Procedia Manufacturing

Volume 1, 2015, Pages 416–428

43rd Proceedings of the North American Manufacturing Research Institution of SME http://www.sme.org/namrc

# A Comparative Study of Machine Vision Based Methods for Fault Detection in an Automated Assembly Machine

Vedang Chauhan* and Brian Surgenor†

*Department of Mechanical and Materials Engineering*
*Queen's University, Kingston, Canada*
*vedang.chauhan@queensu.ca, brian.surgenor@queensu.ca*

**Abstract**

Fault detection and classification in an automated assembly machine using machine vision (MV) based inspection methods is the subject of this paper. A high speed automated assembly machine is used as the test apparatus. The machine is designed to assemble circular O-rings, from a bulk supply, onto continuously moving carriers at a rate of over 100 assemblies per minute. Video data was collected for both normal and abnormal machine conditions, and in particular transfer track jams. Three MV classification methods were adapted to this application and subsequently tested: 1) Gaussian Mixture Models (GMMs) with blob analysis, 2) optical flow and 3) running average. The methods are compared with a previously developed fault detection method based on spatiotemporal volumes (STVs). It is observed that the new methods require less training and processing time and are able to detect faults faster than the STV method. Amongst the three new methods, the running average method is shown experimentally to be the best in terms of having the lowest processing time per frame and the fastest response time. The work continues in order to see how the methods perform as different machine faults are introduced.

*Keywords:* Machine Vision Inspection, Fault detection and classification, Gaussian Mixture Models, Optical flow, Running Average

## 1 Introduction

To gain competitive advantage and to achieve high volume production rate with better product quality, modern manufacturing industries are integrating automated assembly machines with their production lines. These automated machines are equipped with sensors for control of various parts of

* PhD Candidate, Department of Mechanical and Materials Engineering, Queen's University, Canada
† Supervisor and Professor, Department of Mechanical and Materials Engineering, Queen's University, Canada


Peer-review under responsibility of the NAMRI Scientific Committee
doi:10.1016/j.promfg.2015.09.051

the machines and synchronization of events in the assembly process. Even with the state of the art technology, it is hard to prevent the machines from having faults such as missing parts in assemblies, part jams, misalignments, and blockages. These are commonly observed faults in the assembly machines that can cause high production downtime and eventually increase the running cost. Early fault detection and classification can minimize production downtime and help to restore a machine to its online state in the least possible time (Boothroyd, 2005).

Machine fault can be defined as a condition under which a machine is subjected to deviation from its standard acceptable operating condition (Isermann, 2006). Most easy way to detect faults in the machines are traditional techniques such as limit checking and behavioral model (Holloway and Krogh, 1990), where sensors are used to continuously monitor the health of the machines and any deviation from predefined thresholds is considered as a fault. These methods are good for fault detection but they are not best in terms of speed of response and generating necessary data that can help in fast fault diagnosis. Hence, there was a need of fast and accurate supervision of automated machines for fault detection and isolation.

The camera-based methods (Szkilnyk, Hughes et al., 2012; Hughes, Fernando et al., 2014) are fast , more accurate and provide necessary information for fault isolation and diagnosis in case of a fault occurrence. It also helps an operator to understand what went wrong by replaying a video of the machine at the event of the fault. Szkilnyk (2012), proposed the use of such MVI system for fault detection of an automated assembly machine. The idea was derived from the application where MVI system is used for detection of an unusual event in a crowded area (Benezeth, Jodoin et al., 2009). The method models the normal behavior of the crowd using spatiotemporal co-occurrences and compares real time model generated from a video data with the normal model. If it identifies any deviations between both models then it labels that as an abnormal event.

The intent of this research project is to study the scope of a MVI system for fault detection and classification in an automated assembly machine. *Fault detection* is the stage where a fault is recognized as soon as it occurs and *fault classification* means detecting type and location of a detected fault. Three different MVI methods such as fault detection using GMMs and blob analysis, optical flow and running average method are developed, tested and compared in terms of their complexity, efforts required to train them and processing time for AVI of the automated assembly machine. The performances of these methods are compared with the STV method for machine fault detection and classification.

# 2 Related work

Many researchers have worked in the area of fault detection and diagnosis in machines. During literature review, several papers were found that focused on machine condition monitoring and fault detection in assembly automation. A Petri net based method (Viswanadham and Johnson, 1988) was developed for fault detection and diagnosis of automated manufacturing system using a two-level scheme. At level one, an intelligent controller performed the monitoring for each subsystem, such as machine centers, robots, conveyors etc. The level two had Petri net based controller that kept a track of the work piece flow and communicated with level one controller for the system level monitoring. Konrad (1996), presented a fault detection method, for milling machines, using parameter estimation whereas a pattern classifier was built based on model parameters estimated for each insert of milling cutter for a given cutting force. The pattern classifier then used to detect and classify faults in milling machine such as wear or breakage of inserts.

Demetgul, Tansel et al. (2009), implemented two ANNs (Adaptive Resonance Theory 2-ART2 and Backpropagation –Bp) for fault detection in didactic modular production system from Festo Company. Both ANNs were trained using data collected from eight different sensors for both normal and faulty operations and were correctly able to classify faults using trained networks. Sekar, Hsieh et al. (2011),

proposed an e-diagnostic system for PLC based automated assembly systems. The dual robot assembly system in Rockwell Automation Laboratory was considered as an example and three levels of remote diagnostic architecture were employed. At level 1, remote collaboration was established for remote connectivity to the robot. Level 2 had the same capabilities as Level 1 plus remote performance monitoring of operations and Level 3 provided advanced analysis and diagnosis with automated report generation.

An AVI of PCBs for defect detection, classification and localization using image processing techniques such as mathematical morphology and template matching was investigated by Malge P. S. and Nadaf R. S. (2014). A machine vision system for the detection of missing fasteners on steel stampings was proposed by (Killing, Surgenor et al., 2009). A neuro-fuzzy image classification algorithm was developed and compared with a threshold-based classifier. It was reported that once optimized, the neuro-fuzzy classifier degraded in a less abrupt fashion when the input significantly deviated from the trained data. Klein, Masad et al. (2014), investigated early bearing failures and diagnosis using image processing techniques applied to time-frequency representation (TRF) of the vibration signals. The four stages system was developed to detect and classify bearing failure signs based the vibration signals. In stage 1, TRFs of healthy machines obtained using the methods such as wavelets and SIFT. Stage 2 compared the distance TRF between the new TRFs and the baseline. In stage 3, the distance TRF was analyzed using ridge tracking and other image processing techniques. Stage 4 established the relations between the ridges and the characteristic pattern of the bearing failure.

Machine vision systems are often used for detection of unusual events in a scene such as event detection in crowded videos (Ke, Sukthankar et al., 2007). These systems are used for automated surveillance of the area, where they first model the normal behavior of the crowd using image processing techniques like STVs. The STVs are 3D representation of a scene in both space and time coordinates. During real time monitoring of the scene, new STVs are built at fixed interval of a time and compared with the normal STV. Any deviation form a normal behavior is identified and then isolated based on where in the model deviation occurs. These deviations are then later used to classify the video data into a particular condition using predefined rules. Szkilnyk, Hughes et al. (2012), examined the application of a video event detection method, based on STVs, for fault detection in automated assembly machines.

Automatic detection of stamping defects in lead frames (the core part of a semiconductor that functions as a substrate/base in ICs) using machine vision was implemented by Bhuvanesh and Ratnam, (2007). The algorithm used image processing operations such as blob analysis, morphological operations and image subtraction to detect the stamping defects in the presence of transitional and rotational misalignments. The system was capable of detecting defects from both continuous reel and individual cut lead frames with rotational misalignment of +/- 10 degrees.

Xiaokun Li and Porikli (2004), presented a novel approach to automatically detect highway traffic event, such as heavy congestion, vacancy, jam etc. using a MVI system. The algorithm processed the compressed video of the inspected area, extracted event features from DCT domain without decoding and detected traffic event using a Gaussian Mixture Hidden Markov Model (GMHMM). Six traffic patterns were studied and the GMHMM was trained to model these patterns and it detected events in real time with an accuracy of 94 %. Usamentiaga, Molleda et al. (2013), proposed a new system to detect jams in a steel processing line at a crucial step of pickling using a MVI system. The proposed machine vision system acquired images from the pickling line and processed them, extracted features and detected jams in two nozzles using running average method.

Optical flow-based tracking methods are precise and reliable for the analysis of motion. Motion estimation in image sequences using optical flow method was first introduced by Horn and Schunck, (1981) with examples of a rotating sphere and a cylinder on its axis. Patel E. and Shukla D., (2013), implemented vehicle tracking algorithm using the optical flow, calculated using wavelet decompositions by measuring a phase change between representative coarse wavelet coefficients in

subsequent image frames. The results proved that using phase-based optical flow with wavelet approximations for vehicle detection and tracking performed better than the intensity-based optical flow method. However, no publications were found that used optical flow for fault detection in an assembly machine.

# 3 Experimental Setup and Data Collection

## 3.1 Test apparatus

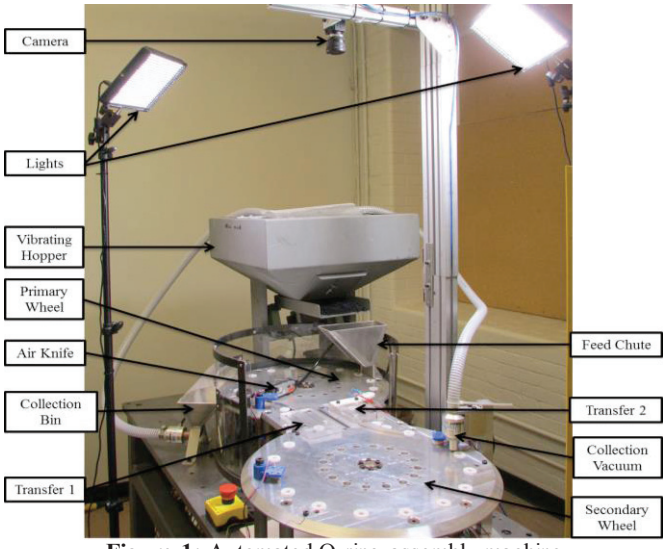


**Figure 1:** Automated O-ring assembly machine

Fault detection using a MVI system for automated assembly machines is carried out on a modified assembly machine as shown in Figure 1. The main parts of the machine are two transfer wheels (primary wheel and secondary wheel) that hold empty carriers and assemblies, a vibrating hopper for storage and supply of O-rings, an air-knife above the primary wheel and a vacuum system for collection of excess O-rings and assembled O-rings. The machine assembles black colored O-rings (Φp= 8mm and Wp= 1mm) onto continuously moving, white, circular carriers. The rate of assembly is 108 per minute at the maximum RPM of the machine.

During an operation, as the hopper vibrates, a steady stream of O-rings fall onto the primary wheel through the feed chute where a single O-ring is assembled onto a circular groove on a carrier using aligning pins. The excess O-rings, that are not assembled, are blown off into the collection bin by the air-knife. In the next stage of the assembly process, the assemblies are transferred, one at a time, to the secondary wheel via the air transfer track 1 (transfer 1). The secondary wheel, when rotates, carriers the assemblies with it until they reach the collection valve. At this point, only O-rings are vacuumed and recirculated back to the hopper. The empty carriers are then returned to the primary wheel via the air transfer track 2 (transfer 2) and the cycle is repeated. The machine operates in a continuous mode using the PowerFlex® 4 AC drive. The machine uses pneumatic system for vacuum suction and for the transfer of assemblies and empty carriers. During a continuous run of the machine, several faults were observed such as transfer track jams, missing O-rings, misaligned O-rings and carriers etc. The purpose of this research is to detect and classify those faults using MVI methods.

## 3.2 Machine Vision System and Data Acquisition

A MVI system developed for the proposed work consists of the USB 3.0 camera, two Aputure Amaran AL-528W LED panels for lights, image processing and computer vision software-MATLAB2014, Allen Bradley MicroLogix 1400 PLC with a Panel View C600 HMI to introduce the controlled faults and to communicate MVI system decision to the machine. In the first step of the experiment, the randomly occurring faults in the machine were minimized and the machine was made to run under a normal operation cycle for the short duration of time. Several commonly observed faults were then introduced by turning on a corresponding solenoid valve using the PLC and HMI.

The MVI system design included steps such as selection of the camera and lens, locating the camera at the appropriate working distance, selection and placement of lights (Chauhan, Surgenor et. al. 2014). The Point Grey Grassshoppe3 (USB 3.0) Camera with 1/1.8" 1928 x 1448 CCD sensor and 2/3" format lens , mounted 1 m above the machine surface, is used for the video data acquisition. The camera was set at the resolution of 724 x 964 with the frame rate of 30 fps. Total 15 videos , 10s each, were recorded for three different machine conditions: 5 for a normal operation, 5 for a transfer track 1 jam (transfer 1 jam) and 5 for a transfer track 2 jam (transfer 2 jam). The time (in terms of a frame number) was recorded when the fault was introduced such that it can be used to compare how fast a fault has been detected by different methods once it is introduced.

# 4 Machine Vision Based Methods for Fault Detection

## 4.1 Method 1: Fault Detection using GMMs and Blob Analysis

This method uses the foreground detector system object to monitoring the machine while it is in operation and classifies the operating condition into a normal operation, a transfer 1 jam or transfer 2 jams. The foreground detector object reads grayscale video frames to compute and return the foreground mask using Gaussian mixture models (GMMs) (Stauffer, C. and Grimson, 2000). The single Gaussian model for background detection may not be reliable in case of lighting changes and scene changes. Hence, the approach of adaptive Gaussian mixture model, where each pixel in the scene in modeled by a mixture of K Gaussian distributions works better for moving objects detection.

**Construction of GMM:**

If $I$ is the image sequence and $\{x_0, y_0\}$ represent the location of a pixel then the history of the pixels value $X_t$ is obtained by equation (1),

$$\{X_1, \ldots \ldots, X_t\} = \{I(x_0, y_0, i): 1 \leq i \leq t\} \quad (1)$$

The recent history of each pixel is modeled by a mixture of K Gaussian distributions. The probability of observing a certain pixel value $X_t$ at time t can be written as Equation (2), where $\omega_{i,t}$ is weight for ith Gaussian component, $\mu_{i,t}$ is the mean of the Gaussian component and $\eta(X_t, \mu, \sum)$ is the normal distribution of ith component with $\mu$ as the mean and $\sum_i = \sigma_i^2 I$ as the covariance of the ith component.

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta(X_t, \mu_{i,t}, \sum_{i,t}) \quad (2)$$

$$\eta(X_t, \mu, \sum) = \frac{1}{2\pi^{\frac{n}{2}} |\sum|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_t)^T \sum_k^{-1} (x-\mu_k)} \quad (3)$$

**Parameters update:**

Each new pixel value is compared with the existing K Gaussian distributions, until a match is found. If a pixel value is within 2.5 standard deviations of a distribution then it is considered as a match. Once the match is found the parameters are update by the following equations (4) to (7).

$$\omega_{k,t} = (1-\alpha)\omega_{k,t-1} + \alpha(M_{k,t}) \quad (4)$$

$$\mu_t = (1-\rho)\mu_{t-1} + \rho(X_t) \quad (5)$$

$$\sigma_t^2 = (1-\rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T(X_t - \mu_t) \quad (6)$$

$$\rho = \alpha\eta(X_t|\mu_k, \sigma_k) \quad (7)$$

Where, $\alpha$ is the learning rate, $\rho$ is the second learning rate, $M_{k,t}$ is 1 for the model that matched and 0 for other models. In case of no match found, the distribution is replaced with a new distribution using the current value as its mean, high initial variance and low prior weight.

**Background estimation:**

The K distributions are ranked as per their fitness value $\omega_k/\sigma_k$, whereas the most likely distributions remain on top and the less probable ones on the bottom , and finally replaced by a new distributions. The first B distributions are used for modelling the background using equation (8), where T is a threshold which is the measure of the minimum portion of the data that should be accounted for by the background. This background $B$ is subtracted from a current frame to obtain the foreground mask.

$$B = \arg min_b \left(\sum_{k=1}^{b} \omega_k > T\right) \quad (8)$$

The steps for classifying the machine condition using foreground estimation by GMMs method are given in Table 1**.**

| Stage | S.No. | Description |
|---|---|---|
| Fault Detection | 1 | Clear workspace variables |
| | 2 | Create system objects *(file read, player, foreground detector, blob analysis)* |
| | 3 | Initialize variables *(No. of frames for training, no. of frames to estimate max. area, ROI)* |
| | 4 | Read a frame at a time, after the last frame read go to ***step 15*** |
| | 5 | Crop ROI and detect foreground using GMMs |
| | 6 | Perform blob analysis of detected objects |
| | 7 | Compute an area and a bounding box and Output a frame to the player |
| | 8 | Is no. of frames greater than no. of frames for training? If No, go to ***step 4***. If yes, continue. |
| | 9 | Calculate the maximum blob area |
| | 10 | Is a current blob area greater the maximum blob area for 5 consecutive frames? If no, go to ***step 11***. If yes, go to ***step 12*** |
| Fault Classification | 11 | Set normal operation LED on and go to ***step 4*** |
| | 12 | Is a bounding box in lower part of the ROI? If yes, go to ***step 13*** else go to ***step 14*** |
| | 13 | Set transfer 1 jam LED on, stop processing and go to ***step 15*** |
| | 14 | Set transfer 2 jam LED on, stop processing and go to ***step 15*** |
| | 15 | Release all objects from the memory |

**Table 1:** The steps for fault detection using foreground estimation by GMMs Method

A GUI is built in MATLAB2014 for the method as shown in Figure 2**.** The machine operating conditions are indicated by three indicators, green for a normal operation and red for the transfer 1 and transfer 2 jams.

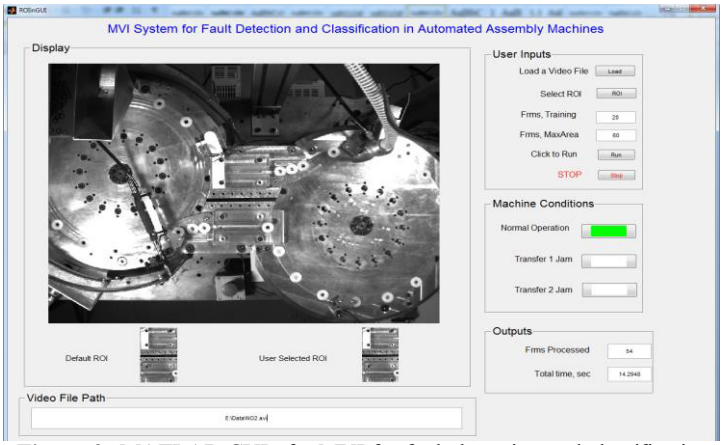


**Figure 2:** MATLAB GUI of a MVI for fault detection and classification

The results of the inspection are given in Figure 3**.** Figure 3 (a) and (d) are results for a normal operation. It can be seen that the blob area is not constant w.r.t no. of frames; this is due to non-uniform flow of carriers through the ROI as a consequence of fluctuations in the air pressure and friction. At some point, the blob area in a particular frame exceeds the threshold but it is only for a frame due to a temporary jam in the track. The method is trained not to classify this false jam as a transfer jam. Once a jam is detected, the method keeps track of it for next 5 consecutive frames and if the jam increases w.r.t time, it is considered as a true jam. In the event of jam, the jam indicator is set red as an alarm and the method stops further processing of the data. Figure 3 (b) and (e) shows transfer 1 jam and Figure 3 (c) and (f) highlights similar results but for the transfer 2 jam. The method was applied to all 15 data sets and no false positive results were reported.

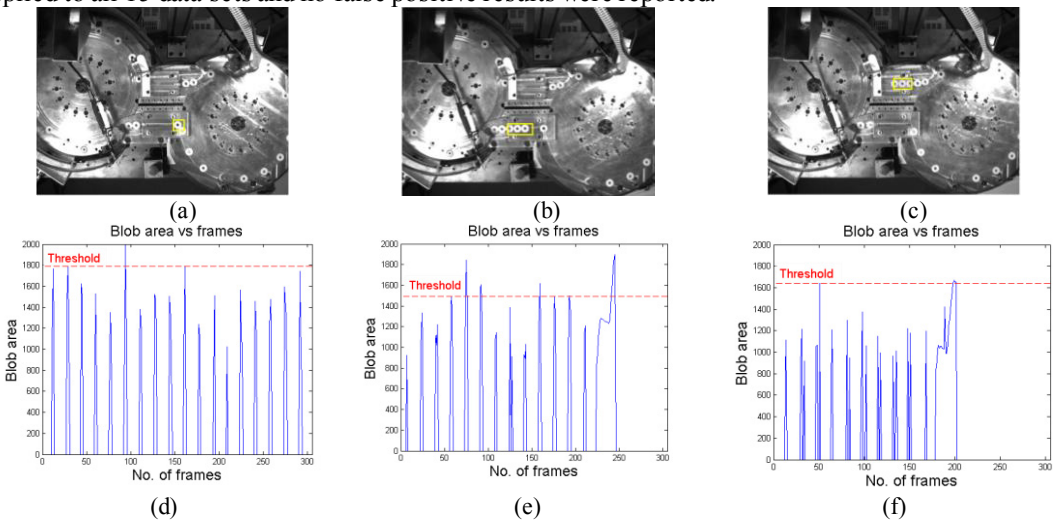


**Figure 3:** Method 1 results: (a), (d) Normal operation; (b), (e) Transfer 1 jam; (c), (f) Transfer 2 jam

## 4.2 Method 2: Fault Detection using Optical Flow Method

Optical flow is a method popularly used to estimate motion from one frame of a video to another frame. It is a two dimensional vector given by equation (9) which is the measure of the motion in x and y directions respectively.

$$[u \;\; v]^T \tag{9}$$

Optical flow between two images is calculated by solving the equation (10), where $I_x$ and $I_y$ image derivatives in horizontal and vertical direction and $I_t$ is the time derivative of the image.

$$I_x u + I_y v + I_t \tag{10}$$

The above equation is unconstrained and can be solved by either of two approaches: Horn-Schunck or the Lucas-Kanade. Horn-Schunck method is based on brightness constancy assumption which assumes that optical flow is smooth over the entire image, while Lucas-Kanade method assumes a constant velocity in each section. For fault detection using optical flow, Horn-Schunck method is used in this research. This method computes the optical flow fields by minimizing the global energy using equation (11) and iteratively solves for horizontal and vertical flow fields using equations (12) and (13).

$$E = \iint (I_x u + I_y v + I_t)^2 dxdy + \alpha \iint \left\{ \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 \right\} dxdy \tag{11}$$

$$u_{x,y}^{k+1} = u_{x,y}^{-k} - \frac{I_x [I_x u_{x,y}^{-k} + I_y v_{x,y}^{-k} + I_t}{\alpha^2 + I_x^2 + I_y^2} \tag{12},$$

$$v_{x,y}^{k+1} = v_{x,y}^{-k} - \frac{I_y [I_x u_{x,y}^{-k} + I_y v_{x,y}^{-k} + I_t}{\alpha^2 + I_x^2 + I_y^2} \tag{13}$$

In above equations, $[u_{x,y}^k, v_{x,y}^k]$ is the velocity estimate for the pixel at location $(x, y)$ and $[u_{x,y}^{-k}, v_{x,y}^{-k}]$ is the neighborhood average of $[u_{x,y}^k, v_{x,y}^k]$. Initial velocity is assumed as zero.

Using above formulas, optical flow for each machine condition video is estimated and the Optical Flow Density (OPF), which is the sum of absolute values of the optical flow fields is computed. The OPF for the normal operation of the machine is calculated from first 50 training frames. It is considered as a base line against which the current video frame's OPF is compared. If the current OPF is below the normal flow density, the machine condition is classified as a normal operation. On the other hand, if the current OPF is more than the normal flow density and increases for next consecutive frames then the machine condition is classified as a faulty condition. The location of the fault is determined depending on where the OPF has exceeded the limit. The steps of the method are given in Table 2 and the results in Figure 4.

| Stage | S. No. | Description |
|---|---|---|
| Fault Detection | 1 | Clear workspace variables |
| | 2 | Create system objects *(file read, player, optical flow, shape inserter)* |
| | 3 | Initialize variables *(No. of frames for training, optical flow matrix , ROI)* |
| | 4 | Read a frame at a time, after the last frame read go to ***step 15*** |
| | 5 | Crop ROI and detect optical flow using the optical flow object |
| | 6 | Down sample optical flow, compute absolute value and optical flow density |
| | 7 | Insert flow lines using the shape inserter and output a frame to the player |
| | 8 | Is no. of frames greater than no. of frames for training? If No, go to ***step 4***. If yes, continue to next step |
| | 9 | Calculate the normal flow density in a frame |
| | 10 | Is a current flow density greater than the normal flow density for 5 consecutive frames? If no, go to ***step 11***. If yes, go to ***step 12*** |
| Fault Classification | 11 | Set normal operation LED on and go to ***step 4*** |
| | 12 | Is a flow density in the lower part of the ROI greater than the upper part? If yes, go to ***step 13*** else go to ***step 14*** |
| | 13 | Set transfer 1 jam LED on, stop processing and go to ***step 15*** |
| | 14 | Set transfer 2 jam LED on, stop processing and go to ***step 15*** |
| | 15 | Release all objects from the memory |

**Table 2:** The steps for fault detection using optical flow method

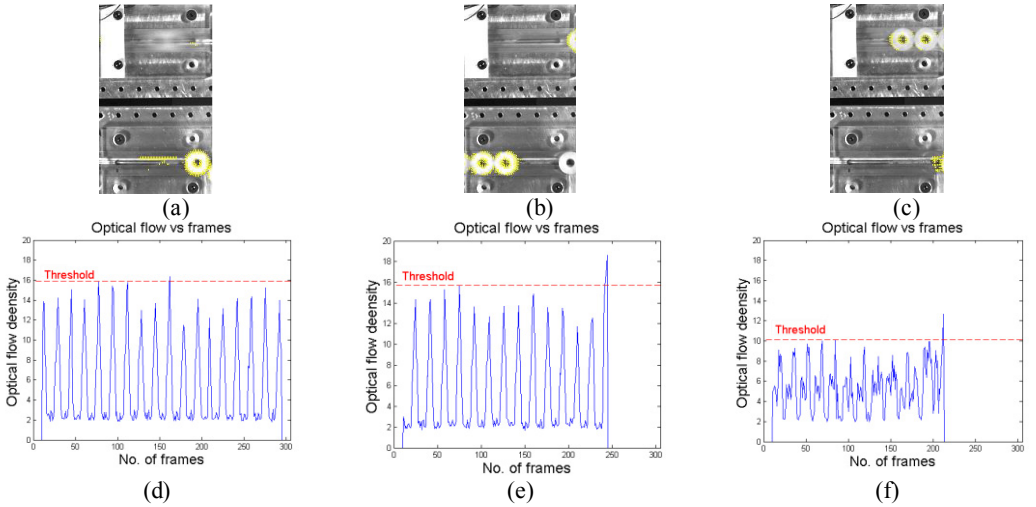


**Figure 4:** Method 2 results: (a), (d) Normal operation; (b), (e) Transfer 1 jam; (c), (f) Transfer 2 jam

The first row in the figure shows images at a given instant of time, while the second row depicts corresponding optical flow density respect to the number of frames. Figure 4 (a) and (d) are results for a normal operation. Figure 4 (b) and (e) shows transfer 1 jam. The high flow density can be seen as number of yellow lines and corresponding plots show that the flow density exceeds the threshold limit and continuously increases w.r.t time. Figure 4 (c) and (f) highlights similar results but for the transfer 2 jam. The method successfully classified the machine conditions from the video data.

## 4.3 Method 3: Fault Detection using Running Average Method

In this approach, a fault in the machine is detected by measuring the area of processed foreground from the ROI. The measured area is then compared against threshold and if it exceeds for few consecutive frames, a machine condition is classified as a jam. The machine is has a rotary motion, hence both the primary and the secondary wheel along with fasteners and assemblies move in the scene. Hence, there was a need of a technique which can properly estimate a background from a video and which is robust to the motion and minor changes in lighting conditions. The methods such as a mixture of Gaussians (Stauffer and Grimson, 2000) or mean-shift are popular but they require high computational efforts and time. The running average method with a filter size of 20 was used to estimate the background from a video. The filter size less than 20 resulted in improper estimate of the background and higher than 20 caused delay in processing. The background at any time $t$ was estimated using equation (15).

$$B_t = \left(\sum_{i=1}^{20} I_i\right)/20 \tag{15}$$

The method is very fast and has low memory requirements compared to advanced background estimation methods. The foreground has both the moving assemblies and some steady objects such as the transfer tracks. The binary image of the detected foreground is obtained by segmentation using thresholding. Morphological closing operation with a disk shape element of size 9 is carried out for noise removal and to get smooth edges of the assemblies. The ROI area, T1 area and T2 area are calculated by running average method. The areas are calculated for each frame and the values are compared with the thresholds. If it exceeds to five consecutive frames, a jam in a particular track is detected by locating the area in the ROI.

The images and corresponding area plots are given in Figure 5 **.** Figure 5 (a) and (d) are results for a normal operation. Figure 5 (b) and (e) shows transfer 1 jam and Figure 1 (c) and (f) highlights similar results but for the transfer 2 jam. The method successfully classified all the machine conditions from the video data. The steps of the method are given in Table 3**.**

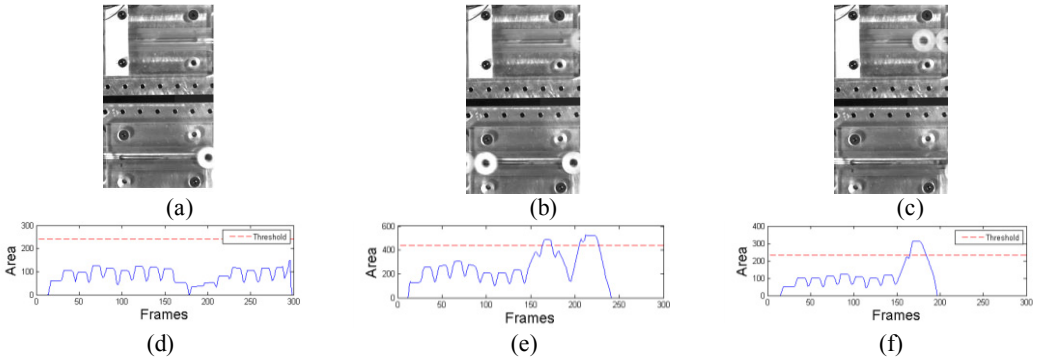


**Figure 5:** Method 3 results: (a), (d) Normal operation; (b), (e) Transfer 1 jam; (c), (f) Transfer 2 jam

| Stage | S. No. | Description |
|---|---|---|
| Fault Detection | 1 | Clear workspace variables |
| | 2 | Create system objects *(file read and a player)* |
| | 3 | Initialize variables *(Running average filter size, structuring element and ROI)* |
| | 4 | Read a frame at a time, after the last frame read go to ***step 17*** |
| | 5 | Crop ROI and apply 5 x5 Gaussian low pass filter |
| | 6 | Estimate a background frame by running average of 20 frames |
| | 7 | Calculate foreground frame by subtracting the background frame from a current frame |
| | 8 | Segmentation using thresholding and perform morphological closing operation |
| | 9 | Calculate Average ROI area, T1 area and T2 area and Output a frame to the player |
| | 10 | Is no. of frames greater than no. of frames for training? If No, go to ***step 4***. If yes, continue to next step |
| | 11 | Calculate the thresholds for ROI area, T1 area and T2 area |
| | 12 | Is average T1 or T2 area greater than the thresholds for 5 consecutive frames? If no, go to ***step 13***. If yes, go to ***step 14*** |
| Fault Classification | 13 | Set normal operation LED on and go to ***step 4*** |
| | 14 | Is average T1 area greater than T2 area? If yes, go to ***step 15*** else go to ***step 16*** |
| | 15 | Set transfer 1 jam LED on, stop processing and go to ***step 17*** |
| | 16 | Set transfer 2 jam LED on, stop processing and go to ***step 17*** |
| | 17 | Release all objects from the memory |

**Table 3:** The steps for fault detection using running average method

# 5 Results and Discussion

The performance of three methods was tested on the 15 video data sets out of which the 5 data sets were for normal operation sequence of the machine, 5 for the transfer track 1 jams and 5 for the transfer track 2 jams. To evaluate the performances, the results obtained by three methods mentioned in this paper were compared with the results obtained by STV method (Szkilnyk, 2012). The STV method gave very good classification rate but it required the training data set and it needs certain number of frames to build the STV and compare it with the normal STV for fault detection. On the

other hand, the new proposed methods do not require separate training data sets but they get trained from the stand alone data set and can classify the machine condition as a normal or a faulty. Moreover, unlike STV method, they can detect a fault within just few frames once it is introduced.

Figure 6 compares average processing time to process video data by proposed methods. The time taken to process a 10 sec of video data by method 1 was 8.47 sec, the method 2 was 6.05 sec and the method 3 was 5.88 sec. It proves that all methods are capable of detecting a fault in real time data acquisition. However, the method 3 performed very well compared to other two methods in terms of processing time. All algorithms were implemented in MATLAB 2014 and run on Intel® core(TM)i5 CPU @2.67 GHz with 4.0 GB RAM.

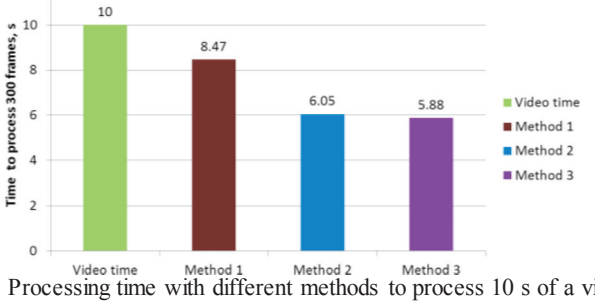


**Figure 6:** Processing time with different methods to process 10 s of a video data

Figure 7 shows comparison of the fault detection time by all methods. Method 1 and method 2 took on and average 30 frames to detect and classify a fault. It is equivalent to 1 sec of lag. This lag is acceptable to stop the further damage once a fault is detected. Both methods detected faults nearly at same time after it was introduced. The method 3 took very less time compared to previous two methods to detect and classify a fault after it was introduced. For example, in a video data of T1JAM 1, the fault was introduced at 222$^{nd}$ frame and it was detected at 244$^{th}$ frame by method 1, at 243$^{rd}$ frame by method 2 and at 226$^{th}$ frame by method 3.

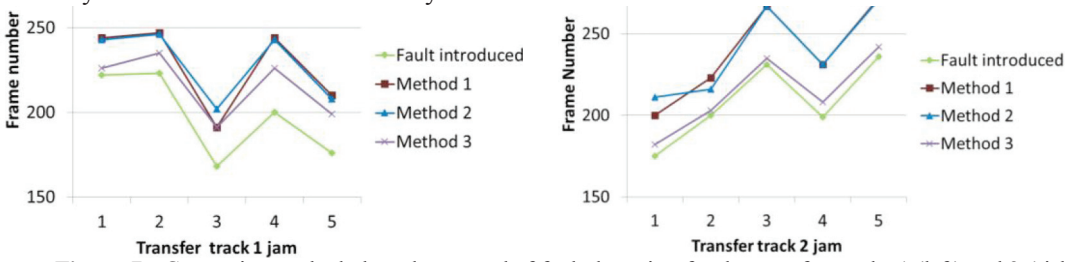


**Figure 7:** Comparing methods based on speed of fault detection for the transfer tracks 1 (left) and 2 (right)

All methods have successfully classified the machine condition without any false positives. The processing time can be reduced further by using high speed processer machine. It was observed that the display command in MATLAB significantly contributed to increase in the processing time. Hence, a parallel processing without display commands and separate code for displaying images may considerably improve the performance by reducing the processing time.

# 6 Conclusion

In this paper, three classification methods for machine vision based fault detection in a high speed assembly machine were studied. The methods were 1) Gaussian mixture models with blob analysis, 2) optical flow and 3) running average. The results were compared with those of a previously documented method that was based upon spatiotemporal volumes (STV). Unlike STV, the new

methods did not require a separate training data set to learn normal operating conditions. Instead, they were able to learn the normal behavior from the initial frames of a video data set. STV was also found to be resource intensive in terms of memory requirements and processing time. The new methods were able to classify a machine condition from a standalone video data set without prior training and were much faster in terms of processing time and fault detection time. It was concluded that all three methods were able to correctly detect whether the targeted fault (transfer jam) had occurred. However, the running average method was considered the best because it took the least time to process video data sets and it detected faults as soon as they were introduced.

Plans for future work include widening the region of interest and expanding on the types of faults to be detected. Thus, the methods would be tested on their ability to detect additional faults such as hopper jams, missing carriers and missing O-rings.

# References

Bhuvanesh, A. and Ratnam, M.M. (2007). Automatic detection of stamping defects in leadframes using machine vision: Overcoming translational and rotational misalignment. *The International Journal of Advanced Manufacturing Technology*, *32*(11), 1201-1210.

Boothroyd, G. (2005). *Assembly Automation and Product Design*. 2nd Edition, Florence: Marcel Dekker.

Chauhan, V., Fernando, H., Surgenor, B. (2014). Effect of Illumination Techniques on Machine Vision Inspection for Automated Assembly Machines, *Proceedings of The Canadian Society for Mechanical Engineering (CSME) International Congress 2014*, 1-6.

Demetgul, M., Tansel, I.N. and Taskin, S. (2009). Fault diagnosis of pneumatic systems with artificial neural network algorithms. *Expert Systems with Applications*, *36*(7),10512-10519.

Patel E. and Shukla D. (2013). Comparison of Optical Flow Algorithms for Speed Determination of Moving Objects. *International Journal of Computer Applications*, *63*(5), 32-37.

Hughes, K., Fernando, H., Szkilnyk, G., Surgenor, B. and Greeenspan, M. (2014). Video event detection for fault monitoring in assembly automation. *International Journal of Intelligent Systems Technologies and Applications*, *13*(1), 103-116.

Isermann, R. (2006). Fault-diagnosis systems: an introduction from fault detection to fault tolerance. *Berlin ; New York: Springer,* 197-229.

Ke, Y., Sukthankar, R. and Hebert, M., 2007. Event detection in crowded videos. *IEEE 11th International Conference on Computer Vision,ICCV 2007*, 1-8.

Killling, J., Surgenor, B. and Mechefske, C. (2009). A machine vision system for the detection of missing fasteners on steel stampings. *The International Journal Of Advanced Manufacturing Technology, 41*(7-8), 808-819.

Klein, R., Masad, E., Rudyk, E. and Winkler, I. (2014). Bearing diagnostics using image processing methods. *Mechanical Systems and Signal Processing*, *45*(1), pp. 105-113.

Konrad, H. (1996). Fault detection in milling, using parameter estimation and classification methods. *Control Engineering Practice*, *4*(11), 1573-1578.

Malge, P. S. and Nadaf, R. S. (2014). PCB Defect Detection, Classification and Localization using Mathematical Morphology and Image Processing Tools. *International Journal of Computer Applications*, *87*(9), 1015-1021.

Sekar, R., Hsieh, S. and Wu, Z. (2011). Remote diagnosis design for a PLC-based automated system: 1-implementation of three levels of architectures. *The International Journal of Advanced Manufacturing Technology*, *57*(5-8), 683-700.

Shahabi, H.H. and Ratnam, M.M. (2009). In-cycle monitoring of tool nose wear and surface roughness of turned parts using machine vision. *The International Journal of Advanced Manufacturing Technology, 40*(11), 1148-1157.

Stauffer, C. and Grimson, W.E.L. (2000). Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(8), 747-757.

Szkilnyk, G., Hughes, K., Fernando, H. and Surgenor, B. (2012). Spatiotemporal volume video event detection for fault monitoring in assembly automation. *19th International Conference on Mechatronics and Machine Vision in Practice (M2VIP),* 20-25.

Szkilnyk, G., 2012. Vision-based Fault Detection in Assembly Automation. *M. A. Sc. Thesis*, *Queen's University*, Kingston, ON.

Usamentiaga, R., Molleda, J., Garcia, D.F., Bulnes, F.G. and Perez, J.M. (2013). Jam Detector for Steel Pickling Lines Using Machine Vision. *IEEE Transactions on Industry Applications*, *49*(5), 1954-1961.

Viswanadham, N. and Johnson, T.L. (1988). Fault detection and diagnosis of automated manufacturing systems. *Proceedings of the 27th IEEE Conference on Decision and Control*, *3*, 2301-2306.

Xiaokun, L. and Porikli, F.M. (2004). A hidden Markov model framework for traffic event detection using video features. *International Conference on Image Processing, ICIP '04, 5*, 2901-2904.

Zezhi, C., Pears, N., Freeman, M. and Austin, J. (2014). A Gaussian mixturemodel and support vector machine approach to vehicle type and colour classification. *Intelligent Transport Systems, IET, 8*(2), 135-144.