International Journal of Innovative Computing, Information and Control
Volume **8**, Number **6**, June **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 3981–3996

IMPLEMENTATION OF A REAL TIME INTEGRATED FILTER SYSTEM WITH GEOMETRIC MATCHING FILTER COMBINED SIFT FOR RELATIVE MOTION ESTIMATION

Kyungsuk Na[1], Sukchang Yun[1], Mingyu Gwak[2], Young Jae Lee[1] and Sangkyung Sung[1,*]

[1]Department of Aerospace Information Engineering
Konkuk University
263 Achasan-ro, Gwangjin-gu, Seoul 143-701, Korea
*Corresponding author: sksung@konkuk.ac.kr
{amerisan; younglee}@konkuk.ac.kr

[2]MicroInfinity
Suwon, Gyeonggi-do, Korea
rolensia@nate.com

Received February 2011; revised June 2011

Abstract. *This paper investigates a real time integrated filter system that measures and estimates a relative motion of mobile vehicles based on a newly presented vision processing algorithm. For this, the relative motion information is processed and computed by an INS/vSLAM integrated system that recognizes environment variation surrounding the vehicle and uses an image-based SLAM algorithm to provide auxiliary navigation information. The presented image processing algorithm improves the tracking performance by combining the conventional SIFT with a newly suggested geometric matching filter. Also, in an effort to manage feature points loss during the process of motion under an unstructured environment, a real time algorithm that dynamically selects feature points based on geometric criterion is considered. For experiment, a low graded IMU sensor and a single vision sensor are chosen and the integrated inertial SLAM system is implemented on a PC based virtual instrumentation platform. With the use of Cartesian robot, two dimensional motion tests are done to verify the performance of the feature tracking algorithm and integrated filter system for an accurate motion estimation.*
Keywords: Integrated filter, Relative motion, INS/vSLAM, Feature point, Geometric matching, Tracking

1. **Introduction.** Measuring sensors data and estimating vehicle's dynamic information such as relative motion is one of the essential requirements for most mobile vehicles, especially for 6 degree-of-freedom (DoF) systems. For instance, aerial vehicles widely use an inertial navigation system (INS) or other dead-reckoning sensors in estimating the position and attitude information for their standalone operation. In particular, INS calculates vehicle's position, velocity, and attitude based on the acceleration and angular velocity information, but it increasingly fails to estimate accurate positions over long periods as integration errors become greater [7-10,13,20,25]. For this reason, an INS/GPS integrated filter system is generally used to avoid accumulation errors resulting from the INS-only system [11,16]. Affected by geographic features and jamming signal, on the other hand, GPS may fail to provide accurate navigation information on a regular basis. To address this problem, active research has recently been underway on an integrated measurement system that uses a vision sensor as the auxiliary sensor [2-6,12,19-24]. A

vision sensor can use geographic information on the surroundings, and as such may provide improved motion estimates where GPS cannot be utilized.

The core principle for implementing vision-based simultaneous localization and mapping (vSLAM) is the use of feature points [1,6]. Major algorithms for extracting and tracking such feature points include Harris Corner, SUZAN, and KLT (Kanade-Lucas-Tomasi) [18,22]. Meanwhile, the scale-invariant-feature-transform (SIFT) algorithm – widely used in object recognition and tracking in computer vision system – can extract feature points that are invariable with change of image size and rotation [1,14,15,17]. Therefore, this algorithm can better extract and match feature points in relatively complicated environments than the aforementioned algorithms, despite its computational complexity. However, as mentioned, using SIFT as the image processing algorithm is greatly limited in a real time application like UAV since the detector and descriptor stages involve quite large dimension in feature point extracting process.

This paper seeks to develop an improved vision-aided relative motion estimation method through a newly proposed geometrically matching filter-combined SIFT (GMF-SIFT). The proposed method contains the following steps. First, feature points that are robust against changes in image size and rotation are extracted to implement feature tracking function through a reduced dimension SIFT algorithm [6]; then a new feature point matching filter based on image's geometric conditions is applied to extracting feature points that remain more consistent and robust against external disturbance and other error factors. With the geometric matching process, the presented tracking performance is revealed to be better than a typical SIFT despite a smaller dimension at descriptor stage, which in turn reduces computational time. Eventually, the suggested GMF-SIFT tracking algorithm is applied to the INS/vSLAM integrated filter system to verify its motion estimation performance. To demonstrate the performance of the suggested algorithm, a comparative analysis is made with an INS-only solution and the KLT algorithm-based integrated filter solution using experimental data. In Section 2, it presented the principle of the proposed GMF-SIFT, which is followed by a real time implementation result. Section 3 describes the construction of inertial SLAM integration filter and Section 4 illustrates the experiment result. Finally, Section 5 provides summary and conclusion.

2. Geometric Matching Filter Combined SIFT for Real Time Feature Point Tracking.

2.1. The proposed geometric matching filter combined SIFT.

Suggested by D. Lowe, SIFT [15] is widely used in detection and object recognition in image processing [27]. The algorithm can be broadly divided into two phases of detection and descriptor. In the detection phase, images are divided into a predefined size and Gaussian smoothing is carried out to find extreme points using difference of Gaussian (DOG) [14]. Therefore, the feature points obtained in the detection phase are relatively robust against scale changes. On the other hand, feature points that are robust against rotation are mainly extracted during the descriptor phase. Here, each feature point generates a descriptor consisting of 128 dimensions. The well-known KD tree is generally used to compare descriptors and consequently match feature points.

Accurate matching of feature points between consecutive image frames is essential to obtain meaningful navigation information from feature point information through the image sequences. For this purpose, this paper devises a robust feature point matching filter method to eliminate feature points that may be inaccurately matched during the KD tree process in the descriptor phase. The suggested matching filter compares relative geometric angles between feature point pairs in continuous image frames to eliminate

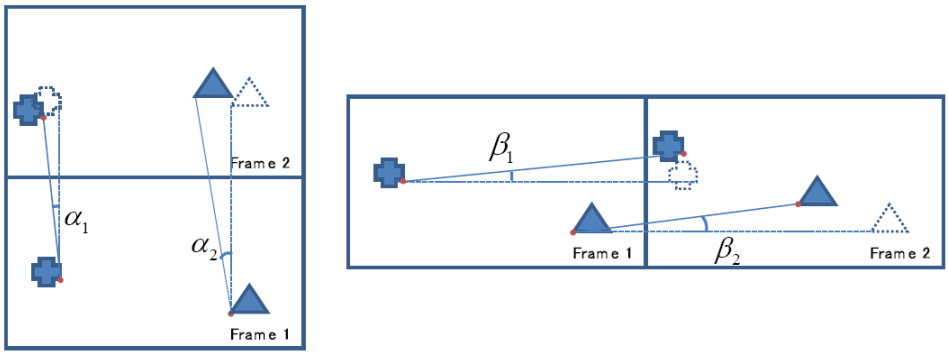

FIGURE 1. Block diagram illustrating the geometrically matching filter principle

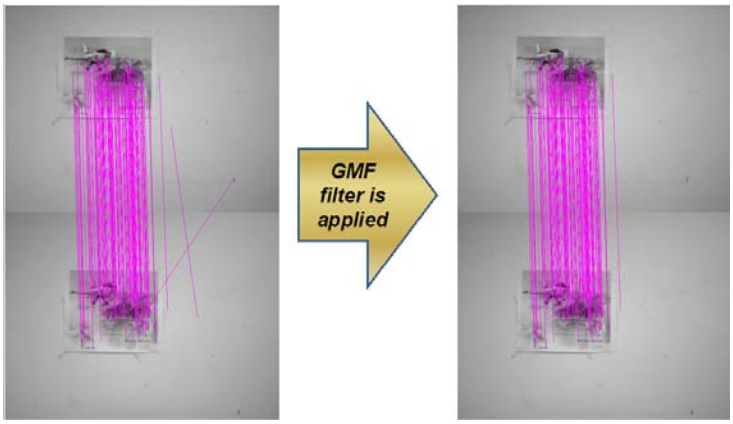

FIGURE 2. Matching performance comparison before and after application of GMF

tentative mismatched points. Figure 1 shows the conceptual diagram about the proposed matching filter. In the figure, note that a longitudinal and lateral expansion is drawn with respect to the original image by using the next image frame. Then each angle deviation from paired points is analyzed to determine the sound feature points. The process of implementation is given in detail as follows: first, the descriptors of feature points extracted in the detection phase are compared to match feature points between the current and previous frames. Then the geometric angles of the x and y axes (i.e., α_i and β_i) are collected for each corresponding pair. Normally, continuous image frames are assumed and therefore consecutive geometric angles involve little variation in measured angles. Therefore, more accurate sets of feature pairs can be obtained in the descriptor phase if a critical threshold (α_T, β_T) is appropriately set to discriminate the induced angle pair (α_i, β_i) of every matched point. The threshold is heuristically chosen such that the total number of acquired feature points is controlled.

The improved performance of the GMF-SIFT algorithm is demonstrated in Figure 2. Assumed a general unstructured vision environment, target images of low contrast level are used. On the left side of the figure, the feature point pairs of two continuous images are visualized using the conventional SIFT algorithm. In the right side, it is shown feature point pairs after the newly proposed matching filter is applied, where a lower descriptor dimension of 64 and the critical threshold of (α_T, β_T) with 5 deg are used. The figure shows that more accurate feature point pairs can be obtained by removing feature points having excessive geometry deviation, after applying the geometric matching filter augmentation.

TABLE 1. Comparisons between original KD tree method and the GMF method

	Original Matching Filter	Modified Matching Filter
Total Matching Point (TMP)	70	67
Miss Matching Point (MMP)	2	0
TMP/MMP (%)	2.8%	0%

Besides, despite a lower descriptor dimension, the proposed method provides a superior matching performance. This suggests an advantage in the real time implementation with better tracking performance. Table 1 compares matching performances from the original KD tree method and the proposed method. In the GMF algorithm, the total matching point is slightly decreased while the mismatched points are completely removed.

2.2. **Feature point tracking principle and configuration.** The enhanced performance in obtaining good feature point set also results in better tracking performance. Unlike feature point matching part through comparison of available feature points in the previous and current frames, tracking part usually takes advantage of fewer feature points to observe relative displacements between image frames. For this, the feature point information of initial images ought to be persistently preserved and compared per frame during tracking process.

In a real time environment, independent matching part alone between image frames cannot provide displacement information on the feature point of interest. This is because the reference position information from the initial frame is not transferred through iteration. Therefore, it is crucial to construct a feature tracking algorithm that continues to hold local feature point information from the initial image frame. Maintaining candidate feature points for tracking function is realized by indexing uniquely each candidate point from initial step. The appended index for feature point continues to remain during the image processing time with a limitation of total number, until it is no longer used for navigation as it disappears in the image. Thus the sequential tracking of candidate feature points in the suggested GMF-SIFT algorithm is realized with the independent indexing function for data association.

Figure 3 shows an overall block diagram containing the structure of the real time feature matching and tracking algorithm. First, a consecutive image pair is fed into the matching part to select candidate to-be-tracked feature points in the initial stage, and the tracking

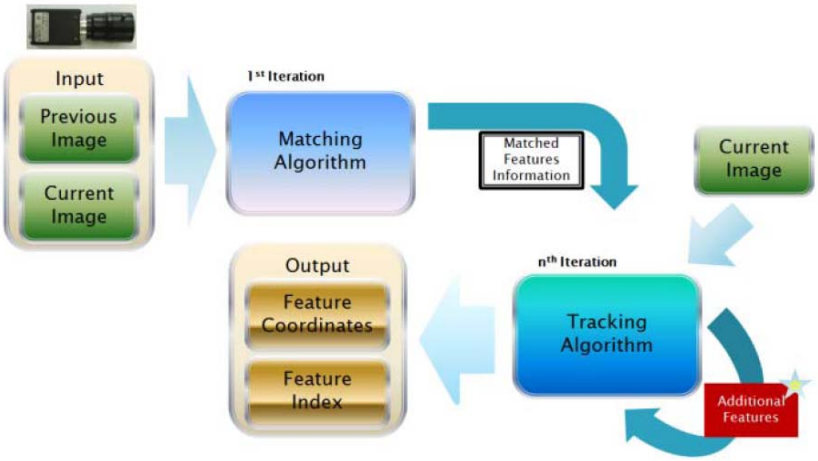

FIGURE 3. Conceptual diagram of real time feature points tracking algorithm

part generates and matches descriptors of these feature points to associate with those of the previously determined feature points. Tracking can be undertaken repeatedly to keep track of feature points via indexing; when the number of to-be-tracked feature points decreases, additional feature points are selected from the candidate group and added to the feature points group of interest to fill up a pre-defined number. For each iteration, the X-Y coordinate values and indices of the tracked feature points are returned as output values, and this information is restored and converted into azimuth and elevation angles in the integrated filter system.

2.3. **Real time implementation and tracking loss management.** This section illustrates the implementation of the proposed GMF-SIFT method under real time embedded system environment. Figure 4 shows block diagram of the GMF-SIFT algorithm implementation. The reduced SIFT core part and geometric matching filter part is realized

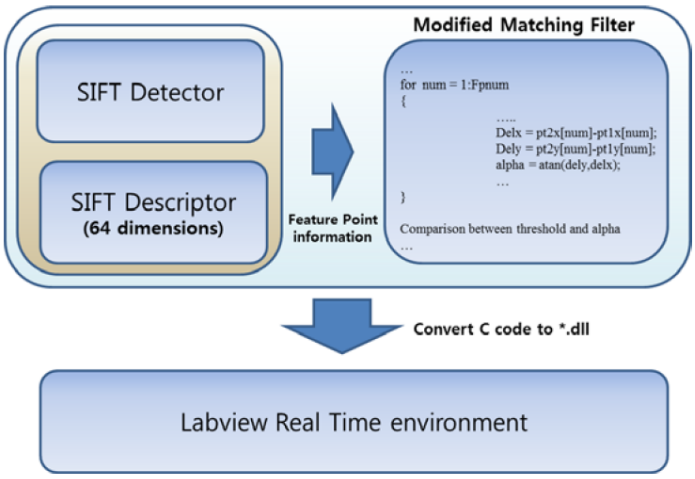

FIGURE 4. Matching block diagram of the GMF-SIFT algorithm implementation in real time platform

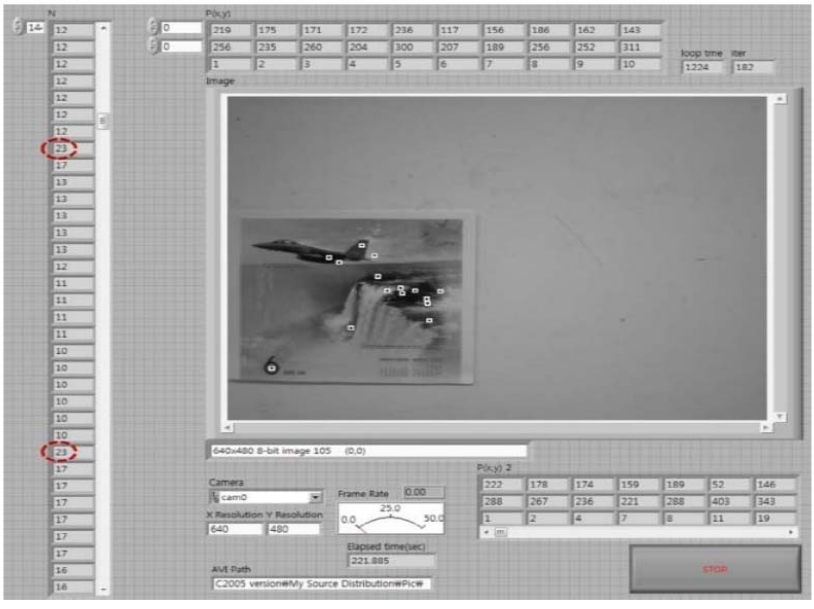

FIGURE 5. Picture of the user interface of the developed real time features tracking

through C code and converted into the $*$.dll execution file, which is downloaded in the real time embedded computer. For user friendly interface, the image and other interface parameters including threshold, feature points number, the X-Y coordinate on the selected feature points in image frame are shown in the front panel of desktop. Figure 5 illustrates the real-time user interface and communications for real-time tracking are realized by Labview graphical programming while the computationally heavy GMF-SIFT algorithm is embedded by a complied C code.

For a visualization of the acquired image and tracked feature points, a VGA-sized central image frame marked with concurrent feature points is provided along with the pixel position data, number of candidate points, frame rate, etc. For the implementation of filter system with GMF-SIFT algorithm, a desktop PC installed with 2.6 GHz, dual core CPU is used. The used image size is 640×480 pixels and feature points managed during each epoch are set to 50.

Notation and descriptions of each interface are provided in Table 2. In the table, each interface variable provides user interactive platform for real time feature point tracking function. For instance, the indices of the initially tracked feature point $P(x, y)$ and the currently tracked feature point $P(x, y)2$ are compared to identify feature points that are lost in the course of tracking. In addition, when the number of feature points currently being tracked, i.e., number of $P(x, y)2$ falls below threshold, feature points kept in the candidate group are retrieved to be added to the to-be-tracked feature points. In Figure 5, the threshold of feature point number is set to 15. Other information for the tracking function is reserved like loop time spent, iteration number, frame rate and camera setting related parameters.

TABLE 2. Interface for real-time feature tracking environment

Parameters	**Note**
N	Number of feature points to-be-tracked in the particular loop
$P(x, y)$	Coordinate and index of the initial feature point
$P(x, y)2$	Coordinate and index of the currently tracked feature point
loop time	Time spent in the current loop
iter	Number of iterations made thus far
Frame Rate	Image frame output speed
Camera	Camera setting
X Resolution, Y Resolution	Camera pixel
AVI Path	Image storage path

Admittedly, feature point matching at each epoch is performed independently from previous matching information, as it only utilizes a pair of the latest two images alone for matching process. This has apparently no relation to the dynamic state of the aircraft; thus as long as the quality of image information is guaranteed, robust information should be provided continuously regardless of any abrupt movement. Feature point tracking, however, is dependent upon the initial information of feature points, and thus a large number of feature points may be lost instantly when the dynamic state of the aircraft changes abruptly, consequently discontinuing the supply of sufficient information to the integrated filter. To address this problem, feature points used through the tracking process are managed and stored according to their quality grades each time the algorithm is

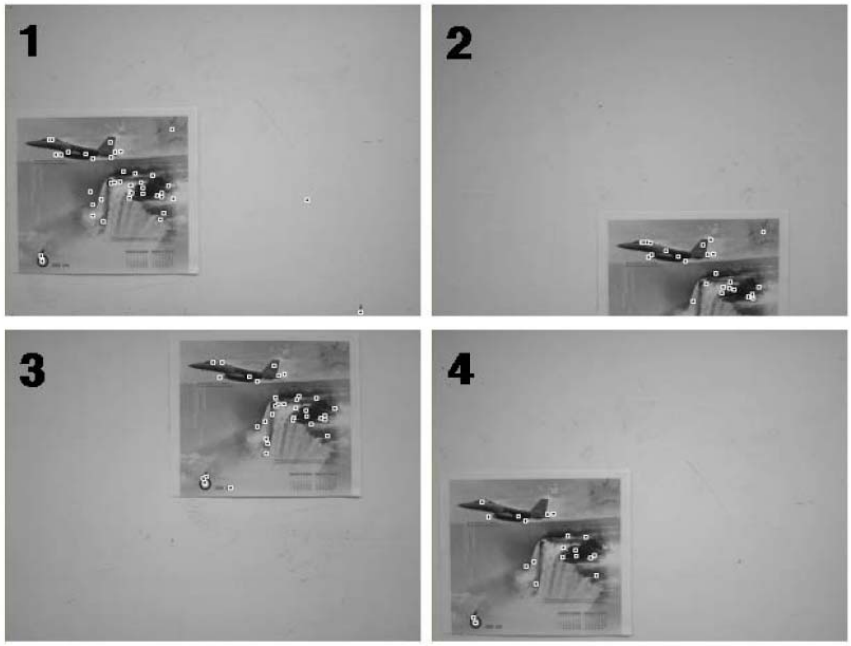

FIGURE 6. SIFT-based real-time tracking of feature points

executed. The sub-features kept in the candidate group are added to the to-be-tracked feature points when it is concluded that some part of current to-be-tracked feature points have been lost.

Finally, Figure 6 demonstrates real-time tracking performance based on the matching filter-combined SIFT method. As observed from the figure, if some of the feature points disappear in the frame – as exemplified in upper left and upper right in the figure – or tracking loss occurs, the process of finding feature points in sub-features set is repeated attempting to hold the number of feature points transmitted from the matching stage. However, if there are limited candidates, the algorithm only provides geometrically matched points less than the threshold value. And when the target appears back, the maintained index information is again applied for tracking. From experimental result under PC environment, the real time tracking performance has been verified with an update rate of 1Hz, with 30 feature points.

3. **Integrated Filter System for Inertial SLAM.**

3.1. **Integrated filter structure and system modeling.** The integrated filter system for inertial SLAM uses the indirect filter structure of leveraging navigation information through INS (inertial navigation system) and vision sensor. In this, the vision sensor based position error information is used as the filter measurement. In an effort to efficiently deal with variation of the number of feature points, it also adopts a distributed filter structure. A brief conceptual diagram of the filter is provided in Figure 7 [4]. Unlike the direct method that uses the system state vector as a filter variable, the indirect method sets the error as the filter state variable for estimation purposes [5]. Thus, the navigation information of the INS, provided at 100Hz, is ultimately generated as a navigation solution after correction of estimation errors offered by the integrated filter. Meanwhile, as for information on feature points in the image frame, the measured azimuth and elevation angles and the delayed initialization technique are used to provide information on the initial positions of those feature points with respect to the navigation coordinate system. The state variable of the filter is then estimated using the initial positions of the feature

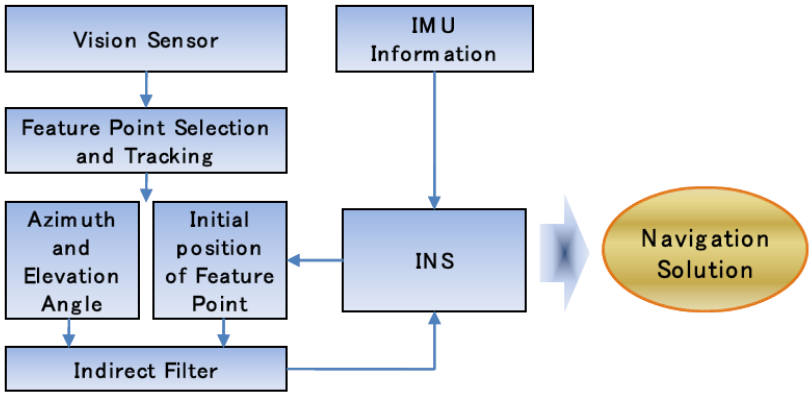

FIGURE 7. Algorithm process and structure of the adapted INS/vSLAM indirect filter

points and the measured line-of-sight vector information between the aircraft and feature points [6,21]. Note that the measured data is expressed as a non-linear function of azimuth and elevation angles.

3.2. **System and observation model.** The system and observation models for the configuration of the inertial SLAM integrated filter are introduced in the following. The system model uses the simple INS error model assuming little effect from earth rotation. Also sensor error models are not included in the filter state vector for computational simplicity. Thus, the system model used in the integrated filter is depicted by

$$x\left(k+1\right)=\Phi_k x\left(k\right)+Gw_k, \tag{1}$$

where state transition matrix and input matrix are given by

$$\Phi_k=\begin{bmatrix} I & I\cdot\Delta t & 0 & \\ 0 & I & [f_b\times]\cdot\Delta t & \\ 0 & 0 & I & \\ & & & I_{n\times n} \end{bmatrix},\quad G=\begin{bmatrix} 0 & 0 \\ C_b^n & 0 \\ 0 & -C_b^n \\ 0_{n\times 1} & 0_{n\times 1} \end{bmatrix}. \tag{2}$$

In deriving (2), state variables are defined as $x\left(k\right):=\begin{bmatrix}\delta x_v^T & \delta x_{f1}^T & \cdots & \delta x_{fn}^T\end{bmatrix}^T$ with δx_v as the vehicle's error state vector position, velocity and attitude. Thus δx_v is defined as $\delta x_v=\begin{bmatrix}\delta p_v^T & \delta v_v^T & \delta\psi_v^T\end{bmatrix}^T$. Input vector in the system model is given by

$$w_k=\begin{bmatrix}\sqrt{\Delta t}f_b \\ \sqrt{\Delta t}w_{ib}^b\end{bmatrix}, \tag{3}$$

where f_b and w_{ib}^b represents specific force and angular velocity in the body axis, respectively. Coordinate system transformation matrix from body to navigation frame is denoted by C_b^n. Next, the observation model uses the equation for calculating the elevation and azimuth angle of the vision sensor, which is expressed in the following nonlinear equation

$$z(k)=h(x(k),v(k)), \tag{4}$$

where z is the observed value and x the state vector, which includes the INS state error vector on the aircraft as well as the position of feature points. For simplicity, the angle measurement noise v, can be assumed as an additive Gaussian random process as in the following equation:

$$z(k)=h(x_v(k),x_{fi}(k))+v(k), \tag{5}$$

where its unit is converted into radian and stochastic property is described by a normal distribution with the covariance of R, i.e., $v\sim N(0,R)$. Finally, the position of feature

points in the navigation coordinate system can be converted in the sensor coordinate frame as delineated in (6)

$$x_{fi}^{s} = C_{b}^{s}\left(C_{n}^{b}\left(x_{fi}^{n} - p_{v}^{n}\right) - p_{s}\right), \tag{6}$$

where x_{fi}^{s}, x_{fi}^{n}, and p_s are feature point position vector in sensor frame and navigation frame and camera lever arm vector, respectively. The formulating equation between the positions of feature points and the measured values in the sensor coordinate frame is written as follows:

$$z_{FP}^{i} = \begin{bmatrix} \tan^{-1}\left(\dfrac{y_i}{x_i}\right) \\ \tan^{-1}\left(\dfrac{z_i}{\sqrt{x_i^2 + y_i^2}}\right) \end{bmatrix}, \tag{7}$$

where measurement vector is defined by $z_{FP}^{i} := [\psi^i \ \theta^i]^T$ and x_i, y_i, z_i represent each element of LOS vector in the sensor frame, i.e., x_{fi}^{s}. In this configuration, note that ψ and θ indicate feature point's azimuth and elevation angle of the local horizontal plane in the sensor frame, respectively.

3.3. **Nonlinear integration filter algorithm.** Given the use of non-linear model and non-gaussian noise characteristics, this paper employs a particle filter as an integrated filter implementation. The particle filter is a recursive Bayesian filter based on Monte Carlo technique that directly takes advantage of probability distribution functions to estimate the state with the help of point mass computation. Owing to its adaptive performance against a nonlinearity and random noise distribution, it is expected to show robust estimation performance for a nonlinear system model and non-gaussian noise distribution considered in this paper.

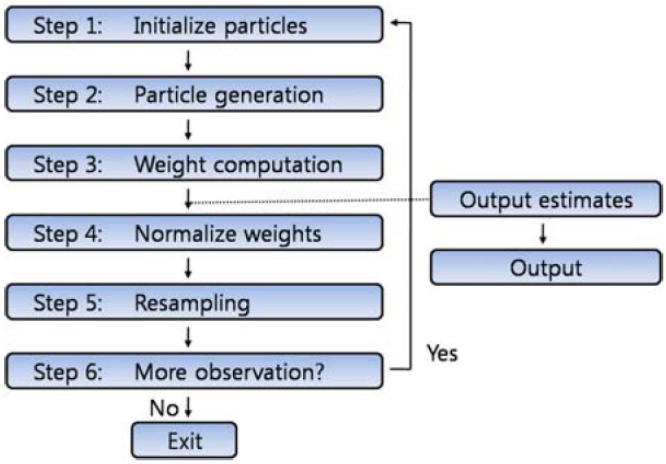

FIGURE 8. Implemented particle filter algorithm flow diagram

A typical sequential importance resampling (SIR) filter approach is taken for the filter implementation as given in [9,12]. The SIR algorithm can be derived from the SIS algorithm by an appropriate choice of the importance sampling and modification of the resampling step as follows. (i-1) the importance density $q(x_k/x_{k-1}^{i}, z_{1:k})$ is chosen to be the prior density $p(x_k\,|x_{k-1})$ and (i-2) the resampling step, which is to be applied at every time index. ii-1) generate and initialize particles with equal weight, ii-2) using measurement, update the weights by the importance function $w_t^i = w_{t-1}^i p(z_t/x_t^i)$, then with normalization of weight, ii-3) apply resampling for modified weight and particles, ii-4)

compute estimate via $\hat{x}(t) = \sum_{j=1}^{N_s} w_t^j x_t^j$, ii-5) predict state by computing system model, ii-6) return to the initialize step, if continued. In a sequential framework, the particles are first drawn from the proposal distribution, and the particle weights are evaluated based on the system transition prior and measurement update. Next, the resampling is applied to avoiding the degeneracy problem. Considered the nonlinear model characteristics, this paper implements a SIR type particle filter as depicted in the process outlined below. Notably in estimating errors via the suggested nonlinear integrated filter configuration, angle measurement for every feature point plays a significant role as it involves a dominant nonlinear model. The elaboration on the detailed particle filter implementation algorithm is out of the focus of the paper and skipped.

4. Inertial SLAM Test and Discussion.

4.1. **Test environment.** In order to demonstrate the performance of the proposed integrated filter system, laboratory tests are done such that comparative estimation results using other algorithms like INS-only and KLT based INS/vSLAM are obtained simultaneously. For this, the following testing device and environment is built. Basler's A601f is used as the vision sensor. The used mono-type sensor provides a resolution performance of 640×480. The pixel size is $9.9\mu\text{m} \times 9.9\mu\text{m}$, and IEEE 1394 is employed as a communication link interface. Micro Infinity's GA3350M is used as the employed IMU with a gyro bias stability of 30deg/h, which supports an RS-232 serial communication interface. Tri-axial Cartesian coordinate robot is used to travel three times around a circle with a diameter of 50cm. Here, the measured value renewal cycle is 50Hz for the IMU, and the vision sensor data transmit rate is 10Hz. The distance between the vision sensor and the image realizing the testing environment is around 2.1m. Camera is built on a tri-axial, orthogonal coordinate motion robot to rotate at a scheduled speed. In the experiment, it provides the programmed circular trajectory during 220 seconds for image acquisition. After obtaining individual data from the IMU and the vision sensor, the suggested feature point matching filter-combined SIFT and the competing KLT algorithm are applied to finally deriving inertial SLAM position estimation as the navigation solutions. The KLT algorithm used for relative comparison of estimation performance is presented by Clemson University [26]. Here, note that both tracking algorithm provides a real time

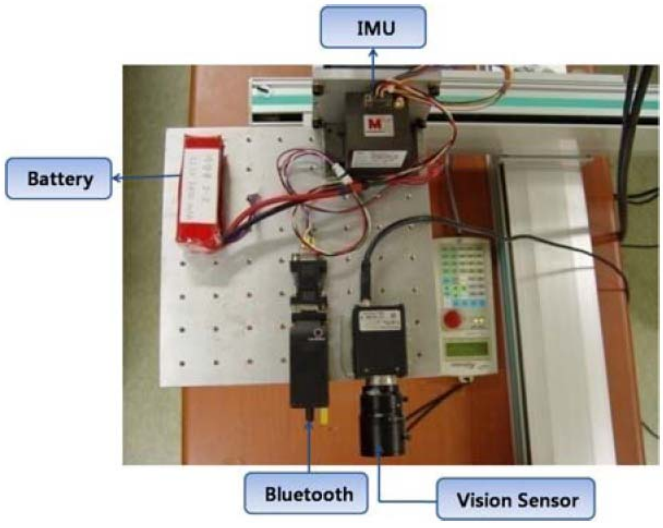

FIGURE 9. Indoor test hardware equipments for vision-aided motion estimation

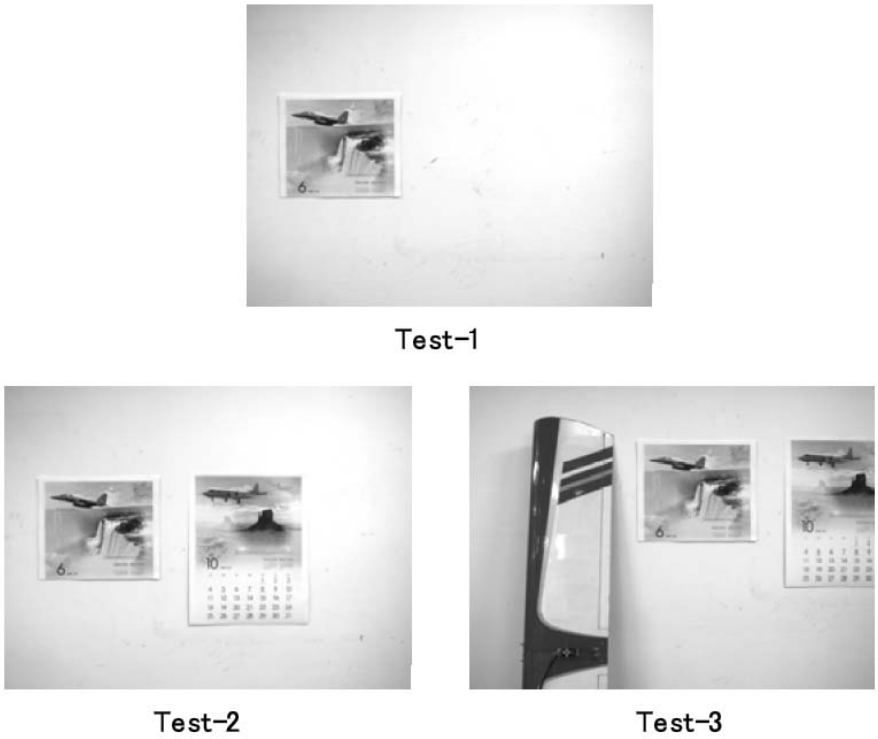

FIGURE 10. Test environment build-up using three cases with different image frames

feature point measurement update to the filter, yet the final integrated filter result in the form of motion estimation is acquired via a post-processing method for computational convenience. Figure 9 shows the equipment arranged on the stabilized platform of the tri-axial Cartesian coordinate robot for testing purposes. The X-axis of the IMU coordinate system is in line with the optical axis of the vision sensor; misalignment errors from sensor arrangement are assumed to be negligible. In an endeavor to verify the performance of vision-aided integrated filter system under wide-ranging unstructured environments, the testing environment is altered using three sets of image conditions. Figure 10 shows the initial images of each test environment.

The images gradually become more complicated as the test proceeds (i.e., from Test 1 through Test 3) to distinguish the inertial SLAM integrated filter's motion estimation performance under various environments. A total of six trajectory tests are carried out – two in each of the three testing environments. In deriving trajectory solutions of inertial SLAM integrated filter through post-processing, 10 Monte Carlo simulations are performed for each image.

4.2. **Analysis of test results.** To verify the performance of the proposed integrated filter system, a circular trajectory data provided by the Cartesian robot is taken as accurate reference data. The robot can generate navigation error under sub-centimeters as it rotates along the pre-determined trajectory. Figure 11 visualizes the results of selecting first data set from each case and performing INS-based pure motion estimation. As the robot moves with the circular trajectory perpendicular to the X-axis, the result is drawn with respect to the Y-Z plane. The red cross-shaped lines represent the reference trajectories obtained through the Cartesian robot and the dotted lines are the trajectories obtained from the pure INS navigation algorithm. As shown in the figures, the convergence characteristics of position estimation results based on an INS-only estimation are

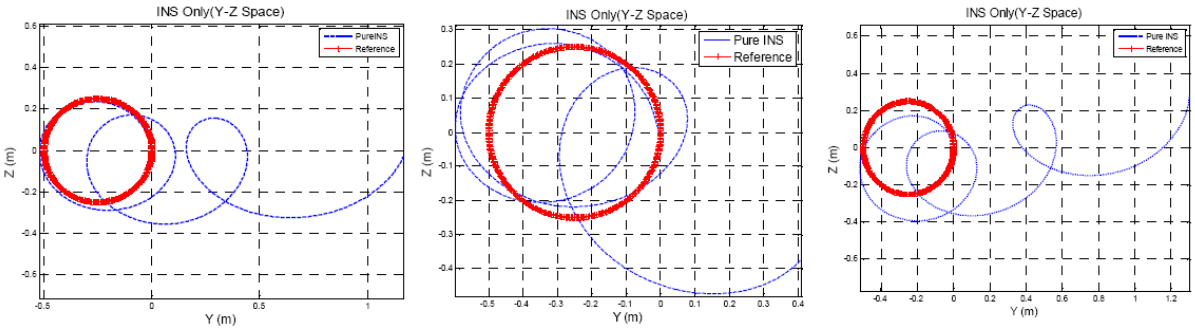

FIGURE 11. Pure navigation-based position estimation (Y-Z plane) – from left, Test A-1, B-2 and C-1

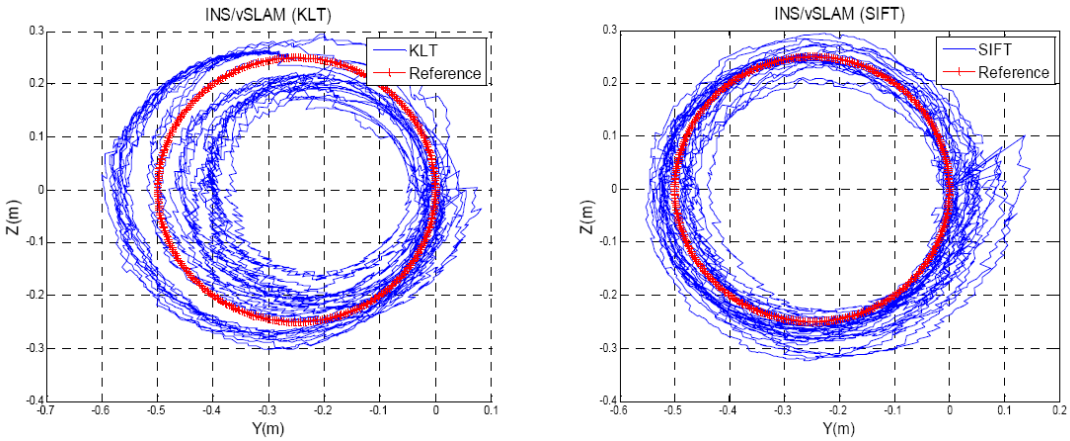

FIGURE 12. Estimated trajectory of INS/vSLAM – Test B-2 (KLT vs. proposed GMF-SIFT)

quickly degraded over time. This mainly results from bias error accumulation of IMU gyro and accelerometer used in the test – caused by bias with random constant model.

Next, the estimation results of the INS/vSLAM integrated filter system are reviewed by investigating data set of each test case. The presented data set are as follows: the second data set for test A and test B (A-2, B-2), and the first data set for test C (C-1). In each case, INS/vSLAM estimation results of two competing tracking algorithms are compared to demonstrate the performance of the proposed matching filter integrated navigation algorithm. For this purpose, 10 Monte Carlo simulations are performed on the feature point matching filter-combined SIFT and the KLT algorithm to show their cumulative trajectory errors, respectively.

As shown in the figure, both KLT and feature point matching filter-combined SIFT demonstrate estimation results that are close to the reference trajectory when compared with the previous INS-only navigation result. However, with the help of more accurate and robust tracking property of feature points presented by the proposed tracking algorithm, the estimated trajectories provides better consistency and repeatability than KLT based one during 10 times Monte Carlo simulations, which is observed in Figure 12.

With respect to estimation results based on each of the Monte Carlo simulations, the initial filter solutions differ by random bias, causing offset errors in the actual values accordingly. The average estimated trajectory, however, is shown to be very close to the real circular trajectory in both cases. This demonstrates that correction of relative position

errors in image-based motion estimation significantly reduces the extent of divergence compared with INS-only case. Meanwhile, the positions estimated by INS/vSLAM are not as smooth as those estimated by INS alone. This is because the update frequency of corrected values in image-based estimation (approximately 1Hz) is considerably lower than that of INS.

A performance analysis can be considered from the perspectives of accuracy and precision. To analyze accuracy levels, the error between the reference trajectory and the INS/vSLAM estimated trajectory is analyzed together. For this, a comprehensive illustration is given statistically by comparing the INS/vSLAM integrated navigation trajectories using the proposed tracking algorithm and KLT, respectively. Particularly, the error and standard deviation between the trajectory obtained from 10 Monte Carlo simulations and the reference trajectory are calculated per test set. The consequent estimation results are summarized in Table 3.

As demonstrated in Table 3, the solution of INS/vSLAM integrated filter – derived using KLT and the proposed matching filter-combined SIFT – is much better than that obtained with INS only. The error of INS-only estimation ranges from 10cm through 40cm, but the error range of the modified SIFT and KLT is relatively small, below 5cm in most cases. Next, performance comparison between the modified SIFT and KLT shows that the enhanced SIFT excels in all of the six tests. Especially in Tests B-2 and C-2, the mean error of SIFT is about half of KLT's error. This implies that the INS/vSLAM filter solution using the proposed matching filter-based SIFT is superior to the other algorithm with KLT.

The performance of each estimation method can also be compared in terms of precision. In this test, positions are estimated by traveling around a circle with a diameter of 50cm repeatedly, and thus a dense distribution of position trajectory values equates with greater precision. The standard deviations of the trajectories can be used to compare their precision levels. As demonstrated in Table 3, not only the mean error but also the standard deviation of the feature point matching filter-combined SIFT are smaller than those of KLT. In other words, in the INS/vSLAM-based comparison of estimated position values, the estimates of feature point matching filter-combined SIFT provide trajectory solutions with a denser distribution and therefore ensure more reliable estimation result.

Further result follows for the performance demonstration. Figures 13 and 14 visualize the estimated trajectories of INS/vSLAM for Tests A-2 and C-1 among the results. As

TABLE 3. Mean error and standard deviation by test set

		INS-only	KLT	GMF-SIFT
Test A-1	Mean error (m)	0.0861	0.0536	0.0340
	Standard deviation	0.1301	0.0672	0.0417
Test A-2	Mean error (m)	0.1459	0.0271	0.0234
	Standard deviation	0.1985	0.036	0.0253
Test B-1	Mean error (m)	0.1299	0.0276	0.0157
	Standard deviation	0.1619	0.0325	0.0182
Test B-2	Mean error (m)	0.2359	0.0295	0.0165
	Standard deviation	0.3070	0.0371	0.0185
Test C-1	Mean error (m)	0.3289	0.0230	0.0208
	Standard deviation	0.3739	0.0281	0.0246
Test C-2	Mean error (m)	0.4439	0.0528	0.0261
	Standard deviation	0.4457	0.0596	0.0295

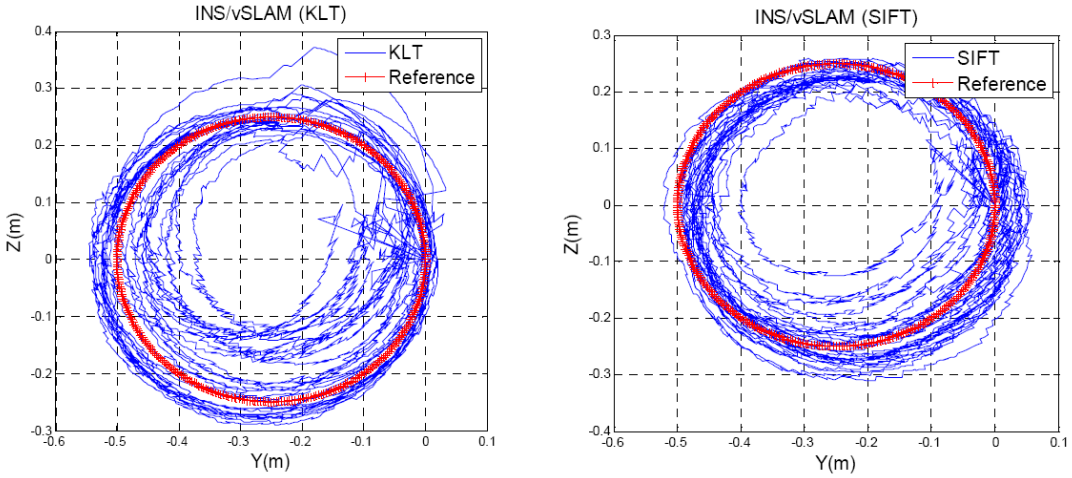

FIGURE 13. Estimated trajectory of INS/vSLAM – Test A-2 (KLT vs. GMF-SIFT)

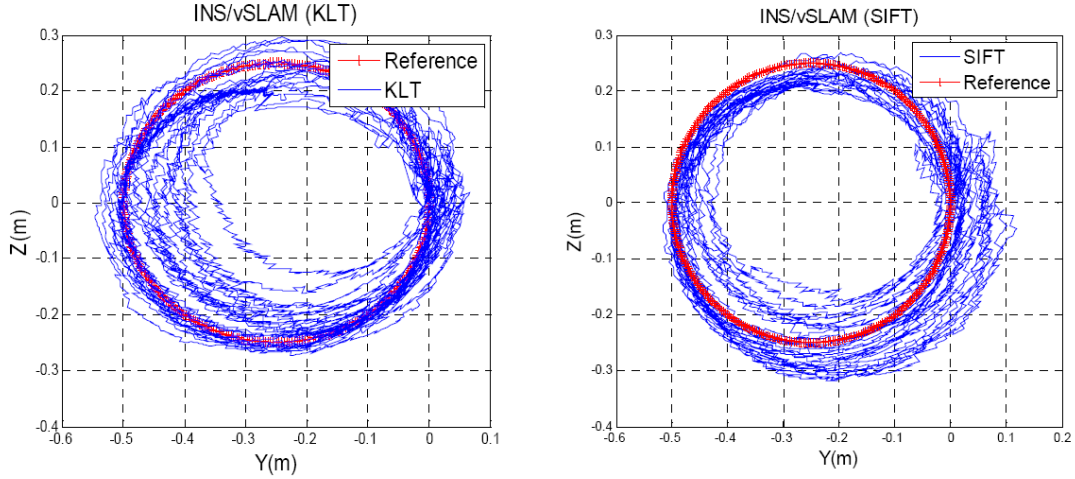

FIGURE 14. Estimated trajectory of INS/vSLAM – Test C-1 (KLT vs. GMF-SIFT)

shown in the figures, the KLT trajectory is, overall, widely stretched, centering on the cross-shaped reference trajectory, while that of matching filter-combined GMF-SIFT is distributed along the reference trajectory with a relatively narrow width. The estimated trajectory of the suggested feature point matching filter-combined SIFT demonstrates the improved data repeatability, implying better precision than KLT's estimation results.

Meanwhile, the estimated trajectory's starting points are distributed differently according to the initial posture of the aircraft (i.e., image and IMU module). Also, the performance of trajectory estimation is significantly undermined by such factors as change in the number of feature points – caused by the dynamic generation and loss of feature points observed during rotating movements – and the jump of feature points. Besides, the bias in estimated values, induced by misalignment between the optical axis of the image sensor and the coordinate axis of the IMU, is also considered to be a non-negligible defect, and thus the performance of relative navigation could be further enhanced if a more accurate initial alignment algorithm among sensors were used.

Finally, Table 4 compares the performance of INS-only estimation, KLT integrated filter, and the proposed GMF-SIFT based integrated filter. For the ease of comparison, the

TABLE 4. Comparison of relative errors from estimation results

		INS-only	KLT	GMF-SIFT
Test A-1	Multiplication	2.53	1.57	1
Test A-2		6.23	1.15	1
Test B-1		8.27	1.75	1
Test B-2		14.29	1.78	1
Test C-1		15.81	1.10	1
Test C-2		17.00	2.02	1
Average		10.68	1.56	1

mean error of the proposed method derived in Table 3 is set as 1, while each corresponding error multiplications from the INS-only and KLT based integration filter methods are computed on this basis. In the comparison of error multiplications, INS-only navigation shows roughly 10 times greater estimated error multiplication than the modified SIFT while the error multiplication of KLT-based motion estimation is around 56% greater than that of the modified SIFT during 3 times rotation.

5. **Conclusion.** This paper has suggested an integrated filter system that measures and estimates a relative motion of mobile vehicles through a newly developed feature point tracking algorithm. The main idea of the proposed tracking method is to incorporate a geometric variation criterion for candidate feature point matching process. In addition to the matching concept, the algorithm and hardware implementation on a real time platform is presented. Based on this algorithm, an inertial SLAM filter system that can dynamically estimate the motion of the moving vehicles is developed. For the experiment, a low- to mid-end IMU and a single vision sensor have been used to realize the test bed. Using the feature point matching filter-combined SIFT and the KLT algorithm as the image tracking function, three experimental environments have been created to carry out repeated tests and analyze the system's performance based on the estimated trajectory solutions. The test and analysis results show that the proposed tracking algorithm and the KLT-based INS/vSLAM outperform IMU-only estimation result. Moreover, the newly proposed, enhanced GMF-SIFT estimation demonstrates relatively better performance in terms of robustness and reliability.

Acknowledgment. This work was supported by the faculty research fund of Konkuk University in 2010.

REFERENCES

[1] J. Jung, Extraction of pupil area using scale invariant feature transform, *Proc. of Information Technology Institute*, vol.14, no.1, pp.81-85, 2006.
[2] S. Chun, D. Won, Y. J. Lee, T. Kang and S. Sung, Precise estimation of relative altitude of aircraft using vision sensor, *Proc. of KSAS Fall Conference*, pp.1252-1255, 2007.
[3] D. Won, *DR Navigation Performance Improvement for Mobile Robot Using Vision Based SLAM Technology*, Master Thesis, Konkuk University, 2008.
[4] S. Yun, *Implementation of INS/vSLAM Integrated System and Relative Navigation Experiment*, Master Thesis, Konkuk University, 2009.
[5] S. Chun, *INS/GPS Performance Improvement Using Multiple Vision Sensor*, Ph.D Thesis, Konkuk University, 2008.
[6] M. Gwak, S. Sung, S. Yun, D. Won and Y. J. Lee, Relative position estimation via modified scale invariant feature transform, *Trans. on KSAS*, vol.37, no.8, pp.759-766, 2009.
[7] A. Lawrence, *Modern Inertial Technology*, 2nd Edition, Springer, 1998.
[8] A. B. Chatfield, *Fundamentals of High Accuracy Inertial Navigation*, AIAA Press, 1997.

[9] D. J. Biezad, *Intergrated Navigation and Guidance Systems*, AIAA Press, 1999.
[10] D. H. Titterton, *Strapdown Inertial Navigation Technology*, Peter Peregrinus Ltd., 1997.
[11] M. Kayton, *Avionics Navigation System*, 2nd Edition, John Wiley & Sons, 1969.
[12] J. Farrell, *The Global Positioning System and Inertial Navigation*, Mc-Graw-Hill, 1999.
[13] O. Salychev, *Inertial Systems in Navigation and Geophysics*, Bauman MSTU Press, 1998.
[14] J. L. Bentley, K-d trees for semidynamic point sets, *Proc. of the 6th Annual Symposium on Computational Geometry*, pp.187-197, 1990.
[15] D. G. Lowe, Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision*, vol.60, no.2, pp.91-110, 2004.
[16] F. M. Mizaei and S. I. Roumeliotis, A kalman filter-based algorithm for IMU-camera calibration: Observability analysis and performance evaluation, *IEEE Transactions on Robotics*, vol.24, no.5, pp.1143-1156, 2008.
[17] K. Mikolajczyk and C. Schmid, Scale & affine invariant interest point detectors, *International Journal of Computer Vision*, vol.60, no.1, pp.63-86, 2004.
[18] K. Mikolajczyk and C. Schmid, A performance evaluation of local descriptors, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.27, no.10, pp.1615-1630, 2005.
[19] X. Wang and H. Zhang, Good image features for bearing-only SLAM, *Proc. of the 2006 IEEE/RSJ, International Conference on Intelligent Robots and Systems*, Beijing, China, pp.2576-2581, 2006.
[20] J. W. Langelaan, *State Estimation for Autonomous Flight in Cluttered Environments*, Ph.D. Thesis, Stanford University, 2007.
[21] J. Kim, *Autonomous Navigation for Airborne Application*, Ph.D. Thesis, The University of Sydney, 2004.
[22] C. Tomasi and T. Kanade, Detection and tracking of point features, *Technical Report CMU-CS-91-132*, Carnegie Mellon University, 1991.
[23] S. Zhang, F. Gao and L. Kong, A self-occlusion detection approach based on range image of vision object, *ICIC Express Letters*, vol.5, no.6, pp.2041-2046, 2011.
[24] X. Wu, Y. Wang and Z. Zhu, Nonlinear filtering approaches based pose, motion and variable focal length estimation in monocular active vision, *ICIC Express Letters*, vol.5, no.3, pp.741-745, 2011.
[25] S. Wan, Triangular fuzzy number method for robot non-vision multi-sensor fusion, *ICIC Express Letters*, vol.4, no.3(B), pp.857-862, 2010.
[26] *http://www.ces.clemson.edu/∼stb/klt/.*
[27] *http://web.engr.oregonstate.edu/∼hess/.*