Video Annotation of TV Content using Audiovisual Information

João Mateus Frederico Malheiro Sofia Cavaco
Nuno Correia
CITI, Departamento de Informática
Faculdade de Ciências e Tecnologia
Universidade Nova de Lisboa
2829-516 Caparica, Portugal
{sc, nmc}@di.fct.unl.pt

Rui Jesus
Multimedia and Machine Learning Group
Instituto Superior de Engenharia de Lisboa
Rua Conselheiro Emídio Navarro, 1
1959-007 Lisboa, Portugal
rjesus@deetc.isel.ipl.pt

Abstract— **Content provider companies and televisions have large video archives but usually do not take full advantage of them. In order to assist the exploration of video archives, we developed ViewProcFlow, a tool to automatically extract metadata from video. This metadata is obtained by applying several state of the art video processing methods applied to a real world challenge: cut detection, face detection and object identification. In addition, we also developed a method to annotate videos with concepts from audio and visual information. The main novelty of this technique is the use of environmental sound recognition to annotate video. The goal is to supply the system with more information such that it has a better understanding of the content and also to enable better browse and search functionalities. This tool has been included in the workflow of a video production company, which confirms its success.**

Keywords- video production; video analysis; media annotation; object identification

I. INTRODUCTION

Nowadays, the content available has a very strong multimedia component. This makes television networks and video production companies rethink the way they produce more and better content, in a fast and convenient way. The overall process of obtaining media from the initial production concepts until the archiving phase can be very time consuming. A more efficient workflow can provide a better management of the available manpower and reduce the overall costs. One possible solution to speed the content production workflow is to reuse footage that is already available, thus reducing the time spent on capturing new footage. Yet, besides, capturing new footage, the editing and annotation stages are other tasks with a major impact on the workflow duration. In order to improve the workflow there is a need to automate its different tasks.

Another problem faced by many video production companies is user subjectivity. Most video annotations are done manually, which is not only a hard and tedious job, but it also introduces the problem of being prone to the user subjectivity.

Nonetheless, many benefits arise if the media content is annotated with semantic metadata including content personalization in interactive TV or media retrieval in digital video archives. Therefore, there is a need for tools that create relevant semantic metadata in order to provide ways to better navigate and search the video archives.

This paper describes a tool that automatically extracts metadata from video and which has been included in the workflow of a multimedia content provider company. The tool analyzes the audiovisual information available in the data in order to extract metadata like scenes, faces and concepts.

Most previous work that combines audio and visual information to annotate video has used information extracted from speech, which can be, for example: recognition of the speech present in the video itself [1, 2], speech recognition of voice annotations [3, 4], or using speech annotation to create audio-visual stories [5, 6]. For more examples and details of work in digital video, the reader can refer to TREC Video Retrieval Evaluation (TRECVID) [7]. On the other hand, here we propose a video annotation tool that combines information extracted directly from the video's environmental sound, such as traffic noise or sounds from crowds, with features extracted from its visual content. A similar strategy was also used by Jiang et al., who used the multiple instance learning technique to construct discriminative audio-visual codebooks [8]. They extract color and texture from local regions of video segments, and audio features from the sound track. Then, they train several concepts using audio-visual atoms. Instead of using audio-visual atoms, we train several concepts separately using only image features and only audio information, and we give the user the option of using only one modality or both.

The remaining paper is structured as follows. The next section presents an overview of the developed tool. Section III introduces the metadata extraction tools used for media annotation. Section IV describes the user interfaces developed to access to the video content and section V discusses the evaluations of the results obtained. Finally, we present conclusions and directions for further development in section VI.

II. OVERVIEW OF THE VIEWPROCFLOW TOOL

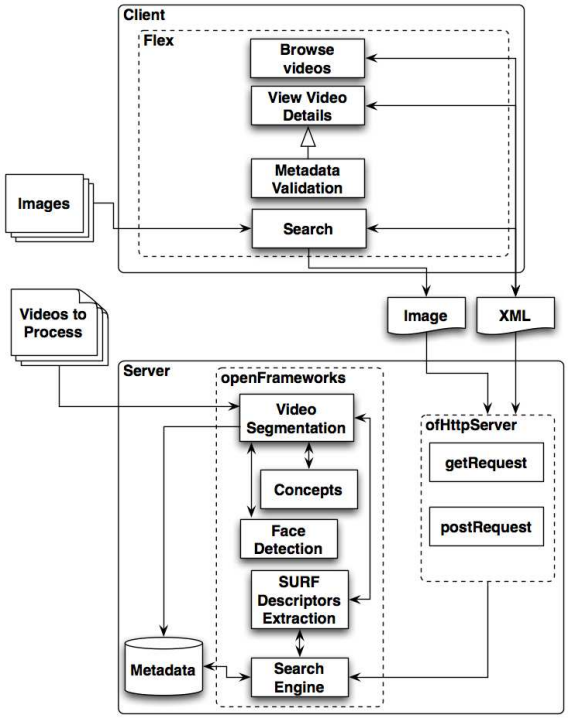
While developing the ViewProcFlow tool, we worked in close proximity with Duvideo, a multimedia content provider

company, which has a video archive with tons of disorganized videos. Working in close proximity with a real video production company gave us the opportunity not only to use real data (from its video archive), but also to interact with video production professionals (such as journalists, screenwriters, producers and directors). We had, therefore, the chance to better understand their needs.

The ViewProcFlow tool includes several search and browse interfaces that facilitate reusing the extracted metadata (for instance, to create the content for a program by defining stories with the edited footage) and therefore contribute to improve the production processes. The tool works with videos recorded in Material eXchange Format (MXF) [9].

The proposed system is split into a Server-side application and a Web Client-side interface. The Server-side application does the video and audio processing and deals with the requests from the Web Client-side. The Client-side application is used to perform several operations on the video archive, such as search and browse, and to visualize the metadata associated with it (see Figure 1). It also provides mechanisms to validate the extracted content. Moreover, with the Web Client, users can access the system wherever they are, not being restricted to a local area or specific software, as they will only need an Internet connection and a browser.

The next two sections explain the major video processing tasks performed in the server (section III) and the client-side with the Web user interface used to search and browse media content (section IV).



architecture.

III. MEDIA ANOOTATION

The server-side application creates hierarchical metadata to describe the videos' contents: their segments, faces, audio and visual concepts, and matching shapes obtained by SURF descriptors. Here we explain how these features are extracted from the videos.

Video segmentation is essential to extract the scenes from the video clips, which are needed in the remaining server components. In order to detect a new scene change, we used a simple difference of histograms [10]. Once the scenes are detected, one *keyframe* is chosen to identify it. We chose to use the middle frame of the shot to represent the whole scene. The frames obtained in this way are the input of the remaining server components.

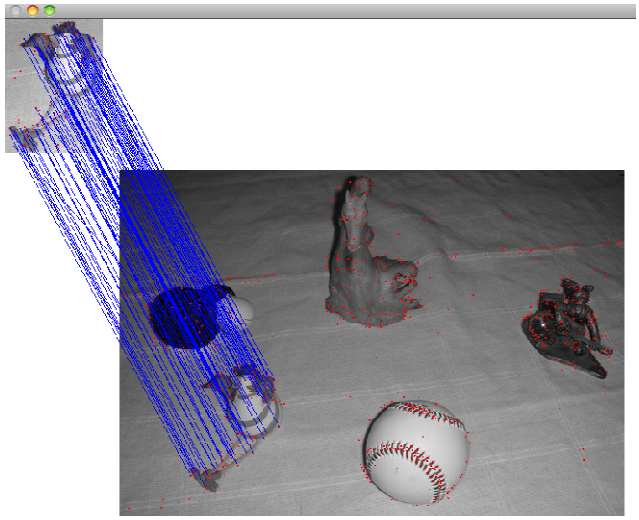
A. *Face Detection*

Faces are pervasive in video content and, therefore, can provide preliminary indexing. We integrated the Viola and Jones algorithm to detect faces that appear in images [11]. This algorithm is based on a set of cascades of previously trained classifiers that inspect image regions.

This algorithm has some limitations, for instance, it does not detect partial faces or faces in a profile view. To overcome these problems we allow the user to be included in the process, such that he/she is able to eliminate the false positives obtained.

B. *Image Descriptors*

The ViewProcFlow tool supports queries that require the comparison of images or image regions. For this purpose, we use the information extracted with the Scale-Invariant Feature Transform (SIFT) and the Speeded Up Robust Features (SURF) [12, 13].



mages with SURF keypoints. Query
iat represents a video (lower right).

These algorithms find keypoints in the images that are invariant to scale and rotation and extract the descriptor that represents the area around the keypoints. This descriptor is used for matching purposes between images, for instance, to find the logo of the TV channel. Figure 2 shows an example of matching keypoints between a query image and a keyframe that represents a video shot. The red dots mark the location of

the SURF keypoints both in the query image and the video frame. The blue lines connect matching points in both images.

C. Semantic Concepts

To automatically annotate video keyframes with keywords describing their content we developed an algorithm based on visual content and audio information. Once the video keyframes are annotated, it is possible to browse a large database of videos based on different concepts and have access from several client applications.

The part of the algorithm that explores visual content gave very satisfactory results when evaluated with standard databases; we obtained a MAP of 0.5 [14]. In addition, in a user study with about 50 users, in average, users classified their level of satisfaction with 4 (in a Liberty-type scale from 1 to 5, where 1 is not satisfactory and 5 is excellent).

To improve this method, here we extended it to include audio information. The metadata used to annotate a video, uses not only visual information but also audio information.

People working at Duvideo usually use a set of categories to access the archives. We selected a subset of the thesaurus used by Duvideo that is also used in ImageCLEF for submissions on "Visual Concept Detection and Annotation Task"[1]. Table I presents these concepts. The following sections present the techniques used to annotate video pictures.

TABLE I. EXAMPLES OF CONCEPTS MATCHED WITH THESAURUS CATEGORIES

Concepts	Thesaurus Category
Car, Bicycle, Vehicle, Train	4816 – Land Transport
Airplane	4826 – Air and Space Transport
Nature, Plants, Flowers	5211 – Natural Environment
Trees	5636 – Forestry
Partylife	2821 – Social Affairs
Church	2831 – Culture and Religion
Food	60 – Agri-Foodstuffs
Fish	5641 – Fisheries
Baby, Adult, Child, Teenager, Old Person	2806 – Family, 2816 – Demography and Population
Mountains, River	72 – Geography

1) Visual Information

Each image is represented by visual features, which are automatically extracted. The image representation consist of the Marginal HSV color Moments and features obtained by applying a bank of Gabor Filters [14].

In order to classify visual information, we use a Regularized Least Squares (RLS) classifier that performs binary classification on the database (e.g., Indoor versus Outdoor or People versus No People) [14]. It also uses a sigmoid function to convert the output of the classifier into a pseudo-probability. Each concept is trained using a training set composed of manually labeled images with and without the concept. After estimating the parameters of the classifier (that is, after training), the classifier is able to label new images. Using this classifier, the tool was capable of executing interesting queries like "Beach with People" or "Indoor without People".

2) Audio Information

As mentioned above, the main novelty of this work is the combination of visual and audio information to annotate video, where the audio information is extracted from the environmental sound and not from speech. In order to recognize concepts from audio information we developed a recognizer that uses non-negative matrix factorization (NMF) [15] to learn spectral features that are then fed to a *k*-NN classifier. This recognizer consists of two modules as depicted in Figure 3: the training and testing modules, which we describe below.

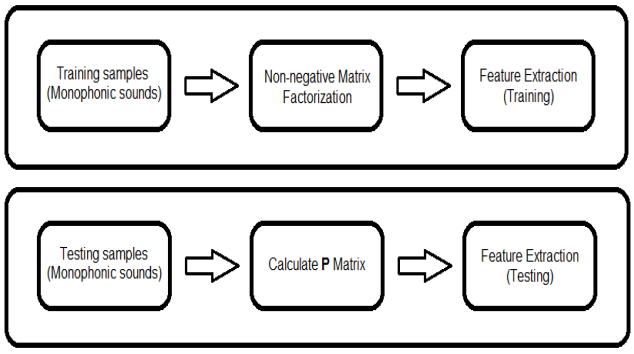


Figure 3. Audio information. The training module (top) and testing module (bottom).

First, we need to define which features are used to classify the data and we need to represent the training data with those features. Instead of using a set of predefined features, here we use NMF to learn a set of features that is appropriate to separate the data.

To start with, the training data samples are transformed into a more suitable representation: The amplitude of the signals is normalized, which guarantees that all the sounds have a similar volume level, reducing the possibility of discrepancies in terms of spectral properties. Finally, from each sound sample, we extract 20 segments of 0.2 seconds uniformly throughout its length (if the sound is less than 4 seconds long, the segments will overlap) and calculate the magnitude spectrogram of these segments.

The magnitude spectrograms for all the sound segments are then concatenated into a unique matrix **S**, which will be used as the input for the next step, the NMF. The NMF algorithm requires the definition of two parameters: the cost function, for which we use a divergence function, and the update rule, for which we use a multiplicative update.

Given a non-negative matrix **S**, NMF calculates two, also non-negative, matrices $\boldsymbol{\Theta}$ and **P**, such that

[1] ImageCLEF, http://www.imageclef.org/2010, June 2010.

$$\mathbf{S} = \mathbf{\Theta} \times \mathbf{P}, \quad (1)$$

where matrix $\mathbf{\Theta}$ is the mixing matrix (whose columns contain the spectra that characterize the sounds in the training set, and which are the axis of the new space where the data is now represented) and $\mathbf{P}$ is the source signal matrix (whose lines contain time-varying functions that describe how each spectra in $\mathbf{\Theta}$ contributes to the whole signal – each value in $\mathbf{P}$ specifies a weight, or coordinate, associated to one of the spectra in $\mathbf{\Theta}$). $\mathbf{P}$ is then processed in the feature extraction phase, and $\mathbf{\Theta}$ will later be crucial for the testing phase.

Now, each column of matrix $\mathbf{S}$ (that is, each time frame of the sequence of spectrograms) is represented by a point in the new space defined by matrix $\mathbf{\Theta}$. This point is a (column) vector of coordinates extracted from matrix $\mathbf{P}$. Therefore, each 0.2 seconds segment is represented by a sequence of points in the new space. We create a feature vector of that segment by calculating the average value and median of those points, as well as the spectral energy (the sum of the values). This process produces a training matrix, $\mathbf{F}_{training}$, composed of the feature vectors for each of the individual sound segments.

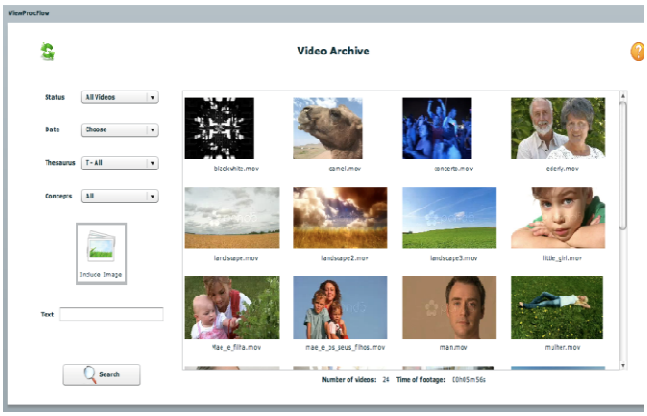
After computing $\mathbf{F}_{training}$, the recognizer is ready to classify new data. The samples that will be processed for testing can be either individual sounds or sequences of sounds. The initial audio processing of these sounds is the same as that of the training stage.

The testing features values are extracted from a matrix $\mathbf{P}_{test\_sound}$ that we obtain from a test sample. However, in this stage we do not use NMF to compute this matrix. Instead, this is obtained by the following equation

$$\mathbf{P}_{test\_sound} = \mathbf{\Theta}^{-1} \mathbf{S}_{test\_sound} ; \quad (2)$$

where $\mathbf{\Theta}^{-1}$ is the pseudoinverse of $\mathbf{\Theta}$ and $\mathbf{S}_{test\_sound}$ is the spectrogram of the test sound. Similarly to what is done in the training phase, the test sound is segmented and each segment is represented by a sequence of points represented in $\mathbf{P}_{test\_sound}$. Again, the feature vector of each segment consists of the average and median of the sequence of points, as well as the spectral energy. These feature vectors are then gathered to create matrix $\mathbf{F}_{testing}$.

The system uses a *k*-NN classifier with a Euclidean distance metric. A matrix with the *k* nearest neighbors is calculated for each test sound, where *k* is determined dynamically by the formula $\text{ceiling}(\sqrt{\text{mean}(\text{sounds})})$, where *sounds* is a vector with the number of training samples for each training class. The class of a test segment is assigned by its most occurring neighbors, and the class of a complete sound is determined by the most occurring class.



ı for media in the digital video archive:
that represent videos and at the left the

IV. USER INTERFACES

The work developed results into a graphic user interface with several functionalities that allow the users to access the archives. As an illustration, we describe two examples of the windows used in the developed interface.

In order to take the most of the metadata produced, the interface for its visualization and manipulation is a key part of the system. Preliminary specifications were done based on input from the potential users and as a result, we developed an interface that is divided in two main groups of functionalities: browsing and searching. The interface starts with an overall view of the whole video archive on the right side and the main search parameters on the left (see Figure 4).

The following search options are available:

- Date: “Before” a specific date, “After” a specific date or “Between” two dates.
- Status: “Valid Videos”, “Videos to Validate”. Once the videos are processed on the server, they are labeled as “Videos to Validate”. After the user approves the metadata, the video is made “Valid”. This option allows having some feedback from the users.
- Thesaurus: a set of categories to identify the context of the video such as “Science and Technology”, “Art”, “Sports”, “History” among others, as presented above in section C.
- Concepts: a second set of options to identify concepts such as “Indoor”, “Outdoor”, “Nature”, and “People”. The user can choose to use visual concepts or audio concepts.
- Image: the possibility of conducting an image-based search (as illustrated by Figure 2).
- Text Input: searches into the annotations, titles and all textual data stored with the video.

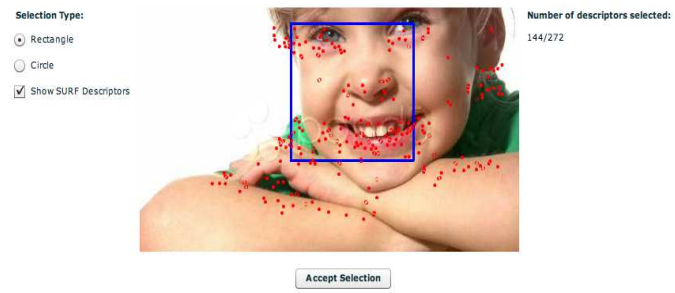


Figure 5. Image editor.

In case that the user wants to add an image to the search, it is possible to use one that already is on the library of images or upload one to the server. However, sometimes the image chosen has more elements than the user wants and for that, we provide a simple editor that allows the user to select the area of interest, which can be either a rectangle or a circle (see Figure 5, where the crop area is marked by the blue line). Since the image search is based on the SURF descriptors, these are marked as red dots in the editor (see Figure 5), in order to provide the user with a visual aid to assist him/her in choosing the crop area (as the user may want to avoid choosing areas with few or none descriptors).

When the search is performed, the library view will be updated with the current results (see the keyframes in Figure 4). A popup window will appear once the user selects one of the videos from the results (see Figure 6). This new window contains a visualization screen that allows the user to observe the video.

The extracted metadata (faces, scenes, concepts) is organized in *timelines* and when one type is selected; all the corresponding data (from the scenes that got a hit in the search) will appear and will be used as anchors (i.e., shortcuts) to its position on the video, thus facilitating to have direct access to the corresponding scenes. In the example shown in Figure 6, the chosen metadata are scenes and faces, and this metadata (whole scene or faces extracted from the scene) appears at the bottom of the window. In order to give a better perspective of where the data occurs, a timeline appears below the video with marks showing the locations of all the scenes that got a hit from the search.

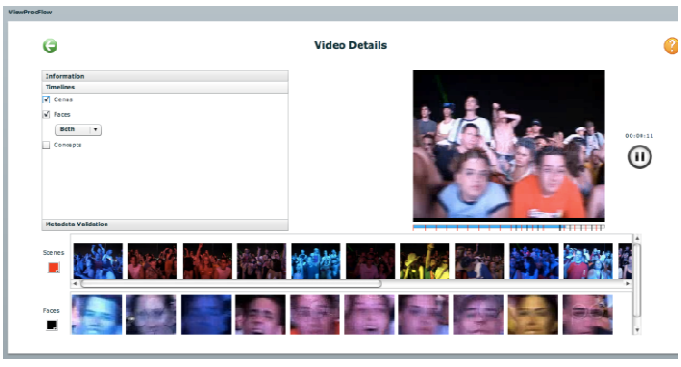


Figure 6. Image Visualization.

V. RESULTS

We received very positive feedback from the Duvideo professionals who evaluated our tool. They reported that the tool was easy to use, gave very satisfactory results and was a better solution to access the data than the solution they used previously. Nonetheless, to have a more precise evaluation, we also measured the accuracy of a set of queries. Below, we present the later results.

As mentioned above, even though we could have used standard databases in the evaluation process, we wanted to use real data. In order to evaluate our algorithms, we used manually labeled data from Duvideo's archive and we compared the tool's results with the manual labels. Since building the manually labeled database was a very time consuming task, so far, we have evaluated three concepts ("people", "car" and "water") using a database with about 1000 keyframes. The results obtained using visual information and audio information are presented in tables II and III respectively.

TABLE II. IMAGE ANNOTATION RESULTS: MEAN AVERAGE PRECISION (MAP) OBTAINED FOR THREE CONCEPTS USING ONLY VISUAL INFORMATION.

Concepts	Visual Information (%)
"people"	82
"water"	69
"car"	83
MAP	**78**

TABLE III. IMAGE ANNOTATION RESULTS: PRECISION OBTAINED FOR THREE CONCEPTS USING ONLY AUDIO INFORMATION.

Concepts	Audio Information (%)
"people"	77
"water"	30
"car"	53
Mean	**53**

The results obtained with visual information are better than the results obtained with audio information. The main reason is related to the lack of training data. While we use hundreds of images to train the classifier that uses visual features, the method that uses audio only uses dozens of audio clips for the training task. This lack of data has more relevance in the results obtained by the concept "water" because we have many different sounds related to the water (e.g., rain, river or ocean). The sound environment provided by people talking is very characteristic and that is the main reason why the concept "people" presents the best result using audio information. Although these audio results are not better than the visual results they are useful because they are obtained with different data. Therefore, we believe the combination of visual information with the audio data will increase the results.

VI. CONCLUSIONS AND FUTURE WORK

Here we present a tool that uses audiovisual information to annotate video, and that has been included in the workflow of a video production company. The metadata is hierarchical and contains information about the video segments, faces, visual concepts, audio concepts and matching images obtained with SURF keypoints. We use several state of the art video processing methods for segmentation, face detection and object

matching. The main novelty of this technique is the use of environmental sound recognition to annotate video. The audio is processed with NMF of the spectrograms of the sounds, which allows not only to separate the data present in the audio but also to characterize it with spectral properties that adapt to the data used for training.

There are also some functionalities that would enrich the current version of the application. For example, while the tool includes a face detection algorithm, it would also be useful to have a face recognition functionality A first step, could be gender classification, a technique describe in [16]. Also since most professionals use the proposed tool to search for data that then is edited to create new stories, the users could benefit from the possibility of doing some level of video edition and creating new stories by cutting and joining scenes, all using the same application. Finally, we are now exploring ways of jointly using audio and visual information in the classification process. So far, that information is used separately to create annotations of visual concepts and audio concepts, but using both together would improve the concepts classification performance.

TABLE IV. RUNNING TIME FOR SEVERAL TASKS.

Task	Average Running Time
Histogram Difference Between Two Images	0.17s
Face Detection	0.02s
SURF Descriptors Extraction	0.94s
Matching Two Images (400 descriptors)	0.21s

Regarding the existing tasks, increasing the performance with the introduction of parallel computing could lead to better results (Table IV shows the average running time of several tasks from the tool, which would benefit from parallel computing). In a similar way, the usage of a native XML database, like sedna[2], will help on accessing data and executing queries for the textual parameters.

VII. ACKNOWLEDGMENTS

This work is part of the Project VideoFlow (3096) funded by QREN and by *Fundo Europeu de Desenvolvimento Regional* (FEDER) through the *Programa Operacional de Lisboa*.

Our thanks to Duvideo for all the help and cooperation on all the life cycles of the project, from requirements, elicitation to the validation of the prototype.

VIII. REFERENCES

[1] V. Tseng,.; J.-H. Su; J.-H. Huang; C.-J. Chen, "Integrated Mining of Visual Features, Speech Features, and Frequent Patterns for Semantic Video Annotation". In IEEE Transactions on Multimedia, vol. 10, issue 2, pp. 260 – 267, 2008.

[2] L Xie, L Kennedy, S.-F. Chang, A. Divakaran, H. Sun, C.-Y. Lin, "Discovering meaningful multimedia patterns with audio-visual concepts and associated text". In IEEE ICIP 2004: 2383-2386.

[3] K. Rodden and K. Woods, "How do people manage their digital photographs?," in SIGCHI Conference on Human Factors in Computing Systems, 2003, pp. 409–416.

[4] T. Mills, D.Pye, D Sinclair, K. Wood, "SHOEBOX: a digital photo management system", Technical Report, AT&T Research, 2000.

[5] T. Jokela, J. Lehikoinen, and H. Korhonen. "Mobile multimedia presentation editor: enabling creation of audio-visual stories on mobile devices". In CHI '08: Proceeding of the twenty-sixth Annual SIGCHI Conference on Human Factors in Computing Systems, Florence, Italy, pages 63–72, 2008.

[6] D. Frohlich and M. Jones. "Audiophoto narratives for semi-literate communities". In Interactions, 15(6):61–64, ACM, 2008.

[7] P. Over, G. Awad, M. Michel, J. Fiscus, W. Kraaij, A. Smeaton, "TRECVID 2011 - An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics". In Proceedings of TRECVID 2011

[8] W. Jiang, C. Cotton, S.-F. Chang, D. Ellis, and A. Loui, "Audio-visual atoms for generic video concept classification". In ACM Trans. Multimedia Comput. Commun. Appl., vol. 6, no. 3, 2010.

[9] B. Devlin, J. Wilkinson, M. Beard, and P. Tudor, The MXF Book: Introduction to the Material eXchange Format. Elsevier, March 2006.

[10] A. Dailianas, R. Allen, and P. England, "Comparison of automatic video segmentation algorithms". In SPIE Photonics West (1995), pp. 2–16.

[11] P. Viola, and M. Jones, "Robust real-time object detection". In International Journal of Computer Vision (2001).

[12] D. Lowe, "Distinctive image features from scale-invariant keypoints". In International Journal of Computer Vision 60, 2 (2004), 91–110.

[13] H. Bay, T. Tuytelaars, and L. Gool, "Surf: Speeded up robust features". In ECCV (1), pp. 404–417, 2006.

[14] R. Jesus, A. Abrantes, N. Correia, "Methods for automatic and assisted image annotation". In Multimedia Tools and Applications, volume 1380:7501, Springer Netherlands, pp. 1-20,2010.

[15] D. Lee and H. Seung. "Algorithms for non-negative matrix factorization" .In NIPS, pages 556–562, 2000.

[16] F. Grangeiro, R. Jesus, and N. Correia, N, "Face recognition and gender classification in personal memories". In Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE Computer Society, pp. 1945–1948, 2009.

[2] MODIS, sedna - Native XML Database System. http://modis.ispras.ru/sedna/.