Gaze Movement-driven Random Forests for Query Clustering in Automatic Video Annotation

Stefanos Vrochidis[1], Ioannis Patras[2], Ioannis Kompatsiaris[1]

Affiliations

1: Centre for Research and Technology Hellas - Information Technologies Institute, Thessaloniki, Greece {stefanos@iti.gr, ikom@iti.gr}

2: Queen Mary, University of London, London, UK {I.Patras@eecs.qmul.ac.uk}

Corresponding Author Stefanos Vrochidis {stefanos@iti.gr}, Centre for Research and Technology Hellas - Information Technologies Institute, Thessaloniki, Greece, +30 2311 257754.

ABSTRACT

In the recent years, the rapid increase of the volume of multimedia content has led to the development of several automatic annotation approaches. In parallel, the high availability of large amounts of user interaction data, revealed the need for developing automatic annotation techniques that exploit the implicit user feedback during interactive multimedia retrieval tasks. In this context, this paper proposes a method for automatic video annotation by exploiting implicit user feedback during interactive video retrieval, as this is expressed with gaze movements, mouse clicks and queries submitted to a content-based video search engine. We exploit this interaction data to represent video shots with feature vectors based on aggregated gaze movements. This information is used to train a classifier that can identify shots of interest for new users. Subsequently, we propose a framework that during testing: a) identifies topics (expressed by query clusters), for which new users are searching for, based on a novel clustering algorithm and b) associates multimedia data (i.e. video shots) to the identified topics using supervised classification. The novel clustering algorithm is based on random forests and is driven by two factors: first, by the distance measures between different sets of queries and second by the homogeneity of the shots viewed during each query cluster defined by the clustering procedure; this homogeneity is inferred from the performance of the gaze-based classifier on these shots. The evaluation shows that the use of aggregated gaze data can be exploited for video annotation purposes.

Keywords

Implicit feedback, eye-tracking, interactive video retrieval, clustering, random forests.

1. INTRODUCTION

In the previous decades, manual annotation of multimedia was one of the usual practices to support video retrieval. However, in the recent years, the rapid increase of the volume of the available content has made such practices very costly in terms of human effort. To this end, several automatic annotation approaches have been proposed. Most of them are based on content-based analysis and therefore they face the well-known problem of semantic gap (i.e. the inability of translating the low-level features to human understandable high level concepts). More recently, the high availability of sensors and the fast communication networks (including social media), which allowed for the generation of large amounts of user interaction data, directed the research trends (e.g. [1], [2]) to move towards exploiting the implicit user feedback during interactive multimedia retrieval tasks for annotation purposes.

In this context, several multimedia retrieval and recommendation systems have built upon the idea of linking queries submitted by the users with items that are subsequently selected or viewed and therefore they can be considered of interest to the user with respect to this specific query. For instance, if a user, who performs interactive video search, views a video directly after she has submitted the query "car", it is highly possible that this video contains a car and could be annotated using this tag. Towards to exploiting this fact, several research works rely upon graph-based representations, in order to link viewed or clicked items with queries ([1], [3], [4]), while others focus on identifying user interest by considering gaze movements ([5], [6]) and predefined (known) topics.

In this paper we deal with the problem of automatically annotating video content in interactive video retrieval tasks, during which, several different users are searching for unknown topics. Inspired by the aforementioned works, we attempt to identify the topics, for which the users are searching for, as well as to detect the items that were of interest for these users in the context of these topics. To this end we employ a supervised framework, which collects and

aggregates the implicit feedback (i.e. gaze fixations and queries) of users (in off-line mode), when they search for similar topics. During training, the aggregated gaze movements of these users are processed, in order to extract gaze fixations (i.e. spatially stable gaze) and generate a set of features that describes each video shot. Subsequently, we train a classifier that could predict which of the items viewed by a new user (i.e. different from the ones involved in the training) could be classified as interesting and match the topic she searches for. During testing we assume that we have no knowledge about the topics that new users search for. To identify the unknown topics, we apply a novel unsupervised hybrid random forest clustering algorithm.

Contrary to the typically used random forest clustering algorithm (e.g. [27]), which constructs the trees of the forest based solely on an affinity matrix, the proposed hybrid random forest clustering algorithm is driven by two factors: a) an affinity matrix including textual and visual distances between groups (i.e. clusters) of queries, and b) by the performance of the classifier (which depends on the gaze movements). After the query clusters have been defined, the positive results of the classifier (i.e. shots of interest for a given topic) are associated with the cluster labels, which derive from the queries and describe the topic, annotating in that way the content.

We evaluate this approach by performing a video retrieval experiment, in which users are retrieving video content with an interactive video search engine, while their gaze movements are captured with the aid of an eye tracker and their click-throughs and query submissions are recorded in log files.

The research contributions of this work are: a) the methodology of automatic annotation during interactive video retrieval tasks by using gaze features and b) the query clustering algorithm based on hybrid unsupervised random forests, which are driven by textual and visual query distances, as well as indirectly by the gaze movements;

This paper is structured as follows: section 2 presents the related work and section 3 provides background knowledge about unsupervised random forests. The problem description is provided in section 4. Section 5 presents the video annotation framework, while the gaze-movement classification is described in section 6. Then, the hybrid random forests for query clustering are described in section 7. The video retrieval search engine is presented in section 8, while the experiment and the results are reported in sections 9 and 10 respectively. Finally, section 11 concludes the paper.

2. RELATED WORK

Research on multimedia annotation and retrieval up to now has been based mostly on content characteristics. Typical supervised methods such as those used in TRECVID systems/submissions use supervised methods (e.g. Support Vector Machines (SVMs)) in order to learn a pre-selected set of concepts and events (e.g. [7]). Other efforts on supervised annotation have focused on correlative tagging, which exploits annotation co-occurrences in the labeling process [8] and active learning [9]. In another approach [37] a multiple-instance neural network (MINN) for content-based image retrieval (CBIR) is trained to learn the user's preferred image concept from the positive and negative examples. With respect to video annotation, several works deal with action detection and recognition, which has many applications including gait recognition, group activity analysis, abnormal activity detection, and sport activity recognition [43]. Recently, several works propose multiview characters and manifold information to deal with human action recognition [44]. In this context, the authors in [43] propose a framework for activity recognition based on multiview Hessian regularized logistic regression, while in [42] the application of Hessian-regularized co-training for social activity recognition in videos is proposed. Such approaches have been applied in a variety of multimedia content such as personal videos, movies and TV news [38]. In other works, the content-based approaches were enhanced by considering implicit user feedback. For instance in [2], automatic image annotation is proposed by training concept classifiers with labeled example images, which result from click-through data, while in [10], the authors attempt to find neuro-physiological indicators to validate tags attached to video content. Given the fact that the proposed work exploits mainly the implicit user feedback for performing annotation, we will further discuss the research works dealing with such problems.

In general case information retrieval tasks, the implicit user feedback can be divided into two main categories: the query actions and the physical involuntarily user reactions. The first category includes the patterns of user interaction with the search engine, as series of mouse movements and clicks, shot selections, key strokes and keyboard inputs, while the second includes physical user unconscious behavior as eye movements (e.g. [11]), heart rate and brain neuron reactions that can be gathered with electroencephalography (e.g. [10]). On the one hand, the feedback of the first category can be easily gathered even during a web search session, while physical reactions can be recorded with the aid of special wearable devices or other sensors (e.g. cameras) capturing and analyzing user behavior.

Implicit user feedback techniques have not been fully explored in the multimedia domain. In text retrieval, the usual implicit information that can be taken into account is the user selection (i.e. the user clicks on an interesting link or

textual description to view the complete document), while in video retrieval we have multiple interactions between the user and the system, which could be utilized to provide meaningful feedback. The main idea to exploit the user feedback during video retrieval interactive sessions is to extend the idea of "query chains" [12] and construct a graph that describes a series of user actions.

Such a graph is transformed to a weighted graph by aggregating the links between the same nodes and weights are introduced based on the different actions taken into account. Recent works [4], [1] employ the aforementioned technique to deal with user clicks. Specifically, in [4] the authors propose to use community based feedback mined from the interactions of previous users of a video retrieval system to aid users in their search tasks. In [1] the construction of click-graphs based on smaller search sessions (sub-sessions) is proposed, and it is used to provide recommendations and improve the results of visual search. Finally, in [13], relevance feedback and multimodal fusion of textual, visual and click-through data is proposed, in order to generate recommendations for the user. These works either represent past user click-through data with graphs or use them to adjust fusion weight, however generation of click-based features is not attempted and gaze data are not considered.

As far as the physical involuntarily reactions of the users are concerned, the gaze movements are considered as one of the most important indicator of user interest. Specifically, research conducted in the last three decades showed that eye movement data have proven to be very valuable in studying information processing tasks [14].

The first works that employed gaze analysis in the area of Information Retrieval (IR) have focused on textual document search. In one of the first attempts to study eye movements during IR tasks [15], the authors investigate how the users interact with the results of a web search engine by employing eye-tracking techniques. A very interesting approach is described in [16], in which proactive information retrieval is proposed by combining implicit relevance feedback and collaborative filtering. More specifically, implicit feedback is inferred from eye movements, with discriminative Hidden Markov Models estimated from data, for which explicit relevance feedback is available. In a more recent work [17], the authors introduce a search strategy, in which a query is inferred from information extracted either from eye movements measured when the user is reading text during an IR task or from a combination of eye movements and explicit relevance feedback.

The first applications of eye-tracking in image and video retrieval were in the area of studying the user behavior and evaluating visual interface representations (e.g. [18]). More recent works in image and video retrieval deal with deriving user interest based on eye movements and also utilize this technique to develop gaze-based interactive interfaces. In [19] the real time interface GaZIR for searching images is proposed. In this case, the relevance of the viewed images is predicted using fixation and saccade-based features. In another work [5], the authors propose a nine-feature vector from fixations and saccades and use a classifier to predict one relevant image from four candidates.

Recently, approaches for performing relevance feedback based on eye features are proposed in [11] and [20], while a gaze-based relevance feedback approach for region-based image search is presented in [21]. In [6] the authors propose the generation of recommendations based on a SVM classifier trained with fixation and pupil dilation based features, while [22] extends this work by performing query clustering based on dominant sets. Other recent works in image retrieval attempt to combine image features with eye movements, either by using a ranking SVM approach [23], or by identifying areas of interest in an image to extract local visual features [24]. In [25] a content-based image retrieval system is presented, which exploits both pointer clicks and eye movements; this system learns a similarity metric between images, which depends on the current interests of the user and it then retrieves images with a specialized online learning algorithm that balances the tradeoff between exploring new images and exploiting the already inferred interests of the user.

These works focus on more controlled retrieval scenarios and do not deal with predicting the relevance of results for a new query by a new user [5], they do not consider unknown search topics [6] and they do not incorporate gaze movements to support query clustering [22]. Also, none of them considers combination of gaze movements and click-through data for video retrieval (e.g. [20], [21], [25] focus on image search).

The proposed approach differentiates from these works by introducing a novel video annotation framework, which builds upon gaze movement driven random forest for query clustering that combines gaze movement data and click-throughs.

3. RANDOM FORESTS

Random Forest (RF) [26] is an ensemble of several decision trees. It has been used with great success for several machine learning problems, such as classification, clustering and regression.

3.1 Construction of random forests

A RF may contain hundreds or thousands of trees depending on the application and the dataset. Let us assume that we want to build a forest with T trees. The following algorithm is used to construct each tree of the forest: we assume that we have N training examples. Each of them is represented in the multidimensional space by a vector $v = (x_1, \dots, x_d) \in \mathbb{R}^d$, where d is the cardinality of the features. Each training example is associated with one of the k classes $\{c_1, \dots, c_k\}$. To construct the decision tree, we take a *bootstrap sample*, which is used as a training set to grow each tree. This is performed by sampling n samples with replacement (i.e. by putting back in the collection the selected sample) from all the N available training cases.

In order to grow each tree, we set a number of $m < d$ random variables, which correspond to the m of the d dimensions that are evaluated to find the best split. The latter is chosen by maximizing an impurity function. Each tree is fully grown and not pruned. After the RF has been constructed, we calculate the probability that a new sample x belongs to class c as:

$$p(c|x) = \frac{1}{T}\sum_{1}^{T} p_t(c|x) \tag{1}$$

where $p_t(c|y)$ is the probability that the sample y belongs to class c as calculated by tree t. $p_t(c|x)$ can be calculated either by using the absolute class frequencies at every leaf or by applying more sophisticated approaches such as smoothing, geometric ranking and density estimation at the leafs.

3.2 Impurity Function

The impurity function measures the purity for a region containing data points that possibly belong to different classes. Let us assume that the number of classes is K. Then, the impurity function is a function of $p_1, .., p_i, \dots p_K$, where p_i is the probability for any data point in the region belonging to class i. During training, we are not aware of the real probabilities. However, an acceptable compromise is to associate these probabilities with the percentage of points that belongs to each class in the region we are interested in exploiting the labels of the training data set.

Formally, an **impurity function** Φ is defined on the set of all K-tuples of numbers $(p_1, \dots p_K)$ satisfying $p_j \geq 0$, $j = 1, \dots, K$, $\sum_1^K p_j = 1$ and has the following properties: a) it achieves maximum only for the uniform distribution of p_j, which means that is all the p_j are equal, b) it achieves minimum when the probability to belong in a certain class is 1 and 0 for all the other classes and c) it is a symmetric function of p_j.

Given an impurity function Φ, we define the impurity measure, denoted as $i(t)$, of a node t as follows:

$$i(t) = \Phi(p(1|t), p(2|t), \dots, p(K|t)) \tag{2}$$

where $p(j|t)$ is the estimated posterior probability of class j given a point is in node t. This is called the impurity function (or the impurity measure) for node t. Once we have defined $i(t)$, we can estimate the goodness of split s for node t, denoted by as $\Delta i(s,t)$:

$$\Delta i(s,t) = i(t) - w_R i(t_R) - w_L i(t_L) \tag{3}$$

$\Delta i(s,t)$ represents the difference between the impurity measure for node t and the weighted sum of the impurity measures for the right child and the left child nodes. The weights, w_R and w_L , are the proportions of the samples in node t that go to the right node t_R and the left node t_L respectively.

3.3 Unsupervised Random Forests

Many supervised methods can be converted into unsupervised using the following idea. An artificial class is created, which distinguishes the observed data from suitably generated synthetic data. The supervised learning methods that attempt to distinguish observed from synthetic data, yield a dissimilarity measure that can be used as input in subsequent unsupervised learning methods. In [27] it is proposed to use RF predictors to distinguish observed from synthetic data.

Since an individual tree is unpruned in RF, the terminal nodes will contain only a small number of observations. The training data are run down each tree. In case two observations i and j end up to the same terminal node, the similarity between i and j is increased by one. After the forest is finalised, the similarities are normalised and divided by the number of trees. The similarities between objects form a symmetric matrix, which is positive definite, and each entry lies in the unit interval $[0\ 1]$. The RF dissimilarity is mathematically defined as:

$$DS_{ij} = \sqrt{1 - SM_{ij}} \quad (4)$$

where SM_{ij} stands for the similarity between i and j.

4. PROBLEM DESCRIPTION

During a common interactive video retrieval scenario many users search for several different topics using the same search engine (e.g. YouTube[1]). As discussed in the introduction, many annotation systems have exploited the idea of linking queries submitted by the user with items that are selected or viewed and therefore they are of interest to the user in the context of a specific topic. In order to generate such links two important parts need to be defined: a) the topic that the user has in her mind and b) the items that are of interest. Due to the fact that our purpose is video annotation we aim at linking items of interest to topics submitted by many different users. Therefore two problems have to be solved by considering aggregated user data: a) the topic identification and b) the detection of interesting items for the users.

4.1 Topic Identification

In most of the cases the queries submitted can give a good idea of the query topic the user searches for. However, the user usually submits specific keywords and not the whole topic itself. For instance a user might be looking for "a red car in motion" but she submits as query only the keyword "car". In addition, the user might submit irrelevant queries (e.g. due to an external unpredicted distraction) to the search topic, which might further complicate the situation. In many cases new users search for a topic already addressed by past users. The aim is to identify query groups submitted by different users that correspond to topics.

We consider a set of N search sessions $S = \{S_1, \dots S_i, \dots S_N\}$. During a session S_i the user i is searching for a specific topic t_i and submits M_i queries $\{Q_1, \dots Q_{M_i}\}$. We assume that the user can submit textual, visual and temporal queries. The actual goal in this case is to group the semantically relevant queries into topics regardless of the user, who search and the time, at which they were submitted. The output of this task will be K query clusters $\{c_1, \dots c_K\}$ that correspond to topics. Each cluster c_j corresponds to a topic and is consists of a set of queries. This cluster c_x can be described by the most frequent queries, which are defined as the topic labels $\{l_{x,1}, \dots l_{x,Z}\}$, where Z is the maximum number of labels per topic.

4.2 Detection of User Interest

On the other hand, the items of interest can be revealed by specific voluntarily and involuntarily actions made by the user. One characteristic indicator for this is the mouse clicks (e.g. [1], [2]), while additional information can be provided by sensors such as gaze trackers (e.g. [5], [6]) or biometric devices.

After the query clusters are formed, we need to identify items that are interesting for the users in the context of these topics. Assuming that we have identified K query clusters $\{c_1, \dots c_K\}$ (section 4.1), the aim is to recognize all the video shots v_y, which were of interest for the users when submitted queries that belong to cluster c_x. This will allow for linking the shot a_y, with cluster c_x and assigning $\{l_x\}$ to shot a_y.

5. VIDEO RETRIEVAL AND ANNOTATION FRAMEWORK

We consider a video retrieval and annotation framework (Figure 1), which deals with the aforementioned problems. This framework builds upon supervised learning (classification) for identifying shots that are of interest to the user (detection of user interest problem). During testing, the framework also includes an unsupervised learning phase (clustering), in order to identify the query clusters that correspond to topics (topic detection problem).

The training phase aims at constructing a classifier that could classify shots based on user interest. The shots are described by processing aggregated gaze movement data from many users, who search for the same topic or query cluster. During this phase, we assume we have explicit knowledge of the query topics that the users are searching for, how much time they search for each topic and the queries they submit (e.g. for every topic a user could submit several queries), as well as the results they are interested in.

In this phase, the gaze movements of the users searching for the same topic are collected (User Gaze Data) and merged. Then, gaze-based features for each video shot are extracted and aggregated along the same topic (topic-based merging). In the following, we use the gaze features and the results for each topic submitted by the users as relevance of each shot to a topic, in order to train a Support Vector Machine (SVM) classifier that could classify as

[1] https://www.youtube.com/

relevant or non relevant the items viewed by new users (Gaze movement-based shot classification). Since this classifier is trained only with gaze movements, it can be considered as predictor of user interest for a certain viewed item in the context of a query topic. It should be clarified that the developed classifier aims at distinguishing between interesting and non interesting shots for a user. This means that the classifier is topic and user independent.

In the testing phase, we assume that we have no knowledge regarding the query topics (i.e. topic subject, time boundaries of the search sessions). To identify the unknown topics during testing, we first perform query clustering using the unsupervised random forests taking into account the WordNet similarity between textual queries, the visual similarity between clicked keyframes and the homogeneity of the shots of interest for each calculated query cluster (Query clustering using Hybrid Random Forests). This homogeneity is measured based on the output of the SVM classifier, which is driven by the gaze movements. Then, we generate gaze-based features for each shot by merging the gaze movement data from different users along the same query clusters (Cluster-based merging) and employ the classifier to predict the relevance of the shots with respect to the query clusters. The positive results are associated with the cluster labels, annotating in that way the video shots.

Therefore the main two components of the framework are: a) the gaze-based shot classification, which deals with the problem of detecting items of user interest (section 6), and b) the hybrid unsupervised random forests, which are employed during the testing phase to identify query clusters and deal with the query clustering problem (section 7). These are presented in the following sections.

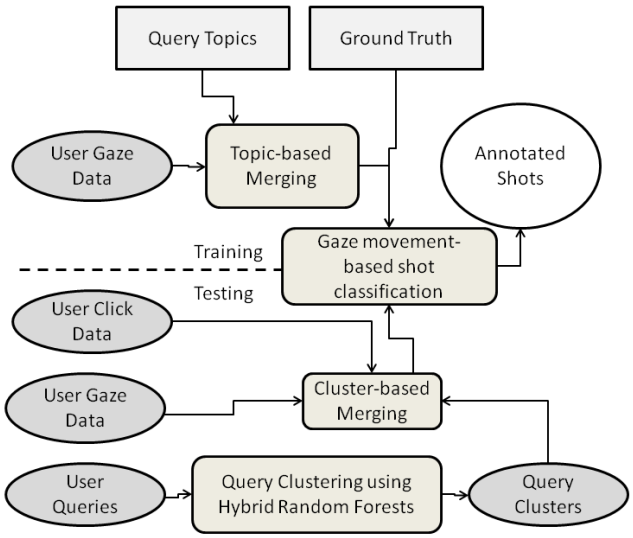

Figure 1. Video annotation framework.

6. GAZE MOVEMENT-BASED SHOT CLASSIFICATION

Generally, eye movements can be characterized according to the following ocular behaviors: fixations, saccades and scan paths [14]. Fixations are defined as a spatially stable gaze lasting at least 100 milliseconds, during which visual attention is directed to a specific area of the visual interface. The eye fixations could be considered as the most relevant and reliable indicator for evaluating information processing during an online video search. On the other hand, saccades, which are the continuous and rapid movements of eye gazes between fixation points, are believed to occur so quickly across the stable visual stimulus that only a blur would be perceived.

Figure 2. Fixations during video search

During a video retrieval session the user interacts with a visual interface illustrating several videos. Apparently, the user focuses her gaze on the items that are of interest with respect to what she searches for. However, it seems that despite the fact that many parts of the graphical interface are viewed constantly for a specific amount of time (i.e. a fixation point was identified) during a video search session, not all of them could be considered as items of interest. This is made clear in the example of Figure 2, in which a user is searching for video scenes that depict books. After the analysis of the gaze movements, many fixations are identified, pointing at different parts of the interface. It is obvious that many fixations on relevant items are reported (e.g. shots on the top left corner of the interface), however it is also clear that some of the video shots that draw the attention of the user (as shown by the fixations) are not relevant to the query (e.g. the shot on the top right corner of the interface). Therefore, the simplistic approach of labeling as interesting the shots with the longest fixations would not be very effective. In order to be able to discriminate between relevant and irrelevant items to a query topic, we need to analyze the characteristics of the fixations and identify correlations between the fixation frequency, duration and the user interest depicted in our case by the search topic.

Based on previous studies, fixation-based features have shown discrimination power over items of interest for users both in image ([5], [11]) and video search [6].

Table 1. Gaze-based features

#	Feature description	Mathematical Definition
1	Total number of Fixations for shot a	$F_a = \frac{\sum_{S_{j,k} \in Y} F_{a,S_{j,k}}}{L \cdot K}$
2	Total fixation time for shot a	$T_a = \frac{\sum_{S_{j,k} \in Y} T_{a,S_{j,k}}}{L \cdot K}$
3	Average fixation time for shot a	$A_a = \frac{T_a}{F_a} = \frac{\sum_{S_{j,k} \in Y} T_{a,S_{j,k}}}{\sum_{S_{j,k} \in Y} N_{a,S_{j,k}}}$
4	Average fixations for shot a per search session	$V_a = \frac{F_a}{\sum_{S_{j,k} \in Y} t_{S_{j,k}}} = \frac{\sum_{S_{j,k} \in Y} F_{a,S_{j,k}}}{\sum_{S_{j,k} \in Y} t_{S_{j,k}}}$
5	Average fixation time for shot a per search session	$M_a = \frac{T_a}{\sum_{S_{j,k} \in Y} t_{S_{j,k}}} = \frac{\sum_{S_{j,k} \in Y} T_{a,S_{j,k}}}{\sum_{S_{j,k} \in Y} t_{S_{j,k}}}$

Here, we employ the fixation-based features, which were used in [6] and shown discrimination power between relevant and irrelevant shots in the context of a specific query. These features describe the way that a representative keyframe of a video shot was viewed (i.e. an aggregated view) by several users in the video search engine interface during the retrieval tasks. We define as search session $S_{j,k}$ the time period, during which, user j is searching for a specific topic k. We assume that each search session $S_{j,k}$ lasts $t_{S_{j,k}}$ time. We declare as $F_{\alpha,S_{j,k}}$ the total number of fixations and $T_{\alpha,S_{j,k}}$ the total fixation duration time that were reported for a shot a during a search session $S_{j,k}$. Let us

assume that we want to describe a shot a with information retrieved during a set of sessions $Y = \{S_{j,k}\}$, where $j, k \in \mathbb{N}, 0 < j \leq L, 0 < k \leq K$, where L is the number of different users and K the number of topics in these sessions. Given the fact that the gaze input could be aggregated by many users, the features are normalized against the number of search sessions $L \times K$. The features are described in Table 1. Hence, the final feature vector for shot a is:

$$f_a = [F_{a,}, T_{a,} A_a, V_a, M_a] \quad (5)$$

Then, we employ a SVM to classify the viewed items according to the user interest exploiting this feature vector. We make use of the LIBSVM library [28] and we consider a binary C-Support Vector Classification, using as kernel the radial basis function:

$$K(f_i, f_j) = e^{-\gamma |f_a - f_j|^2} \quad (6)$$

where g is a constant value, f_a the feature vector of a shot and f_j the support vectors. The created model will be able to identify, which shots are of interest to new users in the context of new topics. In case the topics were known both in the train and testing phase, this step would have been adequate to annotate shots with topic information as it is proposed in [6]. However, since in this problem the topics are considered unknown during the testing phase, an additional step of query clustering is required.

7. QUERY CLUSTERING USING HYBRID RANDOM FORESTS

The plethora of the user interaction data and search logs motivated several works to focus on query clustering, in order to define broader semantic query groups (i.e. topics). Most of these approaches perform query clustering by computing similarity metrics between query pairs. When dealing with query classification (i.e. predefined categories), it may be sufficient to compute the similarity based only on textual features to obtain good results [29]. However, if predefined categories are not available, lexical and content-based information taken separately are not sufficient to obtain good clusters. In this context, an attempt to cluster queries from user logs [30] showed that query-to-query similarity metrics that combine textual features with click-through data can be used much more profitably in query clustering than single-attribute similarities, while in [22] the temporal dimension is used to enhance dominant-set text-based query clustering in video retrieval. Inspired by such approaches, this work proposes to enhance textual and content-based query clustering by considering the user gaze movements.

Instead of performing a direct query clustering (i.e. by comparing queries directly) we propose to cluster the queries at subsession level. Following the definition of [1] a subsession is defined by one *autonomous* query (i.e. doesn't depend on previous results) and could include several *dependant* queries (depend on previous results). Let us provide an example to further clarify the definition of autonomous and dependant queries. In such a case, the user wants to retrieve video shots that depict "musicians playing the guitar". Therefore she submits a textual query "guitar" to the system. This is an autonomous query, since it doesn't depend on any previous results the user has seen. The system provides several results and the user selects one of them to perform visual search and retrieve visually similar shots. This query is considered dependant, since it is strongly related to the previous results. Therefore the subsession can be considered as a small group of queries addressing the same topic. The advantages of dealing with subsessions instead of single queries are that: a) we limit the number of items to be clustered and b) we can derive more descriptive features compared to a single query, since a subsession is formed by a group of queries. Examples of subsessions are provided below in Figure 3. In this work, we consider only the textual queries as autonomous, while visual and temporal queries (which use a keyframe as an input) are dependent. These definitions depend on the type of the search engine (section 8) employed. For instance if a different search engine was used, visual queries that would allow the user to upload a new image as paradigm would have been also defined as autonomous.

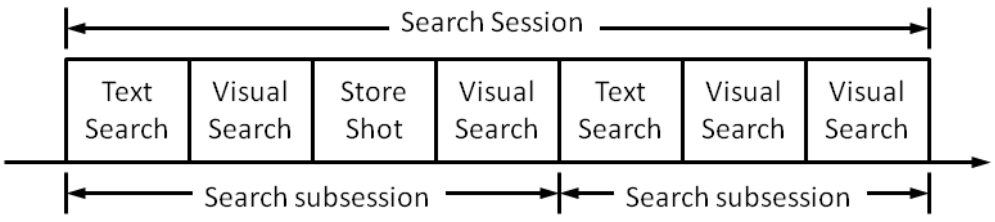

Figure 3: A search session split into subsessions

Our goal is to group the subsessions (and therefore the queries included in them) in such a way, so that they form specific topics. The shots that are viewed during the submission of the queries that belong to the same cluster will be

classified as relevant and non-relevant for this query cluster by the SVM classifier based on the aggregated gaze movement-based features (section 6). Based on the fact that the more gaze data (i.e. the more different users) we aggregate for each shot, the best results are achieved with respect to the user interest [6], we aim at clustering the queries into topics regardless of the temporal information assuming that searches for specific topics will be repeated from time to time. This means that if the same topic is repeated more than once (by the same or another user), the queries that are submitted should be clustered into the same group. We also make the assumption that when the cluster converges to a topic, the best shot separation is achieved by the classifier (section 6). This assumption is based on the fact that if we aggregated gaze data from shots that were viewed along different topics, the separation by the classifier will not be optimal, since contradictory information will be merged. For example let us assume that "image 1" is relevant to topic A and irrelevant to topic B. During the interactive retrieval tasks for these topics different users have viewed "image 1". If we aggregate the gaze movement features during the time that users were searching for topic A (e.g. high number of fixations, etc.), "image 1" will be classified as relevant. If we merge these features with the gaze movement features extracted during topic B search (e.g. low number of fixations, etc) the aggregated features will result in average values and therefore "image 1" will be probably positioned very close to the SVM hyperplane.

Therefore the performance of the gaze classifier can be considered as a homogeneity measure for the query clusters that have been defined. In order to incorporate the result of the classifier into the clustering process, we propose to use and adapt the unsupervised random forest algorithm by introducing additional criteria during the decision tree construction. We select the random forest algorithm because it allows for incorporating the homogeneity measure in the splitting criterion of the decision trees (Section 7.2), without the need of making the algorithm converging (as it would have been the case for instance when using K-means). To this end we perform hybrid RF query clustering, by considering two factors: a) the distances between the subsessions and b) the level of interest of the shots that were viewed during these subsessions, which is described by a homogeneity factor.

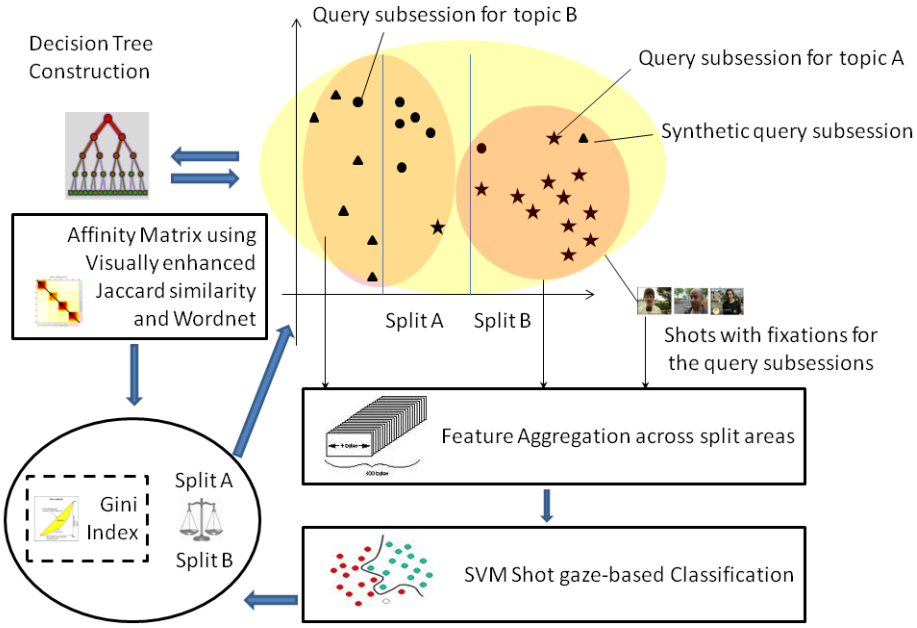

Figure 4. Decision tree construction in gaze movement driven random forests

In order to incorporate the homogeneity of the subsessions based on the gaze movements, we propose the framework in Figure 4 for decision tree construction. In the following, we describe the assumption we are based on and the decision tree construction through an example. To construct a decision tree, several splits are compared. Let's assume that subsessions for topics A and B are to be clustered. In this example we show the separation in feature k and two different splits (SplitA and SplitB) that are to be compared. The basis of this comparison is the purity of the sets created after the splits take place. Both SplitA and SplitB attempt to separate the observations (initial subsessions denoted with stars and circles) from the synthetic data (fake subsessions denoted with triangles). A normal purity criterion would have considered SpitA better compared to SplitB, since it better splits the observed

from the synthetic data. However, in this case SplitB should have been preferable, since it also separates topic A and B. Therefore, we need to introduce an additional criterion during the decision tree construction, which will perform a better split selection for our requirements.

As we have discussed in section 6, during each subsession several fixations are identified on the resulted shots. In order to incorporate also the preference of separating the observations that belong to different topics, we consider two different aggregations for each split (i.e. the split separates the subsessions in two groups and the gaze-movement features of the shots viewed during these subsessions are aggregated). Then, the quality of the classifier separation is incorporated into the splitting criterion (based on Gini Index) by introducing a homogeneity coefficient, which is discussed in detail in section 7.2. It should be noted that by using the Gini index, the algorithm attempts to separate the synthetic class from the observed values. By incorporating the homogeneity coefficient, we attempt also to separate the queries of different topics.

After the splitting is performed, we calculate the dissimilarities using (4). Then the RF dissimilarity is used as input of multi-dimensional scaling. Finally, we perform K-Means clustering using the output of the multi-dimensional scaling [31]. In the following, we discuss the construction of affinity matrix and the proposed splitting criterion for the decision tree construction for the unsupervised random forests.

7.1 Affinity Matrix

The affinity matrix M describes the relevance between subsessions. Let's define the semantic similarity between two subsessions A and B. The idea of this comparison is illustrated in Figure 5. Each subsession includes one autonomous query and a set of dependent queries. We calculate the semantic similarity between the two autonomous queries using the WordNet similarity as this is described in 7.1.1. On the other hand, the dependent queries consider keyframes as input and therefore each subsession includes a set of images that were clicked by the user. To calculate a distance between two sets of images we need to consider a metric that represents such a similarity (section 7.1.2).

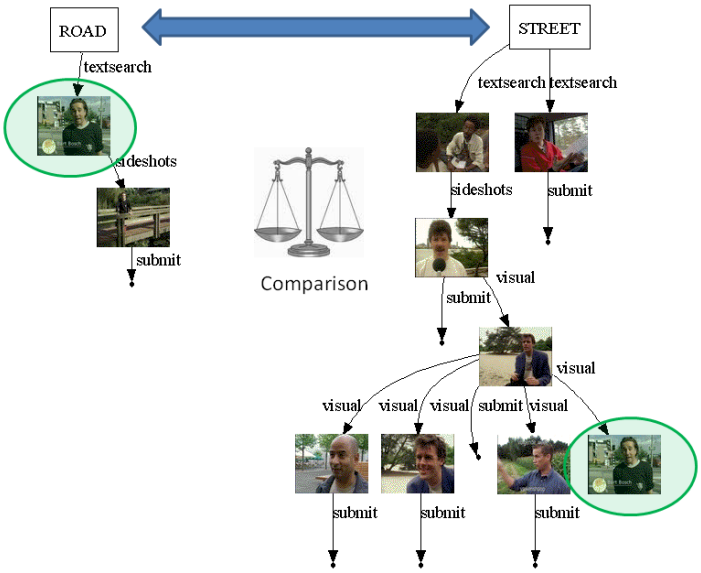

Figure 5. Comparison of subsessions. On the top the direct comparison between the textual queries is illustrated. The duplicate images identified in the dependent queries are shown in green circles.

Assuming that the similarity between the terms of the textual query is $v_{i,j}$ as described in section 7.1.1 and $eJ_{i,j}$ is similarity between sets of queries by visual example discussed in 7.1.2, the final similarity $w_{i,j}$ is defined as:

$$w_{i,j} = v_{i,j} + eJ_{i,j} \quad (7)$$

7.1.1 Textual queries similarity

One of the widely used techniques for comparing textual information is to use thesaurus as WordNet. In this approach, we applied the WordNet "vector" similarity. Each concept in WordNet is defined by a short gloss. The vector measure uses the text of that gloss as a unique representation for the underlying concept. The vector measure creates a co–occurrence matrix from a corpus made up of the WordNet glosses. Each content word used in a WordNet gloss has an associated context vector. Every gloss is represented by a gloss vector that is the average of all the context vectors of the words in the gloss. Relatedness between concepts is measured by finding the cosine between a pair of gloss vectors [32]. An additional problem in our case is the inability of dealing with term disambiguation (as the search topics are considered unknown). To overcome this problem we calculate the maximum similarity between the senses of the two textual queries, which is defined as $v_{i,j}$.

7.1.2 Visual queries similarity

A well known metric for set comparison is the Jaccard coefficient [33]. Formally, for two sets A and B, the Jaccard similarity coefficient $J(A,B)$ is given by:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{8}$$

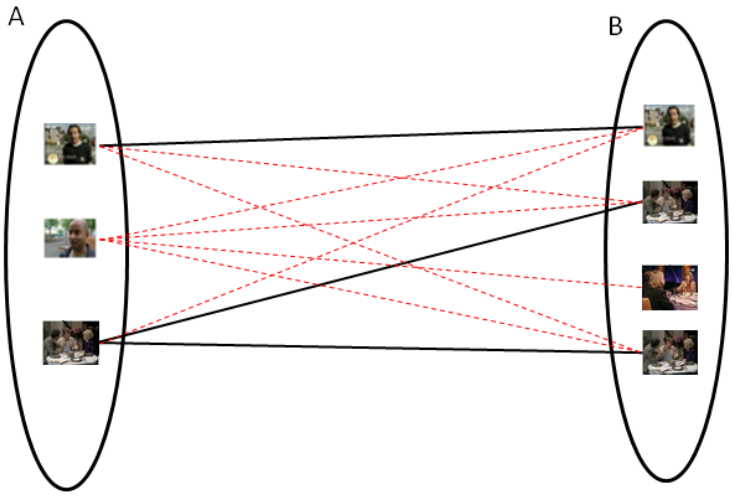

Figure 6. Sets A and B in a bipartite graph representation. Distances for non duplicate shots are represented with red dashed edges, while black solid edges indicate distances between near duplicates

However, when a set is comprised of images (keyframes) there are cases, in which several images are considered near duplicates. In this context we proposed to enhance the Jaccard similarity coefficient and introduce the *visually enhanced Jaccard similarity*, which takes into account near duplicates. The idea is to identify near duplicate images between the different sets and consider them identical in order to compute the Jaccard similarity. However, the problem is not that simple, since each image might have more than one near duplicates and a random selection would lead to different results. For instance let us assume set $A = \{a, b\}$ and $B = \{c, d\}$, where a is near duplicate with c and d, while b is near duplicate only with d. A random assignment would result to different similarity coefficients depending on the sequence we consider. In this case one assignment could be $a \equiv c$ and $b \equiv d$, which leads to Jaccard similarity equal to 1, while another assignment would be only $a \equiv d$ ($b \equiv d$ cannot be considered, since each image is allowed to have only one near duplicate), which leads to Jaccard similarity equal to 0.5. It is obvious that from such a case the most meaningful result is the first, while in the second case important parts of information are neglected.

Since the members of A are linked only with the ones of B, we can represent these connections by considering a bipartite graph (Figure 6). Then, we model the problem of identifying the maximum number of duplicates as a minimum weight perfect matching problem (or assignment problem) [34] in the bipartite graph. To this end we assign in each edge a cost $c = 0$, when the interconnected vertices represent duplicate images and $c = 1$ when the images are not considered duplicates. This is performed by considering a distance threshold T as shown below:

$$c_{i,j} = \begin{cases} 0 \; if \; c_{i,j} \leq T \\ 1 \; if \; c_{i,j} > T \end{cases} \quad (9)$$

This way, the problem is considered as a minimum weight matching, in which we want to identify a matching M, which minimises c. Then we transform the problem to a linear one by introducing dummy shots that do not have any near duplicate. To solve this problem we apply the Hungarian algorithm [35]. Finally, we compute the Jaccard similarity after we have identified the maximum number of assignments between the images of the different sets based on near duplicates. The algorithm for calculating the visually enhanced Jaccard similarity is presented in Table 2.

Table 2. Visually enhanced Jaccard similarity algorithm

Input: the two image sets $A = \{a_j\}$, $B = \{b_i\}$

1. Eliminate any duplicate images separately in A and B
2. Calculate all the visual distances $d_{i,j}$
3. Transform the problem to a linear one by removing or introducing dummy shots.
4. Apply the Hungarian Algorithm to identify the best matching
5. Update the two sets A and B to $\acute{A}$ and $\acute{B}$ respectively after the identification of near duplicates (i.e. if i and j are duplicates replace all j with i).
6. Calculate the Jaccard similarity of the two sets $\acute{A}$ and $\acute{B}$

Output = $eJ(A,B) = \frac{|\acute{A} \cap \acute{B}|}{|\acute{A} \cup \acute{B}|}$

7.2 Splitting criterion for decision tree construction based on the homogeneity co-efficient

In this section we define the splitting criterion we apply during the decision tree construction of the gaze-driven hybrid RF (Figure 4). We select the Gini Index, which is also used by Breiman for RF construction [26] as the basis of our impurity function:

$$G = \sum_{j=1}^{K} p_j(1 - p_j) = 1 - \sum_{j=1}^{K} (p_j)^2 \quad (10)$$

In order to incorporate the homogeneity of the samples that are clustered after the split, we introduce a new variable called the *homogeneity co-efficient*. This represents the homogeneity of a set of samples considering the user interest reflected by the aggregated gaze movements. It should be clarified that, while the Gini index is based on p_j, which is the probability of a sample belonging to the observed or the synthetic class, the homogeneity co-efficient depends on the p_i', which corresponds to the probability that a query belongs to topic i.

We calculate the homogeneity co-efficient by employing the gaze trained SVM model. Let's assume that we have M points in node t. It should be noted that these would include $K < M$ queries and $L = M - K$ vectors that belong to the synthetic data. For the queries that fall into the same split, we assume that they belong to the same cluster (topic) and we aggregate the gaze features for the S shots that resulted from these queries and for which, fixations have been identified. The output of the classifier provides as result the distance d_i between each shot i and the hyperplane. Then, the homogeneity co-efficient is calculated by considering these distances in a sigmoid function:

$$h = \frac{1}{1 - e^{-\frac{\sum_{i=1}^{S} |d_i|)}{S}}} \quad (11)$$

We incorporate the homogeneity coefficient in the impurity function of (10):

$$i(t) = \frac{1}{h}(1 - \sum_{j=1}^{K} (p_j)^2) \quad (12)$$

Given the fact that h is based on a sigmoid function it ranges in [0 1]. Finally, based on (3), (12) and that $K = 2$, since only two classes are considered (i.e. synthetic and observed data) the splitting criterion, which we need to maximize, is:

$$\Delta i(s,t) = \frac{1}{h}(1 - {p_{T1}}^2 - {p_{T2}}^2) - \frac{w_R}{h_R}(1 - {p_{TR1}}^2 - {p_{TR2}}^2) - \frac{w_L}{h_L}(1 - {p_{TL1}}^2 - {p_{TL2}}^2) \quad (13)$$

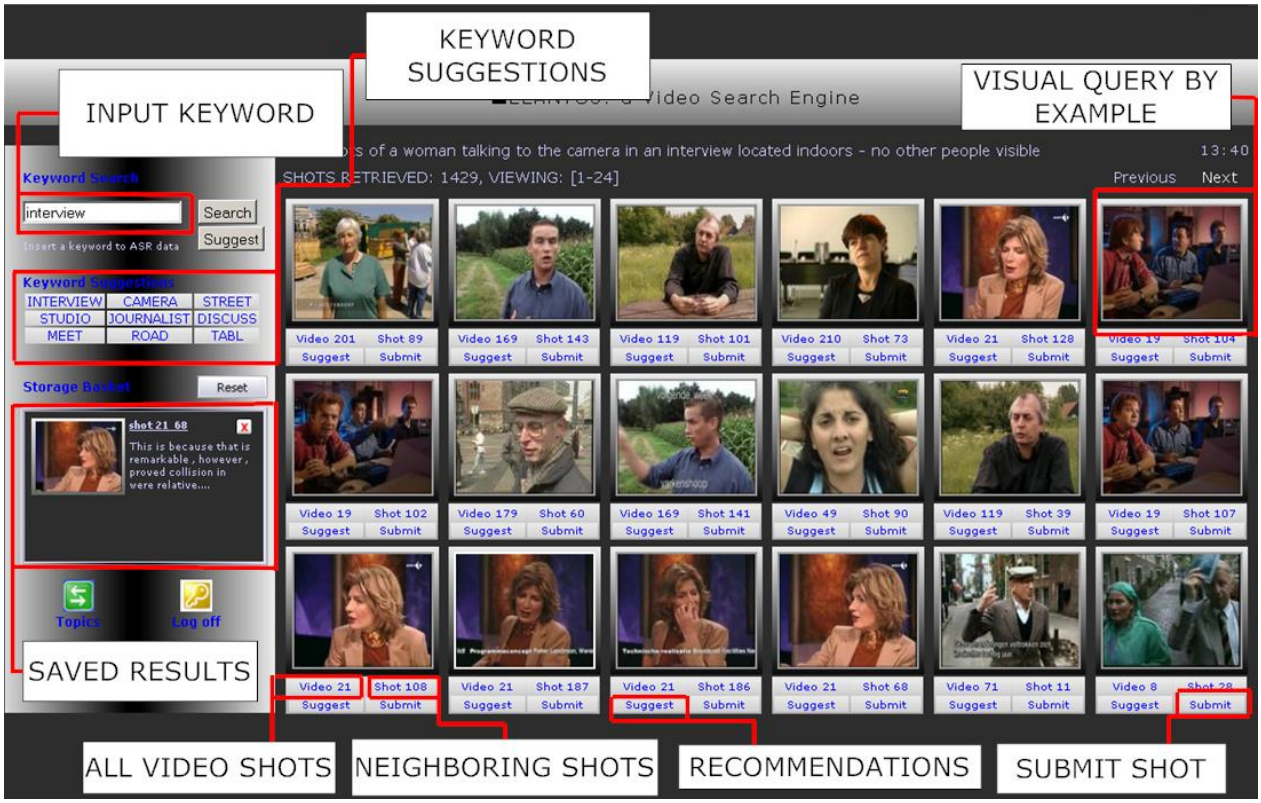

Figure 7. Video Search Engine Interface.

The h_R is the homogeneity co-efficient, which corresponds to the probability that the samples in the right split of node T belong to the observed or synthetic data and measures the purity of the observations in a node. In a similar way the h_L is the homogeneity co-efficient for the left split, while h represents the homogeneity co-efficient before the split. As p_{Tj}, p_{TRj} and p_{TLj} the declared the probabilities of the observations in node T belonging to class j before the split, and in the right and left split respectively.

8. VIDEO SEARCH ENGINE

In this section, we present the interactive video search engine LELANTUS, which was used for our experiments. The search engine interface (Figure 7) is composed of two main parts: the left column, which offers text-based search options and the main container where the results are presented offering at the same time options for queries by visual example and temporal search.

At the left column the user is allowed to enter a keyword in order to fire a text-based search exploiting the ASR information. In addition, the left column includes a basket storage structure, where the user can store the results he/she founds. The main container is the part where the results are presented. Four different options are available for each shot: i) to perform a query by visual example using the MPEG descriptors by clicking on the representative image, ii) to mark a shot as relevant to the topic (i.e. submit a shot), iii) to view all the shots of the same video and iv) to view the temporally adjacent (i.e. neighboring) shots of a selected video shot with the associated textual transcription.

9. EXPERIMENTAL SETUP

To evaluate our approach, we perform an experiment with LELANTUS and an eye-tracker. We used the TRECVID 2008 test video set by NIST[2], which includes around 100 hours of video, and the following query topics:

A. Find shots of people with one or more horses
B. Find shots of a map
C. Find shots of people with one or more books
D. Find shots of food and/or drinks on a table

The experiment took place in the laboratories of Queen Mary, University of London and 8 subjects (4 male and 4 female) were recruited to participate. The participants were mostly postgraduatetudents or postgraduate researchers with an average age of 30.5 years old. All of them had a very good knowledge of English and a computer science background. In addition most of them had a good understanding of retrieval tasks and were familiar with search engines. The task for each user was to search during a time window of 10 minutes per topic and find as many results that satisfy the given search topic using the LELANTUS search engine. In order to imitate as much as possible a real world video retrieval task we instructed the users to search as they normally do (i.e. without making extra effort to focus their gaze on the shots of interest as they were instructed to do in [5]). A tutorial session preceded this task to familiarize the users with the search engine and the existence of the eye-tracker.

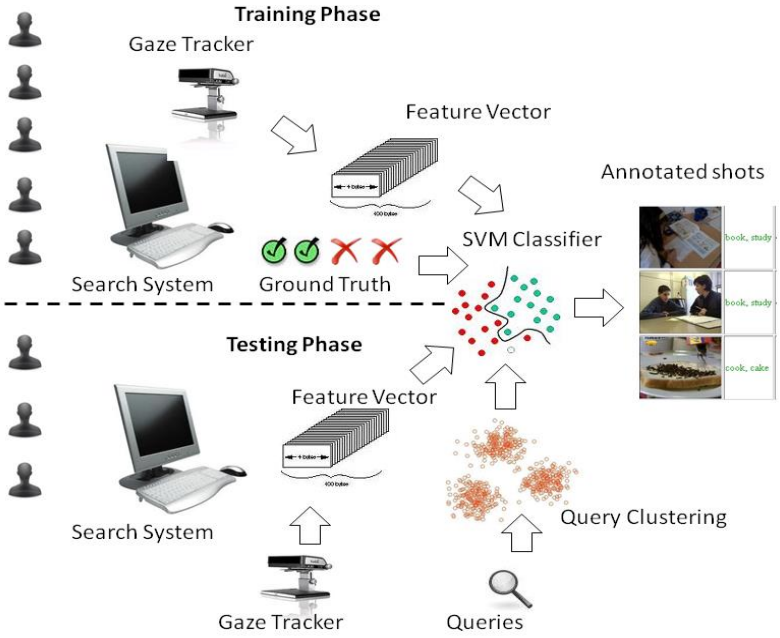

Figure 8. Interactive experiment and video annotation

The whole experiment is depicted in the schematic view of Figure 8. To record the gaze movement of the users we employed a binocular set of 60Hz cameras with Infra-Red filters and the faceLAB 5.0 software package as the eye-tracking technology4. This system requires a user calibration phase, which takes less than one minute. It offers an error of less than 0.5 degrees that suggests approximately less than 5mm diversion from the actual gaze point, when the user is looking at the screen from a distance of 50cm. We used the output of the eye-tracker, in order to gain knowledge regarding the coordinates of each users gaze for a given time. Then, we processed this information to identify eye fixations and pupil dilations on the video shots. We considered as minimum time of 100ms to define a fixation, during which the gaze was stable.

9.1 Training and Testing

In this experiment, we have considered as training data the search sessions performed by 5 users. The results submitted by these users constitute an explicit relevance metric with respect to the query topics for all the viewed items. In order to evaluate the framework, we consider three cases, in which different clustering algorithms,

[2] NIST: http://www.nist.gov/

combinations of topics and users are used for training and testing. More specifically, the following cases, which are shown in Table 3 are considered:

In the first case, we train recursively 6 classifiers that detect the user interest (models 1-6 in Table 3) by selecting each time a different combination of two topics (i.e. (A, B), (A, C), etc.) and using as vector the fixation-based features (Table 1) and we consider the topics known during testing. This means that during testing we apply a topic-based merging of gaze data, since we are able to aggregate the gaze information along the topics that were searched by the users. We also assume 2 different user variations for training and testing. The first variation uses training data from users 1-5 and test data from users 6-8, while the second assumes training data from users 4-8 and test data from users 1-3.

In the second case (model 7 in Table 3), we employ the gaze-driven RF clustering approach and we consider the same topics for training and testing. In order to evaluate the clustering algorithm during testing we average the results from all the possible 5-3 combinations of training and testing users, which leads to 56 user variations. In this case we assume that the topics are unknown during testing and therefore query clustering is employed to identify the topics.

Finally, in the third case (models 8-13 in Table 3) we employ again the gaze-driven RF and we consider all the different combination of two topics, as well as the same 2 user variations as in case 1. The topics during testing are considered again unknown.

In all cases, grid search is employed to select the best SVM training parameters.

Table 3.Training and testing cases

Case	Model	Training-Testing topics	Users	Clustering	Merging
1	1-6	2-2	2 variations	Initial topics	Topic-based
2	7	4 same	56 variations	Gaze driven RF	Cluster-based
3	8-13	2-2	2 variations	Gaze driven RF	Cluster-based

10. RESULTS AND EVALUATION

In this section we present results and evaluate the gaze movement-based classification step, the gaze movement driven RF clustering algorithm and the finally the produced annotations.

10.1 Gaze movement-based classification evaluation

First, we evaluate the gaze movement-based classification by reporting the classification accuracy, the precision, the recall and the F-score over the items returned by the system as positive results. We mostly judge the performance of the system using F-Score, due to the fact that the considered data are imbalanced (i.e. very few positive examples compared to negatives) and therefore judging only by the accuracy could be misleading (e.g. marking all the results as negative could provide an accuracy of 90%). The results for the aforementioned training cases using train data from two different user variations are reported in Table 4. Based on these results the average F-Score is around 55%, while the accuracy reaches 96.4%. Depending on the application of the method we could train the classifier to give more weight to the precision and less to the recall. This will reduce the False detection rate (currently around 38%).

In Figure 9 we provide some indicative annotated shots for topic A that could be used as recommendations for future users, who search for the same topic.

In average the training query submissions are 396.6. The training of each model including the feature extraction and aggregation process took about 205 seconds in a PC Intel(R) Core(T) TM2 Quad CPU Q6600 @ 2,4GHz, 3,25GB RAM. Given the fact that this is an offline procedure the processing time is reasonable and could be further reduced with a more powerful PC.

Figure 9. Annotated shots for topic A using model 6.

Table 4. Topic-based merging

Model	Train topics	Test topics	Accuracy	Precision	Recall	False detection rate	F-Score
1	A,B	C,D	96.91%	69.8%	38.2%	30.2%	49.38%
2	A,C	B,D	95.83%	71.97%	45.22%	28.03%	55.54%
3	A,D	B,C	95.44%	66.33%	41.12%	33.67%	50.76%
4	B,C	A,D	96.9%	52.12%	66.7%	47.88%	58.51%
5	B,D	A,C	96.5%	43.3%	69.7%	56.7%	53.42%
6	C,D	A,B	96.83%	67.12%	52.12%	32.88%	58.67%
Average			**96.40%**	**61.77%**	**52.17%**	**38.23%**	**54.38%**

10.2 Clustering evaluation methodology

To evaluate the performance of clustering we use the normalized version of the Mutual Information metric NMI [36]. Assuming that we have a set of N data items and two clustering solutions U and V (e.g. the proposed clustering solution and the ground truth), $U = \{U_1, U_2, \dots, U_R\}$ with R clusters and $V = \{V_1, V_2, \dots, V_C\}$ with C clusters, the NMI is defined as:

$$NMI(U,V) = \frac{I(U,V)}{\sqrt{H(U)H(V)}} \tag{14}$$

In (14) $H(U)$ is the information entropy of a clustering solution U:

$$H(U) = -\sum_{i=1}^{R} P(i) log P(i) \tag{15}$$

where $P(i) = \frac{|U_i|}{N}$. Similarly the $H(V)$ is calculated as:

$$H(V) = -\sum_{j=1}^{C} K(j) log K(j) \tag{16}$$

where $K(j) = \frac{|V_j|}{N}$. Finally the mutual information between these two clustering solutions is calculated as:

$$I(U,V) = \sum_{i=1}^{R}\sum_{j=1}^{C} P(i,j) log \frac{P(i,j)}{P(i)K(j)} \tag{17}$$

where $P(i,j)$ denotes the probability that a point belongs to cluster U_i in U and cluster V_j in V: $P(i,j) = \frac{|U_i \cap V_j|}{N}$ [36].

In order to evaluate the clustering approach, we compare the gaze-driven RF clustering (RF-G) with a baseline such as K-means, as well as with the traditional random forest clustering (RF), (i.e. without considering the gaze movements). In the following, we discuss the second case, in which all the 4 topics are involved. In average the training query submissions are 396.6, while the testing queries were 238.2. An ideal clustering method would group these queries into 4 different clusters. Given the fact that we consider the topics unknown, we attempt to cluster the queries to different number of clusters and evaluate the performance of different algorithms using the NMI metric.

For each RF we have constructed a variety of trees ranging from 10 to 500, and we assume 15 random variables. The number of random variables is selected to be close to the square root of the feature cardinality (i.e. 238.2 in average for all the user variations). We decided to stop at 500 trees, since we observe that the results have started to converge. After tuning experimentally the parameters for h (section 7.2) we set a = 40 and b = 40. In the following, we present the average cross-validated results (56 user variations) for different cardinality of clusters (4-10). Given the fact that the initial topics were 4, we don't provide results for more than 10 clusters as the performance starts to drop, since the clustering in so many groups is not meaningful.

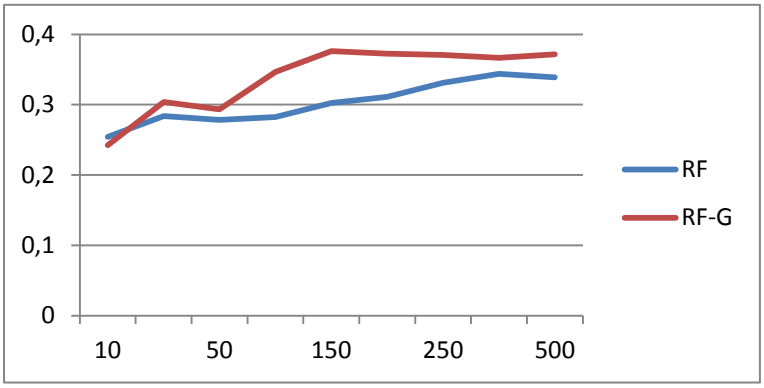

Figure 10. The gaze-driven RF NMI for 4 clusters

In Figure 10 we report the results for 4 clusters. The gaze-driven RF outperforms the traditional RF method by an average of 8.9%.

When the number of clusters is increased to 6, the results (Figure 11) are improved reaching a final NMI of 0.39. However, the increase compared to the traditional RF is lower, since it reaches 5.4%.

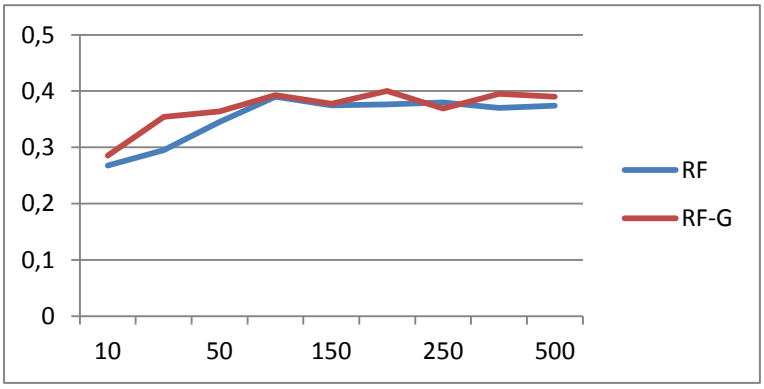

Figure 11. The gaze-driven RF NMI for 6 clusters

In Figure 12 we present the results for 8 clusters. The algorithm performance increases in comparison with the 4 and 6 clusters and outperforms RF with an average of 8%.

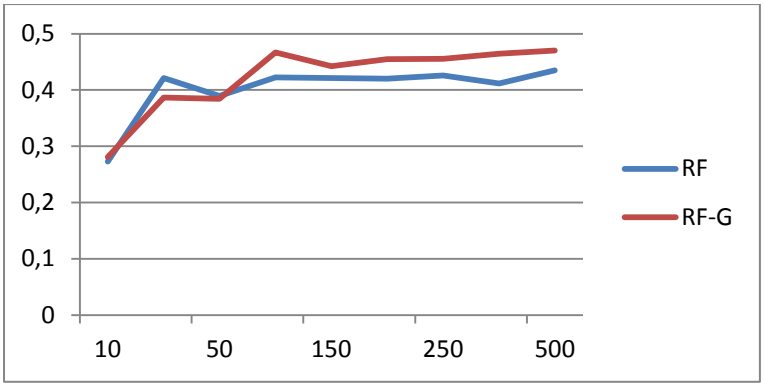

Figure 12. The gaze-driven RF NMI for 8 clusters

Finally we report the results (Figure 13) in the case of 10 clusters. The gaze-driven RF demonstrates a performance of 0.48, which improves the performance of traditional RF by a 11.8%.

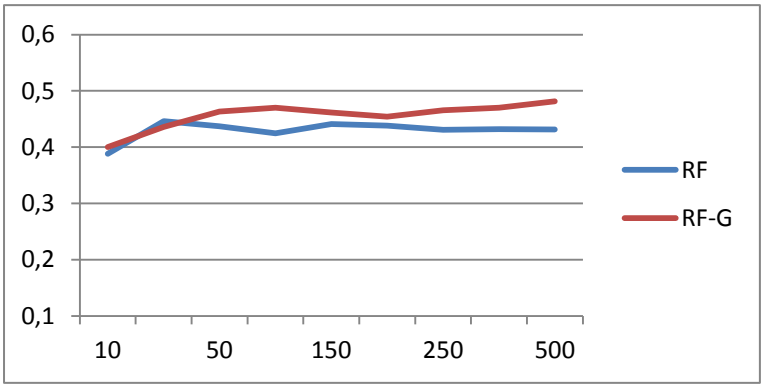

Figure 13. The gaze-driven RF NMI for 10 clusters

Summarizing, in Table 5 we report the results for 500 trees and compare them with the K-means baseline. A graphical comparison of these results is available in Figure 14. First it is interesting to notice that NMI is in general increasing together with the number of clusters. This is probably due to the fact that in this way we avoid associating queries with totally irrelevant clusters. In average the gaze-driven RF (G-RF) performs better.

Table 5. NMI for the 3 clustering techniques (500 trees)

Num of clusters/ Technique	K-means	RF	RF-G
4 clusters	0.2834	0.339	0.372
6 clusters	0.3534	0.374	0.39
8 clusters	0.3677	0.435	0.47
10 clusters	0.3734	0.432	0.48
Average	**0.344**	**0.3949**	**0.4283**

Figure 14. The performance for the 3 clustering algorithms

In Figure 14 we present the performance of the three clustering algorithms along all the clusters. It is clear that in all cases RF-G outperforms the other two baselines. Finally the average performance for all clusters is presented in Figure 15.

As discussed, in average the testing query submissions are 238,2. The clustering procedure regardless of the number of the clusters was pretty fast for the RF reaching in average 9 seconds. On the other hand, when the RF-G was employed the computational time was substantially increased since for each split the homogeneity efficient had to be calculated. In average the calculation time was 272 seconds. The PC used was an Intel(R) Core(T) TM2 Quad CPU Q6600 @ 2,4GHz, 3,25GB RAM. Since the clustering is an offline procedure the computational time is reasonable and could be further reduced with a more powerful PC.

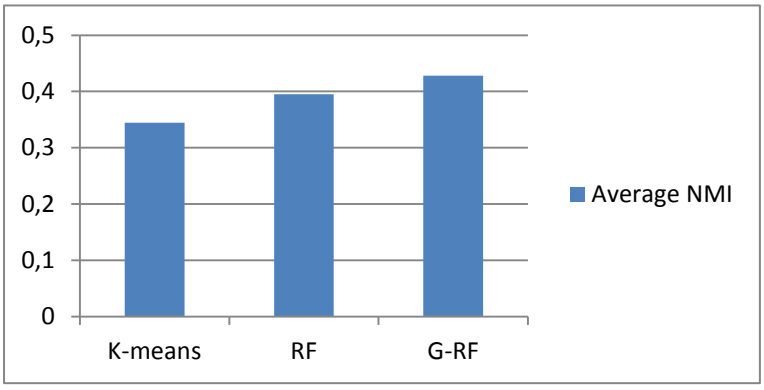

Figure 15. The average performance for the 3 clustering algorithms

10.3 Evaluation of the video annotations

In this section we provide results on the annotated shots based on gaze movement driven RF clustering and gaze classification. We report results of the third case (models 8-13), after considering two different user variations. In this case more than 116 shots were annotated in average. It is also interesting to notice how the initial classification precision (gaze-movement based classification) is decreasing from 60.46% to 55.57% due to the error introduced by the clustering algorithm. In case the topics during testing were known, the initial classification precision and the final precision would have been the same. Also the false detection rate is about 44%, which means that almost 4 out of 10 shots were not annotated correctly. However, we could reduce the False detection rate if we train the classifier to give more weight to the classification precision and less to recall.

Table 5. Annotations of the third case (Table 3)

Model	Class. Prec.	F-score	NMI	Correctly annotated	Annotated shots	Final False detection rate	Final Precision
8	64.4%	50.65%	0.51	52	121	57%	43%

9	70.5%	58.90%	0.58	81	122	33.7%	66.3%
10	65%	52.21%	0.51	62	100	38%	62%
11	58.8%	58.01%	0.57	68	114	40.4%	59.6%
12	52.9%	52.5%	0.54	61	119	48.74%	51.26%
13	51.2%	55.51%	0.5	59	123	52.1%	47.9%
Average	**60.5%**	**54.63%**	**0.535**	**63.83**	**116.5**	**44.43%**	**55.57%**

Figure 16 demonstrates some examples of annotated shots. In this case we can see that shots depicting library and people studying have been successfully annotated with the labels "book" and "study", while shots relevant to food (people eating, bread, fruits) have been annotated with relevant labels i.e. "cook" and "cake".

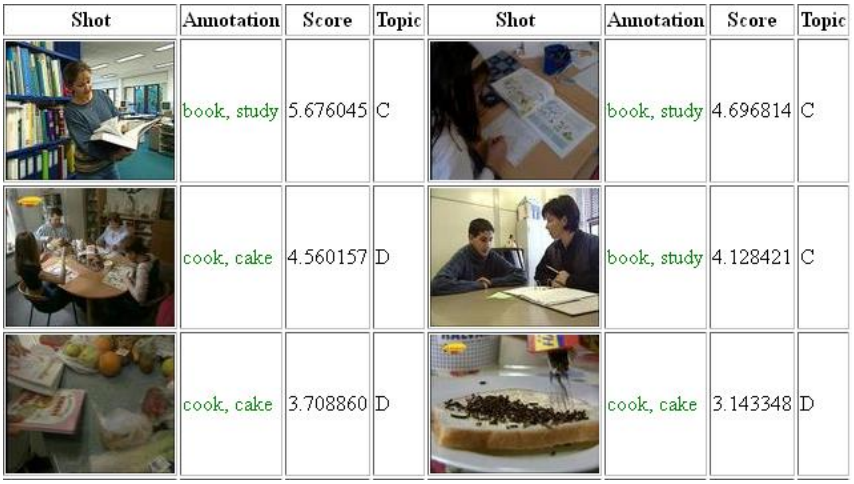

Figure 16. Automatic annotations using model 9

11. CONCLUSIONS

In this paper we have investigated the potential of automatic video annotation by combining aggregated gaze movement data and click-throughs of users in interactive video retrieval. In this context we have introduced a novel query clustering algorithm based on gaze movement-driven random forest, which relies not only upon textual and visual similarity between the queries but also on the gaze movements.

Although the results are based in an experiment with limited number of users and topics, they can be considered as an important indication that such techniques could be used effectively for video annotation. However, it should be noted that the approach can be scalable, since it builds upon a fast performing clustering algorithm (i.e. random forests), while dimensionality reduction techniques such as PCA or sparse KPCA [39] could be applied to reduce the affinity matrix. In addition, the processing required to perform the automatic annotation is realized off-line (i.e. not at real time) and therefore this ensures that the proposed technique could be applied to real-world applications.

Future work includes experiments with more users and topics in order to further test the scalability of the method and evaluate its results for real-world application. Also, we plan to incorporate mouse click-based features into the shot classification process to further strengthen the classifier that detects shots of interest. To this end we will investigate discriminant parallel feature methods [40], as well as multiview learning approaches such as the application of Multiview Hessian discriminative sparse coding [41].

12. ACKNOWLEDGMENTS

This work was partially supported by the projects MULTISENSOR (FP7-610411), HOMER (FP7-312388) and PetaMedia (FP7-216444).

13. REFERENCES

[1] S. Vrochidis, I. Kompatsiaris, and I. Patras, "Utilizing implicit user feedback to improve interactive video retrieval," Advances in Multimedia, vol. 2011, no. 15, 2011.

[2] I. Sarafis, C. Diou, A. Delopoulos, “Building effective SVM concept detectors from clickthrough data for large-scale image retrieval”, International Journal of Multimedia Information Retrieval, Volume 4, Issue 2, 2015, pp 129-142.

[3] F. Hopfgartner and J. Jose, “Evaluating the implicit feedback models for adaptive video retrieval,” in Proc. 9th ACM SIGMM International Workshop on Multimedia Information Retrieval, Augsburg, Germany, 2007, pp. 323–331.

[4] F. Hopfgartner, D. Vallet, M. Halvey, and J. Jose, “Search trails using user feedback to improve video search,” in Proc. 2008 ACM Multimedia, Vancouver, Canada, 2008, pp. 339–348.

[5] A. Klami, C. Saunders, T. D. Campos, and S. Kaski, “Can relevance of images be inferred from eye movements?” in Proc. 1st ACM International Conference on Multimedia Information Retrieval, Vancouver, Canada, 2008, pp. 134–140.

[6] S. Vrochidis, I. Patras, and I. Kompatsiaris, “An eye-tracking-based approach to facilitate interactive video search,” in Proc. 1st ACM International Conference on Multimedia Retrieval (ICMR’11), Trento, Italy, 2011.

[7] L. Jiang, X. Chang, Z. Mao, A. Armagan, Z. Lan, X. Li, S. Yu, Y. Yang, D. Meng, P. Duygulu-Sahin, A. Hauptmann,, “CMU Informedia @ TRECVID 2014: Semantic Indexing” in Proc. TRECVID 2014 Workshop, Gaithersburg, USA, 2014.

[8] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, T. Mei, and H.-J. Zhang, “Correlative multi-label video annotation,” in Proc. ACM Multimedia 2007, Augsburg, Germany, 2007.

[9] S. Ayache and G. Queenot, “Video corpus annotation using active learning,” in Proc. European Conference on Information Retrieval (ECIR’08), Glasgow, Scotland, 2008.

[10] S. Koelstra, C. Muehl, and I. Patras, “Eeg analysis for implicit tagging of video data,” in Proc. Workshop on Affective Brain-Computer Interfaces (ABCI’09), Amsterdam, Canada, 2009, pp. 27–32.

[11] Y. Zhang, H. Fu, Z. Liang, Z. Chi, and D. Feng, “Eye movement as an interaction mechanism for relevance feedback in a content-based image retrieval system,” in Proc. Symposium on Eye-Tracking Research and Applications, Austin, Texas, 2010, pp. 37–40.

[12] T. J. F. Radlinski, “Query chains: learning to rank from implicit feedback,” in Proc. 11th ACM SIGKDD international conference on Knowledge discovery in data mining, Chicago, Illinois, USA, 2005.

[13] B. Yang, T. Mei, X.-S. Hua, L. Yang, S.-Q. Yang, and M. Li, “Online video recommendation based on multimodal fusion and relevance feedback.” in Proc. 6th ACM international Conference on Image and Video Retrieval (CIVR’09), Amsterdam, Canada, 2007, pp. 73–80.

[14] K. Rayner, “Eye movements in reading and information processing,” Psychological Bulletin, vol. 124, pp. 372–422, 1998.

[15] T. J. L. A. Granka and G. Gay, “Eye-tracking analysis of user behavior in www search,” in Proc. 27th Annual International ACM SIGIR conference on Research and Development in Information Retrieval, New York, USA, 2004.

[16] E. S. J. S. S. K. K. Puolamaki, J. Salojarvi, “Combining eye movements and collaborative filtering for proactive information retrieval,” in Proc. 28th Annual International ACM SIGIR conference on Research and Development in Information Retrieval, Salvador, Brazil, 2005.

[17] A. A. K. P. S. K. D. R. Hardoon, J. Shawe-Taylor, “Information retrieval by inferring implicit queries from eye movements,” in Proc. 11th International Conference on Artificial Intelligence and Statistics, San Juan, Puerto Rico, 2007.

[18] B. W. G. M. A. Hughes, T. Wilkens, “Text or pictures? an eyetracking study of how people view digital video surrogates,” in Proc. 2nd International Conference on Image and Video Retrieval (CIVR’03), Urbana, IL, USA, 2003, pp. 271–280.

[19] L. Kozma, A. Klami, and S. Kaski, “Gazir: gaze-based zooming interface for image retrieval,” in Proc. 2009 International Conference on Multimodal interfaces (ICMI09), New York, USA, 2009, pp. 305–312.

[20] Q. Li, B. Key, J. Liu and J. Sun, “A Novel Image Retrieval System with Real-Time Eye Tracking”, in Proc. of International Conference on Internet Multimedia Computing and Service (ICIMCS '14), 2014.

[21] G.T. Papadopoulos, K. C. Apostolakis and P. Daras, “Gaze-Based Relevance Feedback for Realizing Region-Based Image Retrieval”, IEEE Transactions on Multimedia, Volume 16, Issue 2, pp 440-454.

[22] S. Vrochidis, I. Patras, and I. Kompatsiaris, “Exploiting gaze movements for automatic video annotation,” in Proc. 13th International Workshop on Image Analysis for Multimedia Interactive Services, Dublin, Ireland, 2012.

[23] K. P. D. R. Hardoon, “Image ranking with implicit feedback from eye movements,” in Proc. Symposium on Eye-Tracking Research and Applications, Austin, Texas, USA, 2010, pp. 291–298.

[24] Y. Z. Z. C. D. F. Z. Liang, H. Fu, “Content-based image retrieval using a combination of visual features and eye tracking data,” in Proc. Symposium on Eye-Tracking Research and Applications, Austin, Texas, USA, 2010, pp. 41–44.

[25] P. Auer, Z. Hussain, S. Kaski, A. Klami, J. Kujala, J. Laaksonen, A. P. Leung, K. Pasupa, and J. Shawe-Taylor. Pinview: Implicit feedback in content-based image retrieval. In ICML Workshop on Reinforcement Learning and Search in Very Large Spaces, Haifa, Israel, June 2010.

[26] L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[27] L. Breiman and A. Cutler, “Random forests manual v4.0.” Berkeley, University of California, California, USA, Technical Report 99-02, 1999.

[28] C. Chang and C. Lin. Libsvm: a library for support vector machines.

[29] S. Beitzel, E. Jensen, D. Lewis, A. Chowdhury, and O. Frieder, “Automatic classification of web queries using very large unlabeled query logs,” ACM Transactions on Information Systems, vol. 25, no. 2, 2007.

[30] J.-R. Wen, J.-Y. Nie, and Z. Hong-Jiang, “Query clustering using user logs,” ACM Transactions on Information Systems, vol. 20, pp. 59–81, 2002.

[31] S. Lloyd, “Least squares quantization in pcm,” IEEE Transactions on Information Theory, vol. 28, no. 2, pp. 129–137, 1982.

[32] T. Pedersen, S. Patwardhan, and J. Michelizzi, “Wordnet::similarity-measuring the relatedness of concepts,” in Proc. 19th National Conference on Artificial Intelligence (AAAI’04), California, USA, 2004, pp. 1024–1025.

[33] P. Jaccard, “Nouvelles recherches sur la distribution florale.” IEEE Transactions on Information Theory, vol. 44, pp. 223–270, 1908.

[34] R. Burkard, M. Dell’Amico, and S. Martello, Assignment Problems. SIAM, 2009.

[35] H. Kuhn, “The hungarian method for the assignment problem.” Naval Research Logistics Quarterly, vol. 2, pp. 83–97, 1955.

[36] N. Vinh, J. Epps, and J. Bailey, “Information theoretic measures for clusterings comparison: is a correction for chance necessary?” in Proc. 26th Annual International Conference on Machine Learning (IMCL’09), Montreal, Canada, 2009, pp. 1073–1080.

[37] S.C. Chuang, Y.Y. Xu, H.C. Fu, and H.C. Huang, "A Multiple-Instance Neural Networks Based Image Content Retrieval System," Proc. Int'l Conf. on Innovative Computing, Information and Control, vol. 2, pp. 412-415, Beijing, China, 2006.

[38] 4.d. P.S. Lai et al., "Automated Information Mining on Multimedia TV News Archives," Lecture Notes in Artificial Intelligence (LNAI), vol. 3682, pp. 1238-1244, 2005.

[39] X. Xie, B. Li, and X. Chai, "Adaptive Sparse Kernel Principal Component Analysis for Computation and Store Space Constrained-based Feature Extraction," Journal of Information Hiding and Multimedia Signal Processing, vol. 6, no. 4, pp. 824-832, July 2015.

[40] X.F. Liu and X.X. Zhu, "Parallel Feature Extraction through Preserving Global and Discriminative Property for Kernel-Based Image Classification," Journal of Information Hiding and Multimedia Signal Processing, vol. 6, no. 5, pp. 977-986, Sep. 2015.

[41] W. Liu, D. Tao, J. Cheng, Y. Tang, “Multiview Hessian discriminative sparse coding for image annotation”, Computer Vision and Image Understanding, vol 118, 2014, pp 50–60.

[42] W. Liu, Y. Li, X. Lin, D. Tao, Y. Wang, “Hessian-Regularized Co-Training for Social Activity Recognition”, PLoS ONE vol 9, no 9, 2014.

[43] W. Liu, H. Liu, D. Tao, Y. Wang, K. Lu, "Multiview Hessian regularized logistic regression for action recognition", Signal Processing, Volume 110, 2015, pp 101-107.

[44] A. Iosifidis, A. Tefas, I. Pitas, “Multi-view action recognition based on action volumes”, fuzzy distances and cluster discriminant analysis, Signal Process. 93 (2013) 1445–1457