Video Indexing Based on Mosaic Representations

Michal Irani   P. Anandan

Abstract— **Video is a rich source of information. It provides visual information about scenes. However, this information is implicitly buried inside the raw video data, and is provided with the cost of very high temporal redundancy. While the standard sequential form of video storage is adequate for viewing in a "movie mode", it fails to support rapid access to information of interest that is required in many of the emerging applications of video. This paper presents an approach for efficient access, use and manipulation of video data. The video data is first transformed from its *sequential* and *redundant* frame-based representation in which the information about the scene is distributed over many frames, to an *explicit* and *compact* scene-based representation, to which each frame can be *directly* related.**

This compact reorganization of the video data supports *non-linear* browsing and efficient indexing to provide rapid access *directly* to information of interest. The paper describes a new set of methods for indexing into the video sequence based on the scene-based representation. These indexing methods are based on *geometric* and *dynamic* information contained in the video. These methods complement the more traditional "content-based indexing" methods which utilizes image appearance information (namely color and texture properties), but are considerably simpler to achieve and are highly computationally efficient.

Keywords—**video indexing, video browsing, compact video representations, mosaics, video manipulation, video annotation, video compression, video databases.**

I. Introduction

The emergence of video as data and a source of information on the computer opens the potential for new ways of accessing, viewing and manipulating the contents of video. These include direct non-linear access to video frames and sequences of interest, new modes of viewing that gives the viewer the control over how the video is viewed, the annotation and manipulation of objects and scenes in the video, and the merging of text and graphics with the video data.

While the standard manner of representing video as a sequence of frames is adequate for viewing it in a movie mode, it does not support the type of interaction with video information described above. Currently the only way to access the information of interest is by sequentially scanning the video. The only way to manipulate, annotate, or edit the video is by processing the video frame-by-frame. This process is both slow and tedious.

This paper presents a new approach for efficient access, storage, and manipulation of video data. Our approach is based on the fact that a video sequence contains many views of the same *scene* taken over time, either from a moving or a stationary camera. Hence, the information that is common to all the frames is the scene itself. However, this information is distributed over many frames, at the cost of very high temporal redundancy, and is found only implicitly in the video data. We transform the video data from a sequential *frame-based* representation, in which this common scene information is *distributed* over many frames, into a single common *scene-based* representation to which each frame can be *directly* related. This representation then allows *direct* and *immediate* access to the *scene* information, such as static locations and dynamically moving objects. It also eliminates the redundancy between the different views of the scene contained in the frames, and results in a highly efficient and compact representation of the video information. Hence, the scene-based representation forms the basis for direct and efficient access to and manipulation of the video information, and supports efficient storage and transmission of the video data.

The scene-representation is composed of three components: (i) *extended spatial information*: this captures the appearance of the entire scene imaged in the video clip, and is represented in the form of a few (often just one) panoramic mosaic images constructed by composing the information from the different views of the scene in the individual frames into a single image, (ii) *extended temporal information*: this captures the motion of independently moving objects in the scene (e.g., in the form of their trajectories), and (iii) *geometric information*: this captures the 3D scene structure, as well as the geometric transformations which are induced by the motion of the camera and map the frames to the common mosaic image. Taken together, these three components provide a *compact* description of the video data.

We construct the common scene-based representation by measuring and interpreting the image motion within the video clip. Regions of the video frames, corresponding to the static and dynamic portions of the scene are determined. The geometric transformations and the 3D scene structure are recovered as a part of this process. This process is done automatically, without any information about the camera calibration or the scene.

Once the common scene-based representation is constructed, it forms the basis for direct and efficient browsing, indexing, and manipulation of the video data. *Browsing* is done by skimming a collection of images that "summarize" the video data. We refer to these images as *visual summaries*. These summaries visually describe the video information in a compact and succinct fashion, and can serve as a *visual table-of-contents* for the video.

M. Irani is with the Dept. of Applied Math and Computer Science, The Weizmann Institute of Science, 76100 Rehovot, ISRAEL.

P. Anandan is with Microsoft Corporation, One Microsoft Way, Redmond, WA 98052, USA.

This work was done while the authors were with David Sarnoff Research Center.

Since the mosaics capture the information that is common to all the frames, they provide the means to *directly* index into and manipulate the individual frames. Both the static and dynamic portions of the video sequence can be accessed this way. These indexing methods are based on *geometric* and *dynamic* information contained in the video. These complement the more traditional approach to "content-based indexing" which utilizes image *appearance* information (namely color and texture properties) [9], [10], [7], [26], but are considerably simpler to achieve and are computationally highly efficient. The existing appearance-based methods themselves can also be used more efficiently within the scene-based representation, when applied directly to the mosaic image (i.e., to the appearance component of our representation), rather than to the individual video frames one-by-one.

The rest of the paper is organized as follows: Section II presents the common and compact scene-based representation, to which each frame are *directly* related. Section III explains how to use the scene-based representation to efficiently and rapidly browse, index, and manipulate video data. Section IV reviews the techniques used for constructing the scene-based representation from raw video sequences. Section V concludes the paper.

II. From Frames to Scenes

Video is a rich data source. It provides information about scenes. However, this information is buried inside the raw video data, and is provided with the cost of very hight temporal redundancy (e.g., every scene point is displayed repeatedly in numerous consecutive frames). In this section we first review the fundamental components of information in a video stream (Section II-A). Then we make use of these information components to *transform* the video from an implicit and redundant *frame-based representation*, to an explicit and non-redundant *scene-based representation*, which is common to all frames (Section II-B).

A. The Three Fundamental Information Components of Video

Video extends the imaging capabilities of a still camera in three ways. First, although the field-of-view of each single image frame may be small, the camera can be panned or otherwise moved around in order to cover an extended spatial area. However, the *extended spatial information* acquired by the video is not available in a coherent form. It is distributed among a sequence of frames, and is hard to use.

Second, and perhaps the most common use of video is to record the evolution of events over time. Once again, however, this *extended temporal information* is not explicitly represented, but distributed over a sequence of video frames. While it is natural for a human to view it as a movie, this representation is not particularly suitable for analytic purposes.

Third, a video camera can be moved in order to acquire views from a continuously varying set of vantage points. This induces image motion, which depends on the three-dimensional geometric layout of the scene and the motion of the camera. However, this *geometric information* is also only implicitly present, and is not directly accessible from the standard sequential video representation.

Thus, the total information contained in the video data consists of the three *scene* components mentioned above. However, this information is distributed among the frames and is implicitly encoded in terms of image motion. Therefore, a natural way to reorganize the video data is in terms of these three scene components. Moreover, such a reorganization removes the tremendous redundancy that is present in the source video data. This *scene-based* organization is highly efficient, since it directly and *uniquely* maps onto the information in the scene. Therefore, it facilitates efficient interaction and manipulation, and supports very efficient storage and transmission.

B. The Scene-Based Representation

To bring out the *common* scene information contained in the video, and make it more directly accessible, we first transform the video from its *implicit* and *redundant* frame-based representation, to an *explicit* and *compact* scene-based representation. In this section we introduce the scene-based representation. In Section IV we elaborate on the details of the representation and explain how it is constructed from the video data.

The video stream is first *temporally* segmented into *scene segments*, which are sub-sequences of the input video sequence. A beginning or an end of a scene segment is automatically detected wherever a scene-cut or scene-change occurs in the video. The scene cuts are characterized typically by *drastic* changes in the frame content, which is directly refelected in the distribution of color and the greylevels in the image, or in the image motion (e.g., see [9], [37]). These changes are relatively simple to detect.

Each scene segment is subsequently parsed into the three fundamental components of video (see Section II-A), namely, the static background scene, the dynamic moving objects, the geometric information. These components are organized as described below.

Corresponding to the three fundamental components, the scene-based representation is divided into three parts.

1. A **panoramic mosaic image**, which captures an extended spatial view of the entire scene visible in the video clip, in a single (or sometimes few) "snapshot" image (e.g., see Figure 1). This image captures the appearance of the *static* portions of the scene.
   The mosaic image is constructed by first aligning all the frames with respect to the common coordinate system (which becomes also the mosaic coordinate system), and then integrating all these frames to form a single image. Different methods of integration can be employed (e.g., temporal average, temporal median, super-resolution, etc). These are described in more detail in [12].
   The mosaic representation removes the redundancy contained in the overlap between successive frames and represents each spatial point only once. Mosaics have

been previously used as an effective way of creating panoramic views of a scene from video sequences [23], [31], [32], [16], [20], [3]. However, until now they have not been used as an information component within a scene-based representation, which provides direct and efficient access to video data.

Section IV describes a hierarchy of mosaic representations. The hierarchy corresponds to increasing complexity levels in the camera motion and in the 3D scene structure.

2. The **geometric transformations** that relate the different video frames to the mosaic coordinate system. The geometric transformations contain the information necessary to map the location of each scene point back and forth between the panoramic mosaic image(s) and the individual frames. Corresponding to the hierarchy of the panoramic mosaic representations, there exists a hierarchy of representations of the geometric transformations. These range from global parametric 2D transformations to more complex 3D transformations, and are described in Section IV.
3. The **dynamic information**, e.g., information about **moving objects**, which are not captured by the static panoramic mosaic image. Moving object information is *completely* captured by representing the extended time trajectories of those objects, as well as their appearance. Such a complete representation is needed, e.g., for video compression (since the video frames need to be reconstructed from the scene-representation). However, to access, browse, index and annotate the video (as presented in Section III), the trajectory information alone is sufficient. The trajectory of the center-of-mass of each detected moving object (i.e., a single image-point per moving object per frame) is maintained. These trajectories are represented in the coordinate system of the mosaic image, which is common to all the frames. In the common coordinate system, time continuity, continuous tracking, and the temporal behavior of the moving object, can be analyzed more effectively (see Figures 3 and 5).

Thus, the three components of our scene-based representation form a *compact* representation of the video clip. The compactness results from the fact that every scene point is presented *only once* in the mosaic image, while in the original video clip it is observed in multiple frames. This compactness of the scene-based representation facilitates very high *compression* (and we have developed such algorithms for VLBR compression [13]). In this paper, we focus on the power of this representation for video indexing and manipulation. Section III describes how this representation can be used for efficiently accessing and manipulating the video data. Section IV describes the methods for constructing the scene-based representation.

III. From Scenes to Visual Summaries and Indexing

Once a video sequence is transformed from the *frame-based* representation to the *scene-based representation*, it forms the basis for the user's interaction with the video. The user can initially preview the video by browsing through *visual summaries* of the various video clips. These visual summaries can serve as a *visual table-of-contents* of the video data. When a scene of interest is detected by the user, he/she can either request to view only that portion of the video, or can further index into individual video frames. The detected frames of interest can then be either *viewed* or *manipulated* by the user.

A. Visual Summaries – A Visual Table of Content

There are two types of visual summaries of video clips that a user can browse through. These are captured by two types of mosaic images which are constructed from the video clip of a scene:

- **The Static Background Mosaic:**
  The video frames of a single video segment (clip) are aligned and integrated into a single mosaic image. This image provides an extended (panoramic) spatial view of the entire static background scene viewed in the clip in a single "snapshot" image and represents the scene better than any single frame. This image does not include any moving objects. The user can visually browse through the collection of such mosaic images to select a scene (clip) of interest.
  Figures 1 and 2 display some examples of static background mosaic images.
- **The Synopsis Mosaic:**
  While the static mosaic image effectively captures the background scene, it contains no representation of the dynamic events in the scene. To provide a summary of the events, we create a new type of mosaic called the *synopsis* mosaic. This is constructed by overlaying the trajectories of the moving objects on top of the background mosaic. This single "snapshot" image provides a visual summary of the entire dynamic foreground *event* that occurred in the video clip.
  Figure 3 graphically illustrates the trajectory associated with a moving object in a synopsis mosaic.
  Figure 2.c provides a summary of the entire event in the baseball video clip.
  To allow for comprehensive display of multiple trajectories (corresponding to multiple moving objects), the trajectory of each moving object is uniquely color coded.
  Figures 4 and 5 provide visual summaries of airborne (UAV) video clips each with multiple moving objects. Figure 4 shows a flying airplane and a moving car on the road. Figure 5 shows a flying airplane, three parachuters that were dropped from the plane, and a moving car.

The natural mode of operation for the user is to first browse through the visual summary mosaics to identify a few scenes of interest. Once the user has identified a scene (i.e., mosaic) of interest, he proceeds to directly access and/or manipulate individual video frames associated with only a *portion* of the scene which is of interest to him. The scene-based representation supports this type

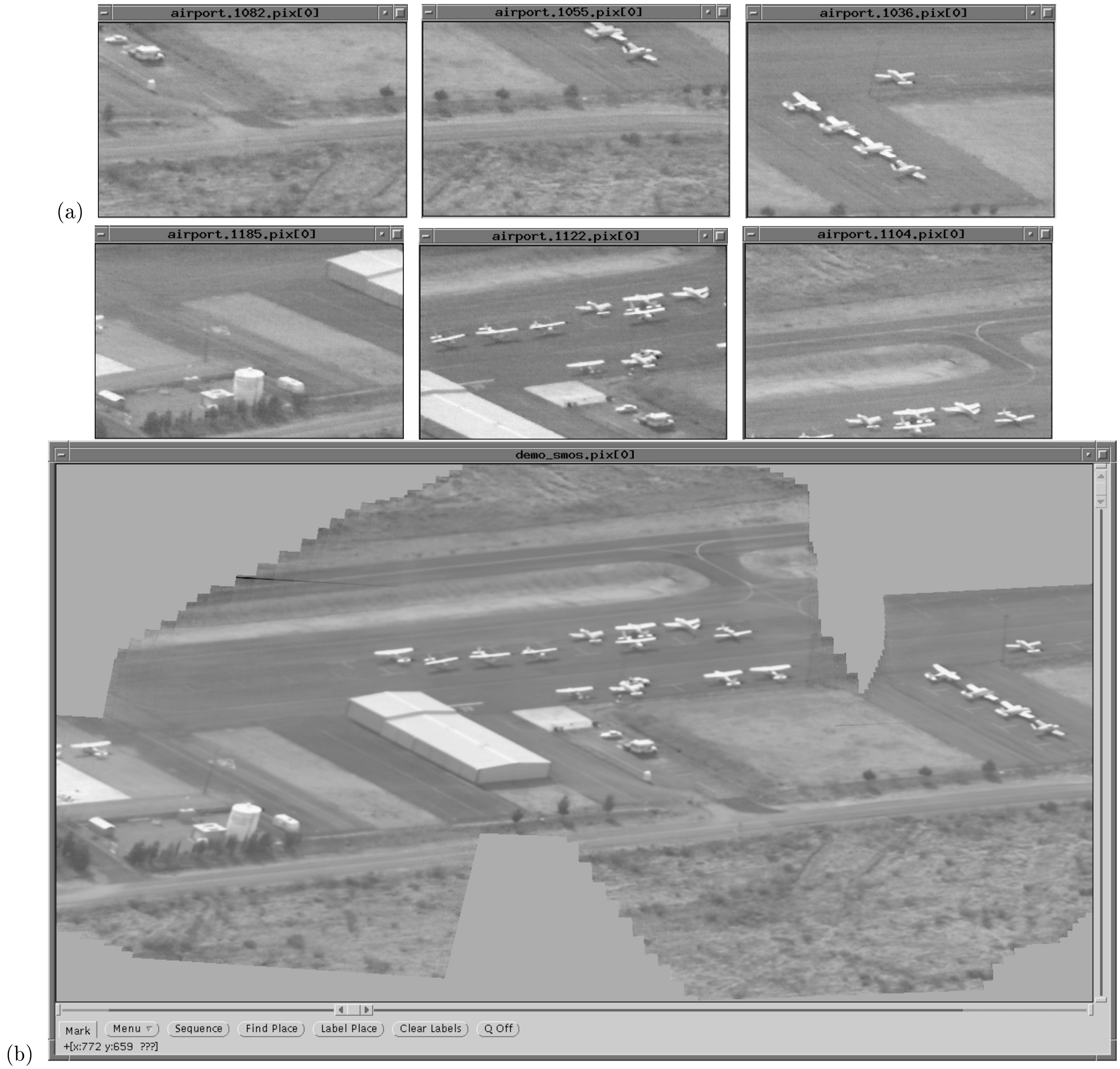

Fig. 1. Static background mosaic of an airport video clip.
(a) A few representative frames from the minute-long video clip. The video shows an airport being imaged from the air with a moving camera. The scene itself is static (i.e., no moving objects). (b) The static background mosaic image which provides an extended view of the entire scene imaged by the camera in the one-minute video clip.

of indexing. Two new types of indexing methods are presented: (i) indexing based on *location* (geometric) information, and (ii) indexing based on *dynamic* information. These are made possible directly via the geometric coordinate transformations that relate the different frames to the mosaic image, and through the moving objects information which was estimated in the formation of the mosaic-based scene representation (Section II-B). The access and manipulation of selected video frames is done directly from the mosaic-based visual summaries. These location and dynamic indexing methods complement the more traditional approach to "content-based indexing", which utilizes image appearance information (e.g., color and texture) [9], [10], [7], [26]. However, our methods are considerably simpler to achieve and are highly computationally efficient.

The remainder of this section describes these modes of video indexing and manipulation.

B. Location (Geometric) Based Indexing

Once a few scenes of interest (in the form of visual summaries) have been selected, the user proceeds to access the video frames themselves. The user selects a scene point (or

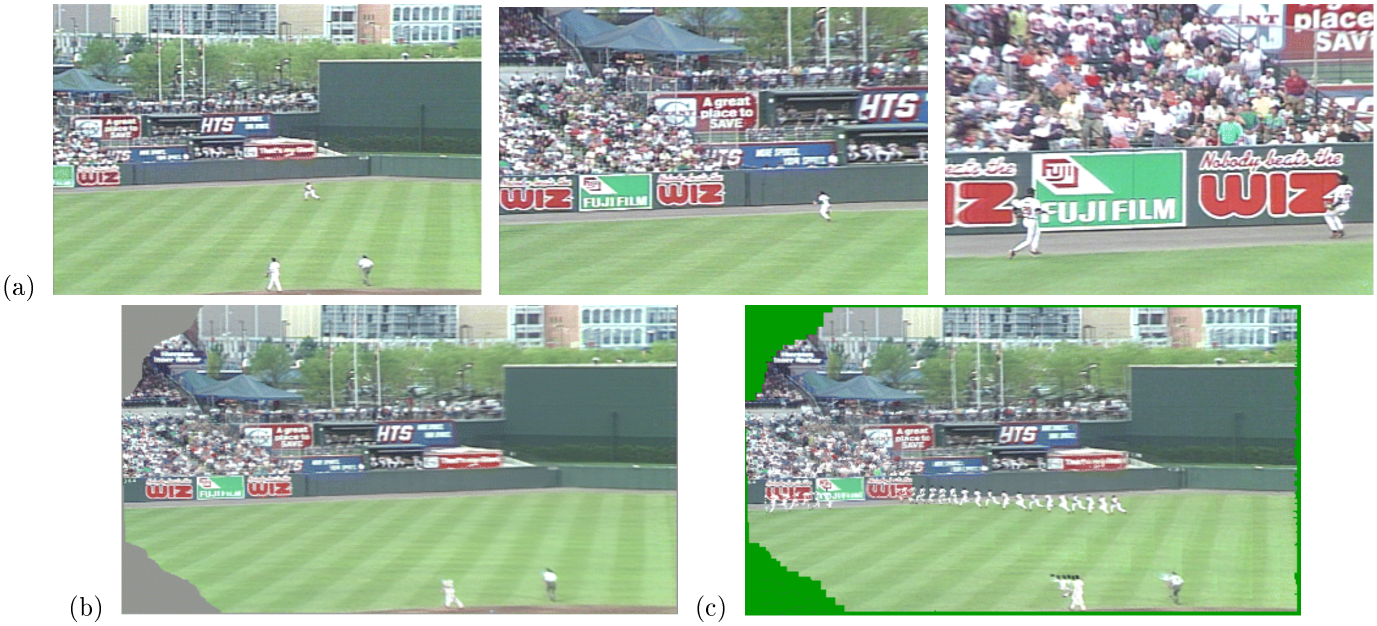

Fig. 2. Visual summaries of a baseball video clip.
(a) A few representative frames from the video clip. The video shows two outfielders running, while the camera is panning to the left and zooming on the two baseball players. (b) The static background mosaic image which provides an extended view of the entire scene captured by the camera in the video clip. The "missing" regions at the top-left and bottom-left were never imaged by the camera, because at that point it was zoomed on the two players (e.g., frame 80). (c) The synopsis mosaic which provides a visual summary of the entire event. It shows the trajectories of the two outfielders in the context of the mosaic image.

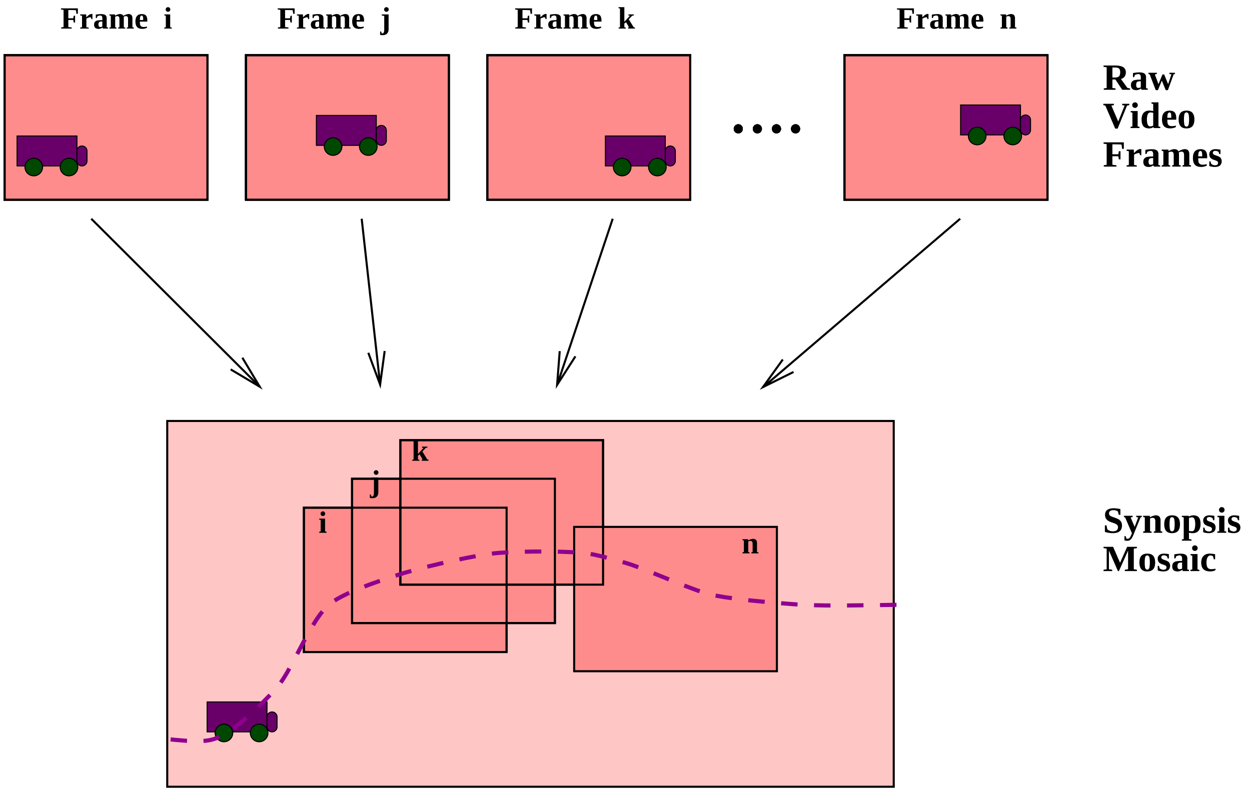

Fig. 3. Synopsis of a Moving Object.
The trajectory of the moving object is depicted in the synopsis mosaic. This shows the motion of the moving object, after cancellation of the background (camera-induced) motion. With each point on the trajectory is associated a frame number (i.e., the "time" when the moving object was at that location).

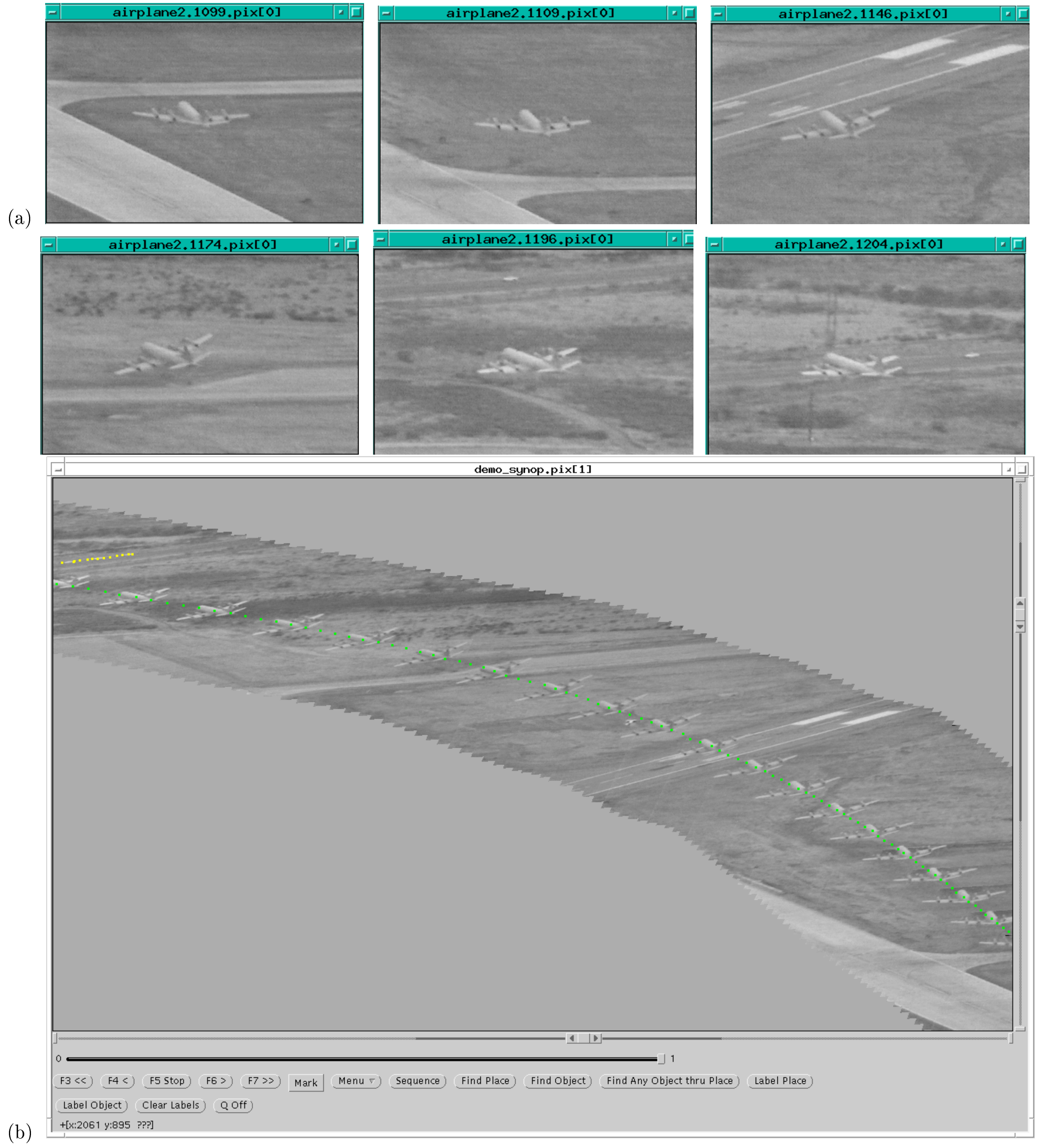

Fig. 4. The visual summary of a flying plane video clip.
(a) A few representative frames from the minute-long video clip. The video shows an airplane flying from right to left (during takeoff). A car driving on a road is visible for a few frames. (b) The synopsis mosaic which provides a visual summary of the entire video clip, showing the trajectories of all moving object in the context of the mosaic image. Each detected and tracked moving object is color coded uniquely (plane: green. car: yellow).

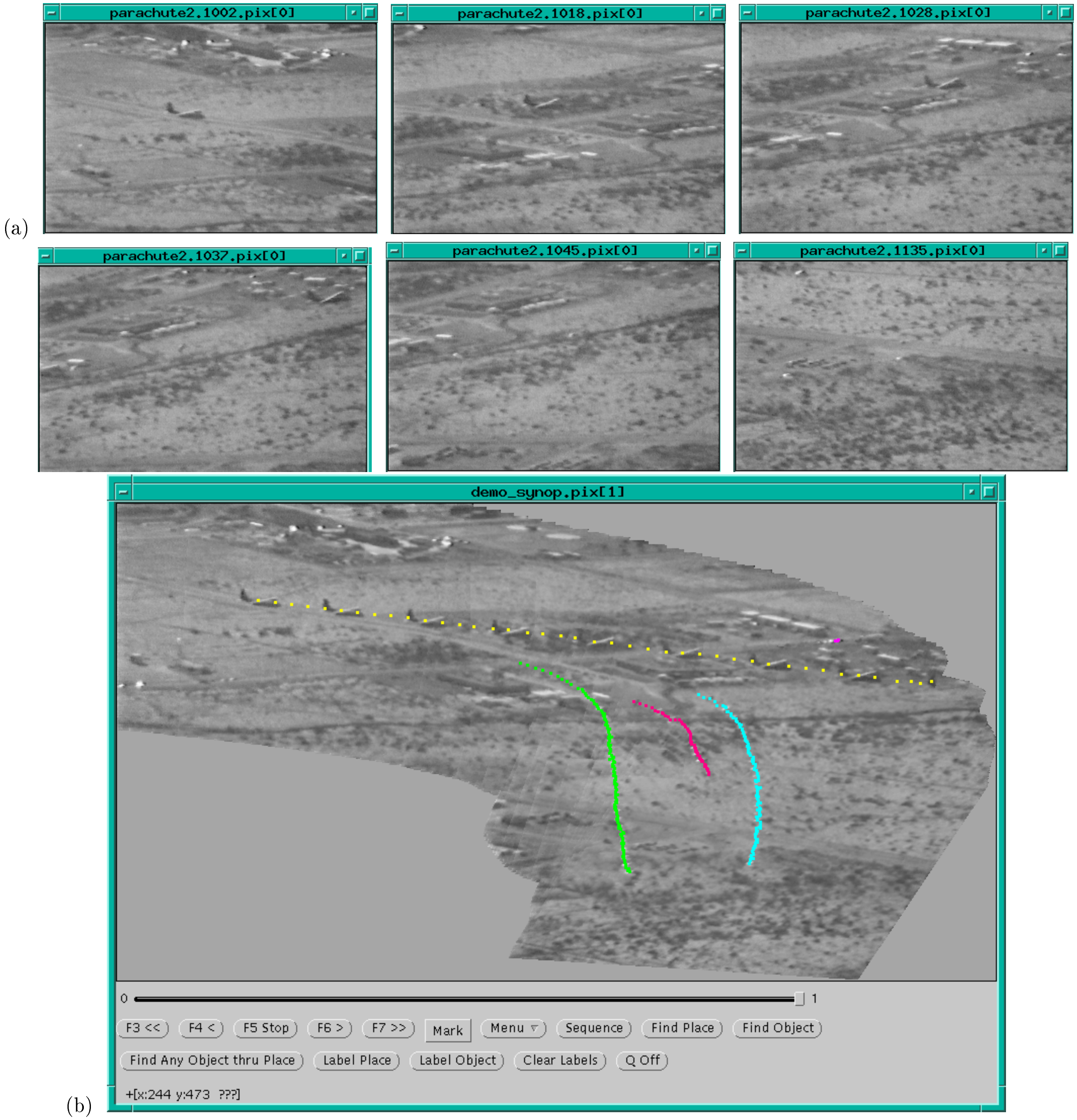

Fig. 5. The visual summary of a parachuters video clip.
(a) A few representative frames from the 30-second-long video clip. The video shows an airplane flying from left to right, dropping three parachuters. A car driving on a road is visible for a few frames. The parachuters are very small (tiny white dots) and difficult to see in a static image, but they are easily detectable in video, as they have different motion than the background. They are depicted in the synopsis mosaic by the green, red, and blue trajectories. In the video sequence, they become visible gradually, as their parachutes open – first the left parachuter, then the right one, and last the middle one. This becomes clearer in the annotated video displayed in Figure 10.
(b) The synopsis mosaic which provides a visual summary of the entire video clip, showing the trajectories of all moving object in the context of the mosaic image. Each detected and tracked moving object is color coded uniquely (parachuters: green, red, blue. car: purple. plane: yellow).

several points) in the mosaic image. The geometric coordinate transformations map the selected scene point(s) from the mosaic image to its location in the coordinate system of each of the video frames. All frames containing the selected scene point inside their field of view are therefore instantaneously determined. The user can view the subsequence of the video that contains only the frames with the selected scene point (or points). When these frames are not consecutive in time (e.g., if the selected portion of the scene was revisited by the camera multiple times), then multiple sub-sequences (corresponding to *consecutive* frame groups) are displayed to the user.

Figure 6 demonstrates an indexing process. Selection of a scene point in the mosaic image generates a display of all frames whose field of view contains the selected scene point. These are frames i, j, and k. In the figure, these frames are displayed as a collection of frames, but in reality, they are displayed as a video sequence.

In addition to manual scene-point selection, this representation also provides a basis for *efficiently* indexing into the video using existing *automatic* detection methods. For example, if a region is searched using an appearance-based detection method (e.g., template correlation, or search based on color or texture attributes [9], [10], [7], [26]), then instead of applying these search methods individually to each frame, it can be applied just *once* to the common mosaic image. Once it is detected in the mosaic image, the location-based indexing mechanism can be used to retrieve the corresponding frames.

Editing and Annotation: The compact mosaic representation can be used not only to access video frames, but also to edit, annotate, and manipulate these frames. For example, the same mechanism used for indexing is also used to efficiently inherit annotations from the mosaic image onto scene locations in the video frames.

The annotation is specified by the user just *once* on the mosaic image, rather than tediously specifying it for each and every frame. This can be further extended to efficiently edit video clips, by inserting or deleting an object in the mosaic image, hence inserting or deleting that object in all corresponding video frames.

Figure 7 graphically illustrates a video annotation process.

Figure 8 shows an example of annotating airborne video of an airport scene.

C. Dynamic (Moving-Objects) Based Indexing

Since the *synopsis* mosaic provides a snapshot view of an entire dynamic event, it can be used for indexing based on temporal events. In the synopsis mosaic, the motion of an object is represented as a trajectory in the common coordinate system, hence, the temporal event has been transformed into a spatial representation. Marking a segment on the trajectory is thus equivalent to marking a time interval, which enables access and display of all frames in this time interval.

More specifically, all frames containing a selected moving object can be immediately determined and accessed, as well as the location of the moving object in each of these frames. The user can select an object of interest whose track is marked on the synopsis mosaic. Since the trajectories of the moving objects in the mosaic coordinate system are precomputed (as well as which point on the trajectory corresponds to which frame), all frames containing that object are immediately accessed and viewed. The location of that object in each frame is estimated through the basic geometric coordinate transformations (the ones that correspond to the camera-induced motion). In a similar manner, the moving objects in the video frames are efficiently annotated or manipulated by annotating the synopsis mosaic, without the need for the user to repeatedly perform the operation on a frame-by-frame basis.

Figure 9 shows an example of annotating moving objects using the plane video, whose synopsis mosaic was shown in Figure 4. The figure displays the selected annotations on the synopsis mosaic. Representative output frames are shown, in which the annotations are automatically inherited from the mosaic. Note that the annotations "move" together with the moving objects.

Figure 10 shows an example of video annotation using the airborne parachuters video. The figure displays the selected annotations on top of the synopsis mosaic image. Both moving objects and stationary scene points are annotated. Representative frames from the automatically-annotated video clip are also displayed. Note that annotations of moving objects "move" together with the moving objects, while annotations of static scene points (e.g., "building") remain stationary with respect to the background scene (i.e., they preserve the background motion induced by the moving camera).

Note also that estimating the trajectories of moving objects in the common mosaic coordinate system allows more reliable detection and tracking of moving objects, even when they are very small (such as the three parachuters in Figure 5). This is because a "temporal coherence" constraint can be used during moving object detection and tracking after removal of the background motion. Assuming that object velocities do not change too rapidly, the detection of moving objects within each frame can be guided by the trajectory of the objects in a few previous frames. This leads to better separation between small moving objects and noise, as well as enables recovery from losing an object for a few frames (e.g., due to occlusion or bad detection). The missing portion of the trajectory is smoothly interpolate/extrapolated from the neighboring frames.

IV. Building the Scene-Based Representation

In Section II-B we introduced the basic components of the scene-based representation. In this section we provide the *details* of the scene-based representation (Section IV-A), followed by a review of the methods used for its construction (Sections IV-B and IV-C). This section serves mainly as a review of methods which have been previously published; these methods are briefly outlined here in order to make the paper self contained.

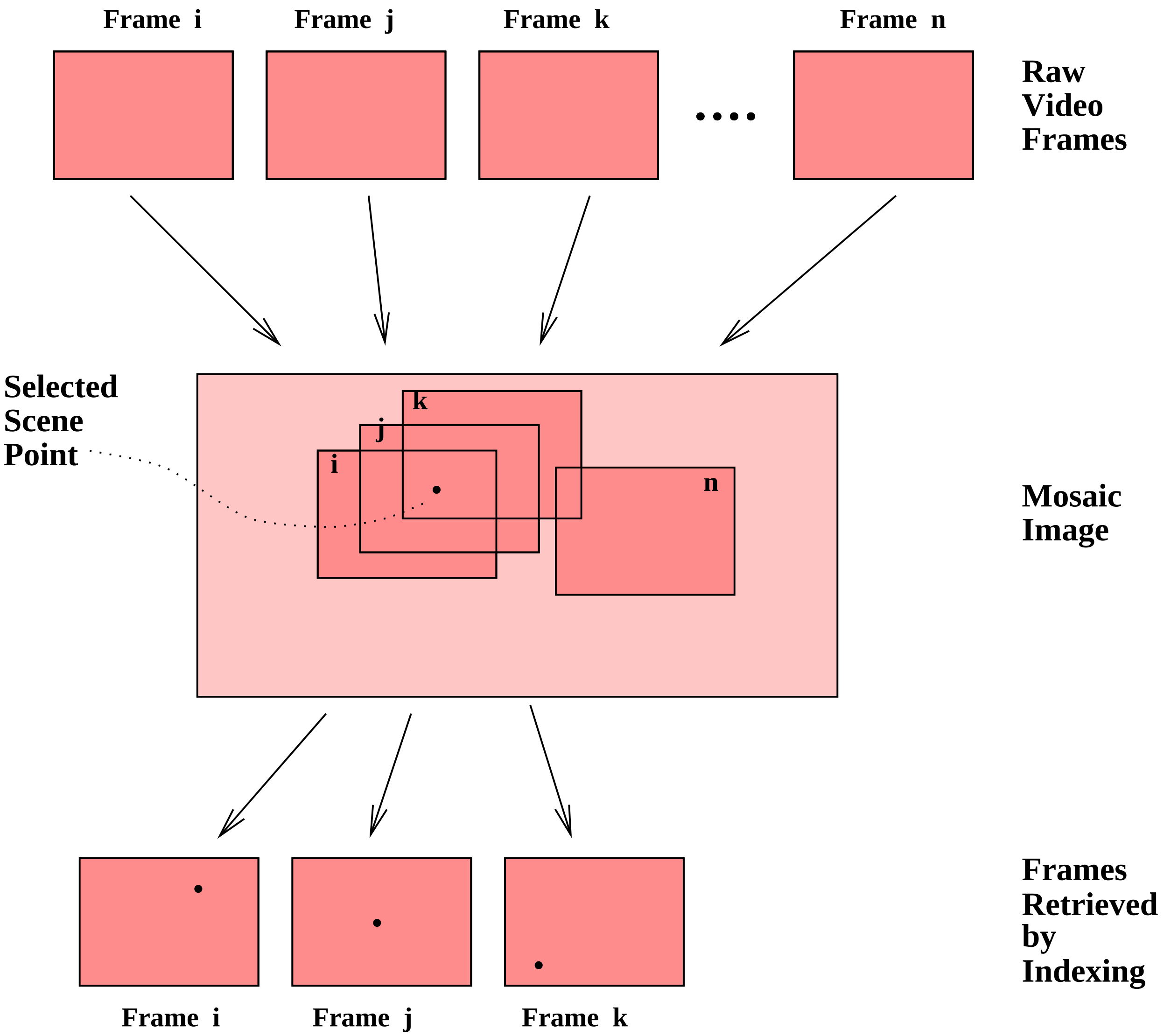

Fig. 6. Location Based Indexing.
Selection of a scene point in the mosaic image generates a display of all frames whose field of view contains the selected scene point. These are frames i, j, and k. In the figure, these frames are displayed as a collection of frames, but in reality, they are displayed as a video sequence.

A. *The Detailed Scene-Based Representation*

1. The **panoramic view** of the scene is captured by one or several mosaic images. We present a *hierarchy* of such mosaic representations. The hierarchy corresponds to increasing complexity levels in the camera motion and in the 3D scene structure:
   (a) The simplest representation is a mosaic image constructed by aligning all the frames to a single coordinate system using 2D parametric coordinate transformations. We refer to such a mosaic as a *2D parametric mosaic image*. The cases when the camera induced motion can be modeled as a 2D parametric transformation can be divided broadly into three categories (see Section IV-B.1): (i) when the translational motion of the camera is negligible, i.e., camera motion can be approximated by only 3D rotations and zooms, (ii) when the scene is planar, or (iii) when the 3D scene is sufficiently distant from the camera, such that it can be approximated by a nearly flat 2D surface. We refer to these scenarios as *2D scenes*.
   The examples given in Section III belong to this class of scenarios. For example, the baseball sequence (Figure 2) was captured by a panning camera (i.e., pure rotation), while the other sequences in that section (Figures 1, 4, and 5) were taken by an *airborne* camera, hence the scene was sufficiently distant from the camera and could be well approximated by a flat 2D surface.
   (b) The next level of complexity arises when the 3D deviations from the 2D planar surface approximation (when combined with the camera translation) results in measurable parallax image motion relative to the surface. In this case, the visual appearance of the scene is still captured by a *mosaic image* as in the previous case, while the geometric component of

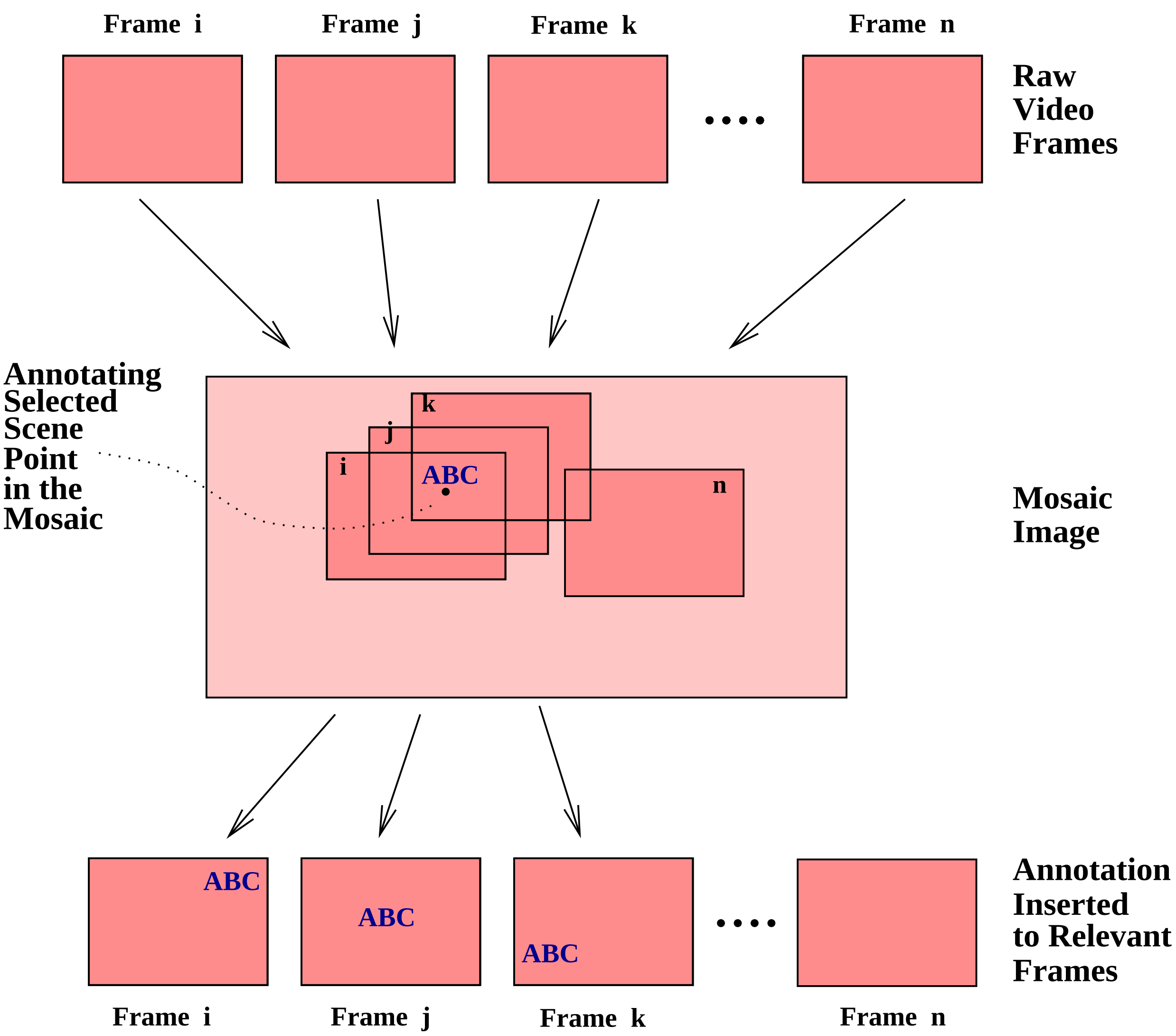

Fig. 7. Location Based Annotation.
Annotation of a selected scene point in the mosaic image leads to automatic annotation of all relevant frames (i, j, and k) with the selected annotation, and at the appropriate image coordinate, i.e., that which corresponds to the selected scene point in each of the frames.

the representation also encodes the 3D parallax relative to the planar surface (see Section IV-B.2). The parallax information is captured in the geometric component of the representation and is taken into account while combining the different frames into a single mosaic [12]. We refer to this representation as the *plane+parallax* representation. The estimation of the parallax motion is briefly described in Section IV-B.2. An example of such a mosaic image constructed from a real video sequence is shown in [12]

(c) The third level of complexity involves using multiple *layers* of *plane+parallax* representations to handle scenes that may contain surfaces at different depths. Each layer captures a collection of points in the scene that when taken together can be approximated by a planar surface with small fluctuations. Points that are not on the planes are associated with one of the layers based on their proximity in the 3D scene to those planes. The visual appearance of each layer is captured by a *plane+parallax* mosaic image as in the case above. The same approach can also be used to handle reflections and transparency.

2. The **geometric transformations** that relate the different video frames to the mosaic coordinate system contain the information necessary to map the location of each point between the panoramic mosaic image(s) and the individual frames. Corresponding to the hierarchy of the panoramic mosaic representations, there exists a hierarchy of representations of the geometric transformation. Below we briefly summarize this component of the representation. The details of their estimation are described in Section IV-B.2.

(a) For the 2D parametric mosaic, the geometric transformations consist of the *2D parametric transformations* that align each frame to the mosaic. These transformations capture the effect of rotations, translations, and zooms of the camera rela-

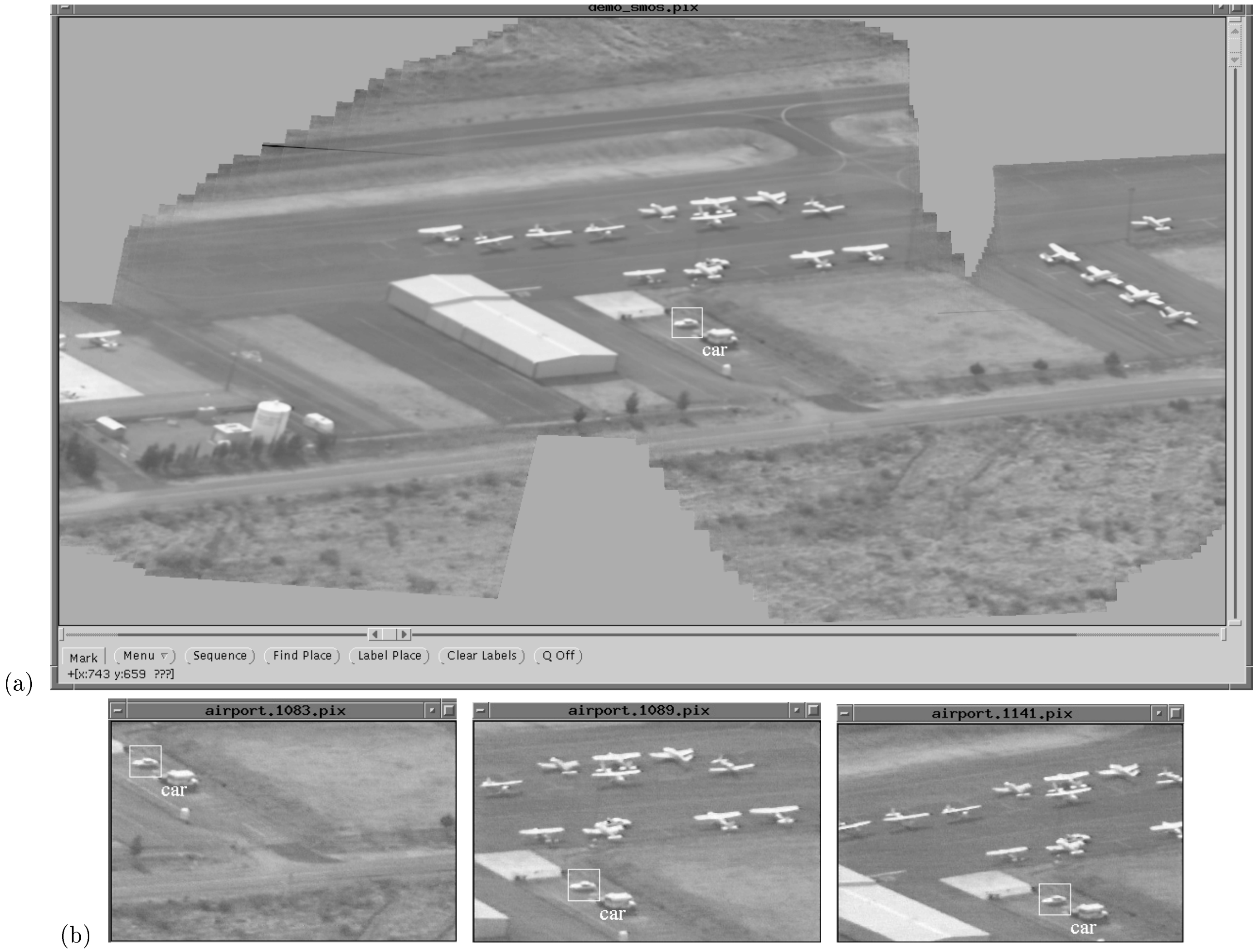

Fig. 8. Annotation of the airport video clip.
(a) A stationary car is annotated *once* on the mosaic image ("car"). (b) A few representative frames from the video clip with the annotations inherited from the mosaic image. The annotations are incorporated into the video frames *automatically* and *instantly* through the geometric coordinate transformations that map each frame onto the mosaic image. Some video frames from the raw video clip are displayed in Figure 1.

tive to a planar surface. They can be described by 6 or 8 parameters per frame. The estimation of these transformations is reviewed in Section IV-B.1.

(b) The plane+parallax representation requires, in addition to the parametric transformation that aligns a dominant plane in the scene, the information required to describe the *3D parallax* of the points that deviate from the plane. The residual parallax displacements after 2D alignment, depend both on the 3D distance the scene points from the plane, as well as the translational motion of the camera. These can be represented in terms of a pointwise "relative structure" measure and the coordinates of the camera epipoles with respect to the panoramic view. The relative structure is once again a property of the scene, which is common to all frames, and therefore represented only once in the same coordinate system as the mosaic. This is reviewed in Section IV-B.2.

(c) In the multiple layer case, the geometric transformation information for each layer consists of the following: (i) the parametric transformations associated with the dominant plane corresponding to that layer, (ii) a layer "ownership" map (typically a binary image) that indicates which points "belong" to that layer, and (iii) the 3D relative structure of the points relative to the plane. The camera translation is common to all the layers, and can be represented in a number of different ways. Since the number of layers is usually small, it is usually convenient to repeat it for each layer.

3. The **dynamic information**, e.g., *moving objects and their trajectories*, which are not captured by the static panoramic view representations. Typically the moving objects are small relative to the background and can be represented as templates along with their motion

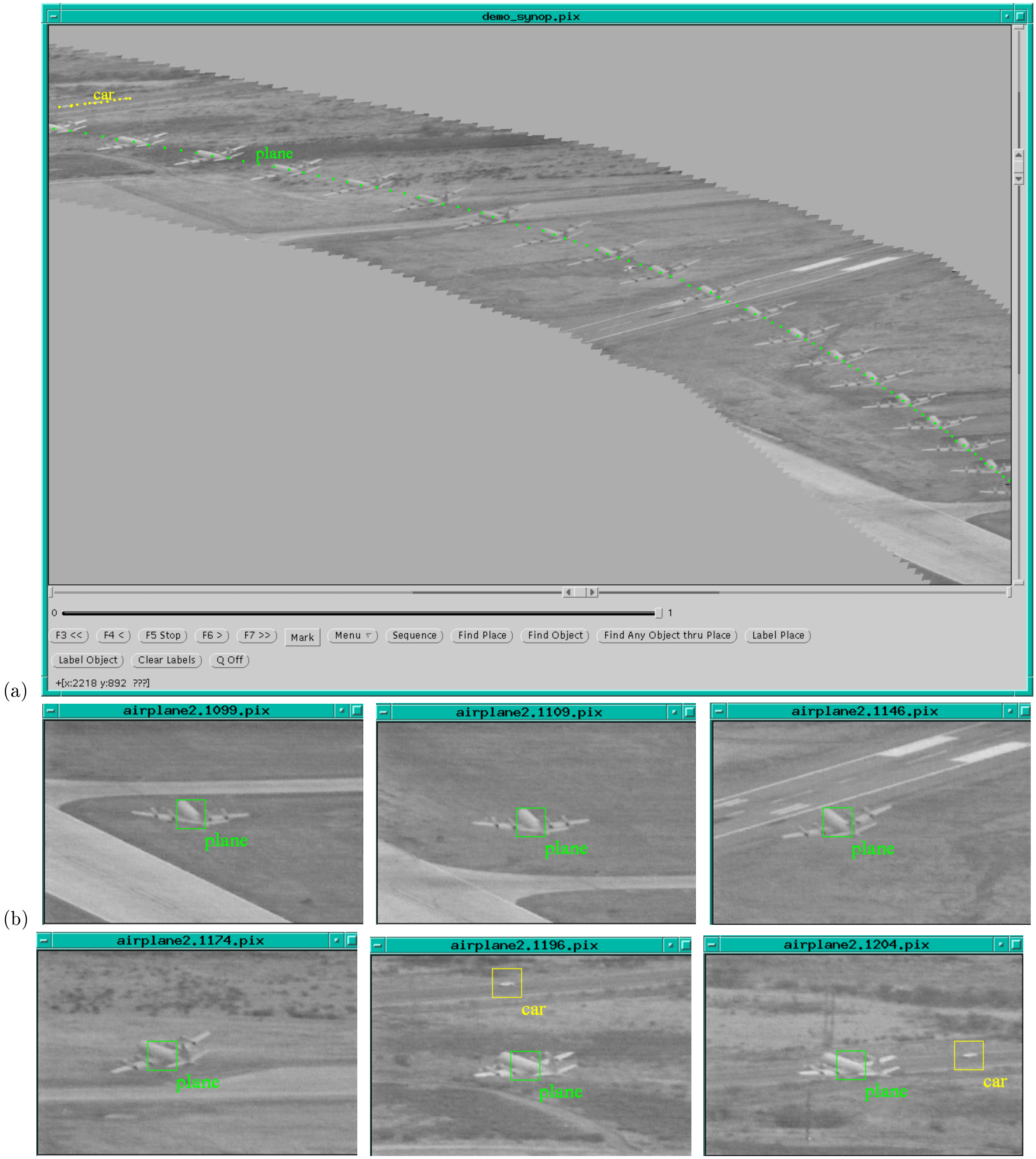

Fig. 9. Annotation of the flying plane video clip.
(a) The annotations are defined *once* on the synopsis mosaic image. The moving objects are being annotated ("plane" and "car").
(b) A few representative frames from the video clip with the annotations inherited from the mosaic image. The original video frames are displayed in Figure 4. The annotations are incorporated into the video frames *automatically* and *instantly* through the geometric coordinate transformations that map each frame onto the mosaic image.

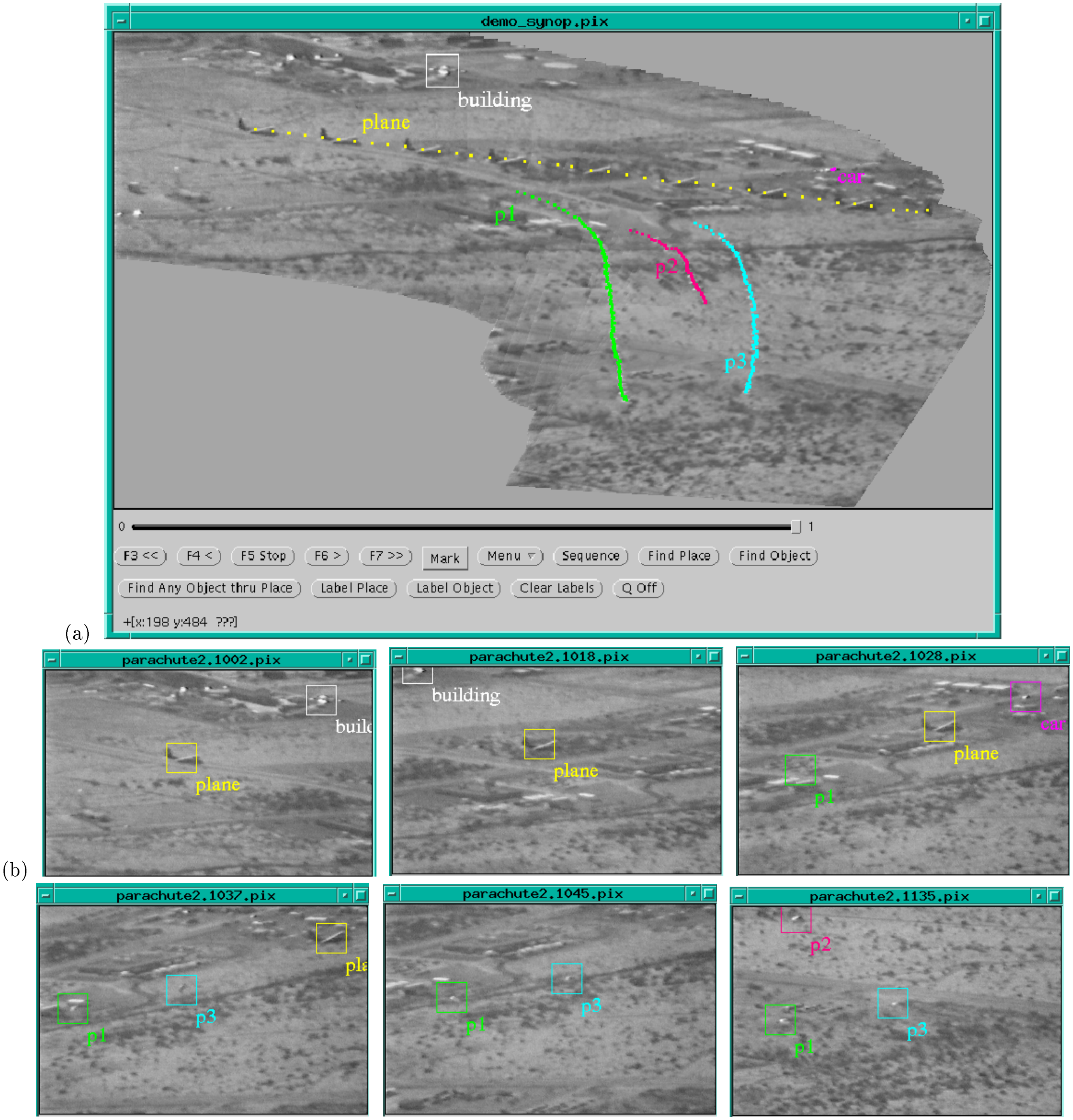

Fig. 10. Annotation of the parachuters video clip.
(a) The annotations are defined *once* on the synopsis mosaic image. Both static scene points ("building") and dynamic scene points ("plane", "car", "p1", "p2", "p3") are being annotated. (b) A few representative frames from the video clip with the annotations inherited from the mosaic image. The original video frames are displayed in Figure 5. The annotations are incorporated into the video frames *automatically* and *instantly* through the geometric coordinate transformations that map each frame onto the mosaic image. The parachuters become visible one-by-one, as their parachutes open: first the left parachuter (green), then the right one (blue), and last the middle one (red).

trajectories[1].

The three levels of the representation described above can capture the vast majority of situations effectively and efficiently. However, there are situations for which our current representation may not suffice, i.e., it will not produce compact or visually meaningful representation. Such situations arise when a a camera is moving around an object (or equivalently an object is rotating in front of the camera), or when the scene contains significant 3D clutter, with many objects at many different depths. These situations require further study and treatment. An example of the type of representations that may be useful in the future to handle such scenarios is the "manifold mosaic" method described in [25], [27].

B. Estimating the Geometric Coordinate Transformations

To relate each frame to a common representation, we need to determine the geometric coordinate transformations between the video frames. This is based on analyzing and interpreting the image motion between the video frames.

Existing methods for interpreting image motion can broadly be classified into two groups: (i) 3D techniques [2], [21], [34], [35], which try to model and interpret the camera-induced motion in terms of the 3D components (namely, 3D camera motion components R and T and the 3D scene structure $Z(x, y)$), and (ii) 2D techniques [14], [5], [6], [8], [30], [24], [36], [3], which do *not* try to decompose the image motion into its 3D components, but instead model the camera induced motion as a single global 2D *parametric* transformation (e.g., 2D affine, 2D quadratic, or 2D projective).

2D techniques have been proven to be very robust, even in the presence of independently moving objects in the scene [14]. As explained earlier (see Section IV-A), these are, however, good models for the camera induced motion only in a restricted set of scenarios ("2D scenes").

3D techniques, on the other hand, can handle general "3D scenes", but their estimation is more difficult [33]. They require *dense* 3D information in the scene (i.e., lots of depth variations), the frames need to be taken with a large baseline (i.e., large camera translation), and are less robust in presence of moving objects. More importantly, if applied to the 2D scenarios, they fail, since these become singular cases in the 3D analysis.

Our hierarchy of mosaic representations matches scenarios that gradually increase in their complexity from 2D to 3D. The same approach of progressive complexity analysis applies also to our estimation process. Our analysis of a video clip always starts with 2D analysis. We first estimate the *dominant* 2D geometric transformation between frames (see Section IV-B.1). Such alignment completely compensates for the camera induced motion in 2D scenes. In 3D scenes, it locks and compensates for the image motion of a dominant planar surface in the scene. The residual parallax motion of the points that are not on the dominant plane is then estimated via a 3D plane+parallax estimation process (see Section IV-B.2). Thus our overall estimation approach consists of two major steps: (i) the estimation of 2D parametric transformations, and (ii) the estimation of residual planar parallax displacements. When the scene is composed of several layers at a few distinct depths, multiple 2D models with residual 3D parallax may be required. The layered alignment is achieved via recursive 2D alignment [14].

B.1 The estimation of 2D parametric transformation

The instantaneous image motion of a general 3D scene can be expressed as [22], [1]:

$$\begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \begin{bmatrix} -(\frac{T_X}{Z}+\Omega_Y)+x\frac{T_Z}{Z}+y\Omega_Z-x^2\Omega_Y+xy\Omega_X \\ -(\frac{T_Y}{Z}-\Omega_X)-x\Omega_Z+y\frac{T_Z}{Z}-xy\Omega_Y+y^2\Omega_X \end{bmatrix} \quad (1)$$

where $(u(x,y), v(x,y))$ denotes the image velocity at image location (x, y), $T = (T_X, T_Y, T_Z)^t$ denotes the translational motion of the camera, $R = (\Omega_X, \Omega_Y, \Omega_Z)^t$ denotes the camera rotation, and Z denotes the depth of the scene point corresponding to (x, y).

Although, strictly speaking, the above equations represent instantaneous image velocity fields, they are very good approximations of interframe displacements even in discretely time sampled images, provided the following requirements concerning the camera motion and the 3D scene are satisfied: (i) the field-of-view of the camera is small (e.g., less than 30 degrees), (ii) the rotational motion between the frames is small (within a few degrees), and (iii) the translational motion component along the optical axis (T_Z) is small relative to Z. Note that these conditions are often satisfied in real video sequences sampled at 15 or 30 frames/sec.

The instantaneous image motion (Equation 1) can often be approximated by a single 2D parametric transformation of the form,

$$\begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \begin{bmatrix} a + b \cdot x + c \cdot y + g \cdot x^2 + h \cdot xy \\ d + e \cdot x + f \cdot y + g \cdot xy + h \cdot y^2 \end{bmatrix} \quad (2)$$

This approximation is valid under the following conditions associated with the scene geometry and/or camera motion: (i) *A planar scene* ($Z(X, Y) = A + B \cdot X + C \cdot Y$)*:* in this case, the parameters (a, b, c, d, e, f, g, h) are functions of the camera motion and the planar surface parameters (A, B, C) (see [11]), (ii) *Distant Scene:* i.e., when the scene is very distant from the camera (i.e., $Z \to \infty$), or when the deviations from a planar surface are small relative to the overall distance of the scene from the camera ($\Delta Z \ll Z$), (iii) *Camera Rotation*–i.e., when the camera undergoes a pure rotational motion (i.e., $T = 0$) or when the camera translation is negligible ($|T| \ll Z$); the rotation will not have any effect on the parameters b and f, and (iv) *Camera Zoom* – when the camera zooms in or out, the image undergoes a dilation. The resulting image motion field can be still be modeled by Equation 2; the zoom will influence the parameters b and f.

[1] In some cases, the objects may be large and each frame may only view a portion of the object. In these cases, the object can be represented as a layer and a panoramic view may be created for these objects as for the background.

We refer to scenes that satisfy any combination of the abovementioned conditions (and hence Equation (2) is applicable), as *2D scenes.*

Under these conditions, we can use a previously developed method [4], [14] in order to compute the $2D$ parametric motion. This technique "locks" onto a "dominant" parametric motion between an image pair, even in the presence of independently moving objects. It does not require prior knowledge of their regions of support in the image plane (see [14]). This computation provides only the $2D$ motion parameters of the camera-induced motion, but no explicit $3D$ shape or motion information. To make this paper self-contained, we briefly outline the technique below.

We will refer to the two image frames (whose image motion is being estimated) by the names "inspection" image and "reference" image, respectively. A Laplacian pyramid is first constructed 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 unknown 2D image motion parameters.

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

$$E(\vec{\alpha}) = \sum_{\mathbf{x}} \left(I(x,y,t) - I(x-u(x,y;\vec{\alpha}), y-v(x,y;\vec{\alpha}), t-1)\right)^2 \tag{3}$$

where I the (Laplacian pyramid) image intensity, $\vec{\alpha} = (a,b,c,d,e,f,g,h)$ denotes the parameters of the quadratic transformation (Equation 2), $(u(x,y;\vec{\alpha}), v(x,y;\vec{\alpha}))$ denotes the image velocity at the location (x,y) induced by the quadratic transformation with parameters $\vec{\alpha}$. The sum is computed over all the points within the region, often the entire image.

The objective function E given in Equation (3) is minimized w.r.t. the unknown parameters (a,b,c,d,e,f,g,h) via the Gauss-Newton optimization technique. Let $\vec{\alpha}_i = (a_i, b_i, c_i, d_i, e_i, f_i, g_i, h_i)$ denote the current estimate of the quadratic parameters. After warping the inspection image (towards the reference image) by applying the quadratic transformation based on these parameters, an incremental estimate $\vec{\delta\alpha} = (\delta a, \delta b, \delta c, \delta d, \delta e, \delta f, \delta g, \delta h)$ can be determined. After iterating a few times within a pyramid level, the process continues at the next finer level. We refer to this process as the *iterative warp estimation* process.

With the above technique, the reference and inspection images are registered so that the desired image region is aligned, and the quadratic transformation (2) is estimated. The above estimation technique is a least-squares based approach and hence possibly sensitive to outliers. However, as reported in [5] 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.

A robust version of the above method [14] handles scenes with multiple moving objects. It incorporates a gradual refinement of the complexity of the motion model (ranging from pure translation at low resolution levels, to a 2D affine model at intermediate levels, to the 2D quadratic model at the highest resolution level). Outlier rejection is performed before each refinement step within the multiscale analysis. This robust analysis further enhances the locking property of the abovementioned algorithm onto a single *dominant* motion.

B.2 Residual 3D Parallax Motion Estimation

The key observation that enables us to extend the 2D parametric registration approach to general 3D scenes is the following: the plane registration process (using the dominant 2D parametric transformation) removes all effects of camera rotation, zoom, and calibration, *without explicitly computing them* [15], [18], [28], [29]. The residual image motion after the plane registration is due only to the *translational* motion of the camera and to the *deviations* of the scene structure from the planar surface. Hence, the residual motion is an *epipolar flow field.* This observation has led to the so-called "plane+parallax" approach to 3D scene analysis [17], [15], [18], [28], [29].

It can be shown (see [19], [15], [28], [29]) that the displacement $\vec{u}$ of a pixel can be decomposed as follows:

$$\vec{u} = \vec{u_\pi} + \vec{\mu}, \tag{4}$$

where $\vec{u_\pi}$ denotes the *planar* part of the $2D$ image motion (which aligns a reference plane Π in the scene). As noted earlier, $\vec{u}_\pi$ can be described by a quadratic transformation as in Equation 2. $\vec{\mu}$ denotes the residual *planar parallax* displacement[2]:

$$\vec{\mu} = \gamma \frac{T_z}{d'_\pi}(\vec{e} - \vec{p_w}) \tag{5}$$

where $\vec{p_w}$ denotes the image point (in homogeneous coordinates) in the first frame which results from warping the corresponding point $\vec{p'}$ in the second image, by the 2D parametric transformation of the reference plane Π. We will refer to the first frame as the *reference frame.* Also, d'_π is the perpendicular distance from the second camera center to the reference plane Π, and $\vec{e}$ denotes the epipole (or FOE), which is the point of intersection of the translational motion vector with the reference image plane. γ is a measure of the 3D shape of the point $\vec{P}$. In particular, $\gamma = \frac{H}{Z}$, where H is the perpendicular distance from the $\vec{P}$ to the reference plane Π, and Z is the "range" (or "depth") of the point $\vec{P}$ with respect to the first camera. We refer to γ as the relative 3D structure of point $\vec{P}$, as it provides 3D structure relative to the plane Π.

Equation 5 indicates that at each image point, the residual planar parallax displacement is a function of the 3D relative structure γ of the point, and the camera translation (as denoted by the epipole $\vec{e}$). For points belonging to the static background scene, the relative structure γ is

[2] When $T_z = 0$, the parallax motion $\vec{\mu}$ has a slightly different form: $\vec{\mu} = \frac{\gamma}{d'_\pi}\vec{t}$, where $t = (T_X, T_Y)$.

constant over the entire sequence, hence common to all the frames, whereas the epipole $\vec{e}$, and the scale factor $\frac{T_z}{d'_\pi}$ is unique to each frame (but common to all the points in the frame). Hence, the geometric transformation due to the 3D parallax motion for the entire sequence relative to the dominant plane, can be represented by two components: (i) a map $\gamma(x,y)$ of the relative structure, which is a "structure" mosaic (aligned with the panoramic mosaic image) that represents the extended geometric information, and (ii) for each frame, the epipole $\vec{e}$ and the scale factor $\frac{T_z}{d'_\pi}$.

The estimation of the camera translation (namely the epipole) by analyzing the residual parallax motion is described in [15], and the estimation of the 3D projective structure γ together with the epipole is described in [18]. The estimation technique is similar to the 2D parametric estimation technique in that, (i) a multi-resolution coarse-to-fine estimation strategy is used, (ii) at each pyramid level, an SSD measure is used as a minimization criterion (however in this case, the measure is a function of the unknown $\gamma(x,y)$ map and the epipole vector $\vec{e}$, as opposed to the parameter vector α) in Equation 3, and (iii) the *iterative warp-refine estimation* strategy is used for obtaining the solution. At each step of the iterative process, the epipole vector $\vec{e}$, and the projective structure map $\gamma(x,y)$ are refined via the Gauss-Newton minimization technique.

C. Moving Object Detection and Tracking

The geometric coordinate transformations that relate the frames to the mosaic image (and to each other) describe the *dominant* detected motion. The dominant motion is assumed to be that of the static portions of the scene (i.e., only due to camera motion). This is a strong assumption which requires treatment in future work. However, this is a valid assumption in a wide range of scenario scenarios, when the camera is not zoomed in on a moving object. This is especially true in airborne video or remote surveillance type of applications.

After dominant-motion alignment, all static portions of the scene are in full alignment, and the only remaining misaligned portions of the image are those that move due to *independent* motion. This is used for detecting potential moving objects [14]. To verify the hypothesis and distinguish moving objects from noise, these image regions are tracked over time. The tracking is performed at a symbolic level, based on "blobs" that represent the misaligned regions. No template correlation or flow estimation is used. This has the benefit that it can effectively track even very small moving objects (e.g., objects that may be a few pixels in size), textureless objects, and non-rigidly moving objects. The objects are required to be detected and tracked over a minimum time period – typically a few (say 6) consecutive frames – before they are believed to be moving objects.

Note also that estimating the trajectories of moving objects in the common mosaic coordinate system allows more reliable detection and tracking of moving objects, even when they are very small (such as the three parachuters in Figure 5). This is because a "temporal coherence" constraint can be used during moving object detection and tracking after removal of the background motion. Assuming that object sizes and velocities do not change too rapidly, the detection of moving objects within each frame can be guided by the trajectory of the objects in a few previous frames. This leads to better separation between small moving objects and noise, as well as enables recovery from losing an object for a few frames (e.g., due to occlusion or bad detection). It also allows handling multiple moving objects with intersecting trajectories. The missing portion of each trajectory is smoothly interpolated/extrapolated from the neighboring frames.

V. Conclusion

This paper described a new approach for efficient access, storage, and manipulation of video data. Our approach is based on transforming the video data from a sequential *frame-based* representation, in which the common scene information is *distributed* over many frames, into a single common *scene-based* representation to which each frame can be *directly* related. This representation then allows *direct* and *immediate* access to the *scene* information, such as static locations and dynamically moving objects. It also eliminates the redundancy between the different views of the scene contained in the frames, and results in a highly efficient and compact representation of the video information. Hence, the scene-based representation forms the basis for direct and efficient access and manipulation of the video data.

As part of the scene-based representation, panoramic mosaic images are created, which provide a snapshot view of the information available in the video data. Two types of mosaics are described: a *static* mosaic, which captures the appearance of the static background portions of the scene, and a *synopsis* mosaic, which in addition visually captures the trajectories of moving objects. These mosaics allow the user to rapidly browse through a large collection of video sequence, and can serve as *visual table-of-contents* for a video database.

The paper also described two new types of indexing methods, based on *geometric* and *dynamic* scene information. While the major research effort in the area of *content-based video indexing* is based on appearance information (e.g., texture and color), the two methods described in this paper have been overlooked. These methods are complementary to the appearance based methods, and are substantially simpler to achieve. The existing appearance-based methods themselves can also be used more efficiently within the scene-based representation, when applied directly to the mosaic image (i.e., to the appearance component of our representation), rather than to the individual video frames one-by-one.

The scene-based representation described in this paper is intended to apply to all types of scenarios. However, there are situations for which our current methods for constructing panoramic views may not suffice, i.e., it will not produce compact or visually meaningful representation. Such situations arise when a camera is moving around an object (or equivalently an object is rotating in front of the camera), or when the scene contains significant 3D clutter, with many objects at many different depths. These situations require further study and treatment.

Acknowledgement

The authors would like to thank Steve Hsu for the development of the symbolic tracking of moving objects and the design of the user interface of the Indexing demo system, Nurit Binenbaum for her support in the implementation of the demo system, and Shmuel Peleg for helpful discussions about the paper.

References

[1] G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 7(4):384–401, July 1985.

[2] Y. Aloimonos, editor. *Active Perception.* Erlbaum, 1993.

[3] S. Ayer and H. Sawhney. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and mdl encoding. In *International Conference on Computer Vision*, pages 777–784, Cambridge, MA, June 1995.

[4] 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.

[5] 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*, 14:886–895, September 1992.

[6] P.J. Burt, R. Hingorani, and R.J. Kolczynski. Mechanisms for isolating component patterns in the sequential analysis of multiple motion. In *IEEE Workshop on Visual Motion*, pages 187–193, Princeton, New Jersey, October 1991.

[7] A. Finkelstein C.E. Jacobs and D.H. Salesin. Fast multiresolution image querying. In *SIGGRAPH*, pages 277–286, 1995.

[8] T. Darrell and A. Pentland. Robust estimation of a multi-layered motion representation. In *IEEE Workshop on Visual Motion*, pages 173–178, Princeton, New Jersey, October 1991.

[9] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: The qbic system. *Computer*, 28(9):23–32, 1995.

[10] D.J. Heeger and J.R. Bergen. Pyramid-based texture analysis/synthesis. In *SIGGRAPH*, pages 229–238, 1997.

[11] M. Irani and P. Anandan. A unified approach to moving object detection in 2D and 3D scenes. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, to appear.

[12] M. Irani, P. Anandan, J. Bergen, R. Kumar, , and S. Hsu. Efficient representations of video sequences and their application. *Signal Processing: Image Communication*, 8(4), 1996.

[13] M. Irani, S. Hsu, and P. Anandan. Video compression using mosaic representations. *Signal Processing: Image Communication*, 7(4–6), 1995.

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

[15] M. Irani, B. Rousso, and S. Peleg. Recovery of ego-motion using image stabilization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 454–460, Seattle, WA, June 1994.

[16] Michal Irani, P. Anandan, and S. Hsu. Mosaic based representations of video sequences and their applications. In *International Conference on Computer Vision*, pages 605–611, Cambridge, MA, November 1995.

[17] J.J. Koenderink and A.J. van Doorn. Representation of local geometry in the visual system. *Biol. Cybern.*, 55:367 – 375, 1987.

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

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

[20] 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.

[21] J.M. Lawn and R. Cipolla. Robust egomotion estimation from affine motion parallax. In *European Conference on Computer Vision*, pages 205–210, May 1994.

[22] H.C. Longuet-Higgins. Visual ambiguity of a moving plane. *Proceedings of The Royal Society of London B*, 223:165–175, 1984.

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

[24] F. Meyer and P. Bouthemy. Region-based tracking in image sequences. In *European Conference on Computer Vision*, pages 476–484, Santa Margarita Ligure, May 1992.

[25] S. Peleg and J. Herman. Panoramic mosaics by manifold projection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 338–343, San-Juan, June 1997.

[26] S. Ravela, R. Manmatha, and E.M. Riseman. Image retrieval using scale space matching. In *European Conference on Computer Vision*, 1996.

[27] B. Rousso, S. Peleg, and I. Finci. Generalized panoramic mosaics. In *DARPA IU Workshop*, 1997.

[28] Harpreet Sawhney. 3d geometry from planar parallax. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 1994.

[29] 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.

[30] M. Shizawa and K. Mase. Principle of superposition: A common computational framework for analysis of multiple motion. In *IEEE Workshop on Visual Motion*, pages 164–172, Princeton, New Jersey, October 1991.

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

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

[33] W.B. Thompson and T.C. Pong. Detecting moving objects. *International Journal of Computer Vision*, 4:29–57, 1990.

[34] P.H.S. Torr and D.W. Murray. Stochastic motion clustering. In *European Conference on Computer Vision*, pages 328–337, May 1994.

[35] P.H.S. Torr, A. Zisserman, and S.J. Maybank. Robust detection of degenerate configurations for the fundamental matrix. In *International Conference on Computer Vision*, pages 1037–1042, Cambridge, MA, June 1995.

[36] J. Wang and E. Adelson. Layered representation for motion analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 361–366, New York, June 1993.

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