

# EXPERIENCE ON VIDEO PROCESSING LABORATORY

Sandra Cancino Suárez<sup>1</sup>, Enrique Estupiñán Escalante<sup>2</sup>, Dominik Makowski<sup>3</sup>

**Abstract** — Noise in digital video is typically handled with simple inter-frame operations and since these operations could be used in more advance video processing is a good starting point for any research on video. Although, there are many barriers to learn and comprehend video processing topics; these are related mainly to the learning curve of the hardware and software for video processing and the inherent complexity of video. In order to overcome the former, a set of guides ranging from video standards, to noise reduction algorithms, in simulation, Simulink and in hardware TMS320DM642, were developed to aid the self-learning processing either for researchers or students. In this paper we present the result of two different courses where this self-learning methodology was used and the results of selected noise removal algorithms.

**Index Terms** — Video, noise, video processing, self-learning.

## 1. GENERAL INFORMATION

Digital Video Processing is a field in constant evolution, its applications range from video surveillance to medical imaging, with many research groups from universities and from private investors, around the world working to get new products in new trends like 3D or enhancing established products. Noise reduction is the simplest video processing task, yet it is the gate to more complex processing so people can gradually improve its abilities and work in topics like segmentation and feature extraction. Motivation to work in this field comes from the wide range of emerging applications such as mobile games, interactive programs for the Digital Terrestrial Television, DTT (DVB-T), IPTV services but also traditional ones like medical imaging or traffic control [1]. In the region, Colombian production of Television is well positioned, but in the mentioned areas of video processing, qualified labor is required. As an answer some undergraduate programs include elective courses on video. Our interest is to develop self-learning strategies so any person with an electronics engineering background would be able to improve his competences in video processing.

The group wants to take advantage of the knowledge on image processing implementations either in software and hardware and use them to explore video systems, specially the acquisition, processing and the segmentation stages in

this first project. At the same time, tutorials developed based on the experiences of the group, will be used in the elective courses covering tasks such as video acquisition, handling of drivers on Matlab® or Labview®, toolbox functions usages; DSP hardware configuration and interface between the software and hardware. Besides, in an intensive course on signal processing in cooperation with National University, the tutorials were used.

## 2. PROJECT DESCRIPTION

The project is focused in processing video in order to reduce the noise, to extract movement and segment objects in video. Several algorithms are studied, five of them are implemented and compared in simulation and then they are implemented in hardware. Objective and subjective measurements are made in order to determine the performance of the algorithm.

### 2.1 Hardware

The DM642 EVM board is the hardware on which the selected algorithms are implemented, it has a 720 Mhz, TMS320DM642 media processor, 32 Mb SDRAM, 32 Kb I2C, 2 composite video inputs, 1 composite video output, 1 VGA video output; as suggested by TI, JTAG (XDS510) is used to communicate with board.

### 2.2 Software

The simulation of algorithms is done in Simulink v7.4 with Matlab R2009b embedded functions and the *Video and Image Processing Blockset* blocks to reduce the number of function needed to create by users, thus diminishing the computational cost of Matlab embedded functions. To download the algorithms to the evaluation board, Code Composer Studio, CCS v3.3 is used. The video samples were selected to have variety, some contains people moving fast, other video contains slow people motion; in addition, some contains static background as follows:

- **Video 1 (ice60.avi):** it has fast and almost unpredictable motion of the persons; the background is not complete static with limited gray range.

1 Sandra Cancino Suárez, Instructor Professor and Researcher in the University Escuela Colombiana de Ingeniería in Bogotá, Colombia, sandra.cancino@escuelaing.edu.co

2 Enrique Estupiñán Escalante, Assistant Professor and Researcher in the University Escuela Colombiana de Ingeniería in Bogotá, Colombia, enrique.estupiñan@escuelaing.edu.co

3 Dominik Makowski, MSc. Eng. Poznan University of Technology in Poznan, Poland, makowski.dominik@gmail.com

- **Video 2 (highway60.avi):** it has fast and predictable road lines motion, almost static background and few gray levels.
  - **Video 3 (container60.avi):** it has slow and predictable motion, static background and wide gray levels.
- This video selection helps to make a better test of the quality of the algorithms.

### 2.3 Selected Algorithms

#### 2.3.1 Averaging Algorithm

In this algorithm the output pixel is calculated as the average of the same position pixels of the previous  $I_{t-1}$  actual  $I_t$  and next frame  $I_{t+1}$  as follows (Ec.1):

$$P_{filtered}(x,y) = \frac{1}{3}(P_{(t-1)}(x,y) + P_{(t)}(x,y) + P_{(t+1)}(x,y)) \quad (1)$$

Processed pixels create new frame, which corresponds (in time) to  $I_t$  input frame (see Figure 1 and Figure 2). If more frames are included a better noise reduction is obtained, but the image is blurred [2].

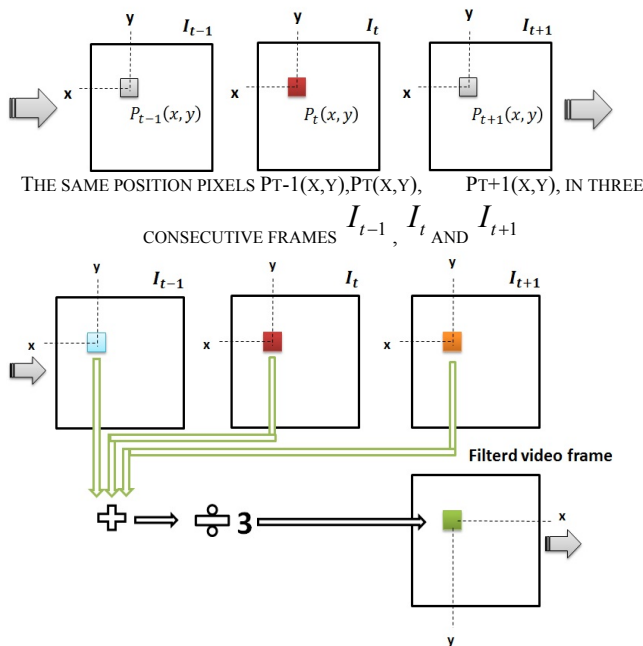


FIGURE. 1  
GRAPHIC EXPLANATION OF AVERAGING ALGORITHM

In Figure 2, there is an example of application of averaging filtering in a noisy video.



FIGURE. 2

NOISED VIDEO FRAME (LEFT) AND FILTERED ONE (RIGHT) USING AVERAGING ALGORITHM

#### 2.3.2 Best Neighborhood Algorithm

In this algorithm 3x3 masks are used, see Figure 3, in this case the pixels are ordered in an increasing fashion according to its difference with the central mask pixel  $P_t(x,y)$  of the actual frame  $I_t$  and they are stored in a vector. The output pixel value which corresponds to  $P_t(x,y)$  is calculated as follows (Ec.2) [3]:

$$V = [P_{(1)}, P_{(2)}, P_{(3)}, \dots, P_{(N)}] \quad (2)$$

$$|P_{(i)} - P_{(j)}| \leq |P_{(i)} - P_{(k)}|$$

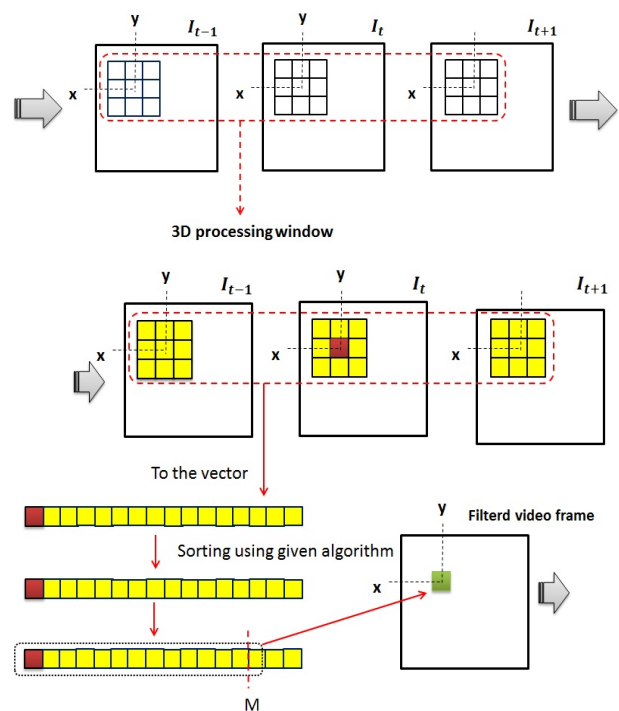
Where :  $i = 2 \dots N, j = i \dots N$  and  $P_{(i)} = P_{(i)}(x, y)$ ;

$$P_{filtered}(x, y) = \frac{1}{\gamma} \sum_{i=1}^N a_i P_{(i)} \quad \gamma = \sum_{i=1}^N a_i$$

In this implementation:

$$a_i = 1 \text{ for } i = 1, 2, \dots, M$$

$$a_i = 0 \text{ for } i > M$$



New pixel value is obtained by averaging pixels of indexes between 1 and M.

FIGURE. 3

GRAPHIC EXPLANATION OF BEST NEIGHBORHOOD ALGORITHM

In Figure 4, there is an example of application of best neighborhood filtering in a noisy video.



FIGURE. 4

NOISED VIDEO FRAME (LEFT) AND FILTERED ONE (RIGHT) USING BEST NEIGHBORHOOD ALGORITHM (M PARAMETER = 16)

### 2.3.3 Alpha Trimmed Algorithm

As in the previous algorithm 27 pixel values are used, corresponding to the three dimensional pixel masks, see Figure 5. Here the pixels are ordered in a descending fashion. The value of filtered pixel is calculated as follows (Ec.3) [2]:

$$U = [P_{(1)}, P_{(2)}, P_{(3)}, \dots, P_{(N)}] \quad (3)$$

$$P_{(1)} > P_{(2)} > \dots > P_{(N)}$$

$$P_{filtered}(x, y) = \frac{1}{\Theta} \sum_{i=floor(\alpha*N)+1}^{N-floor(\alpha*N)} P_{(i)}$$

$$\Theta = \mathcal{V}_1 - \mathcal{V}_{000} \quad (\alpha * \mathcal{V}_1)$$

If  $\alpha = 0.5$  the algorithm is a variant of median algorithm, if  $\alpha = 0$  it is an average algorithm.

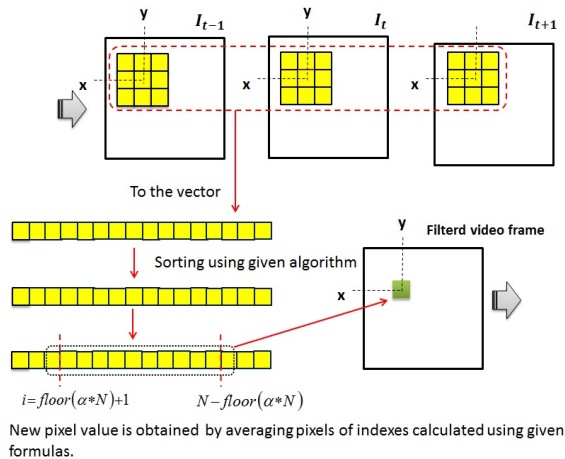


FIGURE. 5

GRAPHIC EXPLANATION OF ALPHA TRIMMED ALGORITHM

In Figure 6, there is an example of application of alpha trimmed filtering in a noisy video.



FIGURE. 6

NOISED VIDEO FRAME (LEFT) AND FILTERED ONE (RIGHT) USING ALPHA TRIMMED ALGORITHM (ALPHA PARAMETER = 0.3)

### 2.3.4 Median and Best Neighborhood Algorithm

This is a combination of the median and best neighborhood algorithms. First, it is determined if there is movement in the video or just noise in the frame; this is achieved by subtracting two consecutive frames thus eliminating constant pixels. The threshold is determined with the aid of the histogram value. In the case of movement in the video, the median algorithm is used otherwise best neighborhood is applied [4]. In Figure 7, there is an example of application of Median and Best Neighborhood algorithm in a noisy video.



FIGURE. 7

NOISED VIDEO FRAME (LEFT) AND FILTERED ONE (RIGHT) USING MEDIAN AND BEST NEIGHBORHOOD ALGORITHM

### 2.3.5 CCD and Averaging Algorithm

This algorithm uses luminance information to find dynamic frames in input video stream and dynamic areas in stationary frames. Using this knowledge, averaging filter is applied to stationary frames/areas. Pixels of dynamic frames/areas are passed to the output without filtering (see Figure 8) [5][6].

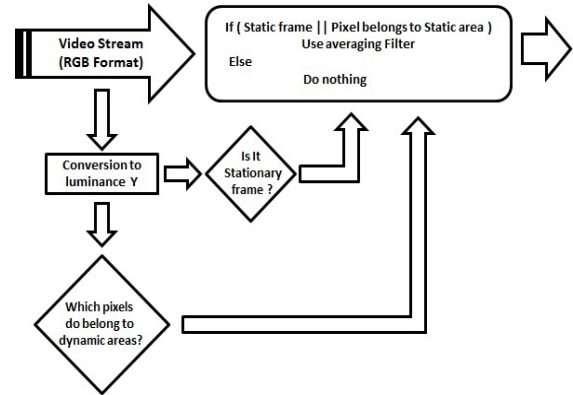


FIGURE. 8

GRAPHIC EXPLANATION OF CCD AND AVERAGING ALGORITHM

In Figure 9, there is an example of application of CCD and averaging algorithm in a noisy video.



FIGURE. 9

NOISED VIDEO FRAME (LEFT) AND FILTERED ONE (RIGHT) USING CCD AND AVERAGING ALGORITHM

## 2.4 Tutorial

### 2.4.1 Simulation Tutorials

These self-learning material cover: How to acquire webcam video, the guides to implement the five algorithms and how to use the Simulation Tool for the development board with CCS in order to run them on the DSP.

### 2.4.2 Hardware Tutorials

These include: Configuration of the CCS, Simulink project configuration, connections scheme and instructions to compile and download a Simulink block to the DSP board.

### 2.5 Performance Criteria

In order to measure the effectiveness of the algorithms, the Peak to Noise Ratio, PSNR is used. In this case only the capability to remove the noise is evaluated (*Ec.4*) [7].

$$PSNR = 10 \log_{10} \left( \frac{Max^2}{MSE} \right) \quad (4)$$

$$MSE = \frac{1}{m * n} \left( \sum_{x=1}^m \sum_{y=1}^n (P_r(x,y) - P_{r-1}(x,y))^2 \right)$$

Since the pixel is represented with 8 bits, the maximum pixel value *Max* is 255. The higher value in dB the better the outcome.

## 3. PROJECT RESULTS

For the performance evaluation, gaussian noise was added to the three sample videos. The added gaussian noise has a mean of 0 and a standard deviation of 10.

Quantitative results are presented in *Table I*, showing the comparison between processed and original video in its implementation on DSP TMS320DM642 hardware.

TABLE I  
QUANTITATIVE EVALUATION OF PROCESSING ALGORITHMS ON DSP  
HARDWARE IMPLEMENTATION

Algorithm	Video	PSNR (dB)
		Processed - Original
Averaging filtering	ice60.avi	-21,38
	highway60.avi	-16,37
	container60.avi	-16,9
Best neighborhood & median filtering	ice60.avi	-20,3
	highway60.avi	-16,2
	container60.avi	-21,71
Best neighborhood algorithm	ice60.avi	-20
	highway60.avi	-17,5
	container60.avi	-21,2
Alpha trimmed filtering	ice60.avi	-21,75
	highway60.avi	-17,81
	container60.avi	-22,72
CCD & averaging filtering	ice60.avi	-19,74
	highway60.avi	-15,96
	container60.avi	-17,07

It is observed on *Table I*, that alpha trimmed algorithm and averaging filtering algorithm have higher performance on ice60.avi video processing. On the other hand, for highway60.avi video, alpha trimmed filtering and best neighborhood were the winners. In the case of container60.avi video, the best processing algorithm was alpha trimmed filtering and in second place was best neighborhood with median filtering.

## 3. TEACHING USING GUIDE AND DEVELOPED ALGORITHM

The set of developed algorithms was introduced to students during one semester course of video processing (PDVD). During classes, teacher explained step-by-step theory connected to noise in video and noise filters.

Students, working in groups, had to implement applications that illustrated the theoretical knowledge. Certain time to develop application for every new algorithm was defined by teacher. During classes, students had opportunity to ask questions regarding the presented theory and implementation details intended for those practical activities.

In order to check quality of teaching, during course, some quizzes were made. Questions given to students were directly and indirectly connected to the introduced theoretical knowledge. The main idea was to check if this way of handling the classes allows students not only to reproduce knowledge (memorized), but as well to find out solutions for presented problems using techniques they have learnt.

Video processing techniques mentioned above were introduced during inter - semester course too. The number of hours was equal to the course described before. In this case the class was conducted in a different way, and a self-learning video processing guide was created with the following characteristics:

- Knowledge about video processing techniques usually provided by teacher.
- Exercises for students – usually in the form of applications to develop, which allow them to understand how particular techniques work, through writing programs and solving common problems.
- Questions connected to provide exercises.
- Exercises for students based on prepared and already working algorithms provided to students during course.
- Questions and tasks connected to given exercises and algorithms.

Is worth to mention that laboratory guides have been used for education in engineer as a useful tool where knowledge comes from a constructive approach [8].

An additional third course using developed algorithms and guide was introduced for students participated in conference “*Digital signal processing and processors: principles and applications*” (<http://www.ing.unal.edu.co/catedra/20/index.php/digital->

signal-processing-and-processors-principles-and-applications.html).

Has to be noticed that participants were students of many different pre-graduate programs and most of them have no experiences with video and even with image processing techniques. Course included four theoretical hours of image/video processing basics and more sophisticated algorithms. Video processing techniques introduced during courses described before, were included in this speech. Efficiency of teaching method using the guides and prepared algorithms was evaluated using results obtained from participants of lecture previously mentioned. Those students have four hours to make exercises and fulfill tasks described by provided video processing guide.

#### 4. TEACHING RESULTS

During first course, as mentioned before, students worked without guide and had to develop video processing algorithms by themselves using knowledge introduced by an instructor. In this way, it was possible to estimate average time needed by group of students to present a working application. It helps to find main common problems and difficulties presented by students:

- Understanding problems related to sophisticated techniques.
- Problems with formulas interpretation.
- Problems connected to programming language syntax and signal flow.
- Problems with developed algorithms efficiency.

Most of the lecture was dedicated to explain the noise reduction algorithms explained above.

As mentioned above, the second inter – semester course covered the same amount of hours as the first course. The evaluation was made by special prepared quizzes. Every group received grades depending on:

- Developed solution – if the problem was solved.
- If answer was connected exactly to given question.
- If presented application works properly.
- Efficiency of presented application.
- Abilities of students to use available features of programming environment.

Has to be noticed that students were divided into two kinds of groups:

- Groups using guides and developed applications.
- Groups that were allowed to use only provided applications.

Instructor introduced theoretical knowledge to every group with the same detail.

In the case of participants of “Digital signal processing and processors: principles and applications” conference, every group was allowed to use the video processing guide and also applications provided by teacher.

#### Obtained results:

- Using video processing guide combined with video processing applications reduce the time of the instructor without reducing the quality of learning. Should be mentioned that 32 hours of a regular semester course finally were compress into eight hours including four hours of theoretical lectures. Such time reduction can be achieved because students can focus on process of knowledge understanding. Process of learning based on application development, was replaced by results observation and deal with provided applications; is good to mention that the mentioned learning process was tested before with good results in other lectures like in [9].
- Comparing groups working with guide and applications with groups working only with applications highlight the difference in knowledge understanding. Making exercises, fulfilling tasks and reviewing once again knowledge introduced by teacher allows to students to understand better and easier the required video processing techniques.
- Provided applications are configured and prepared in the way which allows to use Matlab embedded code to compile them using C language and run on DM642 DSP processor. In this way students can experiment with DSP hardware and easily go from Windows applications development to DSP development.

#### REFERENCES

- [1] Quinnet R., Smarter Video Analysis Techniques mine more data. Electronic Design. Sept. 2009, Cover Feature
- [2] Bovik A., Handbook of image and video processing, Ed. Academic Press, 2000.
- [3] Zlokolica V., Philips W., Van De Ville D., Robust non-linear filtering for video processing, Ghent University, Dept. of Telecommunications and Information Processing (TELIN), IPI Sint-Pietersnieuwstraat 41, 9000 Gent, Belgica IEEE 2002.
- [4] Espinosa-Duró V., Faúndez-Zanuy M., Ortega J. A., “Face detection from a video camera image sequence”, Polytechnic University of Catalonia, Electronic and Automatic Department, 2004.
- [5] Chen T-H., Lin Z-H., Chen C-H., Kao C-L., A fast video noise reduction method by using object-based temporal filtering, Department of Electronic Engineering, Huper Laboratories Co., Tapei, Taiwan, R.O.C. IEEE 2002.
- [6] Hakan N., Bengt O., Youshi X., Spatio-Temporal Noise Reduction ASIC for Real-Time Video Processing, Department of Information Technology, ITE Mid Sweden University, IEEE.
- [7] Kanjanawanishkul K., Uyyanonvara B., Fast Adaptive Algorithm for Time-Critical Color Quantization Application, Department of Information Technology Sirindhorn International Institute of Technology (SIIT) Thailand, pag. 811, DICTA 2003.
- [8] Montes G. Willer, “Prácticas de laboratorio en ingeniería: Una estrategia efectiva de aprendizaje”, Revistas Notas Universitaria, ISSN 1797-1997. En: Escritos para la docencia #4, CEDIP No.11, Ibagué, December 2004.
- [9] Xiao-Hua Li, Jian-Hua Zhao, Bao-Hong Li, “A Methodology for evaluation college student comprehensive quality”, Proceedings of the Fifth International Conference on Machine Learning and Cybernetics, Dalian, 13-16 August 2006.