Three-Dimensional Wavelet Coding of Video with Global Motion Compensation

Albert Wang†, Zixiang Xiong‡,
Philip A. Chou†, and Sanjeev Mehrotra†

† *Microsoft Corporation, One Microsoft Way, Redmond, WA 98052-6399*

‡ *Dept. of Electrical Engineering, Univ. of Hawaii, 2540 Dole St., Honolulu, HI 96822*

alwang, pachou, sanjeevm@microsoft.com, zx@lena.eng.hawaii.edu

Abstract

Three-dimensional (2D+T) wavelet coding of video using SPIHT has been shown to outperform standard predictive video coders on complex high-motion sequences, and is competitive with standard predictive video coders on simple low-motion sequences. However, on a number of typical moderate-motion sequences characterized by largely rigid motions, 3D SPIHT performs several dB worse than motion-compensated predictive coders, because it is does not take advantage of the real physical motion underlying the scene. In this paper, we introduce global motion compensation for 3D subband video coders, and find .5–2 dB gain on sequences with dominant background motion. Our approach is a hybrid of video coding based on sprites, or mosaics, and subband coding.

1 Introduction

Motion compensated predictive video coders have been highly successful since their introduction in the '70s. Today they are quickly being deployed commercially in the form of the standards H.261/3 and MPEG-1/2/4. However, they lack a feature that is becoming increasingly important in the emerging world of heterogeneous packet networks and wireless communication: fine scalability. Experiments with MPEG-2, MPEG-4, and H.263 scalability modes show that 0.5–1.5 dB is lost with every layer, relative to a monolithic (nonlayered) coding [1, 2]. This is the motivation for considering structures for video coding other than backward adaptive differential prediction.

An alternative to differential predictive coding is transform coding, in the temporal direction. Fine scalability is easy to accomplish with transform coding, with no loss in performance relative to monolithic encoding, as has been repeatedly demonstrated in image coding systems e.g., [3, 4]. However, transform coding of video in the temporal direction does introduce far more delay than predictive coding, which is typically unacceptable for bi-directional interactive applications such as videotelephony and

1068-0314/99 $10.00 © 1999 IEEE

videoconferencing. On the other hand, for uni-directional, low-interactivity applications such as broadcast or multicast video, or video-on-demand, the additional delay is typically acceptable. It is with this latter class of applications, where scalability is most important, that we are here concerned.

Three-dimensional spatio-temporal (2D+T) transform coding of video was well investigated in the late '80s and early '90s e.g., [5, 6, 7, 8, 9, 10]. However, it turned out that 3D transform coding could not compete effectively against traditional motion compensated prediction in terms of rate-distortion performance and visual quality. The reason is that motion compensated prediction of video takes advantage of a motion model, which is highly effective in most real world cases, while 3D transform coders have not yet been able to take advantage of such a model.

Recently, wavelet transform coding methods based on zerotrees [3, 4] have become so effective that even without the benefit of a motion model, 3D wavelet transform coding of video using these methods is competitive with standard motion compensated predictive coding. In particular, Kim and Pearlman [11] reported two years ago at DCC that their 3D SPIHT algorithm outperforms MPEG-2 by an average of 0.8 dB on the standard 30 fps SIF (352×240) sequences *table tennis* and *football* at 760–2530 kbps (0.3–1.0 bits per pixel). At lower bit rates, Kim, Xiong, and Pearlman [12] recently reported that 3D SPIHT is within 1.39 dB of H.263 (TMN 2.0) on the standard 10 fps QCIF (176×120) sequences *carphone*, *mother and daughter*, and *hall monitor* at 30–60 kbps (0.14–0.28 bits per pixel).

There have been a number of attempts to incorporate motion compensation into 3D transform coding of video. Kronander [13], Ohm [14], and Choi [15, 16] use various methods to fill holes and treat overlapping motion trajectories created (mostly) by block matching motion compensation. Unfortunately, the coding gain due to these methods appears to be slight. Kim et al. [12] report at most 0.23 db gain, and on one sequence a substantial loss, when the best of these methods [15] is used with SPIHT. On the other hand, Taubman and Zakhor [17] use a simple pan compensation, and demonstrate a 0.56–1.29 dB gain compared to their 3D subband coder without motion compensation, on three sequences at various bit rates.

From a separate direction, a number of researchers have investigated a method of video coding based on layers, sprites, or mosaics [18, 19, 20, 21, 22]. A simplified version of this method is currently being incorporated into MPEG-4 [23]. Sprite video coding is particularly good for sequences in which the camera pans and/or zooms over a stationary background. In this method, the encoder constructs a panoramic image from the sequence of video frames by warping each frame into a common coordinate system, typically the coordinate system of the first frame of the sequence. The warping for each frame is represented by an invertible geometric coordinate transformation, such as an affine or perspective warping. The encoder then resamples the warped, registered frames, and stitches them together, such as by averaging or taking the median, to produce a single, large mosaiced image. The mosaiced image is then compressed using a still image coder, with the geometric transformations sent as side information. The decoder decompresses the image mosaic, and then uses the inverses of the geometric transformations to appropriately resample the image mosaic to reconstruct each frame in the video sequence. A variation is to use these

reconstructions as predictions for the original frames; the prediction residuals are then encoded and decoded as well [20, 21]. Such a scheme has been known to reduce the bit rate by a factor of 3–6 [22], or to increase the SNR by 1–4 dB [21], over traditional motion compensated predictive video on selected sequences.

In this paper, we take an approach that can be considered a generalization of the mosaic method, in the framework of 3D wavelet video coding. It can also be considered a generalization of 3D wavelet video coding, equipped with global affine motion compensation. Like the mosaic method, our approach has high gains on panning and zooming sequences, while performing similarly to 3D wavelet video coding on other sequences.

Essentially, our method is the following. As in the mosaic method, we warp each video frame into a common coordinate system. The sequence of warped frames thereby forms a warped 3D volume. However, rather than averaging the frames together at this point, we perform 3D wavelet coding over this volume, which has an arbitrary region of support in 3D. When the warped volume is decompressed, it is unwarped using the inverses of the geometric transformations which have been sent as side information. Using this method, compared to 3D SPIHT with no motion compensation, we gain an average of 0.56–1.02 dB on the 30fps QCIF sequence *coast guard* at 50–100 kbps, and gain an average of 1.93–2.42 dB on the 30 fps QCIF sequence *stefan* at 100–200 kbps. Our performance on these sequences is similar to H.263 (-0.57 dB to +0.50 dB), while featuring fine scalability.

The essential elements of this method are 1) how we perform the motion estimation and compensation, i.e., the warping and unwarping; 2) how we perform the 3D wavelet transform over an arbitrary region of support in 3D; and 3) how we perform the quantization and entropy coding of the wavelet coefficients over an arbitrary region of support in 3D. Section 2 addresses the first of these issues. A separate paper [24] addresses the second and third of these issues, which cannot be discussed fully in this paper due to lack of space. Section 3 provides experimental results. Section 4 concludes and suggests further avenues of research.

2 Global Motion Compensation

Global motion compensation is the process of warping an entire frame with a single coordinate transformation. This can be especially effective when compensating for global motion due to camera pan, rotation, and/or zoom relative to the scene as a whole, and is often combined with standard block matching motion compensation. Indeed, both H.263 Annex P and MPEG-4 version 2 provide a mechanism for global motion compensation within the standard [23]. In the standard, the global motion compensation parameters are quantized and sent as side information to the decoder so that the previously decoded frame can be warped and used as a reference frame to predict the current frame.

In this paper, we use global motion compensation to warp a sequence of video images prior to three-dimensional wavelet coding. The global motion compensation parameters are again quantized and sent as side information to the decoder, this time

so that the coded images can be unwarped prior to display.

Methods for estimating the global motion parameters for video coding have been well studied, e.g., [25, 26], as they are essentially the same as the methods for estimating motion flow for image mosaicing, e.g., [27, 22], and computer vision, e.g., [28, 29]. Most of the global motion estimation methods, however, are based on gradient descent. Since gradient descent methods are iterative, and subject to being caught in local minima, these methods are best used within a hierarchical coarse-to-fine optimization framework [30]. In this paper, we prefer to solve for the coefficients in a single shot, using least squares based on feature correspondances, which has essentially the same complexity as ordinary block matching motion estimation in traditional video coding. For this purpose, we restrict our attention to an affine motion model. However, planar pespective or even more complicated motion models can be used if computation permits.

Let $I_0(x,y), I_1(x,y), \ldots, I_T(x,y)$ be a sequence of video images, or frames, each having a coordinate system in which x increases to the right and y increases downwards. Between each pair of frames $I_t(x,y)$ and $I_{t-1}(x,y)$, for $t = 1, 2, \ldots, T$, let the affine coordinate transformation $A_t : (x_t, y_t) \rightarrow (x_{t-1}, y_{t-1})$ be the six-parameter transformation

$$\begin{aligned} x_{t-1} &= a_{xx}x_t + a_{xy}y_t + b_x \\ y_{t-1} &= a_{yx}x_t + a_{yy}y_t + b_y \end{aligned} \tag{1}$$

that warps the coordinate system of frame t into the coordinate system of frame $t-1$, such that the images $I_t(x,y)$ and $I_{t-1}(A_t(x,y))$ are optimally aligned in the squared error sense:

$$A_t = \arg\min_A \sum_{x_t, y_t} \left[I_t(x_t, y_t) - I_{t-1}(A(x_t, y_t)) \right]^2 . \tag{2}$$

The optimal alignment can be approximated with low computational cost, using standard block matching to establish a set of point correspondences, and using a linear least squares technique to solve for the affine parameters. The details of this procedure are too lengthy to describe here. However, many procedures for establishing this alignment will work as well.

Figure 1 shows the sequence of frames $I_t(x,y)$, $t = 0, 1, \ldots, T$ and the affine motion compensation transformations A_t, $t = 1, \ldots, T$ between them. With these transformations, which are invertible, one can warp the coordinate system of any frame into the coordinate system of any other frame. In particular, let $S_{t,0}$ be the composition $A_1 A_2 \cdots A_t$, which maps the coordinate system of frame t into the coordinate system of frame 0. Furthermore, let $S_{t,s}$ be the composition $S_{s,0}^{-1} S_{t,0}$, which maps the coordinate system of frame t into the coordinate system of frame s, for any s, t in $0, 1, \ldots, T$.

Figure 2 shows frames $t = 0, 1, \ldots, T$ mapped by coordinate transformations $S_{t,s}$ onto the coordinate system of some selected reference frame s. In the hypothetical video sequence from which this figure was derived, the camera panned to the right while rotating slowly clockwise, at first zooming in, and then out. Frame s is selected as the reference frame because its coordinate system depicts the scene to the highest

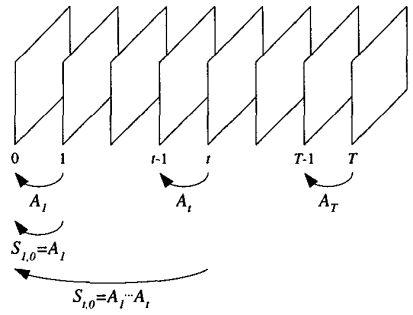

Figure 1: Sequence of unwarped frames and the affine motion compensation transformations between them.

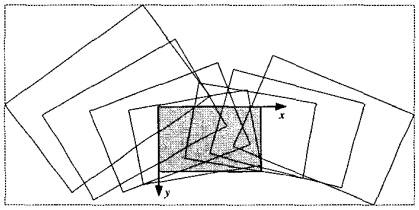

Figure 2: Sequence of frames warped into a common coordinate system (the coordinate system of frame s, shown in gray).

spatial resolution, where the camera was zoomed in. In other words, pixel resolution in the coordinate system of frame s is equivalent to sub-pixel resolution in the coordinate systems of the other frames. More precisely, a unit-area square pixel in frame s maps to a $\det(S_{s,t})$-area parallelogram in frame t, where $\det(S_{s,t}) \leq 1$. In fact, one way to choose the reference frame is to choose the frame s minimizing the determinant $\det(S_{s,t})$, for an arbitrary frame t. The selection of t is irrelevant, since if s minimizes $\det(S_{s,t})$, then it also minimizes $\det(S_{s,t'}) = \det(S_{s,t})\det(S_{t,t'})$, for any other t'. Thus selection of the reference frame is well-defined.

Once a reference frame s is selected, the six parameters a_{xx}, a_{xy}, a_{yx}, a_{yy}, b_x, and b_y of each coordinate transformation $S_{t,s}$, $t = 0, 1, \ldots, T$, must be quantized, for it is these quantized values that will be transmitted to the decoder and used by the decoder to invert the warping. A method for quantizing the parameters that makes the concomitant geometric warping error particularly robust to the parameter quantization error is to uniformly scalar quantize the coordinates (in the coordinate system of frame s) of three corners (totalling six parameters) of each frame [22]. It is easiest to use a simple fixed rate encoding with 16 bits per coefficient. We therefore allocate 96 bits per frame for global motion information. This is a small fraction of the overall bit rate.

Next it is necessary to choose a bounding box for all the frames in the coordinate system of the reference frame. Such a bounding box is shown as a dotted line in Figure 2. Once the bounding box in the reference coordinate system has been computed, a three-dimensional volume of pixels may be set up, with coordinates $x \in [x^{min}, \ldots, x^{max}]$, $y \in [y^{min}, \ldots, y^{max}]$, and $t \in [0, \ldots, T]$, to contain the warped frames. Let $J_t(x, y)$ denote a pixel in this volume, so that the image J_t is the warped and resampled image I_t in the coordinate system of frame s. More precisely, letting $[x_t, y_t, 1]' = \hat{S}_{t,s}^{-1}[x, y, 1]'$, if $(x_t, y_t) \in [0, W-1] \times [0, H-1]$, set $J_t(x, y)$ to the bicubic interpolation [31] of the point $I_t(x_t, y_t)$. Otherwise let $J_t(x, y)$ be undefined.

The defined pixels in the volume $J_t(x, y)$, $x \in [x^{min}, \ldots, x^{max}]$, $y \in [y^{min}, \ldots, y^{max}]$, $t \in [0, \ldots, T]$, now form a three-dimensional arbitrary region of support (3D AROS) within a rectilinear bounding volume. Perform a critically sampled wavelet decomposition, and quantize and code the resulting transform coefficients, as described in [24]. (Limited space prevents a fuller description here.) Pad the quantized values $\hat{J}_t(x, y)$ out from the 3D AROS to the entire volume, using a padding scheme such as in MPEG-4 [23].

Reconstruct the unwarped images $\hat{I}_t(x_t, y_t)$ again by bicubic interpolation, from the padded version of $\hat{J}_t(x, y)$. Letting $[x, y, 1]' = \hat{S}_{t,s}[x_t, y_t, 1]'$, $\hat{I}_t(x_t, y_t)$ is equal to the bicubic interpolation of the point $\hat{J}_t(x, y)$. The padding ensures that there will always be some context for the bilinear interpolation at the boundaries of the 3D AROS.

Finally, display the image sequence $\hat{I}_t(x, y)$, $t = 0, 1, \ldots, T$. In practice, to reduce delay, limit the number of frames to 16 or 32.

3 Experimental Results

Our 3D SPIHT coder was run on two standard test sequences that contain a substantial amount of global motion: stefan and coast guard. The frames were padded along the border by replication in order to remove the black lines in the original sequence. This was done to prevent artificial black edges from being coded in the warped volume. The H.263+ results were generated by the latest coder available, which as of this writing was version 3.2, based on ITU H.263+ Draft 21 and TMN9 documents.

The first 288 frames from stefan and coast guard were coded using both coders, and the PSNR of the luminance was calculated for each frame and averaged over the entire sequence. The results are shown in Tables 1 and 2.

	50 kbps	100 kbps
H.263+	26.71	29.64
3D-SPIHT with compensation	27.21	29.77
3D-SPIHT without compensation	26.65	28.75

Table 1: Results for Coast Guard, QCIF 30 fps (avg. luminance PSNR in dB)

	100 kbps	200 kbps
H.263+	24.66	27.50
3D-SPIHT with compensation	24.46	26.93
3D-SPIHT without compensation	22.53	24.51

Table 2: Results for Stefan, QCIF 30 fps (avg. luminance PSNR in dB)

The results indicate that at low bitrates, motion compensated 3D SPIHT performs about as well as H.263+, while still preserving a fully embedded syntax. Figures 3 and 4 show the luminance PSNR plot by frame number for the two sequences. The

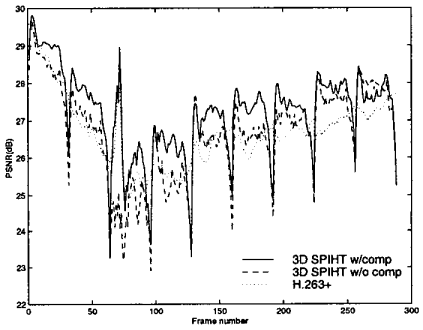

Figure 3: PSNR plot of luminance for Coast Guard at 50 kbps

sharp dips in PSNR for the 3D SPIHT coders are characteristic of 3D subband coders in general, and arise from the boundary effects in the temporal transform. This can be mitigated by normalizing the basis functions at the boundaries before coding.

It should also be noted that at the time these results were generated, the transform used over the arbitrary region of support was slightly oversampled. In addition, all of the coefficients were coded in the bounding rectangular volume, with those coefficients outside the region of support being set to zero. Fixing these inefficiencies should yield slightly improved results.

Computationally, our unoptimized encoder and decoder are at least an order of magnitude slower than a reasonably fast implementation of H.263+. The primary bottleneck is in the large number of floating point calculations needed to perform the high quality affine resampling used to warp the image. In terms of memory and delay, H.263+ only needs to store one to two frames for prediction. However, 3D SPIHT processes all the frames in a GOF as a single block, and thus will be unsuitable for low latency applications such as videoconferencing. The memory requirements are also substantially larger since the wavelet coefficients for a large spatio-temporal block must be stored in memory.

One interesting effect to note is that the compensated 3D SPIHT coder will typi-

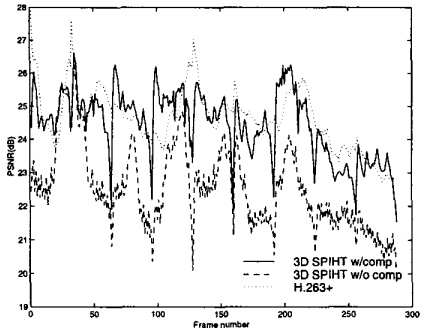

Figure 4: PSNR plot of luminance for Stefan at 100 kbps

cally code the dominant motion to higher fidelity than any subordinate motions. In fact, the compensation of the dominant motion frequently penalizes the coding of subordinate motions by decreasing the spatial overlap between successive frames of any subordinate objects. This causes the subordinate objects to decompose into a wideband temporal signal that contains energy in a large number of subbands. Although all subbands are quantized to the same stepsize during each pass, the compensated dominant object has more subbands which are well approximated by zero than the subordinate object. Thus, the subordinate objects contain more error energy because of the contribution of quantization error from a larger number of nonzero subbands. Since many subordinate motions tend to be foreground objects the viewer is interested in, this may be a particularly undesirable effect, and we will address this issue in future work.

4 Conclusion and Further Research

In this paper we have demonstrated a way to incorporate global motion compensation into three-dimensional subband coding of video. Three-dimensional subband video coders are important for producing finely layered codes, but heretofore their performance has suffered, relative to motion-compensated predictive video coders, on sequences in which motion is easily compensated, such as simple panning and rigid motion. We find that on such sequences, global motion compensated 3D subband coding performs competitively with the most recent version of H.263+, while still preserving a fully embedded syntax. On sequences with multiple rigid motions, global motion compensated 3D subband coding still appears to have performance somewhat inferior to that of standard predictive video coders. This can be addressed by segmenting and coding multiple motion flows, which is the subject of future research.

References

[1] B. G. Haskell and A. Puri and. *Digital Video: An Introduction to MPEG-2.* Chapman & Hall, New York, 1997.

[2] L. Yang, F. C. M. Martins, and T. R. Gardos. Improving H.263+ scalability performance for very low bit rate applications. In *Proc. Visual Communications and Image Processing*, San Jose, CA, January 1999. SPIE.

[3] J. M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. Signal Processing*, 41(12):3445–3463, December 1993.

[4] A. Said and W. A. Pearlman. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. Circuits and Systems for Video Technology*, 6(3):243–250, June 1996.

[5] G. Karlsson and M. Vetterli. Subband coding of video signals for packet switched networks. In *Proc. Visual Communications and Image Processing*, volume 845, pages 446–456. SPIE, 1987.

[6] G. Karlsson and M. Vetterli. Three dimensional subband coding of video. In *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, pages 1100–1103. IEEE, April 1988.

[7] J. W. Woods, editor. *Subband Coding of Video Signals.* Kluwer, Boston, 1990.

[8] F. Bosveld, R. L. Lagendijk, and J. Biemond. Hierarchical video coding using a spatio-temporal subband decomposition. In *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, volume 3, pages 221–224. IEEE, March 1992.

[9] A. Jacquin and C. Podilchuk. Very low bit rate coding with a dynamic bit allocation. In *Proc. Int'l Symp. Video Commun. Fiber Optic Services*, volume 1977, pages 156–167. SPIE, April 1993.

[10] C. Podilchuk, N. S. Jayant, and N. Farvardin. Three-dimensional subband coding of video. *IEEE Trans. Circuits and Systems for Video Technology*, 6(3):243–250, June 1996.

[11] B.-J. Kim and W. A. Pearlman. An embedded wavelet video coder using three-dimensional set partitioning in hierarchical trees (SPIHT). In *Proc. Data Compression Conference*, pages 251–260, Snowbird, UT, March 1997. IEEE Computer Society.

[12] B.-J. Kim, Z. Xiong, , and W. A. Pearlman. Very low bit-rate embedded video coding with 3D set partitioning in hierarchical trees (3D SPIHT). *IEEE Trans. Circuits and Systems for Video Technology*, October 1997. submitted.

[13] T. Kronander. *Some Aspects of Perception Based Image Coding.* PhD thesis, Linköping University, Linköping, Sweden, 1989.

[14] J.-R. Ohm. Three-dimensional subband coding with motion compensation. *IEEE Trans. Image Processing*, 3(5):559–571, September 1994.

[15] S. J. Choi. *Three-Dimensional Subband/Wavelet Coding of Video with Motion Compensation.* PhD thesis, Renssalaer Polytechnic Institute, Troy, NY, 1996.

[16] S. J. Choi and J. W. Woods. Motion-compensated 3-D subband coding of video. *IEEE Trans. Image Processing*, 1997. submitted.

[17] D. Taubman and A. Zakhor. Multirate 3-D subband coding of video. *IEEE Trans. Image Processing*, 3(5):572–589, September 1994.

[18] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Trans. Image Processing*, 3(5):625–638, September 1994.

[19] P. Anandan, M. Irani, R. Kumar, and J. Bergen. Video as an image data source: efficient representations and applications. In *Proc. Int'l Conf. Image Processing*, Washington, DC, October 1995. IEEE.

[20] M. Irani, P. Anandan, J. Bergen, R. Kumar, and S. Hsu. Efficient representations of video sequences and their applications. *Image Communication*, 8:327–351, 1996.

[21] F. Dufaux and F. Moscheni. Background mosaicking for low bit rate video coding. In *Proc. Int'l Conf. Image Processing*, Lausanne, September 1996. IEEE.

[22] M.-C. Lee, W.-G. Chen, C.-L. Lin, C. Gu, T. Markok, S. I. Zabinsky, and R. Szeliski. A layered video object coding system using sprite and affine motion model. *IEEE Trans. Circuits and Systems for Video Technology*, 7(1):130–145, February 1997.

[23] ISO/IEC. Information technology — coding of audio-visual objects: Visual. Final Committee Draft 14496-2, JTC1/SC29/WG11, Tokyo, March 1998.

[24] Z. Xiong, A. Wang, P. A. Chou, and S. Mehrotra. Three-dimensional wavelet coding of video with arbitrary regions of support. *IEEE Signal Processing Letters*, 1998. in preparation.

[25] K. Zhang and J. Kittler. Global motion estimation and robust regression for video coding. In *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, Seattle, WA, May 1998. IEEE.

[26] T. Wiegand, E. Steinbach, A. Stensrud, and B. Girod. Multiple reference picture video coding using polynomial motion models. In *Proc. Visual Communications and Image Processing*, San Jose, CA, January 1998. SPIE.

[27] R. Szeliski and H.-Y. Shum. Creating full view panoramic image mosaics and environment maps. In *Computer Graphics Proc., Ann. Conf. Series*, Los Angeles, CA, August 1997. ACM.

[28] L. S. Shapiro. *Affine Analysis of Image Sequences.* Cambridge University Press, Cambride, UK, 1995.

[29] Z. Zhang. Determining the epipolar geometry and its uncertainty: a review. *Int'l J. Computer Vision*, 27(2):161–195, 1998.

[30] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proc. European Conf. on Computer Vision*, pages 237–252, Santa Marguerita, Ligure, May 1992.

[31] R. G. Keys. Cubic convolution interpolation for digital image processing. *IEEE Trans. Acoustics Speech and Signal Processing*, 29(6):1153–1160, December 1981.