



# **Secured Transmission of Video by Advantageous KLT Algorithm**

Kapil, Anurag Singh

B.Tech Scholar, Department of ECE, Om Institute of Technology and Management, Guru Jambheshwar University of  
Science and Technology, Hisar, Haryana, India

Assistant Professor, Department of ECE, Om Institute of Technology and Management, Guru Jambheshwar University  
of Science and Technology, Hisar, Haryana, India

**ABSTRACT:** In this paper the frames are acquired from the camera signals and those are compared with CAM shift and KLT algorithm, this paper focuses on CAM shift limitation that observed while applying the algorithm, such as tilting of face, skipping tracking, these limitations were removed successfully using KLT algorithm the results were saved and after that digital watermarking was applied on these results for security purpose.

**KEYWORDS:** CAM Shift, KLT algorithm, Watermarking, PSNR, Face Detection

## **I. INTRODUCTION**

Today technology in this world is at different level, the time we think or perform a task, an invention took place in the digital world. The advancement of technology is the need of the world in every field of area. In the era of 21st century the field of imaging is also playing an important role in every humans life , from mobile camera to the video conferencing , from anti-theft system to the character recognition systems are reached to an next level of advancement, one of the most common imaging element is the camera used to produce video signals corresponding to the frames received and processed by its internal circuitry, the output of an camera signal mostly depends on the resolution of lens used , due to the rapid expansion of internet in recent years there is a rapid increase in the content of digital data such as audio ,video, images , a country put lot of resources on security and surveillance , so the techniques used to achieve such important task should be efficient and cost effective every signal might be not important to us , but while transmission of desired video content our most of the signal are important , everyone need to transmit its data in a systematic and secure manner, so that the fidelity of the system increases , while using Matlab different types of processing techniques can be performed on camera signals from basic cropping to the features extraction from video,(as the digital data is increasing day by day there is an mandatory need to handle it an optimum manner), so there is an major requirement in the field of electronics and communication that these sequence of frames transmits securely , such as to prevent from unauthorized access, [5]the false access can be prevented by variety of techniques such as from simply assigning a key to the data or encoding the data at the transmission side and then decoding it at the receiver side. To the digital water marking techniques, or cryptography (such as DES, RSA, AES etc.). In the recent years the Uav's are grabbing the attention, these are sometimes termed as quad-copters [2] or by any other name is a type of flying gadget consisting of a camera for capturing images. The sensed camera output are send over the air link as in this case mostly the camera unit and the receiving unit or the server or the processing station is far from the transmitting section, due to the disadvantage of ground cameras that are restricted to the limited field of view, unlike in Uav's that are used for monitoring , surveillance. The purpose of this paper is to grab the attention towards different techniques used for the process of video signal processing and then transmitting in a secure manner, the paper also provides information regarding the technique that is best suited for specific application or the tasks. The techniques are analysed by the Matlab which act as an integrated development environment for the effective analysis of various techniques

# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

## II. PROPOSED ALGORITHM

A. Different techniques are being used for processing of camera signals figure 1 shows the proposed architecture. The first one which we are discussing is the “Face detection and tracking using cam shift” in this the following steps are to be followed, the main condition required for this technique is the presence of suitable camera signal consisting of moving frames of mostly the above neck portion or the facial part of the body and presence of a single object. [9] When this mandatory condition is satisfied then following steps are performed. [11]

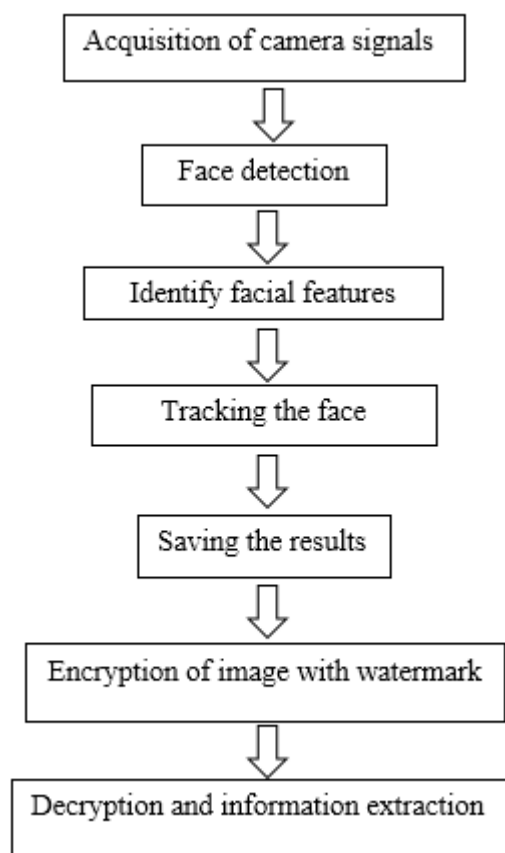


Fig: 1 proposed system architecture

### 1- Detect the face to track.

Before tracking a face, we first need to detect the face from the video provided by the camera. For this we use vision.CascadeObjectDetector, this detect the location of the face in a video frame. The cascade object detector has a Viola Jones detection algorithm and a trained classification model for detection. Initially the detector is calibrated to detect faces but it can be configured for other object types the cascade object detector is being used to track a face across successive video frames.

### 2- Identify facial features to track

Once the face is located in the video, the next step is to identify a feature that will helps us to track the face. For example, we can use the shape, texture, or colour. Choose a feature that is unique to the object and remains invariant even when the object moves. In the camera signal used, the skin tone is used as the feature to track. The skin tone provides a good deal of contrast between the face and the background and does not change as the face rotates or moves. [8]

# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

### 3- Track the face

We select the skin tone as the feature to track, we use the *vision.HistogramBasedTracker* for tracking. The histogram based tracker used here is hold by the CAMShift algorithm, [7] is used to provide the capability to track an object using the HSV values, the outputs are shown in fig 2. From the provided video, the Hue channel pixels are extracted from the nose and the nearby region of the detected face. The extracted pixels information is used to initialize the histogram for the tracker. The steps of the detection of a mode can be summarized as in equation 1:

$$I = f(x, y) \quad (1)$$

Skin colour likelihood image ( $P_{\text{skin}}$ ) obtained by backprojection of a histogram in the colour space HSV (for example) built off-line from a base of examples of skin colour pixels. Initialization of the window  $W$  of the shift mean: position and size. As  $W$  is moved more than a certain threshold and the maximum number of iterations has not been reached: Calculate the centroid of the pixels  $(x, y)$  included in the window, and the sum of (time zero order):

$$M_{00} = \sum_{x,y} I(x, y) \quad (2)$$

$$M_{01} = \sum_{x,y} x I(x, y) \quad (3)$$

$$M_{02} = \sum_{x,y} y I(x, y) \quad (4)$$

From equation (2), (3), (4) we conclude the below equation

$$x_c = \frac{M_{01}}{M_{00}} \quad (5)$$

Centre Won the centroid  $(x_c, y_c)$ .



Fig: 2 (a) Detected face with no error (b) Detected face with tilt error(c) error while moving backward (d) error while tilting at some distance.

### Limitation

During experiments it was observed that when the face tilts or the person turns their head in the video used for processing, we may lose tracking. This limitation is due to the type of trained classification model used for detection. Performing face detection for every video frame is computationally expensive in case of CAM Shift also there need of enough contrast between the face ROI and the background region.

B. The second technique we are using is the face detection and tracking using KLT algorithm, "Face Detection and Tracking Using the KLT Algorithm". Object detection and tracking are important in many computer vision applications including activity recognition, automotive safety, and surveillance. In this example, we will develop a simple face tracking system by dividing the tracking process into three parts:

### 1- Capture the face

To track the face over time, we use the Kanade-Lucas-Tomasi (KLT) algorithm. While it is possible to use the cascade object detector on every frame, it is somewhat expensive. It also sometimes fail to detect the face, when the subject turns or tilts his head. This limitation comes from the type of trained classification model used for detection. It detects the face only once, and then the KLT algorithm tracks the face across the video frames. A problem called translational image registration problem may arise it can be characterized as two function and, the two values  $F_x$  and  $G_x$  representing values at each location, where is a vector, in two images, respectively, we wish to find the disparity vector that minimizes some measure of the difference between  $F_{x+h}$  and  $G_x$  for  $x$  in some region of interest of  $R$ .

# International Journal of Innovative Research in Computer and Communication Engineering

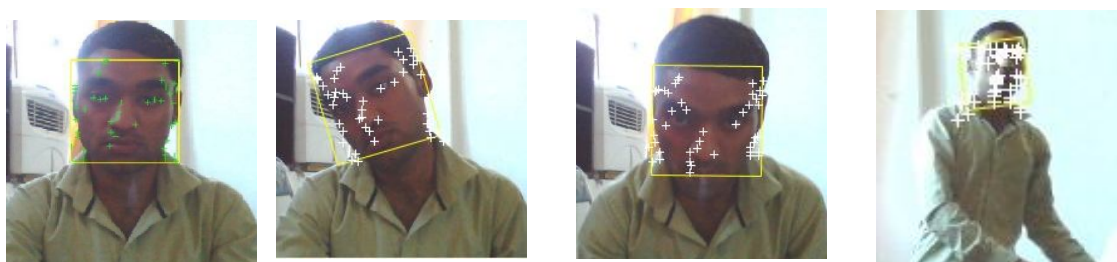
(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

2- *Extract the facial features need to be processed* KLT algorithm tracks a variety of feature points across the video frames. As the detection locates the face, the next step is to identify the feature points that can be reliably tracked.

3- *Initialize the tracker*

Further the tracker is need to be in initialized for identifying the feature points , for this we use the *vision.PointTracker* System object to track them. For each point in the previous frame, the point tracker will now attempts to find the corresponding point in the current frame. Then the estimate Geometric Transform function is used to estimate the translation, rotation, and scale between the previous points and the new points. This transformation is applied to the bounding box around the face.



(a) (b) (c) (d)

Fig: 3 (a), (b), (c), and (d), shows the efficient outputs by using KLT algorithm

4- *Tracking the Face*

For tracking the face points from frame to frame, we use Estimate Geometric Transform function to estimate the motion of the face and Make a copy of the points to be used for computing the geometric and then Transformation between the points in the old frames and the current frames occurs [1]. The output is shown in fig 3.

### C. Digital Watermarking

As from the two algorithms discussed we know that KLT is somewhat better so its outputs are saved and, is digitally watermarked [3] for secure propagation of signals the following process is followed for watermarking the camera signals [4]

1. *Reading the host image:-*

The host image is read by the Matlab by using syntax `Host_image=imread('image_name.extension')`. Where host image can be any color image for the experiment we have used the detected face as a host image of resolution 512x512.

2. *Reading the watermark image:-*

The watermark image can be read or made to read to the Matlab by using syntax `watermark_image=imread('image_name.extension')` Where watermark image can be any color image for the experiment we have used the detected face as a host image of resolution 512x512.

3. *Applying dwt2 technique*

The `dwt2` command performs a single-level two-dimensional wavelet decomposition with respect to either a particular wavelet ('wname') or particular wavelet decomposition filters (Lo\_D and Hi\_D) we specify. [10]

4. *Performing single value decomposition*

The `svd` command computes the matrix singular value decomposition. `[6]s = svd(X)` returns a vector of singular values. `[U,S,V] = svd(X)` produces a diagonal matrix S of the same dimension as X, with nonnegative diagonal elements in decreasing order, and unitary matrices U and V so that  $X = U*S*V'$ .

5. *Concatenating the results*

`C = cat(dim, A, B)` concatenates the arrays A and B along array the dimension specified by dim. The dim argument must be a real, positive, integer value.

6. *Saving the results*

For saving the results `imwrite` command is used in the `matlabimwrite(A,image_name)` where A is the file in which the image is initially stored , `image_name.extension` is provided to give the name to the output image need to be permanently stored. The resultant image is stored in data type `uint8`, if `uint8` is used to store the image then `imwrite` outputs 8-bit values. [11]

# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

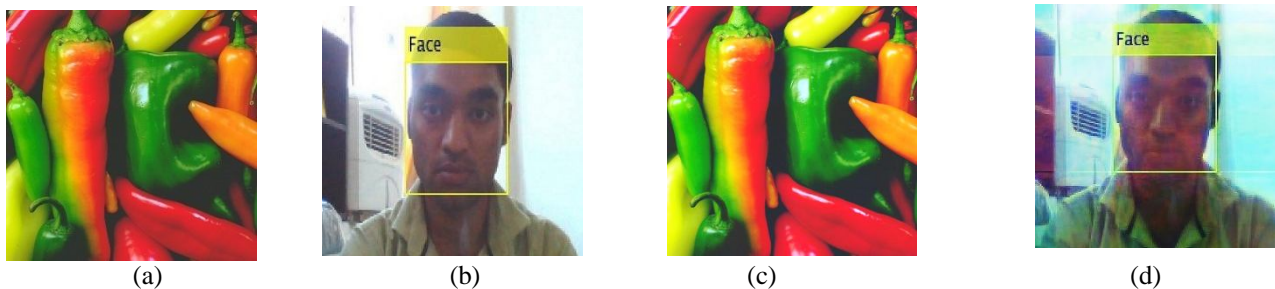


Fig: 4 a) shows the host image, b) shows the image to be watermarked, c) shows the embedded watermarked image that is to be transmitted, d) shows the extracted watermarked image

Note that the dimension of the host image and the image to be watermarked must be same in our case the dimension of both the images is 512x512 also after embedding, the size of the host image is reduced from 60.8kb to 30.3kb, we can observe these slight change from fig 4 (a) and (c) due to compression techniques applied, there is slight change in quality which is not be visible with naked eyes. During transmission of signals fig (c) would be transmitted, at the destination image in fig (a) would be required to extract image in fig (d) this will protect the unauthorized access to data.

### Limitation

Good results are not produced by all types of images used for watermark.

TABLE I Comparison of Different Parameters

Image name (Fig 4)	Level of judgment	Output value
Image (b) input	PSNR by Salt and pepper	21.5017
	SNR value	18.5302
	PSNR by Poisson noise	27.0776
	SNR Value	24.1060
	Correlation coefficient	.9979
	Mean SSIM value	.9482
Image (a) host image	PSNR by Salt and pepper	21.563
	SNR value	14.5614
	PSNR by Poisson noise	28.8588
	SNR value	21.8589
	Correlation coefficient	.9981
	Mean SSIM value	.9875
Image (c) watermarked image	PSNR by Salt and pepper	21.7292





# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

	SNR value	15.8058
	PSNR by Poisson noise	28.2846
	SNR value	22.3612
	Correlation coefficient	.9987
	Mean SSIM value	.9895
Image (d) output image	PSNR by Salt and pepper	21.9617
	SNR value	18.1032
	PSNR by Poisson noise	26.8085
	SNR value	22.9500
	Correlation coefficient	.9901
	Mean SSIM value	.9508

As shown in the table 1 the values of PSNR change slightly from the standard image shown in fig 4 (b) to the PSNR values of output extracted image shown in fig 4 (d) this shows that our frame is sent to destination with minimum error of probability, the correlation coefficient produce values between the image and the same image processed with a median filter in our case the value calculated also seems to be quite similar , the SSIM value is close to 1 in both the cases and signifies that the image quality of extracted image is good.

### III. CONCLUSIONS AND FUTURE SCOPE

In this paper the two techniques cam shift and KLT algorithm are analyzed and as per the visible results from the figures 1 and 2 we conclude that KLT algorithm is more accurate and advantageous than cam shift technique, also we found that the DWT2 and SVD technique used for watermarking can be easily applied to the outputs of KLT algorithm and from the calculated parameters mentioned in the in table 1 represents that with minimum or negligible loss effective outputs can be produced for secure transmission of camera signals. The digital watermarking had played an important role in enhancing the security of the system as without the proper key the useful information cannot be extracted.

Future work can be directed towards following issues:

- Tracking multiple objects at a time
- Improve extraction technique for better outputs
- Missing forehead error sometimes can be removed

### ACKNOWLEDGMENT

I want to acknowledge Mr.Krishan Mohan Pandey trainer futuronixpvt. Ltd for his help and guidance, they provide pathway to my interest in image processing.

### REFERENCES

- [1] CaiXuejun, *A New, Self-Adaptive, KLT-based Algorithm for Visual Tracking*, International Journal of Signal Processing, Image Processing and Pattern Recognition Vol. 8, No. 1 (2015), pp. 61-68
- [2] Daniel Wischounig-Struel, Markus Quartisch and Bernhard Rinner, *Prioritized Data Transmission in Airborne Camera Networks for Wide Area Surveillance and Image Mosaicking*, In: IEEE CVPR Workshops, pp.17-24(2011)
- [3] Poonam N. Mahamuni, Prof. R.T. Patil, and Mr. S. P. Adure, *image watermarking for images captured by Digital camera*, International Journal of Advanced Engineering Technology E-ISSN 0976-3945



ISSN(Online): 2320-9801  
ISSN (Print): 2320-9798

# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

- [4] Shamim Ahmed Laskar1 and Kattamanchi Hemachandran, *secure data transmission using Steganography and encryption Technique*, International Journal on Cryptography and Information Security (IJCIS), Vol.2, No.3, September 2012
- [5] S.Manikandaprabu, *An Efficient Watermarking Algorithm Based on DWT and FFT Approach*, S.Manikandaprabu et.al / International Journal on Computer Science and Engineering (IJCSE) ISSN : 0975-3397 Vol. 6 No.06 Jun 2014
- [6] Hongqin Shi, *DWT and SVD based Watermarking Scheme with Circulation*, journal of software, vol. 9, no. 3, march 2014
- [7] Snekha, Chetna Sachdeva, and Rajesh Birok, *Real Time Object Tracking Using Different Mean Shift Techniques*, International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-3, Issue-3, July 2013
- [8] J.Savitha and Dr.A.V.Senthilkumar, *Face Tracking and Detection using S-PCA & KLT Method*, International Journal of Advance Research in Computer Science and Management Studies, ISSN: 2321-7782 Volume 2, Issue 2, February 2014
- [9] Yang Li, Hui Ding, Shudong Zhang, and Yilei Wang, *The Research and Implementation of Camshift Algorithm Based on Automatically Target Extraction*, Journal of Convergence Information Technology (JCIT) Volume8, Number6, Mar 2013
- [10] <http://in.mathworks.com/help/wavelet/ref/dwt2.html>
- [11] <http://in.mathworks.com/help/matlab/ref/imwrite.html>