Middle-East Journal of Scientific Research 24 (9): 2786-2793, 2016
ISSN 1990-9233
© IDOSI Publications, 2016
DOI: 10.5829/idosi.mejsr.2016.24.09.23934

Authentication of Video Streams Using Interpolated Digital Watermarking System Using QR Code

[1]B. Shanmugham, [1]A. Asokan, [2]K.P. Ramakrishnan and [1]D. Sivakumar

[1]Department of Electronics and Instrumentation Engineering, Annamalai University, Tamil Nadu, India
[2]Department of Electronics and Communication Engineering, Rajiv Gandhi College of Engineering and Technology, Pondicherry, India

Abstract: This paper presents a software implementation of an interpolated digital watermarking system that can insert Quick Response (QR) Code into any part of the splitted video which provides authentication. 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. To achieve high PSNR value, the proposed system employs two main systems namely, Video splitting system and Video merging system. Experimental results show that a video authentication system using interpolated watermarking features minimum video quality degradation and provides good PSNR values.

Key words: Video Authentication · Digital Watermarking · Quick Response (QR) Code · Video splitting · matrix concatenation · Visible watermarking · Sub-matrices · Bilinear interpolation

INTRODUCTION

The advances in electronics and information technology, together with the rapid growth of techniques for powerful digital signal and multimedia processing, have made the distribution of video data much easier and faster [1-3]. However, concerns regarding authentication of the digital video are mounting, since digital video sequences are very susceptible 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. In such cases, the video data should be credible. 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 [4, 5]. 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 [6]. 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 potential commercial 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 [6], uncompressed digital video both on software [6] and hardware platforms. Contrary to still image WM techniques, new problems and new challenges have emerged in video WM applications. The Japanese corporation Denso Wave invited a quick response (QR) code which is a two dimensional barcode. In this information is encoded in both the vertical as well as horizontal direction, thus by a traditional bar code it holds up to several hundred times more data (Fig 1). A considerably greater volume of information than a 1D Barcode holds by QR Codes (Fig 1).

Corresponding Author: B. Shanmugham, Department of Electronics and Instrumentation Engineering, Annamalai University, Tamil Nadu, India.

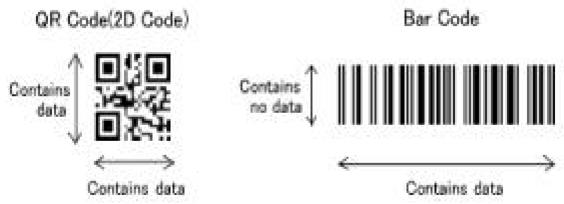

Fig. 1: QR Code and D bar code

The main objective of the paper is to describe efficient software-based digital video watermarking system that can insert QR code in the splitted video with minimum video quality degradation which is achieved by Video Splitting system and video merging system and 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 [7]. Then row mean and column mean or each part is obtained. By using splitting higher precision and display in different Screen [8]. 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 [9].

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, image elements and 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 2.

Video Splitting System: Video splitting is not an easier task when compared to an image splitting. 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.

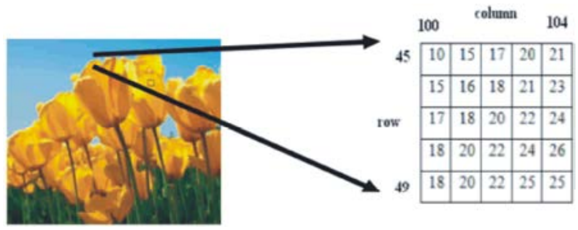

Fig. 2: Structure of Digital Image/Video

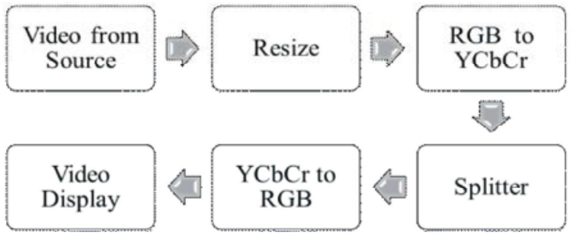

Fig. 3: Block Diagram of Video Splitting

The block diagram of video splitting system is shown in Fig. 3. Video from the Multimedia source is applied to the video input block. 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.

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 to use 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 can select Output size as a percentage of input size, the Resize factor in% parameter appears in the dialog box.

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:

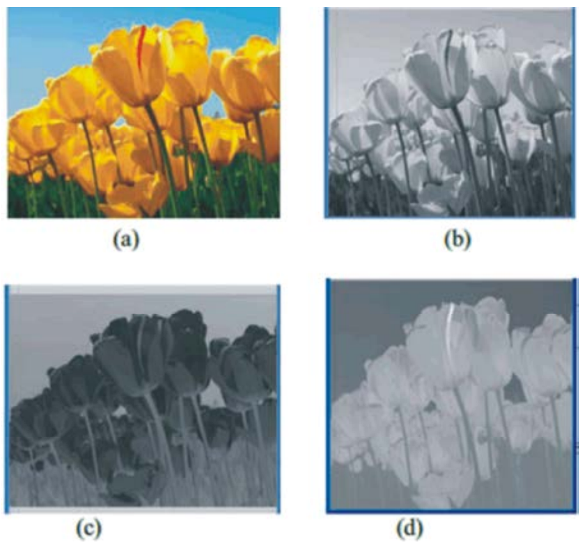

Fig. 4: Color Representation (a) RGB image (b) Y Component (c) Cb Component (d) Cr Component

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

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

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 are 120 million in number and cons are 6-7 million. Rods are much more sensitive than cons. The rods are not sensitive to color, while the cons which provide much of the eyes sensitivity are found to be located close to a central region called the mocula [7]. And another reason is conversion reduces the simulation time in other words increases the data transfer rate and 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. 4. a, b, c, d shows RGB, Y, Cb, Cr image respectively.

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

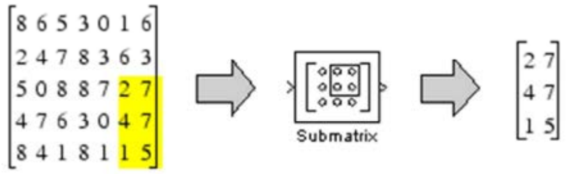

Fig. 5: Submatrix Selection

The Submatrix block extracts a contiguous submatrix from the M-by-N input matrix u as shown in Fig. 5. The Row span parameter provides three options for specifying the range of rows in u to be retained in submatrix output y:

All Rows: Specifies that y contains all M rows of ‘u’.

One Row: Specifies that y contains only one row from u. The Starting row parameter is enabled to allow selection of the desired row.

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.

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.

Video Display: To 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 To 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 (QR code) information are converted to single data type using Datatype conversion block. A standard symbol structure of QR code consists of encoding region, quiet zone and function pattern. Specially, function pattern is composed by position detection patterns with separators and Timing and Alignment patterns. It is worth to figure out that function patterns and quiet zone maintain the same after masking. What's more, the ability of error correcting with Reed-Solomon code is the conspicuous feature of QR Code. Consequently, the incomplete extracted watermark image can be rectified for effective decoding.

The Matrix sum is used to accumulate the two signals using matrix addition process.

Bilinear Interpolator: The watermark embedding system uses a bilinear Interpolation technique [10] 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 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.

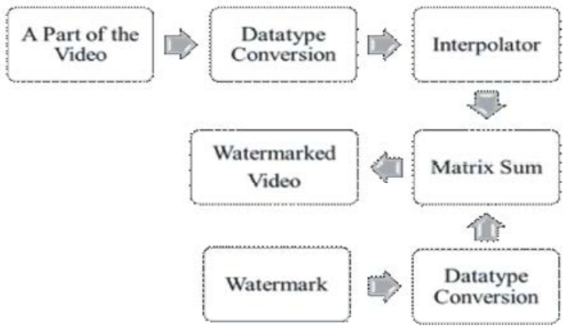

Fig. 6: Block diagram for watermark Embedding System

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.

Datatype Conversion: The Image Datatype 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 data types, 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.

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.

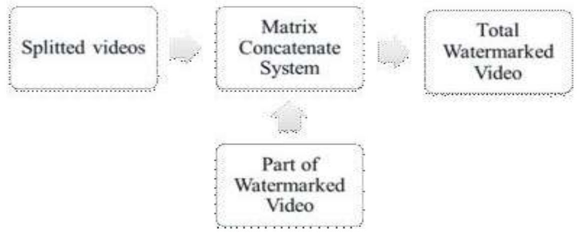

Fig. 7: Block diagram for Video Merging System

In embedding process Floor mode is used. It rounds both positive and negative numbers towards negative infinity. A proper embedding need the accumulator datatype is in 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.7. 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 u1, u2,. .., un 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.

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

y = [u1 u2 u3. .. un]

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- Ni, where Ni is the number of columns in input ui (i = 1, 2,. .., n).When some of the inputs are length-M 1-D vectors while others are M-by-Ni matrices, the vector inputs are treated as M-by-1 matrices.

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

y = [u1;u2;u3;...;un]

For vertical concatenation, inputs must all have the same column dimension, N, but can have different row dimensions. The output matrix has dimension Mi-by-N, where Mi is the number of rows in input ui (i = 1, 2,. .., n).When some of the inputs are length-Mi 1-D vectors while others are Mi-by-1 matrices, the vector inputs are treated as Mi-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. Its primary interface is a graphical block diagramming tool and a customizable set of block libraries. It offers tight integration with the rest of the MATLAB environment and can either drive MATLAB or be scripted from it. 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 [11]. 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. 8. 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.

Experimental Results

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.

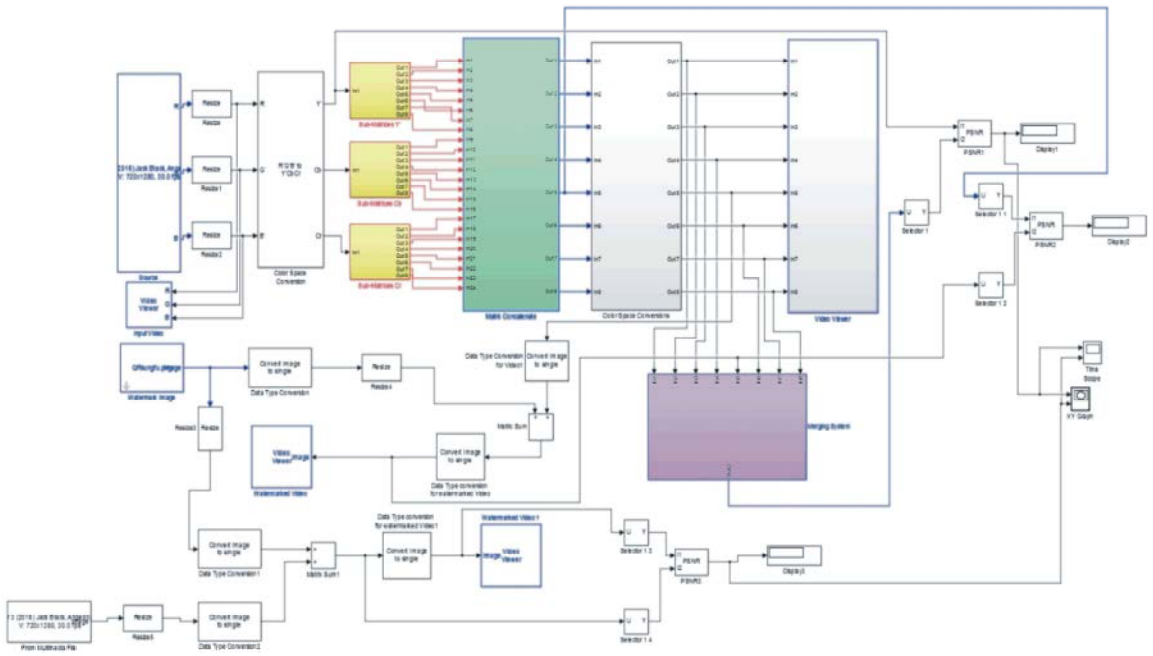

Fig. 8: Overall Simulink Model

Fig. 9: QR code

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 *et al*. [3] and Strycker *et al*. The PSNR, measured in decibels (dB), is computed using

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

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

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 × N pixels (256 × 256 pixels in our experiment).

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 x 64 with horizontal and vertical resolution of 200 dpi is embedded into the 3D Human Brain Scan video of frame length of 480 x 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 x 256. The Splitting system splits the video of size 256 x 256 into eight parts of frame length 128 x64. Then the watermark is embedded into the source video by watermark embedder. Then the watermarked video of frame length 128 x 64 is merged with other seven parts of the video to form the total watermarked video of frame length 256 x 256.

There are many quality measurement parameters, Peak signal-to-noise ratio, often abbreviated as PSNR, is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. Because many signals have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel scale.

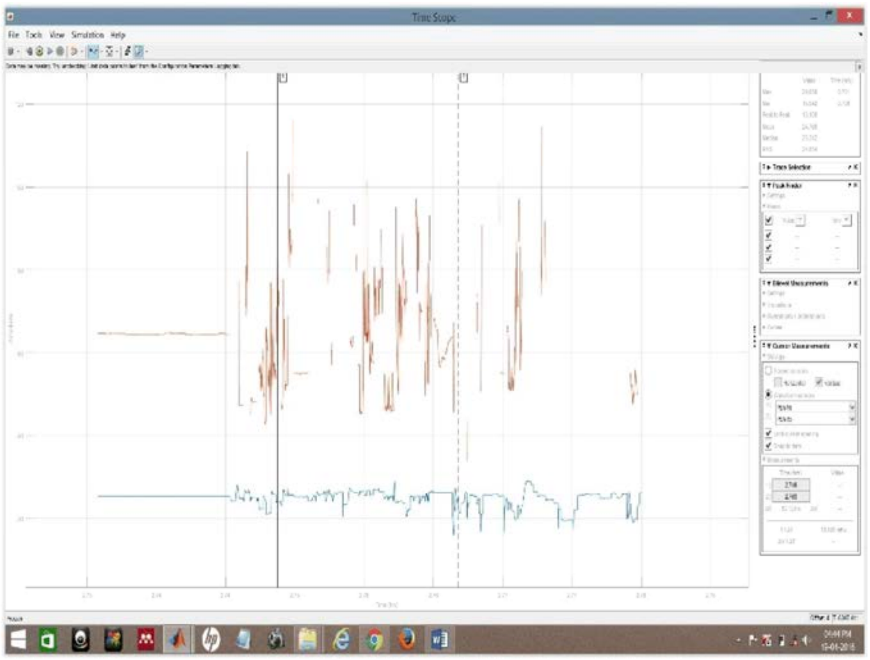

Fig. 10: PSNR

PSNR is most commonly used to measure the quality of reconstruction of lossy compression codecs (e.g., for image compression). The signal in this case is the original data and the noise is the error introduced by compression. When comparing compression codecs, PSNR is an approximation to human perception of reconstruction quality. Although a higher PSNR generally indicates that the reconstruction is of higher quality, in some cases it may not. A proper caution is to be taken with the range of validity with this metric as it is only conclusively valid when it is used to compare results from the same codec (or codec type) and same content.

The above result Fig.10 shows two PSNR values namely PSNR1 and PSNR3, these values are obtained from the PSNR acquiring block from the block diagram. From the figure it is observed that the max value is 29.050 and min value is 15.942.

Typical values for the PSNR in lossy image and video compression are between 30 and 50 dB, provided the bit depth is 8 bits, where higher is better. In our prosed technique we are getting higher PSNR value when compared to existing technique which is clearly evident from the result.

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.

CONCLUSION

The Implementation of Digital video watermarking scheme based on the Interpolated Digital Watermarking System using QR code 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.

REFERENCES

1. Shanmugham, B., Dr. A. Asokan, Dr. K.P. Ramakrishnan and Dr. D. Sivakumar, 2014. Authentication of Video Streams using Interpolated Digital Watermarking System, Advances in Natural and Applied Sciences, 8(22): 1-9.

2. Sonoy Deb Roy, Xin Li, Yonatan Shoshan, Alexander Fish and Orly Yadid-Pecht, 2013. Hardware Implementation of a Digital Watermarking System for Video Authentication, IEEE Trans. On circuits and systems for Video Tech., 23(2).
3. Gwenael, A.D. and J.L. Dugelay, 2003. A guide tour of video watermarking, Signal Process. Image Commun., 18(4): 263-282.
4. Piva, A., F. Bartolini and M. Barni, 2002. Managing copyright in opennetworks, IEEE Trans. Internet Comput., 6(3): 18-26.
5. Kougianos, E. and S.P. Mohanty, XXXX. Simulink Based Architecture Prototyping of Compressed Domain MPEG-4 Watermarking.
6. Shoshan, Y., A. Fish, X. Li, G.A. Jullien and O. Yadid-Pecht, 2008. VLSIwatermark implementations and applications, 2008. Int. J. Information Technol. Knowl., 2(4): 379-386.
7. 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, no. 5. Sofia, Bulgaria: Inst. In- form. Theories Applicat. FOI ITHEA, Jun. 2008, pp: 9-16. (supplement to the Int. J. Inform. Technol. Knowledge, pp: 2.
8. Naveen, B., Dr. K.R. Nataraj and Dr. K.R. Rekha, XXXX. Design of Simulink Model for Real Time Image Splitting, International Journal of Computational Engineering Research (ijceronline.com), 3(1).
9. 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.
10. Sarawathi, M., 2011. Lossless Visible Watermarking for Video, (IJCSIT) International Journal of Computer Science and Information Technologies, 2(3): 1109-1113.
11. Rafael C. Gonzalez and Richard E. Woods, 2008. Digital Image Processing", Pearson Prentice Hall, 3 Edition, 2008.