

2-Stage Camera Motion Parameter Estimation Technique based on Geometric Invariant Features

JeongHee Cha NaYoung Lee GyeYoung Kim HyungIl Choi
School of Computing, School of Media,
Soongsil University
Sangdo 5 Dong, DongJak Gu, Seoul, Korea
pelly@vision.ssu.ac.kr nayoung4493@ssu.ac.kr {gykim11,hic}@ssu.ac.kr

Abstract - Precise camera calibration is a core requirement of location system and augmented reality. In this paper, we propose a method to estimate camera motion parameter based on invariant point features. Typically, feature information of image has drawback, it is variable to camera viewpoint, and therefore information quantity increases after time. The LM(Levenberg-Marquardt) method using nonlinear minimum square evaluation also has a weak point, which has different iteration number for approaching the minimal point according to the initial values and convergence time increases if the process run into a local minimum. In order to complement these shortfalls, we, first propose constructing invariant features of geometry using similarity function and Graham search method. Secondly, we propose a two-stage camera parameter calculation method to improve accuracy and convergence by using 2D homography and LM method. In the experiment, we compare and analyze the proposed method with existing method to demonstrate the superiority of the proposed algorithms.

Keywords: Camera Calibration, Levenberg-Marquardt Method, 2D Homography, Invariants Vectors

1 Introduction

Ubiquitous computing involves every computer being connected, invisible to the user's eyes, always available regardless of time and place, incorporated into our everyday lives and offers an autonomous support for people. Especially, the ubiquitous location-based service (u-LBS), that provides location information of objects such as people and things leads the van of these services, localization technique is an important elementary technology to implement such services. Methods for acquiring location information include using sensors, using vision technology and combining the two approaches. Sensors such as the odometer and the inertial navigation system (INS) yield accurate data only when the inherent error of sensor itself is

resolved. If we use the vision technology that recognizes the location with image information, we can get more accurate data than the sensor. However, extracting visual characteristic data for recognition is not an easy task. A typical example of vision-based localization is employing artificial marks to recognize the location. Such application has disadvantages of generating mismatches during camera operation because image feature information is varying according to the camera viewpoint, so must set numerous assumptions and restrictions [1]. The nonlinear minimization method being used to evaluate the camera's extrinsic factors calculates an optimized solution by minimizing the error. However, the number of iterations for approaching the minimal point varies according to the initial values and the time required for convergence increases if the process runs into a local minimum point [2]. In order to complement these shortfalls, we first propose constructing improved feature models using mathematical tools from projective geometry for identification. Invariant vector in this paper is a characteristic value of corner point unrelated to the camera viewpoint. Secondly, we propose a two-stage calculation method to improve accuracy and convergence by using the information acquired by homography as initial values of the LM method. A block diagram of proposed framework is shown in Fig. 1.

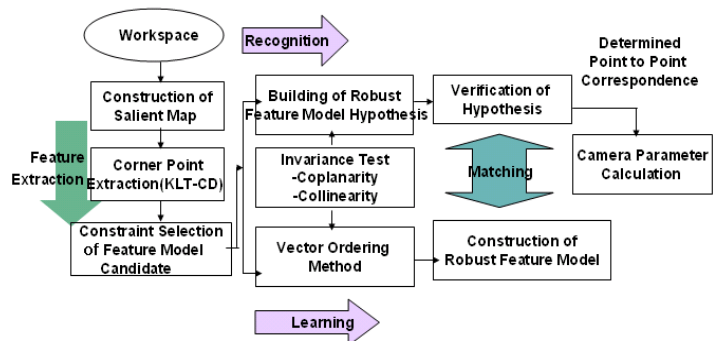


Fig. 1. Block Diagram of Proposed Configuration

2 Methodology

2.1 Cross Ratio, PPIV, Feature Extraction

In order to extract features unrelated to the viewpoint, this paper uses cross ratio of five points on a single plane defined in projective geometry. According to the theorem, when there are five points in a homogeneous coordinate system of a two-dimensional space, if the points exist on a single plane, and three of the five points are not on a same line, cross ratios of two independent projective invariant values exist as shown in Eq.(1) [3].

$$\lambda_1 = \frac{\det(m_{431})\det(m_{521})}{\det(m_{421})\det(m_{531})}, \lambda_2 = \frac{\det(m_{432})\det(m_{512})}{\det(m_{412})\det(m_{532})} \quad (1)$$

$\det(m_{431})$ is the matrix determinant calculated with coordinate data (x, y) of the corner points c_1, c_3, c_4 . Since cross ratio is permutation-sensitive like most projective invariants, we employ the permutation invariant J vector as descriptor since its value is bounded between 2.0 and 2.8, therefore, does not suffer from instabilities due to singularities, and it can provide direct point to point correspondences across matched quintuples. Eq.(3), which indicates each element of the five 2-dimensional invariant J vectors are calculated with Eq.(2), and it is called *PPIV* (Projective and Permutation Invariant Vector) in geometry [4]. Each of the five points is in a 1:1 mapping relationship with each invariant vector $J^{(i)}$.

$$J[\lambda] = \frac{2\lambda^6 - 6\lambda^5 + 9\lambda^4 - 8\lambda^3 + 9\lambda^2 - 6\lambda + 2}{\lambda^6 - 3\lambda^5 + 3\lambda^4 - \lambda^3 + 3\lambda^2 - 3\lambda + 1} \quad (2)$$

$$J^{(1)} = J[\lambda_1], J^{(2)} = J[\lambda_2], J^{(3)} = J\left[\frac{\lambda_1}{\lambda_2}\right], J^{(4)} = J\left[\frac{\lambda_2 - 1}{\lambda_1 - 1}\right]$$

$$J^{(5)} = J\left[\frac{\lambda_1(\lambda_2 - 1)}{\lambda_2(\lambda_1 - 1)}\right] \quad (3)$$

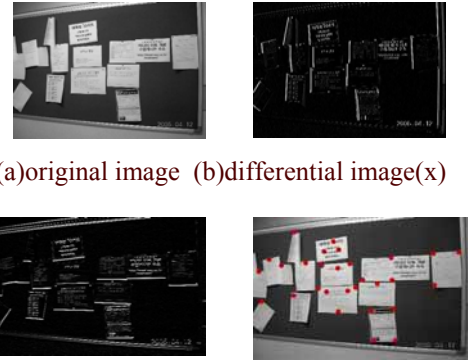
In order to reduce the amount of calculation and create robust feature models, salient map of the image can be used, and the corner points are extracted using the KLT-CD algorithm [5] within the salient region. The salient map is based on the theory that the distinctive area of the image has a higher value than other smooth areas [6]. The KLT-CD algorithm differentiates the image within the region in x and y direction as shown in Eq.(4), multiplies the transposed matrix and adds all the matrix determinants in the region.

$$g = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \nabla I, \quad gg^T = \begin{bmatrix} g_x \\ g_y \end{bmatrix} \begin{bmatrix} g_x & g_y \end{bmatrix} = \begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{bmatrix},$$

$$Z = \iint_w \begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{bmatrix} w dx \quad (4)$$

$x = (x, y)$, w : weighting function

Since matrix Z contains the pure texture information, analyzing the eigenvalue of Z allows categorization of characteristics. If two of the unique values are large, it signifies that there are corner points to be extracted. The permutation combination is a set of numbers of cases arranged in order by drawing five points from the corner points within the region without duplication. *PPIV* for each permutation combination is calculated using Eq.(1), (2) and (3). Fig.2 displays the image of the process of extracting corner points using the KLT-CD algorithm



(a)original image (b)differential image(x)

(c)differential image(y) (d)result image

Fig. 2. Process of Corner Points Extraction

2.2 Learning and Recognition

If the *PPIV* were to be preserved under projective transformation, the first condition is that there must not be three points that exist on a single line because it causes the matrix determinant to be 0, creating a cross ratio of 0 or ∞ . Secondly, the points must be on an identical plane. The restriction of coplanarity stems from the requirement of invariance of the cross ratio. In order to eliminate permutation combinations with three points on a single line, a Grimm matrix [3] was used in experiment. For the purpose of checking whether the points are on an identical plane, each vector was checked to see if the value was between 2.0 and 2.8. In this paper, we call quintuple points after collinearity and coplanarity test as feature model candidate. The vector ordering techniques [7] was applied to the feature model candidates to extract robust outliers. This approach based on the Euclidean distance from mean invariant vector as Eq. (5), Eq (6). *PPIV*_i in Eq.(5) is the

vector of the i th feature model candidate, where the superscript j in Eq.(6) denotes the j th vector component.

$$PPIV_{mean} = (1/q) \sum_{i=0}^{q-1} PPIV_i, q: \text{feature model candidate}, \quad (5)$$

$$d_i = \sqrt{\sum_{j=0}^4 (PPIV_i^j - PPIV_{mean}^j)^2} \quad (6)$$

2.3 Matching using Similarity Function and Graham Search Method

If matching is performed for all feasible blocks by searching the entire salient map, it increases the time complexity as well as the mismatch rate. Therefore, we propose a method for conducting the matching stage for only the blocks with a high degree of similarity between two blocks based on a similarity assessment function. A similarity function R_N uses a correlation equation as shown in Eq.(7).

$$R_N = \frac{\mu_{mn} - \mu_m \mu_n}{\sigma_n \sigma_m} \quad (7)$$

Where μ_n, μ_m denote the mean in N, M, μ_{mn} is the mean intensity value of the pixel-wise product values of the pixels from two window positions, M, N. And $\sigma_n = \mu_{nn} - \mu_n \mu_n$ is the standard deviation in N.

While matching has been performed solely based on the $PPIV$ value in previous studies [8], it is irrational to use only the $PPIV$ value that has instabilities due to noise or small distortions in corner detections. In turn, this study applied the threshold value (0.08). Then a convex hull test was performed using the Graham searching algorithm to filter incorrect matching candidates, and get point- to-point correspondences. Number of points and neighboring relations, corresponding points lie on the convex hull are preserved during projective transformation [9]. Therefore, convex hull test can be additionally used to determine accurate 1:1 correspondence of a feature models. The Graham searching method uses back tracking that makes up for the disadvantages of the conventional method with numerous searching rounds. It assigns candidates to a solution set and finds an optimal solution through back tracking method. Whereas a general algorithm requires an exponential time, Graham searching algorithm only involves a polynomial time since is completely eliminates the solutions not meeting the conditions. Fig.3 displays the process of seeking points on the convex hull using the Graham algorithm.

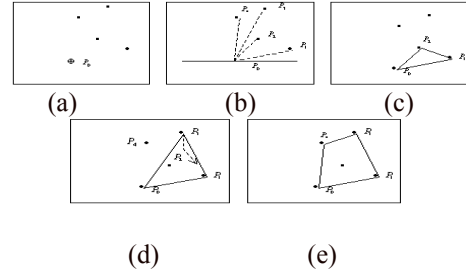


Fig. 3 Search Process of Graham Search algorithm

A From the five points, select P_0 with the least y-axis value and the greatest x-axis value as the axis point in Fig.3(a). Align the remaining points around P_0 with the angle with the positive x direction as the key in Fig.3(b). Then as shown in Fig.3(c), consider the three points P_0, P_1, P_2 as the points on the convex hull and add as another point P_3 on the convex hull. In order to verify whether P_2 is a point on the convex hull, check if P_3, P_2, P_1 are in the counter clockwise direction in Fig.3(d). If not, eliminate P_2 on of the three points as a point on the convex hull. In Fig.3 (e), P_4 is again added as a point on the convex hull, the direction of P_4, P_3, P_2 is examined, and P_4 is considered as a point on the convex hull. From the set of feature model candidates, the most outliers according to Eq.(6) were finally selected as robust feature models. Their graphical representation, along with the corresponding values of the invariant vectors $PPIV_i, 0 \leq i \leq 2$, is given in Fig. 4(left). These three candidates constitute robust feature models. During navigation same scene is viewed from a different vantage-point in Fig. 4(right)). Under a similar procedure coplanar quintuples are extracted and successful matches with reference image are examined. These quintuples constitute the recognized feature models.

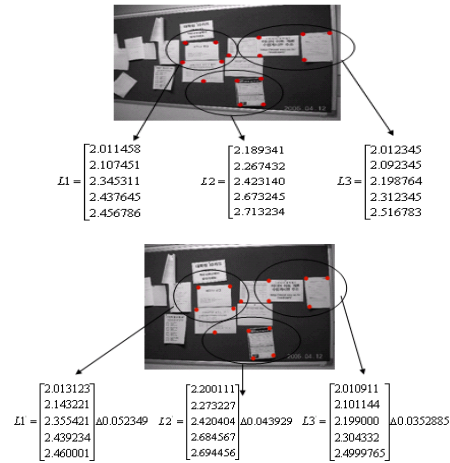


Fig. 4 Robust Feature Model Identified during learning(upper) and Recognized Results(below)

Finally, a transformation matrix of two images is calculated using feature model with determined point-to-point correspondence. It is obtained from the solutions of the linear system constituting the eight equations [10].

2.4 Camera Motion Parameter Estimation

This section describes a method for estimation of the camera motion factors using the acquired feature model set.

2.4.1 Coordinate Transformation

Where there is a 2-dimensional image coordinate x, y that corresponds with the 3-dimensional coordinate system under the perspective projection, the image coordinates x', y' according to translation and rotation can be expressed with 8 variables as in Eq.(8). If camera rotation is not substantial and the focal length is large and consistent, the relationship between the 3-dimensional coordinates and the corresponding 2-dimensional image coordinates can be expressed with an affine model as in Eq.(9) [11].

$$\begin{aligned} x' &= \frac{a_1x + a_2y + a_3}{a_7x + a_8y + 1}, y' = \frac{a_4x + a_5y + a_6}{a_7x + a_8y + 1} \end{aligned} \quad (8)$$

$$x' = x + \gamma y - \beta f = a_1x + a_2y + a_3, \quad y' = -\gamma x + y + \alpha f = a_4x + a_5y + a_6 \quad (9)$$

Furthermore, the camera rotation data including the focal length (f), tilt angle (α), pan angle (β) and swing angle (γ) can be extracted from the affine factors in Eq.(9).

2.4.2 Initial Value Estimation by Homography

Extraction of the camera motion factors using image can be regarded as a problem of finding a 2-dimensional homography. This is due to the fact that when points on a single plane in a 3-dimensional space are expressed in 2 dimensions under perspective projection, movements of these points are expressed with a 3x3 homography matrix [12]. If the homogeneous coordinates of the reference image is $X = (X, Y, W)^T$, and that of the input image $x = (x, y, w)^T$, and the correspondence relation between the two coordinates is $X \leftrightarrow x$, if the relationship in Eq.(10) is satisfied, there exists a homography matrix M between two images.

$$X = Mx \quad (M = [\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{t}]) \quad (10)$$

In the experiment, a homography matrix was calculated using the direct linear transformation algorithm

from the correspondence relationship of the four points of the extracted feature models. Assuming that the internal parameters are known, we can obtain the first two column vectors, and the remaining column vectors can be acquired using orthogonality of the camera's rotation matrix R with a cross product ($\mathbf{r}_1 \times \mathbf{r}_2$) of Eq.(11).

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \quad P = [\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{r}_1 \times \mathbf{r}_2 \ t] \quad (11)$$

Once the rotation matrix R is calculated, the rotation information of each axis is inferred as in Eq.(12).

$$\alpha = \text{atan}\left(\frac{r_{12}}{r_{11}}\right), \beta = \text{asin}(-r_{13}), \gamma = \text{atan}\left(\frac{r_{23}}{r_{33}}\right) \quad (12)$$

However, since the obtained α, β, γ are induced from a periodic function (\sin, \tan), there is the ambiguity of yielding different angles at similar values. Therefore, we use these data as initial values of the LM algorithm during the experiment to obtain the final solution with improved accuracy and convergence.

2.4.3 Nonlinear Minimizing Method by Levenberg-Marquardt

The LM algorithm is one of the minimization methods using nonlinear minimum square evaluation [2]. If the object function correctly approximated to a localized 2-dimensional function, the Gauss Newton method should be applied. Otherwise, a gradient reduction method is applied. The LM algorithm defines the average square error to evaluate the similarity between the output and the actual output. In turn, if the object function based on the affine model in Eq.(13) is denoted as $\chi^2(a)$, it can be set as Eq.(14).

$$y(x, y, \mathbf{a}) = \begin{bmatrix} u(x, y) \\ v(x, y) \end{bmatrix} = \begin{bmatrix} a_1x + a_2y + a_3 \\ a_4x + a_5y + a_6 \end{bmatrix} \quad (13)$$

$$\chi^2(\mathbf{a}) = \sum_{i=1}^N w_i \left[\frac{y'_i - y(x_i, y_i, \mathbf{a})}{\sigma_i} \right]^2 \quad (14)$$

In Eq.(14), y'_i denotes the i th input feature model, σ_i the i th data variance, and w_i the weight value of 0 or 1. If the object function of Eq.(14) is approximated to the form of a second-order equation for the factor \mathbf{a} using Taylor series, it can be expressed as Eq.(15).

$$\chi^2(\mathbf{a}) \approx r - \mathbf{d} \cdot \mathbf{a} + \frac{1}{2} \mathbf{a} \cdot \mathbf{D} \cdot \mathbf{a} \quad (15)$$

In the above equation, \mathbf{d} denotes the first-order differential matrix of the factor \mathbf{a} for the object function, and \mathbf{D} denotes the Hessian matrix, which is the second-order differential matrix of the factor \mathbf{a} . Furthermore, the factor \mathbf{a} can be calculated with the reverse-Hessian method as in Eq.(16). Conversely, if the above equation cannot be approximated to a second-order equation, it should be resolved with a gradient reduction method as in Eq.(17).

$$\mathbf{a}_{min} = \mathbf{a}_{cur} + \mathbf{D}^{-1} \cdot [-\nabla \chi^2(\mathbf{a}_{cur})] \quad (16)$$

$$\mathbf{a}_{next} = \mathbf{a}_{cur} - const \ ant \times \nabla \chi^2(\mathbf{a}_{cur}) \quad (17)$$

3 Experimental Results

The proposed algorithm has been implemented and experimentally verified in an indoor environment. The image was captures with Nikon Coolpix 3200 and standardized into 640×480 pixels. For the feature models recognition experiment, three robust feature models were extracted from the reference image through learning, and the recognition rate was calculated using 50 input images with different camera views. Table 1 displays the recognition results. CR is the percentage of correct recognitions of robust feature models, and MR (Miss-Recognitions), FP (False-Positives) and FN (False-Negatives) are percentage of incorrect recognitions, recognitions of other feature models as robust models and elimination of robust feature models, respectively.

Table1. Recognition Results of Feature Model

Feature Model	CR	MR	FP	RN
L1	48(96%)	1(2%)	0(0%)	1(2%)
L2	45(90%)	1(2%)	3(6%)	2(4%)
L3	49(98%)	0(0%)	1(2%)	0(0%)
Total	47.3(94.6%)	0.7(1.4%)	1.3(2.7%)	1(2%)

From the way that comparison of recognition rates using the conventional method and the proposed method of using the similarity function and the Graham search algorithm, the recognition rate of the proposed method was 94.6%, a 1.3% improvement from the conventional method of 93.3%. Furthermore, when the robust feature models were not included, the rate decreased from 2.7% to 2%, indicating that performance had been improved for the proposed method. Figure 5 compares the accuracies when motion parameter is obtained with the homography matrix and using the improved LM algorithm for 10 different images with known rotation information. The x axis contains index of 10 images used in the experiment and the y axis displays difference between known rotation data and

the data acquired from homography and the data obtained by using the improved LM algorithm with homography's results as initial values of LM. Since the rotation information obtained from homography is induced from a periodic function, there is a significant error between the actual rotation data. However, when the LM algorithm is deployed, the result improved substantially.

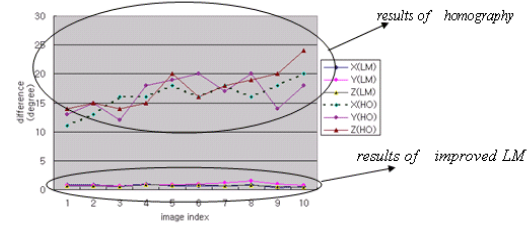


Fig.5. Accuracy Comparison

4 Discussion

In this paper, we proposed a method for extracting features unrelated to the camera viewpoint to calculate the camera's motion factors. Projective and point permutation invariant vectors have been employed to characterize feature patterns. The method proposed for the motion factor calculation involves making up for the disadvantages inherent in the conventional LM algorithm to improve convergence and accuracy, allowing extraction of optimal camera motion factors. Features used in this paper exist on a coplane, and there are restrictions under indoor environments where there is a geometric model. In turn, there must be continuous studies on extracting features that is robust to the outdoor noise and without restrictions. Additionally, uncertainty issues due to total or partial occlusions of stored features at recognition need further research.

5 Acknowledgement

This work was supported in part by IITA through IT Leading R&D Support Project.

6 Reference

- [1] Christian Drewniok and Karl Rohr, "High-Precision Localization of Circular Landmarks in Aerial Images," Proc. 17. Dagm-Symposium, Musterkennung 1995, pp.594-601, Bielefeld, Germany, 13-15. September 1995.
- [2] Martin T.Hagan and Mohammad B.Menhaj, "Training Feedback Networks with the Marquardt Algorithm", IEEE Transactions on Neural Networks, Vol. 5, No. 6, November 1994.

[3] K. Kanatani, "Computational Projective Geometry," CVGIP:Image Understanding Workshop, Washington, DC, pp. 745-753, 1993.

[4] Reiner Lenz and Peter Meer, "Point Configuration Invariants under Simultaneous Projective and Permutation Transformations," Pattern Recognition, Vol. 27, No. 11, pp. 1523-1532, 1994.

[5] S. Birchfield, "KLT:An Implementation of the Kanade-Lucas-Tomasi Feature Tracker, <http://vision.stanford.edu/~birch/klt/>".

[6] Panos E. Trahanias, Savvas Velissaris and Thodoris Garavelos, "Visual Landmark Extraction and Recognition for Autonomous Robot Navigation," Proc. IROS 97, pp. 1036-1042, 1997.

[7] V. Barnett, "The Ordering of Multivariate Data," Journal of Royal Statistical Society A, Part 3 139 pp. 318-343, 1976.

[8] Vicente, M.A., Gil, P., Reinoso., Torres, F, "Object Recognition by Means of Projective Invariants Considering Corner-Points," Proc. SPIE. Vol. 4570. pp. 105-112. 2002.

[9] J.L. Mundy, A. Zisserman, "Geometric Invariance in Computer Vision," MIT Press, Cambridge, MA, 1992.

[10] Fishler, M.A. and Bolles, R.C., "Random Sample Consensus: A Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography," Commumination ACM, vol. 24, no. 6, pp. 381-395, 1981.

[11] Jang SeokWoo, "Shot Transition Detection by Compensating Camera Operations," Soongsil University Press, 2000.

[12] Hartley, R. I, Zisserman, A, "Multiple View Geometry in Computer Vision," Cambridge University Press, 2000.