An Automatic Web-Oriented Multimedia Extraction and Multiresolution Visualization Scheme

KLIMIS NTALIANIS[1], NIKOLAOS PAPADAKIS[2]

[1]Department of Marketing
Technological Educational Institute of Athens
12210, Athens, Egaleo
GREECE
kdal75@gmail.com

[2]Electrical and Computer Engineering Department
National Technical University of Athens
9, Iroon Polytechniou str., Zografou 15773, Athens
GREECE
nkpap@telecom.ntua.gr

Abstract: - When no explicit annotation exists, web-based image search engines cannot understand the actual content of images. As a result they return several noisy data in most searches, leading to low precision and recall. Furthermore, usually a large number of images are returned in several sequential pages, a visualization way that is not view- and transmission-efficient. To overcome these problems an automatic wrapper-based image retrieval and presentation system is proposed that performs significantly better than other search engines. In particular our novel system exploits the format of multimedia sharing web sites to discover the underlying structure in order to finally infer and extract multimedia files and corresponding associated keywords from the web pages. Afterwards the gathered content is properly organized in a multiscale tree structure for efficient visualization. By this way, users can select only paths-of-interest in the tree structure, thus detecting the needed content much faster and spending less bandwidth compared to current solutions. Experimental results are presented that indicate the interesting performance of the proposed architecture.

Key-Words: - Content retrieval, wrapper technologies, multi-resolution visualization, web data extraction

1 Introduction

Year after year more and more multimedia files are published on the Web. Most of them are either poorly or not annotated and thus it is difficult to efficiently search, retrieve and present them. To confront this problem, in the recent years more than 200 content-based retrieval systems have been developed [1], the majority of which are based on low-level features. In particular they can be classified into two main categories: (a) those that mine semantics by analyzing associated textual information, such as annotations, assigned keywords, captions, alt or surrounding text and (b) those that extract low-level visual features. Methods of the first category depend on laborious annotation, while the latter methods usually cannot capture semantics effectively. Some characteristic approaches use techniques that automatically annotate multimedia content by applying a series of image/video processing algorithms and then extract the semantic meaning of the content [2], [3]. However, their main drawback is that they support only limited annotation vocabulary (e.g., indoor/outdoor). Specialized research approaches are also proposed in case of content of specific types, such as sports, news etc [4], [5]. Another approach is to semantically tag multimedia content through multiple textual annotations [6]. However, this approach cannot be applied to generic unstructured content.

On the other hand, the last decade research has moved towards automatically acquired web data, in order to be used for training concept classifiers or retagging [7], [8]. Such data have been annotated by user-defined tags (e.g., Picasa, Flickr, Youtube etc). However concept classifiers have most of the problems of traditional CBIR systems. Furthermore leading search engines such as Google and Yahoo retrieve web images by checking captions, the html page content and the surrounding text, information that may be irrelevant to the content of an image.

To overcome these problems some wrapper-based methods have been proposed in literature.

ISBN: 978-1-61804-074-9

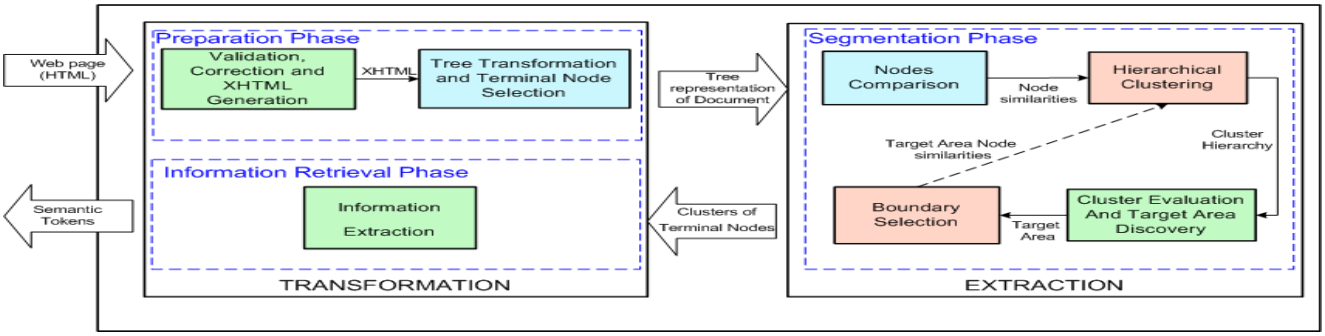

Figure 1: Overview of the automatic wrapper module

For example in [9] the user has to perform a sample query on a component called provider and then mark the important elements in the web pages, thus guiding the generation process of the wrapper. The work in [10] is based on two observations about data records on the Web and the use of a string matching algorithm. The first is that a group of data records containing descriptions of a set of similar objects are typically presented in a particular region of a page and are formatted using similar HTML tags (which are regarded as strings). The second observation is that a group of similar data records being placed in a specific region is reflected in the tag tree by the fact that they are under one parent node.

In order to avoid human guidance and raw-tag manipulations in this paper we propose a novel scheme that automatically segments pages of web image sharing sites into structural tokens and organizes the extracted content in a visualization-efficient way. In particular images are commonly presented in mostly structured HTML pages, but this structure is not known. In our case, managing this task is made somewhat easier by the fact that web image sharing sites do have some structure. The image is presented in a part of a web page, while its keywords are placed in another part. Content extraction is performed by a novel fully automated wrapper-based scheme that is able to segment a web page into structural tokens and select image and associated keywords. Then gathered images are fed to a multiscale content organization module, which produces a tree structure. Nodes of the tree correspond to images, while this structure enables users to detect content of interest much faster and using less bandwidth compared to current web multimedia search engines and web sharing sites.

The rest of the paper is organized as follows: Section 2 provides a short overview of the automatic wrapper module, while in Section 3 a detailed description of the multiscale content organization module is given. Experimental results are presented in Section 4 and Section 5 concludes this paper.

2 The Automatic Wrapper Module

Multimedia files and associated keywords extraction is accomplished by an automatic wrapper-based mechanism. The system first identifies each section of the web page that contains multimedia files and then extracts them by using clustering and statistical techniques. Afterwards the gathered content is properly organized in a multi-resolution form for efficient visualization. The proposed system is based on STAVIES [11] and comprises of 2 modules: the transformation and the extraction module. The transformation module is divided into 2 phases, the preparation and the information retrieval phase, while the extraction module includes a segmentation phase. An overview of the proposed wrapper module can be seen in Figure 1. Next a short description is provided for the two main phases, while further information can be found in [11].

2.1 Preparation Phase

Initially validation, correction and XHTML generation is performed to syntactically correct the source's HTML (by transforming it into XHTML). This is necessary because due to the leniency in HTML parsing by modern web browsers, a major portion of the web page is not well-formed. Data sources often either contain invalid tags or their tags are placed in a wrong manner.

From the cleaned and normalized page a tree representation is produced. The root of the tree corresponds to the whole document. The intermediate nodes represent HTML tags that determine the layout of the page. Finally, the terminal nodes (leaf nodes) correspond to visual elements on the web page, namely images, links and/or text. Once the tree construction is completed, the terminal nodes are selected. Non-terminal nodes are not further analyzed, since they represent layout descriptive elements.

2.2 Segmentation Phase

In this phase initially the sections in the input page are identified and one of them is characterized as the area where the semantic tokens reside.

ISBN: 978-1-61804-074-9

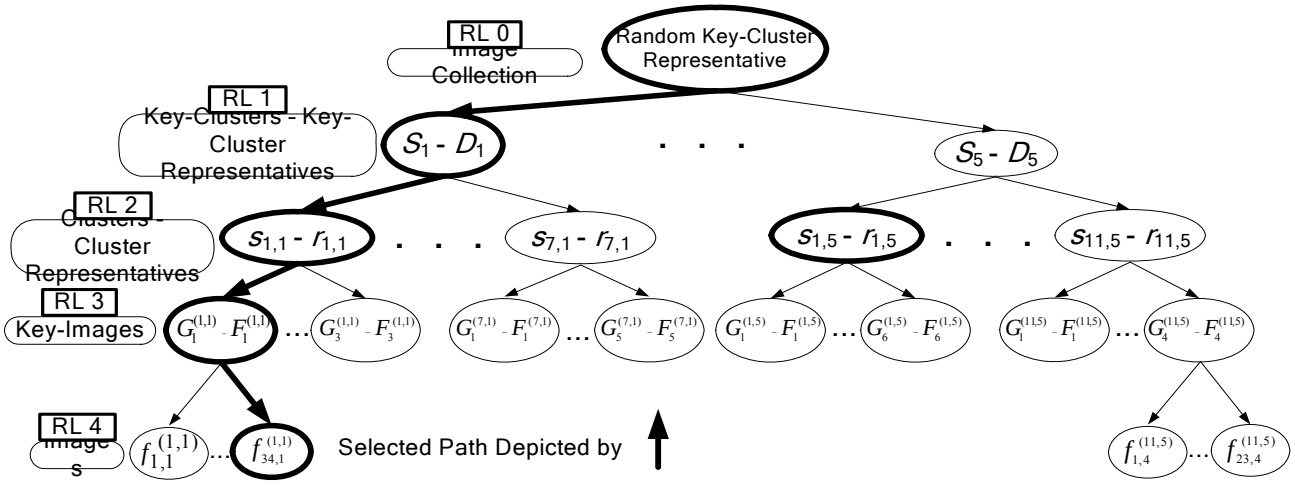

Figure 2: The tree structure

Afterwards segmentation is performed in two steps: the "Target Area Discovery" and the "Target Area Segmentation" steps. Given the list of terminal nodes, the "Target Area Discovery" process aims at selecting a subset of these nodes corresponding to the elements belonging to the target area. This is achieved by applying hierarchical clustering. In the "Target Area Segmentation" step, a segmentation of the target area into further segments that represent the semantic tokens is performed, by locating a "cut-off" level. Once the cut-off level is determined, the system performs a clustering similar to the one performed before. Each output cluster now represents a semantic token.

3 The Multiscale Content Organization Module

3.1 Overview of the Multiscale Module

As previously mentioned, the wrapper feeds the multiscale organization module with images together with their associated keywords. Aim of this module is to organize images into different resolution levels, providing a tree structure, so that they are effectively visualized. An overview of the proposed structure is presented in Figure 2. Each node of the tree corresponds to a particular resolution level. More specifically let us assume that a user searches a database using the keyword "beach" and 20.000 images match his criterion. These images can be grouped using a clustering algorithm. For each cluster one representative image is extracted, namely the cluster representative. Gathering together all cluster representatives, the cluster representatives' set is formed. On the next step key-cluster representatives are extracted from the cluster representatives' set and an influence zone is estimated for each key-cluster representative to classify the remaining cluster representatives. As a result, key-clusters are created (level 1 of tree) and each key-cluster is visualized by the respective key-cluster representative. In our example five key-clusters are constructed (S_1,…, S_5), each represented by one key-cluster representative (D_1,…, D_5). Next, in level 2 clusters are represented, each one connected to a specific key-cluster. In particular, a node of level 1, which corresponds to a specific key-cluster, is partitioned into the clusters that this key-cluster contains. Cluster $s_{i,k}$ is represented by one image, namely the cluster representative, which is denoted as $r_{i,k}$.

Afterwards the content of each cluster is further decomposed in level 3 by extracting key-images so that the cluster's visual content is analyzed in more detail. The remaining images of the cluster are classified with respect to key-images generating image classes. In the presented example, three key-images are extracted for cluster $s_{1,1}$ ($F_1^{(1,1)}$, $F_2^{(1,1)}$, $F_3^{(1,1)}$) and 34 images are associated with key-image $F_1^{(1,1)}$ ($f_{1,1}^{(1,1)}$,…, $f_{34,1}^{(1,1)}$).

3.2 Content Preprocessing and Clustering

Assuming that we have JPEG images, initially the traditional pixel-based image representation is transformed to a feature-based one. This is accomplished by extracting *global-based* and *object-based descriptors*. Global descriptors are extracted from the histograms of color and texture, which, in our case, are computed using information directly available from the JPEG compressed stream. More specifically, the color histogram is calculated using the DC coefficients for every 8×8 block of an image. In a similar way, the texture histogram is computed based on the "energy" (squared sum) of the AC coefficients for every 8×8 block of an image.

To obtain object-based features, each image is first partitioned into several segments by applying the M-RSST algorithm, due to its efficiency and small

computational complexity [12]. For each segment several descriptors are estimated, representing the segment content. In our case, the three average color components of a segment (in the RGB color space), the location of the segment's center, as well as the segment size are used as object-based descriptors. Finally all descriptors are gathered together to form a feature vector, say $\mathbf{g}_i$ for the ith image.

Next spatial clusters are created, the medoids of which are used as cluster representatives. In our work, the CLARANS algorithm [13] has been adopted for spatial clustering due to its low complexity, scalability and quality of results. In particular let us consider that cluster representatives should be extracted from each image set V. According to CLARANS, the process of finding k medoids among n points of a space, can be viewed abstractly as searching through a certain graph. In such a graph, denoted by $G_{n,k}$, each node represents a set of k points $\{MD_{m,1}, \ldots, MD_{m,k}\}$ of an M-dimensional space, indicating that $MD_{m,1}, \ldots, MD_{m,k}$ are the selected medoids. Two nodes ND_1 and ND_2 are considered as neighbors if their sets differ by only one point. More formally, for

$$ND_1 = \{MD_{m,1}, \ldots, MD_{m,k}\} \text{ and } ND_2 = \{MD_{w,1}, \ldots, MD_{w,k}\}, |ND_1 \cap ND_2| = k\text{-}1 \quad (1)$$

where || is the cardinality of the intersection. It is easy to observe that each node has $k(n - k)$ neighbors. Since a node represents a collection of k medoids, each node corresponds to a clustering and can be assigned a cost (e.g. the total dissimilarity between every point and the medoid of its cluster). Here the cost differential defined in [13] is used for cost estimation of each node. The CLARANS algorithm has two parameters: *maxneighbor* and *numlocal*. The first one determines the maximum number of neighbors that are examined in each iteration, while the second determines the number of local minima that should be searched.

3.3 Structural Elements Selection and Organization

As mentioned, the structural elements of the proposed scheme are key-cluster representatives (L1), cluster representatives (L2) and key-images (L3). Here we describe how these fundamental structural elements are extracted and linked together. Initially a cross correlation criterion is formed as a similarity measure between the feature vectors of 2 images or 2 cluster representatives. More particularly let us denote as $\mathbf{g}_i$ and $\mathbf{g}_j$ two feature vectors corresponding either to a pair of images or cluster representatives. Then the correlation coefficient is estimated by [12]:

$$\rho(\mathbf{g}_i, \mathbf{g}_j) = \frac{C(\mathbf{g}_i, \mathbf{g}_j)}{\sqrt{C(\mathbf{g}_i, \mathbf{g}_i)} \cdot \sqrt{C(\mathbf{g}_j, \mathbf{g}_j)}} \quad (2)$$

with $C(\mathbf{g}_i, \mathbf{g}_j) = (\mathbf{g}_i - \mathbf{m})^T (\mathbf{g}_j - \mathbf{m})$, $\mathbf{m} = \frac{1}{L} \sum_{i=1}^{L} \mathbf{g}_i$

where $C(\mathbf{g}_i, \mathbf{g}_j)$ expresses the covariance of $\mathbf{g}_i$ and $\mathbf{g}_j$, while $\mathbf{m}$ is the average feature vector over all images within a cluster or all cluster representatives within an image set V.

Having selected cluster representatives (subsection 3.2), the next step includes optimal selection of key-cluster representatives from the set of cluster representatives (CR) and optimal selection of key-images from each cluster of the image set. In the following, let us consider that $\mathbf{g}_i$ corresponds either to the ith cluster representative or to the ith image of a cluster. In these cases the correlation coefficient between two cluster representatives (or images) with feature vectors $\mathbf{g}_i$ and $\mathbf{g}_j$, $i \neq j$ is given by equation (2). Let now $U = \{0,1,\ldots, L\text{-}1\}$ be a set, containing the indices of all cluster representatives of an image set or images of a cluster. Let us also assume that K indices of U are selected as key-cluster representatives or key-images by minimizing a cross-correlation criterion using equation (2). In particular, initially an index vector $\mathbf{z}$ is formed, which contains possible indices of the K key-cluster representatives or key-images respectively. That is,

$$\mathbf{z} = [z_1, \ldots, z_K]^T \in Z \subset U^K \quad (3a)$$

where $Z = \{[z_1, \ldots, z_K]^T \in U^K : z_1 < \cdots < z_K\}$ (3b)

is the subset of U^K, which contains only sorted indices $\mathbf{z} \in Z$. Then, the correlation measure among the K key-cluster representatives or key-images is:

$$J(\mathbf{z}) = J(z_1, \ldots, z_K) = \frac{2}{K(K-1)} \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} \rho(\mathbf{g}_{z_i}, \mathbf{g}_{z_j})^2 \quad (4)$$

Now it is clear that searching for a set of K minimally correlated feature vectors is equivalent to searching for an index vector $\mathbf{z}$ that minimizes $J(\mathbf{z})$. As a result, the index vector, say $\hat{\mathbf{z}}$, which contains the elements of the K most representative feature vectors is given by the following equation

$$\hat{\mathbf{z}} = [\hat{z}_1, \ldots, \hat{z}_K]^T = \underset{\text{for all } \mathbf{z} \in Z}{\arg\min} \; J(\mathbf{z}) \quad (5)$$

Minimization of (5) is performed using the algorithm proposed in [12].

Next the extracted components are properly linked to the rest of the image set's elements. More specifically, let us recall that the optimal vector $\hat{\mathbf{z}} = [\hat{z}_1, \ldots, \hat{z}_K]^T$ contains the indices of key-cluster representatives (key-images). Then, the remaining

clusters of the sequence (cluster images) are classified with respect to key-cluster representatives (key-images). This is performed by constructing an influence zone for each $\hat{z}_j$, j=1, 2,…,K of $\hat{\mathbf{z}}$:

$$IZ(\hat{z}_j) = \{i \in U : \rho(\mathbf{g}_{\hat{z}_i}, \mathbf{g}_{\hat{z}_j}) > \rho(\mathbf{g}_{\hat{z}_i}, \mathbf{g}_{\hat{z}_m}) \quad \forall\, m \in \{1,2,\mathrm{L},K\} \text{ and } m \neq j\} \tag{6}$$

where $\mathbf{g}_{\hat{z}_i}$ corresponds to the feature vector of the ith key-cluster representative or key-image. Thus, $IZ(\hat{z}_j)$ contains those indices for which the correlation coefficient of the respective feature vectors is closer to $\mathbf{g}_{\hat{z}_i}$ than to $\mathbf{g}_{\hat{z}_m}$, $m \neq j$.

At the final step the tree structure is constructed, in each node of which a viewing element is placed. Let us denote as $v(\cdot)$ an operator that returns the viewing elements of a node. Then, for the tree-root we have:

$v(V)$= random key-cluster representative (7)

The nodes of level 1 of the tree correspond to key-clusters and thus, the viewing elements are key-cluster representatives. Thus for a key-cluster S_k:

$$v(S_k) = D_k \tag{8}$$

In a similar way, at level 2, each node corresponds to a cluster representative and therefore the viewing element associated to cluster s_k is given by

$$v(s_k) = r_k \tag{9}$$

Level 3 is comprised of image classes G_i, each of which is represented by the respective key-image F_i:

$$v(G_i) = F_i \tag{10}$$

Finally, the viewing elements of level 4 are the images themselves,

$$v(f_i) = f_i \tag{11}$$

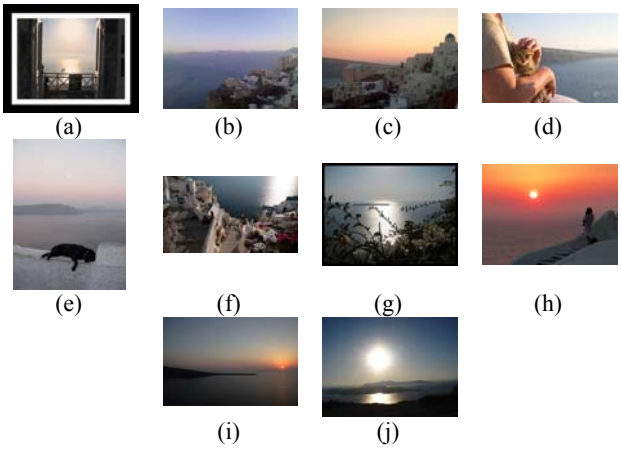

Figure 3: The 10 cluster representatives estimated by CLARANS

4 Experimental Results

Due to presentation limitations and for stressing the capability of specialized search by the proposed system, the query "Santorini Oia Summer Sunset" was submitted to the following image sharing sites: Flickr (www.flickr.com), Picasa Web Albums (picasaweb.google.com) and Smugmug (www.smugmug.com). The respective content (257 images together with their tags) was automatically wrapped, pre-processed and stored locally.

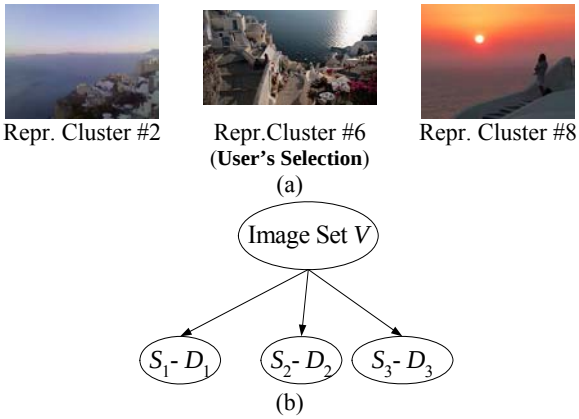

Figure 4: The key-cluster representatives' level for set V. (a) The 3 representative images. (b) The respective decomposition tree.

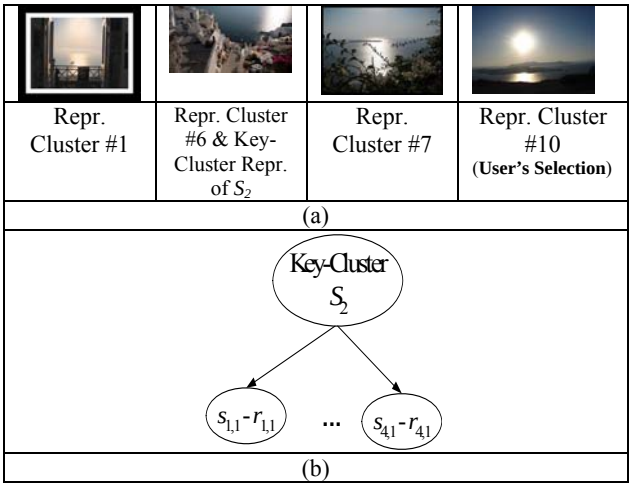

Figure 5: Content decomposition of key-cluster S_2. (a) The cluster representatives belonging to key-cluster S_2. (b) The respective decomposition tree (level 2).

Then the proposed multiresolution visualization module is applied to the 257 wrapped images (set V). Initially the CLARANS spatial clustering algorithm has been applied. Based on the results of [13] *maxneighbor* was set equal to 3.75% of the total number of points of set V, while *numlocal* was estimated for each set according to its dispersion and took values in the interval [1…4]. As a result 10 clusters have been created, the cluster representatives of which are shown in Figure 3. Then, key-cluster representatives are extracted from the set of cluster representatives (CR) by minimizing equation (4), using the proposed stochastic logarithmic scheme. Figure 4(a) shows the 3 extracted key-cluster representatives, while Figure 4(b) presents the tree-structure created in this case.

ISBN: 978-1-61804-074-9

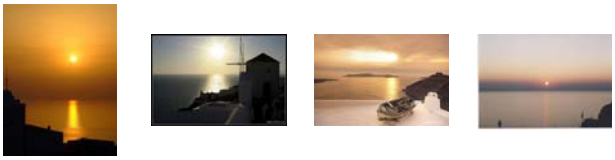

Figure 6: The four key-images of cluster $s_{4,1}$. (level 3).

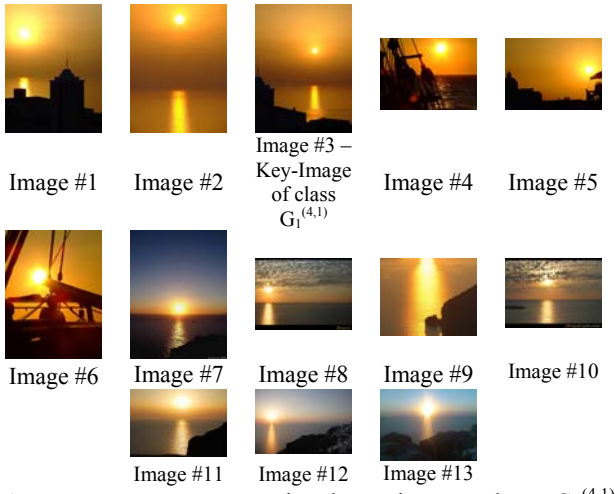

Figure 7: Images associated to image class $G_1^{(4,1)}$ selected by user in Figure 6.

Let us now suppose that a user is interested in the second key-cluster representative, i.e., the D_2, [**User's Selection** in Figure 4(a)]. By selecting this key-cluster representative, the tree structure is expanded as presented in Figure 5(b) and the viewing elements of this key-cluster, associated to the selected key-cluster representative, are presented. Figure 5(a) illustrates the cluster representatives of the clusters associated to the key-cluster S_2. Let us now assume that the fourth cluster of class S_2, denoted as $s_{4,1}$, is selected by the user. Then, the key-images of the selected cluster are presented. In our case, 4 key-images have been estimated, which are depicted in Figure 6. Finally, by selecting the 1st key-image of this cluster, we reach the full resolution of the image set. The respective class of key-image $F_1^{(4,1)}$ contains 13 images, the content of which is shown in Figure 7.

5 Conclusions

The conventional way of seeing the content retrieved by a specific query on a multimedia search engine, is to sequentially scan images presented in several pages. However, this approach is time consuming and usually leads to network congestion.

In this paper, the aforementioned difficulties are addressed using an automatic non-sequential visualization scheme. In particular, a wrapper module initially retrieves web images and feeds them to the content organization module. Then clusters are created and for each cluster a representative is selected. Key-cluster representatives are then determined among the cluster representatives set. Finally in each cluster key-images are selected and the rest of the images are organized based on key-images. The proposed scheme results in a tree structure organization of web multimedia, which is very suitable for interactive navigation, fast browsing and cost-effective transmission of content of interest.

References:

[1] J. Nesvadba, "From push-based passive content consumption to pull-based content experiences," Panel pres. in the *8th IEEE WIAMIS*, Santorini, Greece, 2007.

[2] V.S. Tseng, J.-H. Su, J.-H. Huang and C.-J. Chen, "Integrated Mining of Visual Features, Speech Features, and Frequent Patterns for Semantic Video Annotation," *IEEE Trans. Multimedia*, vol. 10, no. 2, Feb. 2008.

[3] K. Petridis, I. Kompatsiaris, M. G. Strintzis, S. Bloehdorn, S. Handschuh, S. Staab, N. Simou, V. Tzouvaras, and Y. Avrithis, "Knowledge Representation for Semantic Multimedia Content Analysis and Reasoning," Proc. of the 2004 *European Workshop IKSDMT*, pp. 33-46, London, UK, Nov. 2004.

[4] J. Assfalg, M. Bertini, C. Colombo, and A.D. Bimbo, "Semantic annotation of sports videos," *IEEE Multimedia*, vol. 9, no. 2, pp. 52-60, April-June 2002.

[5] N. Tsapatsoulis, and S. Petridis, "Classifying Images from Athletics Based on Spatial Relations," Proc. of the *2nd Intern. Workshop SMAP*, pp. 92-97, Dec. 2007.

[6] S. Gao, D.-H. Wang, and C.-H. Lee, "Automatic Image Annotation through Multi-Topic Text Categorization," Proceedings of the *2006 IEEE ICASSP*, vol. 2, pp. II--II, Toulouse, France, May 2006.

[7] A. Ulges, M. Koch, C. Schulze, and T. Breuel, "Learning TRECVID'08 high-level features from YouTubeTM," In Proc. of *TRECVID 2008*, 2008.

[8] D. Liu, S. Yan, X.-S. Hua, and H.-J. Zhang, "Image Retagging Using Collaborative Tag Propagation," *IEEE Trans. Multimedia*, Vol. 13, No. 4, Aug. 2011.

[9] M. Christoffel, B. Schmitt and J. Schneider, "Semi-automatic wrapper generation and adaption: living with heterogeneity in a market environment," Kluwer Academic Publishers, pp. 60-67, 2003.

[10] B.Liu, R. Grossman, Y. Zhai, "Mining data records in Web pages," Proceedings of the *ninth ACM SIGKDD International conference KDDM*, pp. 601-606, 2003.

[11] N. K. Papadakis, D. Skoutas, K. Raftopoulos, and T. A. Varvarigou, "STAVIES: A System for Information Extraction from Unknown Web Data Sources through Automatic Web Wrapper Generation Using Clustering Techniques," *IEEE Trans. KDE*, Vol. 17(12), Dec. 2005.

[12] N. Doulamis, A. Doulamis, Y. Avrithis, K. Ntalianis and S. Kollias, "Efficient summarization of stereoscopic video sequences," *IEEE Trans. CSVT*, Vol. 10, No. 4, pp. 501-517, June 2000.

[13] R. T. Ng and J. Han, "CLARANS: A method for clustering objects for spatial data mining," *IEEE TKDE*, 14(5), p.p. 1003-1016, 2002.