Our Algorithm with other Algorithms

Algorithm	Precision	Recall	F1
MST	0.957	0.909	0.932
Dispersion	0.944	0.842	0.889
Velocity	0.947	0.887	0.915
Our method	0.957	0.916	0.935

Table 2. Precision, Recall and F1 Measure Comparing Our Algorithm with other Algorithms

in the experiments involving our approach and three other algorithms (minimal spanning tree, dispersion, and velocity-based algorithms). Our results show that the proposed method retrieves efficiently the eye fixation; we miss some saccades compared to other algorithms but we lose less information that concerns eye fixations.

4 Conclusion and Future Work

We presented a new eye fixation identification algorithm based on clustering of eye positions using aggregating clusters formed by uni-dimensional projections of eye-tracking data. The algorithm, based on mathematics foundations, identifies better eye fixations. It is more consistency, robustness and less input parameters, with asymptotic cost $O(n \log n)$, where n is the number of fixation points. We will extend our algorithm to detect saccades and fixations in the context of spatial viewings. This will require separate detection of movements of both eyes and aggregation of the results. Another direction of investigation will be the development of an incremental detection algorithm that will be capable of detecting on-line fixations and saccades.

References

[1] P. Agarwal and N. H. Mustafa. k-means projective clustering. In *Proceedings of PODS*, pages 155–165, 2004.

14th International Conference of Image Analysis and Processing - Workshops (ICIAPW 2007)
0-7695-2921-6/07 $25.00 © 2007 IEEE

[2] C. C. Aggarwal, C. Procopiuc, J. L. Wolf, P. S. Yu, and J. S. Park. Fast algorithms for projected clustering. In *Proceedings of ACM-SIGMOD Conference on Management of Data*, pages 61–72, 1999.

[3] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the ACM-SIGMOD Int. Conf. Management of Data*, 1998.

[4] I. V. C.J. Erkelens. The initial direction and landing position of saccades. *Eye movement research : Mechanisms, Processes, and Applications*, pages 133–144, 1975.

[5] J. H. Goldberg and J. P. Schryver. Eye-gaze contingent control of the computer interface: Methodology and example for zoom detection. *Behavior Research Methods, Instruments and Computers*, pages 338–350, 1995.

[6] R. J. Jacob and K. S. Karn. Eye tracking in human-computer interaction and usability research: Ready to deliver the promises. *The Mind's Eyes: Cognitive and Applied Aspects of Eye Movements.*, 2004.

[7] A. K. Jain and R. Dubes. *Algorithms for Clustering Data.* Prentice Hall, Englewood Cliffs, NJ, 1988.

[8] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31:264–323, 1999.

[9] K. Rayner. *Eye Movements and Information Processing : 20 years of Research.* Eyrolles, 1998.

[10] D. D. Salvucci and J. H. Golberg. Identifying fixations and saccades in eye-tracking protocols. *Proceedings of the Eye Tracking Reseach and Applications Symposium*, 2000.

[11] A. Santella and D. DeCarlo. Robust clustering of eye movement recording for quantification of visual interest. *Eye Tracking Research and Applications (ETRA)*, 2004.

[12] H. Widdel. Operational problems in analysing eye movements. *Theoretical and Applied Aspects of Eye Movement Research*, pages 21–29, 1984.