Figure 13: Transmission compression: results of dynamic MBC vs. MPEG compression of the table-tennis sequence.
(a) Some representative frames. The sequence was constructed of SIF525 size images, and temporally sampled by four (i.e., 7.5 frames/sec).
(b) The reconstructed frames after using dynamic MBC at 64 kbits/sec.
(c) For comparison: The reconstructed frames after using MPEG at 64 kbits/sec.

Figure 12: Transmission compression: results of dynamic MBC vs. MPEG compression on a chopper surveillance sequence.
(a) Some representative frames. The sequence was constructed of SIF525 size images, and temporally sampled by four (i.e., 7.5 frames/sec).
(b) The reconstructed frames after using dynamic MBC with a DCT back-end coder at an average bit rate of 16 kbits/sec.
(c) For comparison: The reconstructed frames after using MPEG compression of the sequence at *twice* the bit rate, i.e., 32 kbits/sec. Despite the higher bitrate, the reconstructed quality of MPEG is of significantly poorer quality than that of the dynamic MBC.

Figure 11: Transmission compression: results of dynamic MBC vs. MPEG compression on a storage-house surveillance sequence.
(a) Some of the sequence frames (50 frames apart).
(b) The reconstructed frames using dynamic MBC at 32 kbits/sec.
(b) The reconstructed frames using dynamic MBC at 16 kbits/sec.
(d) For comparison: The reconstructed frames using MPEG at 32 kbits/sec.

Figure 10: Static MBC compression of the 191–frame storage-house surveillanc sequence.
(a) Some representative frames of the sequence.
(b) The reconstructed frames using static MBC with an EZW back-end coder. An average compression ratio of 110 : 1 was computed for the *entire* sequence.
(c) The reconstructed frames MBC with an EZW back-end coder at an average compression ratio of 195 : 1.
(d) For Comparison: The reconstructed frames using JPEG applied to each individual frame at a compression ratio of 97 : 1.
(e) For Comparison: The reconstructed frames using interframe MPEG to subsequences of 15 frames (i.e., a new I-frame every 15 frames) at an average compression ratio of 97 : 1. Both JPEG and MPEG reconstructions are very blocky, and do not preserve edge information very well. In particular, note the soldiers approaching the house from the right.

Figure 9: Static MBC compression of the 270–frame table–tennis sequence *with* residuals.
(a) Some representative frames of the sequence.
(b) The reconstructed frames (*with* residuals) using MBC with an EZW (wavelet) back-end coder at an average compression ratio of 106 : 1.
(c) For Comparison: The reconstructed frames using JPEG applied to each individual frame at a compression ratio of 87 : 1.
(e) For Comparison: The reconstructed frames using interframe MPEG to subsequences of 15 frames (i.e., a new I-frame every 15 frames) at an average compression ratio of 94 : 1. Both JPEG and MPEG reconstructions are very blocky, and do not preserve edge information very well. Note the significant differences in the reconstruction quality of the reflections in the tennis table, the text in the pictures on the walls, and the player's arm and bat.

Figure 8: Static MBC compression of the 270–frame table–tennis sequence *without* residuals.
(a) The static mosaic image constructed using a temporal median.
(b) The reconstructed static mosaic image using EZW compression on the mosaic of Figure 8.a with a compression ratio of 13 : 1.
(c) Some representative frames of the sequence.
(d) The predicted frames from the reconstructed mosaic of Figure 8.b *without adding the coded residuals*. This gives an effective compression ratio of 1260 : 1 for the *entire* sequence.
(e) The significance measure maps of the residuals.

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

[31] A. Zakhor and F. Lari. Edge-based 3-D camera motion estimation with application to video coding. *IEEE Trans. on Image Processing*, 2(4):481–498, October 1993.

[32] H.-J. Zhang, A. Kankanhalli, and S.W. Smoliar. Automatic partitioning of full-motion video. *Multimedia Systems*, 1(1):10–28, 1993.

[14] R. Koch. Dynamic 3d scene analysis through synthesis feedback control. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(6):556–568, June 1993.

[15] R. Kumar, P. Anandan, and K. Hanna. Direct recovery of shape from multiple views: a parallax based approach. In *Proc 12th ICPR*, pages 685–688, October 1994.

[16] Rakesh Kumar, P. Anandan, and K. Hanna. Shape recovery from multiple views: a parallax based approach. In *DARPA IU Workshop*, Monterey, CA, November 1994.

[17] J. Lee and B.W. Dickinson. Temporally adaptive motion interpolation exploiting temporal masking in visual perception. *IEEE Trans. on Image Processing*, 3(5):512–526, September 1994.

[18] D. LeGall. Mpeg: A video compression standard for multimedia applications. *Commun. ACM*, 34(4):46–58, April 1991.

[19] S. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989.

[20] P.C. McLean. Structured video coding. Master's thesis, MIT, June 1991.

[21] H. Miyamori, T. Hanamura, and H. Tominaga. Processable video coder on camera operations; PVCC. In *Proc. International Picture Coding Symposium*, pages 333–336, September 1994.

[22] H.G. Musmann, M. Hoetter, and J. Ostermann. Object–oriented analysis–synthesis coding of moving images. *Signal Processing: Image Commun.*, 1(2):117–138, 1991.

[23] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. The QBIC project: Querying images by content using color, texture, and shape. Technical Report RJ 9203 (81511), IBM, February 1993.

[24] Harpreet Sawhney. 3d geometry from planar parallax. In *Proc. CVPR 92*, June 1994.

[25] J.M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. on Signal Processing*, 41(12):3445–3462, December 1993.

[26] R. Szeliski. Image mosaicing for tele-reality applications. Technical Report CRL 94/2, DEC Cambridge Research Lab, May 1994.

[27] L. Teodosio and W. Bender. Salient video stills: Content and context preserved. In *Proc. ACM Multimedia Conf.*, 1993.

[28] Jr. V.M. Bove, B.D. Granger, and J.A. Watlington. Real-time decoding and display of structured video. In *Proc. IEEE ICMCS*, May 1994.

[29] G.K. Wallace. The jpeg still picture compression standard. *Commun. ACM*, 34(4):30–44, April 1991.

References

[1] E.H. Adelson. Layered representations for image coding. Technical Report 181, MIT Media Lab. Vision and Modeling Group, December 1991.

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

[3] J.R. Bergen, P.J. Burt, R. Hingorani, and S. Peleg. A three frame algorithm for estimating two-component image motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14:886–896, September 1992.

[4] R.M. Haralick, S.R. Sternberg, and X. Zhuang. Image analysis using mathematical morphology. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(4):532–550, July 1987.

[5] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1–3):185–203, August 1981.

[6] S. Hsu, P. Anandan, and S. Peleg. Accurate computation of optical flow by using layered motion representations. In *Proc. 12th ICPR*, pages 743–746, October 1994.

[7] M. Irani and S. Peleg. Improving resolution by image registration. *CVGIP: Graphical Models and Image Processing*, 53:231–239, May 1991.

[8] M. Irani and S. Peleg. Image sequence enhancement using multiple motion analysis. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 216–221, June 1992.

[9] M. Irani, B. Rousso, and S. Peleg. Detecting and tracking multiple moving objects using temporal integration. In *European Conference on Computer Vision*, pages 282–287, June 1992.

[10] M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. *International Journal of Computer Vision*, 12(1):5–16, February 1994.

[11] ISO/IEC. 11172: Coding of moving pictures and associated audio—for digital storage media at up to about 1.5 mbit/s.

[12] ISO/IEC/JTC1/SC29/WG11. Coded representation of picture and audio information, Test Model 5, April 1993.

[13] ISO/IEC/JTC1/SC2/WG10. 11918-2: Digital compression and coding of continuous-tone still images.

7 Concluding Remarks

We have presented a technique for video compression, based on a *mosaic image* representations, and have demonstrated its benefits for transmission and database storage applications. We have applied the technique (in software simulation) to several real video examples of surveillance imagery and other types of scenarios. In each case, we have shown that the dynamic MBC provides usable output video at very-low bitrates (32 kbps or less) in comparison to MPEG which does not. We have also demonstrated that static MBC provides usable pictures at compression ratios of 100:1. Static MBC is suited for storage applications, since it provides the advantage of random access to individual frames and supports rapid browsing and editing functions, At these types of compression ratios, JPEG (which is the current choice for many editing systems) and MPEG provide poorer quality pictures than MBC.

The compression efficiency achieved in the current implementation is due to a combination of (i) parametric motion models to achieve accurate and efficient representation of interframe motion, (ii) mosaics to exploit long-term temporal correlations in the image sequence, and (iii) significance based coding of residuals.

We believe that the parametric motion models always provide a significant benefit. There are two reasons for this: first, they encode the interframe more compactly than local flow or block motion vectors, and second they provide a more accurate and numerically stable pixel-wise motion compensation than most other representations. Long-term temporal correlations is high when the camera is surveying the same scene for an extended period of time, although due to its limited field-of-view the camera may be panned and zoomed over the various portions of that scene during the course of that time period. Significance based coding allows us to introduce semantic information, albeit in a limited fashion in the current version. Our current approach to significance analysis is based on the observation that residual motion is often a reliable way of distinguishing between real changes and noise. On the other hand, our overall approach to coding does not forbid the use of other measures (e.g., based on human vision models).

The MBC approach using 2D mosaics is currently under implementation on real-time hardware. We have also described three extensions to the 2D mosaic representations that are currently under investigation: parallax-based mosaics, layered mosaics, and the temporal pyramid of residuals. We believe these will further increase the compression efficiency of MBC.

6.3 Temporal pyramid of residuals

As discussed in Section 4, static MBC does not remove as much short–term temporal redundancy as interframe prediction. For example, when some object appears in the scene but not in the mosaic, similar residuals will be coded repeatedly in a number of frames. The static MBC scheme could be extended to use a hierarchy of mosaics whose levels correspond to different amounts of temporal integration. The finest level contains the original images and other levels contain static mosaics integrated from images or mosaics of the next finer level. Only the residuals between levels need be coded and stored. The result is a representation which achieves temporal decorrelation in a manner analogous to spatial Laplacian pyramids.

2. Compute the residual parallax motion of the objects and surfaces, but while doing so use the constraint that all camera rotations and other camera induced deformations have been removed and only the effects of translation remain. This will increase the efficiency and accuracy of the estimation process. (This approach has been implemented and is described in detail in [15].)
3. Decompose the residual motions into "heights" and camera translation.
4. Integrate the height information estimated from multiple frames into a single height map mosaic. This can be done either in a batch mode (suitable for static mosaics) or recursively over time (suitable for dynamic mosaic).

To predict any given frame from the intensity mosaic and the height mosaic, both the 2D global motion transformation between the mosaic and the current frame, and the residual 3D motion computed based on the height information should be compensated. The resulting prediction image will resemble the original frames, except for other changes such as moving objects.

We have partially implemented this approach and have obtained preliminary results. However, the integration of the 3D analysis with the spatial coding process is still under way. It is worth noting that this approach to 3D representation does not require significant image understanding and segmentation of the scene into objects. While the ability to do such segmentation will further enhance the compression efficiency, in practice, such a task is quite complex, and as yet unsolved for natural scenes. By relying on simple point-wise parallax representations of 3D scene structure, we can extend the 2D mosaic representations to handle more complex range of scenes in the near term.

6.2 Multiple motion analysis and layered representations

Another extension to the 2D mosaic is to use multiple layers of 2D mosaics in the manner suggested by Adelson[1]. Each layer can either represent a different moving object or may represent surface at a different 3D depth. The benefits of multiple motion analysis and layered representation has been previously described in [1, 8, 3, 30, 6].

The layered approach is particularly advantageous when the scene contains a significant depth of field, where it will provide further efficiency beyond the parallax based representation. Since each 2D layer captures a dominant surface, additional parallax information should be maintained relative to each layer to complete the 3D representation.

We have currently developed algorithms for multiple motion segmentation[8] and layered motion recovery [30, 6]. We plan to combine these with the 2D mosaics, the 3D representations described above, and the spatial coding techniques in order to develop complete end-to-end simulations using all the ideas described here.

6 Extensions to the 2D Mosaic Representation

The 2D mosaic representation aligns a single surface, usually the dominant surface in the scene. As noted in Section 4.3, the residual changes with respect to that surface may either be due to illumination or scene changes, or due to residual motion (ignoring alignment errors and noise). Of these two categories, the residual motions may arise either because of the parallax motion of objects relative to the mosaic surface or due to independent object motion. To handle residual motions, two extensions of the 2D mosaic are proposed, one involving 3D "height" representation, and the other involving multiple-motion analysis leading to a "multi-layered" representation[1] that will efficiently represent and code these two types of motion-based residuals. In one sense, the static and dynamic mosaics are extremes of a continuum. One uses a completely static image, which may be based on an arbitrarily long sequence, and predicts each frame. In this case, there may be an arbitrarily long time interval between the current frame and the static mosaic. The other approach is completely current and always has the most recent available information. In order to bridge the gap between these two extremes, we propose a third extension based on a "temporal pyramid" representation.

6.1 Parallax based 3D representation

Our approach to 3D representation is based on the observation that the 2D alignment of a dominant surface removes all the motions due to camera rotations and other camera induced deformations (e.g., changes in zoom, lens distortion, etc.) as well as the effects of camera translation on the dominant surface. What remains is purely due to the combination of camera translation relative to the scene and the distance of the other objects and surfaces in the scene from the dominant aligned surface[15, 24].

Thus, the 3D structure of the scene directly appears in the form of a "parallax" motion of the objects and surfaces in the scene relative to the dominant surface. Since the 3D structure is usually invariant over time (at least over the duration of several seconds or minutes), it can be represented as a single static image that can be used to predict the parallax induced motion over that duration. We refer to the parallax based 3D representation as a "height" map, since it can be shown to be directly related to the distance or "height" of objects and surfaces relative to the dominant surface. This height map together with the computed translational motion of the camera between successive frames can be used to predict the motion of all the points between frames, relative to the mosaic surface.

The computation of the 2D mosaic representation plus 3D height information involves the following steps:

1. First align each image to the mosaic coordinate system using 2D alignment.

poorer visual quality. In both cases (dynamic-MBC and interframe-MPEG), only the first frame was intracoded, no bidirectional frames were employed, and rate control followed MPEG–2 Test Model 5.

The figure shows the 70th, 120th, and 170th predicted frames. The same initial I-frame was used for both the dynamic MBC at 32 kbits/sec and for MPEG at 32 kbits/sec. Note that the MBC algorithm preserves many of the details of the images even after 170 frames, whereas the MPEG based reconstruction gradually degrades in quality and after about 50 frames, only the gross outlines of the building are visible. In particular, the rightmost frame in the figure (frame 170) contains images of soldiers approaching the house from the right. For this example, the activities of these people are critical pieces of information. Note that MBC preserves this detail, whereas it is completely lost in the MPEG based reconstruction.

Another surveillance sequence showing a helicopter has frames of size 352×240 (i.e., SIF525), and was *temporally* sampled by four (i.e., 7.5 frames/sec). Since the sequence contained only 15 frames, it was played in palindromic mode (i.e., back and forth). The sequence shows a flying chopper viewed from a camera mounted on another flying chopper. There are two different types of motions in this sequence: that of the background (due to camera motion), and that of the imaged chopper. Figure 12.a shows some representative frames from the sequence. Figure 12.b shows the reconstructed image frames after using the dynamic MBC coding scheme at 16 kbits/sec.. For comparison, the sequence was compressed by MPEG (without MBC) at 32 kbits/sec (we couldn't bring our current version of MPEG to run at a lower bit rate). The corresponding reconstructed frames are shown in Figure 12.c. The MPEG reconstruction is of significantly poorer visual quality than that of dynamic MBC.

The table–tennis sequence was *temporally* sampled by four (i.e., 7.5 frames/sec) and coded at the average bit rate of 64 kbits/sec using MPEG–2 Test Model rate control [12]. The first frame was intracoded, and all other frames were forward predicted (no B frames). Chrominance was dropped. Figure 13.b shows the reconstructed image frames after using dynamic MBC coding. For comparison, the sequence was compressed by MPEG (without MBC) at the same bitrate (Figure 13.c), which results in poorer visual quality.

differences (between the original frames and the frames predicted from the mosaic image) are determined as insignificant residuals by our significance measure.

MBC is particularly suited for these types of sequences, as the objects in the scene which move with respect to the background consume only a small portion of each frame, and the camera revisits the same potions of the scene repeatedly. Therefore, an attempt was made to compress the sequence even further. Figure 10.c shows the reconstructed image frames after using static MBC coding scheme (with an EZW back-end coder) at an average compression ratio of 195 : 1. (The mosaic image itself was coded using EZW coding at 10 : 1. The additional compression was achieved by allotting less bits to code the residuals with respect to the mosaic). The additional compression still maintains good image quality.

For comparison, Figure 10.d shows the reconstructed image frames after applying JPEG coding to each individual frame at a compression ratio of 97 : 1. This was the highest compression ratio achieved by our version of JPEG. Figure 10.e shows the reconstructed image frames after using MPEG coding with a new I-frame every 15 frames at an average compression ratio of 97 : 1 (The I-frames were compressed at an average compression ratio 50 : 1). Once again, this was the highest compression ratio that could be achieved by our version of MPEG.

5.2 Transmission Codec Results

In the development of dynamic MBC for transmission applications, both the DCT–based coder and the wavelet–based coder were evaluated for compressing the first intra frame and subsequent residual frames. Since our MPEG system includes local motion compensation, exploiting temporal correlations of objects moving with respect to the background motion, it proved to be more efficient than the EZW coder in the dynamic case, which in our current implementation did not include motion compensation. Therefore, the examples shown in this section display results using MPEG as a back–end coder.

The coding efficiency of MBC with any back-end coder will generally be greater than the efficiency of back-end coder alone. At *very low bitrates*, however, the improvement of MBC coding over MPEG coding is particularly striking. In surveillance applications, often only very low bitrate channels are available. Moderate image quality and motion rendition are acceptable, as long as objects of interest are recognizable.

Accordingly, the original 764-frame storage-house surveillance sequence (Figure 11.a) was sampled at 7.5 frames/sec, yielding 191 frames of size 320×208. Once again, chrominance was dropped. Figure 11.b shows the reconstructed image frames after using the dynamic MBC coding scheme at 32 kbits/sec. Figure 11.c shows the reconstructed image frames after using the dynamic MBC coding scheme at 16 kbits/sec. For comparison, the sequence was compressed by interframe MPEG (without MBC) at 32 kbits/sec (Figure 11.d), which resulted in significantly

to predict all the frames in the sequence according to the computed motion (alignment) parameters. The residuals were computed between the original frames (Figure 8.c) and the predicted frames (Figure 8.d). The corresponding significance measures of residuals (see Section 4.3) are shown in Figure 8.e .

Applying intra-frame compression to the static mosaic alone, *without* compressing any residuals, yields a reconstructed sequence that conveys the effect of camera motion only (Figure 8.d, i.e., without the residual motion of object with respect to the background) at an effective compression ratio of 1260 : 1 for the entire 270 frame sequence (corresponding to an even higher compression ratio if chrominance were not dropped). However, in most applications a more reliable reconstruction of the sequence is required, and therefore residuals must be coded as well.

Figures 9.b shows the reconstructions after using the static MBC compression scheme with a wavelet (EZW) back-end coding. The overall observed compression ratios for the *entire* sequence, *after dropping chrominance* was 106 : 1. Also the coding was applied to images at half their original resolution. Applying static MBC to chrominance as well would increase the *overall compression ratio*. Using full resolution images would yield further increase in compression ratios.

For comparison, the sequence was compressed by intraframe JPEG applied frame-by-frame to the entire sequence, and by interframe MPEG applied to groups of 15 frames. We tried achieving similar compression ratios (i.e., 106 : 1), however, our existing versions of MPEG and JPEG could not obtain such high compression ratios. Figure 9.c shows the reconstructed frames from intraframe JPEG compression at 87 : 1. Results obtained using MPEG at an average compression ratio of 94 : 1 are shown in Figure 9.d. (The I-frames were compressed at an average compression ratio of 26 : 1) Both JPEG and MPEG reconstructions are very blocky, and do not preserve edge information very well. Note the significant differences in the reconstruction quality of the pictures on the walls, the reflections in the tennis table, and the player's arm and bat.

Figure 10 shows results of static MBC applied to a surveillance sequence. Figure 10.a displays some representative frames of the sequence, which shows monitoring of outdoor activity at a storage-house from a distant airborne platform, in which frequent camera vibrations induce undesirable image shifts. The sequence contains 191 frames of size 320×208. Once again, chrominance was dropped. Figure 10.b shows the reconstructed image frames after using static MBC coding scheme with an EZW back-end coder at an average compression ratio of 110 : 1. (The mosaic image itself was coded using EZW coding at 10 : 1). Note that the reconstructed images of the storage house are less noisy than the original ones. The noise reduction is due to the temporal integration process of the aligned frames when constructing the mosaic. These

5 Experimental Results

In this section results are provided for compressing sequences for storage and transmission applications. The results shown here use MPEG as the spatial coding component for the MBC *transmission* coder and an EZW (wavelet) coding for the spatial coding component of the MBC *storage* coder (this portion of the MBC codec is referred to as the "back–end coder"). The results are compared to results obtained by applying MPEG compression (in transmission) and intraframe JPEG compression (in storage) of the same sequences. The performance of MBC for storage was also compared to interframe MPEG coding over short groups of pictures (GOP), i.e., using an intra-coded frame (I-frame) every 15 frames. This GOP size was selected in order to make retrieval from a database efficient in MPEG coding than a full sequential reconstruction would provide (i.e., with a single I-frame). The results shown in this section demonstrate that the compression efficiency of MBC is generally higher than these other methods for both transmission and storage.

5.1 Storage Codec Results

In the development of the static MBC for storage, both EZW and DCT were evaluated as the spatial back-end coders. In all cases the EZW gave better picture quality at the selected compression ratios than DCT (which had block artifacts). Hence, only the results using the EZW back-end coder are shown here. The EZW coder was used for compressing both the mosaic and the significance–weighted residuals, which in many regions were zero (i.e., below threshold) so that no wavelet coefficients needed be stored. We used only *global* parametric motion compensation for the storage examples, and these motion (alignment) parameters were quantized to about 50 bits/frame. Chrominance was dropped in our experiments. The compression ratios for static MBC were computed as follows:

$$\frac{\text{volume of sequence}}{\text{volume of coded mosaic} + \text{volume of coded residuals} + \text{alignment prms}} \tag{4}$$

The static mosaic image itself, without the residuals, is useful for video browsing. The entire sequence can be reconstructed from the static mosaic image to convey the effect of camera motion, without the residual object motions. Applying intraframe compression to the static mosaic *alone* yields a very large effective compression ratio for the entire sequence in such an application (as the denominator in Equation (4) is small since *volume of coded residuals* = 0).

Figure 8 shows such an example. The table–tennis sequence has 270 frames of size 352×240 (i.e., SIF525). The static mosaic shown in Figure 8.a was compressed at a moderate intra-frame compression ratio ($13:1$). The reconstructed mosaic image, shown in Figure 8.b, was then used

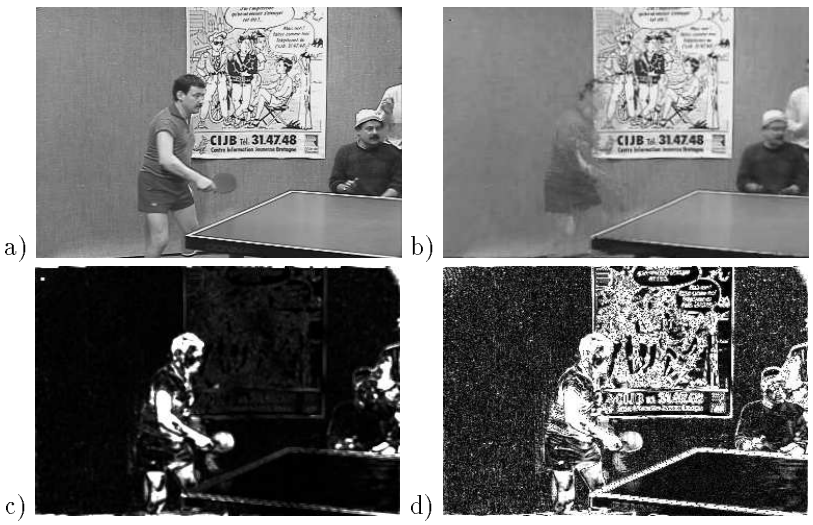

Figure 7: The significance measure map.
a) An actual frame in the table-tennis sequence.
b) A predicted frame from the static mosaic displayed in Figure 2.
c) The significance map $S_t(x, y)$ computed for the residuals between images 7.a and 7.b. The brighter values signify more significant values. The significance map indeed captures regions where significant changes have occurred, while it is relatively *insensitive* to noise and gray-level interpolation errors, e.g., on the background wall and posters.
d) The absolute difference taken between images 7.a and 7.b. The brighter values correspond to larger differences. Despite the image registration, large gray-level differences occur on stationary edges, where interpolation errors were introduced in the alignment process (e.g., on the background wall and posters). Images 7.c and 7.d were stretched to the same gray-level range, for comparison purposes. This demonstrates the benefits of the significance measure S_t over simple differencing.

static case, there are more residuals to be coded, yet, on the other hand, it is easier to distinguish between significant residuals (i.e., due to moving objects) and nonsignificant residuals (e.g., due to noise, interpolation error, or slight misregistration due to low order parametric modeling). Therefore, the threshold applied to the significance mask for the block selection can be set higher in the static case than in the dynamic case. Furthermore, if the change in some neighborhood is large, coding that area directly (intraframe) without prediction is an option that may be more efficient.

For the results shown in this paper, fixed significance thresholds were chosen heuristically, and transmission or storage bitrate was controlled by varying quantization step size. However, significance–based coding provides a supplementary control parameter, namely the effective threshold. Pixels or blocks could be prioritized and coded in order of highest to lowest significance until the target bitrate is used up. The resulting embedded bitstream would support progressive transmission and precise rate control, without the need to heuristically choose a threshold on the significance measures.

Simple image differencing between the predicted and actual image does *not* prove to be a good way of determining semantically significant residuals, since it highlights interpolation errors and noise as well as the first two types of residuals. An effective way of determining semantically significant residuals is to consider not only the residual intensity but also the magnitude of *local residual motions* (i.e., the local misalignments) between the predicted frame and the actual frame. Note that the residual motion computation need *not* be accurate, as it is itself not used for motion compensation, but rather used to determine the significance of the residuals for coding purposes.

To approximate the magnitudes of the residual motions, a rough estimate $S_t(x,y)$ of the normal flow magnitude at each pixel (x,y) at time t is computed. (The normal flow is the component of the optical flow in the direction of the spatial gradient [5].)

$$S_t(x,y) = \frac{\sum_{(x_i,y_i)\in N(x,y)} \left| I_t(x_i,y_i) - I_t^{Pred}(x_i,y_i) \right|}{\sum_{(x_i,y_i)\in N(x,y)} |\nabla I_t(x_i,y_i)| + C} \tag{3}$$

where:

I_t is the frame at time t.

I_t^{Pred} is the predicted frame from the mosaic at time t.

$\nabla I_t(x,y)$ is the spatial intensity gradient at pixel (x,y) in frame I_t.

$N(x,y)$ is a small neighborhood of pixel (x,y) (typically a 3×3 neighborhood).

C is used to avoid numerical instabilities and to suppress noise.

Figure 7.c shows the computed significance map $S_t(x,y)$ of residuals between a predicted frame (from the static mosaic of Figure 2) and the actual frame in the table–tennis sequence. The absolute difference of the two images is shown in Figure 7.d (stretched to the same gray–level range as Figure 7.c), to convey the advantage of our significance measure over simple differencing.

In areas within the image where $S_t(x,y)$ is above some threshold, the prediction error is weighted by $S_t(x,y)$ (i.e., multiplied by $min(S_t(x,y),1)$); otherwise, the error is zeroed out. To decrease noise sensitivity, the significance mask $S_t(x,y)$ undergoes morphological opening by a 3×3 structuring element [4]. When block based spatial coding like DCT is applied to the prediction errors, the maximum significance value in each block compared to a threshold is used to select blocks to be ignored (by forcing all weights to 0) or to be encoded.

Significant prediction errors are usually larger when predicting from the static than the dynamic mosaic, since there is less temporal correlation—both because objects change and because moving objects tend to blur out or even disappear in the static mosaic. Therefore, in the

Figure 6.a illustrates the storage codec using static mosaic for storage applications. The sequence is processed in batch mode, with these major steps:

1. For each pair of successive frames, the motion model needed to align them is computed using the parametric motion estimation algorithm (see Section 3.1).
2. A key frame is chosen to fix a coordinate system for the static mosaic, and the motion models are numerically composed into a set of motion models that establish the correspondence between the mosaic and each source frame.
3. All source frames are warped to the mosaic coordinate system using their motion models and integrated to form a static mosaic image (Section 3.2).
4. The static mosaic and motion models are then compressed and stored.
5. Each pixel of an individual frame to be coded is predicted from the corresponding pixel of the coded mosaic.
6. If desired, local nonparametric motion compensation is estimated and applied to this prediction.
7. The residuals $\Delta I_t = I_t - I_t^p$ are computed, and are analyzed for their "significance".
8. Only the significant residuals are coded and stored, these also being weighted and prioritized according to the measure of significance.

The receiver adds decoded residuals to predictions based on the decoded static mosaic to form reconstructed frames.

4.3 The Significance Measure of Residuals

In this section we describe the significance measure used for weighting the residuals before coding them. This measure is used in both the transmission codec and the storage codec. Residuals between the current frame and the mosaic–based predicted frame occur for several reasons:

- Object and illumination change.
- Misalignments that remain after motion compensation. This can be caused by moving objects or higher order global motion components that are not represented by the parametric plus local nonparametric motion model.
- Intensity errors that occur due to the gray level interpolation performed while warping the images in the alignment process. Those errors will be largest in edge pixels.
- Noise.

Of these four types of errors, for very–low bit rate and high–compression ratio applications, it is more important to correct and code residuals of the first two types.

(ii) or (iii) often becomes zero, thereby forcing that mode to be chosen and requiring no DCT coefficients to be sent. A penalty on the residual of mode (iii) is added to compensated for the bitrate cost of motion vectors. The parametric motion model is coded by uniform quantization of the translations of N points (3 for affine), and block motion vectors are coded in the usual MPEG manner.

Wavelet compression exploits the sparseness of image energy in scale space (position/frequency signal representation). In EZW specifically, a map of insignificant coefficients in the discrete wavelet transform of an image is efficiently described by a quadtree code that accounts for statistical dependence between such locations. EZW is a generic spatial compression algorithm and could be applied to intraframe or residual images. Since EZW does not intrinsically provide support for motion compensation or coding of motion vectors and mode selections, the wavelet–based dynamic MBC coder presented in this paper employs only parametric global motion compensation. (Coding of, and prediction using, nonparametric motion representations other than block–wise translation is presently under development for dynamic MBC.)

Since the spatial coders that are used to code the residuals after mosaic based motion–compensated prediction can literally be any algorithm, the MBC approach provides a natural extension to existing (or future) inter– or intra– frame coders. As a result, the coding efficiency of using MBC with such codecs will generally be greater than the efficiency of using these codecs alone. The magnitude of this increase depends not only on the amount of large scale spatiotemporal correlation available in the image sequence (e.g., in the form of global motions and camera revisits), but also on the accuracy of significance analysis. Our current approach to significance analysis described in Section 4.3 is based on the observation that residual motion is often a reliable way of distinguishing between real changes and noise. On the other hand, our overall approach to coding does not forbid the use of other measures (e.g., based on human vision models).

As is true of any temporal prediction and update scheme, if a gradually changing region (due to motion or background change) changes too slowly to be detected as being significant during a short interval, its accumulated change over a longer interval will be large. Hence, it will be eventually detected as significant and be updated.

4.2 The Storage Codec

For storage applications, the coding process need not be done in real time and can be done in an offline mode (although the sheer volume of data demands that the processing algorithms be efficient). This allows considerably greater flexibility in the techniques representation and coding, specifically allowing direct access to stored frames.

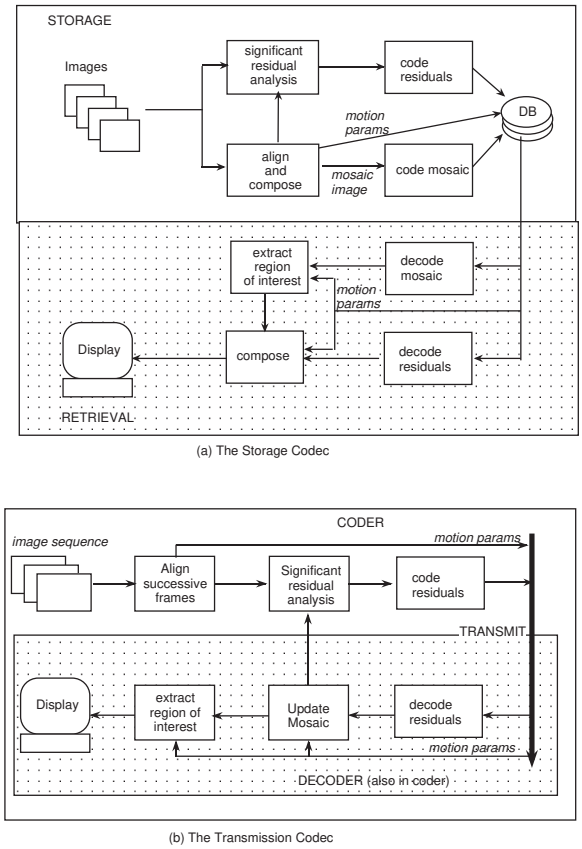

Figure 6: Codecs for static and dynamic mosaic-based compression.

7. The decoded residuals are added to the prediction, providing the current reconstructed frame. They are also used to update the mosaic.

The decoder performs the same steps, except motion estimation and significance analysis are omitted and display of the reconstructed frame is added.

Spatial decorrelation of the significance–weighted residuals can be achieved by a variety of techniques. For the purposes of this paper, we have used two different types of spatial compression, DCT and wavelet. Specifically, DCT coding used MPEG, and wavelet coding used the embedded zerotree algorithm (EZW) described in [25].

Since MPEG readily supports block–based motion compensation and adaptive block mode selection, the DCT–based dynamic MBC coder employs a motion representation consisting of parametric global model plus block–wise local translation. Each macroblock in a P frame (forward predicted frame) is coded in one of three possible modes: (i) it may be intracoded, (ii) it may be predicted by parametric motion compensation only, (iii) it may be predicted by both parametric and block motion compensation. The mode is chosen based on lowest overall significance–weighted residual. As explained in Section 4.3, the weighted residual for mode

4 The Use of Mosaics for Compression

In this section we detail the steps used for compression of an image sequence using the mosaic representations. The differences between static and dynamic mosaic representations that were outlined in the previous section lead to differences in the two types of codecs. However, they have a number of features in common, since they are both based on constructing a mosaic image data structure.

The static mosaic, or the first frame in the dynamic case, may be compressed by any known method for lossy still image coding. All subsequent frames to be coded are predicted by motion compensation from the static or dynamic mosaic. The residuals are compressed using a lossy spatial coder based on semantic and perceptual criteria. A significance measure is used to mask out image regions that are accurately predicted from the mosaic, and to weight and prioritize the residuals in other regions before coding them. The same significance measure is used in both codecs, and is described in Section 4.3.

The motion representation contains a parametric model for the global motion component, possibly augmented by a nonparametric field for local motion deviations from the model. The nonparametric part could be optical flow, a deformable mesh, block–wise translations (or 3D height, see Section 6). For each frame such motion information, coded in a lossless or lossy manner, needs to be stored or transmitted along with the residuals.

4.1 The Transmission Codec

Figure 6.b illustrates the codec using dynamic mosaic for transmission applications. The major steps in the dynamic coding are:

1. The motion parameters between the previous frame to the current frame are computed using the parametric motion estimation algorithm outlined in Section 3.1. The output of this process is a set of motion parameters (6 or 8, depending on the model chosen).
2. The current motion parameters are numerically composed with the previous ones to establish the correspondence between the current frame and the dynamic mosaic image.
3. The current frame is predicted from the mosaic image.
4. If desired, local nonparametric motion compensation is estimated and applied to this prediction.
5. The residuals $\Delta I_t = I_t - I_t^p$ are computed, and are analyzed for their "significance". The significance measure is described in Section 4.3.
6. Only the significant residuals are coded and transmitted, these also being weighted and prioritized according to the measure of significance.

2. A temporal median filtering of the intensity values of the aligned images. Both a temporal average and a temporal median applied to a registered scene sequence will produce a panoramic image of the dominant "background" scene, where moving objects either disappear or leave "ghost-like" traces. Temporal averages usually result in blurrier mosaic images than those obtained by temporal medians. (e.g., see Figure 2).

3. A *weighted* temporal median or a *weighted* temporal average where the weights decrease with the distance of a pixel from its frame center. This scheme aims at ignoring alignment inaccuracies near image boundaries due to the use of low order 2D parametric transformations (especially when the field of view is wide).

4. A weighted temporal average where the weights correspond to the outlier rejection maps computed in the motion estimation process of the dominant "background" (Section 3.1). This scheme prefers the dominant "background" data over "foreground" data in the mosaic construction, and therefore gives less "ghost-like" traces of "foreground" objects, and a more complete image of the dominant "background" scene.

5. Alternative integration schemes for image enhancement, such as Super-resolution [7], to produce mosaic image whose resolution and image quality surpasses those of any of the original image frames.

Besides being an exact description of the instantaneous motion field of a quadratic surface, this transformation also describes well the 2D image motion of an *arbitrary* 3D scene undergoing camera rotations, zooms, and small camera translations. It also approximates well the 2D image motion under larger camera translations, when the overall 3D range (Z) to the scene is much greater than the variation of the range within the scene (ΔZ).

The objective function E given in Equation (1) is minimized via the Gauss-Newton optimization technique. Let $\mathbf{p}_i$ denote the current estimate of the quadratic parameters. After warping the inspection image (towards the reference image) by applying the parametric transformation $\mathbf{p}_i$ to it, an incremental estimate $\delta\mathbf{p}$ can be determined. After iterating certain number of times within a pyramid level, the process continues at the next finer level.

With the above technique, the reference and inspection images are registered so that the desired image region is aligned. The above estimation technique is a least-squares based approach and hence possibly sensitive to outliers. However, as reported in [3] this sensitivity is minimized by doing the least-squares estimation over a pyramid. The pyramid based approach locks on to the dominant image motion in the scene.

We have also experimented and obtained good results with robust versions [10] of the above direct method. The robust version of the above method handles scenes with multiple moving objects. It computes the *dominant* parametric motion, where all other image regions are detected as outliers [10, 9]. The outlier mask is used to segment the image region into the dominant layer (that whose image motion can be explained by the computed dominant 2D transformation) and to the layer which corresponds to the remaining parts of the image (whose motion cannot be explained by the computed 2D dominant parametric transformation). The same technique can then be applied *recursively* to the layer which corresponds to the remaining parts of the image, to find the next dominant parametric transformation and its region within the image, etc.

Note that the outlier mask computed for each 2D transformation is a *continuous* mask. Its values can be used for *weighting* purposes in the robust parametric motion estimation. They are also occasionally used for weighting image frames in the *integration* process to construct the mosaic image (Section 3.2).

3.2 Image Integration

Once the frames are aligned (or, in the dynamic case, the current mosaic and new frame are aligned), they can be integrated to construct the mosaic image (or *update* the mosaic, in the dynamic case). One of several schemes can be chosen for integrating the aligned images:

1. A regular temporal average of the intensity values of the aligned images.

3 Mosaic Construction

The input video sequence can be segmented into contiguous *scene subsequences* via some type of scene change detection technique [32]. A mosaic image is then constructed from all frames in a scene sequence, giving a panoramic view of that scene. Two steps are involved in this process: the *alignment* of the images in the sequence, followed by the *integration* of the images into a mosaic image.

3.1 Image Alignment

Image alignment depends on the chosen world model and motion model. The alignment can be limited to 2D parametric motion models, or can utilize more complex 3D motion models and layered representations. This paper focuses on 2D alignment models. Work on 3D image alignment is currently in progress and described in [16, 15].

The parametric motion that is used to register (align) images represents the motion of a dominant surface in the scene, usually the background scene. In the current implementation, 2D parametric motion models (a 6-parameter affine transformation or an 8-parameter quadratic transformation) are used to approximate the motions between two images.

To align two images (an "inspection" image and a "reference" image), we use the hierarchical direct registration technique described in [2, 8] with a planar surface image motion model. This technique first constructs a Laplacian pyramid from each of the two input images, and then estimates the motion parameters in a coarse-fine manner. Within each level the Sum of squared difference (SSD) measure integrated over regions of interest (which is *initially* the entire image region) is used as a match measure. This measure is minimized with respect to the quadratic image motion parameters.

The SSD error measure for estimating the image motion within a region is:

$$E(\{\mathbf{u}\}) = \sum_{\mathbf{x}} \left(I(\mathbf{x},t) - I(\mathbf{x}-\mathbf{u}(\mathbf{x}),t-1)\right)^2 \tag{1}$$

where $\mathbf{x} = (x,y)$ denotes the spatial image position of a point, I the (Laplacian pyramid) image intensity and $\mathbf{u}(\mathbf{x}) = (u(x,y),v(x,y))$ denotes the image velocity at that point, and the sum is computed over all the points within the region and $\{\mathbf{u}\}$ is used to denote the entire motion field within that region.

The 2D motion field of a 3D planar surface can be presented by a 2D quadratic transformation:

$$\begin{aligned} u(\mathbf{x}) &= p_1 x + p_2 y + p_5 + p_7 x^2 + p_8 xy \\ v(\mathbf{x}) &= p_3 x + p_4 y + p_6 + p_7 xy + p_8 y^2 \end{aligned} \tag{2}$$

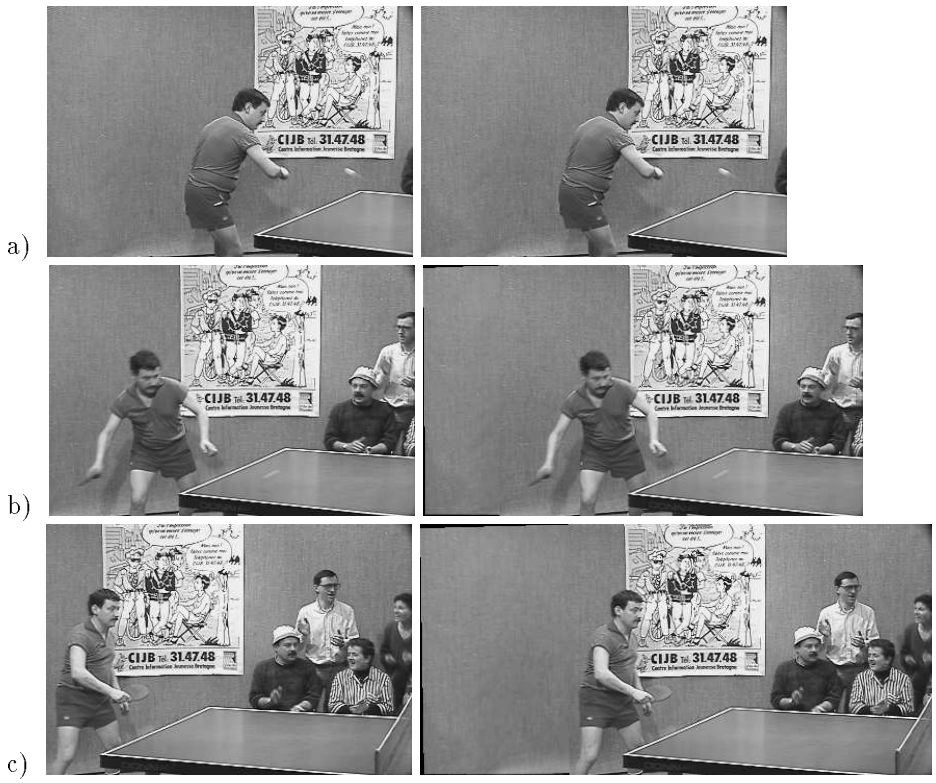

Figure 5: Evolution of the dynamic mosaic images of the table-tennis game sequence.
Left column: Three frames from the sequence.
Right column: The corresponding dynamic mosaic images. Note that the position of the player and the crowd are constantly being updated to match the information in the current frame.

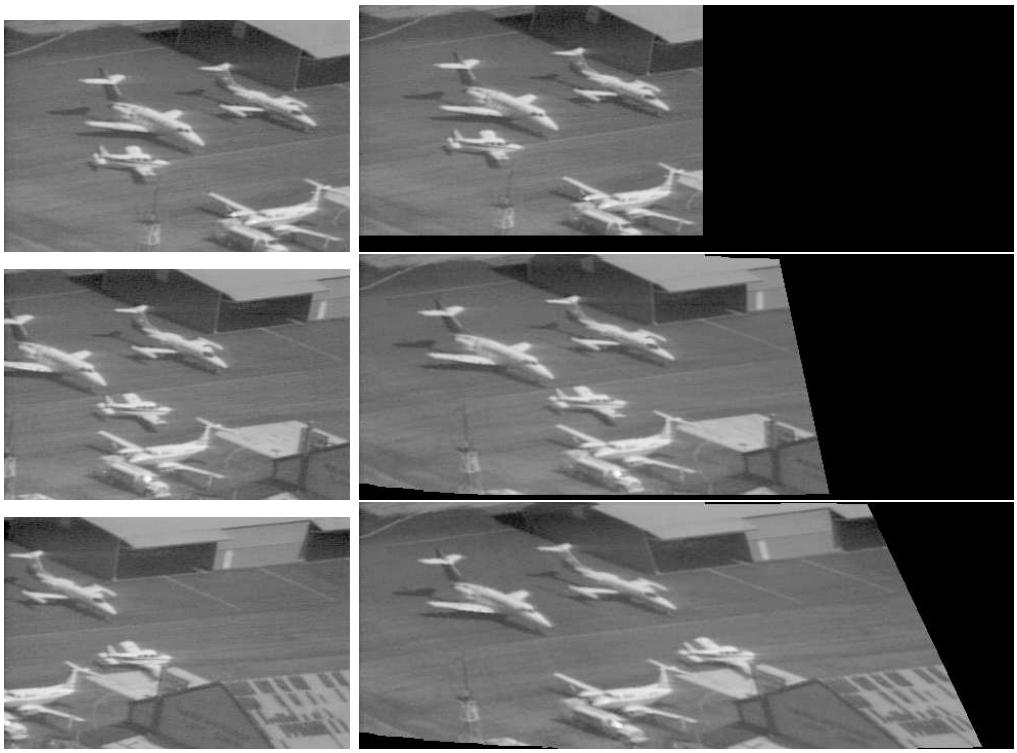

Figure 4: Evolution of dynamic mosaic image of the airport sequence.
Left column: Frames taken from the sequence.
Right column: The corresponding dynamic mosaic images. Note that the position of the taxiing plane is constantly being updated to match the current frame.

the mosaic that correspond to high resolution frames. Constructing the mosaic image at the highest detected resolution, on the other hand, has the penalty of oversampling the low resolution frames. Moreover, in the dynamic mosaic case, the resolution variation is not known in advance.

Varying image resolutions are handled by a *multi-resolution* mosaic data structure, which captures information from each new frame at its closest corresponding resolution level in a *mosaic pyramid*. It is a sparse pyramid in the sense that the resolution levels are not complete (certain mosaic regions may be represented at high resolution, others only at low resolution). When a frame is predicted/reconstructed from the mosaic pyramid, the highest existing resolution data in the mosaic which corresponds to the frame (i.e., is within its region of support) is projected onto that frame's resolution. The multiresolution mosaic data structure can be directly applied to the static mosaic, but for dynamic mosaics, it should be combined with the selective memory representation.

image quality. As opposed to storage applications, where the processing can be done in batch mode, transmission applications require on-line processing of the frames. Therefore, in this case, the mosaic image is constructed *dynamically*, by constantly updating its content to match that of the current frame. The details of this process are described in Section 4.

This scheme assures that the mosaic image is updated with the most recent information. The prediction based on dynamic mosaic is therefore usually more "current" than the static mosaic. The term "dynamic" refers to the contents of the mosaic. However, the coordinate system of the mosaic may also be dynamically updated (depending on visualization requirements).

When a *fixed* coordinate system is chosen for the dynamic mosaic, each new image frame is warped towards the current dynamic mosaic image, and the information within its field of view is updated according to the update criterion (e.g., most recent, average, weighted average, etc. (see Section 3.2)).

When the coordinate system of the mosaic is chosen to be *dynamically* updated to match that of the input sequence, the current dynamic mosaic image is warped towards each new frame, and then the information within the current field of view is updated according to the update criterion. However, such a process involves some type of sub-pixel grey-scale interpolation (e.g., typically local bilinear or bicubic interpolation is used), and repeating this process over time will lead to excessive blurring of the image. In order to avoid this blurring, the dynamic mosaic is implemented using a *selective memory* scheme. The selective memory scheme predicts each pixel of the current frame with the pixel from the most recent occurrence of that scene point without incurring the blurring introduced by repeated warping. In order to achieve this, the source of each prediction is a previously received frame in its original coordinate system. To avoid storing all the previous frames, the decoder maintains a segmented or "tiled" representation of the mosaic, keeping only the most recent occurrences of the imaged scene.

Figures 4 and 5 show examples of the evolution of some dynamic mosaics. Figure 4 shows an evolving dynamic mosaic image of the airport sequence. Note that in the dynamic mosaic the taxiing airplane does not disappear (as opposed to the static mosaic case in Figure 3), and its position is constantly being updated.

Figure 5 shows an evolving dynamic mosaic image of the table-tennis game, where the player and the crowd move with respect to the background, while the camera pans to the right. Note that in the dynamic mosaic case the crowd and the player do not blur out (as opposed to the static mosaic case shown in Figure 2), and are constantly being updated.

2.3 Multiresolution Mosaic

Changes in image resolution occur, e.g., as the camera zooms in and out. Constructing the mosaic image in low resolution results in loss of high frequency information in the regions of

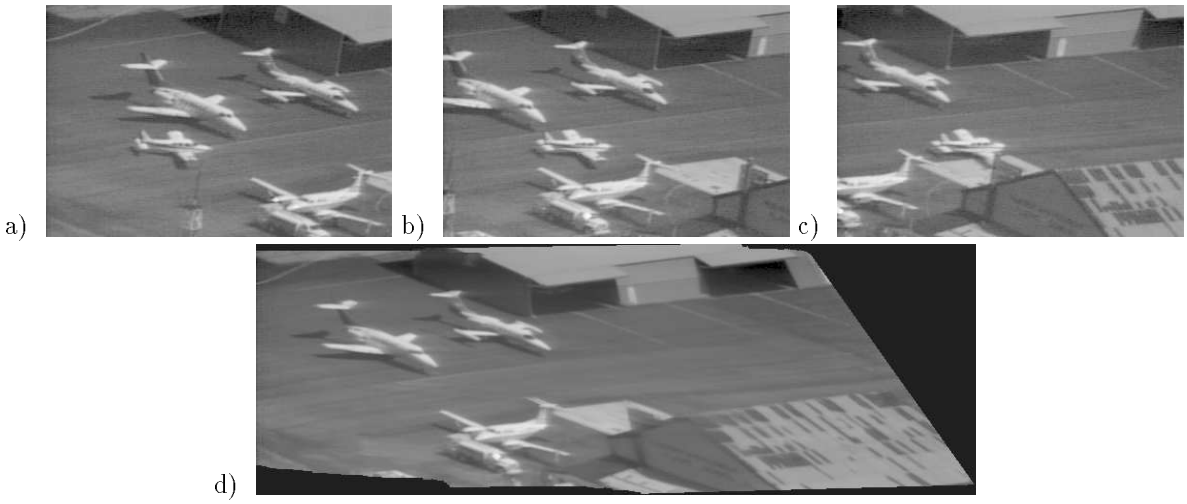

Figure 3: Static mosaic image of an airport sequence.
a,b,c) Three frames out of a 85-frame sequence obtained by a camera panning to the right following a small taxiing airplane.
d) The static mosaic image constructed using a temporal median. Note that the taxiing airplane does not appear in the mosaic image.

a database of sequences. For such browsing, it suffices to retrieve the mosaic images alone (without the residuals). Once a scene of interest has been detected, the part which corresponds to a requested frame of interest is directly reconstructed from the mosaic image of that scene and displayed. In other words, the reconstruction of a given frame does not require the reconstruction of previous frames, as would be needed in an MPEG type sequential coding technique.

Current video-editing systems tend to not use sequential coding but instead rely on JPEG [29, 13] type intra-frame compression techniques. One of the reasons for this choice is the requirement of sequential reconstruction imposed by MPEG techniques. As a result, the compression efficiency achieved by these systems is low (15:1), and the resulting storage volumes are large. On the other hand, the static mosaic based representation provides some of the benefits of motion-compensated prediction while retaining the random access capability.

An important additional benefit of mosaic based representations for video editing is the ability to manipulate the contents of the sequence by simply manipulating the single static mosaic image. During reconstruction, the changes made to the static image can be automatically applied to each of the individual frames of the sequence, since the coordinate relationships between the frames and the mosaic are known.

2.2 Dynamic Mosaic

In real-time transmission applications the need for random access to individual frames is less important. The emphasis here is on very low bit rate coding, with reasonable reconstructed

Figure 2: Static mosaic image of a table-tennis game sequence.
a,b,c) Three frames out of a 270-frame sequence obtained by a camera panning across the scene.
d) The static mosaic image constructed using a temporal median.
e) The static mosaic image constructed using a temporal average.

of the background scene, where moving objects either disappear or leave "ghost-like" traces. Temporal averages usually result in blurrier mosaic images than those obtained by temporal medians.

Examples of static mosaic images are shown in Figures 2 and 3. In Figure 2 a static mosaic image of a table-tennis game sequence is constructed, once using a temporal median, and once using a temporal average. In this sequence, the player and the crowd move with respect to the background, while the camera pans. The constructed mosaic image displays a sharp background, with blurry crowd, and a ghost-like player. Figure 3 shows a static mosaic image of an airport sequence produced using a temporal median. In this sequence a small airplane (in the center of the frames) is taxiing from left to right, while the camera pans to the right. The constructed mosaic image in this case displays a sharp image of the background airport with no trace of the taxiing plane.

A collection of static mosaics for different scene segments is useful for rapid browsing of

2 The Mosaic Representation

In this section the general concept of the mosaic image representation is described. Although the basic idea of mosaic-based compression applies to both transmission and storage applications, these two types of applications place slightly different sets of requirements on the compression scheme. These different sets of requirements lead to slightly different mosaic representations. In storage applications, the coding and decoding processes need not be real-time, although they must be fast, and the coded information should preferably support non-sequential access to individual frames. For these purposes, we develop a *static* mosaic representation (Section 2.1). On the other hand, transmission applications require real-time coding and decoding, and generally need only a sequential access to the incoming video stream. For this, we develop a *dynamic* mosaic representation (Section 2.2). To effectively deal with scenes containing significant changes in scale due to camera zoom, we develop a multiresolution mosaic representation (Section 2.3).

2.1 Static Mosaic

Typical storage applications include video-editing and interactive video manipulation (e.g., for special effects), and general video database browsing and search. In these cases, it is important to be able to rapidly examine all available video clips, and get random access to any frame. Also, for database systems that provide any degree of content-based image retrieval [23], it is important to achieve access rates that are more or less independent of the length of the video sequence. Although coding and decoding must be fast, it need not happen in real-time (e.g., 30 frame/sec.).

For storage applications, we use a "static" mosaic representation. The input video sequence is segmented into contiguous *scene subsequences* (via some type of scene change detection technique [32]) and a static mosaic image is constructed for each subsequence to provide a snapshot view of that scene. This is done by registering all frames of that subsequence to a *fixed* coordinate system (which can be either user-defined or chosen automatically based on motion and activity analysis[17]). The registered frames are then integrated over a single grid (the mosaic image). These static mosaic images can be used as key frames for browsing and other purposes. The *single* static mosaic image is then compressed and stored along with the 2D parametric transformation relating each frame to the mosaic coordinate system. The residuals of each frame relative to the static mosaic are also coded and stored. The integration of the aligned frames into a single mosaic image can be done in a number of different ways involving different types of temporal filters, such as average, median, spatially weighted average (or median) to reduce the effect of outliers, etc. These options are discussed in more detail in Section 3.

Any of these schemes applied to a registered scene segment will produce a panoramic image

changing mosaic image, always containing the most recent information available.

- A *multiresolution* mosaic representation.
- Significance based coding of residuals.
- The choice of the motion model for image alignment.

Mosaic based compression (MBC) applies to both video-storage and video transmission applications. However, the requirements on representation and usage placed by these two types of applications are slightly different. As a result, the exact method of using MBC and its benefits are slightly different for these two types of applications. In either case, the fundamental insight is that in many situations, the camera scans the same scene for some length of time (usually in the order of seconds or minutes). Therefore significant compression can be achieved by representing an extended view of the scene collected from many frames and encoding the residuals of the images relative to that extended view. It should be noted that no significant "image understanding" is required for this approach, and it relies on simple, generic models of the imaging process that are not restricted to particular types of objects. The choice of the motion computation technique used for aligning the images to form the mosaic may impose some implicit assumptions about the scene model or the camera motion, however, our current implementation uses global 2D parametric transformations, which apply to a wide variety of situations (see Section 3.1). This approach is realizable immediately, and is currently under implementation on real-time hardware.

In general, since MBC is consistent with motion compensated prediction and extends it to larger spatial and temporal extents, it will outperform techniques based on pure inter-frame block-based prediction. It can be directly used with any spatial coding scheme for images or residuals such as wavelet coding[19, 25], DCT, etc.

The issues addressed in this paper are presented in the following order: Section 2 addresses the issues related to the mosaic representation, Section 3 describes the technique used for constructing mosaic images from image sequences. These two sections discuss mosaic representations without regard to the compression and coding issues. These issues are addressed in Section 4, which presents the codecs for storage and transmission applications. Since significance-based coding is one of key elements of our approach, it is separately described in Section 4.3. The presented method has been applied successfully to several standard video sequences of real dynamic scenes, resulting in improved reconstructed image quality (both visually and in terms of SNR), as well as substantially lower bit-rates than MPEG. These results are presented in Section 5. Section 6 discusses the extensions that are currently being considered, while Section 7 reviews the ideas described here and the results.

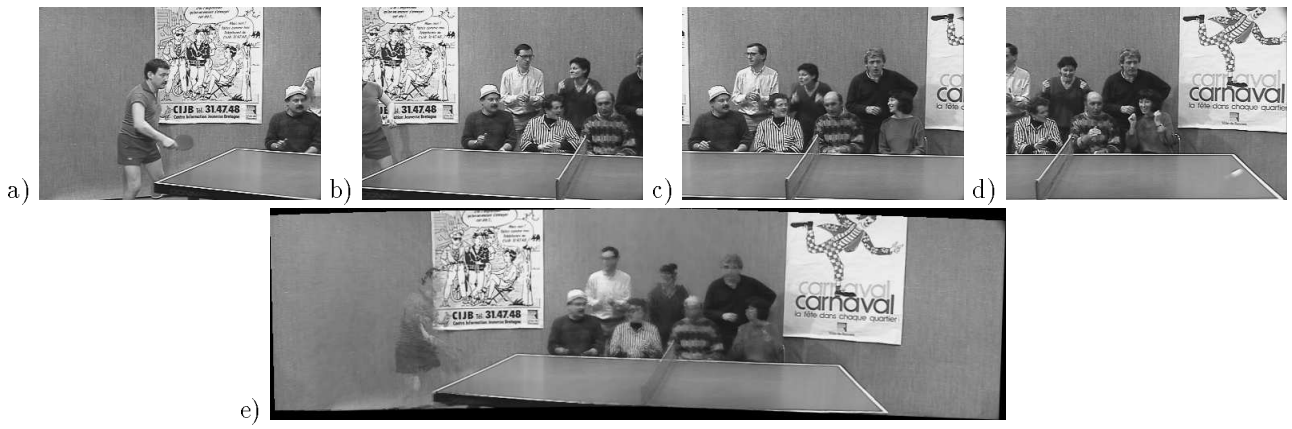

Figure 1: A mosaic image of a table-tennis game sequence.
a,b,c,d) Four frames out of a 270-frame sequence obtained by a camera panning across the scene.
e) The mosaic image.

1.2 Mosaic based compression

A mosaic image is constructed from all frames in a scene sequence, giving a panoramic view of the scene. Parametric motion models are used to align the dominant portion of the scene to construct the mosaic. The parametric motion representations describe interframe motion (and thus large scale spatial correlations of the motion) more efficiently and accurately than block displacements used in MPEG type of methods. Large scale temporal correlations are captured by the mosaic which predicts portions of the scene that are revisited by the camera.

Once a mosaic image of a background scene is constructed and coded, it need not be recoded when the same areas are revisited by the camera or uncovered by moving objects. The only new information that is coded in such cases is changes in the scene with respect to the background (e.g., moving objects) and 3D misalignments with respect to the 2D mosaic. The benefit of mosaic based compression is therefore especially dramatic in scenarios where the camera pans and zooms repeatedly over the same scene.

The mosaic representation is illustrated in Figure 1, where the camera pans across a live table tennis scene

As outlined, mosaic representation and parametric motion models are similar to ideas that are described in the image processing literature [26, 31, 21, 20, 27]. However, in order to maximize the compression efficiency while retaining semantically significant information in the scene, several essential components are developed and incorporated into our mosaic based compression scheme. These include:

- The selection of a *static* mosaic representation which uses a single static mosaic image for predicting all the frames, versus a *dynamic* mosaic representation which uses a dynamically

1 Introduction

1.1 The scope of the paper

Established inter-frame compression algorithms such as MPEG [11, 18], do not give reasonable image quality at very low bit rates (i.e., less than 64 kbits/sec). This is because these methods exploit spatial and temporal correlations only at small scales – blocks and adjacent frames. Moreover, the motion representation used by these techniques is neither parsimonious, nor an adequate representation of the real motion between frames. The use of a predefined block and "group-of-pictures" syntax also does not allow them to be adaptive to the actual variations in the correlations of motion between regions and frames.

Recently, there has been considerable interest in developing techniques based on scene analysis to achieve high compression ratios [22, 1, 14]. Some of these techniques [22] rely on segmentation and 3D shape analysis that would allow prediction of many images based on 3D shape models of objects and surfaces in the scene. Often, these are specifically targeted towards video-telephone types of applications in which there may be a single or small number of human heads occupying the dominant portion of each image and the image sequence.

This paper focuses on the second class of techniques that use parametric motion models to exploit large scale spatial correlations of motion that may be present in many image sequences [28, 1]. In particular, it applies a representation of image sequences called a *mosaic image* [28, 26, 20, 27] to achieve high compression ratios for a number of applications, where strong object models may not be easy to obtain. Such applications include remote surveillance, which typically involves a narrow field of view camera that scans an outdoor natural scene with a few moving objects present, or video databases, where there may be an interest in rapidly browsing through a library of sequences and rapidly accessing particular frames or subsequences.

The main contribution of this paper is an analysis of the issues that arise when considering how to apply mosaics for compression, and the development of preliminary end-to-end simulations that demonstrate the benefits of mosaic based compression (MBC) in different situations. Specifically: (i) we develop two types of mosaic, one called a *static* mosaic which is ideally suited for database storage applications and the other a *dynamic* mosaic that is suited for very-low bit rate transmission applications, (ii) we develop a scheme called "significance based coding" of the residuals that remain after mosaic based prediction of individual frames, and (iii) we show results of applying this approach to several real image sequences. In addition, we also outline the general approach for extending the 2D mosaics that are presented here to incorporate 3D information to achieve even greater compression.

Video Compression Using Mosaic Representations

Michal Irani, Steve Hsu, and P. Anandan
David Sarnoff Research Center
CN5300, Princeton NJ 08543-5300
Phone: 609-734-3047 Email: michal@sarnoff.com

Abstract

We describe a technique for video compression, based on a *mosaic image* representation obtained by aligning all frames of a video sequence, giving a panoramic view of the scene. We describe two types of mosaics, *static* and *dynamic* which are suited for storage and transmission applications respectively. In each case, the mosaic construction process aligns the images using a global parametric motion transformation, usually canceling the effect of camera motion on the dominant portion of the scene. The residual motions that are not compensated by the parametric motion are then analyzed for their significance and coded. The mosaic representation exploits large spatial and temporal correlations in image sequences. In many applications where there is significant camera motion, (e.g., remote surveillance) it performs substantially better than traditional interframe compression methods, and offers the potential for very low bit-rate transmission. In storage applications, such as digital libraries and video editing environments, it has the additional benefit of enabling direct access and retrieval of single frames at a time.