Efficient DSP Implementation of Fractional-Pixel Interpolation for AVS*

Zhigang Yang, Shuhong Jiao, and Lutao Liu

College of Information and Communication, Harbin Engineering University, Harbin 150001, China
{zgyang,jiaoshuhong,liulutao}@hrbeu.edu.cn

Abstract. Fractional-pixel motion compensation can greatly improve the compressing efficiency in video coding, while quarter-pixel interpolation also leads to a significant increment in computational complexity. This paper presents some techniques for efficient implementation of quarter-pixel interpolation in AVS-P2 on a fix-point digital signal processor (DSP). Firstly, the whole interpolation process is divided into five sub-processes from the DSP-oriented viewpoint. Then highly parallel software pipelines are designed for each sub-process with elaborately balancing the resources on each side of the CPU data path. A task-level optimization strategy is also applied to arrange the software pipelines. Finally, the simulated results demonstrate that the execution time of interpolation can be greatly reduced by using this specific design.

Keywords: Interpolation, digital signal processors, AVS, video coding.

1 Introduction

The transform and motion compensation hybrid coding framework is commonly adopted in state-of-the-art video coding standards such as H.264 [1] and AVS [2]. The accuracy of motion compensation is a key technique in this coding scheme. As a real motion has arbitrary precision, enabling motion vectors to have fractional-pixel resolution can greatly improve the prediction accuracy. Half-pixel motion vector accuracy has already been used in prior standards like MPEG-2. Quarter-pixel interpolation is first found in MPEG-4 (Advanced Simple Profile) and then improved in the development process of modern standards H.264 and AVS. In H.264, a 6-tap filter is used for half-pixel interpolation. In AVS-P2, 4-tap filters are used for both half- and quarter-pixel interpolation.

Motion compensation with fractional-pixel accuracy can progressively reduce spatial redundancy as the level of fractional-pixel accuracy increases. Half-pixel motion vector resolution can provide a coding gain of about 2.7dB when compared to

* This work was supported in part by the National Natural Science Foundation of China under Grant Number 61201238 and 61201410, and the Fundamental Research Funds for the Central Universities under Grant Number HEUCFR1017.

W. Lin et al. (Eds.): PCM 2012, LNCS 7674, pp. 587–596, 2012.

integer-pixel resolution, and an additional coding gain of about 0.9dB can be obtained with quarter-pixel motion vector resolution [3]. Some adaptive interpolation techniques also studied to further improve the prediction efficiency [4][5].

However, fractional-pixel interpolation also leads to a significant increase in computational complexity. Some VLSI architectures have been presented to solve the problem. For example, R. Wang *et al.* proposed a parallel and pipeline architecture for the sub-pixel interpolation filter in H.264/AVC [6], and L. Lu. *et al.* presented a reconfigurable sub-pixel interpolation architecture for motion estimation [7]. While on DSP platform, the optimizing methods adopted in VLSI architectures are usually inefficient for DSP pipelines and always lead to worse DSP performance.

The purpose of this paper is to provide a DSP solution to accelerate the interpolation process for AVS-P2, mainly at the encoder side. The rest of this paper is organized as follows. In Section 2, the luma sample interpolation in AVS-P2 is briefly introduced. In Section 3, the luma interpolation process is divided into five software pipelines, and then instruction-level pipelines on the DSP platform are designed in detail. Software pipelines are also arranged from a task-level viewpoint. In Section 4, simulated results are demonstrated to show the effectiveness of the specific design. Finally Section 5 concludes the paper.

2 Luma Sample Interpolation

In Fig. 1, the positions labeled with upper-case letters represent luma samples at full sample locations and the positions labeled with lower-case letters represent luma samples at fractional sample locations. Given the luma samples 'A' to 'L' at full sample locations, the luma samples 'a' to 's' at fractional sample positions are derived by the following rules.

- The luma prediction value at half sample position labeled as 'b' shall be derived by applying a 4-tap filter with tap values (-1, 5, 5, -1) in the horizontal direction.
- The luma prediction values at half sample positions labeled as 'h' and 'j' shall be derived by applying a 4-tap filter with tap values (-1, 5, 5, -1) in the vertical direction.
- The luma prediction values at quarter sample positions labeled as 'a', 'c', 'i', and 'k', shall be derived by applying a 4-tap filter with tap values (1, 7, 7, 1) in the horizontal direction..
- The luma prediction values at quarter sample positions labeled as 'd', 'f', 'n', and 'q', shall be derived by applying a 4-tap filter with tap values (1, 7, 7, 1) in the vertical direction.
- The luma prediction values at quarter sample positions labeled as 'e', 'g', 'p', and 'r' shall be derived by averaging samples at full and half sample positions in the cross direction.

The detailed process for each fractional position is described in the AVS-P2 standard [2].