Storage Processes Optimization In Industrial Environments By Image Analysis On Ethernet Networks

Fabio Navarrete Martínez, John Herrera Guateque and Holman Montiel Ariza
fanavarretem@correo.udistrital.edu.co, jhherrerag@correo.udistrital.edu.co, hmontiela@udistrital.edu.co
Technological Faculty
District University Francisco José de Caldas
Bogotá D.C., Colombia

Abstract: The main objective of this work is to develop a telematic environment, which allows the transfer of information related to image recognition, in order to optimize the storage process for the 208 station of the HAS 200 (Highly Automated System), based on embedded technologies and development of oriented to client-server model applications, speeding up processes and generating an increase of efficiency in the regular operation of the station.

Keywords: Automation, Industrial control, Telematic environment, Client-server model, Image recognition.

1. INTRODUCTION

The highly automated system or HAS 200, is a simulation environment for industrial production processes, designed to provide a safe and low-cost learning environment, which allows students to develop their skills by providing analysis and evaluation knowledge, regarding execution times, quality management processes and activity planning, among others [1]. Its main purpose is providing technological basis attempting to improve the storage process at the 208 station, which is in charge of preserving the final product, due a drawback in its manual operation related to the information of stock's spaces.

The application can reduce consultation times and mechanical errors during the product's storage. The development methodology for the optimization proposal starts by the operational analysis of the station to be intervened; it is developed a description of the module operation where the technique is applied. The second step is the extraction of the main features of a sample images set (orientation, color and position), and thirdly, a classification and information storage is performed in order to optimize the module operation through a telematic infrastructure.

2. METHODOLOGY

The development begins with a global specification of the highly automated system HAS200, focusing on the 208 station and its functioning, providing an approach to the proposed topic.

System definition

The highly automated system HAS-200 (see Fig. 1) is an environment which simulates a modern factory, to provide a learning environment, developing the practical knowledge and capacities of the staff at the production area, where there can be manufactured up to 19 different products according to the established request. The raw material consist of a container with 4 types of labels (red, blue, yellow and multicolor). Each of the labels incorporates a barcode which allows to identify the product during the whole process.

Figure 1: Highly Automated System HAS 200 [1]

Inside those containers "beads" will be poured in quantities of 15, 30 or 45 gr, being the latter measure the one which limits the maximum capacity of the container. Therefore, the finished product may be one produced with a unique type of "beads" or one produced in conjunction with two or three different types of "beads" which will be labeled as multicolor. Within the process, there are measured the weight and the height of the material. Both variables are analyzed by the Statistical Process Control (SPC) for decision making, history generation, among others. The main idea of this work is to optimize the operation of the vertical storage station (see Fig. 2), then its process is described.

Station description

Name: Vertical storage station HAS-208.
Function: The function of the station 208 is to store the products that are generated, in 81 positions destined for this purpose by the use of a robotic arm (see Fig. 3), controlled by a main command that counts on automatic or manual management according to its program.

Figure 2: Front View Vertical Warehouse [2]

Manual Operating Mode: This mode of operation is the result of placing the selector in “I” position, allowing the user to operate manually the station, meaning, the station works independently of the rest of the system stations, assigning the operator a direct control over the previously programmed actions in the electronic control unit of the station (PLC).

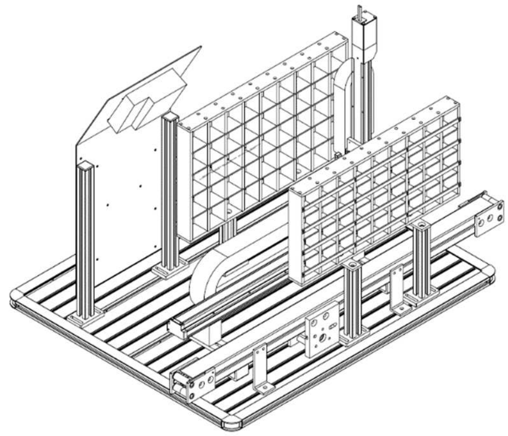

Figure 3: Rear View Vertical Warehouse [3]

Automatic Operating Mode: On the other hand, if the selector is in "II" position, the selected operating mode is automatic, meaning, the user is allowed to work as an integrated factory through a manufacturing execution system or also called M.E.S.

Basically, the 208 station is responsible for storing the products containers, for subsequent dispatch or recirculation, depending on whether it is an elaborated product or a semi-processed product, having the capacity to hold up to 81 containers vertically arranged in unit cells.

The division of those positions is performed in two panels of different dimensions, one of 9x4 cm and the other of 9x5cm; This station is based on a system with two servo-controlled electric axes that allow the storage in any of the 81 possible cells, being this controlled manually by a terminal operator or HMI to ease the interface with the user. Stations communicates through an Ethernet network (see Fig. 4) through a hub, which receives the connections traveling by the UTP cable to connect and transmit between the stations and the management software.

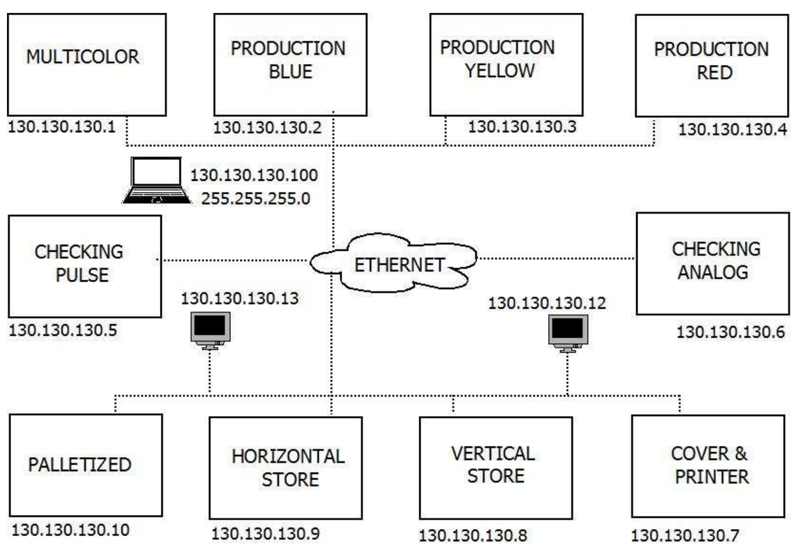

Figure 4: Stations communication diagram

Regarding the software, in order to develop each of the system's capabilities, it is necessary to have a series of applications (see Table 1), (OPC Server, 3D-Supra, EdMES, PLC Programming Software) installed in the PC.

Table 1: Control software

EdMES	
CGIP	Manage Interface Software's DB between HAS-200 and EdMES
PLANNING SYSTEM & WORK ORDER	Planning, to sequence, creating and transmitting to OP.
SPC	Statistics control of the process and the product.
MAINTENANCE	Mantle management: Corrective, Preventive, Predictive.
TRANSPORT	To simulate the product transportation into the production line.
WIP	To consult the database and/or a product traceability.
OEE	To analyze the global system efficiency or performance.
ALARM SUMMARY	Following alarms and generated notices.
AGENTS	To provoke malfunction situations for the system.

3. PROPOSED WORK

It is proposed to develop a telematics environment for the control of the 208 station, from which it is possible to perform the management of the operation, besides having the capacity to indicate the state of its storage, by developing an application that processes the result of the image analysis and communicates through client-server connections [4] , so a communication frame containing information resulting from image processing will be developed to transmit in an Ethernet network, supported on Python software and the Raspberry pi card hardware as platform[5].

Application function

The application is based on a client-server architecture capable of transmitting information obtained from image processing, using sockets to establish the communication using TCP protocol [6] [7][17], and threads to guarantee the system concurrence through an Ethernet network, integrating a database functioning for the storage and generated results management in the entire environment (see Fig. 5); This data transmission will be carried out by a communication network made to size, capable of provide the required information for each of its fields optimizing the storage time of the 208 station this way.

For the data transmission process, it is captured an image of the store indicating the product position, type (labeled by color), and whether there is or not an available storage field; that information is processed by the image recognition module [8] and packaged in a frame which is sent to the storage module that saves the received data in a database, for its later presentation to the station operator [9].

The block diagram of the proposed solution is described with the necessary components for its proper operation.

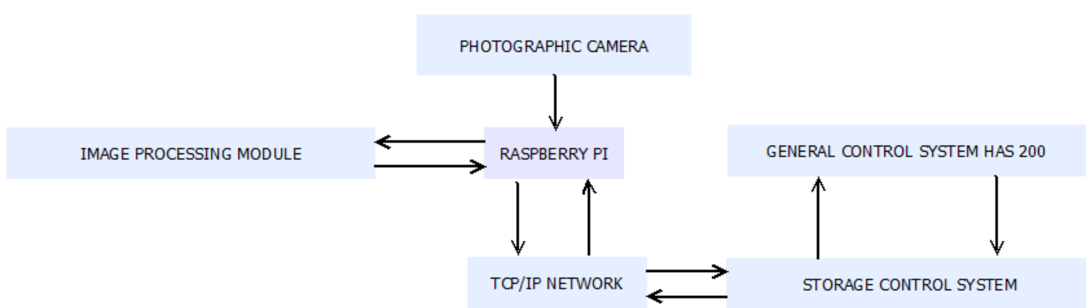

Figure 5: Block Diagram

The images will be analyzed individually, although the application allows to manage a set sequentially (see Fig. 6), the images that are entered in the system will be processed by Raspberry on a Python-coded server [10], then analyze its content, later the application will be able to determine the number of boxes that are in the cells as well as their content and description, this information will be sent through sockets through a communication frame that will contain the results of the analysis to be stored in a database and then consulted by the PLC that manages the station in manual mode [11][18].

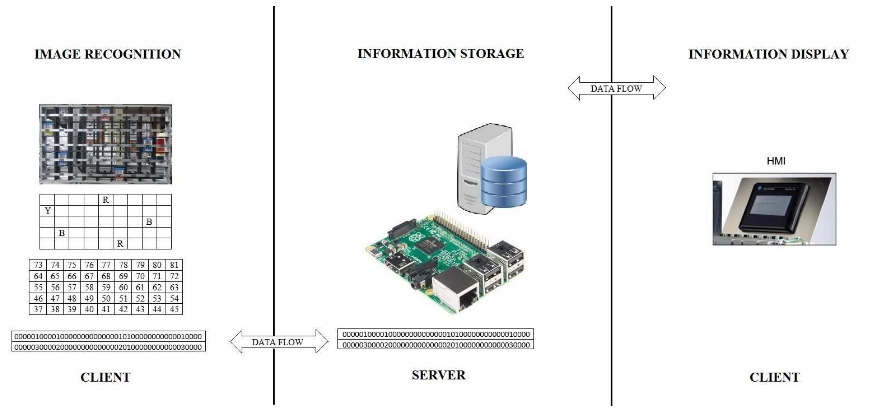

Figure 6: Functioning diagram

Proposed communication frame

Regarding the operation and the transmission process, provided by the above application, it is intended to implement a frame with the structure represented in table 2 [12].

Table 2: Proposed frame scheme

Frame Fields		Offset	Example	Description
Start Delimiter		0	INITX	Frame Beginning
Size		1	12	Size of the Content Information
Frame Specific Data	Type	2	1	Frame Type
	ID	3	1	ID number
	Destination IP Address	MBS 4 LSB 7	130.130.130.12	Destination IP Address
	Product Type	8	R	Product Color
	Product Position	9	43	Field Number
	Available Fields	10	10	Storage Empty Fields
End Delimiter		11	FINTX	

4. IMPLEMENTATION

Figure 7 presents the flow diagram of the proposal for the optimization of storage, focused on the product shelf cells to perform the positions and product states monitoring process [13].

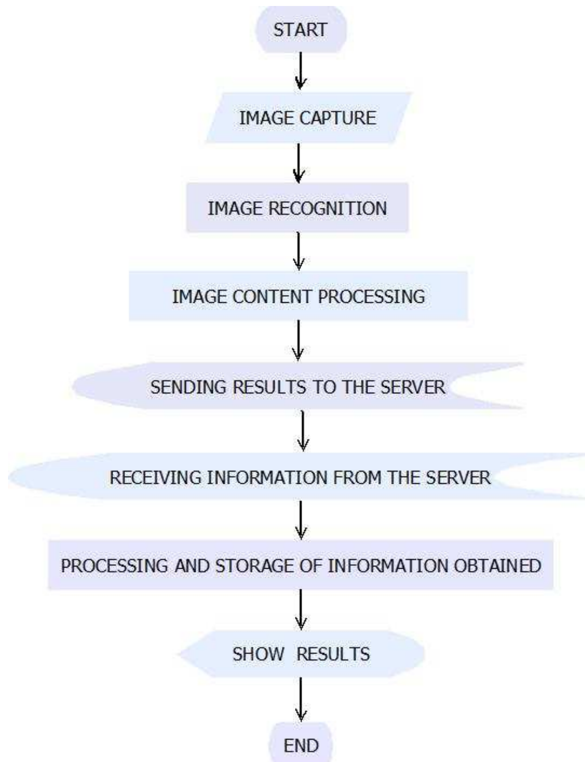

Figure 7: Actions flow

The fundamental key in this optimization targets the image recognition; this process focuses on performing the color detection on each of the warehouse's storage cells. The detection principle starts from analyzing the label's color of the storage cell, this process begin with the division to a matrix of a full color station photo, before the filtering and noise reduction for the image.

The process of image analysis focuses on performing RGB components detection for color images; the main criteria of image analysis is to find the three key colors for each cell [14], which are given by the container cover (Blue) and the content labels (yellow, red and blue). This is where a series of parameters that can affect the performance and efficiency of the image processing take place, since it is necessary to keep in mind that components such as hue, saturation and intensity are involved in the color model.

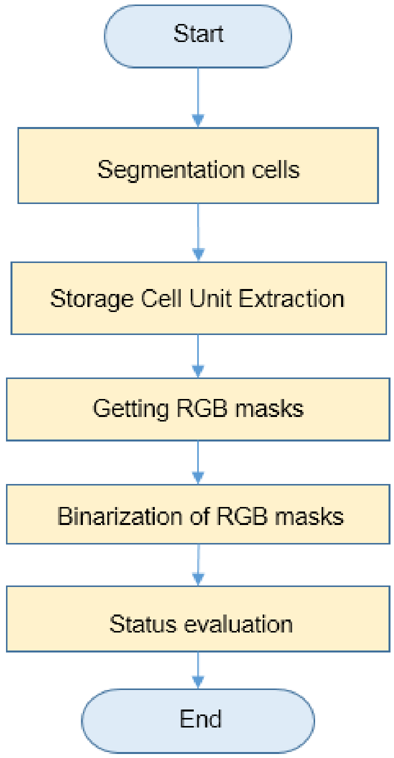

Figure 8: Flow diagram RGB analysis for color detection

The image processing for each cell is described in a specific way in Figure 8, it must be highlighted that in order to obtain the detected colors mask it is necessary to work with a range of detection [15], each color must handle a maximum and minimum value that establishes the precision of the mask to be binarized. Table 3 shows the established values for the detection level used by the Raspberry PI in the detection algorithm.

Table 3: RBG format detection range

Color	Rango RGB (python)
Yellow_H	30,255,210
Yellow_L	16,76,72
Red_H	12,255,255
Red_L	0,65,75
Blue_H	130,255,255
Blue_L	100,65,75

Once the color mask has been detected, a binarization is carried out, meaning that the mask will handle only two possible values; all of the detected pixels for a specific color will be painted white (1) and the rest will be painted black (0). This process is shown in Figure 9, where are displayed the masks that detect yellow and blue colors of a storage cell.

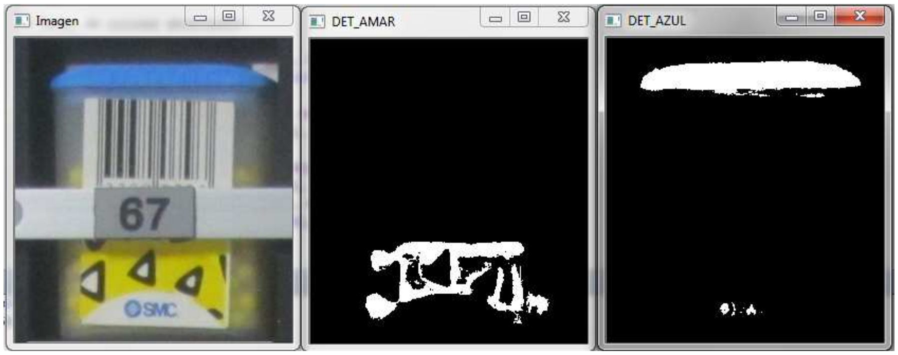

Figure 9: Color masks detected on cell image

Finally, the evaluation process is performed taking as reference a comparison of bits on several pixels of the mask, determining which type of label (color) corresponds to the cell and what is the specific position for the storage matrix [16]. With this information it is proceed to fill the communication frame which will send the Raspberry card to the server reporting the entire warehouse state.

5. RESULTS

The execution times measured for the color detection and space validation step were 42 milliseconds per cell average, calculated within 90 samples per color, including empty space samples, for a total of 360 samples of varied images. The total execution time for the analysis of the 81 storage cells for the station, including the communication frame generation, was 4.212 seconds.

The specific transmission of communication can be analyzed by any tool for network protocol analysis; Wireshark was used for the practical case of functioning and interconnection validation. Through this tool it was possible to capture messages sent from the control card. Fig. 10 shows the frame structure designed for the information transport (see table 2), validating this way the correct operation of the optimization allowing to check the state of the warehouse in real time from any computational device connected to the highly automated system (HAS 200) Ethernet network.

```
▷ Frame 65793: 12 bytes on wire (96 bits), 12 bytes captured (96 bits) on interface 0
▷ Ethernet II, Src: CameoCom_e6:5a:1b (00:40:f4:e6:5a:1b), Dst: Pi_rasp_45:6e:0c (10:fe:ed:45:6e:0c)
▷ Internet Protocol Version 4, Src: 130.130.130.12, Dst: 130.130.130.120
▷ Transmission Control Protocol, Src Port: 49922, Dst Port: 6030, Seq: 838, Ack: 160643, Len: 0
```

Figure 10: Ethernet frame Capture

Detailing the message content, it can be identified the data contained by the frame in a hexadecimal format (see Fig. 11), like positions, type of product and available fields.

```
0000  7c b7 33 d2 8a d0 00 40  f4 e6 5a 1b 08 00 45 00   |.3....@ ..Z...E.
0010  00 28 1f 91 40 00 80 06  4e 6b c0 a8 01 67 43 de   .(..@... Nk...gC.
0020  86 e6 c3 02 00 50 65 21  de 41 99 f6 49 43 50 10   .....Pe! .A..ICP.
0030  00 c4 38 4d 00 00                                  ..8M..
```

Figure 11: Data carried in the Ethernet frame

6. CONCLUSIONS

After investigating the operational features of the Highly Automated System HAS-200, it was possible to identify the limitations presented in the warehouse, which were taken into account for the development of the proposed telematics environment approach.

It has to be done an initial calibration and parameterization process determining a specific range of the RGB format components related to the colors to be detected, as there may be non-systemic factors that affect the accuracy of the image processing. By the same way, it is suggested to implement the image conversion libraries associated with the HSL and HSV models of Python to improve the filtering effects on the images to be treated.

The development of this optimization reduces the response times in addition to generating an automated operation, which increase the efficiency at the moment of operating the warehouse, ensuring a good performance for the storage processes and for the whole system.

Through the elaboration of a specific communication frame model between the client and the server, it was determined that it is possible to decrease the operating times for the different processes of the production environment.

ACKNOWLEDGMENTS

This work was supported by the District University Francisco José de Caldas Technological Faculty. The views expressed in this paper are not necessarily endorsed by District University. The authors thank the research group ARMOS for the evaluation carried out on prototypes of ideas and strategies.

REFERENCES

[1] SMC International Training (2017). Highly automated system manual HAS-200. http://www.smctraining.com/en/webpage/indexpage/517.

[2] SMC International Training (2017). Vertical warehouse technical description HAS-208. http://www.smctraining.com/en/webpage/indexpage/527.

[3] SMC International Training (2017). HAS-200 Photo gallery HAS-200. http://www.smctraining.com/en/webpage/indexpage/519.

[4] Grimmett R. (2016). Raspberry Pi Robotic Projects. Third Edition. Birmingham UK. Packt Publishing Ltd. Chapter 1. Getting Started with the Raspberry Pi. 5–39.

[5] Osorio F, Xinran M, Liu Y. (2015). Sensor network using Power-over-Ethernet. 2015 International Conference and Workshop on Computing and Communication (IEMCON), 1-7. doi: 10.1109/IEMCON.2015.7344442

[6] Molloy D. (2016). Exploring Raspberry Pi Interfacing to the real world with embedded Linux, First Edition, Indiana USA, John Wiley & Sons Inc. Chapter 5. Programming on the Raspberry Pi. 159–211.

[7] Jung C, Kang J, Jin, M. (2014). The development of Ethernet based radar and ECDIS image processing for Voyage Data Recorder. 2014 14th International Conference on Control, Automation and Systems (ICCAS 2014), 963-966. doi: 10.1109/ICCAS.2014.6987924

[8] Norris D. (2014). Raspberry Pi Projects for the Evil Genius. First Edition, New York USA, McGraw-Hill Education. Chapter 4. Camera Controller. 43-56.

[9] Mandliya H, Tandon A, Shah J and Rajpal R. (2014). Gigabit Ethernet Based Image Acquisition System for IR Camera. 2014 International Conference on Computational Intelligence and Communication Networks. 453-457. doi: 10.1109/CICN.2014.106

[10] Shovic J. (2016). Raspberry Pi IoT Projects: Prototyping Experiments for Makers, Washington USA, Technology in action. Chapter 1. Introduction to IoT. 2-8.

[11] Mhaski R, Chopade P and Dale M. (2015). Determination of ripeness and grading of tomato using image analysis on Raspberry Pi. 2015 Communication, Control and Intelligent Systems (CCIS). 241-220. doi: 10.1109/CCIntelS.2015.7437911

[12] Rodriguez R, Sossa J. (2012). Procesamiento y Análisis Digital De Imágenes, Ciudad de México, Alfa omega Grupo Editor. Chapter 2. Proceso de captación y formación de una imagen. 45-70.

[13] Viitanen M, Koivula A, Vanne J. (2015). Kvazaar HEVC still image coding on Raspberry Pi 2 for low-cost remote surveillance. 2015 Visual Communications and Image Processing (VCIP). 1-1. doi: 10.1109/VCIP.2015.7457898

[14] Szabó R, Gontean A. (2016). Industrial robotic automation with Raspberry PI using image processing, 2016 International Conference on Applied Electronics (AE). 265-268. doi: 10.1109/AE.2016.7577287

[15] Karvinen T, Karvinen K, Valtokari V. (2014). Make: Sensors: A Hands-On Primer for Monitoring the Real World with Arduino and Raspberry Pi, Canada, Market Media Inc. Chapter 6. Movement. 145-170.

[16] Sharon V, Karthikeyan B, Chakravarthy S. (2016). Stego Pi: An automated security module for text and image steganography using Raspberry Pi. 2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), 579-583. doi: 10.1109/ICACCCT.2016.7831706

[17] Livingston J, Umamakeswari A. (2015). Internet of things application using IP-enabled sensor node and web server. Indian Journal of Science and Technology. 8(9), 207–12. doi: 10.17485/ijst/2015/v8iS9/65577

[18] Siva D, Sudharson A. (2016). Environment Monitoring System and Traffic Control Using Vehicular Network. Indian Journal of Science and Technology. 9(45), 1–7. doi: 10.17485/ijst/2016/v9i45/99612