Accepted to **Image & Vision Computing Journal**

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

(Revised - September 14, 2005)

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

Guangyou Xu
Department of Computer Science and Technology
Tsinghua University, Beijing 100084, China

Edward M. Riseman and Allen R. Hanson
Department of Computer Science
University of Massachusetts at Amherst, MA 01003

Contact Information:

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

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

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

Guangyou Xu
Department of Computer Science and Technology
Tsinghua University, Beijing 100084, China

Edward M. Riseman and Allen R. Hanson
Department of Computer Science
University of Massachusetts at Amherst, MA 01003

Abstract

This paper presents a unified approach to automatically build dynamic and multi-resolution 360° panoramic (DMP) representations from image sequences captured by hand-held cameras mainly undertaking rotation and zooming for natural scenes with moving targets. A simple (yet stable) rigid motion model and a closed-loop-based mosaicing algorithm are proposed to generate cylindrical mosaics automatically. Multi-resolution representations are built for interesting areas by separating zooming sub-sequences from a pan/zoom sequence. Moving objects are detected and separated from images based on motion information, and then more accurate contours are extracted using a modified active contour algorithm. The DMP construction method is fast, robust, and automatic, achieving 5 frames per second for image sequences with 384×288 color images on a Pentium III 800 MHz PC. The construction of the DMP representation can be used in virtual reality, video surveillance, and very low bit-rate video coding.

Keywords: *Panoramic representation, video mosaicing, multi-resolution, moving object extraction*

1. Introduction

Panoramic representations of visual scenes have 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. A wide field of view (FOV) lens, e.g. a fish-eye [1] or panoramic lens [2-6], can be a solution for obtaining panoramic views. However, in addition to the high cost of these specially designed image sensors, images obtained by such sensors have substantial distortions, and mapping an entire scene into the limited sensor target of a standard 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 the aforementioned applications for high image resolution. However very few of the current algorithms and systems are able to properly detect and represent foreground objects and to efficiently deal with large zoom effect. This motivates our work described in this paper.

1.1. Overview of our approach

Our overall goal in image-based modeling is to create realistic 2D/3D panoramas from video sequences with the more general motion of a hand-held video camera [7-11]. The construction of layered panoramas and stereo mosaics of 3D scenes from *translating* cameras with constrained 6 DOF motion has been proposed in our previous work [8-11], where motion parallax is large and thus is used to recover the 3D structure of the static scenes. In this paper we deal with video sequences of scenes with *moving objects*, which are taken by hand-held cameras undergoing 3D *rotation (mainly panning)*, *zooming*, and small translation. In this case, the motion parallax, if not zero, can be neglected due to the small translation. A new approach is proposed to automatically build a Dynamic and Multi-resolution 360° Panorama (DMP) from such a video sequence. For applications of image-based modeling and rendering, we can control the camera's motion so this requirement can be easily satisfied. Nevertheless, this is often the case for the operation of a video camera by a cameraman for a video program. Therefore, although the description of the DMP construction algorithm in this paper is mostly directed towards image-based modeling, the same algorithm with slight modifications can be used in video analysis and coding, and also in video surveillance.

The system diagram of our approach (as a road map of this paper) is shown in Figure 1. The input of the algorithm is an image sequence captured by a camera, undertaking continuous panning, zooming in for each interesting spot and then zooming out to the continuous capture of the scene around the viewer, and so on so forth, until the camera rotates a full 360 degrees. The zoom in/out operations might be performed at multiple spots, and the focal lengths could be different before and after each zoom in/out operation. However, we assume that the camera will cover the full 360-degree field of

view around the camera, and the rotation is almost around its nodal point. There are three steps in our algorithm: (1) interframe motion estimation, (2) motion accumulation and classification, and (3) DMP model generation. In the first step, a parameter motion model is estimated between each pair of successive frames. Then in the second step, the interframe motion parameters are accumulated to generate a global transformation between each frame and the reference frame (e.g. the first frame). With these global transformation parameters, those frames with significant and continuous changes of the focal lengths (i.e., both zoom-in and zoom-out frames) are separated for further multi-resolution processing, wile the remaining 360-degree panning sub-sequence is used for panoramic mosaicing.

Figure.1

In the third step, there are three parts in generating the DMP representation: panoramic generation, dynamic object extraction, and multi-resolution representation construction. First, a cylindrical panorama is generated by mosaicing those frames in the panning sub-sequence. The full view "closed-loop" constraint is used for rectifying the mosaic to a cylindrical representation. Meanwhile, as the second part of the processing, dynamic objects in the scene are detected and extracted using both motion and shape cues, and they are filtered out from the background panorama and represented separately. Finally, in the third part, a spare multi-resolution pyramid representation is built for each interesting area using the corresponding zoom-in/zoom-out frames. Because of the continuous nature of the video capture, we can effectively register those multiple-resolution frames with the panoramic mosaic. The DMP construction method is fast, robust and automatic; in fact most of the time is spent on interframe motion estimation. The computational performance of image registration is approximately 1 pair of frames per second on a 266 MHz PC, and up to 5 frames per second (5 Hz) on a Pentium III 800 MHz laptop. No camera calibration is needed, and an experimental system has been built and can be easily used by a non-expert. While all the three steps are important, algorithms for the first step have been well studied. Therefore, we focus on the three parts of the third step. The discussion of the second step will be integrated into the description of the multi-resolution representation.

A *panoramic* representation with *dynamic* and *multi-resolution* capacities has the following benefits. For virtual/virtualized reality applications, it has the advantages of simplicity in rendering (just image warping), photographic quality realism (from real images), and 3D illusion experienced by users (virtual camera panning and zooming with dynamic objects). 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, for a class of video sequences with pan/tilt/zoom camera motion and moving objects. These two

aspects of capabilities could be merged into a more general approach for interactive video (e.g. virtual conferencing), which adds the flexibility of synthesizing images with interactivity, selectivity, and enhanced field of view and resolution, all the while making the data streams of video information within a reasonable bandwidth for video conferencing applications over the Internet.

1.2. Related work

Apple's QuickTime VR [12] captures a 360-degree panoramic image of a scene with a camera panning horizontally from a fixed position. The overlap in images is registered first by the user and then "stitched" together by the software using a best match algorithm. Similarly, in [13] 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. Shum & Szeliski [14] 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 [15] 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. Most of the current panoramic mosaicing software systems available with digital cameras follow these approaches, which lack the capabilities to close the panoramic loop, to represent moving targets, and to represent zoom frames properly. Manifold projection [16] enables the fast creation of low distortion panoramic mosaics under a more general motion than an 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 frames. However, the issues of full-view cylindrical 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, with a few exceptions such as a dynamic mosaic approach proposed by Irani, Anandan & Hsu [17] and the motion panoramas [18] to describe dynamic events. However, the accuracy of the contour of a moving object was not addressed, which is important for synthesis of fine details of dynamic objects on the mosaic representation. In our work we utilized a modified active contour method to extract contours of moving objects. Recent work in dealing with matching images with large zoom factors includes several pieces of work to design and use scale-invariant features [19, 20, 21], but these approaches are usually time-consuming.

This paper is organized as follows. In Section 2 a simple inter-frame motion model is introduced and motion estimation and refinement is discussed for panoramic mosaicing. This section also explains why a simple 2D rigid transformation model can result in fine mosaicing of 360-degree panoramas. Section 3 describes how to separate zoom frames from a pan/zoom image sequence and how to build a multi-resolution representation for the selected "interesting" regions. In Section 4, a closed-loop image mosaicing and rectification algorithm is presented in detail. The algorithm for moving object detection and segmentation from the background is presented in Section 5. Interesting results involving the movement of a walking person in a single mosaicing frame is shown. A brief conclusion and some discussions are given in the last section.

2. Motion Estimation and Refinement for Mosaicing

The exact transformation between two frames from pure rigid 3D rotation should be a 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 [14] 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 of a hand-held camera cannot be guaranteed as a pure rotation, which makes 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. This section introduces the interframe motion modeling and estimation. Then, in Sections 3 and 4 we will discuss zoom-frame handling and closed-loop mosaicing algorithms.

2.1. Interframe motion model

Let us first assume 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 later. A coordinate system *XYZ* is attached to the moving camera; the origin O is the optical center of the camera (Figure 2). *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 equivalency is that the scene being viewed moves with 6 degrees of freedom. Considering only an inter-frame case, we represent three rotational angles (roll, tilt and pan) by (α, β, γ) and then a rotation matrix **R**, and three translation components by $\mathbf{T}=(T_x, T_y, T_z)^t$.

Figure 2

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 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

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

If the rotation angle is small (e.g., less than 5 degrees) between the successive frames, then under a pinhole camera model, we have

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

where

$$\begin{cases} t_u = -\gamma f + fT_x / z \\ t_v = \beta f + fT_y / z \\ s = (\gamma u - \beta v + f + fT_z / z) / f' \end{cases} \tag{2}$$

and the camera focal lengths are f' and f before and after the motion.

Figure 3

Under a 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

$$\begin{cases} s \cdot u' = u + \alpha v + t_u \\ s \cdot v' = v - \alpha u + t_v \end{cases} \tag{3}$$

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 (3). 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 (Figure 3). This observation can be easily deduced by comparing equation (3) with equation (1) when β≈0, u≈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 due to model simplification are of only 0~2 pixels in the central strip (with the width w < 16 pixels in Figure 3). In the actual situation for our image sequences, the camera focal length is about 8 mm. More detailed analysis can be found in Appendix 1.

2.2 Motion estimation and refinement

Motion estimation and refinement consists of two embedded iteration cycles. The first (inner) iteration cycle is robust motion estimation based on the motion displacements from the interframe image matches [22, 23]. The inter-frame image displacements are estimated by using a pyramid-based matching algorithm. The hierarchical algorithm consists of four steps: pyramid construction, hierarchical block matching, match evaluation and robust estimation of motion parameters. Details on the algorithms and the performance analysis can be found in [23]. As a highlight, the 2D rigid transformation between two successive frames in equation (3) is estimated using an iterative weighted least mean square method. However, this iterative process is only carried out on the current motion displacements without re-calculating them from the original images. A 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, by either a match correction or a motion refinement step.

When motion parameters are significantly different from the averages of the previous several frames, we assume it is a mismatch. In this case, the initial inter-frame motion parameters of this frame 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

$$h(u,v) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{u^2}{2\pi\sigma^2}} \qquad (4)$$

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 of the image pyramids in which the initial displacements are detectable.

Figure 4

Figure 4 shows a real example in matching correction for a Library scene. 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 under 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.. The improvement is quite clear by looking at the difference images in Figure 4 (c) and (d).

Even if no mismatch occurs, the refinement process is needed when the rotation angle α is large (since we use α instead of sin α in our motion parameter estimation). The refinement is performed by iteratively warping the current image and re-matching the warped image with the reference image.

We emphasize that a more accurate transform matrix $\boldsymbol{M}_{\mathrm{t}}$ than the one in equation (3) is used to warp the current image t as

$$\mathbf{u}_t' \approx \mathbf{M}_t \mathbf{u}_t, \quad \mathbf{M}_t = \frac{1}{s}\begin{pmatrix} \cos\alpha & -\sin\alpha & t_{\mathrm{u}} \\ \sin\alpha & \cos\alpha & t_{\mathrm{v}} \\ 0 & 0 & 1 \end{pmatrix} \tag{5}$$

That fact is, even if we still use equation (3) to estimate the motion parameters $\theta^{(\mathrm{m})}= (T_u^{(\mathrm{m})}, T_v^{(\mathrm{m})}, \alpha^{(\mathrm{m})}, s^{(\mathrm{m})})$, where (m) denotes the iteration count, errors will be reduced with decreasing residual rotating angles $\alpha^{(\mathrm{m})}$. The warping in the mth iteration can be expressed by

$$\mathbf{u}_i^{(m)} = \mathbf{M}_t^{(m-1)}\mathbf{u}_i^{(m-1)}, \mathbf{M}_t^{(m)} \Leftrightarrow \theta^{(m)}, m = 1,... \tag{6}$$

where $\mathbf{u}_i^{(\mathrm{m})}$ is the ith image point after the mth warping of the current frame. The final transformation matrix for the current frame t is

$$\mathbf{M}_t = \prod_m \mathbf{M}_t^{(m)} \tag{7}$$

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 outer match cycles after rectification can achieve fine registration results.

2.3. Global motion accumulation and mosaic basics

A frame (e.g. the first, the last, or a middle frame) of an image sequence could be selected as the reference frame for the mosaic process. The accumulating transformation parameters between each frame and this reference frame are calculated as

$$\Theta_I^{(t)} \Leftrightarrow \mathbf{P}_t = \prod_{j=0}^{t} \mathbf{M}_j = \mathbf{M}_t \mathbf{P}_{t-1}, t = 1,..., F; \ \mathbf{P}_0 = \mathbf{I} \tag{8}$$

when the first frame (t=0) is selected as the reference frame. The accumulated parameter vector $\Theta_I^{(t)} = (T_u^{(t)}, T_v^{(t)}, A^{(t)}, S^{(t)})$ is used to warp frame t in creating a mosaic after zoom-frames have been separated from the entire sequence (as in Section 3). Image frames are warped and pasted frame by frame onto the final mosaic using the following transformation

$$\mathbf{u} = \mathbf{P}_t \mathbf{u}_t \ (\mathrm{t} = 1, 2, \ldots, \mathrm{F}), \quad \mathbf{P}_t = \frac{1}{s}\begin{pmatrix} \cos \mathrm{A} & -\sin \mathrm{A} & T_{\mathrm{u}} \\ \sin \mathrm{A} & \cos \mathrm{A} & T_{\mathrm{v}} \\ 0 & 0 & 1 \end{pmatrix} \tag{9}$$

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).

3. Zoom-Frame Separation and Multi-Resolution Representation

In image-based rendering applications, we want the capability not only to pan but also to zoom the virtual camera to enhance the visual realism. In image coding, we need to handle the video sequence with camera zooming as well as panning. 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". 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. In the following, we first describe the basic steps, then we will give some real examples.

The zoom subsequence separation is in fact performed before panoramic mosaicing and moving object detection (see Figure 1). The whole procedure includes the following five steps: pan/zoom sequence capture, interframe motion estimation, zoom subsequence separation, panning sequence connection, and key frame selection for each zoom subsequence.

Pan/zoom sequence capture. The camera is first panned, zoomed in for an interested scenic spot, and zoomed out to the normal focal length (approximately) to continue the scan of the scene to make a full 360-degree coverage. Multiple zoom in/out operation may be performed for multiple zooming hot spots. Typically, frames are counted as in the zoom sub-sequence if the absolute scale factor of those frames are more than twice (2) of the normal panning frames.

Interframe motion estimation and global motion accumulation. The interframe motion vectors are estimated, and then the global motion parameters related to a reference frame are calculated, using the method described in Section 2.

Zoom sub-sequence separation. The list of the global motion parameters is scanned sequentially. When a frame t_s that have a global scale $S^{(t_s)} >= S_T$, and the interframe translational components t_u and t_v are significant small, then it is selected as the first frame of a zoom sub-sequence. Typically S_T is selected as 2.0 in order that the algorithm works under accumulation error of scales and does not generate too many fragment zoom frames. Translational components t_u and t_v are selected to be smaller than 5 pixels so that the camera mainly performs a zooming in a zoom sub-sequence. Then the frames followed that satisfied the above conditions are added to the sub sequence, ending with a frame t_e with scale factor $S^{(t_e)} < S_T$ (Figure 5). Note in a zoom sub-sequence, the scale factors first increase to a certain level, then decrease back to normal.

Panning sequence connection. Ideally, frames t_s and t_e will be registered by using the global motion parameters, if there is no accumulation error in the interframe matching from the frame t_s, across the zoom subsequence, to the frame t_e. However, accumulation error does exist; therefore, we only use the accumulated "interframe" transformation as the initial estimation between this pair of to-be-connected frames (t_s and t_e), and then perform real matching between them. Since these two frames view almost the same area, we use histogram equalization for both to deal with possible illumination changes due to lighting and auto iris. Then, the refined global transformations are re-calculated after the final panning sequence is determined for final mosaicing.

Key frame selection. From each zoom subsequence, we only select frames when the scale factor changes by a certain number S_c, for example $S_c = 1.5$ when significant zoom happens, until a frame with the highest resolution. So the selected key frames are $t_0 = t_s$, t_1, t_2, …, which satisfy the condition $S_i^{(t)} >= S_T S_c^i$, i = 0, 1, 2,….Frame t_0 can be roughly registered with the panoramic mosaic using its own global motion parameters.

Figure 5

We want to make two notes here. First, an automatic registration between two zoomed frames is achieved in a manner similar to that for the panned frames, but after that the next 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 refinement processing after warping (i.e. re-zooming) is vital for the accurate estimation of the scale parameter. Figure 6 shows a matching example from a zooming clip of a Main Building sequence shown in Figure 7. 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 Figure 6c and Figure 6d. The zoom factor between Figure 6a and Figure 6b is 1.12. The reason for the successful match is that every iteration adjusts the scale factor to approach to the real one.

Figure 6

Second, the effect of the value of the threshold S_T is not significant for both panoramic mosaicing and multi-resolution representation, since the mosaicing procedure handles scale changes between images in the sequence. However, it is safe to select the S_T larger than 1.5 otherwise the accumulation error of the scales over 1.0 may cause false detection of a zoom sub-sequence. Figure 7b shows the cylindrical panorama generated from the panning and zooming image sequence. In this background mosaic, moving objects, some of them are very large, as in the third image of Figure 7a, were removed. Note that the mosaic has been rectified to a 360-degree panoramic view, using the method that will be described in Section 4. Two zoom in/out operations were performed in this sequence, which are marked with rectangles on the mosaic. It is clearly shown that the focal lengths are obviously changed before and after the first zooming operation. Figure 7c shows the three selected zooming frames of the second zooming sub-sequence, with a 1.5 scaling factor between two selected frames. The rectangle in each frame indicates the sub-region that corresponds to the next selected frame. The selection of the scaling factor in the multi-resolution representation should be suitable so that with a minimum number of frames for each interesting area, much better rendering results can be achieved by using image morphing between two preserved zoom frames rather than just using a direct digital zoom from a singe basis resolution of the panorama.

Figure 7

4. Closed-Loop Image Mosaicing and Rectification

Since only one narrow vertical strip in the center of each frame is utilized, a 2D rigid transformation is sufficient to merge the successive frames. Intuitively, the 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 (Figure 8). The principle behind the "conic mosaicing" can be explained as follows. Suppose the central strip is represented in spherical coordinates. Then the four parameters in equation (9) could be interpreted as the 3D rotation (T_u, T_v, A) and zoom (S) of the camera, even though error will be 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 (e.g. in

Figure 8(b)) if the optical axis of the reference frame is slightly downward looking (I_c in Figure 8(a)) and vice versa (I_a in Fig.3 (a), and Figure 10(a)).

4.1 Cylindrical panoramic rectification

Rectifying the unfolded conic mosaic to an unfolded 360° cylindrical panorama is 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 "re-homing" (tail) frames in the image sequence with the first (head) frame using the same pyramid-based matching strategy, and then selecting the frame with minimum difference between the overlapping region of its warped image with the first frame. The "virtual" vertical edge in the head frame is represented by the central column PQ, and the corresponding edge P'Q' is determined by the image matching. To account for the illumination changes between the connecting head and tail frames, histogram specification from the frame in consideration into the head frame is performed. With the head-tail match, the angular range of the unfolded cone, $\angle PoP'$, and the radii of inner and outer arcs of the unfolded cone, oQ and oP , are computed. The re-projection of the conic mosaic to the cylindrical panorama can then be determined (see Figure 8(b)). We will show the algorithm in detail in the following.

Figure 8

If the reference frame coordinate $(u, v, 1)^t$ (i.e. the mosaicing coordinate system) is chosen as the first frame coordinate $(u_1, v_1, 1)^t$, then the transformation ($\mathbf{P_E}$) between the "re-homing" (tail) frame $(u_{E,} v_E, 1)^t$ and the reference frame can be obtained by the successive rigid transformations from the first (head) frame to the last (tail) frame from equation (8) (Fig.3 (b)):

$$\begin{pmatrix} u_E \\ v_E \\ 1 \end{pmatrix} = \mathbf{P}_E \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \tag{10}$$

The physical relation (in the cylindrical mosaicing coordinate system) between the head frame and the tail frame derived from their direct match can be expressed as

$$\begin{pmatrix} u_E \\ v_E \\ 1 \end{pmatrix} = \mathbf{M}_{E1} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} \tag{11}$$

where $\mathbf{M}_{E1}$ is defined in equation (5). For a point $(u_1, v_1)^T$ in the head frame, its coordinates in the panorama frame are simply $Q = (u, v, 1)^t = (u_1, v_1, 1)^t$. But for its corresponding point in the tail frame $(u_E, v_E, 1)^t$, the coordinates Q′ in the panorama frame should be calculated as

$$\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix}^T = \mathbf{P}_E^{-1} \mathbf{M}_{E1} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} \tag{12}$$

using equations (10) and (11). By finding a vertical line segment PQ (a column) in the head frame, the corresponding segment $P'Q'$ in the panorama can be determined by using equation (12). This "vertical" line can be selected as the central column of the head frame when the head frame is not level (see Fig.3(b)). The center of the inner and outer circles of the unfolded conic mosaic, *o*, is the intersection point of PQ and $P'Q'$. For simplicity of notation, we choose the new coordinate system *xoy* with the origin at the center of the circles. Then the angle range of the unfolded cone $\angle PoP'$ is

$$\theta = \theta_0 - \theta_1 \tag{13}$$

where

$$\theta_0 = \tan^{-1}\left(\frac{y_P - y_Q}{x_P - x_Q}\right),\ \theta_1 = \tan^{-1}\left(\frac{y_{P'} - y_{Q'}}{x_{P'} - x_{Q'}}\right) \tag{14}$$

where the two pairs of the end points of the two line segments are used in the calculation: $P(x_P, y_P)$ and $Q(x_Q, y_Q)$; $P'(x_{P'}, y_{P'})$ and $Q'(x_{Q'}, y_{Q'})$. Due to the change of the camera's focal length and accumulating errors, we could have a "deformed" cone with different radii (i.e., $|PQ| \neq |P'Q'|$) in the head and tail of the conic mosaic, e.g.,

$$R = R_P - R_Q \geq R' = R_{P'} - R_{Q'} \tag{15}$$

where $R_P = |oP|, R_Q = |oQ|$, etc. The height (in the vertical direction) and the length (in the angular direction) of rectified cylindrical panorama are set as the larger ones (to preserve image resolution), as

$$R = R_P - R_Q,\ L = R_P \theta \tag{16}$$

So the relation between the conic mosaic (*x,y*) and the cylindrical mosaic (*r,l*) can be expressed as

$$(x, y) = (R_{rl} \cos\theta_l, R_{rl} \sin\theta_l) \tag{17}$$

where

$$\theta_l = \theta_0 + \frac{l}{R_P}$$
$$R_{rl} = R_Q + \frac{l}{L}(R_{Q'} - R_Q) + r + \frac{lr}{LR}(R'-R) \qquad (18)$$

and $l = 0,\ldots,L$ (left to right); $r = 0, \ldots, R$ (bottom-up). The reason for us to use an inverse transformation in equation (17) from the destination *(r,l)* to the source *(x,y)* is that we can easily generate a dense cylindrical mosaic from a conic mosaic. Note that this process also eliminates the accumulating errors from frame-to-frame registration.

Figure 9

4.2. Experimental results

Figures 9 and 10 show a real example of head-tail matching and closed-loop cylindrical rectification. Figure 9 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 numbers are truncated two places after the decimal point, so –0.00 means a very small negative value). The second set of parameters results in a better registration result, which can be observed from the edges in the difference images between the two frames (with the tail image warped to the reference image), especially in the center strip of the image which will be used for the mosaic, e.g., the white lamp in front of the pine tree and the door near that tree.

Figure 10

Figure 10(a) and Figure 10(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 (1). The size of the rectified cylindrical panorama is 3494x323 (please visit http://www-cs.engr.ccny.cuny.edu/~zhu/panorama2D.html for high resolution panoramas). 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 (refer to examples in Figure 7b). The moving object extraction will be presented in the next section.

5. Moving Object Extraction and Representation

As the mosaic is being constructed, difference images between the warped successive frames are analyzed. Regions in the panorama that correspond to those containing large residuals in the difference images are labeled as "dynamic hot spots". The sequences of 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, a region grouping procedure 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 [24-27]. The basic idea of an 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. Finally, 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.

In the following three sub-sections, we will detail each of the three steps: motion detection, initial region grouping, and object extraction via the modified active contour approach.

5.1. Motion detection via three-frame differencing

For moving target extraction, we use three successive frames (f_1, f_2, f_3). Let the second frame f_2 be the reference frame (current frame). After warping the first and the third frame to the reference frame using the corresponding interframe motion parameters, a new triple of frames (f_1', f_2, f_3') is generated. By smoothing each frame using a 3×3 kernel, we generate three smoothed and registered frames noted as $(\overline{f}_1, \overline{f}_2, \overline{f}_3)$. Then, the difference image among three frames is defined as

$$D(i,j) = |\overline{f}_1(i,j) - \overline{f}_2(i,j)| \times |\overline{f}_2(i,j) - \overline{f}_3(i,j)| \tag{19}$$

Note that there will be a difference in a pixel location only if there are differences among all three frames in that location. Moving objects correspond to regions with large values in the difference

image. These regions can be extracted by thresholding the difference image by the following *bi-threshold* method, thus generating a binary *mask* image

$$G(i,j) = \begin{cases} 1 \text{ if } D(i,j) > T_{high}, \text{or } (D(i,j) > T_{low} \ \& \ (i,j) \in C \ \& \ \exists (k,l) \in C) \\ 0, \text{Othewise} \end{cases} \quad (20)$$

where $T_{high} > T_{low}$ are two thresholds, and C is a connected region that have pixels (k,l) assigned as 1.

The advantages of using difference image among three frames are twofold. (1) While the moving target extraction from the difference between only two successive frames often exceeds the object regions in the current frame (Figures. 11a and 11b), most of the points that have such three-frame differences correspond to the image of the moving objects in the current (reference) frame (Figure 11c), since equation (19) indicates that there will be a difference only if there are differences among all the three frames.. (2) The regions in the three-frame difference image are more compact than those in the two-frame difference image if the object texture is smooth and thus generates low difference values within the region, due to the same reason.

Figure 11

5.2. Initial region extraction via region grouping

Generally speaking, for real moving objects that have large portions of homogeneous colors or smooth textures, the binary mask image G cannot give good contours for the moving objects: an object often consists of several dis-connected regions, and the boundary is not accurate. In order to form a single region for each object, a morphological close transformation is first carried out on the mask image G, and then nearby regions are grouped into a single region. The question is how to identify regions corresponding to real moving objects. The following cues are used for this purpose.

- Size and shape constraints. For example, a long horizontal or vertical narrow strip region may correspond to a video scan-line noise instead a moving object.
- Temporal constraints. The existence, size and the shape could not change rapidly in a few frames. Therefore, the determination of the region of an object in the current frame uses the track of this object in previous frames.

The algorithm for the initial region extraction can be summarized in the following steps.

Step 1. A grayscale morphological open operation is applied to the binary mask image in order to eliminate some isolated spots and thin lines. Small spots may due to electrical noise of the camera, and the thin lines may correspond to the edges due to inaccurate registration and the scanline noise of the camera or a videotape. After the grayscale morphological operation, the binary difference image is transformed into a grayscale mask image. The grayscale mask image is then smoothed using a 3×3 average kernel so that the "gray-level" of the difference image is smooth.

Step 2. A threshold is determined for turning the grayscale mask image back into a binary one using the gray-level distribution of the gray-scale mask image. Typically, the threshold is set to 15% of the maximum difference of the current frame so that holes in the original binary mask image are filled.

Step 3. A morphological close operation and a nearby-region grouping procedure are performed to further generate more solid regions for moving objects. Initial contours for moving target regions are generated by extracting boundaries of those regions in the processed mask image.

5.3. Region refinement via an active contour approach

The final step in moving object extraction is to refine the contour of each region on the current original frame and then separate the region out of the frame. For this purpose a modified active contour approach is applied. The concept of active contour algorithms was first proposed by Kass, Witkin & Terzopoulos [24] and many modified methods have been developed since then. Amini, Weymouth & Jain [25] 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 [26] 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 is utilized in our active contour method.

For a closed contour $u(s) = (x(s), y(s)), \quad s \in (0,1)$, the energy function is defined as

$$E_{total} = \int_0^1 \{E_{\text{int}}(u(s)) + E_{image}(u(s)) + E_{con}(u(s))\} ds \tag{21}$$

where

$$E_{\text{int}}(u(s)) = \alpha(s)\,|\,u_s(s)\,|^2 + \beta(s)\,|\,u_{ss}(s)\,|^2$$

$$E_{image}(u(s)) = w_{line}E_{line}(u(s)) + w_{edge}E_{edge}(u(s)) + w_{motion}E_{motion}(u(s)) \qquad (22)$$

$$E_{con}(u(s)) = -k(x1 - x2)^2$$

In the above equations, E_{int} is the internal energy that forces the contour smooth, where u_s、u_{ss} are the 1st and 2nd order differentials along the contour. E_{image} is the external energy from the current image that pull the current contour to the object contour. This term is the weighted average of the following three: $E_{line} = I(x, y)$ - the intensity function of the image, $E_{edge} = -|\nabla I(x, y)|^2$ - the gradient function of the image to account for edges, and $E_{motion} = -|D(x, y)|$ - the temporal difference function of the image defined in equation (19) to account for motion. Finally, E_{con} is the energy terms of the control points, so that the contour will be pulled to the control points (k *i*s the elastic coefficient). The external energy E_{ext} is the sum of E_{image} and E_{con}. We use the same discrete form as in [25]. Suppose the number of the control points is n. Then if the search range of every point is m，then the spatial complexity of the algorithm is $O(nm^2)$ and the time complexity is $O(nm^3)$. Using the searching strategy in [26], the time complexity can be reduced.

For our specific application, there are two improvements on the active contour algorithm:

(1) Both the speed and accuracy of the initial contour extraction is increased. Since we extract contours of moving objects in the motion sequence, motion information is used to speed up the procedure of initial contour extraction; meanwhile, more accurate initial contours speed up the active contour convergence and increase the accuracy of the final contours.

(2) We integrate the advantages of several algorithms [24-27]. 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 spaces are adaptively changed according to the size of the initial contour. The energy function incorporates both shape and motion information. Then, the parameters used in the energy function are automatically assigned according to the point spaces and the curvatures, following [27]. The energy function is minimized using the dynamic programming approach [25] and the line search strategy [26] to obtain the resulting contour.

Figure 12

Figure 12(a) and Figure 12(b) show an original image and the extracted object (a person). Figure 12(c) shows the dynamic mosaic with the walking person pasted onto the mosaic every ten frames.

6. Concluding Remarks

The algorithm proposed in this paper for the construction of the Dynamic and Multi-resolution Panorama (DMP) is fast, robust, and automatic. Key features of this work include: automatic closed-loop mosaicing, accurate moving object extraction, zoom-frame handling, and multi-resolution representation. The processing rate is about 1 frame per second for 384×288 color images using a Pentium II/ 266 MHz PC, and up to 5 frames per second (5 Hz) on a Pentium III 800 MHz laptop.

The objective of this work is to build an image-based representation with panoramic views, multiple resolutions, and dynamic objects in natural scenes. In addition to 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 video surveillance, video enhancement, indexing and manipulation. Future work could include the following interesting topics.

Panoramic view morphing. Seitz and Dyer [28] 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 new images from the set of views. We can apply 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.

3D and layered panoramas. We can generate several interesting representations that have both 3D and multi-resolution representations. For example, by combining the layered panoramic representations we proposed in [8-10] with the dynamic multi-resolution panoramic representation in this paper, we can generate a 3D layered and multi-resolution panorama for 3D scene modeling when a camera undertakes a motion with a dominant translation direction. As another example, using a camera with off-center rotation, we can generate panoramic stereo mosaics [29,30] and then obtain depth maps [30, 31] of the panoramic views. With a camera undertaking off-center rotation plus zooming for a dynamic scene, we will be able to generate a 3D panoramic representation of the scene, together with multi-resolution and moving object representations.

Acknowledgments

This work was partially supported by the China High Tech Program under contract No. 863-306-ZD-10-22, China Natural Science Foundation under contract No. 69805003, NSF Grant Number EIA-9726401, AFRL Award FA8650-05-1-1853, ARO Award 911NF-05-1-0011, New York Institute for Advanced Study (NYIAS) and a CUNY Graduate Research Technology Initiative (GRTI) grant. The basic algorithms of motion detection and estimation was provided by Dr. Yudong Yang of Tsinghua University, and an early experimental modeling system was implemented with the assistance of Mr. Heng Luo and Mr. Qiang Wang at Tsinghua University. The authors also want to thank the anonymous reviewers for their insight comments and suggestions, and Mr. Robert Hill at City College for proofreading the manuscript.

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 (1) and (3) 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.

(|u|, |v|)	(0,0)	(0,128)	(192,128)
(δu|,|δv|)*	(0, 0)	(0, 0)	(4.0, 3.3)
(δu|,|δv|)**	(0, 0)	(0.61. 2.03)	(6.9, 5.3)

* α=β=0, γ=2° ; ** α=β=γ=2°

It is interesting to notice that there are no differences between equations (1) and (3) for the central column if the tilt angle β is zero.

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, K. D. Rajasekar, E. Riseman, A. Hanson. Panoramic Virtual Stereo Vision of Cooperative Mobile Robots for localizing 3D Moving Objects. *Proceedings of IEEE Workshop on Omnidirectional Vision – OMNIVIS'00*, Hilton Head Island, 29-36, JUNE 2000.

[7]. Z. Zhu, G. Xu, E. M. Riseman and A. R. Hanson, Fast Generation of Dynamic and Multi-Resolution 360°Panorama from Video Sequences, *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, Florence, Italy, June 7-11, 1999, vol.1 , pp 400-406.

[8]. Z. Zhu, G. Xu and X. Lin, Panoramic EPI Generation and Analysis of Video from a Moving Platform with Vibration, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 23-25 June, 1999, Fort Collins, Colorado, vol 2, pp. 531-537.

[9]. Z. Zhu, G. Xu and X. Lin, Efficient Fourier-based approach for detecting orientations and occlusions in epipolar plane images for 3D scene modeling, *International Journal of Computer Vision*, 61 (3): 233-258, February - March, 2005.

[10]. Z. Zhu, and A. R. Hanson, LAMP: 3D layered, adaptive-resolution and multi-perspective panorama - a new scene representation, *Computer Vision and Image Understanding,* Special Issue on Model-based and Image-based 3D Scene Representation for Interactive Visualization, 96 (3), December 2004, pp 294–326.

[11]. Z. Zhu, E. M. Riseman, A. R. Hanson, Generalized parallel-perspective stereo mosaics from airborne videos, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, Feb 2004, pp 226-237.

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

[13]. 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.

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

[15]. 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.

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

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

[18]. A. Bartoli, N. Dalal, B. Bose and R. Horaud, From Video sequences to motion panoramas, *IEEE Workshop on Motion and Video Computing*, Orlando, Florida, Dec. 5-6, 2002, pp 201- 207.

[19]. M. Brown and D. G. Lowe, Recognising panoramas, *Proceedings of the 9th International Conference on Computer Vision (ICCV 2003)*, Nice, France, October 2003, pp. 1218-25.

[20]. Y. Dufournaud, C. Schmid, and R. Horaud. Image matching with scale adjustment, *Computer Vision and Image Understanding*, vol 93, no 2, February 2004, pp. 175-194.

[21]. K. Mikolajczyk, C. Schmid, Indexing based on scale invariant interest points, *Proceedings of the 8th International Conference on Computer Vision*, Vancouver, Canada, 2001, pp. 525-531.

[22]. 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.

[23]. Z. Zhu, E. M. Riseman, A. R. Hanson and H. Schultz, An efficient method for geo-referenced video mosaicing for environmental monitoring. *Machine Vision Applications Journal*, 2005.

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

[25]. 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.

[26]. K. F. Lai, R. T. Chin, Deformable contours: modeling and extraction, *IEEE Trans. PAMI*, vol.17 no.11, Nov 1995: pp1084-1089.

[27]. Williams D J, Shah M "A fast algorithm for active contours and curvature estimation", *CVGIP : Image Understanding*, Vol.55 No.1, January 1992: pp.14-26.

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

[29]. S. Peleg, M. Ben-Ezra, and Y. Pritch, OmniStereo: Panoramic Stereo Imaging, *IEEE Trans. on PAMI*, March 2001, pp. 279-290.

[30]. Y. Li, H.-Y. Shum, C.-K. Tang, R. Szeliski, Stereo Reconstruction from Multiperspective Panoramas. *IEEE Trans. on PAMI*, 26(1), 2004: pp 45-62.

[31]. C. Sun and S. Peleg, Fast Panoramic Stereo Matching using Cylindrical Maximum Surfaces, *IEEE Trans. on SMC, Part B*, Vol. 34, Feb. 2004, pp. 760-765.

Figure Captions

Figure 1. System diagram.

Figure 2. Coordinate systems of the camera and the image.

Figure 3. Mosaicing strips from two successive frames.

Figure 4. A match correction example for the 246-frame Library image sequence. (a) the current frame; (b) the previous (reference) frame; (c) difference image of inital matching; (d) difference image after re-matching.

Figure 5. Zoom sub-sequence separation, panning sequence connection, and key frame selection.

Figure 6. Iterative matching after image warping (re-zooming). (a) the current frame; (b) the reference (previous) frame; (c) difference image of initial matching; (d) difference image after matching refinement.

Figure 7. Multi-resolution panorama. The original image sequence has 561 frames, which includes two zooming segments inside the panning sequence. (a). Three frames of the Main Building sequence when the camera was panning from right to left. There are many moving objects (persons, bicycles) in the scene. (b) Cylindrical panorama (image size: 3498x303). Notice that most of the moving objects and noises (e.g. horizontal lines in frame 199) have been successfully filtered out. (c) Three selected zooming frames for one of the "interesting" areas, which is at the right edge of the first segment of the panorama.

Figure 8. The strip-mosaicing geometry. (a) spherical and conic representation; (b) unfolded conic mosaicing and rectification.

Figure 9. Head-tail match and refinement for a 246-frame Library sequence (panning from right to left). (a) the current frame (Frame no. 245) ; (b) the reference frame (Frame no. 0); (c) difference image of inital match; (d) difference image after match refinement.

Figure 10. The panoramic mosaic from the 246-frame Library sequence. (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. (b). Unfolded 360-degree cylindrical panorama (27% display scale; 1^{st} row : 0~180°; 2^{nd} row: 180°~360°). The original true-color image is 3494x323 x24 bits.

Figure 11. Three frame difference illustration, assuming that the rectangular object moving to down-right. (a) frame difference (colored region) between f1 and f2. (b) that between f2 and f3, and (c) the three frame difference. Ideally, the three-frame difference give the entire object region in the current frame f2, whereas the two frame differences usually have "fatter" regions than the true one.

Figure 12. Moving object detection and separation. (a) an original image frame; (b) extracted moving object ; (c) dynamic mosaicing: synopsis of the walking human was pasted on part of the cylindrical panorama.

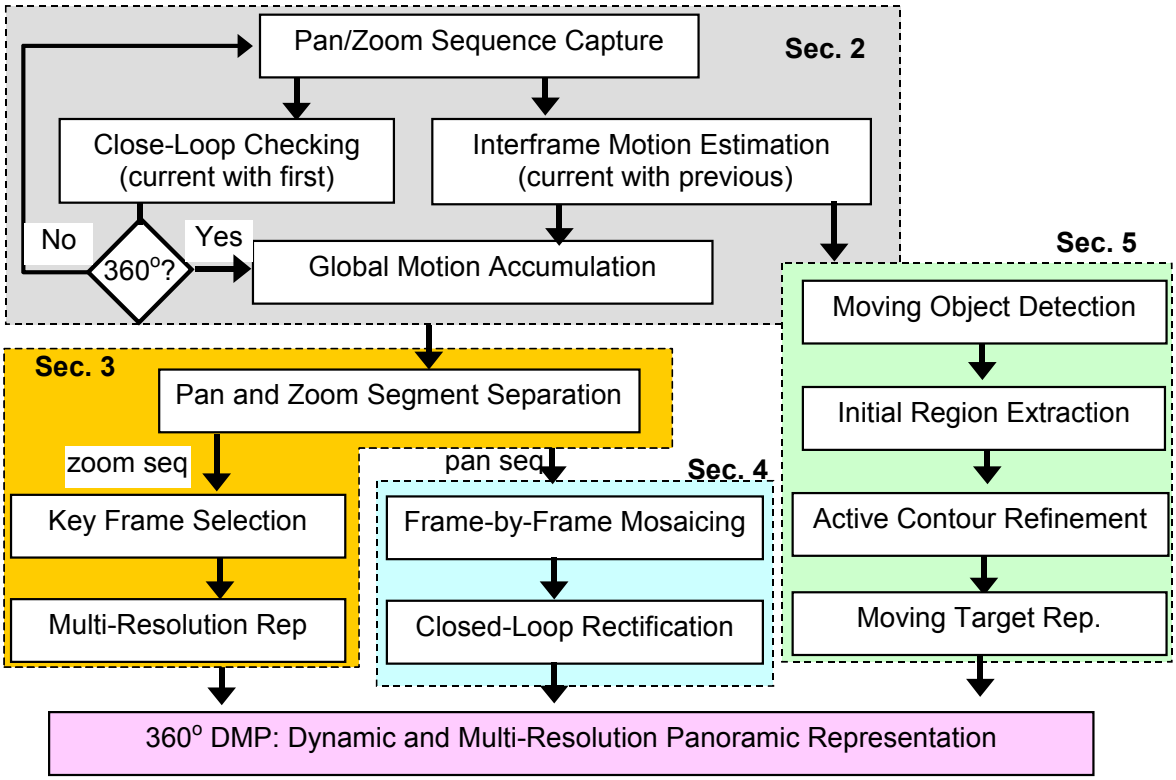


Figure 1. System diagram.

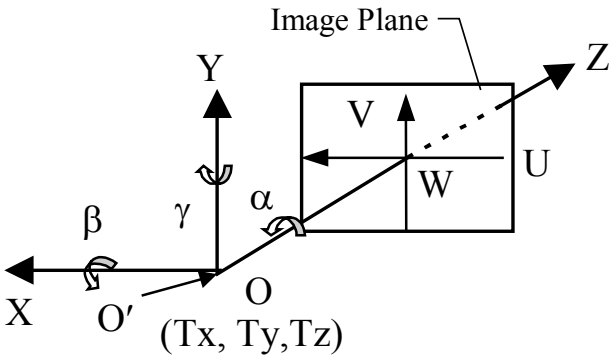


Figure 2. Coordinate systems of the camera and the image.

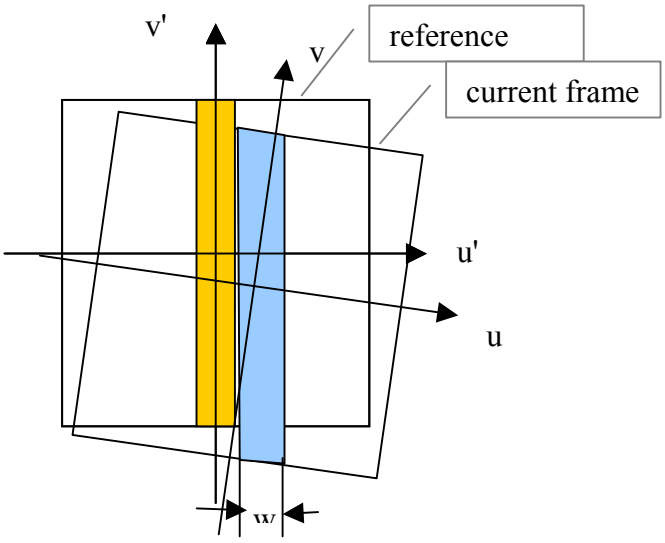
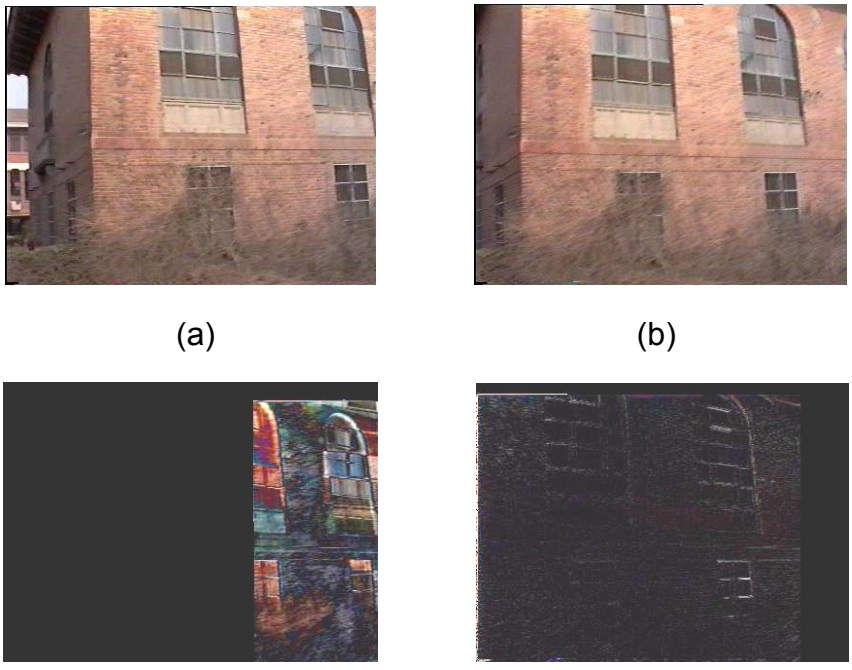


Figure 3. Mosaicing strips from two successive frames.



(a) (b)

(c) (d)

Figure 4. A match correction example for the 246-frame Library sequence. (a) the current frame; (b) the previous (reference) frame; (c) difference image of inital matching; (d) difference image after re-matching.

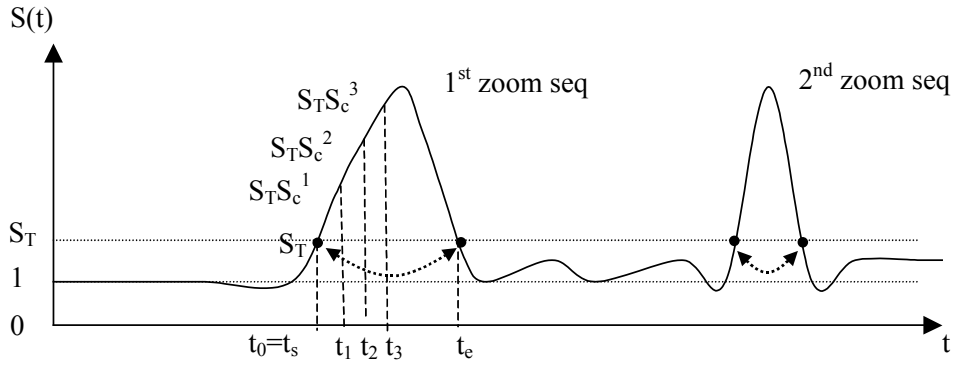
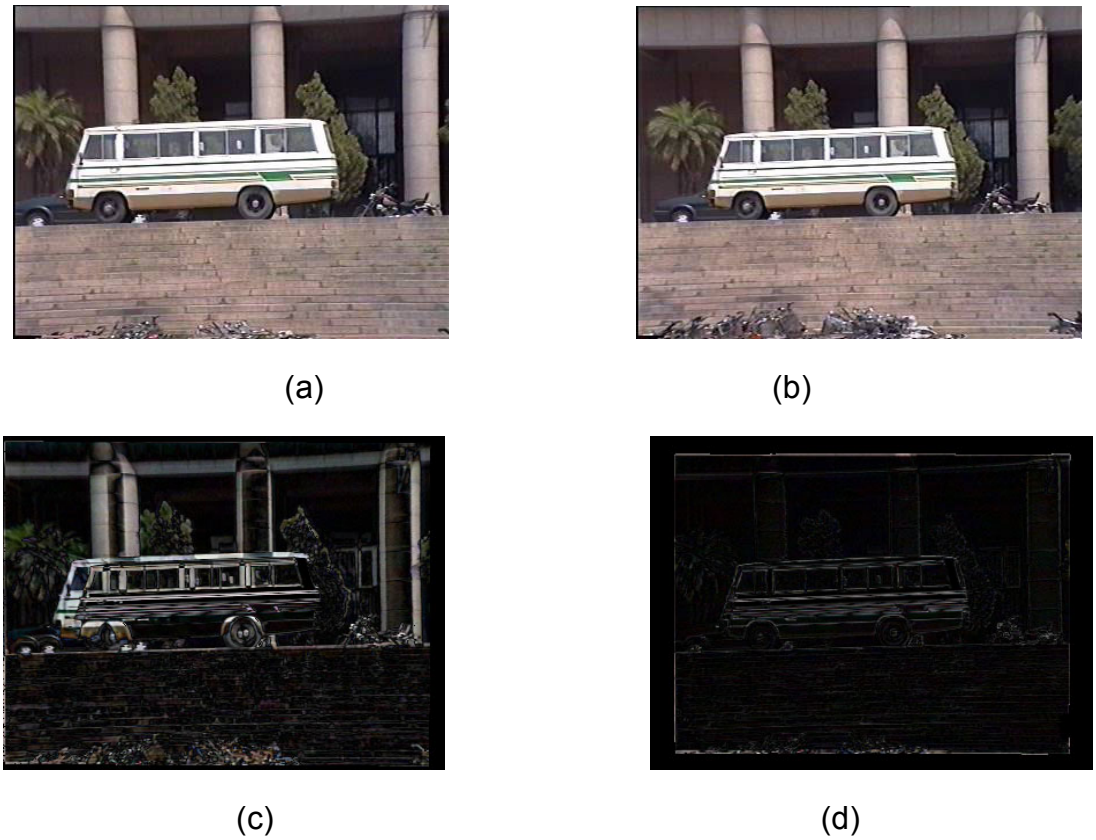


Figure 5. Zoom subsequence separation, panning sequence connection and key frame selection.

.



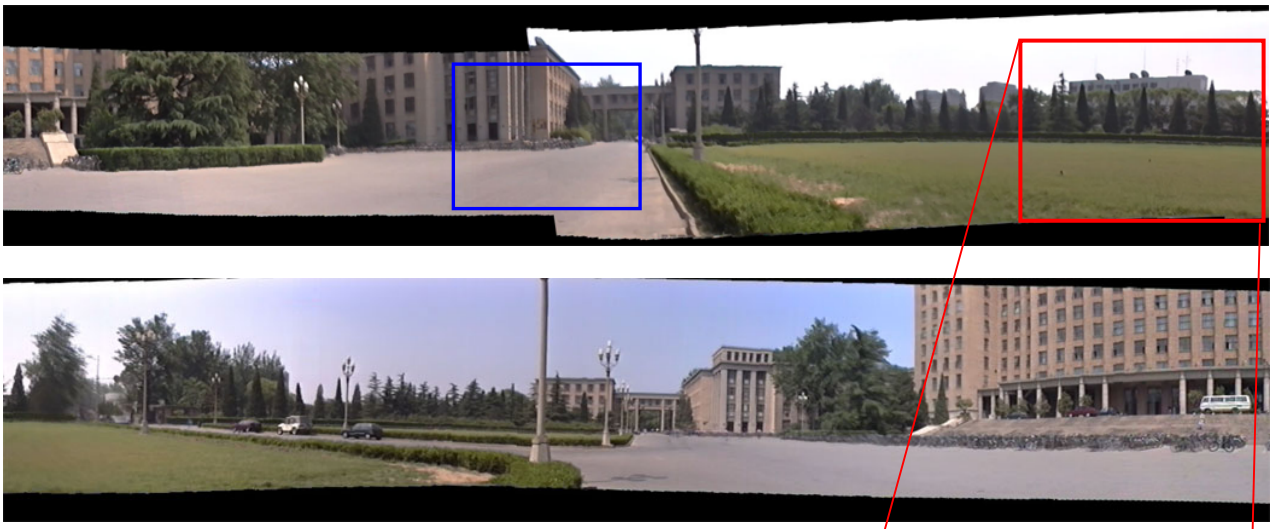
(a) (b)

(c) (d)

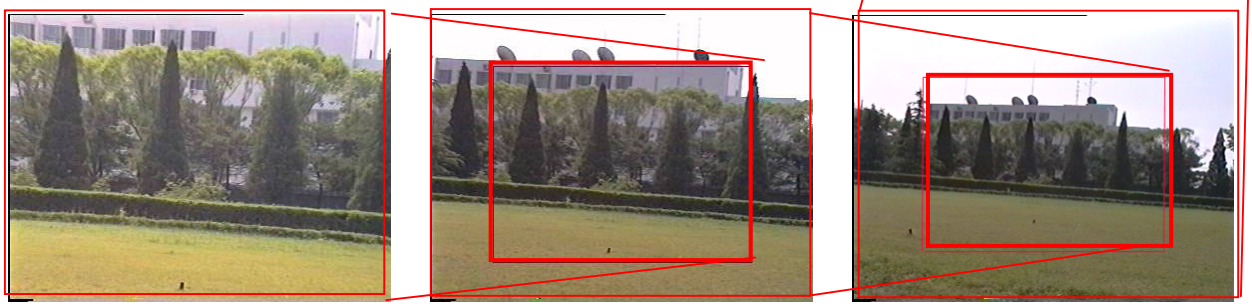
Figure 6. Iterative matching after image warping (re-zooming). (a) the current frame; (b) the reference (previous) frame; (c) difference image of initial matching; (d) difference image after matching refinement.

(a)

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

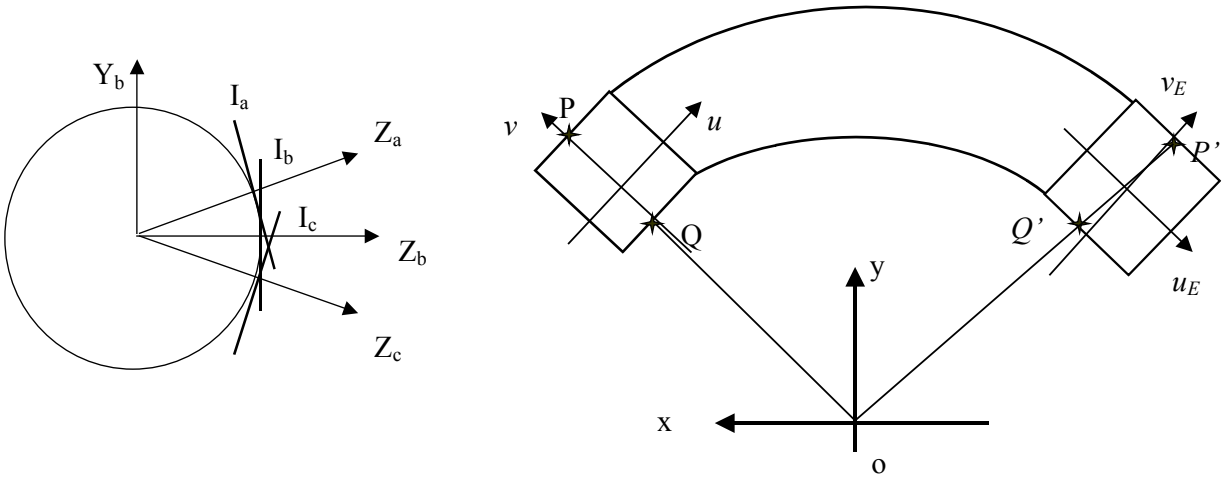


(b)



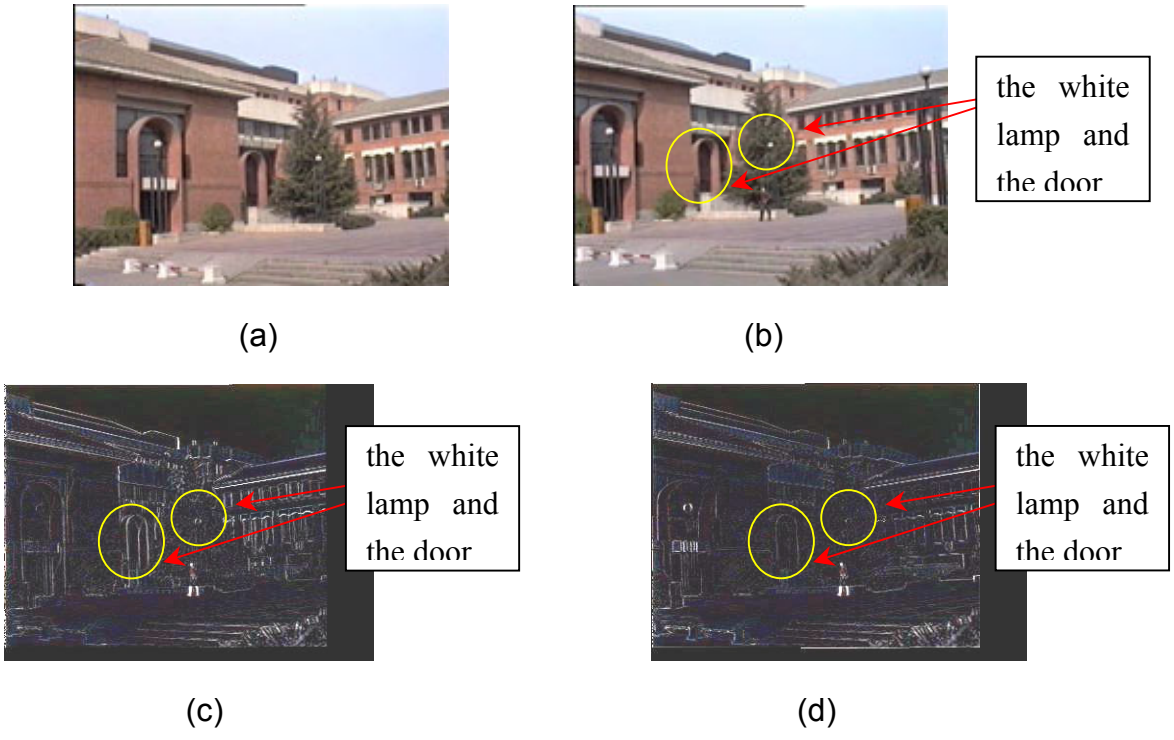
(c)

Figure 7. Multi-resolution panorama. The original image sequence has 561 frames, which includes two zooming segments inside the panning sequence. (a). Three frames of the Main Building sequence when the camera was panning from right to left. There are many moving objects (persons, bicycles) in the scene. (b) Cylindrical panorama (image size: 3498x303). Notice that most of the moving objects and noises (e.g. horizontal lines in frame 199) have been successfully filtered out. (c) Three selected zooming frames for one of the “interesting” areas, which is at the right edge of the first segment of the panorama.



(a) (b)

Figure 8. The strip-mosaicing geometry. (a) spherical and conic representation; (b) unfolded conic mosaicing and rectification.



(a) (b)

(c) (d)

Figure 9. Head-tail match and refinement for a 246-frame Library sequence (panning from right to left). (a) the current frame (Frame no. 245) ; (b) the reference frame (Frame no. 0); (c) difference image of inital match; (d) difference image after match refinement.

(a)

(b)

Figure 10. The panoramic mosaic from the 246-frame Library sequence. (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. (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.

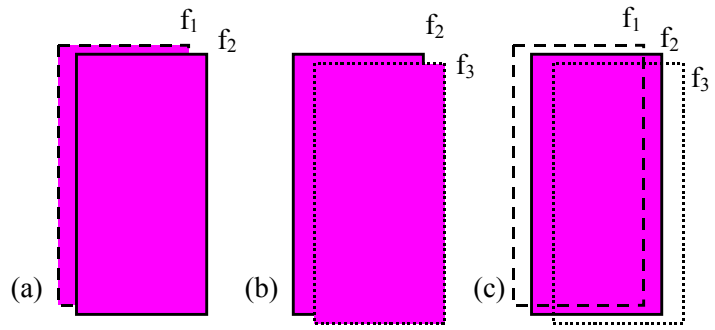


Figure 11. Three frame difference illustration, assuming that the rectangular object moving to down-right. (a) frame difference (shaded region) between f1 and f2. (b) that between f2 and f3, and (c) the three frame difference. Ideally, the three-frame difference give the entire object region in the current frame f2, whereas the two frame differences usually have “fatter” regions than the true one.

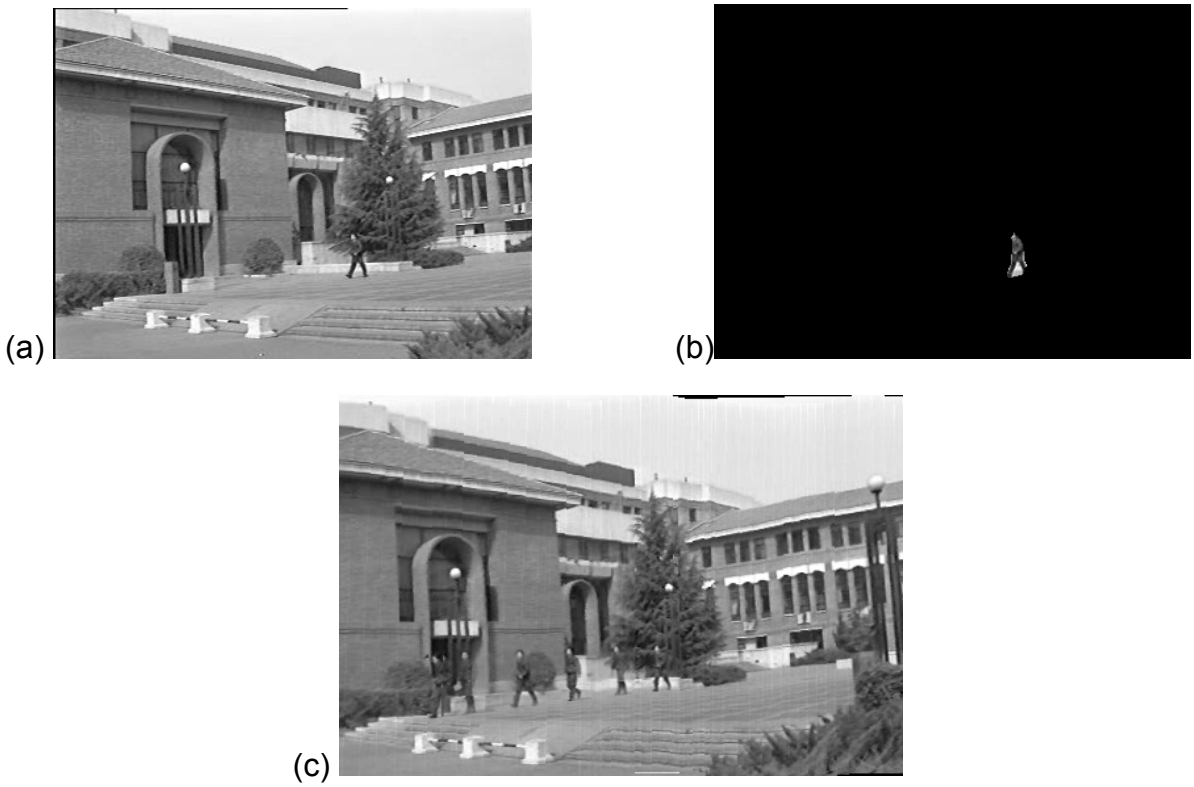


Figure 12. Moving object detection and separation in the Library video sequence. (a) an original image frame; (b) extracted moving object ; (c) dynamic mosaicing: synopsis of the walking human was pasted on part of the cylindrical panorama.