Real Time Rain Removal from Live Video using FPGA and Raspberry Pi

Eman Yassien
Software Engineering Department, The World Islamic Science and Education University, Amman, Jordan

Raja Masadeh
Software Engineering Department, The World Islamic Science and Education University, Amman, Jordan

Omar Almomani
Network Department, The World Islamic Science and Education University, Amman, Jordan

Esra Masadeh
Computer Science Department, Jordan University of Science and Technology, Irbid, Jordan

ABSTRACT

The usage of HD videos has become widely spread into almost every aspect of modern life. There's a need to develop filters and image processing techniques to take a human observer's attention off of any abnormality in a video stream, such as surveillance video footage. Software techniques solely might not have the ability to process a great number of frames efficiently unless operating on a high-tech device such as a state-of-the-art supercomputer however that's not an affordable to most users. Hence there's a demand for a reliable yet, affordable method to filter HD videos, which is the aim of this research.

Removing Raindrops from a real-time video stream requires heavy image processing and computations which may cause an observer to miss a piece of information like a car's plate number. The proposed technique takes raw image input from a video frame and converts it into a binary image using pure hardware implemented with an FPGA circuit. Then the binary image is processed using FastICA technique under Raspberry Pi machine to make raindrops simpler to remove and then renders the video frame to a High Definition Multimedia Interface (HDMI) cable to be displayed on the screen.

General Terms

Image Processing

Keywords

HD video, FPGA, FastICA, Raspberry Pi, HDMI

1. INTRODUCTION

Handling real-time videos have become necessary in many applications and day-to-day activities. Recognizing a trouble maker within a group in a live feed, for example, will help prevent big problems from happening in a society. Also recognizing a suspect that walks outside the parameters of an institute (or even a home) in bad weather conditions will also help to keep them outside the institute, or make the security officer take necessary precautions to prevent this suspect from causing trouble.

Bad weather conditions, like heavy Rain, makes recognition of objects in real-time video feed hard, and an automated system might not have the required speed in performing recognition with high-definition (HD) videos with the current hardware technology that is connected to surveillance systems, or might cost a lot to install such hardware. Still, most of the pattern recognition algorithms might not produce acceptable results because they are not able to process enough number of frames per second.

In this research, we opt to incorporate specific hardware to help in recognition of Raindrops in live HD video stream, to be able to remove the noise caused by this Rain, and produce better pattern recognition results.

We shall use a Field-Programmable Gate Array (FPGA) circuit to process images initially (by converting them to binary code) and then use Raspberry Pi-3 Model B to complete the processing of the frames (Rain detection and displaying the result on connected HDMI screen).

2. REAL-TIME VIDEO PROCESSING RELATED RESEARCH

When dealing with real-time videos, there are mostly two methods for processing them: either by splitting the processing task over multiple processors, or by using pipeline architecture, where set of components fall between the video source (the camera) and a display or output peripheral to speed up the processing of input before producing the required output.

Pipelining hardware components are known to be faster than relying on software processors [1]. The only delay in pipeline processors happens before the pipeline gets full, but once it's filled up, the processed video segments are delivered much faster than parallel processing.

The use of Field-Programmable Gate Array (hereafter FPGA) circuits enhances the pipelining greatly. Bhandari et al [2] used FPGA to perform mean and median filters on received real-time video stream on Virtex-4 FPGA using Partial reconfiguration. This system recorded noticeable enhancement in processing speed and configuring of bit stream size.

When receiving live video, distinguishing between foreground and background items is needed for applications like object detection and tracking. The authors of [3] used a System-on-chip FPGA hardware system, where a processor and an FPGA circuit are integrated, to perform Video Content Analytics (VCA) and Image Signal Processing (ISP) to identify background and foreground entities, under a pipeline a pipeline architecture.

Enhancing the hardware capabilities in a pipelined processing architecture would surely make processing faster, yet, the pattern recognition process itself needs a code and a reliable algorithm to produce the desired output. The Independent Component Analysis (ICA) algorithm was formulated by [4], and enhanced by introducing a faster version in 1995 by [5], which was called the "FastICA algorithm".

In most recent research, ICA algorithm is mostly used for noise removal, by converting Gaussian elements in an image into non-Gaussian, like the work of [6]. They called this principal IPCA-ICA and used it to compute the principal components for a covariance-free matrix. They applied this algorithm to human face recognition, in a real-time image feed. The algorithm merges the output of a Principal Component Analysis (PCA) with the Independent Component Analysis (ICA) that run in parallel to get the most independent component to make the best features recognition.

Working with a stable image is definitely better for processing. But with real-time feed, it's not possible, so the authors in [7] proposed an algorithm for video stabilization, which uses the inter-frame homography estimation as a motion model to reduce error that might build up with every scene change. A Kalman filter is used to smooth frame images, along with a mosaic algorithm to maintain the stability of the video and preserve the high resolution of the streaming video.

Live video streaming was used to monitor the heart pulse in [8] by reading images of the face through a live webcam. The technology used is based on the Principal Component Analysis (PCA) with a convenient channel to monitor the pulse rate in a non-contact manner. The results sound promising for future monitoring of sick people at their homes but still needs more development and enhancements on the processing speed.

3. CURRENT TECHNOLOGY’S CONSTRAINTS

When a video is captured with a fixed camera, it can be viewed as a number of frames in an array, such as 〖[F_1, 〖F〗_2, F_3, 〖…, F〗_n]〗^T (where T is the Transpose and F is the Frame-number). Each frame does contain a mix of a background, and a dynamic component. If we consider Rain as the only dynamic component in the video, then it can be extracted from frames by using FastICA algorithm.

The extraction method depends on multiple frames, by comparing their hysteresis together and finds the odd one in the group, which will be considered as a Raindrop.

The problem with this method is that any motion in the video is considered as Raindrops, such as a walking human or a moving vehicle. Thus the method is useless

Raindrops intensity is so low in comparison with other moving objects in the video. Even while gray values of Raindrops are weak when converting the image to grayscale and try to detect Rain by FastICA, but there would still be contrast among Raindrop and peripheral pixel. Of course, image gradient magnitude is still reflected by the image edges. This means we can depend on edge information to convert the image of the frame into a binary image.

Binarization of an image can be done by either software or hardware. When talking about a video, it means the process should be as fast as possible, and thus depending on software will not provide good results, so instead, we will depend on hardware. To be more specific, we will depend on VHDL and Xilinx processors to make an FPGA for this purpose.

A Binary image is an image that contains 2 colors only, usually black and white. Depending on pixel threshold, its value will be converted to either 0 or 1.

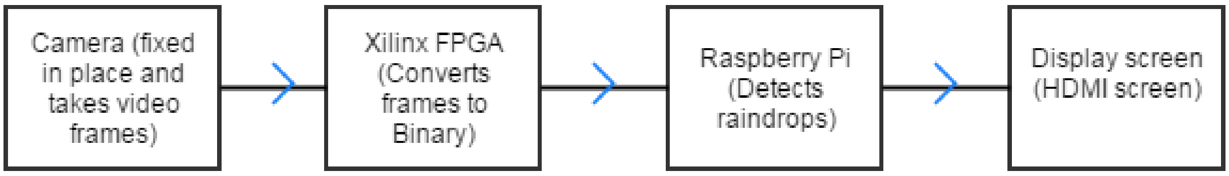
Camera (fixed in place and takes video frames)
Xilinx FPGA (Converts frames to Binary)
Raspberry Pi (Detects raindrops)
Display screen (HDMI screen)

Figure 1: Proposed System's Hardware Layout

4. THE PROPOSED SYSTEM

VHDL is very high-speed hardware description language that does not download software, but instead it connects logic gates that compose the FPGA circuit together in order to achieve a pure hardware system (no actual software in it) that does need job; it is used to write a code on FPGA which consists of a lot of logic gates that can process data fast. VHDL only automates the connection process, but in fact, it does not download software on preconfigured hardware device like C# or android programming.

The streaming video frames provide raw images, so the first step of the algorithm is to convert this image to RGB format, then to binary image. After that, it is sent through RS232 to a cable converter (from RS232 to USB) and input the signal into Raspberry Pi 3 Model B where the software method resides.

Although using this hardware makes processing so much faster but we can't make full processing on hardware alone. This means only part of the process (image binarization) can be done on hardware. Now, Raspberry Pi model B is a single board computer, it is very small in size and runs Linux OS, and can be programmed like any computer, it is just small in size and easier to use with other hardware parts like FPGA, and costs much less than using a supercomputer for almost the same processing capabilities. Figure 1 below shows the hardware components layout of the proposed system.

The binary image conversion system was implemented in Xilinx FPGA Virtex-7 XC7V585T by VHDL. The general circuit design of the Virtex-7 family is shown in figure 2.

The process of converting raw images from the camera to binary images is shown in Figure 3.

The proposed software method on Raspberry Pi would consider adding conditions on Rain identification, such as comparing area and direction.

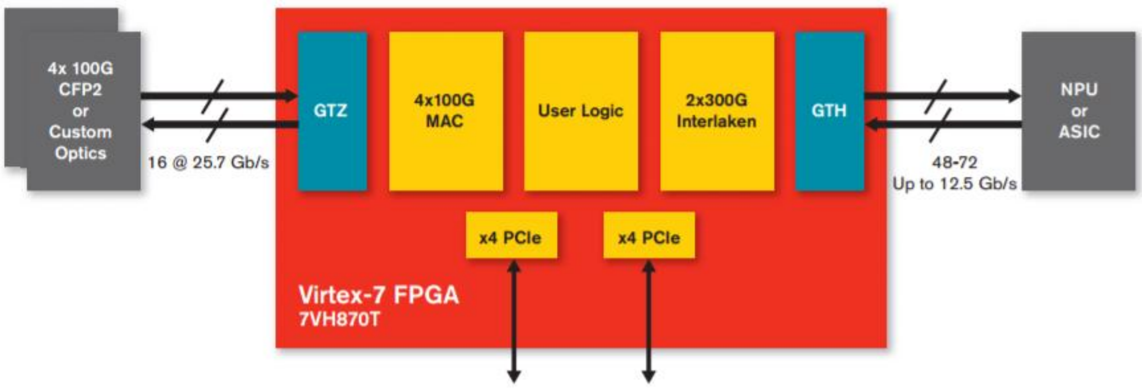

Figure 2: [14] Basic Circuit Design for Virtex-7

4.1 Video reconstruction

After detecting and isolating Rain from a frame, the Raindrops need to be replaced with the original background in the image. Those missing pieces in the background are gotten from neighbor frames.

After getting the binary image from the FPGA circuit, FastICA algorithm is applied to the frame image to detect motion in objects, and frames are reconstructed until all motion is detected and processed.

Usually, Raindrop size is much smaller than a moving object in the video, and thus we can depend on that to identify if the motion is from a Raindrop or not. To decrease the error even more, one can depend on angle, where Rain usually falls vertically or with a maximum angle of 45 degrees, thus, if a moving object is not moving in a straight line within 3 frames, or is moving with an angle greater than 45 (with respect to X axis), it is not considered as Rain. The movement angle can be stated as the angle between major ellipse axis and frame edge because -Raindrop is considered as an area connected domain in a given binary image.

The following chart explains how the software installed on the Raspberry Pi works.

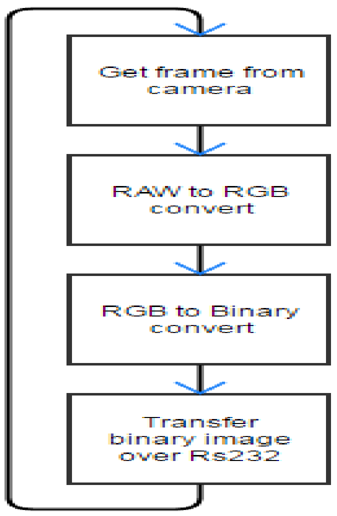

Figure 3: converting raw image to binary sequence of actions

The brewer Method explained in [9] works over a sequence of steps that starts by identifying regions affected by Raindrops if they have a short-duration intensity spike. Raindrops are assumed to have the same characteristics; they are streaks with a consistency range of aspect ratios. If a certain region doesn't fall within this range, it is ignored and considered as having Raindrops. Since Raindrops might change direction, this information is used to reduce the error margin and reduce faulty recognition. This technique is similar to the proposed technique of this research, and the test results are compared with those used by Brewer Technique.

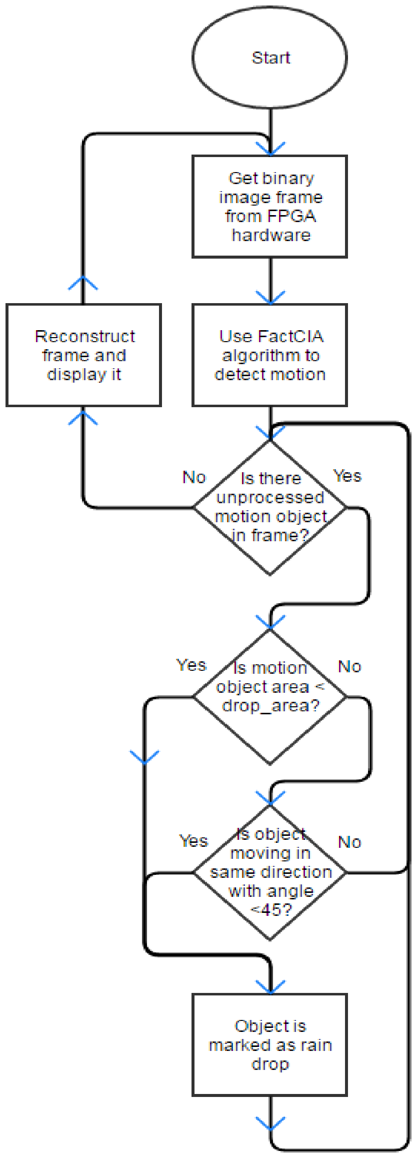
Start
Get binary image frame from FPGA hardware
Use FactCIA algorithm to detect motion
Reconstruct frame and display it
No
Is there unprocessed motion object in frame?
Yes
Yes
Is motion object area < drop_area?
No
Yes
Is object moving in same direction with angle <45?
No
Object is marked as rain drop

Figure 4: Raspberry Pi Software Algorithm

5. SYSTEM PERFORMANCE AND TEST RESULTS

The following image (Figure 5) is a frame from a video recorded in Irbid city near the international grand academy – Jordan. This is the original image with Raindrops that was fed into the proposed system.

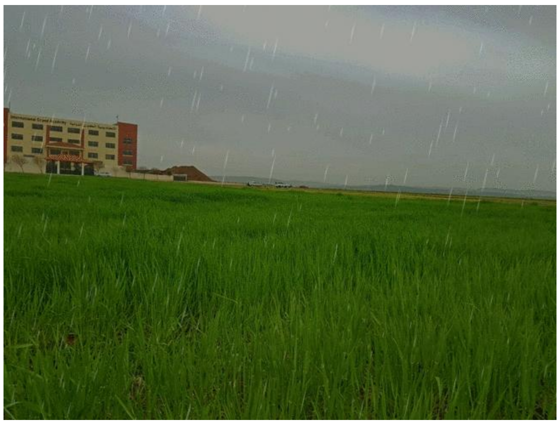

Figure 5: original Video Frame

This video frame was fed into FPGA as a raw image that was converted to RGB coloring scheme. Then the RGB image is converted to binary using the Xilinx FPGA Virtex-7 circuit. Figure 5 shows the image after it was converted to binary.

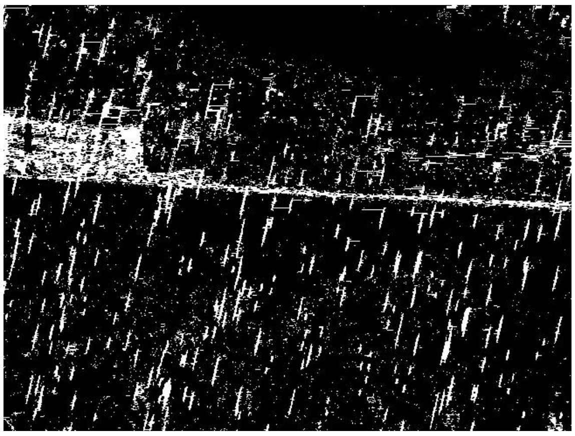

Figure 6: Resulting Binary image

Now, this image was fed into a system that runs Brewer Technique, and another system that runs the proposed system. Figure 7 shows the result after applying Brewer Method.

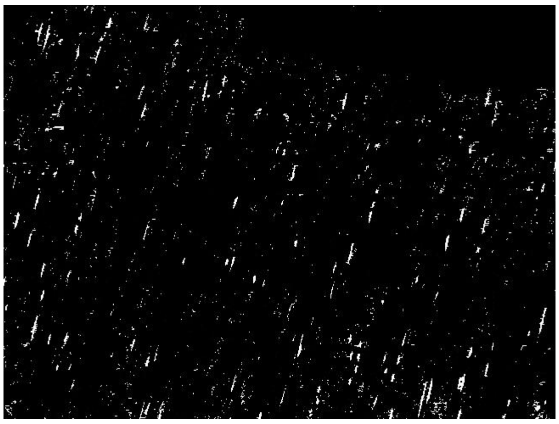

Figure7: raindrops extracted from the binary image

Comparing Brewer Technique with the proposed technique shows better identification of Raindrops for the same image. Figure 8 shows Raindrops extracted from the image using the proposed technique

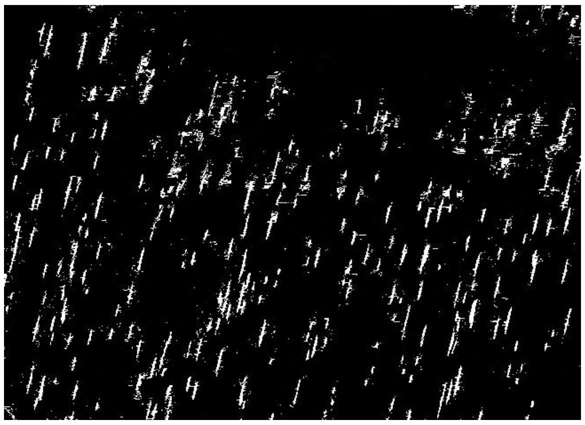

Figure 8: raindrops in image using the proposed technique

Now, the original image is reconstructed after subtracting the identified raindrops from the original image. The result from applying Brewer Method is shown in Figure 9.

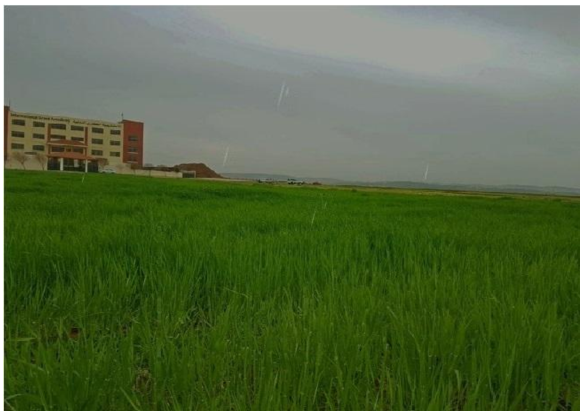

Figure 9: resulted image after applying Brewer method

Reconstructed image from the proposed system is shown in Figure 10. One can notice big enhancement in the reconstructed image using the proposed technique, with less amount of raindrops, and better quality of the image's "background" elements.

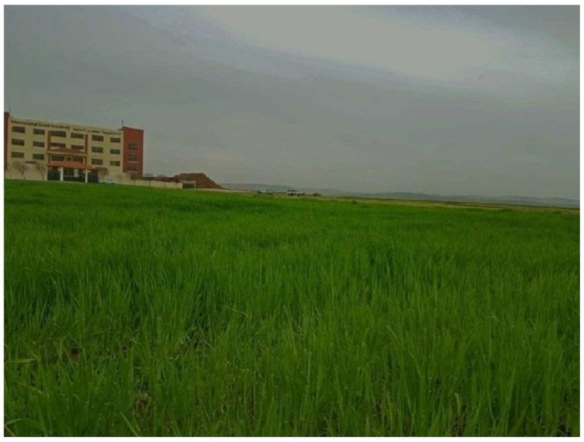

Figure 10: resulted image after applying proposed method

6. CONCLUSION

The proposed system is capable of removing Raindrops from a High-Definition Video frame that is captured live. The processing speed of the frame was very similar to doing the same task on a supercomputer with more complex code, however at a small fraction of the cost, with a simple and small code. This technique could be integrated with monitoring cameras which require fast removal of abnormalities in footage such as Heavy Rain.

7. REFERENCES

[1] Quinn, Heather, Miriam Leeser, and Laurie Smith King. "Implementing image processing pipelines in a hardware/software environment." High Performance Embedded Computing Workshop. 2002.

[2] S. U. Bhandari, S. Subbaraman, S. Pujari and R. Mahajan, "Real Time Video Processing on FPGA Using on the Fly Partial Reconfiguration," 2009 International Conference on Signal Processing Systems, Singapore, 2009, pp. 244-247.

[3] A. Safaei and Q. M. J. Wu, "A system-level design for foreground and background identification in 3D scenes," 2016 IEEE International Symposium on Circuits and Systems (ISCAS), Montreal, QC, 2016, pp. 2571-2574.

[4] Comon, Pierre. "Independent component analysis, a new concept?." Signal processing 36.3 (1994), pp. 287-314.

[5] Lee, Te-Won, Anthony J. Bell, and Russell H. Lambert. "Blind separation of delayed and convolved sources." Advances in neural information processing systems (1997): 758-764.

[6] I. Dagher and R. Nachar, "Face recognition using IPCA-ICA algorithm," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 6, pp. 996-1000, June 2006.

[7] J. Dong and H. Liu, "Video Stabilization for Strict Real-Time Applications," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 27, no. 4, pp. 716-724, April 2017.

[8] M. Lewandowska, J. Rumiński, T. Kocejko and J. Nowak, "Measuring pulse rate with a webcam — A non-contact method for evaluating cardiac activity," 2011 Federated Conference on Computer Science and Information Systems (FedCSIS), Szczecin, 2011, pp. 405-410.

[9] Brewer, Nathan, and Nianjun Liu. "Using the shape characteristics of rain to identify and remove rain from video." Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR). Springer Berlin Heidelberg, 2008.