

The 24th Conference of the Mechanical Engineering Network of Thailand
October 20 – 22, 2010, Ubon Ratchathani

Control of Inertial Stabilization Systems Using Image Tracking of Non-Rigid Objects

Kritsanun Malithong and Viboon Sangveraphunsiri

Department of Mechanical Engineering Chulalongkorn University
254 Phyathai Patumwan Bangkok 10330

E-Mail: Kritsanun.M@student.chula.ac.th, Viboon.S@eng.chula.ac.th, Tel. 0-2218-6449, Fax 0-2218-6583

Abstract

Inertial stabilization systems or camera gimbal is a device that be set up into the aircraft structure. The motion control of camera gimbal can be divided into 2 parts. The first part is a feedback control system in order to move the gimbal according to a reference command and the second part is the tracking moving objects in video. This paper presents an approach to the image tracking for an inertial stabilization system. This method is the real-time tracking of non-rigid objects seen from a moving camera. The computational module is based on the Continuously Adaptive Mean-shift (CAMshift) iterations and finds the most probable target position in the current frame. The CAMshift algorithm applications can track object presenting strong modifications of shape. We can quote the management of the target appearance changes during the sequence. The experimental results illustrate that the pan-tilt camera can automatically follow the moving target and record it. The control loop for the pan-tilt unit was developed that would send rate commands to move the camera position in order to keep the target in the center of the camera image.

Keywords: Inertial Stabilization / Gimbal / Visual Object Tracking / CAMshift

1. Introduction

Unmanned Aerial Vehicle (UAV; also known as a remotely piloted vehicle) is a radio control aircraft equipped with video camera. Its initial occupation has been progressing to useful role such as Reconnaissance and Surveillance. In Fig.1, the current process for visually tracking a target with a pan-tilt camera mounted on UAV requires two operators. One operator controls the UAV while the second with a joystick and display

can provide control commands to the camera gimbal. These two people must work together and continuously communicate in order to keep the camera aimed at the target.

Most target trackers are automatic, in which video imagery is processed to estimate target location. Automatic target trackers require a target recognition process, or manual operator, to initially acquire the target. The tracking process thus begins when the tracker is informed that a

selected portion of a video image represents a target.

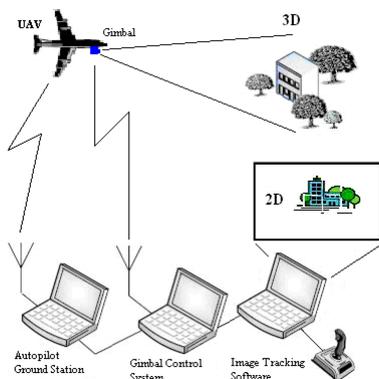


Fig. 1 Overview of Visual Target Tracking

The motion control of camera gimbal can be divided into 2 parts. The first part is controlled by a feedback control system in order to move the gimbal according to a reference command and in the same time to stabilize the gimbal where the camera is attached. The second part is the stabilizing of images done by an image programming technique and autonomous tracking objects in video sequences, which will be focused on this paper.

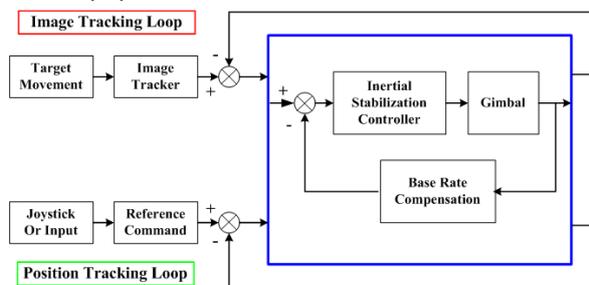


Fig. 2 Motion control of Inertial Stabilization Systems

There are numerous methods that have been developed to solve the object tracking problem. Many approaches are based on the visual primitives tracked in images by employing correlation. Other techniques process the movement of the object in order to track it in the images sequence [6-8]. For tracking Objects from Mobile Platforms, Cohen and Medioni address the

detection and tracking of moving objects in a video stream obtained from a moving airborne platform in their publications [9], [10]. Bell et al. claim that their system is able to follow multiple objects while maintaining the identity of each object [11].

The Regional Center of Robotics Technology at Chulalongkorn University has been extending the capabilities of camera gimbal. Some Previous capabilities include researching for gimbal structure and the controller design [12-14].

This paper presents the Continuously Adaptive Mean-shift algorithm (CAMshift). It is a method to the real-time tracking of non-rigid objects seen from a moving camera based on visual features such as color, whose statistical distributions characterize the object of interest. The proposed tracking is appropriate for a large variety of objects with different color patterns, being robust to partial occlusions, clutter, rotation in depth, and changes in camera position.

2. Object Tracking Using CAMshift

2.1 The CAMshift Algorithm

In 2003, Allen et al. use an algorithm known as the CAMshift Algorithm, which is an adaptation of the Mean-Shift algorithm, for object tracking [2]. The principle of the CAMshift algorithm is given in [1], [2] and [4]. For each video frame, the raw image is converted to a color probability distribution image via a color histogram model of the color being tracked. The center and size of the color object are found via the CAMshift algorithm operating on the color probability image. These new centre and size are employed to place the search window in the next image. This process is then repeated for a continuous target tracking in the video sequence.

The CAMshift algorithm thus employs a 2D probability distribution image produced from a back-projection of the target histogram with the image to process. The CAMshift calls upon the MeanShift one to calculate the target centre in the probability distribution image [5]. It is a matter of finding a rectangle presenting the same moments as those measured on the probability image. These parameters are given from the first and second moments [1], [3].

The main focus of our works was the development of control laws to make autonomous visual tracking of any target. We have implemented is primarily based on the CAMshift algorithm. The steps of this algorithm are stated as follows [1] :

1. Set the region of interest (ROI) of the probability distribution image to the whole image.
2. Choose an initial location of the 2D Mean Shift search window. The selected location is the target distribution to be tracked.
3. Calculate a color probability distribution in the 2D region centered at the search window location in an ROI slightly larger than the mean shift window size.
4. Iterate Mean Shift algorithm to find the center of the probability image. Store the zeroth moment (distribution area) and center location.
5. For the following frame, center the search window at the mean location found in Step 4 and set the window size to a function of the zeroth moment. Go to Step 3.

Fig. 3 demonstrates Block diagram of CAMshift algorithm. This algorithm is a generalization of the Mean Shift algorithm, highlighted in gray in Fig. 3.

2.2 The probability distribution image

The probability distribution image (PDF) may be determined using any method that associates a pixel value with a probability that the given pixel belongs to the target. A common method is known as Histogram Back-Projection. In order to generate the PDF, an initial histogram is computed at Step 1 of the CamShift algorithm from the initial ROI of the filtered image.

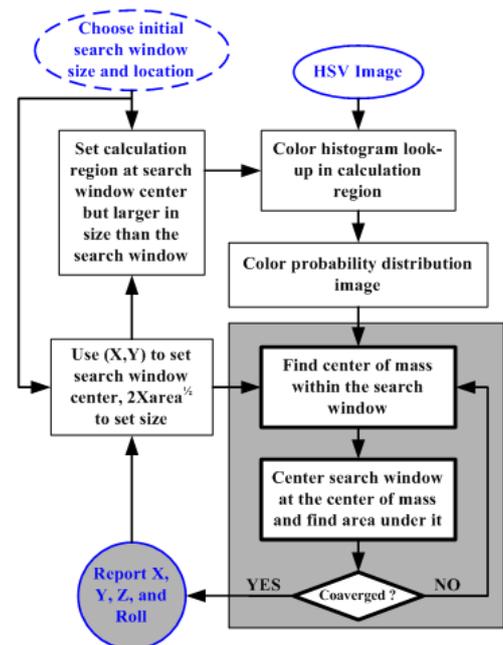


Fig. 3 Block diagram of CAMshift algorithm[1]

The histogram used in Bradski [16] consists of the hue channel in HSV (Hue, Saturation, Value) or HSB (Hue, Saturation, Brightness) color modal, however multidimensional histograms from any color space may be used. The histogram is quantized into bins, which reduces the computational and space complexity and allows similar color values to be clustered together. The histogram bins are then scaled between the minimum and maximum probability image intensities using Eq. (2).

2.3 Histogram Back-Projection

Histogram back-projection is a primitive operation that associates the pixel values in the image with the value of the corresponding

histogram bin. The back-projection of the target histogram with any consecutive frame generates a probability image where the value of each pixel characterizes probability that the input pixel belongs to the histogram that was used.

Given that m -bin histograms are used, we define the n image pixel locations $\{x_i\}_{i=1\dots n}$ and the histogram $\{\hat{q}\}_{u=1\dots m}$. We also define a function $c: \mathbb{R}^2 \rightarrow \{1\dots m\}$ that associates to the pixel at location the x_i^* histogram bin index $c(x_i^*)$. The unweighted histogram is computed as

$$\hat{q}_u = \sum_{i=1}^n \delta[c(x_i^*) - u] \quad (1)$$

The histogram bin values are scaled to be within the discrete pixel range of the 2D probability distribution image using

$$\left\{ \hat{p}_u = \min \left(\frac{255}{\max(\hat{q})} \hat{q}_u, 255 \right) \right\}_{u=1\dots m} \quad (2)$$

That is, the histogram bin values are rescaled from $[0, \max(\hat{q})]$ to the new range $[0, 255]$, where pixels with the highest probability of being in the sample histogram will map as visible intensities in the 2D histogram back-projection image.

2.4 Mass Centre Calculation for 2D Probability Distribution

For discrete 2D image probability distributions, the mean location (the centroid) within the search window of the discrete probability image computed in Step 3 is found using moments [1], [16-18]. Given that $I(x, y)$ is the intensity of the discrete probability image at (x, y) within the search window.

Compute the zeroth moment

$$M_{00} = \sum_x \sum_y I(x, y)$$

Find the first moment for x and y

$$M_{10} = \sum_x \sum_y xI(x, y)$$

$$M_{01} = \sum_x \sum_y yI(x, y)$$

Compute the mean search window location

$$x_c = \frac{M_{10}}{M_{00}}; y_c = \frac{M_{01}}{M_{00}}$$

The direct projection of the model histogram onto the new frame is known to introduce a large bias in the estimated location of the target and the measurement is known to be scale variant.

2.5 Target Model for Localization

2.5.1 Weighted Histogram

The initial selected region contains some pixels from outside the object (background pixels), our 2D probability distribution image will be influenced by their frequency in the histogram back-projection. In order to assign higher weighting to pixels nearer to the region center, a weighted histogram may be used to compute the target histogram [20]

$$\hat{q}_u = \sum_{i=1}^n k \|x_i^*\|^2 \delta[c(x_i^*) - u] \quad (3)$$

The resulting histogram is scaled using Eq. (2) for the discrete quantities we are using and $k(x)$ is any convex, monotonically decreasing kernel profile that assigns higher weight to pixels near the centre of the normalized search window. The most simple kernel profile used to generate the background-weighted histogram in our experiment is shown in Eq. (4)

$$k(r) = \begin{cases} 1 - r & , r \leq 1 \\ 0 & , otherwise \end{cases} \quad \dots(4)$$

It is worth noting that since the Mean Shift iterations are based on moment calculations and do not require an estimate of the probability density gradient, the selected kernel profile does not need to be differentiable or have a constant derivative (kernels with Epanechnikov profile, for instance).

2.5.2 Ratio Histogram

The weighted histogram selected in 2.5.1 is not sufficient to identify the target when histogram back-projection is used to generate the 2D probability distribution image. In a sequence of experiments it has been observed that if the target histogram contains a significant number of features that belong to the background image or neighboring objects, target localization and scale cannot be accurately determined.

A ratio histogram can help to solve the background problem by assigning color features that belong to the background with lower weights [20]. In our experiment, we compute a histogram for a region outside the normalized target location using a kernel a with the following profile

$$k(r) = \begin{cases} ar & , 1 < r \leq h \\ 0 & , otherwise \end{cases} \quad (5)$$

Where a is a scaling factor and h is the bandwidth of the new search window. A background region that is two times as large as the target region ($h = 3$) was used in the experiment. A histogram $\{\hat{O}\}_{u=1\dots m}$ is computed using Eq. (3) with a bandwidth h and then weighted using Eq. (6) where \hat{O}^* is the smallest nonzero entry

$$\left\{ \hat{w}_u = \min \left(\frac{\hat{O}^*}{\hat{O}_u}, 1 \right) \right\}_{u=1\dots m} \quad (6)$$

The background-weighted histogram used in our experiment is therefore given as

$$\hat{q}_u = \hat{w}_u \sum_{i=1}^n k \|x_i^*\|^2 \delta [c(x_i^*) - u] \quad (7)$$

2.6 Orientation and Scale Calculation

The use of moments to determine the scale and orientation of a distribution in robot and computer vision is described in Horn [17].

The orientation (θ) of the major axis and the scale of the distribution are determined by

finding an equivalent rectangle that has the same moments as those measured from the 2D probability distribution image [1], [17]. Defining the first and second moments for x and y

$$\begin{aligned} M_{20} &= \sum_x \sum_y x^2 I(x, y) \\ M_{02} &= \sum_x \sum_y y^2 I(x, y) \\ M_{11} &= \sum_x \sum_y xy I(x, y) \end{aligned}$$

The first two eigenvalues (the length and width of the probability distribution) are calculated in closed form as follows. From the intermediate variables a , b and c

$$\begin{aligned} a &= \frac{M_{20}}{M_{00}} - x_c^2 \\ b &= 2 \left(\frac{M_{11}}{M_{00}} - x_c y_c \right) \\ c &= \frac{M_{02}}{M_{00}} - y_c^2 \end{aligned}$$

Then the object orientation, or direction of the major axis, is

$$\theta = \frac{1}{2} \tan^{-1} \left(\frac{b}{a - c} \right) \quad (8)$$

The distances l_1 and l_2 from the distribution centroid (the dimensions of the equivalent rectangle) are given by,

$$l_1 = \sqrt{\frac{(a + c) + \sqrt{b^2 + (a - c)^2}}{2}} \quad (9)$$

$$l_2 = \sqrt{\frac{(a + c) - \sqrt{b^2 + (a - c)^2}}{2}} \quad (10)$$

Where the extracted parameters are independent of the overall image intensity.

3. The Controller Design

At Regional Center of Robotics Technology, Previous capabilities include researching for the controller design. The robust inverse dynamics control and sliding mode control with the indirect stabilization control configuration is used to control the overall system, so that the gimbal camera can track the reference trajectory

or maintain its line of sight (LOS) while disturbances and base motion [12-14]

In this paper, the robust inverse dynamics control is selected for a feedback control system in order to move the gimbal according to a reference command $q_d, \dot{q}_d, \ddot{q}_d$. The reference command is generated from the Image tracking part.

3.1 Robust Inverse Dynamics Control

The control vector will be expressed by:

$$\tau = \hat{\mathbf{D}}(q)y + \hat{\mathbf{N}}(q, \dot{q}) \quad (11)$$

where $\hat{\mathbf{N}}(q, \dot{q})$ is the estimate of:

$$(\mathbf{C}(q, \dot{q})\dot{q} + \mathbf{F}_s \text{sgn}(\dot{q}) + \mathbf{g}(q))$$

$\hat{\mathbf{D}}(q)$ is the estimate of $\mathbf{D}(q)$

In this expression, q is the vector of joint angles, τ is the torque vector applied to the joints, $\mathbf{D}(q)$ is the inertia matrix, $\mathbf{C}(q, \dot{q})$ is the vector of centripetal and Coriolis forces, \mathbf{F}_s is an approximated friction forces. $\text{sgn}(\dot{q}) = +1$ when \dot{q} is positive and $\text{sgn}(\dot{q}) = -1$ when \dot{q} is negative. $\mathbf{g}(q)$ is the vector of gravitational forces and function.

The input y can be selected as usual:

$$y = \ddot{q}_d + K_D \dot{\tilde{q}} + K_P \tilde{q} + K_I \int_0^t \tilde{q} dt + w \quad (12)$$

Where $\tilde{q} = q_d - q$, The gain K_P, K_D, K_I will be selected. The term w is to be designed to guarantee robustness to the effects of uncertainty. The term w is described in [12].

3.2 The Controller for tracking Objects from Mobile Platforms

Our control system was developed to accomplish a goal; the control loop can control the camera position in order to keep the target in the center of the camera image.

From the CAMshift algorithm, we are tracking a target that is moving across the view of a video camera. At each frame, we make a determination of the location of this target or find an estimate of the position of the target. This estimation is not likely to be extremely accurate. The reasons for this are many. They may include approximations in earlier processing stages, inaccuracies in the sensor, the apparent changing of shape, or issues arising from occlusion. We can think of all these inaccuracies, taken together, as simply adding noise to our tracking process.

The machinery for accomplishing the estimation task falls generally under the heading of estimators, with the Kalman filter being the most widely used technique. Thus, we choose the CAMshift algorithm and Kalman filter to find an estimate of the position of the target.

In digital imaging, a pixel (or picture element) is a single point in an image. Pixels are normally arranged in a 2-dimensional grid (columns, rows). The display resolution of the images is simply the physical number of columns and rows of pixels creating the display (e.g., 640x320).

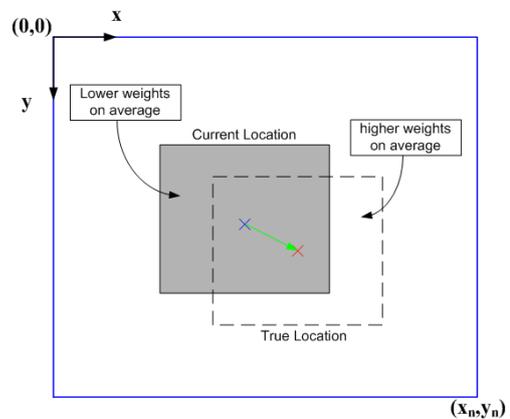


Fig. 4 Shifting of the Mean-Shift tracking window towards higher average weights.

In Fig.4, the display resolution of the image is $X_n \times Y_n$. The center of the image is a haft of the display resolution of the image $\left(\frac{X_n}{2}, \frac{Y_n}{2}\right)$. From the CAMshift algorithm, we know the pixel location of the target on each video frame. If we know the movement of target in the image, we can generate the reference command of the pan-tilt camera.

If the target move in X axis (columns of

pixels), the camera gimbal must be control to move in azimuth axis (pan the camera). Similarly, the target move in Y axis (rows of pixels), the camera gimbal must be control to move in pitch axis (tilt the camera).

Fig. 5 shows the block diagram of the robust inverse dynamics control and the real-time image tracking is added to detect the non-rigid target seen from the camera gimbal.

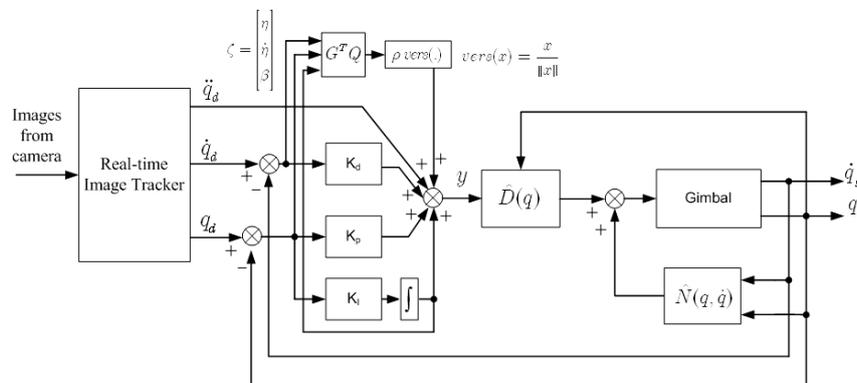


Fig. 5 The block diagram of the robust inverse dynamics control with the real-time image tracking

4. Experiment and result

We have applied our method on various video sequences. The 640x320 images illustrate some obtained results. The search window is initially centered at the position of the object in the image. Initial regions for the video sequences were manually selected by clicking mouse. The program will draw a rectangle around the target area of interest. The histograms were selected in the HSV color space. The target histograms were initialized and scaled to image intensity range. A background region (search window) that is two times as large as the target region was used in the experiment. The algorithm runs comfortably at 30 fps on a 2 GHz notebook.

4.1 Tracking with fixed camera

This section describes the results obtained from experiments with the object tracking module. The camera gimbal is fixed.

The video sequence shown in Fig. 6-7 was used to demonstrate object tracking through occlusion as well as the scale and orientation estimation.

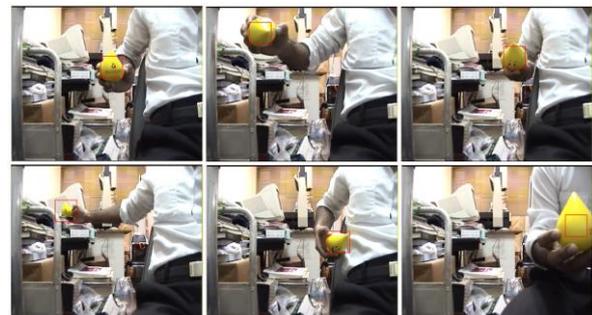


Fig. 6 Tracking the color rubber

By performing several scaling operations automatically, the tracker furthermore showed its

capability to react to different scaling requirements correctly. The color rubber was tracked 100% of the time successfully. In Fig. 7, the tracker can track the car through occlusion if the tracked car is hid only a partial part.

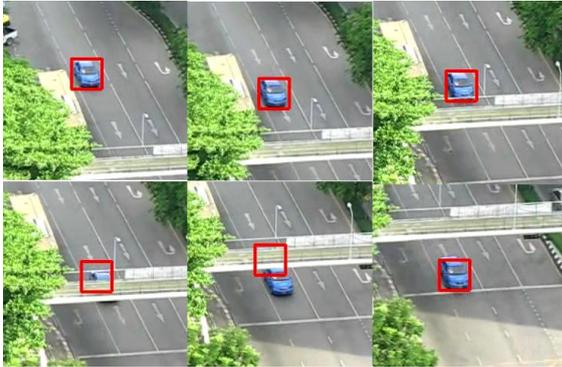


Fig. 7 Tracking the car through occlusion

Fig. 8 demonstrates the similar object-tracking. The tracked target looks very similar to at least two of the other targets it passes during the video sequence.



Fig. 8 Tracking the similar objects

For the experiment shown in Fig. 8, the tracked van looks very similar to the other van. The CAMshift tracker kept up with the tracked van to be tracked until the tracked van is moved closely the other van. The miss tracking problem is happened, if two vans are in the region of interest (ROI) of the tracked object. The tracker

can't decide that the right van or left van is the target because both of vans can be the target. Sometime the tracker chooses the right van and sometime it chooses the left van. For fault detection, the operator can select the correct target again.

4.2 Tracking with controlled camera

The objective of this section is to maintain the camera position in order to keep the target in the center of the camera image. Robust inverse dynamics control is implemented. To test the tracking capability, the reference trajectory $q_d, \dot{q}_d, \ddot{q}_d$ is generated from the image tracker part.

The experimental results show that the gimbal can be controlled in order to keep the target in the center of the camera image.



Fig. 9 Tracking the car, using the controller with image tracking.

4.3 Tracking Objects from Mobile Platforms

In field test, the camera gimbal is mounted on the aircraft structure. In laboratory, the camera gimbal is mounted on the trust frame and hand truck. The video is recoded from the top floor on the high building while the hand truck

is push forward. In this section, the objective of the control is similarly in section 4.2, but the gimbal is set up into the mobile platform, as illustrated in Fig. 10.



Fig. 10 The gimbal mounted on the mobile platform

The gimbal is hung freely, so that a base rate disturbance can be generated to emulate close to the real situation. The controller must track the input and reject the base rate disturbance at the same time. The disturbance is in the form of moving the base.



Fig. 11 Tracking the target from mobile platform.

In Fig.11, the experimental results show that the robust inverse dynamics control with the image tracking control performs very effective for our inertial stabilization system, and is very promising controller. The gimbal can be controlled in order to keep the target in the center of the camera image.

5. Conclusion

We have proposed in this paper an object tracking approach in color images sequences, based on the CAMShift algorithm. The CAMShift shifts its estimate of the target location. So, we can generate the reference command of the pan-tilt camera from the target location. The details of the controllers, the robust inverse dynamic with CAMShift algorithm, of a two-axis gimbal configuration is described. The capability of the tracker to handle in real-time target scale variation, partial occlusion, and significant clutter, is demonstrated. The experiment results are presented to verify the effectiveness of the proposed method in image tracking object. The control loop can move the camera position in order to automatically follow the moving target and keep the target in the center of the camera image.

6. References

- [1] Intel Corporation (2001). Open Source Computer Vision Library Reference Manual, 123456-001.
- [2] Allen, J. G., Xu, Richard, Y. D. and Jin, J. S. (2003). Object tracking using CamShift algorithm and multiple quantized feature spaces, *Conferences in Research and Practice in Information Technology*

- [3] Allen, J. G., Richard, Xu, Y. D. and Jin, J. S. (2004). Object Tracking Using CamShift Algorithm and Multiple Quantized Feature Spaces, Inc. *Australian Computer Society*, vol.36
- [4] Bradski G. R. (1998). Computer vision face tracking for use in a perceptual user interface, *Intel Technology Journal*, 2nd Quarter.
- [5] Cheng Y. (1995). MeanShift, mode seeking, and clustering, *IEEE Trans. on PAMI*, vol.17, pp.790-799.
- [6] Nguyen V. and Tan Y. (2004). Fast Block-Based Motion Estimation Using Integral Frames, *IEEE Signal Processing Letters*, Vol. 11, N° 9, September 2004.
- [7] Besson, S., Barlaud, M., and Aubert, G. (2002). A 3-Step Algorithm Using Region-Based Active Contours For Video Objects Detection, *EURASIP Journal on Applied Signal Processing*, pp. 572-581, Issue 6.
- [8] Irani, M., and Anandan, P. (1998). A Unified Approach to Moving Object Detection in 2D and 3D Scenes, *IEEE Trans. on PAMI*, Vol. 20, pp. 577-589, June 1998.
- [9] Cohen, I. and Medioni, G. (1998). Detecting and Tracking Moving Objects in Video from an Airborne Observer., *DARPA Image Understanding Workshop*, IUW98. Monterey, November 1998
- [10] Cohen, I., Medioni, G. (1999). Detecting and Tracking Moving Objects for Video Surveillance., *Proc. of the IEEE Computer Vision and Pattern Recognition*, 99. Fort Collins, June 1999
- [11] Bell, Walter W., Felzenszwalb, Pedro F. and Huttenlocher, Daniel P. (1999). Detection and Long Term Tracking of Moving Objects in Aerial Video, Computer Science Department, Cornell University. Version: 1999.
- [12] Wongkamchang, P. and Sangveraphunsiri, V. (2008). Control of Inertial Stabilization Systems Using Robust Inverse Dynamics Control and Adaptive Control, *The Thammasat International Journal of Science and Technology*, 2008, Volume 13, No. 2, pp. 20 – 32.
- [13] Malithong, K. and Sangveraphunsiri, V. (2009). Control of Inertial Stabilization Systems Using Robust Inverse Dynamics Control and Adaptive Control, *The 23th Conference of the Mechanical Engineering Network of Thailand*, November 4-7, 2009
- [14] Sangveraphunsiri, V. and Malithong, K. (2010). Control of Inertial Stabilization Systems Using Robust Inverse Dynamics Control and Adaptive Control, *The 6th International Conference on Automotive Engineering (ICAE-6)*, March 29 – April 2, 2010, Thailand
- [15] Nouar, O.-D., Ali, G. and Raphael, C. (2006) Improved Object Tracking With Camshift Algorithm, *Acoustics, Speech and Signal Processing*, ICASSP 2006 Proceedings. 2006 IEEE International Conference on Volume: 2
- [16] Bradski , G. R. (1998). Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, 2nd Quarter, 1998.
- [17] Horn, B. K. P. (1986): Robot vision. MIT Press, 1986.
- [18] Freeman, W. T., Tanaka, K., Ohta, J. and Kyuma, K. (1996). Computer Vision for Computer Games. Int. Conf. On *Automatic Face and Gesture Recognition*, pp.100-105, 1996.
- [20] Comaniciu, D., Ramesh, V. and Meer, P: (2003). Kernel-Based Object Tracking, *IEEE Trans. Pattern Analysis Machine Intell.*, Vol. 25, No. 5, 564-575, 2003