EMBEDDED STEREOSCOPIC 3D CAMERA SYSTEM FOR MOBILE PLATFORMS

Alberto Aguirre, Aziz Umit Batur, Gregory Hewes, Ibrahim Pekkucuksen, Narasimhan Venkatraman, Fred Ware, Buyue Zhang
Texas Instruments Incorporated, 12500 TI Blvd, Dallas TX 75243

ABSTRACT

In this paper, we describe an embedded programmable stereoscopic 3D camera system for mobile platforms. With the proliferation of stereoscopic televisions, computer monitors, and even auto-stereoscopic LCDs, as well as a growing inventory of stereoscopic games and movies, there is a growing demand for stereoscopic 3D content creation. We present each of the key components that enable the stereoscopic 3D video capture and playback on the embedded platform, including imaging, codec, graphics, and display subsystems. We describe several unique and new 3D image processing algorithms and explain how they are integrated into the stereo 3D system.

Index Terms – stereoscopic, 3D, imaging, video, camera

1. INTRODUCTION

With the proliferation of stereoscopic televisions, computer monitors, and even auto-stereoscopic LCDs, there is a growing demand for stereoscopic 3D content creation. It is a natural trend for camera phones to include stereo cameras in order to enable consumers to capture their own stereoscopic 3D videos and images. The mobile phone is the ideal end-to-end 3D device, supporting stereoscopic 3D capture, playback and gaming. With built-in connectivity, consumers can easily share their videos and photos. A mobile phone is a personal device with a single user, so when it is coupled with an auto-stereoscopic display, consumers can enjoy a compelling 3D experience locally on their phone without the need for special glasses. We also describe several new algorithms that are specific to a stereo 3D system and explain how these new algorithms are incorporated into a stereo 3D video processing flow.

2. STEREOSCOPIC SYSTEM

An overview of our stereoscopic system is shown in Figure 1. It includes 4 major subsystems: imaging, coding, graphics, and display. The imaging subsystem receives a pair of synchronized left/right frames from the sensors, and applies stereoscopic 3A (Auto Exposure, Auto Focus, and Auto White Balance), and stereoscopic 3D correction and quality enhancement algorithms. The codec subsystem encodes/decodes high definition stereoscopic video and high resolution images. The display subsystem renders the left and right frames in the format required by the display. Example display types include interlaced or side-by-side auto-stereoscopic panel, HDMIv1.4 3D formats, anaglyph, and even single-view. Each of these subsystems will be covered in more detail in the following sections.

Our system is implemented on a Texas Instruments OMAP4430 applications processor, which is well-suited for implementing stereoscopic systems with its dual bus architecture, high performance image processing, high definition video coding, and powerful display subsystem [1]. The dual–bus architecture allows for parallel SDRAM memories to be accessed simultaneously by the dual core processor via parallel dual-lane memory bus.

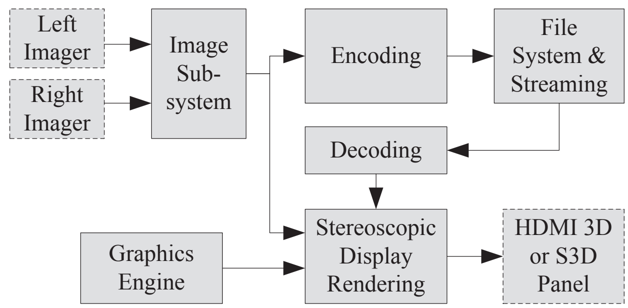

Figure 1 - Overall view of the stereoscopic system

3. IMAGING SUBSYSTEM

The imaging subsystem receives pairs of raw Bayer frames from the camera sensor, processes them through the stereoscopic image pipe, and outputs them to the encoder and display subsystems. The image pipe includes both conventional 2D image processing algorithms, such as CFA interpolation, as well as 3D specific algorithms, such as stereoscopic 3A, stereo auto convergence, and stereo misalignment correction. The purpose of the new 3D specific algorithms is to enhance the left and right frames such that optimal 3D viewing experience can be achieved on the target display. Details on the image pipe and the stereoscopic 3D algorithms are given in the subsections below.

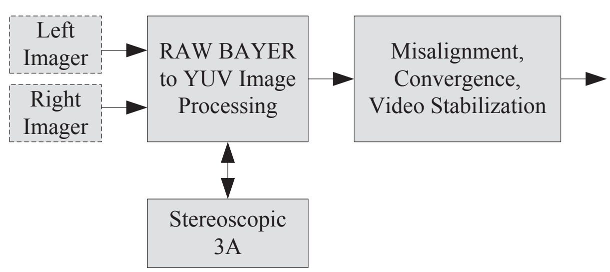

Figure 2 - Overall view of stereoscopic imaging subsystem

3.1 2D Image Pipe

The 2D image pipe receives a left/right pair of images from the imaging sensor. These RAW Bayer images are buffered into memory so that they can be individually processed and converted into the YUV color space using conventional 2D image processing algorithms. These include black level correction, noise filtering, defect pixel correction, CFA interpolation, color correction, edge enhancement, gamma correction, and color space conversion.

3.2 Stereoscopic 3A

Stereoscopic 3A includes auto-focus (AF), auto white balance (AWB), and auto exposure (AE) for the left and right views. Compared to conventional 3A algorithms for 2D images, stereoscopic 3A has additional responsibility for ensuring that both left and right views match each other in terms of their brightness, focus, and color balance. Any mismatch in these dimensions would cause eye strain in viewers. If conventional 3A algorithms run independently for the left and right cameras, a mismatch will likely occur because of the variation in sensor sensitivity, lens focal length, actuator positions, field of view, etc.

To eliminate the mismatch, the stereo 3A algorithm coordinates the AE, AWB, and AF between the two sensors. For stereo AWB, two strategies can be used: 1) align the white balance gains for the left and right views, or 2) align the color of the white balanced image. The first option is less computationally intensive, while the second option produces better color matching. For stereo AF, variations in the focal length of the lens as well as lens actuator positions require the AF algorithm to analyze each view and set each lens position such that the same focal plane is achieved in both views. For stereo AE, a simple strategy is to make sure that the same exposure value is applied to both sensors. A more advanced method is to adjust the exposures independently such that the scene has the same brightness in both views.

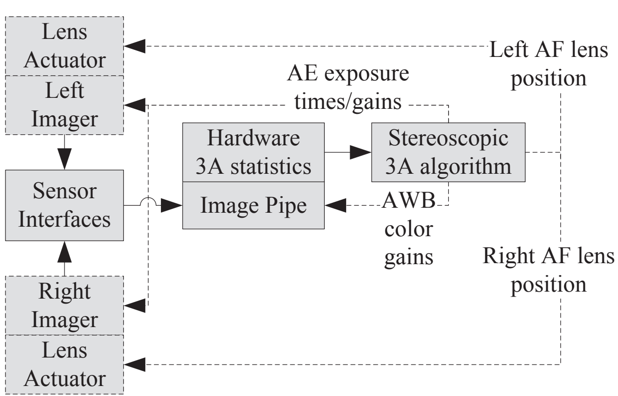

Figure 3 - Overall view of stereoscopic 3A algorithm and control

3.3 Stereo Misalignment Correction

Although we expect stereo cameras to be identical and parallel, in practice there is mismatch between the two cameras and their alignment is not perfect [2]. These can be from manufacturing tolerances, focal length differences, consumers dropping their phones, or even temperature variations. The most straight-forward approach to correct for the misalignment is factory calibration, which is not ideal because it is expensive. Misalignment can also change over time due to environmental factors, which makes the factory calibration less effective. Therefore, a real-time misalignment correction algorithm inside the camera is needed. Our misalignment correction algorithm automatically analyzes the left and right views and fits the difference between them to a modified affine transformation model. This model gives translational, rotational, and scaling parameters that characterize the misalignment between the cameras. Based on these parameters, the left and right views are modified to correct for the misalignment for every frame. As the misalignment changes over time, our algorithm automatically detects and corrects it.

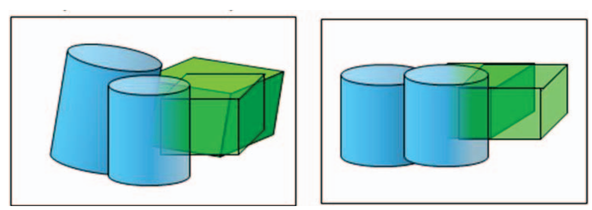

Figure 4 - Misalignment Correction simulation

3.4 Stereo Auto Convergence

With a fixed camera separation and angle, the 3D content captured by a stereo camera can be very difficult to view because of the large positive or negative disparities. A Stereo Auto Convergence (SAC) algorithm addresses this problem by adjusting the convergence plane of the 3D content, which makes the scene much more comfortable to view on a 3D display. SAC estimates the disparity between the left and the right views, decides the convergence point, and then shifts the two views in the horizontal direction to put the convergence plane perceptually onto the display. Different strategies for selecting the convergence point can be used. For example, one strategy is to always put the main object on or near the screen. In general, objects that are close to the screen are most comfortable to view because this reduces the vergence-accommodation conflict. Objects that are too far in front or behind the screen are more difficult to fuse; therefore, these situations should be minimized. Temporal stability of the convergence plane is also an important concern as viewers find it difficult to adapt to a fast moving convergence.

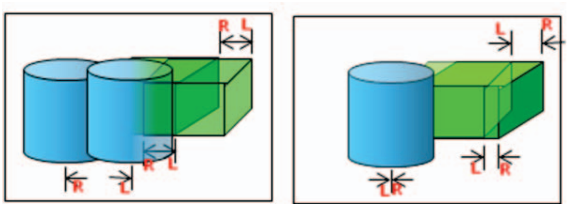

Figure 5 - Convergence simulation

3.5 Video Stabilization

Handheld camera motion jitter makes it harder for viewers to stay fused and focused on a stereoscopic object [3]; therefore, jitter needs to be reduced. Stereoscopic video stabilization analyzes input video and automatically estimates this jitter motion. Because the cameras are rigidly mounted, the camera motion can be calculated from just a single view (e.g. only the left view). Then, the algorithm compensates for jitter by cropping a subwindow from the input video frame. The same correction is applied to both the left and right views.

4. CODEC SUBSYSTEM

The codec subsystem performs the encoding and decoding of the image and video frames. The image and video encoders and decoders are designed to support multiple stereoscopic frame packing arrangements for end system interoperability. The H.264 SEI stereoscopic coding encodes the stereo video content in various arrangements (e.g. top-over-bottom, side-by-side, interleaving, interlacing, etc) for full resolution or subsampled resolutions. The two major H.264 SEI formats are H.264 legacy 2004 supplemental enhancement information (SEI) 3D messaging and the more recent 2010 SEI 3D messaging, which are specified in the H.264 Annex D specification [4]. MultiView Video Coding (MVC) can also be used for stereoscopic video coding, and is specified in the H.264 Annex H specification [4]. The two main stereoscopic still image coding standards are stereo JPEG (JPS) and JPEG Multi-Picture Object (MPO). The image coding system encapsulates the stereo data layout description using the Exchangeable Image File format specification (Exif).

4.1 Encoding

The encoder compresses the camera frames, significantly reduces the storage size for storage. The encoder supports real-time performance for stereoscopic high definition video resolutions and full resolution stereo image capture for the latest mobile platform configurations. The encoder also allows the image or video to be placed into a recognizable file container for interoperability between systems. The frame orientation is stored in the bitstream or file container so that the decoder will be able to recognize and display the stereoscopic 3D content correctly.

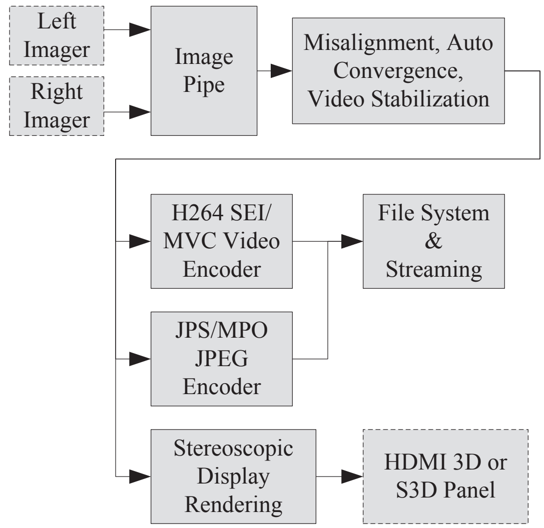

Figure 6 - Overall view of stereoscopic imaging and video encoding subsystem

4.2 Decoding

The decoder receives an image or video bitstream from a file or internet stream, and decodes the frame so that it can be displayed by the display sub-system. The decoding system provides the frame packing information to the display subsystem for proper orientation and management of the decoded frames to efficiently render on local auto stereoscopic and external displays. In addition, non-standard stereoscopic streams are also supported from user input. When the user observes two images or frames, he or she can manually enter a top-over-bottom or side-by-side command to complement the resolution details from the decoder for rendering the content on the display in stereoscopic 3D.

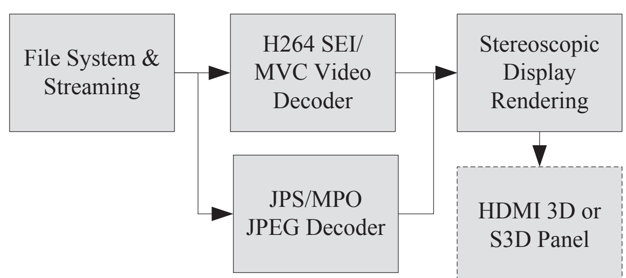

Figure 7 - Overall view of stereoscopic imaging and video decoding subsystem

5. GRAPHICS SUBSYSTEM

The graphics subsystem renders the system's graphical user interface (GUI), as well as allowing stereoscopic rendering of 3D games. Using OpenGL, the application can setup a viewport configuration that matches the orientation of the auto-stereoscopic screen to maximize the per view pixel resolution. The scene is then drawn twice, using either a toed-in or an asymmetric frustum virtual camera setup. An asymmetric frustum setup is generally preferred as it avoids visual distortion at edges of the scene. Other techniques may be used, such as drawing only once then synthesizing the other view from the z-buffer using parallax occlusion mapping. The rasterized end result is a side-by-side or top-bottom frame buffer which can then be post processed by the display subsystem. In this way, the application does not require special signaling with the display driver to synchronize which view is being rendered.

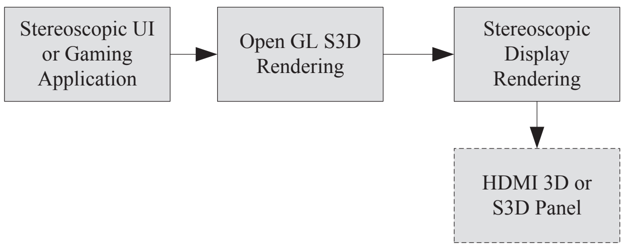

Figure 8 - Overall view of graphic pipeline

6. DISPLAY SUBSYSTEM

The display subsystem receives image, video, and graphic layers from the graphics engine, camera subsystem and/or codec subsystem. The driver of the display subsystem determines how to best render the frames for the supported stereoscopic displays. HDMI v1.4a 3D defines many mandatory and optional frame-packing options. Auto-stereoscopic display panels commonly require column and/or row interleaving, side-by-side frame packing, and even frame interleaving.

The OMAP4430 processor's display sub-system includes four concurrent 2D-DMA read engines, three with color conversion and scaling capabilities as well as a one 2D-DMA write engine. Using these overlay pipelines, one can post-process a given frame-buffer to support different types of stereoscopic displays.

For example, in order to row-interleave a decoded stereoscopic video frame in NV12 format, the display subsystem: 1) reads the left view of the frame-buffer, color converts it into RGB space and downscales to the required panel resolution on-the-fly. 2) The left view RGB data is fed to the 2D-DMA write pipeline which has been configured to skip every other line when writing the result into SDRAM. 3) The process is then repeated for the right view to produce the final row interleaved frame.

Similarly, to perform column interleaving we introduce a 90 degree rotation when the data is read in from SDRAM and a -90 degree rotation as the data is written out to SDRAM. The combination of rotation and row interleaving results in column interleaving without having to disable DMA write bursting, which would be necessary if we simply skipped pixel positions when writing to memory. DMA write bursting can be enabled in this case by using the OMAP4430 TILER engine, which virtualizes access to the frame buffer and optimizes memory fetches for 90 degree increment rotations.

For the mandatory HDMI 1.4a formats which specify an active space region between the left and right view, the overlay composition engine of the display subsystem is used along with a left and right overlay layer. These layers are positioned such that the active space requirement is met.

For example, to conform to the S3D 720P frame packing timings described in the standard, the overlay engine generates 1280 pixels by 1470 lines. The left view layer is positioned at (0, 0) and the right view layer is positioned at (0, 720+30). It is important to note this processing is done on-the-fly without recomposing back into SDRAM.

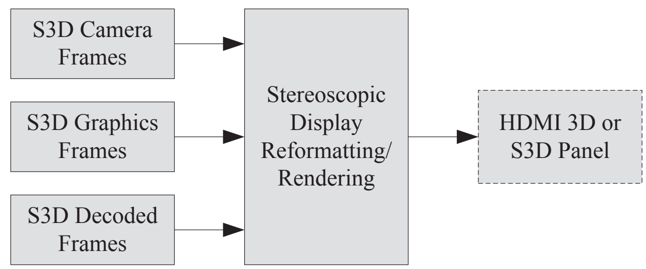

Figure 9 - Overall view of stereoscopic display subsystem

7. CONCLUSION

We presented an embedded programmable stereo camera system based on the OMAP4430 applications processor. Our solution implements new, stereo specific image processing algorithms such as stereo 3A, auto convergence, and camera misalignment correction, which are essential to produce a compelling 3D viewing experience without creating any eye strain.

8. REFERENCES

[1] Texas Instruments Incorporated OMAP4430, http://www.ti.com/omap4430

[2] L. Lipton, *Foundations of the Stereoscopic Cinema*, Van Nostrand-Reinhold: New York , 1982.

[3] A.U. Batur and B. Flinchbaugh, "Video Stabilization with Optimized Motion Estimation Resolution," *Image Processing, 2006 IEEE International Conference on*, pp.465-468, 8-11 Oct. 2006.

[4] A. Vetro, T. Weigand, and G.J. Sullivan, "Overview of the Stereo and Multiview Video Coding Extensions of the H.264/MPEG-4 AVC Standard," Invited Paper, Proceedings of the IEEE, Vol. 99, No. 4, April 2011