Automatic Determination of Constraint Parameter for Improving Homography Matrix Calculation in RANSAC Algorithm

Devy Chandra*, Kee-Sung Lee*, Geun-Sik Jo**
*Department of Information Engineering, Inha University
**School of Computer & Information Engineering, Inha University
e-mail: wiy_ch4n@hotmail.com, lks@eslab.inha.ac.kr, gsjo@inha.ac.kr

Abstract

This paper proposes dynamic constraint parameter to filter out degenerate configurations (i.e. set of collinear or adjacent features) in RANSAC algorithm. We define five different groups of image based on the feature distribution pattern. We apply the same linear and distance constraints for every image, but we use different constraint parameter for every group, which will affect the filtering result. An evaluation is done by comparing the proposed dynamic CS-RANSAC algorithm with the classic RANSAC and regular CS-RANSAC algorithms in the calculation of a homography matrix. The experimental results show that dynamic CS-RANSAC algorithm provides the lowest error rate compared to the other two algorithms.

1. Introduction

Augmented reality (AR) is a novel technology that allows computer-generated virtual objects to be seamlessly superimposed upon image sequences of real world scene [7]. Azuma identifies three key characteristics of AR: (1) combining real and virtual images, (2) the virtual images are registered with the real world, and (3) interactive in real time [11]. In AR system, accurate registration is required to ensure the integration of virtual objects into image sequences [13].

Homography matrix is extensively used in computer vision system to calculate the relation between two images related by a geometric transformation T, such that under perspective projection, T between a pair of correspondence points $p$and p' is projective linear, or a homography [12].

$$p' = Hp \text{ and } H^{-1}p' = p$$

where H is the homography matrix relating two views of planar target, which describes a point-to-point imaging, represented by a 3×3 matrix with 8 degrees of freedom.

$$H = \begin{bmatrix} H_{00} & H_{01} & H_{02} \\ H_{10} & H_{11} & H_{12} \\ H_{20} & H_{21} & 1 \end{bmatrix}$$

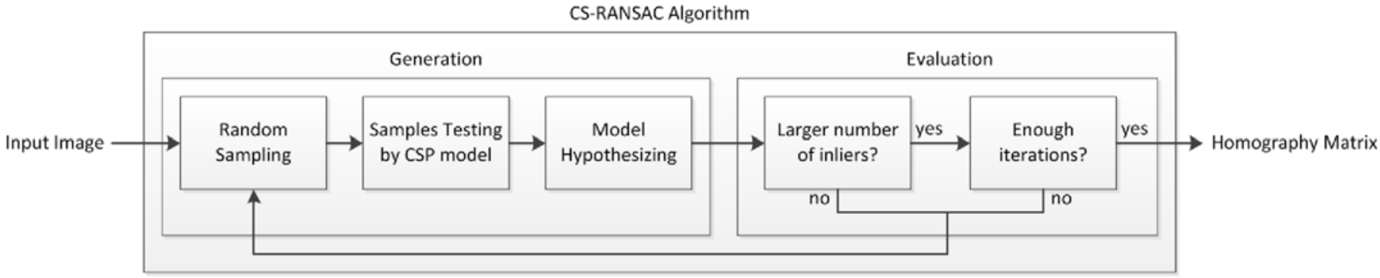
Random Sample Consensus (RANSAC) algorithm is often employed to detect planar homographies in uncalibrated image pairs [8, 9]. It is an iterative algorithm consisting of two main steps: generation and evaluation. In generation step, a minimum subset of data is randomly selected to compute a model to fit the whole dataset. In evaluation step, the computed model is used to determine the consensus set (i.e. inlier set) for model evaluation. These two steps are iteratively repeated until convergence.

The selection of random samples in RANSAC algorithm highly impacts the accuracy of the homography matrix generated in each iteration. In addition, it picks the random samples without considering their locations. This may lead to the selection of samples distributed linearly or too close to one another.

CS-RANSAC algorithm introduces CSP into a computer vision problem of estimating a homography matrix [2, 3]. The sampling problem of a RANSAC algorithm is represented as a CSP model to filter out degenerate configurations (i.e. set of collinear or adjacent features). Thus, only non-degenerate samples are used to calculate the homography matrix of an image pair. An experiment is done by computing homography matrices for several pairs of image, which are divided into five different groups depending on the feature distribution pattern. The experimental results show that applying same parameter to different groups yields different results in error rate and processing time. Some are very good, while some others are just fair. These results conclude that different feature distribution needs different parameter to obtain better results. Therefore, this paper proposes a dynamic constraint parameter for CS-RANSAC algorithm, where it will automatically select a correct parameter based on the feature distribution pattern of the input image.

2. Related Works

Many researches have been conducted to improve RANSAC algorithm for homography matrix calculation. Some studies focus on finding the true set of inliers, while some other focus on the sampling step by filtering degenerate samples according to some conditions. LO-RS [10] improve the inlier rate by locally optimizing a specific area of the image that is highly condensed with features. It runs an inner RANSAC algorithm within each iteration of an outer RANSAC algorithm. T-RS [5] computes the areas of four triangles determined by the random points and retains only those whose areas are greater than a given threshold. MFF-RANSAC [6] applies two filters on sampling step: angle and length filters. The feature is retained as an inlier only if its value below the median flow value. These techniques have successfully improved



(Figure 1) CS-RANSAC algorithm

RANSAC algorithm for calculating a homography matrix of an image pair. However, the resulting homography matrix might not be able to accurately estimate the pose of objects in the whole images, because they focus only to estimate a specific object in a specific area of the image.

As described in Section 1, CS-RANSAC algorithm considers the sampling problem of RANSAC algorithm as CSP, where the variable is defined as a set of feature points denoted as f_k, $k \in \{1,2,\dots,n\}$. The input image I_s is converted into an $N \times N$ grid, such that each cell can be represented by a two-tuple (row, col) defining the location of the corresponding cell. Using this two-tuple as the value, the domain of each feature f_k is defined as a set of any possible location of the corresponding cell (i.e. $\{(1,1),(1,2),\dots,(N,N)\}$). The relationship between all the selected random features is defined using a set of constraints: linear and distance constraints. Linear constraints ensure that no selected samples lie on the same linear sequence, whereas distance constraints ensure that all samples lie far enough from one another. With row_i, row_j, col_i, and col_j are the row and column indices of the corresponding cells of features f_i and f_j, respectively, the linear and distance constraints are defined as follows.

(Definition 1) Linear constraints

A set of n features satisfies linear constraints iff:

$i,j \in \{0,\dots,n\} \Rightarrow$

$<row_i \neq row_j>$ *is true and*

$<col_i \neq col_j>$ *is true and*

$<col_j \neq col_i \pm |row_i - row_j|>$ *is true*

(Definition 2) Distance constraints

A set of n features satisfies distance constraints iff:

$i,j \in \{0,\dots,n\} \Rightarrow$

$<[|row_i - row_j| = 1]\ AND\ [|col_i - col_j| > 2]>$ *is true*

and

$<[|Row_i - Row_j| = 2]\ AND\ [|Col_i - Col_j| > 1]>$ *is true*

Figure 1 briefly shows the procedure of CS-RANSAC algorithm. A set of n random features is tested by the constraints set defined in Definition 1 and 2. The feature set is considered for the next step only if all the constraints are satisfied. If there is any pair of the feature set fails to satisfy any constraint, CS-RANSAC algorithm will first check the sampling iteration denoted as $sIterations$. If $sIterations$ exceeds a predefined threshold θ_s, then CS-RANSAC will instead sample another set of n features with the highest similarity ranking based on the Euclidean distance. The rest of the CS-RANSAC algorithm is same to that of classic RANSAC algorithm, in which the homography matrix is iteratively calculated and updated until convergence.

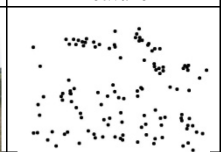
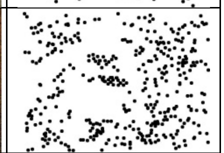
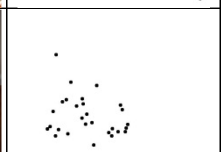
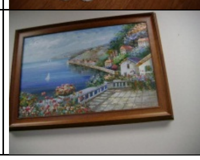
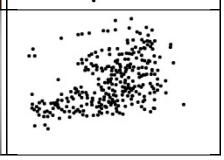
3. Dynamic CS-RANSAC Algorithm

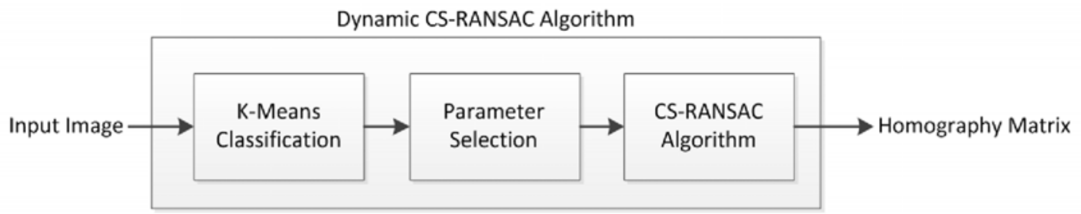
Figure 2 shows the procedure of dynamic CS-RANSAC algorithm. We divide the algorithm into three main parts: (1) image classification, (2) parameter selection, and (3) CS-RANSAC algorithm.

Image Classification

We employ k-means algorithm for image classification [1], such that according to the feature distribution pattern, each input image is classified into a specific group. We define five image classes as listed in Table 1.

<Table 1> Image classification based on feature distribution

Group	Definition	Image	Feature
G1	Few features distributed over the whole image		
G2	Many features distributed over the whole image		
G3	Many features distributed over the center of the image		
G4	Few features distributed over a specific area of the image		
G5	Many features distributed over a specific area of the image		



(Figure 2) Dynamic CS-RANSAC Algorithm

Parameter Selection

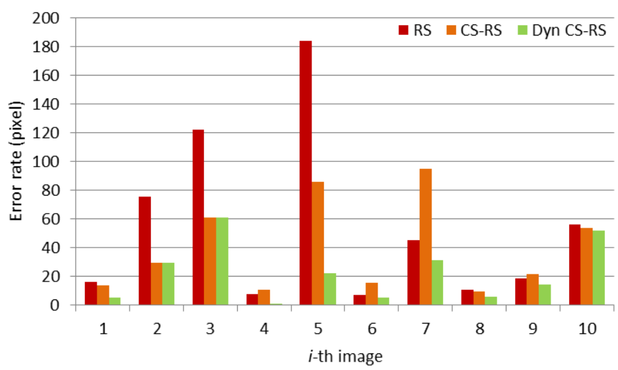
We apply the same linear and distance constraints to every image, but we employ different size of grid depends on which class the input image belongs to.

CS-RANSAC algorithm

Finally, to calculate the homography between the input image and its reference image, we employ the regular CS-RANSAC algorithm as describe in Section 2.

4. Experiments

For evaluation, we compare the proposed dynamic CS-RANSAC algorithm with the classic RANSAC and the regular CS-RANSAC algorithms. We use 10 images from UKBench dataset [4]. Each image has size of 640 × 480. We execute the algorithms to calculate a homography matrix for each image pair and measure their error rate by Euclidean distance. The results are shown in Figure 3, where we can see clearly that dynamic CS-RANSAC algorithm provides the best accuracy among other two algorithms.



(Figure 3) Error rate comparison of RANSAC, CS-RANSAC, and dynamic CS-RANSAC algorithms

5. Conclusions

We propose a dynamic environment for CS-RANSAC algorithm to automatically select the correct parameter for any input image. We define different parameter for each image class. The experimental results show that using different parameter for different classes provides better results in accuracy.

References

[1] A. Coates, A. Y. Ng, "Learning Feature Representations with K-means," in Neural Network: Tricks of the Trade Second Edition, 2012, pp. 561-580.
[2] C-H Jang, K-S Lee, G-S Jo, "CSP driven RANSAC Algorithm for Improving the Accuracy of Homography," in Proceedings of Korean Institute of Information Science and Engineers, 2012, pp. 318-320.
[3] C-H Jang, K-S Lee, G-S Jo, "CSP Driven RANSAC Algorithm for Improving Accuracy of Planar Homography," Journal of Korean Institute of Information Science and Engineers, vol. 39, no. 11, pp. 876-888, 2012.
[4] D. Nister, H. Stewenius, "Scalable Recognition with A Vocabulary Tree," in IEEE Conference on CVPR 2, 2006, pp. 2161-2168.
[5] E. Vincent, R. Laganiere, "Detecting Planar Homographies in an Image Pair," in IEEE International Symposium on Image and Signal Processing and Analysis, 2001, pp. 182-187.
[6] F. Wu, X. Fang, "An Improved RANSAC Homography Algorithm for Feature Based Image Mosaic," in WSEAS International Conference on Signal Processing, Computational Geometry & Artificial Vision, 2007, pp. 202-207.
[7] I. E. Sutherland, "The Ultimate Display," in Proceedings of the Congress of the International Federation for Information Processing, 1965, pp. 506-508.
[8] M. Zuliani, "RANSAC for Dummies," 2012.
[9] M. A. Fischler, R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," in Communications of the ACM, vol. 24, no. 6, pp. 381-395, 1981.
[10] O. Chum, J. Matas, J. Kittler, "Locally Optimized RANSAC", in DAGM-Symposium, 2003, pp. 236-243.
[11] R. T. Azuma, "A Survey of Augmented Reality," in Presence: Teleoperators and Virtual Environments, vol. 6, no. 4, pp. 355-385, 1997.
[12] S. Walt, "Homography Estimation," 2010.
[13] X. Li, Y. Liu, Y. Wang, D. Yan, "Computing Homography with RANSAC Algorithm: A Novel Method of Registration," in Proceedings of SPIE 5637, 2005, pp. 109-112. 2005

<Table 2> Test images uses in the experiment


Image 1: G3


Image 2: G1

Image 3: G1


Image 4: G2


Image 5: G4


Image 6: G5


Image 7: G5

Image 8: G3


Image 9: G2

Image 10: G3