Communication-Efficient Multi-view Keyframe Extraction in Distributed Video Sensors

Shun-Hsing Ou [1], Yu-Chen Lu [2], Jui-Pin Wang [2], Shao-Yi Chien [1], Shou-De Lin [2], Mi-Yen Yeh [3],
Chia-Han Lee [3], Phillip B. Gibbons [4], V. Srinivasa Somayazulu [4], Yen-Kuang Chen [4]

[1] *Graduate Institute of Electronics Engineering and Department of Electrical Engineering, National Taiwan University*
[2] *Department of Computer Science and Information Engineering, National Taiwan University*
[3] *Academia Sinica, Taiwan*
[4] *Intel Cooperation, USA*

***Abstract*—Video sensors are widely used in many applications such as security monitoring and home care. However, the growth of the number of sensors makes it impractical to stream all videos back to a central server for further processing, due to communication bandwidth and server storage constraints. Multi-view video summarization allows us to discard redundant data in the video streams taken by a group of sensors. All prior multi-view summarization methods, however, process video data in an off-line and centralized manner, which means that all videos are still required to be streamed back to the server before conducting the summarization. This paper proposes an *on-line, distributed* multi-view summarization system, which integrates the ideas of Maximal Marginal Relevance (MMR) and MS-Wave, a bandwidth-efficient distributed algorithm for finding k-nearest-neighbors and k-farthest-neighbors. Empirical studies show that our proposed system can discard redundant videos and keep important keyframes as effectively as centralized approaches, while transmitting only 1/6 to 1/3 as much data.**

I. Introduction

With the rapid development of communication and video sensing technologies, video sensors have been widely used in many applications, such as security monitoring, home care, and large-scale environment analysis. Nevertheless, as the number of sensors grows, it is becoming impractical to stream all videos back to the central server for computation due to the limited communication bandwidth and server storage. Some approaches store the video data in local storage aside the sensor. However, they usually just keep the most recent data, which may lead to severe information loss.

To enable large-scale video indexing, techniques of automatic video summarization, [1], [2], [3], [4], [5], are proposed to generate short representations of original videos. Recently, multi-view video summarization, [6], [7], [8], [9], which further focuses on reducing redundancy across multiple cameras with overlapped field-of-view, has attracted more and more attention. Such techniques can thus be applied to a group of video sensors to discard the redundant video data. However, most prior multi-view summarization methods process videos in a centralized and off-line manner, which means all videos have to be streamed back to the central server. In such systems, the bandwidth and the server storage requirement is still large, making it impractical when the number of video sensors grows larger.

In this paper, we propose an on-line and distributed multi-view video summarization system. Our system is based on the previous multi-video summarization work that applied Video Maximal Marginal Relevance (Video-MMR) [8]. The original version of Video-MMR is centralized and off-line, which requires features of all frames to be streamed back to the server. Searching for a better solution, we found MsWave [10], which is originally a communication efficient method to search k nearest and farthest neighbors (kNNs and kFNs) of a set of reference patterns in a distributed environment, suitable for our purpose. We modified Video-MMR to fit in the framework of MsWave to enable on-line and distributed summarization.

As shown in our experiments, the performance of the proposed system is comparable with other centralized or off-line summarization approaches, while large communication overhead can be reduced.

II. Related Work

Generally, video summarization can be divided into two categories: keyframe selection and video skimming. The keyframe selection algorithms [11], [3], [4], [7], [8], [5] select the best representative frames of the original video while the video skimming algorithms [6], [4], [5] generate a short highlight of the original video. Interested readers may find more detailed reviews in [1], [2], [11].

In this paper, we focus on the problem of multi-view video summarization, where redundancy across several cameras are further reduced. The representations of multi-view video summarization can also be divided into keyframes and video skimming. Fu et al. [6] proposed a graph-based multi-view video skimming algorithms. All input videos are divided into small shots first, and a graph is built and cut into small groups. The skimming results are generated by the representative shot of each group. Leo and Manjunath [9] proposed a multi-view summarization algorithm by learning an occurrence model from the correspondence between different areas of

978-1-4799-6139-9/14/$31.00 ©2014 IEEE.

different views. Summarization is generated by choosing those segments that can best reconstruct the original videos.

In contrast to the above two reports that based on video skimming, Li et al. [7] focused on the multi-view keyframe selection problem. By clustering all interesting frames, the keyframes are determined by choosing the representative frames of each cluster. Last but not least, Li and Merialdo [8] formulated the multi-view keyframe selection problem as a text summarization problem and utilized the idea of Maximal Marginal Relevance (MMR) [12].

Although the aforementioned works successfully generate summarization from multiple videos, they can only process the videos in a centralized manner. In this paper, we focus on the multi-view keyframe selection problem and propose an algorithm that can generate the summarization in a distributed manner by exchanging only small data between multiple video sensors and the server. Significant storage and bandwidth can then be saved with the proposed system.

III. Proposed System

Our multi-view keyframe selection algorithm is inspired by the work of Li and Merialdo [8], where the multi-view keyframe selection problem is formulated the same as the text summarization problem and is solved using the idea of Maximal Marginal Relevance (MMR) [12] in a centralized and off-line manner. We propose to on-line generate the keyframes distributedly by exchanging small data between multiple sensors and the server. The MMR problem is solved with the help of MsWave [10] to generate the keyframes.

A. Problem Formulation

1) Video Maximal Marginal Relevance: In [8], the multi-video summarization problem is solved by iteratively selecting keyframes in a centralized manner. In the kth iteration, the Video Marginal Relevance (Video-MR) of all frames are computed as

$$\text{Video-MR}(f_i) = \lambda \text{Sim}_1(f_i, V \setminus S_k) - (1-\lambda) \max_{g \in S_k} \text{Sim}_2(f_i, g), \quad (1)$$

where f_i is the ith frame, V is the set of all video frames, S_k is the set of selected keyframes before the kth iteration, and $\setminus$ is the set minus operation. Sim_1 is the average similarity measure between a frame to a set of frames, Sim_2 is the similarity measure between two frames, and λ is a parameter that controls the weighting between the two terms.

The first term of Video-MR in (1), $\text{Sim}_1(f_i, V \setminus S_k)$, measures the similarity of f_i with all the unselected frames, $V \setminus S_k$. This measures the ability of frame f_i to represent the remaining frames. A frame more similar to all the remaining frames is a better keyframe candidate. The second term of Video-MR in (1), $\max_{g \in S_k} \text{Sim}_2(f_i, g)$, measures the similarity between frame f_i and the most similar one in the already selected keyframes. High similarity means the frame f_i is redundant and should not be selected. Video-MR (1) combines the two terms into a score for each frame, frames with higher Video-MR are better choices for keyframes.

In the kth iteration, the frame with the highest Video-MR, i.e., Video Maximal Marginal Relevance (Video-MMR), is selected into the summary,

$$S_{k+1} = S_k \cup \arg \max_{f_i \in V \setminus S_k} (\text{Video-MR}(f_i)). \quad (2)$$

The algorithm ends when the number of keyframes meets the predefined number.

2) Feature Extraction: To reduce the computation and communication when computing Video-MR, we extract a representative feature for each frame, and perform the computation in the feature space. Any kind of features that can represent a frame can be applied. In our experiment, we simply use a 256-bin color histogram in HSV color space as our features.

For the sequences captured by fixed cameras, such as videos taken by surveillance systems, we further apply background subtraction [13], and compute the histogram only in the foreground region, since only foreground regions are the informative part of such videos. If the number of the foreground pixels is smaller than a given threshold, the frame is skipped directly. Eq. (1) is then modified using the distance between features instead of frame similarity, as shown below:

$$\text{Video-MR}(f_i) = \\ -\lambda \text{Diff}_1(f_i, V \setminus S_k) + (1-\lambda) \min_{g \in S_k} \text{Diff}_2(f_i, g), \quad (3)$$

where the difference function is defined using L2 distance between the frame features, i.e., $\text{Diff}_1(f_i, V) = \frac{1}{\text{Size}(V)} \sum_{f_j \in V} ||f_i - f_j||_2$, $\text{Diff}_2(f_i, g) = ||f_i - g||_2$.

3) On-line and Distributed Keyframe Selection: The iterations of Video-MMR in Eq. (2) are performed off-line and centralized. To modify the process into an on-line system, we perform keyframe selection iteration for every fixed time period T. In the iteration at time $t + T$, the set of frames captured in the time period from t to $t+T$, which is denoted as V_t, is used instead of $V \setminus S_k$ to avoid buffering all frames. The first term of Video-MR in Eq. (3) is modified as $\text{Diff}_1(f_i, V_t)$. All the frames with Video-MR larger than a given threshold $D_{threshold}$ in each iteration are chosen as the keyframes instead of the maximal one, i.e.,

$$S_{t+T} = S_t \cup \{\forall f_i \mid \text{Video-MR}(f_i) > D_{threshold}\}, \quad (4)$$

where S_t is the set of already selected keyframes at time t.

To select keyframes from multiple sensors, it is required to preform (4) distributedly from all sensors. Assuming there are M cameras, V_t can be expressed as $V_t = V_{t,1} \cup V_{t,2} \cup ... \cup V_{t,M}$, where $V_{t,m}$ is the set of frames belonging to the mth camera taken from time t to $t + T$. To reduce the communication burden, we compute the first term of Video-MR locally. For a frame $f_{i,m}$ from the mth camera, we compute Video-MR as

$$\text{Video-MR}(f_{i,m}) = \\ -\lambda \text{Diff}_1(f_{i,m}, V_{t,m}) + (1-\lambda) \min_{g \in S_t} \text{Diff}_2(f_{i,m}, g), \quad (5)$$

where the first term can be computed locally in each sensor for each iteration.

To compute the second term of (5), we store all features of already selected keyframes, S_t, in the server, and perform nearest neighbor search between the server to each sensor.

This can be done naively by streaming all features from the sensors to server. However, such approach introduces large communication overhead. As a result, the second term is computed with the help of MsWave, where much smaller data compared to the features are sent. After the computation of Eq. (5), the features of the selected keyframes are streamed to the server for next iteration.

B. MsWave

In this section, we introduce how to leverage the MsWave [10] approach to efficiently calculate the second term in the Eq. (5) in each iteration. MsWave is a general communication-efficient framework to identify kNN instances given multiple time-series-based reference patterns in a distributed environment. By leveraging the similarity ranges provided by MsWave and regarding each frame feature as a time series, we can find efficiently all frames with Video-MR larger than a given threshold $D_{threshold}$ in each iteration.

At the iteration for S_{t+T}, given the features of S_t in the server and a frame f taken in the period from t to $t+T$ in the mth camera, we want to calculate $\min_{g \in S_t} \text{Diff}_2(f,g) \forall f \in V_m$. By the Haar Wavelet decomposition [14], each feature is decomposed into an error tree of multiple resolutions. Then, instead of simultaneously distributing all the relevant coefficients of S_t to the mth camera, the server only sends the coefficients one level at a time, starting from the top (the coarsest) level. By these meta-data, the mth camera calculates the following lower bounds and upper bounds in each level $\ell, \forall f \in V_m, \forall g \in S_t$:

$$\begin{aligned} &LB(f,g) \leq Diff_2^2(f,g) \leq UB(f,g), \\ &LB(f,g) = accDiff^{\ell}(f,g), \\ &UB(f,g) = \\ &\quad accDiff^{\ell}(f,g) + \sum_{l=1}^{\ell-1} \sum_{p} ([n_{(l,p)}^{(f)}]^2 + [n_{(l,p)}^{(g)}]^2) \times 2^l \\ &\quad + 2 \times \sqrt{\sum_{l=1}^{\ell-1} \sum_{p} [n_{(l,p)}^{(f)} \times 2^l]^2 \times \sum_{l=1}^{\ell-1} \sum_{p} [n_{(l,p)}^{(g)}]^2}, \end{aligned} \tag{6}$$

where $Diff^l(f,g) = 2^l \times \sum_p [n_{(l,p)}^{(f)} - n_{(l,p)}^{(g)}]^2$, $accDiff^{\ell}(f,g) = \sum_{l=\ell}^{L} Diff^l(f,g)$, ℓ represents the current level, L is the height of the error tree, and $n_{(l,p)}^{(f)}$ is used to represent the coefficient at level l having offset p of time series f.

After the m^{th} camera calculates these bounds, the lower and upper bounds of $\min_{g \in S_t} \text{Diff}_2(f,g) \forall f \in V_m$ are the following:

$$LB_{sin}(f, S_t) = \min_{g \in S_t} LB(f,g), \tag{7}$$

$$UB_{sin}(f, S_t) = \min_{g \in S_t} UB(f,g). \tag{8}$$

Then, the mth camera returns the range of Video-MR$(f) \forall f \in V_m$ back to server by these bounds and the first term of Eq. (5) computed locally. Using such information, the server then knows that the frame f should be picked into S_{t+T}, dropped, or kept to the next round in MsWave depending on the relation between $D_{threshold}$ and this range. This procedure proceeds by sending each layer of coefficients iteratively until all frames in V_m are picked or dropped. By using MsWave for every camera, we can pick all keyframes with Video-MR larger than $D_{threshold}$ into S_{t+T} without sending the whole S_t to each camera, which helps save significant bandwidth.

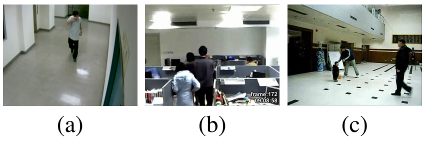

Fig. 1. The example images of each dataset: (a) BL-7F, (b) Office 1, and (c) Lobby.

IV. Experiments

A. Dataset

We used two multi-view summarization datasets given in [6] for our experiments: *Office 1* and *Lobby*, which consist of four and three videos respectively taken by cameras with overlapped field-of-view in an office room and a lobby. We also installed 19 cameras in the 7th floor of the BarryLam Building in National Taiwan University to collect our own dataset, *BL-7F*. Fig. 1 shows the example images of each dataset.

In the experiments, $T = 10$ frames and $\lambda = 0.2$ is selected for our system. Note that the selection of T should be determined by the memory limitation of sensors. Larger T requires more memory space, but generates more optimal results.

B. Baseline

Similar to other summarization works [3], [4], [7], we implemented several baseline algorithms for comparison, including three single-view methods: random sampling (RS), uniform sampling (US), and visual attention (VA)-based [5] method, and one multi-view method: k-mean clustering (KM). We also implemented the original Video-MMR (MMR) [8] using Eq. 5 for comparison. For single-view methods, summarization is performed locally in each sensor, and the multi-view summary is generated by combining all keyframes from all single-view results.

The visual attention based [5] algorithm selects keyframes by computing the attention index of each frame first, and the frames with large attention index are picked. In our implementation, we select keyframes by detecting peaks in the attention curve. As a result, the algorithm runs on-line and generates single-view summarization.

The k-means clustering method selects keyframes by performing clustering on all frames first, and the frames closest to each cluster center are picked. To exploit the redundancy across different views, we perform the clustering on all frames from all videos. Such method can be seen as an off-line, centralized approach for multi-view summarization.

C. Result

As suggested by many previous works [7][15], we use events recall and precision as an objective for evaluation.

TABLE I

THE EXPERIMENTAL RESULTS. *Data Sent* STANDS FOR THE SIZE OF EXTRA DATA, E.G. FEATURES, SENT WHEN COMPUTING THE SUMMARIZATION BY THE PERCENTAGE COMPARED WITH CENTRALIZED SOLUTIONS, IN WHICH FEATURES OF ALL FRAMES ARE SENT.

	Single-view			Multi-view		
	RS	US	VA	KM	MMR	Ours
BL-7F						
Keyframe	77	77	82	77	77	77
Recall (%)	22	30	74	74	67	74
Precision (%)	10	15	90	85	68	68
F1 Score	0.14	0.20	0.81	0.79	0.67	0.71
Redundant Frame	1	3	64	38	36	32
Data Sent (%)	0	0	0	100	100	33
Office 1						
Keyframe	94	94	116	94	94	94
Recall (%)	13	18	52	52	66	63
Precision (%)	6	6	51	64	66	41
F1 Score	0.08	0.09	0.51	0.57	0.66	0.50
Redundant Frame	2	0	44	45	38	21
Data Sent (%)	0	0	0	100	100	26
Lobby						
Keyframe	70	70	117	70	70	70
Recall (%)	66	63	72	72	64	76
Precision (%)	43	45	79	75	71	64
F1 Score	0.52	0.53	0.75	0.73	0.67	0.69
Redundant Frame	8	11	69	29	28	14
Data Sent (%)	0	0	0	100	100	16

Higher recall implies more important information is preserved. We also show the number of the keyframes and the redundant keyframes. Good summary should have high recall with small number of keyframes. Redundant keyframes are those keyframes that capture the same events thus should be minimized. The procedure used in [7] was applied, where ground-truth events are labeled manually for each dataset first. The evaluation results are shown in Table I.

As expected, the random sampling method and the uniform sampling method generate the worst summarization results on all datasets since both of them do not exploit the content of the videos. Note that since both methods perform summarization locally in each sensor, there is no need to send any extra data to the server when performing the summarization.

The visual attention-based [5] algorithm generate acceptable result for a single video. Since the summarization is performed locally in each sensor, no extra data is required to be sent to the server. However, since the method does not exploit the redundancy between different views, the summarization of multiple videos may contain many redundant keyframes. As a result, this approach requires more keyframes than the k-means approach, as shown in Table I.

The k-means clustering method and Video-MMR method generate multi-view summarization centralized and off-line. Since they exploit all video frames at different time, both of them generate more compact summaries compared with single-view methods. However, to perform summarization, features of all frames are required to be streamed back to the server, and all videos are required to be buffered. The communication and the storage overhead can be very large.

As shown in Table I, the performance of our method is similar to k-means and original Video-MMR. However, with the help of MsWave, our method only transmits $\frac{1}{3} \sim \frac{1}{6}$ of the data to perform summarization. Since our method runs online, only a small buffer is required for each sensor comparing to centralized methods.

V. Conclusion

In this paper, we have proposed a distributed multi-view keyframe selection system that can be applied to the video sensor network to help saving bandwidth and server storage. In the era of IoT where image or video sensors are likely to be pervasive, we believe the proposed algorithm can be a good initial solution for online, distributed summarization.

VI. Acknowledgement

This work was also supported by Ministry of Science and Technology, National Taiwan University and Intel Corporation under Grants MOST102-2911-I-002-001 and NTU103R7501.

References

[1] B. T. Truong and S. Venkatesh, “Video abstraction: A systematic review and classification,” *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 3, no. 1, Feb. 2007.

[2] A. G. Money and H. Agius, “Video summarisation: A conceptual framework and survey of the state of the art,” *J. Visual Commun. and Image Representation*, vol. 19, no. 2, pp. 121 – 143, 2008.

[3] A. Khosla, R. Hamid, C.-J. Lin, and N. Sundaresan, “Large-scale video summarization using web-image priors,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, Jun. 2013, pp. 2698–2705.

[4] Y. Cong, J. Yuan, and J. Luo, “Towards scalable summarization of consumer videos via sparse dictionary selection,” *IEEE Trans. Multimedia*, Feb. 2012.

[5] J. Peng and Q. Xiaolin, “Keyframe-based video summary using visual attention clues,” *IEEE MultiMedia*, vol. 17, no. 2, pp. 64–73, 2010.

[6] Y. Fu, Y. Guo, Y. Zhu, F. Liu, C. Song, and Z.-H. Zhou, “Multi-view video summarization,” *IEEE, Trans. Multimedia*, vol. 12, no. 7, pp. 717–729, Nov. 2010.

[7] P. Li, Y. Guo, and H. Sun, “Multi-keyframe abstraction from videos,” in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2011, pp. 2473–2476.

[8] Y. Li and B. Merialdo, “Multi-video summarization based on Video-MMR,” in *Int. Workshop Image Anal. Multimedia Interactive Services*, Apr. 2010, pp. 1–4.

[9] C. Leo and B. Manjunath, “Multicamera video summarization from optimal reconstruction,” in *Asian Conf. Computer Vision Workshop*, vol. 6468, 2011, pp. 94–103.

[10] J.-P. Wang, Y.-C. Lu, M.-Y. Yeh, S.-D. Lin, and P. B. Gibbons, “Communication-efficient distributed multiple reference pattern matching for M2M systems,” in *Proc. of IEEE Int. Conf. on Data Mining*, 2013.

[11] C. Sujatha and U. Mudenagudi, “A study on keyframe extraction methods for video summary,” in *Int. Conf. Computational Intell. and Commun. Networks (CICN)*, 2011, pp. 73–77.

[12] J. Carbonell and J. Goldstein, “The use of MMR, diversity-based reranking for reordering documents and producing summaries,” in *Proc. 21st Annu.Int. ACM SIGIR Conf. Research and Develop. in Inform. Retrieval*, ser. SIGIR ’98, 1998, pp. 335–336.

[13] Z. Zivkovic, “Improved adaptive gaussian mixture model for background subtraction,” in *Proc. 17th Int. Conf. Pattern Recognition*, vol. 2, 2004, pp. 28–31 Vol.2.

[14] A. Haar, “Zur theorie der orthogonalen funktionensysteme,” *Mathematische Annalen*, vol. 69, 1910.

[15] C. Cotsaces, N. Nikolaidis, and I. Pitas, “Video shot detection and condensed representation. a review,” *IEEE Signal Process. Mag.*, vol. 23, no. 2, pp. 28–37, 2006.