# CCTV Footage Summarization

Tanishk Sachdeva
*Department of Information Technology*
*MSIT, Janakpuri, New Delhi, India*

Dr. Bharti Sharma
*Department of Information Technology*
*MSIT, Janakpuri, New Delhi, India*

Gunjan
*Department of Computer Science & Engineering*
*MSIT, Janakpuri, New Delhi, India*

**Abstract- The use of CCTVs has been on the steady rise in within the last decade, getting used for the safety of public and personal establishments alike. With increasing number of accidents & incidents in cities, there's a requirement for a better method to survey the video being recorded, instead of manually browsing hours of footage, saving both time and labor. We propose an efficient method to get a summary of sparse CCTV footage (footage with events occurring very rarely). A temporal re-arrangement of events increases the density of summary, consequently reducing the duration of the summary further than what's possible with motion-detection. The project also aims to simplify the task further, by generating video summaries from a user query supported time and objects, thus showing exactly what's required.**

**Keywords – CCTV, video summary, computer vision, image processing**

## I. Introduction

The number of CCTVs surveillance cameras is increasing everyday resulting in an enormous amount digital video information being captured and stored. Many CCTV cameras run 24 hours each day, sometimes even streaming the content over the web for people to watch. This data is however during a raw, unprocessed format. In most cases, video with little to no motion is being captured, which wastes lot of storage. The method of watching or analyzing the footage is additionally time consuming and laborious.

We propose to create a CCTV video summarizer to eliminate this problem in stored data and provides the user a brief summary [1], [2], [3], [4] consisting of important information that's relevant for the precise use case. The system proposes to include a query-based summary generation to get more relevant summaries. Users can choose the specified objects and events to get a brief yet precise summary with only relevant information, saving longer. The method takes place over two phases, the real-time and query phase. The real-time phase reads the CCTV footage, identifies clips of interest, extracts the “flow-tubes”, and stores them after tagging them with the respective object tags. a question phase would then involve the user selecting the specified tags and objects of interest, which are chosen. The tubes are rearranged using simulated annealing algorithm, before being blended into a generated time-lapsed background using an optimal technique like poisson blending. This method reduces the manual work of browsing hours of footage trying to find relevant events by automatically creating a summary.

### *1.1. Real-time phase*

The real-time phase reads CCTV footage, identifies the clips of interest, and introduces specific image processing algorithms on the footage of interest to be stored in the database and tagged from the clips [10] the phase is divided into the following steps:

#### 1.1.1. *Motion Detection*

Find significant motion in the footage and identify any clips with significant motion while ignoring objects due to changing environmental conditions and other significant disruptions. In this step there is no movement to see if the MOG is used, and if there is a significant foreground in the image, which is determined by the static threshold, the clip is assumed to have motion in it. The MOG is especially useful here due to its dynamic nature and the rapid adoption of gradual changes in the environment and, moreover, the availability of efficient implementation of this highly effective algorithm.

#### *1.1.2. Background Masking*

Foreground extractor such as a mixture of Gaussians [8], [11], [12], [16] are used to capture topics of interest in clips identified by motion detection. The same technique used in the previous step is also employed here to make the foreground mask and from there to the foreground only. Many techniques were experimented on MOG. This is the most cost effective and accurate technique available for use case.

#### *1.1.3. Computation of Objects flow-tubes*

In the previous stage, the flow-tube is calculated from the foreground. Morphological operations are performed by racing the flow tube and removing many redundant foreground blobs in this step. Moreover, the individual subjects present in each frame are identified and related to the subjects present in the previous frame, there generates a flow-tube array.

#### *1.1.4. Object Tagging*

After identifying the actual subjects in the previous stage, the subjects are categorized into several popular categories using the popular deep-learning model known as the “you only look once” model, and these tags are calculated. The pre-trained 26-tier YOLOv3 model is used as one of the most common categories presented in normal CCTV video footage, already present in a set of identifiable categories on the standard COCO dataset trained on YOLOv3.

#### *1.1.5. Storage*

At this stage, the connection is established in the database and the metadata such as texts, length, and importance as well as the events are stored in the database.

### *1.2. Query Phase*

Query Phase Processes user input queries, compares the corresponding tube and generates the corresponding summary. This phase is divided into the following steps:

#### *1.2.1. Tube Selection*

A user query with various parameters such as duration, tags and summary length is taken from the user and the corresponding flow-tube is selected from the database. This stage is easily implemented by writing a logic to create a query with all the parameters specified by the user in the input query.

#### *1.2.2. Tube Rearrangement*

An optimization algorithm, in this case, simulated annealing [13], is used to rearrange event tubes over a period of time to summarize the desired length. While many heuristic-based search algorithms are recommended for these purposes by various authors, simulated annealing is the most successful and most popular cited method. Therefore, simulated annealing with custom cost function has been implemented depending on the requirements.

#### *1.2.3. Time-lapsed background generation*

This step produces a background based on the duration and summary length required by the user. A weighted approach, with the period where most of the activity is, it is considered more to create a time-lapse background.

#### *1.2.4. Blending*

Poison mixing [14], [15] is done by mixing reconfigured flow-tubes with a time lapsed background to create a summary video. This summary is then saved to the user's computer. A timestamp is added to the original input to indicate the time of that event.

It are often mainly used for security purposes, by the police forces for detection of crimes and suspicious activities.

## II. LITERATURE REVIEW

Yael Pritch associated Alex Rav-Acha in [5], proposed a way to effectively generate an outline of associate endless video stream which can be used as an index into the most video. An internet section includes tube detection in spatio-temporal domain, insertion of those tubes into associate object queue, and removal upon reaching an area limit. The response section then constructs a time-lapse video of the dynamical background, choice and sewing of tubes into a coherent video. Min-cut algorithmic rule beside background subtraction has been used for extracting moving objects. Activity, collision and temporal consistency prices are used as parameters for optimum tube arrangement.

Shmuel Peleg and Yael Pritch, in [6], have bestowed a dynamic video outline technique wherever most of the activity within the video is condensed by at the same time showing many actions, even after they originally occurred at totally different times.

CRAM: Compact illustration of Action in Movies, Mikel Rodriguez in [7], generated a compact video illustration of an extended sequence, that whereas protective the final dynamics of the video options solely the essential parts. From the given input video, optical flows area unit generated. These area unit then described as vectors in Clifford Fourier domain. Dynamic regions of flow area unit then known inside the section spectrum volume. The probability of activities of relevancy area unit then computed by correlating it with spatiotemporal most average correlation height filters. The ultimate outline is then generated by a temporal-shift improvement. This technique may sight specific actions [8] presented a really flourishing and extremely used technique for adaptive pixel-level background subtraction. Every picture element has chance density perform on an individual basis. A picture element is taken into account to be a part of the background if its new price if well represented by its density perform. This paper was associate improvement on previous models that used Gaussian mixture models with economical update equations. An especially quick object detection model in [9], the YOLO model was represented. Whereas previous object sightors used classifiers to detect, this paper proposed object detection as a regression drawback to dimensionally separated bounding boxes and associated category possibilities. One neural network is employed to predict each bounding boxes and its category chance, creating end-to-end improvement simple. Though YOLO makes a lot of errors, it's less possible to predict false positives compared to different techniques.

## III. METHODOLOGY

The fundamental idea of this project is to generate a short summary which include all the important events to reduce the amount of manual effort and labor. The internal working of certain core modules related to real time phase and query phase is explained in this section. It also construes the functional description of components and sub-components of the system.

### *A. Motion Detection Module*

This is the main module of the system which is responsible identifying clips of interest in a sparse CCTV footage.

- Purpose: The purpose of this module is to determine whether each frame is relevant.
- Input: The input to this module are frames and timestamps of the input video source.
- Output: The output is whether the input should be saved for further processing or ignored
- Functionality: The functionality of this module is to detect motion and save those clips of interest
- Flowchart: The flowchart shown in figure 1 explains the motion detection module. The module starts by reading a frame and applies the stored MOG model onto the frame. If the generated background mask has more foreground pixels than a specified threshold then it is considered as a frame with motion and is stored as clip of interest.

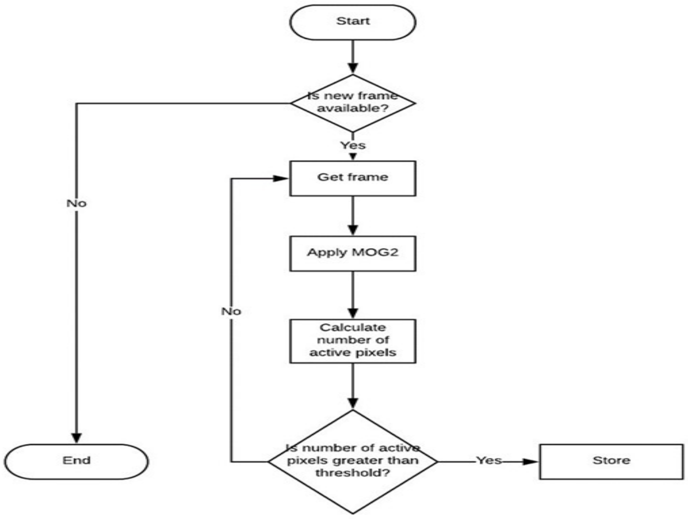

Start
Is new frame available?
Yes
No
Get frame
Apply MOG2
Calculate number of active pixels
No
Is number of active pixels greater than threshold?
Yes
Store
End

Figure1: Motion Detection Flowchart

*B. Background Creation Module*

This is the second module of the system which is responsible for the generation of a time-lapsed background.

• Purpose: The purpose of this module is to generate a time-lapsed background for the input time-period and specified duration to overlay the rearranged flow-tubes.

• Input: The input is a stream of frames in the specified time-period, and length of summary which is required to overlay on this background.

• Output: A background clip representing the background for the given duration condensed into a clip as long as the rearranged summary generated.

• Functionality: The module implements a queue type buffer, which is continuously updated and median value for each pixel row in the buffer is calculated and stored as background.

• Flowchart: The flowchart shown in the Figure 2 explains the procedure followed in the Background creation module. The module uses a buffer which stores the history of frames from previous 4 minutes or more, depending on the length of summary, and background is calculated by pixel-wise calculation of the median across the whole buffer. The buffer is updated as new frames are read into the queue data structure. The median implementation is parallelized to optimize performance.

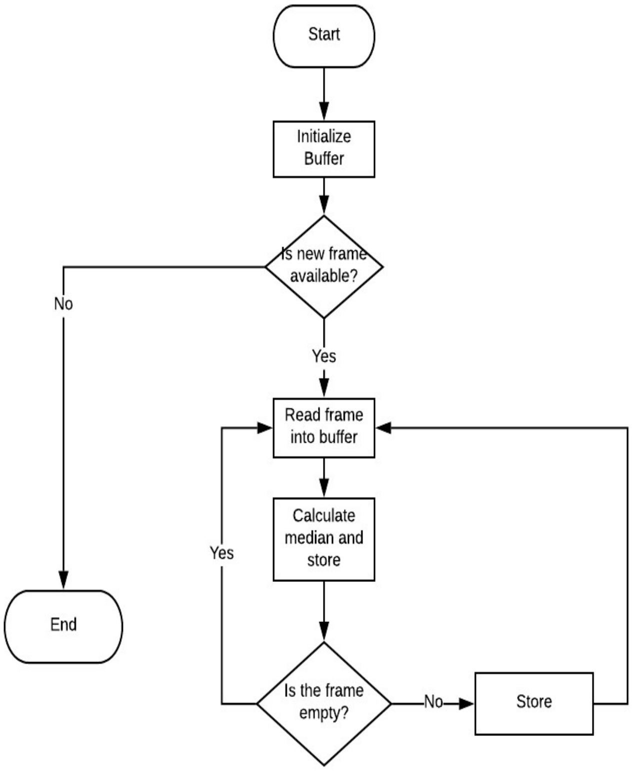


Figure2: Background Creation Flowchart

*C. Optimization Module*

This is module of the system handles the optimization and rearrangement of selected tubes.

- Purpose: The purpose of this module is to rearrange tubes and produce a compact meaningful summary.
- Input: 3D arrays of flow-tubes extracted and their original timestamps.
- Output: The module computes an optimal configuration of the flow-tubes based on a pre-defined cost function to generate the summary.
- Functionality: The module generates an optimized configuration of flow-tubes which is both condense yet meaningful in nature.
- Flowchart: The flowchart shown in the Figure 3 explains the procedure followed in the optimization module. A popular heuristic based search algorithm has been used to implement this module called Simulated Annealing. The algorithm trades-off from exploration to exploitation as the number of iterations and epochs increases. The module uses a pre-defined cost function to evaluate a fitness score for each configuration and the global optimized configuration (GOC) is updated as per the sigmoid value obtained from applying the sigmoid function on the difference in fitness value from the current and previous globally optimized configurations. Higher the sigmoid value, higher are the chances of the GOC getting updated.

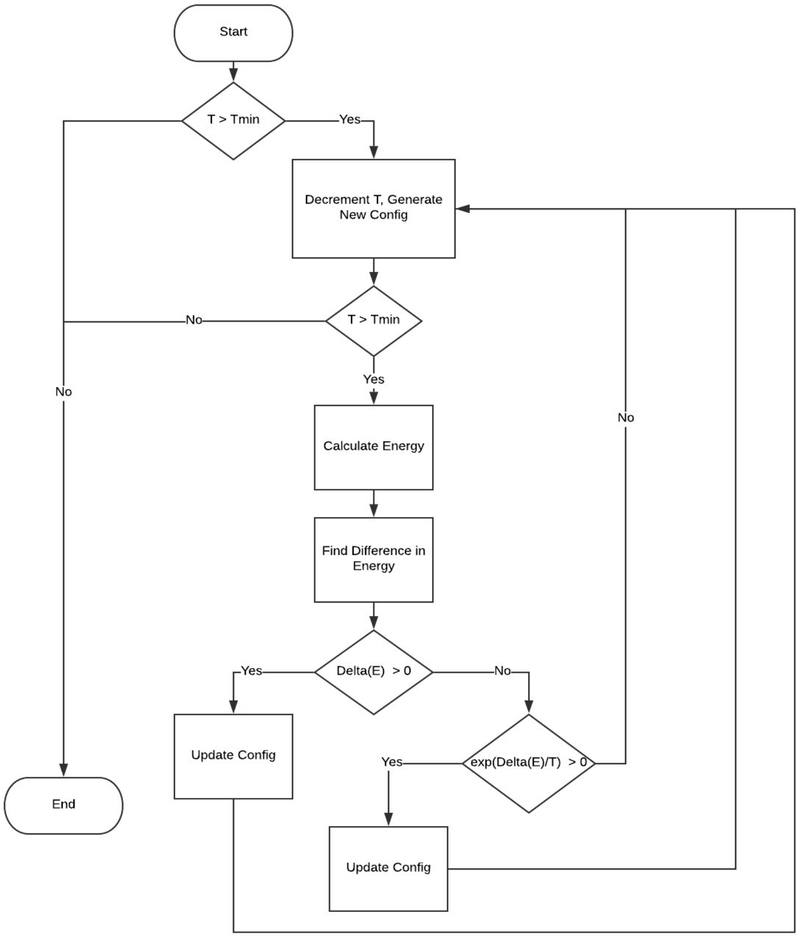


Figure3: Optimization Module Flowchart

*D. Tube Extraction Module*

This module of the system which handles the extraction of flow-tubes of several subjects present in the clips of interest identified by motion detection module.

- Purpose: The purpose of this module is to identify and extract flow-tubes from each clip of interest.
- Input: 3D arrays of MOG masks of video frames of the clips of interest.
- Output: 3D arrays of masks which represent the flow-tubes for each subject tracked in each clip of interest.
- Functionality: The module identifies flow-tubes whose length is beyond the specified user-threshold and extracts the flow-tubes as mask arrays and stores them for further processing later.
- Flowchart: The flowchart shown in the Figure 4 explains the procedure followed in the tube extraction module. The module's core component is the connected component search implementation. Each frame is processed to identify the number of individual blobs present in the frame and are correlated with the blobs seen in the previous frame to track existing subjects and create new subjects as and when they occur in each clip of interest. After having individually identified each subject by annotating the subjects as such in the input video feed, individual mask flow-tube arrays are created for each subject and then stored with their respective timestamps.

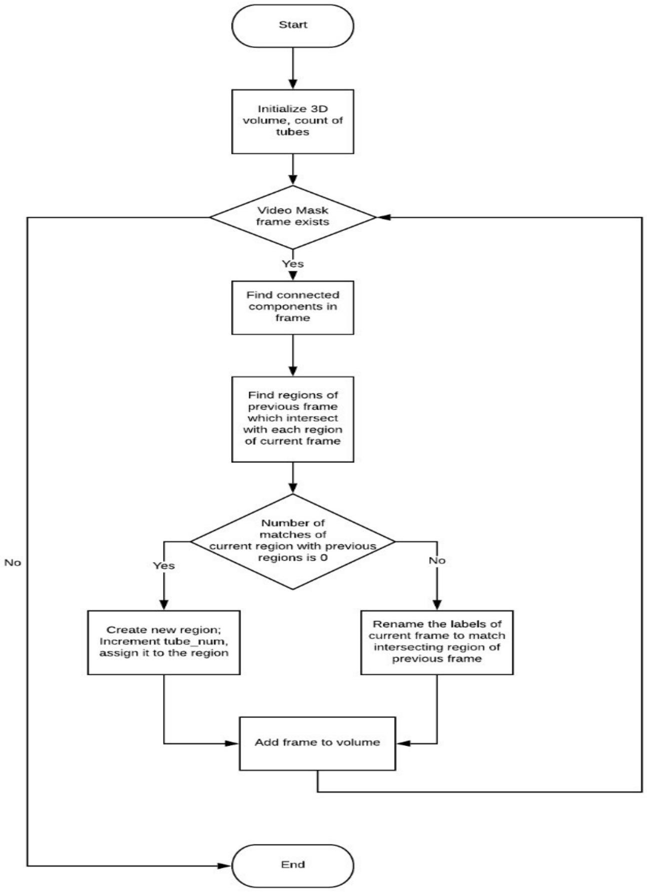

Start
Initialize 3D volume, count of tubes
Video Mask frame exists
Yes
Find connected components in frame
Find regions of previous frame which intersect with each region of current frame
Number of matches of current region with previous regions is 0
Yes
No
Create new region; Increment tube_num, assign it to the region
Rename the labels of current frame to match intersecting region of previous frame
Add frame to volume
No
End

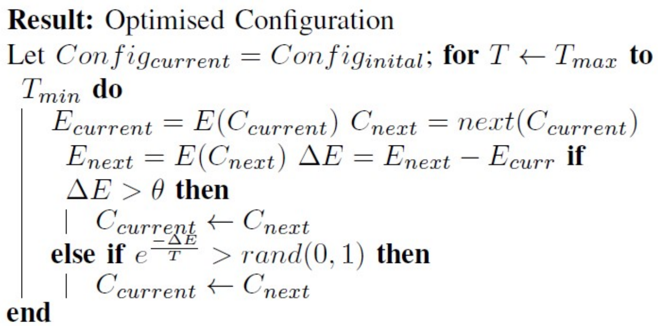
Figure4: Tube Extraction Flowchart

*Algorithms:*
Algorithm shows the simulated annealing algorithm which is used for the tube optimization process.

*Generation of initial configuration:*
The initial configuration is generated by randomly initializing the starting time of each clips within a certain range. Start time is given as:
start_time ~ U[0,max_length * 0.5]
where max_length is the sum of the lengths of all the tubes

*Generation of updated configuration*
In each iteration of the optimization process, a shift is generated for each tube.
We found that generating progressively smaller shift values, based on the ratio of current temperature to maximum temperature values was effective at reducing the collision cost.
shift ~ U[-max_length * ratio * 0.5,max_length * ratio * 0.5]
where, max_length is the sum of the lengths of all the tubes
ratio = T_curr \/T_max of temp values of simulated annealing
Suitable T_max values were found to be around 20-50 & suitable T_min values around 1-5.

Algorithm 1 [13] shows the tube extraction algorithm which is used for the extracting the event tubes from the labelled motion volume.



```
Result: Optimised Configuration
Let $Config_{current} = Config_{inital}$; for $T \leftarrow T_{max}$ to
 $T_{min}$ do
    $E_{current} = E(C_{current})$ $C_{next} = next(C_{current})$
     $E_{next} = E(C_{next})$ $\Delta E = E_{next} - E_{curr}$ if
     $\Delta E > \theta$ then
        $C_{current} \leftarrow C_{next}$
    else if $e^{\frac{-\Delta E}{T}} > rand(0,1)$ then
        $C_{current} \leftarrow C_{next}$
end
```

**Algorithm 1:** Simulated Annealing

## IV. Results

Sample video clips were collected in different urban areas with scattered motion and activity, with only 1-2 events taking place once every few seconds. These clips were used to develop and test the algorithm. Figure 5 shows some frames from our experimental videos. Only 1-2 events occur at any given time. People and bikes are the objects that are seen. Figure 6 shows the output frame generated by our video summary. Many people and bikes are seen together and the timestamp shows the time at which the event occurred in the original input video. It is clear from this picture that the density of events has improved dramatically.

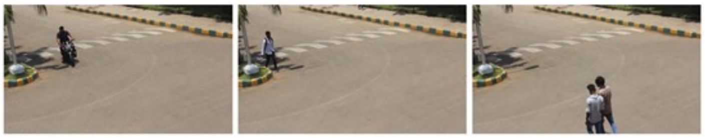


Figure5: Selection of input frames from one of our sample videos

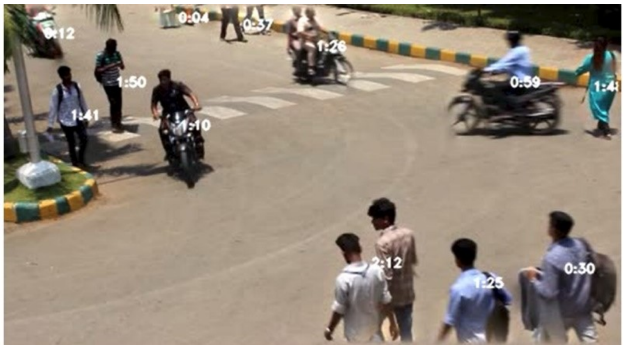


Figure6: A frame from the generated summary video

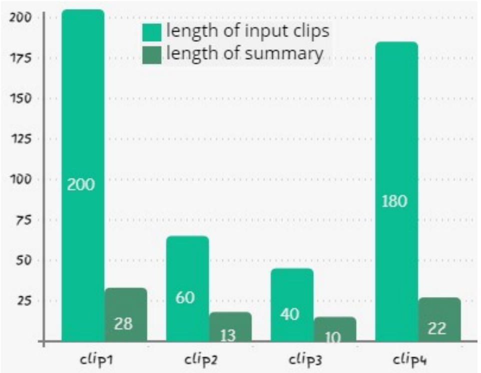


Figure7: comparison of input video vs summary video

Above graph as shown in Figure 7, depicts the variation in the lengths of Original video and summary video. Summary video apart from being considerably short in length compared to original video also shows only the highlights which the user wants to see.

*Formulas used:*

*Motion detection and Background Masking phase:*

Probability of every pixel being the foreground or background is calculated as:

$$\mathrm{P}(X_t) = \sum_{i=1}^{K} \omega_{i,t} \cdot \eta\left(X_t, \mu_{i,t}, \Sigma_{i,t}\right)$$

……….. Equation 1. Gaussian Mixture Model

$X_t$: current pixel in frame t

$K$: the number of distributions in the mixture

$\omega_{i,t}$: the weight of the k$^{th}$ distribution in frame t

$\mu_{i,t}$: the mean of the k$^{th}$ distribution in frame t

$\Sigma_{i,t}$: the standard deviation of the k$^{th}$ distribution in frame t

where $\eta\left(X_t, \mu_{i,t}, \Sigma_{i,t}\right)$is a probability density function defined in equation 2 as:

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{n/2}\Sigma^{1/2}} exp^{\frac{-1}{2}(X_t-\mu)\Sigma^{-1}(X_t-\mu)}$$

……………Equation 2. Probability Density Function

*Optimisation module:*

The heart of the project resides in the rearrangement phase. A heuristic based approach has been adopted to solve this NP combinatorial problem. In order to solve a combinatorial problem using an algorithm like simulated annealing [13], an Energy or a Cost function [5] must be defined, which embodies the various parameters to be optimised. In this case, the two main parameters to optimise are:

– **Collision**: The amount of collision between events in the rearranged set of events.

– **Length**: The total length of the generated summary must be as small as possible.

Collision Cost is calculated as:

$$Cost_{Collision} = \frac{Collision}{TotalPixels}$$

…………………..Equation 3. Collision Cost

where,

$$Collision = \sum_{i=0}^{N}\sum_{j=i}^{N}\left(\sum_{k=\max(s_i,s_j)}^{\min(e_i,e_j)} T_i[k]\,.T_j[k]\right)$$

……………Equation 4. Collision

$$\text{TotalPixels} = \sum_{i=0}^{N}\sum_{j=i}^{N}\left(\sum_{k=\max(s_i,s_j)}^{\min(end_i,end_j)} \sum\left(T_i[k == w] + \sum T_j[k == w]\right)\right) - Collision$$

……………….Equation 5. Total Pixels

$s_i$: the time at which the clip i starts

$e_i$: the time at which the clip i ends

$T_i$: the 3D array (tube) representing the event in a Boolean map format

$w$: the value of all foreground pixels in the $T_i$

Length cost is calculated as:

$$Cost_{length} = \frac{Length - Lowerlimit}{Upperlimit - Lowerlimit}$$

…………….. Equation 6. Cost of Length

where

$$Length = \max_{\forall i \in \tau}(end_i) - \min_{\forall i \in \tau}(start_i)$$

..……………. Equation 6.1

$$Lowerlimit = \max_{\forall i \in \tau}(len_i)$$

………………. Equation 6.2

$$Upperlimit = \sum_{i}^{N}(end_i)$$

………..……… Equation 6.3

$start_i$: the time at which the clip i starts

$end_i$: the time at which the clip i ends

The total cost is given as:

$$\text{TotalCost}(W, Cost) = W.Cost^T$$

……………….. Equation 7 Total Cost

where,

- W is weight vector of the form $[Weight_{Collision}, Weight_{Length}]$ assigning different priorities for the two factors
- Cost is a cost vector of the form $[Cost_{Collision}, Cost_{Length}]$

*Objective Evaluation Parameters:*

*Exhaustiveness of summary:* The summary must retain all the events in the group-based summary page or the type of events specified.

Result: All notable events in the event have been selected.

*Compression factor:* Given by the length of the original input / summary length. The compression factor should be as high as possible while keeping the overlap factor to a minimum.

Result: There is a compression factor of 4-7x for our sample videos.

*Subjective Evaluation Parameters:*

*Semantic structure of events:* Must be shown with interacting events (which take place at the same time and place).

Result: All interacting events are always shown together

*Realistic appearance:* Events must be taken out and blended seamlessly into the background and look real.

Result: Real appearance in most cases, when it is with some halo around the ha object when it is at the edge of the frame, or when it is intersected.

## V. CONCLUSION

This paper aims to use an innovative method to create video summaries to reduce the time spent analyzing CCTV video footage. Implementation in this work summarized by the temporal rearrangement of events, only improves on other methods of finding frames of motion. Tag-based summaries only select the type of events needed, such as people, cars, bikes, etc., further reducing the summary length. A Proof-F concept has been introduced, and can be used in commercial CCTV systems with further development.

## REFERENCES

[1] W.-S. Chu, Y. Song and A. Jaimes, “Video Co-Summarization: Video Summarization by Visual Co-Occurrence,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[2] Y. Li, T. Zhang and D. Tretter, “An overview of video abstraction techniques,” 2001.

[3] J. Nam and A. H. Tewfik, “Video abstract of video,” in 1999 IEEE Third Workshop on Multimedia Signal Processing (Cat. No. 99TH8451), 1999.

[4] J. H. Oh, Q. Wen, S. Hwang and J. Lee, “Video abstraction,” in Video data management and information retrieval, IGI Global, 2005, pp. 321-346.

[5] A. Rav-Acha, Y. Pritch and S. Peleg, “Making a long video short: Dynamic video synopsis,” in 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), 2006.

[6] Y. Pritch, A. Rav-Acha and S. Peleg, “Nonchronological video synopsis and indexing,” IEEE transactions on pattern analysis and machine intelligence, vol. 30, pp. 1971-1984, 2008.

[7] M. Rodriguez, “CRAM: Compact representation of actions in movies,” in 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010.

[8] Z. Zivkovic and others, “Improved adaptive Gaussian mixture model for background subtraction.,” in ICPR (2), 2004.

[9] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.

[10] Y. Pritch, A. Rav-Acha, A. Gutman and S. Peleg, “Webcam synopsis: Peeking around the world,” in 2007 IEEE 11th International Conference on Computer Vision, 2007.

[11] P. L. M. Bouttefroy, A. Bouzerdoum, S. L. Phung and A. Beghdadi, “On the analysis of background subtraction techniques using Gaussian mixture models,” in 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, 2010.

[12] J. Sun, W. Zhang, X. Tang and H.-Y. Shum, “Background cut,” in European Conference on Computer Vision, 2006.

[13] X. Yao, “A new simulated annealing algorithm,” International Journal of Computer Mathematics, vol. 56, pp. 161-168, 1995.

[14] P. Pérez, M. Gangnet and A. Blake, “Poisson Image Editing,” in ACM SIGGRAPH 2003 Papers, 2003.

[15] P. Pérez, M. Gangnet and A. Blake, “Poisson Image Editing,” in ACM SIGGRAPH 2003 Papers, 2003.

[16] R. Szeliski, M. Uyttendaele and D. Steedly, “Fast poisson blending using multi-splines,” in 2011 IEEE International Conference on Computational Photography (ICCP), 2011. G. O. Young, “Synthetic structure of industrial plastics (Book style with paper title and editor),” in *Plastics*, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.