Applied Soft Computing 14 (2014) 338–346

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Fuzzy logic-based embedded system for video de-interlacing

Piedad Brox[a,*], Iluminada Baturone[a,b], Santiago Sánchez-Solano[a]

[a] Instituto de Microelectrónica de Sevilla (IMSE-CNM-CSIC), Américo Vespucio s/n, 41092 Seville, Spain
[b] Departamento de Electrónica y Electromagnetismo, Universidad de Sevilla, Av. Reina Mercedes s/n, 41012 Seville, Spain

ARTICLE INFO

Article history:
Received 30 June 2012
Received in revised form
18 December 2012
Accepted 7 January 2013
Available online 1 February 2013

Keywords:
Embedded system
Video de-interlacing
Fuzzy system
IP core

ABSTRACT

Video de-interlacing algorithms perform a crucial task in video processing. Despite these algorithms are developed using software implementations, their implementations in hardware are required to achieve real-time operation. This paper describes the development of an embedded system for video de-interlacing. The algorithm for video de-interlacing uses three fuzzy logic-based systems to tackle three relevant features in video sequences: motion, edges, and picture repetition. The proposed strategy implements the algorithm as a hardware IP core on a FPGA-based embedded system. The paper details the proposed architecture and the design methodology to develop it. The resulting embedded system is verified on a FPGA development board and it is able to de-interlace in real-time.

1. Introduction

Video de-interlacing is a key task in digital video processing. Some current digital transmission standards use interlaced scan format to halve video bandwidth. However, modern display devices require a progressive scan. Therefore, algorithms to interpolate the missing lines during the transmission have to be implemented at the receiver side. These kind of algorithms are called de-interlacing since they perform the reverse operation of interlacing [1].

Digital video processing chain usually includes the four stages shown in the block diagram of Fig. 1:

- Reception: video TV is received by a TV decoder, which is in charge of decoding the analog or digital signal.
- Removal of artifacts: analog signal suffers white Gaussian noise whereas digital signal is affected by video compression artifacts, which introduces two kinds of artifacts: 'block' and 'mosquito' noise.
- Conversion of resolution: signal is firstly converted from interlaced to progressive. After performing de-interlacing, an algorithm for down or up scaling is applied accordingly to the format required by the device.
- Picture improvement: the quality of video signal is enhanced by applying several picture enhancement algorithms such as color improvement, sharpness or edge enhancement, and improvements of contrast, details and textures.

Despite video processing algorithms are developed using software implementations; their implementations in hardware are required to achieve real-time operation. Although primitive consumer equipments had several ASICs to perform each of these four tasks, the current high-end products in the market contain a highly integrated chip to perform the last three tasks and even all the tasks in Fig. 1. Video processing chips are included in numerous consumer devices, such as Liquid Crystal Display (LCD) TVs, plasma TVs, Audio–Video (AV) receivers, DVD players, High Definition (HD)-DVDs, Blu-ray players/recorders, and projectors. Independently of the video source (during the last years multiple video sources are proliferating), these chips normally perform intelligent and costly de-interlacing methods to obtain a high quality output progressive signal. Concerning video de-interlacing, several choices for hardware implementations are currently available in the market, such as Application-Specific Integrated Circuits (ASICs) [2–9], programmable solutions on Digital Signal Processing (DSPs) [10–14] and/or Field Programmable Gate Arrays (FPGAs) [15–18], and Intellectual Property (IP) cores [19–23] that can be used as building blocks within ASICs or FPGA designs. Recently, application specific instruction-set processors (ASIPs) for high speed computation of intra-field de-interlacing have been also proposed [24]. Each of these alternatives offers advantages and disadvantages regarding high performance, flexibility and easy upgrade, and low development cost.

In commercial equipments, ASICs offer the most efficient solution justified by the huge demand of video products. In the other

* Corresponding author. Tel.: +34 95446666; fax: +34 95446600.
E-mail address: brox@imse-cnm.csic.es (P. Brox).

1568-4946/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2013.01.015

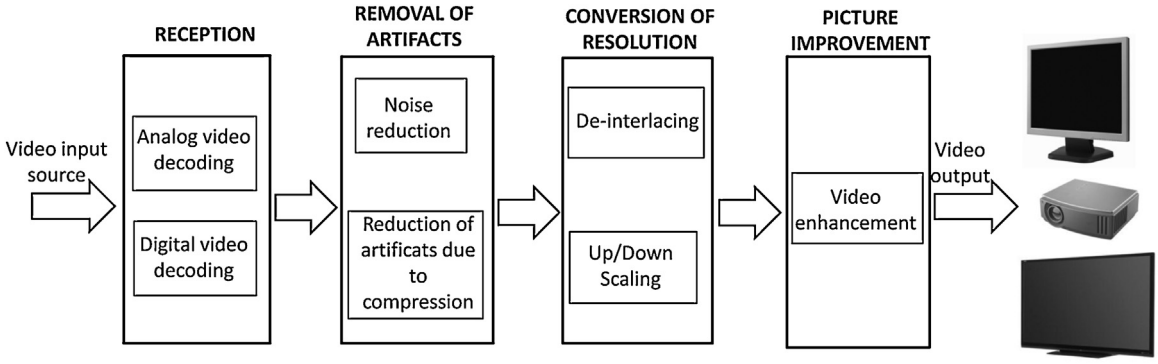

Fig. 1. The four typical stages of digital video processing.

side, FPGAs are a good solution when there is a low volume of consumer products or in the case that the aim is to develop a prototype. Furthermore, FPGAs address with the requirement of flexibility and easiness to upgrade. Taking into account these considerations, FPGA implementation is the option chosen herein to implement the de-interlacing algorithm on an embedded system.

There are many proposals of fuzzy logic-based embedded systems for control applications [25,26]. Recent advances in de-interlacing algorithms propose the inclusion of soft computing techniques to increase the picture quality [27–31]. An embedded system that implements the algorithm of [27] in real-time is proposed in this work. To the best of our knowledge this is the first hardware implementation of a fuzzy logic-based video de-interlacing algorithm. This paper details the implementation strategy that consists of an IP core on a FPGA-based embedded system. The paper is organized as follows. Section 2 summarizes a description of the algorithm and the proposed architecture to develop its hardware implementation. Section 3 explains the design methodology to build the IP core for video de-interlacing. The development of the embedded system and its validation on a development board from Xilinx is detailed in Section 4. Finally, the conclusions of this work are expounded in Section 5.

2. Architecture of fuzzy IP core for video de-interlacing

The algorithm for video de-interlacing is the result of combining three fuzzy logic-based systems, each of them tackling a relevant feature: motion, edges, and possible repetition of areas in fields. The edge-adaptive system is called spatial interpolator since it performs a non-linear interpolation among pixels in the spatial neighborhood. The system that is capable of detecting repeated areas in the fields is called temporal interpolator since it interpolates pixels in the temporal neighborhood. The third system combines the outputs of the spatial and temporal interpolators according to a motion measurement in the current pixel. A block diagram of the complete system is shown in Fig. 2.

The three fuzzy systems are simple since they contain a low number of inputs and a low number of rules in their rulebases.

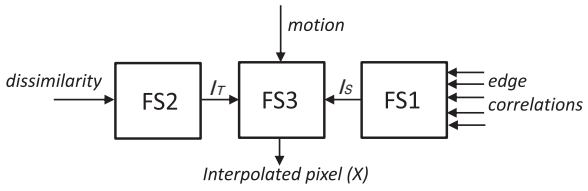

Fig. 2. Block diagram of the video de-interlacing algorithm that employs the three fuzzy logic-based systems.

The spatial interpolator (denoted as FS1 in Fig. 2) is an edge-adaptive algorithm that uses five potential edge directions and six fuzzy rules to adapt the interpolation to the presence of edges. The second interpolator (FS2) is a fuzzy area-repetition-dependent temporal interpolator that uses a simple convolution to measure the dissimilarity between consecutive fields. It employs two simple fuzzy rules to adapt interpolation to repetition of areas in fields. Despite its simplicity, it reduces considerably annoying artifacts such as feathering. The feature of motion is considered by a fuzzy motion-adaptive interpolator (FS3) that uses a simple convolution to measure the motion at each pixel. This third system evaluates how this motion measurement should influence on the interpolation decisions.

The algorithm for video de-interlacing employs an off-line tuning process to obtain the values of the parameters in the fuzzy systems as detailed in [27]. The tuning stage has been successfully performed by using a supervised learning algorithm that minimizes the mean square error between a set of data corresponding to progressive and de-interlaced results of different standard sequences.

A detailed description of the complete system and its comparison (in terms of PSNR and visual inspection) with other state-of-the-art de-interlacers are presented in [27]. The algorithm is able to improve the results obtained by several Motion-Compensated (MC) algorithms in areas of the images with small and large motion, with clear and unclear edges, and with film and video material mixed. Concerning embedded system solutions, the hardware implementation of this algorithm is advantageous over the considered MC algorithms, as depicted in Table 1. The number of Primitive Operations (OPs) required is quite low. The number of field memories is three since it is necessary to store the values of the interpolated pixels in the previous field (as shown in Fig. 3).

Table 1
Analysis of storage resources and POs of several state-of-the-art de-interlacers.

De-interlacing algorithm	No. of field memories	No. of primitive operations (POs)
MC field insertion [32]	1	529
MC VT filtering [32]	1	536
MC TBP [32]	2	535
MC TR [32]	2	529
MC AR [32]	2	544
GST [32]	1	543
Robust GST [32]	1	555
GST-2D [32]	2	559
Robust GST-2D [32]	2	571
Algorithm in [27]	3	153