Submission to International Journal of Computer Vision (Revised) [**VISI338-00M**]

Efficient Fourier-Based Approach for Detecting Orientations and Occlusions in Epipolar Plane Images for 3D Scene Modeling

Zhigang Zhu, Guangyou Xu and Xueyin Lin

(March 31, 2004)

Index Terms:

Motion analysis

3D reconstruction

Epipolar plane image

Energy model

Occlusion recovery

Layered representation

Panoramic representation

Efficient Fourier-Based Approach for Detecting Orientations and Occlusions in Epipolar Plane Images for 3D Scene Modeling

Zhigang Zhu
Department of Computer Science, the City College
The City University of New York, New York, NY 10031
Email: zhu@cs.ccny.cuny.edu

Guangyou Xu, Xueyin Lin
Department of Computer Science and Technology
Tsinghua University, Beijing 100084, P. R. China

Abstract

This paper presents a Fourier-based approach for automatically constructing a 3D panoramic model of a natural scene from a video sequence. The video sequences could be captured by an unstabilized camera mounted on a moving platform on a common road surface. As the input of the algorithms, "seamless" panoramic view images (PVIs) and epipolar plane images (EPIs) are generated after image stabilization if the camera is unstabilized. A novel panoramic EPI analysis method is proposed that combines the advantages of both PVIs and EPIs efficiently in three important steps: locus orientation detection in the Fourier frequency domain, motion boundary localization in the spatio-temporal domain, and occlusion/resolution recovery only at motion boundaries. The Fourier energy-based approaches in literature were usually for low-level local motion analysis and are therefore not accurate for 3D reconstruction and are also computationally expensive. Our panoramic EPI analysis approach is both accurate and efficient for 3D reconstruction. Examples of layered panoramic representations for large-scale 3D scenes from real world video sequences are given.

1. Introduction

The problem of modeling and rendering real scenes has received increasing attention in recent years in both computer vision and computer graphics communities. However, how to automatically build a 3D visual representation from image sequences for re-rendering real 3D natural scenes is still an open problem, and at the heart of it are two challenging issues: the correspondence problem between two or multiple views, and visual representations of large scale scenes with occlusions. Many of the successful image-based modeling and rendering approaches have tried to simplify or avoid the correspondence problem by using 2D image interpolation or mosaicing, polygon scene constraints or human-computer interaction. Other more general approaches need sophisticated vision algorithms for handling the correspondence problem, such as in multi-view stereo or general motion analysis of an image sequence. Building and maintaining a suitable visual representation of a large-scale scene has always been a hot topic of research and applications in image-based modeling and rendering. We will discuss some related work in dealing with these two issues for 3D scene modeling in Section 1.1.

In this paper, we will address these two issues with the application of automatically constructing a 3D panoramic model of a static natural scene from an easily obtained video sequence. We do not attempt to solve the general structure from motion problem; instead, the motion of the camera is somewhat constrained. We assume that a long and dense image sequence will be the input of our system. Ideally, the video sequence is captured by a video camera undertaking strictly 1D translation. However, our method tolerates the vibrations of video sequences captured by an unstabilized camera mounted on an ordinary vehicle, moving on a common and often bumpy road surface. In the latter case, an image stabilization pre-processing step is necessary to generate a pure translational sequence (e.g., Hansen, et al, 1994; Morimoto and Chellappa, 1997, Zhu, et al 1999). Then, a multi-perspective panoramic view image (PVI) and a set of epipolar plane images (EPIs) are extracted from the long (and rectified) image sequence. This paper will discuss an approach to recover depth for each pixel of the spatial-temporal panoramic view image (PVI) by analyzing the corresponding EPIs. Thus, the representation and correspondence problems are efficiently tackled by integrating the two spatio-temporal images: the EPI analysis eases the correspondence problem, whereas the panoramic representation makes the EPI analysis efficient for long image sequences.

1.1. Related Work in 3D Scene Modeling

Existing work in modeling a 3D scene from image sequences can be divided into four categories of approaches: 3D model-based, mosaic-based, multi-view-based and layered representations.

Model-based approach - A 3D model-based method first constructs a 3D CAD-like model of a scene, then the model is reprojected to generate new images of desired views. For modeling and rendering large-sale scenes, several important projects have been reported. Faugeras, et al (1998) address the problem of recovering a realistic textured model of a scene from a sequence of images without any prior knowledge either about the parameters of the cameras or about their motion. Correspondences between images are obtained by either corner matching or feature point tracking, and the complete set of perspective projection matrices for all camera positions is computed. Relying on information of the scene such as parallel lines or known angles, the geometry of the scene can be reconstructed up to an unknown affine transformation. Alternatively, if this information is not available, the Euclidean structure of the scene can be recovered through self-calibration techniques. The scene geometry is modeled as a set of polyhedra, and textures to be mapped on the polygons are extracted automatically from the images. A similar system has been presented by Baillard and Zisserman (1999) which takes sequence of images from an uncalibrated camera or cameras and automatically recovers camera positions and 3D point and line structure from these sequences. This method allows a piecewise planar model of a scene to be built automatically. In Coorg and Teller (1998), a large pose-mosaic dataset is generated in order to model urban views. Several thousand digital images are then grouped by spatial position into spherical mosaics, and each of them is annotated with estimates of the acquiring camera's six DOF poses. Due to the difficulty of automatically recovering realistic 3-D models from images, the authors exploit the geometric structure inherent in typical urban environments, e.g., vertical facades, and apply the space sweep algorithms proposed by Collin (1996) used in aerial image site modeling (Collins , et al, 1998). Similar scene constraints are used in Debevec, et al (1996) where the 3D model of a building is recovered interactively by using a photogrammetric modeling method based on a small number of user-supplied correspondences, followed by a model-based stereo algorithm. Non-polygonal objects such as trees are flattened to the ground plane.

Mosaic-based approach - Recently, the construction of panoramic images and high quality mosaic images from video sequences has attracted significant attention. However, many of the current successful image mosaic algorithms only generate 2D mosaics (either a 360-degree panorama or a full sphere omni-directional image) from a camera rotating around its nodal point (e.g., Chen, 1995; Xiong and Turkowski, 1997; Shum and Szeliski, 2000; Sawhney , et al, 1998）. A

plenoptic modeling approach (McMillan and Bishop, 1995) is proposed to use two cylindrical panoramas from rotating cameras in two viewpoints to estimate the disparity map. However, the emphasis of this work is the rendering from this representation rather than the modeling from an image sequence. Later, Shum , et al (1998) made an effort to construct 3D models from two cylindrical panoramic mosaics from rotating cameras interactively, using environmental constraints such as parallel lines and lines with known orientations. Creating multi-perspective stereo panoramas from one rotating camera off the nodal point was proposed by Ishiguro , et al (1990), Peleg, et al (1999, 2001), and Shum and Szeliski (1999). Shum, et al (1999) extended this idea to capture omnivergent stereo data using a rotating camera. They also showed synthetic examples of spherical stereo mosaics. Nayar & Karmarkar (2000) showed stereoscopic spherical mosaics by using catadioptric (omnidirectional) slice cameras. In these kinds of stereo mosaics, the viewpoints of new views are limited within a very small area, usually a circular area of a few meters in diameter. 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) was proposed by Zheng and Tsuji (1992; 1998). The moving slit paradigm was used as the basis of 2D manifold projection for image mosaicing (Peleg and Herman 1997, Peleg, et al, 2000), multiple-center-of-projection image representation for image-based rendering (Rademacher and Bishop 1998), and creating multi-perspective stereo mosaics from a translating camera for environmental monitoring (Zhu, et al, 2001; Zhu, et al, 2004). Multi-perspective panoramas (or mosaics) show very attractive properties in visual representation and epipolar geometry. However, 3D recovery of stereo mosaics faces the same problems as in traditional stereo - the correspondence problem and the handling of occlusion boundaries.

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 (1997, 2001) obtained depth information of some pre-specified “reference frames” of an image sequence captured by an uncalibrated camera that scans a stationary scene, then transformed the points of reference frames onto an image of the desired virtual viewpoint. However, reference frames were chosen quite arbitrarily, and 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. Szeliski (1999) presents 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.

Layered representation - In a layered representation, a set of depth surfaces is first estimated from an image sequence of a single camera and then combined to generate a new view. Wang and Adelson (1994) addressed the problem as a computation of 2D affine motion models and a set of support layers from an image sequence. Optical flow is used as the input of iterative motion clustering. The *layered representation* that they proposed 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). This representation is a good choice for image compression of a video sequence and for limited image synthesis of selected layers. However, it cannot be used to generate synthesized images of arbitrary views. Sawhney and Ayer (1996) proposed a multiple motion estimation method based on Minimum Description Length (MDL) and modified Expectation-Maximization (EM) algorithms. The algorithm is computationally expensive and requires 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, et al (1998) proposed a framework for extracting structure from stereo pairs, and the scene is represented 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 offset relative to the plane. Initialization of layers (which is a difficult task) is performed by humans, and the initial estimates of the layers are recovered using techniques from parametric motion estimation. These initial estimates are then refined using a re-synthesis algorithm which takes into account both occlusion and mixed pixels. For more complex geometry of a scene, a *layered depth image* (LDI) is proposed (Shade, et al, 1998) which is a representation of a scene from a single input camera view but with multiple pixels along each line of sight.

1.2. Panoramic Images and Epipolar Plane Image Analysis

It has been shown that under strict translation, a panoramic view image (PVI) can be generated by extracting a vertical column from each frame and piling them up to form a wide angle multi-perspective image (Zheng and Tsuji, 1992). Similarly, an epipolar plane image (EPI) (first proposed by Bolles, et al (1987)) can be generated by extracting a horizontal scan-line from each frame and piling them up to form a spatio-temporal (ST) image. In other words, an EPI image is an x-t section of a video sequence (each row comes from the same row index but from different frames). The

slope of a locus (i.e., a line induced by motion of a single point) in an EPI is proportional to its depth. Techniques based on 2D ST image formation (panoramic view images and epipolar plane images) meet the need for a compact representation and fast 3D recovery (e.g., Ishiguro, et al, 1990; Zheng and Tsuji, 1992; McMillan and Bishop, 1995; Dalmia and Trivedi, 1996; Murray, 1995; Shum and Szeliski, 1999; Li, et al, 2004; Zhu, et al, 2004). Baker & Bolles (1989) extended their earlier work that only addresses translation to general motion. However, they used locus tracking for constructing generalized epipolar planes on 3D spatio-temporal surfaces. For EPI-based depth recovery methods, locus extraction is a hard problem for image sequence of a natural scene with complex textures, particularly when the EPIs are generated from video sequences with unpredictable vibrations in camera motion. In addition, the large amount of data in EPIs of a lone video sequence often makes it prohibitive in terms of computational costs.

In order to apply these two kinds of compact representations to an easily-captured image sequence, an efficient and robust Fourier-based method is proposed to robustly detect multiple orientations of the EPI's motion texture in the frequency domain. This approach is different from the locus tracking methods (Murray, 1995; Allmen and Dyer, 1991), the 1D multibaseline matching technique (Li, et al 2004), or the local operator methods, such as Gabor filters (Adelson and Bergen, 1985; Heeger, 1987) and Steerable filters (Freeman and Adelson, 1991; Niyogi, 1995; Fleet, et al, 1998). The local operator methods only provide limited angular resolution for orientation calculation since a local motion operator is usually performed in a small ST neighborhood. This paper provides a first attempt to use large neighborhood windows (e.g. 64×64 pixels) for detecting local motion more robustly and accurately. Furthermore, motion boundaries are accurately located in the spatio-temporal domain by measuring global intensity similarities only along the detected orientations, and the occluded regions are recovered by further exploring extra information near motion boundaries in the EPI. We emphasize that only a small amount of selected data in the EPIs that correspond to the PVI representation is processed in our approach, and the processing for all the epipolar planes can be done in parallel. Thus, it is possible to implement the proposed algorithms in real time with parallel computing hardware. Even the current sequential implementation on a Pentium 400 MHz PC can achieve a frame rate of about 2 frames per second (fps) for 128×128 images. The frame rate increases to 10.8 fps on a Xeon 2.4 GHz dual-CPU Dell Linux workstation. Three-dimensional layered panoramic models have been constructed from several image sequences, some of which have more than 1000 frames. In addition, direct methods

for all the steps have been developed in which image segmentation, feature extraction, and matching are avoided.

The rest of the paper is organized as follows. In Section 2, we describe the data collection and pre-processing issues for generating panoramic view images and epipolar plane images, then we introduce our panoramic EPI analysis approach. In section 3, a new motion occlusion model is presented and analyzed in both the spatio-temporal and the frequency domains. Then, a Gaussian-windowed Fourier orientation detector (GFOD) is proposed for multiple-motion orientation detection. In Section 4, we further discuss the use of the occlusion model and the GFOD for multiple orientation estimation and methods for motion boundary localization and dense depth acquisition. In Section 5, methods for occlusion handling and perspective resolution recovery are presented. In Section 6, we discuss data selection and representation issues for efficient EPI analysis and 3D scene representation. In Section 7, we give a comparison study and show experimental results on panoramic layered representations of several image sequences for natural scene modeling. Brief conclusions are given in the last section.

2. Panoramic Epipolar Plane Analysis

2.1. Data Collection and Spatio-Temporal Image Generation

In order to construct the 3D model of a roadside scene, a camera is mounted on a vehicle moving on an approximately flat road surface. The camera's optical axis is perpendicular to the motion direction and its horizontal axis is parallel to the motion direction[1]. We assume that the motion of the vehicle (camera) consists of a smooth planar motion and an unpredictable small fluctuation due to the vehicle's motion over a rough surface. In many real cases, the smooth motion can be approximated as a constant velocity (*V)* translation. The small fluctuation between two successive frames is modeled by three small rotation angles around the three coordinate axes and three translation components along the three axes (denoted as a rotation matrix R and a translational vector T in Fig. 1(a)). An image stabilization algorithm is used as a pre-processing step to reduce or remove fluctuations so that the motion after stabilization is a translation motion with constant velocity *V*. Image stabilization is not the focus of this paper, but we want to show the conditions in which our algorithms work. Therefore, after we formally define the spatio-temporal (ST) images we will provide some experimental results in generating two kinds of ST images - panoramic and

epipolar plane images - from real-world video. Interested readers can refer to our previous papers (Zhu, et al, 1998; Zhu, et al 1999; Zhu 2001) for details of the image stabilization algorithms.

Fig. 1

Without loss of generality, the effective focal length f of the camera is assumed to be fixed for a rectified (stabilized) image sequence. The image sequence obeys the following spatio-temporal (ST) perspective projection model

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

where (X,Y,Z) represent the 3D coordinate at time t=0. A feature point (x,y) forms a straight locus and its depth is

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

where $v = dx/dt$ is the slope of the straight locus. In order words, two kinds of useful 2D ST images can be extracted (Fig. 1(b)). One is the Panoramic View Image (PVI), which possesses most of the 2D information of a roadside scene. The other is the Epipolar Plane Image (EPI), whose ST texture orientations represent depths of scene points. Fig 2(a) shows two PVIs (x=0 and x=-56) that are extracted from a 1024-frame BUILDING sequence of 128×128 images. These two PVIs are parallel-perspective images with multiple viewpoints in the t axis, and depth information can be derived from this ST stereo pair. A better approach is to make use of the continuous information in the epipolar plane images. Fig 2(b) shows an EPI (x = 9) between these two PVIs. In addition to obtaining depths from locus orientations, occluded (and side) regions will also be recovered by the method proposed in this paper.

Fig. 2

Here, we show two examples to see what are the inputs of our method. Fig. 3 shows the stabilization results of the BUILDING sequence when small fluctuations occurred. Better PVIs and EPIs are obtained after image stabilization, which means better depth estimation. Fig. 4 compares PVIs and EPIs with and without image stabilization for a TREE sequence taken by a camera mounted on a hand-pushed tricycle on a bumpy road. The sequence consists of 1024 frames of 128×128 images. It is obvious that stabilization plays a vital role in the construction of good panoramic and epipolar plane images when the camera's vibrations are large as in this example. The vibrations include both x/y translation components and small in-plane rotation when the vehicle

bumps up and down (y translation – see Fig. 4(a)), waves left and right (x translation, see Fig. 4(b)), and tilts up and down (in-plane rotation), since the camera's direction is perpendicular to the motion direction. The depth map (Fig. 4(c)) can be obtained through our epipolar plane image analysis on the stabilized EPIs. This is almost impossible without image stabilization (see Fig. 4(b)).

In summary, in order to use any EPI-based methods the input image sequences need to be constrained under 1D translation. This can be achieved either by a precise control of the camera motion or by a software process of video stabilization. We note here that our method is robust enough to deal with EPIs generated in the latter case from difficult image sequences as in Fig. 4.

Fig. 3

Fig. 4

2.2. Panoramic Epipolar Plane Analysis

Spatio-temporal panoramic view images(PVIs) provide a compact representation for large-scale scene. Stereo PVIs can be used to estimate the depth information of the scene. The difference between panoramic stereo and the traditional stereo is that panoramic images are parallel-perspective projections. The depth of a point is proportional to the "displacement" in the t direction in a pair of stereo PVIs (Fig. 2 (a); in Eq. (2) "disparity" dx is fixed and $D \propto dt$), which means that depth resolutions are the same for different depths (Zhu, et al, 2001). However, there are some disadvantages when we use stereo PVIs to recover the depth of a scene. First, stereo PVI approach faces the same correspondence problem as in any traditional stereo methods. Second, occluding regions in two panoramic views cannot be easily handled due to the lack of information. The solution to these two problems is to effectively use the information in between, i.e. that of the epipolar plane images. Our panoramic epipolar plane analysis approach consists of three important modules:

- Module 1: *frequency domain orientation detection* by using large neighborhood windows (e.g. 64×64) for detecting local motion more robustly and accurately in the EPIs (Section 3);
- Module 2: *spatio-temporal domain motion boundary localization* by measuring global intensity similarities only along the detected orientations of the loci (Section 4); and
- Module 3: *occlusion and resolution recovery* by further exploring extra information near motion boundaries in the EPIs (Section 5).

Our approach has the following three advantages. (1) It is *robust* since with a spatio-temporal-frequency domain analysis, feature detection, hard thresholding and locus tracking are avoided in our algorithm. (2) It is *efficient* in that it only processes a small fraction of the necessary data instead of the entire 3D ST images. (3). It is *accurate* in both depth estimation and occlusion boundary localization since we use a large ST window for orientation estimation and apply depth boundary localization. In the following sections, after we derive the motion occlusion model in spatio-temporal and frequency domains, we will describe each of the three modules in details, with discussions for the three advantages in the proposed algorithms.

3. Motion and Occlusion Modeling and Detection

The first order motion texture model of an EPI can be expressed in the spatio-temporal domain as (Allmen and Dyer, 1991; Adelson and Bergen, 1985; Heeger, 1987)

$$g(x,t) = f(x - vt) \tag{3}$$

where $f(x)$ is the image of a single scan line at time t =0. By Fourier transform, the model in the frequency domain can be derived as

$$G(\xi, \omega) = F(\xi)\, \delta(v\xi + \omega) \tag{4}$$

Eq. (4) indicates that object points with the same depth values and the same constant translation occupy a single straight line passing through the origin in the frequency domain, i.e.,

$$v\xi + \omega = 0 .$$

It is well known that orientation can be easier to detect in the frequency domain than in the spatio-temporal domain when a single orientation is presented in the window of the processing (Jahne, 1991). In this paper, we will deal with multiple orientations due to depth changes. Consequently we will study two important issues: motion occlusion modeling, and accurate and robust orientation estimation.

Fig. 5

3.1. Motion Occlusion Model

We model the motion occlusion in an *x-t* image (EPI) (following Wang and Adelson, 1994) as

$$g(x,t) = m_s(x,t)g_1(x,t) + (1 - m_s(x,t))g_2(x,t) \tag{5}$$

where the first layer $g_1(x,t)$ is occluded by the second layer $g_2(x,t)$, and $m_s(x,t)$ is a occluding mask (Fig. 5(a)). Under a 1st order translation, the ith layer with velocity v_i can be expressed as

$$g_i(x,t) = f_i(x - v_i t), (i = 1,2)$$

where $v_1 < v_2$. The occluding mask is a step function moving with velocity v_2, i.e.,

$$m_s(x,t) = u(x - v_2 t) \tag{6}$$

and the value of the function is 0 or 1. Hence the *1st order motion occlusion model* can be written as

$$g(x,t) = u(x - v_2 t) f_1(x - v_1 t) + (1 - u(x - v_2 t)) f_2(x - v_2 t) \tag{7}$$

It should be noted here that many other approaches for multiple motion analysis look only at the summation of the single motion model but do not take real occlusion into account. Now we want to further look at this occlusion model in the frequency domain. The Fourier transform of the model can be derived as (Appendix 1)

$$G(\xi,\omega) = \frac{1}{v_1 - v_2} F_1(\frac{v_2\xi + \omega}{v_2 - v_1}) U(\frac{v_1\xi + \omega}{v_1 - v_2}) + F_{u2}(\xi)\delta(v_2\xi + \omega) \tag{8}$$

where $F_1(\xi)$ is the Fourier transform of $f_1(x)$, $F_{u2}(\xi) = F_2(\xi) - F_2(\xi) * U(\xi)$ is the Fourier transform of $f_2(x)(1-u(x))$, the visible parts of $f(x)$, and $U(\xi)$ is the Fourier transform of $u(x)$. Without loss of generality, we assume the step function is

$$u(x) = \begin{cases} 0, x < 0 \\ 1, x \geq 0 \end{cases}, \tag{9}$$

so we have

$$U(\xi) = \frac{1}{j\xi} + \pi\delta(\xi) \tag{10}$$

which implies that the peak value of $U(\xi)$ is at $\xi = 0$ (Fig. 5(c)). From Eq. (8) and (10) we can obtain the following important conclusion (Fig. 5(b)):

Most of the energy spectra of a spatio-temporal texture at a motion boundary of two depth layers lie in two lines in the frequency domain that correspond to the two depth layers.

Eq. (8) is not as obvious as Eq. (4), therefore we will give a little bit more explanation. The Fourier transforms along the first line $\xi = -\omega / v_1$ *is*

$$G(\xi, -v_1\xi) = \frac{1}{v_1 - v_2} F_1(\xi) U(0) \tag{11}$$

which displays a peak corresponding to the occluding layer. The Fourier transform along the second line $\xi = -\omega / v_2$ *is*

$$G(\xi, -v_2\xi) = F_{u2}(\xi) + \frac{1}{v_1 - v_2} F_1(0) U(\xi) \tag{12}$$

which shows a peak corresponding to the occluded layer, with an addition that only has an obvious effect when $\xi = 0$. This conclusion indicates that we can easily detect two layers with occlusion in the frequency domain. In the following we will see how we can use this property in real applications.

3.2. Gaussian-Fourier Orientation Detector (GFOD)

In order to detect multiple orientations more precisely and robustly, we need the Fourier transform performed in a large ST window on an EPI. The angular resolution of orientation estimation is proportional to the size of the window *m*. For example, the angular resolution is four times better when the window size *m*×*m* is 64×64 pixels than when it is 16×16. However, the side effect of the large window is that all the oriented textures in the large window will contribute to the energy spectrum. So for a multiple orientation pattern, multiple peaks could be detected when the window slides thorough a quite wide region near the depth boundary. Therefore, the question is how to accurately localize the depth/motion boundaries. In this paper, a *Gaussian-Fourier Orientation Detector* (GFOD) is designed in order to keep the precision for both orientations of motion textures and localization of motion boundaries.

A spatio-temporal Gaussian window is defined as

$$w(x,t) = \exp(\frac{x^2+t^2}{2\sigma^2}) \tag{13}$$

where $\sigma^2 = \frac{m-1}{4}$. Applying this Gaussian mask to the motion texture image, the Gaussian windowed result of the motion texture can be represented as

$$g_w(x,t) = f(x - vt)w(x,t) \tag{14}$$

Again, we want to look at this model in the frequency domain. Its Fourier transform, which is derived in the same way as for Eq. (8), is

$$G_w(\xi,\omega) = c(v)F_w(\frac{\xi - v\omega}{v^2+1})W_v(v\xi+\omega) \tag{15}$$

where

$$\begin{aligned} &c(v) = \frac{2v}{v^2+1}, \\ &F_w(\omega) \Leftrightarrow f_w(x) = f(x)e^{-\frac{x^2}{2(\sigma\sqrt{v^2+1})^2}} \\ &W_v(\omega) = e^{-2(\sigma/\sqrt{v^2+1})^2\omega^2} \Leftrightarrow w_v(t) = e^{-\frac{t^2}{2(\sigma/\sqrt{v^2+1})^2}}. \end{aligned} \tag{16}$$

The Gaussian windowing still preserves the important frequency property of the motion texture, i.e., most of the energy still lies in the line $\xi = -\omega / v$, which is

$$G_w(\xi,-v\xi) = F_w(\xi)W_v(0) \tag{17}$$

since $W_v(\omega)$ is a Gaussian function with peak at $\omega = 0$.

The derivation of the energy model for the cases of multiple orientation and motion occlusions is a combination of Eq. (8) and Eq. (15). The Gaussian-windowed spatio-temporal image with occlusion and its frequency spectrum can be represented as

$$\begin{aligned} &g_w(x,t) = g(x,t)w(x,t) \\ &G_w(\xi,\omega) = G(\xi,\omega) \otimes W(\xi,\omega) \end{aligned} \tag{18}$$

where $\otimes$ denotes convolution operation. Here we only give a qualitative analysis. From the principle of the Fourier transform, the multiplication of a Gaussian window $w(x,t)$ with variance σ^2 in the ST domain is equivalent to the convolution of a Gaussian function $W(\xi,\omega)$ with variance inversely proportional to σ^2 in the frequency domain, which will smooth the energy distribution. For this reason, the GFOD has the following two advantages. First, it is insensitive to noise. Since the Gaussian window acts like a smoothing operator in the frequency domain, the Fourier spectrum is smoother than that without Gaussian weighting. The smaller the variance σ^2 (i.e. the narrower the Gaussian window), the more smoothing to the energy spectrum[2]. Second, it favors the motion texture closer to the center of the window. By applying the Gaussian window in the ST domain, the ST patterns that are farther from the center of the window have less contribution to the final energy spectrum, but they are not eliminated. So the design of the GFOD operator tries to reach a balance

between the orientation resolution (over a large window) and the localization accuracy of depth boundaries (in the center of the window). We will give real examples in Section 4.4 to show the effectiveness of large Gaussian windows versus small rectangular windows in motion orientation and motion boundary detection.

It should be noted here that the proposed method could be used in other application domains where multiple oriented texture patterns need to be analyzed.

4. Multiple Orientations, Motion Boundaries, and Dense Depths

4.1. Multiple Orientation Detection

First, we show how to use the Gaussian-Fourier Orientation Detector (GFOD) for multiple orientation detection in an EPI. The GOFD is applied only along a scan-line ($x=x_0$) in the EPI, which is the intersection line of this EPI and the PVI extracted at x_0 in the xyt cube (Fig. 1(b). 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. The Fourier transform $G_w(\xi,\omega)$ is obtained for a 64×64 Gaussian-windowed EPI pattern centered at (x_0, t) for any t coordinate along the t axis. The "energy spectrum" $P(\xi,\omega)=\log(1+G_w^2(\xi,\omega))$ 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

$$P_d(\phi)=\int_{r_1}^{r_2} P(r,\phi)dr \quad \phi\in[0,\pi] \tag{19}$$

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 spatio-temporal resolution of the image. Initially, r_1 and r_2 are set to 8 and 30, respectively, for a 64×64 window.

As noted in Chang and Zakhor (2001), because of visibility limitations, real-world scenes typically do not have more than three occlusion levels. We assume that at a depth boundary there are only two depth levels when observed locally. Therefore our method picks up one or two peaks in the 1D orientation histogram (Eq. (19). The highest peak is always selected. The second highest peak is selected if it exists and is significantly high, for example, more than half of the highest in

our implementation. Fig. 6 shows the peak selections for a real EPI (please refer to Fig. 8 for more details in close-up displays). An *orientation energy distribution map* $P_d(\phi,t)$ can be constructed which visually represents the depths of the points along the time (t) axis in the EPI. The two peaks are shown on the distribution map. However the highest does not necessarily correspond to the correct orientation. This will be fixed in the following motion boundary localization.

Fig. 6

4.2. Motion Boundary Localization

Multiple orientations will be detected for a certain temporal range when the GFOD operator moves across a depth boundary. 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. Therefore a *Motion Boundary Localizer (MBL)* is designed to determine whether or not the depth/motion boundary is precisely 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). The intuition behind the MBL algorithm is that statistically intensities of a scene point seen in an image sequence will remain similar (if not the same) over time (here the time period is 64 frames). Therefore we check the pixel similarity along all possible directions estimated by the GFOD under the window to determine the correct orientation. 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 of the locus. For introducing the basic idea, a single scale similarity measurement is derived first. Then, we extend the principle to multiple scale measurements. Note that only a *comparison-and-selection* operation is used, without assuming any detection of feature points or using any troublesome thresholds.

Fig. 7

Consider the case in which two orientations θ_1 and θ_2 ($\theta_1 > \theta_2$) are detected within a Gaussian window. Dissimilarity 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. 7(a) and (b))

$$C(\theta_k + i\pi, R) = \frac{1}{R}\sum_{r=1}^{R} I^2(r, \theta_k + i\pi) - \bar{I}^2(\theta_k + i\pi, R) \quad \text{(k=1,2)} \tag{20}$$

where $r = \sqrt{(x - x_0)^2 + (t - t_0)^2}$, $\bar{I}(\theta_k + i\pi, R) = \frac{1}{R}\sum_{r=1}^{R} I(r, \theta_k + i\pi)$ and $\theta + i\pi$ (i=0,1) indicates the measurements in two opposite radial directions (θ_k and $\theta_k + \pi$) along the detected orientation θ_k (k=1, 2). 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 of the radial directions (θ_1 and $\theta_1 + \pi$), but the occluded (i.e., farther) object can only be seen in one of the two directions (θ_2 or $\theta_2 + \pi$, Fig. 7(a)). Thus, the dissimilarity measurements for closer and farther objects are defined as

$$\begin{aligned} E(\theta_1, R) &= \frac{1}{2}(C(\theta_1, R) + C(\theta_1 + \pi, R))/P_d(\theta_1) \\ E(\theta_2, R) &= \min(C(\theta_2, R), C(\theta_2 + \pi, R))/P_d(\theta_2) \end{aligned} \tag{21}$$

respectively. Notice the difference in the two measurements: the measurement for the closer object is the average of those in both radial directions, but the occluded (i.e. farther) object only takes the smaller measurement of those in the two radial directions. In addition, we give more weights to stronger oriented texture patterns: $P_d(\theta_k)$ is the value of the orientation histogram (Eq. (19)) at θ_k (k=1,2). The higher the value is, the lower the dissimilarity measurement will be.

Fig. 7(a) shows how to use these measurements to localize a depth boundary when the farther object is occluded by the closer object (which is defined as the *occlusion* case). Two 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. $E(\theta_1, R)$) along the locus direction of the occluding object will be larger, since the measurement is performed across the loci pattern of the to-be-occluded object (left of Fig. 7(a)). On the other hand, the dissimilarity measurement (i.e., $E(\theta_1, R)$) along the locus direction of the to-be-occluded object will be much smaller, since the measurement is right along the locus of the to-be-occluded object. As 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 occlusion boundary of the occluding (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. 7(a)). 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. 7(a)). Similar arguments hold for the *reappearance* case, when the occluded object reappears behind the occluding (closer) object (Fig. 7(b)). Therefore a simple verification criterion can be expressed as

$$\theta = \begin{cases} \theta_1, \ if \ E(\theta_1, R) \le E(\theta_2, R) \\ \theta_2, \text{Otherwise} \end{cases} \tag{22}$$

In fact, the condition of occlusion and reappearance can be judged either by comparing $C(\theta_2, R)$ and $C(\theta_2 + \pi, R)$ (see Fig. 7) 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. 7(a)) we have $C(\theta_2 + \pi, R) < C(\theta_2, R)$, and in reappearance (near to far, Fig. 7(b)) we have $C(\theta_2, R) < C(\theta_2 + \pi, R)$.

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. 7(c)). By defining the following ratio

$$D_i = \frac{\max\left(E(\theta_1, R_i), E(\theta_2, R_i)\right)}{\min\left(E(\theta_1, R_i), E(\theta_2, R_i)\right)} \tag{23}$$

a scale p (p=1,2 or 3) with maximum D_p is selected for comparing the intensity similarities. For example, in Fig. 7(c), R_2 will be selected.

The real example of motion boundary localization is shown in Fig. 6. The "correct" peaks are marked as the long dashed curves (red in color version) in the orientation map, while the second peaks are shown in solid small pieces (blue in color version). They are detected by the GFOD but are discarded by the motion boundary localizer.

4.3. Orientation Refinement and Depth Interpolation

The selected orientation angle θ can be refined by searching for a minimum dissimilarity measurement for a small-angle range around θ. The calculation is very simple since for each angle in this small range we only need to calculate a variance value along a 1D line ((as in Eq. (21)), then pick up the angle with the minimum variance value. The accuracy of the orientation angle, especially that of a far object, can be improved by using more frames. The frame number can be decided by examining the occluding relations near the far object.

In order to obtain a dense depth map, interpolations are applied to textureless or weak-textured regions /points where no orientation can be detected (see Section 6 for data selection). The proposed interpolation method (Fig. 7(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 for a real EPI (after applying the GFOD, the motion boundary localizer and depth interpolation) are shown in the last row of Fig. 6 by the histogram of orientation angles. Note that the accuracy of the depth boundaries at locations marked from (1) to (5).

4.4. Experimental Analysis

In this subsection, we will discuss the impact of three aspects of our approach with real data: large window, Gaussian weighting, and motion boundary localization.

Gaussian versus rectangular windows with the same size - Fig. 8 compares experimental results for the BUILDING sequence using a rectangular window and a Gaussian window ($\sigma^2 = 2\frac{m-1}{4}$). The size of the windows in both cases is 64×64 pixels. Simply using a rectangular window has ambiguity in localizing depth boundaries. For example, when the rectangular window is used (see columns (a) to (d) in Fig. 8 (1)), two peaks can be detected within a large neighborhood (27 frames - from frame 296 to frame 322) of a depth boundary (at frame 131). By using the Gaussian window, motion boundaries can be located in a much smaller range. In columns (e) to (h) of Fig. 8(1), two peaks are detected only in 8 frames from frame 305 to frame 313 and without obviously degrading the angular resolution of orientation. This is because the magnitudes of ST texture off the center are reduced but are not eliminated by using a Gaussian window. Note that multiple orientations are still detected within a region of the motion boundary even if Gaussian windows are used. Therefore, motion boundaries are further localized by using the proposed motion boundary localizer, which results in the final dense histogram of the orientation angles in Fig. 6. The motion boundary localizer could also solve the ambiguity problem when the rectangular window is used in this example. In both cases, the solid dark lines show the correct orientation, while the dashed dark lines show the second orientations that are discarded by the motion boundary localizer.

So the question is, since the spatial-temporal domain motion boundary localizer is used anyway, why do we use the more expensive Gaussian windowing? Fig. 8(2) shows that the detection ambiguity by the rectangle windowed Fourier method could not even be correctly fixed by using the motion boundary localizer. This set of images is taken from the EPI of a region with gradually changing depths (side façade of a building - see Fig. 2 and Fig. 6). The reason for this kind of problem is that the stronger oriented texture off the center has different orientations from that of the weaker motion texture at the center. In frame 661, none of the two detected orientations is that of the locus in the center; rather, they are the orientations of the loci of the left and right stronger textures. In frames 613, 661 and 670, the correct orientation is among the two detected peaks, but the orientation selection by the motion boundary localizer is not correct since it is almost textureless along one of the orientations. However, if the Gaussian windowed Fourier orientation detector is used, the motion boundary localizer helps in most of the cases. In Fig. 8(2), the correct orientations can be found in three frames (613, 661 and 670) of the four frames listed here, except for the frame 661. Note that the correct orientation of the weak texture in frame 661 is detected by the GFOD, so the remaining issue is how to correct pick it up.

Fig. 8

Large Gaussian versus small rectangular windows – From the above discussion, it is obvious that large rectangular window won't work well. Now the natural question is: how about using a smaller rectangular window instead of a large Gaussian window for the sake of motion boundary localization as well as computational efficiency? Fig. 9 shows the depth estimation result using a 16x16 rectangular window and without using the motion boundary localizer for the same EPI shown in Fig. 6. Using a smaller window is computationally more efficient; however, compared to Fig 6, the resolution of the orientation energy spectrum is much lower due to the smaller x-t window used. Therefore, the depth resolution is lower when we use the smaller x-t window. The motion boundary localizer won't help in this case since it will bring more noise to the depth estimation and since the depth estimates are noisy and not accurate in the first place.

Fig. 9

Depth w/o motion boundary localizer – We have shown in Fig. 8 that the motion boundary localizer is required even when the GFOD is used. As a comparison, Fig. 10. shows the result of depth estimates *without* using the motion boundary localizer for the same EPI in Fig. 6. In this example, whenever multiple peaks occur, the highest peak in the orientation histogram is selected to determine the orientation of the locus in the center of the window. By comparing Fig. 10 and Fig. 6, the output of the motion boundary localizer is two-sided. In many cases, it improves the accuracy of the motion boundary locations, such as location (2) marked in both figures. But in some other cases,

the motion boundary localizer brings in errors in orientation selection, such as location (5) where the gradual depth change of the side face is actually detected without the motion boundary localization. As a useful observation, the motion boundary localizer can provide accurate depth boundaries in many cases, but might also bring in inaccurate estimates due to noisy data and/or weak texture.

A systematic comparison of depth maps under various cases will be shown in Section 7.

Fig. 10

5. Occlusion and Resolution

5.1. Occlusion Recovery

Because a panoramic view image only reserves information from a single viewing direction, for example, the direction perpendicular to motion direction when the PVI (y, t) is selected at $x_0 = 0$ (Fig. 1(b)), some parts of the scene that are visible in other parts of the images of an original (or a stabilized) video sequence are lost due to occlusions. 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 depth boundary – Since we use PVI representation with index (y,t) and at x_0, a point on a depth boundary (y_0, t_0) in the PVI corresponds to a depth boundary point $p_0(x_0, t_0)$ in the corresponding EPI. The depth boundary point p_0 and the two associated orientation angles (θ_2 and θ_1) of the occluded (far) and occluding (near) objects are all encoded in the panoramic depth map at location (y_0, t_0). A point is considered as a depth boundary point when depth discontinuity occurs, for example, when $|\theta_1 - \theta_2| > 2^\circ$.

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. Points on this segment are projected from the same parallel viewing directions of a moving viewpoint since they have the same x coordinate. The segment is determined by the slopes of the two orientation patterns that have generated the depth boundary, and it is denoted by an x coordinate and start/end times (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. 12(b) 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.

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. For simplicity, the missing parts are classified into two types in our current implementation: OCCLUDED and SIDE. This judgment of classification can be made by calculating and comparing the dissimilarity measurements within the triangular regions for the two cases, E_o for OCCLUDED and E_s for SIDE:

$$E_o = \frac{1}{t_e - t_s}\sum_{t=t_s}^{t_e} C(\theta_2 + i\pi, R_t)$$
$$E_s = \frac{1}{t_e - t_s}\sum_{t=t_s}^{t_e} C(\theta_t + i\pi, R_t), \quad \theta_t = \theta_1 + \frac{\theta_2 - \theta_1}{t_e - t_s}(t - t_s) \tag{24}$$

where the computing of E_o is centered at point p along $p_Sp_e, R_t = |pq|$, and E_s is centered at $p_0, R_t = |p_0p|$. The radial direction for measuring the dissimilarity is determined by i (=0 or 1): for reappearance (near to far, Fig. 11 (a) and (b)), $i=0$, and for occlusion (far to near, Fig. 12(a) and (b)) $i=1$. The region is classified as OCCLUDED if we have $E_o < E_s$ within the triangle $p_0p_sp_e$. Otherwise it is classified as a SIDE region. The angle θ of an OCCLUDED region will be the same orientation θ_2 as the occluded object, whereas angle θ_t of a SIDE region will have gradually changing orientations from θ_1 to θ_2 (or from θ_2 to θ_1), as expressed in Eq. (24).

Fig. 11

Fig. 12

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 images. Fig 12(a) shows an example of recovering an occluded region (building façade) behind a tree, as indicated by the first circle in Fig. 12(c). 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. 12(c), which is separated into two pieces by a tree in front of

it. The third side region corresponds to the second side façade indicated by the third circle in Fig. 12(c). 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.

5.2. Resolution Enhancement

If an object shifts more than one pixel between two successive frames, i.e., its image velocity v is greater than 1, the spatio-temporal panoramic view image (PVI) only preserves one pixel out of the total number of the shifted pixels. The rest of the pixels are not encoded in the PVI. Fortunately, those pixels are preserved in the x directions of the xyt cube when the motion is along the x axis. Therefore the image resolution can be recovered by further EPI analysis. When the image velocity v of a point in the panorama is greater than 1, i.e. the orientation angle of it $\theta = arctan(v)$ is greater than $\pi/4$, then a v-pixel segment in x direction is extracted from the EPI instead of just using the single pixel in the PVI. Fig. 11(c) shows the idea. A noticeable feature is that the end point $p_x(x,t)$ of a segment p_0p_x at time t will exactly connect with the start point $p_1(0,t+1)$ of the segment at time $t+1$. This property is used to generate seamless, adaptive-time panoramas. In Fig. 12(a) and (b), the thickness of the central black horizontal line indicates the number of points to be extracted in the x direction of this epipolar plane image. Fig 12 (c) shows a seamless panoramic mosaic after resolution enhancement, where the width of each vertical strip from the corresponding original frame is determined by the dominant image velocity v along the y-axis in the corresponding PVI. The algorithms for occlusion recovery and resolution enhancement are two key modules that enable the creation of our layered, adaptive resolution and multi-perspective panoramic (LAMP) representation presented in Zhu and Hanson (2001).

6. Data Selection and Fast Algorithm

This section discuss our data selection and representation approach for efficient computation of the panoramic depth map. A fast GLOD algorithm is also designed to speed up the computation.

6.1. Data Selection and Representation

Suppose that a video sequence has F frames, each of size $W \times H$, and the size of the Gaussian window is $m \times m$. Instead of integrating F depth maps, each of them of size $W \times H$, the panoramic depth map corresponds to a single spatio-temporal panoramic view image (PVI). The H×F depth map is acquired by the independent and parallel processing of H images of 2D panoramic epipolar

planes. After the belief map for depth measurement (Fig. 13) is calculated from the panorama, depth information for each scan line of the PVI is obtained by executing the algorithms of multiple orientation detection, motion boundary localization and depth interpolation in the corresponding epipolar planes as described in Section 4.

Basically, our panoramic epipolar plane analysis method processes only the EPI data around a panoramic view image (e.g., the centered horizontal line in the EPI of Fig. 6). A small amount of additional data is processed only for the motion boundaries (see Fig. 12). Moreover, the algorithms also deal with the following two problems: the aperture problems of horizontal edges that run along the motion direction, and depth interpolation in textureless regions. Depth estimates at vertical edge points are more robust. To take this observation into account, a belief map corresponding to a PVI $I_{PVI}(y,t)$ is calculated as

$$B(y,t) = \frac{\partial I_{PVI}(y,t)}{\partial t} - \frac{\partial I_{PVI}(y,t)}{\partial y} \tag{25}$$

Fig. 13 shows the belief map corresponding to a PVI. The brighter intensity in the belief map shows stronger belief.

Fig. 13

The basic data selection is as follows. For the epipolar plane image $I_{EPI}(x,t)$ corresponding to a y coordinate of a given PVI, orientations are detected only at the x coordinate from which the panorama has been taken (typically $x_0 = 0$). The GFOD is applied only to each location (x_0,t_I) where the belief value $B(y,t_I)$ is greater than a given threshold; typically it is a very small value (e.g., 2). Those points with belief values lower than the threshold are interpolated (see Section 4.3). Single or multiple orientation angles $\theta_k (k = 1,\cdots,K)$ are determined by detecting peaks in an orientation histogram. Image velocity can be calculated for each orientation as $v_k = \tan\theta_k$. A motion boundary will appear within the Gaussian window if the orientation number K is greater than 1 (K=2 for double peaks). The additional data selection in motion boundary localization, depth interpolation, occlusion recovery and resolution enhancement have been discussed in the Section 5.

6.2. Fast GFOD Algorithm

The implementation of the GFOD is based on a 1D Fast Fourier Transform (FFT) algorithm. It is performed in an m×m moving window along a 1D scanline (e.g. x_0 = 0), so the 1D Fourier transforms are performed first in the column direction (i.e. the x axis) to obtain $G(\xi,t)$ then in the

row direction (i.e. the t axis) to obtain the final $G(\xi,\omega)$. Generally speaking, a moving window technique can be designed to make use of the overlapping successive windows along the time axis in order to save computation time. However, the multiplication of a Gaussian window to the spatio-temporal epipolar plane image will increase the complexity of reusing the results in the previous window locations. Therefore, we have designed a fast GFOD algorithm that uses the temporal coherence and meanwhile allows adaptive moving intervals of the Gaussian window along the time axis, depending on the belief map.

Fig. 14

A 2D Gaussian function can be separated as the product of two 1D Gaussian function as

$$w(x,t) = w_1(x)w_1(t) \tag{26}$$

then we have (Appendix 2)

$$G_{t_2}(\xi,t) = G_{t_1}(\xi,t+\Delta T)\frac{w_1(t)}{w_1(t+\Delta T)} \tag{27}$$

where $G_{t_1}(\xi,t)$ and $G_{t_2}(\xi,t)$ are the 1D Fourier transforms of the function $g_w(x,t)$ (in Eq. (18)) along the x axis when the Gaussian window is at time t_1 and time $t_2 = t_1 + \Delta T$, respectively. This indicates that the 1D Gaussian-Fourier transform of the column t in the window centered at t_2 can use the 1D Gaussian-Fourier transform of the column t+ΔT in the window of time t_1, providing $t + \Delta T \le \frac{m}{2}$, where $t \in \left[-\frac{m}{2},\frac{m}{2}\right]$ and m is the size of the window (Fig. 14). For such a column, the computation complexity is reduced to $O(m)$, compared with $O(m\log_2 m)$ if 1D FFT is directly applied. If there is no overlap between the current and the previous windows (i.e., $t+\Delta T > \frac{m}{2}$), then we need to initialize the calculation of the current window. When ΔT is smaller (i.e. the ST texture is denser hence the orientation estimation is denser and better) the speedup in computation is more obvious. In the extreme case when ΔT=1, for a W(row)×F(column) x-t image, the multiplication and the addition of a 2D FFT (using 1D FFT directly) is $O(2Fm^2\log_2 m)$. Using the proposed fast GFOD algorithm, the computation complexity reduces to $O(Fm^2(\log_2 m+1))$, which means an increase of speed by nearly a factor of 2. The total computation for an H(height)×W(width)× F(frames) xyt cube is $O(HFm^2(\log_2 m+1))$ which is independent to the width of the original video frames.

With the fast GFOD algorithm and the data selection approach, the current sequential implementation on a Pentium 400 MHz PC can achieve a frame rate of about 2 frame per second (fps) for an image sequence with 128×128 images. The frame rate increases to 10.8 fps on a Xeon 2.4 GHz dual-CPU Dell Linux workstation. The processing for all the epipolar planes can be done in parallel. Thus it is possible to implement the proposed algorithms in real time with a parallel processing hardware system.

Fig. 15

6.3. More on Data Selection and Representation

We show here 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 pixels). Fig. 15 shows two frames of this 115-frame sequence. A PVI and an EPI is shown in Fig. 16(a) and Fig. 16(b). 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. 16(c)).

Fig. 16

7. Experimental Results

We will provide two examples of complete 3D reconstruction, occlusion recovery and resolution recovery from long video sequences. First we will use the first example to show the effectiveness of large Gaussian windows in orientation estimation and the motion boundary localizer in obtaining panoramic depth maps.

7.1. Comparison: Window Sizes, Gaussian and Motion Boundary Localization

A systematic comparison of depth estimation with various parameter selections will be useful to understand the importance of large Gaussian windowing in accurate and robust depth estimation. The selections of the parameters are

G - rectangular windowing (G = 0) or Gaussian windowing (G = 1);

B – with motion Boundary localization (B = 1) or without boundary localization (B = 0); and

M - different window sizes (M = 16, 32, or 64 pixels)

The qualitative rating of the depth maps under all of the 12 possible parameter combinations are shown in Table 1. In the table rank "1" is the best and rank "6" is the worst. The large GFOD with the motion boundary localizer yields the best result in terms of depth resolution, accuracy in depth boundaries and robustness. Some typical results (raw depth maps) for the BUILDING sequence are shown in Fig. 17. As a summary, we have the following observations:

(1) ***Large Gaussian windowing*** is important in the accuracy of both locus orientation estimation and depth boundary localization. Rectangular windowing does not work whether the window sizes are small or large. Large rectangular windowing cannot detect depths with fine structures (Fig. 17(c)), while small rectangular windowing does not provide sufficient accuracy in locus orientation (Fig. 17(a)).

(2) ***Motion boundary localization*** in the spatial domain could be helpful in obtaining more accurate depth boundaries with well-presented texture patterns, but could also bring in errors with weak and complex texture patterns. Let us compare Fig. 17(d) and Fig. 17(e). Both of them use a 64×64 pixel Gaussian window, but Fig. 17(d) is obtained without depth boundary localization. The motion boundary localizer yields better result in most part of the depth map.

Table 1

Fig. 17

7.2. 3D Panoramic Layered Representation Results

In this subsection, we present results of dense depth estimation, occlusion recovery and resolution enhancement for two real examples. To reduce the noise of the depth map, a simple two-step algorithm is used:

(1). Median filtering on the depth map preserves each depth boundary while eliminating errors due to aperture problems and complex non-rigid motion of trees, etc.

(2). Intensity boundaries and depth boundaries are labeled in vertical directions. If there is no intensity boundary at a depth boundary, then the depth boundary is moved to the location of the most suitable intensity boundary.

Fig. 17(f) shows the filtering result for the BUILDING sequence. Depth boundaries of the depth map, superimposed on the panoramic intensity image as red lines in Fig. 17(g), show the accuracy

of localization. We note here that better results of occlusion recovery and resolution enhancement can be obtained when operated on the filtered, panoramic depth map.

Fig. 18

Starting from a panoramic depth map, resolutions of nearer objects are enhanced and the occluded and side regions that are not visible in the basic panoramic view image are recovered. Fig. 18 shows the results of constructing a LAMP representation (Zhu and Hanson, 2001) based on the occlusion and resolution recovery results for the BUILDING sequence. With resolution enhancement, each index (y,t) in the PVI could have more than one pixel. Fig. 18(a) shows the internal data of the resolution-enhanced PVI (without occluded regions), where all the pixels are shown sequentially in a 2D image. With occlusion recovery, additional pixels are obtained at depth boundaries. Fig. 18(b) shows both the occluded region and the resolution-enhanced pixels, in a similar way as in Fig. 18(a). The occluded portion of the facade by the tree and the side façade of the building are partially recovered.

Fig. 19 shows the results of 3D construction of the FG sequence. In the depth map, the tree stands out distinctly from the background, and the gradual changes of depths of the flower-covered foreground are detected. Fig. 20 shows the two layers of the layered representation for the FG sequence, each of which has both texture and depth maps. Note that the background layer is an extended panoramic image (XPI) representation of *xy-ty-xy* images.

Fig. 19

Fig. 20

8. Conclusions

This paper presents a Fourier-based approach for automatically constructing a 3D panoramic model of a natural scene from a video sequence. The video sequences could be captured by an unstabilized camera mounted on a moving platform on a common road surface. As the input of the algorithms, "seamless" panoramic view images (PVIs) and epipolar plane images (EPIs) are generated after an image stabilization step to eliminate fluctuation from the vehicle's motion. A novel panoramic EPI analysis method is proposed that combines the advantages of both PVIs and EPIs efficiently in three important steps: locus orientation detection in the Fourier frequency domain, motion boundary localization in the spatio-temporal domain, and occlusion/resolution recovery only at motion boundaries. The Fourier energy-based approaches in literature have been

proposed for low-level local motion analysis, but the previous approaches are not accurate for 3D reconstruction and are computationally expensive. We have presented and analyzed an occlusion model in both spatio-temporal and frequency domains. Based on the occlusion model, and spatio-temporal-frequency analysis, we have proposed the Gaussian-windowed Fourier Orientation Detector (GFOD). The GFOD is accurate in both locus orientation estimation and depth boundary localization. With the GFOD, effective data selection and fast GFOD algorithm, our panoramic EPI analysis approach is both accurate and efficient for 3D reconstruction. Examples of layered panoramic representations for large-scale 3D scenes from real world video sequences are given.

While the proposed method is a practical solution for 3D scene modeling, there exist some open problems that need further study. The current algorithms can work well only with dense image sequences with constrained motions. The motion boundary localizer is not as robust as we have expected. The fusion of depth/motion and spatial structures (textures, edges) also need further study. The applications of the proposed GFOD to other areas could also be investigated. We hope these open issues will attract research interests in the computer vision and related communities.

Appendix 1. Energy model of the occluding boundary

Define the following functions for Eq. (7)

$$g_1(x,t) = u(x - v_2 t) f_1(x - v_1 t) \tag{a-1}$$

$$g_2(x,t) = (1 - u(x - v_2 t)) f_2(x - v_2 t) \tag{a-2}$$

It is easy to prove that

$$G_2(\xi,\omega) = (F_2(\xi) - F_2(\xi) * U(\xi))\delta(v_2\xi + \omega) \tag{a-3}$$

Now we will prove

$$G_1(\zeta,\omega) = \frac{1}{v_1 - v_2} F_1(\frac{v_2\zeta + \omega}{v_2 - v_1}) U(\frac{v_1\zeta + \omega}{v_1 - v_2}) \tag{a-4}$$

In Eq. (a-1), let $x' = x - v_1 t$, then $x = x' + v_1 t$, $x - v_2 t = x' + (v_1 - v_2)t$, therefore

$$G_1(\xi,\omega) = \int_{x'}\int_t f_1(x') u(x' + (v_1 - v_2)t) e^{-j2\pi(\xi x' + (\xi v_1 + \omega))t} dx' dt$$

$$= \int_{x'} f_1(x') e^{-j2\pi\xi x'} \left\{ \int_t u(x' + (v_1 - v_2)t) e^{-j2\pi(\xi v_1 + \omega)t} dt \right\} dx'$$

Let $t' = x' + (v_1 - v_2)t$, then $t = \frac{t' - x'}{v_1 - v_2}$, hence

$$G_1(\xi,w)=\int_{x'} f_1(x')e^{-j2\pi\xi x'}\left\{\int_{t'} u(t')e^{-j2\pi\frac{\xi v_1+\omega}{v_1-v_2}t'}e^{j2\pi\frac{\xi v_1+\omega}{v_1-v_2}x'}\frac{dt'}{v_1-v_2}\right\}dx'$$

$$=\frac{1}{v_1-v_2}\int_{x'} f_1(x')e^{j2\pi\frac{\xi v_2+\omega}{v_2-v_1}x'}dx'\int_{t'} u(t')e^{-j2\pi\frac{\xi v_1+\omega}{v_1-v_2}t'}dt'$$

$$\frac{1}{v_1-v_2}F_1\left(\frac{\xi v_2+\omega}{v_2-v_1}\right)U\left(\frac{\xi v_1+\omega}{v_1-v_2}\right)$$

Appendix 2. Fast GFOD algorithm

The m×m Fourier transform, $G(\xi,\omega)$, of function $g_w(x,t) = f(x,t)w(x,t) = f(x,t)w_1(x)w_2(t)$ is calculated in two steps. First, for each column in the x direction, performing a 1D FFT obtains an intermediate result $G(\xi,t)$. Second, applying 1D FFTs along the *t* direction to obtain the final Fourier transform $G(\xi,\omega)$. When the Gaussian window is centered at (x_0, t_1), the origin of an m×m sub-image, the Gaussian Fourier transform for column *t* in this sub-image is

$$G_{t_1}(\xi,t)=F\left[g(x,t+t_1)w_1(x)w_1(t)\right]=F\left[g(x,t+t_1)w_1(x)\right]w_1(t) \tag{a-5}$$

Assume that the next location that a Gaussian window will be applied is $t_2 = t_1 + \Delta T$, then the Gaussian-Fourier transform of column *t* with origin (x_0, t_2) should be

$$\begin{aligned} G_{t_2}(\xi,t) &= F\left[g(x,t+t_2)w_1(x)w_1(t)\right] \\ &= F\left[g(x,t+t_1+\Delta T)w_1(t+\Delta T)\right]w_1(t+\Delta T)\frac{w_1(t)}{w_1(t+\Delta T)} \\ &= G_{t_1}(\xi,t+\Delta T)\frac{w_1(t)}{w_1(t+\Delta T)} \end{aligned} \tag{a-6}$$

Acknowledgments

This work was supported by the China Advanced Research Project, by the China High Technology Program under contract No. 863-306-ZD-10-22 and partially by the China Natural Science Foundation under contact No. 69805003. The first author is also supported by the CUNY Graduate Research Technology Initiative program. We would like to thank Prof. Edward M. Riseman and Prof. Allen R. Hanson of UMass at Amherst for their valuable discussions and comments that lead to the first version of this paper. We also thank the anonymous reviewers for their insightful comments and constructive suggestions for revising the paper, and Mr. Robert Hill at the City College of New York for proofreading the revised manuscript.

Notes

1. Camera settings other than this standard stetting are also applicable but an image rectification procedure should be applied first (Zhu, 2001).

2. So in practice, the variance will be selected adaptively according to the real situation of an ST texture instead of using $\sigma^2 = \frac{m-1}{4}$ directly.

References

[1]. Adelson, E. H. and Bergen, J. R. 1985. Spatiotemporal energy model for the perception of motion. *J. Opt. Soc. Am.*, A2: 284-299.

[2]. Allmen, M. and Dyer, C. R. 1991. Long range spatiotemporal motion understanding using spatiotemporal flow curves. In *IEEE Conf. Computer Vision and Pattern Recognition,* pp. 303-309.

[3]. Baillard, C. and Zisserman, A. 1999. Automatic reconstruction of piecewise planar models from multiple views. In *IEEE Conf. Computer Vision and Pattern Recognition,* pp. 559-565.

[4]. Baker, H. H. and Bolles, R. C., 1989. Generalizing epipolar-plane image analysis on the spatiotemporal surface. *Int. J. Computer Vision*, 3, pp 33-49.

[5]. Baker, S., Szeliski, R. and Anandan, P. 1998. A layered approach to stereo reconstruction. *IEEE Conf. Computer Vision and Pattern Recognition,* pp. 434-441.

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

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

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

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

[10]. Chen, S. E. 1995. QuickTime VR - an image based approach to virtual environment navigation. In *ACM Conf. Proc. SIGGRAPH 95*, pp. 29-38.

[11]. Collins, R. 1996. A space-sweep approach to true multi-image matching. In *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 358-363.

[12]. Collins, R., Jaynes, C., Cheng, Y., Wang, X., Stolle, F., Schultz, H., Hanson, A. and Riseman, E. 1998. The Ascender System: Automated Site Modeling from Multiple Aerial Images," *Computer Vision and Image Understanding*, 72(2): 143-162.

[13]. Coorg, S. and Teller, S. 1998. Automatic extraction of textured vertical facades from pose imagery. *MIT LCS TR-729*.

[14]. Dalmia, A. K. and Trivedi, M. 1996. High speed extraction of 3D structure of selectable quality using a translating camera. *Computer Vision and Image Understanding*, 64(1): 97-110.

[15]. Debevec, P., Taylor, C. and Malik, J. 1996. Modeling and rendering architecture from photographs: a hybrid geometry- and image- based approach. In *ACM Conf. Proc. SIGGRPAH 96*, pp. 11-20.

[16]. Faugeras, O., Robert, L., Laveau, S., Csurka, G., Zeller, C., Gauclin, C. and Zoghlami, I. 1998. 3-D reconstruction of urban scenes from image sequences. *Computer Vision and Image Understanding*, 69(3): 292-309.

[17]. Fleet, D. J., Black, M. J., and Jepson, A. D. 1998, Motion feature detection using steerable flow fields. In *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 274-281.

[18]. Freeman, W. T. and Adelson, E. H. 1991. The design and use of steerable filters. *IEEE Trans Pattern Analysis and Machine Intelligence,* 13(9): 891-906.

[19]. Hansen, M., Anandan, P., Dana, K., van de Wal, G., and Burt, P., 1994. Real-time scene stabilization and mosaic construction. In *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 54-62.

[20]. Heeger , D. J. 1987. Optical flow from spatio-temporal filters. In *Proc. IEEE Int. Conf. Computer Vision,* pp. 181-190.

[21]. Ishiguro, H., Yamamoto, M. and Tsuji S. 1990, Omni-directional stereo for making global map. In *Proc. IEEE Int. Conf. Computer Vision*, pp. 540-547.

[22]. Jahne B, *Digital Image Processing , Concept, Algorithms and Scientific Applications*, Springer-Verlag, 1991.

[23]. Li, Y.,. Shum, H.-Y., Tang, C.-K., and Szeliski, R., 2004. Stereo reconstruction from multiperspective panoramas. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(1):44-62, January 2004.

[24]. McMillan L. and Bishop, G. 1995. Plenoptic modeling: an image-based rendering system. In *ACM Conf. Proc. SIGGRAPH 95*, pp. 39-46.

[25]. Morimoto, C. and Chellappa, R. 1997. Fast 3-D stabilization and mosaic construction. In *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 660-665.

[26]. Murray, D. W. 1995. Recovering range using virtual multicamera stereo, *Computer Vision and Image Understanding.* 61(2): 285-291.

[27]. Nayar, S. and Karmarkar, 2000. 360x360 mosaics. In *IEEE Conf. Computer Vision and Pattern Recognition*: II 388-395.

[28]. Niyogi, S. A. 1995. Detecting kinetic occlusion. In *Proc. IEEE Int. Conf. Computer Vision*, pp. 1044-1049.

[29]. Peleg, S. and Ben-Ezra, M. 1999. Stereo panorama with a single camera. In *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 395-401.

[30]. Peleg, S., Ben-Ezra, M. and Pritch, Y., 2001. OmniStereo: Panoramic Stereo Imaging, *IEEE Trans. on PAMI*, vol 23, no 3, March 2001, pp. 279-290.

[31]. Peleg, S. and Herman, J. 1997. Panoramic mosaics by manifold projection. In *IEEE Conf. Computer Vision and Pattern Recognition*: pp. 338-343.

[32]. Peleg, S., Rousso, B., Rav-Akha, A. and Zomet, A., 2000. Mosaicing on Adaptive Manifolds, *IEEE Trans. Pattern Recognition and Machine Analysis*, 22(10), October 2000: 1144-1154.

[33]. Rademacher, P. and Bishop, G. 1998. Multiple-center-of-projection images. In *Proc. SIGGRAPH'98*, pp. 199-206.

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

[35]. Sawhney, H. S., Kumar, R., Gendel, G., Bergen, J., Dixon, D. and Paragano, V. 1998. VideoBrush™: Experiences with consumer video mosaicing. In *IEEE Workshop on Application of Computer Vision*, pp. 56-62.

[36]. Shade, J., Gortler, S., He. L. and Szeliski, R. 1998. Layered depth image. In *Proc. SIGGRAPH'98,* pp. 231-242.

[37]. H.-Y. Shum and R. Szeliski, Construction of panoramic image mosaics with global and local alignment, *Int. J. of Computer Vision*, vol. 36, no. 2, 2000: 101-130.

[38]. Shum, H.-Y., Han, M. and Szeliski, R. 1998. Interactive construction of 3D models from panoramic mosaics. *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 427-433.

[39]. Shum, H.-Y. and Szeliski, R., 1999. Stereo reconstruction from multiperspective panoramas. In *Proc. IEEE Int. Conf. Computer Vision*, pp. 14-21.

[40]. Shum H-Y, Kalai A and Seitz S M. Omnivergent stereo. *Proc. IEEE Int. Conf. Computer Vision,* pp 22 – 29, 1999.

[41]. Szeliski, R. 1999. A multi-view approach to motion and stereo. In *IEEE Conf. Computer Vision and Pattern Recognition,* pp. 157-163.

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

[43]. Xiong, Y. and Turkowski, K. 1997. Creating image-based VR using a self-calibrating fisheye lens. In *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 237-243.

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

[45]. Zheng, J. Y. and Tsuji, S. 1998. Generating Dynamic Projection Images for Scene Representation and Understanding. *Computer Vision and Image Understanding*, 72(3): 237-256.

[46]. Zhu, Z., Xu, G. and Lin, X. 1998. Constructing 3D natural scene from video sequences with vibrating motions. In *Proc. IEEE Virtual Reality Annual International Symposium (VRAIS-98)*, pp. 105 – 112.

[47]. Zhu, Z., Xu, G. and Lin, X., 1999. Panoramic EPI Generation and Analysis of Video from a Moving Platform with Vibration. In *IEEE Conf .Computer Vision and Pattern Recognition*, pp. 531-537.

[48]. Zhu, Z. 2001. *Full View Spatio-Temporal Visual Navigation - Imaging, Modeling and Representation of Real Scenes*, China Higher Education Press, December 2001 (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/).

[49]. Zhu, Z. and A. R. Hanson, 2001. 3D LAMP: a new layered panoramic representation. In *Proc. IEEE Int. Conf. Computer Vision*, vol II, 723-730.

[50]. Zhu, Z., E. M. Riseman and A. R. Hanson, 2001. Parallel-perspective stereo mosaics, In *Proc. IEEE Int. Conf. Computer Vision,* vol I, 345-352.

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

Figure Captions

Fig. 1. (a) Motion model with a side-looking camera, constant speed translation and small vibrations. (b) ST image model: xyt cube, panoramic view image (PVI) and epipolar plane image (EPI)

Fig. 2. ST images from an image sequence (a) Stereo PVIs (x = 0 and x= -56) (b) Loci, occlusion and side regions in an EPI with y = 9 that starts at the PVI with x=0 and ends at the PVI with x = -56.

Fig. 3. Stabilization and ST image generation of the BUILDING sequence. (a) PVI (x=24) before and after stabilization (b) EPI (y = 0) before and after stabilization. Only a small portion of the 1024-column ST images are shown.

Fig. 4. Stabilization of the TREE sequence (128×128×1024). (a) PVI (x=0) before and after stabilization (b) EPI (y =0) before and after stabilization (c) Depth estimation from the stabilized EPIs (the nearer, the brighter). Only a small portion of the 1024-column ST images are shown.

Fig. 5. The motion occlusion model. (a) an x-t image $g(x,t)$ (b) Fourier magnitude map of the x-t image (c) the Fourier magnitude of the step function u(x)

Fig. 6. Multiple orientation detection by 64×64 GFOD and the motion boundary localizer. Rows 1-4: 64×1024 x-t image (EPI, y = 56); orientation energy distribution map (the long dashed curve (red in color version) indicates the selected peaks (which may not be the highest), and several small pieces of solid lines (blue in color version) indicate the second peaks); histogram of orientation angles; and part of the corresponding PVI (PVI x = 24 is selected in order to shown the side face of building around frame 613; the horizontal line in the PVI corresponds to location of the EPI in the first row). The significant motion boundaries are marked by vertical lines (red in color version).

Fig. 7. The principles of depth boundary localization and depth interpolation. (a) Occlusion case: two peaks can be detected in the Fourier spectrum in all of the tree cases when the center of the window is to the left, just at and to the right of a depth boundary (b) reappearance case (c) orientation detection in multi scales (d) depth interpolation for textureless regions.

Fig. 8. Multiple orientation detection: comparison of rectangular and Gaussian windows. The frame index (t) corresponds to the time in the EPI shown in Fig. 6. In both (1) and (2), (a) – (d) rectangular window: (a) the original 64 ×64 x-t image f(x,t). (b) energy spectra of f(x,t) (c) the orientation histogram with the detected peak(s). (d) motion boundary localization results. Column (e) – (h) Gaussian window: (e) Gaussian weighted x-t image g(x,t) (f) energy spectra of g(x,t) (c) the orientation histogram with the detected peak(s) (d) motion boundary localization results. In each of (c), (d) , (g) or (h), the solid dark line indicates the correct orientation, while the dashed dark line indicates the second peak (if any).

Fig. 9. Multiple orientation detection by 16x16 Fourier orientation detection without motion boundary localization. Rows 1-4: 16x1024 x-t image (EPI, y = 56); orientation energy distribution map (the dark (blue in color version) curves shows the high peaks at each frame t); histogram of orientation angles; and part of the corresponding PVI (the horizontal line in the PVI corresponds to the EPI in the first row).

Fig. 10. Multiple orientation detection by 64×64 GFOD, and without motion boundary localization. Rows 1-4: 16x1024 x-t image (EPI, y = 56); orientation energy distribution map (the dark (blue in color version) curves shows the high peaks at each frame t); histogram of orientation angles; and part of the corresponding PVI (the horizontal line in the PVI corresponds to the EPI in the first row). The significant motion boundaries are marked by vertical lines (red in color version).

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

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). (c) Multi-viewpoint mosaic with adaptive time scales (the right edge of the upper part connects with the left edge of the bottom part). Circles show the corresponding OCCLUDED and SIDE regions in (b).

Fig. 13. Top: panoramic intensity image (x =0); bottom: panoramic belief map. The brighter intensity in the belief map shows stronger belief.

Fig. 14. Fast GFOD using a moving window approach

Fig. 15. Two frames of the FLOWER GARDEN (FG) sequence

Fig. 16 Panorama and epipolar plane images of FG sequence (a) PVI (b) EPI and (c) XPI

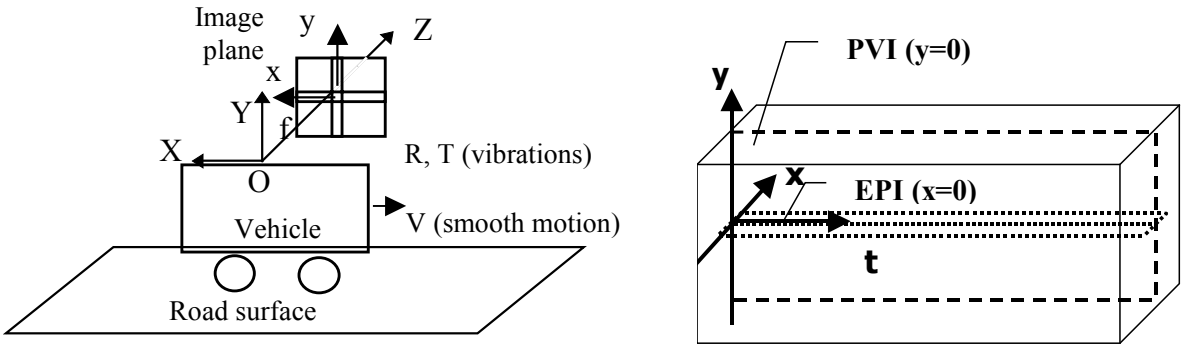
Fig. 17. Panoramic depth maps for the BUILDING sequence: comparison under various parameter selections (B- boundary localization, G – Gaussian windowing, and M – window size). In all depth maps, the nearer depths are represented by brighter intensities. Large GFOD with the motion boundary localizer yields the best result.

Fig. 18. Internal data of (part of) the multi-resolution layered representation of the BUILDING sequence. (a) Resolution enhancement (without occlusion recovery) (b) Occlusion recovery as well resolution enhancement

Fig. 19. Panoramic depth map for the FG sequence. (1) Isometric depth lines overlaid in the intensity map (the isometric depth lines have 2^o intervals). (2)panoramic depth map

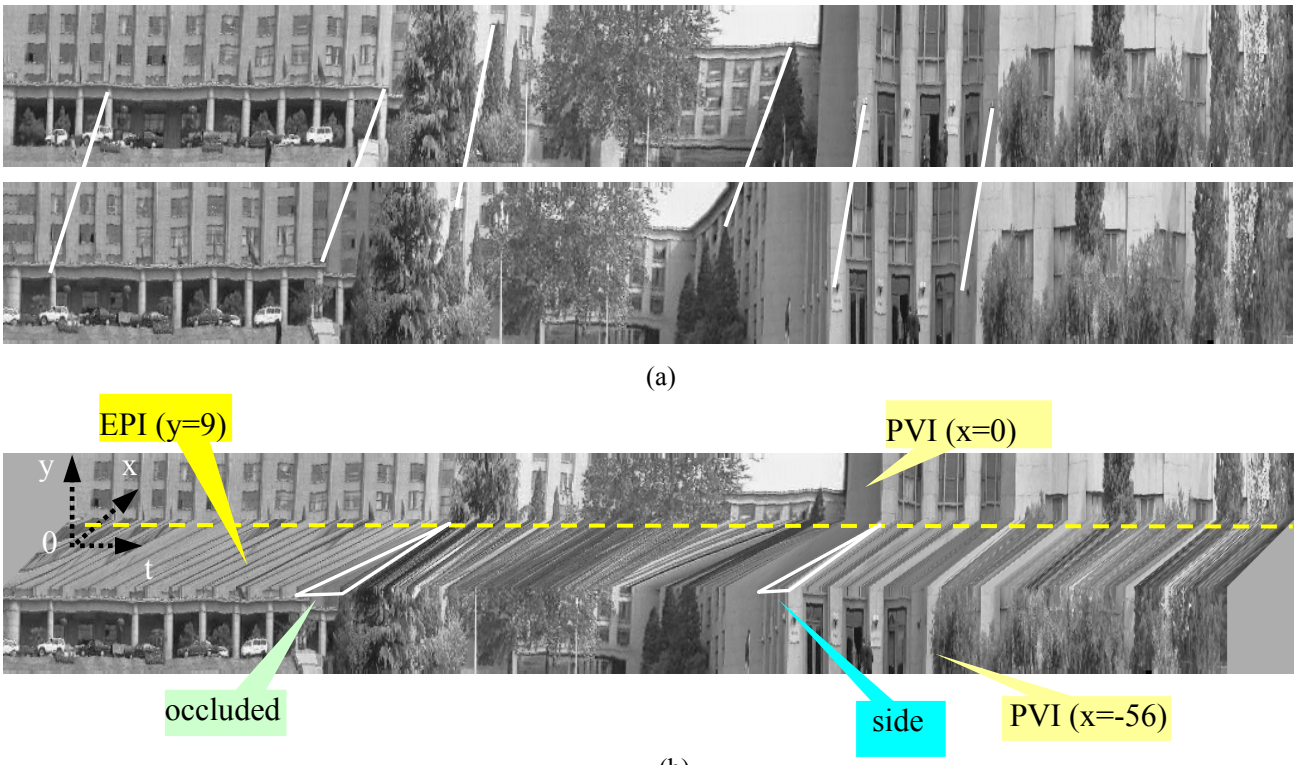
Fig. 20. Layered representation model of the FG sequence

Table 1. A systematic comparison of depth estimation (G =0/1: rectangular /Gaussian windows; B (0/1): motion boundary localizer, M = 16, 32, 64: window sizes. Grading 1-6 is from best to worst. The depth maps corresponding to those marked by * are shown in Fig. 17)



(a) (b)

Fig. 1. (a) Motion model with a side-looking camera, constant speed translation and small vibrations. (b) ST image model: xyt cube, panoramic view image (PVI) and epipolar plane image (EPI)



(a)

(b)

Fig. 2. ST images from an image sequence (a) Stereo PVIs (x = 0 and x= -56) (b) Loci, occlusion and side regions in an EPI with y = 9 that starts at the PVI with x=0 and ends at the PVI with x = -56.

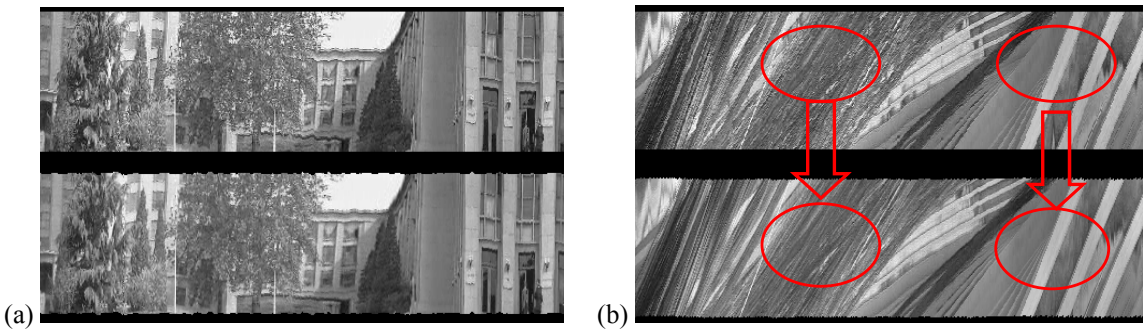


Fig. 3. Stabilization and ST image generation of the BUILDING sequence. (a) PVI (x=24) before and after stabilization (b) EPI (y = 0) before and after stabilization. Only a small portion of the 1024-column ST images are shown.

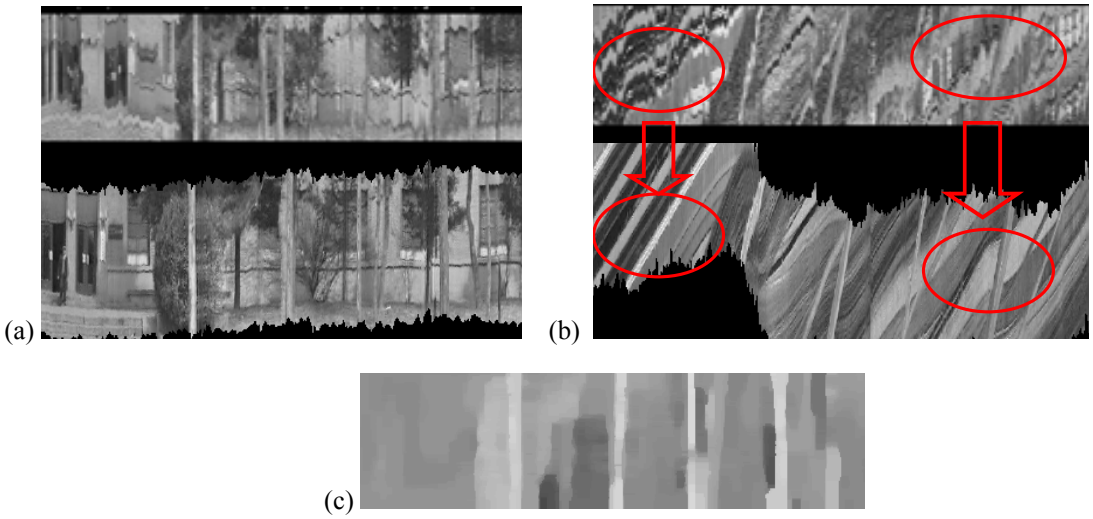


Fig. 4. Stabilization of the TREE sequence (128×128×1024). (a) PVI (x=0) before and after stabilization (b) EPI (y =0) before and after stabilization (c) Depth estimation from the stabilized EPIs (the nearer, the brighter). Only a small portion of the 1024-column ST images are shown.

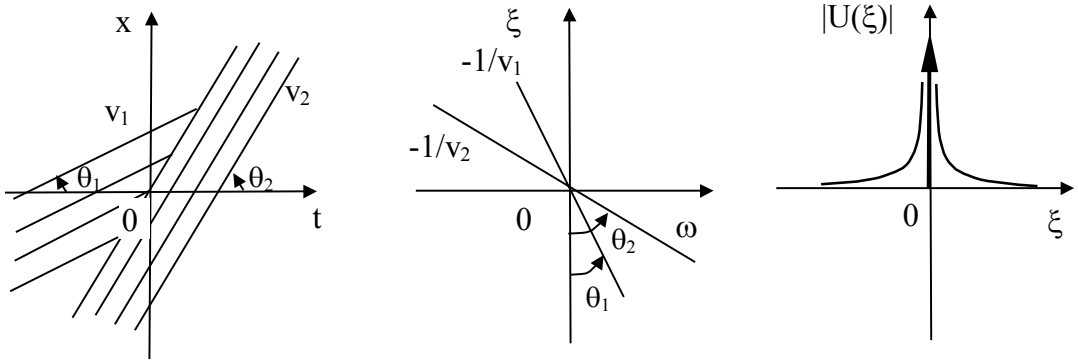


Fig. 5. The motion occlusion model. (a) an x-t image *g(x,t)* (b) Fourier magnitude map of the x-t image (c) the Fourier magnitude of the step function u(x)

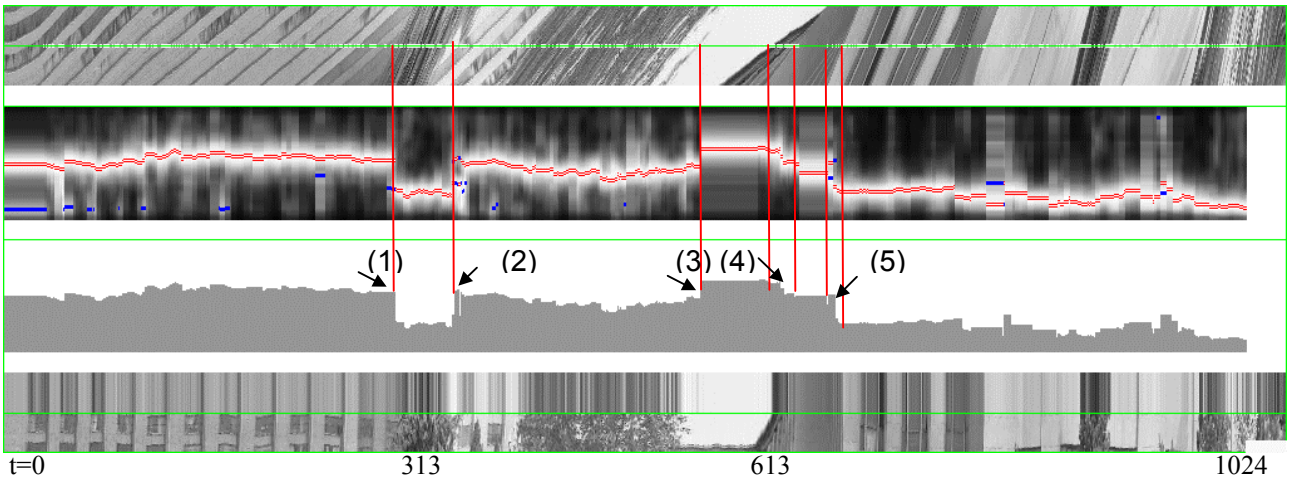


Fig. 6. Multiple orientation detection by 64×64 GFOD and the motion boundary localizer. Rows 1-4: 64×1024 x-t image (EPI, y = 56); orientation energy distribution map (the long dashed curve (red in color version) indicates the selected peaks (which may not be the highest), and several small pieces of solid lines (blue in color version) indicate the second peaks); histogram of orientation angles; and part of the corresponding PVI (PVI x = 24 is selected in order to shown the side face of building around frame 613; the horizontal line in the PVI corresponds to location of the EPI in the first row). The significant motion boundaries are marked by vertical lines (red in color version).

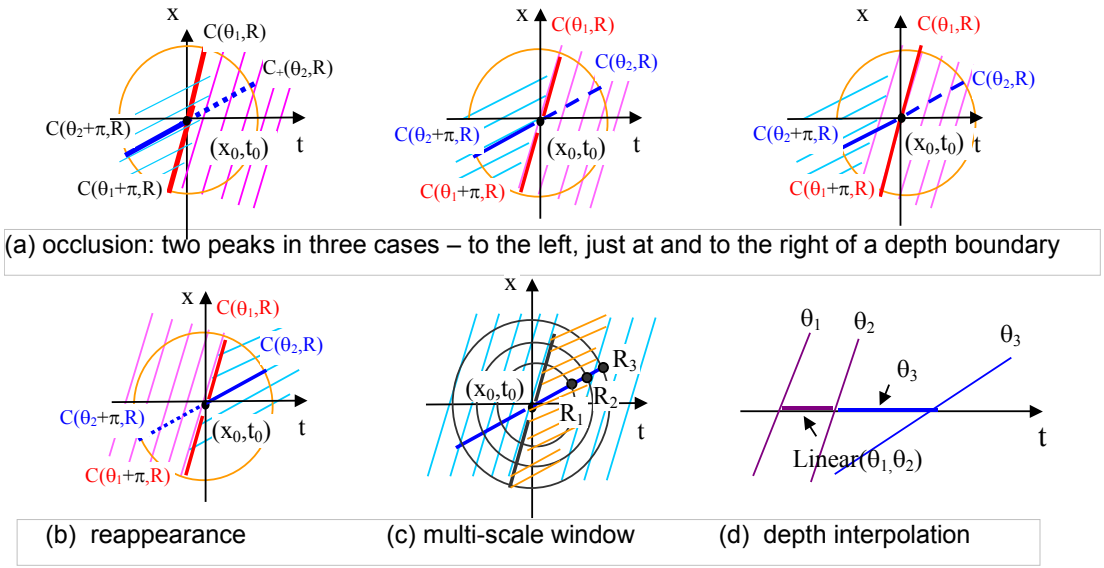
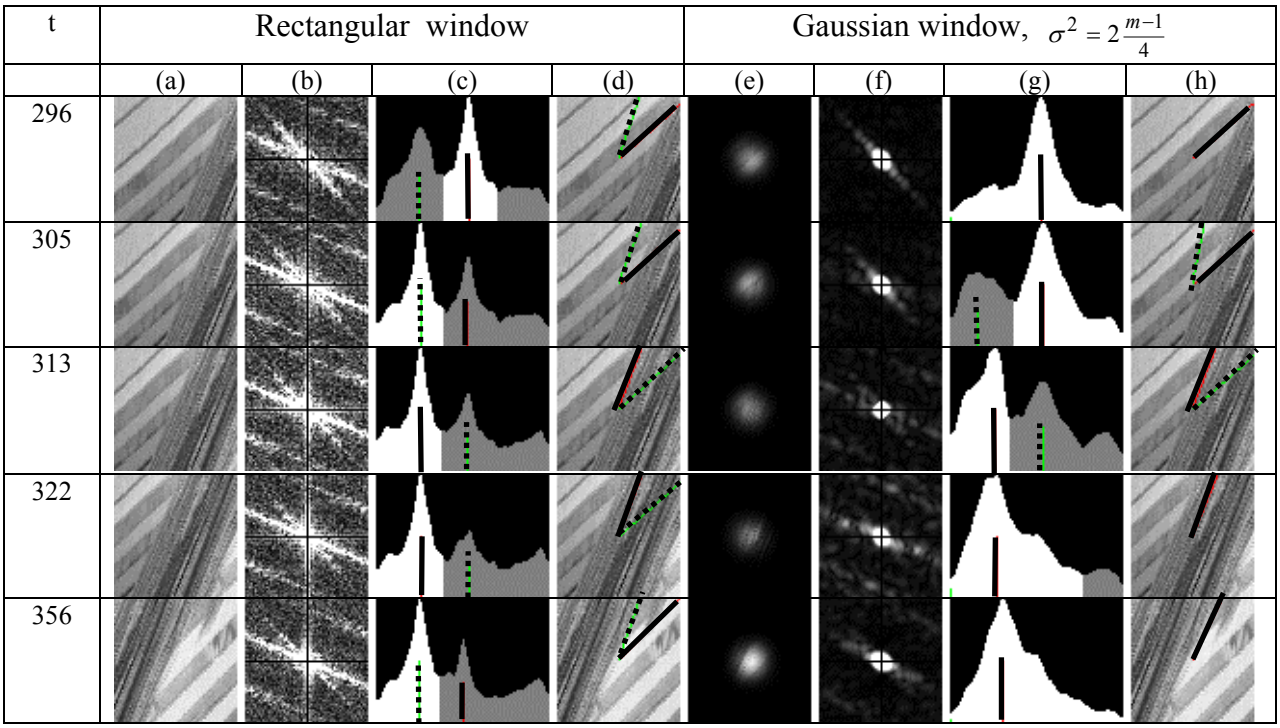
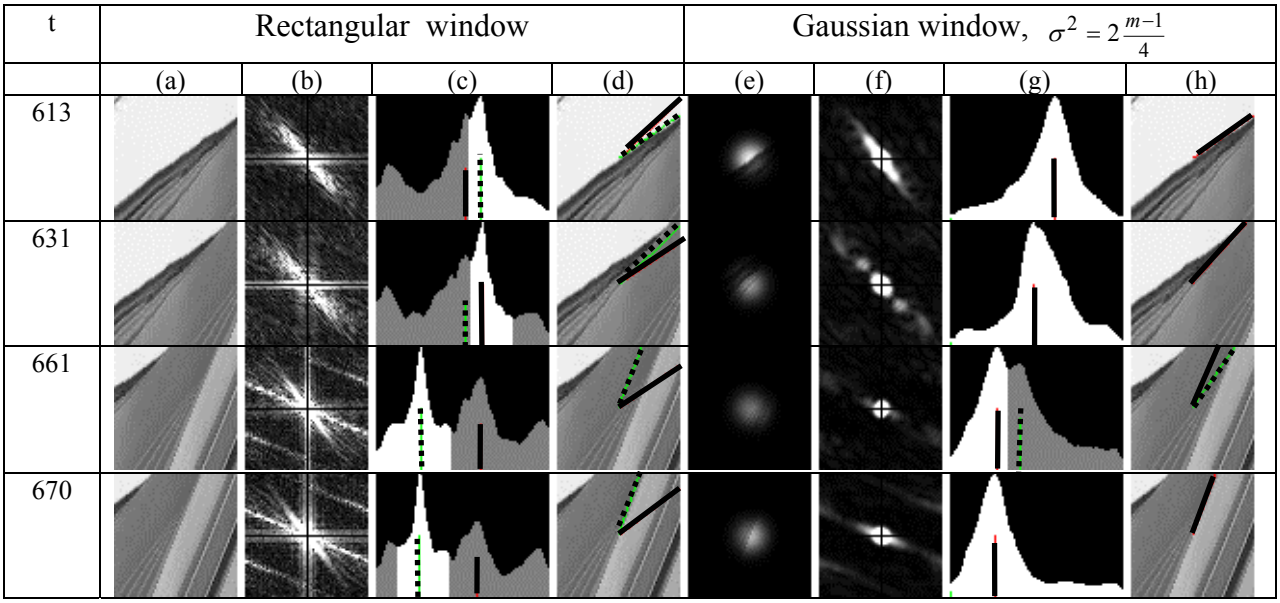


Fig. 7. The principles of depth boundary localization and depth interpolation. (a) Occlusion case: two peaks can be detected in the Fourier spectrum in all of the tree cases when the center of the window is to the left, just at and to the right of a depth boundary (b) reappearance case (c) orientation detection in multi scales (d) depth interpolation for textureless regions.



(1) GFOD operator on occluding boundary



(2)GFOD operator on a side surface

Fig. 8. Multiple orientation detection: comparison of rectangular and Gaussian windows. The frame index (t) corresponds to the time in the EPI shown in Fig. 6. In both (1) and (2), (a) – (d) rectangular window: (a) the original 64 ×64 x-t image f(x,t). (b) energy spectra of f(x,t) (c) the orientation histogram with the detected peak(s). (d) motion boundary localization results. Column (e) – (h) Gaussian window: (e) Gaussian weighted x-t image g(x,t) (f) energy spectra of g(x,t) (c) the orientation histogram with the detected peak(s) (d) motion boundary localization results. In each of (c), (d) , (g) or (h), the solid dark line indicates the correct orientation, while the dashed dark line indicates the second peak (if any).

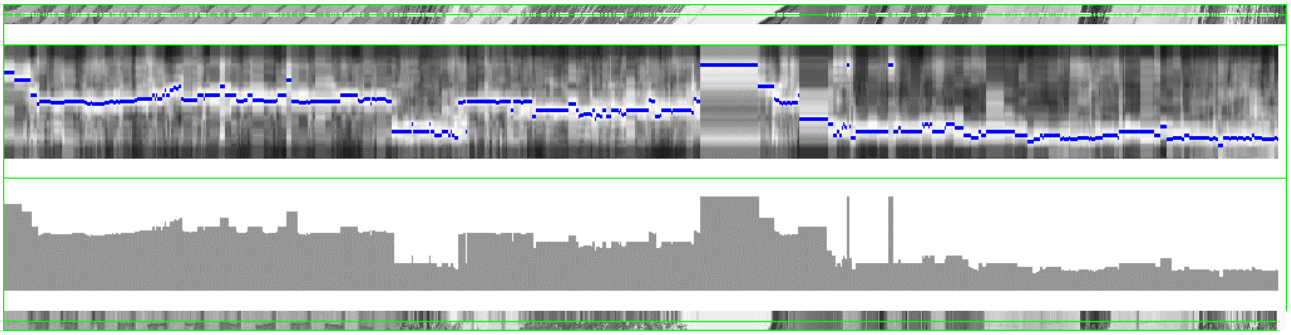


Fig. 9. Multiple orientation detection by 16x16 Fourier orientation detection without motion boundary localization. Rows 1-4: 16x1024 x-t image (EPI, y = 56); orientation energy distribution map (the dark (blue in color version) curves shows the high peaks at each frame t); histogram of orientation angles; and part of the corresponding PVI (the horizontal line in the PVI corresponds to the EPI in the first row).

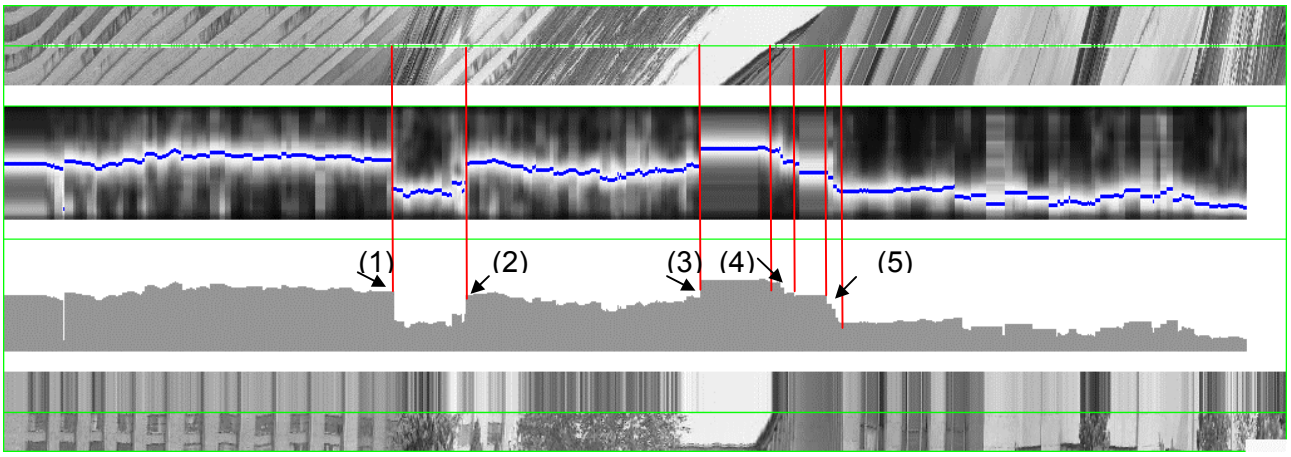


Fig. 10. Multiple orientation detection by 64×64 GFOD, and without motion boundary localization. Rows 1-4: 16x1024 x-t image (EPI, y = 56); orientation energy distribution map (the dark (blue in color version) curves shows the high peaks at each frame t); histogram of orientation angles; and part of the corresponding PVI (the horizontal line in the PVI corresponds to the EPI in the first row). The significant motion boundaries are marked by vertical lines (red in color version).

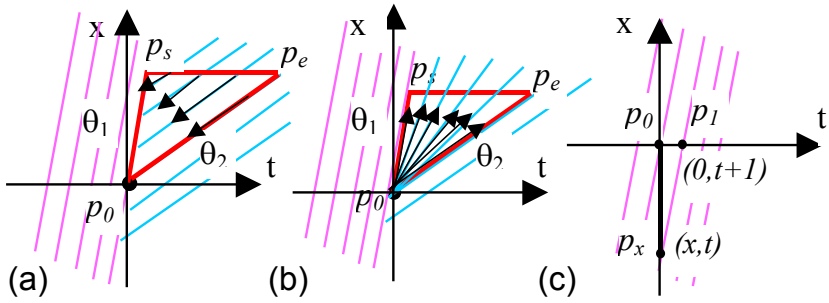


Fig. 11. Region classification and adaptive resolution. (a) locus patterns near an occlusion boundary (b) locus patterns of front- side surfaces (c) temporal resolution enhancement by using spatial resolution

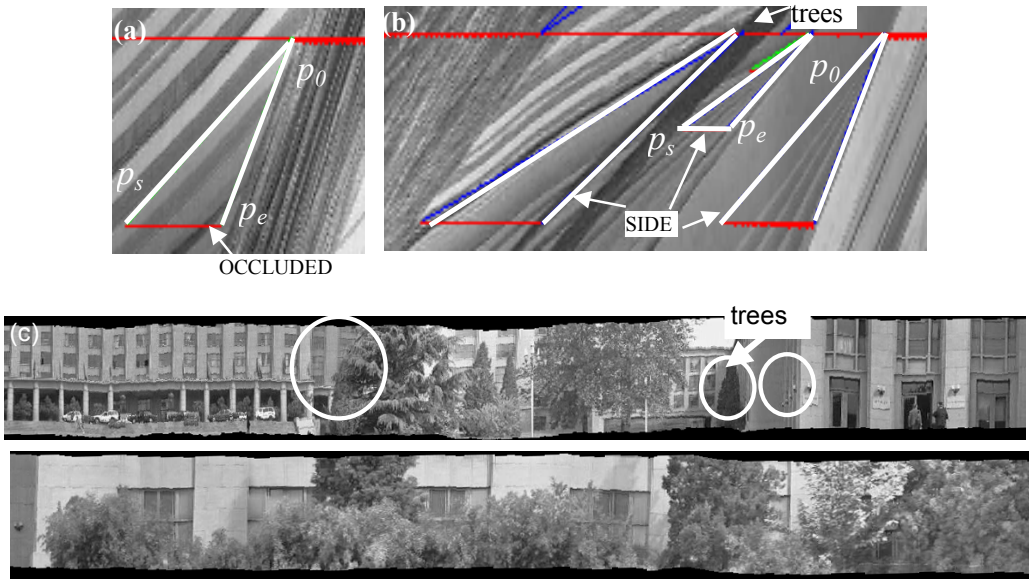


Fig. 12. Occlusion and resolution recovery results in real EPIs. (a) an OCCLUDED region; (b) three SIDE regions (two of them belong to a side facade separated by trees). (c) Multi-viewpoint mosaic with adaptive time scales (the right edge of the upper part connects with the left edge of the bottom part). Circles show the corresponding OCCLUDED and SIDE regions in (b).

Fig. 13. Top: panoramic intensity image (x =0); bottom: panoramic belief map. The brighter intensity in the belief map shows stronger belief.

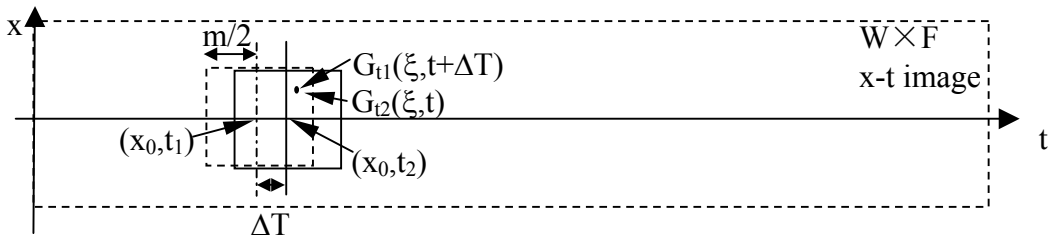
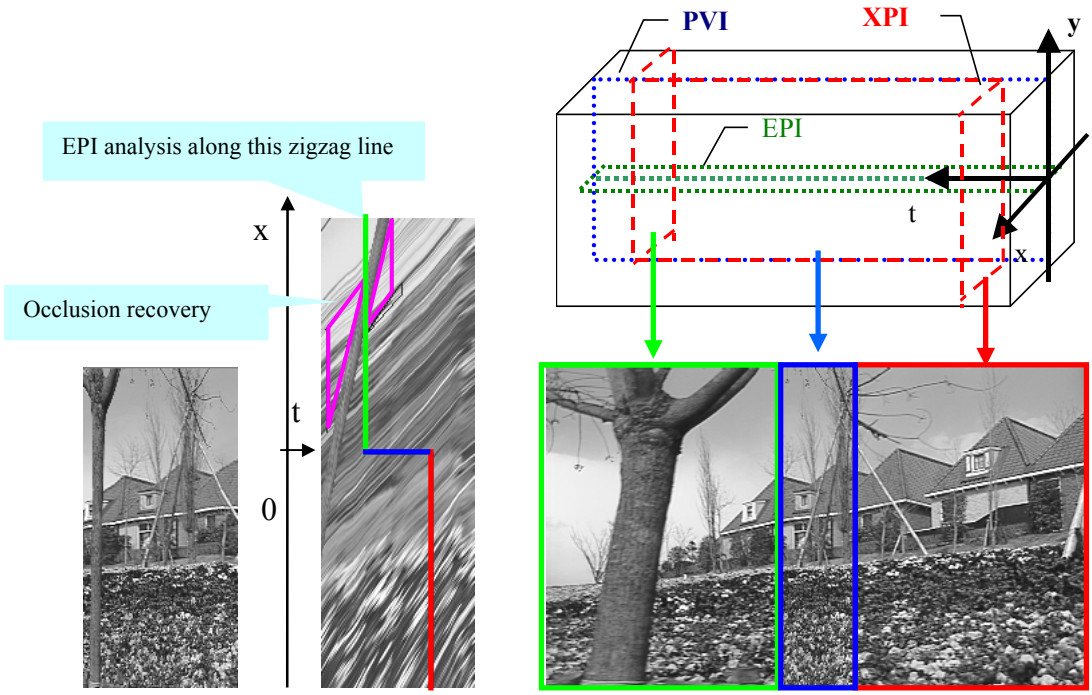


Fig. 14. Fast GFOD using a moving window approach

Fig. 15. Two frames of the FLOWER GARDEN (FG) sequence

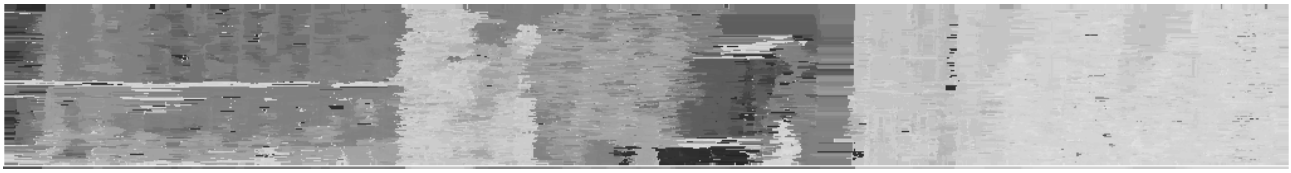


(a) PVI (b) EPI (c) XPI

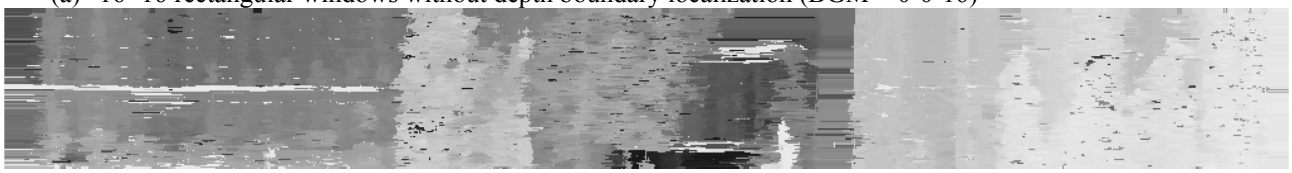
Fig. 16 Panorama and epipolar plane images of FG sequence (a) PVI (b) EPI and (c) XPI

Table 1. A systematic comparison of depth estimation (G =0/1: rectangular /Gaussian windows; B (0/1): motion boundary localizer, M = 16, 32, 64: window sizes. Grading 1-6 is from best to worst. The depth maps corresponding to those marked by * are shown in Fig. 17)

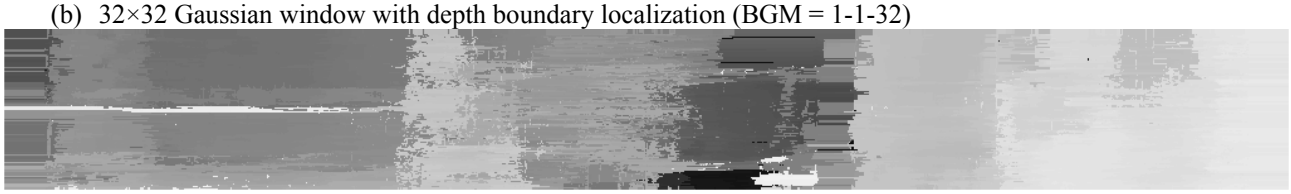
M / B G	16	32	64
0 0	4*	4	6
0 1	4	3	**2***
1 0	6	5	6*
1 1	5	3*	**1***



(a) 16×16 rectangular windows without depth boundary localization (BGM = 0-0-16)



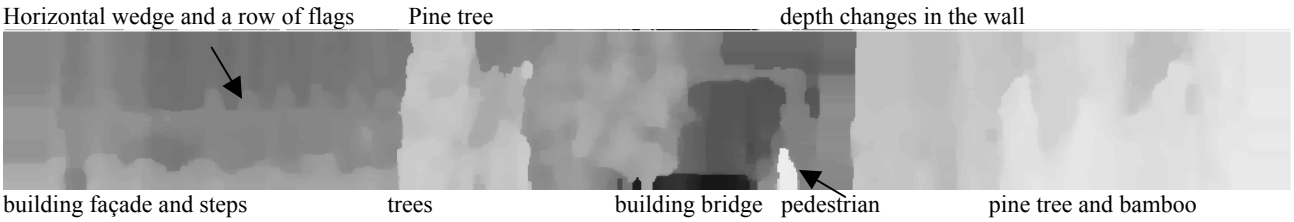
(b) 32×32 Gaussian window with depth boundary localization (BGM = 1-1-32)



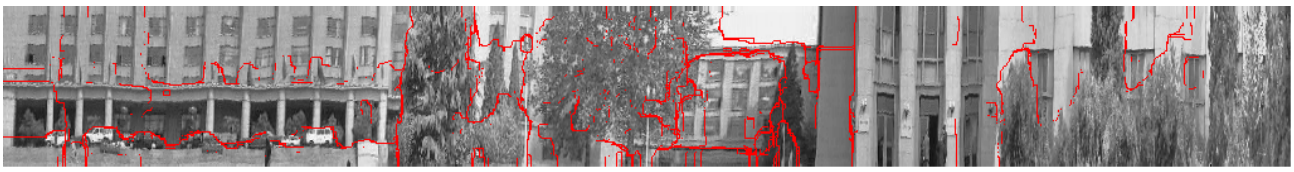
(c) 64×64 rectangular window with depth boundary localization (BGM = 1-0-64)

(d) 64×64 Gaussian window without depth boundary localization (BGM = 0-1-64)

(e) 64×64 Gaussian window with depth boundary localization (BGM = 1-1-64)



(f) panoramic depth map after depth-intensity filtering



(g) depth boundaries (red lines in color version) overlay on the panorama

Fig. 17. Panoramic depth maps for the BUILDING sequence: comparison under various parameter selections (B- boundary localization, G – Gaussian windowing, and M – window size). In all depth maps, the nearer depths are represented by brighter intensities. Large GFOD with the motion boundary localizer yields the best result.

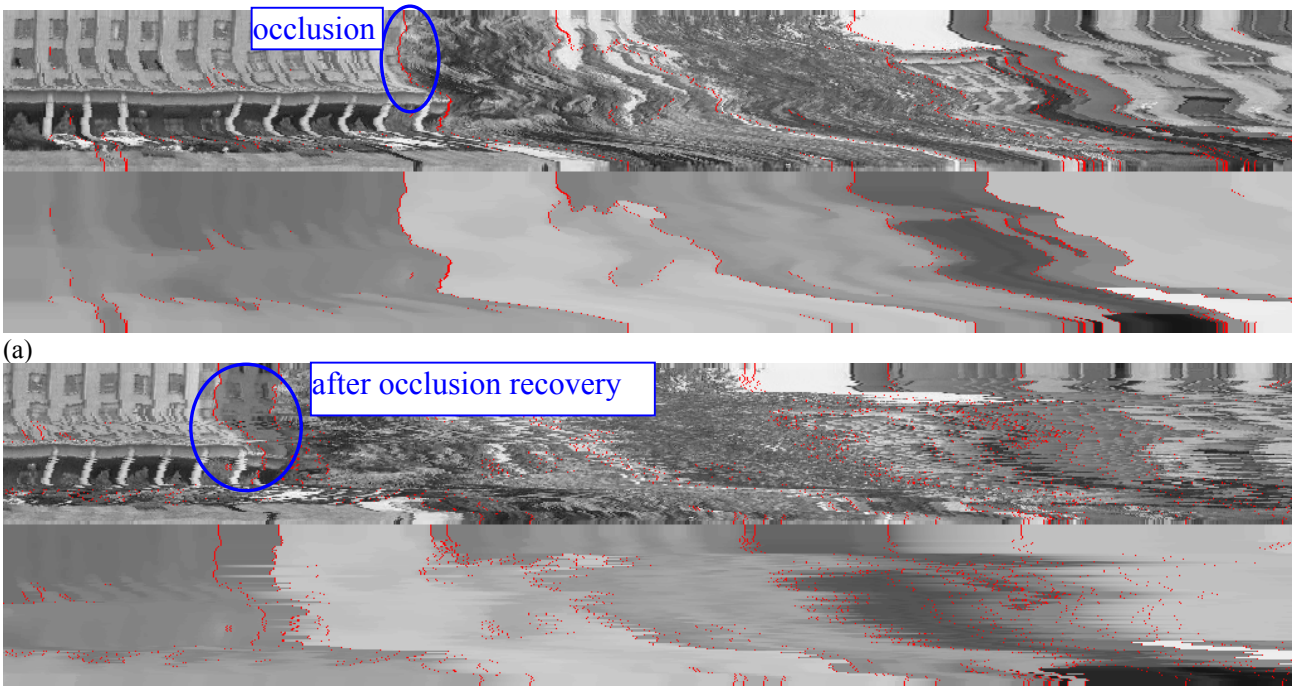


Fig. 18. Internal data of (part of) the multi-resolution layered representation of the BUILDING sequence. (a) Resolution enhancement (without occlusion recovery) (b) Occlusion recovery as well resolution enhancement

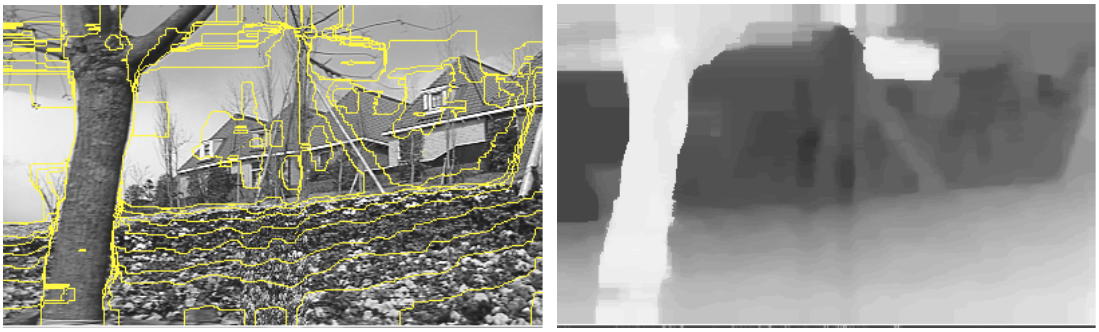
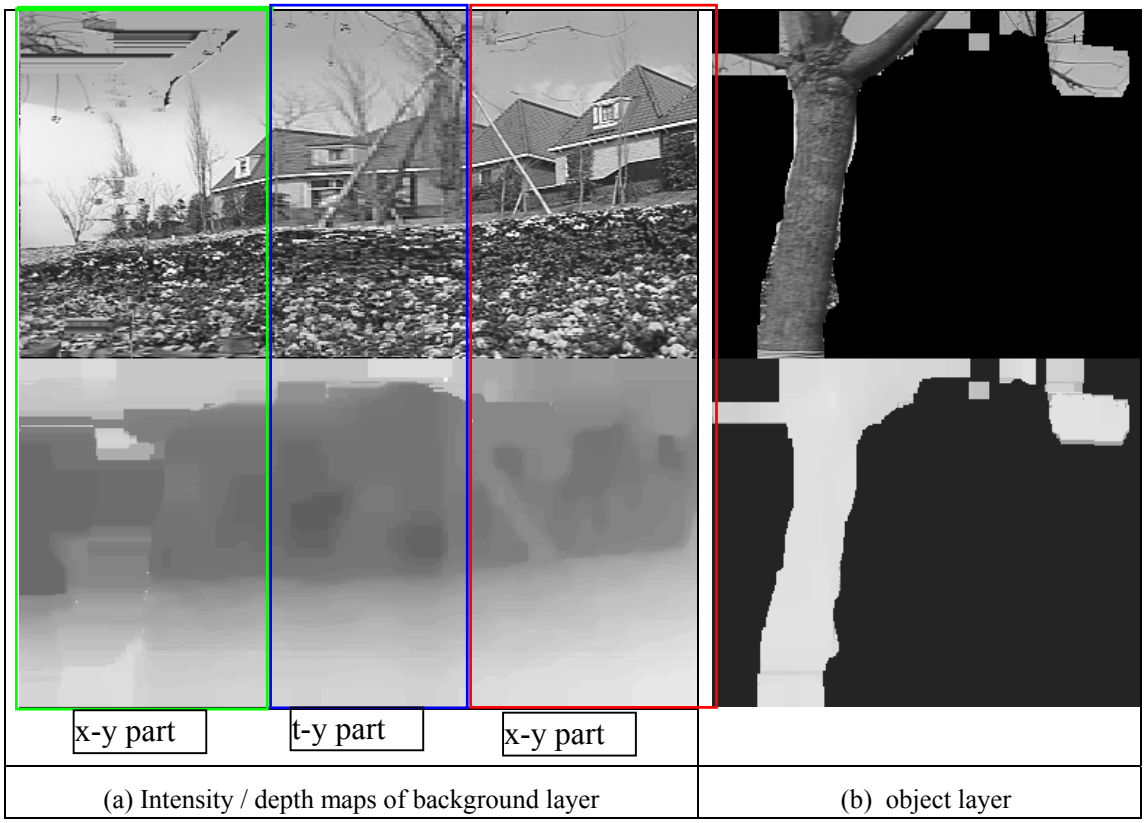


Fig. 19. Panoramic depth map for the FG sequence. (1) Isometric depth lines overlaid in the intensity map (the isometric depth lines have 2^{o} intervals). (2)panoramic depth map



(a) Intensity / depth maps of background layer

(b) object layer

Fig. 20. Layered representation model of the FG sequence