Embedded Sparse Coding for Summarizing Multi-View Videos (Supplementary Material)

Rameswar Panda, Abir Das, Amit K. Roy-Chowdhury

Abstract

This supplemental document provides additional information that does not fit in the paper due to the space limit. Section 1 provides the detailed statistics and source of the experimented datasets. Section 2 presents supplemental experiment results. Section 3 gives detailed information on the QL based MM algorithm that is used to solve Eq. 3 of the main paper. Section 4 gives detail explanation on the fast proximal algorithm (FISTA) to solve Eq. 6 in the sparse representative selection (Section. 3) of the main paper.

1. Detailed information on the Datasets

Three multi-view datasets have been used in our experiments. The datasets are named *Office*, *Campus* and *Lobby*. Their statistics and sources are given in Table 1. Some example frames of the datasets are shown in Fig. 1.

2. Supplemental experiment results

2.1. Summarized events for all the datasets

Due to space limit in the main paper, we have shown only eight summarized events for the *Office* dataset (Figure 4 in the main paper). Here, we will present all the detected events in our generated summaries for all the datasets. Fig. 3, 4 and 5 present the summarized events for *Office*, *Campus* and *Lobby* dataset respectively. The detected events are assembled along the time line across multiple views and are represented by a key frame. In figure, X-axis denotes the time line and the Y-axis represent the view (camera) from which the event is detected.

2.2. Scalability in generating summaries

As mentioned in section 3 of the main paper, our approach provides scalability in generating summaries of different length based on the user constraints without any further analysis of the input videos. Specifically, the non-zero rows of C (sparse coefficient matrix) generate a ranked list of representatives that is subsequently used to provide a scalable representation in generating summaries. In this section, we will present an illustrative example to show the scalability in generating summaries for the *Office* dataset. As per the ground truth marked in [1], the *Office* dataset contains total 26 events throughout the entire duration. Consider a scenario where three user were interested to watch the summary but with different summary length. Let the 1st user want to see only 3 most important events while at the same time, 2nd and 3rd user want to see 5 and 7 most important events respectively occurred in the whole day. Fig. 6 in this connection explains the generated summaries for the *Office* dataset.

Table 1. Dataset statistics and sources. NJU: http://cs.nju.edu.cn/ywguo/summarization.html

Datasets	Number of Views	Video Length(Mins.)	Settings	Camera Type	Source
Office	4	11:16/08:43/11:22/14:58	Indoor	Fixed	NJU
Campus	4	15:19/13:51/12:30/15:03	Outdoor	Non-fixed	NJU
Lobby	3	08:14/08:14/08:14	Indoor	Fixed	NJU

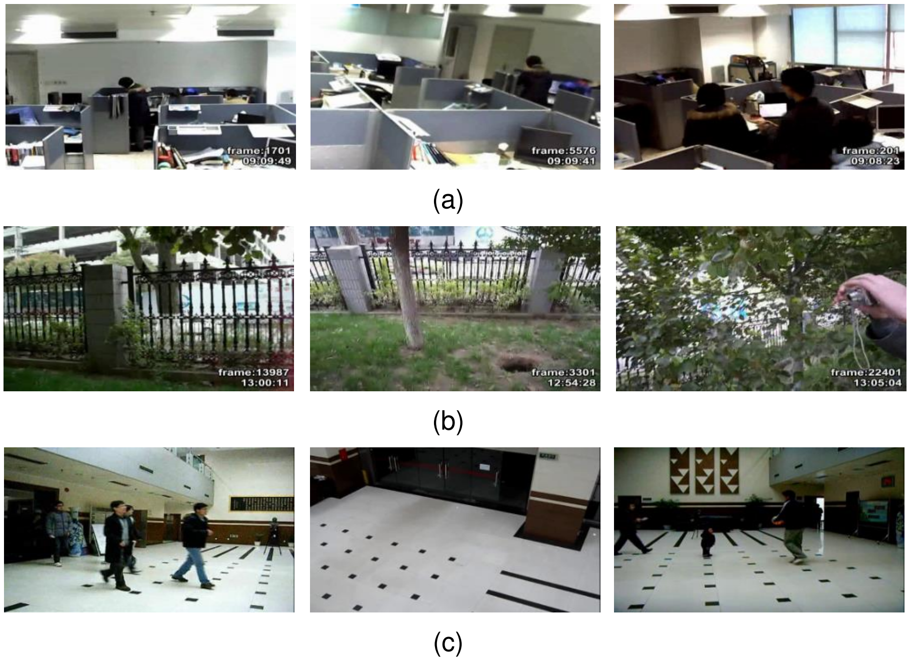

Figure 1. Example frames of the datasets used in the experiments. (a):Office, (b): Campus, and (c): Lobby. As can be seen from figure, the datasets (a) and (c) are captured with fixed cameras in indoor environments whereas the dataset (b) is taken with non-fixed cameras in an outdoor environments.

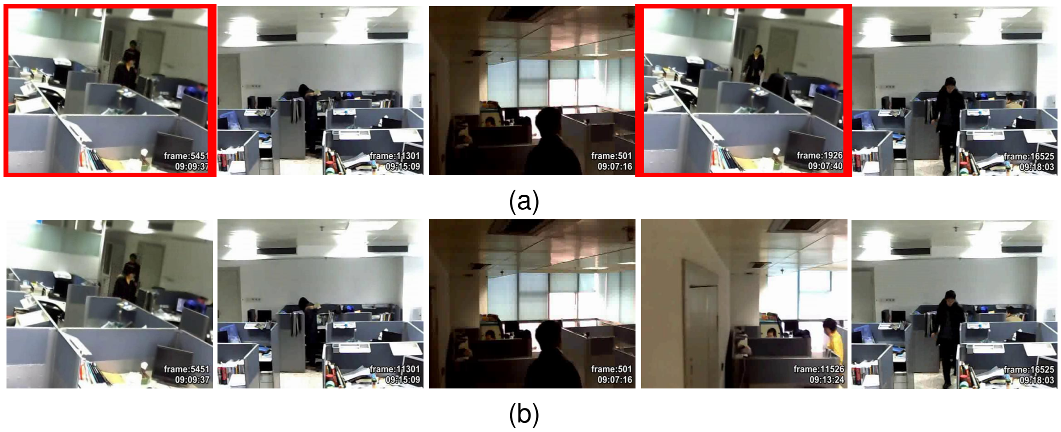

Figure 2. Representative events detected by our method with and without SLH algorithm for *Office* dataset. (a): Our approach without SLH algorithm i.e., only Gaussian kernel is used to calculate the inter-view proximities and (b): Our approach with SLH algorithm. Each event is represented by a key frame and are arranged according to the l_2 norms of corresponding non-zero rows of the sparse coefficient matrix. As an illustration, only top five events are shown in the figure. In (a), both 1st and 4th frame represents the same event as per the ground truth but are detected from view 1 and view 2 respectively. In total, the method without SLH algorithm detects only 15 unique events in contrast to 18 unique events with the use of SLH algorithm. Redundant events are marked with red color borders

2.3. Performance Analysis with Scott and Longuet-Higgins (SLH) algorithm

In this section we make a comparative performance analysis to show the effect of using SLH algorithm in generating better summaries. Fig. 2 shows a comparative analysis of our method with and without using SLH algorithm in computing inter-view proximities for Office dataset. As seen from the Fig. 2(a) the method in absence of SLH algorithm produces redundant events in the output summary however the same method with SLH algorithm produces meaningful summaries (Fig. 2(b)). SLH algorithm tries to maintain the consistency in group sparse coding which guarantees that two similar frames from two different views are not produced at the same time.

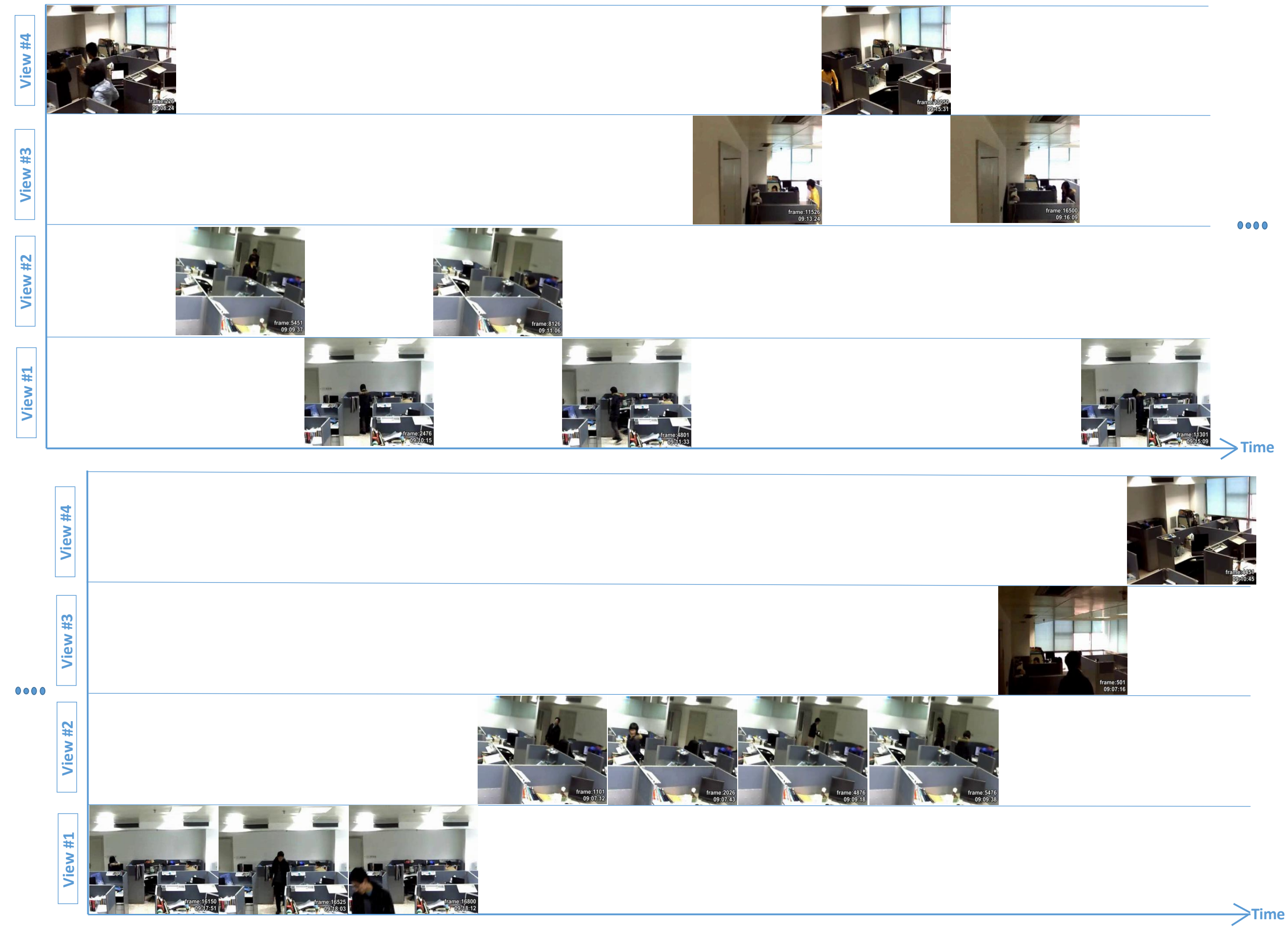

Figure 3. Summarized events for the *Office* dataset. X-axis denotes the time line and the Y-axis represent the view (camera) from which the event is detected. Each event is represented by a key frame. As per the ground truth: A0 represents a girl with a black coat, A1 represents the same girl with a yellow sweater and B0 indicates another girl with a black coat. C and D are two boys. D wears a black topcoat and C wears a dark yellow sweater. E is a old man and F is a young guy about thirty years old. The sequence of events in our summary are: 1st: A0, B, and D go out of the room, 2nd: A0 enters the room, 3rd:A0 stands in Cubicle 1, 4th: A0 is looking for a thick book to read, 5th: A0 leaves the room, 6th: A1 enters the room and stands in Cubicle 1, 7th: A1 goes out of the Cubicle, 8th: B0 enters the room and goes to Cubicle 1, 9th: B0 goes out of the Cubicle, 10th: B0 goes out of the room and D enters the room, 11th: D walks to Cubicle 2 from Cubicle 1, 12th: D walks to Cubicle 3 from Cubicle 2, 13th: F enters the inner office, 14th: A0 is looking for a book in Cubicle 3, 15th: F leaves the room, 16th: E leaves the room, 17th: F enters the room and 18th: The computer screen in Cubicle 2 turns off. As can be seen from the figure, Only 18 events out of 26 events are detected in our summary. It can be noticed that most of the events are detected from the view 1 and 2 as both of the cameras were focused to most activity region in the Fovs which can also be seen from the input videos. (Best viewed in color)

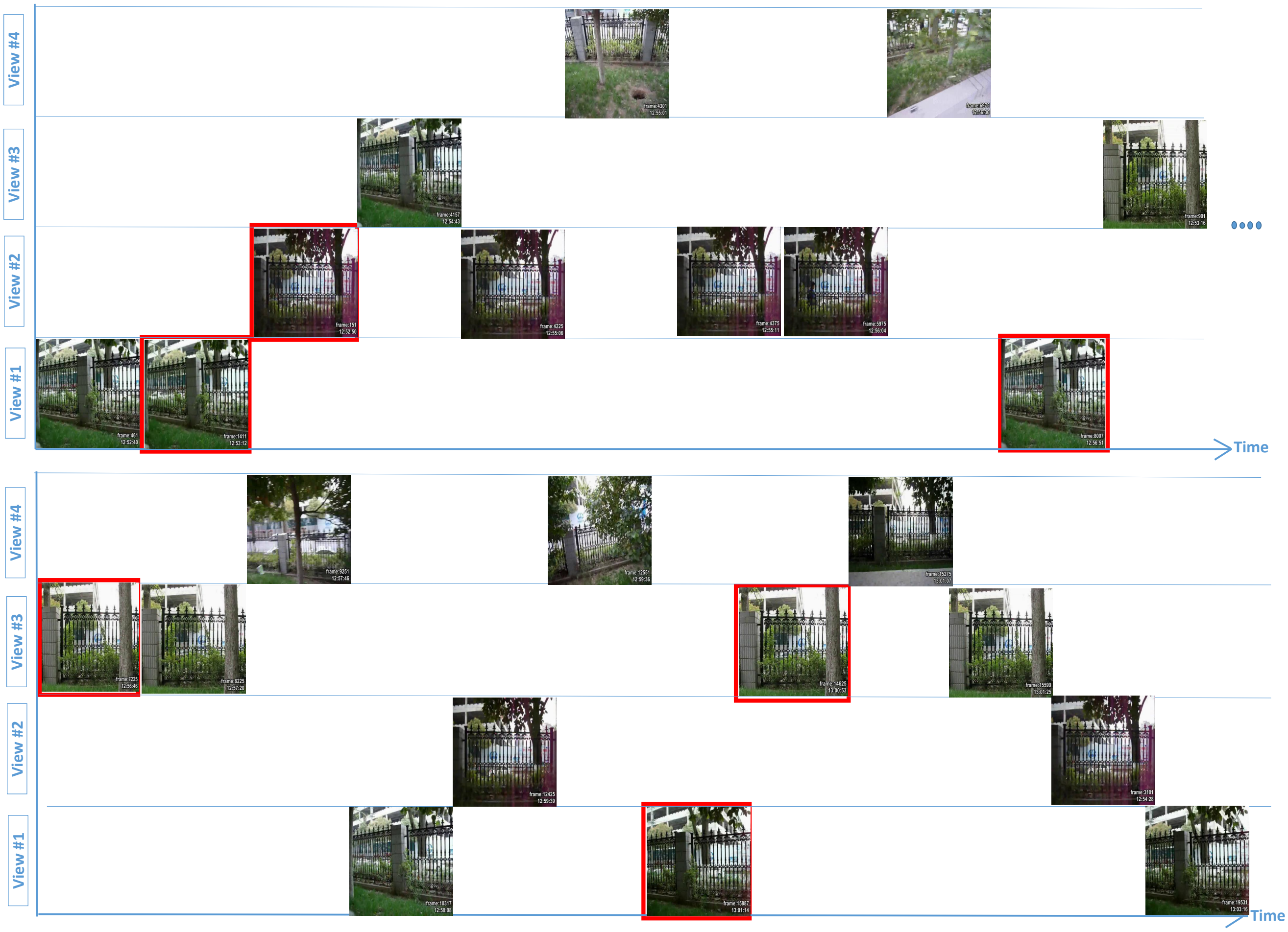

Figure 4. Summarized events for the *Campus* dataset. X-axis denotes the time line and the Y-axis represent the view (camera) from which the event is detected. Each event is represented by a key frame. For this dataset, only start and end frame from an event is mentioned without any labeling. As can be seen from the figure, there exists some redundancies in our output summary as the video is captured using 4 hand-held cameras in an outdoor environment which makes the summarization difficult. Redundant events are marked with red color borders. Only 20 events out of 23 detected events are unique events in our summary. The dataset contains total 29 events.(Best viewed in color)

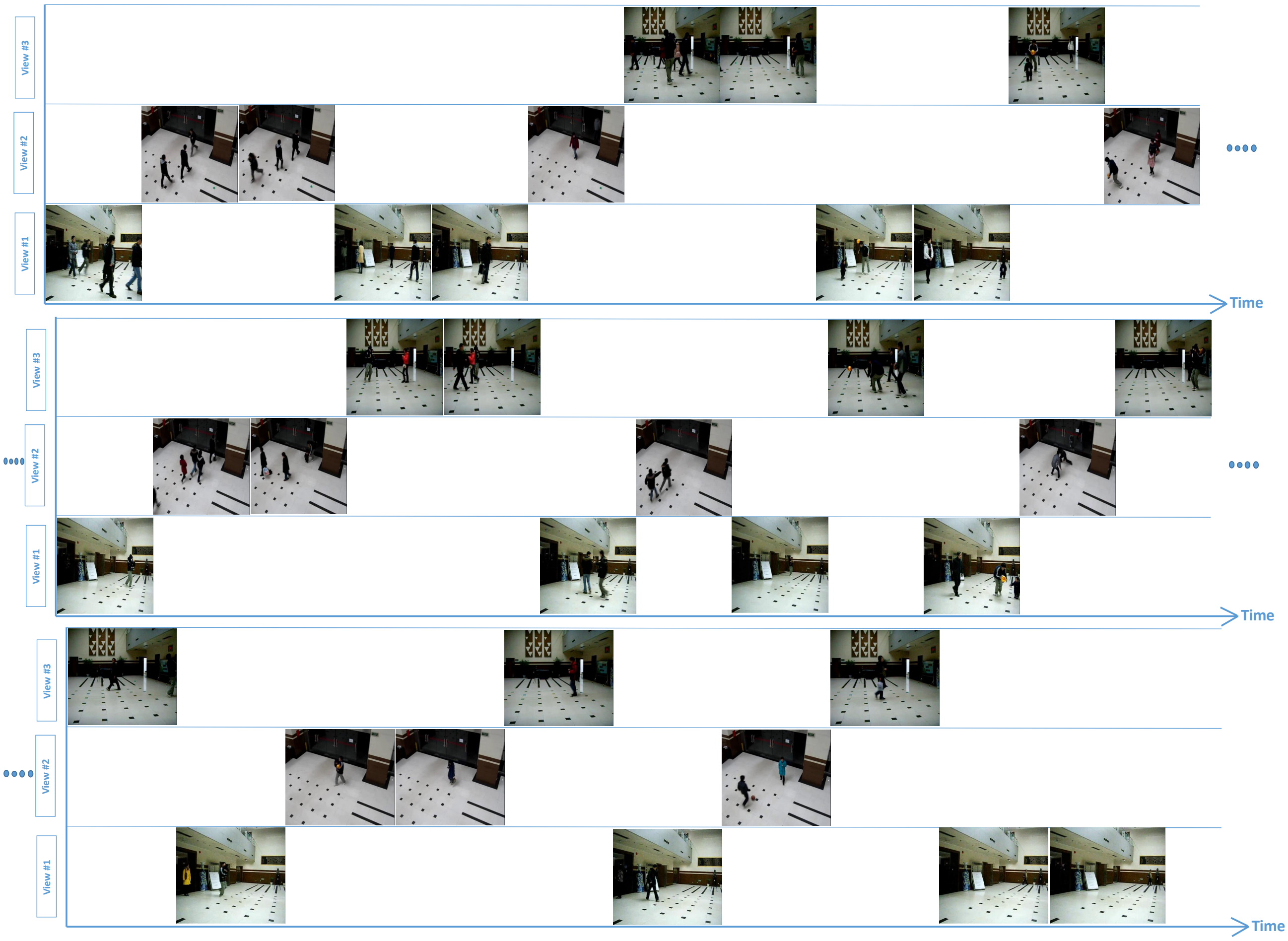

Figure 5. Summarized events for the *Lobby* dataset. X-axis denotes the time line and the Y-axis represent the view (camera) from which the event is detected. Each event is represented by a key frame. The sequence of events in our summary are: 1st: Five persons walk across the lobby towards the gate; a man runs to the gate, 2nd: Two men walks across the lobby towards the gate, and a man walks into the lobby, 3rd: A man run into the lobby from the gate, 4th: Four persons walk into the lobby from the gate, 5th: A0 leaves the room, 6th: A man walks across the lobby towards the gate and at the same time Two men are talking in the lobby, 7th: A man wearing a red coat walks into the lobby from the gate, 8th: Three men and a men with a baby in his arms walk across the lobby towards the gate. A man walk into the lobby, 9th: A man plays a ball with a baby, 10th: A man wearing a black coat walks into the lobby from the gate. A man plays a ball with baby, 11th: A woman wearing a white coat walks across the lobby towards the gate, 12th: A man plays a ball with a baby, 13th: Two women and a man walk across the lobby from the gate, 14th: A man walks across the lobby towards the gate, 15th: Two men walk across the lobby towards the gate, 16th: Two men wearing black coats walk into the lobby from the gate, 17th: A woman wearing a red coat walks across the lobby quickly towards the gate, 18th: A man with a baby in his arms, and four other persons walks into the lobby from the gate, 19th: Two men are running in the lobby to catch each other, 20th: A man with a baby in his arms walks towards the lobby from the inner room, 21st: Two men are passing a basketball to each other, 22nd: A man wearing a black coat walks across the lobby towards the gate, 23rd: Two men are playing basketball, 24th: A man with a baby in his arms walks into the lobby from the gate, 25th: A man with a briefcase taken in hand walks into the lobby from the gate, 26th: A man with a baby in his arms walks across the lobby. And a woman wearing a yellow coat also walks across the lobby, 27th: A man with a baby in his arms runs in the lobby, 28th: A woman wearing a blue coat walks into the lobby slowly, 29th: A man wearing a red coat walks across the lobby towards the gate, 30th: A man wearing a black coat walks across the lobby towards the gate, 31st: A man runs across the lobby quickly with a basketball rolling on the ground, 32nd: A woman wearing a black coat walks across the lobby towards the gate, 33rd: A man is wandering in the lobby, and 34th: A man is going towards the gate across the lobby. As can be seen from the figure, Only 34 events out of 43 events are detected in our summary. (Best viewed in color)

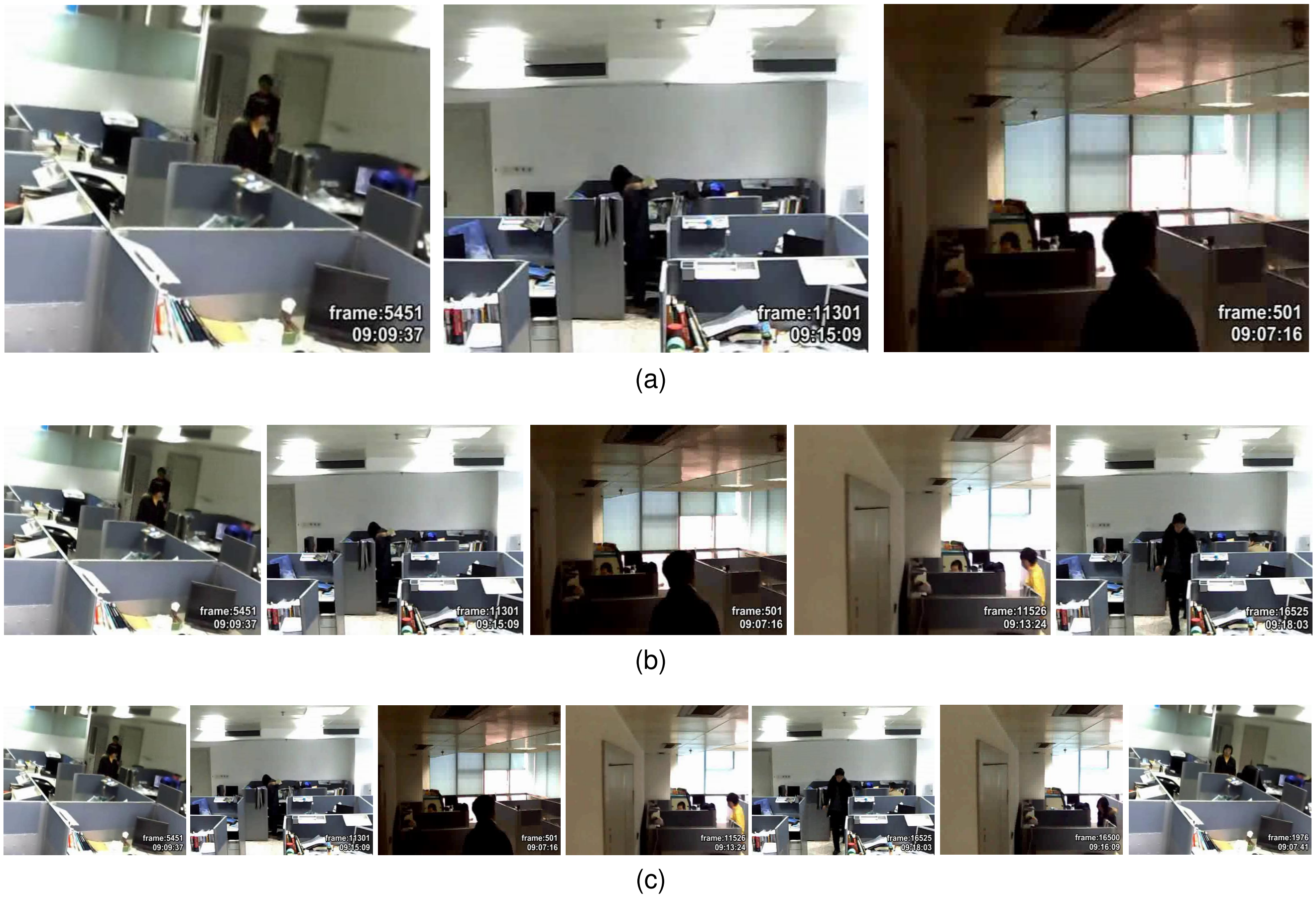

Figure 6. The figure shows an illustrative example of scalability in generating summaries of different length based on the user constraints for the *Office* dataset. Each event is represented by a key frame and are arranged according to the l_2 norms of corresponding non-zero rows of the sparse coefficient matrix. (a): Summary for user length request of 3, (b): Summary for user length request of 5 and (c): Summary for user length request of 7. (Best viewed in color)

3. Detailed Information on QL based MM Algorithm to Solve Eq. 3 of Main Paper

The objective function in Eq.3 of main paper is not convex and gradient descent algorithm can be used to find the local solution. However, constant step sizes in gradient algorithm do not guarantee decrease of the objective; expensive line searches are often needed to decrease the objective in successive steps. In contrast, algorithms such as Majorization-Minimization would be guaranteed to monotonically decrease the cost in each update. MM algorithms are based on finding a tight axillary upper bound of a cost function and then minimizing the cost by analytically updating the parameters at each step.

Recently, Zhirong *et al.* [2] proposed a two phase Quadratification-Lipschitzation (QL) procedure to efficiently find upper bounds for MM which can be analytically minimized. Consider an objective function $\mathcal{F}(\widetilde{Y})$ which can be divided into two parts *i.e.*, $A(P,\widetilde{Q}) + B(P,\widetilde{Q})$, where A has a quadratic upper bound (Quadratification) of the form $A(P,\widetilde{Q}) \leq \sum_{i,j} W_{ij}\big(\|\tilde{y}_i - \tilde{y}_b\|^2\big)$+constant and B has a upper bound by its Lipschitz surrogate (Lipschitzation) of the form $B(P,\widetilde{Q}) \leq \big\langle \Psi, \tilde{Y} - Y \big\rangle + \dfrac{\rho}{2}\|Y - \widetilde{Y}\|^2$+constant, where W_{ij} is a multiplier, $\Psi = \frac{dB}{d\widetilde{Y}}\Big|_{\widetilde{Y}=Y}$ and ρ is the Lipschitz constant of $B(P,\tilde{Q})$. The resulting upper bound on combining both can be minimized analytically by setting the gradient to zero.

In our case with $\sum_{ij} Pij = 1$, we have

$$A(P,\widetilde{Q}) = \sum_{ij} Pij \ln\big(1 + \|\tilde{y}_i - \tilde{y}_j\|^2\big) \tag{1}$$

$$B(P,\widetilde{Q}) = \ln\sum_{ij}\big(1 + \|\tilde{y}_i - \tilde{y}_j\|^2\big)^{-1} \tag{2}$$

Quadratification. Since $A(P,\widetilde{Q})$ is concave in $\|\tilde{y}_i - \tilde{y}_j\|^2$, it can be upper-bounded by its tangent. Hence, $A(P,\widetilde{Q}) \leq \sum_{ij} P_{ij}q_{ij}\|\tilde{y}_i - \tilde{y}_j\|^2$, where $q_{ij} = \big(1 + \|y_i - y_j\|^2\big)^{-1}$.

Lipschitzation. $B(P,\widetilde{Q}) \leq \big\langle -4L_{Q\circ q}Y, \widetilde{Y} - Y\big\rangle + \dfrac{\rho}{2}\|Y - \widetilde{Y}\|^2$, where $L_{Q\circ q}$ denotes the Laplacian of the subscripted matrix and $\circ$ denote the element-wise product.

The final upper bound of Eq.3 and the gradient of the upper bound with respect to $\tilde{Y}$ are given by

$$\widehat{\mathcal{F}}(\widetilde{Y}, Y) = \sum_{ij} P_{ij}q_{ij}\|\tilde{y}_i - \tilde{y}_j\|^2 + \big\langle -4L_{Q\circ q}Y, \widetilde{Y} - Y\big\rangle + \frac{\rho}{2}\|Y - \widetilde{Y}\|^2 \tag{3}$$

where Y, $\tilde{Y}$ and Y^{new} denote current estimate, the variable and new estimates of Y respectively. ρ is the Lipschitz constant of $B(P,\tilde{Q})$ and $q_{ij} = \big(1 + \|y_i - y_j\|^2\big)^{-1}$. $L_{Q\circ q}$ denotes the Laplacian of the subscripted matrix and $\circ$ denote the element-wise product. Finally, the update rule on zeroing the gradient of upper bound, is given as

$$Y^{new} = \big(L_{P\circ q} + \frac{\rho}{4}I\big)^{-1}\big(L_{Q\circ q}Y + \frac{\rho}{4}I\big) \tag{4}$$

The update rule in Eq. 7 is used to yield the mapped locations of the frames in the embedding space. Since, the calculation of Lipschitz constant for $B(P,\tilde{Q})$ is mathematically intractable, we use a simple backtracking approach to impose point-wise maximum at each step. The computational cost of MM algorithm is $O(NlogN)$ where N is the total number of frames in the multi-view videos. We set $c = 2$, $\rho = 10^{-6}$ and $\delta = 10^{-7}$ throughout all experiments.

3.1. Upper bound of $\mathbf{A}(\mathbf{P}, \widetilde{\mathbf{Q}})$

We described that $A(P,\widetilde{Q})$ is concave in $\|\tilde{y}_i - \tilde{y}_j\|^2$ and the upper bound is given by $A(P,\widetilde{Q}) \leq \sum_{ij} P_{ij}q_{ij}\|\tilde{y}_i - \tilde{y}_j\|^2$, where $q_{ij} = \big(1 + \|y_i - y_j\|^2\big)^{-1}$.

Proof. As per the above Equation.1,

$$A(P,\widetilde{Q}) = \sum_{ij} Pij \ln\big(1 + \|\tilde{y}_i - \tilde{y}_j\|^2\big) \tag{5}$$

Since, $\ln\big(1 + \|\tilde{y}_i - \tilde{y}_j\|^2\big)$ is concave in $\|\tilde{y}_i - \tilde{y}_j\|^2$, it can be upper-bounded by its tangent [3]:

$$\begin{aligned}\ln\big(1+\|\tilde{y}_i-\tilde{y}_j\|^2\big) &= \ln\big(1+\|y_i-y_j\|^2\big)+\Big\langle\frac{d\ln\big(1+\|\tilde{y}_i-\tilde{y}_j\|^2\big)}{d\|\tilde{y}_i-\tilde{y}_j\|^2}\Big|_{\tilde{Y}=Y},\tilde{Y}-Y\Big\rangle\\ &=\Big\langle\frac{d\ln\big(1+\|\tilde{y}_i-\tilde{y}_j\|^2\big)}{d\|\tilde{y}_i-\tilde{y}_j\|^2}\Big|_{\tilde{Y}=Y},\|\tilde{y}_i-\tilde{y}_j\|^2-\|y_i-y_j\|^2\Big\rangle+constant\\ &=\Big\langle\frac{1}{1+\|y_i-y_j\|^2},\|\tilde{y}_i-\tilde{y}_j\|^2\Big\rangle+constant\\ &=\frac{\|\tilde{y}_i-\tilde{y}_j\|^2}{1+\|y_i-y_j\|^2}+constant \end{aligned}\tag{6}$$

Now, from Eq. 5 and 6 , we get:

$$\begin{aligned}A(P,\widetilde{Q}) &\leq \sum_{ij}Pij\frac{\|\tilde{y}_i-\tilde{y}_j\|^2}{1+\|y_i-y_j\|^2}\\ &\leq \sum_{ij}P_{ij}q_{ij}\|\tilde{y}_i-\tilde{y}_j\|^2,\end{aligned}\tag{7}$$

where $q_{ij}=\big(1+\|y_i-y_j\|^2\big)^{-1}$.

4. Detailed Information on FISTA to Solve Eq. 6 of Main Paper

Here we state the optimization strategy to solve the convex optimization problem in Eq. 6 of main paper. We use a fast proximal algorithm, FISTA [4] which maintains two variables in each iteration and combines them to find the solution. It finds the minimum of a cost function of the form $g(C)+h(C)$ where g is convex, differentiable but h is closed, convex and non-smooth. New value of the variable, in each iteration is computed by computing the proximal operator of h on a function of the gradient of g. The proximal operator of $h(C)$, denoted as $\text{Prox}_h(C)$ is computed as,

$$\text{Prox}_h(C)=\underset{U\in\mathbb{R}^{N\times N}}{\text{argmin}}\ \big(h(U)+\frac{1}{2}||U-C||^2\big)\tag{8}$$

In Eq. 6 of the main paper, the reconstruction error term is convex, smooth, differentiable function with Lipschitz continuous gradients. Denoting $||Y-YC||_F^2$ as $g(C)$, the gradient $\nabla g(C)$ and the Lipschitz constant L_g of the gradient are given by

$$\begin{aligned}\nabla g(C) &= \big(-Y^TY+Y^TYC\big)\\ L_g &= \big(||Y^TY||_F^2\big)\end{aligned}\tag{9}$$

The sparsity inducing $\ell_{2,1}$ norm is convex but non-smooth functions of their respective variables. Let us denote the non-smooth term $\lambda||C||_{2,1}$ as $h(C)$ and the corresponding proximal operators for this non-smooth function as,

$$\text{Prox}_h(C)=\big(1-\frac{\lambda}{||C^i||_2}\big)_+C^i\tag{10}$$

where i denotes the row numbers with $(C)_+\triangleq\max(C,0)$. Taking a fixed step size α equal to the inverse of the Lipschitz constant L_g, FISTA guarantees a convergence rate of $O\big(\frac{1}{T^2}\big)$ (T is the number of iterations), in contrast to traditional sub-gradient methods of $O\big(\frac{1}{\sqrt{T}}\big)$ to solve non-smooth optimizations.

5. Derivation of Lipschitz constant

This section contains the derivations of the derivatives and Lipschitz's constant that are used in FISTA algorithm to solve the Eq. 6 of the main paper.

$$\begin{aligned} g(C) &= ||Y - YC||_F^2 \\ &= tr\big[(Y - YC)(Y - YC)^T\big] \\ &= tr(YY^T - YC^TY^T - YCY^T + YCC^TY^T) \\ &= tr(YY^T) - tr(YC^TY^T) - tr(YCY^T) + tr(Y^TYCC^T) \end{aligned} \tag{11}$$

Now,

$$\begin{aligned} \nabla g(C) &= 0 - Y^TY - Y^TY + 2Y^TYC \\ &= 2(-Y^TY + Y^TYC) \end{aligned} \tag{12}$$

At the same time the Hessian of $g(C)$ is given by,

$$\nabla^2 g(C) = Y^TY \tag{13}$$

To find the Lipschitz's constant of $g(C)$, we will use the following three facts.

- Consider a mapping $f : \mathbb{R}^d \to \mathbb{R}^n$. If $||\nabla^2 f||$ is bounded in the operator norm on Ω, i.e., $L = \sup_X ||\nabla^2 f(X)||$, then $||\nabla^2 f(X) - \nabla^2 f(Y)|| \leq L||X - Y||, \forall X, Y \in \Omega$ i.e., ∇f is Lipschitz continuous with Lipschitz constant L.
- $g(C)$ has constant Hessian and thus the supremum of the operator norms of the Hessian will be the norm of the corresponding constant Hessian.

Considering operator-2 norm, the Lipschitz constant for $g(C)$ is then given by $||Y^TY||_2$. Again, considering the ease of computation and the fact that the Frobenius norm of a matrix is greater than or equal to its operator 2-norm, we decided to take the Lipschitz's constant as the corresponding Frobenius norms instead of the 2-norms. This is not as tighter a bound as the corresponding operator-2 norm would have been.

Thus, the Lipschitz constant L_g of the gradient of $g(C)$ is given by,

$$L_g = \big(||Y^TY||_F\big) \tag{14}$$

.

References

[1] Y. Fu, Y. Guo, Y. Zhu, F. Liu, C. Song, and Z. H. Zhou, "Multi View Video Summmarization," *TMM*, vol. 12, no. 7, pp. 717–729, 2004. 1

[2] Z. Yang, J. Peltonen, and S. Kaski, "Majorization-minimization for manifold embedding," in *AISTATS*, 2015. 7

[3] Stephen Boyd and Lieven Vandenberghe, *Convex optimization*, Cambridge university press, 2004. 7

[4] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009. 8