Efficient Representations of Video Sequences and Their Applications

Michal Irani P. Anandan Jim Bergen
Rakesh Kumar Steve Hsu

David Sarnoff Research Center
CN5300, Princeton, NJ-08530, U.S.A.
Email: michal@sarnoff.com

Recently, there has been a growing interest in the use of mosaic images to represent the information contained in video sequences. This paper systematically investigates how to go beyond thinking of the mosaic simply as a visualization device, but rather as a basis for an *efficient* and *complete* representation of video sequences. We describe two different types of mosaics called the *static* and the *dynamic* mosaics that are suitable for different needs and scenarios. These two types of mosaics are unified and generalized in a mosaic representation called the *temporal pyramid*. To handle sequences containing large variations in image resolution, we develop a *multiresolution mosaic*. We discuss a series of increasingly complex alignment transformation (ranging from 2D to 3D and layers) for making the mosaics. We describe techniques for the basic elements of the mosaic construction process, namely sequence *alignment*, sequence *integration* into a mosaic image, and *residual analysis* to represent information not captured by the mosaic image. We describe several powerful video applications of mosaic representations including *video compression*, *video enhancement*, *enhanced visualization*, and other applications in *video indexing*, *search*, and *manipulation*.

1 Introduction

Video is a very rich source of information. Its two basic advantages over still images are the ability to obtain a continuously varying set of views of a scene, and the ability to capture the temporal (or "dynamic") evolution of phenomena.

A number of applications have recently emerged that involve processing the entire information within video sequences. These include digital libraries, interactive video analysis and softcopy exploitation environments, low-bitrate video transmission, and interactive video editing and manipulation systems. These applications require efficient representations of large video streams, and efficient methods of accessing and analyzing the information contained in the video data.

There has been a growing interest in the use of a panoramic "mosaic" image as an efficient way to represent a collection of frames (e.g., see Figure 1) [17, 21, 22, 16]. Since successive images within a video sequence usually overlap by a large amount, the mosaic image provides a significant reduction in the total amount of data needed to represent the scene.

Although the idea of the mosaic and even some of its applications have been recognized, there has not been a systematic approach to the characterization of what the "mosaic" is,

or even an attempt to develop any type of standard terminology or taxonomy. In practice, a single type of mosaic, such as a static mosaic image obtained from all the frames of a contiguous sequence, is suitable for only a limited class of applications. Different applications such as video database storage and retrieval and real-time transmission and processing require different types of "mosaics".

Also, while mosaics have been recognized as efficient ways of providing "snapshot" views of scenes, the issue of how to develop a *complete* representation of scenes based on mosaics has not been adequately treated. Specifically, we refer to the question of how to represent the details *not* captured by the mosaics, so that the sequence can be fully recovered from the mosaic representation.

The purpose of this paper is to develop a taxonomy of mosaics by carefully considering the various issues that arise in developing mosaic representations. Once this taxonomy is available, it can be readily seen how the various types of mosaics can be used for different applications. The paper includes examples of several applications of mosaics, including video compression, video visualization, video enhancement, and other applications.

The remainder of the paper is organized as follows: Section 2 presents various types of mosaic representations, and discusses their efficiency and completeness in terms of sequence representation. Section 3 describes the techniques that we use to align the images, construct the mosaics, and detect the significant "residuals" not captured in the mosaics from the input video stream. Section 4 outlines a number of powerful video applications of the mosaic representations with examples and experimental results. Finally Section 5 discusses the salient issues for future research on this topic.

2 The Mosaic Representation

A mosaic image is constructed from all frames in a scene sequence, giving a panoramic view of the scene. Although the idea of a mosaic image is simple and clear, a closer look at the definition reveals a number of subtle variations. For instance, since the different images that comprise a mosaic spatially overlap with each other, but are taken at different time instances, there is a choice regarding how the different grey values available for the same pixel are combined. Similarly, the variations in the pixel resolution between images leads to the issue of choosing the resolution of the mosaic image. Finally, there are also choices

regarding the geometric transformation model used for aligning the images to each other. The different choices in these various issues is typically a result of the type of application for which the mosaic is intended.

In this section we describe different "types" of mosaics that arise out of the types of considerations outlined above.

2.1 Static Mosaic

The static mosaic is the common mosaic representation [17, 22, 21, 16, 14], although it is usually not referred to by this name. It has been previously referred to as "mosaic" or as "salient still" (e.g., see Figures 1 and 2). It will be shown (in Section 4) how the static mosaic can also be extended to represent temporal subsamples of key events in the sequence to produce a static "event" mosaic (or "synopsis" mosaic).

The input video sequence is usually segmented into contiguous *scene subsequences* (e.g., see [23]), and a static mosaic image is constructed for each scene subsequence to provide a snapshot view of the subsequence. This is done *in batch mode*, by aligning all frames of that subsequence to a *fixed* coordinate system (which can be either user-defined or chosen automatically according to some other criteria). The aligned images are then integrated using different types of temporal filters into a mosaic image, and the significant residuals are computed for each frame of relative to the mosaic image. The details of the mosaic construction process are described in Section 3. Note that after integration, the moving objects either disappear or leave "ghost-like" traces in the panoramic mosaic image.

Examples of static mosaic images are shown in Figures 1 and 2. In Figure 1 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 to the right. The constructed mosaic image displays a sharp background, with blurry crowd, and a ghost-like player. Figure 2 shows a static mosaic image of a baseball game sequence produced using a temporal median. In this sequence two players run across the field (from right to left), while the camera pans to the left and zooms in on the players. The constructed mosaic image in this case displays a sharp image of the background with no trace of the two players. In both examples, a 2D motion model was sufficient to align the images (see Section 3).

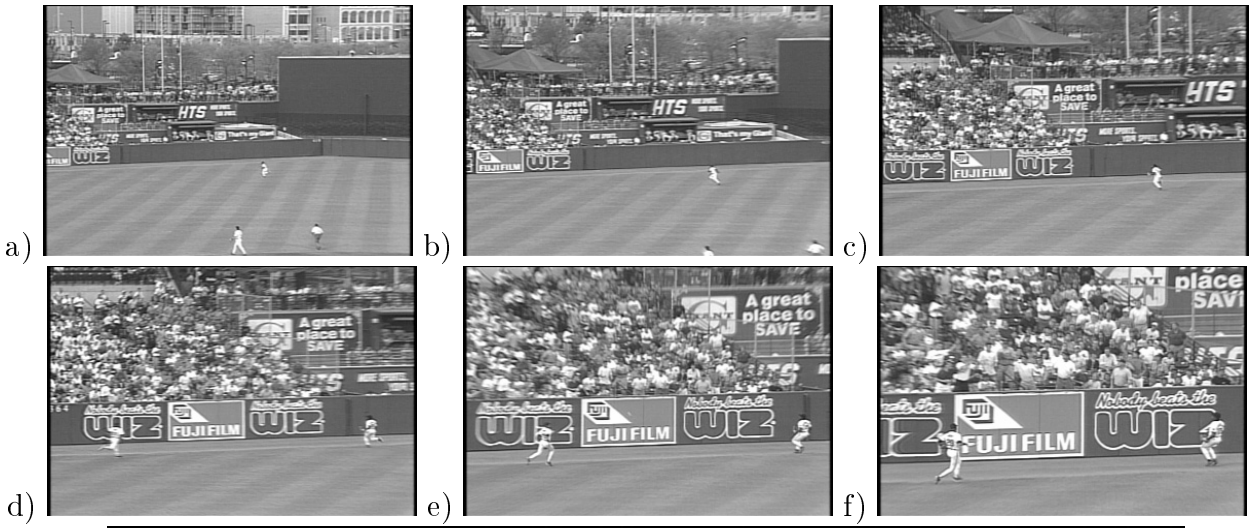
The static mosaic image exploits long term *temporal* redundancies (over the entire scene

Figure 1: Static mosaic image of a table-tennis game sequence.
a,b,c) Three out of a 300 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.

subsequence) and large *spatial* correlations (over large portions of the image frames), and is therefore an efficient scene representation. For examples, in Figures 1 and 2, the *entire* video sequence can be represented by the mosaic image of the background scene with the appropriate transformations that relate each frame to the mosaic image. The only information in the sequence *not* captured by the mosaic image and needing additional representation are the changes in the scene with respect to the background (e.g., moving players). These "residuals" can either be represented independently for each frame, or can frequently be represented more efficiently as another layer using yet another mosaic [1] (see Section 2.5).

The issue of representing "residuals" which are not captured by the mosaic image has frequently been overlooked by handling sequences with no scene activity [21, 16, 14]. The mosaic image, along with the frame alignment transformations, and with the "residuals" together constitute a *complete* and *efficient* representation, from which the video sequence can be *fully* reconstructed. These issues have been addressed to a limited extent with respect to video compression in [1], although that work does not consider how to assign a significance measure to the residuals or how to handle *non-rigid* layers.

The static mosaic, being an efficient scene representation, is ideal for *video storage and*



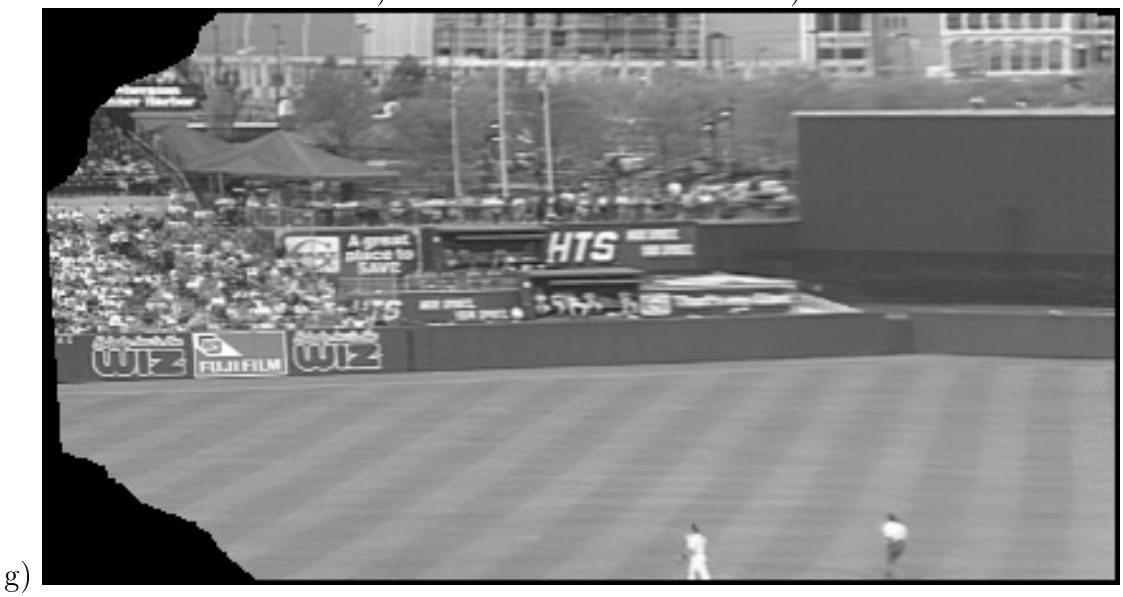


Figure 2: Static mosaic image of a baseball game sequence.
a,b,c,d,e,f) Six out of a 90 frame sequence obtained by a camera panning from right to left and zooming in on the runners.
g) The static mosaic image constructed using a temporal median. The black regions are scene parts that were never imaged by the camera (since the camera zoomed-in on the scene).

retrieval, especially for *rapid browsing* in large digital libraries and to obtain efficient access to individual frames of interest. It can also be used to increase the efficiency of content-based indexing into a video sequence, to reduce the tedium associated with video manipulation and analysis. Last but not least, it can be used for enhanced visualization in the form of panoramic views, as well as a tool for enhancing the contents of the images. These applications are described in greater detail in Section 4.

2.2 Dynamic Mosaic

Since the *static* mosaic is constructed in *batch mode*, it cannot completely depict the dynamic aspects of the video sequence. This requires a *dynamic* mosaic, which is a *sequence* of evolving mosaic images, where the *content* of each new mosaic image is updated with the most current information from the most recent frame. The sequence of dynamic mosaics can be visualized either with a stationary background (e.g., by completely removing any camera induced motion), or in a manner such that each new mosaic image frame is aligned to the corresponding input video image frame. In the former case, the coordinate system of the mosaic is fixed (see Figure 3), whereas in the latter case the mosaic is viewed within a moving coordinate system (see Figure 4). In some cases a third alternative may be more appropriate, wherein a portion of the camera motion (e.g., high frequency jitter) is removed or a preferred camera trajectory is synthesized.

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. When a *virtual coordinate system* is chosen (either predetermined by the user, or computed according to some criterion), both the dynamic mosaic and the current frame are warped towards that coordinate system. Note that the definition of the coordinate system and the warping mechanism will vary according to the world and motion model (see Section 3).

Figures 3 and 4 show examples of the evolution of some dynamic mosaics. Figure 3 shows an evolving dynamic mosaic image of a table-tennis game, where the player and the crowd

Figure 3: Evolution of the dynamic mosaic images of the table-tennis game sequence.

Left column: Three frames from the original sequence.

Right column: The corresponding dynamic mosaic images. Note that the coordinate system as well as the position of the player and the crowd are constantly being updated to match the current frame.

move with respect to the background, while the camera pans to the right. In this example we chose to construct the mosaic in a *fixed* coordinate system (that of the first frame). Note that in the dynamic mosaic the crowd and the player do not blur out (as opposed to the static mosaic shown in Figure 1), and are constantly being updated.

Figure 4 shows an evolving dynamic mosaic image of a baseball game sequence, where two players run across the field (from right to left), while the camera pans to the left and zooms in on the players. In this example we chose to construct the mosaic in a *dynamic* coordinate system that matches that of the input video (i.e, changes with each new frame). Note that in the dynamic mosaic the players do not disappear (as opposed to the static mosaic in Figure 2), but are constantly being updated.

The *complete* dynamic mosaic representation of the video sequence consists of the *first* dynamic mosaic, and the *incremental* alignment parameters and the *incremental* "residuals" that represent the changes. Note that the difference in mosaic content between the static and dynamic mosaics implies a difference in the "residuals" that are not represented by the mosaic. In the dynamic case, since the content of the mosaic is dynamically being updated, the "residuals" reflect only changes in the scene that occur in the time elapsed between successive frames, as well as additional parts of the scene that were revealed for the first time to the camera. These are different than the "residuals" in the static case, that represent objects

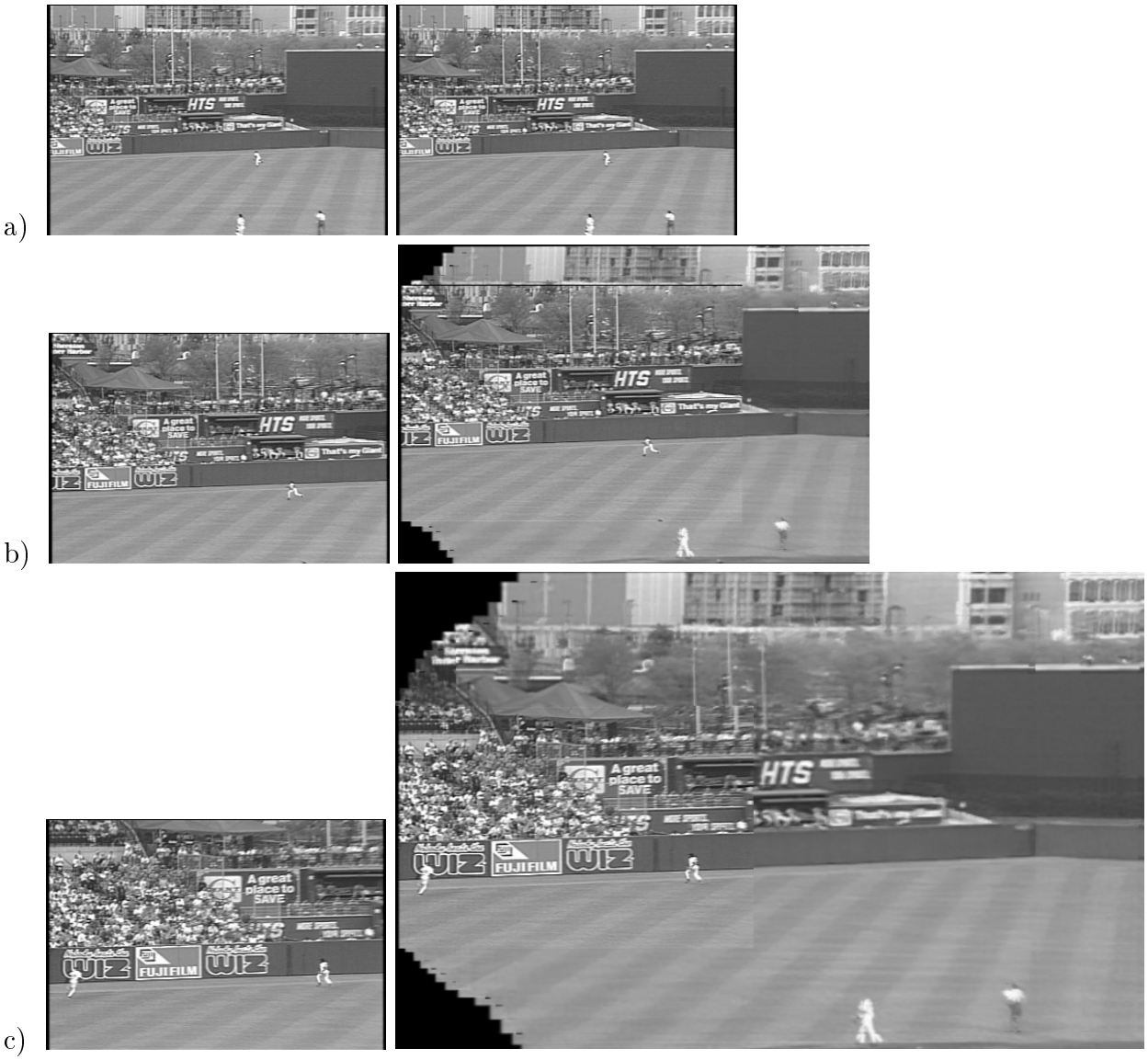


Figure 4: Evolution of the dynamic mosaic images of the baseball game sequence.
Left column: Three frames from the original sequence.
Right column: The corresponding dynamic mosaic images. Note that the coordinate system as well as the position of the players are constantly being updated to match the current frame.

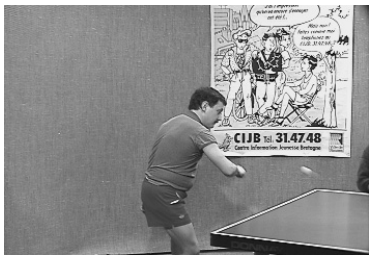
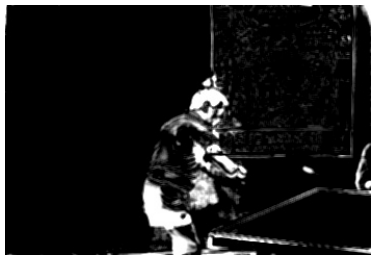
a)  b)  c) 

Figure 5: The residual maps of static vs. dynamic cases.
a) A single frame from the table-tennis sequence.
b) The residual map computed for the corresponding frame in the static representation. The brighter values signify more significant residuals.
c) The residual map computed for the corresponding frame in the dynamic representation. Note that the amount of residuals in the dynamic case is significantly smaller than the amount of residuals in the static case.

that have some motion in any portion of the video sequence (with respect to the background static mosaic). Figure 5 shows an example of frame residuals detected for the static and the dynamic representations in the table-tennis sequence. In general, since changes between successive frames are relatively small, the amount of "residual" information in the dynamic mosaic will be smaller than that in the static case. The dynamic mosaic is therefore a more *efficient* scene representation than the static mosaic. It too allows complete reconstruction of the original video sequence. However, due to its *incremental* frame reconstruction, it lacks the important capability of *random access* to individual frames, which is essential for video manipulation and editing.

The dynamic mosaic is an ideal tool for low bit-rate transmission (see Section 4). The choice of the coordinate system for constructing and visualizing the dynamic mosaic image will depend on the application. For example, in remote surveillance type of applications, which typically involve a narrow field of view camera that repeatedly scans the same outdoor natural scene and is usually very bouncy, it is beneficial to construct a dynamic mosaic with a *fixed* coordinate system, as it will also serve as a *stabilization* mechanism for the remote observer. However, in flight surveillance, it makes more sense to keep a dynamically updating coordinate system that matches that of the view as seen by the pilot (with a gradually growing field of view obtained as the mosaic is constructed). These issues are discussed in Section 4.

2.3 Temporal pyramid

The static and the dynamic mosaics are extremes of a continuum: One uses a completely static image, which may be based on an arbitrarily long sequence, and in principle there may be an arbitrarily long time interval between the current frame and the static mosaic. The dynamic mosaic is completely current and always has the most recent available information. As a result, the dynamic mosaic is more efficient than the static mosaic, but since it requires sequential reconstruction of the frames, it does not provide as immediate an access to the individual frames as the static mosaic. In order to bridge the gap between these two extremes and obtain the benefits of both representations (i.e., representation-efficiency vs. random-access to frames), a "temporal pyramid" mosaic can be used.

As discussed in Section 2.2, the static mosaic does not remove as much short-term temporal redundancy among "residuals" as the dynamic mosaic (see Figure 5). The static mosaic can be extended to use a hierarchy of mosaics whose levels corresponds to different amounts of temporal integration. This hierarchical organization is similar to spatial image pyramid representation. The finest level contains the set of original images, one for each frame of the input sequence. The temporal sampling decreases successively as we go from fine to coarse resolution levels of the pyramid. We will refer to the nodes of the pyramid as "mosaics". For instance, in the manner analogous to the Laplacian pyramid, the sampling rate can be reduced by a factor of 2 between successive levels (although in principle, the factor can be any number). In this case, each "mosaic" at a given level can be obtained in the same fashion as pixel values are computed in the Laplacian pyramid (e.g., as difference of low-pass operators). The *coarsest* level will consist of a single mosaic, which is the same as the static mosaic described in Section 2.1. The succeeding levels represent residuals estimated over various time scales. Reconstruction can be achieved by hierarchically combining the static mosaic with the residual mosaics, namely in logarithmic time.

2.4 Multiresolution Mosaic

Changes in image resolution occur within the sequence, e.g., as the camera zooms in and out. Constructing the mosaic image in low resolution results in the loss of high frequency information in the regions of the mosaic that correspond to high resolution frames. Constructing the mosaic image at the highest detected resolution, on the other hand, incurs 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 can be 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.

Note that the multiresolution mosaic is a completely different representation than the temporal pyramid mosaic, although both use a coarse-to-fine data structure. The unit elements at each level of the multiresolution mosaic are *pixels*, while the unit elements at each level of the temporal pyramid mosaic are *mosaic images*. The multiresolution mosaic data structure can be applied to the static, dynamic, and temporal pyramid mosaic representations.

2.5 Mosaic Representations Versus Scene Complexity

So far, we have described various types of mosaics that address different requirements, specifically representational efficiency and access efficiency. In this subsection, we consider another aspect of mosaic representations, namely the choice of the frame to frame alignment transformation.

The 2D Mosaic : All the examples that have been shown so far in this paper have relied on constructing a mosaic image using 2D alignment parametric transformations. Such a mosaic provides a *complete* representation of the scene segment under the following circumstances: In scenarios where there is no scene activity apart from the motion of the camera and when either there is no translation of the camera, or when the entire scene can be approximated by a single parametric surface (typically a plane).

The 2D motion field of a 3D planar surface is described by the 2D quadratic transformation:

$$u(\mathbf{x}) \;=\; p_1 x + p_2 y + p_5 + p_7 x^2 + p_8 xy$$

$$v(\mathbf{x}) \quad = \quad p_3 x + p_4 y + p_6 + p_7 xy + p_8 y^2 \tag{1}$$

In practice, however, 2D alignment is a good approximation even when these conditions are violated, provided the violations are small. For instances, if the camera translates slowly and/or the relative distances between surface elements in the scene (ΔZ) is small compared to their range (Z), the effects of parallax can be neglected. Similarly, if independent scene activity is confined to a small number of pixels, again these effects can be neglected during the construction of the mosaic. In both these situations, there will be non-zero but small "residuals" associated with the mosaic. These "residuals" are represented separately. The 2D alignment process is described in Section 3.1.1.

Parallax Based Mosaic : As explained above, the 2D alignment models are sufficient when the effects of 3D parallax (relative to the mosaic surface) are small. However, this does not mean that when these effects are significant, we need to abandon the mosaic based approach to representing video sequences. One approach is to represent the parallax effects as *individual* frame "residuals". A better and more efficient approach to modeling parallax effects can be done by naturally extending the mosaic representation to capture parallax in a *single* representation.

The key to the 3D extension lies in the observation that the residual motion after aligning a dominant planar surface in the scene is purely epipolar and is due to the combination of camera translation and the distance of the other parts of the scene from the dominant (aligned) plane [14, 19]. Specifically, we use the following result derived in [14]: Given two views (under perspective projection) of a scene (possibly from two distinct uncalibrated cameras), if the image motion corresponding to an arbitrary plane (called the "reference plane") is compensated (by applying an appropriate 2D parametric warping transformation to one of the images) then the residual parallax displacement field on the reference image plane is an epipolar field. Further, the magnitude of the parallax displacement vector at each point directly depends on the distance of that point from the reference plane.

The total motion vector of a point can be written as the sum of the motion vector due to the planar surface (u_p, v_p) (as represented in equation (1)) and the residual parallax motion vector (u_r, v_r).

$$(u, v) \quad = \quad (u_p, v_p) + (u_r, v_r) \tag{2}$$

The residual vector can be represented as:

$$\begin{aligned} u_r(\mathbf{x}) &= \frac{\gamma}{T_\perp}(fT_x - xT_z) \\ v_r(\mathbf{x}) &= \frac{\gamma}{T_\perp}(fT_y - yT_z) \end{aligned} \tag{3}$$

where

$$\gamma = H/P_z,$$

H is the perpendicular distance of the point of interest from the reference plane and P_z is its depth. (T_x, T_y, T_z) is the displacement of the camera between two views as expressed in the coordinate system of the reference (or "fist") view, and $T_\perp$ is the perpendicular distance from the camera center of the *second* view to the plane and f is the focal length. At each point in the image γ varies directly with the height of the corresponding 3D point from the reference surface and inversely with the depth of the point. In [14, 19, 20] it was shown that the parallax field is a "relative" affine invariant. Finally, it is noted that aligning the reference plane by warping the inspection image using the parametric motion field (u_p, v_p) also removes all of the rotational components of camera motion.

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 mosaic image that can be used to predict the parallax-induced motion over that duration. We refer to this representation of 3D structure as a "height" map (relative to the dominant surface), a term borrowed from aerial imagery analysis.

Thus, the complete mosaic based representation of a 3D scene sequence (without independently moving objects) would contain: (i) an *intensity* mosaic image produced by 3D alignment of the sequence frames, (ii) a corresponding "height" mosaic, (iii) for each frame: the computed 2D alignment transformation of the dominant surface, (iv) for each frame: the computed 3D camera translation. For more details on this representation see [14, 13, 15].

An example of the mosaic image produced by 3D alignment is shown in Figure 6. The three original wall images Frames 1, 2 and 3 are shown in Figures 6a, 6b, and 6c respectively. Figure 6d shows a 2D mosaic built using only 2D affine transformations. The 2D affine transformations used aligns the wall part of the images. However the objects sticking out of the wall exhibit parallax and are not registered by the affine. As a result in the 2D mosaic (Figure 6d), there are many ghost (duplicate) lines in the bottom half of the image.

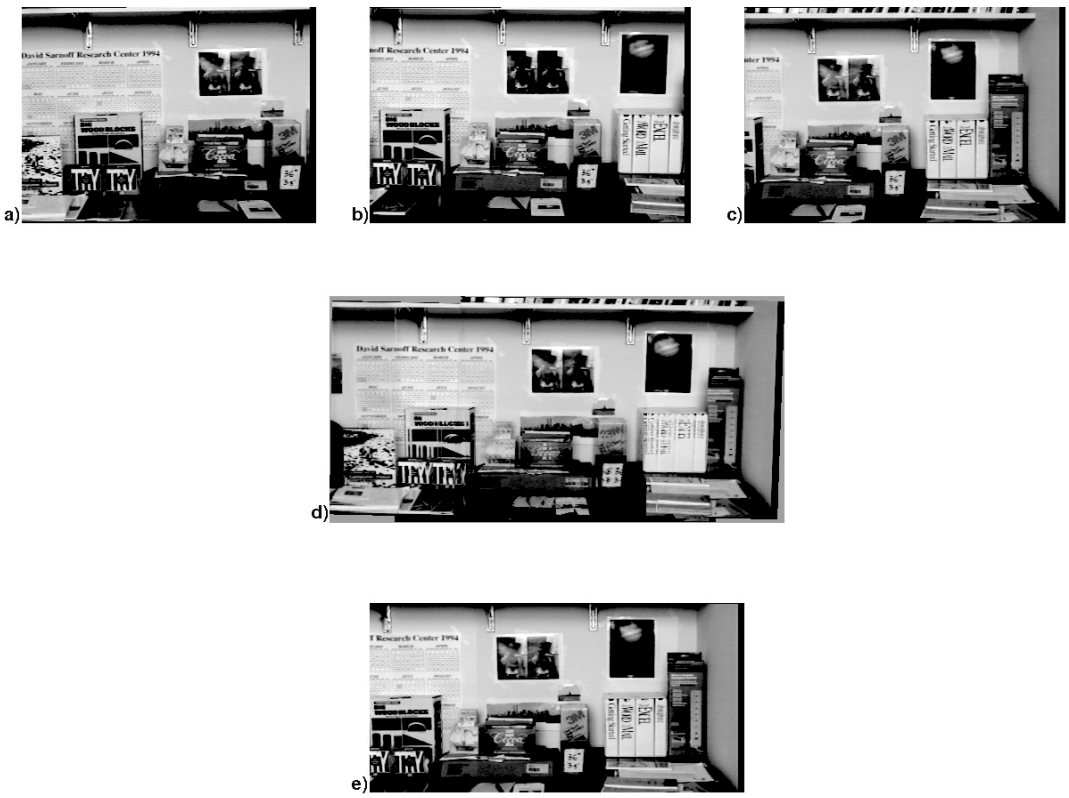


Figure 6: **Parallax Corrected Mosaic to Represent 3D Scenes.**
(a,b,c) Three frames of the input video sequence taken from a sideways moving camera. The third frame (c) is used as the referene image.
(d) The result of constructing a mosaic based on 2D planar surface alignment. Patterns on the wall are perfectly aligned. However, note that objects "sticking out" of the wall are not well aligned, as indicated by the duplicate lines.
(e) The result of constructing a mosaic after correcting for 3D planar parallax. Note that all portions of the scene are well aligned.

The reader's attention is drawn to the image regions corresponding to the duplicate lines in the boxes titled "TRY" and "Wooden blocks" in the left bottom and the smearing on the book title information (e.g. Excel, Word, Getting Started) in the right bottom of Figure 6d respectively.

Figure 6e shows a 3D corrected mosaic image. In this case, using the parallax motion information, the objects sticking out of the wall are correctly positioned and no duplicate lines are visible. The 3D corrected mosaic was made by using Wall Frame 3 (Figure 6c) as the final destination image. Using the parallax computed from Wall frames 1 and 2, Wall frame 2 was reprojected into the frame 3 coordinate system. This reprojected image was then merged with frame 3 to make the Mosaic image shown in Figure 6e. Note in Figure 6c one can not see the boxes entitled "Wooden blocks" or "TRY". In the mosaic image, they however appear and are present in the geometrically correct locations.

The efficiency of representing 3D information with a height map (heights with respect to a surface in the scene) is greater than representing it with a range map (depths relative to the camera). The increased efficiency is due to the fact that the height map is *invariant* to the camera motion, as opposed to the range map. Moreover, the range of values of a typical height map has significantly smaller than that of a typical camera centered range or "depth" map, and can therefore be much more compactly encoded.

The details of the 3D mosaic are described in [15].

Layers and Tiles : In principle, the 2D alignment model augmented with 3D parallax information is adequate for all scenes in which there is no independent object motion in the scene. However, in practice, when the 3D scene begins to be cluttered with objects at widely varying depths, and/or when real or "fence-like" transparency is present, the parallax based representation of 3D is highly inefficient. A natural extension to the 2D mosaic is to use multiple layers of 2D mosaics in the manner suggested by Adelson[1].

In this representation, 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, 10, 9, 11, 4]. In our own work, we have developed algorithms for multiple motion segmentation[8] and layered motion recovery[18]. We plan to combine these with the 2D mosaics, and the parallax based representations described earlier.

Another extension that is necessary in order to handle extended fields of view is a "tiled" representation. To motivate this, consider the simple example of a panning camera. In this case, the use of a single mosaic image plane based on central projection does not meaningfully extend to more than a few degrees of rotation. Reprojecting the views acquired after rotating the camera by 45 degrees or so results in significant distortion of the images, and therefore later in poor image reconstruction. Similarly when the camera is moved around an object (e.g., even a simple object like a box or a table) to get a frontal view of all of its surfaces, projecting these to a single planar mosaic view will lead to the loss of image information from other views. In the panning case, a natural alternative is to use a coordinate system based on a spherical retina. However, this approach does not easily generalize to the second example of a moving camera given above. A better approach may be to have a series of tiles that correspond to different mosaic imaging planes (e.g., in the case of the pan, these will be tangent planes to the sphere) and assemble these "tiles" together into a larger mosaic. Each image can be predicted from the tile that corresponds most to that image in terms of resolution and minimizes the distortion. In this way the representation becomes somewhat more complex, but its efficiency will be preserved while making it more of a complete and effective scene representation.

3 Mosaic Construction

A mosaic based representation is constructed from all frames in a scene sequence, giving a panoramic view of that scene. Three steps are involved in this process: the *alignment* of the images in the sequence, the *integration* of the images into a mosaic image, and the *computation of significant residuals* between the mosaic and the individual frames.

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. Most of the examples in this paper utilize 2D alignment models. In this section we describe the 2D alignment methods in some detail, and briefly outline the 3D alignment methods. More details of 3D alignment can be found in [14, 13, 15]. This section also describes how we compose frame-to-frame alignment parameters to achieve the

alignment of an entire sequence of images.

3.1.1 2D Image Alignment

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 and 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:

$$E(\{\mathbf{u}\}) = \sum_{\mathbf{x}} \left(I(\mathbf{x}, t) - I(\mathbf{x} - \mathbf{u}(\mathbf{x}), t-1)\right)^2 \tag{4}$$

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.

As noted in Equation 1 (Section 2), the 2D motion field of a 3D planar surface can be described by the 2D quadratic transformation:

$$\begin{aligned} u(\mathbf{x}) &= p_1 x + p_2 y + p_5 + p_7 x^2 + p_8 xy \\ v(\mathbf{x}) &= p_3 x + p_4 y + p_6 + p_7 xy + p_8 y^2 \end{aligned} \tag{5}$$

Besides being an exact description of the instantaneous motion field of a

The objective function E given in Equation (4) 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 [4] 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 [11] 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 [11, 10]. 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; e.g., see Figure 5). 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 the pixels of the individual image frames during the *integration* process to construct the mosaic image (Section 3.2). Benefits of using these weights in the mosaic construction for some applications are demonstrated in Section 4.

3.1.2 3D-alignment

Parallax Estimation: The key step involved in the computation of 3D alignment is the estimation of the residual parallax motion with respect to the reference plane that is aligned by the 2D mosaic. The technique for achieving this is described in greater detail in [15, 13]. Below we briefly outline our approach.

The computation of the parallax information can proceed in one of two ways. The first technique takes a *sequential registration* approach, in which the plane is first registered using a 8 parameter quadratic transformation. The residual parallax is then estimated as a separate step. The second technique simultaneously estimates the planar and parallax motion components, and is hence referred to as a *simultaneous registration*.

In the sequential approach, the plane registration is achieved in the same manner as

described in Section 3.1.1. After the plane is aligned in this fashion, the parallax vectors and the direction of translation are simultaneously estimated using the quasi-parametric technique described in [3]. The quasi parametric technique is generally more accurate than using optic flow, but requires an initial estimate for translation. If needed, an initial estimate of the translation direction can be obtained by using the optical flow obtained by using the technique also described in [3].

The sequential registration algorithm is useful when there is a visible planar surface in the scene that occupies a significant portion of the image. However, in many situations, such as images of curved objects and hilly terrains, no such plane may be present in the scene, hence, the sequential registration algorithm may fail in the first step (of plane alignment). However, the plane+parallax representation is still applicable, since a "virtual" reference plane can be used as the basis for computing the residual parallax.

To handle the situations when a "virtual" plane is required, the planar surface alignment and the parallax estimation have to be performed simultaneously. This algorithm consists of two steps:

1. First the plane registration algorithm described in Section 3.1.1 is applied to the entire scene. Although this may not register any real or virtual plane, it provides a good set of initial parameters for the second step.

2. The total motion vector at a point is expressed as a sum of the motion vector due to a planar surface and the residual parallax motion. The initial estimate for the planar motion field is given by the results of the first step given above. The parallax field is initialized to zero, and the translational motion parameters are set to an arbitrary initial value. Both these components are then refined simultaneously—i.e., the 8 parameters of the quadratic transformation is refined as well as the translational motion parameters and the parallax magnitude at each pixel.

The refinement process achieves alignment of every pixel (within the region where the two views overlap) between the two views. The refinement process is done in a manner similar to the quasi-parametric ego motion estimation algorithm described in [3].

3D corrected mosaics: We refer to the mosaic image that is obtained by achieving 3D alignment between multiple views as the "3D corrected mosaic". The recovery of parallax

information requires at least two views of the same portion of the scene. This means that the extension of a single view into a mosaic consisting of information from a second view requires a *third* view to provide the parallax information for the second view (in particular, for those portions of the second view not visible in the first view). The three views should partially (but not completely) overlap with each other. Given such views, the process of construction involves the following steps: The first step is to register the first two images and to build a parallax map in the second frame's coordinate system. With this parallax map, we compute the quadratic transformation parameters $(p_1, \ldots, p_8)$ and the camera translation parameters (T_{2x}, T_{2y}, T_{2z}), which register the second image with the third image. Note that to estimate these 11 "pose" parameters in the mosaic case, we do not need point correspondences. Rather, we directly register the second image with the third image using the estimated parallax map as an input. We again minimize equation (4) but this time only estimate the 11 pose parameters.

After the pose parameters between the second and the third image are estimated, the second image is then reprojected (by forward warping) to create a synthetic image taken from the third view-point. This synthetic image however contains image regions common to the first two images but not present in the third image. The final step to obtain the mosaic is to merge the synthetic third image with the actual third image. The result of this process of constructing the 3D corrected mosaic was previously shown in Figure 6.

To construct the parallax mosaic, we forward-warp the parallax map to the third image coordinate system, much the same way as the second image was reprojected. Given the pose parameters between images 2 and 3, the parallax map of those portions not visible in 1 but only in 2 and 3 can also be estimated. The reprojected parallax map is merged with this additional parallax information to complete the mosaic.

3.1.3 Sequence Alignment

The alignment of *all* image frames in the sequence to form the mosaic can be performed in three ways:

Frame to frame: The alignment parameters are first computed between *successive* frames for the entire sequence. These parameters can then be composed to obtain the alignment parameters between any two frames of the sequence.

When constructing a static mosaic, all the frames are aligned to a fixed coordinate system. If the mosaic coordinate system that is selected is that of a particular frame (called the "reference" frame), then all other images are aligned to that frame. If a *virtual* coordinate system is selected, then the transformation between the virtual coordinate system and one of the input frames (the reference frame) needs to be given. In this case, this additional transformation is simply composed with the transformations required to align each frame to the reference frame.

Note that the sequence alignment process requires only one pass on the sequence (for computing adjacent alignment transformations, and then sequentially composing these transformations for warping the image frames to the mosaic coordinate system).

Frame to mosaic: One problem with frame to frame alignment is that errors may accumulate during the repeated composition of alignment parameters. The alignment can be further refined by directly refining the transformation between each image frame and the mosaic image. To handle the problem of large displacements between the mosaic image and the new image frames, the alignment parameters computed between the previous frame and the mosaic image are used as an initial estimate.

Mosaic to frame: The frame to mosaic alignment is appropriate when the mosaic is constructed with respect to a static coordinate system. However, in some dynamic applications such as real-time video transmission, it is important to maintain the images in their input coordinate systems. In this case, it is more useful to align the mosaic to the current frame. In this case the transformation between the most recent mosaic and the current frame is identical to the transformation between the previous frame and the new frame.

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.

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 1).

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 3.1.1; see also Figure 5). 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. A weighted temporal average where the weights correspond to the *inverse* outlier rejection maps computed in the motion estimation process of the dominant "background" (Section 3.1.1; see also Figure 5). This scheme prefers the *non*-dominant "foreground" data over "background" data in the mosaic construction. The mosaic image constructed by applying such an integration method would contain a panoramic image not only of the scene, but also of the *event* that took place in that scene sequence. We call this type of mosaic an "event mosaic" or a "synopsis mosaic", as it provides a "snapshot" view of the entire synopsis in the sequence. This kind of mosaic can be very useful for rapid browsing (see Figure 10 and Figure 11).

6. Integration in which the *most recent information*, i.e., that which is found in the most recent frame, is used for updating the mosaic. This is especially useful in the dynamic mosaic construction (see Figures 3 and 4). Of course, if desired, the update can be more gradual, e.g., a decaying temporal average which give more weight to more recent information, and tends to forget information more distant in time.

7. 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. See more details in Section 4.

3.3 Significant Residual Estimation

The complete sequence representation includes the mosaic image, the transformation parameters that relate the mosaic to each individual frame, and the residual differences between the mosaic image and the individual frames. To reconstruct any given frame in its own coordinate system, the mosaic image is warped using the corresponding mosaic-to-image transformation and composed with the residuals for that frame. In the case of the static mosaic, the differences are directly estimated between a single reference (static) mosaic and each frames, and the reconstruction is straight forward. In the case of the dynamic mosaic, however, the residuals are incremental, being with respect to the previous mosaic image frame. In this case the reconstruction proceeds sequentially from frame to frame.

Residuals between the current frame and the mosaic–based predicted frame occur for several reasons: object or illumination change, residual misalignments, interpolation errors during warping, and noise. Of these the object changes are the most semantically significant, and in some cases the illumination changes are as well.

The efficiency of the representation can be maximized by assigning a significance measure to the residuals, and using those to weight the residuals. An effective way of determining semantically significant residuals is to consider not only the residual intensity but also the the magnitude of *local residual motions* (i.e., the local misalignments) between the predicted frame and the actual frame. Below, we briefly outline our approach to significance analysis. The details of this measure are described in [11, 6],

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].)

$$S_t(x,y) = \frac{\sum_{(x_i,y_i)\in N(x,y)} \left| I_t(x_i,y_i) - I_t^{Pred}(x_i,y_i) \right|}{\sum_{(x_i,y_i)\in N(x,y)} |\nabla I_t(x_i,y_i)| + C} \tag{6}$$

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 5 shows an example of significant frame residuals detected for the static and the dynamic representations in the table-tennis sequence.

Although the same significance measure is used with the static and the dynamic mosaic, the locations and magnitudes of the significant residuals differ between the two schemes even when applied to the same sequence. In the case of the static mosaic, the significance measures in regions of objects that move with respect to the background are usually larger than in the case of the dynamic mosaic, as moving objects tend to blur out or even disappear in the static mosaic. Therefore, they will not appear in the predicted frame, and hence the changes will be significant. In the dynamic case, the mosaic is constantly being updated with the most recent information, and therefore, the changes in image regions that correspond to independently moving objects will be smaller between the predicted and actual frame. In the dynamic mosaic, however, image boundaries of a new frame may not exist in the predicted frame, and therefore significant residuals will be obtained at those image boundaries. This behavior does not occur in the static case, as the support of the static mosaic is the union of the supports of all frames in the scene sequence.

4 Mosaic Applications

In this section applications of the various mosaic representations will be described. The most obvious applications are *video compression* (as mosaics are efficient scene representations) and as a means of *visualization* (as mosaics provide a wide and stabilized field of view). These will be discussed in sections 4.1 and 4.2. However, mosaics are also useful in other applications, such as scene change detection, efficient video search and video indexing, efficient video editing and manipulation, and others. These applications will also be described in this section along with examples.

The examples shown in this paper are based on 2D alignment for mosaic construction. All image regions not aligned in this manner (e.g., scene changes, parallax) are represented

as "residuals". The 2D alignment was combined with various mosaic representations mentioned in Section 2 (i.e., static, dynamic, etc.), and with different integration techniques (Section 3.2) according to the needs of the desired application.

4.1 Mosaic Based Video Compression

Since mosaics provide an efficient means of representing a video sequence, the most natural application to consider is video image compression. The differences between static and dynamic mosaic representations that were outlined in the previous sections lead to differences in the two types of codecs (see Figures 7 and 8). As mentioned earlier, the static mosaic is more appropriate for storage applications, whereas the dynamic mosaic is ideally suited for real-time transmission applications. In this section, we first outline the codec for transmission, then the one for storage. A detailed description of our approach for using mosaics for compression can be found in [6].

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 are predicted by the computed 2D parametric transformation from the static or dynamic mosaic and only the significant missing residuals are coded. The parametric motion compensation can be 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 2.5). For each frame such motion information, coded in a lossless or lossy manner, needs to be stored or transmitted along with the residuals.

The residuals are compressed using a lossy spatial coder based on semantic and perceptual criteria. The significance mask described in Section 3.3 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 Transmission Codec: The real-time requirements of transmission actually translate into two requirements on mosaic based compression. One is the obvious requirement of real-time on-line processing. The other is the fact that the information maintained in the mosaic must be dynamic. Therefore, the dynamic mosaic is the natural choice for this application. Figure 7 describes at a high level the codec for real-time transmission applications. The major steps in the process are incremental mosaic construction, incremental residual estimation by

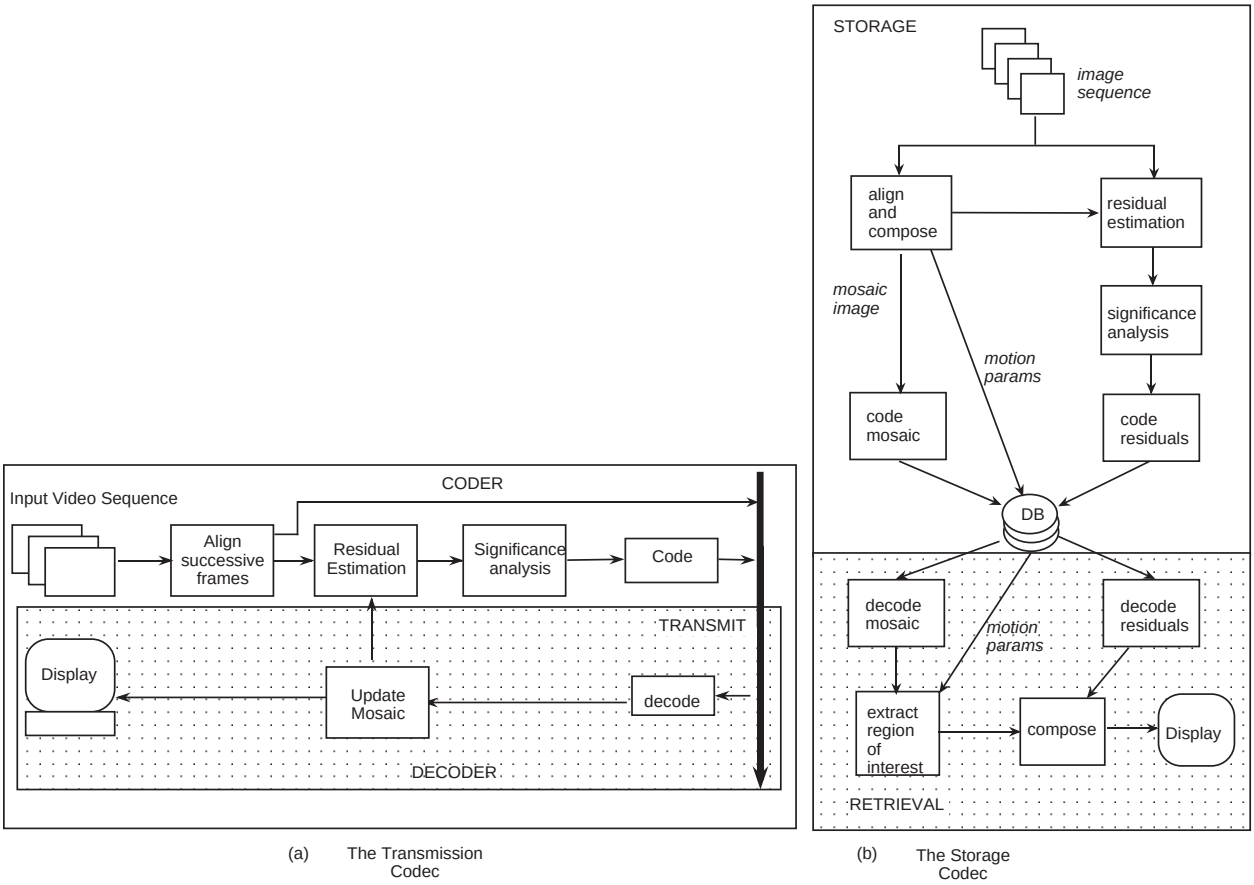
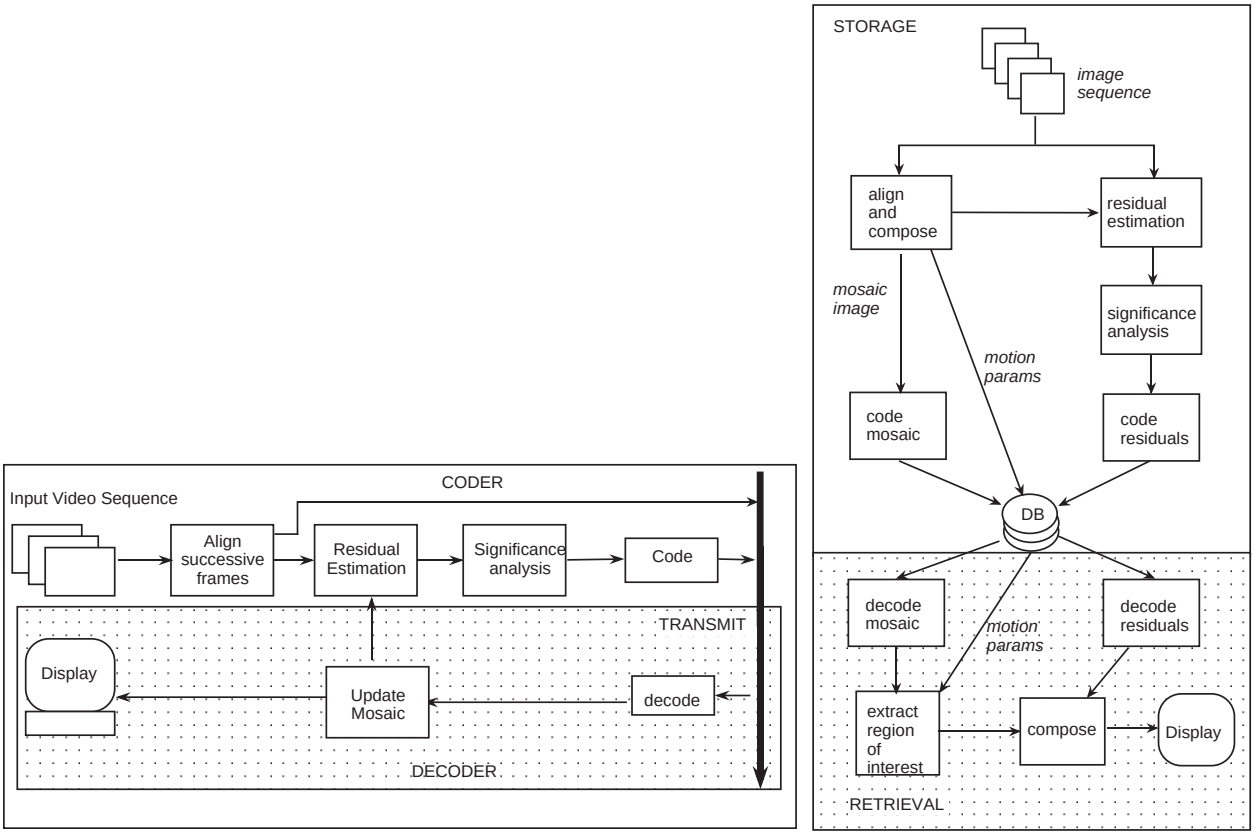


Figure 7: The codec for mosaic based video compression for real-time transmission



(a) The Transmission Codec (b) The Storage Codec

Figure 8: The codec for mosaic based video compression for video database storage

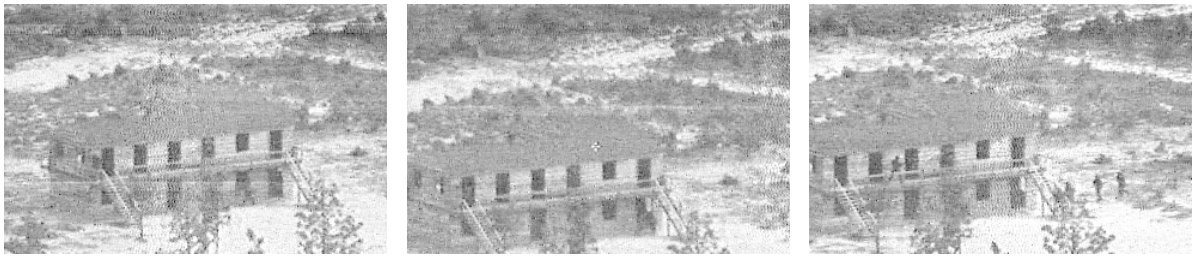
comparison to the reconstructed mosaic from the previous time instance, the computation of significance measures on the residuals, and spatial coding and decoding. As is typical of any predictive coding system, the coder maintains a decoder within itself in order to be in synchrony with the receiver.

The spatial coding of the images and the residuals can be based on any available technique, e.g., Discrete Cosine Transform (DCT) or wavelets. Our locally available DCT based coder is a part of an MPEG simulation software system and includes additional motion compensation at a block level. This exploits temporal correlations of "residual" objects moving with respect to the background motion, and proved to be more efficient than other existing spatial coders Therefore, the example shown in this section was developed using the motion compensated DCT as the spatial coder.

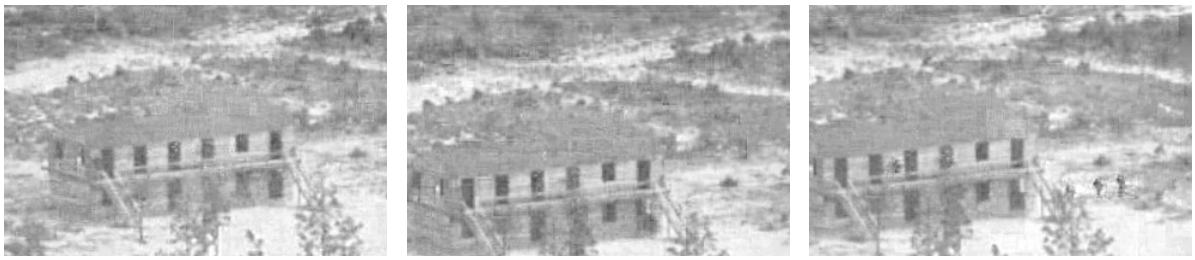
Figure 9.a shows some representative frames of a surveillance video of a storage building viewed from a flying helicopter. This example was selected partly because mosaic based compression is ideally suited for such an application, since usually the same scene is viewed over an extended period of time from a moving platform (or a stationary one with a panning camera). In these applications, typically only very low bitrate channels are available. Accordingly, the video sequence was spatially subsampled at quarter of the original resolution, and *temporally* sampled by four (i.e., 7.5 frames/sec). The sequence was then coded at the constant bit rate of 32 Kbits/sec using mosaic based compression with DCT spatial coding of the first frame and of the detected residuals (Figure 9.b). For comparison, the sequence was compressed by MPEG (without mosaic pre-processing), which is the existing standard video compression method to date, at the *same bitrate* (Figure 9.c), which resulted in significantly poorer visual quality. Note that the soldiers that are running in the scene on the right hand side of the building are visible in the mosaic-based compression results (Figure 9.b), but are invisible in the MPEG compression results (Figure 9.c). More experimental results are found in our paper on mosaic based video compression [6].

The Storage Codec: For storage applications, it is important to provide random access to individual frames. However, 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 for representation and coding.

a) Original frames:



b) Reconstructed frames using dynamic mosaic-based-compression at 32 Kbits/sec:



c) Reconstructed frames using standard MPEG at 32 Kbits/sec:

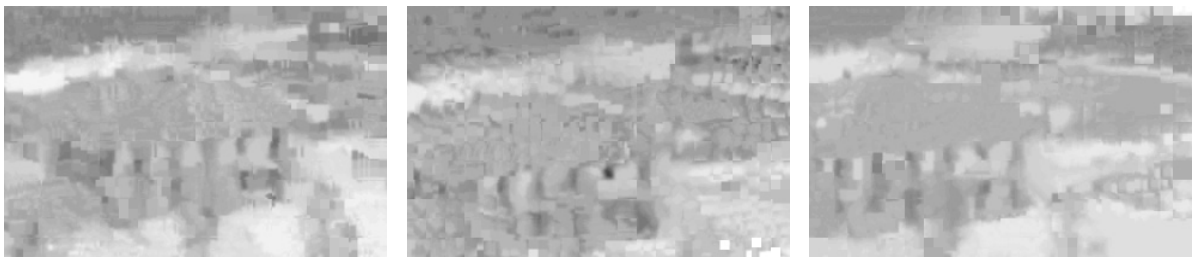


Figure 9: Transmission compression: results of dynamic mosaic-based-compression vs. standard MPEG compression on a storage-house surveillance sequence.
a) Some representative frames of a 24 second sequence. The sequence was constructed of SIF size images, and temporally sampled by four (i.e., 7.5 frames/sec).
b) The reconstructed frames after using dynamic mosaic-based-compression at a constant bit rate of 32 Kbits/sec.
c) For comparison: The reconstructed frames after using standard MPEG compression of the sequence at the same bit rate, i.e., 32 Kbits/sec. Note the differences in the reconstructed quality of the running soldiers in the images of the right column.

Figure 8 illustrates the storage codec using static mosaic for storage applications. The sequence is processed in batch mode, with the major steps being: mosaic construction, residual estimation for each frame, significance analysis, and spatial coding and decoding of the mosaic image and the individual residuals. During retrieval, the decoded individual residuals are composed with the decoded mosaic and after performing appropriate inverse motion transformation and image window selection the individual frames can be displayed. In order to limit the length of this paper, we have omitted any examples of compression for storage. (Such examples can be found in our paper on mosaic based video compression [6].)

4.2 Mosaic Based Visualization

One of the key benefits of mosaics is as a means of enhanced visualization. The panoramic view of the mosaic provides the scene context necessary for the viewer to better appreciate the events that take place in the video [17] (e.g., see Figure 2). Several different types of panoramic visualizations are possible, each highlighting a different type of information. In the following set of figures we will present examples of three useful ways of visualizing the same video sequence using mosaic representations. Each visualization has its own use:

Key frame mosaic: Given a video sequence segmented into contiguous clips of *scene sequences* (e.g., see [23]), a static mosaic image of the most salient features in the scene can be constructed for each scene sequence. The static mosaics images represent their scenes better than any single frame, and can therefore be used as "key frames" for rapid browsing through the entire video sequence [22], which is digitally stored, and for other purposes as well (see Section 4.4). The panoramic visualization shown of the baseball sequence shown in Figure 2.g also illustrates the idea of the key frame mosaic.

A second example of a key frame mosaic is shown in Figure 10.g. This figure displays a *background* mosaic image which was constructed from a 12-second surveillance flight video captured by a camera mounted on a helicopter. Note that two other helicopters are imaged in the sequence, but have been removed during the integration process of the aligned frames.

The key frame mosaic has other applications besides visualizations. These are describe in greater detail Section 4.4.

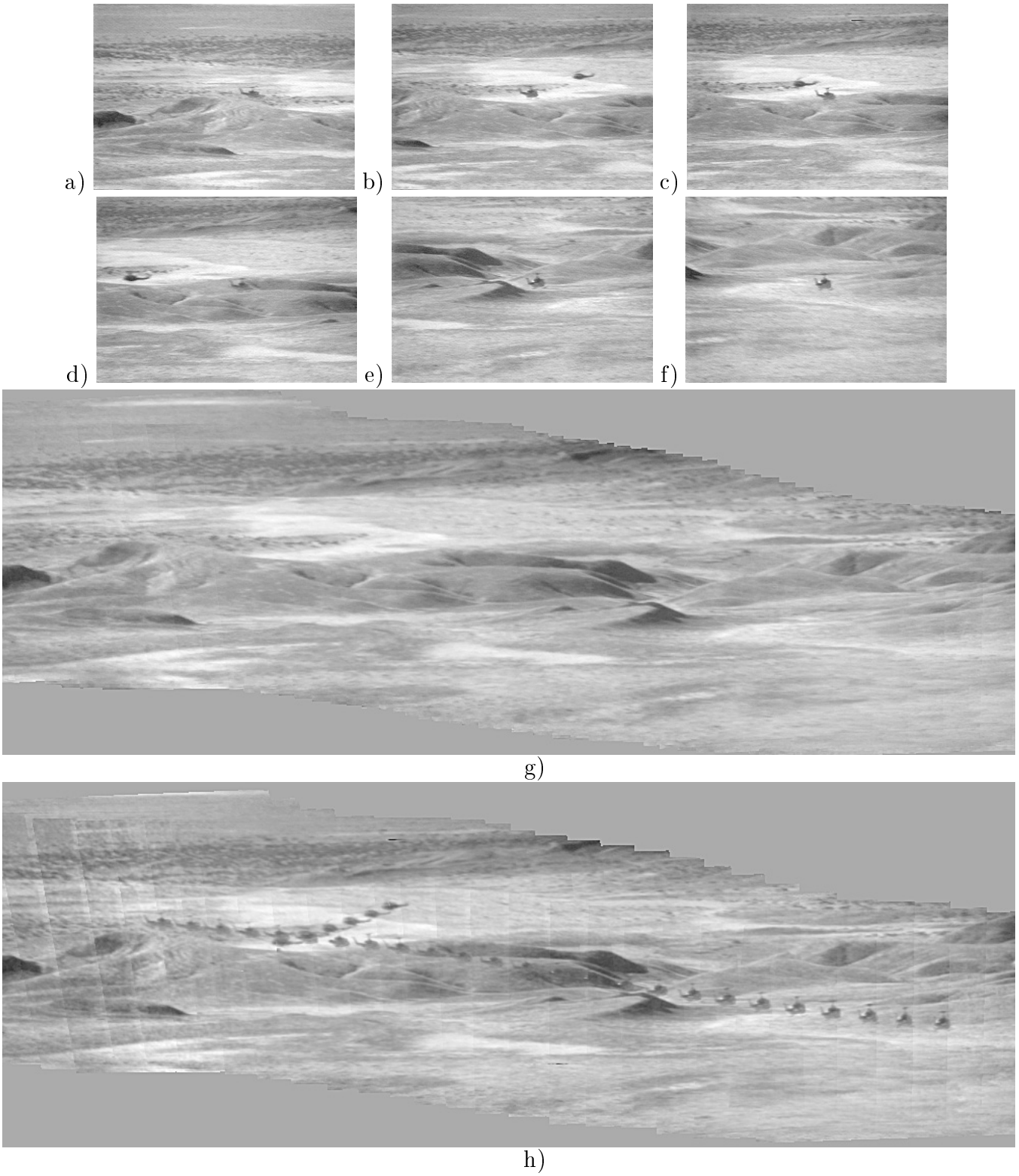


Figure 10: Panoramic mosaic image of a scene captured by a 12-second flight sequence.
a-f) Six frames sampled out of the 360 frame sequence. Note the two imaged helicopters in the foreground.
g) The panoramic mosaic image of the *background* scene (without the two imaged helicopters).
h) The synopsis mosaic image of the sequence showing the trajectories of the two imaged helicopters on top of the background mosaic.

Synopsis mosaic: While the key frame mosaic is useful for capturing the *background*, in some cases, it may be desirable to get a synopsis of the event that takes place within the video sequence. This can be achieved through a mosaic that captures the *foreground event*. The synopsis (or event) mosaic is constructed using the the outlier maps obtained during the background alignment process (see Section 3.1.1). By using the *inverse* of the outlier maps as weights during the integration process, the objects can be retained within the mosaic. Note that *regular* averaging of the aligned frames will *not* maintain the foreground moving objects, but will rather make them either completely disappear or significantly fade out. Figures 10.h and 11.b show examples of synopsis mosaics for the surveillance and baseball sequences, respectively. In order to allow clearer visualization of the moving helicopters and the runners, the input sequences were temporally subsampled at every fifth frame.

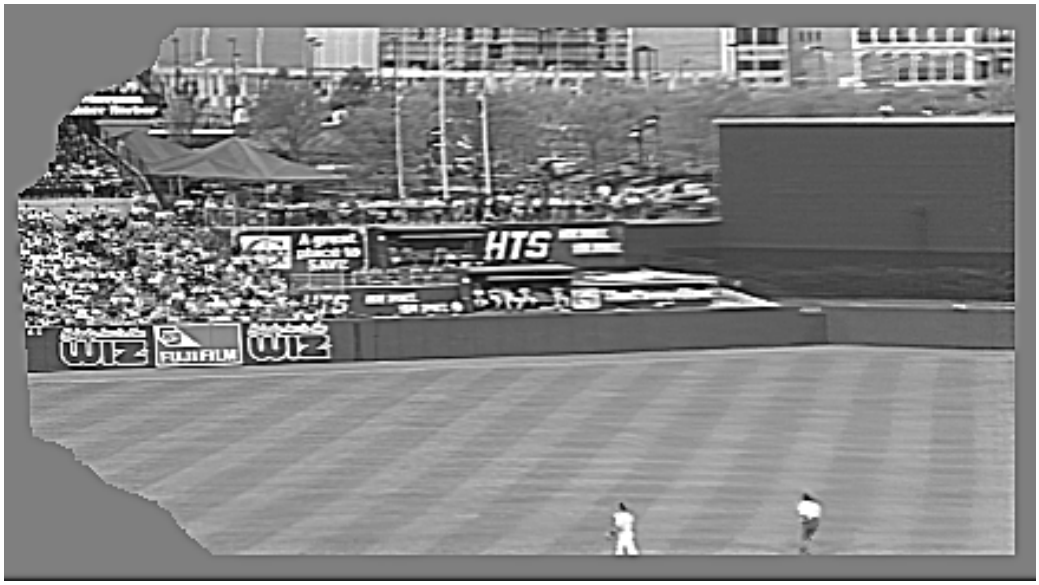
Mosaic video: The panoramic visualization provided by the mosaics is useful not only as a static image, but for dynamic video visualization as well. In this case, a new video sequence is generated (called the "mosaic video") which is a sequence of (dynamic) mosaic images. This type of visualization simulates the output of a virtual camera with desired features. For example, a virtual camera with an expanded field of view can be simulated in this fashion. Alternatively, a desired new camera trajectory can be simulated by applying the appropriate coordinate transformations to each of the mosaic video frames. The simplest example of this is stabilized video mosaic display, in which case the camera motion is completely removed. The previously shown figures 3 and 4 show examples of video mosaics. Such a display has uses in various applications such as *remote navigation*, and *remote surveillance*. Similar mosaic Visualizations have also been suggested by [17].

4.3 Mosaic Based Video Enhancement

Mosaic representations can serve as a useful and efficient tool for producing *high quality stills* from video as well as enhancing an entire video sequence.

The resolution of an image is determined by the physical characteristics of the camera: the optics, the density of the detector elements, and their spatial response. Resolution improvement by modifying the camera can be prohibitive. An increase in the sampling rate could, however, be achieved by obtaining more samples of the imaged scene/object from a sequence of images in which the scene/object appears moving at subpixel displacements.

a)



b)

Figure 11: Synopsis mosaic image of the baseball game sequence.
a) The static *background* mosaic of the scene *without* the event.
b) The synopsis mosaic of the baseball sequence showing the event that occurred in the scene on top of the background mosaic (i.e., showing the trajectories of the two runners).

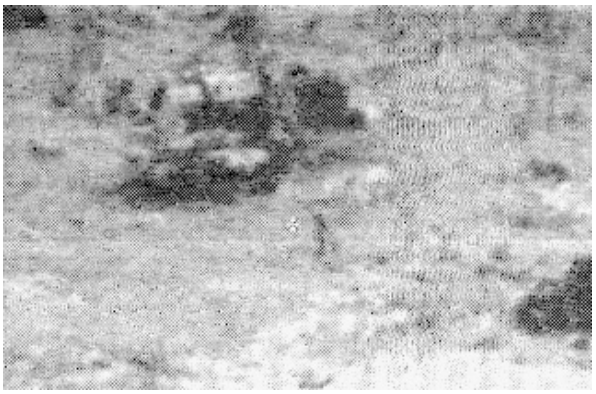
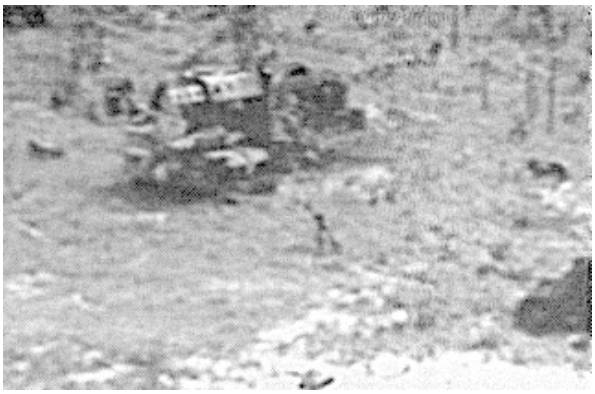
a)  b) 

Figure 12: Mosaic-based video enhancement from a surveillance sequence of a deserted truck.
a) One out of 30 frames (all frames are of the same quality).
b) The corresponding enhanced frame in the enhanced video sequence. (All the frames in the enhanced video are of the same quality.)

Therefore, aligning the sequence frames over a *finer* mosaic grid can provide higher sampling rate of the background scene, and hence integrating over that grid provides higher spatial resolution. When the blur function of the camera is also known or can be computed and used for deblurring, the increase in resolution is even more pronounced. This method is known as Super-resolution [7, 9, 16]. In [12] this idea was incorporated into a framestore of an MPEG like coder.

The *efficiency* of using *mosaics* for *video* enhancement is due to the fact that the mosaic is an *efficient* representation of the video sequence. Rather than enhancing the frames one-by-one (as is suggested in [9]), the enhancement of the entire sequence (or layer) is done in a single step within the mosaic coordinate system, and only then are the enhanced frames retrieved from the enhanced mosaic.

Figure 12 shows an enhanced frame from a sequence of thirty frames of a deserted truck imaged from a remote helicopter surveillance video. In this example, all the input frames were of very poor quality and very noisy. The sequence contained a single static scene that could be completely aligned using 2D alignment. The entire video sequence was enhanced by constructing a single enhanced 2D static mosaic, and then retrieving the frames from the mosaic back into their original coordinate systems (according to the inverse 2D parametric transformations).

4.4 Other Mosaic Based Applications

The benefit of mosaic images for various other applications has been recognized [17, 22]. Some of these relate to managing large digital libraries [22], and with respect to manipulating and editing video in video post-production environments. 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. We are currently working on applying the mosaic representations to these types of applications.

Video Editing and Manipulation: Interactive video editing and manipulation applications can benefit from the mosaic (e.g., the key-frame or the synopsis mosaic) as well. In video editing environments, it is sometimes required to take a video segment and alter the data in it. For example, pulling an existing actor or object out of the sequence (while filling in the occluded regions convincingly, of course), or inserting a non existing object into the video sequence. These processes are currently very tedious, as they are done manually, frame-by-frame. This process can be significantly sped up, as well as done more accurately, using motion analysis and mosaic constructions. During sequence reconstruction from the mosaic, the changes made to the static mosaic 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.

For example, in the baseball video sequence shown in Figure 2, a new sequence was constructed which imitates the same camera motion, but removes the runners from the field. A frame from that "video-deletion" sequence is shown in Figure 13. Such a sequence can then be edited further by inserting a different event on top of this processed video. Such video manipulations were already introduced in our previous work [9], however, in that work, the editing was done on a frame-to-frame basis. The use of the mosaic image as a tool for video editing and manipulation increases the efficiency in processing, and reduces the tedium associated with process.

One type of video-manipulation is the synthesis of new views of the scene corresponding to a desired viewing position from a given set of views. Such a need typically arises in the emerging class of virtual- or "tele"-reality applications. Since the representations developed in this paper incorporate the transformations that relate the views to each other (both in 2D

a) 

b) 

Figure 13: Video editing of the baseball game sequence showing the runners.
a) A single frame from the original sequence.
b) The corresponding frame in the video edited sequence, showing same camera orientation and background, but after removing the runners.

and in 3D), these representations can be used for this purpose. Using our representations has the advantage that the existing views can be directly combined (since the transformation that relate the views to each other is known) to obtain a nearby view, instead of relying on a single 3D world model that is painstakingly constructed from the existing views.

Video Browsing and Search: A collection of static mosaics for the different detected scene segments is useful for rapid browsing of a database of sequences. For such initial browsing, it can suffice to retrieve the key frame mosaic of the background scene alone (i.e., without the residuals), or alternatively the synopsis mosaic (see Section 4.2, Figure 11). Once a scene (mosaic) of interest has been detected, the part of the video tape which corresponds to it can be retrieved on demand. The mosaic images can therefore be used to index into the video data. Furthermore, the user can request retrieval of only a portion of a sequence segment (i.e., frame selection) which displays only a *certain item* in the scene (mosaic) of interest. This does not necessarily require any sophisticated search. It may reduce to the simple operation of indexing frames through pixel (or region) locations in the mosaic image according to the inverse parametric alignment transformations, picking only those frames whose fields of view contain source of that image location.

The mosaic images can also serve as a tool for higher level search in video data. For example, when searching for all frames that have a desired template in them, rather than searching frame-by-frame, one can make the search more efficient by searching the mosaic images and their "residuals" only. As this is a more efficient representation, it speeds up

the search. Furthermore, searching for a template in a mosaic image can prove to be more successful than in individual frames, as a template may appear incomplete in an individual frame, while complete and therefore easier to detect in the mosaic image. We also believe that the *synopsis* mosaic can be used as a cue for *event detection*. These, however, are yet at preliminary research stages.

5 Conclusion

The problem addressed by this paper is that of developing efficient and complete representation of large video streams and efficient methods for accessing and analyzing the information contained in the video data. In this paper, we have systematically explored the issues that arise when considering how such a complete representation may be developed. We have also described a number of different applications of the mosaic representations and illustrated them with real examples.

The work that has begun in this paper is by no means complete. Many issues remain, e.g., the development of the temporal pyramid and multiresolution mosaics, the completion of the more complex alignment models, the design and development of a complete storage system based on mosaics, etc. Finally, we are also working on developing the various applications that were outlined in this paper.

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 *European Conference on Computer Vision*, pages 237–252, Santa Margarita Ligure, May 1992.

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

[4] 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.

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

[6] M. Irani, S. Hsu, and P. Anandan. Mosaic based video compression. In *Proceedings of SPIE Conference on Electronic Imaging, Digital Video Compression: Algorithms and Techniques*, volume 2419, February 1995.

[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 *IEEE Conference on Computer Vision and Pattern Recognition*, June 1992.

[9] M. Irani and S. Peleg. Using motion analysis for image enhancement. *Journal of Visual Communication and Image Representation*, 4(4):324–335, December 1993.

[10] 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, Santa Margarita Ligure, May 1992.

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

[12] R. Kermode. Building the big picture: Enhanced resolution from coding. MSc thesis, MIT, June 1994.

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

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

[15] Rakesh Kumar, P. Anandan, M. Irani, J. R. Bergen, and K. J. Hanna. Representation of scenes from collections of images. In *Workshop on Representations of Visual Scenes*, 1995.

[16] S. Mann and R.W. Picard. Virtual bellows: Constructing high quality stills from video. In *IEEE Int. Conf. on Image Proc.*, November 1994.

[17] P.C. McLean. Structured video coding. MSc thesis, MIT, June 1991.

[18] S. Peleg S. Hsu, P. Anandan. Accurate computation of optical flow by using layered motion representations. In *Proc. ICPR94*, 1994.

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

[20] A. Shashua and N. Navab. Relative affine structure: Theory and application to 3d reconstruction from perspective views. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 483–489, Seattle, Wa., June 1994.

[21] Richard Szeliski. Image mosaicing for tele-reality applications. Technical Report CRL 94/2, Digital Equipment Corporation, 1994.

[22] L. Teodosio and W. Bender. Salient video stills: Content and context preserved. In *Submitted to ACM Multimedia '93*, 1993.

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