



AENSI Journals

Advances in Natural and Applied Sciences

ISSN:1995-0772 EISSN: 1998-1090

Journal home page: www.aensiweb.com/ANAS



Authentication of Video Streams using Interpolated Digital Watermarking System

¹B. Shanmugham, ²Dr. A. Asokan, ³Dr. K.P. Ramakrishnan, ⁴Dr. D. Sivakumar

¹Research Scholar, Department of Electronics and Instrumentation Engineering, Annamalai University, Chidambaram – 608002, Tamil Nadu, India.

²Assistant Professor, Department of Electronics and Instrumentation Engineering, Annamalai University, Chidambaram – 608002, Tamil Nadu, India.

³Professor, Department of Electronics and Communication Engineering, Rajiv Gandhi College of Engineering and Technology affiliated to Pondicherry University, Pondicherry – 607402, Pondicherry, India.

⁴Professor, Department of Electronics and Instrumentation Engineering, Annamalai University, Chidambaram – 608002, Tamil Nadu, India.

ARTICLE INFO

Article history:

Received 3 September 2014

Received in revised form 30 October 2014

2014

Accepted 4 November 2014

Keywords:

Video Authentication, Digital Watermarking, video splitting, matrix concatenation, visible watermarking, sub matrices, bilinear interpolation, PSNR.

ABSTRACT

Authentication techniques are needed in order to maintain authenticity, integrity, and security of digital video content. This paper presents a software implementation of an interpolated digital watermarking system that can insert colored watermark information into any part of the splitted video to provide authentication. To achieve high PSNR value, the proposed system employs two main systems namely, Video splitting system and Video merging system. The video is splitted into eight parts using video splitting system and all the parts of the videos along with the watermarked video is merged using Video merging system. The watermark is embedded using bilinear Interpolation Technique. Experimental results show that this video authentication system using interpolated watermarking features minimum video quality degradation and provides PSNR around 26 dB.

© 2014 AENSI Publisher All rights reserved.

To Cite This Article: B. Shanmugham, Dr. A. Asokan, Dr. K.P. Ramakrishnan, Dr. D. Sivakumar., Authentication of Video Streams using Interpolated Digital Watermarking System, *Aust. J. Basic & Appl. Sci.*, 8(22): 1-9, 2014

INTRODUCTION

The rapid growth of techniques for powerful digital signal and multimedia processing together with the advances in electronics and information technology, have made the distribution of video data much easier and faster (Sonoy Deb Roy *et al.*, 2013, Gwenael.A.D and Dugelay.J.L, 2003, Piva.A *et al.*, 2002). However, concerns regarding authentication of the digital video are growing, since digital video sequences are vulnerable to manipulations and alterations using widely available editing tools. This issue turns to be more significant when the video sequence is to be used as evidence for criminal actions. In such situations, the video data should be trustworthy. Consequently, authentication techniques are needed in order to maintain authenticity, integrity, and security of digital video content. As a result, digital watermarking (WM), a data hiding technique has been considered as one of the key authentication methods (Shoshan.Y *et al.*, 2008, Sarawathi.M, 2011). Digital watermarking is the process of embedding an additional, identifying information within a host multimedia object, such as text, audio, image, or video. By adding a transparent watermark to the multimedia content, it is possible to detect hostile alterations, as well as to verify the integrity and the ownership of the digital media. Today, digital video WM techniques are widely used in various video applications (X. Li, Y. Shoshan, A. Fish, G. A. Jullien, and O. Yadid-Pecht, 2008). For video authentication, WM can ensure that the original content has not been altered. WM is used in fingerprinting to track back a malicious user and also in a copy control system with WM capability to prevent unauthorized copying.

Because of its commercial potential applications, current digital WM techniques have focused on multimedia data and in particular on video contents. Over the past few years, researchers have investigated the embedding process of visible or invisible digital watermarks into raw digital video, uncompressed digital video both on software, and hardware platforms. Contrary to still image WM techniques, new problems and new challenges have emerged in video WM applications (X. Li, Y. Shoshan, A. Fish, G. A. Jullien, and O. Yadid-Pecht, 2008).

The main objective of the paper is to describe efficient software based digital video watermarking system that can insert visible water mark information in the splitted video with minimum video quality degradation

Corresponding Author: B. Shanmugham, Research Scholar, Department of Electronics and Instrumentation Engineering, Annamalai University, Chidambaram – 608002, Tamil Nadu, India.
E-mail: shanbk132@gmail.com

which is achieved by Video Splitting system and video merging system which also provides good Peak to Signal Noise Ratio.

Scheme of Implementation:

Structure of the Video:

Video is the technology of electronically capturing, recording, processing, storing, transmitting, and reconstructing a sequence of still images representing scenes in motion. Video Splitting is the process of dividing the video into non overlapping parts (Naveen B, Dr.K.R.Nataraj,Dr.K.R.Rekha, 2009). Then row mean and column mean of each part is obtained. By using splitting, higher precision is achieved for display in different Screen (Cyril Prassana Raj P, Dr. S.L. Pinjare, and Swamy.T.N, 2012). An image is defined as a two dimensional function, $f(x, y)$, where x and y are spatial coordinates, and the amplitude off at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y and the intensity values of f are all finite, discrete quantities, we call the image a digital image (Rafael C. Gonzalez and Richard E. Woods, 2008).

Digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are called picture elements or image elements or pixels. Pixel is a term used most widely to denote the elements of a digital image. Pixels are normally arranged in a two dimensional grid and are often represented using dots or squares. Number of pixels in an image can be called as resolution. Associated with each pixel is a number known as Digital Number or Brightness Value that depicts the average radiance of a relatively small area within a scene. The matrix structure of the digital image is shown in Fig 1.

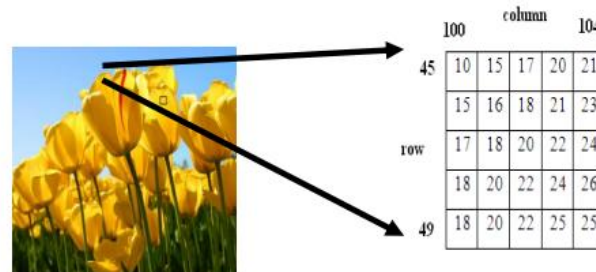


Fig. 1: Structure of Digital Image/Video.

Video Splitting System:

Video splitting is not an easier task when compared to an image splitting. In order to insert a visible watermark in the specific area of the host video with minimum video quality degradation and good PSNR, we introduce a system called Video Splitting System.

The block diagram of video splitting system is shown in Fig.2. Video from the Multimedia source is applied to the video input block. The Resize block enlarge or shrinks image size. Captured video will be in RGB format. It is converted into chroma and luma components. Luma represents the brightness in an image and it represents the achromatic image without any color while the chroma component represents the color information. Image-split block splits the image into number of blocks. Each splitted block is resized using bilinear interpolation technique. The split image is displayed using the video display output block.

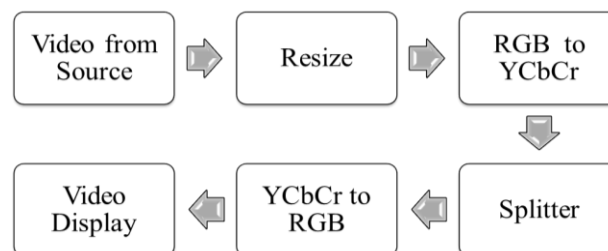


Fig. 2: Block Diagram of Video Splitting.

A. Resize Block:

The Resize block enlarges or shrinks an image by resizing the image along one dimension (row or column). Then, it resizes the image along the other dimension (column or row). If the data type of the input signal is floating point, the output has the same data type. Use the specify parameter to designate the parameters used to resize the image. Important choices in this block are Output size as a percentage of input size, Number of output columns and preserve aspect ratio, Number of output rows and preserve aspect ratio, Number of output rows and

columns. If, for the Specify parameter, we select Output size as a percentage of input size, then the Resize factor in % parameter appears in the dialog box.

B. Need for Color Space Conversion:

The R'G'B' to Y'CbCr conversion and the Y'CbCr to R'G'B' conversion are defined by the following equation:

$$\begin{bmatrix} Y' \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + A \times \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix}$$

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = B \times \left(\begin{bmatrix} Y' \\ Cb \\ Cr \end{bmatrix} - \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \right)$$

The values in the A and B matrices are based on your choices for the Use conversion specified by and scanning standard parameters.

A Bitmap image uses the RGB Planes directly to represent color images. Medical research proved that the human eye has different sensitivity to color and brightness. Thus there came about the transformation of RGB to YCbCr. Medical investigation shows that there are rods about 120 million in number and cones about 6-7 million. Rods are much more sensitive than cones. The rods are not sensitive to color, while the cones which provide much of the eyes sensitivity are found to be located close to a central region called the mocola (Naveen B, Dr.K.R.Nataraj, Dr.K.R.Rekha, 2009). And another reason is that conversion reduces the simulation time or in other words increases the data transfer rate. In Simulink model, the major signal flow is only in two dimensional while the RGB images consists signal more than two dimensional. So the RGB images are converted into Y'CbCr which is two dimensional. Fig. 3. a, b, c, d shows RGB, Y, Cb, Cr image respectively.

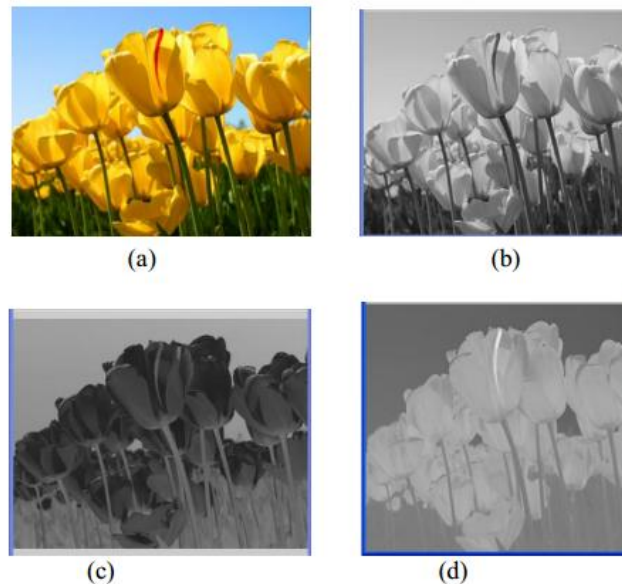


Fig. 3: Color Representation (a) RGB image (b) Y Component (c) Cb Component (d) Cr Component.

C. Splitting System:

The Splitting system consists of two main blocks: sub matrix selection and matrix concatenation.

The Submatrix block extracts a contiguous submatrix from the M-by-N input matrix u as shown in Fig. 4. The Row span parameter provides three options for specifying the range of rows in u to be retained in submatrix output y:

1. All rows: Specifies that y contains all M rows of 'u'.
2. One row: Specifies that y contains only one row from u. The Starting row parameter is enabled to allow selection of the desired row.
3. Range of rows: Specifies that y contains one or more rows from u. The Row and Ending row parameters are enabled to allow selection of the desired range of rows.

The Column span parameter contains a corresponding set of three options for specifying the range of columns in u to be retained in submatrix y: All columns, one column, or Range of columns. The One column option enables the Column parameter, and Range of columns options enables the Starting column and Ending column parameters. The output has the same frame status as the input.

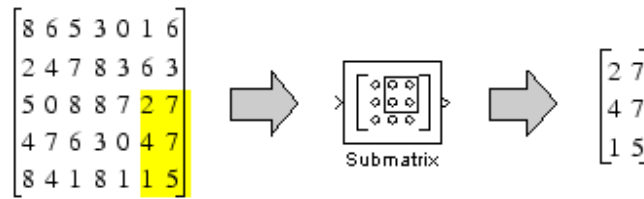


Fig. 4: Submatrix Selection.

The Matrix Concatenate block concatenates the signals at its inputs to create an output signal whose elements reside in contiguous locations in memory. This block operates in either vector or multidimensional array concatenation mode, depending on the setting of its Mode parameter. In either case, the inputs are concatenated from the top to bottom, or left to right, input ports. In vector mode, all input signals must be either vectors or row vectors [1xM matrices] or column vectors [Mx1 matrices] or a combination of vectors and either row or column vectors.

D. Video Display:

The Video Display block sends video data to a DirectX supported video output device or video camera. Alternatively, we can send the video data to a separate monitor or view the data in a window on your own computer screen. For the block to display video data properly, double- and single-precision floating-point pixel values are from 0 to 1. For any other data type, the pixel values must be between the minimum and maximum values supported by their data type. Use the Video output device parameter to specify where we want the video stream to be sent. If On -screen video monitor is selected then video stream is displayed in the Video Display window when we run the model. This window closes automatically when the simulation stops.

Watermark Embedding System:

The block diagram for watermark embedding system is shown in Fig. 5. Both the signals viz., any one part of the splitted video and watermark information are converted to single data type using Datatype conversion block. The Matrix sum is used to accumulate the two signals using matrix addition process.

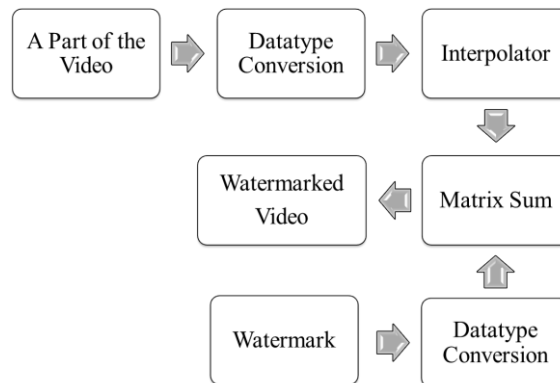


Fig. 5: Block diagram for watermark Embedding System.

A. Bilinear Interpolator:

The watermark embedding system uses a bilinear Interpolation technique to embed a watermark into the host media. In computer vision and image processing, bilinear interpolation is one of the basic resampling techniques.

In texture mapping, it is also known as bilinear filtering or bilinear texture mapping, and it can be used to produce a reasonably realistic image. An algorithm is used to map a screen pixel location to a corresponding point on the texture map. A weighted average of the attributes (color, alpha, etc.) of the four surrounding texels that is the fundamental unit of texture space is computed and applied to the screen pixel. This process is repeated for each pixel forming the object being textured.

When an image needs to be scaled up, each pixel of the original image needs to be moved in a certain direction based on the scale constant. However, when scaling up an image by a non-integral scale factor, there are pixels (i.e., holes) that are not assigned appropriate pixel values. In this case, those holes should be assigned appropriate RGB or gray scale values so that the output image does not have non-valued pixels.

Bilinear interpolation can be used where perfect image transformation with pixel matching is impossible, so that one can calculate and assign appropriate intensity values to pixels. Unlike other interpolation techniques such as nearest neighbor interpolation and bicubic interpolation, bilinear interpolation uses only the 4 nearest pixel values which are located in diagonal directions from a given pixel in order to find the appropriate color intensity values of that pixel.

Bilinear interpolation considers the closest 2x2 neighborhood of known pixel values surrounding the unknown pixel's computed location. It then takes a weighted average of these 4 pixels to arrive at its final, interpolated value. The weight on each of the 4 pixel values is based on the computed pixel's distance (in 2D space) from each of the known points.

B. Datatype Conversion:

The Image Data type conversion is used to convert and scale input image to specified output data type. When converting between floating-point data types, the block casts the input into the output data type and clips values outside the range to 0 or 1.

When converting between all other data types, the block casts the input into the output data type and scales the data type values into the dynamic range of the output data type. For double- and single-precision floating-point datatypes, the dynamic range is between 0 and 1. For fixed-point data types, the dynamic range is between the minimum and maximum values that can be represented by the data type.

C. Matrix Sum:

The Matrix Sum block first converts the input data type to its accumulator data type, and then performs the addition (embedding) operations. The block converts the result to its output data type using the specified rounding and overflow modes. For fixed point operations Integer rounding mode is important. In embedding process Floor mode is used. It rounds both positive and negative numbers towards negative infinity. A proper embedding need the accumulator datatype that inherits via internal rule and also the output datatype has inherited via internal rule. Simulink chooses a combination of output scaling and data type that requires the smallest amount of memory consistent with accommodating the calculated output range and maintaining the output precision of the block and with the word size of the targeted hardware implementation specified for the model.

Video Merging System:

The block Diagram for Video Merging system is shown in Fig.6. The Merging system is used to join all the splitted parts of the video along with the watermarked video. Merging is done by the Matrix Concatenation System (MCS). MCS uses a matrix concatenate block for concatenation.

The Matrix Concatenation block concatenates input matrices u_1, u_2, \dots, u_n along rows or columns, where n is specified by the **Number of inputs** parameter. The block accepts inputs with any combination of built-in Simulink data types and/or fixed-point data types. If all inputs are sample-based, the output is sample-based. Otherwise, the output is frame-based.

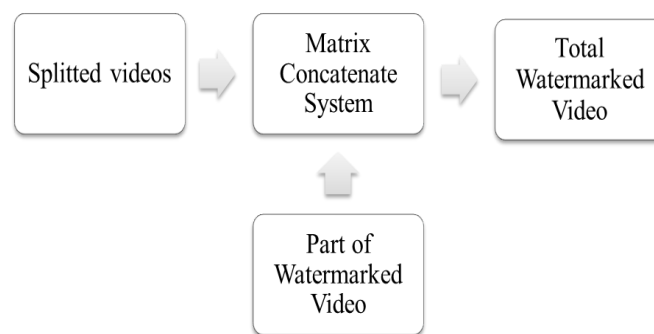


Fig. 6: Block diagram for Video Merging System.

A. Horizontal Matrix Concatenation:

When the Concatenation method parameter is Horizontal, then the block concatenates the input matrices along rows.

$$y = [u_1 \ u_2 \ u_3 \ \dots \ u_n]$$

For horizontal concatenation, inputs must all have the same row dimension, M , but can have different column dimensions. The output matrix has dimension M -by- $\sum N_i$, where N_i is the number of columns in

input u_i ($i = 1, 2, \dots, n$). When some of the inputs are length- M 1-D vectors while others are M -by- N_i matrices, the vector inputs are treated as M -by-1 matrices.

B. Vertical Matrix Concatenation:

When the Concatenation method parameter is Vertical, then the block concatenates the input matrices along columns.

$$y = [u_1; u_2; u_3; \dots; u_n]$$

For vertical concatenation, inputs must all have the same column dimension, N , but can have different row dimensions. The output matrix has dimension $\sum M_i$ -by- N , where M_i is the number of rows in input u_i ($i = 1, 2, \dots, n$). When some of the inputs are length- M_i 1-D vectors while others are M_i -by-1 matrices, the vector inputs are treated as M_i -by-1 matrices. (1-D vector inputs are not accepted for vertical concatenation when the other inputs have column dimension greater than 1.)

Simulink Based Implementation:

Simulink, developed by MathWorks, is a data flow graphical programming language tool for modeling, simulating and analyzing multi-domain dynamic systems. Simulink is widely used in control theory and digital signal processing for multi-domain simulation and Model-Based Design.

Simulink is a block diagram environment and Model-Based Design. It supports system – level design, simulation, automatic code creation and verification of embedded systems. It provides a graphical editor, customizable block libraries for modeling and simulating dynamic systems (Prachi V. Powar, 2013). It enables to integrate MATLAB algorithms into models and export simulation results to MATLAB for further analysis. The overall Simulink architecture model for an interpolated digital watermarking system is shown in Fig.7. The overall architecture is implemented in Simulink using Computer Vision Toolbox. Bilinear Interpolation is achieved by the Resize block in Interpolation mode. The PSNR value is measured using the Statistics toolbox.

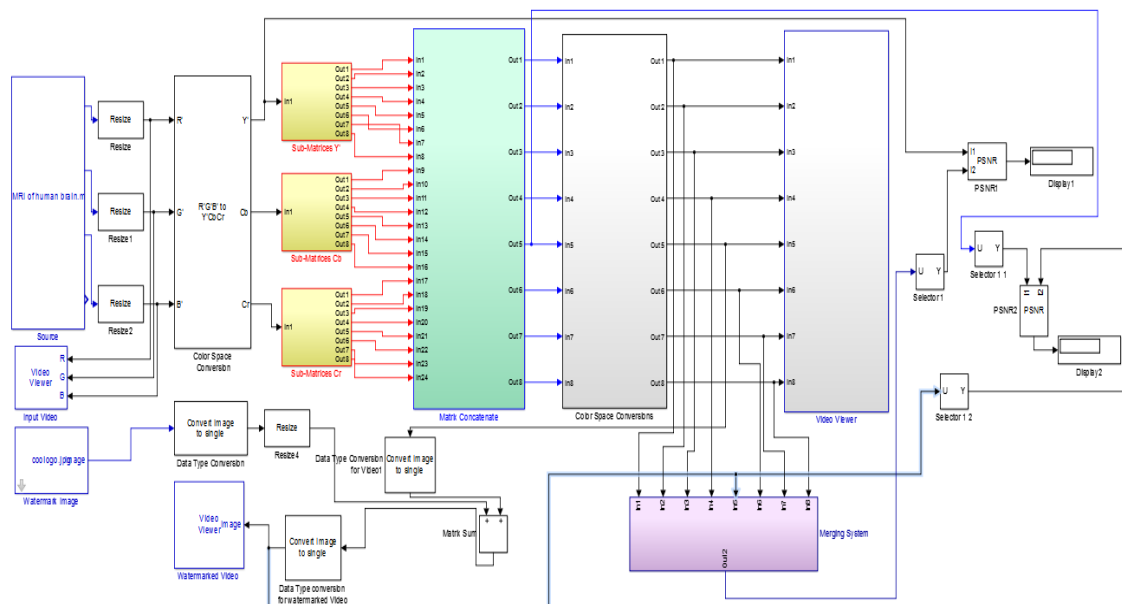


Fig. 7: Overall Simulink Model.

Experimental Results:

A. Methodology for verification:

In order to evaluate the performance of the Simulink model, the model was tested with 3D Human Brain Scan color video clips. The video streams at the rate of 30 frames per second (fps) and 256 x 256 pixels/frame were taken from commercial and medical videos.

For each video stream, a comparison was performed between video stream without watermarking and watermarked video stream. The comparisons were quantified using the standard video quality metric: PSNR, which is a well-known quantitative measure in multimedia processing used to determine the fidelity of a video frame and the amount of distortion found in it, as suggested by Piva.A *et al.*, 2002. The PSNR, measured in decibels (dB), is computed using

$$PSNR = 10 \log_{10} \left(\frac{255}{MSE} \right)$$

$$MSE = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [f(m, n) - k(m, n)]^2$$

where 255 is the maximum pixel value in the grey-scale image and MSE is the average mean-squared error. Here, f and k are the two compared images, the size of each being $M \times N$ pixels (256×256 pixels in our experiment).

B. Analysis of Experimental Results:

The quality of watermark ensures the authenticity of the video frames. To ensure the quality of the video, PSNR values can calculate for the source video and watermarked video. Higher the PSNR values gives higher the quality.

The colored watermark information of size 128×64 with horizontal and vertical resolution of 200 dpi is embedded into the 3D Human Brain Scan video of frame length of 480×360 with frame rate of 30 fps (frames per second). The source video is in rectangular matrix form, for embedding a watermark it is converted into square matrix of fixed frame length say 256×256 . The Splitting system splits the video of size 256×256 into eight parts of frame length 128×64 . Then the watermark is embedded into the source video by watermark embedder. Then the watermarked video of frame length 128×64 is merged with other seven parts of the video to form the total watermarked video of frame length 256×256 .

When Splitting system is not used then the watermark is embedded into the whole frame length of the source video. It will annoy the viewer and also cause reduction in video quality. Splitting system split the video into the size of the watermark and the watermark can be embedded into any part of the splitted video. This method is used to improve the quality of the video and also entertain the viewer.

The figure 8.a gives the resized source video of frame length 256×256 and the figure 8.b gives eight splitted parts of the source video of frame length 128×64 . The figure 8.c gives the watermark information; figure 8.d gives the watermarked video; figure 8.e gives the watermarked video with Splitting system and figure 8.f gives the Watermarked video without splitting system.

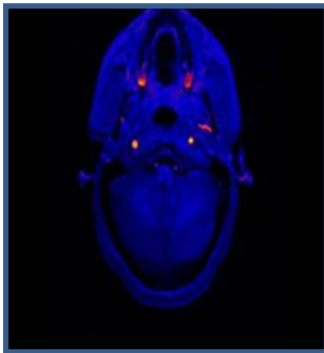


Fig. 8.a: Source Video.

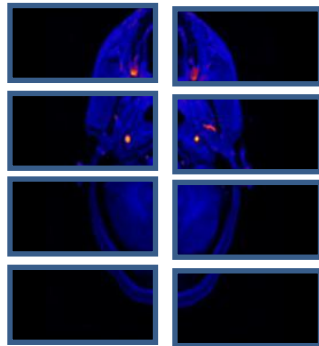


Fig. 8.b: Splitted Videos.



Fig. 8.c: Watermark



Fig. 8.d: Watermarked Video

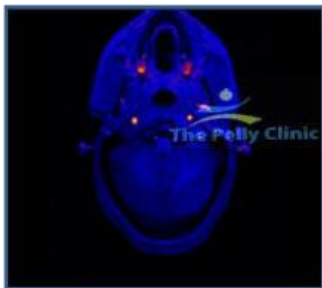


Fig. 8.e: Watermarked Video with Splitting System.



Fig. 8.f: Watermarked Video without Splitting System.



Fig. 9: Comparative PSNR graph.

Table 1: Comparison with other Watermarks.

Research Work	Type of WM	Video Standard	Processing Domain	PSNR (dB)
(Mohanty.S.P and Kougianos.E, 2011)	Visible	AVI	Frequency	22.82
(Poulami Ghosh, Rilok Ghosh, 2012)	Visible	Grey Level Video	Frequency	Around 30 dB
This Paper	Visible	Spatial	Spatial (Interpolation)	26.912 (21-30)

Table 1 gives a comparative perspective with other video Watermarking systems. (Mohanty.S.P and Kougianos.E, 2011) shows only the AVI format video is processed in the frequency domain which gives a PSNR value of 22.82 dB where (Poulami Ghosh, Rilok Ghosh, 2012) implemented Gray level video in the same frequency domain which doesn't contain any Chroma and Luma components and achieves a PSNR value of around 30 dB which is not a consistent one. The proposed watermark system uses source video of any format, where the watermark is processed in the Spatial Domain using Interpolation Technique processed both Chroma and Luma Components and achieves a PSNR value of 26.912 dB and a series of PSNR values from 21-30 dB. The Proposed System achieves a PSNR values which are nearly ahead of Ghosh *et al* system but the main advantage is it uses a colored video instead earlier one uses only a grey level video.

Conclusion:

The Implementation of Digital video watermarking scheme based on the Interpolated Digital Watermarking System splits the source video into eight parts and embeds the colored watermark information into any part of the splitted video using Splitting System and the splitted watermarked video is concatenated with other parts of the video into a watermarked video using Video merging system.

The Splitting and Merging System is used to improve the Quality of the Video by increasing the PSNR values, provides minimum Video Quality degradation and also provides authentication of video with high fidelity. The System model is designed and implemented by using MATLAB/Simulink. When the splitting and merging rate is increased, the PSNR is found to get reduced and also imposes additional computational complexity. In future, the above problem can be addressed to reduce system complexity and improve PSNR further.

REFERENCES

Sonoy Deb Roy, *et al.*, 2013. Hardware Implementation of a Digital Watermarking System for Video Authentication, IEEE Trans. On circuits and systems for Video Tech., 23(2).

- Gwenaél, A.D. and J.L. Dugelay, 2003. A guide tour of video watermarking, *Signal Process. Image Commun.*, 18(4): 263-282.
- Piva, A. *et al.*, 2002. Managing copyright in open networks, *IEEE Trans. Internet Comput.*, 6(3): 18-26.
- Shoshan, Y. *et al.*, 2008. VLSI watermark implementations and applications, *Int. J. Information Technol. Knowl.*, 2(4): 379-386.
- Li, X., Y. Shoshan, A. Fish, G.A. Jullien and O. Yadid-Pecht, 2008. Hardware implementations of video watermarking, in *International Book Series on Information Science and Computing*, 5. Sofia, Bulgaria: Inst.Inform. Theories Applicat. FOITHEA, pp: 9-16.
- Naveen, B., Dr. K.R. Nataraj, Dr. K.R. Rekha, 2009. Design of Simulink Model for Real Time Image Splitting” *International Journal of Computational Engineering Research (ijceronline.com)*, 3(1).
- Cyril Prassana Raj, P., Dr. S.L. Pinjare and T.N. Swamy, 2012. FPGA implementation of efficient algorithm of image splitting for video streaming data, *International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com*, 2(5): 1244-1247.
- Rafael C. Gonzalez and Richard E. Woods, 2008. *Digital Image Processing*, Pearson Prentice Hall, 3 Edition.
- Prachi V. Powar, 2013. Design of Digital Video Watermarking Scheme using Matlab Simulink, *IJRET*, 2(5).
- Mohanty, S.P. and E. Kougianos, 2011. Real-time perceptual watermarking architectures for video broadcasting, *J. Syst. Softw.*, 84(5): 724-738.
- Poulami Ghosh, Rilok Ghosh, 2012. A Novel Digital Watermarking Technique for Video Copyright Protection, *CS & IT-CSCP*.
- Prabhishek Singh, R.S. Chadha, 2013. A Survey of Digital Watermarking Techniques, Applications and Attacks, *IJEIT*, 2(9).
- Kougianos, E., S.P. Mohanty, *Simulink Based Architecture Prototyping of Compressed Domain MPEG-4 Watermarking*.
- Sarawathi, M., 2011. Lossless Visible Watermarking for Video, (*IJCSIT*) *International Journal of Computer Science and Information Technologies*, 2(3): 1109-1113.
- User’s Guide of Computer vision System Toolbox™, *Interpolation Methods, Geometric Transformations*-chapter 5.