Representation of Scenes from Collections of Images

Rakesh Kumar, P. Anandan, Michal Irani, James Bergen, Keith Hanna
David Sarnoff Research Center
CN5300, Princeton NJ 08543-5300
Email: rkumar@sarnoff.com

Abstract

The goal of computer vision is to extract information about the world from collections of images. This information might be used to recognize or manipulate objects, to control movement through the environment, to measure or determine the condition of objects, and for many other purposes. The goal of this paper is to consider the representation of information derived from a collection of images and how it may support some of these tasks. By "collection of images" we mean any set of images relevant to a given scene. This includes video sequences, multiple images from a single still camera, or multiple images from different cameras. The central thesis of this paper is that the traditional approach to representation of information about scenes by relating each image to an abstract three dimensional coordinate system may not always be appropriate. An approach that more directly represents the relationships among the collection of images has a number of advantages. These relationships can also be computed using practical and efficient algorithms.

This paper presents an hierarchical framework for scene representation. Each increasing level in the hierarchy supports additional types of tasks so that the overall structure grows in capability as more information about the scene is acquired. The proposed hierarchy of representations is as follows: (1) The images themselves (2) Two dimensional image mosaics. (3) Image mosaics with parallax and (4) Layers and tiles with parallax. We develop the algorithms used to build these representations and demonstrate results on real image sequences. Finally, the application of these representations to real world problems is discussed.

1 Introduction

The visual information present in a collection of images is due to a combination of camera geometry, scene layout, illumination, and reflectance properties of surfaces. In order to be useful for various tasks that require visual information, a representation of the scene should be invariant to those factors that are irrelevant for that task. In the *image understanding* approach to vision, the individual coordinate systems of images in the collection of views are each related via a transformation to the single 3-d coordinate system of the world model. In addition to the global transformations of coordinate systems, this approach usually involves definition of explicit object models which are also expressed in the world coordinate system.

There are three problems with a three dimensional world model as a representation of visual information; these problems are distinct but related. The first problem is that 30 years of vigorous effort has successfully demonstrated the difficulty of computing 3-d object representations from images. The second problem is that the information in an *arbitrary* collection of images may not be sufficient to build the representation. Thus, an object level representation must remain mute until a certain minimum level of information (or minimum collection of images) is available. In many situations, this minimum level may be quite high. The third problem is that an object model may not be the most effective representation for performing the diverse tasks demanded of artificial vision systems. For example, synthesizing a view of a scene from some new viewpoint to support change detection or tasks such as obstacle detection can in principle be accomplished by constructing (and analyzing) a 3-d model of the environment, but may be more easily and reliably performed by a more direct examination of image properties [BAHH92].

Another practical difficulty is that relating new views to a 3D scene model requires matching image information with abstract model information. This matching problem is difficult because the types of information are very different and because abstract scene models frequently lose potentially valuable information about photometric and textural properties. Matching is more easily achieved using an image based scene representation where the photometric information of the representative images and mosaics have been preserved and the matching can be done in an image coordinate system.

In order to avoid some of the problems encountered in construction of object level representations, we propose an approach to scene representation that begins with the source images themselves, and maintains those basic data as an integral part of the representation. We will gradually build up a network of transformations that describe the relationships among the image coordinate systems, and their relationships to invariant three dimensional properties of objects in the scene. The remainder of this paper describes our approach to the representation of scenes from collections of images, the methods of computing it from images, and some illustrative applications of this representation.

2 Representation Framework

In this section we present an example of a hierarchical framework for scene representation that has some of the properties described above. Each increasing level in the hierarchy supports additional types of tasks so that the overall structure grows in capability as more information

about the scene is acquired. No claim is made that this framework is unique, but it will be argued that it has certain desirable characteristics with respect to a number of practical vision applications. The framework can be seen as an extension of several previous types of descriptions [BAHH92, LF94, KAH94, Sze94]. The proposed hierarchy of representations is as follows:
[1] The images themselves, [2] Two dimensional image mosaics, [3] Image mosaics with parallax, [4] Layers and tiles with parallax.

2.1 The images themselves

This is the representation of a single image, or of any collection of images that cannot be determined to overlap in scene content. Without assuming the availability of non-image information, we cannot say much more about a single image than what its pixel values are. A number of useful things can be computed from an image (image pyramids, feature maps, etc.) but they do not directly give much information about scene geometry.

2.2 Two dimensional image mosaics

Image mosaics are collections of overlapping images together with coordinate transformations that relate the different image coordinate systems. By applying the appropriate transformations via a warping operation and merging the overlapping regions of the warped images, it is possible to construct a single image covering the entire visible area of the scene. This merged single image is the motivation for the term "mosaic". We refer to this representation as "two dimensional" because the images are related by 2D coordinate transformations, not because the scene is assumed to be two dimensional. Two types of collections of images can be represented exactly by 2-d image mosaics: images taken from a single viewpoint by panning and zooming the camera and images in which the entire scene can be described by a parametric surface model (e.g. a planar surface).

Image mosaics essentially allow one to compensate for differences in viewing geometry. Thus they can be used to simplify vision tasks by simulating the condition in which the scene is viewed from a fixed position with a single camera. Mosaics are therefore quite useful in tasks involving motion or change detection since complicating effects of camera motion are reduced. Image mosaics can be used to determine the relative pose (visual direction and scale) of new images that are acquired. They can be used to determine what parts of the scene visible from that point have been observed. Thus, mosaics can be used to support active exploration, as well as to construct concise representations of scenes.

2.3 Image mosaics with parallax

For a general collection of views of a general scene, it is not possible to compute a parametric image coordinate transformation that will compensate for differences in viewing geometry. However, it is possible to represent these collections of images in the form of a mosaic in which portions of the scene that lie on a real or virtual parametric surface are aligned, together with a *parallax field* that relates deviations from this alignment to the distance of points from that surface. The 2-d mosaic representation corresponds to the special case in which this parallax field is everywhere zero.

The motivation for use of this "surface plus parallax" representation is that (as with the 2-d mosaic representation) as much as possible of the effects of changes in sensing geometry are compensated. For example, changes in camera orientation and zoom are compensated by the 2-d transformation that aligns the reference surface. The effects of changing camera position are also compensated by this transformation for point on the surface; the residual change in points not on this surface is represented by the parallax field itself.

Mosaics with parallax allow enhanced change detection since, unlike 2-d mosaics, they do not in general confound deviation from the reference surface with changes or motion of objects. Based on a mosaic plus parallax representation, one can estimate the pose of a new view of the scene with respect to the reference view. The parallax field also provides information about the 3-d structure of the scene relative to the reference surface. This reference surface need not be a physical surface in the scene: a virtual surface in space can serve to parse out the viewpoint variations referred to in the previous paragraph.

2.4 Layers and tiles with parallax

Tiles and layers allow representation of scene segments with respect to multiple surfaces. This provides for transparency and greater scene complexity, as well as for representation of extended scenes. When the 3D scene begins to be cluttered with objects at widely varying depths, and/or when real or "picket-fence" transparency is present, the parallax based representation of 3D is inadequate. A natural extension to the 2D mosaic is to use multiple layers of 2D mosaics in the manner suggested by Adelson[Ade91]. Each 2D layer can be augmented with parallax information for surface elements that are physically near that layer. In addition, the layered approach can also be used to represent multiple moving objects in the scene.

Another extension that is necessary in order to handle extended fields of view is a "tiled" representation. To motivate this, consider the case 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 view will lead to the loss of image information from other views. To handle this situation, we propose to use a series of "tiles" that correspond to different imaging planes 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 least distortion.

3 Estimation of coordinate transformations between images

Construction of the different representations forming the proposed hierarchy involves estimation of parametric and quasi-parametric transformations relating pairs of images [BAHH92]. We will begin by describing a robust and efficient approach to this computation, then we will present algorithms for producing a sufficient set of transformations to relate all of the images for various cases of scene structure and camera/ scene motion.

Consider two camera views, one denoted as the "reference" camera, and the other the "inspection" camera. A

3D point $\vec{P}_1$ in the reference camera coordinate system gets mapped to the 3D point $\vec{P}_2$ in the inspection camera coordinate system by a rigid body transformation:

$$\vec{P}_2 = R(\vec{P}_1) + \vec{T}_2 = R(\vec{P}_1 - \vec{T}_1) \tag{1}$$

The mapping can be represented by a rotation (R) followed by a translation ($\vec{T}_2$) or by a translation ($\vec{T}_1$) followed by a rotation (R).

To align two images (an "inspection" image and a "reference" image), we extend the hierarchical direct registration techniques described in [BAHH92] with different image motion models. 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:

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

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 motion vector at that point. 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 motion field $\{\mathbf{u}\}$ can be modeled by a set of global and local parameters. This modeling is simple for flat scenes or for camera motions such as panning and zooming and more complicated for arbitrary 3D motions and scenes. We derive below a set of increasingly complex algorithms for the various cases.

The general framework for all the direct algorithms is the same. Levenberg-Marquardt minimization is applied to the objective function described in 2 in order to estimate the unknown motion parameters and the resulting motion field $\{\mathbf{u}\}$. Starting with some initial values (typically zero), the hierarchical estimation algorithm iteratively refines the parameters in order to minimize first the SSD error at a coarse resolution, then successively at finer resolutions. After each step of the iteration, the transformation based on the current set of parameters is applied to the inspection images, in order to reduce the residual displacement between the images. 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 sensitive to outliers. However, as reported in [BAHH92] 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. To further improve rejection of noise and unmatched structure, we have also developed robust versions of the above least squares algorithms.

3.1 2-D motion fields

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

$$\begin{aligned} u_p(\mathbf{x}) &= p_1 x + p_2 y + p_5 + p_7 x^2 + p_8 xy \\ v_p(\mathbf{x}) &= p_3 x + p_4 y + p_6 + p_7 xy + p_8 y^2 \end{aligned} \tag{3}$$

Note for discrete views with significant camera rotation, a full 8 parameter projective transformation is required to align a planar surface. However, for closely related views such as those obtained from a video sequence we have found that the above quadratic transformation is a good approximation and more stable to compute. The eight parameter transformation aligns a planar surface undergoing instantaneous rigid motion. It is also valid for general 3D scenes and there is pure camera rotation and/or zoom.

The expressions for (u_p, v_p) from equation (3) are substituted into equation (2) to obtain the complete objective function. This function is minimized using the direct hierarchical registration technique to estimate the quadratic image motion parameters $(p_1, \ldots, p_8)$. We refer to this algorithm as the "Quad" algorithm. A version of the "Quad" algorithm which computes only affine motion parameters is refered to as the "Affine algorithm"[1].

3.2 3-D motion fields

Traditionally, in dynamic image analysis the 3D motion field has been parametrized in terms of rotational and translational motion and depth fields. However, aligning the images using this parameterization requires knowledge of the intrinsic camera parameters such as focal length and optical center. In many of our applications, this information is typically not available or provided to us. In [KAH94], we developed an alternate parametrization which does not require this knowledge. The parameterization is based on the image motion of a 3D plane and the residual parallax field.

In [KAH94], we proved the following theorem: Given two views (under perspective projection) of a scene (possibly from two distinct uncalibrated cameras), if the image motion corresponding to an arbitrary parametric surface 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[2].

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 (3)) and the residual parallax motion vector (u_r, v_r).

$$(u, v) \quad = \quad (u_p, v_p) + (u_r, v_r) \tag{4}$$

The residual vector can be represented as:

$$\begin{aligned} u_r(\mathbf{x}) &= \gamma(fT_{2x} - xT_{2z}) \\ v_r(\mathbf{x}) &= \gamma(fT_{2y} - yT_{2z}) \end{aligned} \tag{5}$$

where $\gamma = H/P_z T_\perp$, H is the perpendicular distance of the point of interest from the plane and P_z is its depth. $T_\perp$ is the perpendicular distance from the center of the *first* camera 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 [KAH94, Saw94, SN94]

[1] We often compute a global set of affine parameters and use the estimated parameters as an initial estimate for the quadratic.

[2] Aligning the reference surface by warping the inspection image to compensate for the motion of a parametric surface removes all of the rotational components of the image motion.

it was shown that the parallax field is a "relative" affine invariant.

The computation of the parallax information can proceed in one of two ways. The first algorithm "P-then-P" takes a *sequential registration* approach, in which the image regions of a physical plane in the scene are first registered and the residual parallax is then estimated as a separate step. The second algorithm "P-and-P" simultaneously estimates the planar and parallax motion components.

3.2.1 "P-then-P": Sequential registration

In the first step, the "Quad" registration algorithm is used to estimate the motion parameters $(p_1 \ldots p_8)$ for a region in the image corresponding to a physical plane in the scene. After the plane is aligned in this fashion, the parallax vectors and the direction of translation are simultaneously estimated using a quasi-parametric alignment technique. The expressions for (u, v) from equation (4) are substituted into equation (2) to obtain the complete objective function. The parameters for (u_p, v_p) computed by the first step are substituted into the SSD error function. The resulting function is then minimized using the direct hierarchical technique to solve for the direction of translation $\mathbf{T_2}$ and the parallax vector field γ.

3.2.2 "P-and-P": Simultaneous registration

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. The expressions for (u, v) from equation (4) are substituted into equation (2) to obtain the complete objective function. The resulting function is then minimized using the Levenberg Marquardt algorithm to solve for the planar motion parameters $(p_1 \ldots p_8)$, direction of translation $\mathbf{T_2}$ and the parallax vector field γ. We normalize the estimated parameters at each iteration so that the planar registration parameters obtained correspond to a virtual 3D plane which gives rise to the smallest parallax field (the average plane of the 3D scene).

3.3 Algorithm for independent motion detection

We have now described three image registration algorithms: "Quad", "P-then-P" and "P-and-P". Each algorithm optimizes an increasingly more complex objective function. We illustrate the performance of these algorithms by developing a two frame algorithm which detects independent motion or change in images accquired by a camera moving arbitrarily in 3D space.

We employ the registration algorithms in sequence, using the output of each as an initial estimate for the next. An overall algorithm for change detection and registration given two views of a general 3D scene and for arbitrary 3D motion is as follows:

1. Align the two images with the "Quad" registration algorithm described in Section 4.1 to compute an estimate for the planar motion parameters $(p_1 \ldots p_8)$.
2. Using the estimate for the quadratic plane parameters, align the two images by the algorithm "P-then-P" to compute a translation direction and parallax field.
3. Test to see if majority of the image is aligned by the algorithm "P-then-P". If yes, stop and label unaligned areas as possible areas of change. If no, proceed to step 3.
4. Use "P-and-P" to refine the parameters estimated by the algorithm "P-then-P" and align the inspection image to the reference image.
5. Label unaligned areas as possible areas of change.

In steps 3 and 5 above, we test for alignment by computing the magnitude of the normal flow [BAHH92] between the aligned image and the reference image. Regions with normal flow above a certain threshold (typically 0.5 or so) are labeled as unaligned or changed.

3.3.1 Example: Independent motion detection

Figure 1 shows the 8'th and 13'th images from a sequence of 24 images taken from a helicopter flying over a road and parking lot. The images have been digitized from an NTSC video tape at 30 frames a second and to make the sequence of 24 frames, we took every 50'th frame in the sequence. We do not have any knowledge of the intrinsic parameters of the camera used to grab these images. The images are of dimension 320 by 240 pixels.

As can be noted from the original images, motion in the image is due to both the 3D camera motion and the motion of the independently moving cars on the road and a person walking on the upper right side of the image. There is also change due to specularities appearing in different images and appearance and disappearance of occluded regions. Our change detection algorithm is insensitive to changes caused by the 3D motion of the camera and is able to correctly identify areas of change due to independently moving objects. We use the following heuristic to minimize detecting areas of change due to specularities and occlusion. For any image in the sequence, we detect changes by comparing it to both the previous and next frames in the sequence. The final areas of change are those which are found in both cases. This simple heuristic works quite well in eliminating many areas of change which are view-point dependent such as specularities and occlusions.

We used the change detection algorithm described above to register the images. The algorithm was able to align static regions of the scene to within 0.2 pixels. For each corresponding image in Figure 1, the detected areas of changes appear in black and have been boxed in Figure 2. As can be noted both the independently moving cars and moving person (top right in the images) have been detected. We have also detected a few small areas of change due to specularities from the metallic car bodies. Although we show

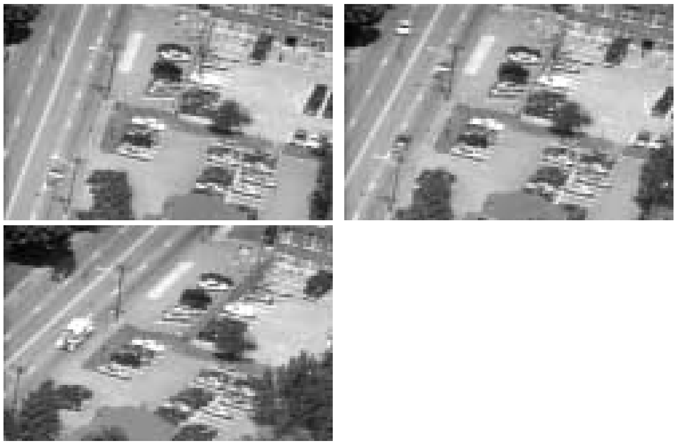

Figure 1: **Three frames from original helicopter sequence**

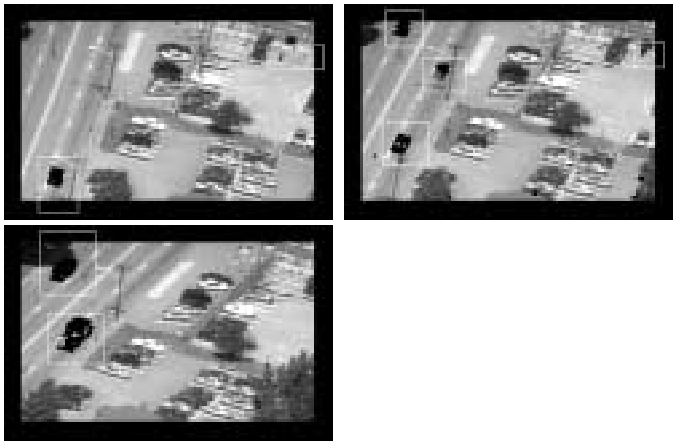

Figure 2: **Change detection: detected moving objects have been boxed and shown in black**

results only for two frames in this sequence, we were correctly able to identify all moving objects in the entire 24 frame sequence.

3.4 Pose Estimation: "direct-pose"

Given a reference and inspection image, we can use algorithms "P-then-P" and "P-and-P" to register them and compute a parallax map. The reference image and the parallax map then serve as an initial representation of the 3D scene. Given a new image taken from a new viewpoint, we wish to compute the pose of this new viewpoint with respect to the reference view. The pose of the new frame can be represented by the 8 planar motion parameters $(p_1, \ldots, p_8)$ and 3 translation parameters (T_{2x}, T_{2y}, T_{2z}) as described in equations (4) and (5). To compute the pose parameters, we use the direct hierarchical technique outlined above. We again minimize equation (2) using equation (4) to estimate the 11 pose parameters given the parallax field γ and the parallax field. We refer to this algorithm as "direct-pose" and will demonstrate its application in Section 4.2. In contrast to feature based pose estimation algorithms which must solve combinatorially intensive matching problems, the direct-pose algorithm is computationally very efficient. It uses the coarse to fine search strategy over the image pyramid to align a new image with a 3D representation.

Note, the algorithm "direct-pose" can be used for change detection. In this scheme, an extended 3D mosaic based scene representation is built. New images are then aligned with this representation and misaligned regions signal possible areas of change. In future work, we plan to develop search algorithms which directly align new images with very large 3D mosaics.

4 Construction of mosaic representations

Given a collection of images of a scene, we wish to build an extended scene representation. We should be able to build this scene representation incrementally and also be able to align and orient new images with this scene representation. An important aspect of the mosaic-based framework is that we store the photometric information (i.e. the gray values) associated with the various mosaics and images. The presence of this photometric information greatly simplifies the problem of registering new views to the current scene representation.

4.1 2D Mosaics

Construction of a 2-d mosaic requires computation of alignment parameters that relate all of the images in the collection. For image collections corresponding to video sequences, the computation of transformations is best structured in one of the following ways: (i) Successive images are aligned, the parameters cascaded to determine the alignment parameters between any two frames, (ii) Each image is aligned directly to the current composite mosaic image using the mosaic coordinate system as the reference, or (iii) the current composite mosaic image is aligned to the new image, using the new image as the reference. We use the "Quad" algorithm described in Section 3 to relate any pair of images.

The result of any of these processes is a set of transformations linking (directly or indirectly) to coordinate systems of all of the relevant images. An example is shown in Figures 3 and 4. In this case, the images form a sequence in which the camera (mounted on an aerial vehicle) moves over an extended segment of terrain while also rotating and zooming. The input frames of the video sequence are shown in Figure 3. The input images are of size 352 x 224 pixels. A 2-d mosaic representation which allows construction of the **map-like panoramic view** is shown in Figure 4. The output mosaic is of size 4814 x 1784 pixels. It has been reduced in resolution to print on the 8.5 x 11 inch pages used in this paper.

4.2 3D Mosaics plus parallax

In this section, we explain how to extend the 2D image mosaics to handle 3D scenes via parallax information. Given the 2-D planar surface alignment (either a real or a virtual plane) and the residual parallax map between each pair of images, we already have a complete representation of the relationship between all the images. However, such a representation is rather abstract, and any efficiency of

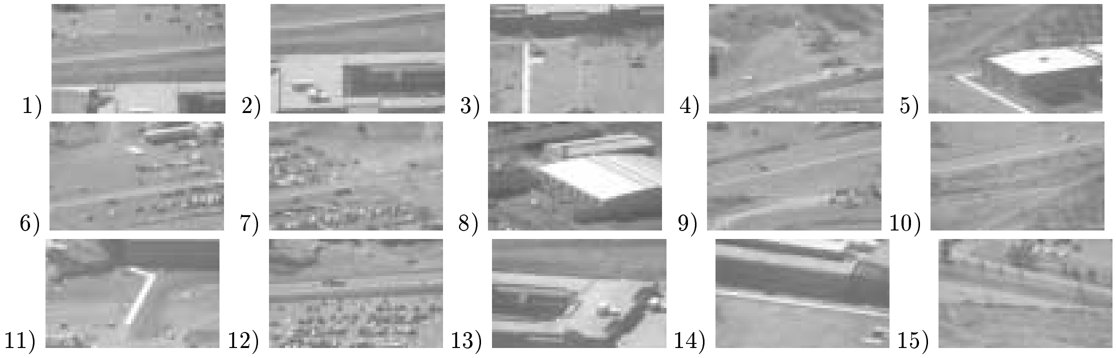

Figure 3: (15) frames sampled out of the 1800 frame (1 minute) sequence obtained by a camera mounted on an airplane. The camera is undergoing translation, rotation, and zoom (in and out).

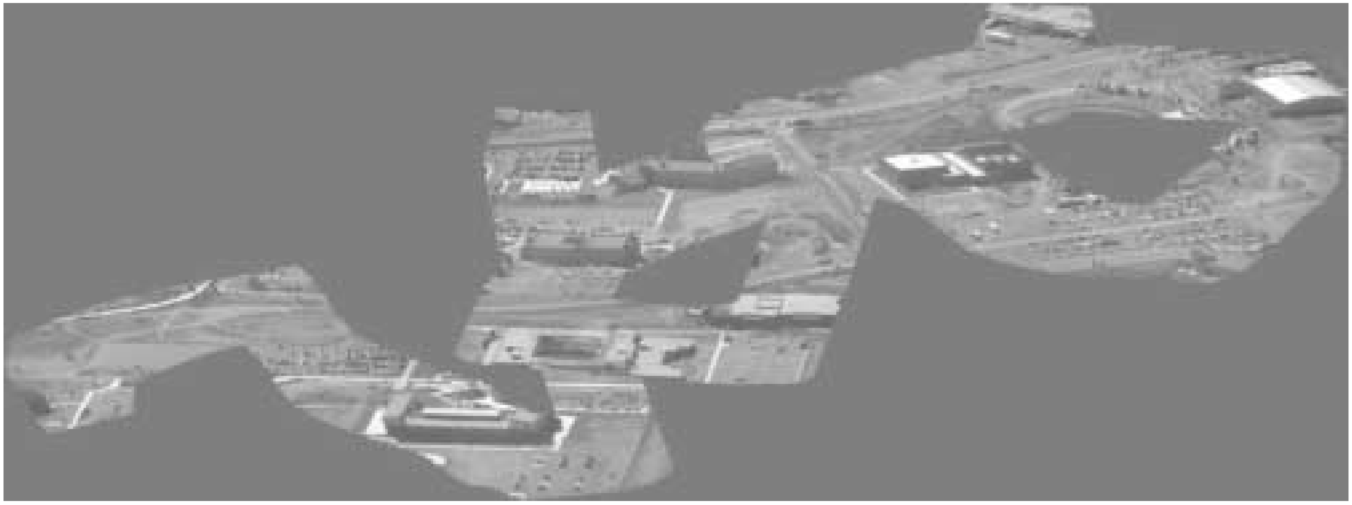

Figure 4: Panoramic mosaic image of a background scene captured by a 1-minute flight sequence. The panoramic mosaic image is constructed using a temporal average.

representation gained by relating the available view is left implicit. In order to illustrate our approach to scene representation in a more concrete fashion and to demonstrate its efficiency, we consider a type of visualization of this representation that we call "3D corrected mosaics". 3D corrected mosaics consist of two parts: A mosaic image, which assembles the various images into a single (real or virtual) camera view, and a parallax mosaic corresponding to that view.

To construct a 3D corrected mosaic, at least three views of the scene are needed. The three views should partially overlap with each other. Given such views, the process of construction involves the following steps:

[1] Use algorithms "P-and-P" or "P-then-P" to register the second and third images and to build a parallax map in the second frame's coordinate system.

[2] Use the "direct-pose" algorithm to compute the 11 pose parameters $(p_1, \ldots, p_8)$ and (T_{2x}, T_{2y}, T_{2z}), which register the second image and parallax map with the first image.

[3] Create a synthetic image taken from the first viewpoint by reprojecting the second image using the estimated pose parameters. The reprojection is done by forward warping the second image using equation (4). Note, we avoid generating any holes in the synthetic image due to forward warping by adaptively super-sampling the second image. The reprojection process must also be sensitive to occlusion and the technique we use for that is described below.

[4] Merge the synthetic first image with the first image to create a new mosaic. The synthetic image contains image regions common to the second and third images but not present in the first image. These new regions get added to the first image and extend its boundaries to create the 3D mosaic.

To construct the parallax mosaic, we forward-warp the parallax map to the first image coordinate system, much the same way as the second image was reprojected. The reprojected parallax map is merged with this additional parallax information to complete the mosaic. As part of our future work, we plan to develop Kalman filter based

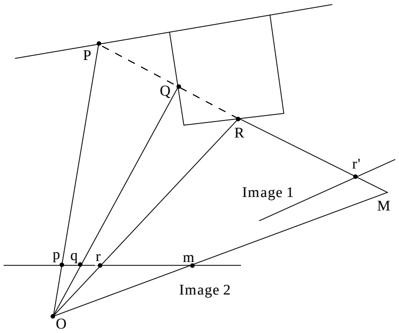

Figure 5: **Occlusion detection by pixel ordering.**

algorithms which sequentially integrate the parallax maps created by subsequent pairs of frames.

4.2.1 Occlusion detection

Finally due to occlusion, in creation of the synthetic image (Step 3 above) more than one image point in the second image may project to the same point in the synthetic image. This can be seen in Figure 5 where the 3D points "P", "Q" and "R" all project to the same point in Image 1. If we knew the depth map of the points in the Image 1 coordinate system then it would be easy to estimate that in Image 1, points "P" and "Q" are occluded by point "R". In our case, the parallax map does not provide this information. However, this information is provided to us by the ordering of the image points "p", "q" and "r" in Image 2. The points "p", "q" and "r" must lie on an epipolar line. The epipole "m" also lies on this line and is nearest to the point "r". Also the focal point "M" for Image 1 is in front of the focal point "O" for Image 2. This implies that the 3D point "R" occludes points "P" and "Q" in Image 1. Note if "M" was behind "O" then the ordering information to detect occlusions would have been reversed and the point furthest from the epipole "m" would have been chosen as the occluding point. In general, to create a new view, we have to make a binary decision as to whether the point "M" is in front or behind point "O". This ordering constraint for detecting occluded points is often used in stereo and was also used by Laveau and Faugeras [LF94] for generating new views using a different technique.

3D corrected Mosaic: Wall sequence The three original wall images Frames 1, 2 and 3 are shown in Figures 6, 7 and 8 respectively. Figure 9 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 9), there are many ghost (duplicate) lines in the bottom half of the image. 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 the smearing on the book title information (e.g. Excel, Word, Getting Started) in the right bottom of Figure 9 respectively.

Figure 10 shows a 3D corrected mosaic image. In this case, using the technique described in Section 3.5, 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 8) 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 10. Note in Figure 8 one can not see the boxes entitled "Wooden blocks" or "TRY". In the mosaic image, however they appear and are present in the geometrically correct locations. The 3D mosaic is necessarily smaller than the 2D mosaic. For a region to appear in the 3D mosaic it must appear in at least two of the input images. Therefore regions in the left side of Frame 1 do not appear in the 3D mosaic shown in Figure 10.

Finally, the translation motion of camera in this data set was parallel to the image plane. This kind of motion has traditionally been a challenge for conventional structure from motion techniques because of the difficulties in disambiguating rotational motion from translational motion. However, this motion is not difficult for our algorithm since we parametrize ego motion to be the sum of a parametric planar motion component and residual epipolar parallax. The parametric planar motion combines the ambiguous rotation and translation components of motion into a single set of parameters and there is no need for disambiguation.

5 Applications of mosaic scene representations

In this section, we consider some of the applications of our approach scene representation. We consider a series of tasks that require increasingly complex levels of the representation. The set of tasks discussed below is not intended to be complete, but rather to indicate representative examples. Also, for the sake of brevity, we will not consider tasks that require only the first level of the representation, namely the images themselves.

The simplest type of tasks to consider include image stabilization, tracking, and change detection. The benefit of image mosaics for these types of applications are discussed in detail in [BA94]. Another family of applications are those that require efficiency of representation. Examples of these include video transmission, video storage and retrieval, video analysis and manipulation (e.g., for video post production environments). In all of these applications, the major obstacle for effective and efficient use of video information is the sheer magnitude of data volume.

To illustrate this family of applications, we consider one of them, namely video compression. Most of the traditional video compression methods can be broadly divided into two categories: waveform based compression, and object based compression. The former class of techniques make minimal use of the semantic information in the images. For instance, MPEG compression methods use interframe motion vectors (which *is* a type of semantic information) to decorrelate the images. But such an approach does not take advantage of the invariance of the scene structures over multiple frames. On the other hand, the object based compression techniques[MHO91] use surface and object level geometric

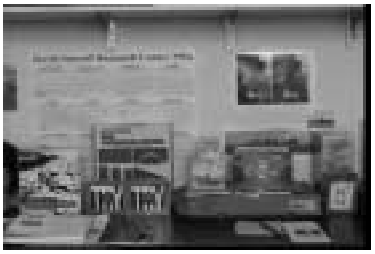

Figure 6: **Wall Frame 1**

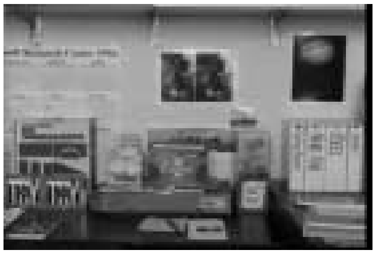

Figure 7: **Wall Frame 2**

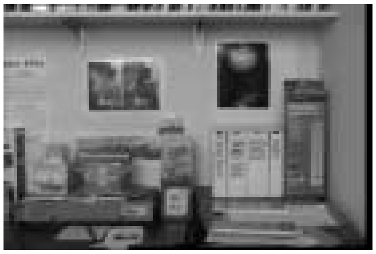

Figure 8: **Wall Frame 3**

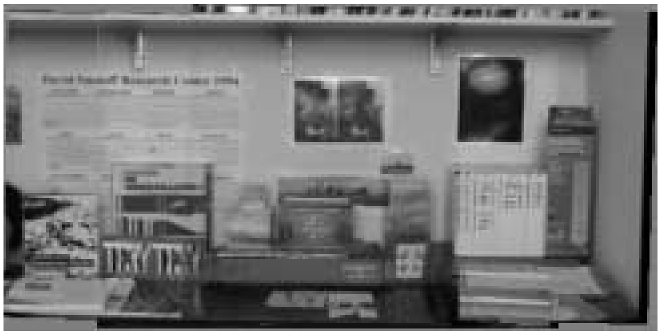

Figure 9: **2D based Mosaic constructed from the 3 Wall frames.**

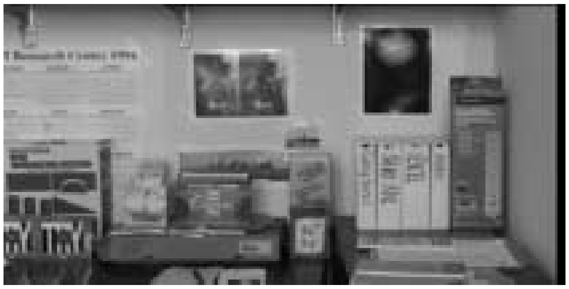

Figure 10: **3D corrected Mosaic constructed from the 3 Wall frames.**

models that can provide very high compression efficiency. However, these techniques suffer from two problems: (i) the computation of such models in arbitrary scenes is difficult, and (ii) the models are generally incomplete (e.g., may not account for photometric variations), hence the predictions are inaccurate. As a result of the dichotomy between these two categories of techniques, they tend to be applied to specific non-overlapping classes of applications. For example, the waveform based techniques are more commonly used when the compression ratios required are not high (e.g., high bitrate transmission) but the picture quality needs to be high, whereas the object based techniques are used when faithful rendering is not required, but high compression rates are needed (e.g., low bit rate applications such as video-phones).

The hierarchy of representations proposed in this paper falls somewhere between the two extremes of waveform and object based compression techniques. Since the images themselves are maintained as models, the faithfulness of rendering can be higher. On the other hand, since the representation explicates 3D scene structure, extended temporal correlations can be detected and removed, thereby leading to greater compression. Our approach to using these techniques for compression is described in greater detail in [IHA95].

The next family of applications that we consider are those that require pose estimation and matching. This requirement arises in a number of tasks such as object recognition, and spatial orientation for navigation. For these applications, mosaic representations can be used within an alignment and verification paradigm. The network of images forms the model. When a new image is compared to the model, the existing images together with the parallax information can be used to determine the pose of the new views with respect to the existing views (e.g., along the lines described in Section 4.2 for generating 3D corrected mosaics). This aligns the new image to the existing views. A direct comparison of the new image can then be made to information derived from the set of existing images.

References

[Ade91] E.H. Adelson. Layered representations for image coding. Technical Report 181, MIT Media Lab. Vision and Modeling Group, December 1991.

[BA94] P. Burt and P. Anandan. Image stabilization by registration to a reference mosaic. In *1994 Image Understanding Workshop*, volume 1, Monterey, CA, November 1994.

[BAHH92] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *European Conference on Computer Vision*, Santa-Margherita Ligure, Italy, 1992.

[IHA95] M. Irani, S. Hsu, and P. Anandan. Mosaic based video compression. In *Proceedings of SPIE Conference on Electronic Imaging*, February 1995.

[KAH94] Rakesh Kumar, P. Anandan, and K. Hanna. Shape recovery from multiple views: a parallax based approach. In *DARPA IU Workshop*, Monterey, CA, November 1994.

[LF94] S. Laveau and O. D. Faugeras. 3d scene representation as a collection of images. *Inter-*

national Conference on Pattern Recognition, A:689–691, October 1994.

[MHO91] 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.

[Saw94] Harpreet Sawhney. 3d geometry from planar parallax. In *Proc. CVPR 94*, June 1994.

[SN94] A. Shashua and N. Navab. Relative affine structure, theory and application to 3d reconstruction from 2d views. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 1994.

[Sze94] Richard Szeliski. Image mosaicing for telereality applications. Technical Report CRL 94/2, Digital Equipment Corporation, 1994.