Detecting Eye Fixations by Projection Clustering

THIERRY URRUTY, STANISLAS LEW, and NACIM IHADADDENE
University of Lille
and
DAN A. SIMOVICI
University of Massachusetts Boston

Eye movements are certainly the most natural and repetitive movement of a human being. The most mundane activity, such as watching television or reading a newspaper, involves this automatic activity which consists of shifting our gaze from one point to another.

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 a number of problems. In this article, we present a new fixation identification technique that is based on clustering of eye positions, using projections and projection aggregation applied to static pictures. We also present a new method that computes dispersion of eye fixations in videos considering a multiuser environment.

To demonstrate the performance and usefulness of our approach we discuss our experimental work with two different applications: on fixed image and video.

Categories and Subject Descriptors: H.5.2 [**Information Interfaces and Presentation (e.g., HCI)**]: User Interfaces; I.4.9 [**Image Processing and Computer Vision**]: Applications

General Terms: Algorithms, Experimentation, Performance

Additional Key Words and Phrases: Eye fixations, interaction modeling, projected clustering, static pictures, videos

ACM Reference Format:

Urruty, T., Lew, S., Ihadaddene, N., and Simovici, D. A. 2007. Detecting eye fixations by projection clustering. ACM Trans. Multimedia Comput. Comm. Appl. 3, 4, Article 23 (December 2007), 20 pages. DOI = 10.1145/1314303.1314308 http://doi.acm.org/10.1145/1314303.1314308

1. INTRODUCTION

Eye movements are certainly the most natural and repetitive movement of a human being. The most mundane activity, like watching television or reading a newspaper, involves this automatic activity which consists of shifting our gaze from one point to another.

Information obtained by tracking the gaze of users is of increasing interest for industry because it represents the interests that users have for specific objects. One of the most important applications is in marketing (the advertising of products or services as images, posters, newspaper advertising, pop-ups,

Authors' addresses: T. Urruty (contact author), S. Lew, N. Ihadaddene, University of Lille, Batiment A3-Cité Scientifique, 59655 Villeneuve d'Ascq Cedex, France; email: Thierry.urruty@lifl.fr; D.A. Simovici, University of Massachusetts Boston, 100 Morrissey Blvd., Boston, MA 02215-3393.

videos, etc.), where the goal is to determine the areas of an image that attract the attention. Another significant area of application is in ergonomic studies.

Estimating the efficacy of an advertising campaign or a web advertising effort was discussed in Cowen et al. [2002] and Baccino and Colombi [2001]. However, relatively few results were published concerning user behavior during video sessions. In 1964, the first study, discussed in Guba [1964], showed that a person watching a movie shows a strong tendency for viewing the faces of the characters present in a scene. This tendency persists even in the presence of strong distracting elements. More recently, Tosi et al. [1992] have shown that during the viewing of fast-changing sequences of images, viewers have a strong tendency of focusing on the center of the screen. In the same framework, Goldstein [2006], using a group of 20 subjects watching several types of video clips, has shown that the gaze has the tendency of focusing on the same spot and that this phenomenon varies with the age of the subjects. Starting from this observation we have investigated the scattering of fixation points, using a study involving 30 participants and three types of videos.

We present in this article a system that detects eye fixations in static images and video. In static pictures, the number of fixations is high. So, to interpret the eye fixations, which is the objective of eye fixation detection, it is necessary to simplify/reduce them in a suitable form. The basic idea is that we understand and better exploit eye fixations when they are represented in simple form. The simplification consists of reducing noise and clustering the fixations that converge in the same region of interest. In this case, we present a new method for extracting the gaze fixations-starting from points captured from an eye-tracking system. In videos, the processing is different. Video is composed of a high-number of frames. For example, three minutes of video is composed of 4500 frames; hence the number of fixations per frame is limited compared to static pictures. In this case, we will not focus on clustering of eye fixations robust to noise, but on the degree of fixation dispersion of several users. We will focus also on the extraction of regions of interest in the frames, based on eye fixations. The limited number of points in video frames give us the opportunity to deal with new problems that are much harder to resolve in static pictures, namely the detection of eye fixations in a multiuser environment.

We begin by defining fixation points relative to saccades and then we present the state-of-art of various methods for detecting saccades and fixations. We propose our fixation detection algorithm using a clustering-by-projection algorithm. To demonstrate the performance and usefulness of our approach we discuss our experimental work with two different applications: on fixed image and video. In the second part of the article we present an application of our algorithm to the detection of gaze fixation on video documents, which aims to analyze user behaviors of several individuals that watch short advertising videos.

2. IDENTIFYING SACCADES AND FIXATIONS, RELATED WORKS

We use our eyes without worrying about the processes that underlie their good functioning. However, these underlying mechanisms are very complex and constitute an entire psychomuscular complex. The greater part of eye movements are realized involuntarily; they are very efficient and used very frequently. They are the consequence of the enormous amount of visual information that surrounds us, which compels us to select certain portions of our visual field in order to process the stimuli more efficiently. Furthermore, 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 *saccadic movement*.

During exploration of a scene or image, the eye has a tendency to remain fixated for a few milliseconds on the most significant areas of an image; 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 one area of interest to another (see Figure 1). The fixation periods, when the eye is almost stationary,

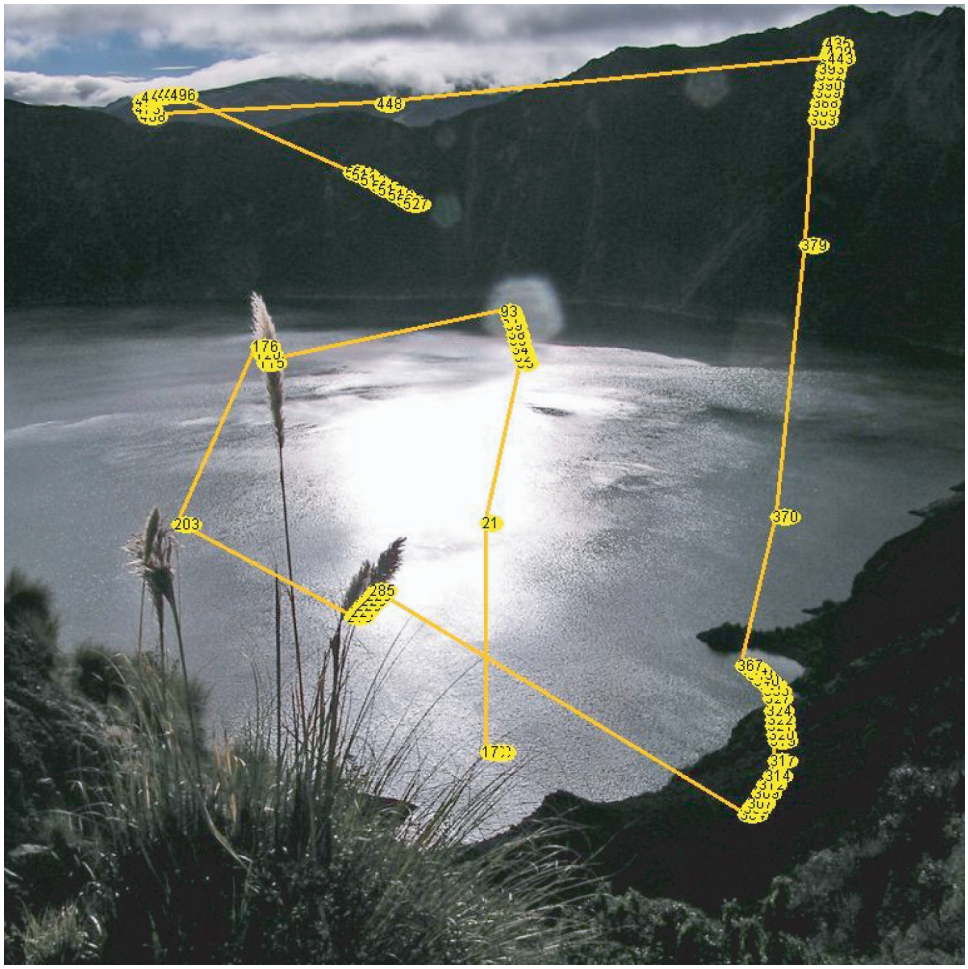

Fig. 1. Eye fixations separated by saccades.

allow the brain to process information. Their duration θ ranges between 70 ms and several hundreds [Poole et al. 2004; Rayner 1998].

During these 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 (see Ross and Burr [1994] and [Banks and Anderson 1991]). Thus, the brain assigns a semantics to the image only during the eye fixations. This implies that study of fixations is essential for the understanding and interpretation of eye movements.

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 important density of points) separated by large spaces where only a few isolated points are present. The sequence of fixations and saccades is also called *scanpath*.

Algorithm 1. Velocity Threshold Algorithm

Input : Ep : EyePath composed of N points of regard
Vt : Velocity threshold parameter
`Calculate point-to-point velocities for each point of` Ep `Label each point below` Vt `as a fixation point, otherwise as a saccade point`
`Collapse consecutive fixation points into fixation groups, removing saccade points Map each fixation group to a fixation at the centroid of its points`
`Return fixations`

Automatic identification of fixation and saccades is an essential part in the analysis of visual behavior. Indeed, 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.). Good surveys on this topic are Jacob and Karn [2004] and Poole et al. [2004].

The most widespread identification technique is by computing the velocity of each point (defined as the angular speed of the eye in degrees per second). The velocity of a point corresponds to the distance that separates it from its predecessor or successor [Erkelens 1975]. Separation of points into fixations and saccades is achieved by using a velocity threshold. Successive points labeled *fixations* are then grouped into what can be considered as an eye fixation. Another threshold involving a minimal duration of a fixation allows the elimination of insignificant groups.

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

Algorithm 2. Dispersion Detection Algorithm

Input : Ep : EyePath composed of N points of regard
Di : Dispersion threshold
Du : Duration threshold
Variables : pr_i : the ith gaze point of Ep
F : a fixation group
for $i = 0$ to N **do**
  while `Duration(F)<Du` **do**
    $F.add(pr_i)$
    $i \leftarrow i + 1$
  end while
  if $Dispersion(F) \leq$ Di **then**
    repeat
      $F.add(pr_i)$
      $i \leftarrow i + 1$
    until $Dispersion(F) >$ Di
    `map F to the centroid of its points`
  end if
end for

Algorithm 3. Minimum Spanning Tree Identification

Input : EyePath Ep consists of N gaze points
Edge ratio : Er
Edge standard deviation : Sd

```
Construct MST from eyepath using Prim's algorithm
Find maximum branching depth for each MST point using depth-first search
Identify saccades as edges whose distances exceed predefined criteria
Compute parametric properties (μ, σ) of local edges
```

$$\frac{EdgeLength}{\mu} > Er$$

Identifying edge as saccade if: and

$$EdgeLength > \mu + Sd$$

```
Identify fixations as clusters of points not separated by saccades
Return fixations
```

number of PORs. Also, the choice of parameters is often difficult and requires experimental work for determining the most appropriate values.

An extension of the velocity algorithm makes use of a hidden Markov model to obtain a more robust classification [Salvucci and Golberg 2000]. However, the method requires the creation of a training set in order to estimate the values of parameters, which makes it quite expensive.

Other algorithms that do not use velocities use the fact that fixation points tend to be close to each other. In Widdel [1984] the auther uses an algorithm that identifies as fixations groups of points having dispersion lower than a certain threshold. In the same reference the eye track is scanned by a window of variable size that computes the dispersion of consecutive points. This window is expanded as long as the dispersion of the points is lower than a certain parameter. This approach is useful for online identification of eye fixations.

Finally, in Goldberg and Schryver [1995] a purely spatial approach is proposed, using a *minimum spanning tree* (MST) of the set of points. This method allows maximization of the intercluster separation by eliminating edges of the MST that exceed a certain length. This technique ignores the temporal aspect of eye movements and confuses points that are closed spatially, but distant temporally. Its use is recommended in the identification of areas of interest.

In Salvucci and Golberg [2000], the reader will find a comprehensive comparative study of identification algorithms for identification of fixations and saccades.

Recently, a new method was proposed by Santella and DeCarlo [2004] 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.

In Karsh and Breitenbach [1983] it is shown that the application of several algorithms for detecting eye fixation can lead to totally divergent result interpretations. Also, the choice of parameters on which the algorithms are based influences considerably the quality of identification of eye fixations. In Goldberg and Wichansky [2003] the authors argue for an increased level of formalization in this domain.

Existing commercial systems propose automatic methods for detection of eye fixations. Unfortunately, most suffer from a certain lack of transparency regarding the algorithms and from a strong dependency on the choice of parameters [Lankford 2000]. Santella and DeCarlo [2004], have identified the following criteria for a good clustering algorithm for fixation identification.

—*Consistency*. The algorithm should not depend on a random initialization.

—*Robustness*. Outliers should not influence the construction of the clusters.

—Moreover, the algorithm should not require the number of clusters as an input parameter.

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 unidimensional projections of eye-tracking data. We begin by presenting the main ideas that underlie our algorithm, then describe its implementation and present experimental results.

3. OUR APPROACH: PROJECTION CLUSTERING

Various data mining applications, including multimedia data mining, make use of clustering [Chaudri et al. 2002; Zaïane et al. 2002; Djeraba 2003]. Periodically, surveys or monographs [Jain et al. 1999; Jain and Dubes 1988] are published that summarize progress made in developing new clustering methods and applications.

Clustering in spaces with low dimensionality can be applied with rather high computational efficacy. For example, in a unidimensional space it is easy to identify regions of high density of points by a simple linear scan. With increased dimensionality the problem grows in complexity.

The notion of projected clustering was introduced by in Agrawal et al. [1998], who made the crucial observation 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 ones. In Aggarwal et al. [1999] the authors focus on a technique to discover clusters in small-dimensional subspaces, which is the focus of their Proclus algorithm. The theoretical support of these techniques can be found in the Johnson-Lindenstrauss lemma [Johnson and Lindenstrauss 1984], which asserts that a set of points in a high-dimensional Euclidean space can be projected into a low-dimensional Euclidean such that the distance between any two points changes by only a factor of $1 \pm \epsilon$ for $\epsilon \in (0, 1)$. Simplifications of the proof of this result have been obtained by Frankl and Maehara [1988] and by Dasgupta and Gupta [1999]. An especially useful source is the monograph Vempala [2004].

The number of clusters is a given parameter in Proclus and the algorithm identifies these clusters, as well as 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 Agarwal and Mustafa [2004], where an objective function is introduced that takes into account a tradeoff 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 one adopted in Agrawal et al. [1998] in that we construct clusters in low-dimensional spaces and then select those dimensions that can best help to identify clusters in the original dataset. Our main contribution consists in choosing a random frame of reference for the dataset and executing the projections on the subspaces that correspond to this randomly chosen axis. We show that this process has a certain advantage over using the "natural" system of coordinates in

that it diminishes the chance of the occultation phenomenon, which occurs when the projections of two distinct clusters of data on a subspace are not disjoint. Static segmentation of images (regarded as partitioning an image) into a number of regions that represent a meaningful part of the image can be helped, as we show, by applying clustering techniques [Jain and Flynn 1996]. Our clustering algorithm combines ideas from random 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 unidimensional 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 nonempty subsets of $\mathbb{R}^n$ (the clusters of κ) which satisfy the following conditions.

(1) The subsets of κ are pairwise disjoint.

(2) 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, that is,

$$\frac{|S \cap C_i|}{\mathsf{vol}(C_i)} \geq \delta.$$

(3) 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 $1 \leq i \leq p$.

The second condition of the previous definition assures us that the point density in each of the sets C_i is sufficiently large; the third condition limits the fraction of the points that are not clustered.

Identifying the clusters in a unidimensional manner 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 (described in Section 3) which constricts the histogram of the distribution of the number of projected points.

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

Another difficulty may be created in which two distinct bidimensional clusters have unidimensional projections on the x-axis or y-axis that overlap and create a single unidimensional cluster. We refer to this phenomenon as x-, 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.

Let C, D be two bidimensional clusters. We make the following simplifying assumptions:

(1) The centers of these clusters $\mathbf{c} = (x_1, y_1)$ and $\mathbf{d} = (x_2, y_2)$, respectively, are bidimensional independent random variables uniformly distributed in the square $[0, \ell]$.

(2) Each cluster is a square of side a.

An x-occultation O_x occurs if $|x_1 - x_2| < a$; similarly, a y-occultation O_y occurs if $|y_1 - y_2| < a$.

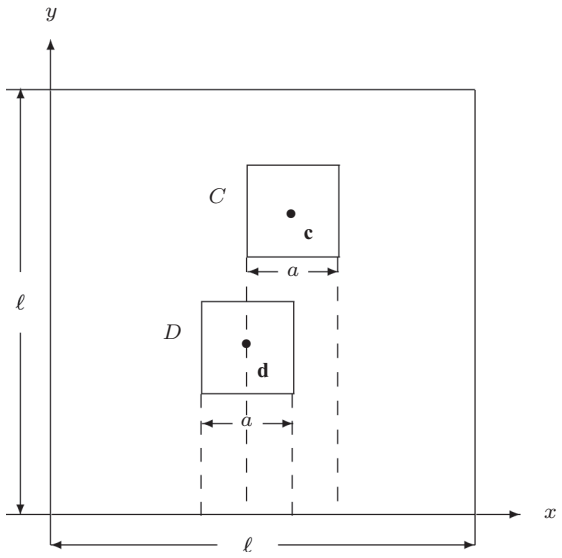

Fig. 2. Occurrence of an x-occultation O_x.

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 $|x_1 - x_2| < a$ (see Figure 2); 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 that shows no occultation is

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

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 unidimensional 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 unidimensional clusters by projecting the dataset 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 bidimensional clusters (see Figure 4).

The second part of the algorithm improves the clustering by using two postprocessing procedures: ϵ-expansion, and temporal slicing of the clusters based on the t-component of the data points.

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 (shown in dark gray in the figure) that may have high densities of points.

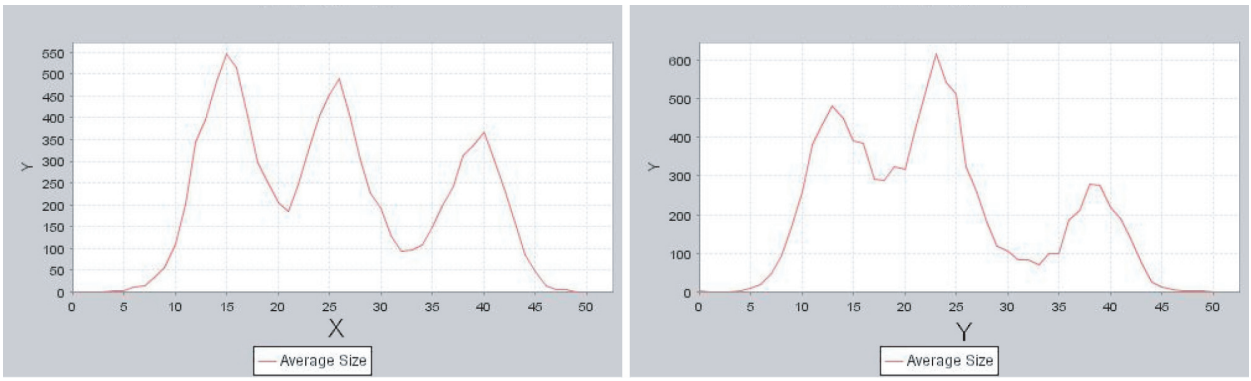

Fig. 3. Density histograms of projections.

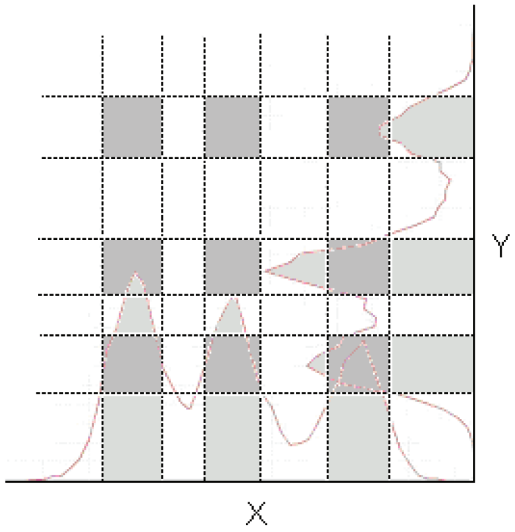

Fig. 4. Combination of histograms.

At the end of this clustering phase, our algorithm yields a number 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 for sorting the projections of the points on the two axes.

This second phase of the algorithm, involving the postprocessing techniques of ϵ-*expansion* and *time-slicing*, allows the improvement of the quality of the clustering. The postprocessing techniques do not affect the asymptotic cost of the algorithm.

During the first part of the algorithm our dataset 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 density 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 postprocessing 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

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

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 whether 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 $\theta = 150$ ms plays an important role in the second, "time-slicing" phase of the algorithm. This second phase consists mainly in the following steps:

—Points that belong to the same cluster may not be separated in time by more than θ, or the cluster will be split.

—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.

4. OUR APPROACH FOR STATIC PICTURES

4.1 Experimental Setup and Procedure

To measure the quality of our method we established the following experimental protocol.

Five subjects whose ages were, on average, 25 years ($\pm$ 5ans) volunteered for this experiment. They had no special vision problems. Each subject was required to view in a natural manner images displayed on a monitor. Each recording session consisted of 20 images divided into four categories (sport, art, landscape, and advertisements). To avoid visual boredom, each image was displayed for 10 s, totalling slightly more than 3 m for each subject.

During data recording, all 20 images were presented one after the other on the screen, separated in time by a black image displayed for 1/2 s. Eye movements were recorded monocularly with an Eye-Response Erica eye-tracking system associated with gazeTracker software. The sampling gaze of the Erica system is 60 Hz with a gaze position accuracy of 0.5°. Prior to each recording, a 16-point spatial calibration was performed.

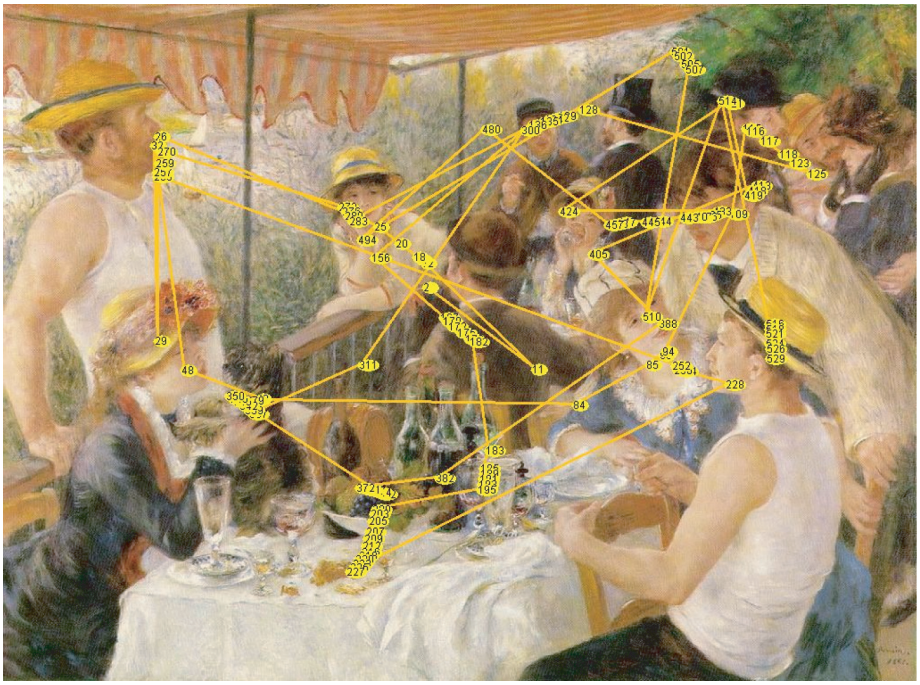

Fig. 5. Eye tracking on a fixed image.

4.2 Data Analysis and Results

We present in this section the results obtained by our projection technique for detection of saccades and eye fixations from a viewer of a fixed image. An image was displayed for about 10 s, and the camera recorded data at a rate of 60 Hz (every 17 ms) for a normal usage. This yields between 600 and 700 viewing points per image. Figure 5 shows the viewing points of a user in chronological order.

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, given as follows:

—store chronologically consecutive points covering the longest trajectories; and

—randomly create noise points between consecutive points and maintain a minimal distance to the first type of points.

Thus, we generate noise about equivalent to 10% of the number of real eye positions which we include in the dataset.

Figure 6 shows the set of viewing points in yellow, with the noisy data added by our algorithm shown in red on the same image.

Figure 7 represents fixation points by small colored squares and saccades with black crosses placed on the positions of the eye. This allows us to validate visually the quality of our classification.

The use of an algorithm for synthesizing saccades allows us to use several customary measures that are popular in evaluation of classifiers. The terms are described in Table I (see Tan et al. [2006, p. 549]).

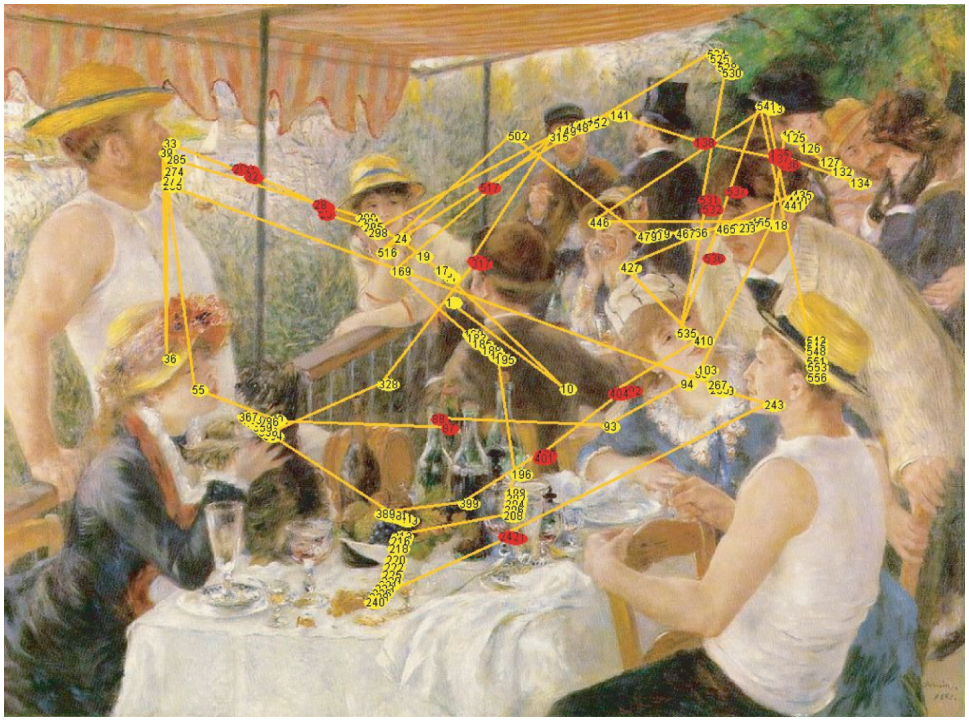

Fig. 6. Example of eye tracking on a fixed image with noise added.

The precision and recall are given by

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

and

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$

In Table II, F_1 is the harmonic average of the precision and recall. In the table we show the averages of the values obtained in experiments involving our approach and three other algorithms (minimal spanning tree, dispersion, and velocity-based algorithms). Our results show that the proposed method efficiently retrieves the eye fixation; we miss some saccades compared to other algorithms, but we lose less information that concerns eye fixations.

5. OUR APPROACH IN VIDEO DOCUMENTS

A video consists of a sequence of images (called *frames*) that are shown at a certain frame rate (see Figure 8). The perception of fluidity of the video presentation depends on the value of this parameter. The most common value of the frame rate is 25 Hz, which means 25 frames per second. Starting from the frame rate, it is possible to determine the frame shown at time t, which allows us to find the frame on which the point of regard is located. The usage of frames imposes a temporal discretization of the video document. By studying the position of the eye gaze in a given frame, we insure temporal consistency of position of the gaze points of several individuals, which allows comparisons between these points. The

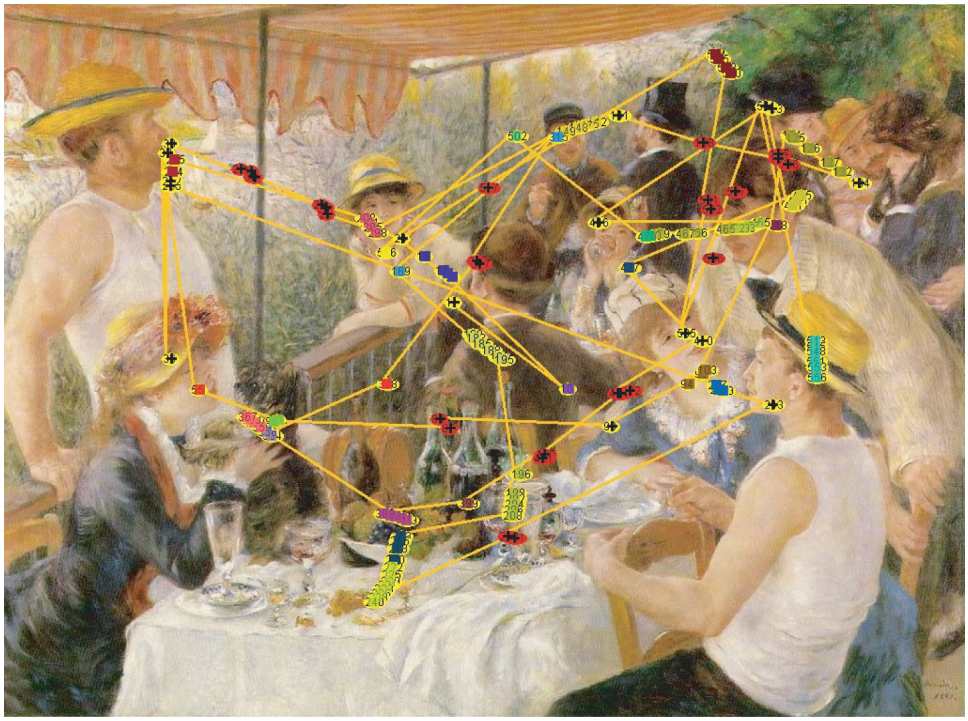

Fig. 7. The results of our classification algorithm on a fixed image.

Table I. Definitions and Results of the Contingency Table

True Positive (TP)	Fixations well identified
False Positives (FP)	Saccades misidentified
True Negative (TN)	Saccades well identified
False Negative (FN)	Fixations misidentified

Table II. Contingeny Tables Comparing Our Algorithm with Other Algorithms

Algorithm	TP	FP	TN	FN	Precision	Recall	F1
MST	0.867	0.038	0.008	0.086	0.957	0.909	0.932
Dispersion	0.803	0.046	0.0	0.150	0.944	0.842	0.889
Velocity	0.846	0.046	0.0002	0.106	0.947	0.887	0.915
Our method	0.873	0.038	0.007	0.079	0.957	0.916	0.935

sampling rate of the eye-tracking system that we use allows us to obtain between 2 and 10 gaze points per frame.

In Gulliver and Ghinea [2004] the authors compute the centroide of a set of points such that the trajectory of a user's gaze on a video consists of one point per frame. The goal of this work is to compute the average trajectory of a group of users. This allows superimposition of the gaze points on their corresponding frames and the visualization of areas that attracted the attention of the viewers (see Figure 9).

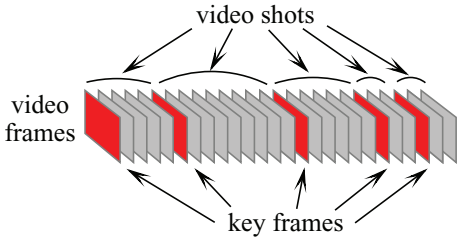

Fig. 8. Decomposition of a video sequence.

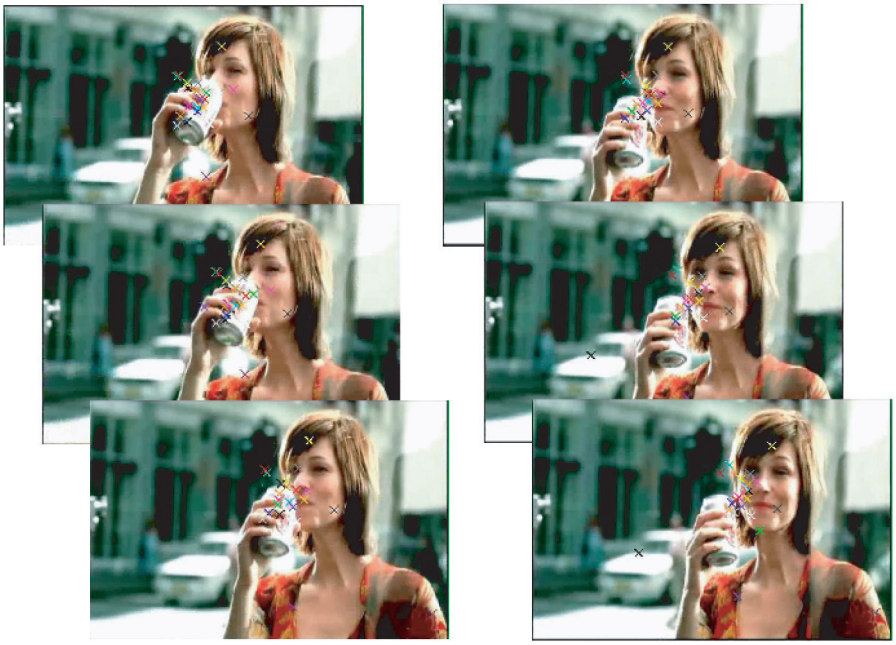

Fig. 9. Superimposed points of regard in video frames.

5.1 Dispersion of Points of Regard Set

In this section we apply a technique proposed in Goldstein [2006] for measuring the dispersion of the set of points of regard on a frame. Our goal is to visualize the evolution of the dispersion of user regards, depending on the content of the video. This allows the determination of key parts of the video documents, when the gazes are concentrated on small areas.

There is a large number of numerical criteria allowing the evaluation of the dispersion on a set X of objects in a given space. The choice of a particular criterion depends on the type of data that is under study and on the goal of the study. In our case, we deal with two-dimensional space and the objects are the points of regard. In Crossland and Rubin [2002] the authors use a measure based on the "bivariate contour ellipse area." This measure evaluates the area of an ellipse, including a certain percentage of the points of the initial set, as

$$\text{BVCEA} = 2k\pi\sigma_H\sigma_V(1-\rho^2)^{1/2}.$$

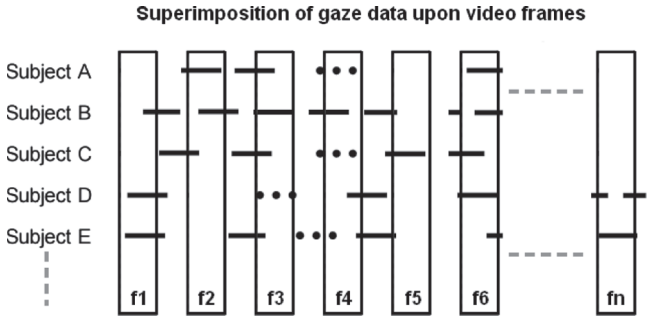

Fig. 10. Only points of regard tagged as fixations (horizontal line) are used in dispersion calculus; saccades (dotted line) are removed.

Here σ_h is the standard deviation of the point locations over the horizontal meridian, σ_h is the standard deviation of the point locations over the vertical meridian, and ρ is the correlation coefficient of the horizontal and vertical values. The k parameter of the BVCEA determines the enclosure of the ellipse. In Goldstein [2006] the value of k is set to 1, for which 63% of the points are enclosed by the ellipse. Calculation of the BVCEA does not require that an ellipse be fitted to all gaze points.

However, this measure of dispersion is very sensitive to outliers; indeed, a small number of scattered points may cause a significant increase of the area of the ellipse. Thus, this measure requires preprocessing the data in order to eliminate outliers. Furthermore, if the set of points possesses strong segmentation in several subsets, then the BVCEA does not at all reflect the dispersion of the points of regard. Starting from these observations, we propose to use the *Gini coefficient* to better quantify the degree of concentration (or dispersion) of the set of points of regard that are present in a frame.

The Gini coefficient was initially used for studying the distribution of income levels in an economy. To compute this coefficient for the set of points of regard, we partition the frame into pq rectangular cells. If the cell (i, j) contains n_{ij} points, then the Gini coefficient is computed as

$$\text{Gini} = 1 - \sum_{i=1}^{p} \sum_{j=1}^{q} \frac{n_{ij}^2}{n^2},$$

where $n = \sum_{i=1}^{p} \sum_{j=1}^{q} n_{ij}^2$ is the total number of points. The value of the Gini coefficient is a number in the interval [0, 1). A value close to 1 indicates a strong dispersion of points; small values close to 0 occur when there is a strong concentration of points. Only those points of regard determined to be fixations by the algorithm discussed in Section 3 are taken into account in the computation of the dispersion.

5.2 Experimental Setup and Procedure

The choice of videos in our experiments is significant because the contents of these videos influence the visual behavior of the subjects, and therefore the information that we will extract from the experiments. Our initial experiments involved three videos.

This choice is motivated by the time availability of our subjects and by the fact that a large number of videos will generate user fatigue which would impact the quality of our experiments. The order of presentation of the videos is randomly chosen for each user.

We used three criteria in our choice for the videos, detailed next.

(1) The similarity of the quality of the videos is intended to reduce or eliminate user bias.

Fig. 11. Measurements performed on three video types.

(2) A second criterion is the duration of the video sequences. We have limited the total duration to 3 m in order to avoid visual fatigue. Moreover, since the visualization constraints are relatively strong, the probability that the regard of the subject leaves the field of the camera increases strongly with increase in length of the video.
(3) Finally, the most important criterion is the content of the sequences. The objective of this experiment is to validate our measure of similarity. This requires videos with very different contents such that there are real disparities at the level of the regard trajectories.

We chose the following videos:

—the movie trailer for Star Wars (the third episode);
—an advertisement for a famous cola drink; and
—a fragment of an air show.

The Movie Trailer. This trailer consists of very short sequences with many moving objects. The slower sequences often signal an important moment of the trailer (initial title, dialog between two characters, etc.). There are many distracting elements.[1] We conjecture that such a video will produce regard trajectories that are specific to each subject. The duration of the video is 55 s.

The Air Show. This clip shows a famous American fighter plane that performs several flying maneuvres. This video is relatively slow and most of the frames consist of a single element that may attract the attention of the viewers. This type of content suggests that the regards of the users will tend to converge towards the same area. The duration is 24 s.

The Advertising. The video advertisement is a good compromise between the preceding two videos. The sequences are rather short and proceed in rapid succeession; however, the movements of the objects in view of the camera are rather few. In this video we felt it would be interesting to examine the timing of occurrences of the product advertised in the image. The duration is 54 s.

Our study involved 26 participants between 22 and 65 years of age, most of whom were engineering school members and computer literate. As in the first experiment, eye movements were recorded with an Erica eye-tracking system.

5.3 Data Analysis and Results

We recorded the eye movements of 26 participants using the three videos, which yielded 78 datasets. This yielded about 10, 5000 regard points. For each of the eye trackings we applied our projection clustering technique in order to identify eye fixations and saccades.

[1]that is, objets that are likely to attract the regard of the viewer.

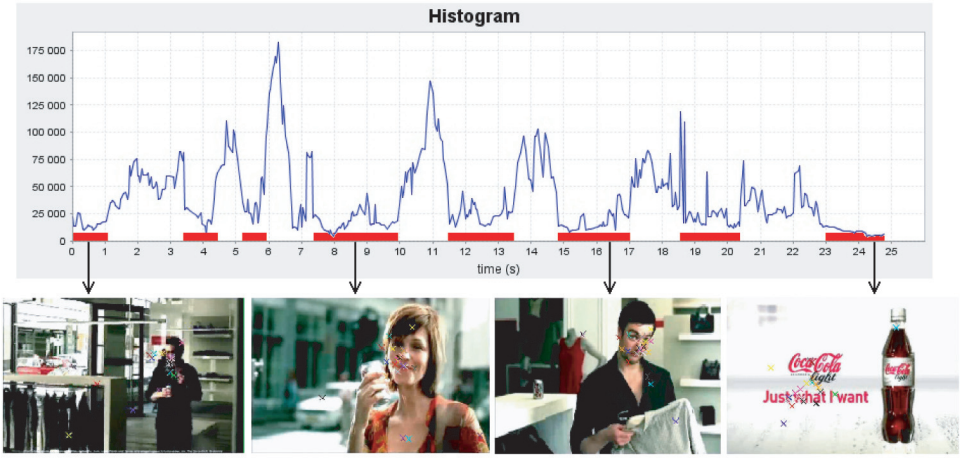

Fig. 12. Extraction of video sequences starting from the dispersion histogram.

Since no cognitive process is possible during a saccade, we have chosen to ignore the points identified as saccades in the evaluation of the dispersion. For each of the 26 eye trackings corresponding to a video, we assigned each fixation point to a frame (corresponding to the time code of the point; refer to Section 5).

We computed the BVCEA measure and Gini index for each frame of the three videos. By tracing the evolution of these measures, it is possible to detect those scenes where the regards of the subjects are concentrated on a small zone. Because of the data acquisition conditions and tracking losses of the system, the number of points on each frame is not constant (see Figure 10). To alleviate this problem we considered only the frames that contain at least 21 points of regard (out of 26 possible ones). The set of accepted frames represents 91% of the total number of frames.

The results indicate that the content of a video has a real impact on the visual behavior of the subjects. The advertising histogram shows that periods of small dispersion values correspond in general to precise time periods of the video (see Figure 12). We also observe a small dispersion when the advertised object appears in the large plan, or when at the end of the advertising spot the logo of the brand appears. A similar phenomenon is observed in the clip of the air show; when the aircraft flies alone, the totality of regards are fixated on this object. In the movie trailer, one observes a weak dispersion at the beginning and end of the clip, when the production company logo and date of film release are shown.

The structure of the diagrams seems to show that the pics of the divergency often correspond to changes of plan. These sudden changes have the tendency to distract the gaze of the subjects; after several milliseconds (and the time needed to assimilate the new content), the gaze is repositioned on a precise area of the scene.

Although the two histograms seem rather distinct, close examination reveals certain similarities. The correlation coefficient between the two sequences is 0.74. However, the Gini coefficient seems to be less sensitive than BVCEA to the segmentation of the set of points.

5.4 Extraction of Region of Interest

Our algorithms for extracting fixation points can be used in several applications. We discuss here two of them.

We used a segmentation algorithm for images on the set of images of the advertising video. In Figure 15 we show the image with the regard fixation points, the segmented image, and finally the

Fig. 13. The histogram of the dispersion computed by using the Gini index.

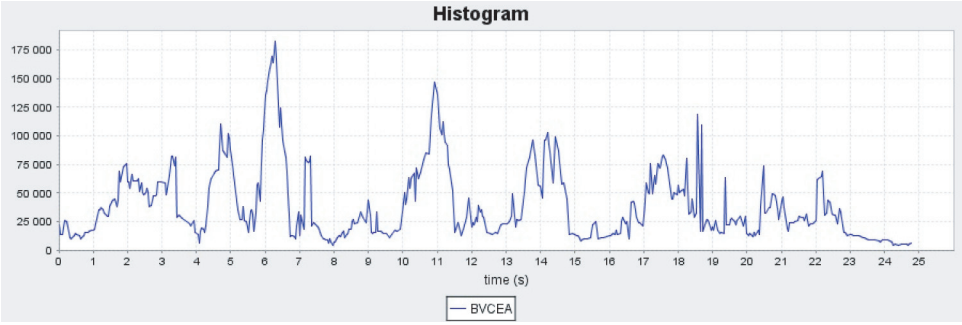

Fig. 14. Histogram of dispersion with BVCEA.

Fig. 15. Extraction of relevant objets of a video using the points of regard.

reconstructed image starting from the segments that contain fixation points. It is next possible to reconstruct the entire video in order to determine the areas of interest in the video. Information obtained in this manner can be a powerful analytic marketing tool.

Another possible application is the summarization of videos by combining various pieces of information: areas of interest of an image (established by using the points of regard), the key images of a video, and the dispersion of points of regard. These summaries may serve afterwards for the indexation of video databases and allow fast access to interesting videos during the searches performed on these databases.

6. DISCUSSION AND FUTURE WORK

In this article, we presented a new method for static pictures that detects eye fixations in a simplified form. The method is a clustering of eye positions using projections and projection aggregation. We

showed how this method is efficient in dealing with eye fixations as multidimensional information. We also presented a method for videos that estimates the degree of dispersion of eye fixations in a multiuser environment.

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 online fixations and saccades.

REFERENCES

AGARWAL, P. AND MUSTAFA, N. H. 2004. k-means projective clustering. In *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, 155–165.

AGGARWAL, C. C., PROCOPIUC, C., WOLF, J. L., YU, P. S., AND PARK, J. S. 1999. Fast algorithms for projected clustering. In *Proceedings of ACM-SIGMOD Conference on Management of Data*, 61–72.

AGRAWAL, R., GEHRKE, J., GUNOPULOS, D., AND RAGHAVAN, P. 1998. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the ACM-SIGMOD International Conference on Management of Data*, 94–105.

BACCINO, T. AND COLOMBI, T. 2001. L'analyse du mouvement des yeux sur le web. *Les Interactions Homme-Systéme: Perspectives et Recherches Psycho-Ergonomiques*, 127–148.

BANKS, A. S. AND ANDERSON, S. 1991. Peripheral spatial vision: Limits imposed by optics, photoreceptors and receptor pooling. *J. Optical Soc. Amer.*, 1775–1787.

CHAUDRI, A. B., UNLAND, R., DJERABA, C., AND LINDNER, W. eds. 2002. *Proceedings of the International Conference on Extending Database Technology XML-Based Data Management and Multimedia Engineering (EDBT)*. Lecture Notes in Computer Science, vol. 2490. Springer, Berlin.

C.J. ERKELENS, I. V. 1975. The initial direction and landing position of saccades. *Eye Movement Res. Mechan. Proc. Appl.*, 133–144.

COWEN, L., BALL, L., AND DELIN, J. 2002. An eye-Movement analysis of web-page usability. In *Proceedings of the Conference on Human-Computer Interaction (HCI)*, 317–335.

CROSSLAND, M. D. AND RUBIN, G. S. 2002. The use of an infrared eyetracker to measure fixation stability. *Optom. Vision Sci.*, 735–739.

DASGUPTA, S. AND GUPTA, A. 1999. An elementary proof of the Johnson-Lindenstrauss lemma. Tech. Rep. TR-99-006, International Computer Science Institute.

DJERABA, C., ed. 2003. *Multimedia Mining - A Highway to Intelligent Multimedia Documents*. Kluwer, Boston.

DJERABA, C., LEW, S., SIMOVICI, D. A., MONGY, S., AND IHADDADENE, N. 2006. Eye/Gaze tracking in web, image and video documents. In *Proceedings of the 14th ACM International Conference on Multimedia*, Santa Barbara, CA, 23–27.

FRANKL, P. AND MAEHARA, H. 1988. The Johnson-Lindenstrauss lemma and the sphericity of some graphs. *J. Comb. Theory B 44*, 355–362.

GOLDBERG, J. H. AND SCHRYVER, J. P. 1995. Eye-Gaze contingent control of the computer interface: Methodology and example for zoom detection. *Behav. Res. Meth. Instrum. Comput.*, 338–350.

GOLDBERG J. H., W. A. M. 2003. Eye tracking in usability evaluation: A practitioner's guide. In *The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research*.

GOLDSTEIN, R.B., RUSSEL, L., AND WOODS, E. P. 2006. Where people look when watching movies: Do all viewers look at the same place? *Comput. Biol. Med.*

GUBA, E. AND WOLF, S. D. G. 1964. Eye movements and tv viewing in children. *Audio-Visual Commun. Rev.*, 386–401.

GULLIVER, S. R. AND GHINEA, G. 2004. Stars in their eyes: What eye-tracking reveals about multimedia perceptual quality. *IEEE Trans. Syst. Man Cybernet. 34*, 4.

JACOB, R. J. AND KARN, K. S. 2004. Icommentary on Section 4. Eye tracking in human-computer interaction and usability research: Ready to deliver the promises. In *The Mind's Eyes: Cognitive and Applied Aspects of Eye Movements.* Elsevier Science., Oxford, UK.

JAIN, A. K. AND DUBES, R. 1988. *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, NJ.

JAIN, A. K. AND FLYNN, P. J. 1996. Image segmentation using clustering. In *Advances in Image Understanding: A Festschrift for Azriel Rosenfeld*. IEEE Press, Piscataway, NJ, 65–83.

JAIN, A. K., MURTY, M. N., AND FLYNN, P. J. 1999. Data clustering: A review. *ACM Comput. Surv. 31*, 264–323.

JOHNSON, W. B. AND LINDENSTRAUSS, J. 1984. Extensions of Lipshitz mappings into Hilbert spaces. *Contemp. Math. 26*, 189–206.

KARSH, R. AND BREITENBACH, F. W. 1983. Looking at looking: The amorphous fixation measure. *Eye Movement Psychol. Functions: Inter. Views*, 53–54.

LANKFORD, C. 2000. Gazetracker a software designed to facilitate eye movement analysis. *Eye Tracking Res. Appl.*, 51–55.

POOLE, A., BALL, L., AND PHILLIPS, P. 2004. In search of salience: A response time and eye movement analysis of bookmark recognition. In *Proceedings of the Conference on Human-Computer Interaction (HCI)*, 19–26.

RAYNER, K. 1998. *Eye Movements and Information Processing: 20 years of Research*. Eyrolles.

ROSS, J., AND MORRONE, M., AND BURR, D. C. 1994. Changes in visual perception at the time of saccades. *Nature*, 511–513.

SALVUCCI, D. D. AND GOLBERG, J. H. 2000. Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the Eye Tracking Reseach and Applications Symposium*, 71–78.

SANTELLA, A. AND DECARLO, D. 2004. Robust clustering of eye movement recording for quantification of visual interest. In *Proceedings of the Eye Tracking Research and Applications Symposium (ETRA)*.

TAN, P. N., STEINBACH, M., AND KUMAR, V. 2006. *Introduction to Data Mining*. Pearson/Addison-Wesley, Boston.

V. TOSI, L. MECACCI, E. P. 1992. Scanning eye movements made when viewing film: Preliminary observations. *Inter. J. Neurosci.* 47–52.

VEMPALA, S. S. 2004. *The Random Projection Method*. American Mathematical Society, Providence, RI.

WIDDEL, H. 1984. Operational problems in analysing eye movements. *Theor. Appl. Aspects Eye Movement Res.*, 21–29.

ZAÏANE, O. R., SIMOFF, S., AND DJERABA, C., eds. 2002. *Mining Multimedia and Complex Data*. Lecture Notes in Artificial Intelligence, vol. 2797. Springer, Berlin.

Received August 2007; accepted August 2007