A Real Time Object Recognition and Counting System for Smart Industrial Camera Sensor

Shih-Hsiung Lee and Chu-Sing Yang

***Abstract*— In the industry 4.0, factories around the world grow automated and intelligent, and where smart camera plays an important role. Smart camera is equipped with processor, memory, communication interface, and operating system, so it can process large amounts of data in advance to assist follow-up automatic inspection and judgment. Additionally, since smart camera is an independent system, it will not affect the original system of factories, which is an immense advantage in troubleshooting. Besides, thanks to technology breakthroughs in recent years, using Graphics Processing Unit (GPU) to implementing tons of parallel computing helps to significantly boost the overall efficiency. Therefore, when a rising number of factories consider improving production capacity of production lines, how to use GPU to assist the improvement is an important issue. Based on this scenario, this paper used NVidia Tegra TX1 platform with 256 GPU CUDA cores and Quad-core ARM Cortex A57 processor and Basler USB 3.0 industrial camera to simulate a smart industrial camera, which has GPU and can perform a myriad of complex computations. This paper designed how to recognize and count objects in a real time manner in a high-speed industrial inspection environment with large volumes of data, so as to verify the concept (smart camera with GPU cores) we proposed. The experimental results proved our ideas, and the software design architecture provided in this paper is a simple and efficient design. In the future application in the Internet of Things or the Internet of Everything, this structure can be a valuable reference.**

***Index Terms*— Smart industrial camera, AOI, GPU.**

I. INTRODUCTION

IN RECENT years, the machine vision system in industrial inspection has gradually become modular, and the world's manufacturers have provided corresponding solutions to simple machines as well as to complex turnkey machine. In the domain of machine vision, system can be divided into three subsystems: image sensor, lighting system and image processor. The current development trend is to integrate image sensor and processor in the same system and to become modular as a smart camera. Smart camera integrates image sensor (CCD or CMOS), lens (C, CS, F or M-mount), DSP (Digital signal processor), light source (LED), communication interface (RS232, RS485, Ethernet or PLC Link), I/O interface and other related preprocessing software design programs. Since smart camera is provided to users for independent parameter setting and calibration, whether it is easy to use depends on the software applications offered by the manufacturers. Manufacturers usually provide users with manuals of application programming interfaces, documentations, quick setting for handling events and exceptions and other mechanisms, or even provide source codes. As a result, smart camera is featured by RAD (Rapid Application Development) and has the advantages of quick completion and modification. Besides, smart camera can adjust parameters to suit different applications and is flexible and cost-effective, like [1] proposing a low-power analog smart camera sensor for edge detection.

There is one indispensable key system in the development of machine vision system, that is, the lighting system. Capturing a good image means that the desired objects must be clearly shown, and the technology involved includes positioning of spare parts, right lighting and appropriate camera system. For example, low-angle lighting must be used to check whether a bright metal surface is scratched, otherwise the image only shows the reflected light from the surface of workpiece. In addition, due to the impact of other light sources in the environment, how to select appropriate lamps and how to overcome the interference of other light sources are also among the technology. Lighting is science and art, and there is no law to be followed presently. In respect of the lighting system proposed in this paper, a suitable way of installation was found through a number of experiments to successfully prevent the interference of other light sources.

The machine vision system needs to apply an array of image processing technology. Specifically, in the image capturing technology, camera calibration, system calibration, and coordinate correction must be completed before images are captured. Image preprocessing technology encompasses image filtering, image denoising, image Fourier transformation, image stitching and other processing.

Image segmentation technology includes segmentation of foreground and background and often uses Otsu's method and threshold level for image segmentation [2]. Image edge detection technology includes first-order derivative operators, such as Sobel [3], Prewitt, Robert et al [4], and second-order derivative operators like Laplacian [5] and LOG (Laplacian of Gaussian) [6], etc.

Image feature extraction technology includes BRIEF (Binary Description Very Fast) [7], BRISK (Binary Robust Invariant Scalable Keypoint) [8], FREAK (Fast Retina Keypoint) [9], ORB (Oriented FAST and Rotated BRIEF) [10], SIFT (Scale-Invarient Feature Transform) [11], and SURF (Speeded Up Robust Feature) [12], etc. Besides, the technology used in the machine vision system also includes dimensional measurement [13], classification technology [14], position or alignment [15], registration [16], etc. However, with the advances in image sensing technology and require-

Manuscript received November 15, 2016; revised January 19, 2017; accepted February 12, 2017. Date of publication February 17, 2017; date of current version March 22, 2017. The associate editor coordinating the review of this paper and approving it for publication was Dr. Akshya Swain.

The authors are with the Institute of Computer and Communication Engineering, National Cheng Kung University, Tainan 701, Taiwan (e-mail: beargoer@gmail.com).

Digital Object Identifier 10.1109/JSEN.2017.2671457

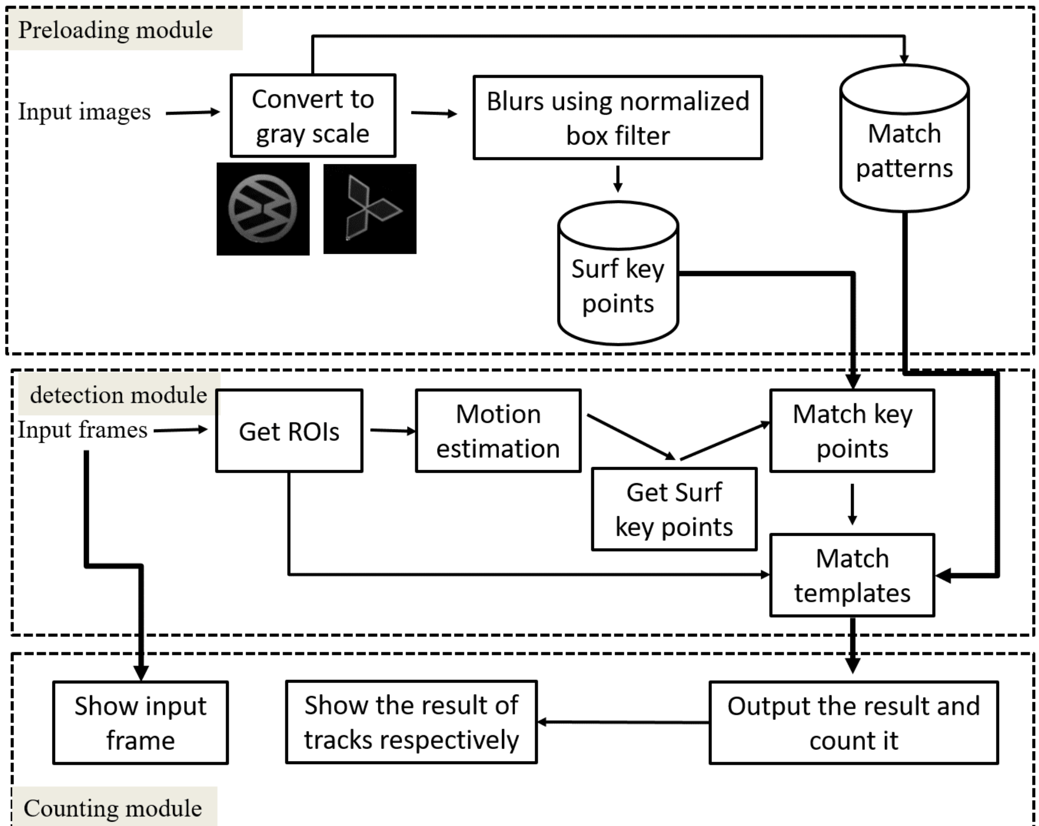

Fig. 1. Overview of designed system.

ments detection accuracy, these applications process high-resolution image data and complex algorithms.

Real-time on-line analysis poses an enormous challenge. The definition of real-time varies in different industries. Generally, real-time in image streaming means that image data of at least 30 frames can be processed in one second, and the data in the industry is definitely far greater than this figure. According to definition of specification, the experimental equipment in this paper can output data about 164 1920 * 1200 pixel gray-scale images per second. Under this demand, this paper used the NVidia Jetson TX1 platform with 256 Graphics Processing Unit (GPU) CUDA cores and Quad-core ARM Cortex A57 processor, Basler USB 3.0 Industrial camera and machine vision lens to create a smart industrial camera which is equipped with GPU and can perform large numbers of complex computations. In addition, in the course of mounting the camera, we considered the design of light source and mounting height of light source and camera, and how to adjust light source and camera parameters to prevent the interference of other light sources. For frame rate of images, we considered that a few clear pictures could be captured at high speed as the data sources of detection. In the design process of image algorithms, we regarded SURF as the feature identified by our objects. Although image features extraction method in recognition has an outstanding performance. However, using the key points match method in the high-resolution image data consumes a lot of computing power. According to our observation, a 1980 * 1200 pixels image was input for capture by SURF key points, 1 image could be processed per second with GPU core. But without GPU core, only 1 image could be processed in 2 to 3 seconds. This is one of main reasons that we proposed the system with GPU core. Hence, considering the size range selection of ROI (Region of Interest) plays an important role in the design which can be seen in the experiment part. In the design course of software architecture, we considered the ways of communication between CPU and GPU to prevent heterogeneous cores from resulting in reduced efficiency in the copy of memory, and the arrangement of multithreading to achieve high efficiency with the least number of threads. Aside from correctly recognizing and counting objects, this paper also considered the efficiency of rendering, and proposed a simple and efficient design.

Most of the applications in the AOI industry are choosing PC-based or customized system by using the DSP (Digital Signal Processor) or GPU card to accelerate the performance. The embedded system with GPU for AOI industry is brand new concept. This concept is not only parallel computing merit but low power consumption. In the age of internet of things, smart end-device or smart sensor is a very important issue. Hence, using the embedded system to design a smart sensor has the advantage of portable and realizing the deep learning with GPU core. Using less power consumption, get more intelligent feedback than traditional design. The GPU and embedded system exactly play an important role in the future. Our main contribution of this paper proposed a scenario and

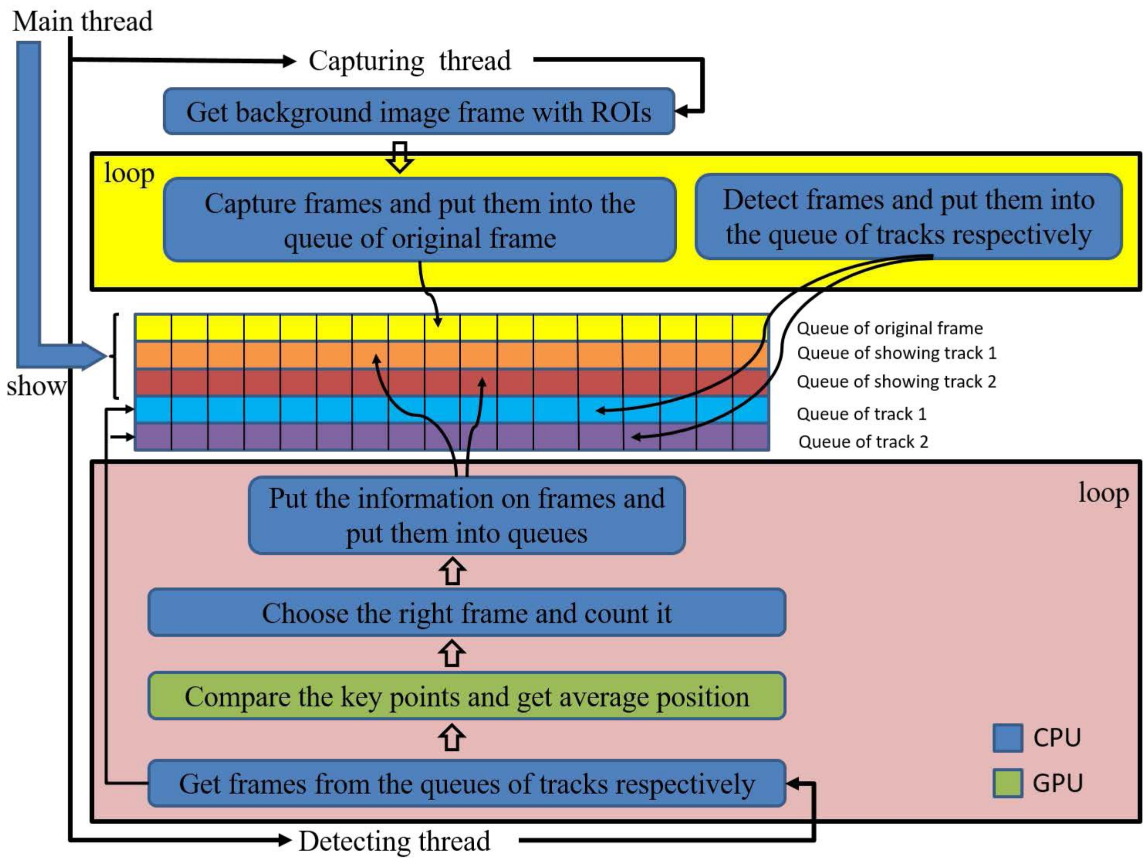

Fig. 2. Overview of theading of designed system.

design the system to prove this fresh concept. We disclosed the very detailed design method in order to help user refer to our system to develop more efficient applications.

The rest of this paper is organized as follows. Sytem overview and preloading are presented in Section II and III. The proposed algorithm is presented in Section IV and V, respectively. The experimental results are discussed in Section VI. Section VII. concludes the paper and offers suggestions for future studies.

II. System Overview

Fig. 1 illustrated the flow diagram of the proposed system, which includes three stages. In the preloading stage, the object images to be identified were converted into gray-scale images and stored in the match patterns database. Normalized box filter was used to blur the images, and then the SURF key points of images were captured and stored in the database. In the detection stage, we employed a motion estimation method to select the images to be detected. The image size was the range captured by ROI (region of interest) and to be judged. Images captured by key points were selected and match was carried out. Then, the average center point of the key points meeting the match was calculated. Besides, the detected images and patterns were compared with the match template to obtain the scope of the match. The basis for output results was whether the average center point fell within the scope of the match template. Finally, the system will output the original image and the result and counting of the detected objects.

In order to effectively boost the efficiency, the system introduced the design of multithreading. Fig. 2 illustrated the arrangement of system threads and the design of queues. Once the system was started, two threads and five queues were generated. The first thread was the capturing thread which can capture images from an industrial camera. In the first execution, the thread would first capture the background image of the track as a reference of motion estimation. After that, the images of the same size as the original image would be placed in the queue. Moreover, in accordance with ROI settings, images were captured to judge whether there were objects. If so, the camera was placed in the respective queue. Because we knew the location of ROI, we could place the image to be judged in the respective queue. The second thread was responsible for identifying whether objects occurred and for counting the number of times. Detecting thread would output images and judgment results from the queues of tracks, and would draw this result information on the image and place it into the corresponding showing queues. Finally, main thread obtained images from queues and output them on the screen. The blue part in Fig. 2 was CPU operation while the green part was GPU operation. Given the reduced efficiency caused by copying memory by the heterogeneous cores, this paper would use GPU in the most resource-consuming SURF features capture and match operation.

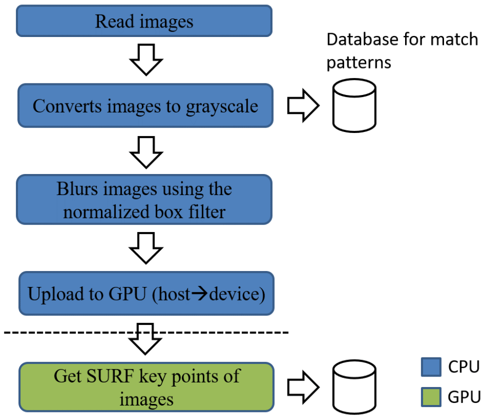

Fig. 3. Preload the features.

III. Preloading

The method proposed in this paper is to regard SURF as features to determine the type of objects and the recognition algorithm runs in the GPU. Unnecessary repeated capture of features can be avoided through preloading mechanism. The object images to be recognized was read and converted into a gray-scale image, copied and stored in the match patterns database. Subsequently, normalized box filter was used to blur the images and transmitted to the GPU. Capture algorithms of SURF key points were carried out in the GPU. Finally, the key points of the images were stored in the GPU memory and used by the algorithms of recognizing objects. The process is shown in Fig. 3.

IV. Motion Estimation for Dropping Frames

The image capture of industrial cameras is exceedingly fast. The camera used in this paper can output image data of 164 1920 * 1200 pixels gray-scale images per second. In this experiment, the camera was set to output 100 1920 * 1200 pixels gray-scale images per second. Generally, in case of no objects, the system did not need to perform object recognition. In order to achieve this effect, this paper proposed a simple motion estimation for dropping frames approach. The background images of the track captured at the beginning were regarded as the absolute difference between reference images and captured images to obtain image information about difference. Normalized box filter was used to blur the images and to find the number of contours in the images. If the number is greater than the threshold, it means that there is an object and this image needs to be judged and will be placed in the corresponding queue. This step plays a crucial role in the overall system. How to reduce unnecessary operation is one of the best ways to enhance the overall efficiency. The process is shown in Fig. 4.

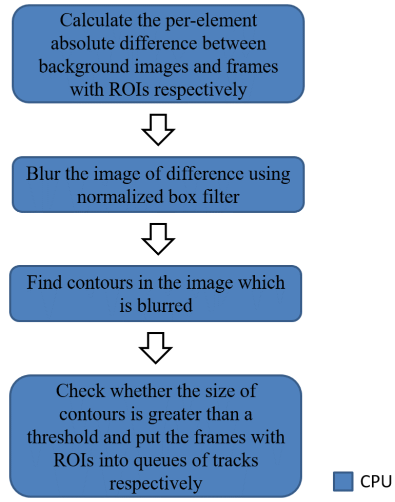

Fig. 4. The flow of motion estimation for dropping frames.

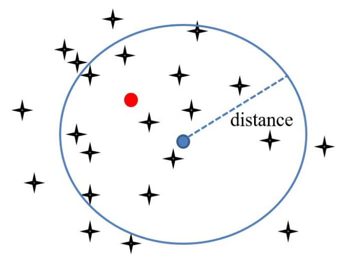

Fig. 5. The distance of key points.

V. Real Time Detection and Counting

A. Object Recognition

The object images to be judged are captured from the queue and transmit them to the GPU. Capture algorithms of SURF key points were carried out in the GPU. Brute force matcher method was used to compare the key points with key points which are in preloaded database. The result of the match was transferred to the CPU. In examining the matched key points, it is checked whether the number of key points, whose distance was less than the threshold set by the system, was greater than a certain number, which was the rule of thumb. In case of less than this number, it means that the object did not appear in the screen. Next was to calculate the average center points of these key points distance which was less than the threshold. It is shown in Fig. 5. The meaning of distance is to describe the similarity between key points in two images. The greater the distance is, the lower the similarity is, and vice versa. The output average center point is used for judgment in the next stage. The overall process is shown in Fig. 6.

B. Matching Template

When the objects entered into the detection area, match template was used to search whether there were objects to be detected in the image and bounding box was output. This paper expects to present the best detection image output on the screen and develop object area judgment methods to achieve the complete presentation effects of detection objects. Let R is the area where the expected object appears, while B is the bounding box output by match template, and c is the average center point of the key points. The output image satisfies the condition of $\{ B$ in R and c in $B \}$. The process is shown

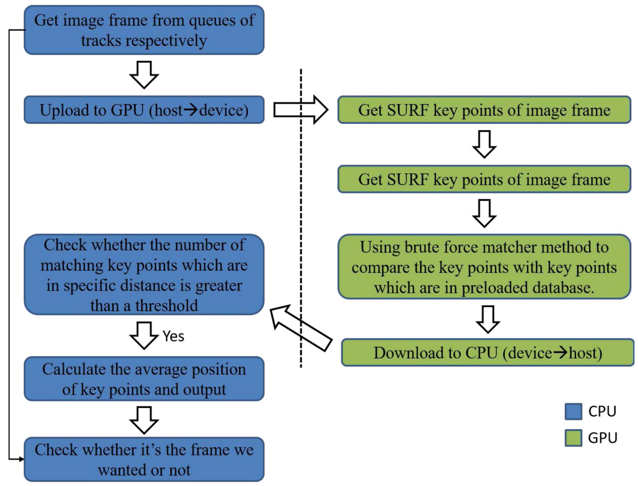

Fig. 6. The flow of detection.

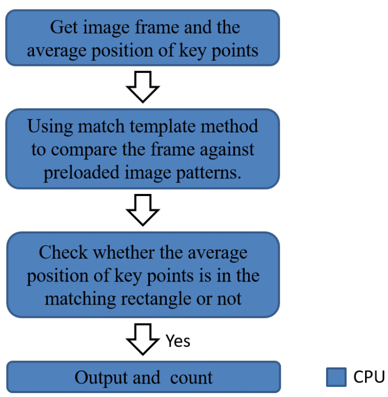

Fig. 7. The flow of matching template.

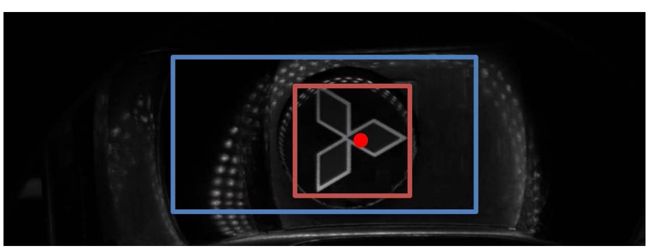

Fig. 8. The situation satisfies the condition.

in Fig. 7. Fig. 8 presented the situations that satisfy the condition. The blue box is the area where the expected object appears, while the red box is the bounding box and the red dot is the average center point of the key points. Fig. 9 presented the situations that do not satisfy the condition.

C. Counting

If the objects entered into the detection area and meet the output conditions, output of approximately 2 to 3 frames would be obtained in the experiment of this paper. 2-3 frames only indicate that the object appears once, and in counting, the repeated frames must be eliminated. The repeated object images are counted via the concept of time difference.

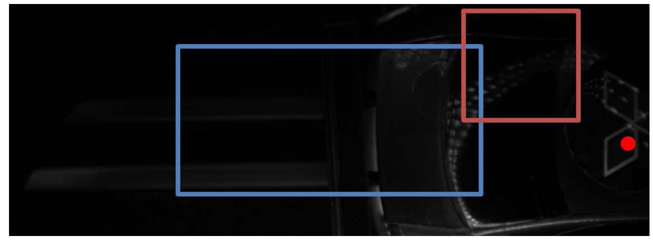

Fig. 9. The situation does not satisfy the condition.

To be more specific, the object images on the same track are continuously counted, and the time difference obtained by subtracting the timestamp of the previous counting from that of this counting must be greater than a threshold. As a result, there will not be repeated counting. The experiment also proved that this method is feasible and fairly simple. The resulting information is drawn on the output image and placed in the corresponding showing queues. Finally, the system gains the images from the queues and outputs to the screen in order to present the best results.

VI. Experimental Result and Environment Setup

This paper used the AOI (Automatic Optical Inspection) environment for experiment. This chapter explores the design of scenarios and the problems encountered in experiment.

A. Scenario Design

This paper simulated an AOI (Automatic Optical Inspection) experimental scenario. Track racing car toys [17] were used to create the scenario of online production lines and one track represented one production line. This experiment had two production lines. The car logo stickers were pasted on the car roof as the detection objects and the brand logo of Volkswagen and Mitsubishi was used. Also, light source and industrial cameras were set up as a detection station. Light source was medium-angle white outer ring light source with an outside diameter of 132mm and an inside diameter of 98mm. To avoid flicker of ambient light and for stable current control, APWLC-24V 2 channels 24V frequency-flash light source controller was employed. The industrial camera was the Basler acA1920-155um USB 3.0 camera with the Sony IMX174 CMOS sensor and delivered 164 frames per second at 2.3 MP resolution [18]. Additionally, a Tokina 3 mega pixel lens (TC2514-3MP) was mounted in front of the camera [19]. The industrial camera was connected by USB 3.0 port to NVidia Jetson TX1 [20] for detection, and output the experimental results to the screen via HDMI. NVidia Jetson TX1 is equipped with 256 GPU CUDA cores and Quad-core ARM Cortex A57 processor and is an embedded system with impressive operation efficiency. The experimental equipments and scenario is shown in Fig. 10. The settings of the industrial camera are listed in Table I.

B. Probe Into Light Source and Object Speed

In planning lighting system, we are not professional manufacturers and have no laws to follow. We tried a lot of lights

TABLE I
PARAMETER SETTINGS FOR INDUSTRIAL CAMERA

Settings:	Value:
Mininum exposure time of the camera	34000ms
Maximum frame rate of the camera	100 frames
Maximum number of buffers	5

Fig. 10. The experimental equipments and scenario.

and general table lamps, including yellow light, white light, LED, etc. In the fast image capturing by industrial cameras, the flicker of light sources will affect recognition and result in poor effects. Finally, we used an industrial ring light source and frequency-flash light source controller to achieve remarkable results. As long as the lighting distance was proper, even under the exposure by other light sources beside the industrial camera, the image capture would not be troubled by the flicker of ambient light sources.

The object speed is controllable on the track. The setting of the speed is also a trick. If the objects run on the track at the fastest speed, industrial cameras can capture about 3 to 5 images of the objects. At the slowest speed, they can probably capture 12 images of the objects. The so-called object images in this paper were the body of racing cars. Fig. 11 showed that the industrial camera captured 3 to 5 images of the objects. As can be observed from Fig. 11, the objects may be missed if the cars run at the fastest speed. Therefore, we adjusted the speed so that the industrial cameras can capture about 7 object images, as shown in Fig. 12. It can be seen from Fig. 12 that the complete objects (the logos of the car) would stably appear in the image. Besides, SURF key points for capturing images are exceedingly resource-consuming despite the operation via GPU. With regard to the experiment set in this experiment, if a 960 * 600 pixel image was input for capture by SURF key points, only 4 images could be processed per second. If an 800*290 pixel image was input, 14 images could be processed per second. As a result, in this experiment, 7 object images were input continuously, the image capture could be completed at least in 0.5 seconds. Therefore, considering the size of ROI (Region of Interest) can more efficiently identify and count images.

Fig. 11. Objects at the fastest speed.

Fig. 12. Objects at the lower speed.

C. Experimental Results

Based on the environment we set up, the experimental results in this paper are 100% accurate in identifying and counting objects. The results of output picture are shown in Fig.13. SOP was provided in the environmental setting methods in this paper, so in any environment, re-constructing experimental scenarios can present 100% accurate results. The reason is that we considered mounting problems and control of object speed when designing the experiment. What is the most important is the control of light source, and the impact of ambient light should be minimized under any circumstances. The best supporting evidence is that these experimental equipment were dissembled and packaged and taken by air to Germany to participate in the 2016 Embedded World Exhibition [21]. In the exhibition, 100% experimental results were presented. The video link is https://www.youtube.com/watch?v=oNlnmQ7HMsU.

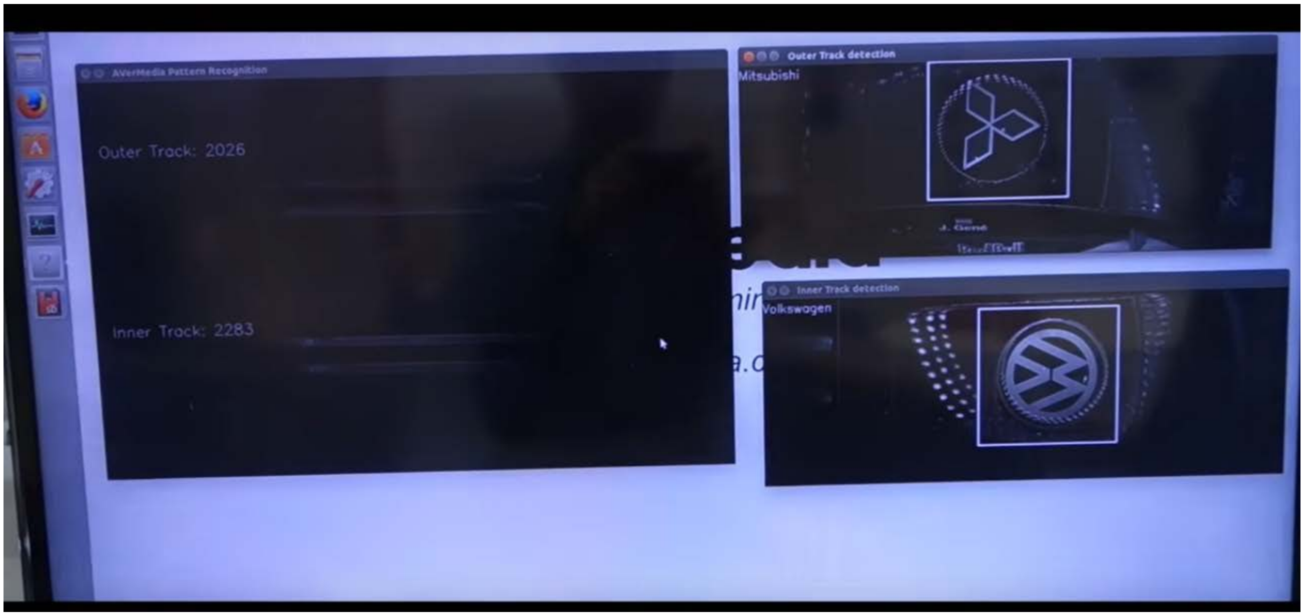

Fig. 13. Overview.

VII. Conclusion

This paper used the NVidia Jetson TX1 platform with 256 GPU CUDA cores and Quad-core ARM Cortex A57 processor, Basler USB 3.0 Industrial camera and machine vision lens to create a smart industrial camera which is equipped with GPU and can perform large numbers of complex computations. In the design process of image algorithms, we regarded SURF as the feature identified by our objects, and considered the size range selection of ROI (Region of Interest). In the design course of software architecture, we considered the ways of communication between CPU and GPU to prevent heterogeneous cores from resulting in reduced efficiency in the copy of memory, and the arrangement of multithreading to achieve high efficiency with the least number of threads. Aside from correctly recognizing and counting objects, this paper also considered the efficiency of rendering, and proposed a simple and efficient design. It is also proved by the experimental results that the discussions and methods in this paper and application of GPU to smart industrial camera is a possible direction. The future research work will focus on studying how to provide more intelligent analytical ability by applying deep learning to smart industrial camera.

Acknowledgment

The authors would like to extend their special gratitude to AVerMedia Technology Ltd. for its sponsorship and support and to the Vice President, Jiun-Kuei Jung, for his support and encouragement. Also, they are also indebted to the participants of the 2016 German Embedded World Exhibition for their valuable feedbacks.

References

[1] C. Soell, L. Shi, J. Roeber, M. Reichenbach, R. Weigel, and A. Hagelauer, “Low-power analog smart camera sensor for edge detection,” in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 4408–4412.

[2] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.

[3] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, “Design of an image edge detection filter using the Sobel operator,” *IEEE J. Solid-State Circuits*, vol. 23, no. 2, pp. 358–367, Apr. 1998.

[4] G. N. Chaple, R. D. Daruwala, and M. S. Gofane, “Comparisions of Robert, Prewitt, Sobel operator based edge detection methods for real time uses on FPGA,” in *Proc. Int. Conf. Technol. Sustain. Develop. (ICTSD)*, Feb. 2015, pp. 1–4.

[5] C. Xiansheng, “An edge detection new algorithm based on Laplacian operator,” in *Proc. IEEE 3rd Int. Conf. Commun. Softw. Netw. (ICCSN)*, May 2011, pp. 202–206.

[6] A. Anand, S. S. Tripathy, and R. S. Kumar, “An improved edge detection using morphological Laplacian of Gaussian operator,” in *Proc. 2nd Int. Conf. Signal Process. Integr. Netw. (SPIN)*, Feb. 2015, pp. 532–536.

[7] C. F. Liew and T. Yairi, “Generalized BRIEF: A novel fast feature extraction method for robust hand detection,” in *Proc. 22nd Int. Conf. Pattern Recognit. (ICPR)*, 2014, pp. 3014–3019.

[8] S. Leutenegger, M. Chli, and R. Y. Siegwart, “BRISK: Binary robust invariant scalable keypoints,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2548–2555.

[9] A. Alahi, R. Ortiz, and P. Vandergheynst, “FREAK: Fast retina keypoint,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 510–517.

[10] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2564–2571.

[11] J.-M. Morel and G. Yu, “ASIFT: A new framework for fully affine invariant image comparison,” *SIAM J. Imag. Sci.*, vol. 2, no. 2, pp. 438–469, 2009.

[12] H. Bay, T. Tuytelaars, and L. Van Gool, “SURF: Speeded up robust features,” in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 404–417.

[13] Y.-C. Wang, J.-C. Lin, and S.-F. Chiu, “The automatic image inspection system for measuring dimensional parameters of a saw blade,” in *Proc. 8th IEEE Int. Conf. Control Autom. (ICCA)*, Jun. 2010, pp. 1557–1561.

[14] G. Duan, Y.-W. Chen, and T. Sakekawa, “Automatic optical inspection of micro drill bit in printed circuit board manufacturing based on pattern classification,” in *Proc. IEEE Instrum. Meas. Technol. Conf. (IMTC)*, May 2008, pp. 279–283.

[15] Y.-C. Lin and K. Wang, “Position determination of a ball grid array by automated optical inspection method,” in *Proc. 9th IEEE Int. Conf. Nano/Micro Eng. Molecular Syst. (NEMS)*, Apr. 2014, pp. 97–101.

[16] J. R. Evans and T. Arslan, “Implementation of a robust image registration algorithm on an ARM system-on-chip platform,” in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2002, pp. 269–272.

[17] *Racing Car Toys*, accessed on Feb. 27, 2017. [Online]. Available: https://www.amazon.com/Artin-Scale-Ultimate-Express-Racing/dp/B013INM0UI/ref=sr
[18] *Basler Industrial Camera*, accessed on Feb. 27, 2017. [Online]. Available: http://www.baslerweb.com/en/products/cameras/area-scan-cameras/ace/aca1920-155um
[19] *Tokina Machine Vision Lens*, accessed on Feb. 27, 2017. [Online]. Available: http://www.tokina.co.jp/en/security/machine-vision-lenses/tc2514-3mp.html
[20] *NVidia Jetson TX1*, accessed on Feb. 27, 2017. [Online]. Available: http://www.nvidia.com/object/jetson-tx1-dev-kit.html
[21] *Embedded World Exhibition*, accessed on Feb. 27, 2017. [Online]. Available: https://www.embedded-world.de/en

Shih-Hsiung Lee received the B.Sc. degree from the Department of Applied Mathematics, National Chung Hsing University, in 2007, and the M.Sc. degree from the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan, in 2009, where he is currently pursuing the Ph.D. degree in electrical engineering with the Institute of Computer and Communication Engineering. His research interests include Internet of Things, intelligent computing, and signal processing.

Chu-Sing Yang received the B.Sc. degree in engineering science and the M.Sc. and Ph.D. degrees in electrical engineering from National Cheng Kung University, Tainan, Taiwan, in 1976, 1984, and 1987, respectively. Since 1993, he has been a Professor with the Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan. He was the Chair of the Department of Computer Science and Engineering, National Sun Yat-sen University, from 1995 to 1999, and the Director of the Computer Center, National Sun Yat-sen University, from 1998 to 2002. He joined the Faculty of the Department of Electrical Engineering, National Sun Yat-sen University, as an Associate Professor, in 1988. He is currently a Professor of Electrical Engineering with the Institute of Computer and Communication Engineering, National Cheng Kung University. He was the Program Chair of ICS-96 and the Program Co-Chair of ICPP-2003 and MTPP-2010. He joined the Faculty of the Department of Electrical Engineering, National Cheng Kung University, as a Professor, in 2006. He participated in the design and deployment of Taiwan Advanced Research and Education Network and served as the Deputy Director of the National Center for High-Performance Computing, Taiwan, from 2007 to 2008. His research interests include future classroom/meeting room, intelligent computing, and network virtualization.