

## Visual Tracking Control for Inertial Stabilization System

Sangveraphunsiri V. and Malithong K.

Department of Mechanical Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand

### Abstract

This paper presents an approach to the image tracking for an inertial stabilization system, comprised of a camera system set up into the aircraft structure. The motion control of the camera system can be divided into two portions; the first portion is a feedback control system, the second portion is the autonomous tracking objects in video sequences. We propose an object tracking algorithm based on a combined HSV color probability histogram model using a CAMshift algorithm that finds the most probable target position in the current frame. The CAMshift algorithm applications can track objects presenting strong modifications of the shape. We can quote the management of the target appearance changes during the sequence. The feedback control loop for the pan-tilt unit would send rate commands to move the camera position in order to keep the target in the center of the camera image. 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.

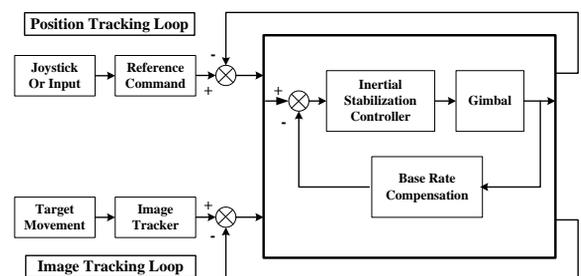
**Keywords:** Visual Object Tracking, CAMshift, Inertial Stabilization system

### 1 Introduction

Unmanned Aerial Vehicles or UAVs are self-piloted or remotely piloted aircraft that can carry cameras, sensors, communications equipment or other payloads. Its initial occupation has been in reconnaissance and surveillance. In Figure 1, the motion control of camera gimbal consists of two portions. The first portion is controlled by a feedback control system in order to move the gimbal according to a reference command and stabilize the gimbal where the camera is attached. The second portion involves the stabilizing of images carried out by an image programming and autonomous object tracking technique. 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.

In the literature, most of the techniques used for this problem deal with a stationary camera or closed world representations which rely on a fixed background or a specific knowledge on the type of actions taking place. For tracking objects from

moving platforms, Cohen [6], [7] proposed the tracking of moving objects in a video stream by the dynamic template from a moving airborne platform. Bell et al. [8] asserted that their system is able to follow multiple objects in aerial video stream while maintaining the identity of each object.



**Figure 1:** Motion control of camera gimbal

Another nonparametric technique is proposed in the Open Source Computer Vision Library Reference Manual “Intel Corporation, 2001” [1], it implements the Continuously Adaptive Mean-shift algorithm (CAMshift) which uses a one-dimensional histogram

to track an object in color images sequences. In [9], it was then stated that the use of a three-dimensional histogram leads to a target localization improvement. In [10-11], the histogram back-projection permits to obtain a probability distribution image which is processed by the iterative CAMShift algorithm in order to find the maximum of the distribution.

Our laboratory (Regional Center of Robotics Technology) has been extending the capabilities of inertial stabilization systems for a number of years. Some Previous capabilities include researching for gimbal development, and the controller design [12-14]. The sliding mode control with indirect stabilization can be used for trajectory tracking in the presence of dynamics uncertainties. This paper focuses on the image tracking with CAMShift algorithm. It is a method used for the real-time tracking of non-rigid objects seen from a moving camera based on color feature, whose statistical distributions characterize the object of interest. Since CAMShift algorithm tracks a combination of colors, it can follow a target through orientation changes. 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 The Tracking System

CAMShift is primarily intended to perform efficient head and face tracking in a perceptual user interface [4]. It is based on an adaptation of Mean Shift that, given a probability density image, finds the mean (mode) of the distribution by iterating in the direction of maximum increase in probability density [1]. The principle of the CAMShift algorithm is given in [1], [2] and [4]. The main focus of our work was the development of control laws to enable the autonomous visual tracking of any target. The method implemented here is primarily based on the CAMShift algorithm. The four steps of the CAMShift algorithm are stated as follows:

1. Select a target and create a color histogram to represent the target.

CAMShift identifies and tracks an object using a histogram (a barchart) of colors. The height of each colored bar indicates how many pixels in an image region have that hue. Hue is one of three values describing a pixel's color in the HSV (Hue, Saturation, and Value) or HSB (Hue, Saturation, Brightness) color model. Figure 2 shows HSV color model.

The histogram used by Bradski [17] consists of the hue channel in the HSV color model; however multidimensional histograms from any colour space may be used. The histogram is quantized into bins, which reduces the computational and space complexity and allows similar colour values to be clustered together. The histogram bins are then scaled between the minimum and maximum probability image intensities using Eq. (2). Figure 3 shows an example histogram produced by a program.

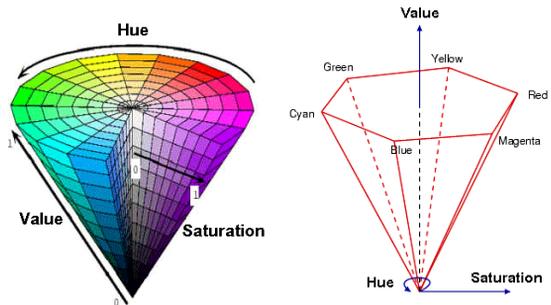


Figure 2: HSV color model.

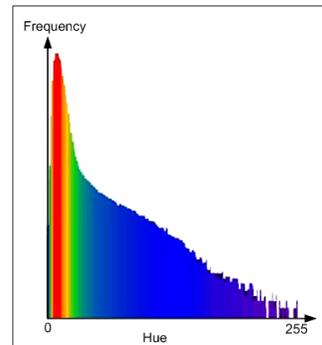


Figure 3: The color histograms

Given that  $m$ -bin histograms are used, we define the  $n$  image pixel locations  $\{x_i\}_{i=1...n}$  and the histogram  $\{\hat{q}\}_{u=1...m}$ . We also define a function  $c: \mathbb{R}^2 \rightarrow \{1...m\}$  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

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

That is, the histogram bin values are rescaled from  $[0, \max(q)]$  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. Calculate a target probability for each pixel in the incoming video frames.

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

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 the probability that the input pixel belongs to the histogram that was used.

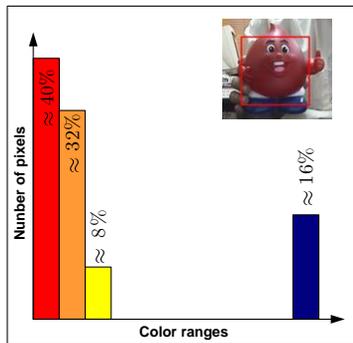


Figure 4: The simple example of the color histogram.

The histogram is created only once, at the start of tracking. Afterwards, it's used to assign a target-probability value to each image pixel in the video frames that follow. For easy understanding, we will use a simple example. Figure 4 shows the bars from a target's histogram. The highest bar accounts for about 40% of the pixels in the region. That means the probability that a pixel selected randomly from the

region would fall into the highest bin is 40%. That's the target probability for a pixel with this hue. The same reasoning indicates that the target probability for the next histogram bin to the left is about 32%. As new video frames arrive, the hue value for each pixel is determined. From that, the target histogram is used to assign a target probability to the pixel. This process is called histogram back-projection.



Figure 5: The normal and target-probability views as CAMshift tracks the coloured doll.

Figure 5 shows the target-probability image in a video frame as CAMshift tracks the coloured doll. White pixels have the highest probability value, and black, the lowest. Gray pixels lie somewhere in the middle.

3. Iterative Mean Shift algorithm to find the center of the target in the probability image and shift the location of the target rectangle in each video frame.

This process of shifting the rectangle to correspond with the center of the target in the probability image, is based on an algorithm called Mean Shift. CAMshift shifts its estimate of the target location, keeping it centered over the area with the highest concentration of white pixels in the target-probability image. It finds the new location by starting at the previous location and computing the center of gravity of the target-probability values within a rectangle. It then shifts the rectangle's centroid to be over the centroid of the target. This is repeated a few times until an optimum alignment is obtained.

For discrete 2D image probability distributions, the mean location (the centroid) within the search window of the discrete probability image is found using moments [1], [17-19]. Given that  $I(x, y)$  is the intensity of the discrete probability image at  $(x, y)$  within the search window.

Compute the zeroth moment

$$M_{00} = \sum_x \sum_y I(x, y) \quad (3)$$

Find the first moment for x and y

$$M_{10} = \sum_x \sum_y xI(x, y) \quad (4)$$

$$M_{01} = \sum_x \sum_y yI(x, y) \quad (5)$$

Compute the mean search window location

$$x_c = \frac{M_{10}}{M_{00}}; y_c = \frac{M_{01}}{M_{00}} \quad (6)$$

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.

4. Calculate the size and angle.

The use of moments to determine the scale and orientation of a distribution in robot and computer vision is described in Horn [18].

The orientation ( $\theta$ ) of the major axis and the scale of the distribution are determined by finding an equivalent rectangle that has the same moments in the x and y axis, as those measured from the 2D probability distribution image [1], [18]. Defining the first and second moments for x and y

$$M_{20} = \sum_x \sum_y x^2 I(x, y) \quad (7)$$

$$M_{02} = \sum_x \sum_y y^2 I(x, y) \quad (8)$$

$$M_{11} = \sum_x \sum_y xy I(x, y) \quad (9)$$

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

$$a = \frac{M_{20}}{M_{00}} - x_c^2 \quad (10)$$

$$b = 2 \left( \frac{M_{11}}{M_{00}} - x_c y_c \right) \quad (11)$$

$$c = \frac{M_{02}}{M_{00}} - y_c^2 \quad (12)$$

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

$$\theta = \frac{1}{2} \tan^{-1} \left( \frac{b}{a - c} \right) \quad (13)$$

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

$$l_1 = \sqrt{\frac{(a + c) + \sqrt{b^2 + (a - c)^2}}{2}} \quad (14)$$

$$l_2 = \sqrt{\frac{(a + c) - \sqrt{b^2 + (a - c)^2}}{2}} \quad (15)$$

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

The primary difference between CAMShift and the Mean Shift algorithm is that CAMShift uses continuously adaptive probability distributions while Mean Shift is based on static distributions, which are not updated unless the target experiences significant changes in size, shape or color. CAMShift adjusts the size and angle of the target rectangle each time it shifts it. It does this by selecting the scale and orientation that are the best fit to the target-probability pixels inside the new rectangle location. After that, the process will go to Step 2.

### 3 The Controller Design

In this paper, the controller structure is divided into two parts – the image tracking loop and the servo control loop.

The image tracking loop has a goal, which is to hold or control the line of sight (LOS) of the gimbal relative to the target. The LOS is the center of the field of view (FOV) of a camera. This loop consists of an image processing and image position controller. The image processing is used to process the signal and find the pixel location of the target. The image position controller attempts to minimize the error between the center of the target and the center of the image. The PI control is used for this loop.

The servo control loop is the inner loop. From the dynamic model, we allow the model to be imprecise. Model imprecision come from actual uncertainty about the plant (e.g., unknown plant parameters) or neglecting structural model. The sliding mode 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.

The block diagram of controller structure is shown in Figure 6. It begins selecting a frame to obtain an input image from the camera. If there is any target detected, it will proceed to the tracking step using the CAMShift algorithm. Next, the current position of the target in the image is compared with the middle position of the image. By doing the scale translation,

the current error can be used as an input value for the servo-control loop. And the servo-control loop uses the corresponding value to vary the speed and

direction of servo and thus making further positional adjustments.

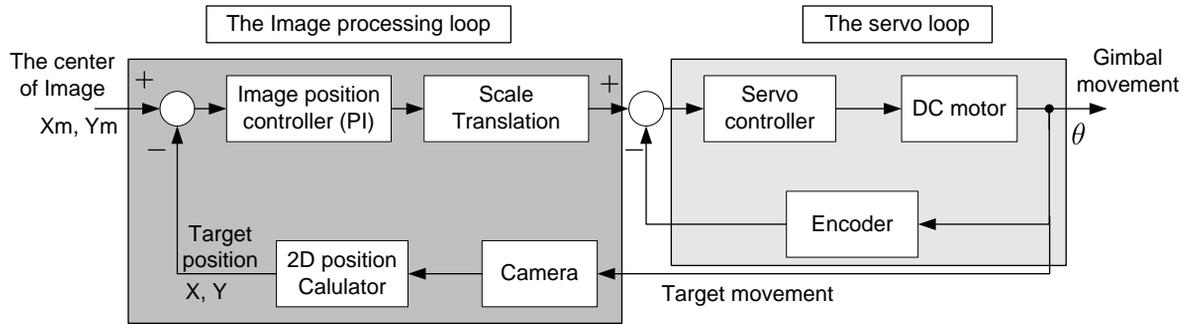


Figure 6: The block diagram of controller structure

### 3.1 The Controller for Tracking Object from Moving 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. The losing target is just a possibility if the target is on the side of image.

Using the CAMshift algorithm, the target is tracked whilst it moves across the view of a video camera. At each frame, an estimated location of the target is made. This estimation is not likely to have a significant degree of inaccuracy. 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.

From the CAMshift algorithm, we know the pixel location of the target on each video frame. If we know the movement of the target in the image, we can generate the reference command of the pan-tilt camera.

If the target moves in X axis (columns of pixels), the camera gimbal must be controlled 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).

### 3.2 The Sliding Mode Control

The dynamic model of the two-axis gimbal can be written in the joint space [12-14]. The friction force,  $F_s$ , will be added to the dynamic equations, so the equation of motion in matrix form can be written as:

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + F_s \operatorname{sgn} \dot{q} + g(q) = \tau \quad (16)$$

In this expression,  $q$  is the vector of joint angles,  $\tau$  is the torque vector applied to the joints,  $D$  is the inertia matrix,  $C(q, \dot{q})$  is the vector of centripetal and Coriolis forces,  $F_s$  is the approximated friction forces.  $\operatorname{sgn} \dot{q} = +1$  when  $\dot{q}$  is positive and  $\operatorname{sgn} \dot{q} = -1$  when  $\dot{q}$  is negative.  $g(q)$  is the vector of gravitational forces and function. For this system, each matrix in the dynamic equations can be written as:

$$D(q) = \begin{bmatrix} \mathbf{I}_{122} + \mathbf{I}_{211} \sin^2 \theta_2 + \mathbf{I}_{233} \cos^2 \theta_2 & 0 \\ 0 & \mathbf{I}_{222} \end{bmatrix}$$

$$C(q, \dot{q}) = \begin{bmatrix} \frac{1}{2} \omega_2 (\mathbf{I}_{211} - \mathbf{I}_{233}) \times \sin(2\theta_2) & \frac{1}{2} \omega_1 (\mathbf{I}_{211} - \mathbf{I}_{233}) \times \sin(2\theta_2) \\ -\frac{1}{2} \omega_1 (\mathbf{I}_{211} - \mathbf{I}_{233}) \times \sin(2\theta_2) & 0 \end{bmatrix}$$

where  $\mathbf{I}_{j,k}$  is a member of row  $j$  and column  $k$  of moment of inertia of link  $i$ .

$$\mathbf{I}_1 = \begin{bmatrix} 0.065 & 0 & 0 \\ 0 & 0.069 & 0 \\ 0 & 0 & 0.07 \end{bmatrix}, \mathbf{I}_2 = \begin{bmatrix} 0.018 & 0 & 0 \\ 0 & 0.024 & 0 \\ 0 & 0 & 0.025 \end{bmatrix} \quad V = \frac{\sigma^T \sigma}{2} \quad (23)$$

$\mathbf{I}_1$  and  $\mathbf{I}_2$  can be obtained by computer aided design software.

From dynamic equation can be written into state space form of a non-linear dynamic system.

$$\dot{x} = f(x) + B(x)u \quad (17)$$

The states are selected as the angular positions and their derivatives

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} q \\ \dot{q} \end{bmatrix} \quad (18)$$

Then, the following state equations are obtained:

$$\dot{x}_1 = x_2 \quad (19)$$

$$\dot{x}_2 = \ddot{q} = \mathbf{D}^{-1}(x_1) \tau - \mathbf{N}(x_1, x_2) \quad (20)$$

$$= -\mathbf{D}^{-1}(x_1)\mathbf{N}(x_1, x_2) + \mathbf{D}^{-1}(x_1)\tau \quad (21)$$

where  $\mathbf{N}(q, \dot{q}) = \mathbf{C}(q, \dot{q})\dot{q} + \mathbf{F}_s \operatorname{sgn} \dot{q} + \mathbf{g}(q)$

For existence and uniqueness of solution of above equation, assume that the functions  $f(x)$  and  $B(x)$  are continuous and sufficiently smooth.

The design of the sliding surface is presented below.

$$\sigma(x, t) = \begin{bmatrix} \sigma_1 \\ \sigma_2 \end{bmatrix} = G_1 e + G_2 \dot{e} = 0. \quad (22)$$

where  $e = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} q_{d1} - q_1 \\ q_{d2} - q_2 \end{bmatrix}$  is position error for each

joint subsystem, The matrices  $G_1$  and  $G_2$  used in this design are

$$G_1 = \begin{bmatrix} c_{11} & 0 \\ 0 & c_{22} \end{bmatrix}, \quad G_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{where } c_{11} \text{ and } c_{22} \text{ are}$$

positive constants.

The derivation of the control involves the selection of a Lyapunov function  $V(\sigma)$  and a desired form for  $\dot{V}$ , the derivative of the Lyapunov function. The selected Lyapunov function is

For the system given by Equation (17), and the sliding surface given by Equation (22), a sufficient condition for the existence of a sliding mode is that

$$\dot{V} = \sigma^T \dot{\sigma} < 0 \quad (24)$$

The derivative of the Lyapunov function will be negative definite, and this will ensure stability. A Stronger condition, guaranteeing an ideal sliding motion, is the  $\eta$ -reachability condition given by

$$\dot{V} = \sigma^T \dot{\sigma} < -\eta|\sigma| \quad (25)$$

where  $\eta$  is a strictly positive constant.

In a neighborhood of  $\sigma = 0$ , This is also a condition for reachability. It is desired that

$$\dot{V} = -\sigma^T K \operatorname{sgn} \sigma. \quad (26)$$

Where  $K$  is a positive-definite matrix. Thus, The last two equations together lead to

$$\sigma^T (K \operatorname{sgn} \sigma + \dot{\sigma}) = 0 \quad (27)$$

A solution for the equation above is

$$\dot{\sigma} = -K \operatorname{sgn} \sigma \quad (28)$$

The expression for the derivative for the sliding function is

$$\dot{\sigma} = G_1 \dot{e} + \ddot{e} \quad (29)$$

$$\dot{\sigma} = G_1 \dot{e} + \ddot{q}_d - \ddot{q} \quad (30)$$

$$\dot{\sigma} = G_1 \dot{e} + \ddot{q}_d - \mathbf{D}^{-1}(x_1)\tau - \mathbf{D}^{-1}(x_1)\mathbf{N}(x_1, x_2) \quad (31)$$

Thus, from Equation (28) and Equation (31), we have that

$$-K \operatorname{sgn} \sigma = G_1 \dot{e} + \ddot{q}_d - \mathbf{D}^{-1}(x_1)\tau + \mathbf{D}^{-1}(x_1)\mathbf{N}(x_1, x_2) \quad (32)$$

$$\tau = \mathbf{D}(x_1) G_1 \dot{e} + \ddot{q}_d + \mathbf{N}(x_1, x_2) + \mathbf{D}(x_1) K sgn \sigma \quad (33)$$

$$sat(\sigma_i) = \begin{cases} +1 & \text{if } \sigma_i > \Phi_i \\ \frac{\sigma_i}{\Phi_i} & \text{if } |\sigma_i| \leq \Phi_i \\ -1 & \text{if } \sigma_i < -\Phi_i \end{cases} \quad (34)$$

From Equation (33), the sliding mode control may cause a significant problem of chattering. The signum function in Equation (33) is changed by saturation function. This method can eliminate chattering effect of the control signal.

Where  $\Phi_i > 0$  is a switching boundary value in joint  $i$ .

Figure 7 shows the block diagram of the sliding mode control and the real-time image tracking is added to detect the non-rigid target seen from the camera gimbal.

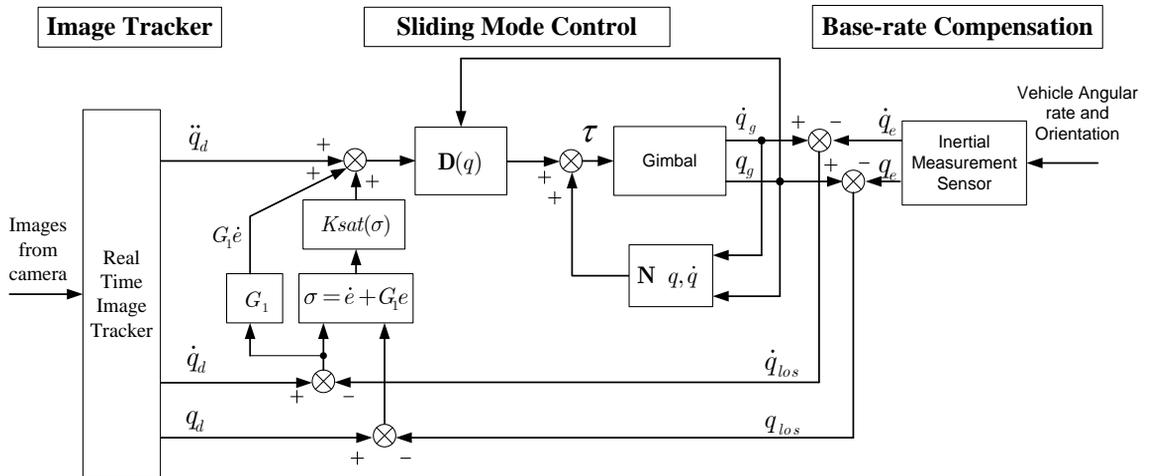


Figure 7: The block diagram of the Sliding mode control with the real-time image tracking

#### 4 Experiment and result

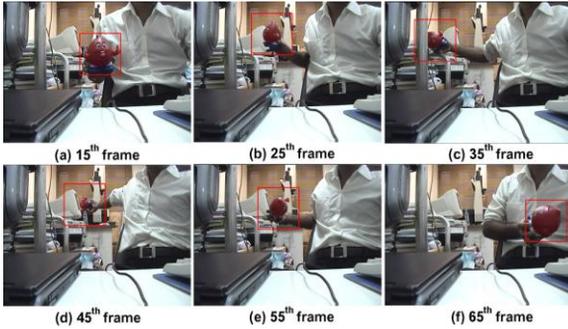
Our system consists of three modules - the image tracking, the position control and the base-rate compensation module. For the first experiment, we conducted experiments to test the image tracking module. The camera gimbal was stationary. In the second experiment, the combination of the image tracking and position control module was carried out experiment. The camera gimbal was controlled to track the target. In order to prove the validity of the overall system, the gimbal is set up into the mobile platform in the last experiment. We have applied our method on various video sequences. The 640x480 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 colour 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

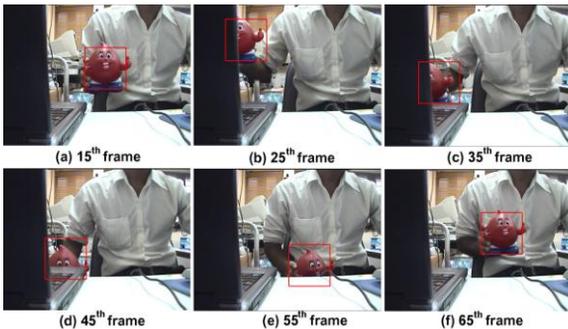
This section describes the results obtained from experiments with the object tracking module. The camera gimbal is fixed.

Figure 8-9 give two examples, showing the object tracking through occlusion as well as the scale and orientation estimation.



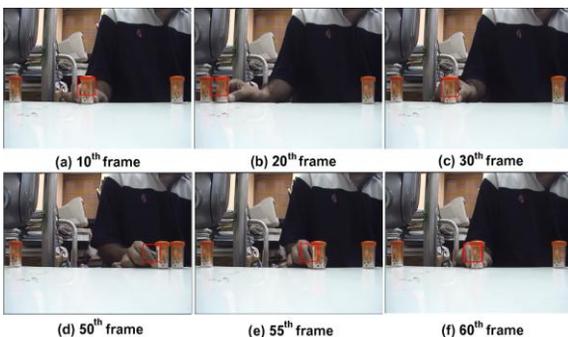
**Figure 8:** Tracking the coloured doll

By performing several scaling operations automatically, the tracker furthermore showed its capability to react to different scaling requirements correctly. The coloured doll was tracked successfully throughout the time period. In Figure 9, the tracker can track the coloured doll through occlusion if the tracked coloured doll is only partially hidden.



**Figure 9:** Tracking the coloured doll through occlusion

Figure 10 demonstrates the similar object-tracking. In this situation the tracked target looks very similar to other objects in the video sequence.



**Figure 10:** Tracking the similar objects

For the experiment shown in Figure 10, the tracked bottle looks very similar to the other bottles. The CAMshift tracker kept up with the target bottle to be tracked until the target bottle is moved within close proximity of the other bottle. The miss tracking problem occurs when other objects with the same characteristics as the target are in the region of interest (ROI) of the tracked object. The tracker is unable to reliably and consistently identify the target as both objects possess the same characteristic identifying properties. Kalman estimator has been used to solve this problem. The tracker can decide what to be the target because Kalman filter estimates the target location in the future.

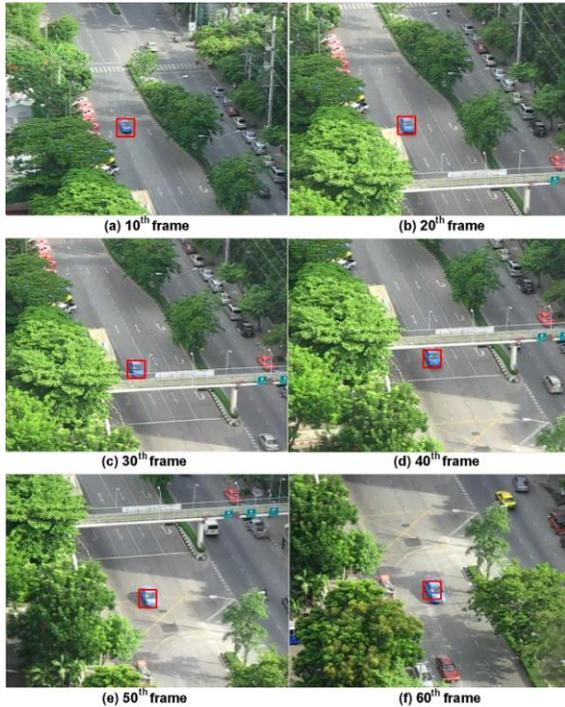
If any tracking errors are present, say because the feature is hidden for some frames, or fault detection to fool the matcher, the operator can select the correct target again. Tracking portions will be restart.

#### 4.2 Tracking with controlled camera

The objective of this section is maintaining the camera position in order to keep the target in the center of the camera image. Sliding mode 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.

Camera motion will blur the captured image and further bring difficulties in image processing. The distance between the target to camera and color space will significantly affect the performance of the system. Color space is more efficient, therefore a variety of color space have been used in tracking. The difference in distance between the camera and the target will result in the difference in the change of error even with the same horizontal or vertical displacement speed. This makes the system even more difficult to control.

Figure 11 demonstrates tracking with the use of a pan-tilt controlled camera. The experimental results show the gimbal can be controlled in order to keep a car in the center of the camera image.



**Figure 11:** Tracking a car, using the developed controller and image tracking system.

### 4.3 Tracking Objects from Moving Platforms

In field tests, the camera gimbal was mounted on the aircraft structure. In the laboratory, the camera gimbal was mounted on the trust frame and hand truck. The video is recorded from the top floor on the high building (70 meters) while the hand truck is push forward. In this section, the objective of the control is similar to that described in section 4.2, but the gimbal is set up into the mobile platform, as illustrated in Figure 12.



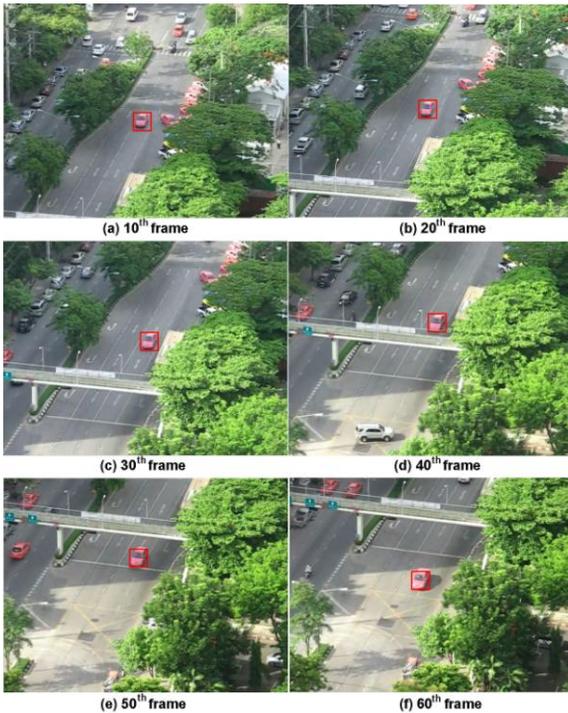
**Figure 12:** The gimbal mounted on the mobile platform



**Figure 13:** Map for the experiment.

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 simulated by moving the base.

In figure 13, distance between a car to the camera gimbal is about 250-300 m. The car was driven at approximately 10 km/h. The hand truck was push forward at a speed of 1 m/s.



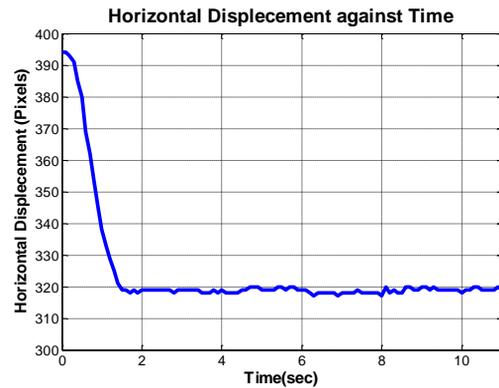
**Figure 14:** Tracking the target from mobile platform.

In figure 14, the experimental results show that the sliding mode control with the image tracking control performs effectively for our inertial stabilization system, and is a promising controller. The gimbal can be controlled in order to keep the target in the center of the camera image.

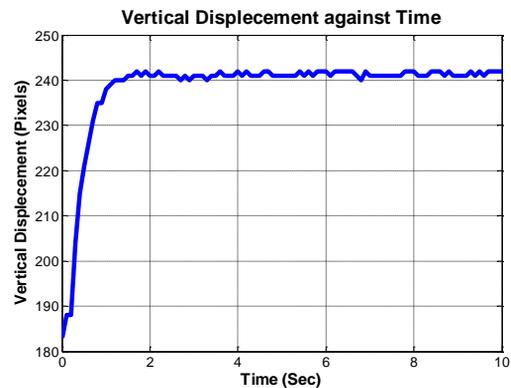
For the target selection, if the position of the target is selected far from the center of frame, the system will take many times to move a tracked target to the center of frame.

Our system is able to capture the video with horizontal rotation angle  $-60$  to  $60$  degree and vertical rotation angle  $-35$  to  $35$  degree with maximum rotation speed  $60$  degree per second for both rotations.

Due to the width and height of the frame size is  $640 \times 480$  pixels, hence the centerd coordinate of horizontal coordinate and vertical coordinate are  $320$  pixels and  $240$  pixels respectively. Figure 15 shows the system takes approximately  $2$  seconds for a tracked car on the right side of the frame to be moved to the centre. And figure 16 shows the system takes approximately  $2$  seconds for a tracked car in top side of the frame to be moved to the center of frame.



**Figure 15:** Horizontal displacement of the target in the sequence frames.



**Figure 16:** Vertical displacement of the target in the sequence frames.

## 5 Conclusions

This paper proposes an object tracking approach using color image sequences, based on the CAMShift algorithm. The CAMShift algorithm shifts its estimate of the target location, so that reference command for the pan-tilt camera based on the target location can be generated. The details of the controllers, the sliding mode control 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 targets were tracked  $90\%$  of the time successfully. 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.

## Acknowledgments

This work is sponsored by Chulalongkorn University under Chulalongkorn University Centenary Academic Development Project.

## 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] 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.
- [7] 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.
- [8] 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.
- [9] John, A. G., Richard Xu, Y. D. and Jin Jesse S., 2004. *Object Tracking Using CamShift Algorithm and Multiple Quantized Feature Spaces*, Inc. Australian Computer Society, vol.36.
- [10] Cheng, Y., 1995. *MeanShift, mode seeking, and clustering*, IEEE Trans. on PAMI, vol.17, pp.790-799.
- [11] Bradski, G. R., 1998. "Computer vision face tracking for use in a perceptual user interface", Intel Technology Journal, 2nd Quarter.
- [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. *Robust Inverse Dynamics and Sliding Mode Control for Inertial Stabilization Systems*, Asian International Journal of Science and Technology in Production and Manufacturing Engineering Vol. 2, No.4, October - December , 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] 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
- [17] Bradski , G. R., 1998. *Computer vision face tracking for use in a perceptual user interface.*, Intel Technology Journal, 2nd Quarter, 1998.
- [18] Horn, B. K. P., 1986: *Robot vision.*, MIT Press, 1986.
- [19] 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.