THESIS APPROVAL

GRADUATE SCHOOL, KASETSART UNIVERSITY

Master of Engineering (Information and Communication Technology for Embedded Systems)
DEGREE

Information and Communication Technology for Embedded | Electrical Engineering
FIELD | **DEPARTMENT**

TITLE: A Vehicle Tracking Algorithm by Fusing Video and GPS Data

NAME: Ms. Kanittha Sae-Lim

THIS THESIS HAS BEEN ACCEPTED BY

(Assistant Professor Teerasit Kasetkasem, Ph.D.)
THESIS ADVISOR

(Mr. Supakorn Siddhichai, Ph.D.)
THESIS CO-ADVISOR

(Ms. Thitiporn Chanwimaluang, Ph.D.)
THESIS CO-ADVISOR

(Associate Professor Tsuyoshi Isshiki, Ph.D.)
THESIS CO-ADVISOR

(Assistant Professor Teerasit Kasetkasem, Ph.D.)
DEPARTMENT HEAD

APPROVED BY THE GRADUATE SCHOOL ON _____

(Associate Professor Gunjana Theeragool, D.Agr.)
DEAN

THESIS

A VEHICLE TRACKING ALGORITHM BY FUSING VIDEO AND GPS DATA

KANITTHA SAE-LIM

A Thesis Submitted in Partial Fulfillment of
the Requirements for the Degree of
Master of Engineering (Information and Communication Technology for Embedded Systems)
Graduate School, Kasetsart University
2011

Kanittha Sae-Lim 2011: A Vehicle Tracking Algorithm by Fusing Video and GPS Data. Master of Engineering (Information and Communication Technology for Embedded Systems), Major Field: Information and Communication Technology for Embedded Systems, Department of Electrical Engineering. Thesis Advisor: Assistant Professor Teerasit Kasetkasem, Ph.D. 53 pages

Accuracy of positioning system is important for many applications ranging from the automatic navigation in the city through the precision agriculture in the field. A regular GPS receiver has a low cost but with a low accuracy (2 to 30 meters). As a result, we propose a low cost system where the positioning data from GPS is fused with the images obtained by the mounted cameras on a moving vehicle. Our method uses the particle filter as the tracking algorithm and improves the accuracy based on epipolar geometry, which represents the relationship of two corresponding images from video sequence. Local feature of the corresponding points among these views are extracted and matched by SIFT algorithm. Associated with the particle motions, these match points are reconstructed in 3D and then reprojected back to measure errors for weighting. The experiment demonstrates all optimum parameter values for each kind of tracking.

_____ _____ ___ / ___ / ___

Student's signature Thesis Advisor's signature

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to my advisor, Dr. Teerasit Kasetkasem from Kasetsart University, for devoted time and for all indescribable assistances until I could publish this paper. As well as Dr. Supakorn and Dr. Thitiporn, my co-adviser from National Electronics and Computer Technology Center, I am grateful for the great guidance and other valuable suggestions for this research. Moreover I also give thank for a new point of view from Prof. Tsuyoshi Isshiki from Tokyo Institute of Technology.

I am thankful all of my friends, especially, Ms. Sayumporn and Mr. Pongthorn, who fulfill other parts of this project "The automatic vehicle system" and whom I can refer to whenever is needed or necessary for the extempore situation. Here, I am very appreciative to Ms. Pornprawee, our official, as well for all of kindnesses to answer questions and remind me to complete each work in time.

Finally, I wish to thank my parents, my sisters, and Mr. Surat for all the support and encouragement.

This research was supported in part by Center for Promoting Research and Development of Satellite Image Applications in Agriculture and Thailand Advanced Institute of Science and Technology (TAIST), National Science and Technology Development Agency (NSTDA), Tokyo Institute of Technology and Kasetsart University for financial support.

Kanittha Sae-Lim

May 2011

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF ABBREVIATIONS

GPS	=	Global Positioning System
PF	=	Particle Filter
RTK-GPS	=	Real Time Kinematic GPS
SIFT	=	Scale Invariant Feature Transform
SSD	=	Sum of Square Difference
UTM	=	Universal Transverse Mercator

A VEHICLE TRACKING ALGORITHM BY FUSING VIDEO AND GPS DATA

INTRODUCTION

As an agricultural country, the majority of Thai population earns their living through the harvestmen of agriculture products, especially crops. The National research council of Thailand specifies rice, rubber, sugarcane, maize and cassava as the main crops of Thailand. However, the demands in the market do not always conform to the volume of crop productions as repeatedly occurred every year. This unconformity results in the leftover of the productions and downfall of the crop prices. Moreover, since the agriculture products are perishable, the government cannot always build the silos to store the leftover to be sold during the under-supply periods. Therefore, the information on crop productions in this country is vital for the living quality of Thai farmers. If the government can estimate the yield of crop productions in each area, additional the oversee markets can be prepared in advance during the overproduction periods.

As a result, a crop yield monitoring system is essential for Thailand economy. This system consists of the remote sensing images to obtain the synoptic images, the yield models of major crops to estimate the crop productions, and the ground survey systems to assist and assess the estimation model. Remote sensing images with the synoptic views provide land use and land cover maps that enable the experts to estimate the crop areas, stage of growth, and length of time to the harvestmen. Due to the large amount of data in remote sensing images, automatic classification and estimation algorithms are employed in creating the land use and land cover maps where the ground survey is very essential since it provides the ground data to train the mapping algorithms. A ground survey system is also important to the yield model development since local information obtained in the survey data can be used to assess the model accuracy and selection of model parameters. From the previous statement,

it is clear that the ground survey system is an import part of the crop yield monitoring system.

Currently, a ground survey carried out in Thailand involves a number of experts to go out into the field of study with the survey equipment by which the true position is determined. After obtaining the position, the experts decide to declare the respective location with one of the predefined labels. Even though this technique yields very accurate mapping information, it is very slow and can only accommodate limited area. As a result, the automatic survey system, where a camera is mounted on a GPS-equipped vehicle, can survey a large area when comparing the existing ground survey techniques. The automatic survey system obtains its location from GPS and decides to map each location with one of the predefined labels as the regular ground survey system. Since the images are taken near the objects, the texture information can be used for classification. Pharsook (Pharsook *et al*., 2011) has used the texture information for classification and obtained over ninety percentages of accuracy. The automatic ground survey system has one major limitation, i.e., the positioning accuracy is limited by the accuracy of the GPS receiver equipped on the vehicle which may not be sufficient in very detail maps. As a result, this research would focus on gaining precise positions accuracy of GPS data by fusing with images taken by video camera.

Even though we acquire positions on the earth easily from GPS device, there are such factors that reduce performance of this system. The most significant one such as atmospheric conditions, ephemeris (orbit position) errors, clock drift, measurement noise and signal multipath these always cause a limited positioning accuracy and availability in a general GPS device or low cost GPS device. However, there are still such a high precision positioning devices. Real Time Kinematic GPS (RTK-GPS) is the one has accuracy in centimeter but with a high cost and using high frequency signal to transmit between GPS devices and reference devices. This kind of transmission requires the permission from The National Broadcasting and Telecommunications Commission (NBTC) in Thailand which may not be convenient to be obtained.

Our method mount video camera and GPS receiver on a car to record video scene and GPS data while the car is wandering around the observed location. Since video camera takes a sequence of images at a high rate (30 images in one second for standard camera). Nearby images in this image sequences have many overlapping regions and, therefore, the relative camera positions can be determined through the shifts of the corresponding overlapping regions. However, the amount of image data in one video scene even for a short recording time, say 10 mins, can easily overwhelm a positioning algorithm. To reduce the computational complexity, we employ the scale invariant feature transform (SIFT) algorithm to extract corresponding pixel pairs between images. Since regions of hundreds thousands pixels are reduced to a group of hundreds pixels, the amount of computation involved can be greatly reduced. Unfortunately, the SIFT algorithm can produce false pixel pairs and the false pixel pairs can significantly lower the positioning accuracy of the proposed algorithm. We develop the pixel pair validation and adjustment algorithm. Here, each point match is verified by measuring its similarity and shifted-distance. Firstly, sum of square difference (SSD) is used to measure the similarity. The real match should have a small difference. To obtain well-localized pixel as accurate as possible, pixels around the corresponding points will be measured their similarities as well. The window that is most similar will be defined the perfect match. On the other hand if all pixels are still different, this match will be defined the false match. The shifted-distance is pixel movement between the pixel in the first image and the second image. Each match should have shifted-distance conform to others. However if it is not, the probability to be mismatch is high so it will be ultimately discarded.

After the validation and adjustment algorithm, the remaining pixel pairs are used to determine the relative positions of cameras though the epipolar geometry. Here, given a set of corresponding pixel pairs between images and the positions of camera, the real world coordinates of these pixel pairs can be estimated through the use of a triangulation technique. Next, we project the real world coordinates back to the pixel locations in the image pair. Under the idea condition, the reprojected and original pixel locations should be exactly the same. In other words, there is no error between reprojected and original pixel locations. However, if the locations and

orientation of the camera given to the reprojection process is incorrect, the reprojected and original pixel locations are not perfectly aligned. The main reason that we do not work on the real world coordinates is due to the fact that the real world coordinate can only be determined only up to a scale factor. Since errors between the reprojected and original pixel locations depends on the correction of the estimated camera positions, the errors can be used to further refine the positioning information obtained from the mounted GPS on the vehicle. In this research, the particle filtering algorithm is employed as the localization algorithm where GPS and captured image are the inputs. Note here that there are many localization algorithms that can be used for determining location of the vehicles. However, we choose the particle filter algorithm due to its simplicity. Other algorithms may be more efficient in term of possibility accuracy and computational complexity. However, the goal of this research is to investigate the use of video images to improve positioning accuracy, not to develop a new localization algorithm. The results show an outperformance compared to the positions from a general GPS and the positions from GPS that fused with original particle filter.

OBJECTIVES

1. To improve the correctness of corresponding points by validation and refinement the positions of local features detected in image.

2. To increase the position accuracy of the GPS mounted vehicle by combining of the video data.

3. To determine the position of camera for image scene without GPS data.

LITERATURE REVIEW

In this chapter, previous and related works on the improvement of GPS accuracy through the use of auxiliary sensors are provided. Specially, the use of video data is thoroughly investigated. Here, we begin the chapter with the detailed information on the existing positioning systems, in particular the GPS. Then, we review the integrations of GPS with many sensors, e.g., RTK, range sensor and camera. As stated in the introduction, we increase the positioning accuracy through the use of video image sequences. Here, corresponding points among images are determined using SIFT algorithms and the relative positions of a camera are estimated through the epipolar geometry. For the sake of clarity, the details of SIFT algorithm and epipolar geometry are provided in separated sections. In the last section, the particle filters are thoroughly stated since it is employed as the tracking algorithm in our work.

1. Positioning System

The favorite global positioning system called GPS constructs from a constellation of 24 satellites orbiting the earth. All GPS satellites are synchronized by a very accurate time reference provided by atomic clock on board the satellites. Therefore they can repeatedly transmit signals to a GPS receiver at the same instant. Each transmitted signal indicates its location and the current time. Once GPS receiver receives this information it will calculate the distance between itself and the positions of at least 3 satellites by comparing the time a signal was transmitted by a satellite with the time it was received. Next, with distance measurements, the receiver applies triangulation to determine a location on the earth surface. Although the GPS has been designed to be as nearly accurate as possible, there are still errors from several sources (Corvallis Microtechnology Inc., 2000) such as

- Atmospheric Conditions: The ionosphere and troposphere both refract the GPS signals. This causes the speed of the GPS signal in ionosphere and troposphere

differs from the speed in space. Therefore, the calculated distance ($signal \times time$) will be different.

- Ephemeris Errors/ Clock Drift/ Measurement Noise: The data concerning ephemeris (orbit position) errors may not exactly model the true satellite motion or the exact rate of clock drift. Distortion of the signal by measurement noise can further increase positional error.

- Selective Availability: This is the intentional alteration of time and ephemeris signal by the Department of Defense. SA can cause positional error up 0-70 meters however this error can be removed by differential correction.

- Multipath: The reflection of GPS signal on object mainly appears in the neighborhood of large building. The reflected signal takes more time to reach the receiver than the direct signal.

These errors bring to the implementation of many methods and devices. RTK-GPS or real time kinematic GPS is the one device which can position a moving vehicle within 30 cm. This is suitable for a high accuracy guidance system. Nevertheless, the cost of the RTK-GPS is higher than a regular GPS receiver and transmitted frequency range is not in the public frequency. Such inconvenience may limit the usage of this device so other researches attempt to develop and propose a novel algorithm instead.

A particle filter is a powerful method in tracking as proposed by Pupilli (Pupilli *et al*., 2005). They emphasize a real-time camera tracking. With minimal pre-calibration of structure and the ability to incorporate new structure of particle filter, these give the potential for a general purpose camera tracking. However the results are not robust enough due to an uncertainty of camera position when initializing new points. Therefore an integration of filtering method with different devices is more widely used in many applications for tracking, navigation and localization.

Several non-linear filter methods were evaluated by Flament (Flament *et al*., 2004) for terrain-aided navigation system together with additional sensors, which are accelerometer and gyro-meter. The results show that particle filter is very sensitive to terrain topology and requires a large number of particles. Meanwhile the unscented particle filter could help reducing the number of particles. But Gaussian-mixture filter gives an overall improvement. Another filtering method called extended Kalman filter (EKF) is applied to solve the unmanned aerial vehicles (UAVs) refueling problem (Mammarella *et al*., 2008). GPS data and a machine vision-based system are integrated in EKF to estimate the tanker-UAV position through the derivation of feature extraction, feature classification, and pose estimation algorithm. The results indicate that the accuracy is improved at least one order of magnitude. Levinson (Levinson *et al*., 2007) presented a localization method that utilizes an urban environment map. First of all, environment map is built using infrared LIDAR measuring the 3D structure and infrared reflectivity of the environment. Then simultaneous localization and mapping algorithm (SLAM) transform this data into a model with only retains the flat road surface. In localization, an inertial navigation system from GPS, IMU, and wheel odometer are integrated in a particle filter to determine vehicle velocity which project particles forward through the time. The results achieved localization in real-time with accuracy in the 10 centimeters but the localization may fail due to an unreliable map. Another localization method with different devices is proposed by Limsoonthrakul (Limsoonthrakul *et al*., 2009). This research integrates GPS, compass, odometry encoder and machine vision in particle filter. A wheel encoders and a steering wheel encoder are used to inform the traveled distance and the angle, respectively. Images from the view in front of vehicle are classified using a color histogram-based method to identify the road regions. The classification result is used as observation data in the localization. The experimental results demonstrate that this method significantly improves lateral localization error compared to a standard extended Kalman filter which does not use image data.

Since we want to apply an image sequence in our approach, camera pose estimation is also considered. To estimate camera position from the capture images, epipolar geometry in stereo vision from Hartley (Hartley *et al*., 2003) is the favorite

one represented the relation between the 3D points and their projections onto the 2D images. Many researches applied this geometry (Kosecka *et al*., 2004; Wang *et al*., 2004; Hakeem *et al*., 2006) by detecting corresponding points between image pair first and then compute essential matrix to recover rotation and translation of the camera. But due to the reconstructions based on essential matrix have a scale ambiguity which leads to the disadvantage of this approach, other researches try to match the captured images to the reference landmarks to solve this problem.

Based on epipolar geometry the desirable on well-localized features of image is emphasized in particular. Due to an evaluation in local descriptor performance proposed by Mikolajczyk (Mikolajczyk *et al*., 2005), they observed that the SIFT-based descriptor perform the best to be distinctive and robust to viewing change in order to extract features. Here we take an interest in this algorithm to be the one parameter in an observation.

2. SIFT Algorithm

SIFT algorithm proposed by Lowe (Lowe, 2004) use to detect, extract, and match feature points. These feature points are invariant to image scaling, rotation, illumination change, and 3D camera view point so they provide a reliable matching between different views of an object or scene. There are four major stages of computation used to generate the set of image features as follow:

2.1.1 Scale-space extrema detection

The first stage of computation searches over all scales and image locations. It is implemented efficiently by using a difference-of-Gaussian function to identify potential interest points that are invariant to scale and orientation.

2.1.2 Keypoint localization

At each candidate location, a detailed model is fit to determine location and scale. Keypoints are selected based on measures of their stability.

2.1.3 Orientation assignment

One or more orientations are assigned to each keypoint location based on local image gradient directions. All future operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations.

2.1.4 Keypoint descriptor

The local image gradients are measured at the selected scale in the region around each keypoint. These are transformed into a representation that allows for significant levels of local shape distortion and change in illumination.

Interesting points for SIFT features correspond to local extrema of Difference-of-Gaussian or DoG at different scales. Given a Gaussian-blurred image

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y) \tag{1}$$

where

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{2}$$

The result of convolving an image with a difference-of-Gaussian is given by

$$D(x,y,\sigma) = \left(G(x,y,k\sigma) - G(x,y,\sigma)\right) * I(x,y) \tag{3}$$

$$= L(x,y,k\sigma) - L(x,y,\sigma) \tag{4}$$

which is the difference of the adjacent Gaussian-blurred images at scale σ and $k\sigma$ as shown in Figure 1. There are a number of reasons for choosing this function. First, it is a particularly efficient function to compare, as the smoothed images, L, need to be computed in any case for scale space feature description, and D can therefore be computed by simple image subtraction. In addition, the difference-of-Gaussian function provide a close approximation to the scale-normalized Laplacian of Gaussian, $\sigma^2\nabla^2 G$. The maxima and minima of $\sigma^2\nabla^2 G$ produce the most stable image features compared to a range of other possible image function.

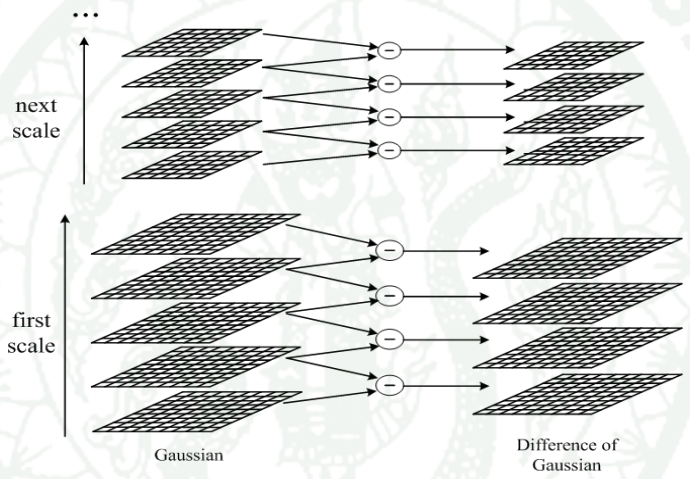


Figure 1 Adjacent Gaussian-blurred images are subtracted to produce the difference-of-Gaussian images.

Source: Lowe (2004)

The convolved images are grouped by octave (an octave corresponds to doubling the value of σ), and the value of k is selected so that we obtain a fixed number of blurred images per octave. This also ensures that we obtain the same number of difference-of-Gaussian images per octave.

In order to detect interesting points or keypoints, each pixel in the DoG images is compared to its eight neighbors at the same scale and nine neighbors in the scale above and below (see in Figure 2).

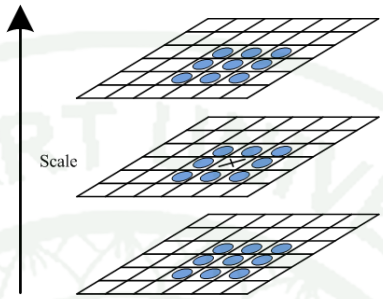


Figure 2 Maxima and minima of the difference-of-Gaussian images are detected by comparing their pixel neighbors.

Source: Lowe (2004)

For each candidate keypoint:

- Interpolation of nearby data is used to accurately determine its position.
- Keypoints with low contrast are removed.
- Responses along edges are eliminated.
- The keypoint is assigned an orientation.

To determine the keypoint orientation, a gradient orientation histogram is computed in the neighborhood of the keypoint. The gradient magnitude, $m(x, y)$, and orientation, $\theta(x, y)$, is precomputed using pixel differences:

$$m(x,y) = \sqrt{\left(L(x+1,y) - L(x-1,y)\right)^2 + \left(L(x,y+1) - L(x,y-1)\right)^2} \quad (5)$$

$$\theta(x,y) = \tan^{-1}\frac{\left(L(x,y+1) - L(x,y-1)\right)}{\left(L(x+1,y) - L(x-1,y)\right)} \quad (6)$$

The contribution of each neighboring pixel is weighted by the gradient magnitude and a Gaussian window with a σ that is 1.5 times the scale of the keypoint. Peak in the histogram correspond to dominant orientations of local gradients. The highest peak in the histogram is detected, and then any other local peak that is within 80% of the highest peak is used to also create a keypoint with that orientation. All the properties of the keypoint are measured relative to the keypoint orientation, this provides invariance to rotation.

Once a keypoint orientation has been selected, a keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the left of Figure 3. These are weighted by a Gaussian window, indicated by the overlaid circle. These samples are then accumulated into orientation histograms summarizing the contents over 4×4 sub regions, as shown on the right of Figure 3.

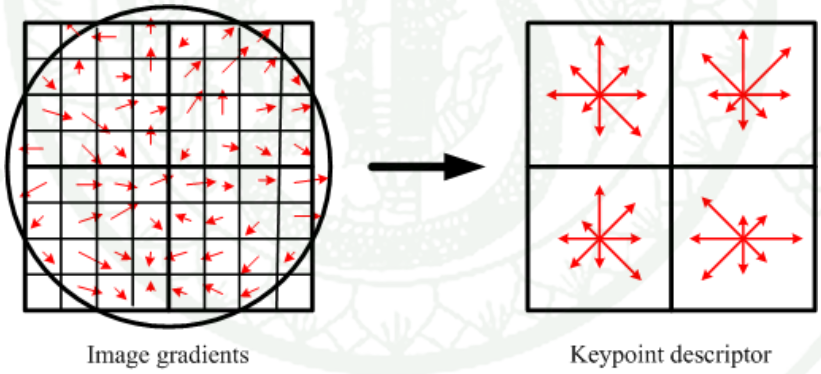


Figure 3 A keypoint descriptor is created by computing the gradient magnitude and orientation.

Source: Lowe (2004)

Histograms contain 8 bins, and each descriptor contains an array of 4×4 histograms around the keypoint. This leads to SIFT feature vector with 4×4×8 = 128 elements. This vector is normalized to enhance invariance to changes in illumination.

The match of these features is accepted if the ratio between the distance to the first nearest descriptor and the distance to the second nearest descriptor is lower than a given threshold.

3. Epipolar Geometry

Geometry between two views is a key point in computer vision to compute essential matrix for three-dimensional camera motion.

Suppose a point P in 3-space is mapped in two views, at p and p', through the camera centres, C and C' as shown in Figure 4. The ray from p through camera centre and the line joining the two camera centre define a plane called epipolar plane. Each epipolar plane intersects with two image planes along the epipolar lines. All the points in an epipolar plane are projected onto epipolar line in the first image and onto the corresponding epipolar line in the second image. This condition/constrain leads to the relation of the corresponding point between two views which satisfy the equation

$$p'^{T} F p = 0 \tag{7}$$

where F is the fundamental matrix relating to intrinsic matrix as

$$E = K_2^T F K_1 \tag{8}$$

Here K_i is the intrinsic matrix or internal camera parameter of the i-th camera defined by

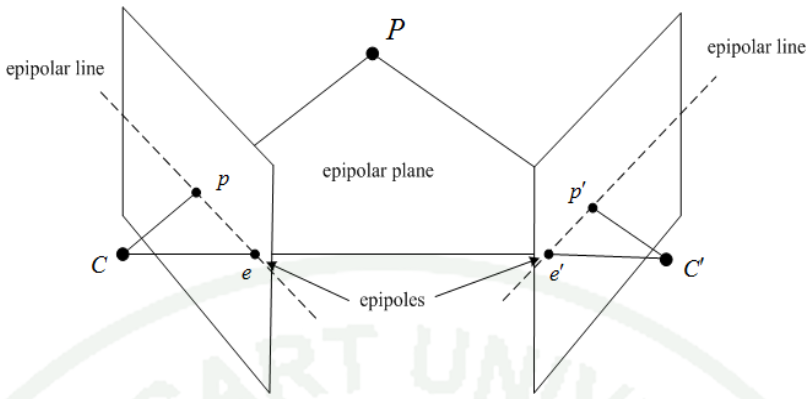


Figure 4 Epipolar Geometry

Source: Hartley *et al*. (2003)

If we consider camera model which map the 3D world points P to image points p via a projective matrix Q. The corresponding expression in homogeneous coordinate can be written as

$$p = QP \tag{9}$$

This projective matrix or camera matrix could be decomposed as

$$Q = KR[I \mid -C] \tag{10}$$

where R and C represent the camera orientation and the coordinate of camera center in the world coordinate which relate to the translation parameter $\mathbf{t}$ as

$$\mathbf{t} = -RC \tag{11}$$

Hence the relative rotation matrix R and translation $\mathbf{t}$ between two views from one camera to another can be expressed as

$$Q = K[R \mid \mathbf{t}] \tag{12}$$

This relation, which is from the triangulation of epipolar geometry, could help to inform the direction of motion in unit vector. This information is useful for the tracking. Here, we prefer the simplicity of particle filtering method as the tracking algorithm which will be described in the next section.

4. Particle Filter

Particle filter is based on the Bayesian filter theory. The main objective of the particle filter is to approximate the *posteriori* probability of the system state given the observed data. In general, the particle filters are implemented in tracking problems (Khoomboon *et al*., 2010) where it uses process model to predict the prior distribution, and then updates that distribution by incorporating new observations to get posterior distribution. At the end, the list of the *posteriori* distributions of the system states is obtained over time.

Let $P(X_0) = P(X_0 \mid Y_0)$ be an initial prior probability of system states where Y_0 is the initial observation. Furthermore, since the system is dynamic, the system states are changed according to some transition probability, $P(X_k \mid X_{k-1})$, i.e., hence, the system states are modeled as the Markov chain. Here, the notation *k* denotes the time that the system being observed. The *posteriori* probability of a system state at time *k* given all the previous observation can be written as

$$P(X_k \mid Y_{1:k-1}) = \int P(X_k \mid X_{k-1}) P(X_{k-1} \mid Y_{1:k-1}) dX_{k-1} \tag{13}$$

The subscript $1:k-1$ implies that the set of data from the time 1 to $k-1$. Next, the k-th observation can incorporated into the *posteriori* probability by using the Bayes' rule as

$$P(X_k \mid Y_{1:k}) = \frac{P(Y_k \mid X_k) P(X_k \mid Y_{1:k-1})}{P(Y_k \mid Y_{1:k-1})} \tag{14}$$

Since the conditional probability $P(Y_k \mid Y_{1:k-1})$ is independent of a choice of X_k, it can be viewed as a normalizing constant. Hence, the equation (14) can be rewritten as

$$P(X_k \mid Y_{1:k}) = aP(Y_k \mid X_k) P(X_k \mid Y_{1:k-1}) \tag{15}$$

where *a* is the normalizing constant that makes the sum of probability equal to one. Furthermore, the conditional probability $P(Y_k \mid X_k)$ is obtained from the observation model of the problem of interest.

The particle filter tries to approximate the *posteriori* probability given in (15) by representing *N* possible system states by a position of *N* particles. In our problem, the position of a particle can be considered as one image transformation. Here, each particle is assigned a weight function according to its *posteriori* probability. Let w_k^i denote a weight function of the *i*-th particle at a time *k*. The integral in (13) can be approximated as

$$P\left(X_k^i \middle| Y_{1:k-1}\right) \approx \sum_{i=1}^{N} P\left(X_k^i \middle| X_{k-1}^i\right) w_{k-1}^i \tag{16}$$

where X_k^i denote the position (system state) of the *i*-th particle. Next, the weight at the next time can be obtained from (15). The particle filter recursively solve equation (15) and (16). The approximation can be accurate if a large number of particles are deployed. The obvious drawback of this approach is the computation burden of the algorithm. To reduce the computational complexity, the particle filter introduces resampling technique to re-locate some of the particles in the area of higher weight. Here, a new set of *N* particles is selected from the set of current particle with replacement. The probability that a particle is chosen is proportion to its weight. Next, a new set of particle are move according to the given system model.

MATERIALS AND METHODS

Materials

1. Computer
2. Logitech Webcam
3. NCS-NAVI Bluetooth GPS receiver
4. MATLAB Simulation software
5. Microsoft Excel software
6. Microsoft Visual Studio 2008

Methods

1. Studied area and experimental setup

Our studied areas based on the experimental setup are separated into two types. The first one is a large patio at the 6th floor Chuchat Kumphu Building, Engineering Faculty, Kasetsart University, Bangkhen Campus (see Figure 5(a)-(b)). This area is used for tracking simulation. Another type comes from the real survey consisting of several locations of agricultural field in Thailand, the examples are shown in Figure 5(c)-(d).

Here, the experiments were setup according to studied areas; the first experiment is for the tracking simulation and the second one is for the field survey. For tracking simulation, a camera was set on a tripod with 105 cm. height. Then move it in x-direction parallel to a plane together with capture image continuously with a known scale, here every 30 cm. For the real survey, a video camera was mounted on the side of vehicle and synchronized with a GPS receiver. While the vehicle was travelling around the considered agricultural area, an image sequences with their locations were recorded as an input of the system.

(a) (b)

(c) (d)

Figure 5 Studied area: (a)-(b) are the locations for the simulation tracking and (c)-(d) are the example of agricultural fields in the real survey.

2. Proposed approach

To estimate the true position of a moving object by combining the data from the GPS and the camera, here, the particle filter is employed as the tracking algorithm. First, the algorithm waits for the GPS receiver transmitting the first location data, $\mathbf{x}(0)$. Once the first location data is received, an image $I(0)$ is taken and N particles are generated at random locations around $\mathbf{x}(0)$. The velocity vectors of these particles are also randomly assigned to between the minimum and maximum values ($\mathbf{v}_{\min}$, $\mathbf{v}_{\max}$). Next, the algorithm moves these particles into new random positions with unknown forces. Hence, the acceleration is assumed to be randomly distributed between $-\mathbf{a}_{\max}$ and $\mathbf{a}_{\max}$. Furthermore, since we assume that the vehicle can move in

the angular motion as well, the particle changes its orientations with a random angular velocity, $\boldsymbol{\omega}$.

Next, the tracking system waits for a new GPS data from a GPS receiver. As soon as a new GPS data is obtained, the mounted camera takes a new image and sends it to the tracking system. Then, the tracking algorithm compares image from current and previous times by creating the corresponding key points between the image pairs as illustrate in Figure 4. Next, the keypoint position in 3D is determined using the method from Hartley (Hartley *et al.*, 2003). Because of noisy problem, the corresponding point does not satisfy the epipolar constrain. The reprojection of the estimated 3-space point $\hat{P}$ to the image plane at $\hat{p}_i$ and $\hat{p}'_i$ is used to measure errors as shown in Figure 6.

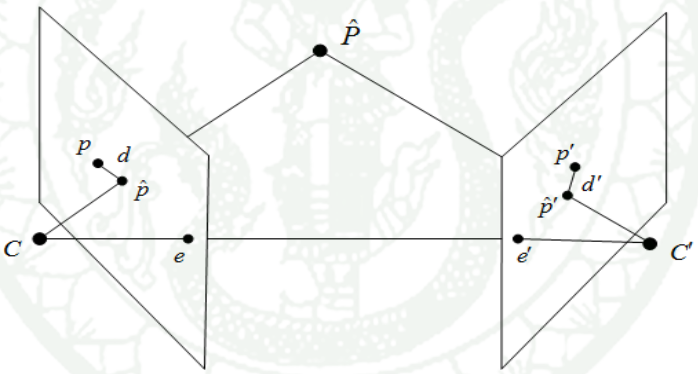


Figure 6 Error distance from the reprojection

Source: Hartley *et al.* (2003)

If the rotation and translation between these two images are correct, a back reprojection of the reconstructed point, $\hat{P}$ to image planes should exactly be the same points as appear on images. In other words, the distance between the corresponding point (p , p') and the reprojection point ($\hat{p}$, $\hat{p}'$) in each view should be minimized. Here, this error is measured using Euclidean distance.

$$E\left(p_i, p_i'\right) = d\left(p_i, \hat{p}_i\right)^2 + d\left(p_i', \hat{p}_i'\right)^2 \quad (17)$$

where $d(*,*)$ is the Euclidean distance between two points in each view and E is the measured error. Therefore, the weight function applies this attribute as the one parameter to identify the right positions of vehicle.

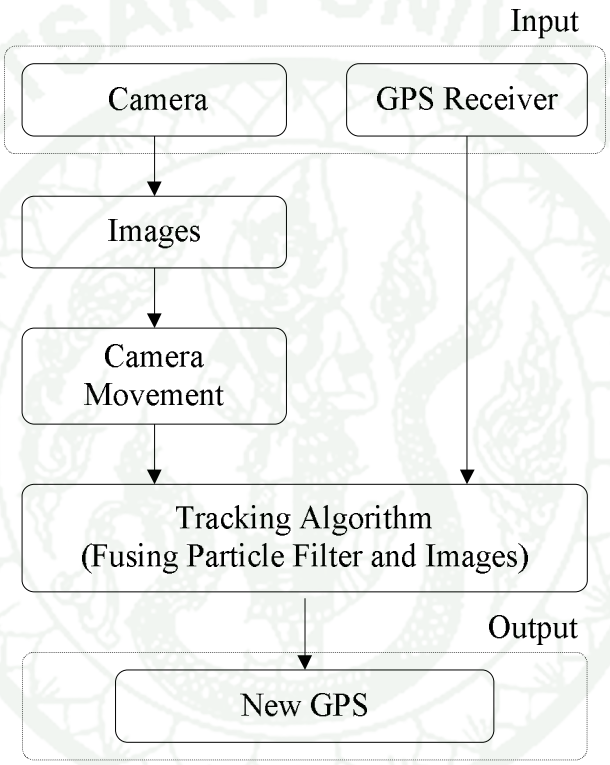


Figure 7 Flow chart of our proposed method

Our proposed algorithm concept can be concluded as a chart of Figure 7. First, we have images and GPS data from video camera and GPS receiver as inputs. The images are used to give direction information or give camera movement using their corresponding points of two adjacent frames. Then, this information is fused with GPS data in the tracking algorithm which employs particle filtering method for localization. After all processes are done, we obtain new GPS positions as the output. Since there are many important processes in the algorithm, hereafter each process will be discussed in details.

3. Accurate Keypoint Matching

From the attribution of epipolar geometry, correspondence between image pairs is required. Thus we choose scale invariant feature transform algorithm (SIFT) for this purpose and validate the correctness of each match as described below.

3.1 Keypoint Validation

Once we obtain the corresponding points between image pair, and then verify the consistency. We found that there are some errors which can be distinguished into two cases:

- Inaccurate in keypoint coordinates.
- False matching of the corresponding points.

These errors affect the quality and performance of many researches which are the important problem that researchers always concerned. There are 3 procedures to improve the correctness.

3.2 Refine Keypoint Coordinate

The correlation-based method which measures the similarity between windows in the two images is used in this section to fine-tune coordinate of the keypoint. Consider a region around keypoint in the first image and a corresponding region in the second image. The intensities in the neighborhood of each keypoint pair are compared to each other. The dissimilarity between two regions is measured using Sum of Square Difference (SSD) cost function which is given by

$$SSD = \frac{1}{(2N+1)^2} \sum_{i=-N}^{N} \sum_{j=-N}^{N} \left(I_1(x_1+i, y_1+j) - I_2(x_2+i, y_2+j) \right)^2 \qquad (18)$$

To match up the most similarity coordinate, the window will be searched around a region of its corresponding point (within 5-pixel radius). The coordinate that SSD is minimum, will have the maximum correlation. Therefore the old coordinate will be shifted to the new one for improvement the precision of localization each keypoint pairs.

3.3 Detect Dissimilarity

Even we gain the precision of keypoint coordinate from keypoint refinement. There are still some complete false matches. The importance of correlation is also used in this section to distinguish inlier and outlier correspondences. The higher SSD means the higher possibilities to be mismatch. So the coordinate that SSD over the defined threshold will be detected as the outliers. We define twice of the average SSD from all keypoints of each image pair as a threshold.

$$Threshold = 2 \times SSD \tag{19}$$

3.4 Filter High-Shifted Distance

Once the camera move, the pixel of the object that is close to the camera would be shifted farther than the pixel of the object that is far away. This causes the difference distance of the shifted pixel. The distance is either too far or too close intends to be mismatch so the shifted distance is the one parameter used to detect this type of error. We use a normal distribution with a range of two standard deviations above and below the mean, ($\bar{x} \pm 2\sigma$) define the perfect matches which are about 95% of the set of data and otherwise will be eliminated.

4. Localization

According to the availability of received GPS data, the localization based on our motion model is separated into two parts. The first part is a general tracking and the second part is a tracking when there are some GPS lost. The motion model is described in details first and follows with our proposed localization.

4.1 Motion Model

Let $\mathbf{y}(n)$ denote the vectors of position data in 2-dimensional space produced by a GPS where $n = 1,2,...,M$ is the acquisition number. Since the positioning system in GPS is erroneous, the position data can be written as

$$\mathbf{y}(n) = \mathbf{x}(n) + \mathbf{e}(n) \tag{20}$$

where $\mathbf{x}(n)$ and $\mathbf{e}(n)$ are the true location and the positioning error vectors at time *n*. When a GPS data is obtained the system issues the command the camera to capture an image of the surrounding environment. As a result, we denote the image taken at time *n* as $I\left(\mathbf{x}(n),\boldsymbol{\theta}(n)\right)$ where $\boldsymbol{\theta}(n)$ is the camera orientation. This notation emphasizes that the image is a function of the location and orientation of the camera at time *n*.

In this work, we assume that the time between two GPS messages are short enough so that $I\left(\mathbf{x}(n-1),\boldsymbol{\theta}(n-1)\right)$ and $I\left(\mathbf{x}(n),\boldsymbol{\theta}(n)\right)$ contain a large overlapping regions. Let $\wp_{n-1,n} = \{P_1, P_2, ..., P_M\}$ be the set of the corresponding key points between images taken at times $n-1$ and *n*. Based on the epipolar geometric function, the corresponding image points are given by

$$p_{m,n-1} = [I \mid 0] P_m \tag{21}$$

and

$$p_{m,n} = Q_{n-1,n} P_m \tag{22}$$

where $Q_{n-1,n} = \left[R\left(\boldsymbol{\theta}(n) - \boldsymbol{\theta}(n-1)\right) | \, \mathbf{x}(n) - \mathbf{x}(n-1) \right]$ is the projection matrix between images taken at time $n-1$ and n. Note here that the relative rotation and translation are functions of the differences in the orientation and position between time $n-1$ and n only.

The goal is of this work is to track the position of the moving vehicle equipped with the GPS and camera. As a result, let us assume that the motion equation model can be applied to track GPS from image sequences. Normally the traveled distance can be computed from $s = ut + \frac{1}{2}at^2$. Therefore, when the camera moves to the new position from the $\mathbf{x}(n-1)$ to $\mathbf{x}(n)$ via an initial velocity $\mathbf{v}(n-1)$ and the unknown acceleration $\mathbf{a}$ within time t, this relation can be expressed as

$$\mathbf{x}(n) = \mathbf{x}(n-1) + \mathbf{v}(n-1)\Delta t + \tfrac{1}{2}\mathbf{a}(\Delta t)^2 \tag{23}$$

where Δt is the time duration between two GPS messages. Next, the velocity $\mathbf{v}(n)$ at any position can be written as

$$\mathbf{v}\left(n\right) = \mathbf{v}\left(n-1\right) + \mathbf{a}\Delta t \tag{24}$$

Since the movement of camera includes both translation and rotation so the translation and the changed angle between two views are

$$\mathbf{t}\left(n\right) = \mathbf{x}\left(n\right) - \mathbf{x}\left(n-1\right) \tag{25}$$

$$\boldsymbol{\theta}\left(n\right) - \boldsymbol{\theta}\left(n-1\right) = \boldsymbol{\omega}\Delta t \tag{26}$$

where $\boldsymbol{\omega}$ is the unknown angular velocity of the moving vehicle. Once all parameters are obtained, the position could be tracked. In this research, we employ the particle filter algorithm to track the moving object.

4.2 Tracking Model

4.2.1 A General Tracking

Here, a general tracking means the tracking that have a complete set of GPS data and images. There is no GPS lost and all frames have their own positions. In this case, the weights are from the reprojection error and GPS error.

4.2.2 Lost GPS Tracking

For the incidence when GPS is lost. The proposed method can be separated into two solutions. Each solution has different information to inform the direction from images. All solutions are described below.

4.2.2.1 Recover GPS lost method I

This method recovers the lost position by calculating the reprojection error cost from each pair of corresponding images, which similar to a general tracking as described before. However when the middle frame has no GPS, in this case it utilizes the relationship of three corresponding frames instead; the first considered frame corresponds to the second frame, the second frame corresponds to the third frame, as well as the first frame should correspond to the third frame. As a result, the reprojection error costs from each pair of corresponding images are accumulated to form a new cost for weighting.

4.2.2.2 Recover GPS lost method II

This solution also uses the relationship of three corresponding images but instead to reconstruct 3D point from each image pair first and use the accumulate error cost, this method reconstructs the 3D points from three corresponding images first and then measure error cost.

RESULTS AND DISCUSSION

Results

The results were separated into two parts. The first part is the results from keypoint validation and refinement and the second part is the results from our tracking algorithm.

1. Results from keypoint validation and refinement approach

The experiment on this section emphasizes the accuracy of corresponding points. The SIFT algorithm was used to extract interesting features in image and then match these features between image pair as shown in Figure 8 and 9, respectively. These matches will be verified both their coordinates and their matches. Figure 10 shows the matches that intend to be inaccurate using SSD measure the similarity of each keypoint pair.

Therefore we move all points to new coordinates that are most similar to their matches or least SSD. The experiment on this approach used three different scenes of view and the results of new coordinate after refinement are the red points in the right hand of Figure 11. Here, there are 2 criterions to define the error match. First, the match that has the coordinate shifted over 5 pixels from the coordinate where it should be. Second, the match that is absolutely mismatched. Table 1 shows the numbers of the remaining error after each validation process. Even this method cannot remove all error, for the best result, error can be reduced up to 95.24%

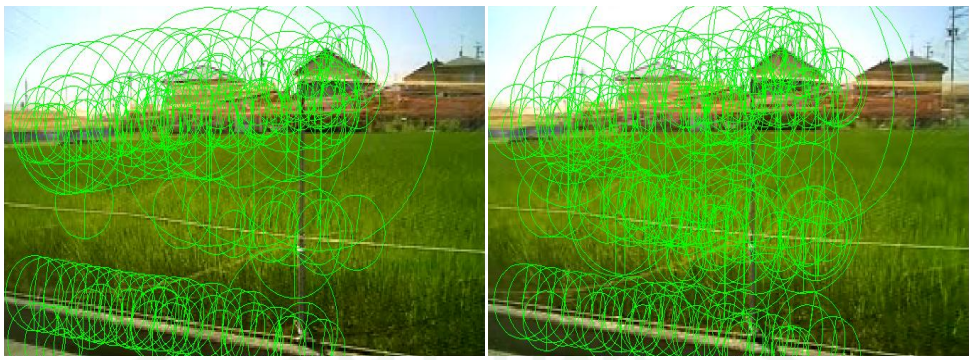


Figure 8 The interesting points extracted from SIFT algorithm.

Figure 9 The matches between two views represented by the blue line joining coordinates of corresponding points.

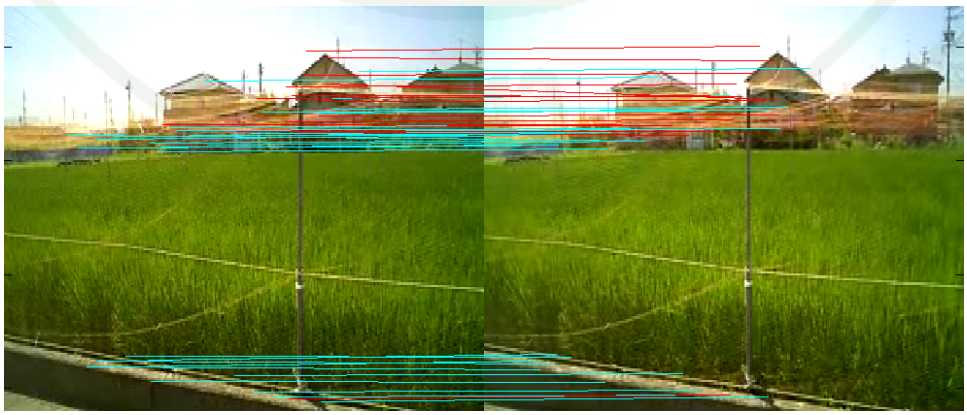


Figure 10 The red lines are the matches that intend to be inaccurate.

(a) Scene 1

(b) Scene 2

(c) Scene 3

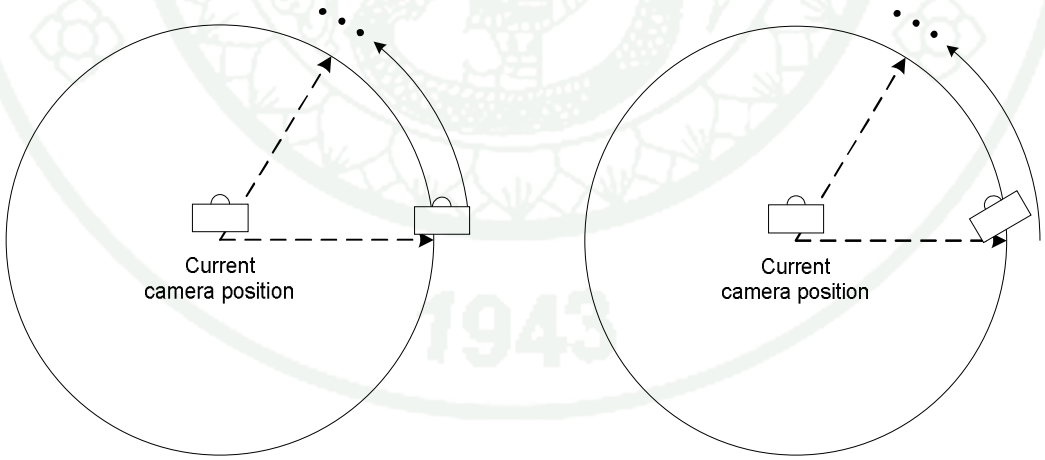
Figure 11 (a) – (c) the red points in the right hand side of these different scenes are the new coordinates after keypoint refinement.

Table 1 Matching errors that remain after the validation

Scene	Total matches	Error					
		Original		After: Fine-tune		After: Delete	
		Total	(%)	Total	(%)	Total	(%)
1	48	18	37.50	4	8.33	3	6.25
2	54	8	14.81	6	11.11	2	4.76
3	52	13	25.00	8	15.38	7	13.73

2. Results from the back reprojection verification

To verify the correctness of the reprojection error from the triangulation, assume that the movement from the center of a circle to any point on the circle represents the difference of camera center's translation position. Here, the different movements are expressed into two cases; only translation case and translation with rotation case as shown in Figure 12(a) and 12(b) respectively.



(a) Translation (b) Rotation and Translation

Figure 12 The difference of camera movement: (a) the camera moves with only translation and (b) the camera moves with both rotation and translation.

In the first case of only translation, the camera moves from the center and stop at the zero-degree position. The corresponding points from the captured images (as in Figure 13(a) and 13(b)) are used for measuring error cost from the reprojection. An error cost in Figure 14(a) is the result when the camera has a different translation position. The position that gives the minimum error cost is at zero-degree which conforms to the actual movement. As well as the case that the camera moves with both rotation and translation, Figure 12(b) shows the camera is translated to the same position as the previous case but its axis is rotated around 30 degree and Figure 13(c) and 13(d) shows the captured images. The result in Figure 14(b) also gives the minimum cost when the camera is in the right position related to the corresponding points.

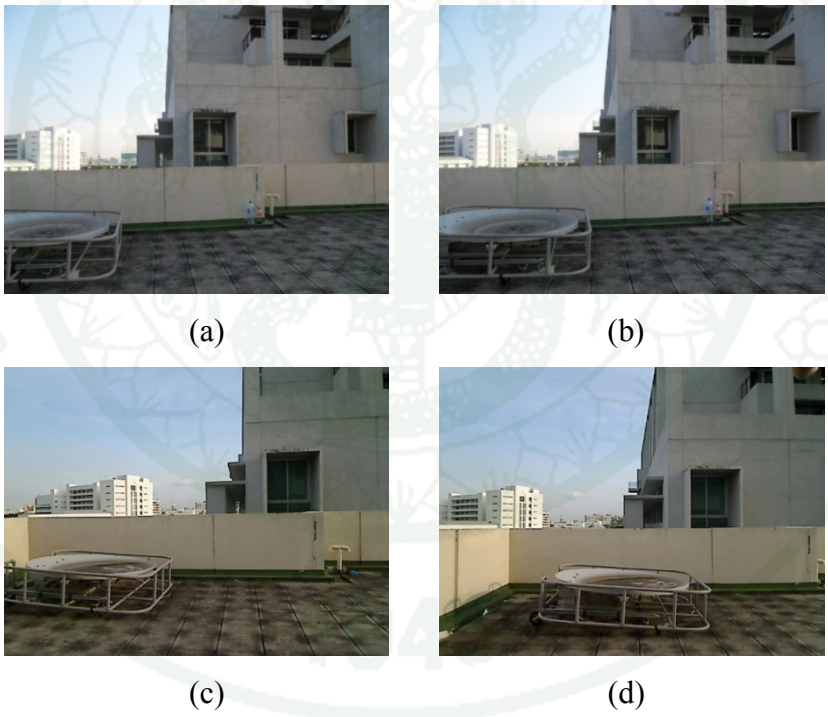
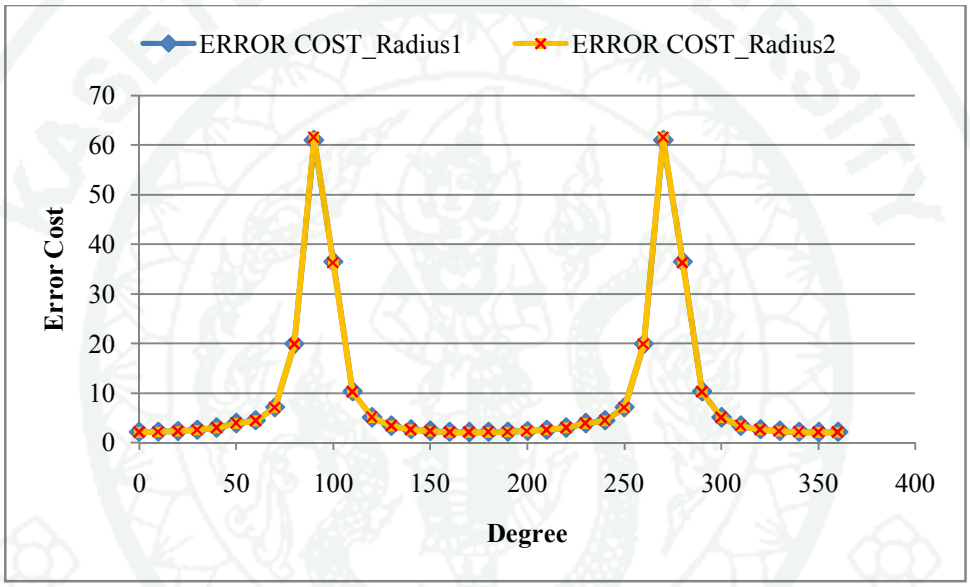


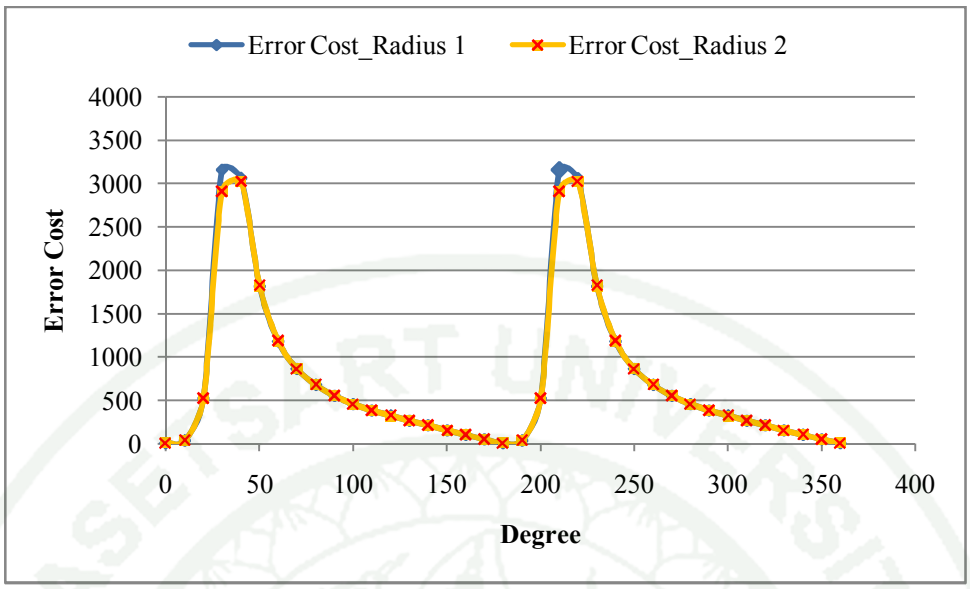
Figure 13 The captured images from different camera movement, the left hand are captured from the first location and the right hand are captured from the second location. (a)-(b) are from the first case with have only translation and (c)-(d) are from the second case when the angle axis is also changed.

Here, different scales of translation are demonstrated as well. The experiment has the process as previous except translation scale. The results in Figure 14(a) and 14(b) show that error cost from the reprojection is minimal when translation direction is correct. This implies that scaling does not affect the direction information. As a result, it is able to determine the camera movement using corresponding points from image pair although the camera is rotated and translated by any scale. If the movement direction is correct, the measured error must be minimal.



(a)

Figure 14 The reprojection error when the camera moves with different motions. (a) shows the error cost when camera is only translated to the new position. (b) shows the error cost when camera is rotated and translated to the new position.



(b)

Figure 14 (Continued)

3. The results from tracking algorithm

3.1 Simulation Tracking

3.1.1 A General Tracking

In the experiment we setup the camera perpendicular to its trajectory and simulated the motion by move the camera in the same direction parallel to the plane with a known scale as shown in Figure 15. Figure 16 is the example of the captured images which have equally distance, 30 centimeters. Here, we define 30 centimeters distance is 1 unit in the experiment. Based on our assumption, Gaussian noise was added to the true position to generate GPS that is usually inaccurate in positioning system. The different velocities include of a constant velocity and a velocity with acceleration, were experiment and discuss below.

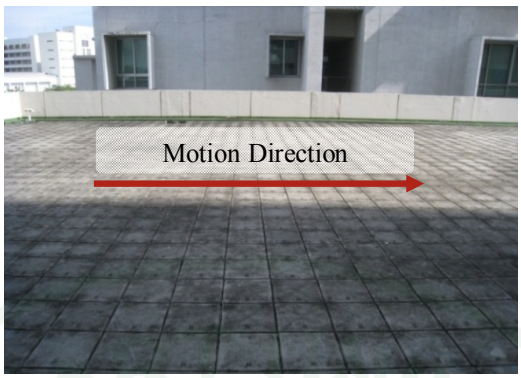


Figure 15 The direction of camera trajectory

Figure 16 The captured images for the simulation tracking

In the first experiment the camera moved with a constant velocity 30 cm/s. All parameter in the tracking model such as initial velocity, maximum acceleration, maximum angular velocity, numbers of particle, and weight parameter (sigma1, σ_1) are varied to obtain the optimum parameter value. The first parameter, an initial velocity, has its initial value in all 3 directional-axis (X, Y, Z). To start with 0 cm/s for all directions are less proper than start with 30 cm/s in x direction only but it is better than start with initial force in all direction, as shown in Table 2. The next parameter is the maximum acceleration used to define the range of the possible acceleration. Here, the acceleration was varied from 30 to 120 cm/s^2, the result in Table 3 show that 30 cm/s^2 is the most suitable for this motion. Another parameter is angular velocity. This also has the maximum range so we tried the maximum value 2, 10, and 20 degree/s with different initial angular velocity values. According to the direction and orientation from frame to frame change not much, 2 degree/s with initial

0 degree/s for all axis give the best result, as shown in Table 4. Next, numbers of particle, based on particle filter, the more particles are need for the more performance. Hence Table 5 did the comparison between 1000 and 5000 particles. The results are accordingly, 5000 particles give the better result. Finally, the last and most important one is the weight parameter, here, called sigma1. This is an additional parameter used for weighting in our algorithm. Table 6 shows that 500 is best for sigma1. Here, we obtained all optimum parameter values and showed in Table 7. As a result, the first experiment is benefit in order to inform the suitable parameter value and trend to define all parameter value for another different movement, which is the best if we can initial them closely to the real values. Hence when we experiment the proposed algorithm with different movement velocity and plot error distance at each position when time is past, the graphs, in Figure 18 and 19, show that most of the time error distance from our approach is lower than error distance from GPS data where the first graph is the result from constant velocity tracking and the second graph is from inconstant velocity tracking.

Furthermore we did the comparison between an original particle filtering algorithm and our algorithm which employs images in particle filter. Assume that GPS data is 100% error, Figure 17 shows the percentage error of each method when the deviation of noise is changed. The result demonstrates that our algorithm gives the better performance when the error noise is not much however the performance is reduced when noise is increasing.

Table 2 Tracking performance with different initial velocity

$\mathbf{v}_p$ (cm/s)	$[0\ 0\ 0]^T$		$[30\ 0\ 0]^T$		$[30\ 30\ 30]^T$	
GPS	Error					
	(cm.)	(%)	(cm.)	(%)	(cm.)	(%)
original	24.19	100	23.93	100	24.04	100
by pf	20.45	84.55	20.01	83.64	20.54	85.45
by fusing pf and image	19.44	80.37	18.82	78.65	19.16	79.68

Table 3 Tracking performance with different maximum accelerations.

$\mathbf{a}_{max}$ (cm/s^2)	30		60		90		120	
GPS	Error							
	(cm.)	(%)	(cm.)	(%)	(cm.)	(%)	(cm.)	(%)
original	24.19	100	23.48	100	24.10	100	24.29	100
by pf	20.45	84.55	22.25	94.76	23.03	95.57	23.80	98.01
by fusing pf and image	19.44	80.37	20.16	85.83	21.66	89.90	22.62	93.13

Table 4 Tracking performance with different maximum angular velocity

Initial angular velocity	$[0\ 0\ 0]^T$				$[10\ 10\ 10]^T$			
$\boldsymbol{\omega}_{max}$ (degree/s)	2		10		10		20	
GPS	Error							
	(cm.)	(%)	(cm.)	(%)	(cm.)	(%)	(cm.)	(%)
original	24.19	100	23.85	100	24.04	100	23.88	100
by pf	20.45	84.55	20.86	87.45	20.68	86.04	20.79	87.06
by fusing pf and image	19.44	80.37	21.15	88.68	21.05	87.55	23.62	98.88

Table 5 Tracking performance with different number of particle

Numbers of Particle	1000		5000	
GPS	Error			
	(cm.)	(%)	(cm.)	(%)
original	23.93	100	23.85	100
by pf	20.01	83.64	20.22	84.77
by fusing pf and image	18.82	78.65	18.76	78.63

Table 6 Tracking performance with different weight

σ_1	10		100		500		1000		5000	
GPS	Error									
	(cm.)	(%)	(cm.)	(%)	(cm.)	(%)	(cm.)	(%)	(cm.)	(%)
original	20.91	100	20.91	100	20.91	100	20.91	100	20.91	100
by fusing pf and image	28.01	133.9	19.72	94.33	18.78	89.83	19.90	95.18	20.15	96.40

Table 7 All optimum parameter values for tracking by fusing video and GPS data

Parameters	Value
Maximum acceleration, $\mathbf{a}_{max}$ (unit/s^2)	1
Maximum angular velocity, $\boldsymbol{\omega}_{max}$ (degree/s)	2
Initial velocity, $\mathbf{v}_p$ (unit/s)	1
Weighted parameter, σ_1	500
Numbers of particle, N	5000
Standard deviation of error, SD	0.5

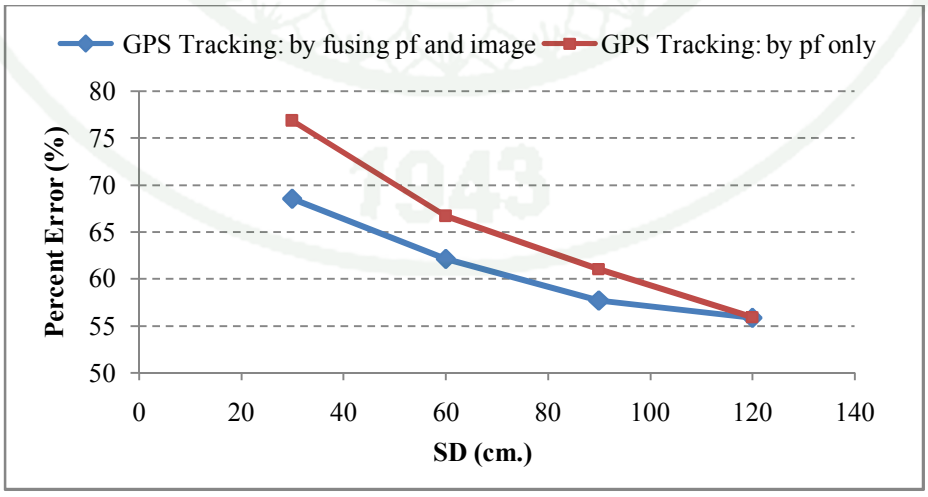


Figure 17 A comparison of percentage error between the position from GPS and our algorithm.

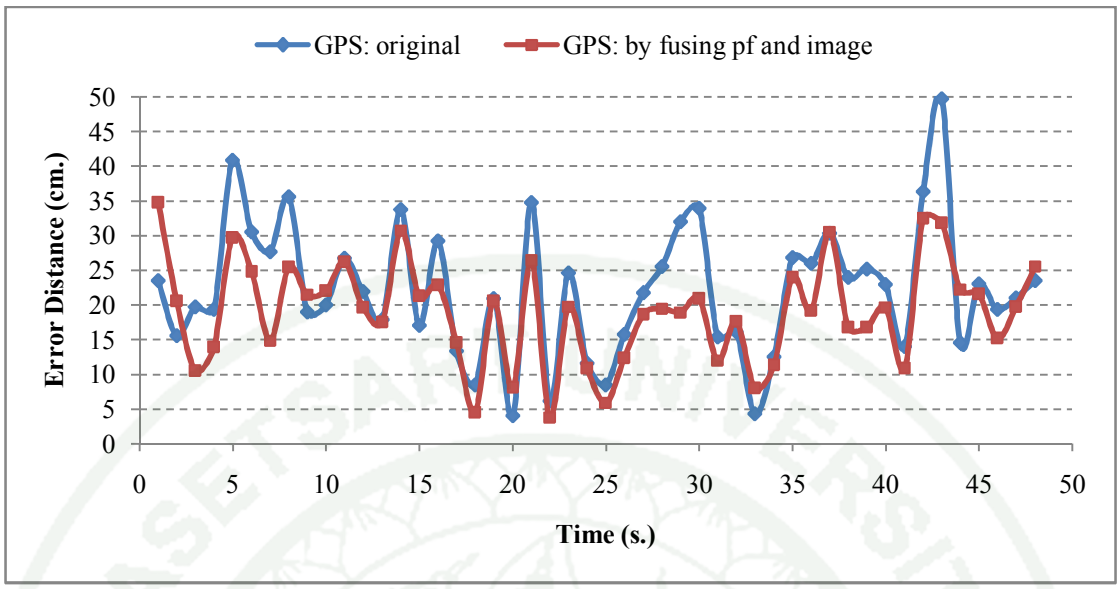


Figure 18 A General Tracking when velocity is constant: the comparison of shifted distance over time between original GPS data and new GPS by fusing particle filter and image

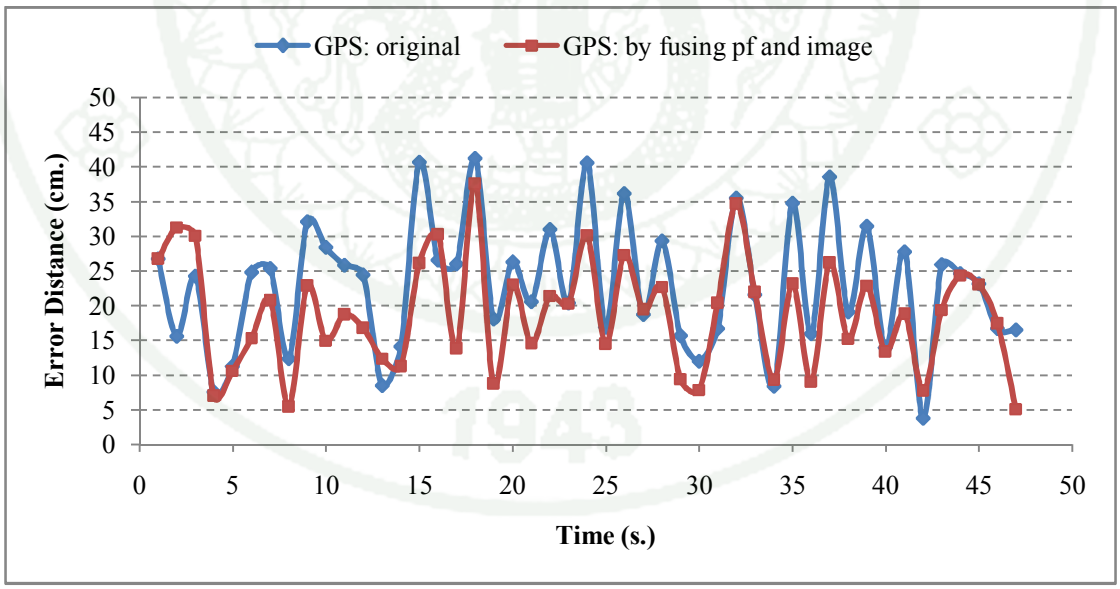


Figure 19 A General Tracking when velocity is not constant: the comparison of shifted distance over time between original GPS data and new GPS by fusing particle filter and image

3.1.2 Lost GPS tracking

We also attempt for the incidence when GPS is lost. In the case of tracking positions from GPS is lost frame skip frame, these positions without GPS are recovered using our method. The main idea uses the translation from the previous frame that has GPS until the current frame that also has GPS. Due to the difference of calculation, the proposed method can be separated into two solutions.

1. Method I uses the relationship of each two views
2. Method II uses the relationship of all three views

In this experiment, the initialized parameters are noise variance 15 cm., initial velocity 0 cm/s, maximum angular velocity 0.0175 radian/s (2 degree/s), and σ_1 500. Due to the difference of motion velocity, the maximum acceleration was changed according to each movement. There are three directional motions X, Y, and Z where X and Z represent the horizontal movement as a planar and Y represents the vertical movement. In the case of moving with constant velocity, the maximum acceleration was set to 30 cm/s^2 for X and Z direction, and 15 cm/s^2 for Y direction while the maximum acceleration for inconstant velocity was set to 45 cm/s^2 for X and Z direction, and remained as ever for Y direction. We compared the ability to recover the lost positions using our method (both method I and II) with a simple recovery by interpolation of the remaining GPS data. The results for 50-time average are shown in Table 8 which can be noticed that both methods give a very close performance for both cases of different velocity. However they take different computational time, method I take around 3 times more than method II because of many back reprojections between two frames. Therefore, method II seems to be more suitable than method I. Here the average error distance at each position is also drawn on Figure 20 and 21 for constant velocity motion and Figure 22 and 23 for inconstant velocity motion.

Once error noise is more than 15 cm., Table 9 and 10 show the average error distance after recovering the lost positions when error noise increases from 15

cm. to 120 cm.. The experiment did a performance comparison when the camera moved with a constant velocity and when the camera moved with inconstant velocity. All parameter were set to the same set, except the maximum acceleration, [30, 15, 30] cm/s^2 for constant velocity and [45, 15, 45] cm/s^2 for inconstant velocity. These tables show error distance in centimeter and percent error over original GPS position, assume GPS data is 100% error. As a result, from both kinds of movement, the recovery of GPS position using method II, which reconstructs 3D positions from three corresponding images at the same time, gives the better performance over method I with less time complexity. The trends of average error occurred in each method with different movement are shown in Figure 24 and 25.

Table 8 The average error distance of each method when moving with different velocity

Error	Constant velocity		Inconstant velocity	
	(cm.)	(%)	(cm.)	(%)
Method I				
GPS: Original	20.45	100	22.93	100
GPS: by fusing pf and image	18.70	91.43	21.99	95.91
Method II				
GPS: Original	20.45	100	22.93	100
GPS: by fusing pf and image	18.84	92.09	21.77	94.95

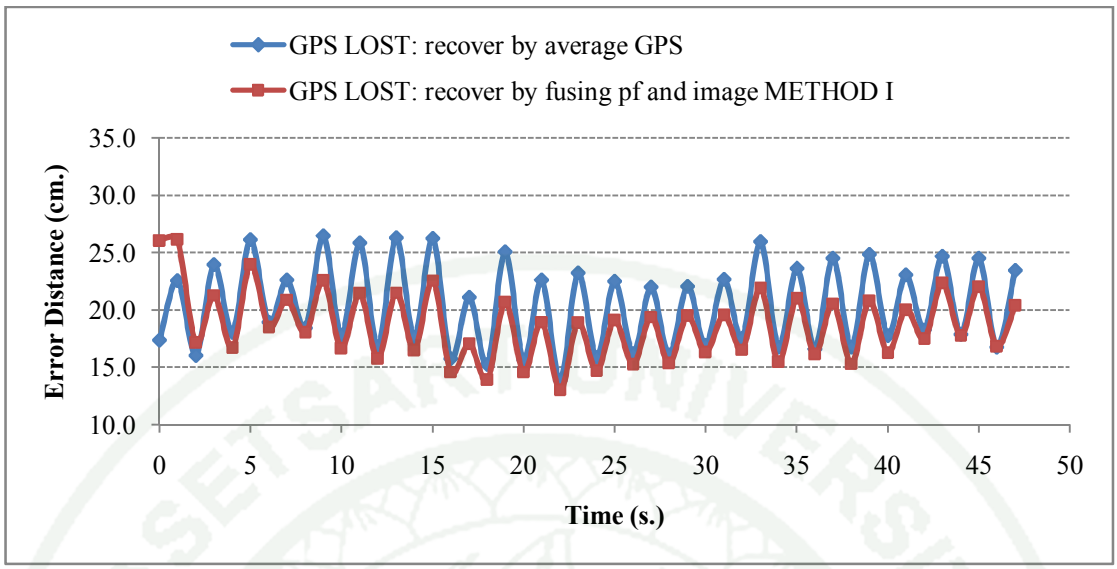


Figure 20 Recover lost position method1 when velocity is constant: the comparison of shifted distance over time between average GPS data and new GPS by fusing particle filter and image method I.

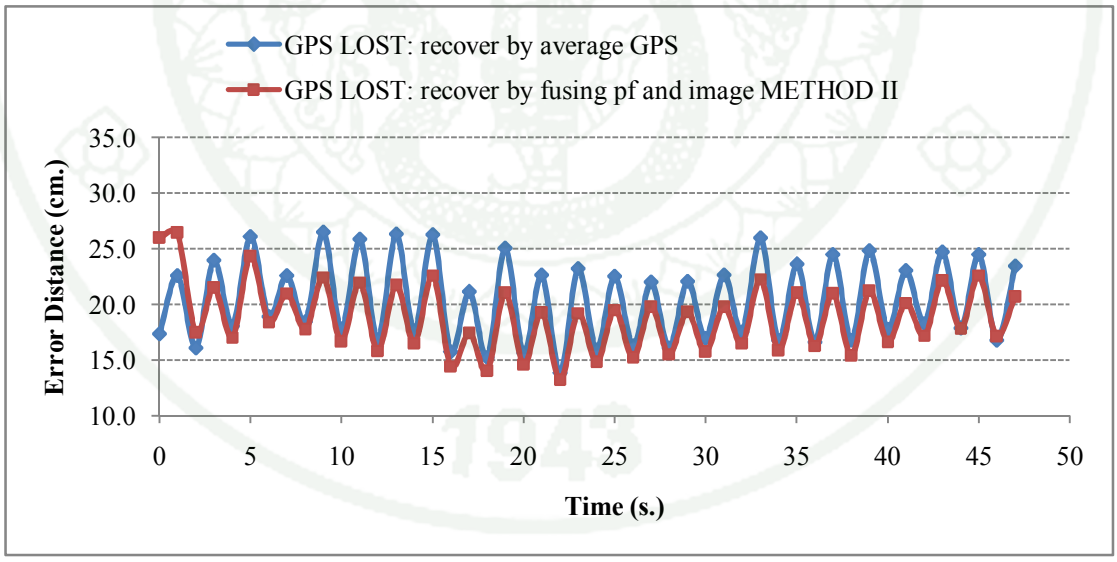


Figure 21 Recover lost position method2 when velocity is constant: the comparison of shifted distance over time between average GPS data and new GPS by fusing particle filter and image method II.

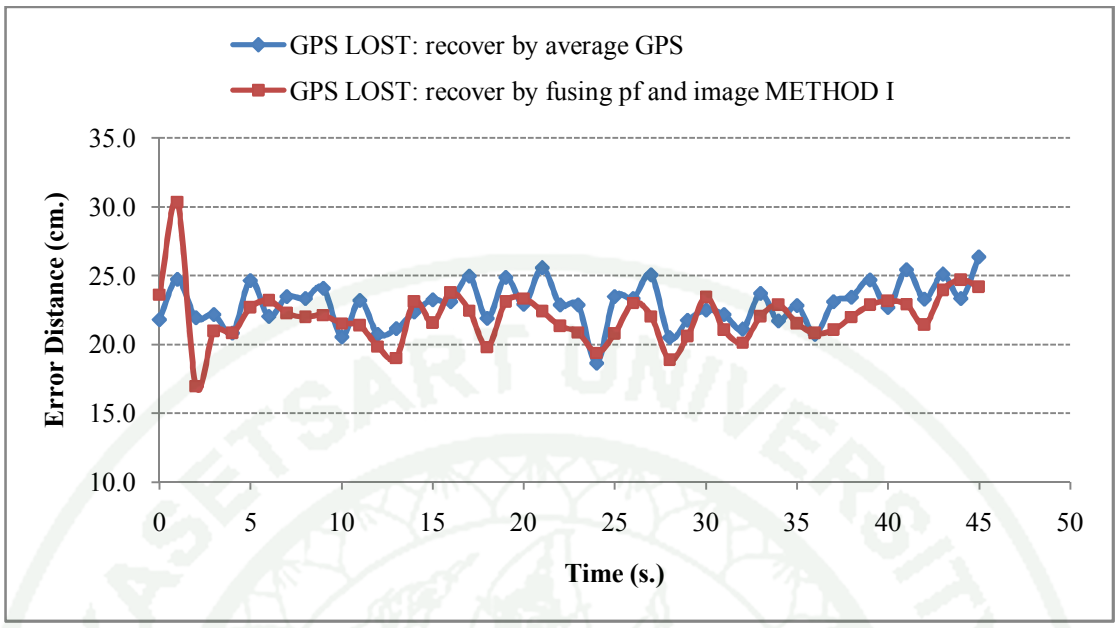


Figure 22 Recover lost position method2 when velocity is not constant: the comparison of shifted distance over time between average GPS data and new GPS by fusing particle filter and image method I.

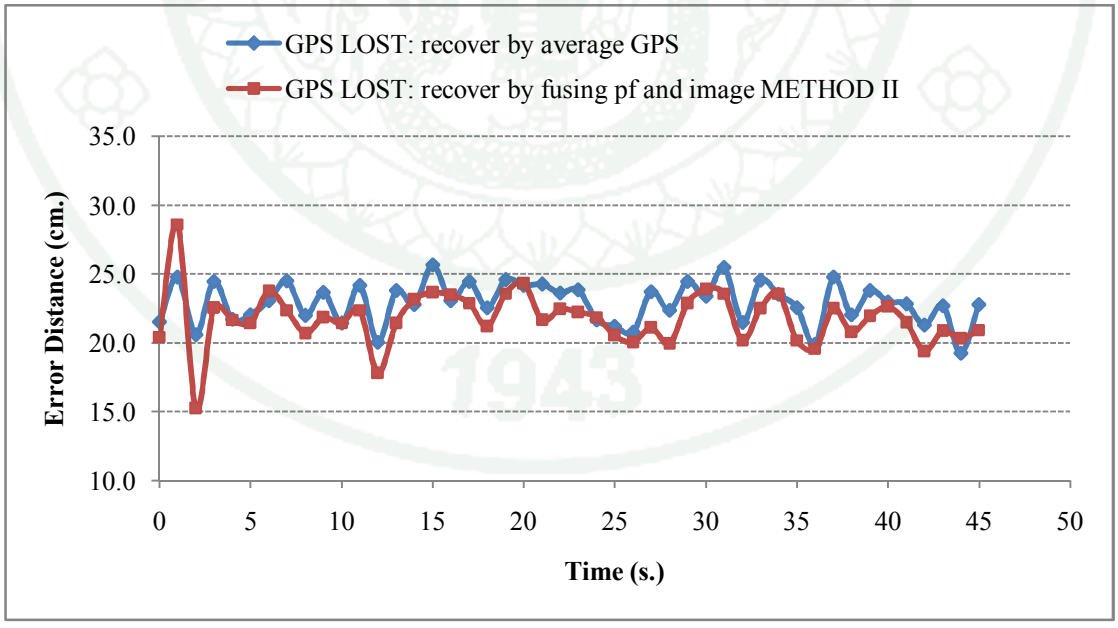


Figure 23 Recover lost position method2 when velocity is not constant: the comparison of shifted distance over time between average GPS data and new GPS by fusing particle filter and image method II.

Table 9 Constant velocity: the performance of each method when errors increase continuously.

Error	SD (cm.)															
	15		30		45		60		75		90		105		120	
	(cm.)	(%)	(cm.)	(%)	(cm.)	(%)	(cm.)	(%)	(cm.)	(%)	(cm.)	(%)	(cm.)	(%)	(cm.)	(%)
Method 1																
GPS	20.45	100	38.99	100	65.93	100	81.67	100	96.22	100	128.34	100	142.14	100	169.07	100
Pf and Image	18.70	91.43	34.12	87.51	54.07	82.00	64.48	78.95	79.68	82.81	95.44	74.37	115.36	81.16	136.16	80.54
Method 2																
GPS	20.45	100	38.71	100	64.17	100	82.34	100	101.63	100	115.46	100	149.73	100	162.45	100
Pf and Image	18.84	92.09	33.86	87.48	51.24	79.85	67.31	81.74	78.32	77.06	96.83	83.86	111.43	74.42	132.09	81.31
Original particle filtering method																
GPS	20.48	100	41.72	100	60.82	100	80.85	100	97.58	100	122.82	100	140.73	100	166.59	100
Pf only	19.79	96.63	38.42	92.10	55.04	90.50	71.61	88.57	84.20	86.29	101.70	82.80	115.20	81.86	134.73	80.87

Table 10 Inconstant velocity: the performance of each method when errors increase continuously.

Error	SD (cm.)															
	15		30		45		60		75		90		105		120	
	(cm.)	(%)	(cm.)	(%)	(cm.)	(%)	(cm.)	(%)	(cm.)	(%)	(cm.)	(%)	(cm.)	(%)	(cm.)	(%)
Method 1																
GPS	22.93	100	39.54	100	60.99	100	85.99	100	106.29	100	133.34	100	133.16	100	170.11	100
Pf and Image	21.99	95.91	37.42	94.63	50.71	83.15	68.16	79.27	89.00	83.74	102.19	76.64	112.84	84.74	131.99	77.59
Method 2																
GPS	22.93	100	39.54	100	65.06	100	84.77	100	102.45	100	122.97	100	146.58	100	165.14	100
Pf and Image	21.77	94.95	37.11	93.86	56.12	86.27	70.07	82.66	82.87	80.89	98.28	79.93	109.57	74.75	118.77	71.92
Original particle filtering method																
GPS	22.51	100	41.70	100	60.20	100	83.17	100	107.93	100	124.96	100	147.28	100	162.77	100
Pf only	21.75	96.61	39.18	93.95	55.46	92.14	73.58	88.47	93.86	86.96	109.94	87.98	123.24	83.68	131.54	80.82

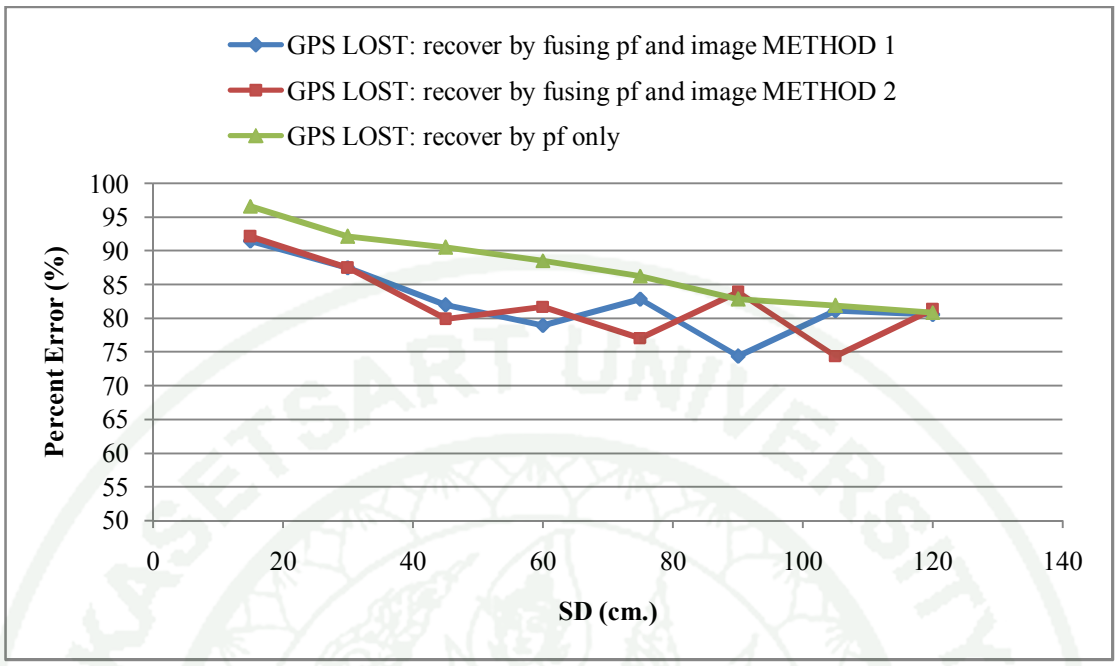


Figure 24 The comparison of percent error from each method when velocity is constant.

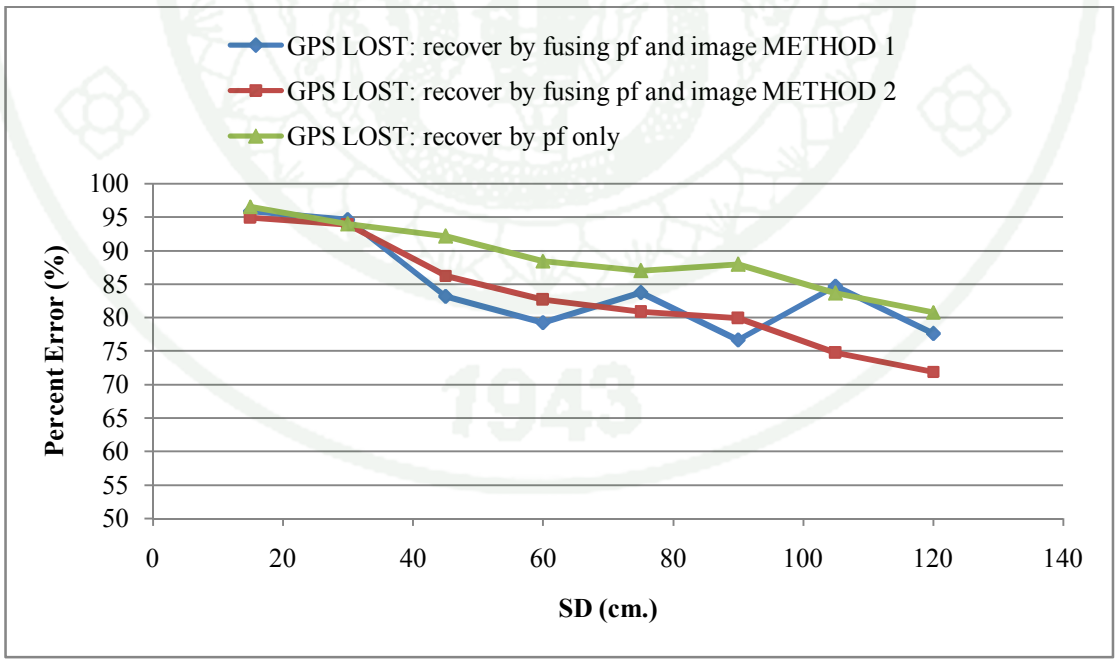


Figure 25 The comparison of percent error from each method when velocity is not constant

3.2 Real Survey Tracking

We observed the rice field where is in the preparation process before farming. Examples of the captured images are shown in Figure 26. There are totally 80 images synchronized with GPS data. After we put all data into our tracking algorithm and set all parameter values; $\mathbf{a}_{max}$ to 0.5 m/s^2, $\boldsymbol{\omega}_{max}$ to 30 degree/s, the deviation of GPS error to 15 meters, the weight parameter to 500, and number of particle to 5000. The result is a path drawn on Figure 27, which is a map in universal transverse Mercator coordinate (UTM) consists of the reference path (gray dash-line), the original path from a GPS receiver (blue line), and the new path from our tracking method (red line). All positions are also drawn on the real map, Google Map, and show in Figure 28 which left hand side comes from GPS data and right hand side comes from the proposed method. The result has 7% advantage over GPS data.

Figure 26 The captured images from the field survey at Kasetsart University

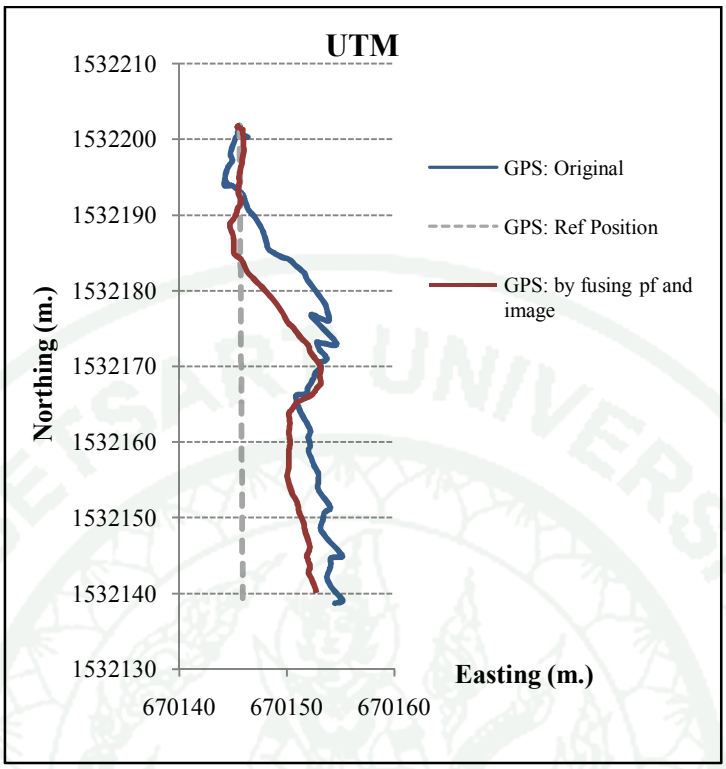


Figure 27 All the tracking paths drawn in UTM coordinate.

(a) (b)

Figure 28 (a) is the positions from GPS receiver and (b) is the position from the proposed method.

Discussion

Accuracy of corresponding points affects the reprojection errors used for weighting. So the more precision of match points could improve the accuracy of vehicle tracking. Here, we choose a powerful method called SIFT which is invariant to image scaling, rotation, illumination, and 3D camera view point. However, there are some errors of these match points. To improve the accuracy of these match points, we propose three procedures; start from refine the keypoint coordinate according to the similarity of neighbor pixel using SSD, after that discard any match that is dissimilar, and then discard any match having the direction and distance of the moved pixel does not conform to other matches. Due to these procedures we can reduce error up to 95%.

For the tracking, our proposed method employs image sequence in particle filter. Because the relationship of two images can inform camera motion direction, there is such a method to measure the correct of the movement, called triangulation. As shown in the second experiment, if the direction is correct, in any case of scale, the error from the reprojection will be minimal. The methods can be separated into two cases according to the availability of GPS data. If all images are synchronized with GPS data, this case is called a general tracking. Another case, if all images are not synchronized with GPS due to GPS lost or inequality of video frame rate and receive GPS rate, we call this case as a GPS lost. In the experiment we both experiment on the simulation tracking and the field survey tracking. For the simulation tracking in the case of a general tracking, the result shows the advantage over original GPS data and original particle filter method. Assume that GPS data is 100 % error, the best result from our algorithm can reduce error to 78.63%. Here, we obtain the tendency of the optimum value that is suitable for the model. Therefore, we are able to define an appropriate parameter value for other experimental tracking. Other simulations are the cases when GPS is lost, here, we have two solutions to recover the lost position. These solutions called method I and method II have different information to inform the directions in tracking. Method I uses the relationship of each two views to reconstruct points in the world coordinate so the reprojection errors are calculated

from different 3D points. On the other hand, method II utilizes the relationship of all three views so we obtain the more certainty of the reconstructed 3D points. Here, different velocities were simulated as the motions of camera. The results show that both methods I and II have ability to recover the positions either constant or inconstant velocity motion. However, they prosper to the constant velocity which gives the better result. Even though method I and II give a similar performance, due to time complexity of method I caused by several reconstructions and back reprojections, method II seems to be more useful than method I.

For the real field survey, the pre-process to collect image sequence and GPS data is very important. Since there are many problems for the gathered data such as the captured images do not have enough features or GPS receiver cannot receive the signal to identify the position due to the weather. These all are not enough quality data which cannot inform any information for the tracking. Actually this procedure is not inconvenient, just driving a car not over 1 minute, but the camera should be mounted in the right position and angle. Here we learn that if the camera only captures the image that has the sky in a frame, this image will have enough features from this background. Once we obtained all data, the most important thing for the post-process is to define an initial value of each parameter. Since we used to do the experiment for optimum parameter values before, now some of them can be used for the real survey data as well. These are weight parameter (σ_1), and numbers of particle (N), set to 500 and 5000. However, there are still such parameters that cannot use old values because of the different motion such as the maximum acceleration, the maximum angular velocity, the deviation of GPS error. These all parameter should be defined relate to its actual movement. In the experiment, they were set to 0.5 m/s^2, 30 degree/s, and 15 m., respectively. Here there is another important process that should be prepared for the algorithm other than define the optimum parameter values. This process is to convert GPS coordinate from latitude and longitude coordinate to the UTM coordinate which is a 2-dimensional Cartesian coordinate in meter. The experimental result shows that there is around 7% improvement. When we draw our coordinates as a tracking path, the map shows that the new path is close to the reference more than the path from GPS data.

CONCLUSION AND RECOMMENDATION

Conclusion

An error from GPS data causes an imprecision of position. This research aims to track the motion which improves the correctness of GPS data. Since we utilize the use of corresponding points between two images to inform the direction of motion, the accuracy of these matches is also considered. Once the validation process is done, errors from the mismatch and inaccurate pixel coordinate of these match points are reduced. In the tracking, particle filtering method is employed with images. The relative rotation and translation between two views which satisfy the epipolar geometry have to minimize the reprojected error cost of the corresponding points. This information guides the direction of camera trajectory thus the experiment shows that we obtain the more precision of positions than GPS data and an original particle filtering method. Moreover, for the incidence when GPS data is lost, the proposed method has the ability to determine the position of camera for image scene without GPS data as well. However, there have additional details which can be separated into two methods; method I apply the relationship of each two views and method II apply the relationship of all three views. The performance to recover the lost positions of both methods is very similar but method II has less time complexity so method II seems to be more advantage.

Recommendation

Our algorithm is an offline tracking using two and three views geometry. The further study would be emphasized on multiple views geometry which has the possibility to give more accurate reference positions. However, this must be trade off with a higher computational time in tracking.

LITERATURE CITED

Corvallis Microtechnology, Inc. 2000. **Chapter Six: The GPS Error Budget**. Introduction to the Global Positioning System for GIS and TRAVERSE. Available Source: http://www.cmtinc.com/gpsbook/, March 21, 2011.

Flament, M., G. Fleury and M.E. Davoust. 2004. Particle Filter and Gaussian-Mixture Filter Efficiency Evaluation for Terrain-Aided Navigation, pp. 605-608. *In* **Proceedings of European Signal Processing Conference**. Vienna, Austria.

Hakeem, A., R. Vezzani M. Shah and R. Cucchiara. 2006. Geospatial Trajectory of a Moving Camera. **IEEE Int. Conf. on Pattern Recognition**.

Hartley, R. and A. Zisserman. 2003. **Multiple view geometry in computer vision**. 2nd ed. Cambridge University Press.

Khoomboon, S., T. Kasetkasem, R. Keinprasit and N. Sugino. 2010. Increase a standalone GPS Positioning Accuracy by using a Proximity Sensor, pp. 584-587. *In* **Proceedings of ECTI conference**.

Kosecka, J. and X. Yang. 2004. Global Localization and Relative Pose Estimation Based on Scale-Invariant Features. **IEEE Int. Conf. on Pattern Recognition**.

Levinson, J., M. Montemerlo and S. Thrun. 2007. Map-Based Precision Vehicle Localization in Urban Environments. **Robotics: Science and Systems**.

Limsoonthrakul, S., M.N. Dailey and M. Parnichkun. 2009. Intelligent Vehicle Localization Using GPS, Compass, and Machine Vision, pp. 3981-3986. *In* **Proceedings of IEEE Transaction on Intelligent Robots and Systems**.

Lowe, D.G. 2004. Distinctive image features from scale-invariant keypoints. **International Journal of Computer Vision** 60(2): 91-110.

Mammarella, M., G.Campa, M.R. Napolitano, M.L. Fravolini, Y. Gu and M.G. Perhinschi. 2008. Machine Vision/GPS Integration using EKF for The UAV Aerial Refueling Problem. **IEEE transaction on Systems, Man, and Cybernetics** 38(6).

Mikolajczyk, K. and C. Schmid. 2005. A Performance Evaluation of Local Descriptors. **IEEE Transactions on Pattern Analysis and Machine Intelligence** 27(10): 1615-1630.

Pharsook, S., P. Larmsrichan, S. Siddhichai, T. Kasetkasem, T. Chanwimaluang and T. Isshiki. 2011. The Comparison of Three Feature Extraction for Classification of Image Texture. **ICICTES 2011**. Thailand.

Pupilli, M. and A. Calway. 2005. Real-Time Camera Tracking Using a Particle Filter, pp. 519-528. *In* **Proceedings of British Machine Vision Conference**.

Wang, J., R. Cipolla and H. Zha. 2004. Image-based Localization and Pose Recovery Using Scale Invariant Features, pp. 711-715. *In* **Proceedings of IEEE Int. Conf. on Robotics and Biomimetrics**.

CIRRICULUM VITAE

NAME : Ms. Kanittha Sae-Lim

BIRTH DATE : February 16, 1987

BIRTH PLACE : Prachuapkhirikhan, Thailand

EDUCATION :

YEAR	INSTITUTE	DEGREE/DIPLOMA
2009	Kasetsart Univ.	B.Eng. (Electrical Engineering)
2011	Kasetsart Univ.	M.Eng (Information and Communication Technology for Embedded System)

POSITION/TITLE : -

WORK PLACE : -

SCHOLARSHIP/AWARDS : TAIST ICTES Master Degree Scholarship

PUBLICATIONS : ICICTES 2011
IC2IT 2011