Real-time panorama image synthesis by fast camera pose estimation

Beom Su Kim, Sang Hwa Lee, and Nam Ik Cho
INMC, School of Electrical Engineering and Computer Science, Seoul National University
E-mail: bskim@ispl.snu.ac.kr, lsh529@snu.ac.kr, nicho@snu.ac.kr

***Abstract*—This paper proposes a fast panorama synthesis algorithm that runs on a mobile devices real-time. Like most existing methods, the proposed method consists of following steps: feature tracking, rotation matrix estimation, and image warping on a targeting plane, where the feature tracking is usually a bottleneck for real-time implementation. Hence, we propose to track the features on a virtual sphere surface instead of projected surface or image domain as in the conventional methods. By performing the feature tracking on the sphere, the camera pose can be found by linear and non-iterative least squares method, which was usually obtained by nonlinear and iterative methods. The fast estimation of camera pose can make outlier rejection more robust since the camera pose can be inferred from the hypotheses by one iteration, which can't be done in real-time by iterative estimation. We also propose a two-step blending algorithm, i.e., celling-filling followed by linear blending along the cell boundary. The panorama canvas is partitioned into many cells where each cell contains pixels from the same shot. Hence there is no stitching seam within the cell and only the boundaries need to be blended, which reduces the stitching artifacts significantly.**

***Index Terms*—Panorama image, feature tracking, real-time AR, mobile system**

I. Introduction

Panorama image synthesis is to generate a large or wide image from a sequence of shot. With the advancement of digital camera technologies and embedded systems, it has become possible to implement the panorama synthesis algorithm on the camera or mobile devices [1]. However, the elaborate panorama synthesis such as [2] cannot be implemented on the embedded system due to their high computational complexity, which degrades the quality of synthesized image on a mobile device. The main bottleneck for the real-time implementation of panorama seems to be feature detection and tracking between the frames, and thus there have been many algorithms to reduce the computations for these steps [3] [4] [5]. Specifically, in order to avoid heavy computations required for feature detection, the early mobile panorama systems assumed the fixed camera motions such as horizontal rotations with fixed angles using user-constrained interfaces [3]. This simplified the calculations of transformation matrix with high accuracy, but the degree of freedom to handle the panoramic images was restricted. Adams proposed a feature tracking method which simplified the descriptor-based matching to reduce the time for the estimation of camera's translational motion [4]. Wagner developed the feature tracking to estimate 3-DOF (degree of freedom) rotation matrix in the mobile systems, which tracks the feature points in the hierarchical multi-resolutions to reduce time consumption, and iteratively updates the rotation matrix using Gauss-Newton method [5]. The panorama system in [5] also generated the panoramic images in real-time, but since it needs iterative computation of feature matching and camera pose estimation, it is difficult to use robust outlier rejection scheme. Moreover, this system warps every input frames with small misalignment caused by matching error and translation of camera which was not assumed in the 3-DOF model, so the artifact such as discontinuous seam may arise in the panorama image. This paper proposes a real-time panorama algorithm using feature matching in the mobile systems. Especially, we focus on the fast estimation of rotation matrices which is the main process for real-time and automatic panorama synthesis. We change the non-linear and iterative problem of rotation matrix estimation into linear and non-iterative problem without loss of performance. This linear and non-iterative process improves the operation speed and eliminates the problem of floating point precision in the fixed integer coding for mobile devices. In addition, we improve quality of panoramic images by adopting robust outlier rejection scheme and by using fast two-step blending. We demonstrate the proposed panorama algorithms in the usual mobile systems such as mobile phones and tablet PC.

The rest of paper is organized as follows. Section 2 describes the proposed panorama system in detail. Experimental results are shown in Section 3. and we conclude the paper in Section 4.

II. Proposed Panorama System

The proposed panorama algorithms consist of feature extraction, feature tracking in the multi-resolutions, rotation matrix estimation, warping, and display interface.

A. Feature Tracking

To estimate the camera motions automatically without user interaction, a robust feature matching scheme is required. However, the feature detection has been a bottleneck for the real-time operation due to heavy computation. Reference [5] proposed a method to reduce the time required for feature detection by tracking the previously detected features in the next image. We also apply the tracking method of [5] for feature matching. Fig. 1 shows the panorama canvas and illustrates feature tracking. The panorama canvas is partitioned into 64×64 pixel blocks, which will be called cells. When the

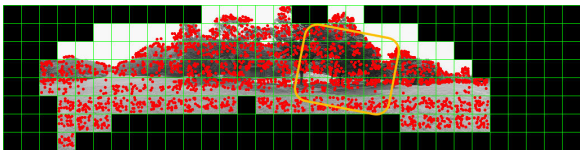


Fig. 1. Panorama canvas and feature tracking. The cell (green block) is a basic unit to extract features (red points). The input frame (yellow rectangle) is located at the predicted position from extended kalman filter.

cell is completely filled with warped images, feature points are extracted in the cell. We implement the feature extraction in the three multi-resolutions for saving the computation. In large and middle resolution, FAST features in [6] is extracted, which is one of the fastest feature detection algorithms. In small resolution, we use fast hessian detector in [7] since the number of reliable features are small in lowest resolution due to noise and down-sampling. The fast hessian detector has the best repeatability even though the number of extracted feature is small [8]. The features to track are picked out from previously extracted features within overlapped cells between the panorama canvas and input image. The overlapped region is initially guessed from previous camera pose, for current camera pose is not known without tracking results. The guessed camera pose has to be close to true value since the search range for feature tracking is limited to small window near guessed position due to limited computation power. To predict initial pose of camera accurately, each parameter related to camera pose is tracked using extended Kalman filter [9]. The selected features in the lowest resolution are first tracked by the block-based matching within the search range, and the tracking result in the lower resolution is refined in the higher resolutions.

B. Estimation of Rotation Matrix

Using the correspondences from features tracking, we estimate the camera pose of the input frame. We model the camera motions as 3-D rotation along a fixed camera center. The 3-D rotation matrix has 3 parameters of angles $\Theta = (\theta_x, \theta_y, \theta_z)$ according to the coordinate axes.

Previous algorithms update the rotation parameters of current input using the previous ones, $\Theta_t = \Theta_{t-1} + \Delta\Theta$, where the incremental parameter vector $\Delta\Theta$ is estimated to update the rotation matrices. Finding the parameters is not linear, so M-estimator was used in the iterative process [5]. In addition, the increment of parameters is too small between the input frames to express by the fixed point coding. This causes the performance degradation in the mobile devices without FPU. We propose a method to transform the non-linear estimation problem into linear and non-iterative process. The correspondences are computed not between input images but between the cylindrical surface and an input image by tracking, and this is the main reason that makes the parameter estimation a non-linear problem. Thus, we first project the feature points on the panorama canvas and input frames to a unit sphere in the 3-D world coordinates. This "projection onto unit sphere" changes the non-linear warping process into a linear transform. For convenience, let us define some notations as: $W(P|O) \rightarrow M$ is the cylindrical warping by rotation matrix O, from a feature coordinate P in the input image onto a point M on the cylindrical surface. The world coordinates of feature point P is described as

$$P_w = (X, Y, Z) = O^{-1}K^{-1}\pi'(P), \tag{1}$$

where π' is a function to map the 2-D coordinates into the homogenous coordinates by adding 1 in the z-coordinate, and K is the camera calibration matrix. Since the panorama canvas is a cylindrical surface, 3-D coordinates (X, Y, Z) are mapped onto cylindrical coordinates (u, v) as,

$$M = (u, v) = \left(R\tan^{-1}\left(\frac{X}{Z}\right),\ R\frac{Y}{\sqrt{X^2+Z^2}} \right), \tag{2}$$

where R is the radius of cylinder as the projection surface.

Now, we get the coordinates of features that are projected on the unit sphere,

$$M_s = \frac{1}{\sqrt{R^2+v^2}} \begin{pmatrix} R\sin(u/R) \\ v \\ R\cos(u/R) \end{pmatrix}, \tag{3}$$

and

$$P_s = \frac{1}{|K^{-1}\pi'(P)|}K^{-1}\pi'(P), \tag{4}$$

where M_s is the coordinates of features on the panorama canvas projected on the unit sphere, and P_s is the coordinates of features on the input frame projected on the unit sphere. After projecting feature points on the panorama canvas and those on the input frame onto the unit sphere, two corresponding points on the unit sphere are linearly related by rotation matrix O of current camera pose,

$$P_s = OM_s. \tag{5}$$

Using the relation between eq. (3) and eq. (5), we estimate rotation parameters in the linear process. Consequently, we derive the formulation using singular value decomposition (SVD) to get the parameters from correlation matrix T of corresponding feature points on the unit sphere,

$$T = \sum_{i=1}^{N} P_{s_i} M_{s_i}^T = U\Sigma V^T, \tag{6}$$

and the rotation matrix is derived as

$$O = U \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & s \end{bmatrix} V^T, \tag{7}$$

where s is a sign values [10], [11],

$$s = \text{sign}\left(det(UV^T)\right). \tag{8}$$

In (8), there is no possibility that s is -1 since the correspondences are matched within a small search range. Thus, s is always 1 in (8). The process from (6) to (8) is a well-known

(a) (b)

Fig. 2. Comparison of blending results, (a) previous method in [5] (b) proposed method

linear algebra problem that needs little computational load for 3×3 matrix.

C. Removal of Outliers

When we match the corresponding feature points while tracking, there are some outliers. They cause the incorrect estimation of camera pose, which distorts the panorama result and tracking performance. For the removal of outliers, we use iterative random sample consensus (RANSAC) method [12]. In the case of M-estimator, many iterative operations are required due to Gauss-Newton method to evaluate each hypothesis, so it is difficult to use iterative RANSAC. In contrast, the proposed method performs the non-iterative linear operation when evaluating the generated hypotheses for RANSAC. Thus, the iterative RANSAC does not delay the panorama process in the proposed algorithm. We need 2 samples to generate a hypothesis for the rotational model and about 10 iterations to remove the outliers sufficiently [12]. For speeding up the evaluation process for each hypothesis, we adopt early rejection scheme which is to reject hypotheses that have large difference from the previous camera pose in the early stage. When removing the outliers in the iterative RANSAC, we evaluate the reliability of refined rotation matrix by checking the ratio of inliers. The number of inliers usually changes very much according to the scene complexity, and it is not suitable for testing the correctness of rotation matrix. Hence, we exploit the ratio of inliers over all initially matched features. When the ratio of inliers is high, the computed rotation matrix is reliable. However, when the ratio of outliers is high, we discard the rotation parameters and track the feature points again until the reliable rotation matrix is obtained.

D. Warping with two-step blending

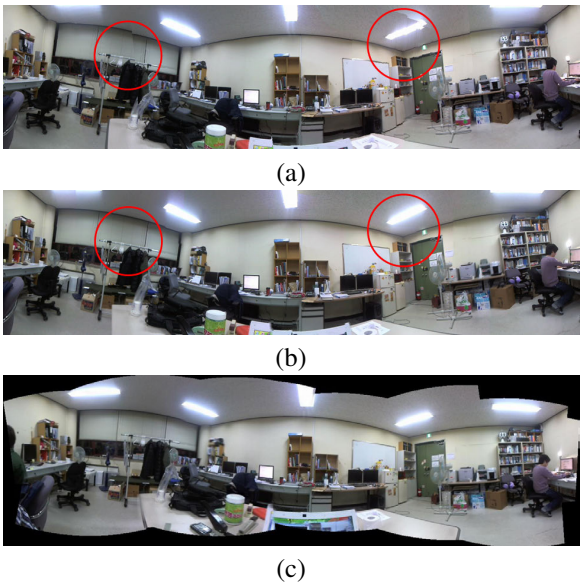
Input images are warped into the panoramic canvas using the refined camera pose from initial value. Basically, we ignore previously filled pixels in the panorama result and fill only unfilled pixels with warped image of the input image. Filling only unfilled pixels can reduce computational time, but the artifact such as discontinuous seam may arise like Fig. 2(a). Since this artifact is caused mainly by an accumulation of small error in estimation of camera pose, we skip the warping of input frames whose projection regions are considerably filled with previous input frames. This makes covered area in panorama canvas by an input image larger, so that number of discontinuous seams can be reduced. Though such sampling of input frames can reduce the artifact caused by small alignment error between consecutive frames, it cannot affect alignment error between sampled frames. Moreover, sampled frames are more vulnerable to alignment error than the consecutive frames due to their sparsity. To compensate for the alignment error between sampled frames, we use blending along stitched boundaries between sampled frames. Considering the computation time, we use linear blending in a narrow window. Since linear blending with small window is insufficient to reduce the alignment error over a large overlap region, we fill each cell of panoramic image again with one input image which can cover the whole cell region. This makes whole pixels in one cell to be brought from an input image, which avoids artifact at least within the filled cell. In case that neighboring cells are brought from different input images, discontinuous seams may arise along the boundaries of cells. The boundaries of cells are also blended linearly. These two step blending reduces artifact in panoramic canvas significantly while satisfying real-time performance.

TABLE I
TIME CONSUMPTION OF PARAMETER ESTIMATION.

Resolution	Iterative M-estimator	Proposed Method
Small (512×128)	0.4 ∼ 0.7ms	0.007 ∼ 0.009ms
Medium (1024×256)	0.8 ∼ 1.0ms	0.01 ∼ 0.02ms
Large (2048×512)	1.0 ∼ 1.5ms	0.03 ∼ 0.05ms

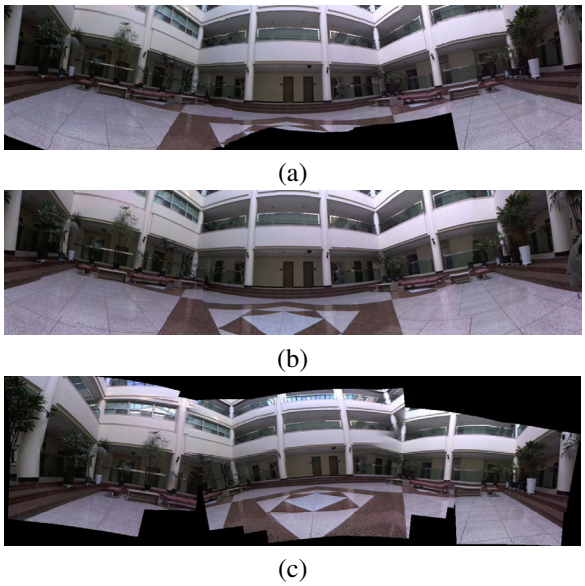
III. EXPERIMENTAL RESULTS

We have implemented the proposed panorama algorithms in the mobile phone with 1GHz CPU. The input video consists of 320×240 frames, and panorama canvas is 2048×512. In the mobile device, the proposed system works over 30 fps and average timing per frame is about 22ms∼30ms. To compare the performance of proposed system, we implemented the method in [5] using M-estimator. Table 1 show the comparison of time consumption in estimating the rotation parameters in Intel 2.6GHz PC. Time consumption is dependent on the number of tracked features. For fair comparison, the number of features to track in each resolution is limited to the same. In Table 1, it can be seen that the proposed non-iterative method is much faster than the iterative M-estimator. In Fig. 3 and Fig. 4, the results from our own implementation of [5] and the result of *Autostitch*[13] which is well-known offline stitching program are compared with the results of proposed method. It can be seen that the proposed results suppress the artifact better than previous method especially inside red circles in Fig. 2 and they are comparable with results of offline stitching algorithm. Moreover, since proposed method removes outliers in tracking process by RANSAC, it shows better tracking performance than the previous method. In Fig. 3(a), previous method in [5] fails to track at the bottom of panorama canvas due to lack of correct matching, but the proposed method tracks camera



(a) (b) (c)

Fig. 3. Comparison of panorama results, (a) previous method in [5] (b) proposed method (c) *Autostitch*[13].



(a) (b) (c)

Fig. 4. Comparison of panorama results, (a) previous method in [5] (b) proposed method (c) *Autostitch*[13].

pose robustly.

IV. CONCLUSION

We have proposed a real-time panorama algorithm for the mobile devices. The proposed panorama system consists of feature extraction, feature tracking, estimation of camera pose, and image warping with blending. Features are extracted from the three multi-resolutions of panoramic image. Then, the detected feature points are tracked on the input images in the three hierarchical resolutions. The camera pose is modeled as a rotation matrix which is estimated using the tracked feature points. For real-time operation of panoramic image synthesis, we have proposed a method to estimate the rotation matrix using non-iterative least squares which is much faster than the previous M-estimator. Fast estimation of camera pose also enables the system to remove outliers in feature tracking by RANSAC, so tracking performance can be improved with little computational load. Finally, we project the input frames onto panorama surface with fast two-step blending. Experimental results shows that the proposed system produces panoramic images with unnoticeable distortion, while satisfying real-time operation on mobile devices.

ACKNOWLEDGEMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (2012-0000913).

REFERENCES

[1] R. Szeliski, "Image alignment and stitching: a tutorial," *Found. Trends. Comput. Graph. Vis.*, vol. 2, no. 1, pp. 1–104, 2006.

[2] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *International Journal of Computer Vision*, vol. 74, pp. 59–73, August 2007.

[3] S. Ha, S. Lee, N. Cho, S. Kim, and B. Son, "Embedded panoramic mosaic system using auto-shot interface," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 1, pp. 16 –24, Febuary 2008.

[4] A. Adams, N. Gelf, and K. Pulli, "Viewfinder alignment," *Computer Graphics Forum (Proceedings of Eurographics)*, vol. 27, pp. 597–606, 2008.

[5] D. Wagner, A. Mulloni, T. Langlotz, and D. Schmalstieg, "Real-time panoramic mapping and tracking on mobile phones," in *Virtual Reality Conference*, March 2010, pp. 211 –218.

[6] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European Conference on Computer Vision*, 2006, pp. 430–443.

[7] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," *Computer Vision and Image Understanding*, vol. 110, pp. 346–359, 2008.

[8] S. Gauglitz, T. Hollerer, and M. Turk, "Evaluation of interest point detectors and feature descriptors for visual tracking," *International Journal of Computer Vision*, vol. 94, pp. 335–360, 2011, 10.1007/s11263-011-0431-5.

[9] M. I. Ribeiro, "Kalman and extended kalman filters : Concept, derivation and properties," *Institute for Systems and Robotics Lisboa Portugal*, Fautostitcheb. 2004.

[10] M. Brown, R.I. Hartley, and D. Nister, "Minimal solutions for panoramic stitching," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2007.

[11] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 5, pp. 698 –700, September 1987.

[12] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, second edition, 2004.

[13] Autostitch, *http://www.autostitch.net.*