

# FACE TRACKING SYSTEM USING A WEB CAMERA

Resmana Lim, Thiang & Iwan Nyoto S.

Electrical Engineering Department  
Petra Christian University  
Siwalankerto 121-131, Surabaya – Indonesia  
Phone: +62-31-8439040, Fax: +62-31-8436418,  
Email: {resmana;thiang}@petra.ac.id

## ABSTRACT

*The paper describes a development of a pan-tilt camera tracking system using a motorized web camera connected to a PC. The main goal of this project is to provide camera tracking capability to locate human face in its close proximity by analyzing color of skin human face. The camera try to continuously track the human face and maintain the face to be located within the center of the screen. The system achieved a real time performance of about 15 frames per second.*

**Keywords:** human face tracking, Camshift algorithm, pan-tilt camera tracking, video tracking, OpenCV.

## 1. INTRODUCTION

The project described in this paper is a part of a program to develop a human recognition system in which the computer is given the ability to sense and recognize the human face, expression or gesture. The first step in this project is to develop a real-time face tracking system that is robust under varying lighting/noise conditions. The goal of this project is to develop a robust real time face tracking system that operates on a Windows platform using an ordinary PC with a low cost USB web-camera.

A pan-tilt motor has been developed and integrated to a web-camera giving the ability to the camera to move in 2 degree of freedom. The Camshift algorithm [1] is used to follow and track the human face based on its skin color. It gives information of the face center in the 2D image which is further the information will be used to move the camera to maintain the face image to be located in the center of the screen.

The remainder of this paper is organized as follows. First, the system description is presented in section 2. Then, the overview of the Camshift algorithm is presented in section 3. In section 4, results on various illumination/noise condition are presented. Finally, a conclusion and directions for future work are briefly covered in the last section.

## 2. SYSTEM DESCRIPTION

The goal of this project is to develop a pan-tilt camera tracking system that uses the available CAMSHIFT Algorithm [1] developed by Intel to track human face under varying lighting conditions.

Specifically, the intension is to develop an application that:

- is capable of tracking human face
- runs in real time
- implements the CAMSHIFT algorithm developed by Intel and supported in OpenCV (Open Source Computer Vision library)
- Integrates the position of tracked object (result of the CAMSHIFT algorithm) with the pan-tilt motor control to move the camera looking the human face.
- runs on an ordinary PC equipped with a USB web camera
- is fairly robust under varying lighting conditions
- can handle image noise

The setup of the system is as pictured on figure 1.

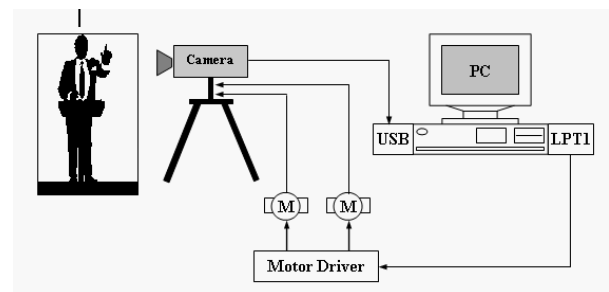


Figure 1. The System Setup.

The construction of pan-tilt motorized web-camera is as follows:



Figure 2. Pan-Tilt Camera Motor Configuration

### 2.1 Motor driver circuit

In the project, the mini DC motors (12 Volt, 350 mA) are used and a motor driver circuit has been developed using a IC L293 D. The IC is a DC motor driver with *push - pull four channel driver* configuration. It capable to drive current of 600 mA per channel simultaneously and can be controlled by TTL - Compatible Logic. In the project, a PC printer port is used to control the motor driver as depicted on figure 3.

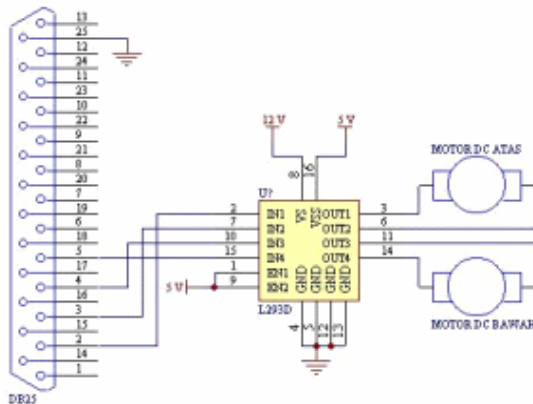


Figure 3. Pan-Tilt DC Motor Driver.

## 3. CAMSHIFT ALGORITHM

The Continuously Adaptive Mean-Shift Algorithm (CAMSHIFT) is a color-based tracking algorithm that is fairly robust and extremely inexpensive in terms of

computer cycles. Thus, the challenge of this project is to develop a pan-tilt camera tracking application that uses the CAMSHIFT Algorithm developed by Intel [1] to track human faces under varying lighting/noise conditions.

CAMSHIFT was designed to handle precise tracking of facial location on a non-moving camera at the range of less than one meter. The algorithm relies heavily on hue value of the human flesh. Human skin forms a very tight cluster in color space even when different races are considered [1]. CAMSHIFT operates on color probability image and applies a non-parametric gradient density claiming called mean shift algorithm to re-center its operating window. By windowing the image, the algorithm will track the location of the face and effectively reduce color segmentation noise by ignoring value outside its search window. Using color model greatly cut down the fault track in noisy environment since color noise has a low probability of being flesh color [2]. The result shows that the algorithm can tolerate up to 30% of Gaussian noise added within reasonable performance degradation. Thus, CAMSHIFT is able to handle noisy image without the need for extra filtering or adaptive smoothing [3].

### 3.1 Color Probability Image

A probability distribution image of the desired color in the video scene must be created, in order to use CAMSHIFT to track colored objects in a video scene. In order to do this, a model of the desired hue is created using a color histogram. A HSV(Hue-Saturation-Value) conversion is done on all of the RGB pixels, and a histogram is created using the H channel component. HSV space separates out hue (color) from saturation (how concentrated the color is) and from brightness (V).

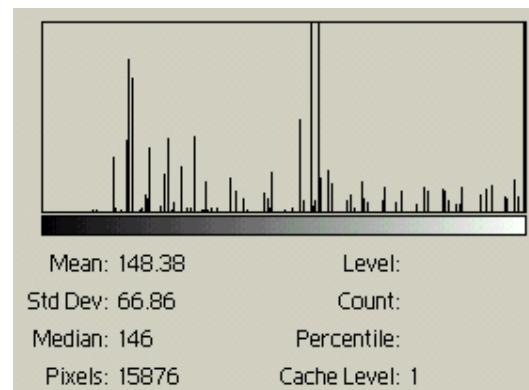


Figure 4. Sample of Image Histogram

For the project of camera tracking to follow human face via a flesh color model, flesh areas from the user are sampled by drawing a box around an area of the face. The

hues derived from flesh pixels in the image are sampled from the H channel and binned into an 1D histogram. The histogram is saved for future use. More robust histograms may be made by sampling flesh hues from multiple people/races. Even simple flesh histograms tend to work well with a wide variety of people without having to be updated. A common misconception is that different color models are needed for different races of people, for example, for blacks and whites. This is not true. Except for albinos, humans are all the same color (hue). Dark-skinned people simply have greater flesh color saturation than light-skinned people, and this is separated out in the HSV color system and ignored in our flesh-tracking color model.

During operation, the stored color histogram is used as a model, or lookup table, to convert incoming video pixels to a corresponding probability of the desired color. This is done for each video frame. Using this method, probabilities range in discrete steps from zero to one. Tracking is then done using the CAMSHIFT algorithm on this probability image.

### 3.2 CAMSHIFT Calculation [1]

The closest existing algorithm to CAMSHIFT is known as the mean shift algorithm. The mean shift algorithm is a technique that climbs the gradient of a probability distribution to find the nearest peak.

#### The Mean Shift Algorithm calculation:

- Choose a search window size.
- Choose the initial location of the search window.
- Compute the mean location in the search window.
- Center the search window at the mean location computed in Step 3.
- Repeat Steps 3 and 4 until convergence (or until the mean location moves less than a preset threshold).

For discrete 2D image probability distributions, the mean location (the centroid) within the search window (Steps 3 and 4 above) is found as follows:

Find the zero-th moment

$$M_{00} = \sum_x \sum_y I(x, y). \quad (1)$$

Find the first moment for  $x$  and  $y$

$$M_{10} = \sum_x \sum_y xI(x, y); \quad M_{01} = \sum_x \sum_y yI(x, y). \quad (2)$$

Then the mean search window location (the centroid) is

$$x_c = \frac{M_{10}}{M_{00}}; \quad y_c = \frac{M_{01}}{M_{00}}; \quad (3)$$

where  $I(x, y)$  is the pixel (probability) value at position  $(x, y)$  in the image, and  $x$  and  $y$  range over the search window.

Unlike the Mean Shift algorithm, which is designed for static color probability distributions, CAMSHIFT is designed for dynamically changing color probability distributions. These occur when objects in video sequences are being tracked and the object moves so that the size and location of the probability distribution changes in time. The CAMSHIFT algorithm adjusts the search window size in the course of its operation. CAMSHIFT relies on the zero-th moment information, extracted as part of the internal workings of the algorithm, to continuously adapt its window size within or over each video frame. One can think of the zero-th moment as the distribution "area" found under the search window. Thus, window height and width, are set to a function of the zero-th moment found during search.

#### The Continuously Adaptive Mean Shift Algorithm calculation:

- First, set the calculation region of the probability distribution to the whole image.
- Choose the initial location of the 2D mean shift search window.
- Calculate the color probability distribution in the 2D region centered at the search window location in an area slightly larger than the mean shift window size.
- Mean shift to convergence or for a set number of iterations. Store the zero-th moment (area or size) and mean location.
- For the next video frame, center the search window at the mean location stored in Step 4 and set the window size to a function of the zero-th moment found there. Go to Step 3.

The complete implementation of the Camshift algorithm is available in the Intel OpenCV library. In the project, we used the Camshift algorithm and incorporated it with our own application of the pan-tilt camera motor control to track the human face.

#### 4. EXPERIMENT RESULTS

The system was implemented in the Windows environment. Several libraries are required to build the complete application which as follows:

- Microsoft Visual C++ standard libraries
- Microsoft Win32 API
- Microsoft DirectX Software Developers Kit 8.0a
- Intel Image Processing Library (IPL) v2.5
- Intel Open Source Computer Vision Library (OpenCV)

Microsoft Visual C++ and the Win32 API are used to build the foundation of the application, and provides a built-in windows interface (push buttons, radio buttons, file view, etc.). Microsoft's DirectX SDK provides the library functions for DirectX 8.0. Microsoft DirectX is a set of low-level application programming interfaces (APIs) for creating games and other high-performance multimedia applications. It includes support for two-dimensional (2-D) and three-dimensional (3-D) graphics, sound effects and music, input devices, and support for networked applications such as multiplayer games. In particular, DirectShow library functions were used in the project. The Microsoft DirectShow application programming interface (API) is a media-streaming architecture for Windows. Using DirectShow, applications can perform high-quality video and audio playback or capture. This project used DirectShow functions to grab video from the USB webcamera, process it using the CAMSHIFT algorithm available in the OpenCV, and display it on the screen.

The system is tested on various color video (size: 320x240 pixels per frame) with various illumination/noise conditions. Figure 5 demonstrates the performance of our human face tracking system with the presence of varying illumination and noise. In this experiment, the Gaussian noise is generated and added to the video stream meanwhile the pan-tilt camera motors are turned off. The results show that the Camshift algorithm is quite robust to track the moving human face in the presence of the varying illumination and noise. We can see here, The human face could be tracked with reasonable good result.

Figure 6 shows the pan-tilt camera tracking performance. Both camera motors are turned on and the camera is moving to try to track the face. The intention is that the face is to be placed in the center of the screen image. The pan-tilt motor control algorithm used in the experiment is just a simple proportional control. That's why the performance is not so good. Here it can be seen that the center of face is not located exactly in the center of image instead of just around the center square. The tracking performance achieved a real time of 15 frame per

second on PC Pentium III 600 MHz.

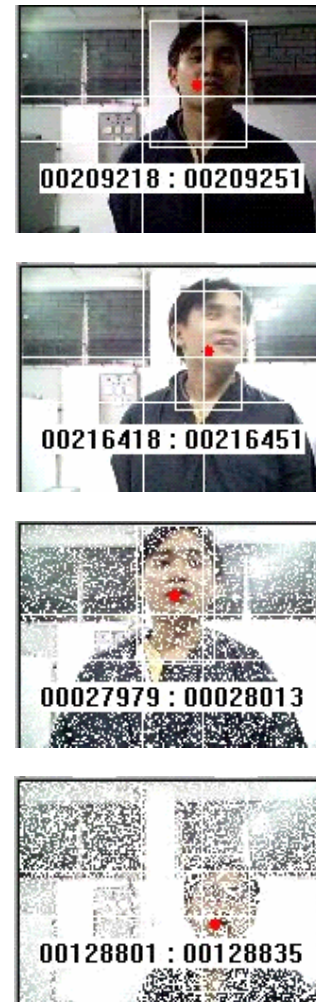


Figure 5. Tracking performance in presence of varying illumination and noise.



Figure 6. Pan-Tilt camera tracking performance.

## 5. CONCLUSIONS

We have developed a pan-tilt real time camera tracking to locate the human face. The performance of the system was demonstrated on various video captured by a web camera containing moving human face. The results are quite promising and achieved a real time tracking of 15 frame per second. The future work should be dedicated to incorporate advanced control methods so that the tracking performance and accuracy will be improved. From the results of the experiment, we conclude that the proposed system has a good prospect and should be considered to be incorporated in the active face recognition systems.

## REFERENCES

- [1] Gary R. Bradski, "Computer Vision Face Tracking For Use in a Perceptual User Interface". Intel Technology Journal, pp. 1-15, Q2 1998.
- [2] D. Comaniciu and P. Meer, "Robust Analysis of Feature Spaces: Color Image Segmentation," CVPR'97, pp. 750-755, 1997.
- [3] K. Sobottka and I. Pitas, "Segmentation and tracking of faces in color images," Proc. Of the Second Intl.

Conf. On Auto. Face and Gesture Recognition, pp. 236-241, 1996.