Technical Report TR #99-10, **Computer Science Department, University of Massachusetts at Amherst, February, 1999**

Fast Construction of Dynamic and Multi-Resolution 360° Panorama from Video Sequences

Zhigang Zhu[!+], Guangyou Xu[!]

Edward M. Riseman[+], Allen R. Hanson[+]

[!] Department of Computer Science and Technology

Tsinghua University, Beijing 100084, China

[+] Computer Vision Lab, Computer Science Department

University of Massachusetts at Amherst, MA 01003

zhu@cs.umass.edu

http://www.cs.umass.edu/~zhu

Fast Construction of Dynamic and Multi-Resolution 360° Panorama from Video Sequences*

Zhigang Zhu[!+], Guangyou Xu[!], Edward M. Riseman[+], Allen R. Hanson[+]

[!] Department of Computer Science and Technology

Tsinghua University, Beijing 100084, China

[+] Computer Vision Research Lab, Computer Science Department

University of Massachusetts, Amherst, MA 01003

Abstract

This paper presents a new approach to automatically build a dynamic and multi-resolution 360° panoramic (DMP) representation from image sequences taken by a rotational and zoom camera. A cylindrical panorama is generated by mosaicing the image sequence taken by a hand-held camera. A multi-resolution representation is built for the more interesting areas by means of camera zooming. The dynamic objects in the scene can be detected and represented separately. Although a simple (yet stable) rigid motion model is used to estimate inter-frame motion parameters, the mosaicing methodology presented in this paper enables precise mosaicing. Moving objects are detected and separated from images based on motion information, and more accurate contours are extracted using a modified active contour algorithm. The DMP construction method is fast, robust and automatic, achieving 1 frame per second in a 266MHz PC. No camera calibration, feature extraction, image segmentation or complicated nonlinear iterative processing is required in our algorithms. The construction of the DMP representation can be used in virtual reality, video surveillance and very low bit-rate video coding.

Keywords: *Image-based VR, panoramic representation, dynamic mosaic, multi-resolution, moving object extraction*

---

* This paper was completed with the first author in the Computer Science Department, the University of Massachusetts at Amherst. A large part of the work was performed when he was at Tsinghua University, Beijing, P. R. China. A short version of this paper is published in the Proceedings of the IEEE International Conference on Multimedia Computing and Systems, Florence, Italy, June 7-11, 1999

1. Introduction

A panoramic representation of image sequences has a wide application scope, including virtual reality (VR), interactive 2D/3D video, tele-conferencing, content-based video compression and manipulation, and full-view video surveillance. For virtual reality, it has advantages of simplicity for rendering, photographic quality realism, and 3D illusion experienced by users. For video analysis and coding, it is superior to existing coding approaches in that it is a content-based representation with a very low bit-rate. In the sense of advanced human-computer interaction (HCI), these two categories could be merged into a more general approach of interactive video (e.g. virtual conferencing), which adds the flexibility of synthesizing images with interactivity, selectivity, and enhanced field of view and resolution. This is quite valuable for virtual and/or augmented video conferencing over the Internet.

A wide field of view (FOV) lens, e.g. a fish-eye [1] or panoramic lens [2-6], can be a solution for generating panoramic presentations. Besides the expense of these specially designed image sensors, the image obtained will have substantial distortions, and mapping an entire scene into the limited resolution of a video camera compromises image quality. Constructing a panoramic representation by mosaicing image sequences captured by ordinary cameras, on the other hand, meets the requirements of applications such as virtual reality, wide FOV surveillance, content-based video manipulation and advanced HCI, where high image resolution, low bit-rate, interactivity and photographic realism are needed.

1.1. Related work

Apple's QuickTime VR [7] captures a 360-degree panoramic image of a scene with a camera panning horizontally from a fixed position. The overlap in images are registered first by the user and then "stitched" together by the software in a best match. Similarly in [8] mosaics were constructed by registering and reducing the set of images into a single, larger resolution frame. However the final image mosaic is not a full 360-degree view. Plenoptic modeling[1] [9] adds ranges (using a disparity map) to each panoramic image, thereby allowing reprojection from other viewpoints. The concept of plenoptic function is further explored by light field methods [10,11], which attempt to fully sample the plenoptic function within a subset of space. Clearly the generation of a full-view panorama is the foundation of these methods. In order to construct a 360-degree panorama, a reasonably good

[1] The "plenoptic function" by Adelson and Bergen [26] is a parameterized function for describing everything that is visible from a given point in space. McMillan and Bishop [9] use this plenoptic function for image-based rendering paradigms, such as morphing and view interpolation. In fact it can be viewed as a panoramic or omnidirectional representation.

estimation of the camera focal length is required *a priori* in Apple's QuickTime VR [7] and McMillan & Bishop's method [9].

Shum & Szeliski [12] proposed a mosaic representation that associates a transformation matrix with each input image, rather than explicitly projecting all of the images onto a common surface (e.g., a cylinder). In particular, to construct a full view panorama, they introduced a rotational mosaic representation that associates a rotation matrix (and optionally a focal length) with each input image. However the decomposition of the projective transformation matrix into rotation angles and the focal length is known to be very sensitive to image noise.

Kang & Weiss [13] analyzed the error in constructing panoramic images and proposed a technique that has the advantage of not having to know the camera focal length *a priori*. However in order to create a panorama, they first had to ensure that the camera is rotating about an axis passing through the nodal point. To achieve this, they manually adjusted the position of the camera relative to an X-Y precision stage (mounted on a tripod) such that the parallax effect disappears when the camera is rotated about the vertical axis. The focal length of the camera cannot be changed throughout the rotation.

Xiong and Turkowski [1] proposed a method to create image based VR using a self-calibrating fisheye lens. The nodal point of the fisheye lens needs to be adjusted so that it lies on the rotation axis of the tripod. They take four pictures by rotating the camera 90 degrees after every shot and formulate the registration and self-calibration constraints as a single nonlinear minimization problem in which 34 parameters need to be determined.

Manifold projection [14] enable the fast creation of low distortion panoramic mosaics under a more general motion than the exact panning. The basic principle is the alignment of the strips that contribute to the mosaic, rather than the alignment of the entire overlap between frame. However the issues of circular panorama, independent object motion and camera zoom are not considered in this approach.

Static scenes are a common assumption in image mosaicing and image-based rendering [1, 7, 9-14], with the exception of a dynamic mosaic approach proposed by Irani, Anandan & Hsu [15] to describe dynamic events. However the accuracy of the contour of a moving object was not addressed, which is important for synthesis of fine detail of the dynamic events based on the mosaic representation. In our work we utilized a modified active contour method to extract the contour of the moving object.

The concept of an active contour algorithms was first proposed by Kass, Witkin & Terzopoulos [16] and many modified methods have been developed since then. Amini, Weymouth & Jian [17] proposed an algorithm to find the minimum of an energy function using dynamic programming. Their

algorithm does not need to calculate the high order differentials and is easy to give a discrete implementation. Lai & Chin [18] proposed a global contour model that could effectively describe both global and local deformations by combining a stable shape matrix method with a Markov random field approach. A line search strategy was presented that encompasses a large search region without significantly increasing the search time.

In general, there are three critical issues for a successful active contour algorithm: iterative convergence, automatic parameter selection and computational complexity. In the aforementioned algorithms, only the intensity information was used. In order to detect and rapidly separate the dynamic and deformable objects from the scene, both motion and shape information will be utilized in our active contour method.

1.2. Overview of our approach

Our goal the generation of realistic 2D/3D panoramas from video sequences with more general motion of a hand-held video camera. The construction of a layered 3D panorama from a vibrating translating camera previously has been reported in [19]. In this paper a new approach is proposed to automatically build a Dynamic and Multi-resolution 360° Panorama (DMP), with good image quality, from a video sequence taken by a hand-held camera undergoing 3D rotation, zooming, and small translations. It should be noted that this is often the case for the general manual operation of a video camera in a video program. For purposes of a realistic virtual environment, this requirement can be easily satisfied. Though the description of the DMP construction algorithms in this paper is mostly directed towards a scenario of virtual environment modeling, the same algorithms with slight modifications can be directly used in video analysis and coding, and in video surveillance.

In this paper, a cylindrical panorama is generated by mosaicing the image sequence captured by a camera with a 360° panning angle as the dominant motion, but in the presence of uncontrollable minor tilt, roll and translation movements. A multi-resolution representation is built for the more interesting areas by using standard camera zoom. Dynamic objects in the scene are detected and extracted using both motion and shape cues, and they are represented separately from the background panorama. The DMP construction method is fast, robust and automatic; the computational performance of image registration is approximately 1 pair of frames per second on a 266 MHz PC without the speedup by available hardware accelerations of MMX coding. No camera calibration, feature extraction, image segmentation or exhaustive iterative processing is needed. An experimental system has been built and can be easily used by a non-expert.

This paper is organized as follows. In Section 2 the inter-frame motion model is derived and a pyramid-based motion detection algorithm is described. This section also explains why a simple 2D

rigid transformation model can result in fine mosaicing of 360-degree panoramas. In Section 3 the image mosaicing and rectification algorithm is presented in detail. The important issues of motion parameter estimation and refinement, as well as automatic stitching of the full-view panorama, are discussed in this section. The algorithm for moving object detection and segmentation from the background is presented in Section 4. Interesting results involving the movement of a walking person in a single mosaicing frame is shown. Section 5 describes how to distinguish the zoomed frames in the image sequence, and how to build the multi-resolution representation for the selected "interesting" regions. Experimental systems of DMP building and rendering are presented in Section 6 and a brief conclusion and discussion are given in the last section.

2. Inter-Frame Motion Model

Let us make a basic assumption that the scene is static and all motions in the image are caused by the movement of the camera. The independent motion of other objects in the scene will be considered in Section 3 and Section 4. A coordinate system *XYZ* is attached to the moving camera; the origin O is the optical center of the camera (Fig. 1). *UV* is the image coordinate system whose origin is the intersection of the optical axis with the image plane. The camera motion has 6 degrees of freedom: three translation components and three rotation components. Since we use the camera as the reference coordinate system an alternative view is that the scene being viewed moves with 6 degree of freedom. Considering only an inter-frame case, we represent three rotational angles (roll, tilt and pan) by (α, β, γ) and three translation components by $\mathbf{T}=(T_x, T_y, T_z)^t$.

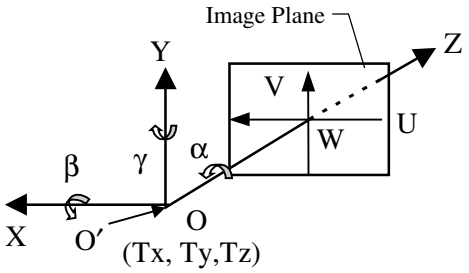


Fig. 1. Coordinate systems of the camera and the image

With current frame at time t and the reference frame at the previous time t', a 3D point $\mathbf{X} = (x, y, z)^t$ with image coordinates $\mathbf{u} = (u, v, 1)^t$ at time t will have moved from point $\mathbf{X'}=(x', y', z')^t$ in the reference time t', with the image point $\mathbf{u'} = (u', v', 1)^t$. The relation between the 3D coordinates is

$$\mathbf{X}' = \mathbf{RX} + \mathbf{T}$$

where $\boldsymbol{R}$ is the rotation matrix

$$\boldsymbol{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \approx \begin{bmatrix} 1 & \alpha & -\gamma \\ -\alpha & 1 & \beta \\ \gamma & -\beta & 1 \end{bmatrix} \tag{1}$$

and it can be approximated by the right-hand matrix for the rotation of successive frames. Suppose the camera focal length f is f' before the motion. Under a pinhole camera model

$$\begin{pmatrix} wu \\ wv \\ w \end{pmatrix} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \end{pmatrix}$$

we have

$$w' \begin{bmatrix} u' \\ v' \\ f' \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} + T_x / z \\ r_{21} & r_{22} & r_{23} + T_y / z \\ r_{31} & r_{32} & r_{33} + T_z / z \end{bmatrix} \begin{bmatrix} u \\ v \\ f \end{bmatrix} \tag{2}$$

In order to construct the 360° panorama, panning must be the dominant motion of the camera. Under pure 3D rotation (i.e. $\mathbf{T} = \mathbf{0}$) , we have the following homogenous rotation transformation

$$w' \begin{bmatrix} u' \\ v' \\ f' \end{bmatrix} = \boldsymbol{R} \begin{bmatrix} u \\ v \\ f \end{bmatrix} \Rightarrow w' \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{3}$$

which is a special case of planar projective transformation. With four pairs of matched points in two successive images, eight parameters (A, B, C, D, E, F, G, H) can be solved from equation (3), and two images can be registered exactly to generate a planar mosaic with a larger field of view. However, the field of view is constrained to be less than 180° since the points in the direction of ±90° from the optical axis of the reference frame are mapped to infinity in the planar mosaic. A full-view cylindrical panorama can be constructed by decomposing f, f', α, β and γ from the eight projective parameters [12], but unfortunately the decomposition of the intrinsic and extrinsic parameters is very sensitive to noise. Thus we look for an alternative way to achieve the goal more robustly and efficiently.

2.1. 2D rigid motion model is plausible

If the rotation angle is small, e.g., less than 5 degrees, between the successive frames, equation (2) can be approximated as (referring to equation(1))

$$w'\begin{bmatrix} u' \\ v' \\ f' \end{bmatrix} = \begin{bmatrix} 1 & \alpha & -\gamma + T_x / z \\ -\alpha & 1 & \beta + T_y / z \\ \gamma & -\beta & 1 + T_z / z \end{bmatrix} \begin{bmatrix} u \\ v \\ f \end{bmatrix} \tag{4}$$

or

$$\mathbf{u}' = \mathbf{M}\mathbf{u}, \quad \mathbf{M} = \frac{1}{s}\begin{bmatrix} 1 & \alpha & T_u \\ -\alpha & 1 & T_v \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

where

$$\begin{cases} T_u = -\gamma f + f T_x / z \\ T_v = \beta f + f T_y / z \\ s = (\gamma u - \beta v + f + f T_z / z) / f' \end{cases} \tag{6}$$

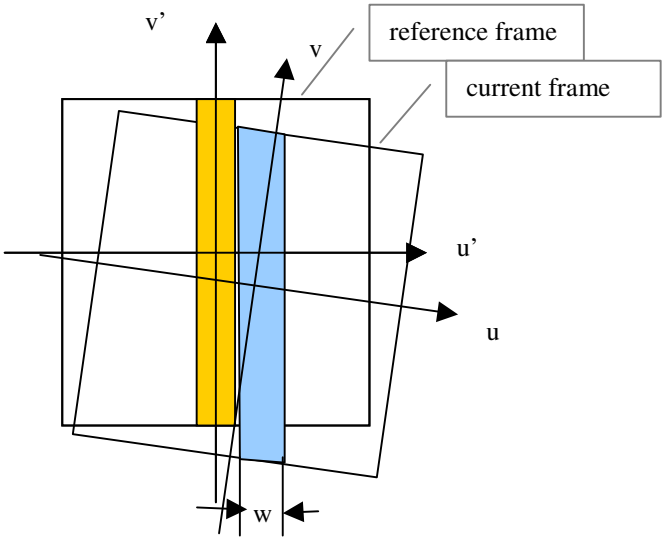


Fig. 2. Mosaicing strips

Under 3D rotation that is dominated by panning motion, possibly with zooming and small translation, we have very small roll α, tilt β and $(T_x / z, T_y / z, T_z / z)$. Therefore a 2D rigid inter-frame motion model can be used

$$\begin{cases} s \cdot u' = u + \alpha v + T_u \\ s \cdot v' = v - \alpha u + T_v \end{cases} \tag{7}$$

where $s \approx f / f'$ is a scale factor associated with zoom, and Z-translation; $(T_u, T_v) \approx (-\gamma f, \beta f)$ is the translation vector representing (pan/X-translation, tilt/Y-translation); and α is the roll angle. This motion model is also plausible if the scene is far away. Given more than 2 pairs of corresponding points between two frames, we can obtain the least square solution of motion parameters, s, T_u, T_v and α, in equation (7). The errors of approximation are especially small for the narrow vertical strip in the center of each image that will be used in our image mosaic algorithm (Fig. 2). This observation can be easily deduced by comparing equation (7) with equation (4) when $\beta \approx 0$, $u \approx 0$, and $(T_x / z, T_y / z, T_z / z) \approx 0$. If the image size is 384×288 and the equivalent focal length of the camera is 384 pixels, numerical analysis shows that when all the three angles are less then 2 degrees, errors are of only 0~2 pixels in the central strip (the width w < 16 pixels in Fig. 2). It satisfies the actual situation for our image sequences where the camera focal length is about 8 mm. More detailed analysis can be found in Appendix 1.

2.2. Inter-frame motion detection algorithm

The inter-frame image displacements are estimated by using a pyramid-based matching algorithm [20]. The hierarchical algorithm consists of four steps: pyramid construction, hierarchical block matching, match evaluation and robust estimation of motion parameters.

Step 1: Generating the pyramids for the current and the reference (preceding) images. For computational efficiency, the final image displacements are only given for non-overlapping image blocks of a given size, say 16×16, in the finest layer (i.e. original image) of the reference frame. The matching process is carried out from coarse to fine resolution layers, starting from a layer with certain image size, e.g. 2 times as large as the matching block size. The list of the blocks is represented by their center coordinates $\{(u'_i, v'_i)$, i=0, ...,B-1$\}$ in the reference frame.

Step 2: Determining the image displacements. For each block in a layer of the reference frame, the absolute difference operation (a simple version of correlation) is carried out in an adaptive search window over the current frame pixel by pixel. Matches with largest correlation values are determined and the one with smallest displacement is selected as the best match. It should be noted that there may be several best matches due to similar patterns within the search window. The initial size of the search window is about half the image size in the first layer, but it is reduced in the finer layers. The motion vectors for these blocks are presented by $\{(\Delta u_i, \Delta v_i)$, i=0, ...,B-1$\}$.

Step 3: Evaluating each match by combining a texture measure with the correlation measurement to produce a confidence value. This step is important because we wish to have blocks with strong

textures and high correlation peaks weighted more. The evaluation of the matching itself is calculated from the normalized absolute difference of each block as

$$d_i = 1.0 - \frac{1}{255 N_w} \sum_{(u',v') \in W(u'_i, v'_i)} | I'(u', v') - I(u' + \Delta u_i, v' + \Delta v_i) |$$

where $W(u'_i, v'_i)$ is the block centered at (u'_i, v'_i), N_w is the pixel number in the block, $I'(.)$ and $I(.)$ are the intensity values (0-255) in the reference and current frames, respectively. The texture is measured as the normalized average magnitude of the gradient image of the reference frame inside a given block i

$$g_i = \frac{1}{g_{\max} N_w} \sum_{(u',v') \in W(u'_i, v'_i)} | (\frac{\partial I(u', v')}{\partial u'}, \frac{\partial I(u', v')}{\partial v'}) |$$

where g_{max} is the maximum value of average magnitudes of all the blocks. The initial weight for the ith match is computed as

$$w_i^{(0)} = \frac{1 - e^{-\kappa d_i g_i}}{1 - e^{-\kappa}} \tag{8}$$

where $\kappa = 8.0$ in our experiments.

Step 4: Robust estimation of inter-frame motion parameters. We use a weighted least mean square (WLMS) method to iteratively estimate the inter-frame motion parameters $\theta = (T_u, T_v, \alpha, s)$ in equation (7). This will severe as the inner iterative loop in the mosaicking process described in the next section.

The objective function is

$$J = \min \sum_i w_i^{(k)} (r_i^{(k)})^2, \ (r_i^{(k)})^2 = | \mathbf{u}'_i - \theta^{(k)} (\mathbf{u}_i) |^2 \tag{9}$$

where $\mathbf{u}'_i = (u'_i, v'_i)^t$, $\mathbf{u}_i = (u'_i + \Delta u_i, v'_i + \Delta v_i)^t$, i = 0,..., B-1, and the weight updating function is

$$w_i^{(k+1)} = \frac{w_i^{(0)}}{1 + (r_i^{(k)} / \rho)^2} \tag{10}$$

where the scale factor ρ is estimated as [21,22]

$$\rho = \underset{i}{median}(| r_i^{(k)} |) * 1.4826 \tag{11}$$

assuming that the residuals can be modeled as a noisy Gaussian distribution (residuals for the non-dominant components are the outliers). It has been pointed out in [22] that a median-based estimate has excellent resistance to outliers. The iterative algorithm is given as follows.

---

(1): Initialize : k=0, $\theta^{(-1)} = (0,0,0,0)$.

(2): Find $\theta^{(k)}$ using WLMS method.

(3). Compute the distance $\Delta\theta^{(k)} = |\theta^{(k)} - \theta^{(k-1)}|$, and estimate the scale factor ρ based on the current residuals.

(4). If $|\frac{\Delta\theta^{(k)}}{\theta^{(k)}}| < \varepsilon$ (e.g. $1.0e^{-3}$), or $\rho < \varepsilon$, or iterating count k > MaxK (e.g. 20), then stop; else update the weights $w_i^{(k)}$, assign k = k+1, and then go to step (2).

The final result from this algorithm is the dominant motion of the points that satisfy the 2D rigid motion. Those points that do not satisfy the motion model – e.g. those on an independent moving object or mismatched - are treated as outliers by automatic reduction in the weights in equation (10). Interested readers can find the difference between our approach and the approach in [22] in the objective function. Our approach is based on residuals between the motion data and the parameter model, rather than intensity difference between two images. A linear method is used instead of a nonlinear iteration method (e.g. Gauss-Newton method [22]) so the computation is very efficient. Instead of directly applying the Geman-McLure function as in [22], the weight function in our approach combines the measures of block match reliability and the data-model difference.

3. Image Mosaicing and Rectification

The relation between two frames from pure rigid 3D rotation is a strict planar projective transformation. However, if we use planar reprojection, the field of view is limited to be less than 180 degrees. In an initial study, we first utilized a direct linear method similar to that in [12] to estimate camera parameters from projective transformation between two frames. The parameters include relative focal length, nodal point, aspect ratio, and the three inter-frame rotational angles of the camera. Theoretically it would be elegant if a cylindrical panorama can be constructed after the focal length and the three rotation angles have been decomposed. However, experimental analysis has shown that this decomposition is very sensitive to image noise and accuracy of the recovered motion

parameters. Since the motion we consider in our domain is not a pure rotation, which make this difficult problem even harder, we adopt an alternative approach when the camera panning is the dominant motion and the pan covers more than 360° around the viewpoint. The algorithm consists of the following three steps: motion estimation, image mosaicing and cylindrical un-warping.

3.1 Estimating and Refining 2D rigid transformation between two successive frames

This step consists of two embedded iteration cycles. The first (inner) iteration cycle is robust motion estimation (Section 2.2) based on the current motion vectors from image matches. The 2D rigid transformation between two successive frames is estimated using an iterative weighted least mean square method. Notice that this iterative process is only carried out on the current motion vectors without re-calculating them from the original images. The re-weighting process accounts for moving objects and other mismatches that are not consistent with the estimated rigid motion model.

The second (outer) iteration cycle is for match correction and refinement. After warping the current frame using the calculated motion parameters, the difference between the warped image and the reference image provides residual errors for the motion model. If the residual is large then the residual motion displacements are estimated between the warped frame and the reference frame, a match correction or a refinement is needed. When motion parameters are significantly different from the averages of the previous several frames, then a mismatch may occur. In this case, the initial inter-frame motion parameters are assigned as the average of the previous several frames. Given that our goal for image registration is to create an image mosaic using only a small portion of the full frame, the weight function employed for the image difference is a 1D Gaussian function

$$h(u,v) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{u^2}{2\pi\sigma^2}} \tag{12}$$

which favors those points near the center scan-lines of the frames that will be used in the mosaic images (Fig.2). With the initial motion vectors of each block from the given initial inter-frame motion parameters, the match process will start from a suitable intermediate layer in which the initial displacements are detectable.

Even if no mismatch occurs, the refinement process is needed when the rotation angle α is large (notice that we use α instead of sin α in our motion estimation). The refinement can be performed by iteratively warping the current image and re-matching the warped image with the reference image. We emphasize that a transform matrix $\boldsymbol{M}_t$ is used to warp the current image t as

$$\mathbf{u}_t' \approx \mathbf{M}_t \mathbf{u}_t \tag{13}$$

where

$$\mathbf{M}_t = \frac{1}{s}\begin{pmatrix} \cos\alpha & -\sin\alpha & T_u \\ \sin\alpha & \cos\alpha & T_v \\ 0 & 0 & 1 \end{pmatrix} \tag{14}$$

even if we still use equation (7) to estimate the motion parameters $\theta^{(m)} = (T_u, T_v, \alpha, s)|_m$, where (m) denotes the iteration count, so that errors will be reduced with decreasing residual rotating angles. The initial motion parameters $\theta^{(0)}$ for refinement are from the initial match, while the initial weighting function is modified as

$$w_i^{(m,0)} = h(\mathbf{u}_i)\frac{1-e^{-\kappa d_i^{(m)} g_i}}{1-e^{-\kappa}} \tag{15}$$

The warping in the *m*th iteration can be expressed by

$$\mathbf{u}_i^{(m)} = \mathbf{M}_t^{(m-1)}\mathbf{u}_i^{(m-1)}, \mathbf{M}_t^{(m)} \Leftrightarrow \theta^{(m)}, m = 1,... \tag{16}$$

while the objective function is modified as

$$J_m = \min \sum_i w_i^{(m,k)} \left| \mathbf{u}'_i - \theta^{(m,k)}(\mathbf{u}_i^{(m)}) \right|^2, \begin{vmatrix} m = 1,... \\ k = 0,... \end{vmatrix} \tag{17}$$

The final transformation matrix for the current frame t is

$$\mathbf{M}_t = \prod_m \mathbf{M}_t^{(m)} \tag{18}$$

Since the residual motion displacements are reduced, the probabilities of mismatches will be reduced hence the matching results will be improved. Experiments show that about two match cycles after rectification can achieve rather fine registration results.

3.2. Mosaicing the image frame by frame

A frame (e.g. the first frame) is selected as the reference frame for the mosaic process, and the accumulating transformation parameters between each frame and this reference frame are calculated as

$$\Theta_I^{(t)} \Leftrightarrow \mathbf{P}_t = \prod_{j=0}^{t} \mathbf{M}_j = \mathbf{M}_t \mathbf{P}_{t-1}, t = 1,..., F;\ \mathbf{P}_0 = \mathbf{I} \tag{19}$$

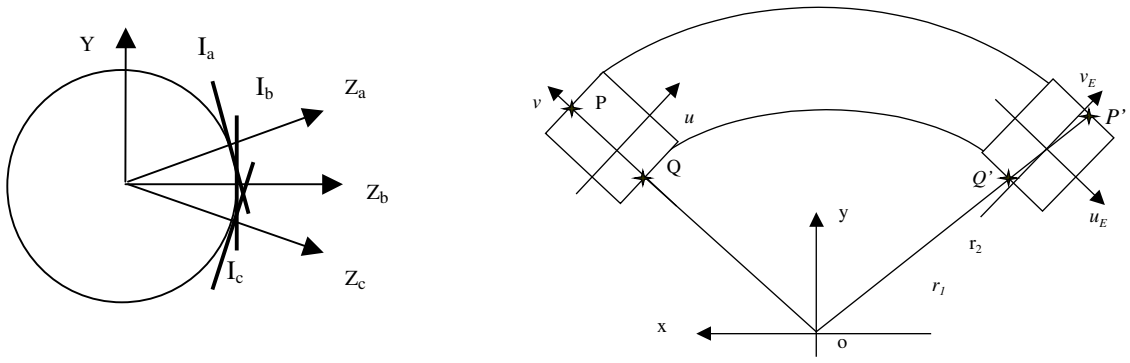
Then images are warped and pasted frame by frame onto the final mosaic using the following transformation

$$\mathbf{u} = \mathbf{P}_t \mathbf{u}_t \tag{20}$$

where $\mathbf{u} = (u,v,1)^t$ is the coordinate in the mosaic coordinate system, i.e. frame t =0, and $\mathbf{u_t} = (u_t,v_t,1)^t$ in the current frame (i.e. time t). If only one narrow vertical strip in the center of each frame is utilized, a 2D rigid transformation is sufficient to merge the successive frames. Moreover 2D rigid mosaicing approximately maps the image to an "unfolded" conic surface, or sometimes an "unfolded" cylindrical surface, depending on the orientation of the optical axis (Fig. 3). The principle behind the conic mosaicing can be explained as follows. Suppose the central strip is represented in spherical coordinates. Then the 2D rigid transformation in equation (13) more closely describes the 3D rotation and zoom of the camera, even though error is introduced by the approximation of the circular arc by a planar strip. If the roll and tilt angles are significantly smaller than the pan angle, then this error is small since the distortion is mostly in the vertical direction (see also Appendix 1). It also implies that the actual mosaic is an unfolded conic surface since the strip is planar. A true cylindrical panorama can be obtained only if the optical axis is strictly horizontal (I_b in Fig.3 (a)). The cone is upward if the optical axis of the reference frame is slightly downward looking and vice versa (I_c and I_a in Fig.3 (a)).

3.3 Rectifying the unfolded conic mosaic to an unfolded 360° cylindrical panorama

Rectifying the unfolded conic mosaic to an unfolded 360° cylindrical panorama can be achieved by finding the correspondence of a (virtual) vertical edge in the head and tail of the conic mosaic. The correspondence is established automatically by matching the possible "connecting" frames in the image sequence with the first frame through the same pyramid-based matching strategy and selecting the frame with minimum difference with the first frame. To account for the illumination changes between the connecting head and tail frames, histogram specification from the frame in consideration to the first frame is performed. After the angle range, and the radii of inner and outer arcs of the unfolded cone are computed from the head-tail match, the re-projection of the conic mosaic to the cylindrical panorama can be determined (see Fig. 3(b)).



(a) spherical and conic representation (b) unfolded conic mosaicing and rectification

Fig. 3. The strip-mosaicing geometry

3.3.1. Head-tail stitching

If the reference frame coordinate $(u, v,1)^t$ is chosen as the first frame coordinate $(u_1, v_1,1)^t$, then the relationship between the last frame $(u_{E,} v_E,1)^t$ and the reference frame can be obtained by successive rigid transformations from the first frame to the last frame from equation (19) (Fig.3 (b)):

$$\begin{pmatrix} u_E \\ v_E \\ 1 \end{pmatrix} = \mathbf{P}_E \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \tag{21}$$

The relation between the first (home) and the last (re-homing) frame derived from their match can be expressed as

$$\begin{pmatrix} u_E \\ v_E \\ 1 \end{pmatrix} = \mathbf{M}_{E1} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix}_1 \tag{22}$$

where $\mathbf{M}_{E1}$ is defined in equation (14). For a point $(u_1, v_1)^T$ in the first frame, its coordinates in the panorama frame are Q = $(u, v,1)^t = (u_1, v_1,1)^t$. For its corresponding point in the last frame $(u_E, v_E,1)^t$, the coordinates Q′ in the panorama frame can be calculated as

$$\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix}^T = \mathbf{P}_E^{-1} \mathbf{M}_{E1} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} \tag{23}$$

By finding a vertical line segment PQ across the first frame, the corresponding segment $P'Q'$ can be determined by using equation (23). This "vertical" line can be selected as the central column of the first frame when the first frame is not bevel (see Fig.3(b)). The center of the circles of the unfolded conic mosaic is the intersection point of PQ and $P'Q'$. For simplicity we choose the new coordinate system *xoy* with the origin *o* at the center of the circles. Then the angle range of the unfolded cone is

$$\theta = \theta_0 - \theta_1 \tag{24}$$

where

$$\theta_0 = \tan^{-1}\left(\frac{y_P - y_Q}{x_P - x_Q}\right),\ \theta_1 = \tan^{-1}\left(\frac{y_{P'} - y_{Q'}}{x_{P'} - x_{Q'}}\right) \tag{25}$$

Due to the change of the camera's focal length and accumulating errors, we could have a "deformed" cone with different radii in the head and tail of the conic mosaic, e.g.

$$R = R_P - R_Q \geq R' = R_{P'} - R_{Q'} \tag{26}$$

The height and the length of rectified cylindrical panorama are set as

$$R = R_P - R_Q,\ L = R_P\theta \tag{27}$$

So the relation between the conic mosaic (x,y) and the cylindrical mosaic (r,l) is

$$(x, y) = (R_{rl}\cos\theta_l, R_{rl}\sin\theta_l) \tag{28}$$

where

$$\begin{aligned} \theta_l &= \theta_0 + \frac{l}{R_P} \\ R_{rl} &= R_Q + \frac{l}{L}(R_{Q'} - R_Q) + r + \frac{lr}{LR}(R'-R) \end{aligned} \tag{29}$$

and $l = 0,\ldots,L$ (left to right)*;* $r = 0, \ldots, R$ (bottom-up). This process also eliminates the accumulating errors from frame-to-frame registration.

(a) the current frame (Frame no. 245)

(b) the reference frame (Frame no. 0)

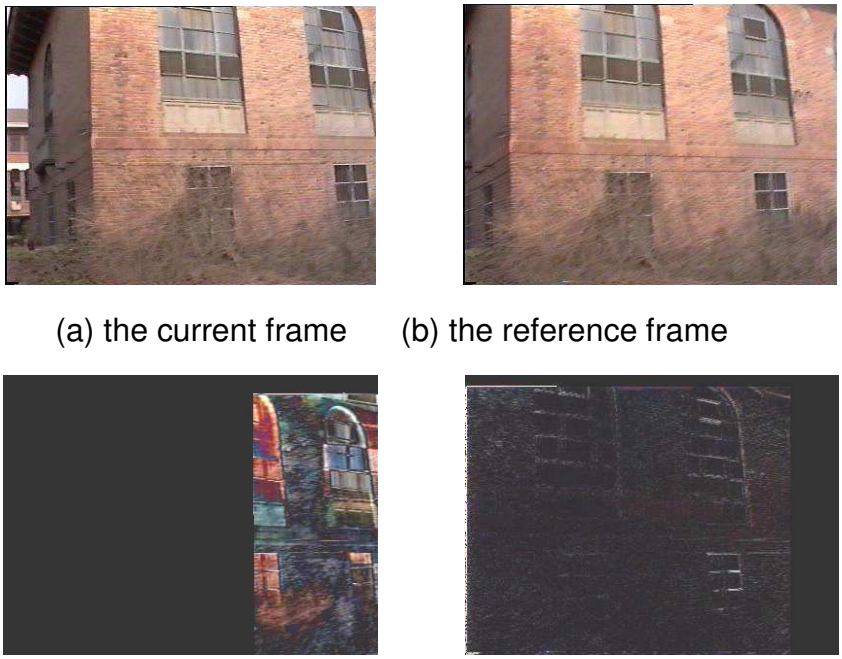
(c) difference image of inital matching

(d) difference image after re-matching

Fig. 4. A match refinement example from the 246-frame original image sequence (panning from right to left): the first and the last frame

3.4. Experimental results

It should be emphasized that no camera calibration or intrinsic camera parameters are needed, and the algorithm is completely automatic. Fig. 4 shows the matching process of the head and the tail frame from a 246-frame image sequence of the Library scene. The motion parameters from the initial match are $T_u = 49.07$, $T_v = 13.72$, $s = 1.00$ and $\alpha = -0.00$, while the motion parameters resulting from the second match are $T_u = 48.05$, $T_v = 13.74$, $s = 1.00$ and $\alpha = -0.00$ (These number are truncated after the second numbers after the decimal point, so –0.00 means a very small negative value). The second set of parameters result in a better registration result, which can be observed from the edges in the difference images between the two frames, especially in the center strip of the image which will be used for mosaic, e.g., the white lamp in front of the pine tree and the door near that tree.



(a) the current frame (b) the reference frame

(c) difference image of inital matching (d) difference image after re-matching

Fig. 5. A match correction example from the 246-frame original image sequence

Gross match errors can be detected and corrected by using the smoothness constraint on interframe motion. If the displacement for the current frame is obviously much larger or smaller than the previous ones (the average of several preceding frames), and the frame difference is quite large, a mismatch will be assumed. Then the previous average motion parameters are used as the initial estimates for the current match. Fig. 5 shows an example. The initial match obtained wrong motion

parameters $T_u = 206.35$, $T_v = 19.84$, $s = 0.99$, $\alpha = -0.00$ due to repetitive patterns and the large search window of the motion estimation algorithm (Note that the search range is the entire image at the top of hierarchical match process at the beginning). The motion parameters are far from the average of the preceding values, and the sum of absolute frame difference (SAD, average of the R,G and B bands) is 17533 for an overlapping region of 1/3 of the image size. By using the match-correction technique, the new motion parameters are $t_u = -49.25$, $t_v = 11.01$, $s = 1.00$, $\alpha = 0.00$ and the SAD reduced to 3950. Form the difference images in Fig. 5 (c) and (d) , the improvement is quite clear.

Fig 6(a). Unfolded conic mosaic (13% display scale). The original color image is 3806 x 773x24 bits. Notice the curved and uneven boundary created by the up-tilted angle and unstabilized motion of the hand-held camera.

Fig.6(b) Unfolded 360-degree cylindrical panorama (27% display scale; 1st row : 0~180°; 2nd row: 180°~360°). The original true-color image is 3494x323 x24 bits.

Fig. 6(a) and Fig. 6(b) show the panoramas before and after cylindrical rectification and head-tail stitching. The original image sequence has 246 frames of 384 x 288 color images, so the average

panning angle between two frames is about 1.5 degrees, which satisfies the small rotation assumption in equation (4). The size of the rectified cylindrical panorama is 3494x323. If the compression ratio of the panorama in JPEG format is 20:1, the total compression ratio between the JPEG panorama and the original image sequence is about 500. Moreover new images of arbitrary viewing angles can be synthesized interactively, which is essential for applications of virtual reality and content-based video manipulation. When there are moving objects in the scene, median values of the corresponding points in multiple frames are used to generate the conic panoramic background. The resulting image is somewhat blurred since a wider strip is used in each frame (see Fig. 9b). The moving object extraction is presented in the next section.

4. Moving Object Extraction

As the mosaic is being constructed, each difference image between the warped successive frames is analyzed. The region in the panorama that corresponds to that of the current frame containing large residuals in the difference image is labeled as a "dynamic hot spot". The dynamic sub-images of objects are coded separately, for example, using MPEG format.

In practice, a difference image is calculated from three successive images for robustness. Then region analysis is carried out to determine those regions that may contain moving objects. In order to achieve the best figure-ground separation, the contour of the moving object in each region needs to be extracted. We apply an active contour model to extract contours from a noisy image [16, 17,18]. The basic idea of the active contour algorithm is to constrain the contour of an object onto a controllable continuous spline. The task is to minimize an energy function that takes into account both input image information and constraints on the continuity of the contour. Our modified active contour algorithm uses both motion and gradient cues of the images, and the control parameters are adaptively adjusted according to objects in the current image.

The algorithm consists of the following four steps:

(1) A difference image is calculated from the current image and its predecessor and successor frames in the sequence. Regions with large residuals are detected through a region grouping algorithm. Then, in each region, the difference value is thresholded to a binary image, gaps and holes are filled using a morphology-based method, and large scale grouping is used (if necessary) to generate a single mask for each moving object; this mask is then used as an initial contour in the following step.

(2) Evenly spaced control points are placed on the initial contour, and curvatures at the control points are estimated. The control points are evenly spaced and the space is adaptively changed according to the size of the initial contour. Then the parameters used in the energy function are automatically assigned according to the point spaces and the curvatures.

(3) The energy function is minimized using a dynamic programming approach to obtain the resulting contour [17, 18].

(4) Each dynamic object is separated along its contour from the original frame and is labeled on the corresponding location of the panorama, and the dynamic sub-images of objects are represented individually.

(a) original image (b) moving object

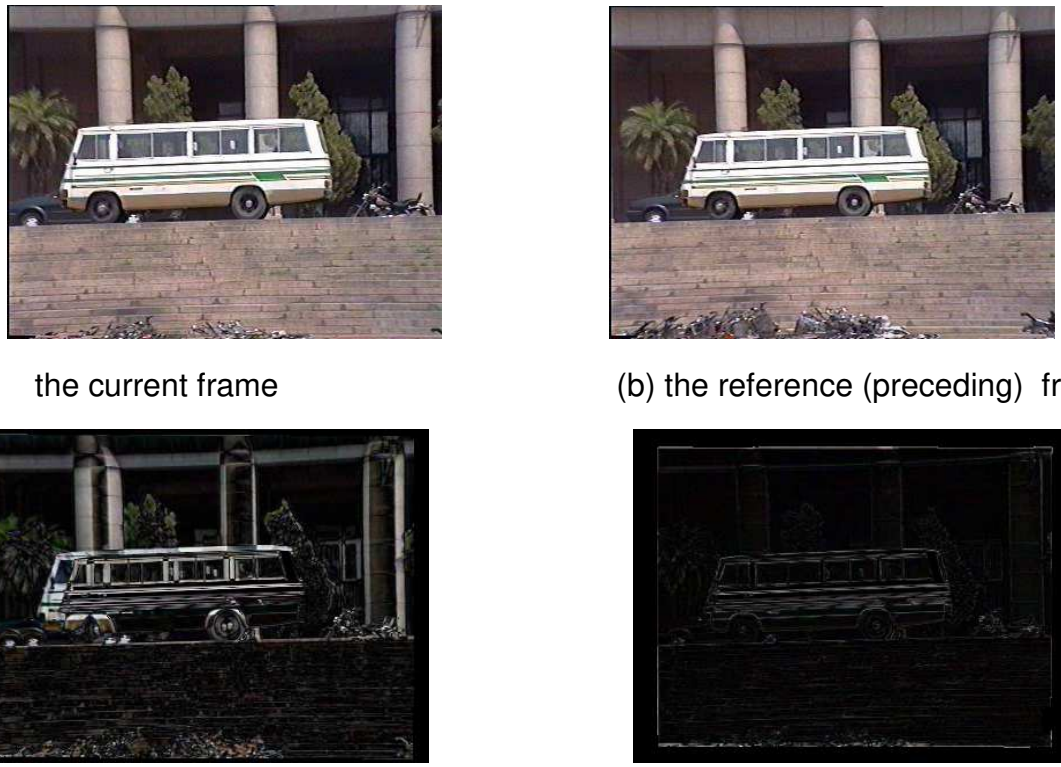
(c). dynamic mosaicing (part of the cylindrical panorama)

Fig. 7. Moving object detection and separation

Fig. 7 (a) and (b) show an original image and the extracted object (a person). Fig. 7(c) shows the dynamic mosaic with the walking person pasted onto the mosaic every ten frames.

5. Multi-Resolution Representation

In VR applications we want the ability to zoom and pan (under controlled motions) to enhance the visual realism; in image coding we need to handle the video sequence with zoom as well as pan. Therefore, we introduce a multi-resolution representation for each user specified "interesting" portion of the panorama. Each of those regions on the panorama is labeled as a "zooming hot spot". This is similar to the sparse pyramid in [15], but our representation is more purposive and compact. The representation is constructed by physically zooming the camera when the more interesting regions of the scene are viewed. The zoomed frames are separated automatically from the original panning and zooming image sequence by observing the accumulating scaling (zoom) factor. An automatic registration between two zoomed frames is achieved in a manner similar to that for the panned frames, but the following step is to select representative frames as the components of a multi-resolution representation (instead of mosaicing the frames). It should be noted here that it is more difficult to accurately assess similarity in the zooming case than in the panning case, especially when the scale change is large between successive frames (e.g. $s > 1.1$), since the scales of the match blocks are not the same in the two images. In this case re-match processing after warping (i.e. re-zooming) is vital for the accurate estimation of the scale parameter.



(a) the current frame (b) the reference (preceding) frame

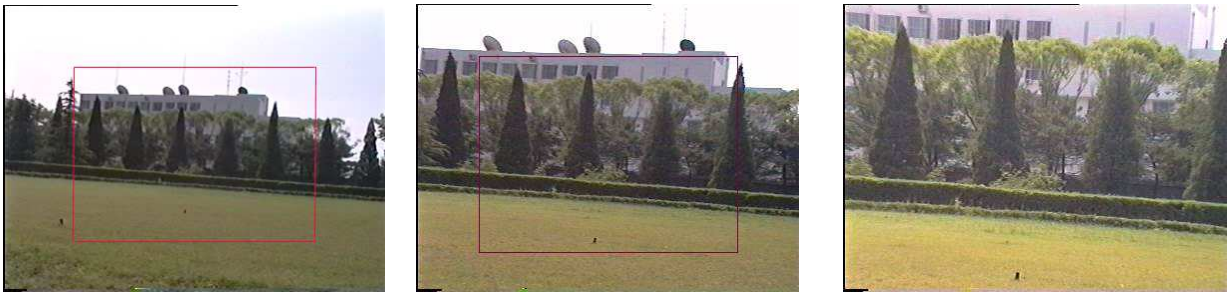
(c) the difference image of initial matching (d) the difference image after re-matching

Fig. 8. Iteration matching after image warping (zooming).

Frame 199 (cars parked at the road side); Frame 149 (many small moving objects); Frame 90 (large moving object)

(a) Several frames of the Main Building sequence with moving objects in the scene (the camera is panning from right to left)

(b) Cylindrical panorama after eliminating the moving objects (image size:3498x303). Notice that the zoom factor is changed.



(c) Three selected zooming frames. This “interesting” area is to the right of the first part of the panorama

Fig. 9. Multi-resolution panorama. The original image sequence has 561 frames, which consists of 3 zooming segments among the panning sequence. There are many moving objects (persons, bicycles) in the scene. Notice that most of the moving objects and noises (e.g. horizontal lines in frame 199) have been successfully filtered out.

Fig. 8 shows a matching example from the zoomed segment of the Main Building sequence shown in Fig. 9. The motion parameters from the initial matching process are $T_u = 7.29$, $T_v = -0.52$, $s = 1.03$ and $\alpha = 0.00$, while the motion parameters from the second (final) matching process are $T_u = 0.34$, $T_v = -0.99$, $s = 1.12$ and $\alpha = -0.00$. The second set of parameters results in a much better registration of the frames, as can be seen by comparing Fig.8(c) and Fig. 8(d). The zoom factor between Fig. 8a and 8b is 1.12. The reason for successful match is that every iteration adjusts the scale factor to approach to the real one. Fig. 9(c) shows the selected zooming frames with 1.5 zooming factors between two selected frames for the Main Building image sequence. The rectangle in each frame indicates the sub-region that corresponds to the next selected frame.

6. Modeling and Rendering System

We have built experimental systems for both DMP modeling and DMP rendering. The DMP modeling system was built using Borland Builder C++ 3.0, and an interface is shown in Fig. 10. On the top of the windows there are menu and buttons, and two image windows and a text windows are shown in the interface. The two images may be the two successive frames, or the optical flow superimposed on the left image, and the difference image shown in the left. The final mosaic results are shown on a separate scrollable window.

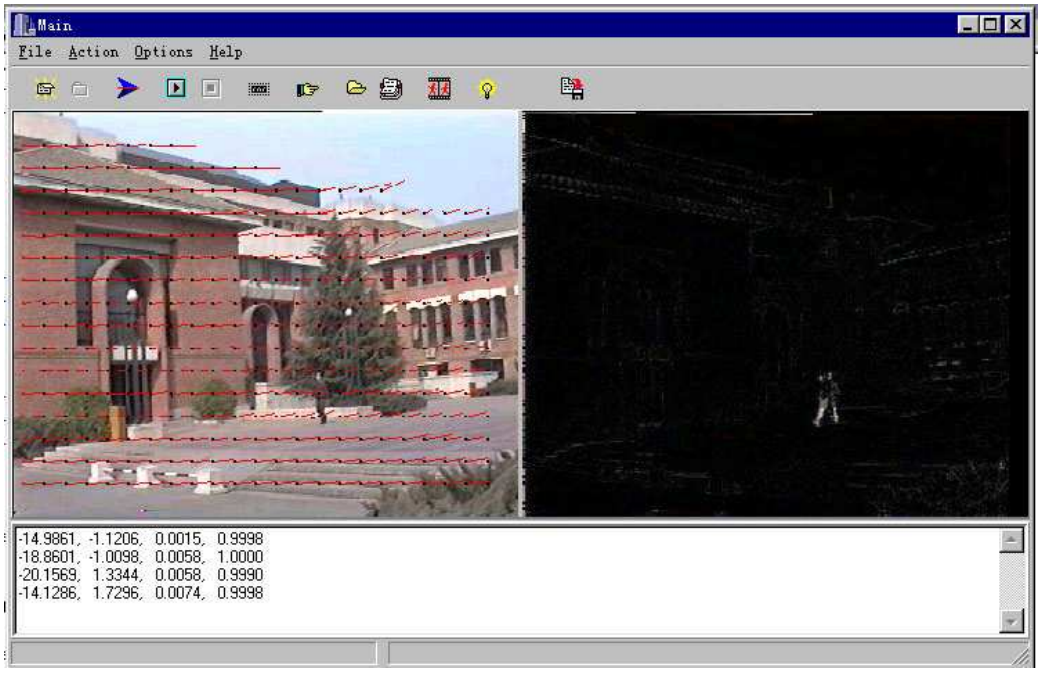


Fig. 10. Interface of DMP modeling system

An experimental system of DMP rendering was built using Netscape Plug-In and Java Interface. Fig. 11 shown an example of image-based VR browsing of the Tsinghua Library. On the left of the window is a layout of the locations that have panoramic images, both indoors and outdoors. The left window is a user-controled view window to allow wandering in the virtual environment using a mouse. We defined several kinds of hot spots on the panoramic images: a Hyperlink to a Web Site, a media to a audio/video clip, and a Travel link to other panorama. Currently, traveling between two panoramas is realized by mtaching the images of the multi-resolution images in the two mosaics.

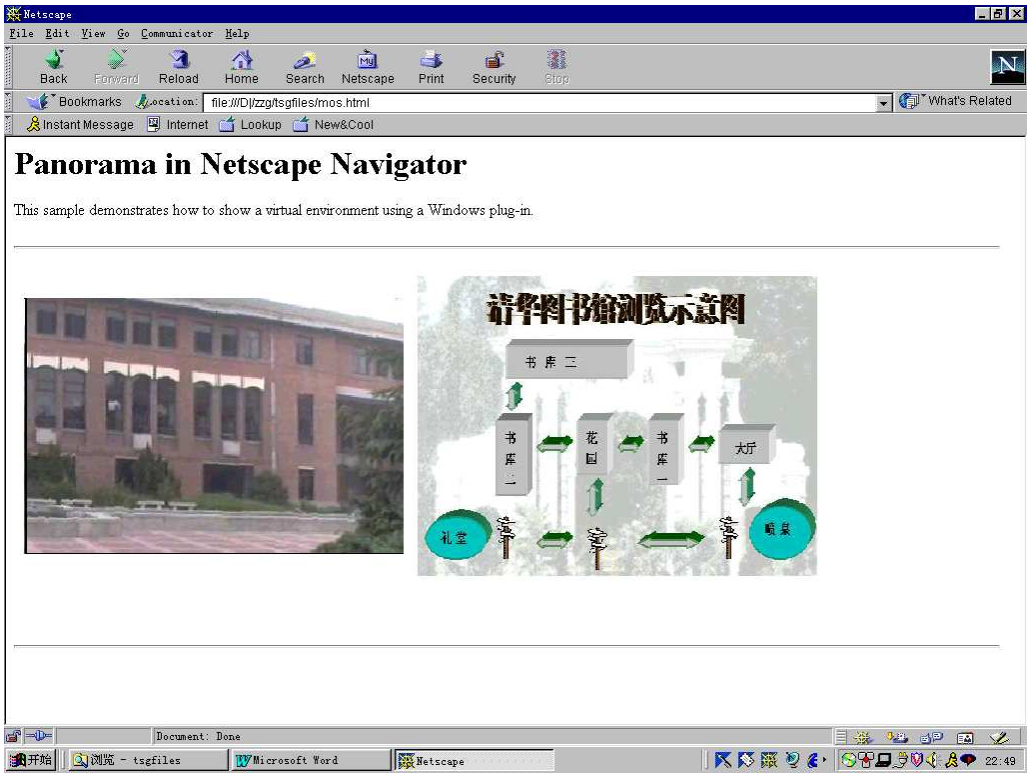


Fig. 11. Interface of DMP rendering system

7. Conclusion and Discussion

The construction of the DMP (Dynamic and Multi-resolution Panorama) is fast, robust and automatic. The processing rate is about 1 frame per second for 384×288 color images using a Pentium

II/ 266 MHz PC. A factor of 2 speed-up can be expected by algorithm optimization and MMX utilization. Besides the most obvious applications such as virtual reality scene modeling and very low bit rate video coding, the DMP and the algorithm is also useful in other applications such as surveillance, change detection, video enhancement, indexing and manipulation. Ongoing and future work includes the following topics.

1. *Panoramic view morphing.* Seitz and Dyer[23] showed that two basis views of a static scene uniquely determine the set of views on the line between their optical centers when a visibility constraint is satisfied, and then a simple view morphing algorithm can generate the new images from the set of views. We are extending this method to a discrete set of panoramic images to generate scene appearance for a continuous range of viewpoints. With a suitable view planning strategy for collecting the discrete panoramic samples, a panoramic view morphing method can generate the scene appearance with arbitrary viewpoints and viewing directions.

2. *Layered panorama.* By combining the layered representation [19,24] with the DMP, we are generating a Layered and Multi-resolution Panorama (LAMP) for 3D scene modeling. This goal can be reached by using a panoramic stereo vision method, which generates a panoramic disparity map from two panoramic images constructed at two calibrated viewpoints. It can be also viewed as the merging of the VRML model with the MPEG content-based video coding model.

3. Air-ground site modeling. By combining algorithms of the site modeling from aerial images [25] and scene modeling from ground image sequences, we can generate a hierarchical site model for the city and the scene. This approach will be useful for VR and surveillance applications because you can fly over and walk-through the scene from air to ground, overcoming the difficulty in viewing aerial reconstruction from ground level view points.

4. *Real-time panoramic scene modeling and monitoring.* We are considering a virtual stereo vision system using two panoramic annular lens (PAL) cameras mounted on two separate platforms [6]. Work is currently underway to understand the geometric model and to find calibration algorithms of the PAL camera system. Both cylindrical and perspective images can be generated from the PAL image in real-time. Although the image sharpness and resolution is not as good as that of typical CCD cameras, the real-time panoramic generation of the PAL camera is very attractive for fast scene modeling. By using a high-resolution camera sensor, this drawback can be somewhat compensated for. The PAL camera system is especially useful for virtual video conferencing and real-time wide FOV video moving object tracking. Moreover, the combination of the full view property of the PAL camera and the high resolution of the narrow view camera will lead to an algorithm for building the dynamic and multi-resolution panorama more efficiently.

Acknowledgments

This work is supported by the China High Technology Program under contract No. 863-306-ZD-10-22 and partially by DARPA under contract No. F30602-97-2-0032. The basic algorithm of motion detection and estimation was provided by Dr. Yudong Yang of Tsinghua University, and the experimental modeling system is realized with the assistance of Mr. Heng Luo, and Mr. Qiang Wang at Tsinghua University. Thanks are also given to Dr. Howard Schultz and Mr. Kristopher Denio at the University of Massachusetts at Amherst for their discussions on panoramic sensing and processing.

References

[1]. Y. Xiong, K. Turkowski, Creating image-based VR using a self-calibrating fisheye lens, *IEEE Proceedings of Computer Vision and Pattern Recognition*, pp. 237-243, Washington, June 1997.

[2]. S. K. Nayar, Omnidirectional video camera. *Prof. DARPA Image Understanding Workshop*, May 1997:235-241

[3]. V. Peri and S. K. Nayar, Generation of perspective and panoramic video from omnidirectional video, *Prof. DARPA Image Understanding Workshop*, May 1997:243-245

[4]. P. Greguass, Panoramic imaging block for three dimensional space, *U.S. Patent 4,566,763* (28 Jan 1986).

[5]. I. Powell, Panoramic lens, *Applied Optics*, vol. 33, no 31, Nov 1994: 7356-7361

[6]. Z. Zhu, E. M. Riseman, A. R. Hanson, Geometrical modeling and real-time vision applications of panoramic annular lens (PAL) camera, *Technical Report TR#99-11*, Computer Science Department, University of Massachusetts Amherst, February, 1999.

[7]. S. E. Chen, QuickTime VR - an image based approach to virtual environment navigation, *Proceedings of SIGGRAPH 95*, pp. 29-38, New York, 1995. ACM.

[8]. S. Mann, R. W. Picard, Video orbit of the projective group: a new perspective on image mosaicing, *Technical Report No.338*, MIT Media Lab Perceptual Computing Section,1995

[9]. L. McMillan and G. Bishop, Plenoptic modeling: an image-based rendering system, *Proceedings of SIGGRAPH 95*, pp. 39-46, New York, 1995. ACM.

[10]. S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, The lumigraph. *Proceedings of SIGGRAPH 96*, pp. 43-54, New York, 1996. ACM.

[11]. M. Levoy and P. Hanrahan. Light Field Rendering. *Proceedings of SIGGRAPH 96*, pp. 31-42, New York, 1996. ACM.

[12]. H.-Y. Shum and R. Szeliski, Panoramic Image Mosaics, *Microsoft Research, Technical Report, MSR-TR-97-23, 1997*

[13]. S. B. Kang, R. Weiss, Characteristics of errors in compositing panoramic images, *IEEE Proceedings of Computer Vision and Pattern Recognition*, pp. 103-109, Washington, June 1997.

[14]. S. Peleg, J. Herman, Panoramic Mosaics by Manifold Projection. *IEEE Proceedings of Computer Vision and Pattern Recognition*, pp. 338-343, Washington, June 1997.

[15]. M. Irani, P. Anandan, S. Hsu, Mosaic based representation of video sequence and their applications, *IEEE Proc ICCV'95*, pp605-611.

[16]. M. Kass, A. Witkin, and D. Terzopoulos, Snakes: Active contour models, *Proceedings of First International Conference on Computer Vision*, London, 1987: pp259-269.

[17]. A. Amini, T. Weymouth, and R. Jain, Using dynamic programming for solving variational problems in vision, *IEEE Trans. PAMI*, vol.12 no.9, 1990: pp855- 867.

[18]. K. F. Lai, R. T. Chin, Deformable Contours: Modeling and Extraction, *IEEE Trans. PAMI*, vol.17 no.11, Nov 1995: pp1084-1089

[19]. Z. Zhu , G. Xu, X. Lin, Constructing 3D natural scene from video sequences with vibrated motions, *Proceedings of the IEEE Virtual Reality Annual International Symposium*, Atlanta, March 14-18,1998:105–112.

[20]. Z. Zhu, G. Xu, Y. Yang, J. S. Jin, "Camera stabilization based on 2.5D motion estimation and inertial motion filtering," IEEE Int. Conf. on Intelligent Vehicles, Oct 28-30, 1998, Stuttgart, Germany.

[21]. P.J. Rousseeuw, A.M. Leroy, *Robust Regression and Outlier Detection*, J. Wiley & Sons, New York, 1987.

[22]. Sawhney H S, Ayer S, Compact representation of videos through dominant and multiple motion estimation", *IEEE Trans. PAMI,* Vol. 18, No. 8, Aug 1996, pp814-830

[23]. S. M. Seitz, C. R. Dyer, Viewing morphing: uniquely predicting scene appearance from basis images, *Prof. DARPA Image Understanding Workshop*, May 1997:881-887.

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

[25]. C. Jaynes, E. M. Riseman, and A. R. Hanson, Building construction from optical and range images, *Proc. ARPA Image Understanding workshop,* 1996: 479-490.

[26]. E. H. Adelson and J. R. Bergen, The plenoptic function and the element of early vision, *Computational Models of Visual Processing*, Chapter 1, eds. M. Landy and J. A. Movshon, The MIT Press, Cambridge, MA, 1991.

Appendix 1. Error analysis

For simplicity, we only consider the case of pure 3D rotation. When $(T_x / z, T_y / z, T_z / z) \approx 0$ and $f = f'$, subtracting equation (4) and (7) yields the following error terms

$$\begin{cases} \delta u \approx \dfrac{u + \alpha v - \gamma f}{\gamma u - \beta v + f}(-\gamma u + \beta v) \\ \delta v \approx \dfrac{-\alpha u + v + \beta f}{\gamma u - \beta v + f}(-\gamma u + \beta v) \end{cases}$$

The errors in pixels are shown in the following table for the edge and central points along the central strip and off the center strip with different rotation angles.

angles \ *(|u|, |v|)* / *(|δu|,|δv|)*	*(0,0)*	*(0,128)*	*(192,128)*
α=β=0, γ=2°	*(0, 0)*	*(0, 0)*	*(4.0, 3.3)*
α=β=γ=2°	*(0, 0)*	*(0.61. 2.03)*	*(6.9, 5.3)*

It is interesting to notice that there are no differences between equation (4) and (7) for the central column if the tilt angle β is zero.