Increasing the Accuracy of Video Abstraction from Multiple Sources in the Internet of Things

KLIMIS NTALIANIS[1] and NIKOS MASTORAKIS[2]

[1]Department of Marketing
Athens University of Applied Sciences (TEI of Athens)
Agiou Spyridonos, Egaleo, Athens
GREECE
kntal@teiath.gr

[2]Industrial Engineering Department
Technical University of Sofia,
Sofia,
BULGARIA
mastor@tu-sofia.bg

Abstract: - *I*n this paper a multiple sources video abstraction scheme is proposed for summarizing video content in the Internet of Things IoT). The proposed scheme borrows concepts from the IoT in case of events' recording from different cameras around a specific geo-location. In particular the proposed abstraction method assumes that frame trajectories are created and an interpolation scheme is designed to optimally approximate these trajectories. The selected points on every trajectory are extracted as key-frames. Key-frames from all different sources are put in chronological order to produce the video abstract. In case a user demands more accuracy, he/she can increase it by refining the number of key-frames. In this case an optimal accuracy increasing approach is proposed and the video abstract is tuned according to the needs of the user. Experimental results on real world videos exhibit the promising performance of the proposed scheme.

Key-Words: - Video Abstracts, Internet of Things, Interpolation, Accuracy Increase

1 Introduction

Generally speaking, the IoT refers to the networked interconnection of everyday objects, which are often equipped with ubiquitous intelligence. The IoT will increase the ubiquity of the Internet by integrating every object for interaction via embedded systems. By this way a highly distributed network of devices will be developed which will communicate with human beings as well as other devices. Analysts predict that the IoT will comprise up to 26 billion interconnected devices by 2020, a 30-fold increase from 2009 [1].

On the other hand, most of today's Internet users have at least two personal devices that they use to create, share, and consume multimedia content anywhere and anytime. Both the amount of content and the amount of time people spend viewing it have increased significantly in recent years. It is estimated that by 2019 it would take a viewer more than 5 million years to view all the Internet videos that will be generated each month! According to the June 2016 Sandvine Global Internet Phenomena Report, "streaming audio and video now accounts for 71 percent of evening traffic in North American fixed access networks. Sandvine expects this figure will reach 80 percent by 2020 [2]. In the coming years and especially in the era of the IoT, the majority of multimedia Internet traffic will be transmitted wirelessly. Quality issues will become increasingly important as user expectations rise. To address these and other demands intelligent algorithms should be proposed to support the analysis, storage, transmission, visualization and consumption of multimedia.

This paper focuses on video abstraction in the IoT. In particular it is anticipated that in the IoT multiple video cameras will be placed in neighboring geo-locations and record events. The recordings from these multiple sources need to be properly analyzed and intelligently orchestrated to provide a very high quality of service. More specifically, our work focuses on the creation of video abstracts from multiple recordings of an event from adjacent geo-locations. The recordings may cover different angles, positions, heights etc. and in order to provide a video abstract of the total content, key-frames are extracted from each recording. Towards this direction, for each frame object-based and global features are extracted to form a content description feature vector. Then feature vectors are plotted to provide frame trajectories. In order to extract key-frames from each trajectory, an interpolation scheme is proposed which is based on error minimization over the approximated trajectory. A genetic algorithm is also incorporated to provide optimal selection of trajectory points. Additionally an accuracy increasing method is also proposed in case a user needs more information from the summarized content. Experimental results on real world multiple sources recordings illustrate the promising performance of the proposed scheme.

The rest of this paper is organized as follows: in Section 2 state-of-art work is presented. Section 3 focuses on video content representation. In Section 4 the proposed multiple sources video abstraction scheme is described as well as the method for increasing the summary's resolution. Section 5 provides the experimental results while Section 6 concludes this paper.

2 Previous Works

In [3] a multitask feature selection is introduced to discover the semantically important features. Then, the key frames are selected based on their contributions to reconstruct the video semantics. Thereafter, a probabilistic model is proposed to dynamically fit the key frames into an aesthetically pleasing video summary, wherein its constituent frames are adaptively destabilized.

In [4] a video is summarized by finding shots that co-occur most frequently. The main technical challenge is dealing with the sparsity of co-occurring patterns, and for this reason a Maximal Biclique Finding algorithm is developed to find sparsely co-occurring patterns, discarding less co-occurring patterns even if they are dominant in one video.

Much work has been proposed to measure the shot importance through supervised learning. Egocentric videos can be summarized by learning important faces, hands, and objects [5], or learning the overall energy of storiness, importance, and diversity of selected video shots [6]. To predict per-frame interestingness, low-level, high-level, and spatial-temporal features were combined to train a linear regression model [7]. Similarly, shot importance was measured with a pre-trained topic-specific binary SVM classifier [8] or a SVM ranker [9]. Furthermore, with a small number of labels, a hierarchical model was learned to generate a video summary that contains objects of interests [10]. In [11] multi-view summarization is applied to wireless video sensors so that redundant content is reduced. There are also some other recent works and surveys such as [12] – [14].

Even though interesting, the aforementioned schemes do not focus on the IoT. Furthermore no resolution increasing algorithms are considered.

3 Vector-Based Video Content Representation

One of the most challenging areas in video analysis is video representation. Or in other words, how to describe video content in such a form that represents the real video and is processable by a computer. The better the representation, the better the performance of a video abstraction scheme. In the following the features used for representing videos is described.

In particular two different types of descriptors have been adopted in our case for video content representation. The first type refers to global frame properties (global-based descriptors), while the second to object characteristics (object-based descriptors).

Global-based descriptors include the color, texture and motion histograms of a video frame. The histograms are estimated directly over the MPEG compressed stream as in [15] to reduce the cost and storage requirements.

Let us denote as $\mathbf{h}_g^c$, $\mathbf{h}_g^t$ and $\mathbf{h}_g^m$ the vectors which map the histogram bins of global color, texture and motion. Then the feature vector $\mathbf{f}_g$ of the global-based descriptors is given by:

$$\mathbf{f}_g = \left[(\mathbf{h}_g^c)^T(\mathbf{h}_g^t)^T(\mathbf{h}_g^m)^T\right] \qquad (1)$$

In order to extract object-based descriptors, initially each frame is split into segments by applying the adapted multiresolution Recursive Shortest Spanning Tree algorithm (aM-RSST) [16]. This algorithm is very efficient and has low computational complexity. For each color segment, the three color components in the RGB space, the segment's location as well as the size of the segment are selected as segment descriptors. Similarly, for each motion segment, the average motion vector and the respective location and area are extracted.

However, since different frames may have different numbers of segments, all color/motion segment descriptors are classified into pre-determined classes forming an object-based color and motion histogram denoted as $\mathbf{h}_o^c$ and $\mathbf{h}_o^m$. As a result, the feature vector for the object-based descriptors $\mathbf{f}_o$ is given by:

$$\mathbf{f}_o = [(\mathbf{h}_o^c)^T (\mathbf{h}_o^m)^T] \quad (2)$$

Thus, by gathering the vectors $\mathbf{f}_g$ and $\mathbf{f}_o$, the full feature vector **f** of a video frame is formed as:

$$\mathbf{f} = \left[\mathbf{f}_g^T \ \mathbf{f}_o^T\right]^T \quad (3)$$

Other types of descriptors could also be taken into consideration and used in the full feature vector, such as semantic objects' characteristics.

4 Video Abstraction from Multiple Sources

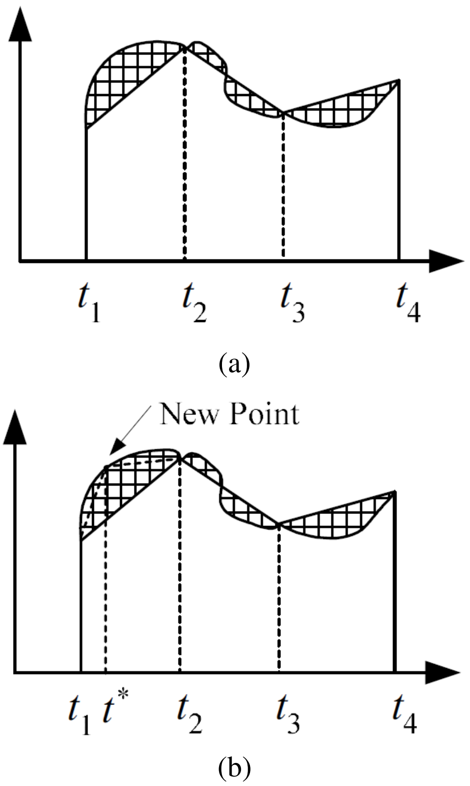
Let us assume that a real world event is covered by several video caption devices from several different geo-locations around the event and under variable conditions (different lighting, motion, angle, zoom, resolution etc.). For simplicity, let us also assume that all these different video sequences are captured as one-shot sequences. Moreover and without loss of generality let us assume that these one-shot sequences are also temporally synchronized. In this paper we focus on providing an overall summary of the event, by abstracting each shot and by temporally combining the different abstracts into a final summary. By this way the summary will provide a multi-view sight of the event under consideration. To give an example, it is like a penalty scene in football, where the director provides the content of all different cameras to the audience so that the audience has a better understanding of the situation.

In our work, the plot of all feature vectors of a shot forms a trajectory, which actually expresses the temporal variation of the content of video frames of this shot. Thus, shot-abstraction can be carried out by selecting key-frames from the shot. Towards this direction key-frames extraction can be considered as a problem of selecting appropriate curve points (time instances), able to represent the corresponding trajectory. The selected curve points should provide sufficient information about the trajectory, so that it can be reproduced using some kind of interpolation.

Let us denote as $g(t)$ the magnitude of a frame's feature vector for frame number t:

$$g(t) = |\mathbf{f}(t)|, \ t \in [T_a T_b] \quad (4)$$

where $\mathbf{f}(t)$ is the same as the vector of Eq.(3) but the time index t has been added to indicate the frame number. T_a and T_b correspond to the first and last frame of the shot respectively.



(a)

(b)

Figure 1: (a) Example of the proposed video abstraction scheme based on interpolation. (b) Graphical representation of the resolution increase.

In the proposed scheme, selection of points is optimally performed using interpolation theory. In particular let us assume that K points are to be selected to approximate the curve $g(t)$. In this case the objective is to estimate the frame numbers $\mathbf{t}= [t_1, t_2, \ldots, t_{K-1}, t_K]^T$, which provide the minimum approximation error of the curve $g(t)$, using an interpolation scheme. A simple example of this concept is depicted in Figure 1(a). In this case, the curve line corresponds to function $g(t)$, while the stepwise lines are used to approximate $g(t)$ at points t_1, t_2, t_3 and t_4. The area between the two curves corresponds to the approximation error and the frame indices are estimated so that this error is minimized. Furthermore and without loss of generality a linear interpolation scheme is adopted. Thus, the obtained approximation error $E(\mathbf{t})$ is given by:

$$E(\mathbf{t}) = \sum_{j=1}^{K-1} \int_{t_i}^{t_{i+1}} |g(t) - h_i(t)| dt \quad (5)$$

where $h_i(t)$ is the line which approximates the curve $g(t)$ at the interval $[t_i \ t_{i+1}]$. In a linear interpolation scheme, $h_i(t)$ is given by

$$h_i(t) = g(t_i) + \frac{g(t_{i+1}) - g(t_i)}{t_{i+1} - t_i}(t - t_i), \forall i = 1, \ldots, K - 1 \qquad (6)$$

In Eq. (5), we have assumed that $t_1 \equiv T_a$ and $t_K \equiv T_b$. This means that the first and last points of the shot are always selected as key-frames. Extension to the case that points t_1 and t_K can take any other value can be also similarly performed.

Thus, the optimal vector of selected points $\mathbf{t} = [t_1, t_2, \ldots, t_{K-1}, t_K]^T$, i.e., the optimal key-frames, are given by:

$$\hat{\mathbf{t}} = \arg\min_{\mathbf{t}} E(\mathbf{t}) = \sum_{j=1}^{K-1} \int_{t_i}^{t_{i+1}} |g(t) - h_i(t)| dt \qquad (7)$$

Since $g(t)$ can be any generic function, the optimal vector $\hat{\mathbf{t}}$ cannot be directly calculated. Since the complexity of an exhaustive search to obtain the minimum value of Eq. (7) may be very large (in case of shots with long duration), a genetic algorithm (GA) approach is adopted [17].

4.1 Optimal Key-Frames Selection by a Genetic Algorithm

In the GA approach, the index vector $\mathbf{t} = [t_1, t_2, \ldots, t_{K-1}, t_K]^T$ is considered as a chromosome, while the elements $t_1, t_2, \ldots, t_{K-1}, t_K$ as the genetic material of the respective chromosome. Initially, a population of m chromosomes is created, say $P(0)$, consisting of m randomly selected index vectors $\mathbf{t}$. That is, $P(0)=\{\mathbf{t}_1(0), \ldots, \mathbf{t}_m(0)\}$, where $\mathbf{t}_i(0)$, $i=1, \ldots, m$ corresponds to the i-th chromosome of the population $P(0)$.

The approximation error $E(\mathbf{t}_i(\mathrm{n}))$ of Eq.(5) is considered as the objective function of the GA for the ith chromosome of the nth population $P(n)$. Based on $E(\mathbf{t}_i(\mathrm{n}))$, appropriate "*parents*" are selected so that a fitter chromosome gives a higher number of offspring and thus has a higher chance of survival in the next generation. In particular, in our case, a probability is assigned to each chromosome:

$$PR_{t_i} = E(\mathbf{t}_i(n)) / \sum_{i=1}^{m} E(\mathbf{t}_i(n)) \qquad (8)$$

Then chromosomes are randomly selected based on their assigned probabilities as candidate parents (*roulette wheel selection* procedure [17]).

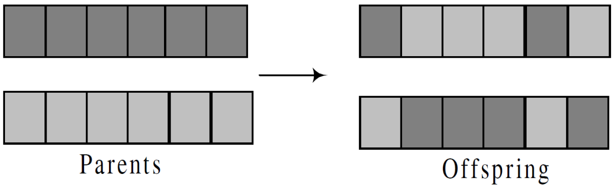


Figure 2: uniform crossover operator mechanism

A set of new chromosomes (offspring) is then produced by mating the genetic material of the parents using a *crossover* operator. In our case, the selected material of two parents is randomly mated resulting in a uniform-crossover operator as in Figure 2.

In the next we apply *mutation* to the newly created chromosomes. This process introduces random gene variations that are useful for restoring lost genetic material, or for producing new material that corresponds to new search areas. In our case, a random mutation scheme is adopted, meaning that each gene undergoes mutation with a probability p_m.

By applying the parent selection, the crossover operator and the mutation mechanisms, the next population $P(n+1)$ is created by inserting the new chromosomes and deleting the older ones. Several GA cycles take place by repeating the above-mentioned procedures until the algorithm converges to the optimal solution.

By the aforementioned procedure, key-frames are detected for each shot. The final summary of the whole event is produced by gathering all key-frames from all shots and putting them in order of occurrence (chronological order of all key-frames).

4.2 Increasing the Abstract's Resolution

Of course the number K of selected key-frames cannot be constant for all shots of the same event (since some videos are captured by still cameras and other videos have a lot of motion, zoom, etc). Furthermore K cannot be constant also for all users, since different users that follow the event may need more information (a larger video abstract). In this case it would be interesting to let different users to ask for more content from some shots and less content from other shots (personalized adaptation of video summary). However this approach also needs users' feedback, which is not currently available in our experiments.

In order to implement an adapted-K policy (where K is the number of key-frames per shot), let us assume that at the beginning an initial estimation of K is provided for each shot, by applying the heuristic but fast technique of [18].

In case K yields an approximation error greater than the required one, additional points (key-frames) should be selected. Then, instead of running from scratch the aforementioned minimization algorithm (something that is very time consuming especially in case of several different users with different tastes), an approximation scheme is proposed, the details of which are provided in the following paragraphs.

Let us suppose, without loss of generality that we would like to increase the extracted points of a specific shot from K_o to K_{o+1}. Based on Eq. (5), the error E can be written as:

$$E = \sum_{i=1}^{K_o - 1} e_i \tag{9}$$

where e_i, i=1, 2, …, K_{o-1} corresponds to the approximation error in the interval [t_i, t_{i+1}]. Then, the new point t^* should be selected from the interval [t_k, t_{k+1}], which provides the maximum approximation error, i.e.,

$$[t_k, t_{k+1}] : k = \arg\max_i e_i \tag{10}$$

The error e_k of [t_k, t_{k+1}] can be split as:

$$e_k = e_k^- + e_k^+ \tag{11}$$

where

$$e_k^- = \int_{t_k}^{t^*} \left| g(t) - h_k^-(t) \right| dt \tag{12a}$$

$$e_k^+ = \int_{t^*}^{t_{k+1}} \left| g(t) - h_k^+(t) \right| dt \tag{12b}$$

The $h_k^-(t)$ is the linear approximation of $g(t)$ in the interval [t_k t^*], while $h_k^+(t)$ the linear approximation in the interval [t^* t_k]. Then the optimal new point t^* can be obtained by the following minimization criterion:

$$t^* = \arg\min(e_k^- + e_k^+) \tag{13}$$

Minimization of Eq. (13) is performed similarly to Eq. (7) using the GA approach. However, the computational time in this case is much smaller than applying the algorithm from scratch, since only one new point is calculated. A graphical representation of this concept is provided in Figure 1(b), where the new point t^* is selected between points t_1 and t_2.

Similarly to the abovementioned technique, we can decrease the resolution of the abstract (number K of key-frames for one or more shots), by removing the key-frame which provides the minimum increase to the interpolation error.

5 Experimental Results

In this section, we evaluate the performance of the proposed multiple-sources video abstraction scheme. Since currently there are not any IoT standard video datasets for the described or other similar scenarios, the proposed multiple-sources video abstraction scheme has been evaluated on several synchronized video sequences available on Youtube. In particular 80 multiple-sources video shots have been gathered of a total duration of 197 minutes and 32 seconds (an average of 148.15 seconds). The longest shot lasted 6 minutes and 11 seconds, while the shortest 27 seconds. These shots contain complicated content, with zooming, panning, complex camera effects, motion etc.

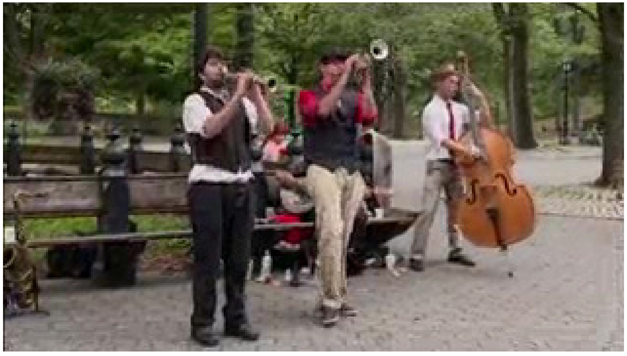


Figure 3: One randomly selected frame from camera #1 of Vid#27 – Overall view.

Due to space limitations, results on a 50 seconds video (Vid#27) consisting of three sources (sequence #1, sequence #2 and sequence #3) are presented in the following Figures. Diagrams are estimated for another much larger video (Vid#63, also with three sources and duration of 4 minutes and 47 seconds). Vid#27 is captured by three video cameras which are located at three different locations around the event. The event they are covering is an outdoor performance (park area) of four musicians. Figure 3 illustrates one randomly selected frame of camera #1 (sequence #1) which provides an overall view of the event.

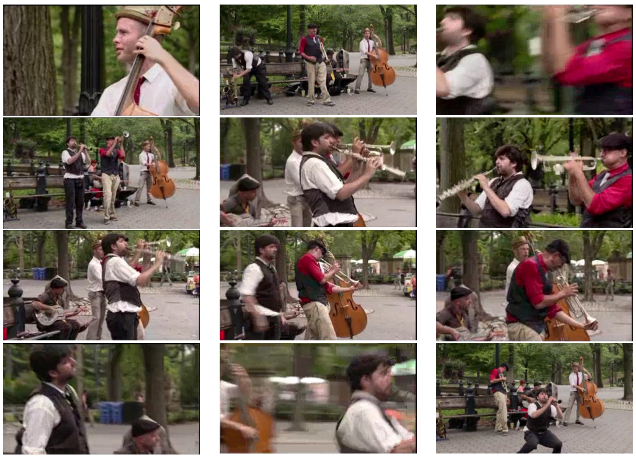
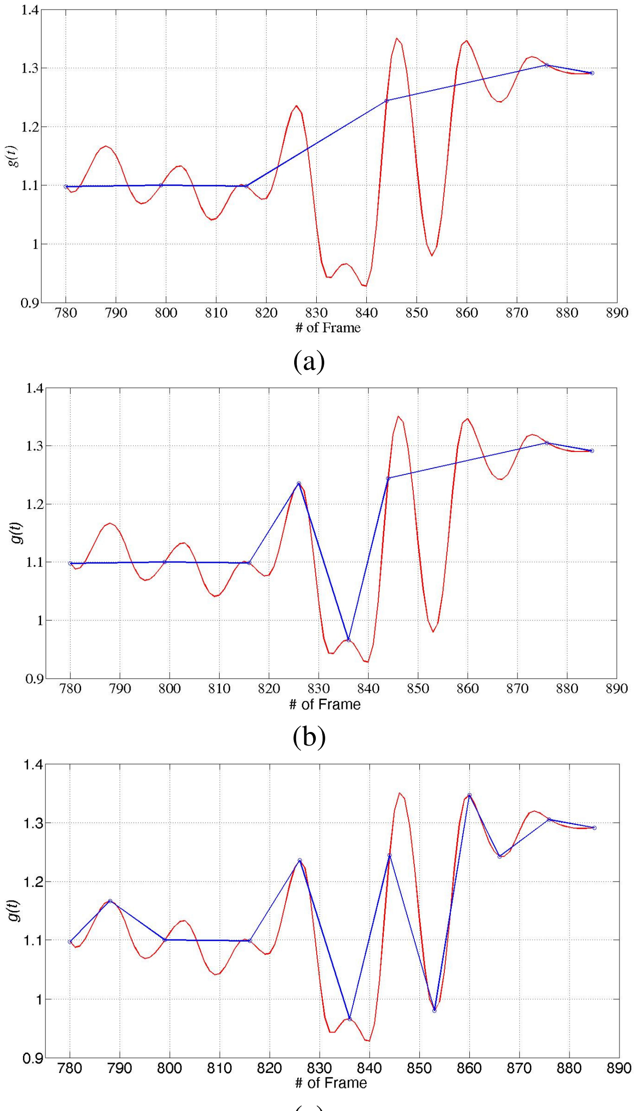


Figure 4: The video abstract consisting of 12 extracted key-frames in total (E<15%), from the 3 different sequences. Key-frames are put in chronological order.

The selected key-frames extracted from the shot for an approximation error E<15% (for all three sequences) are shown in Figure 4. Three key-frames are extracted from Sequence #1 and Sequence #2, while Sequence #3 provides six key-frames. Here it should be mentioned that key-frames from all three cameras are put in order of occurrence (chronological order), so that the final abstract is created. As it can be observed, the final abstract describes with high efficiency the event's visual content, since several different views and time instances are considered.



(a)

(b)

(c)

Figure 5: The feature vector $g(t)$ of a part of a shot along with the respective optimal approximation versus the number of key-frames. (a) K=6 key-frames. (b) K=8 key-frames. (c) K=12 key-frames.

The magnitude of the frame feature vectors for a part of Sequence #2 of Vid#63 is shown in Figure 5. In particular the period from frame 780 to frame 885 is presented and three different cases are visualized for K=6, 8, and 12 respectively. Thus initially sequence#2 provides six key-frames and the accuracy of abstraction increases in the following two rounds where eight and twelve key-frames are extracted respectively. The selected points (key-frames) are shown as circles in this figure. The respective approximation error E obtained is 0.329 in case of Figure 5(a), 0.257 in 5(b) and 0.116 in 5(c). As observed, the approximation accuracy increases with the number of key-frames. Here it should be mentioned that the quality of the summary does not linearly increase with the number of key-frames. Furthermore many users are interested in seeing key-frames in specific time intervals of the summary (interested in zooming in, in specific themes of the event, e.g. a goal in a football match). Thus an adaptive interpolation scheme could better meet these needs.

Table I: Storage / Transmission reduction and computational cost requirements for five video sequences.

# of Video	Number of Sources	Total Number of Frames	Total Number of Key-Frames	Storage / Transmission Reduction (%)	Average Comp. Cost per frame (sec)
1	2	8,450	17	99.799	0.18
7	3	23,325	34	99.854	0.21
22	3	15,675	28	99.821	0.16
34	4	24,100	46	99.809	0.23
76	2	5,700	15	99.737	0.17

Finally the numbers of key-frames extracted for five randomly selected multiple-source video sequences and for a fixed E=0.22, are presented in Table I, along with storage reduction, which lies in the interval [99.737% 99.854%]. In the same table the computational cost is also provided. As it can be seen, in all cases, the required time is relatively low (about 0.19 seconds per frame on average or 4.75 processing seconds per video second). However, currently the algorithm cannot be incorporated in real time applications, but better meets the needs of offline post-processing cases. Furthermore it should be mentioned that the average time per frame depends on the complexity of the content of each video. Finally it should also be stressed that in the experiments of Table I, each source initially provided six key-frames and the rest were estimated by the proposed accuracy increasing algorithm.

6 Conclusion

In this paper a multiple sources video abstraction scheme has been proposed, focusing on the Internet of Things. The proposed scheme assumes several different cameras recording the same event. In order to summarize the whole content an interpolation-based scheme has been proposed. A resolution increase algorithm has also been introduced. Experimental results on real life sequences seem promising.

In the future different questions should be answered. What about real time summarization in the IoT ? How complexity issues are going to be solved in case of millions of recordings ? Which of the views is the most important ? These and other issues should be further examined in future works.

References:

[1] www.gartner.com/newsroom/id/2636073, accessed: 25th May 2017.

[2] https://www.computer.org/web/computingnow/archive/future-of-multimedia-on-the-internet-november-2016-introduction, accessed: 25th May 2017.

[3] L. Zhang, Y. Xia, K. Mao, H. Ma, Z. Shan, “An Effective Video Summarization Framework Toward Handheld Devices,” *IEEE Transactions on Industrial Electronics*, Vol. 62, No. 2, Feb. 2015.

[4] W.-S. Chu, Y. Song, A. Jaimes, “Video Co-summarization: Video Summarization by Visual Co-occurrence,” *CVPR 2015.*

[5] Y. J. Lee, J. Ghosh, and K. Grauman, “Discovering important people and objects for egocentric video summarization,” In *CVPR, 2012.*

[6] Z. Lu and K. Grauman, “Story-driven summarization for egocentric video,” In *CVPR, 2013.*

[7] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool, “Creating summaries from user videos,” In *ECCV, 2014.*

[8] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid, “Category-specific video summarization,” In *ECCV, 2014.*

[9] M. Sun, A. Farhadi, and S. Seitz, “Ranking domain-specific highlights by analyzing edited videos,” In *ECCV, 2014.*

[10] D. Liu, G. Hua, and T. Chen, “A hierarchical visual model for video object summarization,” TPAMI, Vol. 32, No. 12, p.p. 2178–2190, 2010.

[11] S.-H. Ou, C.-H. Lee, V. S. Somayazulu, Y.-K. Chen, S.-Y. Chien, “On-Line Multi-View Video Summarization for Wireless Video Sensor Network,” *IEEE Journal of Selected Topics in Signal Processing*, Vol. 9, No. 1, Feb. 2015.

[12] S. Sheinfeld, Y. Gingold, A. Shamir, “Video Summarization using Causality Graphs,” Proceedings of Workshop on *Human Computation for Image and Video Analysis*, HCOMP 2016.

[13] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, “Video summarization with long short-term memory,” arXiv preprint arXiv:1605.08110, 2016.

[14] A. G. del Molino, C. Tan, J.-H. Lim, A.-H. Tan, “Summarization of Egocentric Videos: A Comprehensive Survey,” *IEEE Transactions on Human-Machine Systems*, Vol. 47, No. 1, Feb. 2017.

[15] Y. Avrithis, N. Doulamis, A. Doulamis, and S. Kollias, “Optimization Methods for Key Frames and Scenes Extraction,” *Journal of Computer Vision and Image. Under.*, Vol. 75, pp. 3-24, July/August 1999.

[16] K. S. Ntalianis, A. D. Doulamis, N. D. Doulamis, and S. D. Kollias, “Multiresolution GVF field: A Fast Implementation Towards VOP Segmentation,” in Proceedings of the *IEEE International Conference on Multimedia and Exposition* (ICME’01), Tokyo, Japan, August 2001.

[17] E. Goldberg, *Genetic Algorithm in Search, Optimization and Machine Learning*, Addison Wesley, 1989.

[18] A. Doulamis, N. Doulamis, and S. Kollias, "Non-Sequential Video Content Representation Using Temporal Variation Of Feature Vectors," *IEEE Trans. on Consumer Electr.*, Vol. 46, No. 3, pp. 758-768, 2000.