Appendix c
Parallel-Perspective Stereo Mosaics for UVIS

Zhigang Zhu, Edward M. Riseman, Allen R. Hanson and Howard Schultz
Department of Computer Science, University of Massachusetts at Amherst, MA 01003
{zhu, riseman, hanson, hschultz}@cs.umass.edu

1. Introduction

There have been attempts in a variety of applications to add 3D information into an image-based mosaic representation. Creating stereo mosaics from two rotating cameras was proposed by Huang & Hung [1], and from a single off-center rotating camera by Ishiguro, et al [2], Peleg & Ben-Ezra [3], and Shum & Szeliski [4]. In these kinds of stereo mosaics, however, the viewpoint -- therefore the parallax -- is limited to images taken from a very small area. Recently our work [5,6,7] has been focused on parallel-perspective stereo mosaics from a dominantly translating camera, which is the typical prevalent sensor motion during aerial surveys. A rotating camera can be easily controlled to achieve the desired motion. On the contrary, the translation of a camera over a large distance is much hard to control in real vision applications such as robot navigation [8] and environmental monitoring [6, 9]. We have previously shown [5-7] that image mosaicing from a translating camera raises a set of different problems from that of circular projections of a rotating camera. These include suitable mosaic representations, the generation of a seamless image mosaic under a rather general motion with motion parallax, and epipolar geometry associated with multiple viewpoint geometry.

First we will show why an efficient "3D mosaicing" techniques are important for accurate 3D reconstruction from stereo mosaics. Obviously use of standard 2D mosaicing techniques based on 2D image transformations such as a manifold projection [11] cannot generate a seamless mosaic in the presence of large motion parallax, particularly in the case of surfaces that are highly irregular or with large different heights. Moreover, perspective distortion causing the geometric seams will introduce errors in 3D reconstruction using the parallel-perspective geometry of stereo mosaics. In generating image mosaics with parallax, several techniques have been proposed to explicitly estimate the camera motion and residual parallax [9,12,13]. These approaches, however, are computationally intense, and since a final mosaic is represented in a reference perspective view, there could be serious occlusion problems due to large viewpoint differences between a single reference view and the rest of the views in the image sequence.

We have proposed a novel "3D mosaicing" technique called PRISM (parallel ray interpolation for stereo mosaicing) [7] to efficiently convert the sequence of *perspective* images with 6 DOF motion into the parallel-perspective stereo mosaics. In the PRISM approach, global image rectification eliminates rotation effects, followed by a fine local transformation that accounts for the interframe motion parallax due to 3D structure of the scene, resulting in a stereo pair of mosaics that embodies 3D information of the scene with optimal baseline.

The purpose of this effort is to study how to apply the PRISM approach to video mosaicing for an under-vehicle inspection system (UVIS) that will be able to inspect the undersides of vehicles. The system will create entire images of the vehicle undersides for both visual inspection and automatic comparison with previous images stored in a database. Since cameras are very close to the underside of a vehicle under inspection, each camera only covers a small portion of the underside. Thus a composite picture covering the entire under-vehicle will be created by mosaicing the images from a virtual 2D "array of cameras". Challenging technical issues include (1) calibration of the 1D camera array; (2) estimation of the motion of the vehicle while creating the mosaics and (3) seamless mosaicing with 2D "array of cameras" with different viewpoints.

Our approach for solving this problem is to use a line of cameras as a scanner. The system will continuously take images as the vehicle drives over, then mosaic all of those images into a single image used for inspection. The dense coverage of the vehicle bottom by camera images as the vehicle drives over allows for relatively easy mosaicing of the vehicle underside image. As a result of camera view overlap, we can also create multiple mosaics with pseudo-parallel projection representations that preserve the occlusion information from different viewing angles in favor of vehicle inspection.

2. Parallel-Perspective Stereo Geometry

Fig. 1 illustrates the basic idea of the parallel-perspective stereo mosaics. Let us first assume the motion of a camera is an ideal 1D translation, the optical axis is perpendicular to the motion, and the frames are dense enough. We can generate two spatio-temporal images by extracting two columns of pixels (perpendicular to the motion) at the front and rear edges of each frame in motion. The mosaic images thus generated are similar to *parallel-perspective* images captured by a linear pushbroom camera [14], which has *perspective projection in the direction perpendicular to the motion and parallel projection in the motion direction*. In contrast to the common pushbroom aerial image, these mosaics are obtained from two different oblique viewing angles of a single camera's field of view, one set of rays looking forward and the other set of rays looking backward, so that a stereo pair of left and right mosaics can be generated as the sensor moves forward, capturing the inherent 3D information.

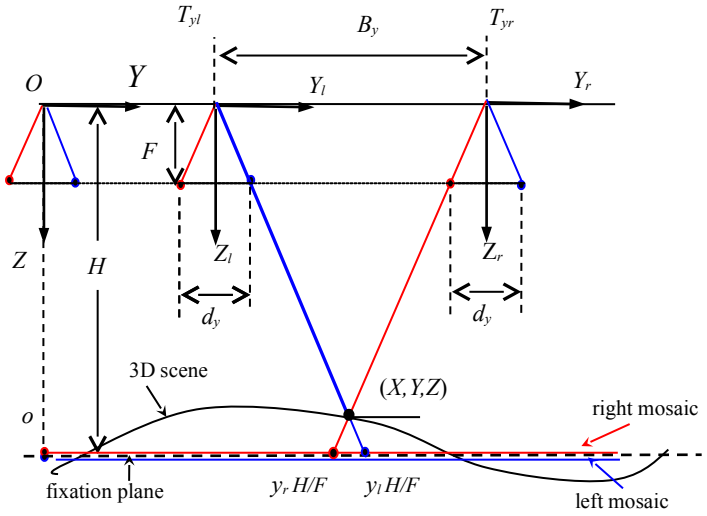

Fig. 1. Parallel-perspective stereo geometry. Both mosaics are built on the fixation plane, but their unit is in pixel – each pixel represents H/F world distances.

Without loss of generality, we assume that two vertical 1-column slit windows have $d_y/2$ offsets to the left and right of the center of the image respectively (Fig. 1). The "left eye" view (left mosaic) is generated from the front slit window, while the "right eye" view (right mosaic) is generated from the rear slit window. The *parallel-perspective projection model* of the stereo mosaics thus generated can be represented by the following equations [6]

$$\begin{aligned} x_l &= x_r = F\,X/Z \\ y_r &= FY/H + (Z/H\text{-}1)\,d_y/2 \\ y_l &= FY/H - (Z/H\text{-}1)\,d_y/2 \end{aligned} \qquad (1)$$

where F is the focal length of the camera, H is the height of a *fixation plane* (e.g., average height of the terrain). Eq.(1) gives the relation between a pair of 2D points (one from each mosaic), (x_l,y_l) and (x_r,y_r), and the corresponding 3D point (X,Y,Z). It serves a function similar to the classical pin-hole perspective camera model. The depth can be computed as (from Eq. (1))

$$Z = H\frac{b_y}{d_y} = H(1+\frac{\Delta y}{d_y}) \qquad (2)$$

where

$$b_y = d_y + \Delta y = FB_y/H \qquad (3)$$

is the "scaled" version of the baseline B_y, and

$$\Delta y = y_r - y_l \qquad (4)$$

is the "mosaic displacement"[1] in the stereo mosaics. Displacement Δy is a function of the depth variation of the scene around the fixation plane H. Since a fixed angle between the two viewing rays is selected for generating the stereo mosaics, the "disparities" (d_y) of all points are fixed; instead a geometry of optimal/adaptive baselines (b_y) for all the points is created. In other words, for any point in the left mosaic, searching for the match point in the right mosaic means finding an original frame in which this match pair has a pre-defined disparity (by the distance of the two slit windows) and hence has an adaptive baseline depending on the depth of the point (Fig. 1).

3. Stereo Mosaicing from Real Video

In the PRISM approach for video mosaicing and 3D reconstruction from real video, the computation of "matching" is efficiently distributed in three steps: camera pose estimation, image mosaicing and 3D reconstruction. In estimating camera poses (for image rectification), only sparse tie points widely distributed in the two images are needed. In generating stereo mosaics, matches are only performed for parallel-perspective rays between small overlapping regions of successive frames. In using stereo mosaics for 3D recovery, matches are only carried out between the two final mosaics. This section gives a brief summary of the techniques in the three steps, which have been discussed in detail can be found in [6,7].

3.1. Image rectification

The stereo mosaicing mechanism can be generalized to the case of 3D translation if the 3D curved motion track has a dominant translational motion for generating a parallel projection in that direction [7]. Under 3D translation, seamless stereo mosaics can be generated in the same way as in the case of 1D translation. The only difference is that viewpoints of the mosaics form a 3D curve instead of a 1D straight line. Further, the motion of the camera can be generalized to a 6 DOF motion with some reasonable constraints on the values and rates of changes of motion parameters of a camera [6,7] (Fig. 3a), which are satisfied by a sensor mounted in an air or ground vehicle, or a sensor looking at the underside of a moving vehicle. There are two steps necessary to generate a rectified image sequence that exhibits only 3D translation, from which we can generate seamless mosaics:

1) *Camera orientation estimation*. Assuming an internally pre-calibrated camera, the extrinsic camera parameters (camera orientations) can be determined from a bundle adjustment technique [16]. The detail is out the scope of this report, but the main point here is that we do not need to carry out dense match between two successive frames. Instead only sparse tie points widely distributed in the two images are needed to estimate the camera orientations.

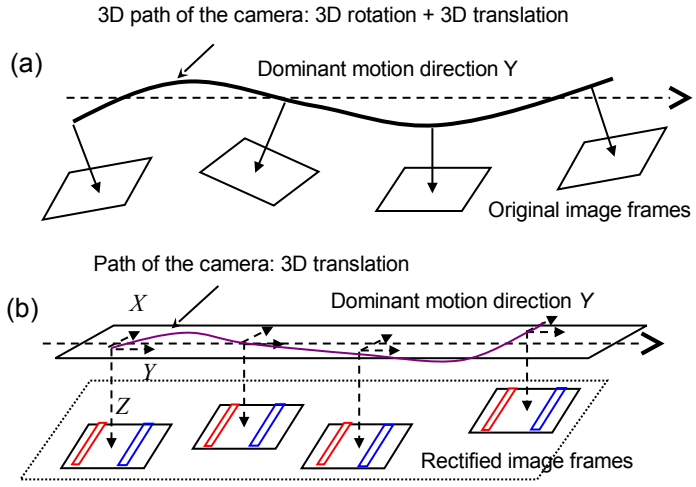

Fig. 3. Image rectification. (a) Original and (b) rectified image sequence.

2) *Image rectification*. A 2D projective transformation is applied to each frame in order to eliminate the rotational components(Fig. 3b). In fact we only need to do this kind of transformation on two narrow slices in each frame that will

[1] We use "displacement" instead of "disparity" since it is related to the baseline in a two view-perspective stereo system.

contribute incrementally to each of the stereo mosaics. The 3D motion track formed by the viewpoints of the moving camera will have a dominant motion direction (Y) that is perpendicular to the optical axis of the "rectified" images.

3.2. Ray interpolation

How can we generate seamless mosaic from video of a translating camera in a computational effective way? The key to our approach lies in the parallel-perspective representation and an interframe ray interpolation approach. For each of the left and right mosaics, we only need to take a front (or rear) slice of a certain width (determined by interframe motion) from each frame, and perform local registration between the overlapping slices of successive frames (Fig. 4), then generate parallel *interpolated rays* between two known discrete perspective views for the left (or right) mosaic.

Let us examine this idea more rigorously in the case of 2D translation after image rectification when the translational components in the Z direction is small [6]. We take the left mosaic as an example (Fig. 4). First we define the central column of the front (or rear) mosaicing slice in each frame as a *fixed line*, which has been determined by the camera's location of each frame and the pre-selection of the front (or rear) slice window (Fig. 4, Fig. 5). An interpretation plane (IP) of the fixed line is a plane passing through the nodal point and the fixed line. By the definition of parallel-perspective stereo mosaics, the IPs of fixed lines for the left (or right) mosaic are parallel to each other. Suppose that (S_x, S_y) is the translational vector of the camera between the previous (1st) frame of viewpoint (T_x, T_y) and the current (2nd) frame of view point (T_x+S_x, T_y+S_y) (Fig. 4). We need to interpolate parallel rays between the two *fixed lines* of the 1st and the 2nd frames. For each point (x_1, y_1)(to the right of the 1st fixed line $y_0=d_y/2$) in frame(T_x, T_y), which will contribute to the left mosaic, we can find a corresponding point (x_2, y_2) (to the left of the 2nd fixed line) in frame (T_x+S_x, T_y+S_y). We assume that (x_1, y_1) and (x_2, y_2) are represented in their own frame coordinate systems, and intersect at a 3D point (X,Y,Z). Then the parallel reprojected viewpoint (T_{xi}, T_{yi}) of the correspondence pair can be computed as

$$T_{yi} = T_y + \frac{(y_1 - d_y/2)}{y_1 - y_2} S_y, \quad T_{xi} = T_x + \frac{S_x}{S_y}(T_{yi} - T_y) \qquad (5)$$

where T_{yi} is calculated in a synthetic IP that passes through the point (X,Y,Z) and is parallel to the IPs of the fixed lines of the first and second frames, and T_{xi} is calculated in a way that all the viewpoints between (T_x,T_y) and (T_x+S_x, T_y+S_y) lie in a straight line. Note that Eq. (6) also holds for the two fixed lines such that when $y_1 = d_y/2$ (the first fixed line), we have (T_{xi}, T_{yi})=(T_x, T_y), and when $y_2 = d_y/2$ (the second fixed line), we have (T_{xi}, T_{yi})=(T_x+S_x, T_y+S_y). We assume that normally the interframe motion is large enough to have $y_1-1 \geq d_y/2 \geq y_2+1$. A super dense image sequence could generate a pair of stereo mosaics with super-resolution, but this will not be discussed in this paper.

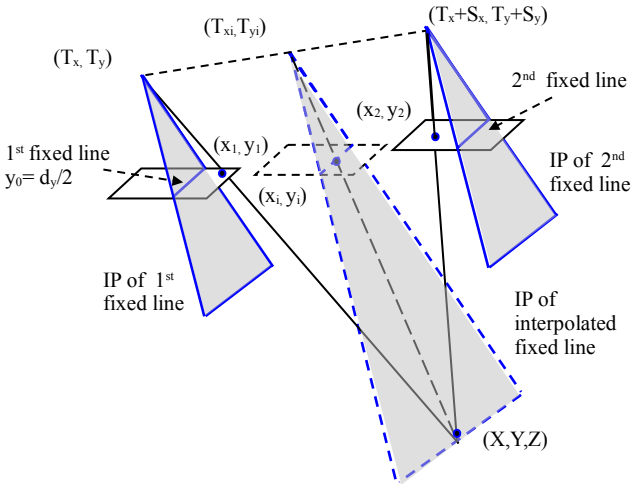

Fig. 4. View interpolation by ray re-projection

The reprojected ray of the point (X,Y,Z) from the interpolated viewpoint (T_{xi}, T_{yi}) is

$$(x_i, y_i) = [x_1 - \frac{S_x}{S_y}(y_1 - \frac{d_y}{2}), \frac{d_y}{2}] \qquad (6)$$

and the mosaicing coordinates of this point is

$$(x_m,\ y_m) = [t_{xi} + x_1 - \frac{S_x}{S_y}(y_1 - \frac{d_y}{2}),\ t_{yi} + \frac{d_y}{2}] \quad (7)$$

where

$$t_{xi} = F\,T_{xi}/H,\ t_{yi} = F\,T_{yi}/H. \quad (8)$$

are the "scaled" translational components of the interpolated view. Note that the interpolated rays are also parallel-perspective, with perspective in the *x* direction and parallel in the *y* direction.

3.3. 3D reconstruction from stereo mosaics

In the general case, the viewpoints of both left and right mosaics will be on the same smooth 3D motion track. Therefore the corresponding point in the right mosaic of any point in the left mosaic will be on an epipolar curve determined by the coordinates of the left point and the 3D motion track. We have derived the epipolar geometry of the stereo mosaics generated from a rectified image sequence exhibiting 3D translation with the *y* component dominant [7]. Under 2D translation (T_x, T_y), the corresponding point (x_r, y_r) in the right-view mosaic of any point (x_l, y_l) in the left-view mosaic will be constrained to an *epipolar curve*

$$\Delta x = b_x(y_l, \Delta y)\frac{\Delta y}{\Delta y + d_y}, \quad (9)$$
$$\Delta x = x_r - x_l,\ \ \Delta y = y_r - y_l$$

where $b_x(y_l, \Delta y) = [t_{xl}(y_l + d_y + \Delta y) - t_{xl}(y_l)]$ is the baseline function of y_l and Δy, and $t_{xl}(y_l)$ is the "scaled" *x* translational component (as in Eq. (3) or (8)) of the original frames corresponding to column y_l in the left mosaic. Hence Δx is a nonlinear function of position y_l as well as displacement Δy, which is quite different from the epipolar geometry of a two-view perspective stereo. The reason is that image columns of different y_l in parallel-perspective mosaics are projected from different viewpoints. In the ideal case where the viewpoints of stereo mosaics form a 1D straight line, the epipolar curves will turn out to be horizontal lines.

The depth maps of stereo mosaics were obtained by using the Terrest system designed for perspective stereo match[17] without modification. The Terrest system was designed to account the illumination differences and perspective distortion of stereo images with largely separated views by using normalized correlation and multi-resolution un-warping. Further work is needed to apply the epipolar curve constraints into the search of correspondence points in the Terrest to speedup the match process. Currently we perform matches with 2D search regions estimated from the motion track and the maximum depth variations of a scene.

4. Experimental Results

4.1. Comparison of 3D vs. 2D mosaicing

First, we show why "3D mosaicing" is so important for 3D reconstruction from stereo mosaics by a real example. Fig. 5 shows the local match and ray interpolation of a successive frame pair of a UMass campus scene, where the interframe motion is $(s_x, s_y) = (27, 48)$ pixels, and points on the top of a tall building (the Campus Center) have about 4 pixels of additional motion parallax. As we will see next, these geometric misalignments, especially of linear structures, will be clearly visible to human eyes. Moreover, perspective distortion causing the geometric seams will introduce errors in 3D reconstruction using the parallel-perspective geometry of stereo mosaics. In the example of stereo mosaics of the UMass campus scene [18], the distance between the front and the rear slice windows is $d_y = 192$ pixels, and the average height of the aerial camera from the ground is $H = 300$ meters (m). The relative *y* displacement of the building roof (to the ground) in the stereo mosaics is about $\Delta y = -29$ pixels. Using Eq. (2) we can compute that the "absolute" depth of the roof from the camera is $Z = 254.68$ m, and the "relative" height of the roof to the ground is $\Delta Z = 45.31$ m. A 4-pixel misalignment in the stereo mosaics will introduce a depth (height) error of $\delta Z = 6.25$ m, though stereo mosaics have rather large "disparity" ($d_y = 192$). While the relative error of the "absolute" depth of the roof ($\delta Z/Z$) is only about 2.45%, the relative error of its "relative" height ($\delta Z/\Delta Z$) is as high as 13.8%. This clearly shows that geometric-seamless mosaicing is very important for accurate 3D estimation as well as good visual appearance. It is especially true when sub-pixel accuracy in depth recovery is needed [17].

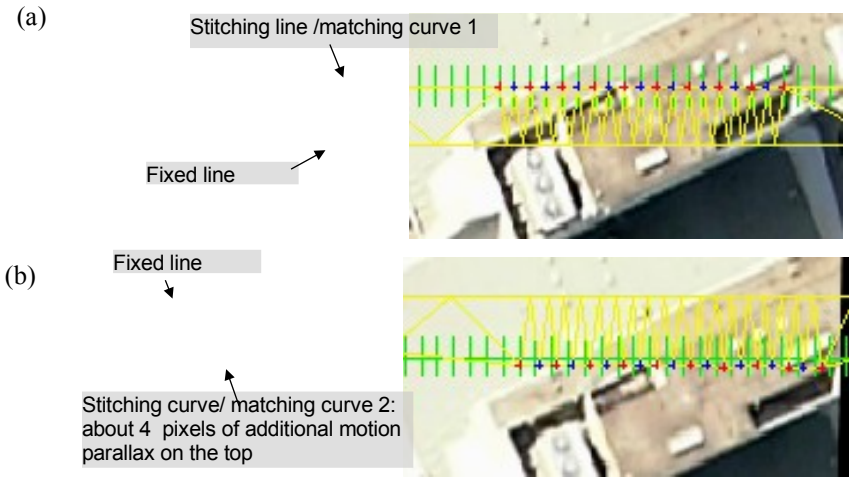

Fig. 5. Examples of local match and triangulation for the left mosaic. Close-up windows of (a)the previous and (b) the current frame. The green crosses show the initially selected points (which are evenly distributed along the ideal stitching line) in the previous frame and its initial matches in the current frame by using the global transformation. The blue and red crosses show the correct match pairs by feature selection and correlation (red matches red, blue matches blue). The fixed lines, stitching lines/curves and the triangulation results are shown as yellow.

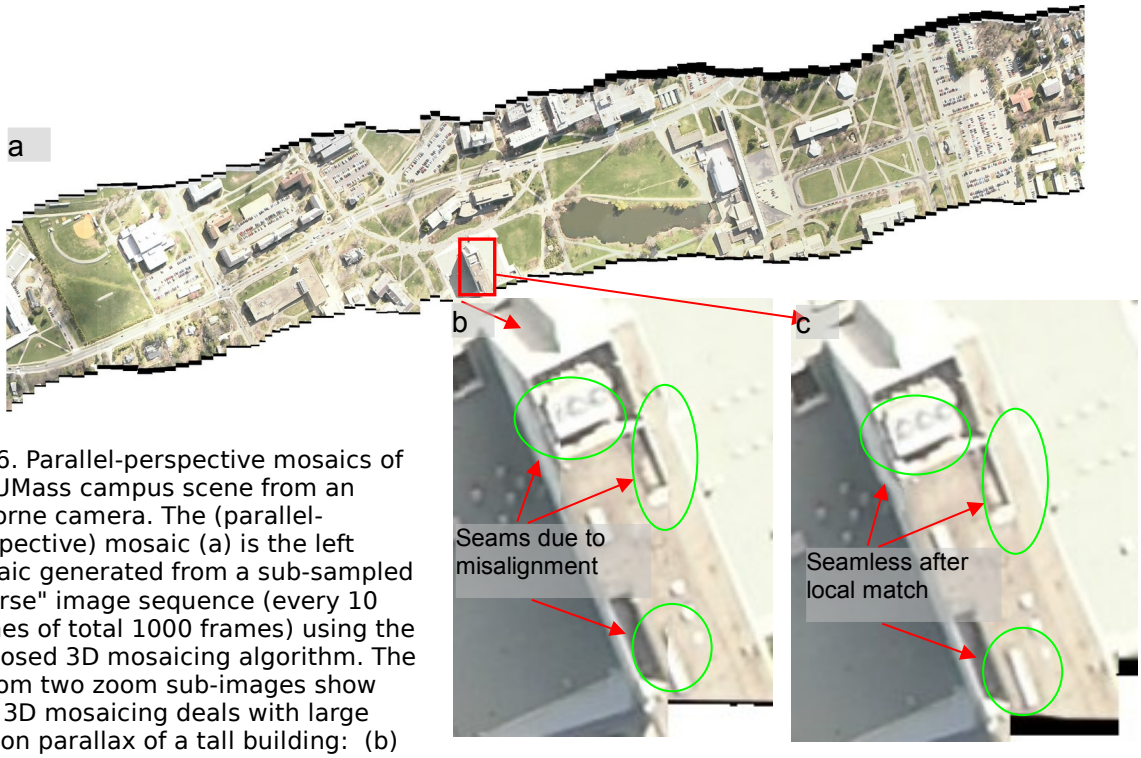

Fig. 6. Parallel-perspective mosaics of the UMass campus scene from an airborne camera. The (parallel-perspective) mosaic (a) is the left mosaic generated from a sub-sampled "sparse" image sequence (every 10 frames of total 1000 frames) using the proposed 3D mosaicing algorithm. The bottom two zoom sub-images show how 3D mosaicing deals with large motion parallax of a tall building: (b) 2D mosaic result with obvious seams (c) 3D mosaic result without seam.

In principle, we need to match all the points between the two fixed lines of the successive frames to generate a complete parallel-perspective mosaic. In an effort to reduce the computational complexity, we have designed a fast *3D mosaicing* algorithm [7] based on the proposed PRISM method. It only requires matches between a set of point pairs in two successive images around their *stitching line*, which is defined as a virtual line in the middle of the two fixed lines (see Fig. 5). The pair of *matching curves* in the two frames is then mapped into the mosaic as a *stitching curve* by using the ray interpolation equation (7). The rest of the points are generated by warping a set of triangulated regions defined

by the control points on the matching curve (that correspond to the stitching curve) and the fixed line in each of the two frames. Here we assume that each triangle is small enough to be treated as a planar region.

Using sparse control points and image warping, the proposed 3D mosaicing algorithm only approximates the parallel-perspective geometry in stereo mosaics (Fig. 6), but it is good enough when the interframe motion is small. Moreover, the proposed 3D mosaicing algorithm can be easily extended to use more feature points (thus smaller triangles) in the overlapping slices so that each triangle really covers a planar patch or a patch that is visually indistinguishable from a planar patch, or to perform pixel-wise dense matches to achieve true parallel-perspective geometry.

While we are still working on 3D camera orientation estimation using our instrumentation and the bundle adjustments [15], Fig. 6 shows mosaic results where camera orientations were estimated by registering the planar ground surface of the scene via dominant motion analysis. However the effect of seamless mosaicing is clearly shown in this example. Please compare the results of 3D mosaicing (parallel-perspective mosaicing) vs. 2D mosaicing (multi-perspective mosaicing) by looking along many building boundaries associating with depth changes in the entire 4160x1536 mosaics at our web site [18]. Since it is hard to see subtle errors in the 2D mosaics of the size of Fig. 6a, Fig. 6b and Fig. 6c show close-up windows of the 2D and 3D mosaics for the portion of the scene with the tall Campus Center building. In Fig. 6b the multi-perspective mosaic via 2D mosaicing has obvious seams along the stitching boundaries between two frames. It can be observed by looking at the region indicated by circles where some fine structures (parts of a white blob and two rectangles) are missing due to misalignments. As expected, the parallel-perspective mosaic via 3D mosaicing (Fig. 6c) does not exhibit these problems.

4.2. Tests for UVIS sequences

For the UVIS, the data processing system is responsible for building mosaics from a collection of video frames, which have been stored in memory by the image acquisition system. In order to study how to apply the PRISM approach to UVIS, we tested the PRISM algorithm on many single-camera sequences of digital images using the laboratory prototype. Cameras moved across the bottom of the "car" when capturing the image sequences. In the current implementation, we used a real-time global image matching technique (rather than bundle adjustment) for determining the relative positions of each frames. In this technique, four parameters are used to characterize the position and orientation of the camera at each frame – two translation components in the row and scan directions, one rotation angle around the vertical axis, and a scaling factor accounting for the depth changes. A pyramid-based matching algorithm was used to determine the relative motion parameters between a pair of successive frames.

We studied when the mosaicing algorithm would break by simulating different distances between consecutive "camera locations" when capture the image sequences. We also studied how good the mosaics would be with and without using 3D mosaicing. Figures 7-12 show the mosaicing results by using every frame, every other frame, every six frame, and every twelve frame of the same image sequence. This simulates the cases with different camera distances from 0.5 inches to 6 inches, corresponding to image displacements of from 8 to 100 pixels. We also compare the results with 2D mosaicing, 3D mosaicing using automatic local matching, and 3D mosaicing using manual local matching. Here are several important conclusions.

1. In the current setup where the average distance between cameras and the bottom of the car is about 32 inches, the break-down point of the mosaicing is when the distances between consecutive "camera locations" are larger than 6 inches (see Figures 7-11). Camera distance of 3-4 inches achieve a good balance between cost of computation and qualitiy of mosaics. Note that these numbers are connected with the setup of the system, but this condition approximately corresponds to a rather large image motion of 100 pixels between two successive frames.
2. The reason for the breakdown with large camera distance is that the system cannot effectively find the right camera location by the global image matching algorithm. Therefore the local matching algorithm in 3D mosaicing approach (PRISM) cannot make up the failure in the global matching anyway (see Fig. 11). However, if we can find the camera location by other means, for example, by calibrating the 1D camera array, or by using rigid motion constraints to estimate the 3D motion of the car, we can still build reasonably good mosaic with rather large camera distance. For example, when the camera distance is about 6 inches, 3D mosaicing results with manual local matches shows reasonably good stitching in mosaicing (Figure 12).
3. It turns out that when the image motion between two frames is small, we can apply a simple cut-and-paste algorithm (i.e., 2D mosaicing technique) to generate mosaics from the image sequence. In this set of specific

experiments, it does not make obvious differences in the mosaicing results by using 2D or 3D mosaicing techniques when the image motion is small (Figure 8, Figure 9).

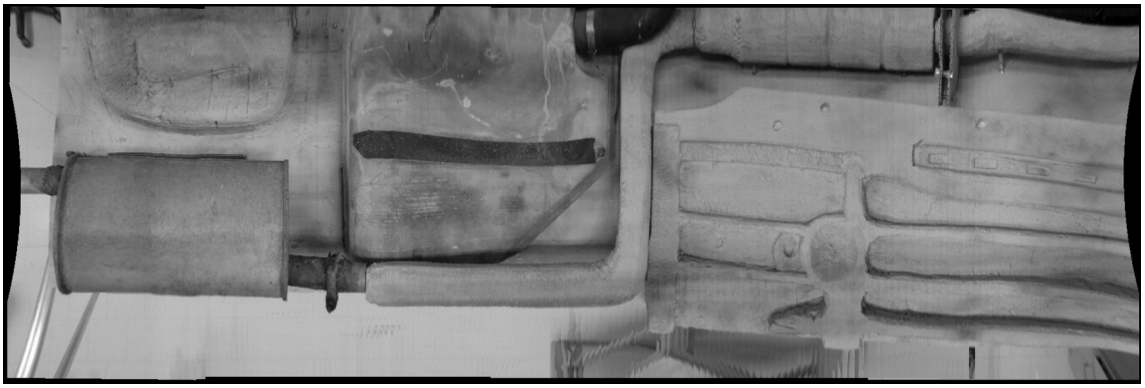

Figure 7. The 3D mosaicing result of image sequence with 0.5-inch camera distances. Since the image motion is only 8-pixel on average, the mosaic is almost perfect.

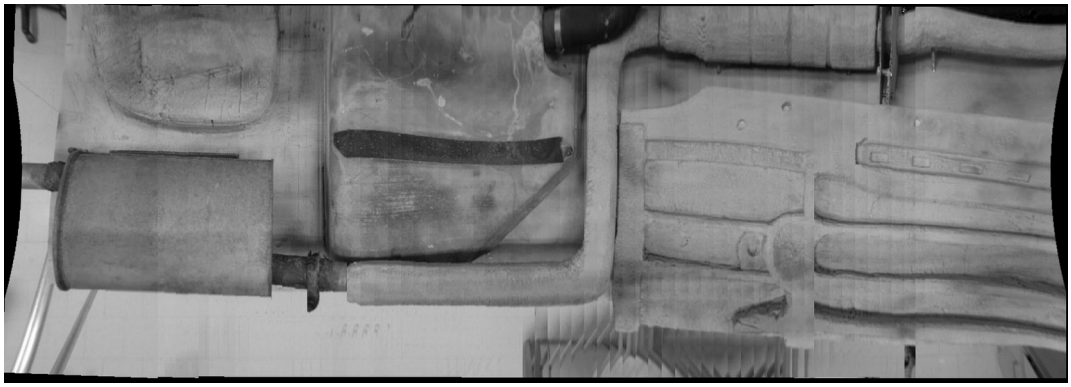

Figure 8. The 2D mosaicing result of image sequence with 1-inch camera distances (16-pixel image motion on average). The 2D mosaicing technique uses a rectangular cut-and-paste approach, so geometrical seams can be seen in some parts of the mosaic.

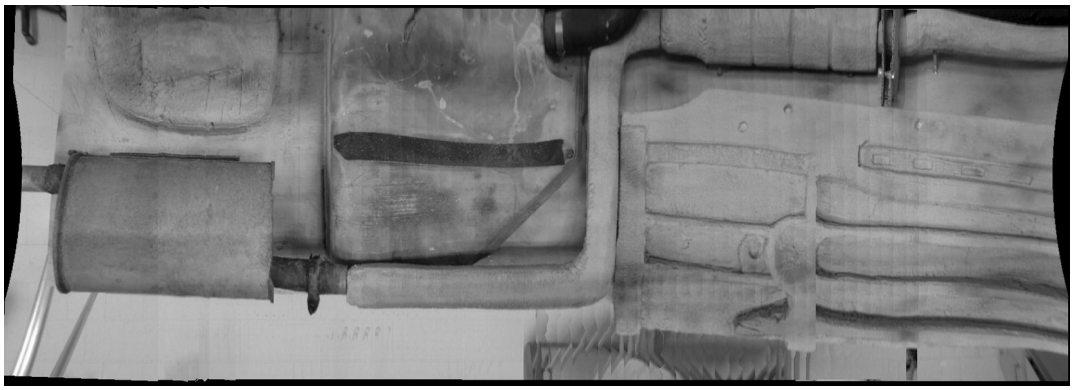

Figure 9. The 3D mosaicing result of image sequence with 1-inch camera distances (16-pixel image motion on average). Local match and view interpolation technique is applied between two successive slices therefore the mosaic is slightly better than the 2D mosaic (Fig. 8). However the cost is additional computation for local matching and view interpolation.

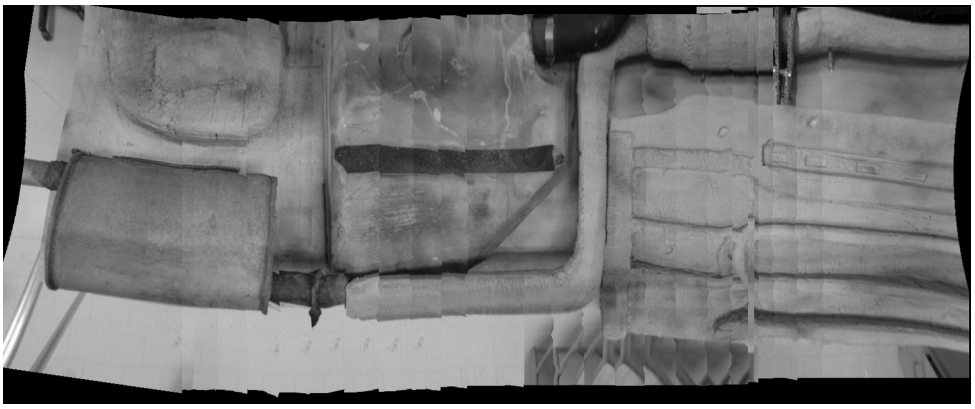

Figure 10. The 3D mosaicing result of image sequence with 3-inch camera distances (50-pixel image motion on average). The mosaic is not perfect but is good enough for understanding the under-vehicle.

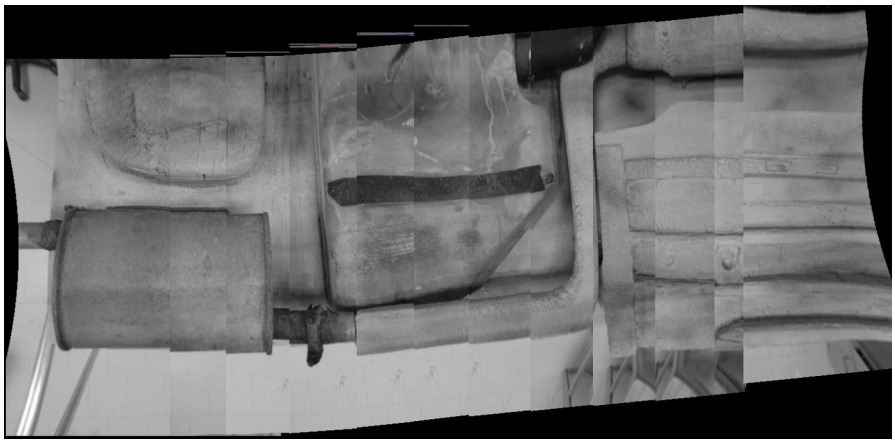

Figure 11. The 3D mosaicing results for image sequence with 6-inch camera distances (100-pixel image motion on average). The local matches were established by automatic method. Since the global matching algorithm fails to find the correct camera parameter in this case, the local matching step does not compensate the errors (e.g. the missing of some parts).

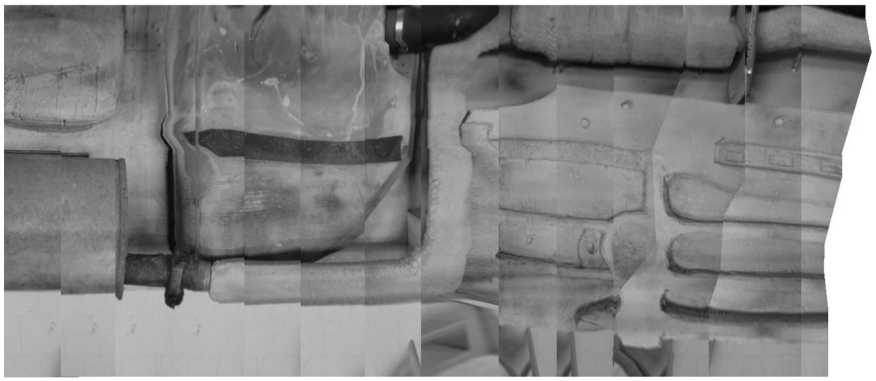

Figure 12. The 3D mosaicing result of image sequence with 6-inch camera distances (100-pixel image motion on average). The "camera locations" of the moving camera was estimated by a simple calibration procedure with metric measurements of the camera movements, and the local matches were established manually. The result shows that if we can develop an effective local matching algorithm for view interpolation, we can still make good image mosaic with rather sparse camera array (e.g. 6 inch apart).

5. Mosaics for UVIS with a Virtual bed-of-cameras

The algorithm design and implementation effort focused on generating mosaics from a *virtual bed-of-cameras*. Three technical issues were studied and algorithms were implemented and tested: (1) calibration of the 1D camera array (including lens distortion removal); (2) estimation of the motion of the vehicle while creating the mosaics and (3) seamless mosaicing from different viewpoints.

The first step is to remove the lens distortion from the images. Before the cameras are installed in the system, each camera goes through a calibration procedure in which their lens distortion characteristics are measured. The lens distortion characteristics are then used by the data processing system to remove the geometric distortion introduced by the lenses.

The mosaic generation program stitches the individual images from the camera array together to form a row mosaic. Three row mosaics are made – one from the left side of each frame, one from the center of each frame, and one from the right side of each frame. These mosaics are equivalent to looking at the undercarriage from the left, middle and right.

The mosaic generation continues the process in the along scan direction. The row mosaics, which are equivalent to a single view of the undercarriage, are then stitched together to form a complete view of the undercarriage. In a similar process to forming the row mosaics, three full view mosaics are created – one from the leading edge of the center row mosaic, one from the middle of the center row mosaic, and one from the trailing edge of the center row mosaic. The result is five images, which are equivalent to viewing the undercarriage from the left, right, front, back and middle.

A key feature of UVIS is its ability to capture a full view of the undercarriage from multiple viewpoints. Thus, if an object were occluded in one view, it would likely be visible in another view. This is a remarkable result, which will significantly enhance the capabilities of the proposed system.

To study some of these issues we captured several single-camera sequences of digital images using the laboratory prototype. In the current implementation, we used the same technique for determining the relative positions of each frames – both for the raw mosaics and in the along scan direction. We used four parameters to characterize the position and orientation of the camera at each frame – two translation components in the row and scan directions, one rotation angle around the vertical axis, and a scaling factor accounting for the depth changes. Again, a pyramid-based matching algorithm was used to determine the relative motion parameters between a pair of successive frames. Because the image motion between two frames was small, we applied a simple adaptive cut-and-paste algorithm to generate each mosaic from the set of images, as indicated in Conclusion 3. The results were very encouraging. Although some minor geometric distortion was observed, we clearly demonstrated the ability of the UMass mosaic algorithm to fast create smooth, topologically correct mosaics under each of the test conditions. Figure 13 shows a test mosaic made from 4 cameras spaced 4 inches apart with a 45 degree field-of-view.

We also studied the seamless mosaicing issue in case of larger motion between frames. Two aspects to this issue were investigated. First, we studied adaptive window matching techniques, which have been shown to be robust for matching scenes with weak texture. The importance of the matching algorithm has been shown in Figure 12 for view interpolation. Second, we tested the UMass fast PRISM algorithm using surface triangulation as we did in Section 4.2. We tested both algorithms for some examples of under-vehicle images as well as other type of images, and obtained promising results (Figures 6-12). Further investigation and implementation is needed in the phase II.

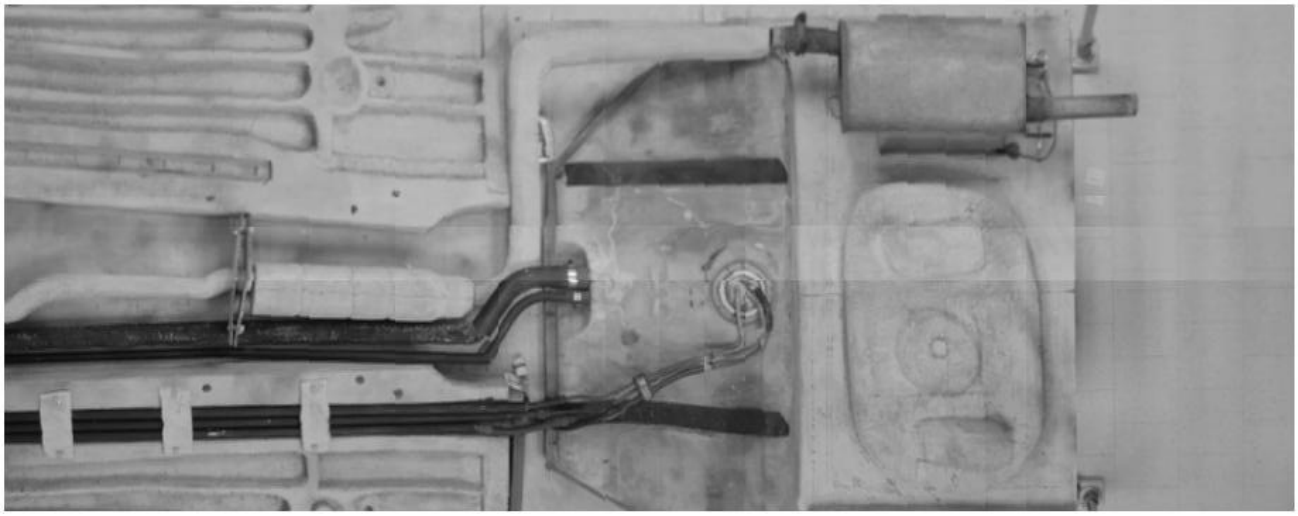

Figure 13. A test mosaic made from 4 cameras spaced 4 inches apart with a 45 degree field-of-view

References

[1]. H.-C. Huang and Y.-P. Hung, Panoramic stereo imaging system with automatic disparity warping and seaming, *Graphical Models and Image Processing*, 60(3): 196-208, 1998.

[2]. H. Ishiguro, M. Yamamoto, and Tsuji, Omni-directional stereo for making global map, *ICCV'90*, 540-547.

[3]. S. Peleg, M. Ben-Ezra, Stereo panorama with a single camera, *CVPR'99*: 395-401

[4]. H. Shum and R, Szeliski, Stereo reconstruction from multiperspective panoramas, *Proc. IEEE ICCV99*, 14-21, 1999.

[5]. Z. Zhu, A. R. Hanson, H. Schultz, F. Stolle, E. M. Riseman, Stereo mosaics from a moving video camera for environmental monitoring, *Int. Workshop on Digital and Computational Video*, 1999, Tampa, Florida, pp 45-54.

[6]. Z. Zhu, E. M. Riseman, A. R. Hanson, Theory and practice in making seamless stereo mosaics from airborne video, *Technical Report #01-01*, CS Dept., UMass-Amherst, Jan. 2001 (http://www.cs.umass.edu/~zhu/UM-CS-2001-001.pdf).

[7]. Z. Zhu, E. M. Riseman, A. R. Hanson, Parallel-perspective stereo mosaics, In *ICCV'01*, Vancouver, Canada, July 2001.

[8]. J. Y. Zheng and S. Tsuji, Panoramic representation for route recognition by a mobile robot. IJCV 9 (1), 1992, 55-76

[9]. R. Kumar, P. Anandan, M. Irani, J. Bergen and K. Hanna, Representation of scenes from collections of images, In *IEEE Workshop on Presentation of Visual Scenes*, 1995: 10-17.

[10]. J. Chai and H. -Y. Shum, Parallel projections for stereo reconstruction, *CVPR'00*: II 493-500.

[11]. S. Peleg, J. Herman, Panoramic Mosaics by Manifold Projection. *CVPR'97*: 338-343.

[12]. H.S. Sawhney, Simplifying motion and structure analysis using planar parallax and image warping. ICPR'94: 403- 408

[13]. R. Szeliski and S. B. Kang, Direct methods for visual scene reconstruction, In *IEEE Workshop on Presentation of Visual Scenes,* 1995: 26-33

[14]. R. Gupta , R. Hartley, Linear pushbroom cameras, *IEEE Trans PAMI*, 19(9), Sep. 1997: 963-975

[15]. Schultz, H., Hanson, A., Riseman, E., Stolle, F., Zhu. Z., A system for real-time generation of geo-referenced terrain models, *SPIE Symposium on Enabling Technologies for Law Enforcement*, Boston MA, Nov 5-8, 2000

[16]. C. C. Slama (Ed.), Manual of Photogrammetry, Fourth Edition, *American Society of Photogrammetry*, 1980.

[17]. H. Schultz. Terrain Reconstruction from Widely Separated Images, In *SPIE*. Orlando, FL, 1995.

[18]. Z. Zhu, PRISM: Parallel ray interpolation for stereo mosaics, http://www.cs.umass.edu/~zhu/StereoMosaic.html