



# Bandwidth Efficient Remote Video Surveillance System

<sup>1</sup>Megha Kolhekar, Ujjwal Trivedi, Vedant Sarode, Akshata Khardikar

\*Associate Professor, Department of Electronics and Telecommunication, FCRIT,

\*\*B.E. Students, Department of Electronics and Telecommunication, FCRIT, Vashi

Email: Vashimeghakolhekar@gmail.com, ujjwaltrivedi.1993@gmail.com

**Abstract**—This paper presents an experimental work on video surveillance system which is capable of monitoring a desired room for security purpose. The system will continuously keep track of particular location with help of camera installed and obtain database of image frames from video. Adjacent frames are compared for scene change and upon successful detection it will notify remote device such as mobile which is wirelessly connected to central system. Image compression techniques are used to utilize storage or transmission efficiency. Scene change detection is based upon MAFD (Mean Absolute Frame Difference) and Histogram equalized MAFD. Transform domain and spatial domain techniques are used for compressing the video frames.

**Index Terms**—DCT (Discrete Cosine Transform), MSE (Mean Square Error), surveillance, scene change, lossy compression, JPEG (Joint Picture Expert Group), MAFD (Mean Absolute Frame Difference).

## I. INTRODUCTION

Video surveillance systems are very important in our daily life. Video surveillance applications exist in airports, banks, offices and even in our homes to keep us secure. Video surveillance systems currently are undergoing a transition where more and more traditional analog solutions are being replaced by digital ones. Digital video surveillance system offers much better flexibility in video content processing and transmission along with implementation of advanced features like motion detection. For such surveillance system, image compression forms an integral part. If images are sent in uncompressed form then time required for transmission to end user will be more. If image compression is performed before transmission, then very high speed can be achieved and even real time view is possible.

### 1.1 IMAGE COMPRESSION

Image compression is an application of data compression that encodes the original image in such a way that compressed image requires less storage. The objective of image compression is to reduce the redundancy of the image and to store or transmit data in an efficient way. The process of image compression consists of two components, an encoding algorithm that

takes a input image and generates a compressed representation (with fewer bits), and a decoding algorithm that reconstructs the original image or some approximation of it from the compressed representation. These two components typically work in tandem, since they both have to understand the shared compressed representation.

Compression techniques can be of two types: one is lossy another is lossless. Lossless compression reduces size by identifying and eliminating statistical redundancy without any loss. In lossless compression techniques, the original image can be perfectly recovered from the compressed (encoded) image. These are also called noiseless since they do not add noise to the signal (image). It is also known as entropy coding since it uses statistics/decomposition techniques to eliminate/minimize redundancy.

Lossless compression is used only for a few applications with stringent requirements such as medical imaging.

Following techniques are included in lossless compression:[1]

1. Run length encoding
2. Huffman encoding

Lossy compression reduces size by identifying unnecessary information and removing it. It is used for compressing images, videos generally. [1] The main advantage of lossy compression method is that, it not only considers redundancy in data but also factors that involve human perception, resulting in very high compression rate. Image compression is based on the fact that human eye is less sensitive to high frequency components and hence it can be easily removed without much loss of detail in image thus achieving considerable reduction in size. Lossy is used in an abstract sense, however, and does not mean random lost pixels, but instead means loss of a quantity such as a frequency component, or perhaps loss of noise.

Image compression forms the backbone for video surveillance system where the database is so large that practical transmission without compression is very time-

and resource-consuming. Image compression is a smart technique which takes advantage of the redundancy in the data. Image compression can be achieved with the help of lossy compression technique. Lossy compression techniques includes following coding schemes

1. Transform coding
2. Vector quantization
3. Fractal coding

video surveillance system uses two approaches for image compression namely frequency transfer domain and spatial domain. Former technique consists of three important processing stages: Image transform, quantization and entropy encoding. In transformation encoding scheme, DCT (Discrete Cosine Transform) is used to change the pixels in the original image into frequency domain coefficients (called transform coefficients). These coefficients have several desirable properties. One is the energy compaction property that results in most of the energy of the original data being concentrated in only a few of the significant transform coefficients which are low frequency components. This is the basis of achieving the compression. Only those few significant coefficients are selected and the remaining high frequency components are discarded. The selected coefficients are considered for further quantization and entropy encoding. There are two important factors governing the compression of video sequence namely, Temporal redundancy and Spatial redundancy which can be exploited for scene change detection.

Before compression of particular frames scene change detection algorithm is applied on data base. Two algorithms are implemented on images. First method is based on average mean difference of frames. The first method gives many instances of false and missed detections when high motions and brightness variations are present in the video. Second method has advantage over first method. This method is based on MAFD and histogram equalized MAFD. MAFD alone does not provide robust detection, we introduce a new top-down approach consisting of two main phases: (a) rejecting a large portion of easily detected non-scene change frames using MAFD, and (b) further refining the detection process for the rest of the frames. In this second phase, frames are firstly normalized via a histogram equalization process. The refined decisions are then based on a combined metrics of MAFD and MAFD\*.

## II. PROPOSED METHOD

The proposed method consists of three major sections namely transmitter section, wireless section and receiver section as shown in fig. (1)

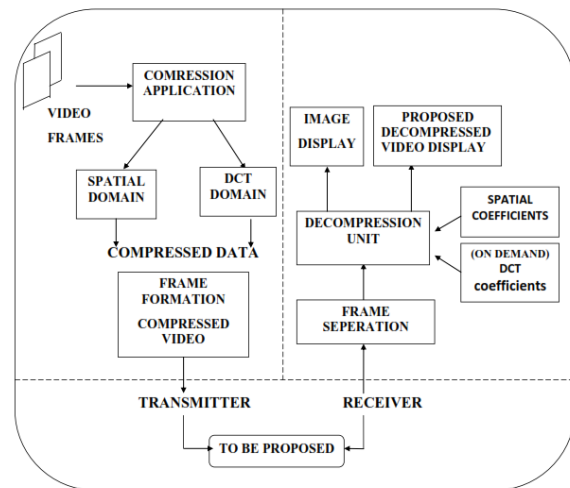


Fig. 1: Block diagram of the surveillance system

### 2.1 TRANSMITTER SECTION

This section deals with video acquisition, parsing video in the frames, scene change detection and the compression. A typical webcam of resolution 640x480 is used to capture video frames. Then scene change detection algorithm is implemented on those frames. After successful motion detection image compression algorithm is applied in DCT and Spatial domain on java platform. In either domains, size of data to be sent is significantly reduced. After detection of scene change the remote device is alarmed by sending command via wireless network and the corresponding compressed image is sent immediately.

### 2.2 RECEIVER SECTION

This section consists of remote device like mobile to display compressed images and real time video. The remote device receives compressed data from transmitter via wireless network. In addition to this the remote device can also request for required image of desired quality by sending command with proper parameters to transmitter via wireless network.

## III. IMPLEMENTATION

The following stages have been successfully implemented:

### 3.1 EXTRACTION OF REAL TIME IMAGES FROM WEBCAM TO CREATE DATABASE

For obtaining images from webcam, Java library files are imported into Eclipse IDE (Integrated Development Environment) under java platform. JavaCV is a library of programming functions which occurs mainly on real time image processing.

An object of class Frame Grabber is created and Start() method is invoked from that object to capture frame from webcam. Another object of Ipl Image is created to store captured frames. After capturing, images are stored in particular directory as list of images governed by memory availability. Image1, Image2, and

Image3,.....Image N, where N is governed by memory availability.

### 3.2 DETECTION OF SCENE CHANGE

#### 3.2.1 METHODOLOGY

##### METHOD I :

Divide the frames into square blocks of size 'b' and frame size of (f x f). Let  $B_i$  represent  $i^{th}$  block of image. Let  $F_n$  and  $F_{n+1}$  be the consecutive frames to be compared.

$B_i = m \times n$        $\bar{B}$ : average of blocks

$F_n = U_i B_i$       where  $i=1, 2, 3, \dots, f/n$

$B_i'$  is the  $i^{th}$  block of next frame.

$F_{n+1} = U_i B_i'$       where  $i=1, 2, 3, \dots$

$\Delta_j = [\bar{B}_i - \bar{B}_i'] \forall i$        $j=1, 2, 3 \dots N$

where N: no. of blocks

$\Delta_j$  represents the difference matrix obtained by taking difference of consecutive block matrices

$\Delta_g = \text{mean}(\Delta_j)$

$\Delta_g$  is the mean of all values in  $\Delta_j$

$\bar{\Delta}_{k_i} = 1$       if  $(\Delta_i > \Delta_g)$

0      otherwise

For every value of  $\Delta_i$  greater than  $\Delta_g$ ,  $\bar{\Delta}_{k_i}$  is filled with 1 else 0.

$G_{av} = \text{mean}(\bar{\Delta}_k)$

$G_{av}$  is the mean of all values present in  $\bar{\Delta}_k$ .

If  $G_{av} > 0.5$  then scene change is detected.

##### METHOD II :[2]

The method described in 3.2.1 gives many instances of false and missed detection. The method described below is more effective. Let  $F_n$  and  $F_{n-1}$  be the consecutive frames to be compared. A simple mean absolute frame differences (MAFD) for the  $n^{th}$  frame is obtained as follows:

$$MAFD_n = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |f_n(i, j) - f_{n-1}(i, j)| \dots (1)$$

Where M and N are the width and height of the frames,  $f_n(i, j)$  is the pixel intensity at position (i,j). MAFD corresponds to the first-order derivative  $off_n$ , and it measures the degree of dissimilarity at every frame transition. We focus on abrupt scene change detection issues by introducing a two-phase strategy. Frames are firstly tested against MAFD, based on the fact of that most of the frames are not scene changes and a first decision can be made quickly to screen out many non-scene change frames. It is observed that MAFD alone does not provide robust detection. We introduce a new top-down approach consisting of two main phases:

(a) Rejecting a large portion of easily detected non-scene change frames using MAFD

(b) Further refining the detection process for the rest of the frames. In this second phase, frames are firstly normalized via a histogram equalization process. The refined decisions are then based on a combined metrics of MAFD and MAFD\*. Here \* represents Histogram equalized image.

Scene change decision is made on a top-down fashion. That is, most surely decisions are made in the beginning via relaxed thresholds, followed by more refined decisions via more rigid thresholds.

The algorithm is implemented as follows:

/\* First phase \*/

Compute MAFD;

if (MAFD < Threshold\_value\_1) THEN

Return false; Reject via threshold of MAFD

End if

/\* Second phase \*/

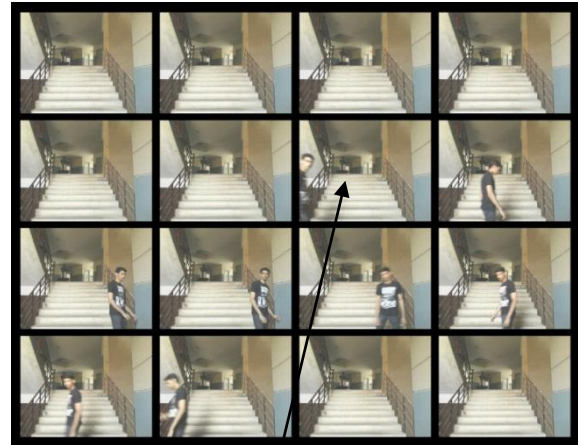
Compute MAFD\*;

if (MAFD\* < Threshold\_value\_2) THEN

Return false; Reject via threshold of MAFD\*

Endif

Result of Method II:



Occurrence of Scene change

### 3.3 COMPRESSION OF IMAGES

Compression is performed either in spatial or frequency(DCT) domain. Color images so obtained from previous stage after scene change detection are forwarded for compression. But before compression RGB values of pixels are extracted and stored in corresponding arrays for further processing. Hence three functions for separating RGB components from single pixel are created. Original pixel is of 32 bit. First 8 bits are not taken into consideration, it represents transparency and only next 24 bits are considered. Out

of those 24 bits, 8 bits each define red, green and blue colors. Following is the logic for separating red, blue and green pixels.

Now shifting 16 bit out of 24 bits and then ANDing with FF (11111111), to get red pixels.

red = (pixel\_val >> 16) && 0xFF;

Now shifting 8 bit out of 24 bits and then ANDing with FF (11111111), to get green pixels.

green = (pixel\_val >> 8) && 0xFF;

Now last 8 bits of original pixel ANDing with FF (11111111), to get blue pixels.

blue = pixel\_val && 0xFF;

After obtaining RGB components of all pixels, each component is stored in different array.

### 3.3.1 DCT IMPLEMENTATION

A discrete cosine transform[3] (DCT) expresses a sequence of finitely many data points in terms of a sum of cosine functions oscillating at different frequencies.

Using DCT-2D, pixel values of a spatial image in the region will be transformed into a set of DCT coefficients in the frequency region. Before compression, pixel values in arrays are divided into several blocks. Each block consists of 8x8 pixels. Compression operations including DCT-2D in it will be done on each block.

#### DCT EQUATION

The DCT equation (Eq.1) computes the  $i, j^{\text{th}}$  entry of the DCT of an image.

$$D(i, j) = \frac{1}{\sqrt{2N}} C(i) C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} p(x, y) \cos \left[ \frac{(2x+1)i\pi}{2N} \right] \cos \left[ \frac{(2y+1)j\pi}{2N} \right] \quad \dots(2)$$

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } u = 0 \\ 1 & \text{if } u > 0 \end{cases} \quad \dots(3)$$

$p(x, y)$  is the  $x, y^{\text{th}}$  element of the image represented by the matrix  $p$ .  $N$  is the size of each block on which DCT has to be performed.

The equation calculates one entry  $(i, j^{\text{th}})$  of the transformed image from the pixel values of the original image matrix. For the standard 8x8 block.  $N$  equals 8 and  $x$  and  $y$  ranges from 0 to 7. Therefore  $D(i, j)$  is as follows:

$$D(i, j) = \frac{1}{4} C(i) C(j) \sum_{x=0}^7 \sum_{y=0}^7 p(x, y) \cos \left[ \frac{(2x+1)i\pi}{16} \right] \cos \left[ \frac{(2y+1)j\pi}{16} \right] \quad \dots(4)$$

Because adjacent image pixels are highly correlated, the forward DCT processing step lays the foundation for

achieving data compression by concentrating most of the signal in the lower spatial frequencies. For a typical 8x8 sample block from a typical source image, most of the spatial frequencies have zero or near-zero amplitude and need not be encoded. In principle, the DCT introduces no loss to the source image samples; it merely transforms them to a domain in which they can be more efficiently encoded.

Consider original 8x8 block of red pixel:

|     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 120 | 119 | 123 | 116 | 102 | 98  | 97  | 92  |
| 111 | 113 | 114 | 109 | 104 | 102 | 101 | 104 |
| 108 | 114 | 109 | 105 | 107 | 107 | 108 | 104 |
| 112 | 105 | 105 | 107 | 104 | 100 | 98  | 104 |
| 125 | 114 | 113 | 110 | 99  | 99  | 103 | 100 |
| 116 | 112 | 106 | 102 | 99  | 98  | 100 | 116 |
| 107 | 100 | 95  | 94  | 95  | 96  | 109 | 135 |
| 98  | 90  | 94  | 96  | 97  | 120 | 147 | 155 |

Table No. (1)

After Applying DCT

|        |       |        |       |       |      |       |       |
|--------|-------|--------|-------|-------|------|-------|-------|
| 924    | -103  | 35.9   | -3.5  | 14.5  | -4.8 | 10.2  | -5.6  |
| -108   | 114   | -2.5   | -1.0  | 1.20  | -0.5 | -1.2  | 1.92  |
| 25.76  | -10.1 | -25.50 | 6.771 | 1.88  | 4.60 | -1.65 | 1.36  |
| -13.45 | -9.23 | 26.36  | 12.54 | -9.73 | 5.32 | 0.47  | -0.20 |
| -1     | 13.3  | -4.0   | -4.4  | 6     | 7.17 | 1.39  | -1.3  |
| -10.9  | 0.26  | -0.971 | 13.88 | -10.2 | -1.1 | -2.5  | 0.84  |
| 3.40   | 6.72  | -3.6   | -7.4  | 1.92  | 1.50 | 1.00  | -1.1  |
| 2.04   | -4.5  | 6.75   | -7.0  | 2.86  | 5.41 | -3.6  | -0.6  |

Table No. (2)

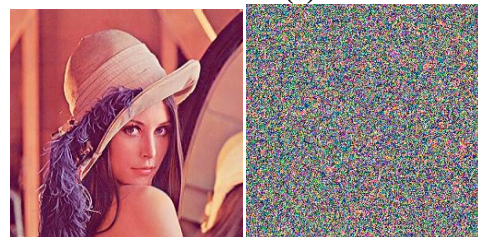


Fig.(1)Lena image before DCT Fig.(2)Lena image after DCT

### 3.3.2 APPLYING QUANTIZATION

8x8 block of DCT coefficient is quantized by quantization matrix. In this step varying levels of image compression and quality are obtainable through selection of specific quantization matrices. This helps the user to decide on quality of images ranging from 1 to 100, where 1 gives the highest compression but poorest

35

Matrix after applying IDCT

|     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 163 | 146 | 129 | 119 | 107 | 97  | 98  | 105 |
| 151 | 138 | 126 | 120 | 110 | 99  | 96  | 102 |
| 135 | 126 | 120 | 119 | 112 | 101 | 98  | 102 |
| 124 | 115 | 110 | 112 | 110 | 104 | 105 | 112 |
| 118 | 107 | 101 | 104 | 106 | 108 | 117 | 130 |
| 112 | 102 | 97  | 102 | 109 | 116 | 130 | 145 |
| 104 | 97  | 98  | 109 | 118 | 126 | 139 | 153 |
| 97  | 94  | 100 | 116 | 127 | 133 | 143 | 155 |

Table No. (7)

Mean squared error

Comparing restoration results requires a measure of Image quality. The mean-squared error (MSE) between two images  $p(x,y)$  and  $p'(x,y)$ ,

Formula for calculating MSE

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [p(i,j) - p'(i,j)]^2 \dots (8)$$

RECONSTRUCTION OF IMAGE AFTER IDCT



Fig.(3)

Fig. (4)

Fig. (5)

Fig.(3) represents lena.jpg image without applying quantization having size of 166kb with MSE is equal to 0 .Fig.(4) represents lena.jpg image with applying quantization of quality level 50 having size of 9.91kb with MSE is equal to 360. Fig.(5) represents lena.jpg image with applying quantization of quality level 90 having size of 10.6kb with MSE is equal to 226.

### 3.5 SPATIAL DOMAIN APPROACH

In this compression technique original pixels of image are stored in matrix but only alternate pixels are used in reconstruction step. Alternate pixel of row and column are sent for reconstruction. Hence size and dimension of image reduces. If original image having dimension of 256x256 then reconstructed image using this method having dimension of 128\*128. Thus image size is reduced by four times. This technique has advantage over frequency domain that it is fast and less complicated.

Original Lena imageLena image after processing



Fig.(6)Lena.jpg(166kb)-256x256Fig.(7)Lena.jpg (5.25kb)-128x128

### COMPRESSION RATIO

Compression ratio is calculated as follows:

1. For DCT domain

Compression ratio

$$= \frac{\text{Total no. of DCT coefficients sent}}{64 \times \text{no. of DCT blocks}} \dots (8)$$

2. For Spatial domain

Resolution factor itself is the compression ratio. Following are possible resolution factors : $(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots)$

### VII. CONCLUSION & FUTURE WORK

In this paper, we propose a bandwidth efficient lossy compression based video surveillance system using scene change detection. The idea is to save on transmission bandwidth by selecting only those frames for transmission which show some change in steady scene. Compression is implemented in spatial as well as transform domain and only one of them selected practically. We also plan to send this compressed scene on a mobile device along with an SMS alert. On demand, an uncompressed video clip could be made available by storing some frames before and after scene change. The testing will be done and false alarm rate, clarity of pictures will be quantified.

### IV. REFERENCES

- [1] Wei-Yi Wei, "An introduction to image compression", Graduate Institute of Communication Engineering, National Taiwan University, Tapei, Taiwan, ROC
- [2] Xiaoquan Yi and Nam Ling, "Fast Pixel-Based Video Scene Change Detection", Department of Computer Engineering, Santa Clara University, Santa Clara, California 95053, USA
- [3] R. Gonzales, R. Woods, Digital Image Processing, Pearson Education.

