

*ARMY RESEARCH LABORATORY*



**Automatic Mosaicking of 360° Panorama  
in Video Surveillance**

**by Sean Ho and Philip David**

**ARL-TR-4670**

**December 2008**

## **NOTICES**

### **Disclaimers**

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

**Army Research Laboratory**

Adelphi, MD 20783-1197

---

---

**ARL-TR-4670**

**December 2008**

---

**Automatic Mosaicking of 360° Panorama  
in Video Surveillance**

**Sean Ho and Philip David**  
**Computational and Information Sciences Directorate, ARL**

---

---

**Approved for public release; distribution unlimited.**

---

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188		
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. <b>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</b>					
1. REPORT DATE (DD-MM-YYYY) December 2008		2. REPORT TYPE Final		3. DATES COVERED (From - To) October 2007 to September 2008	
4. TITLE AND SUBTITLE Automatic Mosaicking of 360° Panorama in Video Surveillance			5a. CONTRACT NUMBER		
			5b. GRANT NUMBER		
			5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S) Sean Ho and Philip David			5d. PROJECT NUMBER		
			5e. TASK NUMBER		
			5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Research Laboratory ATTN: AMSRD-ARL-CI-IA 2800 Powder Mill Road Adelphi, MD 20783-1197			8. PERFORMING ORGANIZATION REPORT NUMBER ARL-TR-4670		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSOR/MONITOR'S ACRONYM(S)		
			11. SPONSOR/MONITOR'S REPORT NUMBER(S)		
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT Recently, there has been an increasing interest in using panoramic images in surveillance and target tracking applications. With the wide availability of off-the-shelf, Web-based pan-tilt-zoom (PTZ) cameras and the advances of CPUs and Graphics Processing Units (GPUs), object tracking using mosaicked images that cover a scene of 360° in near real-time has become a reality. This report presents a system that automatically constructs and maps full-view panoramic mosaics to a cube-map from images captured from an active PTZ camera with 1–25× optical zoom. A hierarchical approach is used in storing and mosaicking multi-resolution images captured from a PTZ camera. Techniques based on scale-invariant local features and probabilistic models for verification are used in the mosaicking process. Our algorithm is automatic and robust in mapping each incoming image to one of the six faces of a cube with no prior knowledge of the scene structure. This work can be easily integrated to a surveillance system that wishes to track moving objects in its 360° surroundings.					
15. SUBJECT TERMS Image mosaicking, surveillance					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 20	19a. NAME OF RESPONSIBLE PERSON Sean Ho
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER (Include area code) (301) 394-3927

---

## Contents

---

<b>List of Figures</b>	<b>iv</b>
<b>1. Introduction</b>	<b>1</b>
<b>2. Background Theory</b>	<b>2</b>
2.1 Camera Model .....	2
2.2 Camera Rotation.....	2
2.3 Distortion.....	3
<b>3. Methodology</b>	<b>4</b>
3.1 Cube-map .....	4
3.2 Image Acquisition .....	6
3.3 Feature Extraction .....	7
3.4 Feature Matching.....	8
3.5 Outliers Screening.....	8
3.6 Warping.....	9
3.7 Blending.....	9
<b>4. Results</b>	<b>10</b>
<b>5. Conclusion</b>	<b>10</b>
<b>6. References</b>	<b>12</b>
<b>Acronyms</b>	<b>13</b>
<b>Distribution</b>	<b>14</b>

---

## List of Figures

---

Figure 1. Images mosaicked to the faces of a cube, showing (a) the SONY camera, mounted to the rooftop that was used in the experiment; and (b) the back, left, front, right, and bottom faces mapped onto a cube and (c) unfolded onto a plane. ....	1
Figure 2. (a) Camera rotation model and (b) cube-map representation. ....	3
Figure 3. The 640 x 480 image data needed to fill a cube-map, showing (a) the camera's FOV at various zoom settings and (b) the amount of image data (3 bytes per pixel RGB images) needed to build one face of the mosaic cube. ....	5
Figure 4. Flowchart showing the methods used in the construction of a mosaic image on a cube face. ....	5
Figure 5. Images acquired in the construction of the front face. The order from which image is acquired and stitched is shown by the first number followed by the camera's pan and tilt angles ( $p,t$ ). ....	6
Figure 6. Mosaicking of the front face: (a) New image captured at (0, 15); (b) a submosaic is selected from the mosaic at (0,15) as indicated by the green ROI box; (c) matched features between the new image (top) and the submosaic (bottom); and the (d) mosaicked image. ....	7
Figure 7. All four sides of the unfolded cube. ....	10

---

## 1. Introduction

---

The Army is increasingly interested in ground-based assets for use in urban Reconnaissance, Surveillance, and Target Acquisition (RSTA) applications. Persistent or continuous surveillance is a crucial part of this RSTA operation. Being able to detect threats early provides opportunities to neutralize the threats before they occur. Our goal is to develop a wide-area, 360° field of view (FOV) target detection and acquisition system that is suitable for an urban environment. An omnidirectional camera can image a scene with a full 360° FOV; however, it comes with a high price tag, limited resolution, and substantial image distortion. On the other hand, an off-the-shelf pan-tilt-zoom (PTZ) camera is affordable and scalable with low image distortion. The drawback in using a PTZ camera is that the entire scene cannot be captured from a single image, and the delays introduced in panning and tilting can be an issue for certain applications. We opt to use a PTZ camera, instead of an omnidirectional camera, for its advantage of being able to provide a virtual 360° FOV while behaving as a high-resolution synthetic ultra wide-angle camera. Using a PTZ camera, however, brings up the problem of image mosaicking. In a complex and changing scene, the mosaicking method should be efficient and robust against illumination variations, moving objects, camera rotations, zooming, and other unexpected changes in the scene.

In this report, we will present the method and image mosaicking algorithms that are used to project a complete 360° panorama of the scene onto a cube (figure 1(b) and (c)). The work presented here is part of a RSTA system that is being developed.

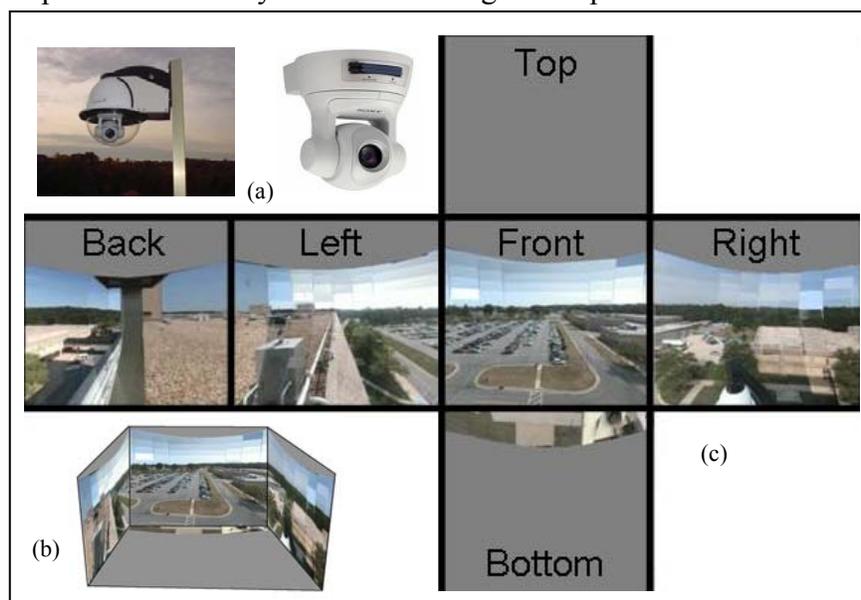


Figure 1. Images mosaicked to the faces of a cube, showing (a) the SONY camera, mounted to the rooftop that was used in the experiment; and (b) the back, left, front, right, and bottom faces mapped onto a cube and (c) unfolded onto a plane.

---

## 2. Background Theory

---

### 2.1 Camera Model

In our experiment, a SONY SNC-RZ30 PTZ network camera was used in acquiring all of our live images. The camera is stationary and mounted to a pole on the rooftop (see figure 1(a)). Because the distance of the scene from the camera is large, we can safely assume that the center of rotation of the camera is fixed and coincides with the camera's center of projection.

For the perspective camera model of a pinhole camera, a point  $\mathbf{X} = (X, Y, Z, 1)$  in three-dimensional (3-D) projective space  $\mathbb{P}^3$  projects to a point  $(x, y, 1)$  on the two-dimensional (2-D) image plane  $\mathbb{P}^2$ . This can be represented by a mapping from  $\mathbb{P}^3$  to  $\mathbb{P}^2$  such that  $(x, y, 1)^T = P(X, Y, Z, 1)^T$ , where  $P$  is a 3x4 camera projection matrix of rank-3. The matrix  $P$  can be written as

$$P = K[R \mid -R\mathbf{t}], \text{ where } K = \begin{bmatrix} \alpha f & s & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

and  $R$  and  $\mathbf{t}$  represent the camera rotation matrix and translation vector, respectively, in the world coordinate system. The  $K$  in equation 1 is the camera calibration matrix, which maps the camera coordinate to the image coordinate in pixels. The calibration matrix is made up of the camera intrinsic parameters  $\alpha, s, f, c_x,$  and  $c_y$ . Parameters  $\alpha$  and  $s$  are constants representing the camera's pixel aspect ratio and skew, respectively. Zooming has no effect on these two values.  $f$  is the focal length of the camera at the current zoom setting in pixels.  $(c_x, c_y)$  is the camera principal point in pixel coordinates where the optical axis intersects the image plane. For cameras with fixed optics, none of the parameters changes from one image to the next. For cameras with zoom capability, however, both the focal length and the principal point change with zoom. For most cameras, the skew  $s$  is very close to zero since pixels on the charge coupled device (CCD) are almost perfectly rectangular. In our experiment, these camera intrinsics were pre-computed at various zoom settings from collections of images of a planar checkerboard held at different orientations ( $I$ ).

### 2.2 Camera Rotation

In the case where the camera is stationary, only rotation is possible; we can safely set the translation vector  $\mathbf{t}$  to  $\mathbf{0}$ . Let  $\mathbf{x}$  and  $\mathbf{x}'$  be images of the 3-D scene point  $\mathbf{X}$  in images  $I$  and  $I'$ , respectively, taken at different times with rotation and zooming (see figure 2(a)). Applying equation 1, we can express  $\mathbf{x}$  and  $\mathbf{x}'$  as  $\mathbf{x} = KR\mathbf{X}$  and  $\mathbf{x}' = K'R'\mathbf{X}$ , and via substitution,  $\mathbf{x}' = KR'R^{-1}K^{-1}\mathbf{x}$ . When the camera undergoes pure rotation at a fixed zoom, its calibration matrix does not change. The equation can be simplified to

$$\mathbf{x}' = KR_{\text{rel}}K^{-1}\mathbf{x} \quad (2)$$

where  $R_{\text{rel}} = R'R^{-1}$  represents the relative camera rotation about its projection center between the two views.  $KR_{\text{rel}}K^{-1}$  is a 3x3 matrix defining a homography from any point  $\mathbf{x}$  in one image to the corresponding point  $\mathbf{x}'$  in the other image.

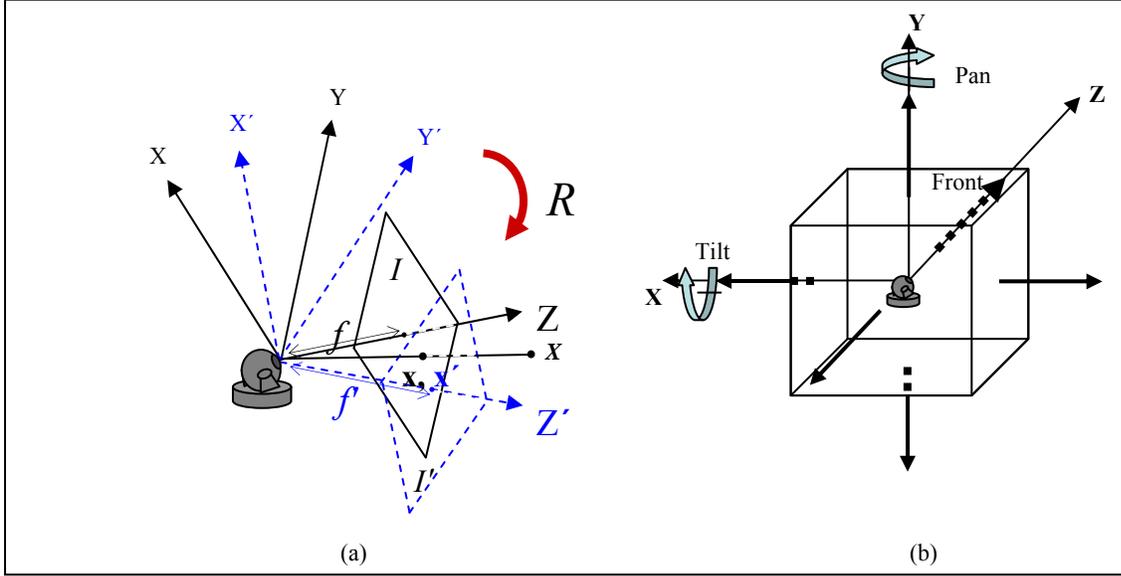


Figure 2. (a) Camera rotation model and (b) cube-map representation.

### 2.3 Distortion

Radial distortion exists in even the most expensive lenses. Its effect increases with decreasing focal length or wider FOV. The removal of the radial distortion is performed by determining two distortion coefficients,  $k_1$  and  $k_2$ , in a parametric radial distortion model. We assume that the center of radial distortion is the image principal point  $(c_x, c_y)$ . In our experiment,  $k_1$  and  $k_2$  are determined by manual camera calibration at various zoom levels using a planar checkerboard. Interpolation is used to determine  $k_1$  and  $k_2$  between calibration points. Let  $(x_d, y_d)$  be a measured, radially distorted image point in camera coordinates. In our distortion model, the distortion-free image point in camera coordinates is

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} c_x \\ c_y \end{pmatrix} + (1 + k_1 r^2 + k_2 r^4) \begin{pmatrix} x_d - c_x \\ y_d - c_y \end{pmatrix} \quad (3)$$

where  $r^2 = (x_d - c_x)^2 + (y_d - c_y)^2$ . The corrected, distortion-free point in pixel coordinates is then  $(u, v, 1)^T = K(x, y, 1)^T$ .

---

### 3. Methodology

---

#### 3.1 Cube-map

A cube is often used to represent and reflect the world in environment mapping. This technique is often used in computer graphics to render the 3-D environment around a viewpoint (2). In order to represent or reflect a full 360° view of a scene, we first construct a cube (see figure 2(b)) with the camera located at the center of the cube. This is done by projecting a 90° horizontal and vertical FOV to each of its six faces. Each face of the cube represents a mosaicked image composed of all projected scene points within the 90° FOV. The top and bottom images may not be needed since they encompass mostly the sky and ground. Cube-mapping has several advantages over using a single plane or a spherical representation. These advantages include less perspective distortion and Graphics Processing Unit (GPU) support for plane-to-plane mapping, and easy implementation and adaption to more planar faces (octahedron, dodecahedron, etc.). Another advantage is that higher resolution cube-maps generated at higher zoom settings can be linearly aligned to the base cube-map taken at the lowest zoom setting (3). With the SONY SNC-RZ30 network camera set to acquire 640 × 480 resolution images at its widest zoom (1×), which produces a FOV of 45°, we can determine the number of bytes of image data that is required to fill a face of the cube. From figure 3(a), one can see that  $\tan(\theta/2) = 320/f$  and  $\tan(45^\circ) = (S/2)/f$ , where  $S$  is the width and height of one face of the cube in pixels, and  $\theta$  is the camera FOV in degrees. Solving for  $S$  as a function of  $\theta$ , we obtain

$$S = \frac{640 \tan(45^\circ)}{\tan(\theta/2)} \quad (4)$$

Figure 3(b) shows the relationship between the amount of image data (3 bytes per pixel red, green, blue (RGB) images) that is required to fill a face for various camera FOVs. This table shows that almost 4 GB of data are required to build one face of a mosaic cube when the camera is zoomed into a FOV of 2°.

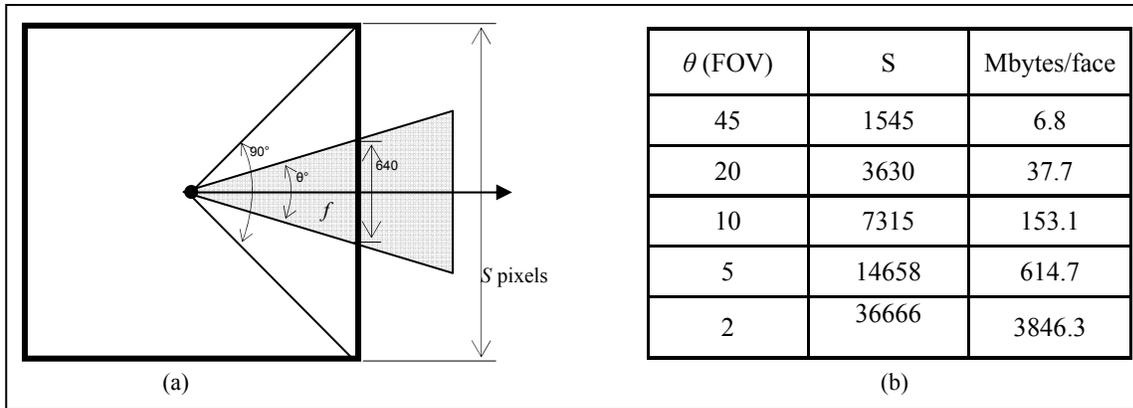


Figure 3. The 640 x 480 image data needed to fill a cube-map, showing (a) the camera's FOV at various zoom settings and (b) the amount of image data (3 bytes per pixel RGB images) needed to build one face of the mosaic cube.

In the following sections, we present the method that is used in building the cube-map mosaic. The proposed method can be divided into six stages: image acquisition, feature detection, feature matching, estimation of the geometric transformation, image warping, and image blending. Each stage is described below and the overall system diagram is shown in figure 4.

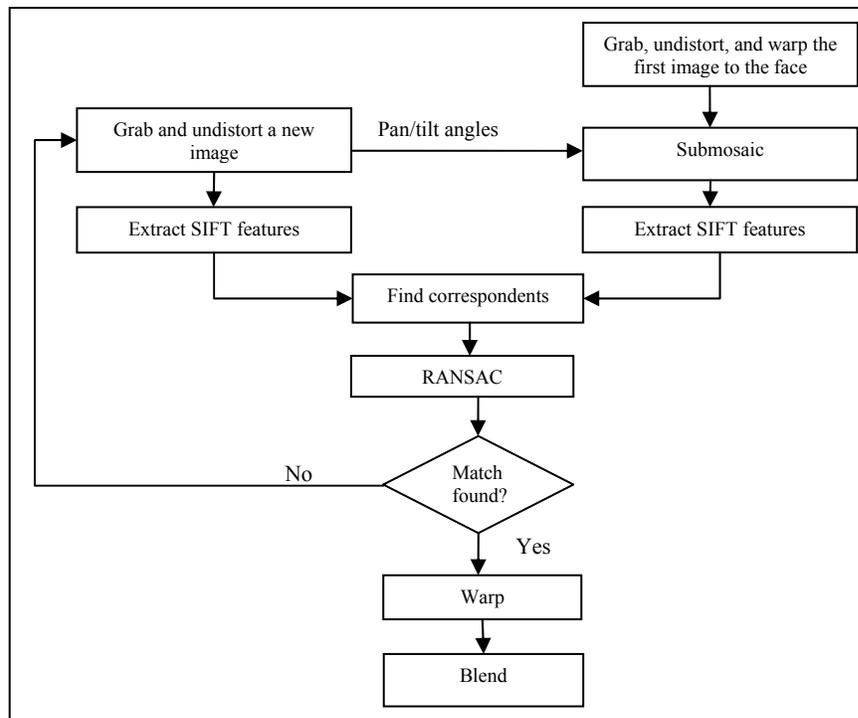


Figure 4. Flowchart showing the methods used in the construction of a mosaic image on a cube face.

Note: SIFT = Scale-Invariant Feature Transform.

Once the mosaic cube is constructed, any relevant side of the cube can be updated with a newly acquired image using the same six steps as outlined in the flowchart and described in sections 3.2

through 3.7. In determining side(s) of the mosaic cube that need(s) to be updated given a new image, the following equation is used:

$$\theta = \cos^{-1}(\mathbf{v} \cdot \mathbf{u}) \quad (5)$$

where  $\mathbf{v}$  is the unit vector parallel to the camera's optical axis and  $\mathbf{u}$  is the unit vector orthogonal to a face of the cube. When  $\theta$  is less than  $45^\circ$  plus half of the camera's current FOV, the respective mosaic-map for that side of the cube is updated.

### 3.2 Image Acquisition

Nine images are required in the construction of one face of the cube-map (figure 5). The first image for each face of the cube is the image that is normal to the camera optical axis; this image is inserted at the center of the face. All subsequent eight images that make up the mosaic for a face are registered to this first image. Each image is undistorted after it is acquired using the distortion coefficients obtained in section 2.3. In order to speed up the feature extraction process in the next stage, a submosaic is cropped from the mosaic. The submosaic is a rectangular region of interest (ROI) cropped from the existing mosaic image (see figure 6(b)). It is obtained from equation 2 based on the known approximate rotation of the camera.

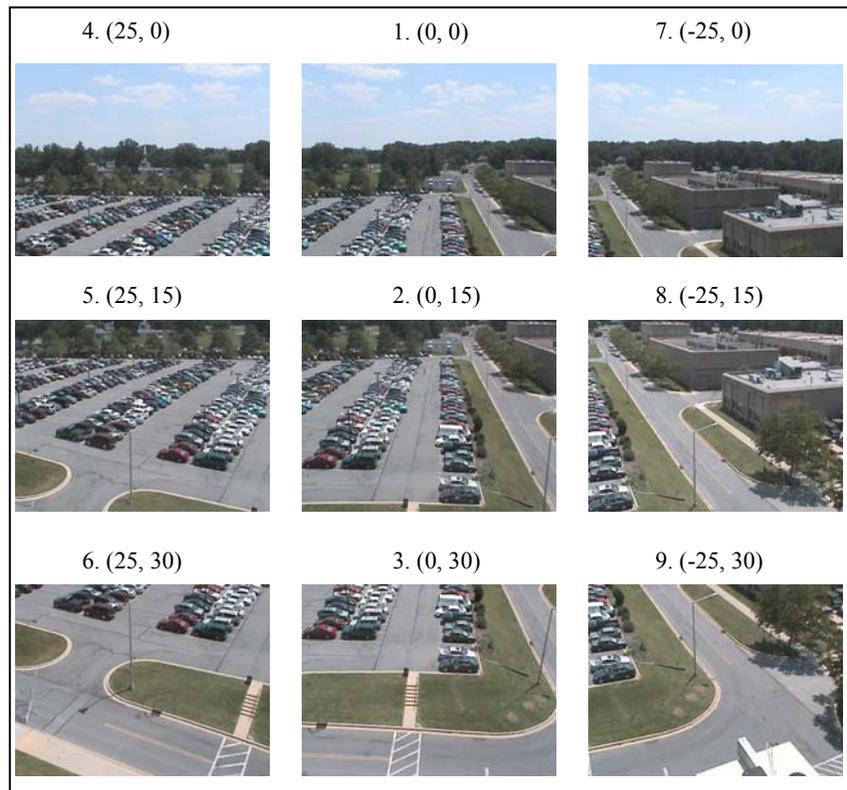


Figure 5. Images acquired in the construction of the front face. The order from which image is acquired and stitched is shown by the first number followed by the camera's pan and tilt angles  $(p, t)$ .

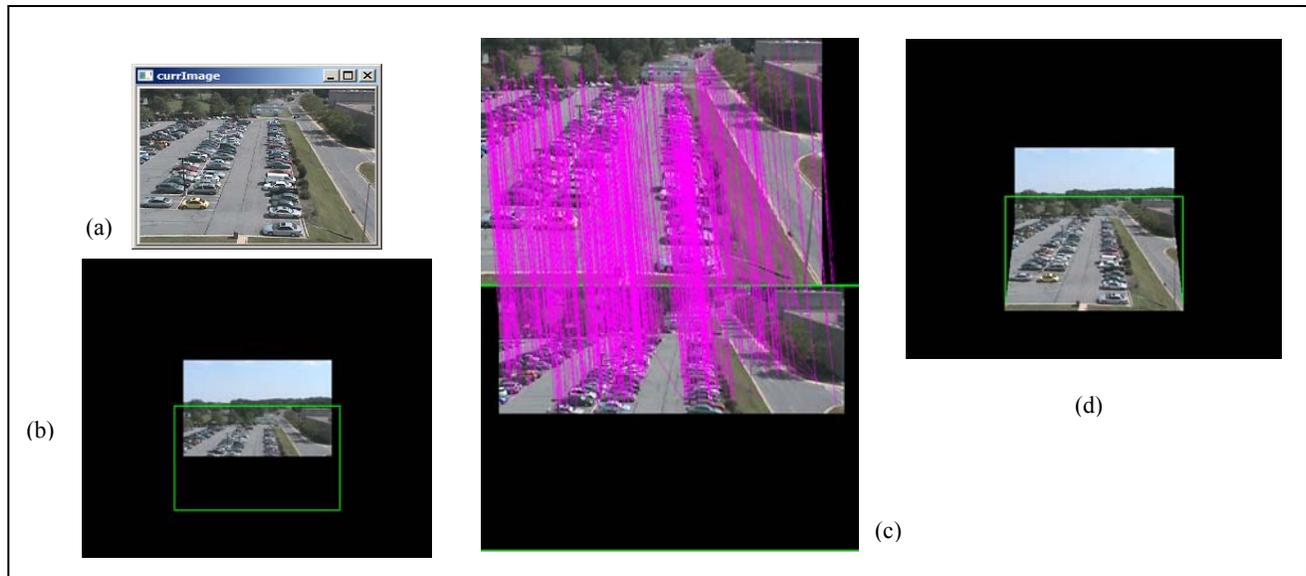


Figure 6. Mosaicking of the front face: (a) New image captured at (0, 15); (b) a submosaic is selected from the mosaic at (0,15) as indicated by the green ROI box; (c) matched features between the new image (top) and the submosaic (bottom); and the (d) mosaicked image.

### 3.3 Feature Extraction

The most time-consuming stage of the mosaicking process is where correspondences between the new and submosaic images are found. Its success depends on how good the feature tracker can accurately and distinctively determine feature points in each image. There is much literature devoted to various feature detectors. Some well-established and popular ones include the Kanade-Lucas-Tomasi (KLT) tracker (4), the Harris Corners detector (5), and the Scale-Invariant Feature Transform (SIFT) (6, 7). The algorithm has to be able to work reliably under adverse outdoor conditions; and it must be robust against illumination changes, noise in the scene, and orientation, scaling, and distortions in the images. These are important especially in our approach where a new image is matched to a mosaicked image that may contain noise and perspective distortion. We chose SIFT for the above reason. Also, SIFT features are easy to extract and highly distinctive, so they have a low probability of mismatch. The major stages to feature matching using the SIFT detector are described below:

1. *Scale-space extrema detection*: Interest points or keypoints are identified by searching over all scales and image locations. A difference-of-Gaussians (DoG) function is used to identify potential interest points, minima and maxima, that are invariant to scale and orientation.
2. *Keypoint localization*: Unstable keypoints with low contrast are rejected in this stage. This is done by fitting a detailed model at each candidate's location in order to determine its location and scale. Final keypoints are selected based on measures of their stability.

3. *Orientation assignment*: Each keypoint is assigned one or more orientations based on local image gradient directions. Image data are transformed relative to the assigned orientation, scale, and location according to each feature thus providing invariance to image rotation.
4. *Keypoint descriptor*: The local image gradients are measured at the selected scale in a region around each keypoint. These are transformed into a representation that is tolerant to local shape distortion and changes in illumination. Each keypoint is represented by a 128 element feature vector.

### 3.4 Feature Matching

For each feature found in the new image, the two closest matches in the submosaic are determined using the Euclidean distance. If the two distances are too close to each other, the matching cannot be done reliably and the feature is discarded. Otherwise, the closest match is included to the match set. The output of this stage is a set of feature matches between the new image and the submosaic image (see figure 6(c)).

### 3.5 Outliers Screening

Though most of the incorrect matching features are removed in the previous stage, mismatched features or so-called outliers may still exist. These are due to scene clutter as well as non-rigid objects (i.e., moving vegetation, people, and vehicles) in the scene. This condition is particularly prevalent in a dynamic scene such as ours, a parking lot where cars and pedestrians are constantly moving in and out. In order to obtain the best estimate of the projection model or homography, we opt to use Random Sample Consensus (RANSAC) (8) to screen out the outliers. RANSAC is a robust estimation procedure that uses a minimum set of randomly sampled correspondences to estimate image transformation parameters, and finds a solution that has the best consensus with the data. It is able to select the best set of inliers among all correspondents with a high degree of accuracy when a significant number of outliers are present. The RANSAC algorithm that is used in this stage is described below.

Steps 1 through 3 are repeated  $N$  times where  $N$  is the number of random samples required to examine in order to guarantee with a given probability that at least one of these samples contains only inliers. In other words,  $N$  is the number of samples to examine before quitting and is defined below:

$$N = \frac{\log(1 - S)}{\log(1 - (1 - \alpha)^4)} \quad (6)$$

In this equation,  $\alpha$  is the maximum fraction of correspondences that are outliers and  $S$  is the required probably of success:

1. Randomly select a set of four correspondences from all the correspondences obtained from the last stage and solve the homography matrix using those four correspondences.

2. Calculate the Euclidean distance between each of the correspondences and the calculated position using the homography  $H$  obtained from Step 1. If the distance is less than a certain threshold, it is an inlier.
3. Keep the homography matrix computed from the set of four correspondences that has the most inliers.

At the end of this stage, the best estimate for the projection model is obtained and will be used in the warping stage where the new image is transformed to a face on the cube representing its mosaic.

### 3.6 Warping

The projection model, or homography, obtained from the previous stage is now used to transform (warp) the new image so it is registered to the cube-map image. A translation matrix is then applied to the homography to compensate for the translation performed in determining the submosaic from the cube-map mosaic. Bilinear interpolation is used in registering the new image to the mosaic (see figure 6(d)).

### 3.7 Blending

The new image, after it is registered to the mosaic, must be stitched into the mosaic. For areas in the mosaic where there is no overlap, pixels values from the warped new image are directly copied into the mosaic. Discrepancy in intensity may be large in the area of overlap due to illumination changes in the scene and the effect of the camera auto iris. In order to “hide” this unpleasant stitch look, blending is applied to the mosaic. Blending is a process of finding the updated pixel values in the area of overlap by applying a blending function  $b(x)$  that outputs a weight between 0 and 1 for each pixel in the image. The updated pixel values are computed using the following equation:

$$I'(x') = b(x)I(x) + (1 - b(x))I'(x') \quad (7)$$

where  $I$  and  $I'$  are the pixel values of the warped new image and mosaic, respectively. A blending function that decreases near the boundary of an image will effectively prevent visible discontinuities from occurring. We used a 2-D Gaussian blending function. Blending not only makes the intensities of the mosaic image more uniform, it can also reduce the effect of the registration errors.

---

## 4. Results

---

We successfully constructed four sides of the mosaic cube using the methods discussed in section 3 of the report (figure 7). The top and bottom faces of the cube-map were omitted because only the sky and rooftop were in view. Each face of the mosaic cube is made up of nine  $640 \times 480$  images captured live via the SONY PTZ camera on our building's rooftop. The mosaicked parking lot scene showed good registration even when moving objects were present. This demonstrates that the method is very effective in selecting the inliers and dropping the outliers in the feature matching process.



Figure 7. All four sides of the unfolded cube.

All of our code is written in C/C++ running on a PC with an Intel Core2 Quad 3.4 GHz processor with 3 GB of memory. The main shortfall of the method is the mosaicking speed. Computation time can be significantly reduced if we limit the number of extracted features to hundreds while still obtaining a good result. It takes about 1.5 s to extract about 500 features on a  $640 \times 480$  image using SIFT. This means it takes 3 s to extract features from both the new image and the submosaic in sequence. Consequently, 24 s are spent in feature extraction in the construction of a single face. We are currently looking into a GPU-based SIFT implementation that can provide a speedup of 10 times over a CPU-only based implementation (9,10). Even greater performance can be achieved by also parallelizing the SIFT algorithm to take advantage of the current, and widely available, multi-core processors (11). Our ultimate goal is to integrate the module into a surveillance system running on a small robot platform in an urban environment.

---

## 5. Conclusion

---

We have presented an automatic method to construct a full  $360^\circ$  panorama onto cube-maps. The SIFT feature detector is used in extracting features from images. The method worked well in a dynamic scene of a parking lot with moving cars and pedestrians. Higher resolution cube-maps can be built by aligning images to a lower resolution cube-map in a coarse-to-fine manner (3),

which is not in the scope of this report. Ongoing work includes an auto-calibration module (12) that calibrates the camera on-the-fly without human intervention and a moving object tracker that can detect and track objects using the cube-maps.

---

## 6. References

---

1. Zhang, Z. Flexible camera calibration by viewing a plane from unknown orientations. *ICCV* **1999**, 666–673.
2. Green, N. Environment mapping and other applications of world projections. *IEEE Computer Graphics and Applications* **1986**, 6 (11), 21–29.
3. Sinha, S. N.; Pollefeys, M. Pan-tilt-zoom camera calibration and high-resolution mosaic generation. *Computer Vision and Image Understanding* **2006**, 103, 170–183.
4. Shi, J.; Tomasi, C. Good features to track. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* **1994**, 593–600.
5. Harris, C.; Stephens, M. A combined corner and edge detector. *4th Alvey Vision Conference* **1988**, 147–151.
6. Lowe, D. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* **2004**, 60 (2), 91–110.
7. Brown, D.; Lowe, G. Recognizing panoramas, *Proceedings of the ICCV*, Vol. 1, 1218-1225, 2003.
8. Fischler, M. A.; Bolles, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of ACM* **1981**, 24 (6), 381–395.
9. Sinha, Sudipta N.; Frahm, Jan-Michael; Pollefeys, Marc; Gene, Yakup. *GPU-based video feature tracking and matching*; TR 06-012; Department of Computer Science: UNC Chapel Hill, NC, 2006.
10. Heymann, K. Müller; Somolic, A.; Froehlich, B.; Wiegand, T. SIFT implementation and optimization for general-purpose GPU, *International Conference in Center Europe on Computer Graphics, Visualization and Computer Vision*, 2007.
11. Zhang, Q.; Chen, Y.; Zhang, Y.; Xu, Y. SIFT implementation and optimization for multi-core systems, *IEEE International Symposium on Parallel and Distributed Processing* **2008**, 1–9.
12. Heikkilä, J.; Silvén, O. A four-step camera calibration procedure with implicit image correction. *CVPR* **1997**, 1106–1112.

---

## Acronyms

---

2-D	two-dimensional
3-D	three-dimensional
DoG	difference-of-Gaussians
FOV	field of view
GPU	Graphics Processing Unit
KLT	Kanade-Lucas-Tomasi
PTZ	pan-tilt-zoom
RANSAC	Random Sample Consensus
RGB	red, green, blue
ROI	region of interest
RSTA	Reconnaissance, Surveillance, and Target Acquisition
SIFT	Scale-Invariant Feature Transform

NO. OF COPIES	ORGANIZATION	NO. OF COPIES	ORGANIZATION
1 ELEC	ADMNSTR DEFNS TECHL INFO CTR ATTN DTIC OCP 8725 JOHN J KINGMAN RD STE 0944 FT BELVOIR VA 22060-6218	1	US GOVERNMENT PRINT OFF DEPOSITORY RECEIVING SECTION ATTN MAIL STOP IDAD J TATE 732 NORTH CAPITOL ST NW WASHINGTON DC 20402
1	DARPA ATTN IXO S WELBY 3701 N FAIRFAX DR ARLINGTON VA 22203-1714	1	US ARMY RSRCH LAB ATTN AMSRD ARL CI OK TP TECHL LIB T LANDFRIED BLDG 4600 ABERDEEN PROVING GROUND MD 21005-5066
1 CD	OFC OF THE SECY OF DEFNS ATTN ODDRE (R&AT) THE PENTAGON WASHINGTON DC 20301-3080	1	DIRECTOR US ARMY RSRCH LAB ATTN AMSRD ARL RO EV W D BACH PO BOX 12211 RESEARCH TRIANGLE PARK NC 27709
1	US ARMY RSRCH DEV AND ENGRG CMND ARMAMENT RSRCH DEV AND ENGRG CTR ARMAMENT ENGRG AND TECHNLGY CTR ATTN AMSRD AAR AEF T J MATTS BLDG 305 ABERDEEN PROVING GROUND MD 21005-5001	9	US ARMY RSRCH LAB ATTN AMSRD ARL CI OK PE TECHL PUB ATTN AMSRD ARL CI OK TL TECHL LIB ATTN IMNE ALC IMS MAIL & RECORDS MGMT ATTN AMSRD ARL CI IA S HO (5 copies) ATTN AMSRD ARL CI IA P DAVID ADELPHI MD 20783-1197
1	PM TIMS, PROFILER (MMS-P) AN/TMQ-52 ATTN B GRIFFIES BUILDING 563 FT MONMOUTH NJ 07703	TOTAL: 19 (1 ELEC, 1 CD, 17 HCS)	
1	US ARMY INFO SYS ENGRG CMND ATTN AMSEL IE TD F JENIA FT HUACHUCA AZ 85613-5300		
1	COMMANDER US ARMY RDECOM ATTN AMSRD AMR W C MCCORKLE 5400 FOWLER RD REDSTONE ARSENAL AL 35898-5000		