



142  
820  
THS



1  
2011

This is to certify that the  
thesis entitled

VISUAL TRACKING IN MANUFACTURING PROCESS

presented by

RUI ZHANG

has been accepted towards fulfillment  
of the requirements for the

M.S. degree in Electrical Engineering



Major Professor's Signature

8/27/10

Date

*MSU is an Affirmative Action/Equal Opportunity Employer*

LIBRARY  
Michigan State  
University

**PLACE IN RETURN BOX** to remove this checkout from your record.  
**TO AVOID FINES** return on or before date due.  
**MAY BE RECALLED** with earlier due date if requested.

| DATE DUE               | DATE DUE | DATE DUE |
|------------------------|----------|----------|
| AUG 31 2013<br>05 04 1 |          |          |
|                        |          |          |
|                        |          |          |
|                        |          |          |
|                        |          |          |
|                        |          |          |
|                        |          |          |
|                        |          |          |
|                        |          |          |
|                        |          |          |

# **VISUAL TRACKING IN MANUFACTURING PROCESS**

By

Rui Zhang

A THESIS

Submitted to  
Michigan State University  
in partial fulfillment of the requirements  
for the degree of

**MASTER OF SCIENCE**

Electrical Engineering

2010



## ABSTRACT

### VISUAL TRACKING IN MANUFACTURING PROCESS

By

Rui Zhang

Visual tracking system is a very active research area. The application of visual tracking system is not limited to research but expended to manufacture process automation. A complete visual tracking system can provide useful information based on the analysis of the input image. The core of visual tracking system is its image processing and analysis procedure. In this research a new hardware system set up is presented as well as a new approach of edge detection to achieve a more accurate tracking. The new system set up includes the utilization of server machines to separate computation process from display and operation process which are the main function of client PC. This new system set up greatly improved system reliability and enabled the system to handle multiple camera multiple object tracking. The new approach of edge detection overcomes the inconsistent lighting condition in manufacture process and generates target object properties for the manufacturer. Human-machine interaction is also introduced to this tracking system to improve reliability and simplifies the problem.

## TABLE OF CONTENTS

|                                   |    |
|-----------------------------------|----|
| LIST OF FIGURES.....              | iv |
| INTRODUCTION.....                 | 1  |
| LITERATURE REVIEW.....            | 5  |
| PROBLEM STATEMENT.....            | 7  |
| SYSTEM OVERVIEW.....              | 10 |
| Hardware architecture.....        | 10 |
| Software architecture.....        | 13 |
| VISUAL TRACKING.....              | 16 |
| Noise removal.....                | 16 |
| Segmentation.....                 | 17 |
| Unwanted features within ROI..... | 20 |
| Edge fitting.....                 | 21 |
| Calibration.....                  | 22 |
| RESULT AND ANALYSIS.....          | 26 |
| CONCLUSION AND FUTURE WORK.....   | 40 |
| REFERENCES.....                   | 43 |

## **LIST OF FIGURES**

|  |    |
|--|----|
| Figure 1 Sample tin bath images .....                                  | 3  |
| Figure 2 Hardware architecture .....                                   | 12 |
| Figure 3 Software architecture .....                                   | 14 |
| Figure 4 Glass manufacture process.....                                | 26 |
| Figure 5 Sample ROI image and tin bath image.....                      | 28 |
| Figure 6 Adaptive threshold with and without noise removal .....       | 30 |
| Figure 7 Global threshold.Result.....                                  | 32 |
| Figure 8 Opening and Closing Operation Effects .....                   | 33 |
| Figure 9 Registry of non-interest edges .....                          | 35 |
| Figure 10 Result of XOR between registry and new processed frame ..... | 35 |
| Figure 11 ROI and line scan sample .....                               | 36 |
| Figure 12 Result of fitted edge .....                                  | 38 |
| Figure 13 Calibration result....                                       | 39 |

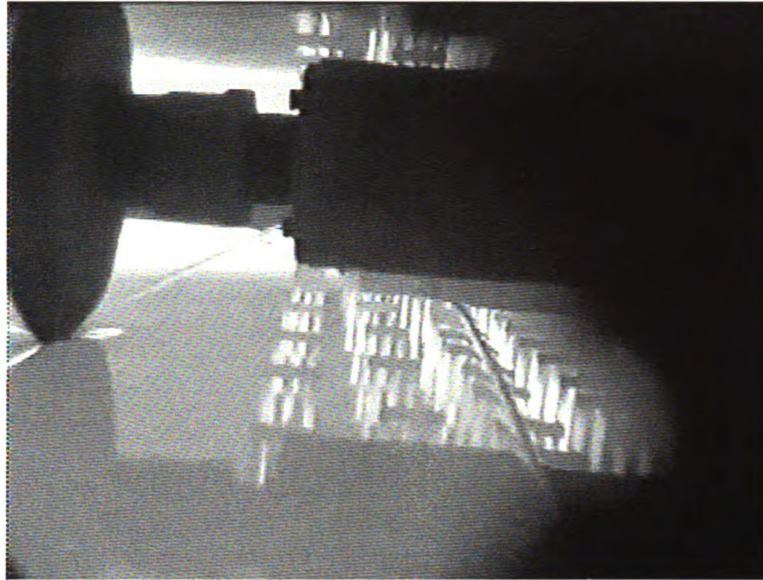


## **1. Introduction**

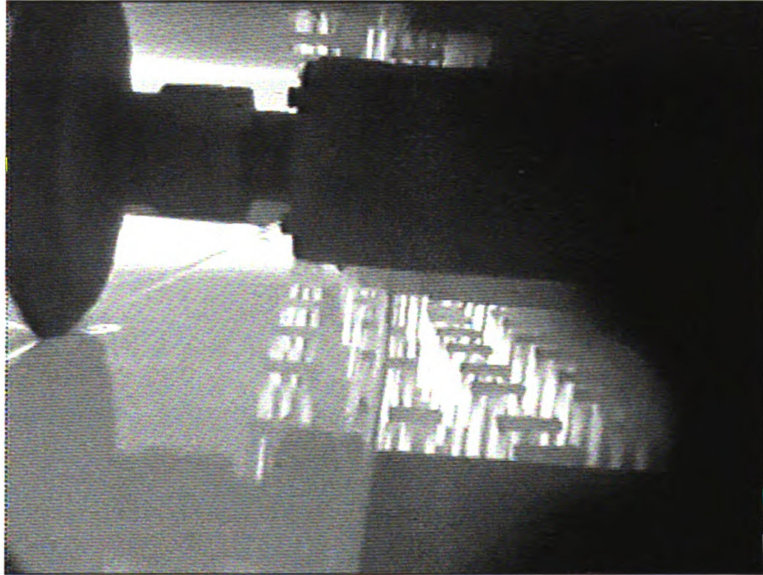
Visual tracking system is a very active research topic. Its primary function is to detect and track the motion of certain object in image sequences. It can be applied in many fields such as surveillance, robot control, human gesture recognition, medical imaging and manufacture process automation. Real-time Visual tracking system is becoming more and more popular in manufacture process automation since it provides a necessary feedback measurement which can greatly improve the accuracy of process automation [1][2]. Visual tracking has many unique advantages comparing to other tracking methods in both functionality and practicality. It can capture rich information such as color, shape, and size which are very useful at recognizing object. It is also usually cheap and easy to implement which is important to industrial applications.

However, visual tracking system still has lots of problems which preventing it to play a bigger part in manufacture process. The performance of visual tracking system highly depends on the efficiency and accuracy of the image processing technique being used. Generally, image processing has a very high demand of computation power. With complicated image processing techniques, the system suffers less from error rate. However, it also increases the time needed to complete the analysis of an image and fail to meet the on-line requirement of manufacture process. Therefore, in order to achieve real-time tracking, either the complexity of image processing is reduced so that the overall process time is

shorted, or very expensive equipment is purchased to provide enough computation power. In addition to that, one major problem the tracking system faces is that the inconsistent lighting condition of manufacture environment. Inconsistent lighting condition can cause the camera either fail to grab a clear image to be analyzed or capture a large variation of illumination which can cause malfunction of image processing and eventually fails the tracking system. People tried to solve this problem by adding additional lighting source so that camera can catch a clearer image and illumination variance is reduced. Unfortunately, sometime it is impossible to add another lighting source to the manufacture process and images caught by the camera becomes inconsistent in its overall brightness and contrast. For example, figure 1 shows two camera images retrieved from a glass production tin bath in which the average temperature is 1475 degree Fahrenheit. There is no light bulb can sustain that heat and the picture is in low contrast and contains lots of noise.



(a)



(b)

Figure 1 Sample tin bath images

In order to provide a more generic solution to tracking system, a new hardware set up and a new image processing method on image segmentation is presented in this research. The new hardware set up greatly improved the capability of the tracking system which can handle multiply camera multiple object tracking. It also improved the hardware system's reliability and provided the



possibility of remote access and control. The new image processing procedure simplified the complexity of the algorithm and reduced the computation power and time needed to ensure that this system can handle real-time tasks.

## **2. Literature review**

There exist many visual tracking systems in the literature [2] [4] [5] [6] [7] [8], however, none of them is focused on low contrast image-based visual tracking. In the above glass production example, the main purpose is to track the edge of liquid glass flowing above liquid tin and calculate the distance from the edge to machine tip. The contrast difference between them is not significant and is surrounded by a lot of other features which are very hard to be distinguished.

Using CAD model based visual tracking [2] can easily find the machine tip because a determined shape is available and the rigid body will not alter a lot just because of the camera is pointing from a difference angle. However, this method fails to deal with target object without a defined CAD model.

Others use template matching [1] [6] which in general works fine as long as the target object doesn't change its shape over time. The draw back of template matching is that it requires a lot of computation resource and usually takes longer to finish the task of locating the object since it performs a search over the entire image. Another problem of template matching is that it cannot provide any further properties of the target object since only the template is located but not the object within it. Further analysis method is required in order to get more detailed property of the target object.

Lack of attention on hardware architecture design also imposed restrictions on the system. A normal approach usually includes all the tracking system

function on a single machine which has a very limited computation power.

Sawasaki [12] and others have noticed this issue and started separating the image processing procedures from other functions of the tracking system. This idea of distributing computation is the right path to take and inspired the system hardware architecture set up presented in this research. In this research, a new visual tracking system set up which has more efficiency and ability to analyze low contrast noisy images is presented. This system can handle more generic situations of manufacture process.



### **3. Problem statement**

From section 1 and section 2, we come to the following conclusions of the problems that the current visual tracking system faces. The current visual tracking system can handle well defined simple rigid object tracking under stable lighting condition with evenly distributed illumination. However, the current tracking system cannot handle object tracking under uneven distribution of illumination. The visual tracking system also fails to perform its task under unstable lighting condition. The main cause is that the core image processing algorithm used in visual tracking system is not targeted to real world applications. Because the lighting condition is usually much better at a research environment than a manufacture plant, these factors is not taken into consideration. At current state, the tracking system does not meet the hardware reliability requirements for manufacture process which usually require system running at 24/7. In addition to that, the tracking system cannot use more sophisticated algorithm for image processing due to the limitation of hardware's computation power.

These are the main problems we want to solve in this research. The difficulties in overcoming these questions involve hardware architecture modification and software algorithm improvement. On hardware side, the challenge is to increase the computation ability and hardware reliability while maintaining the cost of system to an acceptable level. It is also the manufacture's interest to give this system remote control ability so that the flexibility of system

set up is maximized. Another challenge in hardware set up is to minimize the electromagnetic interference. Usually in a manufacture plant, many power cables run around the plant to provide power to production equipments. These energy cables have a high current running through them and will generate electromagnetic fields which will interference with the video signal running in cable from camera to frame grabber. This effect will degrade the image quality at the frame grabber end and will be worse if the video signal goes through a longer cable which has a higher risk of being exposed to electromagnetic field interference.

The solution to these problems is proposed in the following sections. In order to reduce the meet manufacture hardware reliability requirement, server machine is used for image acquiring, image processing and information generation. To reduce interference of electromagnetic field to video signal and include remote control ability, network communication is introduced and a client PC machine is included for display and human machine interaction. The network communication provides the option of putting server closer to camera and reduces video signal cable length in order to minimize effect of electromagnetic field interference. The image processing algorithm that is used on server machine includes additional filter to reduce high frequency noise. In addition to that, an adaptive threshold method is utilized to reduce uneven distributed illumination and unstable lighting conditions. This new approach is very efficient and has a low cost of computation

power which increased the ability to add more cameras to the tracking system to best utilize the powerful server machine.

## **4. System overview**

### ***4.1 Hardware architecture***

The general system set up is shown in figure 2. The main idea is to distribute computation load to reduce the time needed for image processing and information retrieval and improve the overall system efficiency by maximizing the computation ability.

In the system set up, cameras will catch image continuously and send them to a splitter. Increasing the number of camera used can greatly increase adaptability of the system. The cameras can set up in two ways. In the first set up, each camera focuses on different view areas and act like end point sensors. Therefore, all together, the cameras can form a surveillance network with maximum coverage. Another way to set up the cameras is to pair them up and each pair is configured to stereo vision which is very effective in generating 3D image and avoiding camera calibration.

From the splitter, the video signal is separated to 3 feeds entering frame grabber, encoder and monitor respectively. The splitter must have the ability to strengthen the video signals otherwise the separated video signal will be too weak for the devices at the next step to pick up. The frame grabber will grab an image for the software application containing the image processing routines to perform necessary steps to analyze the current frame of image and produce information for next step. The information produced by image processing routines is then

either stored to a database or sent to client machine through Ethernet connection using TCP/IP protocol. The TCP/IP protocol provides reliable communication between the server and client and guarantee that no information loss during data transmission. The second output from splitter goes into the encoder and it will compress the video signal and send them directly to client through network connection. The client has decoder installed and the image is decoded for display or other simple tasks at the client machine. User at the client machine can also use a graphic user interface to send out command to the server machine. A direct connection from splitter to client is available, but longer cable is needed for transmitting the video signal. Usually there is lots of high energy cables used in manufacture plant and most of them are running strong current which generate strong electromagnetic field can interference with the video transmitted cable and degrade its quality. With the use of encoder and decoder, the image can be transmitted over the internet with slight loss of information due to quantization. Since the quantization effect only appears on the client machine where no image processing is performed, the effect is minimized. The third feed from splitter goes directly to analog monitor who performs as a backup display in case the server machine malfunctions and stop video transmitting through the network.

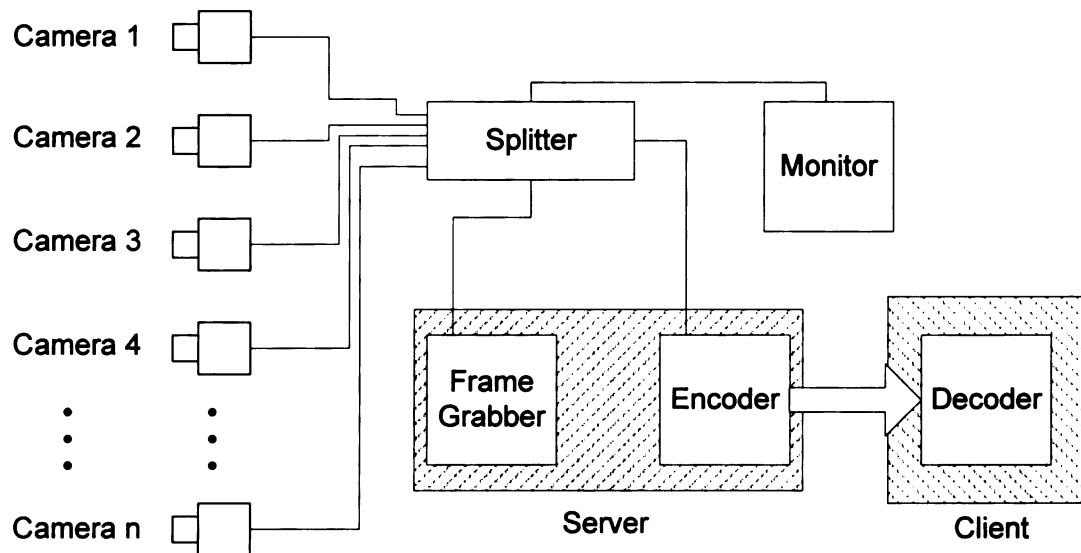


Figure 2 Hardware architecture

Another advantage of using server to perform most computation demanding process is that the parallel processing can be best utilized. Servers are usually designed to handle multiple applications at the same time and have passed a much higher reliability test so that they can operate 24/7 which is the same of any manufacture production. In general, servers are much more reliable than regular PCs. With multi-threading programming technique along with the powerful server machines, the tracking system can be pushed to its extreme to handle multiple cameras, multiple object tracking simultaneously.

A graphic user interface (GUI) is created on the client machine and important information of the tracking system can be visualized. The GUI also provides the possibility for human machine cooperation where the user can input some key parameter of the object to be tracked and these parameters are sent to server to help improve accuracy and efficiency of the tracking system.

## ***4.2 Software Architecture***

The majority of computation and image processing is completed on the server and Figure 3 shows the software architecture within the server machines. Region of interest (ROI) is utilized in this tracking algorithm because it greatly reduces the computation load. A ROI containing the most useful information of the object being tracked is created and the initial position of this ROI is provided by user in this system's set up. User at the client machine can interface with the tracking system through a graphic user interface and provide the information needed.

While using an automated tracking module, a sufficient prior knowledge of tracking object is required. Moreover, initial detection of location of object being tracked contains the most uncertainties and very likely to go wrong. Therefore, instead of writing the prior knowledge into hard code and risking the software's reliability, this initial detection function is left open for user to define the tracking object's characteristics at the initial processing. User-machine cooperation can save a lot of effort and reduce the chance of error to improve software reliability.



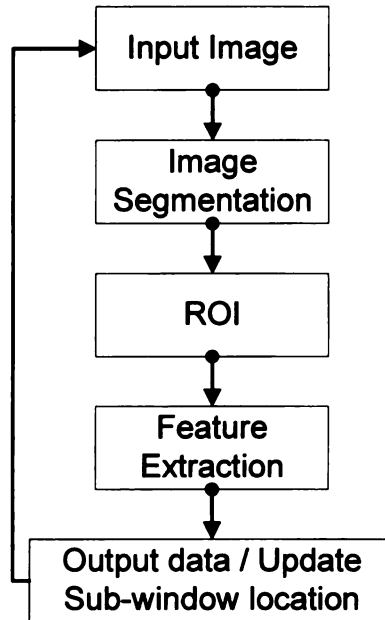


Figure 3 Software architecture

This tracking system also utilizes a recursive framework of the core image processing algorithm to accomplish the tracking objective. At the end of each image analysis, essential extracted information will be stored and passed to the next frame analysis as inputs. The advantage of recursive framework is that it greatly reduces the complexity of the program while increased its reliability. The draw back of recursive framework is that the program either has to perform a search for the tracking object over entire image for its location before it can extract any feature from the object being tracked, or use prior information from analysis to estimate the location of the tracking object. Either method has its affection on the overall system performance. The first method obviously requires more effort on computation which will increase the total processing time needed for each frame analysis which is not preferred in on-line tracking system. The second method will

have the potential risk of false estimation of the location of tracking object.

However, with added condition and constraints such as a history of object moving speed and direction, natural law of physics, etc, the false estimation rate can be minimized.

After analyzing input image through image segmentation and other processes, server will extract useful information of the object being tracked and then send them to client through an Ethernet connection. A TCP connection between server and client is used to ensure no loss of information during network data transmission.

## 5. Visual Tracking

### 5.1 Noise removal:

Noise is removed by 2-D adaptive noise-removal filtering.

$$\mu = \frac{1}{NM} \sum_{n_1, n_2 \in \eta} a(n_1, n_2)$$
$$\sigma^2 = \frac{1}{NM} \sum_{n_1, n_2 \in \eta} a^2(n_1, n_2) - \mu^2$$

The above equations produce the mean and variance of local neighborhood.

Use the above statistics a Wiener filter can be created, which is very efficient for noise removal. Noise removal at the initial step is part of the image enhancement process. Normally, under very good and stable environment, the camera can catch a very clean image. By clean, we mean there is no high frequency component of the captured image in the frequency domain. In other words, it means no “salt and pepper” noise in the spatial domain. Other noise removal filters using spatial domain method such as Gaussian blur usually perform an average operation on adjacent pixels. These methods work very fast since they are pixel-wise operations. However, when removing the “salt and pepper” noise, they also blurred the edges in the image since the adjacent gray level difference is weakened by the averaging operation. The other noise removal methods using frequency domain operations such as frequency cut-off don't have blurring effect on edge. However, these methods require the image transformed to frequency

domain from spatial domain and transfer them back to spatial domain after the operation is performed. Obviously this technique requires more processing time and is not favored by on-line applications.

The result of noise removed image, in this case, “salt and pepper” noise removed image has clearer edge information and provided a better platform for image segmentation which is the next step.

## ***5.2 Segmentation***

The next step is to get the edge image. Edge detection is very useful in image processing since the resulting edge image can be used for image segmentation and many other applications. It is a fundamental step in image processing. Edges are the sign of lack of continuity, usually a jump of intensity in the image.

Discontinuities in the image intensity can be either step edge, where the image intensity abruptly changes from one value on one side of the discontinuity to a different value on the opposite side, or line edges, where the image intensity abruptly changes value but then returns to the starting value within some short distance. However, step and line edges are very rare in real images. Because of low frequency components or the smoothing introduced by most sensing devices, sharp discontinuities rarely exist in real signals. Step edges become ramp edges and line edges become roof edges, where intensity changes are not instantaneous but occur over a finite distance. The common edge detection methods use a variety of masks and perform convolution across the entire image.

Because the resulting edge images from masks are always directional due to the nature of masks, these edge images are then added together to form one final edge image which includes edge of all directions. The obvious draw back of these methods is the directional effect of the resulting edge image. When the edge's direction is know, these methods can save a lot of effort on edge detection. But when dealing with a more generic situation, using masks to detect edge is not a good choice and can be very time consuming.

Therefore, adaptive threshold method is introduced to get possible edge points in this project.

$$p(z) = \frac{P_1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(z-\mu_1)^2}{2\sigma_1^2}} + \frac{P_2}{\sqrt{2\pi}\sigma_2} e^{-\frac{(z-\mu_2)^2}{2\sigma_2^2}}$$

$$AT^2 + BT + C = 0$$

$$A = \sigma_1^2 - \sigma_2^2$$

$$B = 2(\mu_1\sigma_2^2 - \mu_2\sigma_1^2)$$

$$C = \sigma_1^2\mu_2^2 - \sigma_2^2\mu_1^2 + 2\sigma_1^2\sigma_2^2 \ln\left(\frac{\sigma_2 P_1}{\sigma_1 P_2}\right)$$

Using the above equations, software can quickly threshold the image using an adaptive method and not worry about different illumination effect on the image.

Not like traditional threshold, the result of adaptive threshold is an edge image.

Adaptive threshold method will look at a smaller region of the entire image and produce a proper threshold value for that particular region. Generally, the region either contains pixels from a variety of gray levels or all the pixels are in one gray level. In the former case, adaptive threshold can apply a proper threshold value to the region to distinguish the difference in gray levels. In the later case, adaptive threshold can simply set all pixels to high since they all belong to one gray level. Notice that in the later case, no matter the pixels in the region has a higher gray level or lower gray level, they all get assigned to high in the output. This is very different comparing to traditional threshold operation. That is the main reason why adaptive threshold operation can provide an edge image and can be considered as an edge detection method.

The resulting edge image may still contain some unwanted features. Additional morphological operators such opening and closing operator can eliminate small noisy features. Connected component is another helpful algorithm can be used to further analyze the image and remove noisy elements.

As mentioned in section 3.2, a ROI is used to roughly locate the desired object and its initial location is provided by user. User draws a line or shape roughly near the objected being tracked and a ROI is created based on that drawing's starting and ending point. The initial location of object being tracked and ROI is usually hard to search automatically by software since it requires extensive computation and has a higher error rate. User-machine cooperation reduces the

computation cost and possibility of error. The ROI location is updated by software after the initial step using the detected desired object's properties.

### ***5.3 Unwanted features within ROI***

No matter how careful the image segmentation is performed, usually the resulting edge image in the ROI still contains a lot of features other than the desired target object. To prevent them from confusing the software with the desired target object in future analysis, a registry of these features is generated. The registry is constructed in a way that erases the pixel values around the initial detected desired object. The registry updated after every frame analysis as mentioned in previous sections.

When the next frame of image is grabbed by the frame grabber and the image segmentation is performed, an XOR function of the registry generated by last frame analysis and the new edge image can successfully remove confusing features and leaving only the desired target object. This method works both when the object being tracked is stationary and moving within the estimated ROI. In stationary case, since the registry doesn't contain the desired target object, the XOR function can bring the object back and remove all the background. In moving case, the target object traveled to a new location and changed the image in that area. Therefore, the XOR function can detect the difference and isolate the moved target object. After the new location of target object is found, use the new frame to create an updated registry for next frame's analysis. Therefore, if the



camera doesn't malfunction during image caption, the registry is always updated and the moving/stationary target object can always be isolated from the background or noisy features.

#### ***5.4 Edge fitting***

Only detecting the moving target object will not help the manufacturer much if no further information is generated. Therefore, this system includes a function to calculate the target object's certain properties and provide this information. This information can be applied by the manufacturer to perform any future task they want. Meanwhile, this information is also feed back to the tracking system as input for estimation purpose. An estimation of next frame's ROI location is produced based on this useful information. In this system, the function to calculate line property is included due to the nature of the test example in the following sections.

Since this computation must be done in a fast and accurate way, simple form of linear regression model is adapted to fit the resulting edge points obtained to a straight line. This method avoids any minimization of a function which has the possibility being trapped in a local minimum and either providing wrong results or end up in a dead loop. The following equations are used for calculating the line parameters.  $\bar{x}$ ,  $\bar{y}$  represent the average of edge points coordinates.

$$y_i = \alpha + \beta x_i + \varepsilon_i$$

$$\hat{\beta} = \frac{\sum x_i y_i - \frac{1}{n} \sum x_i \sum y_i}{\sum x_i^2 - \frac{1}{n} (\sum x_i)^2}$$

$$\hat{\alpha} = \bar{y} - \hat{\beta} \bar{x}$$

To reduce computation time, only area within ROI is used. The area is chopped into 10-15 stripes. In each stripe, a horizontal line scan is performed to get multiple line scan profiles. Using these profiles, gradient method is applied to find the edge points and get an average of the total line points of each stripe. Therefore, each stripe can generate one point and totally 10 to 15 points is produced for the linear regression model to calculate the straight line's slope and offset. This information is then sent to client through Ethernet connection between the server and client.

### ***5.5 Calibration***

Once a camera is set up and position is fixed, it can be calibrated to provide measurements in engineer units instead of pixel values. This is very useful in manufacture process since the real world results can be applied to control systems as inputs to form a close loop control system and guiding a robot arm to finish a task. As mentioned in previous sections, stereo vision can avoid the calibration process of using CAD model which usually include the task of accurately locate certain special points in the image. In this research, due to the

nature of the test example, stereo vision is not an option so that calibration using CAD model is utilized. From another perspective, stereo vision can't be applied to all system while there must be CAD model for machines used in manufacture process. Therefore, calibration using CAD model is a more generic method which can be applied to all tracking system works in manufacture process.

This process of using CAD model for calibration usually includes a lot of image processing effort in order to accurately identify certain points and provide their image coordinates. In order to reduce the effort on identifying useful points, a better approach is to choose unique points such as points at a particular corner or some other unique properties which made them stands out and easy to identify. This extra effort invested before programming the software is very important because it saves a lot of computation power and the calibration is more robust and reliable. An alternative method to avoid this effort in identifying these unique points is to take the advantage of the system's set up to use human-machine interaction. This means a list of particular points on the CAD model is stored orderly in the software and a user will locate these points in the same order. In this case, the points should have similar properties so that the software can use the same method to refine the user's selection and provide more accurate coordination for these points. To sum this up, basically, these certain points' image coordinates in pixel values can be paired up with their CAD model coordinates in engineering units. A set of equations can then use these pairs to

produce a transformation which describes the camera model and transform the pixels on image plane to the real world coordinates. There is one more important factor will affect the performance calibration process. All these pre-defined points in both methods must be presented in the image. If the image doesn't contain these points, the result will be either a false location or lack of enough points to perform proper calibration. Either way the resulting transformation matrix of the camera model will have a large error of converting the image coordinates to engineering units.

There are many existing proven calibration methods and in this research we use affine transformation [13] to get the camera model. This method is very straight forward to understand and the performance is very reliable.

$$\begin{aligned}
{}^C P &= T(t_x, t_y, t_z) R(\alpha, \beta, \gamma) {}^W P \\
{}^C P &= {}^W C T R(\alpha, \beta, \gamma, t_x, t_y, t_z) {}^W P \\
{}^I P &= \left( {}^I S {}^F C \Pi(f) {}^W C T R(\alpha, \beta, \gamma, t_x, t_y, t_z) \right) {}^W P \\
\begin{bmatrix} s {}^I P_r \\ s {}^I P_c \\ s \end{bmatrix} &= \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & 1 \end{bmatrix} \begin{bmatrix} {}^W P_x \\ {}^W P_y \\ {}^W P_z \\ 1 \end{bmatrix}
\end{aligned}$$

The superscript “C”, “I” represents camera plain and image plain respectively.

Symbols “ $T$ ” and “ $R$ ” are transformation and rotation matrix. “ $s$ ” is scaling factor.

There are total of 12 parameters need to be determined in the equations which means at least 6 unique points with known CAD model coordinates must be located within the image. However, the number of points used by this method is not limited to 6. The points used toward this method are identified by software or user. Either way there may be an offset embedded in the points' coordination. In order to refine the camera model and improve the accuracy of the transformation, more points with coordination can be introduced to the method. All together, these points can be applied to the above equations along with least square fit to minimize the error of the camera model.

Note that at the beginning of this section, camera calibration is a one time procedure. Therefore, as long as the camera doesn't move, the camera model will not change. Because of this nature, the camera calibration procedure doesn't have a strict time limit requirement as the tracking procedure. It is acceptable if the calibration process requires a reasonable longer time to finish its procedure in order to provide an accurate camera model.

## 6. Result and analysis

In this research, glass manufacture video is used as example to test system performance. A standard glass manufacture can be break down into five main stages shown in Figure 4.

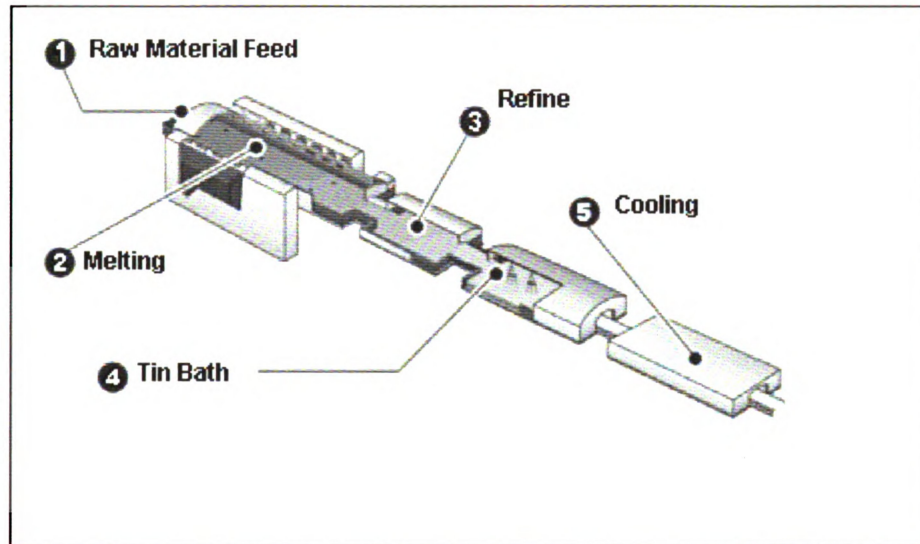


Figure 4 Glass manufacture process.

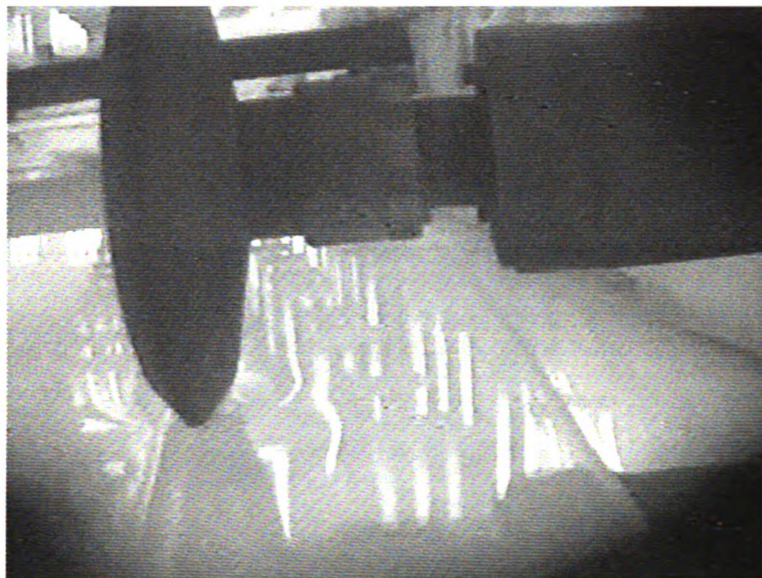
Firstly, the raw materials of glass which usually include a mixture of silica sand, soda ash and so on are blended and fed into the production machine. Secondly, the mixture of raw material is melted into liquid form under extreme temperature usually  $1600^{\circ}\text{C}$  or higher. Thirdly, the liquid glass is cooled down to temperature between  $1100^{\circ}\text{C}$  and  $1300^{\circ}\text{C}$ . It is during this stage all bubbles formed in the viscous liquid glass flow are removed so that the liquid glass is ready for the next step. Fourthly, while maintaining its temperature, liquid glass float smoothly and uniformly distributed on melted tin to form the shape and thickness desired. The equipment completing this task is called tin bath. It is at

this stage that the tracking system came in to play an important role of tracking the liquid glass edge and calculate its distance between the edge and the machine grabbing the liquid glass flow. The machine is the plate shaped object in Figure 1. The machine has the ability to grab the glass flow and by controlling the machine's position and angle of operation, certain thickness and width of glass can be produced. The glass between the actual liquid glass flow edge and machine is eventually cut and put back to stage one as raw material. Therefore, by minimizing the distance between liquid glass flow edge and machine can save the glass manufacturer a lot of money. However, a certain threshold of the distance is desired to ensure that the liquid glass doesn't loose from the machine and shrink causing the production process to halt and waste a lot of money. The tracking system can provide necessary data for the production line to operate safely and continuously while optimizing its cost. It is shown from Figure 1 that the image is dark and contains a lot of noise both from the environment and the various objects inside the view of camera which can also be shown from Figure 5 in an enlarged region of interest (ROI) view. Therefore, this situation matches the purpose of this research and is chosen to be the experiment to test the proposed image processing algorithm and hardware/software system structure. Hardware engineers have figured out a way to retrieve video signals from the very hot tin bath. However, only one camera can be used to point at a particular view inside the tin bath which in this case is the machine and its adjacent liquid glass flow



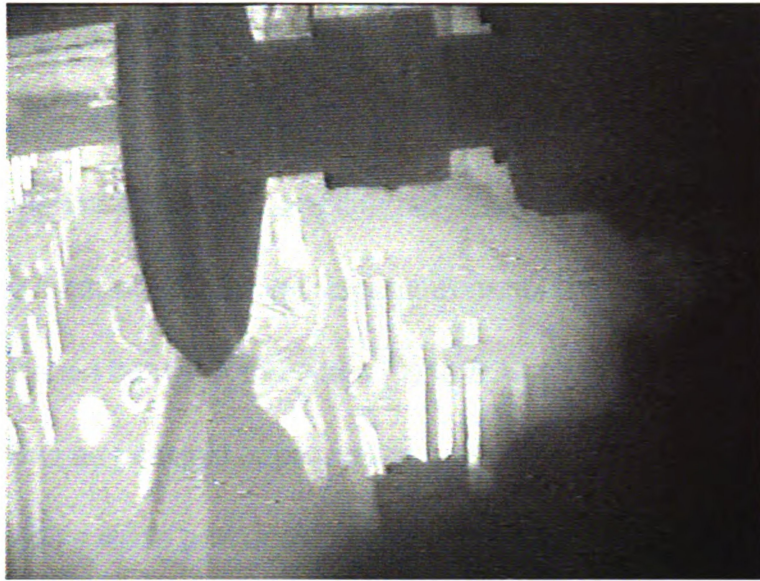
edge. This hardware setup limited the ability of using stereo vision technique but it is still possible to calibrate the camera using various unique points of machine and its corresponding CAD model. The last stage of glass manufacture is just cooling the glass down to return its solid form.

Results and analysis is presented in the following section.



(a)

Figure 5 Sample ROI image and tin bath images



(b)



(c)

Figure 5 continue

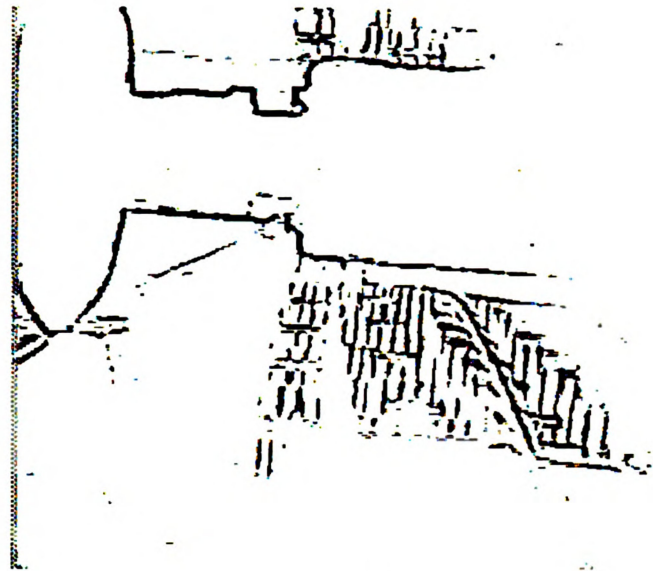
Figure 5 shows a sample ROI image in which line indicates glass edge. It is obvious that without further knowledge, even human can not separate glass edge with other edges in the image. The multiple short white stripes are heating elements reflections. There is a slight grayscale difference between the liquid glass and the melted tin in this ROI view. This grayscale difference is hard to distinguish at most part of the camera view but much easier to identify at the bottom where shadow from the machine presents. Unfortunately, the shadow is

not always presented in the camera view and the edge inside the shadow is too short for edge property extraction. The lack of edge data points will result a large error of the slope and offset of the estimated straight line fitted using these points. Therefore, the image processing algorithm must be able to distinguish the grayscale difference between the liquid glass flow and the melted tin and use as many edge points as possible to provide enough data for the line fitting function to estimate the edge's slope and offset and calculate the distance between the liquid glass edge and the machine.



(a)

Figure 6 Adaptive threshold with and without noise removal



(d)

Figure 6 continue

Figure 6 shows the result of both stand alone adaptive threshold and adaptive threshold with 2-D noise removal filter. Figure 6a is threshold image without noise removal and 5b is threshold with noise removal. The parameter of adaptive threshold is a region size of 15x15. This parameter is obtained by many experiments done on different situations of input images. Different region size of threshold will have different effect on the output result image. If the region size is too large, there are two defects. Firstly, it will increase the processing time needed to calculate the threshold value for each block. Secondly, since the region is increased, a lot of fine details are included in the region. After threshold is applied, many of them will be eliminated. This is a really unwanted because the main purpose of apply threshold to the image is to reduce the image data size while maintaining as many detail as possible. On the other hand, if the region size is too



small, there won't be enough data within the region to produce a correct threshold value. Therefore, while applying adaptive threshold method, region size is crucial and must be tuned to get the best performance. Since this is an empirical result, it is made tunable in the different situations. Comparing Figure 6a and Figure 6b, one can easily notice that without noise filtering, the resulting image will contain many salt and pepper noises. These are some high frequency components presented in the original image. If the salt and pepper noise is left untouched in the image, it will greatly reduce the accuracy of the following steps and eventually reduce the overall system performance. Therefore, it can be shown that filtering is a must do step for real application image processing.



Figure 7 Global threshold

Figure 7 here shows the result of a simple global threshold. All fine features including the edge information are lost using this method. This is because that the

liquid glass flow edge only has a slight difference from melted tin and these gray levels all fall into one region on the gray level histogram on the entire image. Therefore, after global threshold, they all merged into one level. In real applications, especially when lighting is inconsistent and causing uneven distribution of illumination, adaptive threshold is much better than global threshold.

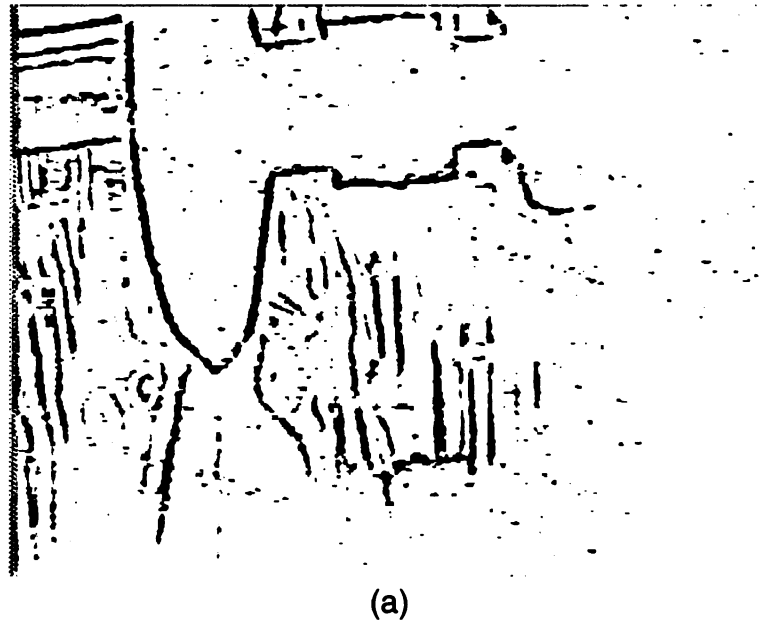


Figure 8 Opening and Closing Operation Effects



(b)

Figure 8 Continue

Figure 8(b) show result after opening followed by closing operations of the adaptive threshold image of Figure 8(a). This operation further removes small objects in the image. From the above figures one can notice that in some case, even with noise filtering, the output image still contains noise in the form of small objects. These objects are small compare to the actual features in the image, but they are large enough to pass the high frequency removal filter. There are several ways to remove these small objects. A simple one is to apply connected component algorithm on the entire image and then remove the ones with small area property. Connected component is very easy to implement but requires a relatively large portion of memory to complete the algorithm as well as longer process time depending on the image's size. These are really bad for an on-line tracking system. So we move to an even simpler pixel-wise method to get rid of



the small objects. An opening operation followed by a closing operation is proven very efficient at getting rid of most noisy small objects. These operations also have a very low demand of memory since they are pixel-wise operations and timing efficient. In addition to its effectiveness, the opening and closing operation connected some of the broken edges to regenerate its correct form which provides a good platform for the data analysis in the later steps.

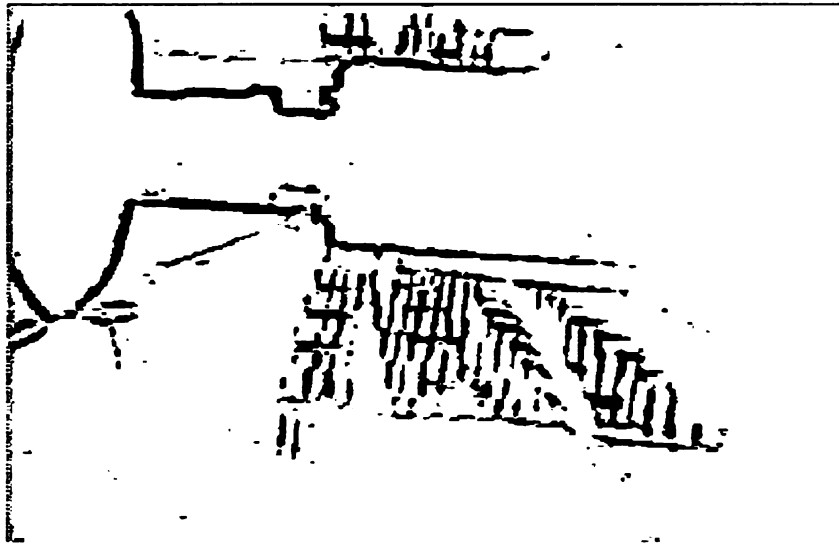


Figure 9 Registry of non-interest edges



Figure 10 Result of XOR between registry and new processed frame

In the iterative framework, it is important to distinguish the object to be tracked from the background. In the glass manufacture case, the background image is

very complicated and very hard to be distinguished since the heating element reflection has the similar shape to the liquid glass edge. This situation may also happen to many other applications. Therefore, creating some kind of registry to store the background information is important. This registry can then be used as a reference to distinguish the object being tracked from the background image easily. Since this is an iterative framework, the registry must also be updated because the background may have some changes over a period of time. The change can either be caused by a move of the camera or the lighting is different. If it is within the scope of processing time, it is recommended to update the registry for each frame but it is not required. Different applications may have a different approach on this matter. In Figure 9 and Figure 10, shows the result of edge registry and XOR operation result of processed new frame image. It can be shown that with the correct background registry, the moving object is easily identified and separated. The result of Figure 10 can be used for line fitting of the next step.

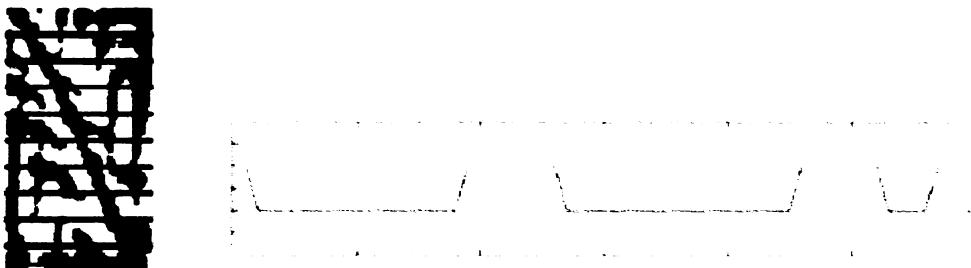


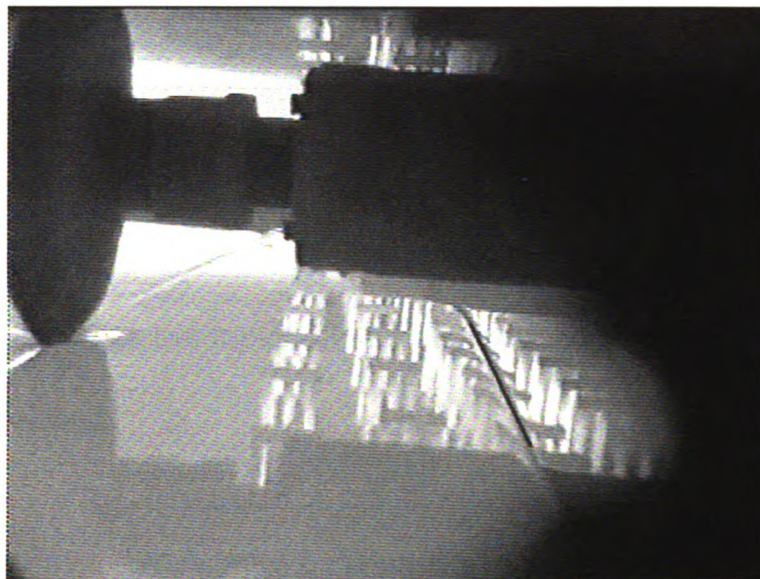
Figure 11 ROI and line scan sample

Figure 11 show the ROI image and a sample line scan profile. The horizontal lines divide the ROI into multiple stripes. Within each stripe, line scan generates

multiple line profiles. The spikes of the line profile indicate edges of the line scan. This ROI image contains the liquid glass edge and some background component as well. Therefore in the line scan profile, multiple spikes are observed. This has proven that correctly separate object being tracked from the background is very important and can save a lot of analysis effort in the following steps. With the correct separation, the ROI will only contain the moving object, and the line scan should only have one spike for data retrieving and analysis. However, if the ROI contains some noise like it is shown in Figure 11, there is still possible method to identify the correct spike in the line scan to be used for data analysis. By looking at previous frames result, program can give an estimate of the moving object's position. Based on the estimation, the program can look for the nearest spike in the line scan profile and treat that as the object.

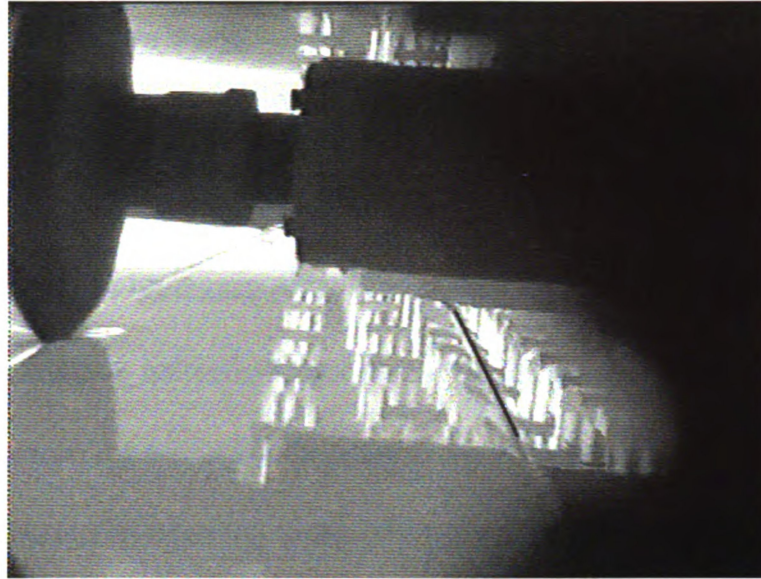
The spike contains a rising edge and falling edge. The mid-point is considered to be the actual object location and is generated as the line scan profile's output. There are multiple line scan profiles in each stripe and many object location points are produced by analyzing the line scan profiles. Averaging multiple line scan within each stripe will generate one object location point for further analysis. In this glass manufacture case, the liquid glass edge's property is the primary goal and it can be fitted into a straight line. Using the object location points produced by each stripe of the ROI and applying the line fitting method in section 4.4, the result is the edge's slope and offset. These values are used to calculate the

distance between the liquid glass edge and the machine using standard distance from point to straight line formula. The result can be either in pixel distance or in engineering units since the camera has been calibrated using method provided in section 4.5. The results in engineering units have the error rate less than 5% from Figure 13.



(a)

Figure 12 Result of fitted edge



(b)

Figure 12 Continue

Figure 12 shows result of two consequent frame with result edge showing on top of original image as overlay drawing. The total process time of each frame is less than 0.5 second.

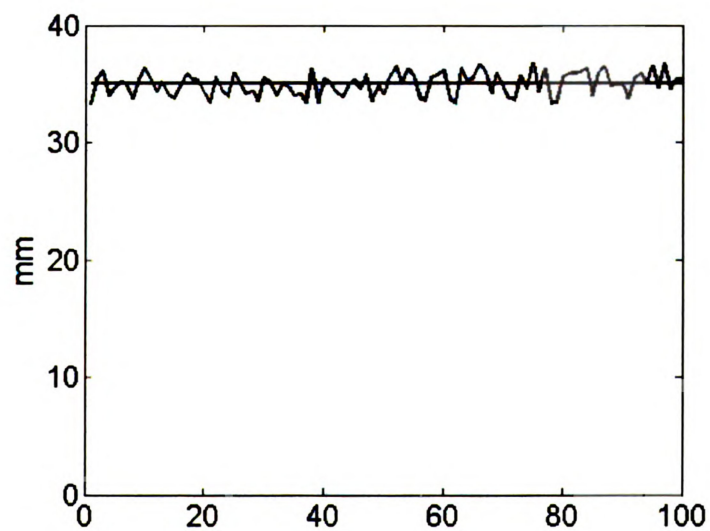


Figure 13 Calibration result

## **7. Conclusion and future works**

The distributed system set up enabled the tracking system to handle multiple cameras multiple objects tracking simultaneously. The utilization of server machine ensured the hardware reliability to meet manufacture standard and can run 24/7. The iterative software structure simplified the computational complexity and provided more robustness with proper constrains setting. The calibration process of camera provided the ability of transforming pixel value and image coordinates to more meaningful engineering units which can be used as feedback to engineers to determine how to optimize the manufacture line. The image processing techniques included in this system provides accurate and robust image analysis results. Moreover, human-machine interaction is introduced to this tracking system so that operators can provide necessary support to the tracking algorithm to further improve the automation process' reliability.

Because image processing is separated to server which has a higher computation power, image acquisition and analysis time is much shorter compare to the object's movement. This system can be used for edge tracking or other or rigid object tracking. The advantage of the hardware system set up also provides the ability of remote accessing and control for future works since Ethernet communication is introduced. After all, this system is developed focusing on industrial applications. Therefore, reliability is the most important issue while accuracy has a higher tolerance comparing to research applications. From the

experiment result, this visual tracking system takes less than 0.5 second to finish analysis for a single frame of image and produce information. For a single camera tracking, the processor usage on server machine is less than 20%. Note that multiple tracking modules can work at the same time to accomplish multiple camera tracking and the total processing time will not be affected.

Future works of this project can include backup method for edge detection like template matching to guarantee edge tracking of all time. This effort must be made within the hardware resource's availability since template machining is also a computational demanding algorithm. An image understanding process which can automatically determine the image's characteristics will also help complete the automation of this system. With an image understanding process and a high level decision rule which can also be implemented to determine under what circumstance which tracking method is preferred or has the higher probability of the best performance, this system can be pushed to its limit of best performance possible. As mentioned in previous sections, camera set up can be altered to form a stereo vision system to avoid CAD model for camera calibration and reduces the calibration error. This system only records a small number of historical data to provide necessary data to support estimation procedure. A database system can also be introduced to interface with this tracking system to record all historical data which can be used to either provide data log for the manufacturer or to statistical model analysis to improve the manufacture process. Adding a database

to the system also includes the option of interfacing with other manufacture process.

This tracking system is a base model which can still be further improved in many ways to achieve the goal of industrial manufacture process automation and optimization.



## REFERENCES:

- [1] Heping Chen, William Eakins (2009). Robotic Wheel Loading Process in Automotive Manufacturing Automation. IEEE/RSJ International Conference on Intelligent Robots and Systems.
- [2] Kemal Berk Yesin, Bradley J. Nelson (2004). Robust CAD Model Based Visual Tracking for 3D Microassembly Using Image Space Potentials. IEEE International Conference on Robotics and Automation.
- [3] Zhenbang Gong, Wei Ding and Hairong Zou (2006). Data-logging and Monitoring of Production Auto-lines Based on Visual-tracking Tech. IEEE.
- [4] Hong Liu, Ying Shi (2009). Robust Visual Tracking Based On Selective Attention Shift. IEEE Multi-conference on Systems and Control.
- [5] Masaaki Shibata and Nobuaki Kobayashi (2006). Image-based visual tracking for moving targets with active stereo vision robot. SICE-ICASE International Joint Conference
- [6] A. Aouf, H. Rajabi, N. Rajabi, H. Alanbari and C. Perron (2004). Visual object tracking by a camera mounted on a 6DOF industrial robot. IEEE conference on Robotics, Automation and Mechatronics.
- [7] Tae Hyoung Park and Beom Hee Lee (1997). Dynamic Tracking Line: Feasible Tracking Region of a Robot in Conveyor Systems. IEEE Transactions on Systems, Man, and Cybernetics, vol. 27, pp. 1022-1030, 1997
- [8] N. J. Ferrier (1998). Performance of Visual Tracking Systems: Implications for Visual Controlled Motion. Conference on Decision & Control
- [9] LARRY S. DAVIS (1975), A Survey of Edge Detection Techniques, Computer Graphics and Image Processing 1975, 4, 248-270
- [10] Rafeal Gonzalez, Richard Woods (2002). Digital Image Processing, Prentice Hall
- [11] Alexander Borst (2007). Correlation Versus Gradient Type Motion Detectors: the pros and cons. Phil. Trans. R. Soc. B 362, 369–374

- [12] Naoyuki Sawasaki, Toshihiko Morita, Takashi Uchiyama (1996). Design and Implementation of High-speed Visual Tracking System for Real-Time Motion Analysis. International Conference on Pattern Recognition
- [13] Linda Shapiro, George Stockman (2001). Computer Vision. Prentice Hall
- [14] Kazuhiko Kawamoto (2008). Adaptive Sampling for Bayesian Visual Tracking
- [15] P. F. Mclauchlan, J. Malik (1997). Vision for Longitudinal Vehicle Control. ITSC 97, IEEE Conference, Nov. 1997 pp 918-923
- [16] M. P. Groover, M. Weiss, R. Nagel, and N. G. Odrey (1986) Industrial Robotics: Technology, Programming and Applications. McGraw-Hill
- [17] T. H. Park and B. H. Lee, An approach to robot motion analysis and planning for conveyor tracking, IEEE Trans. Syst., Man, Cybern., vol. 22, pp. 378–384,
- [18] Jung Uk Cho, Seung Hun Jin, Xuan Dai Pham (2007), FPGA-Based Real-Time Visual Tracking System Using Adaptive Color Histograms, IEEE International Conference on Robotics and Biomimetics
- [19] G. R. Bradski (1998), Real Time Face and Object Tracking as a Component of a Perceptual User Interface, IEEE workshop on Applications of Computer Vision, Princeton, pp. 214-219
- [20] K. Nummiaro, E. Koller-Meier, L. Van Gool (2002), A Color-Based Particle Filter, Generative-Model-Based Vision, pp. 53-60
- [21] S. Fleck, W. Straber (2005), Adaptive Probabilistic Tracking Embedded in a Smart Camera, IEEE Computer Vision and Pattern Recognition, Vol. 3, pp. 134-142
- [22] Andrew W. B. Smith, Brian C. Lovell (2005), Measurement Function Design for Visual Tracking Applications, International Conference on Pattern Recognition
- [23] T. F. Cootes, C. J. Taylor (1992). Active shape models - 'Smart Snakes'. British Machine Vision Conference
- [24] A. Blake, R. Curwen, A. Zisserman (1993). A framework for spatio-temporal control in the tracking of visual contours. International Journal of Computer Vision

- [25] P. Beardsley, A. Zisserman (1995). Affine structure from motion. International Conference on Computer Vision
- [26] O. Faugeras, F. Lustman, G. Toscani (1987). Motion and structure from motion from point and line matches. Conference on Computer Vision
- [27] D. Koller, D. Danilidis, H.-H. Nagel (1993). Modelbased object tracking in monocular image sequences of road traffic scenes. Journal of Computer Vision, 10(3):257-281
- [28] Bijoy K. Ghosh, Di Xiao (1997), Sensor-Guided Manipulation in a Manufacturing workcell. International Conference on Intelligent Robots and Systems, pp. 1403-1407
- [29] Hesheng Wang, Yun-Hui Liu (2006). Uncalibrated Visual Tracking Control without Visual Velocity. IEEE International Conference on Robotics and Automation
- [30] A. Astolfi, L. Hsu, M. Netto, R. Ortega (2002), Two Solutions to the Adaptive Visual Servoing Problem, IEEE Transaction on Robotics and Automation, vol. 18, no.3, pp. 387-392



MICHIGAN STATE UNIVERSITY LIBRARIES



3 1293 03163 7014