Performance-Efficient Architecture for Free-Viewpoint 3DTV Receiver

E. Bondarev[1], S. Zinger[1], and P. H. N. de With[1,2], *Fellow*, *IEEE*

[1]Eindhoven University of Technology / [2]CycloMedia Technology, The Netherlands

Abstract -- **This paper presents algorithmic and architectural solutions for a free-viewpoint 3DTV receiver system. We describe our rendering algorithm and evaluate performance-related challenges in mapping of the algorithm on a receiver board of which the architecture is outlined. It is found that the required processing load exceeds the provisioning of dual Virtex5 FPGAs. We develop several mapping optimizations to fit the rendering algorithm into a platform.**

I. INTRODUCTION

Three-dimensional television (3DTV) at high resolution is likely to be the succeeding step after the broad acceptance of HDTV. The introduction of several formats such as depth/disparity maps [1] along with texture images, or stereo video signals enables rendering of two different views for right and left eyes of a user. As a result, the user obtains 3D experience in viewing. *Free-viewpoint (FVP)* technology is even more advanced, as it enables a user to view scenes from different angles. The user chooses the position from which he would like to watch a video. We assume that we have several input video streams, captured by multi-view cameras. Each video stream consists of texture and depth images. In this case, we perform *Depth Image Based Rendering (DIBR)*.

In the iGlance project [2], we aim at combining the above-mentioned technologies, by developing a real-time FVP 3DTV receiver. In this paper, we analyze the complexity of the rendering algorithm and explore its implementation with FPGAs. The analysis reveals severe bottlenecks, for which we identify possible solutions.

II. INTERPOLATION AND RENDERING ALGORITHM

Our research on 3D rendering further extends the current research on DIBR [4, 5]. A novel aspect is that not only reference camera views, but also the rendered views should be accompanied by depth information. Due to the real-time requirements of the algorithm, we keep its complexity limited while maintaining an acceptable rendering quality.

Generally, DIBR algorithms are based on 3D image warping, which enables the synthesis of the virtual view from a reference texture view and a corresponding depth map. This warping specifies the computation for one pixel only, so that it has to be performed for the entire image. In multi-view video, the information for warping is taken from the two surrounding camera views. Typically, two warped images are blended to create a synthetic view at the new position. Such an approach requires several post-filtering algorithms for improving the visual quality of the results.

The latest research has shown that a better rendering quality can be obtained when we first create a *depth map* for a new image [4, 5]. Using this depth map, we perform an *inverse mapping* in order to obtain texture values for the new image. In this case, we create two main stages for the novel rendering algorithm: depth map creation and texture creation.

Depth map creation consists of the following steps: (a) combining warped depth maps from the closest left and right cameras; (b) median filtering of the resulting depth map; and (c) processing of occlusions.

To create the texture, we proceed as following: (a) warping textures for the new view; (b) blending the obtained textures; and (c) filling the occlusions.

Our rendering involves median filtering, which has important advantages over other filters. Median filters have an anisotropic nature - they preserve edges which are important for depth maps, and the only parameter is window size. The quality of rendering depends not only on the quality of the depth map. The occlusion filling process is error prone. To reduce these errors, we use the depth information to fill in the disoccluded regions of the depth maps and textures. This avoids blurring and ensures that the disocclusions are filled with correct background data.

III. PERFORMANCE CHALLENGES

An FVP 3DTV receiver should provide the following real-time functionality: (a) decoding of multiple streams, each containing a pair of Texture and Depth Map (T+DM) for one view, and (b) interpolation and rendering of an intermediate view. We have found the following *performance challenges*.

A. Processing of High-Definition video stream

The 1080p@60Hz HD format imposes high load on the internal communication bandwidth. An internal bus should transfer at least 1.6 Gbit/s from the decoder to the rendering unit. Existing HW-based decoding solutions, e.g. Systems-on-Chip (SoC), and bus infrastructures (HDMI, DVI, Multiplex) are able to satisfy this requirement.

B. Processing of multiple pairs of video streams

Realization of the *3DTV* concepts requires not a single video stream, but *pairs* of streams (T+DM or L+R textures). This doubles performance requirements for the decoder, memory and bus infrastructure. Moreover, the *FVP* concept requires processing of *multiple pairs* of streams, one pair per view. These two aspects boost the internal bus-load requirements at minimum to 6.4 Gbit/s.

C. Interpolation of artificial views

The generation of intermediate views is a computationally

expensive process. It poses high requirements on processing units. Our profiling results on the interpolation and rendering algorithm [3] are shown in the Table I.

TABLE I: PERFORMANCE REQUIREMENTS FOR INTERPOLATION ALGORITHM

Stream Type	Internal bus bandwidth, Gbit/s	Processor operations, GOP/s
HD Ready: 720p@120Hz, 24b	10.668	302
Full HD: 1080p@120Hz, 24b	23.887	675
Downscaled format:1080p@30Hz	5.971	168

None of the current FPGA or GPU processors are able to handle 675 GOP/s. The currently available Virtex5 FPGAs, which are priced within the cost limits for a set-top box receiver, can handle only 20-50 GOP/s.

The same holds for the bandwidth requirement (23.9 Gbit/s). Current bus technologies, such as HDMI 1.3 and DVI, provide a maximum bandwidth of 8.16 Gbit/s, while PCIe 3.0 just exceeds 8 Gbit/s.

Concluding, the provisioning of currently available hardware IP-blocks is not sufficient for straightforward architectural solutions. Therefore, careful architecting and trade-off analysis are required to build a FVP 3DTV system.

IV. ARCHITECTURE DISCUSSION

We have identified and validated a number of architectural decisions to make the receiver functionality feasible for the performance requirements. Figure 1 depicts a principal architecture of the receiver board. As an input, the board receives multiple T+DM streams captured by real cameras. The output to the 3D Display is a T+DM stream of a 3D scene from the user-selected viewpoint.

The MPEG-2 decoding algorithm is deployed on two synchronized ST40 SoCs. Each SoC is powerful enough to decode one T+DM stream in HD format. The decoded streams are transmitted via two HDMI interfaces for FVP interpolation and rendering. The algorithm is mapped onto a synchronized pair of Virtex5 FPGA processing units. The interpolated video stream containing the texture and depth map is sent to a 3D Display via the HDMI interface.

In the described architecture, the high processing power requirements for decoding dual texture and depth map HD streams is successfully handled by deployment of two SoCs synchronized within 10 ms.

The internal bus-load is handled by two HDMI connections, going from the SoCs to the FPGAs. Even these two HDMIs cannot handle two T+DM streams in the 1080p@120Hz format. Therefore, in the iGlance project, we downscaled the requirements to the 1080p@30Hz format.
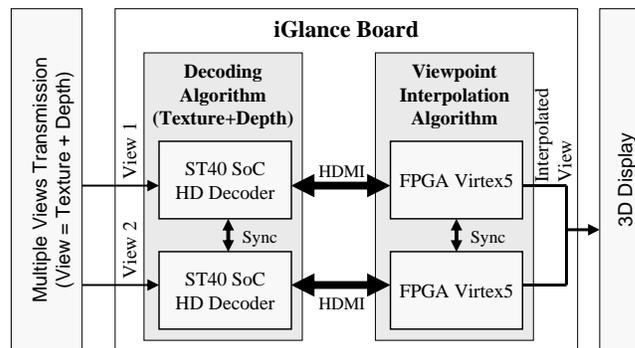

Figure 1. Principal architecture of iGlance board.

Finally, we address the extremely high processing requirements of the interpolation algorithm by mapping it on two full-fledged dual-core Virtex5 FPGAs. Note that even the processing capacity of these two FPGAs is not sufficient for these downscaled streams (see Table I). An efficient algorithm-to-hardware mapping is required, to make the architecture feasible. Therefore, we took the following mapping optimizations: (a) map the expensive median filtering algorithm (requiring 70 GOP/s) onto a separate dedicated hardware unit, thereby leaving only the remaining 98 GOP/s to FPGAs, (b) fully pipeline the processing on FPGAs, and (c) vectorize the streaming data for computations. With these optimizations, the dual-core FPGA tandem and dedicated hardware units are able to handle 168 GOP/s.

V. CONCLUSIONS

In this paper, we have presented our algorithmic and architectural advances in an FVP 3DTV receiver. We have described our FVP rendering algorithm and evaluated a performance feasibility of an FVP 3DTV receiver. The feasibility study shows that the current hardware provisioning is insufficient for straightforward architectural solutions. Our principal architecture combines the following aspects: (a) duplication and synchronization of hardware units, (b) format downscaling and (c) pipelining and efficient mapping of algorithmic blocks onto FPGA units, to match the required functionality with the real-time constraints. The principal architecture is used for development of a receiver prototype within the iGlance project.

REFERENCES

[1] Tzovaras, D., et. al, "Disparity field and depth map coding for multiview 3D image generation", SP: IC (11), No. 3, January 1998, pp. 205-230.

[2] Official website of iGlance project [www.iglance.org]

[3] P. H. N. de With, S. Zinger, "Free-viewpoint rendering algorithm for 3D TV", Proc. of the 2nd Workshop of Advances in Communication, Germany, May 2009

[4] Y. Morvan, "Acquisition, compression and rendering of depth and texture for multi-view video", Ph.D. thesis, Eindhoven University of Technology, 2009.

[5] Y. Mori, N. Fukushima, T. Fujii, and M. Tanimoto, "View generation with 3D warping using depth information for FTV," in Proc. of 3DTV-Conference, pp. 229-232, 2008.