Implementation of Deinterlacing on an Embedded Processor

Krishnaraj Bhat and Dr. U. C. Niranjan, Senior Member, IEEE.

***Abstract*--As the video industry transitions from analog to digital, video processing equipments will also need transitions from analog to digital. The analog television standards like NTSC, PAL, etc are based on interlaced formats. Current television standards like HDTV, LCD TV etc need video in the progressive format. So the demand for video processing products with high quality deinterlacing techniques increasing.**

There are many methods to perform deinterlacing techniques. Each method differs with other in quality variations and computational complexity. Video deinterlacing relies heavily on the correctness of motion. In this work we used per-pixel motion adaptive deinterlacing[2], in which we detected the motion of objects between two frames in pixel level and depending on this motion values we selected either spatial domain deinterlacing or temporal domain deinterlacing. We implemented this technique on the Stretch's Software Configurable Processor along with Video Development Platform(VDP) board.

***Index Terms*-- Interlacing, ISEF, Progressive frame, Motion Adaptive Deinterlacing, Software Configurable Processor, Stretch S5530, Video Development Platform.**

I. Introduction

The interlaced format has been used widely on the TV industry and consumer cam coders. The merit of interlaced format lies in that the refreshing rate is double without increasing the bandwidth. However when displaying an interlaced video on a progressive device may cause visual artifacts like edge flickering, line crawling, and interline flickering. To overcome these artifacts deinterlacing technique has been widely used.

Deinterlacing is the process of converting an interlaced video (a sequence of fields) into a non-interlaced format or progressive video. Now a days display devices are developed with progressive scanning instead of interlaced scanning. So there is a need of conversion from interlaced video to progressive video. A number of deinterlacing techniques are available. These methods can be generally classified into two categories i.e. spatial domain methods and temporal domain methods. Spatial domain method is one in which only one field is involved in forming a progressive frame. In temporal domain method, multiple fields are used to get a resulting progressive frame. By using temporal domain method one can get better result than spatial domain methods. This is because the interlaced fields are captured at different time instants and hence there is a motion of objects between different fields. As a result temporal methods, which utilize the motion information between neighboring fields, often offer a significantly improvement in progressive video at the cost of more computational complexity. To overcome this computational complexity, in our work we used Stretch's Software Configurable Processor.

The paper is organized as follows: in the next section the details of materials used for this work is been explained. In the section 3 the implementation details are given. Section 4 is about results of this work. In section 5 the references used for this work are given.

II. MATERIALS USED

A. Stretch Software Configurable Processor:

A Software Configurable Processor (SCP) is one in which we can modify a part of the hardware to suite the needs of the application. The Stretch Software-Configurable Processor is targeted at a wide range of embedded applications with demanding cost-performance requirements. Unlike traditional RISC processor architectures that have fixed instruction sets, we can easily extend SCP's instruction set to address the specific needs of the end application. The main goal of the SCP architecture are to achieve unparalleled processor cost/performance in areas where SoC and ASIC design are economically infeasible and to ensure effective optimization of application software using modern compiler technologies. The advantages of SCP are

1. SCP is flexible for designing multiple applications.
2. Performance improvement in compute-intensive applications.
3. The processor can be programmed using C/C++.
4. Increased efficiency to address high-volume applications.
5. Lowered cost.

U. C. Niranjan is with the Manipal Dot Net Pvt. Ltd., Manipal-576104, Karnataka, India. (email: ucniranjan@yahoo.com).

Krishnaraj Bhat is with the Manipal Dot Net Pvt. Ltd., Manipal-576104, Karnataka, India. (email: krishnarajbhat@rediffmail.com).

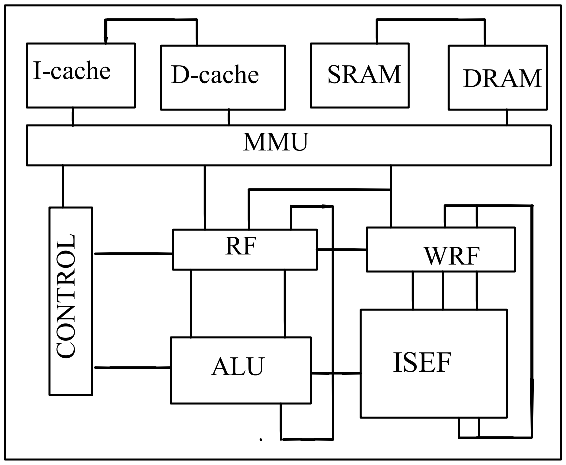

Fig 1. Block Diagram of SCP

The main components in the SCP architecture are

1. Tensilica Xtensa ISA : This is the fixed base instruction set.
2. One or more Instruction Set Extension Fabric (ISEF) units : These units can be customized to execute Extension Instructions.
3. A 128-bit Wide Register (WR) file : This register file is used for holding data.
4. A set of load - store instructions : These instructions move data efficiently between the processor data memory and the WR registers.
5. A set of one-bit State Registers (SR) : These registers can be easily grouped to form Extension Registers (ER) of any size. The use of one or more ERs is implicitly defined in the Extension Instructions.

B. *Stretch's VDP board:*

In our work we used Stretch's Video Development Platform(VDP) to get input as the interlaced image format and to store it in memory. VDP provides a development environment for video applications.

C. *Stretch Basic Input Output Systems(SBIOS):*

A set of basic input and output streams are available with the Stretch. It includes run-time libraries for C and C++, reset, exception and interrupt handlers, drivers for the peripherals available on the S5530.

III. IMPLEMENTATION DETAILS

A. *Practical Considerations:*

The S5530 processor has 128 bit wide special registers named as wide registers. These registers are used to send or receive data to or from ISEF. The processor has four streaming functions; three of these are used to stream the data into the ISEF and one is used to send data out of the ISEF. The ISEF allows all the data types except float values. It also allows signed and unsigned values with variable number of bits.

B. *Image Dimensions:*

The input image, we used is 720X240 interlaced RGB format. The output progressive frame is in 720X480 RGB format.

C. *Motion Adaptive Deinterlacing:*

The VDP board gets the RGB interlaced image from the real time system and stores it in the main memory. This image data must be send to ISEF for fast processing. But loading the image data from main memory to ISEF takes much larger time. To increase the data transfer, data must be send to on chip RAM. For data transfer we used DMA channels. The main purpose of using the DMA channels is to let the processor to do the processing work, instead of copying data. As the copying of image data goes on, the processor can be used to load the data to ISEF for processing. By doing this we achieve the parallelism between the data copying and loading of data to ISEF. Since, on chip RAM and ISEF both are of limited size, we need to do this process of data transfer and loading of data to ISEF in steps. The process of data transfer and data load is shown by using the following block diagram.

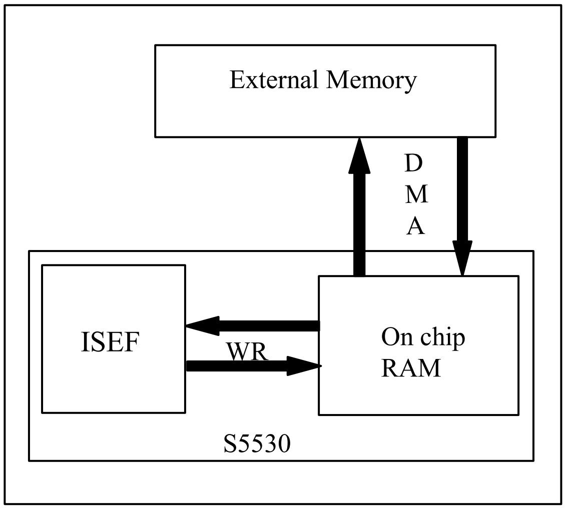

Fig. 2. Block Diagram of DMA Data Transfer

In this work we used three successive field informations to create a progressive frame. By using the three input streams, we are loading sixteen pixels of each of the three field informations to the ISEF at a time. By using these field informations, the motion of objects is calculated using per pixel difference. Depending on this difference we are using either spatial domain deinterlacing [2], [3] or temporal domain deinterlacing to produce a progressive frame. The process of per pixel[4] motion detection and selecting the algorithm can be represented as shown below.

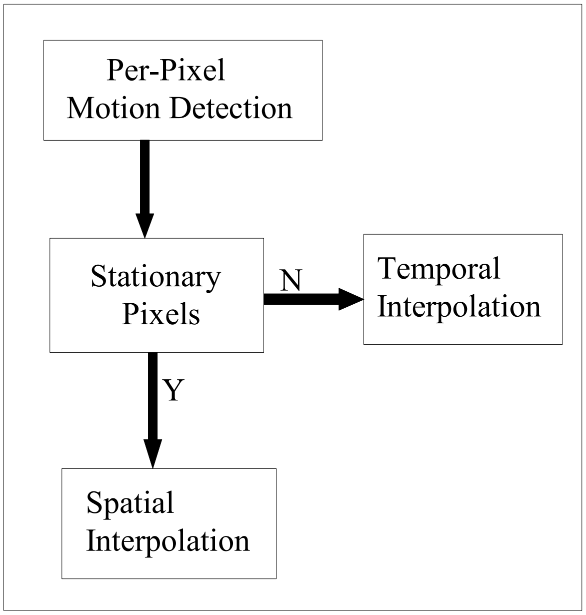

Fig. 3. Block Diagram of Motion Adaptive Deinterlacing

In our work we are creating two progressive frames using four interlaced fields. Out of four progressive frames first three are used for creating first progressive frame and next three are used for creating second progressive frame.

IV. RESULTS

We done this motion adaptive deinterlacing for real time applications. We generated 30 Progressive frames per second by using 60 interlaced fields per second. Per pixel motion detection used in this work judge the motion of objects within the fields more accurately and generate progressive frames. In this work we mainly focused on the luminance values of interlaced image. We did not considered the chroma values as it does not change by the motion of objects.

V. ACKNOWLEDGEMENT

We are grateful to Dr. Narasimha B. Bhat for providing us with opportunity to carry out this work at Manipal Dot Net Pvt. Ltd., Manipal. We also thank Stretch Inc., CA, USA.

VI. REFERENCES

[1] De-interlacing: A Key Technology for Scan Rate Conversion Volume 9 by E.B. Bellers and G.de Haan.

[2] Kefei Ouyang, Goubin Shen, Shipeng Li, Ming Gu, “Advanced Motion Search and Adaption Techniques for Deinterlacing”.

[3] Charles Poynton, “Digital Video and HDTV Algorithms and Interfaces, 2003, page number 437-443.

[4] http://www.hometheaterhifi.com/volume_7_4/dvd-benchmark-part-5-progressive-10-2000.html.

[5] http://100fps.com/