SHIATSU: Semantic-Based Hierarchical Automatic Tagging of Videos by Segmentation using Cuts*

Ilaria Bartolini
DEIS – Alma Mater Studiorum
Università di Bologna
Italy
i.bartolini@unibo.it

Marco Patella
DEIS – Alma Mater Studiorum
Università di Bologna
Italy
marco.patella@unibo.it

Corrado Romani
DEIS – Alma Mater Studiorum
Università di Bologna
Italy
corrado.romani@unibo.it

ABSTRACT
The recent dramatic widespread of multimedia content over the Internet and other media channels (such as television or mobile phone platforms) points the interest of media broadcasters to the topics of video retrieval and content browsing. Semantic indexing based on content is a good starting point for an effective retrieval system, since it allows an intuitive categorization of videos. However, the annotation process is usually done manually, leading to ambiguity, lack of information, and translation problems. In this paper, we propose SHIATSU, a novel technique for automatic video tagging which is based on shot boundaries detection and hierarchical annotation processes. Our shot detection module uses a simple yet efficient algorithm based on HSV histograms and edge features. The tagging module assigns semantic concepts to both shot sequences and whole videos, by exploiting visual features extracted from key frames. We present preliminary results of our technique on the *Mammie* platform video set by proving its effectiveness in real scenarios.

Categories and Subject Descriptors
I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding – *Video analysis*

I.4.6 [**Image Processing and Computer Vision**]: Segmentation – *Edge and Feature Detection*

H.3.1 **[Information Storage and Retrieval**]: Content Analysis and Indexing – *Abstracting methods*

General Terms
Algorithms.

Keywords
Shot boundary detection; video tagging; semantic concepts.

1. INTRODUCTION
The emergence of multimedia sharing web portals (such as Youtube [1] and Google Videos [2]) and pay-per-view services over the Internet, mobile phones, and television is dealing more and more with the problem of correctly indexing and categorize a huge amount of video data in order to allow the development of efficient and user-friendly browsing tools. The main challenge is to find an effective way to retrieve needed information from large collections of videos (such as visual content and genre) in a (semi-)automatic way, in order to allow a generic user to correctly retrieve videos of interest. Since it is not always possible to analyze the context where the video is enclosed (which could be missing or simply unrelated to the information contained in the video), many studies focus on video retrieval based on visual content [3][4]. Literature on the topic tends to decompose the problem into two major tasks: *shot boundaries detection* and *video semantic indexing*. Shot detection has been an important topic of video retrieval research throughout the last 20 years [3][5], aiming to divide a video into scenes sharing the same visual features in order to help the content analysis.

This goal is achieved by detecting hard cuts and gradual transitions (such as fade-in/fade-out, wipe, and dissolve), a task which involves the computation of visual differences between consecutive frames and the subsequent application of criteria to declare the presence of a shot cut (see Figure 1 for an example).

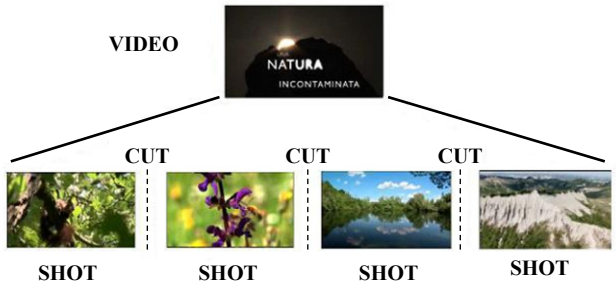

Figure 1. Shot boundaries detection example.

On the other hand, video semantic indexing has the purpose of extracting semantic concepts from video data in order to provide a compact description, useful to categorize video content. Towards this end, image retrieval techniques applied on the key frames are commonly used to detect the meaning of video scenes and generate an appropriate description [6][7]. At present, the most common way to express this description is represented by video annotation (or tagging) techniques, aiming to express the video content through a series of textual labels. In this way, the semantic concept conveyed by each textual label can be transferred to the video scene (refer to Figure 2 for an example). In this paper, we pursue this approach, focusing on video tagging techniques.

In this context, we note that segmenting a video into a sequence of key frames for video tagging purposes is inherently different with respect to other video segmentation tasks. For example, a great amount of accuracy in detecting the precise timing of a transition is not required, since we basically want to identify the shots carrying the visual "content" of a sequence.

* This work was partially supported by the CoOPERARE MIUR Project

AIEMPro'10, October 29, 2010, Florence, Italy.

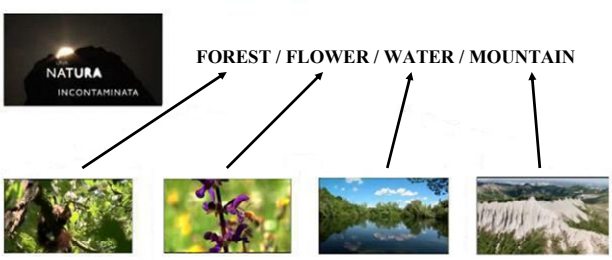

Figure 2. Video tagging example.

In this paper we present SHIATSU (Semantic HIerarchical Automatic Tagging of videos by Segmentation Using cuts), a complete video processing system which offers an accurate description of video content (Section 3). A video submitted to SHIATSU is first segmented into shots (Section 3.1): we chose a balanced approach for the shot detection process in order to mediate between effectiveness, efficiency, and ease of implementation. Then, from each sequence of shots, a number of key frames are identified and submitted to the tagging module (Section 3.2), producing a number of candidate tags for the sequence. The user can then review such tags or accept them. Finally, the whole video is hierarchically tagged using a weighted propagation of shot tags. The user can then search videos using both video and shot tags (see Figure 3). We experimentally validate the performance of SHIATSU using the *Mammie* platform video set for the Item Segmentation task (Section 4) and discuss future improvements and real use scenarios for our prototype system (Section 5).

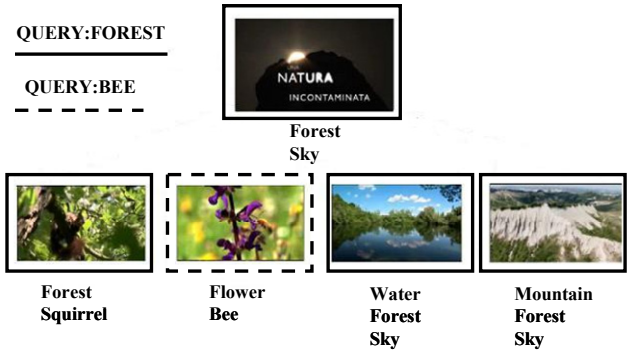

Figure 3. An example of shot tags propagation.

With respect to other existing automatic video tagging tools, SHIATSU only considers the visual content of each video in order to suggest description labels: this can be easily complemented with other (more complex) techniques that analyze audio and speech so as to provide a more accurate tagging of videos. For the moment, we stress the fact that (as we will show in the experimental section) SHIATSU is able to obtain a good tagging accuracy, in spite of its simplicity and efficiency.

2. RELATED WORK

Selecting the appropriate features is crucial to guarantee the effectiveness and efficiency of a shot detection algorithm: every frame is described by a vector of properties and differences between those vectors represent differences in content for video frames. Color histograms [8][9][10][11] are widely used to describe frames, because they are easy to compute and mostly insensitive to translational, rotational and zooming camera motions, hence being quite effective to represent frame content difference.

Edges extracted from frame objects [8][9][10][12], although more computational-intensive than color histograms, are also commonly used to detect scene changes with good results, often in conjunction with color features. We chose to limit ourselves to a combination of this two features: the use of other features, like motion vector detectors [8][9][13] or wavelet filters [14][15], might improve the accuracy of video segmentation at the cost of a much higher computational overhead; as said, the shot detection task is, in our case, only half of the picture, our video tagging results being sufficiently accurate even without a perfect segmentation. Deciding whether a cut exists between two consecutive frames is usually carried out by comparing features of the two frames using a threshold [8][9][10][12][13], while other works rely on clustering techniques [14][15][16]. Fixed thresholds generally cannot deal well with different types of videos, while clustering performs poorly when the differences calculated around probable cuts are not very high compared to the mean of the differences of non-cut frames, since grouping the frames in two clusters (normal frames and cut frames) could lead to miss a large number of shots (false negatives). The best available option seems to be dynamic thresholding, i.e., calculating a threshold value based on the content of each processed video. An interesting approach is the one described in [10], which uses edge features and HSV histograms with a dynamic threshold system. Although the performance exhibited by the system are adequate in most cases, the similarity metric used in [10] to evaluate HSV histograms and the proposed edge features led in some other cases to an excessive number of false positives (i.e., detecting a transition between shots belonging to a same scene). We therefore modified the approach presented in [10] to improve the accuracy of shot detection, while keeping the complexity of the process at reasonable levels.

Video tagging has become a popular field of study due to the emergence of video sharing platforms such as YouTube [1]: the task of assign tags to content is usually left to users who upload videos, creating problems of ambiguity, lack of information, and translation. The task of (semi-)automatically tagging videos based on semantic concepts [17][18] is usually carried out using annotation of key frames [4][7][19][20][21], so as to detect and categorize objects inside video scenes. Image annotation aims to overcome the so-called "Semantic Gap" [17][22], i.e., the distance between visual features and the image/video meaning, which could lead to misinterpretation of the extracted data.

3. OVERVIEW OF SHIATSU

SHIATSU consists of two main components: the Shot Detection Module and the Video Tagging Module. They work sequentially (Figure 4) to segment a video into coherent frame sequences and to attach semantic concepts to them. Shot tags are then propagated to the whole video, so as to obtain semantic indices for both the video and its shots: this allows the realization of a hierarchical, two-level browsing platform. When a video is processed, the shot detection module analyzes its frames and computes its shot boundaries, marking their timestamps (begin and end of each shot). Every detected shot is then analyzed by the tagging module, which automatically assigns tags depending on the visual content of the shot; the user can then review the proposed tags and possibly modify the annotation results. After processing all the shots, the module selects the most appropriate tags for the whole video and saves all the information into a database.

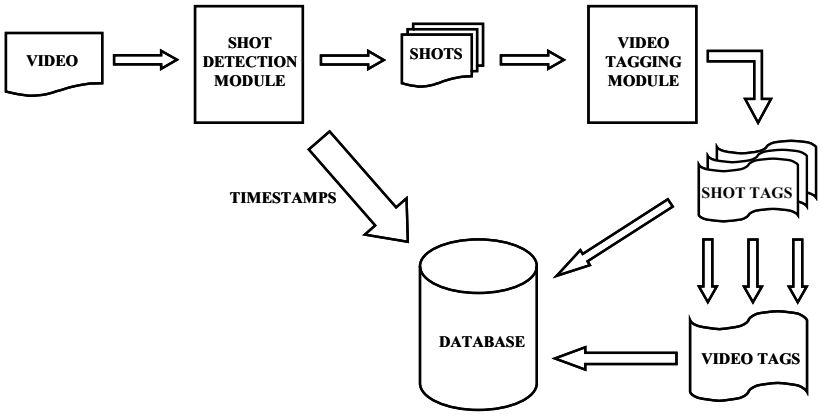

Figure 4. Tagging of a video using SHIATSU.

3.1 Shot Detection

Our shot detection algorithm computes color histograms and object edges of every frame and uses such features to compare consecutive frames by applying two different distance metrics: this is because usually a shot transition produces a change in both the color and the texture structure of the frames. Shot selection process is done with a double dynamic threshold system which takes into account video content in order to adapt to different video types: frames are filtered on their color features and then on their edge features. In the following, we will detail how videos are segmented in SHIATSU, highlighting differences from the approach described in [10].

HSV Color Histogram

We characterize the color information of each frame as the distribution of HSV values of its pixels: each frame is represented using 3 histograms (we have 12 bins for Hue, 5 for Saturation and 5 for Value). The difference between histograms of two frames is computed using a L_1 bin-to-bin metric [24]:

$$d_{b2b}(h_1, h_2) = \frac{1}{2N} \sum_i | h_1[i] - h_2[i] | \qquad (1)$$

where N denotes the number of frame pixels. From (1), the HSV distance between two consecutive frames k and k+1, $d_{HSV}(k, k+1)$, is defined as the sum of bin to bin difference for the three HSV histograms:

$$d_{HSV}(k, k+1) = \frac{1}{6N} \sum_{i=1}^{22} | h_k[i] - h_{k+1}[i] | \qquad (2)$$

With respect to [10], where an histogram intersection formula is used to measure frame similarity, Equation (2) allows a better discrimination between cut and non-cut frames, because distance values between non-cut frames generally remain in the same range throughout the whole video, thus avoiding having distance values whose average changes sensibly between different parts of a video, leading to problems during the cut selection phase.

Edge Change Ratio

The edges of frame objects usually change across a shot boundary. Exploiting this fact, we can compute the percentage of entering (new) edges and the percentage of exiting edges those edges which are present in the actual frame but not in the next frame. We compute edge pixels using a Canny filter [25] and determine entering and exiting edges by analyzing the difference of edge pixels between frames: edge pixels which are 3 pixels away from edges in the previous/next frame are not counted as changed edges to compensate for object motion into the scene. The Edge Change Ratio (ECR) [8] between frames k and k+1 is calculated as follows:

$$ECR(k, k+1) = \max\left(\frac{X_k^{IN}}{\sigma_k}, \frac{X_{k+1}^{OUT}}{\sigma_{k+1}}\right) \qquad (3)$$

where σ_k is the number of edge pixels in frame k, X_k^{IN} and X_{k+1}^{OUT} are the entering and exiting edge pixels in frames k and k+1 respectively. In [10] edge density and average gray variation are used to detect object edges: we believe that the use of ECR could help in better defining frame differences and avoiding a large amount of false positive cuts in the result.

Cut Selection Process

Once all the frames have been processed (note that this has a O(N) complexity, since each frame is only compared with the next frame), the system calculates a double threshold based on frame features. To determine the HSV threshold θ_{HSV}, SHIATSU computes the mean of the highest $M - (M/fr)$ HSV distances:

$$\theta_{HSV} = \frac{\beta_{HSV}}{M - (M/fr)} \sum_{i=M-(M/fr)}^{M} L_{HSV}(i) \qquad (4)$$

where β_{HSV} is a sensitivity parameter (the default value is 1), M is the total number of video frames, fr is the video framerate, and L_{HSV} is the ascending ordered list of HSV distances among all consecutive shots. To determine the ECR threshold θ_{ECR}, the average of the highest $M - 2(M/fr)$ ECR values is computed:

$$\theta_{ECR} = \frac{\beta_{ECR}}{M - 2(M/fr)} \sum_{i=M-2(M/fr)}^{M} L_{ECR}(i) \qquad (5)$$

where β_{ECR} is a sensitivity parameter (the default value is 1) and L_{ECR} is the ascending ordered list of ECR values. We consider a larger window of data for the computation of θ_{ECR} because ECR distances are usually sparser than HSV distances. Our algorithm filters out all frames having HSV distance $\leq \theta_{HSV}$ and considers ECR values of the remaining candidates: they are again

excluded from the result if ECR $\leq \theta_{ECR}$; the frames that exceed both thresholds are considered as shot cut frames. The approach described in [10] uses two different dynamic thresholds to detect abrupt shot cuts and gradual cuts: as we will show in the experimental section, SHIATSU is able to detect both hard and gradual cuts and is also easier to implement.

3.2 Video Tagging

The tagging module of SHIATSU exploits the Imagination system [19] and uses a set of pre-annotated images as a knowledge base. The system extracts a set of visual features from each image and saves the information in a database, indexing them efficiently with an implementation of the M-Tree metric index [23]. The semantic concepts in the knowledge base can be organized into either a tree-shaped taxonomy (where terms are linked with a father/child relationship) or as a flat structure (all terms at the same level) and can be easily modified and expanded. When provided with a key frame to be labeled, the tagging module extracts its visual features, exploits the M-Tree index to efficiently retrieve images having similar features and proposes semantic concepts depending on the similarity of the shot with the images in the knowledge base (more details can be found in [19]). Every time a new image/shot is processed and tagged, its information are inserted into the database, hence improving the system accuracy and quality.

Shot Tagging

Using the cuts timestamps, SHIATSU extracts a set of key frames as representatives of each shot sequence, computes their visual features and compares them with those contained in the knowledge base. The module suggests a set of concepts for each key frame: only terms recurring in the majority of key frames are selected as suitable concepts to describe the whole shot sequence. The number of key frames N_k processed for every shot s depends on the shot length:

$$N_k(s)=l(s) / fr \tag{6}$$

where $l(s)$ represents the shot length, so the algorithm takes approximately 1 key frame for every second of the shot. To avoid producing an overwhelming number of tags for each shot, only the most frequent tags retrieved for each key frame in the sequence are maintained (the total number of tags for each shot is also limited to 6). The proposed tags can then be reviewed by the user and, in case, are finally stored into the database.

Hierarchical Tagging

Shot tags are useful to browse sequences across different videos, but they could be too specific to index a whole video, especially if this contains a large range of different visual content. A simple criterion to select video tags from the set of shot tags is to weigh every tag depending on its frequency and the length of the shot it is associated to. We first compute the relevance of each shot s as the length of s with respect to the whole video V:

$$W(s) = l(s) / l(V) \tag{7}$$

Then, we define the rank $R(t)$ for every shot tag t as:

$$R(t) = \frac{1}{N_s} \sum_s W(s)A(t,s) \tag{8}$$

where N_s is the total number of shots and $A(t,s)$ is 1 if shot s contains tag t, and 0 otherwise. Tags are ordered by descending $R(t)$ values and the first 10 tags (if available) become video tags.[1] The rationale behind the proposed propagation method relies on the fact that concepts extracted from long shot sequences and/or that appear in several shots are probably more relevant, to describe the content of a whole video, than concepts occurring rarely or in short sequences. Finally, both shot and video tags are stored in the SHIATSU database, thus the user can exploit both of them when searching for videos of interest.

4. EXPERIMENTAL RESULTS

We present here results obtained by applying SHIATSU to the video set for the Item Segmentation task of the *Mammie* system (http://media.ibbt.be/mammie). The benchmark consists of 43 videos containing a total of 1.935.000 video frames (4225 shot cuts) for a playback time of over 1075 minutes. Regarding efficiency, SHIATSU is quite inexpensive, processing frames at a rate of 15 FPS for the segmentation task on an AMD 3.1 GHz Dual Core processor with 2GB of RAM running the Windows 7 Professional OS: the tagging task requires around 1 sec for each key frame; since around 1 key frame for second is considered for tagging (see Section 3.2), the time needed to tag a video is about equal to the video length.

4.1 Shot Detection

We evaluated the performance of the shot detection module of SHIATSU using classical precision/recall metrics:

$$P = \frac{CC}{CC + FC} \qquad R = \frac{CC}{CC + MC} \tag{9}$$

where CC denotes the number of correct cuts detected, MC the number of missed cuts, and FC the number of false cuts detected, respectively. We accept a tolerance of 250 milliseconds from shot boundaries timestamps indicated in the ground truth to declare a correct cut. We compare results of SHIATSU with those obtained by the technique proposed in [10] (named QLR in the following). The central columns of Table 1 show the results of SHIATSU using default and optimal sensitivities values, the last column shows the performance of the QLR technique [10].

Table 1. Video segmentation results.

	SHIATSU		QLR
parameters	β_{HSV}=1 β_{ECR}=1	β_{HSV}=1.2 β_{ECR}=1	—
CC	3784	3721	3625
FC	672	494	1612
MC	441	504	600
recall %	89.56	88.07	85.8
precision %	84.92	88.28	69.22

SHIATSU clearly outperforms the reference algorithm in both recall and precision and achieves good overall values with default and optimal sensitivity values. The use of bin to bin differences for HSV histograms and of ECR for frame object edges dramatically reduce the number of false positive shot boundaries detected: experimental results evidence how our choice of image features leads to significantly better performances. We also evaluated how recall and precision

[1] We note here that the total number of tags for each shot and for the whole video are input parameters that can be easily modified by the user, without any significant impact on the system performance.

values change when modifying the sensitivity parameters β_{HSV} and β_{ECR} (Figure 5 and Figure 6).

As expected, increasing sensitivity parameters leads to increasing values of precision and decreasing values of recall (since we obtain higher thresholds, thus a lower number of cuts). After tweaking sensitivity parameters, the most balanced setting seems to be β_{HSV}=1.2 and β_{ECR}=1 when the segmentation algorithm achieves a recall of 88.07% and a precision of 88.28%; we however note that SHIATSU is quite robust in performance since a variation of 20% in sensitivity parameters does not lead to a dramatic drop in neither recall nor precision.

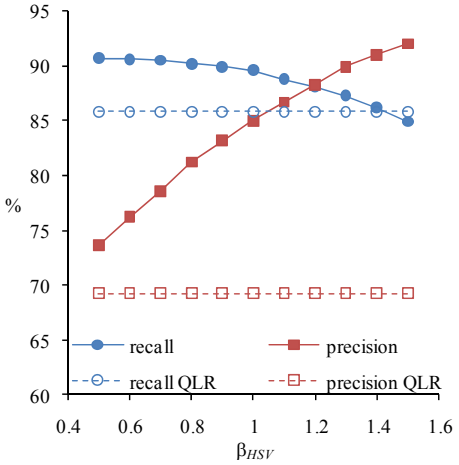

Figure 5. P/R values when varying β_{HSV} (β_{ECR}=1).

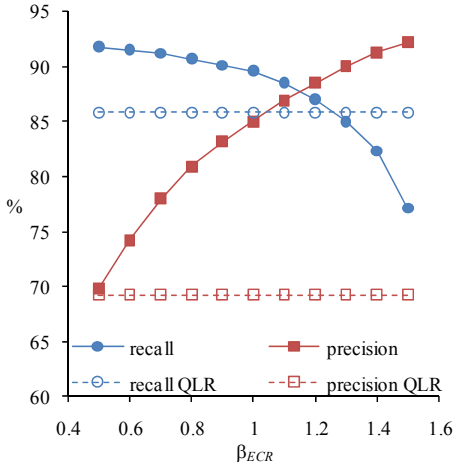

Figure 6. P/R values when varying β_{ECR} (β_{HSV} =1).

4.2 Video Tagging Interface

In order to review tagging results and allow control over proposed tags, we developed a prototype interface for video/shot tagging (Figure 7). In the upper section of the interface, the user can open a video, extract and review the list of shot boundaries computed by the system. In the middle part, the user can play single shots and execute the tagging process, possibly specifying a dimension for the semantic concepts (if tags are specified in different semantic dimensions, such as subject, location, device, etc.). The lower section allows the browsing of the tag taxonomies and shows the results of shot tagging, that can be reviewed/modified by the user. Once all shots have been tagged, SHIATSU derives the video tags, saves all tags into the database and shows the result in the "Recommended Tags" section of the interface.

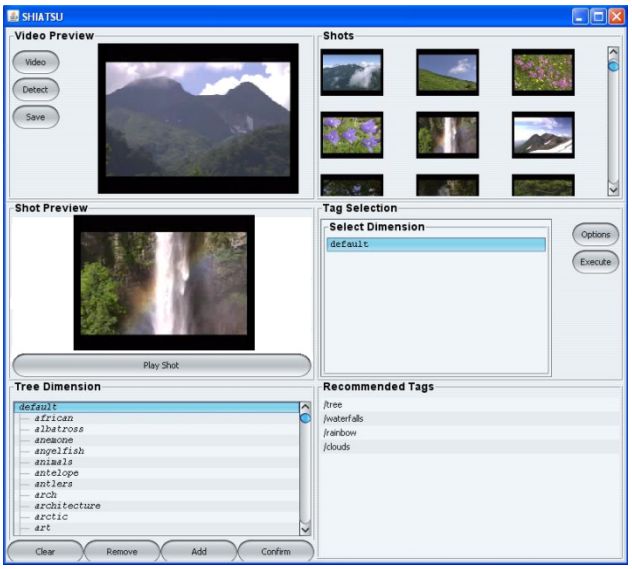

Figure 7. Video tagging interface of SHIATSU.

4.3 Video Tagging Performance

We performed a series of tagging experiments on the 43 videos of the *Mammie* Item Segmentation dataset. Unfortunately, the provided data does not include tagged videos, thus it is quite difficult to assess the quality of tags proposed by the SHIATSU tagging module. In order to provide at least some result on the tagging accuracy, we decided to exploit the Corel image dataset [26] that includes 5000 tagged images (with 371 different semantic labels) and to submit proposed labels to the judgment of real users to assess their relevance to each tagged shot/video. The bottom line of this approach is that the Corel dataset includes general-purpose tags that are not always relevant to the content of videos; for example, tags in the Corel dataset include concepts like "bear", "forest", "sea", while the several videos are focused on people, cities and so on. In this light, the precision of 27% obtained by SHIATSU should not be considered so negative as the mere value might suggest. Indeed, for those few shots/videos whose content is somewhat related to the Corel image tags, the results are quite satisfactory: Figure 8 shows one of such examples (video JSB20040304). We see that the tagging of key frames is quite accurate (since we rely on an image tagging technique which has been previously demonstrated as precise [19]); the same can be said for the propagation of tags to shots and videos. In the example of Figure 8, the overall precision for the video is 70% (correct tags are shown in bold). Again, we expect such precision to be even higher when a knowledge base more related to the visual content of video is used.

5. CONCLUSIONS

In this paper we have presented SHIATSU, a system for the (semi-)automatic tagging of videos which is based on segmentation of cuts and analysis of visual content of the video. Our experiments demonstrate the validity of our approach to the problem, particularly in the light of the good trade-off between efficiency and accuracy of proposed tags.

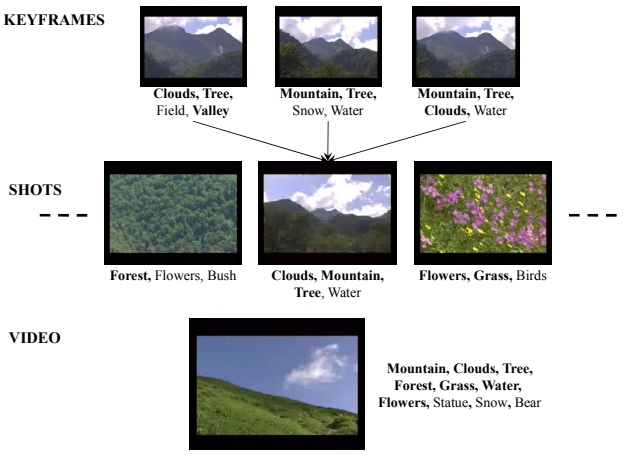

Figure 8. Visual example of shot/video tagging.

Since SHIATSU does not rely on context data but just on video content, it can be effective under many circumstances, especially when textual information about video data is lacking. On the other hand, since only the visual content of videos is considered, SHIATSU can be accompanied to other tagging/classification tools that take into account other aspects, like audio and/or speech. Classification of videos by semantic concepts eases the task of building an intuitive browsing interface, and SHIATSU also offers two different granularity levels of search, since the user can navigate through both videos or shot sequences using their semantic tags.

We are currently working to improve our shot detection algorithm and we want to test it on TRECVID [3] benchmark: although performance on the *Mammie* Item Segmentation task are acceptable, we believe that recognizing the shot cut type (i.e., discriminating between hard cuts and various types of gradual cuts) could improve performances for gradual cut detection. As to the shot tagging module, its accuracy highly depends on the quality of the knowledge base, so we are working on the generation of a suitable dataset of tagged images, in order to provide a quality assessment of the precision of tags proposed by SHIATSU. Moreover, further improvements are needed in order to transform the shot tagging from a semi-automatic to a fully-automatic process. Finally, we are also planning to develop a browsing interface to fully exploit features of SHIATSU, creating a complete video processing and browsing platform.

6. REFERENCES

[1] YouTube, http://www.youtube.com

[2] Google videos, http://video.google.com

[3] TRECVID video retrieval evaluation, http://trecvid.nist.gov

[4] Geetha, P. and Narayanan, V. A survey of content-based video retrieval. *Journal of Computer Science 4 (6)*, pp. 474-486, 2008.

[5] Yuan, J., Wang, H., Xiao, L., Zheng, W., Li, J., and Zhang, B. A formal study of shot boundary detection. *IEEE Trans. on Circuits and Systems for Video Technology*, 17 (2), pp. 168-186, Feb. 2007.

[6] Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., and Jain, R. Content-based image retrieval at the end of the early years. *IEEE TPAMI*, 22 (12), pp. 1349-1380, Dec. 2000.

[7] Datta, R., Joshi, D., Li, J. and Wang, J.Z. Image retrieval: ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40 (2), article 5, Apr. 2008.

[8] Jacobs, A., Miene, A., Ioannidis, G.T., and Herzog, O. Automatic shot boundary detection combining color, edge, and motion features of adjacent frames. In *TRECVID 2004*, pp. 197–206, 2004.

[9] Liu, Z., Zavesky, E., Gibbon, D., Shahraray, B. and Haffner, P. AT&T research at TRECVID 2007. AT&T Labs – Research, Middletown, NJ, USA, 2007.

[10] Qu, Z., Liu, Y., Ren, L., Chen, Y. and Zheng, R. A method of shot detection based on color and edge features. School of Information Science and Engineering, Lanzhou University, China, 2009.

[11] Chasanis, V., Likas, A., and Galatsanos, N. Simultaneous detection of abrupt cuts and dissolves in videos using Support Vector Machines. *Pattern Recognition Letters*, 30 (1), pp. 55-65, Jan. 2009.

[12] Zhao H., Hu B., Zheng M., and Li X., Shot boundary detection based on mutual information and Canny edge detector. *Journal of Communication and Computer*, 6 (10), pp. 17-22, Oct. 2009.

[13] Su, C.-W., Liao, H.-Y. M., Tyan, H.-R., Lin, C.-W., Chen, D.-Y. and Fan, K.-C. Motion flow-based video retrieval. *IEEE Trans. on Multimedia*, 9 (6), pp. 1193-1201, Oct., 2007.

[14] Liao, J. and Zhang, B. A robust clustering algorithm for video shots using Haar wavelet transformation. College of Information Science, Northeastern University, China, 2008.

[15] Barbu, T. Novel automatic video cut detection technique using Gabor filtering. *Computers and Electrical Engineering*, 35 (5), pp. 712-721, Sept. 2009.

[16] Barbu, T. An automatic unsupervised pattern recognition approach. *Proc Romanian Acad Ser A.*, 7(1), pp. 73-78, 2006.

[17] Hauptmann, A.G., Yan, R., Lin, W., Christel, M.G., and Wactlar, H. Can high-level concepts fill the semantic gap in video retrieval? A case study with broadcast news. *IEEE Trans. on Multimedia*, 9 (5), pp. 958-966, Aug. 2007.

[18] Hauptmann, A.G., Christel, M.G., and Yan, R. Video retrieval based on semantic concepts. *Proc. of IEEE*, 96 (4), pp. 602-622, Apr. 2008.

[19] Bartolini, I. and Ciaccia, P. Imagination: Exploiting link analysis for accurate image annotation. *AMR 2007*, Paris, France, Jul. 2007.

[20] Lew, M.S., Sebe, N., Djeraba, D., and Rain, J. Content-based multimedia information retrieval: State of the art and challenges. *ACM Trans. on Multimedia Computing, Communications and Applications*, 2(1), pp. 1–19, Feb. 2006.

[21] Li, J., and .Wang, J.Z. Real-time computerized annotation of pictures. *ACM Multimedia 2006*, Santa Barbara, CA, Oct. 2006.

[22] Rasiwasia, N., Moreno, P.J., and Vasconcelos, N. Bridging the gap: query by semantic example. *IEEE Trans. on Multimedia*, 9 (5), pp. 923-938, Aug. 2007.

[23] Ciaccia, P., Patella, M., and Zezula, P. M-tree: an efficient access method for similarity search in metric spaces. *VLDB '97*, Athens, Greece, Sept. 1997.

[24] Kasturi, R., Strayer, S.H., Gargi, U., and Antani, S. An evaluation of color histogram based methods in video indexing. *1st Int'l Workshop on Image Database and Multi-Media Search*, Amsterdam, The Netherlands, Aug. 1996.

[25] Canny, J. A computational approach to edge detection. *IEEE TPAMI*, 8(6), pp. 679–698, Nov. 1986.

[26] Duygulu, P., Barnard, K., de Freitas, J.F.G., and Forsyth, D.A. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. *ECCV 2002*, Copenhagen, Denmark. pp. 97-112, May 2002.