Continuous Real-Time Optical Measuring of Strip Width and Edge Inspection in Stainless Steel Production Lines

Juan M. Cañero-Nieto*1, Carlos G. Spinola2, G. Moreno-Aranda3, M. J. Martín-Vázquez4, J. M. Bonelo5

1, 2, 4 Department of Electronics, University of Malaga, Boulevar Louis Pasteur 35, 29071-Málaga, Spain

1, 3 T.C.C., S.A. Robinson Crusoe 5, Local 2, 29006 Málaga, Spain

5 University of Cádiz. Algeciras (Cádiz), Spain

*1 jmcanero@ctima.uma.es; 2, 3 cspinola@uma.es; 4 martinv@ctima.uma.es; 5 josemaria.bonelo@uca.es

***Abstract*-An automated optical inspection (AOI) system designed to continuously inspect the width and dual edges of a stainless steel strip is presented in this paper. The study follows this design from its initial conception to its installation, calibration and integration on a production line. The inspection process is based on real time image processing, which measures the strip width, while detects defects and anomalies that could appear at the perimeter of the strip. This is done by processing the acquired images using two linear cameras. Next, the geometrical model and required adjustment are presented. This virtual instrument has been installed for quality control purposes in an annealing and pickling line of a stainless steel factory.**

Keywords- Automatic Optical Inspection; Image Defect Detection; Image Processing; Size Measurement; Calibration

I. INTRODUCTION

Nowadays, stainless steel is a metal alloy with great added value, and is appropriate for many applications. After the melting shop, where the precise chemical composition is attained, the produced stainless steel is usually shaped at a hot mill where it is rolled, which forms strip coils of different widths, diameters, and lengths (Fig. 1). The strip width and thickness must be accurately supervised to fulfill the dimensional tolerance specified by the customers [1]. Moreover, it is mandatory to detect any kind of anomaly on the surface and edges of the steel coils to produce the exceptional quality expected by the market and to also to avoid trouble in subsequent processing lines (e.g., wearing out and breaking of machine elements, breakage of the strip, etc.).

Fig. 1 Stainless steel coils

Due to the width range of the strip coils, from 900 to $1600\ mm$, and the rough industrial environment found on the production of the iron and steel industry, it is very difficult for an instrument to be robust and precise enough to do the tasks mentioned above. Typically an array of barrier sensors or a couple of mobile laser based sensors issued to measure the width of the coils. In these applications, however, the range to be covered is too vast to use sufficiently precise barrier sensors or the sensors would be mounted in mobile parts over a precision linear guide to track each edge and therefore their installation and maintenance costs would increase considerably. Other width measurement systems are based on measuring the distance to the edge from a fixed position on both sides of a strip, but in addition to their cost, they usually require thicker material and very stable positioning of the sheet. High resolution CCD cameras are becoming more competitive and image processing and embedded computer performance are becoming more and more efficient.

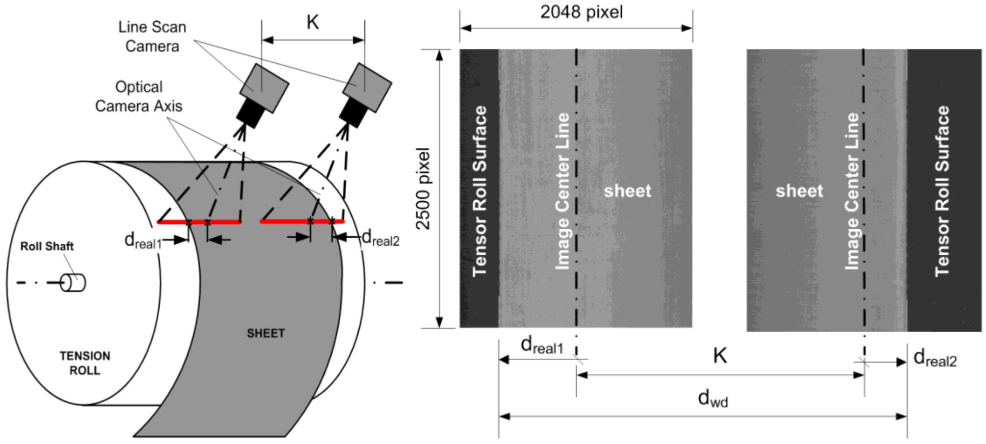
So, a decision was made to implement an AOI system based on CCD cameras, which could have both two functions: continuous measurement of the width of the strip and detection of defects on its edges.

Several real time AOI systems using area [2, 3] and linear cameras [4-7] in an industrial environment had been presented earlier in this report. But, in this paper a real time AOI system is presented. This AOI continuously measures the width of strips while also offering the ability to detect defects on the edges (e.g., cracks, scratches, strike marks, etc.). The wide range of the width of strips produced, together with the high precision required for the measurement, led our team to use a system with two linear cameras. The geometrical model and adjustments required to tackle the practical problems presented throughout this project are described in the following paragraph.

This AOI system is now working in the factory of one of the greatest stainless steel producers, and part of this system had been presented previously in [8, 9].

II. SYSTEM DESCRIPTION

The proposed AOI system integrates two linear cameras (Fig. 2a), which concurrently inspect both edges of the stainless steel sheet along the whole coil length. The cameras must be placed in a very stable location without displacements and vibrations produced from the sheet, to be able to obtain the required precision. A location where these requirements can be fulfilled is within a tension roll, where the steel sheet is very close to the surface of the roll. The use of linear cameras in this location avoids distortion along the vertical direction which area cameras introduce.



(a) (b)

Fig. 2 Width measurement system: (a) general view and (b) image acquired from line scan cameras

The placement of both cameras must be done delicately so that their optical axes are parallel to each other but perpendicular to the roll shaft, and so that their exploration lines are along the same straight line (Fig. 2a). A procedure and templates have been designed to achieve that aim accurately enough that the introduced error will be negligible.

The system's objective is to measure the width of the coil along its entire length (typically, coils are several hundred meters long). The distance between the edge position and the center line of the picture, d_{real}, is calculated for each pair of images acquired by the cameras (Fig. 2b). As the distance K between the optical centers of the cameras becomes known in the system calibration, the coil width, d_{wd}, can be calculated as in Eq. (1).

$$d_{wd} = d_{real1} + d_{real2} + K \tag{1}$$

The additional goal of the study is to accomplish real time detection of edge anomalies in enough time to warn quality control staff, and to store all information about those defects saved in a data base for subsequent analysis.

The continuous processing of coils in annealing and pickling lines requires welding the end of a coil to the beginning of the following one. As all measurements of width and edge anomalies detected must be positioned along the correct coil, a weld line detection algorithm is essential to make a suitable assignment. The weld detection event is used to reset the coil length.

The detection demands real-time image processing due to the line speed and to effectively discriminate other defects that may appear on the sheet surface. To better fulfill these requirements, the designed processing algorithms have been adapted to be implemented cooperatively between the main CPU and the GPU (Graphic Processor Unit) of the graphics card. In this way, it was possible to separate image processing tasks from other tasks such as data base accesses, data storing, interaction with other processes, data communication, etc. Furthermore, a high reduction during the image processing time was achieved, not only through the basic division of the tasks, but via the additional parallel processing which the GPU allowed. After taking advantage of these strategies, there is more processing time available, and now it is possible to undertake a more detailed inspection in the future.

III. GEOMETRICAL MODEL

The core of an instrument is to transform the row data supplied by the sensors to measurements. In this case the sensors are a pair of CCD linear arrays as shown in Fig. 2a. The row information supplied is a pair of images, which are shown in Fig. 2b. Any measuring application based on processing of optical images must have a geometrical model to extract the measurements from, and a fixed calibration method to set all of the pertinent constants [10-12]. The geometrical model is presented in this paragraph and the calibration procedure is exhibited in VI.

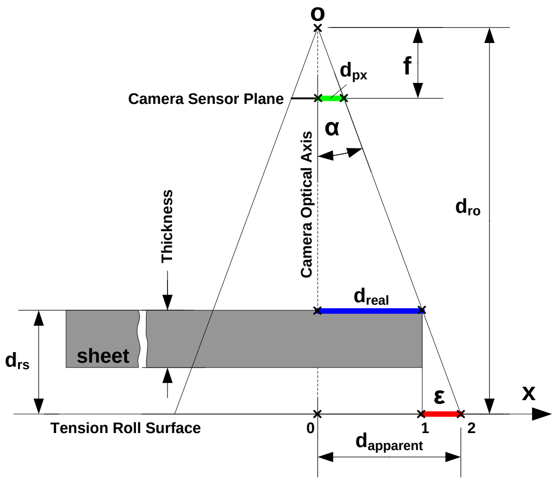


Fig. 3 Distance and perspective corrections

Although images acquired by a linear camera do not have distortion along the vertical axis, they suffer distortion along the horizontal one. In Fig. 3, a simple geometrical model is presented. The cameras have fixed placement in regards to their distance from the roll d_{ro}, and the focal distance of the camera optics f are constants. So the distance of any point in the roll surface to the optical center of the image $d_{apparent}$ can be measured (2) depending on the pixel distance d_{px} in the CCD array sensor. The pinhole optical model has been adopted [13].

$$d_{apparent} = d_{px} \cdot \frac{d_{ro}}{f} \tag{2}$$

But the steel strips to be measured have different diameters, and the distance to the camera varies. This introduces a difference ε (Fig. 3) regarding the $d_{apparent}$ measured in the roll, which needs to be corrected. Eq. (3) gives the real distance d_{real} measured in the strip's surface, which was calculated according to the apparent $d_{apparent}$ distance

$$d_{real} = d_{apparent} \cdot \left(1 - \frac{d_{rs}}{d_{ro}}\right) \tag{3}$$

being d_{rs} the distance between the sheet surface and the roll due to the strip thickness and any accidental gap between them. However, it is not constant, and must be independently measured, which is accomplished by a distance sensor. A precision ultrasonic sensor has been selected, which gives accurate measurements of distance, and is more resilient against changes in surface finish of the steel.

The images taken by the cameras have the following characteristics:

- Scanned line: 2048 squared pixels equivalent to a real width of 400 mm (5 pixels/mm ratio).
- Camera sensor size: 28.67 mm.
- Distance between each camera: 750 mm (approximately).
- Lens focal distance: 50 mm.
- Lens F-number: 16 (depth of field obtained is approximately 94.4 mm).

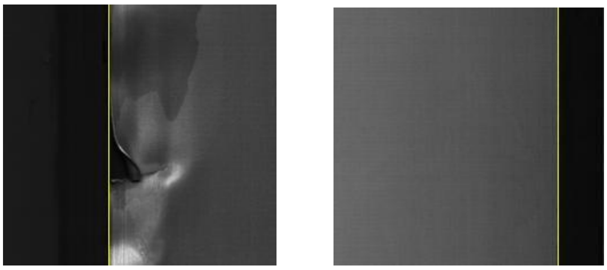
IV. EDGE LINE DETECTION AND COIL WIDTH MEASUREMENT

The width of the sheet is usually quite constant along the whole length of the coil. This allows one or two width measurements per meter length to be sufficient enough for a real-time representation and data storage. With this in mind, a 500 mm (or 2560 pixels) vertical size for each image was selected to obtain a global width measurement after applying a specific image processing algorithm (Fig. 4).

The coil edge detection approach is based on luminance variance between the tensor roll and steel surface. The designed algorithm must fulfill the following requirements:

- It must be robust against different anomalies and defects that could appear in the coil surface.
- It must be flexible enough to support the different luminance and superficial finish of all steel grades processed in the production line.

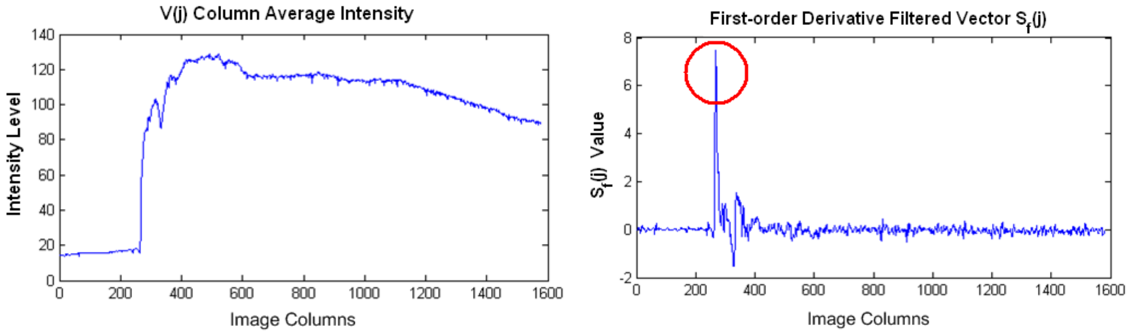
- Its computational cost must be efficient enough to allow real-time processing of every few images taken while the line is working.



(a) (b)

Fig. 4 Acquired coil edge images: (a) left and (b) right ones

Consider a digital image as a two-dimensional function $I(i,j)$ where i and j are respectively the row and col indexes which identify a determined pixel in the image and I is the intensity or level of gray within the pixel. Also, m and n can be established as the total number of rows and columns of the whole image.



(a) (b)

Fig. 5 Algorithm steps, (a): average intensity of each column *V(j)*; (b): low-pass filtered first-order derivative vector $S_f(j)$

Basically, the edge line detection and width measurement algorithm integrates the following steps for each camera image:

1) The image $\mathrm{I(i,j)}$ is transformed in an array $\mathrm{V(j)}$ of n elements averaging the intensity values of each column (Fig. 5a).

$$V(j) = \frac{1}{m} \cdot \sum_{i=1}^{m} I(i,j) \tag{4}$$

2) Obtaining the array $\mathrm{S(j)}$ as the first-order derivative of $\mathrm{V(j)}$array.

$$S(j) \equiv \Delta \mathrm{V(j)} = \mathrm{V(j+1)} - \mathrm{V(j)} \tag{5}$$

3) Smoothing $\mathrm{S(j)}$ array by means of a low-pass filter (moving average algorithm) to obtain $\mathrm{S_f(j)}$ array,

$$S_f(j) = \frac{1}{2\alpha + 1} \sum_{k=-\alpha}^{+\alpha} \mathrm{S(j+k)} \tag{6}$$

where $2\alpha + 1$ is the filter width, an odd value which determines the smoothing window. The best results were obtained with filter width fixed to 5.

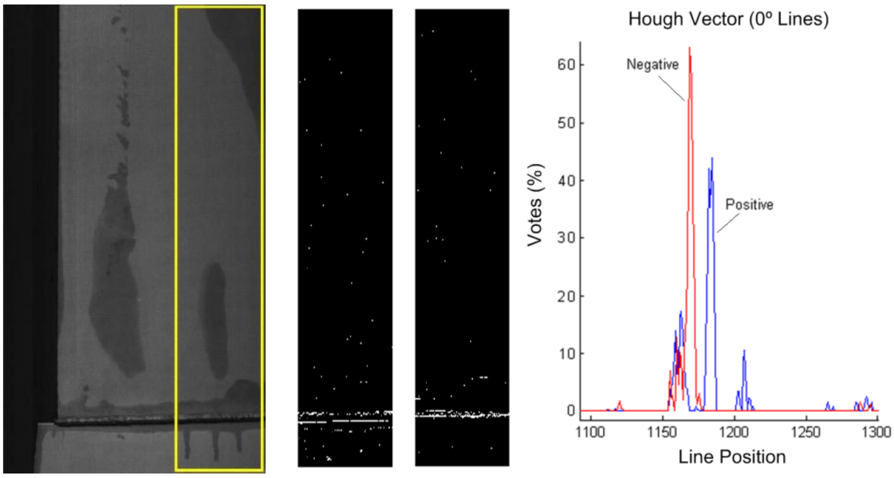
4) Select the pixel $\mathrm{d_{px}}$ which corresponds with the edge of the image (Fig. 5b). This value will be the j index which maximizes the value of $\mathrm{S_f}$ array,

$$\mathrm{j} \mid \max_{\mathrm{j=1\ldots 2048}} \mathrm{S_f(j)} \tag{7}$$

This position gives the d_{px} values in each twin image, which are used to calculate each d_{real1} and d_{real2} distances to be used in Eq. (1) to extract the width of the coil at a given position.

V. WELDING LINE DETECTION

The assigning of the width values measured and edge defects detected in a specified part of the coil will require knowing the exact coil length that has been passed by the AOI system. This obstacle requires accomplishing two tasks: measuring coil length, and resetting such length once the coil has been finished. The first task can be easily accomplished using an incremental encoder fixed in the tension roll, then counting the pulse output from the encoder. The second task requires capturing the reset length event, which can occasionally become a difficult or impossible task.



(a) (b) (c)

Fig. 6 (a) Image with welding line and ROI, (b) Positive (left) and negative (right) gradients, and (c) Horizontal line voting graph

To avoid this problem, the reset length event can be generated by the same AOI system through the detection of the welding, via image processing. In the images, the welding pattern appears as horizontal lines spanning the duration of the coil's width (Fig. 6). Using a simplified and specialized version of the classic approach of line detection, the welding could be detected as described:

a) *Selection of a rectangular region of interest (ROI) within the image* (Fig. 6a). For the detection of the welding, it is unnecessary to analyze the entire image so that an important reduction in computational costs is achieved. The ROI selected must be located on the opposite side of the sheet edge where the image will always show a part of the material. To find the welding line in images, it was sufficient to use a ROI of 1315 x 400 pixels.

b) *Horizontal edge detection*. The ROI image is processed by administering a classical Sobel operator [14] taking into account only the horizontal gradient, and namely using the vertical gradient kernel. Based on their positive and negative values, the result is separated in two 2D arrays, which are binarized afterwards (Fig. 6b). The intensity threshold I_{th} used to binarize the images is then computed using the intensity average I_{avg} and its standard deviation σ as shown in Eq. (8).

$$\mathrm{I_{th}} = \mathrm{I_{avg}} + 3\sigma \tag{8}$$

c) *Search for horizontal straight lines*. The positive and negative arrays obtained in the horizontal edge detection step are used to locate horizontal straight lines through a simplified Hough Transform [15]. The searching can be conducted swiftly and more efficiently, so that it is looking for horizontal lines with 0° degree slopes. Due to this restriction, the 2D parameter array must be reduced to a 1D vector on each image obtaining two vectors with the number of votes of horizontal lines, and with negative and positive gradient (Fig. 6c).

d) *Welding detection event*. Positive and negative lines at a distance less or equal to the typical welding line mark with a minimum number of votes must be found to consider that the welding has been matched (Fig. 6c).

VI. SYSTEM ADJUSTMENTS AND CALIBRATION

A very important step during system installation is its correct placement and calibration. In this study, considerable attention has been devoted to it, as the final precision and accuracy obtained during the study depends greatly on these tasks.

The three possible turns around the axis have been considered (Fig. 7a). The errors introduced by them have been geometrically modeled, and their value represented against the angle. The goal is to align the camera precisely, to avoid any errors being neglected. So, the maximum allowable deviation has been calculated to avoid this error being effected (i.e. at least one order of magnitude less than the required precision). With this information, templates can be designed for control camera placement at installation time. As shown in Fig. 7b, in case of a rotation in Z-axis of an angle γ, the error δ is

$$\delta = |x - x'| \tag{9}$$

or depending on γ it can be expressed as

$$\delta = |x' \cdot \cos(\gamma) - 1| \tag{10}$$

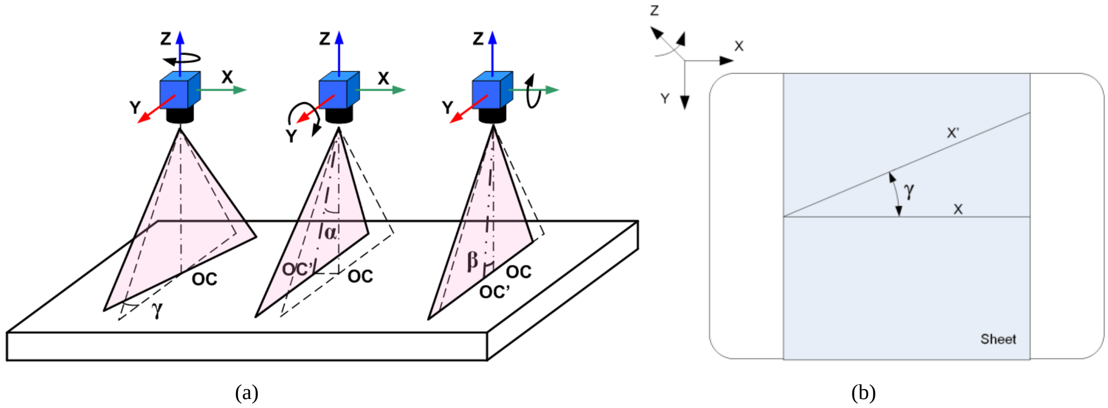


Fig. 7 Camera rotations around X, Y, and Z axis

According to Eq. (10), if we select 0.05 mm as the maximum allowable value, a value of $\gamma = 0.9°$ is obtained. This information is used to design a template for adjusting the z-axis (Fig. 8a and 8b), as shown in Eq. (11).

$$\begin{cases} \gamma_{max} = 0.9° \rightarrow \tan(\gamma_{max}) = 0.015 \\ \tan(\gamma_{max}) = \dfrac{d_v}{d_h} = \dfrac{d_v}{400\ \text{mm}} = 0.015 \rightarrow d_v = 6.28\text{mm} \end{cases} \tag{11}$$

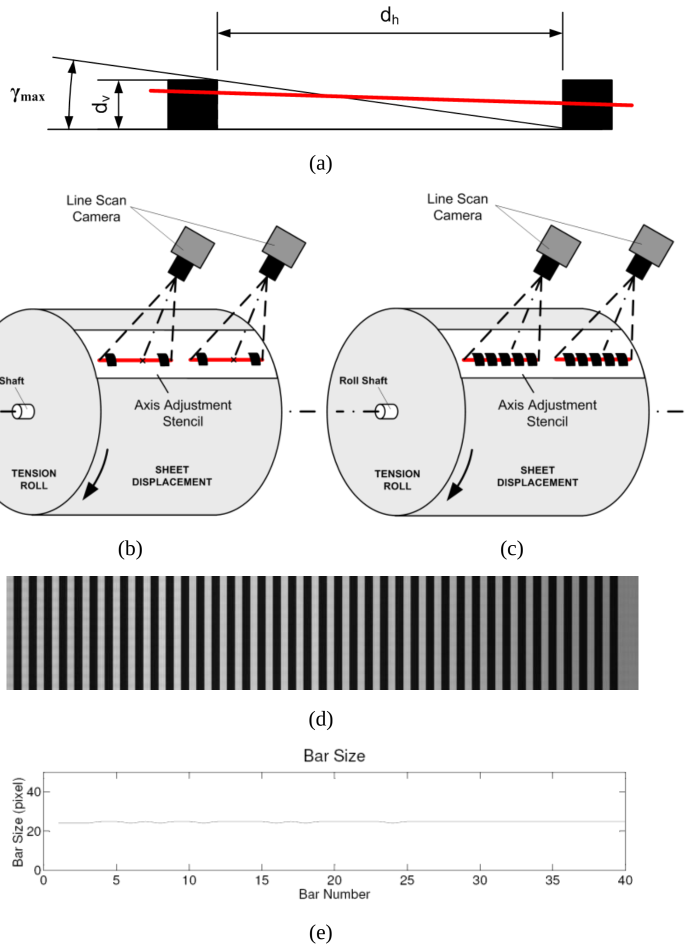


Fig. 8 Adjusting and calibration templates: (a) template design for Z-axis maximum error limitation, (b) camera Z-axis rotation correcting stencil, (c) X-axis rotation correcting stencil, (d) camera image acquired for X-axis calibration procedure, and (e) graphical representation of black bar sizes

The template in Fig. 8b is fixed in tension roll to adjust the z-axis rotation of each camera. A similar template to that in Fig.8c is used to adjust the x-axis rotation, while processing the acquired image (Fig. 8d), and trying to obtain the bar width size graph (Fig. 8e) with constant value.

In spite of all these considerations, residual errors are possible. One method of correcting them is via the value of K so far considered constant in Eq. (1). This can be done after some days of working, when routinely monitored measurements are taken of hundreds of coils' different thicknesses and widths. So K is computed as in Eq. (12).

$$\mathrm{K} = \mathrm{k}_1 + \mathrm{k}_2(\mathrm{width}, \mathrm{thickness}) \tag{12}$$

All the width measurements manually taken by quality control operators are compared with those acquired by the AOI system as a routine maintenance task. Table 1 shows the results which are sorted by the range of width and diameters, where value is the number of samples taken for each set of data. Statistics values, mean difference, and standard deviation (STD) of the difference between manual and automatic measurements are shown as well.

TABLE 1 COMPARISON OF AUTOMATIC MEASUREMENTS (WITH AND WITHOUT K CORRECTION APPLIED)

Width*	Thick.*	n	w/o K Correction*		w/ K Correction	
			Mean	SDT	Mean	SDT
900-1000	[2,3]	448	-4.49	1.78	0.64	1.78
	[3,4]	445	-4.34	1.92	0.62	1.93
1000-1100	[2,3]	621	-2.89	1.90	0.12	1.91
	[3,4]	1045	-2.77	1.98	0.06	1.99
	[4,5]	896	-2.65	2.02	0.02	2.03
	[5,6]	470	-2.48	1.90	-0.02	1.91
	[6,7]	225	-2.39	1.68	-0.11	1.70
1200-1300	[2,3]	486	-0.88	2.03	0.51	2.03
	[3,4]	1637	-1.03	1.91	0.08	1.93
	[4,5]	1810	-0.79	1.91	0.05	1.91
	[5,6]	924	-0.41	1.91	0.15	1.91
	[6,7]	338	-0.43	1.71	-0.19	1.72
	[7,8]	114	0.31	1.72	0.29	1.72
1500-1600	[3,4]	680	2.05	1.73	-0.11	1.71
	[4,5]	1715	2.48	1.66	-0.03	1.67
	[5,6]	1269	2.76	1.65	-0.11	1.64
	[6,7]	706	3.20	1.59	-0.05	1.58
	[7,8]	241	3.42	1.67	-0.18	1.62
	[8,9]	230	4.01	1.62	0.00	1.56
	[9,10]	95	4.26	1.72	-0.02	1.56

*Units are in millimeters

After a statistical study of the differences, a second order correction of K is carried out, which is accomplished by building the piecewise linear approximation that best matches the distribution of mean differences. A statistical package can be used to build the model. The one used for this system is shown in Fig. 9. In this case a linear correction is applied along the thickness values and piecewise along different widths. This correction significantly enhances the precision as is shown in the last two columns (mean and STD with K correction headers) of Table 1.

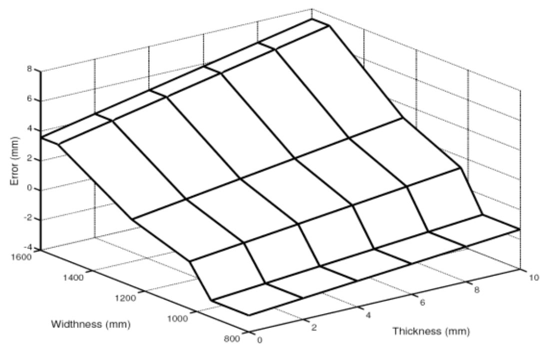


Fig. 9 Error correction model

VII. EDGE DEFECT DETECTION

An important added value of this system is its ability to detect edge anomalies besides the width measurement, providing alarm and event notifications to quality control staff.

Usually, the defects which appear in the edge of coils affect the continuity and straightness of the edge. These defects are produced by a hit-and-strike over the coil's side (Fig. 10a) or breakdowns in previous production processes.

Thereby, the defect detection algorithm in the image is constructed to take accommodate both common continuity and straightness properties. It is executed after the coil position is determined in the image by the edge detection algorithm, as shown in IV section.

This approach consists of a highly detailed analysis regarding a region around the edge detected. The algorithm steps are described as follow:

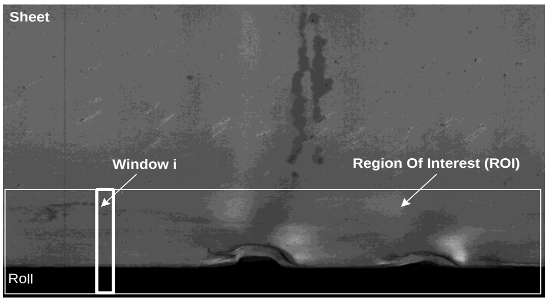
a) ROI selection around the coil's edge (bigger white rectangle in Fig. 10a).

b) Binarization of ROI to separate the pixels belonging to roll from those of the coil.

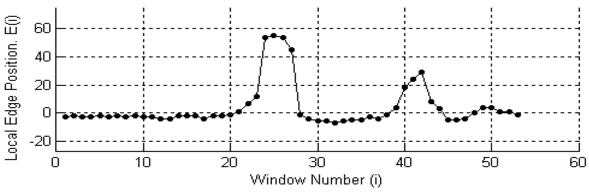
c) Division of the ROI in *N-windows* along the edge. The number of windows N must be selected to reach a compromise between the accuracy desired and low computational cost, bearing in mind the real time restrictions of vision task. For this vision task, a window of 50 pixel x 600 pixel (10 mm x 120 mm) has been selected.

d) Processing of each binarized window i (analogously to edge detection algorithm) to obtain edge relative position vector E where each element $E(i)$ contains the relative edge position in the *i-th* window, which is the global edge line determined in section IV (Fig. 10a and 10b).

e) Analyzing the vector E along *N-windows* to detect sharp variations in relative edge positions (see windows $i = 25$ and $i = 42$ values in Fig. 10b).



(a)



(b)

Fig. 10 (a) Elements in image (ROI and Window i; image left-rotated 90 º); (b) Edge relative position vector E

VIII. GPU IMAGE PROCESSING

Nowadays, the image processing algorithms developed accommodate a specific vision or task tends to be executed in a Graphic Processing Unit (GPU) instead of in a standard CPU. Such graphic processors are preferable when compared with the others because of their high computational power and parallel processing capability, which is executed by using SIMD (Single Instruction stream Multiple Data) architecture [16-21].

NVIDIA Company was the inventor of the GPU unit, and were also the developers of a general programming framework called CUDA (Compute Unified Device Architecture) [22, 23] which is used in general purpose C language applications [24]. The underlying concept of CUDArevolves around the parallel processing capability based on the grouping of *threads* (or elemental processors) in 1, 2 or 3 dimensional *grid* which runs the same operator over a data matrix. This simple idea could minimize the processing time of a specific task, freeing that time to do other processing tasks.

Analogously, it could be beneficial if the described algorithms in precedent sections were adapted to the GPU. If this were done it would be possible to combine, submit, and run all whole algorithms (welding detection, coil edge detection, width measurement and defect detection) through the GPU, fulfilling them with real time restrictions.

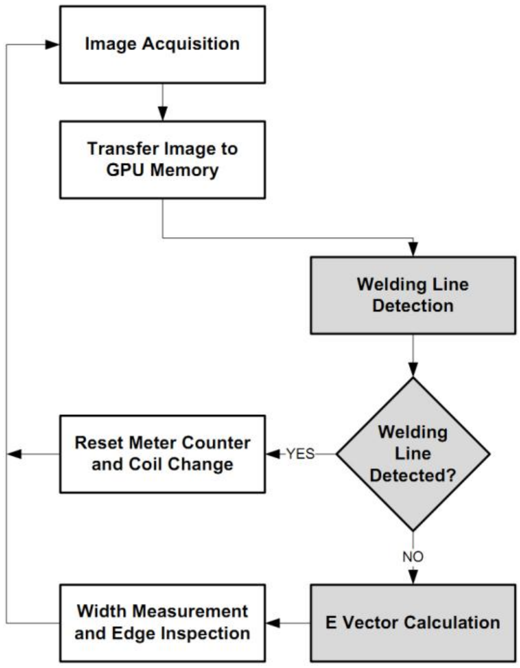


Fig. 11 GPU-CPU processing task integration sequential scheme (CPU tasks in white and GPU tasks in light grey)

The following steps show the sequential scheme for CPU-GPU processing task integration (see Fig. 11):

1. *Image acquisition.* Image acquired with a resolution of H pixels height by W pixels wide.

2. *GPU memory allocating and image matrix transference*.

3. *Welding detection.* Detection of the welding line as described in section V. The Sobel filter adapted for GPU has been developed, which is exhibited in [25, 26].

4. *Image intensity averaging.* The image is divided into *N-windows*, or regions, with the same height and width, where each individual *i-window* will be composed by pixels. Sequentially, for each *i-window*, a *thread* with an averaging operator devoted to each column, and constituting a *grid of threads* processes all columns in parallel. When the averaging of the window has been completed, the result is stored. Lastly, a matrix is obtained where each row holds the intensity average values of the cols for each *i-window* (Fig. 12, step 4).

5. *Matrix $I(N,W)$ pre-processing.* Each row of the matrix $I(N,W)$ is filtered and differentiated to obtain a new matrix $G(N,W)$ with the gradient values. Sequentially, a *thread* with a filtering and differentiating operator is devoted to each row of each matrix column, constituting $N \times 1$ *grid of threads* which process all rows in parallel (Fig. 12, step 5).

6. *Local edge position vector E computing.* Each row of $G(N,W)$ is analyzed to secure a value with the local edge position. For each matrix column, a thread with a maximum operator is devoted to each row constituting $N \times 1$ *grid of threads* which processes all rows in parallel to obtain the maximum value of the gradient in the row. Finally, a vector E with N elements is obtained where each element $E(i)$ contains the local edge's position (Fig. 12, step 6).

After that, the vector E is transferred to the CPU to be analyzed and to compute its mean value. The value is considered the “coil edge location” in the image (d_{real1} or d_{real2} depending on the image side). By applying Eq. (1) in section II, it will obtain the coil width. Additionally, the E vector is analyzed looking for sharp variations in relative edge positions to determine if it contains any edge defects.

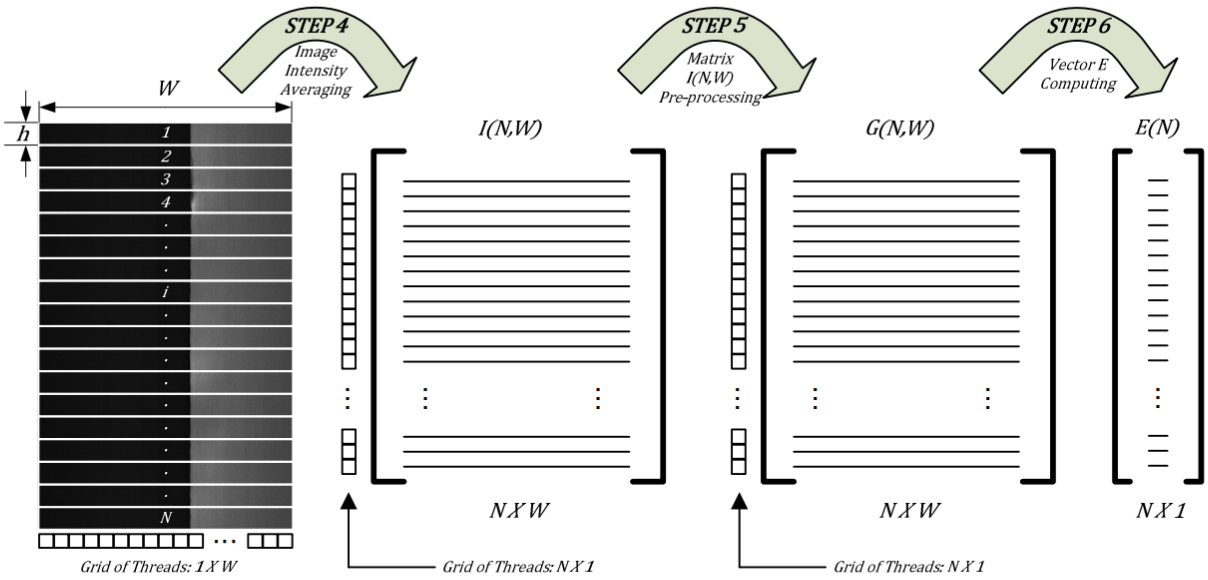


Fig. 12 Sequential GPU processing scheme and Grids of threads

IX. PERFORMANCE IMPROVEMENTS

The algorithms presented before have been applied over 2048×2048 pixel size images to compare the processing time according to the following implementation: in a CPU, a CPU/GPU with independent tasks and a CPU/GPU with overlapped tasks (Fig. 13). The image transferring time from CPU to GPU was also considered in calculations. The hardware used as the test-bench was:

- CPU Processor: Intel Pentium Dual Core E5200 2.52GHz, 1GB RAM.
- GPU Processor: NVIDIA GeForce GTX 285, 1 GB Global Memory. 240 CUDA cores. Compute capability 1.3.

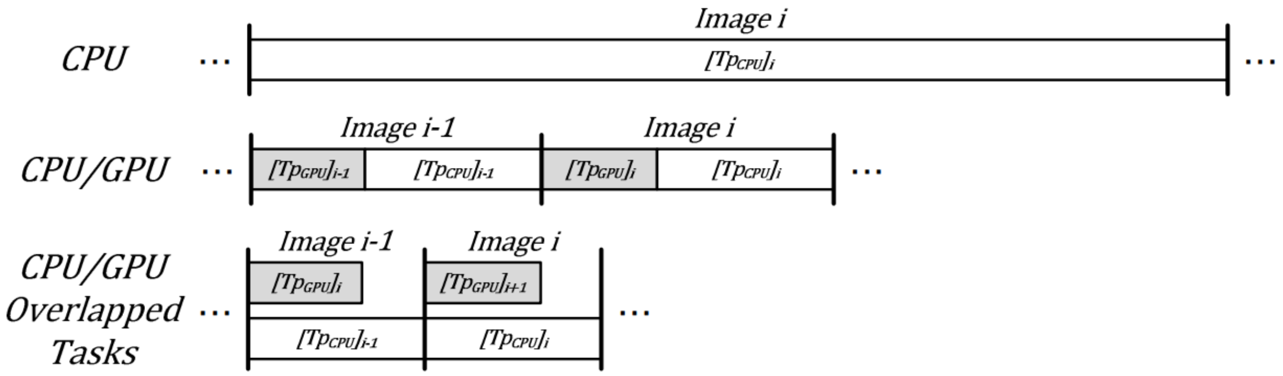


Fig. 13 Time improvement using overlapped tasks

Table 2 shows the processing times and the acceleration achieved by implementing the algorithms in a CPU/GPU with independent tasks.

TABLE 2 COMPARISON OF TASK PROCESSING TIMES

Task	Time (ms)		
	CPU	**GPU**	**Speed-Up**
Load Image	-	3.75	-
Weld Line Detection	78	7.81	10x
Edge Detection	63	4.40	14x
TOTAL	**141**	**15.96**	**9x**

Moreover the CPU processing time used during the image processing tasks, an additional CPU processing time must be spent for miscellaneous tasks as vector E analyzing, results storing, image compression, data base accessing, etc. When be necessary or required, additional processing time can be obtained by implementing another algorithm strategy, such as a CPU/GPU with overlapped tasks. Such a strategy runs the presented tasks in parallel rather than sequentially. The GPU processing time required for the refinement of the i image $[Tp_{GPU}]_i$ and the CPU processing time required for the i image miscellaneous tasks $[Tp_{CPU}]_i$ can be overlapped, which is shown in Fig. 13. It can be assumed that there is always a new image available at the beginning of each period, and that the i image is processed in a GPU while the miscellaneous tasks corresponding with $i-1$ image is performed in the CPU.

X. SYSTEM IMPLEMENTATION

A general connection diagram of sensors and processing devices is shown in Fig. 14. The image acquisition of the two line scan cameras is triggered a pulse train, which is generated an incremental encoder fixed in the roll shaft. The coil speed in the production line usually ranges between 20 and 60 m/min depending on the operation conditions. The acquired images by both cameras are sent to an industrial embedded computer by means of a gigabit Ethernet link. The image processing and other miscellaneous tasks are performed in the industrial computer. This computer also receives the ultrasonic sensor signal via serial RS-422 to correct the apparent position of the edge according to the geometrical model presented in section III.

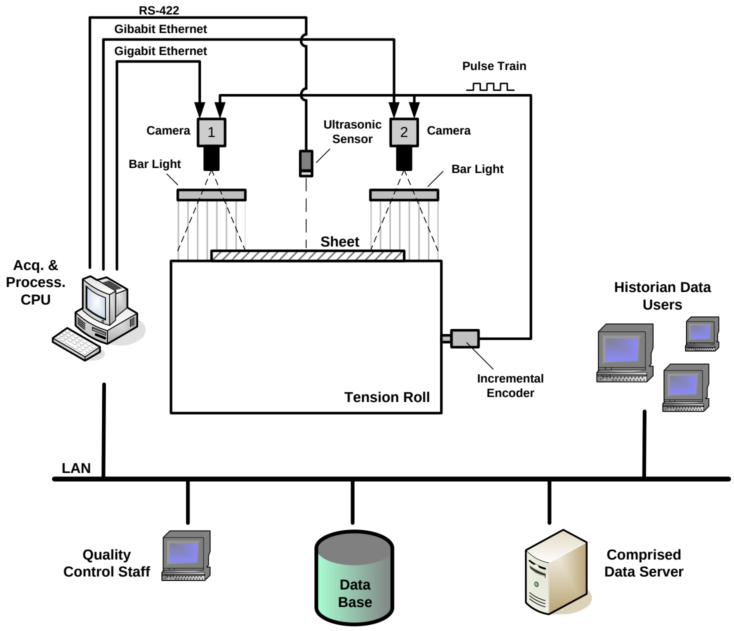


Fig. 14 System architecture

The industrial computer is linked to the factory LAN to present the acquired images and processing results to the quality control operators in order to show them a real time synoptic screen, and to build and store the information in a data base and server for historical review and analysis purposes. Fig. 15 shows a view of the installed system in the production line.

Fig. 15 View of the system installed in the production line

XI. RESULTS AND CONCLUSIONS

The AOI system presented in this paper exhibits its capacity to continuously inspect the width and edges of the stainless steel strip in real time. A geometrical model and corrections constructed from strip thickness variations have been presented. The AOI system has been installed and integrated in an annealing and pickling production line, obtaining a great quantity of processed coil data. Continuous width measurements of each coil and images of detected defects are stored for quality control auditing purposes.

Fig. 16 shows a graph of a coil with 589 meters in length where the nominal and tolerance limits of width are also represented. It is noticeable that the beginning and the end of this coil is outside of the tolerance limits and cannot be delivered.

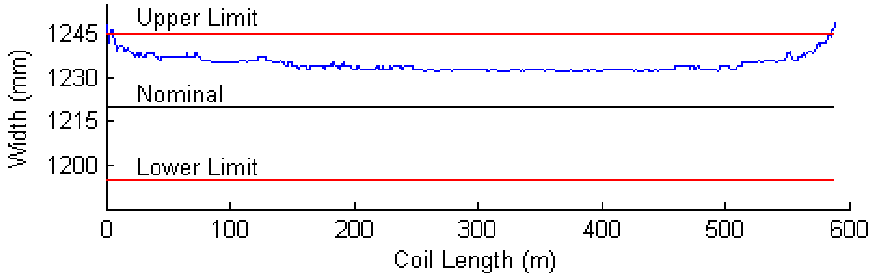


Fig. 16 Measured width of a whole length stainless steel coil

The linear CCD sensor data is converted into images and processed with conventional image processing techniques. The calibration and camera placement tools are essential and an adjusting procedure has been designed.

The second order correction of K value based on a statistical model constructed with the initial working data of the instrument, dramatically improves its precision, and software has been developed to accomplish this task.

The main benefits of this system include the ability to inspect coil edges, and the fact that it can be integrated with other inspection tasks. A high performance in vision task can be achieved if algorithms are adapted and executed in a GPU unit using CUDA architecture. This methodology allows more processing time to become available for integrating other image processing algorithms, which allow passing from a general and coarse defect detection to a more specific and fine one. For now, the population of images with defects collected by the system is not sufficient enough to afford this step.

ACKNOWLEDGMENTS

This research project would not have been possible without the support, help and suggestions of many people. The authors wish to express their gratitude to Information Systems and Quality Control departments of Acerinox Europa S.A.U.

REFERENCES

[1] B. Jha, P. Jha, and C. D. Singh, “Process Technology for the Continuous hot band annealing of 17% Cr ferritic stainless steel,” *in Journal of Material Engineering and performance*, pp. 11, 2002.

[2] C. Fernández, C. Platero, P. Campoy, and R. Aracil, “Vision system for on-line surface inspection in aluminium casting process,” *in International Conference on Industrial Electronics, Control and Instrumentation*, Maui, Hl, vol. 3, pp. 1854-1859, Nov. 15-19, 1993.

[3] H. Zhang, G. Wu, X. Sun, J. Xu, and K. Xu, “A New Fast Border Search Algorithm and Its Expansion,” *in IEEE International Conference on Automation and Logistics*, Jinan, China, pp. 1886-1890, Aug. 18-21, 2007.

[4] Y. Fernández, R. C. Gonzalez, P. F. Díaz, and J. M. Enguita, “On-line wavy strip and off-center analysis of high speed production lines by a linear camera,” *in IEEE Conference on Emerging Technologies and Factory Automation*, Lisbon, Portugal, vol. 2, pp. 552-559, Sept. 16-19, 2003.

[5] Z. Hocenski, T. Keser, and A. Baumgartner, “A Simple and Efficient Method for Ceramic Tile Surface Defects Detection,” *in IEEE International Symposium on Industrial Electronics*, Vigo, Spain, pp. 1606-1611, Jun. 4-7, 2007.

[6] Y. Li, T. Y. Young, and C. Huang, “Noncontact Measurement Using Line-Scan Cameras: Analysis of Positioning Error,” *in IEEE Transactions on Industrial Electronics*, vol. 36, no. 4, pp. 545-551, Nov. 1989.

[7] C. G. Spínola, J. M. Cañero-Nieto, G. Moreno-Aranda, M. J. Martín-Vázquez, J. M. Bonelo, and J. Vizoso, “Sistema de Visión para la Medida del Ancho de Bobinas de Acero,” *in XVII Seminario Anual de Automática, Electrónica Industrial e Instrumentación (SAAEI’10)*, Bilbao, España: 7-9 de Julio 2010.

[8] C. G. Spínola, J. M. Cañero-Nieto, G. Moreno-Aranda, J. M. Bonelo, and M. J. Martín-Vázquez, “Continuous Real-Time Optical Measuring of Strip Width and Edge Inspection in Stainless Steel Production Lines,” *in IEEE Int. Instrumentation and Measurement Technology Conf.*, Binjiang, Hangzhou, China, May 2011.

[9] C. G. Spínola, J. M. Cañero-Nieto, G. Moreno-Aranda, J. M. Bonelo, and M. J. Martín-Vázquez, “Real-Time Image Processing for Edge Inspection and Defects Detection in Stainless Steel Production Lines,” *in IEEE Int. Conf. on Imaging Systems and Techniques*. Batu Ferringhi, Penang, Malaysia, May 2011.

[10] Y. Guo, Z. Fandg, and X. Xu, “An Analytic Solution of a Linear Camera Self-Calibration,” *in IEEE Proceedings of the 6th World Congress on Intelligent Control and Automation*, Dalian, China, vol. 2, pp. 9930-9934, Jun. 21-23, 2006.

[11] W. I. Grosky and L. A. Tamburino, “A unified Approach to the Linear Camera Calibration Problem,” i*n IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 663-671, Jul. 1990.

[12] R. Horaud, R. Mohr, and B. Lorecki, “Linear-Camera Calibration,” *in IEEE Proceedings of the International Conference on Robotics*

and Automation, Nice, France, vol. 2, pp. 1539-1544, May 12-14, 1992.

[13] B. Jahne, *Digital Image Processing*, 6th ed., Berlin, Germany: Springer-Verlag., pp. 192-195, 2005.

[14] I. Sobel, “Camera Models and Machine Perception,” PhD Thesis, Standford University, Standford, CA, U.S.A., 1970.

[15] R. O. Duda and P. E. Hart, “Use of the Hough Transformation to Detect Lines and Curves in Pictures,” *Commun. of the ACM*, vol. 15, no. 1, pp. 11-15, Jan. 1972.

[16] E. Lindholm, J. Nickolls, S. Oberman, and J. Montrym, “NVIDIA Tesla: A Unified Graphic and Computing Architecture,” *in IEEE Micro*, vol. 28, no. 2, pp. 39-55, Mar. 2008.

[17] M. Garland et ál., “Parallel Computing Experiences with CUDA,” *in IEEE Micro*, vol. 28, no. 4, pp. 13-27, Jul. 2008.

[18] J. Nickolls and W. J. Dally, “The GPU Computing Era,” *in IEEE Micro*, vol. 30, no. 2, pp. 56-69, Mar. 2010.

[19] I. K. Park, N. Singhal, M. H. Lee, S. Cho, and C. Kim, “Design and Performance Evaluation of Image Processing Algorithms on GPUs,” *in IEEE Trans. On Parallel and Distributed Syst.*, vol. 22, no. 1, pp. 91-104, Jun. 2010.

[20] Z. Yang, Y. Zhu, and Y. Pu, “Parallel Image Processing Based on CUDA,” *in Int. Conf. on Comput. Sci. and Software Eng.*, Wuhan, Hubei, pp. 198-201, Dec. 2008.

[21] J. Kim, M. Hwangbo, and T. Kanade, “Parallel Algorithms to a Parallel Hardware: Designing Vision Algorithms for a GPU,” *in IEEE 12th Int. Conf. on Comput. Vision Workshops (ICCV Workshops)*, Kyoto, Japan, pp. 862-869, Sept. 2009.

[22] NVIDIA CUDA Programming Guide 3.0, NVIDIA Corporation, Feb. 2010.

[23] S. I. Park, S. P. Ponce, H. Jing, C. Yong, and F. Quek, “Low-cost, High-speed Computer Vision using NVIDIAS’s CUDA Architecture,” *in IEEE 37th Appl. Imaginery Pattern Recognition Workshop (AIPR’08)*. Washington DC, U.S.A., pp. 1-7, Oct. 15-17, 2008.

[24] NVIDIA CUDA C Programming Best Practices Guide. CUDA Toolkit 2.3, NVIDIA Corporation, Jul. 2009.

[25] N. Zhang, Y. Chen, and J. Wang, “Image Parallel Processing Based on GPU,” *in 2nd Int. Conf. on Advanced Comput. Control (ICACC’10)*. Shenyang, China, pp. 367-370, Mar. 27-29, 2010.

[26] J. Fung and S. Mann, “OpenVIDIA: Parallel GPU Computer Vision,” *in Proc. of the 13th Annu, ACM Int. Conf. on Multimedia*, Singapore, pp. 849-852, 2005.