LAMP: 3D Layered, Adaptive-resolution and Multi-perspective Panorama - a New Scene Representation

Zhigang Zhu
Department of Computer Science
The City College, The City University of New York, New York, NY 10031

Allen R. Hanson
Department of Computer Science
University of Massachusetts Amherst, Amherst, MA 01003

Short Running Title: 3D LAMP Representation

Contact Information:

Prof. Zhigang Zhu
Department of Computer Science
The City College of New York /CUNY
Convent Avenue and 138th Street, New York, NY 10031
Tel: (212) 650 - 8799 Fax: (212) 650 - 6248
Email: zhu@cs.ccny.cuny.edu
URL: http://www-cs.engr.ccny.cuny.edu/~zhu/

Abstract

A compact visual representation, called the 3D Layered, Adaptive-resolution and Multi-perspective Panorama (LAMP), is proposed for representing large-scale 3D scenes with large variations of depths and obvious occlusions. Two kinds of 3D LAMP representations are proposed: the relief-like LAMP and the image-based LAMP. Both types of LAMPs concisely represent almost all the information from a long image sequence. Methods to construct LAMP representations from video sequences with dominant translation are provided. The relief-like LAMP is basically a single extended multi-perspective panoramic view image. Each pixel has a pair of texture and depth values, but each pixel may also have multiple pairs of texture-depth values to represent occlusion in layers, in addition to adaptive resolution changing with depth. The image-based LAMP, on the other hand, consists of a set of multi-perspective layers, each of which has a pair of 2D texture and depth maps, but with adaptive time-sampling scales depending on depths of scene points. Several examples of 3D LAMP construction for real image sequences are given. The 3D LAMP is a concise and powerful representation for image-based rendering.

Keywords: image-based modeling and rendering, layered representation, multi-resolution, multi-image processing, spatio-temporal image, epipolar plane image

1. INTRODUCTION

The problem of modeling and rendering real 3D scenes has received increasing attention in recent years in both the computer vision and computer graphics communities [1-3, 5, 6, 8, 17-19, 22, 23, 27]. In order to build a visual representation from image sequences for re-rendering real 3D natural scenes, there are two challenging issues that need to be solved: the correspondence problem between two (or multiple) views, and a suitable geometric representation of large scale scenes. Usually a suitable visual representation will ease the correspondence problem. Many of the successful image-based modeling and rendering approaches [1-3, 5, 6, 8-10] have tried to simplify or avoid the correspondence problem by using 2D image interpolation or video registration/mosaicing. On the other hand, more general approaches need sophisticated vision algorithms, such as in multi-view stereo [17, 18] or general motion analysis [10, 11, 22, 23] of an image sequence. Three classes of image-based representations have been proposed to represent video sequences of large-scale 3D scenes: panoramic (mosaic) representations, multi-view representation and layered representations. Building and maintaining a suitable visual representation of a large-scale scene remains an important research topic in image-based modeling and rendering.

1.1. Related Work

Mosaic-based approach - A video mosaic is a composite of images created by registering overlapping frames. Many of the current successful image mosaic algorithms, however, only generate 2D mosaics from a camera rotating around its nodal point [1,2,3]. Creating multi-perspective stereo panoramas from one rotating camera off its nodal point was proposed by Ishiguro, et al [4], Peleg, et al [5], and Shum & Szeliski [6]. A system for creating a global view for visual navigation by pasting together columns from images taken by a smoothly translating camera (comprising only a vertical slit) has been proposed by Zheng & Tsuji [7]. The moving slit paradigm was later used as the basis of the 2D manifold projection approach for image mosaicing [8], the multiple-center-of-projection approach for image-based rendering [9], the epipolar plane analysis techniques for 3D reconstruction [15], and the parallel-perspective view interpolation methods for stereo mosaicing [16, 27]. Multi-perspective panoramas (or mosaics) exhibit very attractive properties for visual representation and epipolar geometry; however, 3D

recovery from stereo mosaics still faces the same problems as traditional stereo methods - the correspondence problem and occlusion handling. The 3D information for regions that are occluded in one of the views is usually difficult to obtain since no correspondences can be established for those regions.

Layered representation - In a layered representation, a set of depth surfaces is first estimated from an image sequence from a single camera, and then combined to generate a new view. Wang and Adelson [10] addressed the problem as the computation of 2D affine motion models and a set of affine support layers from an image sequence. The *layered representation* that they proposed consists of three maps in each layer: a mosaiced intensity map, an alpha map, and a velocity map. Occlusion boundaries are represented as discontinuities in a layer's alpha map (opacity). This representation is a good choice for image compression of a video sequence and for limited image synthesis of selected layers. Recently, Ke & Kanade [22] proposed a subspace approach to reliably extract *planar* layers from images by taking advantage of the fact that homographies induced by *planar patches* in the scene form a low dimensional linear space. Occlusion regions are excluded from their layered representation. Sawhney and Ayer [11] proposed a multiple motion estimation method based on a Minimum Description Length (MDL) principle. However, their algorithms are computationally expensive and require a combinatorial search to determine the correct number of layers and the "projective depth" of each point in a layer. Occlusion regions are not recovered in their layered model. Baker, Szeliski & Anandan [12] proposed a framework for extracting structure from stereo pairs and represented a scene as a collection of approximately planar layers. Each layer consists of an explicit 3D plane equation, a texture map (a *sprite*), and a map with depth offsets relative to the plane. The initial estimates of the layers are recovered using techniques from parametric motion estimation and then refined using a re-synthesis algorithm which takes into account both occlusion and mixed pixels. More recent work on 3D layer extraction [23] uses an integrated Bayesian approach to automatically determine the number of layers and the assignment of individual pixels to layers. For more complex geometry, a *layered depth image* (LDI) has been proposed [13] which is a representation of a scene from a single input camera view, but with multiple pixels along each line of sight.

Multi-view approach - Rather than constructing a single mosaic from a sequence of images, multi-view approaches represent a scene by multiple images with depth and texture. Chang and Zakhor [17] proposed a method to obtain depth information of some pre-specified "reference frames" of an image sequence captured by an uncalibrated camera scanning a stationary scene, then to transform the points of reference frames onto an image of the desired virtual viewpoint. However, reference frames were chosen quite heuristically; a synthesized image from a viewpoint far away from that of the reference frames leads to erroneous results since occluded or uncovered regions cannot be well represented. Chang and Zakhor later extended this work to a multivalued representation (MVR) [26], which is automatically constructed with respect to a single reference frame from a set of dense depth maps. They pointed out that occlusion levels come naturally from the dense depth information and argued that because of visibility limitations, real-world scenes typically do not have more than three occlusion levels. This distinguishes their MVR from the layered representations, which usually have a much larger number of layers when using affine motion to group regions. Szeliski [18] presented a new approach to computing dense depth and motion estimates from multiple images. Rather than estimating a single depth or motion map, a depth or motion map is associated with each input image (or some subset of them). Furthermore, a motion compatibility constraint is used to ensure consistency between these estimates, and occlusion relationships are maintained by computing pixel visibility.

1.2. 3D LAMP: Overview of Our Approach

The goal of our work is to construct a *layered* and *panoramic* representation of a large-scale 3D scene with large variations of depths and obvious occlusions from primarily translating video sequences. Our approach in this paper is based on a multi-perspective panoramic view image (PVI) [7] and a set of epipolar plane images (EPIs) [14] extracted from a long image sequence, together with a panoramic depth map generated by analyzing the EPIs [15, 21]. To accomplish this, we need to solve four problems: 1) generating seamless PVIs and EPIs from video under more general motion than pure translation in a large and real scene, 2) analyzing the huge amount of data in EPIs robustly and efficiently to obtain dense depth information, 3) enhancing resolution and recovering occlusions in a parallel-perspective PVI representation, and 4) representing a large scale 3D scene with different depths and occlusions efficiently and

compactly. While the first two issues are very important in constructing a 3D model of a scene, they have been discussed in our previous work [15, 21], so this paper will mainly focus on the last two issues.

We propose a new compact representation - *3D Layered, Adaptive-resolution and Multi-perspective Panorama* (LAMP). The motivation for layering is to represent occluded regions and the different spatial resolutions of objects with different depth ranges; meanwhile the model is represented in the form of a seamless multi-perspective mosaic (panorama) with viewpoints spanning a large distance. Two kinds of 3D LAMP representations are constructed: the *relief-like LAMP* and the *2D image-based LAMP*. The relief-like LAMP is basically a single, extended, multi-perspective PVI with both texture and depth values, but each pixel has multiple values of texture-depth pairs to represent results of occlusion recovery and resolution enhancement. The image-based LAMP, on the other hand, consists of a set of multi-perspective layers, each of which has both texture and depth maps with adaptive time-sampling scales depending on depths of scene points. The LAMP representation is related to such representations as the panoramic view image (PVI) [7, 15], sprite [12], and the layered depth image (LDI) [13]. However, it is more than a multi-perspective PVI in that depth, adaptive-resolution and occlusion are added in our representation. It is different from the sprite (or the layered depth image) since the latter is a view of a scene from a single input camera view and is without adaptive image resolution. The 3D LAMP representation is capable of synthesizing images of new views within a reasonably restricted but arbitrary moving space, as its intensity and depth maps contain almost all the information that could be obtained from an image sequence.

1.3. Organization of the Paper

This paper is organized as follows. In section 2, we describe the basic panoramic representation with parallel-perspective geometry, including both texture and depth maps. Based on this "supporting" panoramic representation, we propose a relief-like LAMP representation in Section 3, which features multiple layers, adaptive resolution, multi-perspective, and panoramic views. In Section 4, an image-based LAMP representation is presented with just a set of 2D images in order to simplify the 3D scene representation. Steps to convert a relief-like LAMP to an image-based LAMP are provided. Section 5 discusses a unique EPI-based approach to construct a relief-like LAMP. This approach does not explicitly extract features, track loci, or match

correspondences. Instead, a spatio-temporal-frequency domain cross-analysis is performed to create dense depth maps, to localize accurate depth boundaries, and to extract occluded regions that cannot be seen in the selected panoramic view. In Section 6, experimental results and some practical considerations are given for 3D LAMP construction, and 3D rendering based on LAMP representations are discussed. In Section 7, we make two comparisons. First, we compare our LAMP representation based on parallel-perspective projection with those representations having full perspective or full orthogonal projections. Second, we compare our image-based LAMP representation with several existing layered representations. The last section is a summary of this work.

2. BASIC PANORAMIC GEOMETRY

For completeness, we give a brief introduction to the constructions and representations of Panoramic View Images (PVIs) and Epipolar Plane Images (EPIs) and to the reconstruction of a "supporting PVI surface" - a panoramic map with both depth and texture - from PVIs and EPIs.

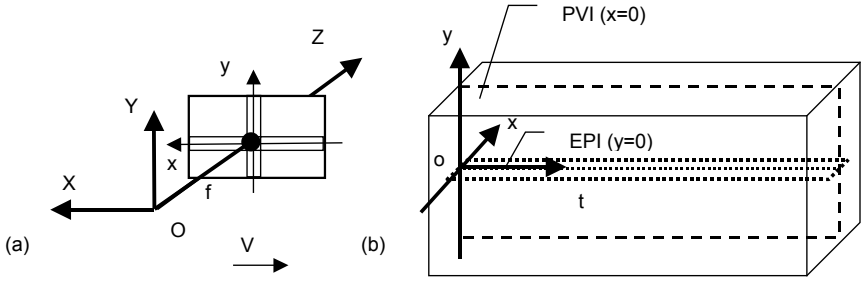


Fig. 1. ST image model. A PVI is a yt image inside the xyt cube and an EPI is an xt image

Let us consider the situation in which a model of a large-scale scene will be constructed from a long and dense video sequence captured by a camera moving in a straight line whose line of sight (i.e., optical axis) is perpendicular to the motion (Fig. 1). The resulting sequence obeys the following spatio-temporal (ST) perspective projection model

$$x(t) = f\frac{X + Vt}{Z}, y(t) = f\frac{Y}{Z} \quad (1)$$

where (X,Y,Z) represents the 3D coordinate of a point at time t=0, f is the camera focal length, and V is its speed . A feature point (x,y) forms a straight locus and its depth value is

$$D = Z = f\frac{V}{v} = f\frac{Vdt}{dx} \tag{2}$$

where

$$v = dx / dt \tag{3}$$

is the slope of the straight locus. In other words, two kinds of useful 2D ST images can be extracted: (1) Panoramic View Images (PVIs) - the *y*-*t* intersections in the *xyt* cube (Fig. 1), which is a parallel-perspective image including most of the 2D information of the scene from multiple viewpoints, and (2) Epipolar Plane Images (EPIs) - the *x*-*t* intersections in the *xyt* cube, whose ST locus orientations represent depths of scene points. Fig. 1 illustrates the central PVI and one of the EPIs.

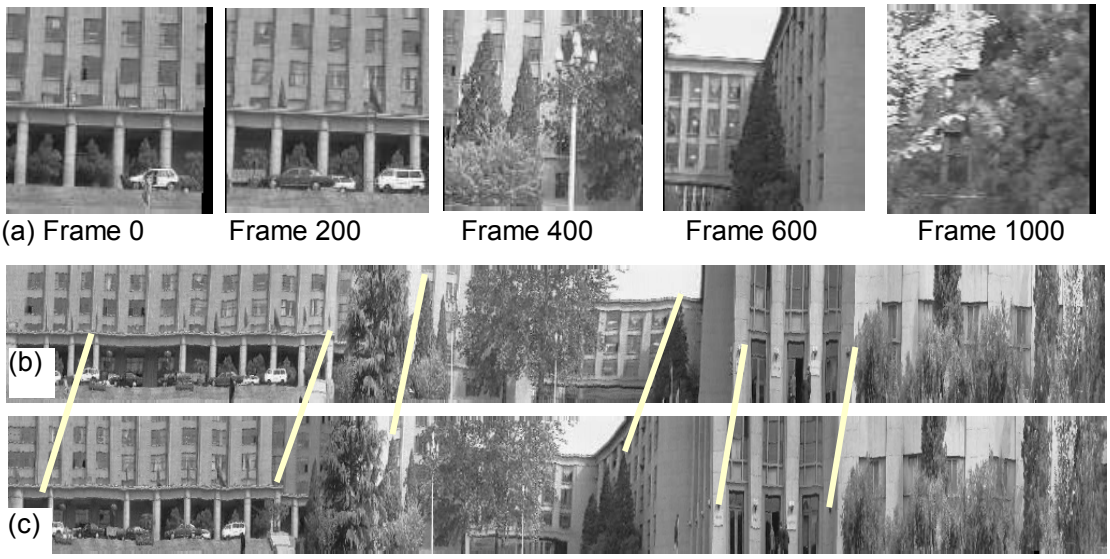


Fig. 2. A few frames of the stabilized Main Building (MB) sequence in (a), and a pair of stereo PVIs: (b) x = 0 and (c) x= -56. White lines indicate matches.

In real scenes, the motion of a camera usually will be composed of a dominant translation plus unpredictable variations. In addition to depth recovery and occlusion handling, a pre-processing step is needed to generate a stabilized image sequence with only a 1D translation by using image stabilization and image mosaicing techniques [15, 21]. Fig. 2 shows two PVIs that are extracted from *x*=0 and *x*=-56 of a 128*128*1024 *xyt* image cube of a building scene - the *stabilized* Main Building (MB) sequence. The MB sequence was captured by a camcorder mounted on a tripod carried by a moving vehicle. Other than the small vibration of the vehicle on a normal outdoor

road, the translational velocity was mostly constant. Fig. 2a shows a few frames after video stabilization; image transformations for removing the camera vibration produce blank regions (shown in black) around the borders of the stabilized frames, particularly visible on the right sides of Frame 0 and Frame 1000. The camera vibration is also demonstrated by the irregular boundaries of the mosaic in Fig. 7.

Multi-perspective PVIs provide a compact representation for a large-scale scene, and stereo PVIs can be used to estimate the depth information of the scene [7, 6, 16, 27]. In panoramic stereo (Eq. (2), Fig. 2), the "disparity" *dx* is fixed; and the distance *D* is proportional to *dt,* the stereoscopic "displacement" in *t*. This indicates that depth resolutions are the same for different depths. However, we still face two problems in order to use stereo PVIs to recover 3D information - the correspondence problem and the occlusion problem.

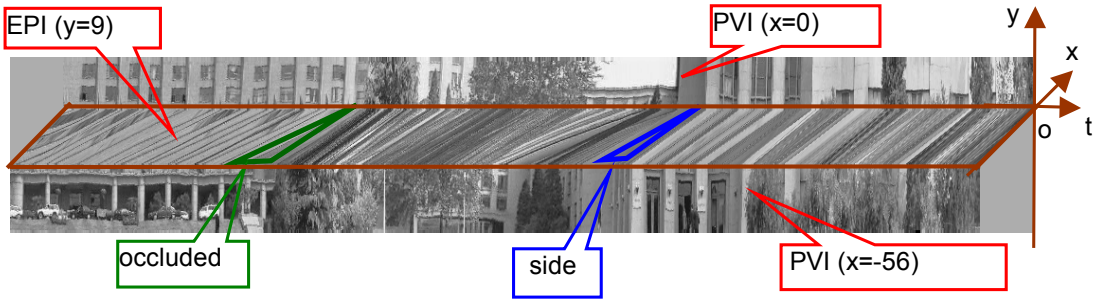
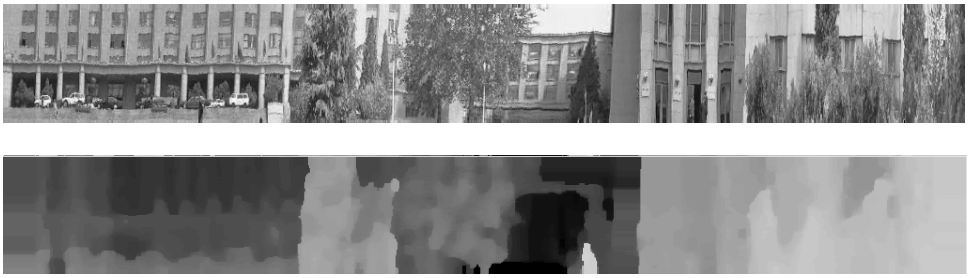


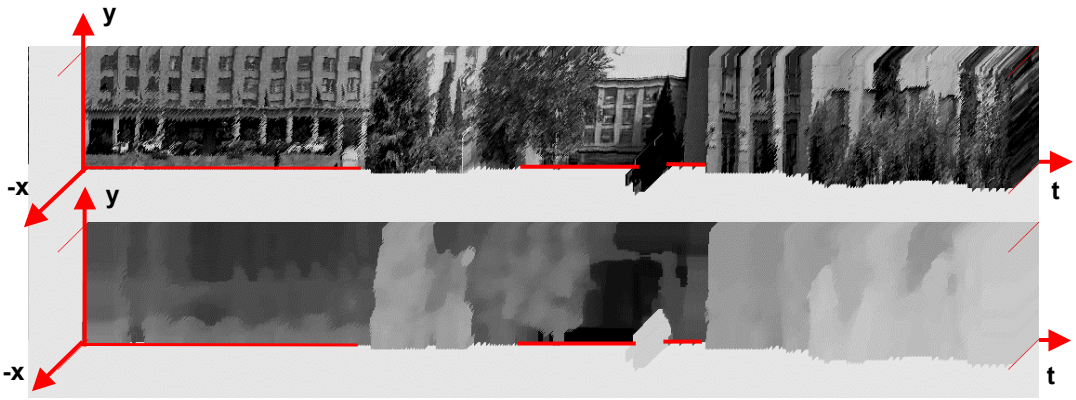
Fig. 3. Each EPI is a composite image of a scanline from all possible views of an image sequence, and includes information about depth, occlusion and spatial resolution of 3D scene points

Our solution to these two problems is to effectively use the information that is continuous between two views, i.e., the epipolar plane images, to obtain a "supporting PVI surface" - a panoramic view image with both texture and depth maps, and then the occluded regions that cannot be seen from the PVI. Fig. 3 shows an EPI from which the occluded (and side) regions as well as depths can be recovered; the algorithms to extract depth and to recover occlusion/resolution will be discussed in Section 5. The *supporting PVI surface* is a 2D panoramic view image with both texture and depth map (Fig. 4a) and is the base for our LAMP representation. The texture map is a *y-t* intersection plane in the *xyt* image cube, and the depth map has depth values for each pixel. In principle, we can extract the supporting PVI surface from

any x location inside the xyt cube. In this paper, we usually select a central PVI (with $x = 0$) that has orthogonal parallel rays along the direction of motion. For constructing a dense depth map for the supporting PVI, we need to process all the EPIs inside the xyt cube. While the 1D motion model will be used to develop our EPI-based approach in this paper, LAMP representations can be constructed under a more general [15, 16, 27] or different motion [5,6], and/or using other approaches [10, 11, 12, 22, 23].



(a) Texture and depth maps of the supporting PVI surface (back of the relief).



(b) Texture and depth maps of the relief surface (front of the relief).

Fig. 4. The base layer of the relief-like LAMP representation for the MB sequence.

3. Relief-like 3D LAMP Representation

Based on the representation of a multi-perspective supporting PVI surface (containing both texture map and depth map), we propose a compact and comprehensive representation called the **3D LAMP**- *Layered, Adaptive-resolution and Multi-perspective Panorama*. We will explain the LAMP model by an illustrative example shown in Fig. 5 using a simple 1D scene. Recall that a 3D ST image (which is a 2D EPI for the 1D scene in Fig. 5) includes everything from an image

sequence. First, we will give a basic LAMP representation - the *relief-like 3D LAMP* - that is directly carved from the 3D ST image (xyt image). The relief-like 3D LAMP basically cuts out some essential part of the 3D ST image depending on the depth value for each pixel. It has the following four properties that make it very suitable for image-based modeling and rendering:

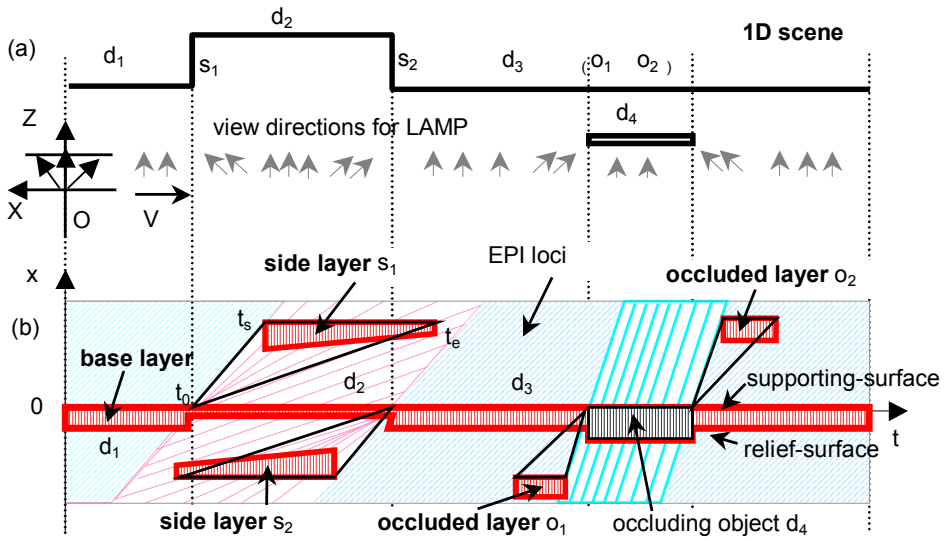


Fig. 5. A 1D illustrative scene and its LAMP representation. (a) 1D scene with four horizontal depths (d_1-d_4) and two sides (s_1,s_2), (b) relief-like LAMP and depth layering for the image-based LAMP from an EPI. Note that in order to represent all the information presented in an image sequence of the scene, different parallel viewing directions in (a) are needed, which are reflected in the LAMP representation as multiple layers in (b).

1). It is a *panoramic* image-based representation with 3D information. A large-scale scene will first be represented by a *base layer*, which is indexed by a seamless 2D panoramic view image (PVI) consisting of both texture and depth maps. This 2D PVI (see Fig. 4a for a real example) is defined as the *supporting surface* (back surface) of the *base layer* in the relief-like LAMP, which makes the LAMP representation very efficient for archiving (modeling) and retrieval (rendering).

2). It has *adaptive* image resolution. In the base layer (which includes depths d_1, d_2, d_3 and d_4 in Fig. 5b), each point has an attached streak of *multiple* pixels in the *x* direction, with both texture and depth information, in order to represent the resolution loss in the multi-perspective PVI (the supporting surface). The slope of the locus, i.e. the image velocity *v* in Eq. (3), gives us the

number of pixels if it is greater than 1 (v>1). Otherwise the number of pixels is 1. Real examples can be found in the right hand side of the panoramic image in Fig. 4a where the doors of the buildings and the bushes in front of it appear narrower than the real ones. In the LAMP representation, the number of pixels ("temporal resolution") at each point in the supporting surface (where x = 0) *adaptively* changes with the depth of that point. The adaptive temporal sampling in the horizontal direction and the inherent adaptive spatio-sampling of perspective projections in the vertical direction recover and preserve image resolutions of the original frames in a satisfying way.

The name "relief-like" comes from the fact that the appearance of the front surface (*relief surface*) of this representation is somewhat like a *relief sculpture*, in which forms and figures (with image velocities $v > 1$) are distinguished from a surrounding planar surface (*the supporting surface*) (Fig. 5b). Fig. 4b shows the corresponding relief surface of the base layer of the relief-like LAMP representation for the MB sequence (refer to Fig. 14 for all the internal data). Each pixel in the relief-like LAMP is associated with a location with (x,y,t) coordinates; for the supporting surface, we have $x = 0$. The end point in the *relief surface* in the location (y,t) is exactly connected with the start point in the supporting surface in the location ($y, t+1$) (Fig. 5). This feature allows us to generate seamless mosaics in the image-based LAMP representation in the next section.

3). It is a *layered* representation. Additional *occluded layers*, which are smaller pieces of multi-perspective view images compared with the "complete" based layer, represent the occluded and side regions (regions o_1, o_2, s_1 and s_2 in Fig. 5b) that are not visible in the selected parallel panoramic view of the base layer (which is orthogonal to the camera path in Fig. 5). They are attached to the base layer with the same representation (texture, depth and adaptive resolution) as the base layer. Each of the occluded x-segments is attached to a depth boundary point (y,t_0), and has an x coordinate, a start time t_s and an end time t_e that mark its position in the xyt image (Fig. 5b). We want to emphasize that in order to represent all the information presented in an image sequence of the scene, different parallel viewing directions (shown in Fig. 5a) are needed, which are reflected in the LAMP representation (Fig. 5b) as multiple layers with different x coordinates.

4). It is a *multiple perspective* image. The PVI is a multi-viewpoint perspective image (i.e., the *y-t* image). Each sub-image (a one-column slit-image in concept) in the multi-perspective panorama is full perspective, but successive slit images have different viewpoints. The multi-perspective representation acts as a bridge in the image-based modeling and rendering pipeline between the modeling end from the original perspective sequences and the rendering end for new perspective images, both with changing viewpoints over a large distance.

In conclusion, a relief-like LAMP is composed of a "complete" base layer and a set of occluded layer pieces. It can be viewed as an essential part of the *xyt* image cube in which each pixel has two attributes - texture (represented by intensity I) and depth (represented by the image velocity v) - connected to its coordinates (x,y,t), where the x coordinate is implicitly represented. The data structure of a relief-like LAMP is a spatio-temporal (ST) 2D array that is defined as follows:

```
struct  x-Streak
{
      int x0; // start x coordinate, indicating the viewing direction of the supporting surface
      int xlen; // length of the x-streak, indicate thickness of the relief
      int I[xlen]; // texture: intensity array of the x-streak
      float v[xlen]; // depth: image velocity array of the x-streak
}
struct  Occlusion-Segment
{
      char type; // segment type: OCCLUDED or SIDE
      int ts, te;  // start frame and end frame of the segment
      x-Streak r[te-ts+1]; // x-Streaks in this segment
}
struct  Relief-LAMP-Element
{
      x-Streak *r; // a x-streak
      Occlusion-Segment *occ; // a occluding segment; NULL if not a depth boundary
}
Relief-LAMP-Element  rLAMP[YDIM][TDIM]; // rLAMP[y][t]: y=0~ YDIM, t = 0~TDIM
```

It is clear that a relief-like LAMP is a 2D spatio-temporal array indexed by y and t; at each (y,t) location there is an **x-Streak**, which is a 1D segment along the x direction inside the *xyt* cube and an optional **Occlusion_Segment** that consists of a number of x-Streaks. In the current implementation, each **x-Streak** assigns the same image velocity v of the start point (x_0,y,t) in the supporting surface. In other words, for resolution enhancement, we only sample v pixels along the x direction and assign them the same v value. An **Occlusion-Segment** is one of two types - OCCLUDED that has the same depth, or SIDE that has linearly interpolated depths. Clearly, the

representation here allows further refinement of the depth map in that each *(x,y,t)* pixel in the relief-like LAMP can has its own depth. The 3D coordinates of *each* pixel (*x,y,t*) in the relief-like LAMP can be recovered by the following equation

$$Z = F\frac{V}{v},\ Y = y\frac{V}{v},\ X = x\frac{V}{v} - Vt \tag{4}$$

given *v(x,y,t)* in the relief-like LAMP representation.

4. IMAGE-BASED 3D LAMP REPRESENTATION

A relief-like LAMP can be viewed as having adaptive time sampling for every single pixel in the panoramic supporting surface as well as including all the occluded regions represented in the same way. This representation is good for a complex scene that has many small depth layers. However, in terms of data structures, the relief-like LAMP is an inhomogeneous representation rather than a set of homogeneous 2D image arrays. In addition, the base layer usually includes objects at different depth levels, and the occluded layers are not merged into the regions they belong to. For example, regions o_1 and o_2 in Fig. 5 belong to the same depth surface as region d_3, but they are segmented into two additional "occluded" layers. Hence a natural extension of the basic LAMP is to extract from it a more concise representation - an *image-based 3D LAMP*. In an image-based 3D LAMP, a scene is represented as multiple layers of 2D mosaiced images in which each layer is truly represented by two 2D arrays - a texture map and a depth map. We wish to create a panoramic mosaic image for each layer that is both seamless and preserves adequate image resolutions. There are two steps necessary to fulfill this goal: depth layering and time re-sampling.

4.1. Depth Layering

The motivation for layering is to represent occluded regions and different spatial resolutions of objects with different depth ranges in different layers. An image based LAMP is layered according to occluding relations rather than merely depths, which is also used by other researchers as a powerful observation to generate a more compact image-based representation [26]. The scene parts with varying (but not discontinuous) depths in a single layer will further use *adaptive* temporal sampling rates in the second step to represent different resolutions for

different depths along the direction of parallel projection. Conceptually, the implementation of depth layering from a relief-like LAMP is straightforward since we can easily determine where the occluding regions in the relief-like *base layer* should be segmented and put into separate *image-based layers*. The same is true for the occluded or side regions in the *occluded layers*. Generally speaking, layers either in the same depth range or with continuous depths will be merged into one single layer. Regions with occluding boundaries will be divided. For example, in Fig. 5b, the two occluded regions (o_1 and o_2) will be merged into the base layer with depth d_3, while the occluding region (d_4) originally included in the base layer will be separated out as a new layer. Ideally, side regions (s_1 and s_2 in Fig. 5b) should be inserted into the base layer since they can connect well (in both depth and texture) with the base layer. However, in our current implementation we put them into two separate layers for simplicity.

After depth layering, the original relief-like LAMP is divided into several *single-layer* relief-like LAMPs, ready for creating image-based layers. Each of them only includes a base layer, with an *x* coordinate to indicate where it is taken from in the *xyt* cube. Different *x* coordinates in the relief-like LAMP imply different viewing directions of parallel projections in order to better capture surfaces with different orientations (see Fig. 5a).

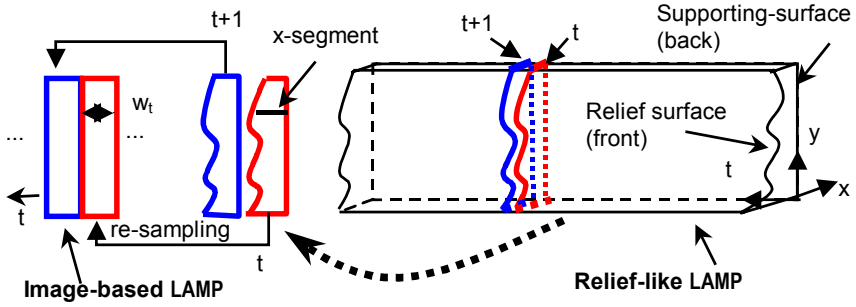


Fig. 6 . Temporal re-sampling and seamless mosaicing

4.2. Temporal Re-Sampling

From each single-layer relief-like LAMP that is obtained by depth layering (as shown in the left part of Fig. 6), we will generate a seamless 2D panoramic image with both texture and depth attributes. In each column (*t*) of such a relief-like LAMP (Fig. 6), the number of pixels in the *x* direction of a point in the supporting surface is inversely proportional to the depth of that point. Since depths may change along the *y* direction, the *x*-*y* slice at each column *t* usually has varying widths in the *x* direction. In order to create a regular 2D image, we want to warp each irregular *x*-

y slice (in the relief-like LAMP) with varying widths in the *x* direction into a rectangular *t-y* slice of equal width w_t ($w_t \geq 1$) in the *t* direction, which will be stitched into a 2D seamless mosaic in the image-based LAMP (see the right part of Fig. 6). The width of the *t-y* slice image at time *t* is calculated as the dominant image velocity of all *v(y, t)* in column *t* of the supporting surface, e.g.

$$w_t = \bar{v}(t) = \underset{y}{median}\{v(y,t)\} \tag{5}$$

Note that each *x*-segment of the *x-y* slice in a relief-like LAMP starts from x_0 and ends at $x_e(y,t) = x_0 - v(y, t)$; for the base layer $x_0=0$. A temporal re-sampling is performed for each *x*-segment by turning it into a *t*-segment of w_t-pixels (Fig. 6 and Eq. (5)). In this way, *super-time sampling* is virtually achieved within a frame's time *t* (usually 1/30 second as the unit) such that each pixel of a transformed w_t-pixel slice represents a point captured in a time unit of $1/w_t$. Hence, the texture map and the depth map of a layer starting from time t_0 are represented as *I(y,k)* and *v(y,k)* where index *k* for time *t* is

$$k \in [\sum_{\tau=t_0}^{t-1} \bar{v}(\tau), \sum_{\tau=t_0}^{t-1} \bar{v}(\tau) + \bar{v}(t)],\ \bar{v}(t_0) = 0,\ t > t_0 \tag{6}$$

For computing 3D coordinates, the super-sampled t_k corresponding to column *k* is stored in a 1D array as part of the image-based LAMP model:

$$t_k = t + \frac{k - \sum_{\tau=t_0}^{t-1} \bar{v}(\tau)}{\bar{v}(t)} \in [t, t+1),\ \bar{v}(t_0) = 0,\ t > t_0 \tag{7}$$

In this manner, each column *k* in the image-based LAMP array is virtually a 1-column perspective image at the "super" viewpoint (time) t_k. The densities of viewpoints adaptively change with depths of the scene. The parallel-perspective views between time *t* and *t+1* are approximated by transforming the corresponding part of the perspective image in frame *t*. Note that this is a *correspondenceless* approach to implementing view interpolation and is different from the global parametric method in [24] or the local match method in [16, 27] for image mosaicing. Fig. 7 shows a seamless adaptive-resolution panoramic mosaic (corresponding to the PVI shown in Fig. 4a), where the time scale in each instant *t* is determined by the dominant

(median) depth of points along the corresponding vertical line (the *y* direction) in frame *t*. Note that the aspect ratios of the objects in the right-hand side of Fig. 4a are restored.

Fig. 7. Multi-viewpoint mosaic with adaptive time scales (the right edge of the upper part connects with the left edge of the bottom part). The width of each vertical slice from the corresponding original frame is determined by the dominant image velocity *v* of pixels along the *y*-axis in the corresponding PVI (the supporting surface of the base layer). Circles in this figure indicate the corresponding OCCLUDED and SIDE regions that are shown on the EPIs in Fig. 12.

In conclusion, in an image-based 3D LAMP each layer is basically a 2D parallel-perspective panorama with an *x* coordinate (to indicate viewing direction). It has three components: 1) texture map *I(y,k)*; 2) depth map *v(y,k)*; and 3) a 1D super time-sampling array $t_k=t(k)$ (to indicate densities of viewpoints, or adaptive temporal resolutions). The data structure is

```
struct Image-LAMP-Layer
{
int I[YDIM][KDIM];  // texture map I[y][t]: y = 0~YDIM; k = 0 ~KDIM
float v[YDIM][KDIM]; //depth map v[y][t] : y = 0~YDIM; k = 0 ~KDIM
float t[KDIM]; // super-time index tk; k = 0~KDIM;
}
Image-LAMP-Layer iLAMP[L]; // layer l = 0~L
```

From this representation, we can find the corresponding 3D coordinates of a point *(y,k)* in the image-based LAMP by

$$Z = F\frac{V}{v},\ Y = y\frac{V}{v},\ X = x\frac{V}{v} - Vt_k \tag{8}$$

given *v(y,k)* and $t_k = t(k)$.

As a comparison, Table 1 shows the tradeoff between relief-like and image-based LAMP representation. While the former is more general and more appropriate for complex scenes, the latter is more compact.

Table 1. Comparison between relief-like LAMP and image-based LAMP representations

	Relief-like LAMP	Image-based LAMP
Scene	general, complex scenes	well-layered scenes, 2-3 layers
Representation	inhomogeneous	homogeneous, more compact
Construction	intermediate level	higher level
Rendering	fast	faster

5. DEPTH, OCCLUSION AND RESOLUTION RECOVERY

In this section, we will discuss one approach – our panoramic EPI approach [15, 21] - to obtain a dense depth map for the supporting surface of the relief-like LAMP. Further, we will show how to generate occluded layers and adaptive resolution by selectively using the most essential information from all the EPIs. The support surface is a PVI with selected parallel viewing direction; for example, a viewing direction orthogonal to the camera motion direction when x=0. The panoramic EPI approach consists of four important steps: locus orientation detection, motion/depth boundary localization, depth-texture fusion, and occlusion/resolution recovery. The results of the first three steps are a panoramic depth map with accurate depth boundaries as well as the corresponding texture map (Fig. 4a).

5.1. Depth Recovery: the Panoramic EPI-Based Approach

The panoramic EPI approach we proposed in [15, 21] uses a frequency-spatio-temporal cross-analysis to estimate loci's orientations in each EPI, without explicitly tracking the loci in EPIs. We will give a brief summary of how this goal is achieved in the first three steps of the panoramic EPI approach. The method consists of three steps: frequency domain loci orientation estimation, spatio-temporal domain depth boundary localization, and depth-texture fusion. Since it is very important to accurately localize the depth boundary for layered representation in image-based rendering applications, we will focus on the method for accurate depth boundary localization.

First, a *Gaussian-Fourier Orientation Detector* (GFOD) is performed along a scanline ($x=x_0$) in the EPI, which is the intersection line of this EPI with the supporting PVI. The GFOD operator uses Gaussian-windowed Fourier transforms to detect orientations of the image under the Gaussian window. A large window (e.g. 64×64) is used in order to detect accurate locus orientations. Multiple orientations are detected for a certain temporal range when the GFOD operator moves across a depth boundary. Thus, the Gaussian window is applied to reduce this range by assigning higher weights for pixels closer to the center of the window. However, the response of multiple (two in our current implementation) orientations does not only happen exactly at the point on the depth boundary (see Fig. 8 for locations with two peaks).

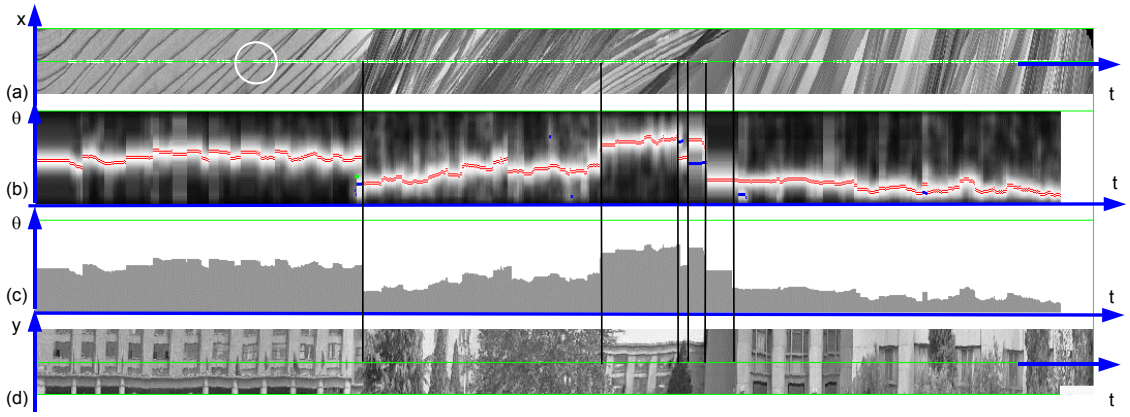


Fig. 8. Multiple orientation detection by Gaussian-windowed Fourier Orientation Detector (GFOD), and depth boundary localization. (a). An x-t image (i.e., EPI, with y = -18) with the processed points (white dots) and a Gaussian window (indicated by a circle). (b) Orientation energy distribution map $P_d(\phi,t)$ and loci orientation detection: the long dashed curve (red in color version) indicates the first selected peaks, and several pieces of solid lines (blue in color version) indicate the second peaks. (c) Loci orientation angles selected after depth boundary localization and depth interpolation. (d) Part of the corresponding PVI (x = 0); the horizontal line in the PVI corresponds to the EPI in (a). Significant depth boundaries are marked by vertical black lines across (a) to (d).

The Fourier transform $G(\xi,\omega)$ is obtained for a 64×64 Gaussian-windowed EPI pattern centered at (x_0, t) (shown as a circle in Fig. 8a). The "energy spectrum" $P(\xi,\omega)=\log(1+G^2(\xi,\omega))$ is mapped into the polar coordinate system (r,ϕ) by a coordinate transformation

$r = \sqrt{\xi^2 + \omega^2}, \phi = \frac{\pi}{2} + \arctan\left(\frac{\xi}{\omega}\right)$. From the resulting polar representation $\mathrm{P}(r, \phi)$, an orientation histogram is constructed as

$$P_d(\phi) = \int_{r_1}^{r_2} P(r, \phi) dr \phi \in [0, \pi] \quad (9)$$

where ϕ corresponds to the orientation angle of the ST texture centered at (x_0, t) and [r_1,r_2] is a frequency range of the bandpass filter, which is selected adaptively according to the spatial-temporal resolution of the image. Initially, r_1 and r_2 are set to 8 and 30, respectively, for a 64×64 window. An orientation energy distribution map $P_d(\phi,t)$ can be constructed (Fig. 8b), which visually represents the depths of the points along the time (t) axis, corresponding to the processed EPI.

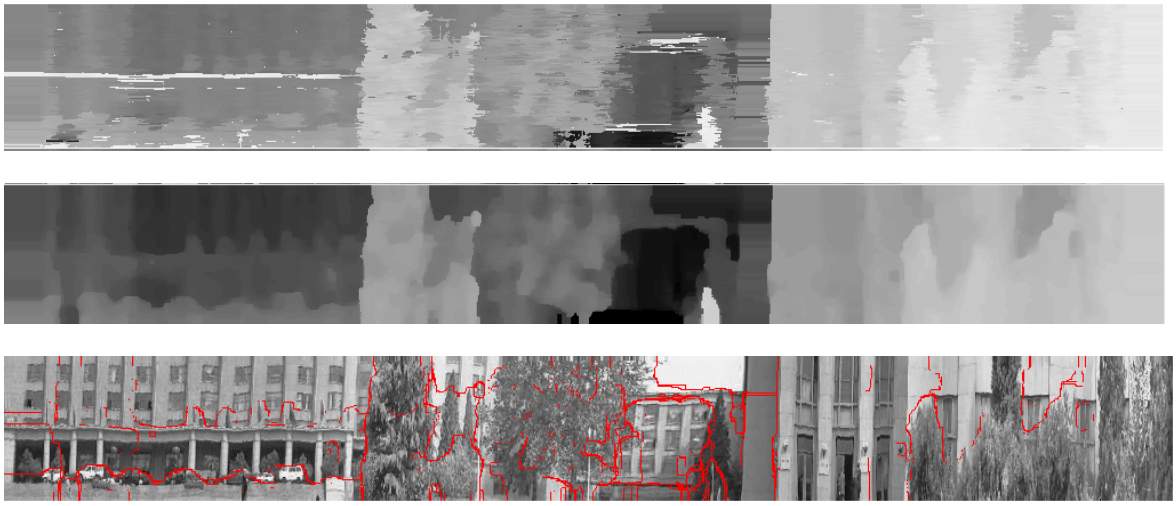


Fig. 9. Panoramic depth construction: raw depth map, refined depth map and the texture maps with depth boundaries superimposed in black lines (red in color version).

In the second step, a *Depth Boundary Localizer* (DBL) is used to accurately localize the depth boundary. It measures intensity consistencies along the two detected orientations with the two highest peaks, taking occlusion/reappearance relations into account. The best consistent measurement should be achieved right at the depth boundary since otherwise one of the measurements will cross both locus patterns (Fig. 8 c; see also Fig. 10a and Fig. 10b).

Then in the third step, a *depth-texture fusion* (DTF) algorithm is applied to reduce the errors produced in each EPI analysis (due to aperture problems, noise, etc.) and to refine depth

boundary localizations. The refinement is based on the observation that a depth boundary almost always coincides with an intensity boundary in a visual scene. This observation is also used by others [22, 25]. Fig. 9 compares the raw depth map (after the first two steps) and the refined depth map, and shows superimposed depth boundaries in the panoramic texture map.

5.2. Depth Boundary Localization

Since multiple orientations are detected not only at but also near the depth/motion boundaries by using the large GFOD operator, a *Depth Boundary Localizer* is designed to determine whether or not the depth/motion boundary is right in the center of the Gaussian window. For the method to be valid for most of the cases encountered in a natural scene and applicable to the EPIs generated by a un-stabilized camera, we use an approach that does not rely on locus tracking (which often fails due to the non-ideal ST textures generated from a complex scene with changing illuminations and un-smooth camera motion). In our algorithm, multiple scale intensity similarities are measured along all the detected orientations $\theta_k\,(k=1,\cdots,K)$ by the GFOD operator. Among them the orientation with the greatest similarity measurement is selected as the correct orientation. Note that only a *comparison-and-selection* operation is used, without assuming any detection of feature points or using any troublesome thresholds.

Consider the case in which two orientations θ_1 and θ_2 ($\theta_1 > \theta_2$) are detected within a Gaussian window. Dissimilarity (i.e., variance) measurements along θ_1 and θ_2 for a given circular window of radius R centered at the point (x_0,t_0) are defined as the variance of intensity values (Fig. 10 (a) and (b); refer to Fig. 8)

$$C_{\pm}(\theta_k, R) = \frac{1}{R}\sum_{r=1}^{R} I^2(\pm r, \theta_k) - \bar{I}_{\pm}^2(\theta_k, R) \quad \text{(k=1,2)} \tag{10}$$

where $r = \sqrt{(x-x_0)^2 + (t-t_0)^2}$, $\quad \bar{I}_{\pm}(\theta_k, R) = \frac{1}{R}\sum_{r=1}^{R} I(\pm r, \theta_k)$.

In the above equations, subscripts '+' and '-' denote the dissimilarity measurements along the detected orientations in positive (+) and negative (-) x directions respectively. This is designed for dealing with the occlusion of a farther object (θ_2) by a closer one (θ_1): the occluding (i.e., closer) object can be seen in both the positive and the negative x directions, but the occluded (i.e., farther) object can only be seen in one of the two directions (Fig. 10a). The dissimilarity measurements for closer and farther objects are defined as

$$E(\theta_1,R)=\frac{1}{2}\left(C_+(\theta_1,R)+C_-(\theta_1,R)\right)/P_d(\theta_1)$$
$$E(\theta_2,R)=\min(C_+(\theta_2,R),C_-(\theta_2,R))/P_d(\theta_2) \qquad (11)$$

respectively. Notice the difference in the two measurements - the occluded object only takes the smaller measurement among the positive and negative directions. In addition, we give more weights to stronger oriented texture patterns: $P_d(\theta_k)$ is the value of the orientation histogram (Eq. (9)) at θ_k (k=1,2). The higher the value is, the lower the dissimilarity measurement should be.

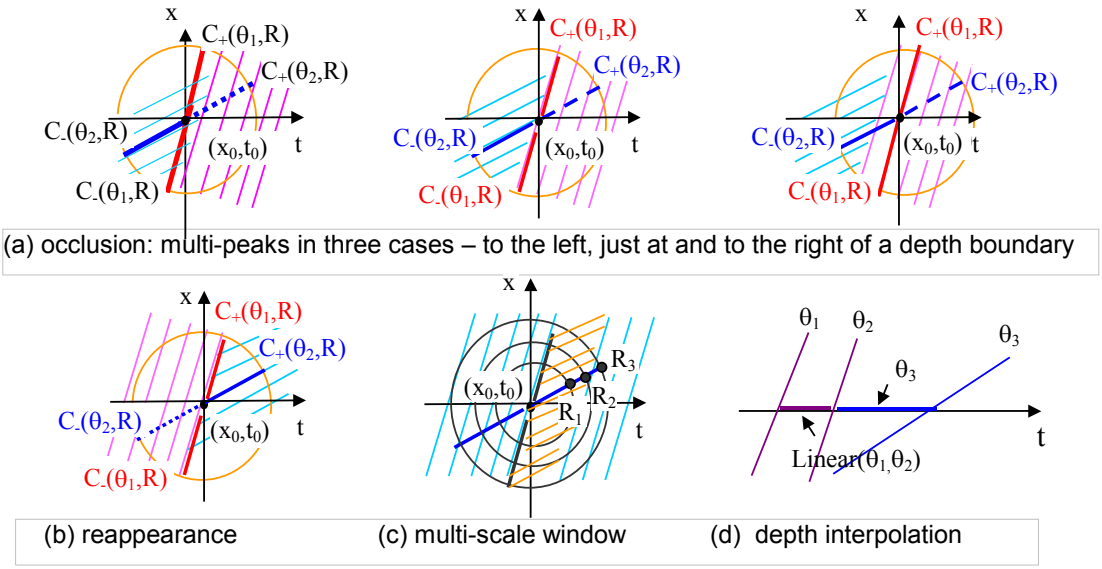


Fig. 10. Principle of the depth boundary localization and depth interpolation

Fig. 10a shows how to use these measurements to localize a depth boundary when the farther object will be occluded by the closer object (which is the *occlusion* case). Multiple peaks are detected by the GFOD operator when the Gaussian window (indicated by circles) is *near* the depth boundary. When the Gaussian window is to the left of the depth boundary, the dissimilarity measurement (i.e. variance) along the locus direction of the occluding object $E(\theta_1,R)$ will be larger, since the measurement is performed across the loci pattern of the to-be-occluded object (left of Fig. 10a). On the other hand, the dissimilarity measurement along the locus direction of the to-be-occluded object $E(\theta_2,R)$ will be much smaller, since the measurement is right along the locus of the to-be-occluded object. Whenever the center of the Gaussian window is precisely *at* the depth boundary, both measurements will be small since both

measurements are along their own loci's directions. However, since the occluding boundary of the closer object usually will be visually stronger than the ST pattern of the occluded object, the measurement will be in favor of the closer object at this location (middle of Fig. 10a). As the center of the window moves into the occluding (closer) object, the dissimilarity measurement of the occluded object will be significantly increased, since the measurement will cross the loci of the occluding object, but the dissimilarity measurement for the occluding object will remain small (right of Fig. 10a). Similar arguments hold for the *reappearance* case, when the occluded object reappears behind the occluding (closer) object (Fig. 10b). Therefore, a simple verification criterion can be expressed as

$$\theta = \begin{cases} \theta_1, \ if\ E(\theta_1, R) \le E(\theta_2, R) \\ \theta_2, \text{Otherwise} \end{cases} \tag{12}$$

In fact, the condition of occlusion and reappearance can be judged either by comparing $C_+(\theta_2, R)$ and $C_-(\theta_2, R)$ (see Fig. 10) or by analyzing the context of the processing (i.e., the change of depths). In the case of occlusion of a far object by a near object (far to near, Fig. 10(a)), we have $C_-(\theta_2, R) < C_+(\theta_2, R)$, and in reappearance (near to far, Fig. 10(b)) we have $C_+(\theta_2, R) < C_-(\theta_2, R)$.

In order to handle cases of various object sizes, different motion velocities, and multiple object occlusions, multiple scale dissimilarity measurements $E(\theta_k, R_i)$ (e.g., i=1,2,3) are calculated within multiple scale windows of radii R_i (i=1,2,3), $R_1<R_2<R_3$. In our experiments, we have selected $R_1=m/8$, $R_2=m/4$, $R_3=m/2$ ($m = 64$ is the window size; see Fig. 10(c)). By defining the following ratio

$$D_i = \frac{\max\left(E(\theta_1, R_i), E(\theta_2, R_i)\right)}{\min\left(E(\theta_1, R_i), E(\theta_2, R_i)\right)} \tag{13}$$

a scale p (p=1,2 or 3) with maximum D_p is selected for comparing the intensity similarities. For example, in Fig. 10(c), R_2 will be selected.

The selected orientation angle θ can be refined by searching for a minimum dissimilarity measurement for a small-angle range around θ. The accuracy of the orientation angle, especially that of a far object, can be improved by using more frames.

In order to obtain a dense depth map, interpolations are applied to textureless or weak-textured regions/points where no orientation can be detected. The proposed interpolation method (Fig. 10(d)) is based on the fact that a depth discontinuity almost always implies an occluding boundary or shading boundary. The value $\theta(t)$ between two instants of time t_1 and t_2 with estimated orientation angles θ_1 and θ_2 is linearly interpolated for smooth depth change (i.e., $|\theta_1 - \theta_2| < T_{dis}$, T_{dis} is a threshold), and is selected as $\min(\theta_1, \theta_2)$, i.e., the angle of the farther object, for depth discontinuity (i.e., $|\theta_1 - \theta_2| \geq T_{dis}$). The processing results of dense depth measurements from a real *x-t* image(EPI) are shown in Fig. 8c.

5.3. Occlusion Recovery

Because the supporting PVI only contains information from a single viewing direction (for example, the direction perpendicular to the motion direction when we select a PVI with $x_0 = 0$), some parts of the scene that are visible in other parts of images from an original (or a stabilized) video sequence are missing due to occlusion. They will be recovered by analyzing depth occlusion relations in the EPIs. The basic algorithm is performed in each EPI after the panoramic depth map and its depth boundaries have been obtained. The algorithm consists of the following steps (Fig. 11, Fig. 12):

Step 1. Find the location of a depth boundary - A point on a depth boundary, p0(x_0,t_0), and orientation angles (θ_2 and θ_1) of the occluded (far) and occluding (near) objects are encoded in the depth map. A point is considered as a point on the depth boundary with a depth discontinuity when, for example, $|\theta_1 - \theta_2| > 2^\circ$.

Step 2. Localize the missing part - The missing (occluded) part is represented by a 1D (horizontal) spatio-temporal segment $p_s p_e$ in the EPI, on which points have the same parallel viewing directions but from moving viewpoints. It is calculated from the slopes of the two orientation patterns that have generated the depth boundary, and it is denoted by an *x* coordinate and start/end frames (t_S/t_E). Basically, the largest possible angle of viewing direction (indicated by the *x* position in the EPI) from the viewing direction of the PVI (indicated by the x_0 coordinate) possesses the most missing information, but possible occlusions by other nearby objects should be considered. For example, the second missing region from the right in Fig. 12b was determined by checking the occlusion of the locus patterns of the missing part against those

of other nearby foreground objects (trees), resulting in an ST segment with smaller x coordinate, i.e. smaller viewing angle from the viewing direction of the PVI. In this way, a triangular region $p_0p_sp_e$ can be determined, and the 1D segment p_sp_e will be used as the texture of the missing part that is occluded by the foreground objects.

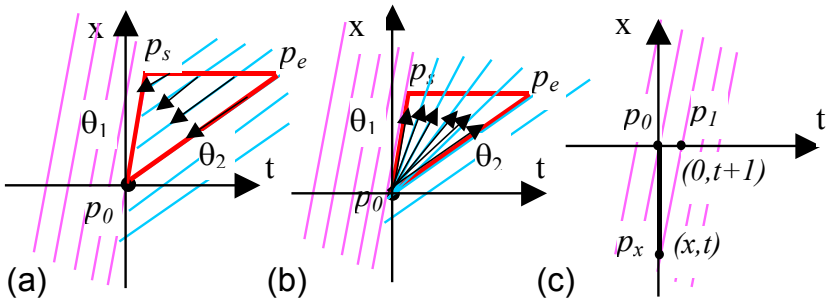


Fig. 11. Region classification and adaptive resolution. (a) Locus patterns near an occlusion boundary; (b) locus patterns of front- side surfaces and (c) temporal resolution enhancement by using spatial resolution.

Step 3. Verify the type of the missing part. The triangular region also contains depth information of the missing part - the 1D segment p_sp_e. In principle, similar treatments can be made here as for the basic depth map as in Eq. (10). For simplicity, the missing parts are classified into two types in our current implementation: OCCLUDED and SIDE. If the loci within the triangular region form a parallel pattern of angle θ_2 (Fig. 11a), then the missing part is classified as OCCLUDED; otherwise, as SIDE if the angle of the oriented pattern

$$\theta_t = \theta_1 + \frac{\theta_2 - \theta_1}{t_e - t_s}(t - t_s) \qquad (14)$$

changes linearly inside the triangle (Fig. 11b). By assuming the missing part as each of the two types, an overall consistent measurement along the hypothesized locus orientations (indicated by arrows in Fig. 11a and b) can be calculated within the triangle region (similar to measurements in motion boundary localization [15]). The type of missing part is selected as the one with the better consistent measurement of the two hypotheses. The loci's angle θ of the OCCLUDED region will be the same angle θ_2 as the occluded object, whereas loci's angle θ of the SIDE region gradually changes from θ_1 to θ_2 (or from θ_2 to θ_1), as expressed in Eq. (14).

In this way, the depths of the occluded or side region can be assigned. Fig. 11 illustrates the situation of reappearance where the farther object re-appears behind the closer object. Fig. 12 shows other situations (occlusion and side) with real image.: Fig 12(a) shows an example of recovering an occluded region (building façade) behind a tree, as indicated by the first circle in Fig. 7. Fig 12(b) shows three recovered "side" regions. The first two correspond to the first side of the building indicated by the second circle in Fig. 7, which is separated into two regions by a tree in front of it. The third side region corresponds to the second side façade indicated by the third circle in Fig. 7. The x location of the second side region is much closer to the supporting PVI (with $x = 0$), since it is occluded by the tree.

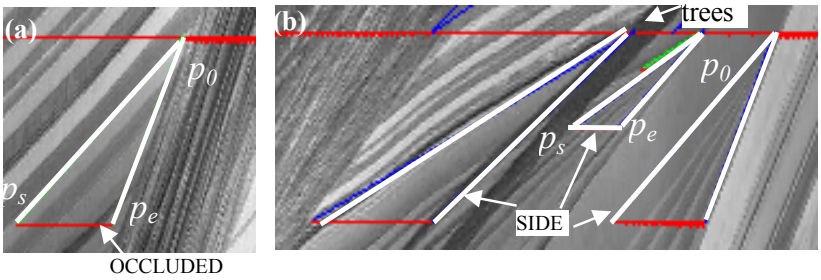


Fig. 12. Occlusion and resolution recovery results in real EPIs. (a) an OCCLUDED region; (b) three SIDE regions (two of them belong to a side facade separated by trees). Circles in Fig. 7 show the corresponding OCCLUDED and SIDE regions in this figure.

5.4. Resolution Recovery

As we have shown, the panoramic view image (PVI) in the t direction is under-sampled if the image velocity v of a point in the PVI is greater than one pixel per frame (as in the right hand part of Fig. 4a, where the images of doors of the building look thinner than the real ones). To enhance temporal resolution, a v-pixel segment in the x direction is extracted from the EPI (as opposed to a single pixel in a PVI, as in Fig. 2 and Fig. 4a). Fig. 11c shows the principle: a nice feature is that the front (*relief*) point $p_x(x,t)$ of an x-segment p_0p_x at time t will exactly connect with the back (*supporting*) point $p_1(0,t+1)$ of the x-segment at time $t+1$ in the relief-like LAMP representation, since both image points are on the same locus of a 3D point. This property has been used to generate seamless, adaptive-time panoramas in Fig. 7. The thickness of the red

(dark in B_W version) horizontal lines (including those of SIDE and OCCLUDED regions) in the two EPI tiles in Fig. 12 indicates the number of points (pixels) to be extracted in the *x* direction of this epipolar plane image. As shown in Fig 7, a seamless panoramic mosaic is generated after resolution enhancement, where much higher temporal resolutions are achieved in the second part of the mosaic.

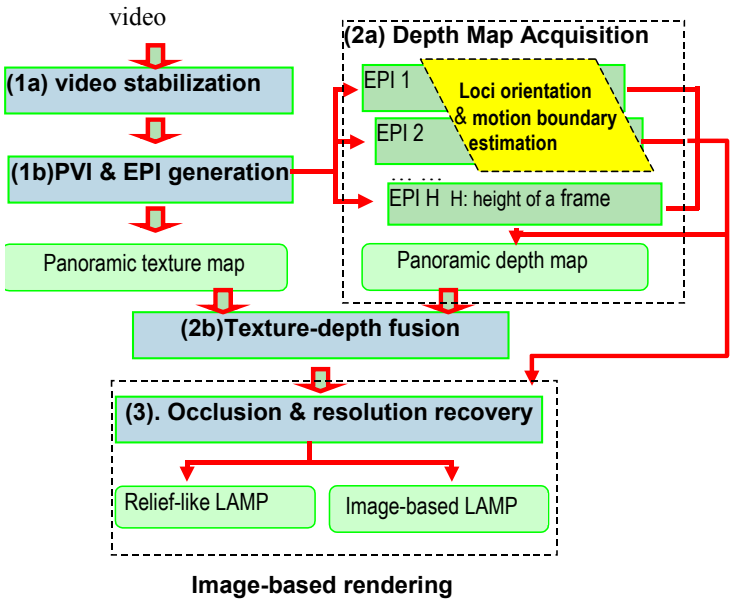


Fig. 13. System diagram of 3D panoramic scene modeling (PVI: Panoramic View image; EPI: Epipolar Plane Image; LAMP: Layered, Adaptive-resolution and Multi-perspective Panorama)

6. LAMP Construction and Rendering: Experimental Results

6.1. 3D LAMP Construction Results

Our 3D panoramic scene modeling approach consists of three modules: (1) video stabilization (and PVI/EPI generation), (2) dense depth map acquisition (with texture-depth fusion), and (3) depth layer construction (with occlusion/resolution recovery). The system diagram is shown in Fig. 13. For using the EPI-based approach to recover dense depth maps, our method requires a dense, long, 1D translational video sequence as the input in order to generate parallel-perspective panoramic mosaics. It is particularly effective when the number of frames is larger than the width of an original video frame in order to generate mosaics with "panoramic" view. In the

ideal case, the representation of parallel projection requires a pure 1D translation along one image axis. However our algorithms still work under more practical motion given that a video stabilization module is used to preprocess the sequence. Our 3D video stabilization module estimates interframe camera motion, models the camera motion of the image sequence as a 1D dominant translation plus a small vibration (the latter is modeled by a homography between successive frames), and removes the vibration by a motion filtering and image warping step so that a rectified video sequence with virtually 1D translational motion can be created. Detailed algorithms and results can be found in [15, 21]. The depth map acquisition module was summarized in Section 5.1. LAMP representations and construction are presented in Sections 3 and 4, while the algorithms for occlusion/resolution recovery has been discussed in Sections 5.2 and 5.3. In this section, we will present some construction results from real image sequences, look at some practical considerations in LAMP representations, and discuss the LAMP-based 3D rending issues.

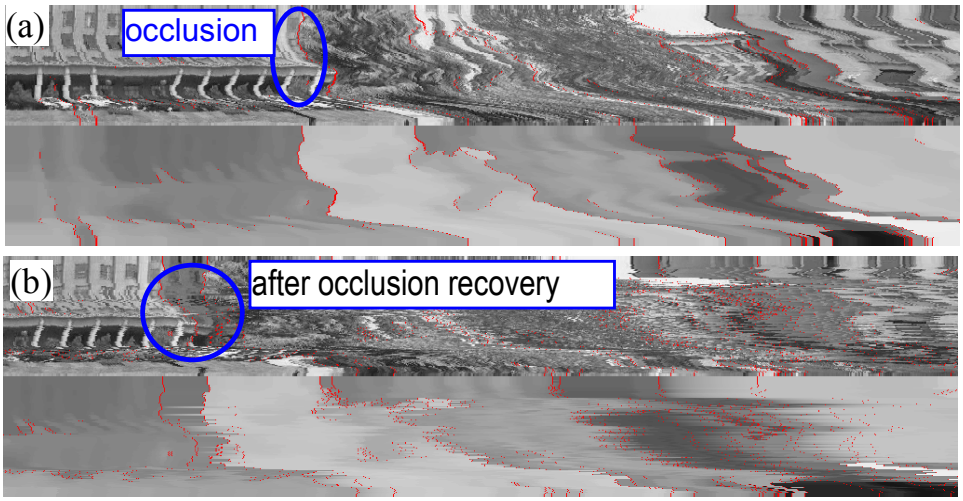


Fig. 14. Internal data of (a) the base layer, and (b) all the layers of (part of) the LAMP of the MB sequence. The images are truncated for fitting into the page.

The MB sequence was captured by a camcorder mounted on a tripod carried by a moving vehicle. Aside from the small vibration of the vehicle on a normal outdoor road, the velocity of translation was mostly constant. The small vibration in the video sequence (which is demonstrated by the irregular boundaries of the mosaic in Fig. 7) was removed by a video stabilization process discussed in our previous work[15, 21]. The frequency-domain depth estimation method is robust for dealing with the non-ideal loci created from this real-world video

sequence with vibration. We have shown the supporting surface and the relief-surface of the relief-like LAMP representation constructed from the MB sequence; Fig. 14 shows the internal data for the base layer and all the layers. Recall that a relief-like LAMP representation is part of a 3D *xyt* cube. That is to say, for each pixel *(y,t)* in the supporting surface, there is an *x*-segment of *v* pixels in the *x* direction. The internal data of the base layer is displayed in Fig. 14a as the sequential head-tail arrangement of all the segments of the base layer in each row. We have found that both the texture and the depth map show almost seamless connections between successive segments, and it is obvious that higher resolutions than in the corresponding PVI are recovered for closer objects. The internal data of the relief-like LAMP, including all the occluded layers, is displayed in Fig. 14b as similar head-tail arrangements of all the *x*-segments in all the layers, including all the occluded layers as well as the base layer. The *x*-segments in the occluded layers are inserted into the locations of their corresponding depth boundaries. Comparing Fig. 14b to Fig. 14a, additional data for the occluded regions are shown. For example, the portion of the building's facade occluded by the trees and the side facades of the building are partially recovered. Note that since both images are truncated to the same length, leaving out different amounts of scene points on the right side.

The compactness of the LAMP representation can be seen from the following key numbers for the MB sequence (W*H*F= width*height*frames, P: number of pixel per location in the PVI):

- Original video sequence: S_0 = W*H*F=128*128*1204 bytes (gray level images) = 16MB
- PVI supporting surface (both texture and depth in 1 byte/pixel): S_p=H*F*2 = 128*1024*2 bytes = 256KB
- LAMP representation (texture 1 byte/pixel, depth 4 byte/pixel, as in the **x_Streak** structure in Section 3): 1.739 MB.

The last number approximately corresponds to P = 3 pixels per location and 5 bytes per pixel on average in the PVI supporting surface, which count for both adaptive resolution and occlusion representation. This ends up with an estimated LAMP size of 128*1024*5*3 = 1.92MB, which is close to the real size. If we use 1 byte/pixel to represent depth, then the LAMP size will be

S_l = H*F*2*P = 128*1024*2*3 = 768 KB,

which is simply three times that of the PVI supporting surface.

In conclusion, the size of the LAMP representation in this example is about 1/10 (if floating point depth is saved) or 1/20 (if byte/pixel depth is saved) of the data size of the original video sequence. In the general case, the LAMP-to-sequence size ratio is $S_l/S_o = 2P/W$, where W is the width of each frame in the original video sequence and P is the number of pixels per location of the PVI supporting surface.

6.2. Extended Panoramic Image (XPI) Representation

We have shown in [15] how to make full use of the original image sequence by generating an extended panoramic image (XPI). Suppose that an image sequence has F frames of images of size W×H. An example is the frequently used flower garden (FG) sequence(W×H×F = 352×240×115). A PVI and an EPI is shown in Fig. 15a and Fig. 15b. It is unfortunate in this case that the field of view of the panoramic view image turns out to be "narrow" due to the small number of frames and large interframe displacements. Therefore, an extended panoramic image (XPI) is constructed. The XPI is composed of the left half of frame m/2 (m is the GFOD window size), the PVI part formed by extracting center vertical lines from frame m/2 to frame F-m/2, and the right half of frame F-m/2 (Fig. 15c). The total width of the XPI is $W_x = W/2 + (F - m) + W/2 = W+F-m$.

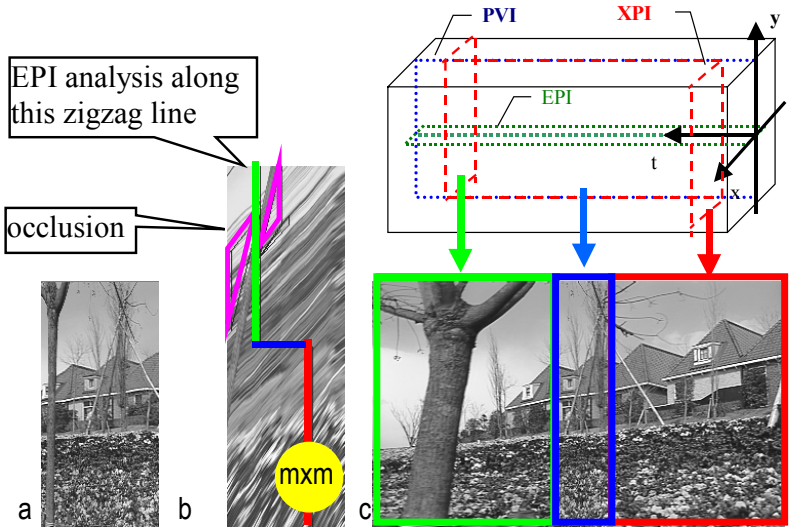


Fig. 15. PVI, EPI and XPI of the flower garden (FG) sequence

Fig. 16 shows the results of 3D recovery of the XPI of the FG sequence. In the depth map, the tree trunk stands out distinctly from the background, and the gradual depth changes of the

flowerbed are detected. The occluded regions and resolutions are recovered in a way similar to that for a PVI image, except that the EPI analysis is performed along a zigzag line (Fig. 15b).

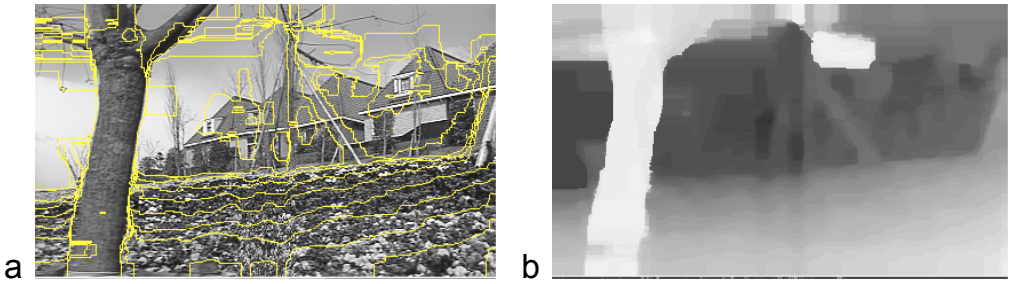


Fig. 16. Panoramic depth map for the FG sequence. (a) Isometric depth lines overlaid in the intensity map (b)panoramic depth map.

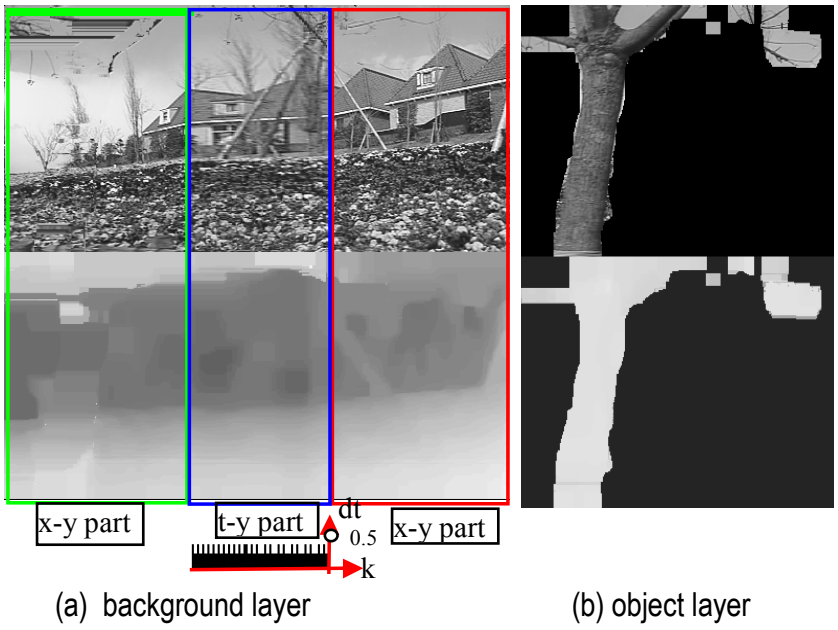


Fig. 17. Image-based LAMP model of the FG sequence

Fig. 17 shows the two extracted layers of the image-based LAMP representation for the FG sequence, each of which has both texture and depth maps. These two pairs of images, along with a 1D temporal sampling rate array (Eq. (7)) for each layer, are constructed from the 115-frame FG image sequence. In the *yt* part of the background layer of this extended image-based LAMP, the time-sampling rate is adaptively changed according to the dominant depth in each time instant. The time scales are less than 0.5 (frames); this means that higher resolution is achieved than that of the original PVI (shown in Fig. 16) in both texture and depth maps (the uneven

super-timing in Fig. 17a is due to quantization). Furthermore, the background regions occluded by the tree trunk have been completely recovered, and have been merged into the background layer with both texture and depth values.

The size of the XPI-based LAMP representation in the general case is

$$S_{xl} = (W+(F-m)*P)*H*2*L \text{ (bytes).}$$

where W*H is the size of an original frame, F is the number of frames in the sequence, m is the size of the Gaussian window (m = 64 in our experiments), P is the number of pixels per location in the PVI part of the XPI image, and L is the total number of layers. The real sizes of the image-based representation of the FG sequence is S_{xl} = (114*2 + 95*2) KB = 418 KB. Compared to the size of the original video W×H×F = 352×240×115 = 9.265MB, the real compression ratio is 22.7. The estimated size using the above equation is

$$S_{xl} = (352+(115-64)*2)*240*2*2 = 423 \text{ KB}$$

assuming P = 2 and L = 2. The theoretical LAMP-to-sequence ratio is $S_{xl}/S_o = 2L/F + 2PL/W$, assuming that F is much larger than m.

6.3. LAMP-Based Rendering

The 3D LAMP model is capable of synthesizing images from new viewpoints differing from the viewpoints of the original image sequence thanks to its multi-perspective representation, adaptive resolution, occlusion representation, and depth information. The mapping from a pixel in a relief-like LAMP model to its 3D coordinate can be computed by using Eq. (4), and then it is straightforward to re-project it to the desired view (in practice an inverse mapping should be applied). Because of the neighborhood relations of pixels in the LAMP representation (refer to Fig. 14), a rendering algorithm can easily perform interpolation between neighborhood pixels. In a relief-like LAMP, an attached layer is always occluded by the layer to which it is attached. So, when rendering the attached layers should be drawn first so that an occluded region could be seen correctly in a new view. Thanks to the adaptive resolution representation, the higher resolutions in the original images can be applied to a new view whose viewpoint is close to those of the original image sequence. Fig. 18 shows the preliminary "rendering" results from the relief-like LAMP of the MB sequence. The development of a rendering system based on relief-like representation is still underway.

(a) (b)

Fig. 18. “Rendering” results from relief-based LAMP representation of the MB sequence. (a) Normal rendering (with correct occlusion checking of different layers), where the building façade is occluded by the tree. (b) The part of the façade is seen through the tree. Note that that part is in shadow in the original video sequence, and therefore it is darker.

The rendering process is easier using an image-based LAMP model. For a LAMP model based on an XPI (i.e., a combination of x-y and y-t images) in general, Eq. (4) and Eq. (8) should be used in the two *x*-*y* parts and the one *y*-*t* part, respectively. Fortunately, both equations are very simple and virtually the same (but reflect the different ways to obtain *x* and *t* indices) so that a fast implementation is feasible. In order to achieve a correct occlusion relation, a rendering from a new view should begin with the farthest layer and end with the nearest layer, from the viewpoint of a synthetic image.

A simple image based 3D rendering program has been developed to demonstrate the capability of image synthesis of arbitrary views using the image-based LAMP representation. Synthetic sequences with a virtual camera of 6 DOF of motion generated from the LAMP model (with and without the object layer) of the flower garden sequence can be found on our web page [20]. The compactness of the image-based LAMP representation enables the rendering algorithm to achieve real-time performance. Fig. 19 shows some snapshots of the synthesis with both the background and foreground layers, without the foreground layers. The perspective distortion shown at the borders of the synthetic images in the second and the third rows in Fig. 19 clearly reveals the accuracy in the recovered depth changes for the tree, house, and garden. The first synthetic images in both the second and the third rows were generated from a viewpoint farther from the original camera path so that the entire scene can be seen.

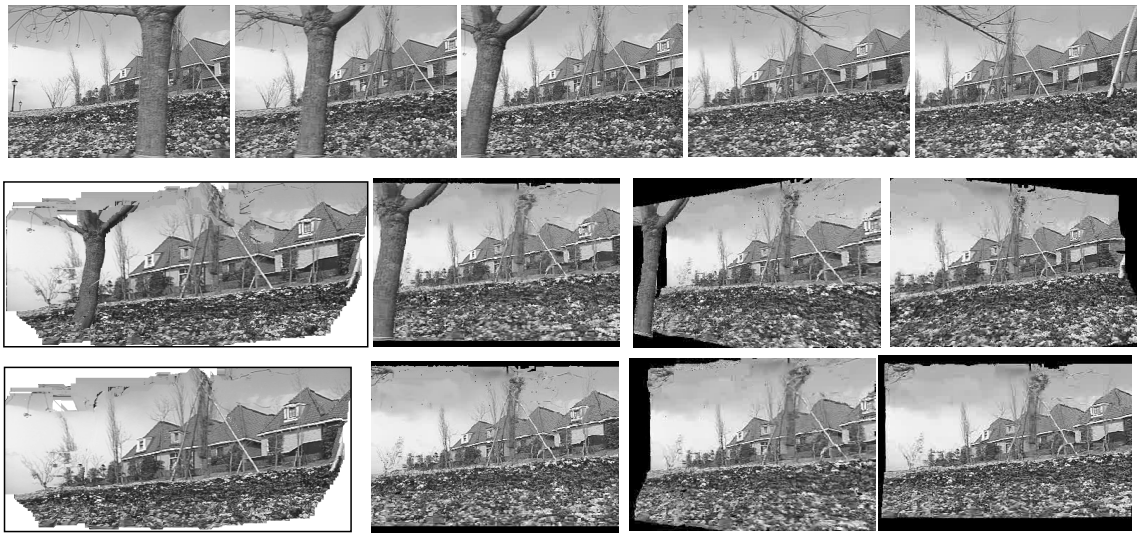


Fig. 19. Synthetic images from the image-based LAMP model. First row: Snapshots of original video sequence; second row: snapshots of the virtual walk-through with both the background and the trees; third row: snapshots of synthetic results without the tree. The first synthetic images in the 2nd and 3rd rows are generated from viewpoint that can see all the scene points in the LAMP model.

7. COMPARISONS AND DISCUSSIONS

7.1. Layered Representation

The LAMP representation is related to such representations as multi-perspective panoramic view images (PVIs), sprites, and layered depth images (LDIs). However, it is more than a multi-perspective PVI in that depth, adaptive-resolution and occlusion are added. It is different from a "sprite" (or LDI) since the sprite or LDI is a view of a scene from a single input camera view and is without adaptive image resolution. We compare our results [17] with other layered representations [10, 11, 13, 22, 23]. A comparison in number of input images, results of layered representations, and algorithm performance for the flower garden sequence is summarized in Table 2.

Usually, in a layered representation a set of depth surfaces is first estimated from an image sequence of a moving camera and then combined to generate a new view. The *layered representation* proposed in [10] consists of three maps in each layer: a mosaicing intensity map,

an alpha map, and a velocity map. The velocity map is actually a set of parameters of the affine transformation between layers. Occlusion boundaries are represented as discontinuities in a layer's alpha map (opacity). In [10], the first 30 frames of the flower garden sequence were used as input. The output were three affine plane layers - the tree, the house, and the flowerbed. Occlusion (by the tree) was recovered, but no depth information was provided (i.e., each layer is only modeled as an affine plane). Processing of the 30 frames of 720*480 images took 40 minutes in a HP 9000 series 700 workstation.

Table 2. Comparison of 4 layered representation results for the flower garden sequence

	Wang-Adelson [10]	**Ke-Kanade [22]**	**Sawhney-Ayer [11]**	**Baker-Szeliski-Anandan[12]**	**Image-based LAMP**
Input	first 30 frames (720x480)	two frames	a few frames	first 9 even frames	all 115 frames (350x240)
Rep.	3 planar layers (tree, flower bed, house)	4 planar layers (tree, branch, house, flowerbed)	4 layers (tree, flower bed, house, sky)	6 layers of Sprites (3 tree, 2 flower bed, 1 house)	2 layers of LAMP (tree, background)
Depth	no	no	no	yes	yes, each obtained from 64 frames
Occlusion	recovered	not recovered	not recovered	recovered	recovered
Multi-view	affine mosaic rep.	2D layer rep.	single view mosaic	single view mosaic	multiple-view mosaic
Adaptive res.	no	no	no	a perspective image	adaptive resolution
Performance	40 mins / 30 frames in a HP 9000 series 700 workstation	not available	not available	not available	14 mins / 115 frames in a 400 MHz PC (general C code)

The subspace approach [22] is an effective method for 2D layer extraction from an uncalibrated image sequence. For the flower garden sequence, four planar layers, which roughly correspond to tree, branch, house, and flowerbed, are extracted from two frames in the sequence. Depth boundaries are accurate after the layer refinement step using color segmentation results (similar to our depth-texture fusion step). However, depth information of each layer is not obtained since they are 2D layers, and the occluded regions are not recovered since only two frames are used.

The multiple motion estimation method based on MDL and EM algorithms in [11] is computationally expensive. It requires a combinatorial search to determine the correct number of layers and the "projective depth" of each point in a layer. For the flower garden sequence, only a few frames were processed, and a six parametric motion model was used to segment the scene

into four layers - tree, house, flowerbed and sky. Occlusion regions are not recovered in the layered model, and depth was not estimated in the flower garden example.

In the sprite representation [12], each layer consists of an explicit 3D plane equation, a texture map (a *sprite*), and a map with depth offset relative to the plane. For the flower garden sequence, eight-parameter homography was used to fit the planar layers. The first nine even frames were used in the experiment, and the result was six layers of "sprites" - three for the tree, two for the flowerbed, and one for the house. Depth information was attached to each pixel, and the occlusion (by the tree) was recovered. Each layer was a single perspective view mosaic. More recent work on 3D layer extraction by the same research group [23] uses an integrated Bayesian approach to automatically determine the number of layers and the assignment of individual pixels to layers. The approach is a very general one. However the segmentation results for the flower garden sequence are not as satisfactory as the results in [12]. As the authors pointed out in their paper, there was always a danger in choosing the prior in order to obtain a desired result of layering, and this remains a challenging research issue.

For the flower garden sequence, our method segments the scene into two complete layers of the image-based LAMP representation: the tree and the background. Each layer has both a texture map and a dense depth map. The segmentation is very accurate along the depth boundaries, the occluded regions in the selected panoramic view are recovered from other views, and the images have adaptive resolutions. Our algorithms can process 115 frames of 350*240 images in 14 minutes on a 400 MHz PC to obtain the final LAMP model. The processing time reduces to 2.6 minutes on a Xeon 2.4 GHz dual-CPU Dell Linux workstation. The time includes reading all of the 240 EPIs from a SCSI hard disk and displaying the processed EPIs and resulting images on the screen.

7.2. Full Perspective, Full Orthogonal, Multi-Perspective and Multivalued Representations

There are at least three existing geometric representations for large image sequences, based on parallel projection (i.e., a textured digital elevation map), single-view perspective mosaicing (e.g. sprite), and parallel-perspective panorama (i.e., PVI). A parallel projection of the panoramic depth in Fig. 4a of the MB sequence is shown in Fig. 20a that correctly shows the aspect ratio of depth and size (height and width) of the scene. However, the resolutions of the nearer objects are significantly decreased (or lost) because of this evenly sampled representation.

A single view perspective representation, on the other hand, is not a good representation method for surfaces of varying orientations and over a large distance (Fig. 20b). A very small depth estimation error in the single-view-based representation could be greatly enlarged when re-projecting to a mosaiced image of a single referenced viewpoint.

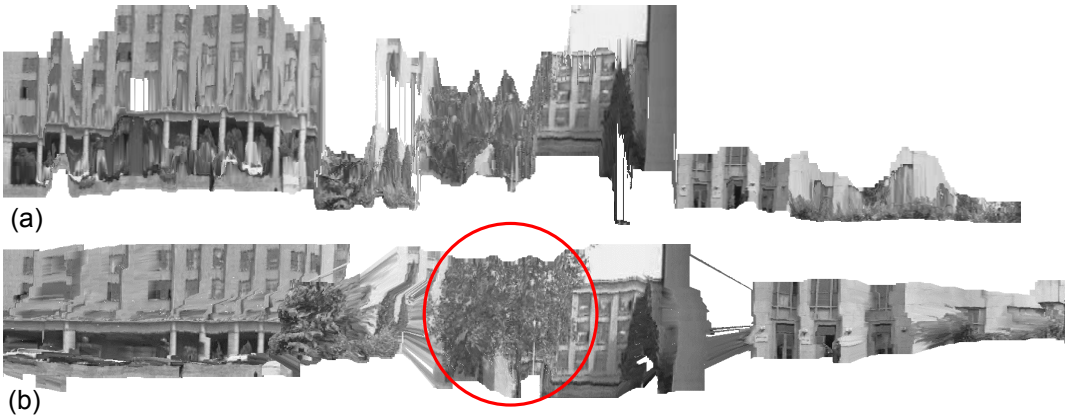


Fig. 20. Two geometric representations of the 3D panorama of the MB sequence. (a) Parallel projection (b) Perspective projection from a viewpoint at the center of the path where the video was captured

A parallel-perspective panoramic view image (PVI) is a compact image-based representation that better represents the image data captured by a camera translating over a long distance. Based on the multi-perspective panorama, we have proposed a still more compact and more powerful representation - 3D layered, adaptive-resolution and multi-perspective panorama (LAMP). To our knowledge, this seems to be the first piece of work that integrates multi-perspective panoramas and layered representations with adaptive image resolutions in a unified model. The relief-like LAMP is basically a single extended multi-perspective panoramic view image (PVI) with both texture and depth values, but each pixel has multiple values to represent results of occlusion recovery and adaptive resolution enhancement. The image-based LAMP, on the other hand, consists of a set of multi-perspective layers, each of which has both texture and depth maps with adaptive densities of viewpoints depending on depths of scene points. No assumption is made on the structures of a scene in constructing its 3D LAMP representations. The LAMP representations are effective for image-based rendering.

We have noticed that our LAMP representations are very similar to the multi-valued representation (MVR) proposed by Chang and Zakhor [26] in that (1) dense depths are

generated, (2) depth regions are grouped by occlusions rather than affine motions, and (3) multiple values are extracted for each pixel. However, there are some fundamental differences. First, with respect to geometry, we use a *single* parallel-perspective geometry rather than *multiple* selected perspective views. Therefore, our LAMP representation is more compact for long (and particularly translational) video sequences. Second, the multiple values per pixel in our LAMP representations come from both occlusion regions and *adaptive resolution*. Third, the methods for constructing dense depth maps and then for generating layered representations are different. A multi-frame stereo approach was used to obtain multiple depth maps for the MVR, while a panoramic EPI-based method was used to recover the depth of only one parallel-perspective panoramic view in our LAMP construction. Therefore, our method is more efficient since we do not recover depth maps with any redundancy, and yet it is more robust since we use many frames (64 frames) instead of just a few frames to estimate the depth for each point. Finally we shall point out that the limitation of our method is that translational motion for a long video sequence is required. Generalization of the LAMP representation to a more general motion is possible, but it needs further research.

8. CONCLUSIONS

We have proposed a compact representation - 3D layered, adaptive-resolution and multi-perspective panorama (LAMP). Two kinds of 3D LAMP representations are constructed, both of which concisely represent almost all the information from a long image sequence, including texture, depth, and occluded regions. The relief-like LAMP is based on a single 2D multi-perspective panoramic view image (PVI) with both texture and depth values, but each pixel has multiple pairs of texture and depth values to represent results of occlusion recovery and resolution enhancement. The image-based LAMP, on the other hand, consists of a set of 2D multi-perspective layers, each of which has both 2D texture and depth maps with adaptive time-sampling scales depending on depths of scene points (in the direction of parallel projection). The 3D LAMP is layered according to the occluding relations of a scene rather than merely the depths. The motivation for layering is to represent occluded regions and the different spatial resolutions of objects with different depth ranges; meanwhile the model is represented in the form of seamless multi-perspective mosaics with viewpoints spanning across a large distance.

The LAMP representation is related to such representations as PVI, sprite, and layered depth image (LDI). However, it is more than a multi-perspective PVI in that depth, adaptive-resolution, and occlusion are added in our representation. It is different from the "Sprite" (or the layered depth image) since the latter is a view of scene from a single input camera view and is without adaptive image resolution. The 3D LAMP representation is capable of synthesizing images of new views within a considerably arbitrary moving space, since the intensity and depth maps derive almost all the information from the original image sequences. The 3D LAMP representations are concise, practical, and powerful representations for image-based modeling and rendering. In future work we plan to extend the 3D LAMP models to represent a large-scale scene where the camera moves along more general paths, and also to explore other approaches for constructing 3D LAMP representations.

9. ACKNOWLEDGEMENTS

This work is partially supported by the China Natural Science Foundation under contact number 69805003, by AFRL under award number F33615-03-1-63-83, by the National Science Foundation under grant EIA-9726401, by DARPA under contract numbers DABT63-99-1-0004 and DABT63-99-1-0022, by the Army Research Office under contract number DAAD19-02-1-0133, and by the CUNY Graduate Research Technology Initiative (GRTI) program. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily represent those of the sponsors. A short version of this work has appeared in *the Proceedings of the Eighth IEEE International Conference on Computer Vision* [19]. We would like to thank the anonymous reviewers for their valuable comments and suggestions, and Mr. Robert Hill at the City College of New York for proofreading the final manuscript.

10. REFERENCES

[1]. S. E. Chen, QuickTime VR - an image based approach to virtual environment navigation. In *Proceedings of SIGGRAPH 95,* 1995: 29-38.

[2]. H.-Y. Shum and R. Szeliski. Construction of panoramic mosaics with global and local alignment. *International Journal of Computer Vision*, 36(2), February 2000:101-130.

[3]. H. S. Sawhney, R. Kumar, G. Gendel, J. Bergen, D. Dixon, V. Paragano, VideoBrush™: Experiences with consumer video mosaicing. In *Proceedings of Workshop on Applications of Computer Vision* (*WACV'98),* 1998: 56-62.

[4]. H. Ishiguro, M. Yamamoto and S. Tsuji, Omni-directional stereo for making global map. In *Proceedings of Computer Vision and Patten Recognition (CVPR'90),* 1990: 540-547.

[5]. S. Peleg, M. Ben-Ezra and Y. Pritch, OmniStereo: panoramic stereo imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, March 2001: 279-290.

[6]. H.-Y. Shum and R. Szeliski, Stereo reconstruction from multiperspective panoramas. In *Proceedings of IEEE Conference on Computer Vision* (*ICCV'99), 1999*: 14-21.

[7]. J. Y. Zheng and S. Tsuji, S Panoramic representation for route recognition by a mobile robot. *International Journal of Computer Vision*, 9(1), 1992: 55-76.

[8]. S. Peleg, B. Rousso, A. Rav-Akha and A. Zomet, Mosaicing on adaptive manifolds. *IEEE Transactions on Pattern Recognition and Machine Analysis*, 22(10), October 2000: 1144-1154.

[9]. P. Rademacher and G. Bishop, Multiple-center-of-projection images. In *Proceedings of SIGGRAPH'98*: 199-206.

[10]. J. Wang and E. H. Adelson, Representation moving images with layers. *IEEE Transactions on Image Processing,* 3(5), 1994: 625-638.

[11]. H. S. Sawhney and S. Ayer, Compact representation of videos through dominant and multiple motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8), 1996: 814-830.

[12]. S. Baker, R. Szeliski and P. Anandan, A layered approach to stereo reconstruction. In *Proceedings of Computer Vision and Patten Recognition (CVPR'98), 1998:* 434-441.

[13]. J. Shade, S. Gortler, L. He. and R. Szeliski, Layered depth image. In *Proceedings of SIGGRAPH'98, 1998:* 231-242.

[14]. R. C. Bolles, H. H. Baker and D. H. Marimont, Epipolar-plane image analysis: an approach to determining structure from motion. *International Journal of Computer Vision,* 1(1), 1987: 7-55.

[15]. Z. Zhu, G. Xu, X. Lin, Panoramic EPI generation and analysis of video from a moving platform with vibration, In *Proceedings of Computer Vision and Patten Recognition (CVPR'99), 1999*: 531-537.

[16]. Z. Zhu, E. M. Riseman and A. R. Hanson, Parallel-perspective stereo mosaics, In *Proceedings of the IEEE International Conference on Computer Vision*, Vancouver, Canada, July 9-12, 2001, vol I, 345-352.

[17]. N. L. Chang and A. Zakhor, View generation for three-dimensional scene from video sequence. *IEEE Transactions on Image Processing*, 6(4), 1997: 584-598.

[18]. R. Szeliski, A multi-view approach to motion and stereo. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition(CVPR'99),* 1999: 157-163.

[19]. Z. Zhu and A. R. Hanson, 3D LAMP: a new layered panoramic representation. In *Proceedings of the IEEE International Conference on Computer Vision*, Vancouver, Canada, July 9-12, 2001, vol II, 723-730.

[20]. Z. Zhu, *3D LAMP: Layered Representation for image-based rendering*,

URL for the project: http://www-cs.engr.ccny.cuny.edu/~zhu/panorama3D.html

URL for a movie: http://www-cs.engr.ccny.cuny.edu/~zhu/fg3dLAMP4M.mpg

[21]. Z. Zhu, *Full View Spatio-Temporal Visual Navigation - Imaging, Modeling and Representation of Real Scenes*, China Higher Education Press, December 2001, First Hundred National Excellent Doctorate Dissertations Series (Based on his Ph.D. Thesis, Department of Computer Science and Technology, Tsinghua University, 1997. English Version may be found at http://www-cs.engr.ccny.cuny.edu/~zhu/PhD-Thesis/).

[22]. Q. Ke and T. Kanade, A subspace approach to layer extraction. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, Hawaii, Dec. 2001.

[23]. P. H. S. Torr, R. Szeliski, and P. Anandan. An integrated Bayesian approach to layer extraction from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):297-303, March 2001.

[24]. B. Rousso, S. Peleg, I. Finci, and A. Rav-Acha, Universal mosaicing using pipe projection. *Proceedings of the IEEE International Conference on Computer Vision (ICCV'98)*, Bombay, January 1998, pp. 945-952.

[25]. M. J. Black and A. D. Jepson, Estimating optical flow in segmented images using variable-order parametric models with local deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10), 1996: 972-986.

[26]. N. L. Chang and A. Zakhor, Constructing a multivalued representation for view synthesis. *International Journal of Computer Vision*, 45(2), November 2001: 157-190.

[27]. Zhu, Z., Riseman, E. M. and Hanson, A. R., 2004. Generalized Parallel-Perspective Stereo Mosaics from Airborne Videos, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 2, Feb 2004, pp 226-237.