Understanding the Impact of Inter-Lens and Temporal Stereoscopic Video Compression

Wu-chi Feng, Feng Liu
Portland State University
Portland, OR 97207
{wuchi, fliu}@cs.pdx.edu

ABSTRACT
As we move toward more ubiquitous stereoscopic video, particularly with multiple (> 2) lenses, the need to understand the efficiency of compression will become increasingly important. In this paper, we explore the impact of spatial (between lenses) and temporal (over time) compression for stereoscopic video images. In particular, because stereoscopic images are taken at the same time, there is expected to be a high correlation between pixels in the horizontal direction due to the fixed nature of the multiple lenses. We propose a vertically reduced search window in order to take advantage of this correlation. Starting with multiple stereoscopic video sequences shot using a production studio 3D camera, we explore the effectiveness of temporal and inter-lens motion compensation for stereoscopic video. Furthermore, the experiments use exhaustive search to remove the effects of heuristic-based motion-compensation techniques.

Categories and Subject Descriptors
H.3.4 [**Information Storage and Retrieval**]: Systems and Software

General Terms
Algorithms, Design, Experimentation,

Keywords
Stereoscopic imaging, stereoscopic compression.

1. INTRODUCTION
One promising area in multimedia systems is stereoscopic imaging, which allows users to capture the feeling of depth in an image by feeding two different images to the left and right eyes. Currently, 3D cameras such as the Fuji FinePix Real 3D Digital and 3D video cameras such as the Panasonic AG-3DA1 3D Camcorder are available. Even smartphones like the LG Optimus 3D and the HTC Evo 3D are equipped with two lenses in order to allow for the capture of 3D video. Furthermore, devices are now emerging that allow for the display of stereoscopic imaging such as the Nintendo 3DS and the LG Thrill 3D without the requirement of shuttered, polarized, or blue/red glasses.

While it might seem that the problem of capture and display of stereoscopic images and video are essentially solved, this is far from the truth. As noted in a recent article, there are problems with such stereoscopic devices [14]. Professor Banks at UC Berkeley has pointed out that viewers can suffer "3D fatigue" from improperly produced sequences. For movies like Avatar, the filmmaker spent particular attention to reducing eye fatigue by drawing viewer focus on just one object at a time.

NOSSDAV'12, June 7-8, 2012, Toronto, Ontario, Canada.

Figure 1. Imaging Devices: This figure shows two mockups of future multi-lens stereoscopic imaging devices

As described in a previous paper [1], we believe that with more general availability of 3D capture hardware, and without careful consideration of how stereo video is being captured, that more than likely many of the stereoscopic streams captured will end up causing 3D fatigue and viewing issues. Rather than making all users aware of stereoscopic composition and cinematography rules, we envision that a stereoscopic camera may one day be made of many linearly aligned lenses to provide a denser sampling of the viewpoints. With the knowledge of the display size and the viewing distance from the screen, the system can then render a stereoscopic image from a subset of the images taken to maximize viewing experience. As a result, the underlying computing system will need to be able to deal with a large number of streams (one from each camera) during capture and display. An example of such an envisioned system is shown in Figure 1.

As will be described shortly, when greater than two lenses are used for stereoscopic video capture, compression and retrieval become more interesting. First, the greatest compression gains are achieved by removing all redundancy within the streams. For current stereoscopic video systems with two lenses this is not a problem as all the content is required for display. In the multi-lens scenario, the compression may need to take into account the fact that only select sub-streams will be retrieved for display. Second, while multi video codecs have been defined, they are typical standards in that *they specify the format of the compressed stream but not how to get there*.

In this paper, we begin the process of understanding the fundamental trade-off between inter-lens compression and temporal compression for stereoscopic video compression. This paper uses several standard stereoscopic video sequences from a local production studio as input and explores, for a number of parameters, how inter-lens frames and temporal frames impact compression. Our results show that we may be able to take advantage of the motion compensation process to help with feature tracking (required for optimal substream selection) for display, or vice versa.

In the next section, we will briefly review some of the related and background work. This will include further motivation for stereoscopic video versus 3D and multiview video. Section 3 we briefly review our proposed multi-lens video system that we presented at NOSSDAV 2011. Section 4 presents our detailed experimental results followed by a discussion and future work.

2. RELATED WORK

2.1 Stereoscopic Imaging

Stereopsis is the process in visual perception that leads to the perception of depth. Each eye can be thought of as an individual point of view. The brain perceives depth by processing the discrepancy between these views, which is known as retinal disparity. Objects that are far away have a small retinal disparity. As objects are brought closer, we perceive them as being closer because the disparity has increased. What this means is that our perceived depth is inversely proportional to the retinal disparity. The goal of stereoscopic imaging is to recreate depth perception in the brain.

To recreate depth perception, stereoscopic systems try to display the appropriate retinal disparity for the object being viewed. This is accomplished with two images of the same object that each represent the left and right eye point of view. The two images need to be delivered to the eyes separately. This is accomplished by using either red-cyan glasses, polarized light with different polarization for each eye, or shuttered glasses that quickly alternate covering each eye and changing the on screen image.

In a simple model of stereoscopic display, the retinal disparity depends on the interocular distance, the distance between the viewer and the screen and the on-screen disparity between two displayed images. The interocular distance is constant, roughly 2.5" for an adult. The on-screen disparity linearly depends on the raw disparity between two stereoscopic images and the screen size. A stereoscopic image pair designed for a certain viewing scenario (a certain viewing distance and screen size) may not be appropriate for another viewing scenario. The only adjustable parameter with a fixed user distance and display size is the raw disparity in the stereoscopic content. Assuming a fixed viewing distance, as the size of the screen increases, the retinal disparity will usually increase. In order to maintain the same perceived depth, the raw disparity will need to be decreased to maintain the on-screen disparity, thus, maintaining the retinal disparity. Similarly, assuming a fixed screen size, as the distance of the user from the screen increases, the retinal disparity will decrease. In order to maintain the same perceived depth, the raw disparity will need to be adjusted to compensate. The dependence of the retinal disparity on these factors is complicated. Readers interested in a more detailed description are referred to [2].

Unlike 2D content, stereoscopic images need to be adapted to different viewing scenarios for the proper experience [10]. Wang and Sawchuk developed a disparity manipulation system that combines image warping and data-filling techniques for novel view synthesis according to the new disparity map [15]. Lang et al. further discussed the important perceptual aspects of stereo vision and their implications for stereoscopic content creation, and then provided a set of basic disparity mapping operators to enable disparity map editing [6].

2.2 Multi-camera and 3D Video

The use of multiple cameras in multimedia applications and systems has been the subject of research for the last several years. Minimal overlap multi-camera systems have been used for tracking and management in surveillance and traffic monitoring systems. While some inter-camera overlap sometimes occurs, the focus of such systems is typically on the coordination amongst multiple cameras and not necessarily efficient compression.

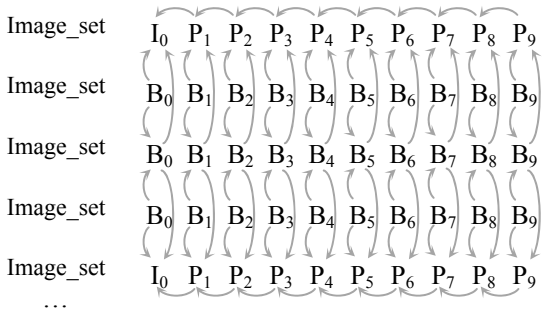


Figure 2. Typical MVC coding example – This figure shows an example compression of multiview images that is typical in the literature. The image sets (horizontal rows) are images that are taken at the same time.

In the multimedia computing and networking community, multi-camera image systems have been used to create better immersion systems. Most notably, efforts from UNC's immersive tele-conferencing system [5] and UIUC's TEEVEE project [16] use multiple cameras pointed towards a small number of object. The purpose of these projects is to capture depth from multiple cameras in order to create 3D geometries of the objects being captured. The main reason for this is that it allows the remote viewer to allow the user to control the view point in the environment being captured. The display in these systems is planar (i.e., displayed to a normal screen).[1]

2.3 Multi-view Video Coding

Several efforts have focused on compression to take advantage of redundancy in multiple camera/video systems. Perhaps the closest work to ours is the recent introduction of 3D stereoscopic Blu-Ray players with content such as Avatar [7]. The underlying standard used for this type of video is the H.264/MPEG-4 AVC standard with amendment for multi-view video coding (MVC) [3]. There are two important points here with respect to our work. First, standards specify the format for a properly formatted stream, not how to get there. Thus, algorithms are still needed to compress the image data into a stream that is useful for the application. Second, current implementations (i.e., 2 channels) use as much compression between frames temporally and between channels as possible. The reason for this is that the entire stream is decompressed when played back so partial access to data is not required. Further details of H.264/MPEG4 AVC and MVC can be found in [9][13].

As an example, a typical compression model found in MVC compression papers typically have a compression structure similar to that found in Figure 2 and in [11]. In this figure, we see that the typical MVC compression approach is to maximize compression. $Image\_set_0$ and $Image\_set_4$ are key frames (i.e. I-frames of traditional MPEG-1 or MPEG-2 video streams). All frames within the image sets are differentially coded.

[1] Capturing 3D depths and texture can, in theory enable stereoscopic display. We, however, focus on systems that are primarily meant for stereoscopic display.

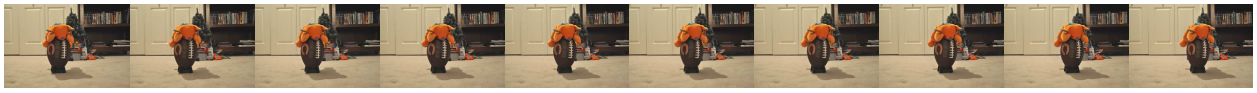


Figure 3 - Stereoscopic Imaging: This figure shows a sequence of images taken using a single DSLR camera with lens spacing of 0.5 inches, similar to what we envision for stereoscopic cameras of the future.

3. A STEREOSCOPIC ARRAY MODEL

Our envisioned system consists of an array of lenses that capture image data in synchrony. Then, depending upon the viewing distance from the screen and screen size, the display system will select or create two images from the array of images that will deliver a pleasant user experience. Our "standard" stereoscopic camera will be an array of 10 image lenses, each 0.5" apart. Given that the standard stereoscopic camera will have the lens 2.5" apart, this configuration will give us, for each eye, two additional images to the left and two additional to the right. Each synchronized and captured frame is referred to as an *image_set*. In Figure 3, we have shown a set of images representing a "multi-lens" stereoscopic array. A standard stereoscopic camera with 2.5" spacing corresponds to the images 2 and 7 in Figure 3.

In the storage and compression of the multi-lens video data we need to do two things. We first need to analyze all the objects within the images to determine the disparity of features. These disparities will then be used in the retrieval process to select the best subset of images for a particular view scenario. Second, we need to compress the data. As previously mentioned, the focus of MVC compression is to typically achieve the highest compression ratio possible, typically sacrificing the ability to retrieve subsets of images.

Our envisioned storage of multi-lens stereoscopic image data will have a *thread* of images that are compressed following a typical video compression algorithm. The thread is chosen based upon the disparity calculations to match the most-likely expected viewing scenario. We note that one of the threads can be predictive coded with respect to the other but that it may affect retrieval times, particularly if the viewing scenario requires images other than from the thread. An example of such compression is shown in Figure 4, where a fixed thread is used.

To accomplish disparity calculations, a method to select important features within an image set, matching them up, and calculating the disparity (horizontal distance) between the corresponding points. Among many local feature descriptors, SIFT [8] is reported to perform best by recent work [12]. We use SIFT points as features for our stereoscopic image sets. The best candidate match for each SIFT point in one image is found by identifying its nearest neighbor in the other image. We do note that the actual best disparity to use is still an open research question, although typically 3-8% disparity is considered to be appropriate working ranges [4].

Because of the large amount of image data expected to be generated from such a system and the need to do both compression and analysis of the image sets, *we need to start understanding the basic trade-offs in terms of inter-lens compression and temporal video compression*. In particular, this paper focuses on the beginning investigations of such compression. We use exhaustive motion compensation for a variety of stereoscopic videos in order to begin understanding this trade-off without bias from heuristic search choices.

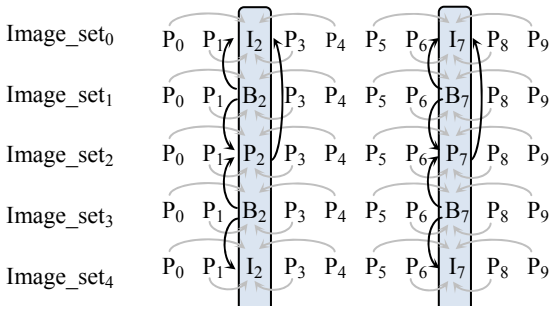


Figure 4 - Stereoscopic Threaded Compression:

4. Experimentation

4.1 Experimental Setup

To study the effect of motion estimation search window size, we have obtained three stereoscopic video sequences from a local production company using a 2-lens Panasonic AG-3DA1 Professional 3D Camera recorder. The first shot was taken from the side of a road using a fixed camera on a tripod. The sequence has two vehicles moving from a far depth towards the screen. The second was taken from a moving vehicle being driven down the road. The sequence contains significant movement as all the scenery is moving past the car. We also note that there was quite a bit of image instability due to the car movement and the camera being hand held. The final sequence is a fixed camera sequence taken at a pier. The movement in the video includes people walking relatively slowly and the ripples from the water surface. Two images from each sequence are shown in Figure 5.

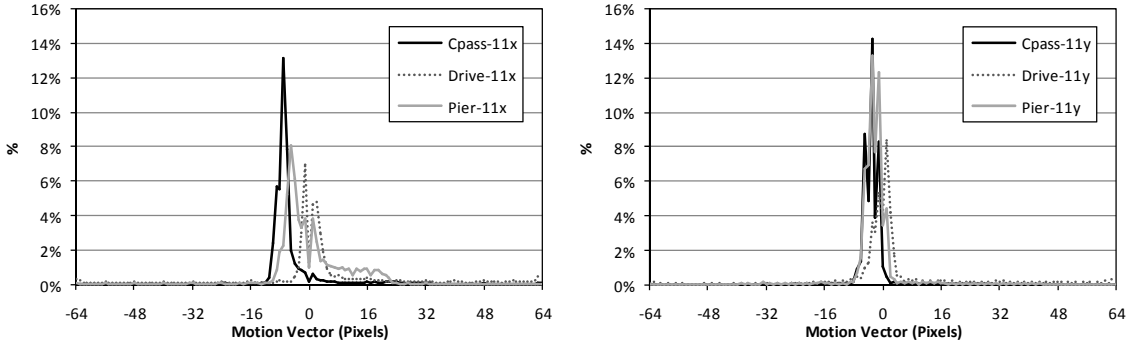
For experimentation, we used the reference MPEG-2 encoder[2]. For all experiments, we chose to use the *exhaustive* search option for all compression. This was to avoid having the heuristics of motion compensation affect the results and to help us establish the upper end for how well the encoder can do for the test sequences. We also note that the popular *ffmpeg* software has long since removed the ability to do exhaustive searches in motion compensation.

Using the software, we encoded each of the three streams at 11 Mbps and 20 Mbps. The former being the average bit-rate of many Blu-ray discs, and the latter being the target bit-rate for HDTV. For each sequence, we encoded each using search ranges of (in width x height), 64 x 64, 64 x 32, 64 x 16, 64 x 8, 64 x 4, 32 x 32, 16 x 16, 8 x 8. Note the software uses these numbers as +/-; hence, 8x8 is actually +/- 8 pixels in width and +/-8 pixels in height.

[2] http://www.mpeg.org/MPEG/video

(a) *Cpass* sequence (b) *Drive* sequence (c) *Pier* sequence

Figure 5 - Stereoscopic Video Data Set: These figures show two sample left eye images from the stereoscopic video sequence that we used for the experimentation in this paper. They were captured using a Panasonic AG-3DA1 Professional 3D Video recorder.



(a) X motion vector distribution (b) Y motion vector distribution

Figure 6 – Inter-lens Predictive-coded Macroblock Distribution: These figures show the X and Y motion vector distributions for macroblocks where at least one of the components was not zero. That is, predictive coded macroblocks with no motion vector are not in the distributions

4.2 Effect of Inter-Lens Compression

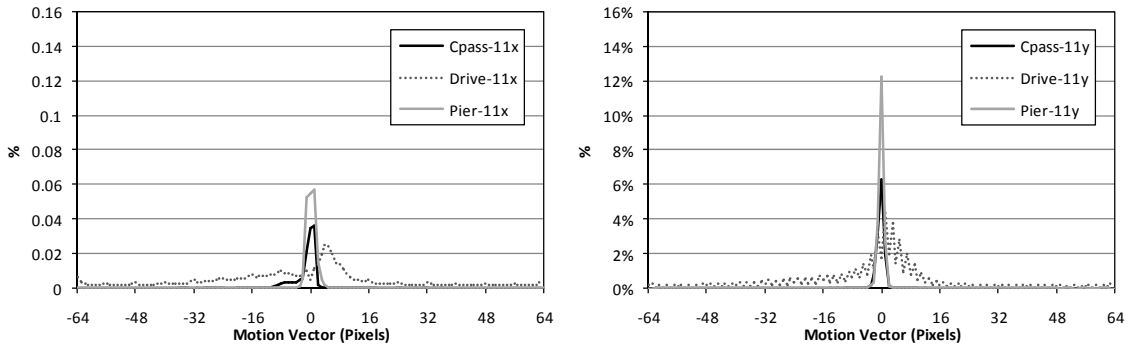
In the first set of experiments, we were interested in understanding the impact of inter-lens compression. To study this, we set up the encoder so that each *left* image served as the reference frame and each *right* image was predicted from the left in the image_set. Thus, each of the left frames was encoded as an I-frame and each of the right images was predictive coded with respect to the left. For these results, we only show the 11 Mbps encodings as the 20 Mbps encoding results were nearly identical.

Table 1 shows the distribution of macroblock encodings for the right images for the 11 Mbps encodings. *Skip* and *Intra* encoding are the skipped and intra-coded macroblocks. *Zero* indicates the percentage of macroblocks with no motion vector but predictive coded coefficients. *Pred* indicates the percentage of macroblocks that were predictive coded. As shown in the table, the Pier sequence is significantly different than the other two sequences. There are many more skipped blocks as a result of the camera being fixed with a static sky, building, and pier. There are also much fewer zero encoded motion vectors due to the water ripples in the bottom of the image causing residuals to be added to the compressed output stream.

Movie	Skip	Intra	Zero	Pred.
Cpass	11.5%	0.22%	36.94%	51.33%
Drive	6.73%	0.79%	41.12%	51.36%
Pier	25.34%	2.17%	5.40%	67.10%

Table 1: This table shows the distribution of macroblock encodings for the inter-lens sequences

For macroblocks that were encoded with a motion vector (i.e. pred), we have graphed the histogram for the distribution of the motion vector magnitude (in pixels) in Figure 6. Figure 6(a)



(a) X motion vector distribution (b) Y motion vector distribution

Figure 7 Temporal Predictive-coded Macroblock Distribution: These figures show the X and Y motion vector distributions for macroblocks where at least one of the components was not zero. That is, predictive coded macroblocks with no motion vector are not in the distributions

shows the distribution of the motion vector sizes in the horizontal direction. We note that there are dips around 0 because the macroblocks with no motion vector (i.e., just predictive coded) are not shown. These values are under Zero in Table 1. The *Drive* sequence is fairly evenly centered around 0. The reason for this is that there is significant movement in between frames of the video, thus, motion compensation tends to be more random as the camera shakes more or less randomly. In the *Cpass* and *Pier* sequences, the motion vectors are clearly biased in the negative horizontal direction. This suggests that the motion compensation algorithm may be useful in helping with feature tracking and disparity calculation (or vice versa). In the *y* direction, the *Cpass* and *Pier* sequences show a slight negative bias in motion vectors. We believe that this is due to the sequential search from the top to the bottom of the motion estimation range in the encoder. A candidate motion vector is only replaced if it exceeds the previous best. Thus, ties go to the motion vector that was encountered first. Finally, this suggests that there is the potential to take advantage of inter-frame compression while reducing the search range in the y direction for stereoscopic image sets.

4.3 Effect of Temporal Compression

To understand the difference between inter-lens compression and temporal compression for stereoscopic video, we have also performed temporal compression between frames. In order to make the comparison more useful to the experiments in 4.2, we compressed each of the sequences in the following way: Every frame, *i,* was compressed as an I-frame. Every *i+1* frame was then predictive coded with respect to frame *i.* Thus, every frame (except the first) is compressed as an I-frame as well as predictive coded with respect to the previous frame. For these experiments, we used only the right images to match which frames were being predicted as in the inter-lens study in section 4.2.

Table 2 shows the distribution of macroblock encodings for the predictive coded right images. Here, we see that in the temporal inter-coding case that the number of skipped macroblocks jumps dramatically for the *CPass* and *Pier*. This is somewhat expected as the camera for the reference frame is fixed. We also see that the number of predictive coded macroblocks drops for the Cpass and Pier sequences also because of the shift to zero-motion-vector macroblocks. The *Drive* sequence had similar numbers to the inter-lens compression case. We believe that this is primarily due to the instability of the camera as the car was being driven.

Movie	Skip	Intra	Zero	Pred.
Cpass	52.31%	0%	34.46%	13.22%
Drive	17.80%	2.87%	20.32%	59.01%
Pier	55.44%	0.33%	23.17%	21.07%

Table 2: This table shows the distribution of macroblock encodings for the temporal compression sequences.

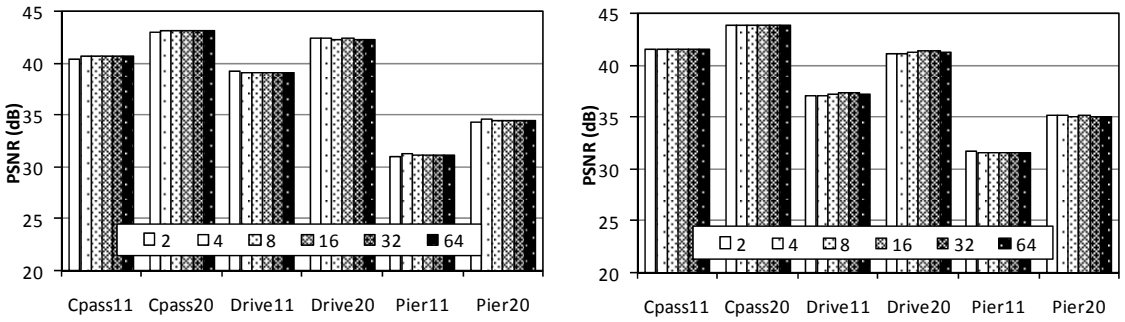
Figure 7 shows the distribution of the resultant motion vectors. From Table 2, these represent 13.22%, 59.01%, and 21.07% of the macroblocks in the predictive coded frames. Compared with Figure 6, we a significant decrease in the number of macroblocks encoded. More importantly, we see very little bias in the *x* direction as we saw in the inter-lens compression case. One peculiar issue that arose was in the distribution of the *y* motion vectors for the *Drive* sequence. The distribution was highly jagged. We believe that this might be due to the video stabilization that is built into the video camera.

4.4 Effect of Constrained Search

In this section, we explore the efficacy of reducing the vertical search range in order to improve compression performance. The underlying premise is that given a constant number of compute cycles that it might be more useful to search horizontally rather in the typical unbiased search of today's encoders. For these experiments, we used the same compression encoding as described in Sections 4.2 and 4.3. Instead of using just the 64x64 exhaustive search, we added the search window sizes of 64x2, 64x4, 64x8, 64x16, and 64x32.

In Figure 8, we have grouped the results by sequence and by maximum vertical search range. We note that the actual search range is 2 times the number specified. We have grouped the results by sequence name, bit rate of encoding, and then the search range. As expected in all cases, the 20 Mbps encodings have higher PSNR results than the 11 Mbps encodings.

Surprisingly, there is very little difference in terms of resulting PSNR regardless of the search range used. In all cases using 64x2 results in the lowest PSNR as one might expect. However, the



(a) Inter-lens Compression (b) Temporal Compression

Figure 8 PSNR Results: These figures show PSNR results for the Inter-lens (a) and Temporal (b) compression as described in Section 4.2 and 4.3, respectively. Each column represents a maximum vertical search range of 2, 4, 8, 16, 32, and 64.

difference is not that large. In comparing inter-lens and temporal compression, we see that as the *Cpass* and *Pier* videos all have better compression under temporal compression than inter-lens compression. This suggests that for sequences that are highly unstable in terms of camera motion that having the fixed relation between the stereoscopic lenses provides better reference images.

4.5 Discussion / Future Work / Conclusion

We have shown through exhaustive (+/- 64 pixel) motion compensation for a number of sequences that temporal compression using standard two-lens spacing has higher coherence than the inter-lens coherence, except for highly unstable sequences. We have also found that for the HDTV sequences that we have obtained that the vertical search range for relatively static shots can be greatly reduced without affecting the image quality while improving motion compensation speed.

We are currently in the process of building a multi-lens array from point-of-view cameras. While inter-lens compression, in general, provides less coherence between images than temporal coherence, we will revisit this as lenses are added at finer granularity, which presumably, will increase the inter-lens coherence.

Our results also show a deficiency in the selection of the input sequences. The *Cpass* and *Pier* sequences are very similar in that they have fixed cameras. In the future, we hope to obtain a wider selection of input sequences to provide to the community, including a panning stereoscopic set. We do note, however, that panning shots are somewhat harder to shoot in order to maintain stereoscopic cinematography rules (e.g., not having an object along the edge of the image.

Future work will also entail moving toward H.264 encoding to add quarter pixel search to the half and full pel search in the MPEG-2 encoder.

5. REFERENCES

[1] Wu-chi Feng, Feng Liu, Yuzhen Niu, Scott Price, "Systems Support for Stereoscopic Video Compression", in *Proc. of NOSSDAV 2011*, Vancouver, BC, pp. 99-104, June 2011.

[2] M. Guttmann, L. Wolf, D. Cohen-Or, "Semiautomatic Stereo Extraction from Video Footage", in *Proc. of the IEEE Inter. Conf. on Computer Vision*, pages 136 – 142, 2009.

[3] Y. He, J. Ostermann, M. Tanimoto, A. Smolic, "Introduction to the Special Section on Multiview Video Coding", in *IEEE Transactions on Circuits and Systems for Video Technology,* Vol. 17, No. 11, pp. 1433-1435, Nov. 2007.

[4] http://apophysisrevealed.com/apo3dblog/2009/07/192

[5] San-Uok Kum, K. Mayer-Patel, H. Fuchs, "Real-Time Compression for Dynamic 3D Environments", in *Proceedings of ACM Multimedia,* 2003.

[6] M. Lang, A. Hornung, O.Wang, S. Poulakos, A. Smolic, M. Gross, "Nonlinear Disparity Mapping for Stereoscopic 3D", *ACM Transaction on. Graphics*, 29(4), 2010.

[7] R. Lawler, "Blue-ray 3D Specifications Finalized, Your PS3 is Ready", Dec. 17, 2009, From: http://www.engadget.com/2009/12/17/blu-ray-3d-specifications-finalized-your-ps3-is-ready/

[8] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision, Vol. 60, No. 2, pp 91-110, 2004.

[9] D. Marpe, T. Wiegand, G.J. Sullivan, "The H.264/MPEG4 Advanced Video Coding Standard and its Applications", *IEEE Communications*, pp. 134-143, August 2006.

[10] B. Mendiburu, 3D Movie Making: Stereoscopic Digital Cinema from Script to Screen, Focal Press, 2009.

[11] P. Merkle, A. Smolicc, K. Muller, T. Wiegand, "Efficient Prediction Structures for Multi-view Video Coding", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 17, No. 11, November, 2007.

[12] K. Mikolajczyk, C. Schmid, "A Performance Evaluation of Local Descriptors", *IEEE Trans. on Pattern Analysis and Machine Intelligence,* Vol. 27, No. 10, pp. 1615-1630, 2005.

[13] MPEG: "Introduction to Multiview Video Coding", ISO/IEC JTC 1/SC 29/WG 11 N9580, Edited by A. Smolic, Jan. 2008.

[14] D. Sanchez, "Are 3D Movies, TV Bad For Your Eyes?", February 24, 2010, Retrieved from KGO News: http://abclocal.go.com/kgo/story?id=7278834

[15] C. Wang, A. A. Sawchuk, "Disparity Manipulation for Stereo Images and Video, in Proc. *SPIE*, Vol. 6803, pages E1– E12, 2008.

[16] Z. Yang, Y. Cui, B. Yu, J. Liang, K. Nahrstedt, S. H. Jung, R. Bajcsy, "TEEVE: The Next Generation Architecture for Tele-Immersive Environments", in *Proceedings of the 7th IEEE International Symposium on* Multimedia *(ISM'05)*, Irvine, CA, 2005.