INSTITUTO TECNOLOGICO Y DE ESTUDIOS
SUPERIORES DE MONTERREY
CAMPUS MONTERREY
ARCHITECTURE AND ENGINEERING
GRADUATE PROGRAM

TECNOLÓGICO
DE MONTERREY®

Design and implementation of a Methodology to integrate the Vision and Robot Technologies

THESIS

MASTER OF SCIENCE IN MANUFACTURING SYSTEMS

BY:

DAVID ALEJANDRO HERNANDEZ CAMPOS

MONTERREY, N. L. DECEMBER 2011

INSTITUTO TECNOLOGICO Y DE ESTUDIOS
SUPERIORES DE MONTERREY
CAMPUS MONTERREY
ARCHITECTURE AND ENGINEERING
GRADUATE PROGRAM

TECNOLÓGICO
DE MONTERREY®

Design and implementation of a Methodology to integrate the Vision and Robot Technologies

THESIS

MASTER OF SCIENCE IN MANUFACTURING
SYSTEMS

BY:

DAVID ALEJANDRO HERNANDEZ CAMPOS

MONTERREY, N. L. DECEMBER 2011

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

CAMPUS MONTERREY

ARCHITECTURE AND ENGINEERING GRADUATE PROGRAM

Design and implementation of a Methodology to integrate the Vision and Robot Technologies

THESIS

MASTER OF SCIENCE IN MANUFACTURING SYSTEMS

by

David Alejandro Hernández Campos

MONTERREY, N.L. DECEMBER 2011

Design and implementation of a Methodology to integrate the Vision and Robot Technologies

A dissertation presented by

David Alejandro Hernández Campos

Submmitted in partial fulfillment of
the requirements to obtain the degree of

Master in Sciences
in Manufacturing Systems

TECNOLÓGICO
DE MONTERREY®

Thesis committee:

M.Sc. Ricardo Jiménez González
Ph.D. Federico Guedea Elizalde
Ph.D. Eduardo González Mendivil

Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Monterrey

December 2011

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

ARCHITECTURE AND ENGINEERING GRADUATE PROGRAM

The committee members hereby recommend the dissertation by **David Alejandro Hernández Campos** to be accepted as a partial fulfillment of the requirements to be admitted to the **Degree of Master in Sciences in Manufacturing Systems**.

Committee members:

RJiménez

M.Sc. Ricardo Jiménez González
Advisor

Federico Guedea E

Ph.D. Federico Guedea Elizalde
Member

Ph.D. Eduardo González Mendívil
Member

Ph.D. Ciro Ángel Rodríguez González
Director of Master in Manufacturing Systems
School of Engineering and Information Technolgies

December 2011

Dedicatory

This study is dedicated:

To God who is always there for me, giving me every day a reason to be thankful for my life.....

To my parents Eva and Abel whose love has always been the engine of my life.....

To my sister Mirna whose kindness and joyness is always a reason to smile....

To my brother Juan who is my guardian angel....

Acknowledgments

I would like to thank:

To M.Sc. Ricardo Jiménez for his support, availability, time and help in the development of this thesis work.

To the members of the committee Ph.D. Federico Guedea and Ph.D. Eduardo Gonzalez for their advices, recommendations and time dedicated on my thesis review.

To Ph.D. Antonio Sánchez from the UPV (Valencia, Spain) who accepted me in the internship where I developed the study case of my thesis, I really appreciate his help, gentleness, availability and patience.

To my friends Orlando, Xavier, Hector, Sergio, Hiram, Elias, Francisco, Angelica, Valeria, Itzel and Varinia, for their unconditional friendship, affection and support.

To Cony and Luis who became mi family in Monterrey and joined me through this stage of my life.

Design and implementation of a Methodology to integrate the Vision and Robot Technologies

The actual manufacturing systems have certain requirements and demands in order to be considered as profitable and efficient, some of these demands are the reconfigurability, flexibility, efficiency and accuracy. The integration of vision and robot technologies provides to the system features that fulfill the requirements mentioned, through the use of vision systems, reconfigurability, flexibility and efficiency are added to the application, in the other hand the use of a robot brings accuracy, repeatability and flexibility.

The integration of vision and robot technologies has various applications as assembly, path tracking, quality inspection, dimensional measuring, material handling, etc., although there are several researches and applications implemented, these studies are very specific and focused on certain devices, applications or products, so it is identified the need of a global and generic methodology that provides the general guidelines to implement a integration of these technologies. This thesis work presents a modular methodology to integrate vision and robot technologies and includes the most important factors considered when integrating these technologies, the methodology is compound by 5 modules: 1)Integration Objective, 2)Working Conditions, 3)Physic Integration, 4)Logic Integration and 5)Evaluation, these modules are divided in specific sub-steps in order to provide hints, recommendations and suggestions for the implementation.

The mentioned methodology was validated through the implementation of an automatic packaging process, the devices integrated were a FANUC$^{\mathrm{TM}}$robot, a Kinect$^{\mathrm{TM}}$camera and a SONY$^{\mathrm{TM}}$PC, the packaging process consists on the identification and three-dimensional location of the products (by the camera), the transference of this data to the robot (through the computer), and the execution of the motions needed to reach, grasp and dispose the products in their belonging spot(by the robot).

The integration of the study case was successfully implemented and evaluated, with those results the methodology proposed is considered as satisfactory accomplishing the main objective of the research.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Nowadays in the manufacturing industry different kinds of fabrication processes are utilized, process selection depends on the type, characteristics and properties of the product that is going to be manufactured. In the next figure it is shown an analysis developed by Kruger (2009) where a representative indication of manufacturing process frequency is presented.

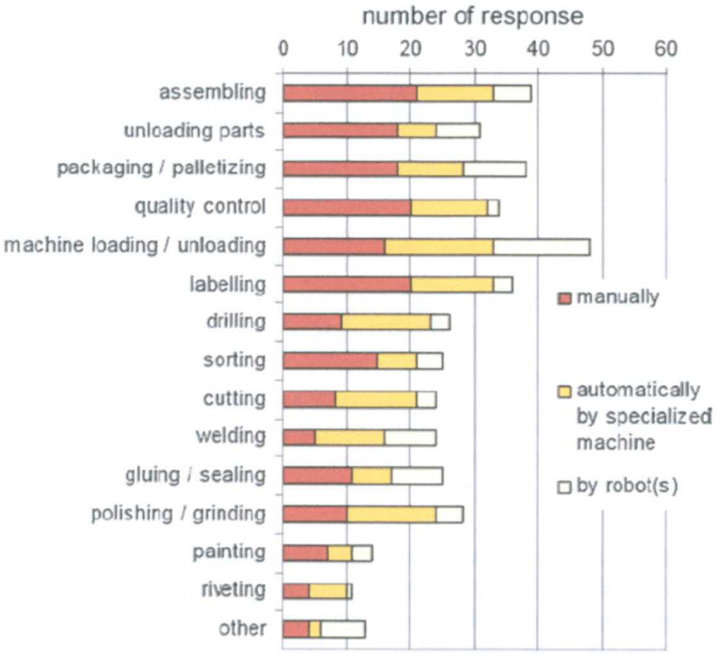


Figure 1.1: Manufacturing process frequency

As it is shown every manufacturing process can be done manually, automatically by specialized machines or by robots. When robots are used, and depending on their degrees of freedom they are able to realize very sophisticated and complex movements required in a accurate, flexible, reliable, and fast way. That is the reason that the use of these devices has been increasing due the pass of the years, and will continue growing as is shown in the projection analysis for the 2015 presented by López (2008), where the percentage of activities carried out by robots and automatic working systems is estimated.

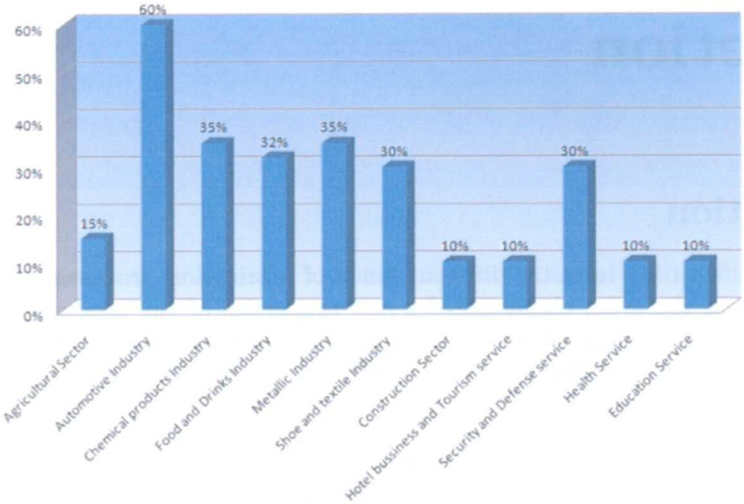


Figure 1.2: Projections about the percentage of activities carried out by robots and automatic working systems in 2015 (from Lopez, 2008).

The use of robotic devices is one of the principal activities that allow and facilitate the establishment of Flexible Manufacturing Systems (FMS), The analysis of the system's flexibility has been a research direction for many authors, specifically for Upton(1994) who in his work presented in 1994 defines the flexibility as the capability of a system to change and react with the minimum affectation in time, effort, cost and performance.

The flexibility in a manufacturing system is also defined as: "The attribute that allows the system to produce a high variety of product models, with a certain variation level and without interruptions or high investments"(Chacón, 2003).

In order to use robots in the manufacturing process it is not enough to count only with a robotic arm, the use of different tools that allow the physic interaction between the robot and the pieces of the final product is inherent, these tools are best known as fastening and referencing devices, examples of these devices are: Grippers, screwdrivers, welding electrodes, proximity sensors, vision systems, lasers, etc.

The improvement of the characteristics and features of vision systems (also called machine vision) has converted this device as one of the most important components in automation, according to Conolly (2009). Machine vision is beneficial in the inspection of components and assemblies, with the advantage of 100 percent inspection rather than spot checks. It also allows pass/fail data to be logged and integrated with other factory statistics, and the machine vision system can be inter faced with factory automation, for example outputting signals to trigger reject mechanisms. On the software side, pattern-matching algorithms are extending into the third dimension, thanks to the availability of multi-core personal computers. This facilitates much more sensitive robot guidance, opening up a wealth of assembly applications from the car-engine building to cake-icing. Machine-vision component gauging in 3D can remove the need for tactile probes.

Brosnan & Sun (2003), Conolly (2008) and Hardin (2009) identify one of the main advantages of the use of machine vision, the capability to implement a reliable and powerful image analyzer with cost effectiveness, the costs of vision systems have decreased in an important quantity, Fabel (1997) compared the cost of a vision system in the 80's decade and one in the end of the 90's decade and identified a cost reduction of more than the 60%, this trend has been continuing through the years, converting the machine vision in one of the principal components of manufacturing systems.

Several researches have been focused in applications of robotic and vision technologies. Conolly (2008) presents the "Artificial intelligence and robotic hand-eye coordination" article, some of the examples presented are:

- A group of eight FlexPickers from ABB, pick and inspect 50,000 croissants per hour at a Spanish bakery. The robots respond very quickly to the random placings of the croissants, directed by the software based on Cognexs geometric pattern matching algorithm PatMax.
- A sheet metal application in the Skoda Auto manufacturing process. An ABB robot cuts sheets up to 1200 mm x 9500 mm with 0.1 mm precision. The robot is guided by Sony ST50CE cameras connected to two Matrox Meteor-II/multichannel frame grabbers. Matroxs GMF module locates the position of the sheet metal and measures it, enabling the robot to follow the required path.

Connolly (2009) presents the "Machine vision advances and applications" article, where some of the most important applications are mentioned, for example:

- ABB Robotics robot guidance system is used in automated engine assembly. Toyota, Ford, Honda GM and Chrysler use TrueView which acquires an image of the robots work area and applies geometric path recognition to find features within the target.
- Aerospace manufacturers use vision for part identification and inspection, and for drilling and riveting.
- At the Mikron factory a multi-camera inspection systems checks that the speakers, diodes, window pads and spring assembly are all presented and correctly located.
- Prettl Appliance Systems Deutschland GmbH is using In Sight cameras to inspect control panels for washing machines. A six-axis robot holds the panel and moves it relative to the camera, which checks the controls, LED indicators, lettering, and quality paintwork.

1.2 Problem statement

In reference with the information presented in the last section it can be noticed the importance and versatility of the Vision Robot Technologies (from now on VRT) Integration, even tough a lot of applications examples, mathematical models and integration improvements are published, it is identified the need to establish a integration methodology of these technologies.

In this research it will be presented an analysis to determine a methodology which will provide objective and generic rules to follow in order to establish a Vision Robot Technologies Integration based in methods and techniques that will propitiate the improvement in the manufacturing operations performance, the application of this methodology enables the VRT integration to realize different manufacturing operations with a better performance talking about cycle time and external devices needed, therefore it is necessary to identify in this methodology all the factors that must be involved in the technologies integration in order to determine the best order of the steps and recommendations.

There are some methodologies referred to VRT integration, but normally are focused in certain application, or delimited for some components of the integration (software, hardware, illumination, network control, etc), this research is intended to develop a generic and general methodology, where generic means the possibility to apply and implement the methodology with no predetermined devices (specific robots or vision systems), and the general attribute is referred to consider as possible the bigger quantity of parameters to include, select and evaluate through the methodology application.

1.3 Objectives

1.3.1 General Objective

Develop and validate a generic methodology in order to establish the Integration of VRT considering the most important factors in the design and implementation of this integration.

1.3.2 Specific Objectives

1. Identify the most important factors involved in the integration between the Robot and the Vision Technologies.
2. Establish the design or implementation methodology for each factor involved in the general methodology.
3. Identify possible links between important issues identified in the last objectives.
4. Develop de methodology in base with a real case (experiment) and bibliographic research.
5. Lead and emphasize in another case of study in order to validate the methodology proposed.

Through the validation process, it is necessary to:

- Establish the integration objective and evaluate the viability of the integration according with the devices and facilities available.
- Develop the Integration of Vision-robot technologies through the proposed methodology.
- Evaluate the results and propose methodology improvements

1.4 Scope of the research

The methodology presented in this research is compound and complemented by other specific methodologies, the establishment of a general methodology will be developed through this study, even tough the methodology presented is generic for any application of a VRT integration, the validation process will be oriented to a certain application which is a Package Operation.

The Package Operation consists on the identification of previously defined pieces, the acquisition of their 3D position, the computation of their orientation in one cartesian plane, the objects grasping, and the placement of the objects in their respective spot of the package fixture.

This research is based in two experiments, the first one (1) refers to a integration of VRT in a Flexible Manufacturing cell (Monterrey, Mexico) where the integration will be realized without a specific methodology, this study case will help to develop the methodology simultaneously with the integration development. The second set up(2), will be implemented with the previously generated methodology, the validation experiment consists on VRT integration to perform a package operation of known components, the detailed description of the experimental set-up is presented in section 5.1.

The experiment 1 is compound by a robotic arm and a vision system, both parts are components of a Flexible Manufacturing Cell developed and installed in the Instituto Tecnologico y de Estudios Superiores de Monterrey, Monterrey Campus, the objective of this integration is the assembly operation of the "demo" processed in the Manufacturing Cell, for more information see Section 3.1.3.

The experiment 2 is compound by a robotic arm, a Personal Computer, a Server and a non-industrial vision system, this integration is intended to perform a package operation of certain pieces which position and orientation are unknown, for more details see Chapter 5.

1.5 Justification

As it is shown in the projection of Lopez (2008) (Figure 1.2) the use of the robots has been and will be increasing due the past of the years, the application of the robots is present in almost every industrial sector, but this technology could not be able to perform as good as actually does without the improvements on its assistance devices, networking and control software, all these capabilities advances have been leaded by the manufacturing needs.

Bruccoleri (2004) and Jozwiak (2010) present this manufacturing trend by the application of reconfigurable manufacturing systems (RMSs), the RMSs improves the FMSs characteristics by not only providing flexibility to the system but adding cost efficiency, reliability and easy software/hardware change ability. All these features are needed in response of the market requirements changes, introduction of new models, increase in jobs arrival rates, and reworks are only few of the situations that currently happen in industry.

The integration of VRT presents advantages that allow it to be part of the RMSs, one of the major advantage is the one referred to fixtureless manufacturing, according to Wang (2010) the fixture can be defined as a "mechanism used in manufacturing to hold a workpiece, position it correctly with respect to a machine tool, and support it during the process". Bone (2003) and Wang(2010) emphasize the cost impact of the fixtures usage, according to Bone (2003) the redesign, manufacture, and installation of the fixtures cost can be very important (eg. in the $ 100 millions order per year for the automotive industry) and according to Wang (2010) the costs associated with fixture design and manufacture generally can account for 10% to 20% of the total cost of a manufacturing system.

The fixture is widely used in manufacturing, e.g. machining (including turning, milling, grinding, drilling, etc), welding, assembly, inspection and testing, also it is good to notice that in the use of the fixtures beside the use of the fixing devices it is required the utilization of different kind of sensors which provides to the robot information about the presence or absence of the components, this function (of the sensor) can be sophisticated and replaced by the use of a vision system which can be able to recognize the presence, orientation and position of the environment communicating this information to the robot in order to execute the actions belonging to the process without the need of fixtures and sensors.

It can be concluded that the use of VRT brings reconfigurable information for robots (by the use of new machine vision algorithms), this advantage allows the system to process images for different products (surfaces, sizes, materials, etc.), furthermore by the use of algorithms for object identification and recognition, the VRT integration brings the capability of **fixtureless** process implementation.

It can be inferred the need to establish techniques and methods to add better characteristics to the manufacturing systems, the use of a vision system could avoid of the problems presented before (as the excessive use of fixtures and sensors, time lost in the model change, etc.), so the need to develop the integration of VRT, and most important the need to establish a generic methodology to realize this Integration is presented and will be considered as the main contribution of this research.

1.6 Background

Several researches have analyzed the integration of VRT, however most of them are focused in just some specific sections of the integration, Sitte (2007) argues that "there are no methods that support or require a complete description of the product system" (talking about robot vision systems) and identifies the next issues:

- Methods proposed consider just specific design domains of mechanics, electronics or computer software, and can not be transferred to other domains.
- There is no system description, and comparison between methods it is hard to realize.
- Methods specify a process, but the steps are not clearly and/or generic enough.

All these needs identified by Sitte(2007) are present in most of the bibliography papers of this research, for example Pauli (2001) applies visual servoing mechanism not only for handling objects, but also for camera calibration, object inspection and assembly, however, there is no consideration for other aspects as lighting, quality inspection, or communication software. Bone (2003) presents a vision guided fixtureless assembly method where grippers design, part locating, 3D pose measurement and system architecture are considered but other aspects as calibration or communication interface are omitted. Watanabe (2005) presents a paper where robot motion is autonomously generated to identify the camera view line, with this algorithm the measurement of Target Position and measurement of workpiece pose can be obtained efficiently but application objective, mechanic considerations in camera placing or and Vision/Robot communication interface are not considered. Pea Cabrera (2005) presents a methodology for online recognition and classification of pieces in assembly tasks, in this methodology it is considered the control architecture description, the communication software, commands between vision system and robot and a object recognition methodology, some other aspects like camera calibration or identification of target points are omitted. Misel Brezak (2008) focus on real time tracking of multiple robots by the use of designed robot marks, camera calibration, vision algorithm and robots pose measurement are other aspects contained in the research, nevertheless lighting technique, architecture control or physical target characteristics were not included. Some other researches like Jiang (2001), Shin (2002) and Gottfried (2008) presents some machine vision applications as object or surface quality inspection, they present vision algorithms and some mechanical mounting considerations, but omit lighting type selection, interface communication development or system architecture.

The "Methodic design of robot vision systems" presented by Sitte(2007) helps to describe the hole complex vision system, find out the gaps or failures of the system design and create new solutions, this method is based on simple ideas organization. The method specifies the presence of demands, components and functions in Robot-Vision systems also finds a relation between those parameters and evaluates the influence level between them. The author proposes some possible or typical demands, functions and components for a generic vision system, then in base with a real robot vision system description develops the matrix analysis and as result obtains the characteristics needed to satisfy the application requirements, even though Sitte (2007) involves more parameters in the Robot-Vision system design, there are some guidelines missing in this methodology, as the application objective consideration, lighting technique and type, interface design, and others.

Section 1.5 and last paragraphs provide important information about the need to establish a generic methodology that could support most of the principal aspects of a VRT integration. The methodology developed will contain as much as aspects possible, this methodology can not address every aspect of VRT integration due to the large number of applications and specific integrations, one of the principal characteristics of this new methodology is to provide a generic guidelines when very variable parameters are being considered, in the other hand for more defined parameters the guidelines should be more specific.

The following table shows the previous analysis, evaluating the background researches with the factors that are considered as the most important in the VRT integration.

Table 1.1: Methodologies and researches analysis

Research	Integration Methodology	Integration Objective				Working conditions			Physic interaction		Logic Integration		Implementation	Evaluation
		Assembly	Inspection and verification	Path Tracking	Others	Hierarchy establishment	Workpiece characteristics	Illumination	Camera Mounting	Mechanical design	PC-Robot	PC-Vision system		
Machine vision advances and applications (C. Conolly, 2009)		●			**O**		○	**O**			**O**	**O**	●	**O**
Robust and accurate global vision system for real time tracking of multiple mobile robots (M. Brezak, 2008)				●		**O**	○			**O**			●	**O**
Autonomous Processes in Assembly Systems (B. Scholz, 2007)		●											●	**O**
Methodic Design of Robot Vision Systems (J. Sitte, 2007)	●			**O**	●								●	**O**
Methodology for the Vision Robot Systems Integration (V. Espinosa, 2006)	●	**O**	●								●	●	●	**O**
FROVIS:A Flexible Robot Vision System (R. Cuevas, 2005)	●	**O**			●						●	●	●	**O**
Autonomous Visual Measurement for Accurate Setting of Workpieces in Robotic Cells (Watanabe, 2005)	●				●								●	**O**
Machine vision approach for robotic assembly (M. Peña, 2005)	●	●									**O**	**O**	●	**O**
Vision-guided fixtureless assembly of automotive components (G. Bone, 2003)	●	●					●				**O**	**O**	●	**O**
An efficient multi-camera, multi-target scheme for the three-dimensional control of robots using uncalibrated vision (E. Gonzalez, 2003)	●			●									●	**O**
A minimum-time algorithm for intercepting an object on a conveyor belt (I. Sang, 2003)					●								●	**O**
Vision system for defect imaging, detection, and characterization on a Specular surface of a 3D object (D. Aluze, 2002)			●		**O**								●	**O**
Vision-based integrated system for object inspection and handling (J. Pauli, 2001)		●	●				**O**						●	**O**
Intelligent robotic assembly (S. Balakrishnan, 2000)		●											●	**O**

●	Presents a detalied explanation of the subject as a part of the methodology proposed or used
O	Presents information regarding the subject, but does not include it as a methodology step
○	Mentions the subject and relate it with the main objective

1.7 Thesis Organization

This thesis work is structured in the following way:

In the second Chapter the description of the two main technologies involved in this study is presented, the Machine Vision and the Robotic Technology are described. In reference with the vision systems, the performance of a generic vision system, different kinds of applications, and illumination characteristics are presented, on the other hand for the robotic technology the application of robots for industrial operations, and fundamentals of robot kinematics are described and analyzed.

In the third Chapter the analysis of the Vision Robot Technologies integration is shown, the main components of a VRT integration are defined and described, some examples of successful VRT integration are presented, and finally the previous methodologies for the VRT integration are presented and analyzed in order to improve or modify them according to the study orientation.

In the fourth Chapter the methodology for VRT integration is developed and explained, the methodology is compound by several sub-methodologies according to the main topic that is being analyzed, the proposal considers 5 main topics when integrating this technologies: Integration Objective; Working Conditions; Physic Integration, Logic Integration and Evaluation.

In the fifth Chapter a case of study is presented, the proposed methodology is implemented and the obtained results are shown. The sixth and last chapter contains the Conclusions and Future work for the proposed methodology.

Chapter 2

Fundamentals of Vision and Robotic Technologies

2.1 Vision Systems

2.1.1 Introduction

Generally the vision systems are compound by different parts as: a camera, the digitalization hardware, a PC, and the hardware and software that enable the communication between them. The following picture presented by Malamas (2002), shows a representative image of a typical industrial vision system and its components.

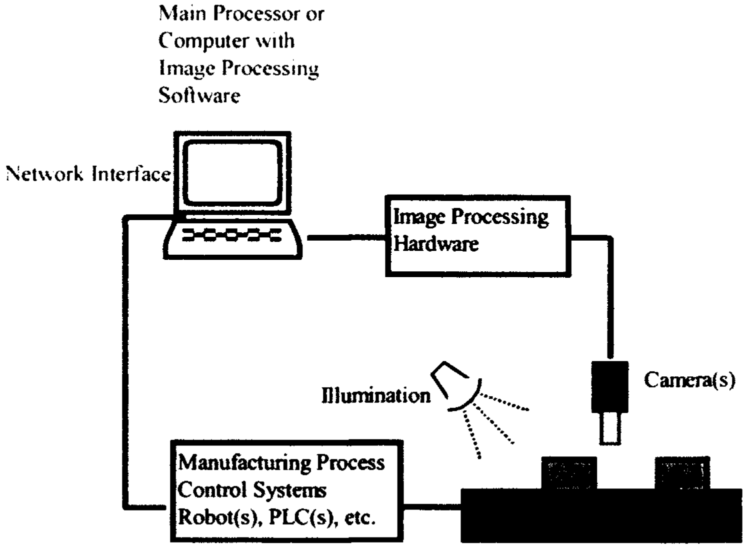


Figure 2.1: A typical industrial vision system

General description of the vision system:

The Personal Computer (PC) is used to process the images that have been acquired, this function is accomplished by the utilization of classification, processing and analysis of images software. The images are obtained by one or more cameras located on the action zone of the system, there are two configurations of mountings for the cameras, the description and analysis of this options is presented in the section 3.1.2. The zone or scene of action in the system must be arranged and illuminated in a proper way to facilitate the reception of the image characteristics needed for the analysis.

Description of the vision system performance:

The operation and interaction of all the components of the vision system allow the realization of the following basic functions (Malamas, 2002; Groover, 2008 & Suarez, 2009):

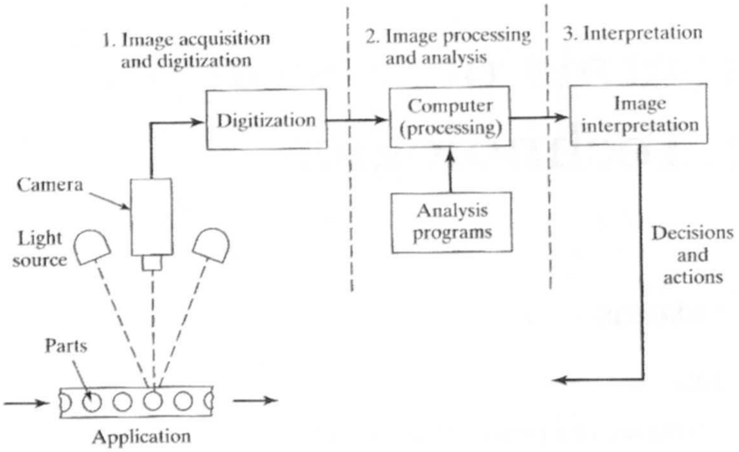


Figure 2.2: Basic functions of vision systems

1. **Image acquisition and digitization**

   The images of interest are detected and acquired, this digital aquisition is realized by a camera or digitizing systems focused to the scene of interest. The image is obtained through the conversion of the visualized area into a matrix of discrete picture elements (called pixels),each element has a value that is proportional to the light intensity of that zone of the image. The next Figure shows this procedure:

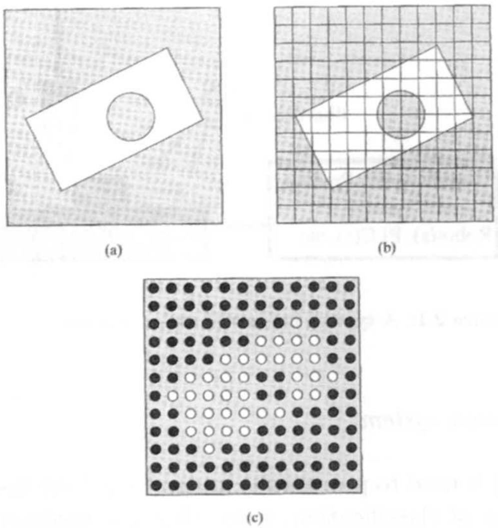


Figure 2.3: Image digitization

One image of the simplest vision system is shown in the Figure 2.3, this is called binary vision system. In the binary vision the light intensity of each pixel is finally reduced to one of two options, black or white, this depends if the intensity exceeds or not a threshold level. A sophisticated vision system is able to distinguish and process different gray scales in the image, these systems are called grayscale vision systems,and they are capable of determine not only an object's outline or area, but also it's superficial characteristics as texture and color. The grayscale systems usually use 4,6 and 8 bits of memory, identifying in the 8 bits case 256 illumination levels. (Groover,2008)

2. **Image processing and analysis**

Once that the images have been acquired , they are filtered in order to remove the digital noise or other unwanted effects caused by the illumination system. A lot of images analysis techniques have been developed, the most important are described below:

Segmentation.- The segmentation techniques are intended to define and separate interest regions of then image, this technique can be applied trough:

- *Thresholding:* involves the conversion of the light intensity value of each pixel into a binary value, white or black. This conversion it is made by the comparison of the light intensity value with a previously defined threshold value.
- *Edge Detection:* involves the determination of the edges location between an object an it's environment

Special features extraction.- The set of characteristics or features of interest is computed, as an example of this features are: size, position, outline measure, zone texture, etc. This set of data forms the image description.

3. **Interpretation and decisions**

Once that the previous functions of the system have been performed, the next step for the vision system is the application or utilization of the information processed and/or acquired before, according to Cuevas (2005), Espinosa (2006), and Suarez (2009),this applications can be categorized as follows:

- *Measure:* Is the basic application of the vision systems, where the system provides to the user dimensional data of he object or region of interest, the features that can be measured are: diameter, length, angles, level, etc.
- *Inspection:* In this application the vision system provides to the user information corresponding to the quality of the object, considering the presence of superficial damages, incomplete pieces or geometries, bad-oriented pieces, recognition of assembly components, etc. The information obtained in this application is usually used in the binary (go, no-go) decisions.
- *Identification:* Application in which the vision system recognizes and classifies an object, afterwards the orientation and position determination, usually these operations are followed by a robotic action. The identification actions include: pieces identification, palletizing, pick an place, etc.
- *Tracking:* In this kind of application the vision system leads the operation of a robot or another visual entry device. Some of the industrial applications are: placement of pieces, moving pieces pick and place, path tracking in arc welding or painting, etc.

It is good to mention that the requirements for the design and development of a vision system vary according to the specific application, it will also depend on the tasks to be executed, the environment where it performs, and the operation velocity.

Malamas *et. al.* (2003) presents a set of desirable requirements for the vision systems used in industrial applications:

- **Multi-level process.-** The processing varies from a low level (filtering, thresholding,etc.), through medium level (segmentation, features computation, etc.) to high level (object identification, images classification, etc.). An image analysis software must be able to realize the analysis in every level mentioned before.
- **Ease of manipulation.-** Graphic User Interfaces, visual programming, and code generation are usual features that allows the application development.
- **Dynamic range.-**The new vision sensors (CMO'S) offer a high dynamic range, and a faster image acquisition. The vision software must be able to handle and process this kind of information.
- **Expansibility.-**The vision software must offer the option to add new or better image processing algorithms.

2.1.2 Optics and calibration

Camera optics

In base with Steger *et. al.* (2008), the light can be considered as rays that propagate in straight lines in a homogeneous medium. The Pinhole camera is the simplest configuration of a optic device which represents the behavior of the vision systems. The Figure 2.4 shows this configuration:

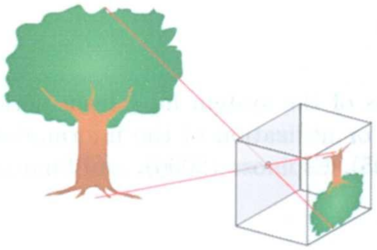


Figure 2.4: Pinhole camera model

The object of the left side is projected on the plane of the right side, this right side is one of the faces of a box into which a pinhole has been made on the opposite plane to the image plane. The pinhole camera model produces an upside-down image of the object.

The size of the projected image will depend on the distance relation between s and c, the h^1 can be computed in base with similar triangles law:

$$h^1 = h\frac{c}{s} \tag{2.1}$$

Where h is the height of the object, s is the distance of the object to the projection center, and c is the distance of the image plane to the projection center.

Using this principle the width of the projected image could be obtained by the same relation:

$$w^1 = w\frac{c}{s} \tag{2.2}$$

Where w is the width of the object, s is the distance of the object to the projection center, and c is the distance of the image plane to the projection center.

These two equations describe the behavior principle of a vision system without considering any extra devices.

Lenses

A lens is an optical device through which light is focused in order to form an image inside of a camera, in vision systems this image is formed on a digital sensor (CCD). The purpose of the lens is to create a sharp image in which fine details can be resolved.

In industrial applications there are certain lenses which are more frequently used (Espinosa, 2006):

- **Wide Angle.**-the focal distance is smaller than 5mm (standard distance), has a wide field depth and a reduced focal length.
- **Telephoto.**-the focal distance is bigger than the standard, has a reduced field depth and increases the far objects.
- **Zoom** The focal length is between 35 and 70 mm.
- **Macro** The projected image size is equal to the object size.
- **Telecentric** Does not have perspective distortion.

Camera Calibration

The camera calibration process is one of the most important step during the vision systems application, the calibration accuracy will determine the measurements exactitude of the scene, that is the reason why a lot of research has been done in this area, Sanchez Antonio (2008) presents a research in which a complete calibration method is proposed, this method is based on a complete review about the previous methods (Tsai,1997; Kanade *et. al* 1997; Raskar, *et al* 1998; Salvi, *et al*, 2002 and Zhang,2002,2008).

As a result of the calibration process the intrinsics parameters of the camera are obtained, the camera's matrix is a 3x3 array which contains these parameters:

$$A = \left\{ \begin{array}{ccc} \alpha_x & \gamma & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{array} \right\} \tag{2.3}$$

Where u_0 *and* v_0 represents the 2D coordinates of the principal or focal point, ideally this point should be in the center of the 2D image. α_x *and* α_y refer to the focal distance expressed in pixels, and γ represents the skew factor between x and y axis.

The extrinsic parameters of the camera regard to the rotation matrix R and the translation vector t, these parameters allow the coordinates transformation between camera's coordinate system and world's coordinate system.

The next image shows the intrinsics parameters representation and the projection of one 2D point(image plane) into a 3D point (in reference with camera Coordinates System), and the transformation to the World Coordinate System.

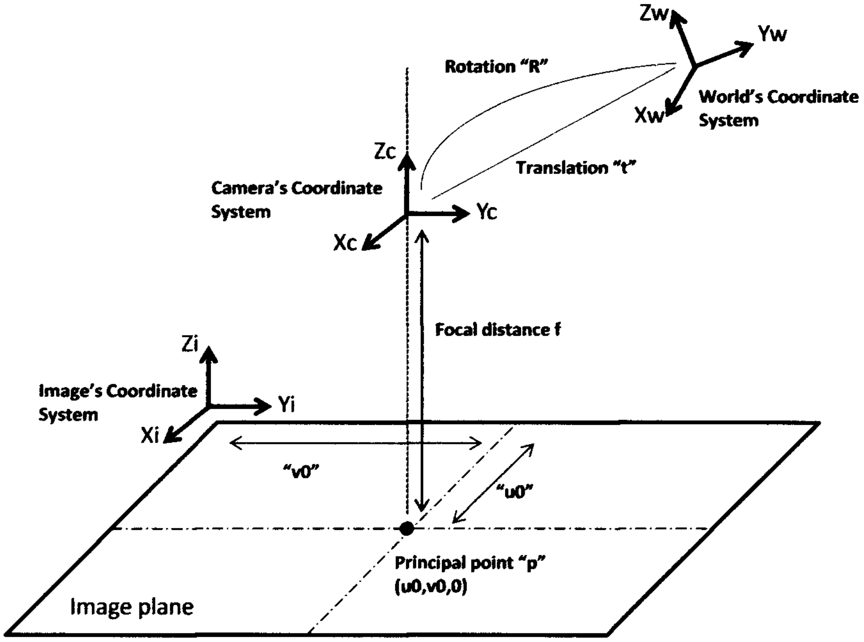


Figure 2.5: 3D projection

2.1.3 Machine Vision Algorithms

Data Structures

Image.-Refers to the set of data delivered by the acquisition device to the computers memory, the RGB (or color) cameras return three samples per pixel so a three channel image will be delivered.

An image channel can be considered as a two-dimensional array of numbers. Hence the gray value at the pixel (r,c) can be interpreted as an entry of a matrix: $g = f_{r,c}$. The gray value will be discretized to 8 bits so the set of possible gray values will be from 0 to 255.

All information above can be formalized as:

An image channel f of width w and height h is a function from a rectangular subset $R = \{0, ..., h-1\}$ x $\{0, ..., w-1\}$of the discrete plane $\mathbb{Z}^2$.

Region.-is an arbitrary subset of data of the discrete plane $R \subset \mathbb{Z}^2$. The Region Of Interest (ROI) is sometimes called the domain of the image because it is the domain of the image function f. Mathematically regions can be described as sets or by the use of characteristic function of the region:

$$XR(r,c) = \left\{ \begin{array}{ll} 1 & (r,c) \in R \\ 0 & (r,c) \notin R \end{array} \right\}. \tag{2.4}$$

This second definition suggests the use of binary spots to represent regions. A binary image has a gray value of 0 for points that are not included in the ROI and 1 (or any number different from 0) for points that are included in the ROI.

Contour.- this structure is not necessary pixel-precise, the application may require an accuracy that is higher than the pixel resolution of the image.

Contours can basically be represented as a polygon, an ordered set of control points (r_1, c_1), where the ordering defines which control points are connected to each other. in the computer, the contours are simply represented as arrays of floating-point row and column coordinates.

Geometric Transformations

If the position and rotation of the objects cannot be kept constant with the mechanical setup, a rotation and translation correction is needed. The mounting conditions sometime can make that the distance between the object and the camera changes, leading to an apparent change in size of the object. The following representation consider the use of basic transformations in order to get or represent an affine transformation:

$$\begin{pmatrix} 1 & 0 & t_r \\ 1 & 0 & t_c \end{pmatrix} Translation \tag{2.5}$$

$$\begin{pmatrix} s_r & 0 & 0 \\ 0 & s_c & 0 \end{pmatrix} \text{Scaling in row and column direction} \tag{2.6}$$

$$\begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \end{pmatrix} \text{Rotation by an angle of } \alpha \tag{2.7}$$

Image Segmentation

Segmentation is an operation that takes an image as input and returns one or more regions or subpixel-precise contours as output, there are a lot of segmentation algorithms, the simplest threshold operation is mathematically represented by:

$$S = \{(r, c) \in R \mid g_{min} \leq f_{r,c} \leq g_{max}\} \tag{2.8}$$

Threshold operation identifies al points in the ROI of the image that lie in a previously chosen range of bright values. Steger et. al (2008) specify that often $g_{min} = 0$ or $g_{max} = 2^b - 1$ is used.

The thresholds g_{min} and g_{max} can be fixed only if the illumination of the image will not change. Since the threshold operation is based on the gray values themselves, it can be used whenever the object to be segmented and the background have significantly different gray values(Steger et. al., 2008).

Feature Extraction

Through the image segmentation the regions and contours can be obtained, but there could be some characteristics which are important, so there is a need to remove unwanted parts of segmentation. As Steger et. al. (2008) presents, this characteristics are called features, typically they are real numbers, and can be related to:

- **Region Features.-** By far the simplest region feature is the area of the region:

$$a = \mid R \mid = \sum_{(r,c)\in R} 1 = \sum_{i=1}^{n} ce_i - cs_i + 1 \tag{2.9}$$

The area of the region is simply the number of points $| R |$ in the ROI. If the region is represented as a binary image, The first sum has to be used to compute the area; whereas if a run-length representation is used, the second sum can be used.

- **Gray Value Features.-** The mean gray value is a measure of the brightness of the region. A single measurement within a reference region can be used to measure additive brightness changes with respect to the conditions when the system was set up. The mean gray value is a statistical feature. and represented by:

$$\overline{g} = \frac{1}{a} \sum_{(r,c)\in R} g_{r,c} \tag{2.10}$$

Another statistical feature is the variance of the gray values:

$$s^2 = \frac{1}{a-1} \sum_{(r,c)\in R} (g_{r,c} - \overline{g})^2 \tag{2.11}$$

- **Contour Features.-** Steger propose to assume that a closed contour is specified by $(r_1, c_1) = (r_n, c_n)$. Let the subpixel-precise region that the contour encloses be denoted by R. Then the moment of order (p, q) is defined as:

$$m_{p,q} = \int\int_{(r,c)\in R} r^p c^q dr dc \tag{2.12}$$

2.1.4 Illumination systems

It is known that the behavior of a vision system will be constrained by two factors that affect directly the quality of the image acquisition. These factors are the Lighting and the lens of the system, a good lighting on the scene will provide enough illumination in order to distinguish the patterns of interest, remembering that the vision system translate the colors of the workpiece into brightness levels that are highly related with the light that is reflected by each pixel.

Bachem *et. al.* (2001) presents a examination of cameras under different illumination conditions, the experimental set up of this works consists on a robot working cell, where four illumination conditions where used, one of the objectives of his work is to present a comparison between this lighting conditions and its impact on the percentage of visible model segments.

The next figure represents the experimental set up used by Bachem *et. al.*(2001):

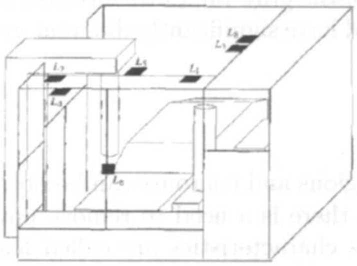


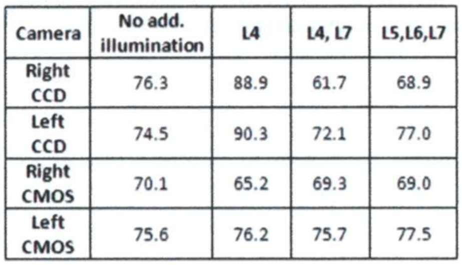
Figure 2.6: Alignment of the light sources in a robot working cell

The investigation was run under the next conditions:

1. no additional artificial light
2. one (L4)
3. two (L4 and L7)
4. three spot lights (L5, L6 and L7)

In the next table the results of Bachem *et. al.*(2001) are exposed, as it can be observed, the variation of illumination conditions strongly affects the vision system performance.

Table 2.1: Percentage of visible model segments covered by the system



Camera	No add. illumination	L4	L4, L7	L5,L6,L7
Right CCD	76.3	88.9	61.7	68.9
Left CCD	74.5	90.3	72.1	77.0
Right CMOS	70.1	65.2	69.3	69.0
Left CMOS	75.6	76.2	75.7	77.5

The way or configuration in which the light impacts the workpiece surface is called the type of illumination, Suarez (2009), Perez (2010) and Valiente (2010) present a general description of the most important types of illumination:

Direct Front Lighting

The light impacts directly the surface to be checked with a small incidence angle, as is shown in the next figures:

Figure 2.7: Direct front lighting

- *Advantages.*-Flood or spot lighting is easy to set up and use and provides maximum contrast in the image.

- *Disadvantages.*- Tridimensional objects will cause shadows and shiny parts will cause specular reflections.

- *Applications.*- It is helpful to maximize contrast in low contrast images and strobes can be used to freeze images of moving parts. Some application examples are shown in the next figure

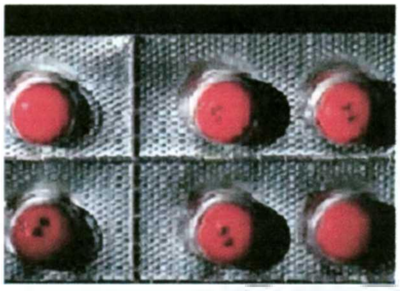


Figure 2.8: Direct front lighting application

- *Hints.*-Use two or more spots lights to minimize shadows also some times the shadows can be used to improve contrast.

Back Light

The object to be inspected is located between the camera and the illumination lamp as is shown in the next figure:

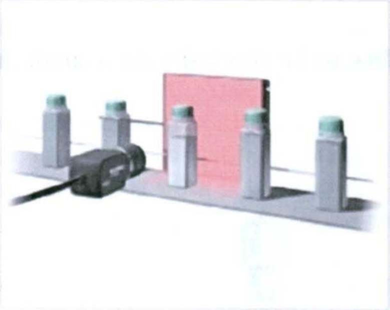


Figure 2.9: Back light

- *Advantages.*- Provides maximum contrast between part and background and simplifies the image providing a silhouette of the part.

- *Disadvantages.*-Surface details is lost (part is black on a white background) and object in fixtures often can not be easily identified.

- *Applications.*-Useful for dimensional measurements such as gauging object edges or through holes. Some application examples are shown in the next figure:

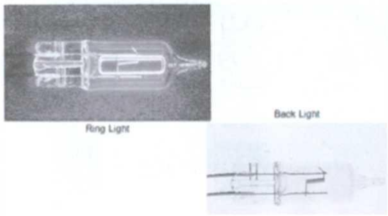


Figure 2.10: Back light application

- *Hints.*-Blacklight must be kept clean and you can collimate the blacklight in order to perform high accuracy gauging operations.

Structured Lighting

It is based on the impact of light patterns on the piece, this patterns are grids, circles points, planes, etc. a representative example is shown in the next figure:

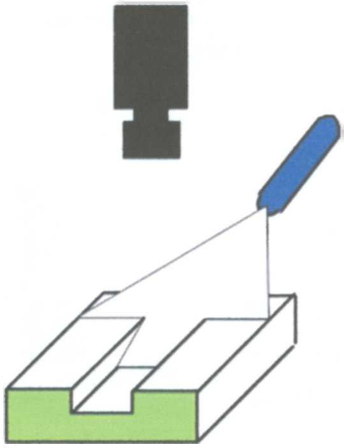


Figure 2.11: Structured lighting

- *Advantages.*- Inexpensive method for measuring height/depth of part also shows surface profile on very low contrast parts.

- *Disadvantages.*-Lasers are expensive and must be handled properly and it is not good for light absorbing surfaces or applications with high accuracy requirements.

- *Applications.*- Gauging continuous features such as steps, gaps edges, etc., Determine part‘s height/depth by observing light‘s deformation on part also is used on very low contrast parts (grey on grey, metal on metal, etc) . Some application examples are shown in the next figure:

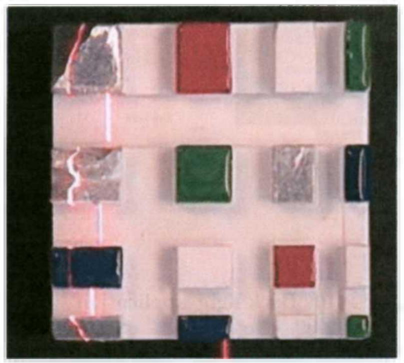


Figure 2.12: Structured lighting application

- *Hints.*-Use a optical fiber line instead a laser when is possible.

Diffuse Light

It is based on the impact of light on the piece but in different distributed directions, this illumination type is used almost in the same situations as the direct light but with soft or bright surfaces, a representative example is shown in the next figure:

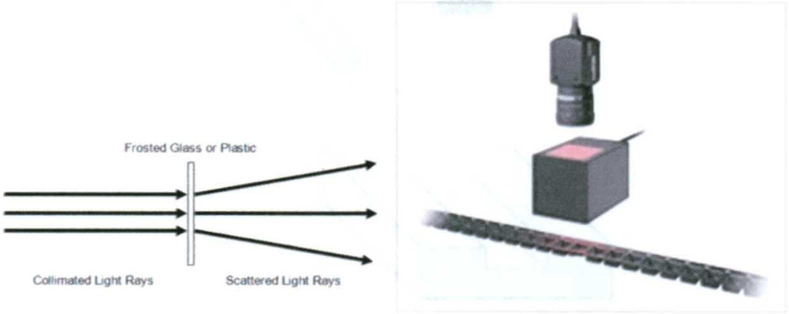


Figure 2.13: Diffuse light

- *Advantages.*- Diffuse light is softer, more even, less likely to cause glare and shadows also diffuse light cover a larger area without creating hot spots.

- *Disadvantages.*-Diffusers lower light intensity, which may be critical when using low intensity sources, such as LED's.

- *Applications.*- Back lighting, on axis lighting, and in-direct lighting of shiny metallic parts. Some application examples are shown in the next figure:

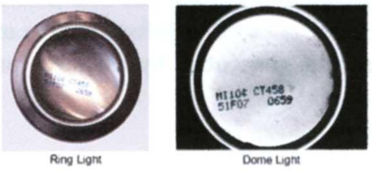


Figure 2.14: Diffuse light application

- *Hints.*-Choose the material based on how much light is lost versus how much light is diffused and Bouncing light off reflective, rough surfaces also diffuse light.

Dark Field

This kind of illumination is used when the object is translucent, by this technique the shape and contour are enhanced, the diffuse surfaces are bright and the flat and the polished surfaces are dark, dark field emphasize height changes in surface inspection and edges detection.

Figure 2.15: Dark field lighting

- *Advantage.*- Shows surface detail on very low contrast parts.
- *Disadvantages.*-Not good for light absorbing surfaces or applications with high accuracy requirements.
- *Applications.*- Very low contrast parts (gray-on-gray, metal-on-metal) and with shiny (specular), metallic parts.

Figure 2.16: Dark light application

Diffuse On-Axis Light

Consists on an uniform and diffuse light generated by a source and leaded to the scene in the same axis that the camera.

Figure 2.17: Diffuse On-Axis lighting

- *Advantages.-* The camera is normal to the object, no perspective distortion. This lighting creates a bright-field effect (specular surfaces are bright, diffuse surfaces are dark).
- *Disadvantages.-* Thickness of mirror can produce a double image.
- *Applications.-* Locating defects or flaws on flat, shiny surfaces. Illuminating the bottom of small cavities.

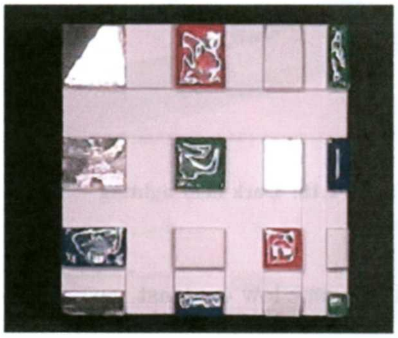


Figure 2.18: Diffuse On-Axis lighting application

- *Hints.-*Use in combination with other source for fill lighting and position mirror at 45 degrees, close to part surface

2.2 Robotics

2.2.1 Introduction

The complementary part of a VRT integration regards to a robot device, the importance of this technology has been remarked in Chapter 1, a robot is defined by the Robot Institute of America as: "A programmable multifunction manipulator designed to move material, parts or specialized devices through variable programmed motions for the performance of a variety of tasks", this is one of the most restricted definition (Fu, 1987 & McKwerrow, 1998)and is the one that is going to be considered in this research, as Lopez (2008) and Fu (1987) show there are a lot of applications kinds, consequently there are a lot of robot types too (mobil, legged, vehicles, arms, etc.), in order to constraint the information presented and in base with the Chapter 1, the robot type that is going to be considered is the robotic arm or manipulator arm.

Robotic arm

Refers to anthropomorphic design of a device which emulate the characteristics of human arm, the two principal elements of this kind of robot are links and joints, its performance is based in the motion of the joints that modifies the links position. According to Groover (2008) the five types of joints are:

1. **Linear.-** in which the relative movement between the input link is translational sliding motion.
2. **Orthogonal.-** which is also a translational sliding motion, but the input and output links are perpendicular to each other during the move.
3. **Rotational.-** which provides rotational relative motion, with the axis of the rotation perpendicular to the axes of the input and output links.
4. **Twisting.-** which also involves rotary motion, but the axis of rotation is parallel to the axes of the two links.
5. **Revolving.-** in which the axis of the input link is parallel to the axis of rotation of the joint, and the axis of the output link is perpendicular to the axis of rotation.

The number of joints will define the degrees of freedom(DOF's), according to the number of DOF's and the type of joint used there a lot of robotic arm configurations (cartesian, polar, SCARA, etc.)

The main components of a robotic arm are:

- Joints
- Links
- Wrists (Links and joints)
- End effectors
- Actuators
- Transmission elements
- Sensors
- Controllers
- User Interfaces

2.2.2 Robot Kinematics

Kinematics of a robotic arm refers to the relationship between the velocities, positions and accelerations of the links of the manipulator. The position of the hand of the robot is probably the most common parameter that every application needs to be developed, in order to compute this position a coordinate frame transformation is used $^{R}T_{H}$, this transformation specifies the orientation and position of the hand in spacewith respect of the base of the robot.

If a coordinate frame is attached to each link, the relationship between two links can be described with a homogenous transformation matrix (**A**). The first **A** matrix relates the first link to the base frame, and the last **A** matrix relates the hand frame to the last link. (McKerrow 1998).

McKerrow (1998) proposes an algorithm to apply direct kinematic(find the location of hand in terms of the angles and displacements of the links) of a robotic arm, a resume of this algorithm is presented:

1. **Move the manipulator to its zero position.**

   Every robot has a zero position where all the joint variables are zero.

2. **Assign a coordinate frame to each link.**

   The author makes reference to Paul (1981) where a coordinate frame assignment is proposed.

3. **Describe the rotations and transitions between joints with link variables.**

   A general **A** matrix can be specified by multiplying the transformations which describe the effect of these parameters. When we relocate from frame $n-1$ to frame n (that is, from joint n to joint $n+1$), the following transformations occur:

   - a rotation about the z_{n-1} axis by the angle between the links (θ_n)
   - a translation along the z_{n-1} axis of the distance between the links (d_n)
   - a translation along the x_n axis (rotated x_{n-1}) of the length of the link (l_n)
   - a rotation about the x_n axis of the twist angle (α_n)

   The last procedure can be represented as:

$$A_n = Rot(z,\theta)Trans(0,0,d)Trans(l,0,0)Rot(x,\alpha) \tag{2.13}$$

$$= \begin{pmatrix} cos(\theta) & -sin(\theta) & 0 & 0 \\ sin(\theta) & cos(\theta) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & l \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\theta) & -sin(\theta) & 0 \\ 0 & sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} cos(\theta) & -sin(\theta)cos(\alpha) & sin(\theta)sin(\alpha) & lcos(\theta) \\ sin(\theta & cos(\theta)cos(\alpha) & -cos(\theta)sin(\alpha) & lsin(\theta) \\ 0 & sin(\alpha) & cos(\alpha) & d \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.14}$$

4. **Define the A matrices relating the links**
   The A matrix for a specific link is found by substituting the link parameters into last equation.

5. **Multiply the A matrices to calculate the manipulator transformation matrix $^{R}T_H$**

6. **Equate the manipulator transformation matrix and the general transformation matrix to obtain Cartesian coordinates in terms of joint coordinates**

$$^{R}T_H = \begin{pmatrix} x_x & y_x & z_x & p_x \\ x_y & y_y & z_y & p_x \\ x_z & y_z & z_z & p_x \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.15}$$

7. **Equate the manipulator transformation matrix and the general orientation matrix to obtain the hand orientation angles**

$$^{R}T_{H} = \begin{pmatrix} x_x & y_x & z_x & p_x \\ x_y & y_y & z_y & p_x \\ x_z & y_z & z_z & p_x \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} (RPY(\phi, \theta, \gamma)) \tag{2.16}$$

2.2.3 Robot teaching

At application level, one of the most important topic about robots is the robot teaching or programming, most of the time the robots should be taken to off-line status in order to modify its program. The robot is taught to interact with its environment in show-and-teach mode and expected to operate subsequently in playback mode, a list of conditions to avoid the high rate of time consumed in teaching process is presented next(Benhabib,2003):

- "The kinematic model of the robot describing the mobility of its end effector with respect to its base must be known"
- "The locations of all devices must be accurately defined with respect to a fixed world coordinate frame and must not vary during the interactions of the robot end-effector with its environment"
- "The motion controller of the industrial robot must allow for off-line programming"

As it can be noticed generally one or two of this considerations is missing, so in industry the robot programming lays on show-and-teach mode, where the task of the robot is taught by locating the end-effector of the robot in space points where it must be, this is manually made trough the use of a teach pendant. The teaching of certain points in space should be sequential and the memorization of the track compound by determined points is expected, so that the robot will repeat this task when asked for it. In base with Benhabib(2003) a typical teach pendant would allow the programmer to move each joint individually or move the end effector along individual Cartesian axis in the end effector or in the world-frame coordinates in order to position the robot end-effector at a desired location.

Chapter 3

Vision Robot Technologies Integration

3.1 Integrated VRT systems

As the name indicates, the Integrated VRT systems are compound by two systems, the vision and the robotic manipulator system. There is a third component, a PC, that can or can not be integrated to the system depending on the complexity level of the application and on the tasks that will be performed, in the development of this research a PC is considered as an integrator of the VRT, the next picture shows the generic architecture of a physical integration of this two systems:

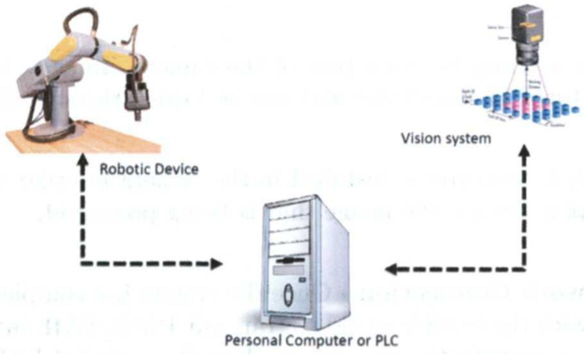


Figure 3.1: VRT integration architecture

3.1.1 Components

The next figure shows the main components of a Integrated VRT system, several components can be included in this figure but their function is basically the same.

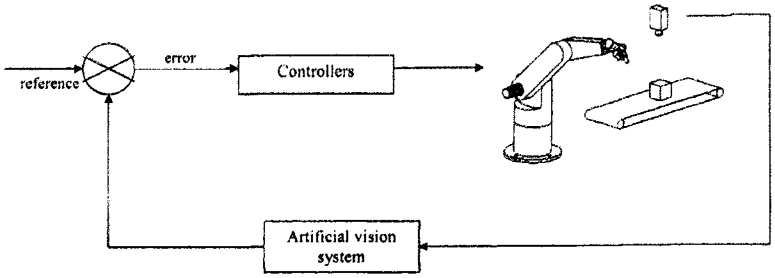


Figure 3.2: Components of a vision-robot system

According to Cuevas (2005) the elements of the VRT system can be classified in the next form:

- **Cameras, Optics and Lighting.-** The most common cameras available on the market are the ones based on charged coupled devices (CCD), charge injection devices (CID), or silicon bipolar sensors. All the cameras need lenses in order to focus the object and/or an illumination unit. The light intensities sensed are a result of the light from the illumination unit reflected by the object. The most common forms of lighting are: Halogen,Incandescent, Fluorescent, Laser, Xenon, etc.

- **A/D converter.-** The matrix of fotosensors is scanned and the light intensities in the form of a continuous analog video signal must be sampled and converted into a digital signal. The A/D converter is part of the digital video camera

- **Image processor.-** The image can be processed for image data reduction, segmentation, feature extraction and/or object recognition. The processor is part of the camera and it is generally a DOS program preinstalled bye the manufacturer. Inspection test are configured through the software installed in the PC and uploaded to the camera.

- **Image memory.-** The memory is also a part of the camera. Images and/or inspection programs are temporally stored in this device and can be loaded through the camera software.

- **D/A converter.-** A D/A converter is installed in the camera in order to visualize in a VGA monitor or another display device, the image that is being processed.

- **Digital I/O and Network Connection.-** Once the system has completed its inspection of a part, it communicates with the outside world to transmit PASS/FAIL information, or another data needed to control a manufacturing process. Usually a digital I/O interface board and a network card build up this interface. Commercially available systems have communication abilities through Profibus,RS 232, Device Net, Ethernet, etc.

- **Inspection Software.-** Machine vision software is used for creating and executing programs, processing incoming image data, and making PASS/FAIL decisions, This software is developed in many different forms (C libraries, ActiveX controls, Point and Click programming environments, etc.) and it can be single or multi function.

- **Robot controller.-** The robot controller is in charge of the task and paths planning, task execution, communication with sensors and generation of reference points.

3.1.2 Camera placing

As it has been mentioned, the use of a camera in an integrated VRT system is inherent, there are two major mounting configurations:

- The first one called by Torres *et. al.* (2002) as eye in hand belongs to the mounting configuration where the camera is placed in the end of the robotic arm, as it is shown in the next figures:

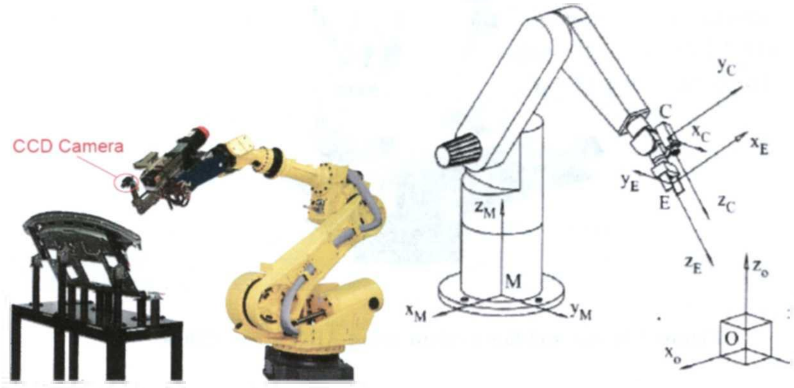


Figure 3.3: Eye in hand configuration

- The second configuration considers the placement of the camera in an external location of the robot, or in predefined spots on the system, the next figures shows systems using this kind of camera placing:

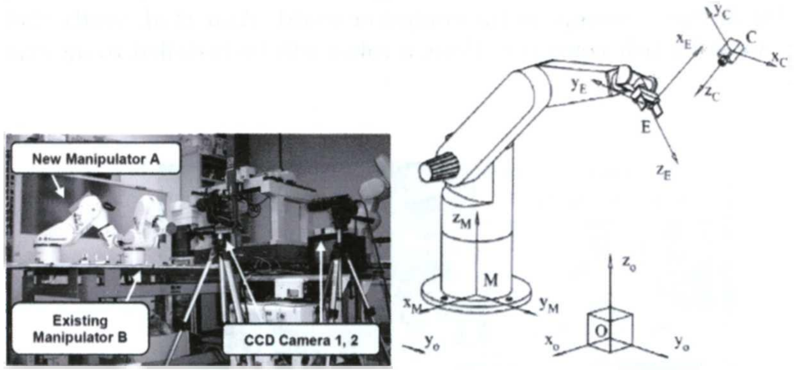


Figure 3.4: External configuration

3.1.3 VRT systems examples

Autonomous Visual Measurement for Accurate Setting of Workpieces in Robotic Cells

Watanabe *et. al.* (2005), present a paper which introduces a new method of adapting the virtual world of an offline programming model to an actual robotic cell by attaching a CCD camera to the robot. This method requires no specific camera attachment location or optical camera calibration. Furthermore, there is no operational requirement for setting robotic camera location. Robot motion is autonomously generated to identify the camera view line. The view line is adjusted to pass through the designated target point, utilizing visual feedback motion control. This method calibrates reference points between the virtual world of an offline model to an actual robotic cell.

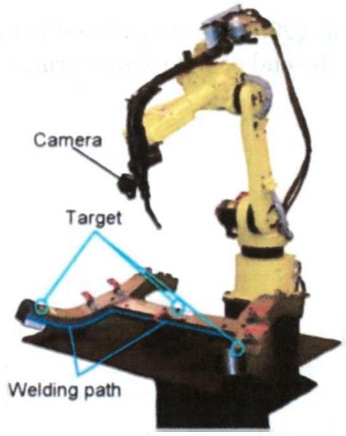


Figure 3.5: Arc welding system using ARC Mate 120iBe

The devices that were integrated to compound the RVT system are: 1) a FANUC robot R-2000/200FO, 2)A spot welding tool as the end of arm tool, 3) An OPTEON USB camera (with 1/3 CCD of 640 x 480 pixels) and a 12 mms lens and 4) a 20 mm diameter target.

Assembly System with "Plug & Produce capabilities"

Arai *et al.*(2000) propose a Holonic Assembly System with "Plug & Produce" capabilities. This is a methodology to introduce a new unit or device into an assembly cell easily and quickly. It is designed by analogy of the "Plug & Play" concept in the computer world. Arai *et al.* verify their concept by experiments with robots and a belt conveyor. Here, a robot will be installed to an existing cell in a short time.

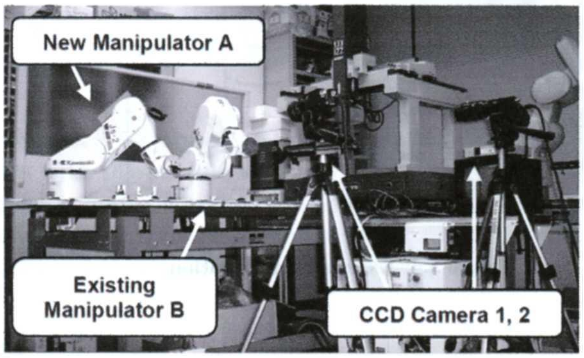


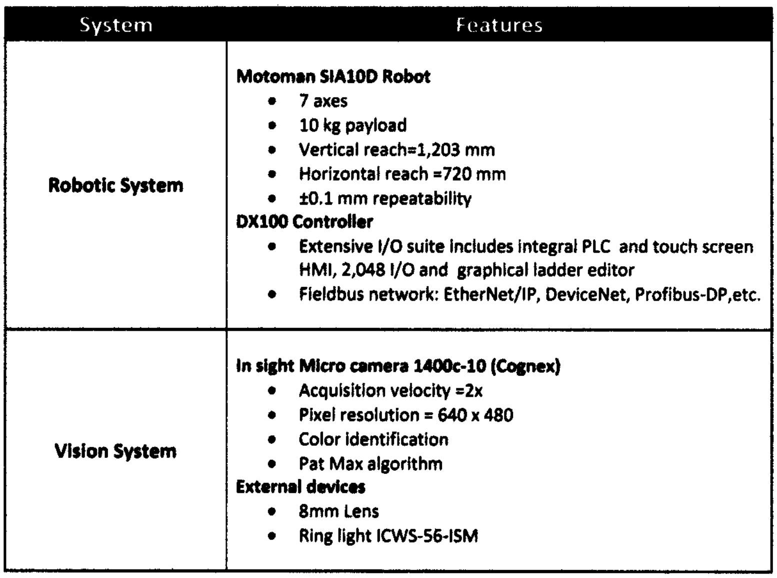
Figure 3.6: Automated calibration system for a "Plug & Produce" assembly cell

When a new device, e.g. a robot is installed into the assembly cell, calibration should be made. Arai et al. propose an automated calibration system of relative position/orientation based on the Direct Linear Transformation method using two CCD cameras. The cameras are freely positioned, and then a set of motions is commanded to each manipulator. By detecting the motion with the cameras, the relative position of the two robots is obtained.

Vision Robot assistance in a Manufacturing cell

This experimental example realized by ITESM in campus Monterrey (Hernandez et. al. 2010) includes a robot and a vision system in the manufacturing cell operation, the robotic arm is developed by MOTOMAN (SIA10D), and the vision system is compound by a In-sight Micro Camera from Cognex, the most important features of these devices are shown in the next chart:

Table 3.1: Features description of the Vision and Robotic devices



System	Features
Robotic System	**Motoman SIA10D Robot** - 7 axes - 10 kg payload - Vertical reach=1,203 mm - Horizontal reach =720 mm - ±0.1 mm repeatability **DX100 Controller** - Extensive I/O suite includes integral PLC and touch screen HMI, 2,048 I/O and graphical ladder editor - Fieldbus network: EtherNet/IP, DeviceNet, Profibus-DP,etc.
Vision System	**In sight Micro camera 1400c-10 (Cognex)** - Acquisition velocity =2x - Pixel resolution = 640 x 480 - Color identification - Pat Max algorithm **External devices** - 8mm Lens - Ring light ICWS-56-ISM

This integration consider an "eye in hand" and interchangeable configuration, where the vision system is the interchangeable device between the Inspection and the Assembly station.

The purpose of having this flexibility is to use the vision system in two different jobs, the first one corresponds to the inspection of the finished good produced in the cell, and the second job consists on the assistance for the SIA10D robot in the assembly process. This assistance delivers to the control unit of the cell the information in order to start or not the assembly routine, by this performance the assembly process is more robust, due to the identification of each of the components to be assembled (improving the fixture and proximity sensors configuration).

The interchangeability is provided by mechanical and electric connectors which activate the vision system depending on the station where it is, this configuration can be represented as in the next figure, the mechanical and electrical adaptors performs as "switches S1 and S2", the lighting system is fixed to the camera casing providing illumination in both stations.

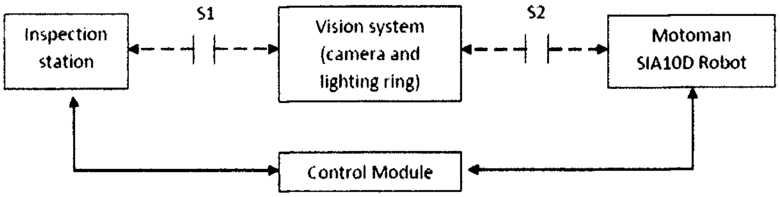


Figure 3.7: Configuration of the VRT system

3.2 Integration of VRT methodologies

3.2.1 Methodologies examples

In this research it will be consider as a background the previous research about this integration applied for the Manufacturing Cells, next it is presented the description of the methodologies that have been successfully implemented and their results.

FROVIS:A Flexible Robot Vision System

FROVIS was developed by Raquel Cuevas (2005), this research consists on the development of a Flexible Robot Vision System comprised by a vision sensor mounted on a robot end effector, Physical and logical methods are proposed for integrating robot and camera systems. The integration architecture proposed allows for different robot controllers and vision systems to work together making the system modular, flexible and generic.

In this methodology the VRT integration is a PC based integration, so talking about communication between the main components, the PC performs as a client and both the camera and the robot controllers are considered as servers, the job of these servers is to listen for a connection, while the job of the PC is to to try to make a connection with both servers, the communication architecture proposed by Cuevas (2005) is shown in the next figure:

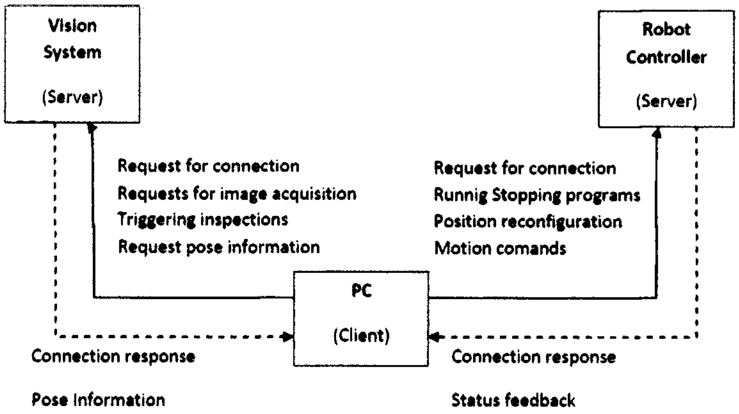


Figure 3.8: Communication architecture for FROVIS development

Cuevas (2005) establish certain generic tasks for the camera an robot that must be performed in order to conform a Flexible Robot Vision System, for the robot the mandatory functions are: Connect,Go Home, Run Program, Update Position, Move, Grab and Disconnect, and for the camera the functions are: Connect, Acquire image, Get Position and Orientation, Get Height and Disconnect.

In order to achieve the correct communication between the robot and the camera, it is proposed a communication sequence and a communication protocol, in the next figure is shown the sequence diagram for the Control Module:

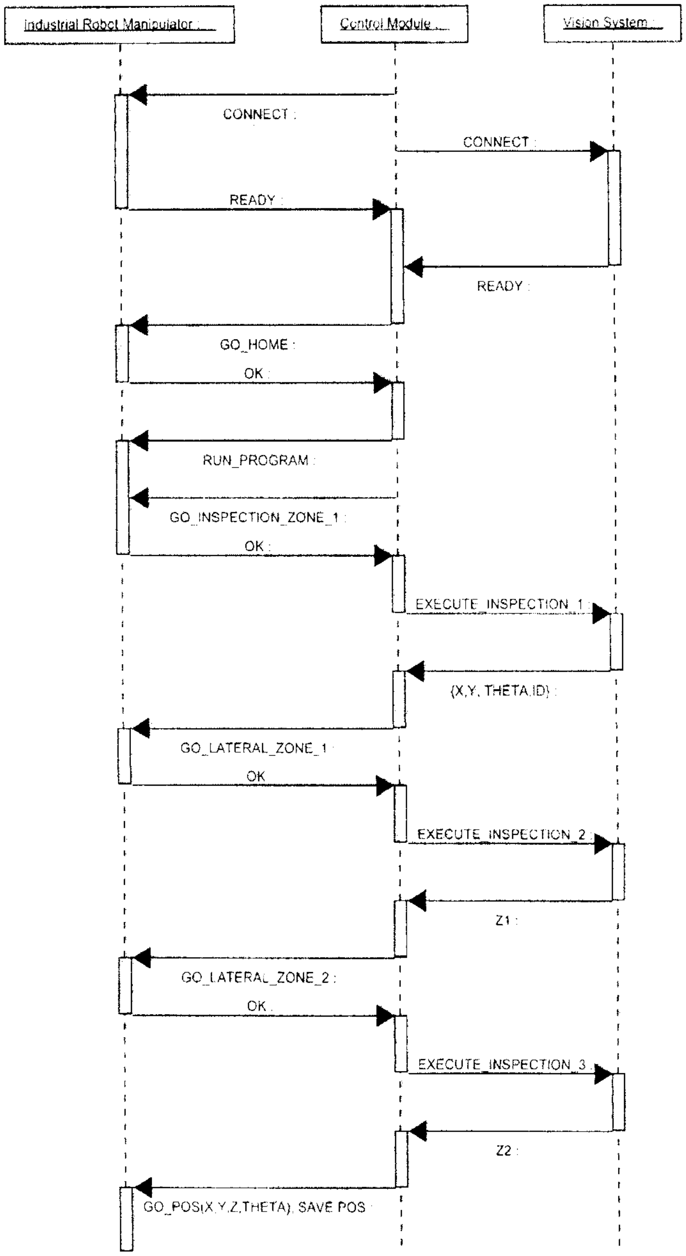


Figure 3.9: Communication sequence

The communication protocol consists mainly on the exchange of strings with defined lengths between the Control Module and the Vision System. The communication protocol dos not apply to the link between the PC and the robot controller because controllers have their own specific communication protocols which can not be changed.

The FROVIS was applied in two study cases obtaining good results specially in the synergy of the integration, in the first experiment it was performed a Integration of an ABB controller and a Simatic Vision Sensor, and in the second experiment a Integration of Mitsubishi controller and a Simatic Vision Sensor was realized.

Methodology for the Vision Robot Systems Integration

Espinosa (2006) presents a research about the integration of two systems, both systems are compound by a robot and a vision system the difference between them is the application of the integration, in the first one associated with FROVIS development, the camera is used by the robot as a sensor which provide information about the environment of the robot, it can be said that the camera is the slave device and the robot is the master device, in the second one called FROVIS II the camera utilizes the robot in order to acquire the best location for the inspection routine, in this case the camera is the master device and the robot performs as a slave device.

A integrator system was developed, the main function of this system is to provide the option of choosing between all integration systems available, in this study there were used 2 integration systems (mentioned above), FROVIS and FROVIS II. Once that the system has been chosen the communication between the robot and the camera will be present.

A communication protocol was developed in order to send and receive the information in a organized and structured form,this protocol communicates the integrator system with FROVIS and FROVIS II, in the next figure is shown an example of the message structure proposed by Espinosa (2006) in order to establish the wanted communication.

Byte	Encabezado	Descripción
0	I	Envía si la pieza fue aceptada o rechazada.
1	P	
2	A	
3	S	
4	S	
5	F	
6	A	
7	L	
8	>	Separador
9	P ó F	"PASS" o "FAIL"
10	A ó A	
11	S ó I	
12	S ó L	
13	CR	Delimitador 1
14	LF	Delimitador 2

Figure 3.10: Message structure for communication protocol establishment

The integrator system has the capability to:

- Select the execution order of the integration systems (FROVIS and FROVIS II.
- Select the program that FROVIS or FROVIS II will execute.
- Configure a execution depending on the result of a previous program execution.

The execution logic of the systems described above is summarized in the next flow diagram:

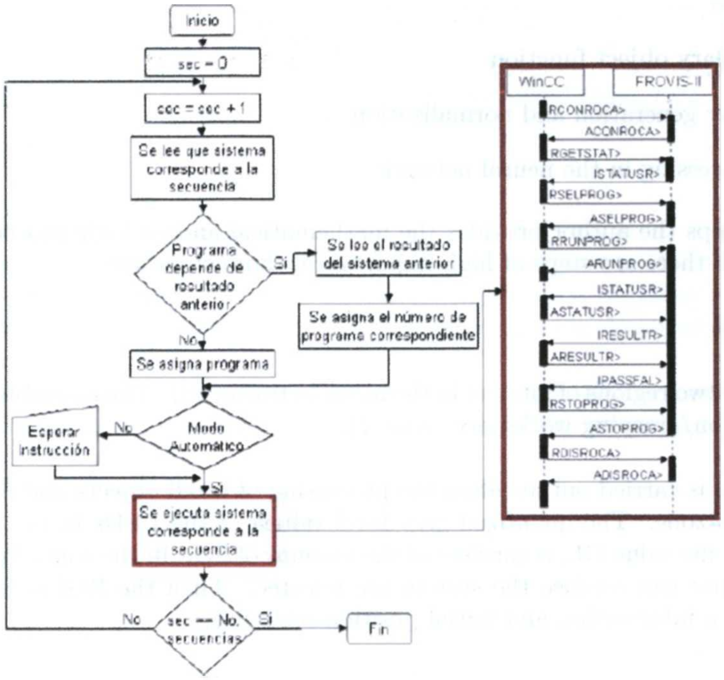


Figure 3.11: Flow diagram of the execution logic

Machine vision approach for robotic assembly

Cabrera (2005) presents a methodology to integrate a robot arm with a camera in a manufacturing cell, this methodology consider two major topics the system integration and the object recognition methodology, the general methodology will be described next.

For the System integration the author propose a serial communication between the camera and the robotic assembly module and identifies three principal commands to be sent between the robot and the camera:

- Send information of zone 1, where the zone 1 is the place where the robot grasps the male components.
- Send information of zone 2, where the zone 2 is the place where the robot is preforming the assembly task.
- Resend information of zone x, where the command is sent when there is an error, or the information sent by the camera is incorrect.

The communication protocol presented by the author is shown in the next figure:

For The object recognition methodology, the author proposes 8 steps:

- Finding the region of interest
- Calculate the histogram of the image
- Search for pieces
- Centroid calculation

- Piece orientation
- Calculate boundary object function
- Descriptor vector generation and normalization
- Information processing in the neural network

For each of this steps the author provides the mathematical and/or logic procedure, for the objective of this research there are steps of high importance, those steps are:

Finding the ROI

The author defines two regions of interest in the manufacturing cell. The assembly workspace(zone 1) and the identification/grasping workspace (zone 2).

Image conditioning is carried out avoiding the processing of small objects and finding the initial position of the desires zone. The quantized grey level valuea of the LEDs in the image is greater than or equal to a specific value GL, regardless of the amount of light in the zone. With this process, most of the objects that can confuse the system are rejected. Then the ROI is first extracted by using the 2D histogram information and initial position reference.

Search for pieces

The pieces search is realized by the perimeter calculation obtaining:

- Number of points around the piece
- Group of points coordinates X and Y, corresponding to the perimeter of the measured clockwise
- Boundaries of the piece 2D Bouncing Box

The perimeter calculation for every piece in the ROI is performed after the binarization. Search is always accomplished from left to right and from top to bottom. Once a pixel is found, all the perimeter is calculated with a search function. The author provides the next definitions to understand the algorithm:

- **Nearer pixel to the boundary.-** is any pixel surrounded mostly by black pixels in connectivity eight.
- **Farther pixel to the boundary.-**is any pixel that is not surrounded by black pixels in connectivity eight.
- **Highest and lowest coordinates.-** The boundaries which create the rectangle (Boundary Box)

The search algorithm proposed is:

1. Searches for the nearer pixel to the boundary that has not been already located
2. Assigns the label of actual pixel to the nearer pixel to the boundary recently found
3. Paints the last pixel as a visited pixel
4. If the new coordinates are higher than the last higher coordinates, it is assigned the new values to the higher coordinates

5. If the new coordinates are lower than the last lower coordinates, it is assigned the new values to the lower coordinates.

6. Steps 1-5 are repeated until the procedure returns to the initial point, or no other nearer pixel to the boundary is found

Methodic Design of Robot Vision Systems

Sitte (2007) presents a research following the Demand Compliant Design (DeCoDe) method developed by the authors, this method is based on the design information organization through simple ideas, then the identification and evaluation of relations between main design elements is proposed.

This research considers three different but related views of the VRT integration; the functional, the structural and the process view. Each view consists of the corresponding hierarchical list of functions, processes or components.

"Connectivity matrices capture relations between pairs of catalogues and with themselves", as shown in the next Figure:

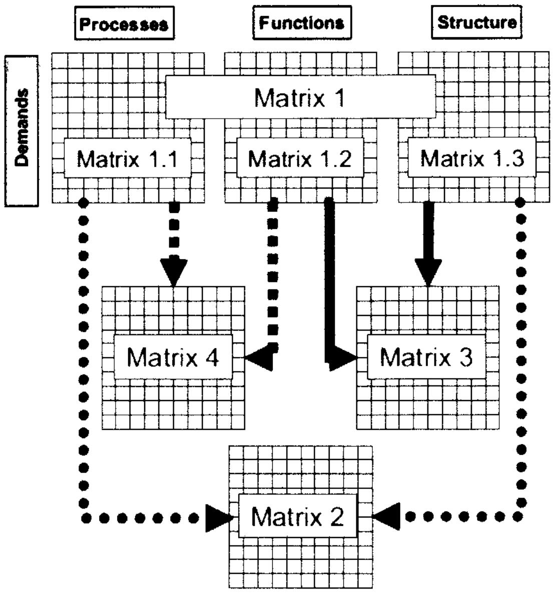


Figure 3.12: Diagram of the DeCoDe method

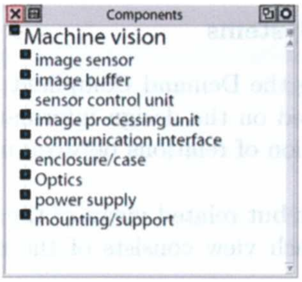
The authors propose to evaluate the relationship between two elements not only by binary factors (block diagram as a result), but with a rate that represents the strength of the interaction on a determined scale. The matrix now captures more information than the block diagram.

The matrix that relates the demands to the components can capture some essential information, each row corresponds to a demand and each column to a component. An element of this matrix can represent an estimate of how much a component contributes to the satisfaction of a demand.

As it has been said before this method consider 3 views of the design, the author describes these views in base with possible demands on a VRT system, these possible demands can be roughly divided into three groups: (1) functional, manufacturable, (2)cost and (3)regulatory and standards.

Structural view

This view is compound by the components that make up a device, this components perform a certain activity simultaneously with all other in order to make a device work. Sitte (2007) presents a list of components of a generic vision system:

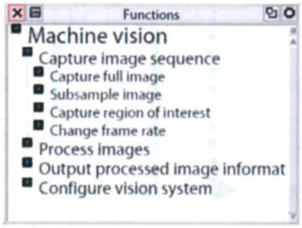


- Machine vision
  - image sensor
  - image buffer
  - sensor control unit
  - image processing unit
  - communication interface
  - enclosure/case
  - Optics
  - power supply
  - mounting/support

Figure 3.13: Components of a generic vision system

Functional view

This view consider the way to perform certain task known as function of the device, there a lot of ways to accomplish certain activity, the functions are closely related to the demands, but there is not always a one to one correspondence. Sitte (2007) presents the functions of a generic vision system:



- Machine vision
  - Capture image sequence
    - Capture full image
    - Subsample image
    - Capture region of interest
    - Change frame rate
  - Process images
  - Output processed image informat
  - Configure vision system

Figure 3.14: Functions of a generic vision system

Processes view

In order to perform a function one ore more components of the device need to carry out one or more processes. In a process the components undergo a dynamic interaction. The next figure shows Top level processes in a vision system presented by Sitte (2007).

- Machine vision
  - Design
  - Manufacture
  - Deployment
  - Configuration
  - Programming
  - Testing
  - Disposal

Figure 3.15: Processes in a vision system

Chapter 4

Methodology for VRT systems integration

The methodology that is going to be developed, includes the most important steps in order to establish the integration of VRT, each of the steps or points to be realized are analyzed and described in this section of the thesis, the figure 4.1 presents the concept diagram that shows the scope of the methodology:

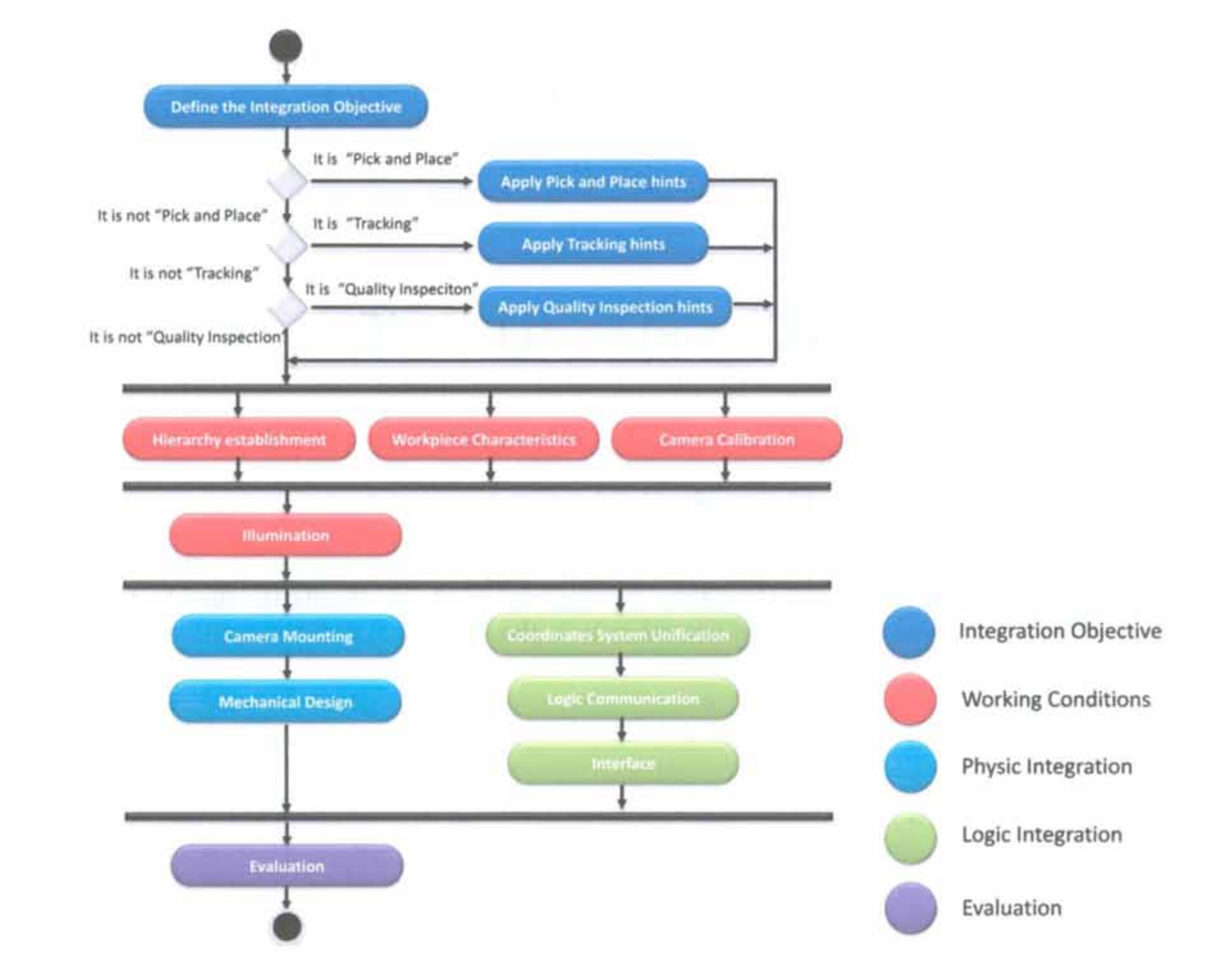


Figure 4.1: Methodology for VRT integration

From now on, each page of the Chapter will include at the beginning a logo indicating which section of the methodology is being explained, in the next figure it is shown the logo and the code to interpret it:

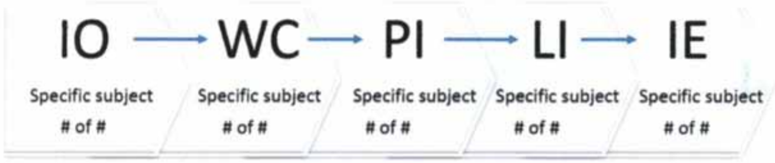


Figure 4.2: Guide logo

The sections of the methodology are shown in the figures 4.1 and 4.2 and their logo codes are:

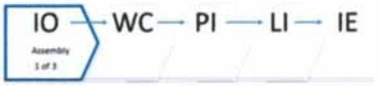
- IO means Integration Objective
- WC means Working Conditions
- IP means Integration (Physic)
- IS means Integration (Logic)
- IE means Integration Evaluation

The section box of the methodology being explained will be remarked, also if there is, the specific subject of the section will be defined in the specific subject field, if there are more than one specific subject, the number of the current subject in reference with the total subjects is represented in the # of # field.

For example:

Figure 4.3: Example of logo interpretation

Indicates that the section of Working Conditions is being explained, specifically the Illumination subject, this subject is the third of a total of five subjects in the Working Conditions section.



4.1 Integration objective

There are many application that involves the integration of VRT, the three main application according to the analysis presented in the section 1.6, are pick and place, path tracking and inspection,

4.1.1 Pick and Place

For Pick and Place applications (assembly, packaging and material handling) the next methodology is suggested:

1. Workpiece Detection (Reference points utilization)

   The Reference Points (RP) identification is one of the principal task and it's realization could define the success of the integration, both, the object to be assembled and the matching fixture or component should contain at least one RP, the RP should have the following characteristics:

   (a) The RP should be a part of a feature of interest

   (b) The RP should be visible for every robot position available (for eye-in-hand and fixed configuration, see Section 3.2.1)

   (c) If stereo vision is used the RP and feature of interest should be visible throughout all the process and by all the cameras.

   As is a stereo image, you should have more than 2 images corresponding to an object, due to perspective issues you should have more than one image for each RP, so that the tangents of the RP will correspond and the position could be obtained. This perspective problem can be observed in the next images presented by Pauli (2001):

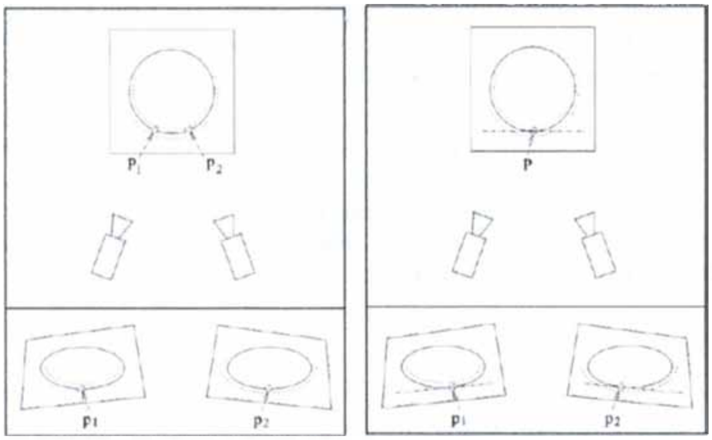
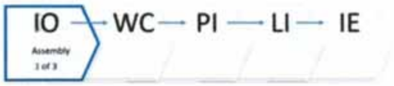


Figure 4.4: On the left: extraction image points p1 and p2 originate from different scene points; on the right extracted image points are corresponding

   In the left figure is shown a configuration with two RP and two cameras, as it can be observed the tangent of the RP can not be obtained because there is only one image per RP available. In the right image an adequate configuration is shown, where there is one RP and it is visible for both cameras, so that there are two stereo images available for the RP, in this configuration the tangents of the RP do correspond.



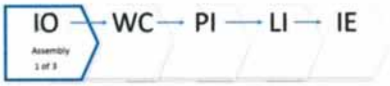
2. Workpiece identification
   As Bone (2003) and Conolly (2009) show, there are a lot of commercial algorithms which can be trained to detect certain patterns belonging to a specific object, in this part of the methodology of integration the following hints are proposed:

   - If the Object Detector Tool bases it's performance on the use of edges contours in the image, use several edge contours to increase the reliability and accuracy of the system.
   - If a family of pieces is considered, and the variation between pieces model allow it, determine a ROI (Region Of Interest) that is suitable for as many as models possible.
   - If a family of pieces is considered, and the variation of the between pieces remains only in the scale of the pieces, adjust the Object Detector Tool in order to disable the size sensitivity(the level of reduction of this sensitivity is determined by quality and model specifications), so you would be able to use just one pattern to detect all pieces of the family.
   - When identifying the region which will be the pattern to look for, try to include critic quality characteristics (number of holes, size of structural details, etc.) in order to apply the first quality inspection of the object through the object detection.
   - If possible identify the most common quality defects, and when identifying the region of interest avoid using patterns that even in "not good" pieces could pass.

3. Picking workpiece up

   - *Disposition of the object*
     A roughly classification of workpiece grabbing consider two options; when the object is in motion and when it is not. When the object is static the issue remains in the adequate use of the information provided by the previous steps (Workpiece identification, Pose acquiring and Reference points localization).

     In the other hand if the object is moving the workpiece picking up requires not only the pattern recognition, pose measurement and localization information, another variables of the system should be considered, usually this information corresponds to the velocity and acceleration behavior of the device which moves the object (usually a belt or conveyor), Sang Shin (2009) identifies three modes for the workpiece picking up:



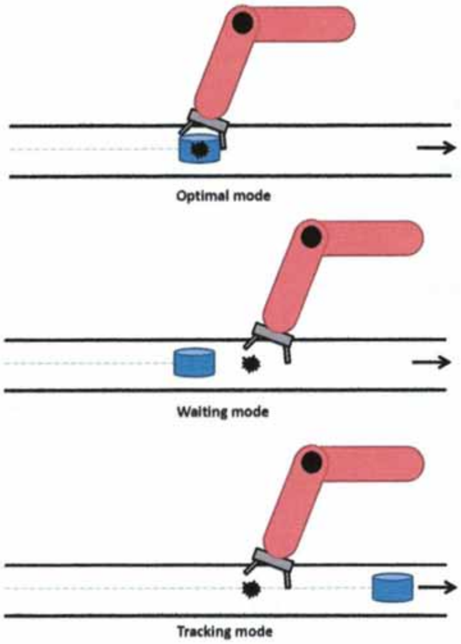


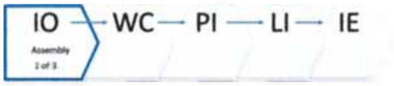
Figure 4.5: Interception modes on the conveyor belt

It is important to remark that the need of optimal scenario implementation, it is not always very significant in comparison with other two, let's consider two workstations, the first one delivers the object of interest to other station where a robotic arm is used. If the cycle time of the first workstation is bigger than the cycle time of the second one, the workpiece picking up mode could be waiting or optimal with the same result (downtime due this material handling task) in both scenarios.

The user should define or identify the scenario present on his application, and if needed, a dynamic analysis between the motion of the object and the motion of the robot should be done, as it is shown in the research of Sang Shin (2009) where he propose an algorithm to determine the minimum-time interception point on a conveyor belt to grab a moving object.

Another important issue in this part of the robotic assembly is the definition of the pick-up point (which could be or not the reference point), this point should be, if the part geometry allows it, as near to the gravity centroid of the part as possible, considering this, most of the times the picking-up action will not modify the piece's rotation about x and y axis, avoiding reorientation after the piece is grasped.

- *End-effectors*
  The shape, surface and size of the object to be grasped will determinate the type of end effector to be used, according to Guedea (2010) and Groover (2008) the most common end effectors are:



(a) Grippers
  - Mechanical gripper
  - Vacuum
  - Adhesive Tongs
  - TailHook
  - Magnetic Tongs

(b) Tools
  - Arc-welding
  - Water-jet cutting
  - Spray coating
  - Laser cutting

The most common end effectors used in industry are grippers and vacuums, the main formulas and parameters are:

For grippers:

$$\mu n_f F_g = wg \tag{4.1}$$

where:

μ = Friction coefficient between the object and the gripper fingers
n_f = Number of touching fingers
F_g = Force provided by each finger of the gripper
w = Object's weight
g = Adjust factor
$g = 1$: When the acceleration of the motion is against the gravity
$g = 2$: When the motion is horizontal
$g = 3$: When the motion is towards the gravity

For Vacuums: the lifting capability of a vacuum actuator depends on the effective area and the pressure between the vacuum and the object. The next formula represent this relationship:

$$F = P * A \tag{4.2}$$

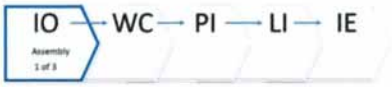
where:

F = Lifting force
P = Air pressure
A = Suction area

4. Workpiece assembly

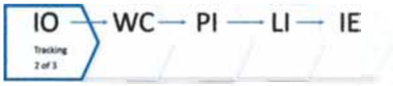
When the integration objective considered refers to assembly operations, and as Boothroyd (2002) presents, we should include general rules or hints for assembly process and some specific hints for robotic assembly, the next hints are referred to assembly process in general:

- Reduce the number of parts to be assembled as much as possible
- Include features such as leads, lips, chamfers, etc., to make parts self-aligning in assembly



The next suggestions are directed to assembly process by the use of robotic arms:

- The parts should be designed in certain shape so that the minimum number of tool(gripper) changes is propitiated. The use of screw fasteners always include one gripper change due to the spin capacity (1 revolution) of the typical robot wrists.
- Consider a layered assembly method, if it is possible the layers should be orthogonal to z-axis, with this the assembly operation can be realized by a simpler(cheaper) robot than in other direction.
- Avoid the need for reorienting the partial assembly or for manipulating previously assembled parts.
- If parts are automatically feeded, then ensure that they can be oriented using simple tooling
- If parts are presented in magazines, make sure that they have a stable resting aspect from which they can be gripped and inserted without any manipulation by the robot.

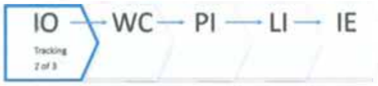


4.1.2 Path Tracking

According to Misel Brezak *et. al.* (2007) if the path tracking is external (when the camera tracks the robot motion by acquiring the image of the robot and linking it with the environment) there are three main problems in the robot navigation: robot localization, map building, and robot motion planning.

In order to solve this problems the following hints are presented(Misel Brezak *et. al.*, 2007):

- To realize the robot localization two tasks are required: the robot detection and the robot pose measuring, although these tasks can be performed by the use of one mark in the robot body, it is recommended to implement a robot localization by two marks one for each task.
- Regarding to the robot detection it is recommended to use color patches as detection cues, which provide robustness under rotation, scale and resolution changes , and the main advantage the processing speed.
- If the application is supposed to track more than one robot,and also identify them, different colors can be used just as the number of colors that can be recognized by the camera. The disadvantage in this is that the system has to be calibrated for each color, and this would result in reduced robustness to light intensity changes and would also require long set up times.
- If the application has to track and identify more than one robot and the number of robots to track is relatively small (2 or 3) the color patch detection method is recommended. On the other hand when the robots to track and identify are 4 or more, another detection method is recommended: this method consists on the execution of predefined routines for each robot, those routines have to be specific for each robot and designed preferably in the way to make impossible to realize it by a non corresponding robot. In this form the robot identification can be accomplished.
- In order to reach unambiguity of robot mark detection, it is convenient to place a key patch in the middle of the color mark, and other color patches around it so that their distances to the key patch are identical. In this way, the key patch is first located in the image and after that other belonging color patches are located on a circle around the key patch with radius equal to the given distance, where it is guaranteed that no color patches of other robots can appear, but only patches that belong to the related robot, which significantly simplifies the color mark detection process. Another criterion is that color patches must be distributed so that the robot orientation can be unambiguously determined once locations of all individual patches are known.
- The patch colors can be selected in a manner that detection reliability is maximized, so that only saturated colors are used, and distance between particular colors on the hue scale is maximized, or alternatively colors that are less likely to appear in the image background can be used.
- Concerning the shape of the color patch, in order to maximize color recognition reliability, the square color patch shape which provides maximum space utilization can be selected. The next figure shows some examples of possible detection mark designs, where black color is used for the background and blue color for the key patch. Of course many other designs are possible.



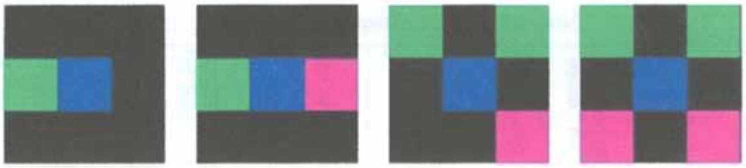
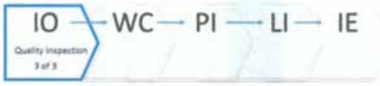


Figure 4.6: Possible designs of the robot detection mark



4.1.3 Quality inspection

Visual inspections most of the times consider the analysis of images which represent assemblies or finished products, the analysis objective is to obtain information to make(generally) yes/no decisions depending on the pre-specified requirements accomplishment.

Wallace (1982) identifies two major phases of the visual inspection, the first one consists on the features of interest “teaching” and the second one is the “run” phase, where the object is analyzed.

Malamas (2003) propose a classification of the visual inspection of products, depending on the features to be checked, this classification considers:

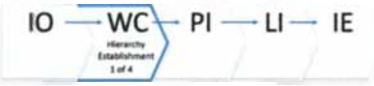
1. Inspection of dimensional quality
2. Inspection of surface quality
3. Inspection of correct assembling (structural quality)

Table 4.1: Quality Inspection Hints

Inspection type	Potential Features to be inspected	Hints
Dimensional	• Dimensions • Shape • Positioning • Orientation • Alignment • Roundness • Corners	• In order to achieve complete inspection, it is often necessary to provide manipulation in three dimensions, by ***moving*** either the object or the camera. • In Printed Circuits Boards (PCB) Current research has shown that ***histogram-based techniques*** perform better than two and three-dimensional feature-based techniques • An interesting application in this category deals with the inspection of screw threads for compliance with manufacturing standards. Edge detection algorithms (***based on linear interpolation to the sub-pixel resolution***) are applied to detect region of interest (ROI). Each such region is matched with multiple models of threads, since the dimensions and positions of the inspected threads are allowed to vary. • Shin(2002) propose an ***imaging mechanism*** (4 Degrees of Freedom) to measure certain dimensions of interest of a ball stud, in his methodology Shin uses an algorithm to determine the optimal Threshold on order to get an ideal histogram of the image.
Surface	• Pitch • Scratches • Cracks • Wear • Finish • Roughness • Texture • Seams-folds-laps Continuity	• In the assessment of surface roughness of machined parts ***Fourier transform*** is applied first for the extraction of roughness measures. Then, NNs (Neural Networks) are employed for the classification of surfaces based on roughness. • Surface inspection is also applied to the aluminum strip casting process. ***Infrared (IR) temperature measurements*** (providing a measure of the distribution of surface temperature) are used to evaluate the quality of aluminum sheets. • For the inspection of defects on objects with directionally textured surfaces (e.g. natural wood, machined surfaces and textile fabrics) researches present a global image restoration scheme based on Fourier transform. ***High frequency Fourier components*** corresponding to line patterns are discriminated from ***low-frequency*** ones corresponding to defective regions

Table 4.2: Quality Inspection Hints (cont.)

Inspection type	Potential Features to be inspected	Hints
Structural	Assembly: • Holes • Slots • Rivets • Screws • Clamps Foreign objects: • Durst • Bur • swam	The use of CAVIS (Computer Aided Visual Inspection System) reduces implementation cost and time, the designed system integrates several modules as: **Product Data Base**.- Should include: • Lighting source type • Position and intensity • Product color • Product Material • Product surface • Effects of lubricants **Layout Design Module**.-elements considered: • Number type and position of the cameras and light sources • Sensor features • Lens focal length **CAD Modeller.-** Generate solid models of the components **Decision Supporting Module.-** structured as a rule-based system, helps the user in selecting the best viewing and lighting conditions and the proper algorithms. **Algorithm DataBase.-** contains a large database of algorithms from developed applications **Programming Module**.- The user can build the complete image analysis software upon the basis of the suggested algorithms or its personal experience. **On-line Test.-** In order to select or modify the current algorithm.



4.2 Working conditions

In this section of the methodology the physic, performance and environmental characteristics are considered, it is very important to look upon these kind of variables due to their influence in the VRT integration performance, it is recommended to define certain features of the integration in order to direct all the previous work and analysis to satisfy the demands established by the features definition. The way how the devices are going to interact, the physic features of the workpieces, the lighting conditions, the space available and another environment conditions, are some examples of characteristics that will change in every application but they need to be defined to lead the work to the own application demands.

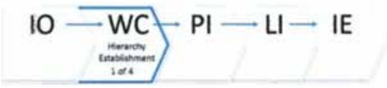
4.2.1 Hierarchy establishment

This feature regards to the determination of the device (from now on Master Device, MD) which status and information will control directly or indirectly the performance of all other devices of the system. As the integration that is being discussed refers to Robotic and Vision Technologies, the two main devices are the Vision System and the Robot involved in the application. As can be inferred and as Espinosa (2006) presents the two arrangements possible in this kind of integration are "Vision system assisted by robot" and "Robot assisted by Machine vision". It is good to remember that as it has been said in section 3.1.1 and section 3.2.1 an integrated Vision Robot system generally includes a third component: The control unit(usually a PC or a PLC).

1. Vision system assisted by robot.- In this case the MD is the Vision system because the main task is performed by the vision system (for example quality inspection of an object) but it needs previous assistance of the robot to locate the camera in the position in which the object of interest is inside the field of view of the camera.
2. Robot assisted by Machine vision.- In this case the MD is the Robot because it performs the main task (arc-welding) but needs information form the camera (the position of the object to be welded) in order to execute the task correctly.

It is important to define the role of each device to identify the kind of integration (1 or 2), so in order to identify the master device, the following methodology is proposed:

1. Identify the main task of the system (mt)
2. Identify which device(d) performs the mt
3. Identify the scenarios of the system (s)
4. Identify which device is controlling in each scenario
5. Get the number of scenarios controlled by each device
6. If the mt is performed by the device that realizes the major quantity of scenarios this device is the MD, if not go to next step.
7. Assign a weight to each scenario in a scale from 1 to 5, where 5 is the rate for the most important task (mt)
8. Sum the score of each device (due to its performed tasks) and the one with higher score should be considered as the MD.



The last method is represented in the next flow diagram:

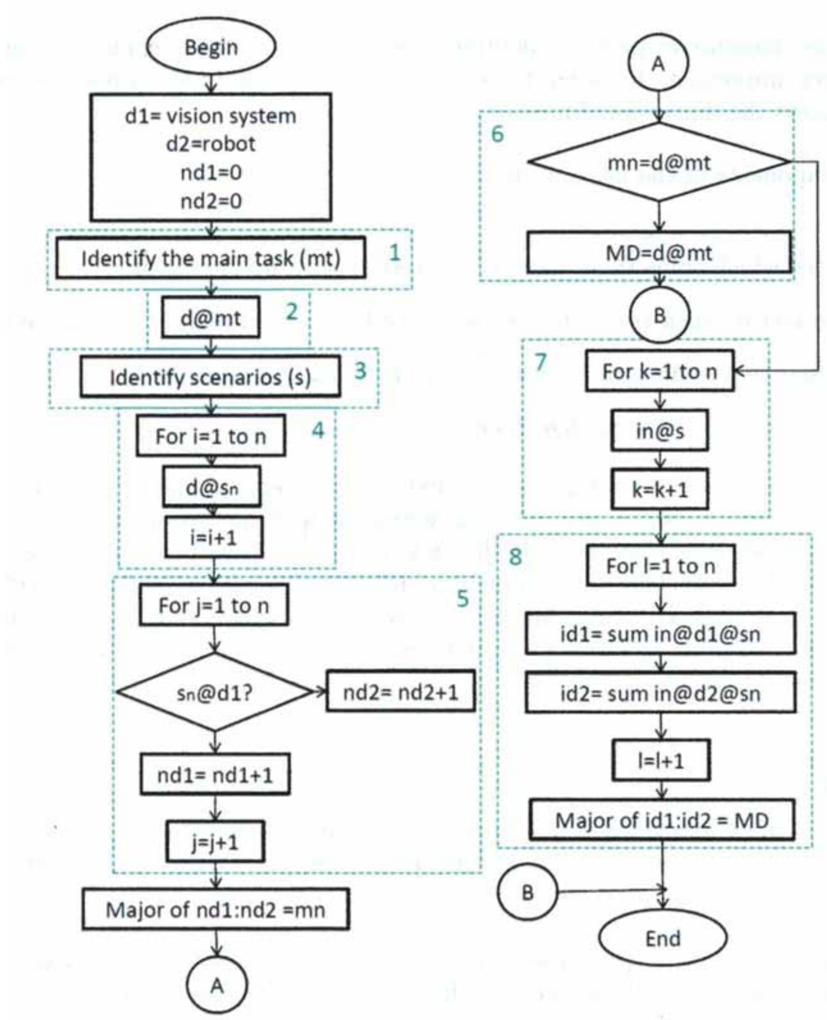
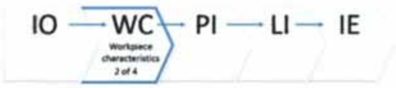


Figure 4.7: Hierarchy establishment method

Now that the MD is identified a better perspective of the device functions can be obtained, so the following analysis can be guided to MD features and demands.



4.2.2 Workpiece characteristics

Workpiece Shape

For 3D image analysis (measure,inspection, identification,etc.) the need for acquiring a great quantity of images is very important, in order to solve this issue, Shin (2002) presents an imaging mechanism which allows the dimensional measuring for a ball stud.

The principal components of the mechanism are:

1. Rotational device which spins the Object of Interest around it's own axis.
2. Vertical sliding device which translate the Object of Interest through (z)vertical axis.
3. Horizontal sliding device which moves the camera forward and backward.
4. Rotational device which adjust the lens focus.

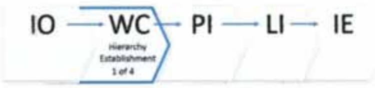
Bone (2003), presents a work in reference with a fixtureless assembly, and in order to locate the camera in the correct position used a pan/tilt unit which allowed the vision system to be aimed at any location in the workcell where visual feedback is required. This unit was necessary due to the size of the assembled components (in millimeters: 180x800x200, 290x110x70, 150x160x60 and 250x640x170), as it can be noticed, the characteristics of the workpiece (shape, size, surface,etc.) will determinate the need of extra mechanisms or devices in order to acquire the information needed by the VRT integration.

Workpiece feeding

Kruger(2009) identifies the importance of the workpieces feeding in the manufacturing operations, due the great diversity of shape, size and complexity of the products there is not a Methodology that gives a straight road forward to an automation solution.

The use of a subdivision of this type of task is proposed by Kruger(2009) and in base with Langmoen(1983) and Rampersad(1994), the general feeding task is classified as follows:

- Production batch volume of a given unique component
  - Large (mass production)
  - Medium
  - Small and single piece (produce to order)
- Component size
  - Large (characteristic dimension ¿1m)
  - Medium (characteristic size between 100 mm and 1 m)
  - Small (characteristic size between 1 mm and 100 mm)
  - Miniature (characteristic size ¡ 1 mm)

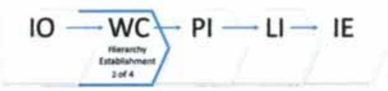


- Component complexity
  - Simple shape, symmetric or semi symmetric
  - Unsymmetrical but simple shape
  - Complex shape
- Component stiffness
  - Stiff, non-deformable objects
  - Semi stiff, deformable (like meat, rubber components, wires, etc.)
  - Limp objects (like textile, tissues, thin plastic sheets, etc.)
- Fragility
  - Unbreakable by normal handling
  - Moderately fragile (requires some attention to handling forces)
  - Very fragile (must be handled very delicately)

Kruger(2009) proposes the subdivision of the feeding process into four major steps:

- Separation: One unique component is separated from the bulk volume.
- Transfer: The component is brought to a point very close to the point of pick up or treatment in the next stage of the manufacturing operation.
- Orientation: The component is brought from a general orientation into the specifically wanted orientation for the operation next in the process.
- Positioning: The component is positioned precisely within required tolerances for the next handling step in the process.

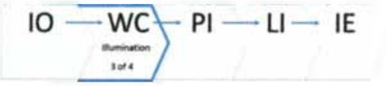
There are several feeder techniques that in base with the product characteristics include or not all the steps of the feeding process. For large volume manufacturing where small rigid parts are handled the feeders usually include the whole process, but for large parts, small volume or fragile parts the feeding process is realized in steps.



Boothroyd(2005) has described a great quantity of feeders and mechanisms, but the most common principles used in nowadays are: Vibratory bowl feeder, elevator feeder, belt feeder and drum feeder.

Figure 4.8: Vibratory brush feeder for gentle feeding of fragile parts.

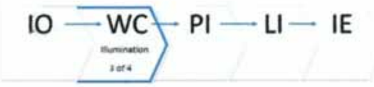
Figure 4.9: Vibratory bowl feeder with active sensing reorientation.



4.2.3 Illumination

As it has been explained in Chapter two, the illumination is a very important factor in the use of vision systems, according to Valiente (2010) most of the times in the machine vision applications, the design of the illumination system is more determinant in a successful identification or inspection than a highly and sophisticated image analysis.

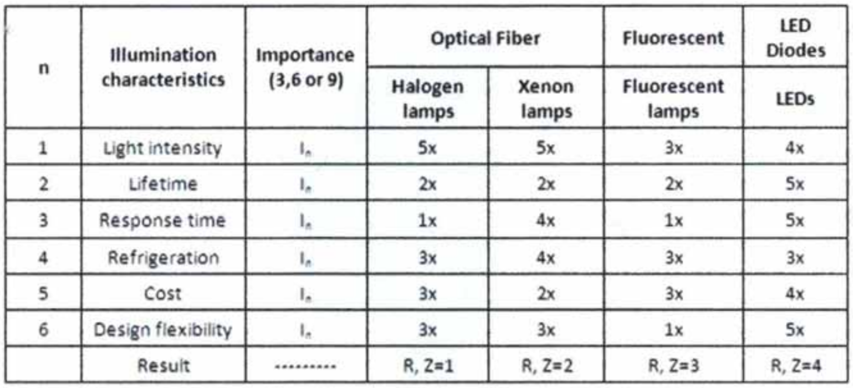
In order to select and/or design an illumination system, there are two general guidelines to follow, the type of illumination and the illumination technique.



Illumination type

For the type of illumination a evaluation and comparison chart is proposed, this chart helps the designer to select a specific illumination system based on a characteristic evaluation, the most important characteristics to be considered in the illumination systems are: light intensity, lifetime, response time, refrigeration, cost and design flexibility. The following figure shows the evaluation chart proposed for the illumination system selection.

Table 4.3: Evaluation chart for illumination type selection



n	Illumination characteristics	Importance (3,6 or 9)	Optical Fiber		Fluorescent	LED Diodes
			Halogen lamps	Xenon lamps	Fluorescent lamps	LEDs
1	Light intensity	I_n	5x	5x	3x	4x
2	Lifetime	I_n	2x	2x	2x	5x
3	Response time	I_n	1x	4x	1x	5x
4	Refrigeration	I_n	3x	4x	3x	3x
5	Cost	I_n	3x	2x	3x	4x
6	Design flexibility	I_n	3x	3x	1x	5x
	Result	---------	R, Z=1	R, Z=2	R, Z=3	R, Z=4

In order to fill the evaluation chart, the first step is to identify if there are any characteristics that due to the specific application should be included in the chart, also the designer must identify if there is any characteristic included that should be removed, if the designer adds a new characteristic he must evaluate it in a scale from 1 to 5 each illumination option (halogen, xenon, fluorescent and LED) and continue with the next step.

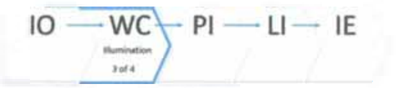
The second step is to assign the importance of each characteristic included, this importance value should be 3, 6 or 9 and will depend on the application characteristics and in the design priorities, is recommended to avoid the situation where 3 or more characteristics have the same importance, because this assignment will not significatively represent the design priorities.

The next step is to calculate the evaluation result for each illumination option, this will be obtained by the application of the next formula.

$$R_z = \sum_{1}^{n} I_n M_{n,z} \tag{4.3}$$

Where z is the identifier for each illumination option, n is the number of system characteristics, I_n is the importance of the n characteristic assigned by the designer, and $M_{n,z}$ is the modifier that evaluates the n characteristic for the z option.

The next step is to select the illumination system for the application according to the evaluation results and working conditions identified in the last sections.



The following step consists on determine if a color filter is needed or not, remembering that the color filters increase contrast in scenes where the color of object and background are different, but they appear to be the same when converted to grey values, a color light could be used instead a color filter, it is recommended to use the light color only if the design flexibility is not a important characteristic of the system, the interchange between color filters make the system more flexible and allows the improvement of the image quality even if the product or background color changes.

If the designer finds the color filter needed, he should use the next chart to determine which color filter (or light) will be the appropriate for the application:

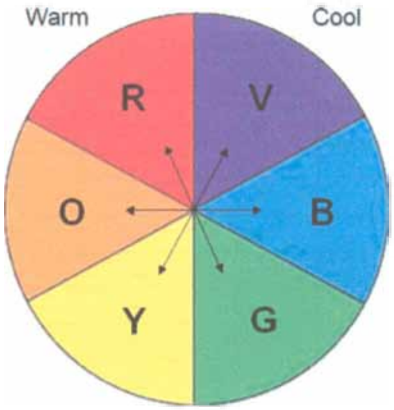
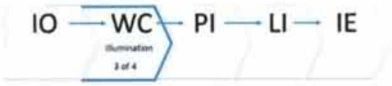


Figure 4.10: Relationship between colors families

Use similar colors or families (warm or cool family) to lighten. For example, Yellow light makes yellow features brighter.

Use opposite colors or families (warm or cool families) to darken. For example, Red light makes green features darker.

The next figure shows a flow diagram that represents the illumination type selection through the use of the evaluation chart.



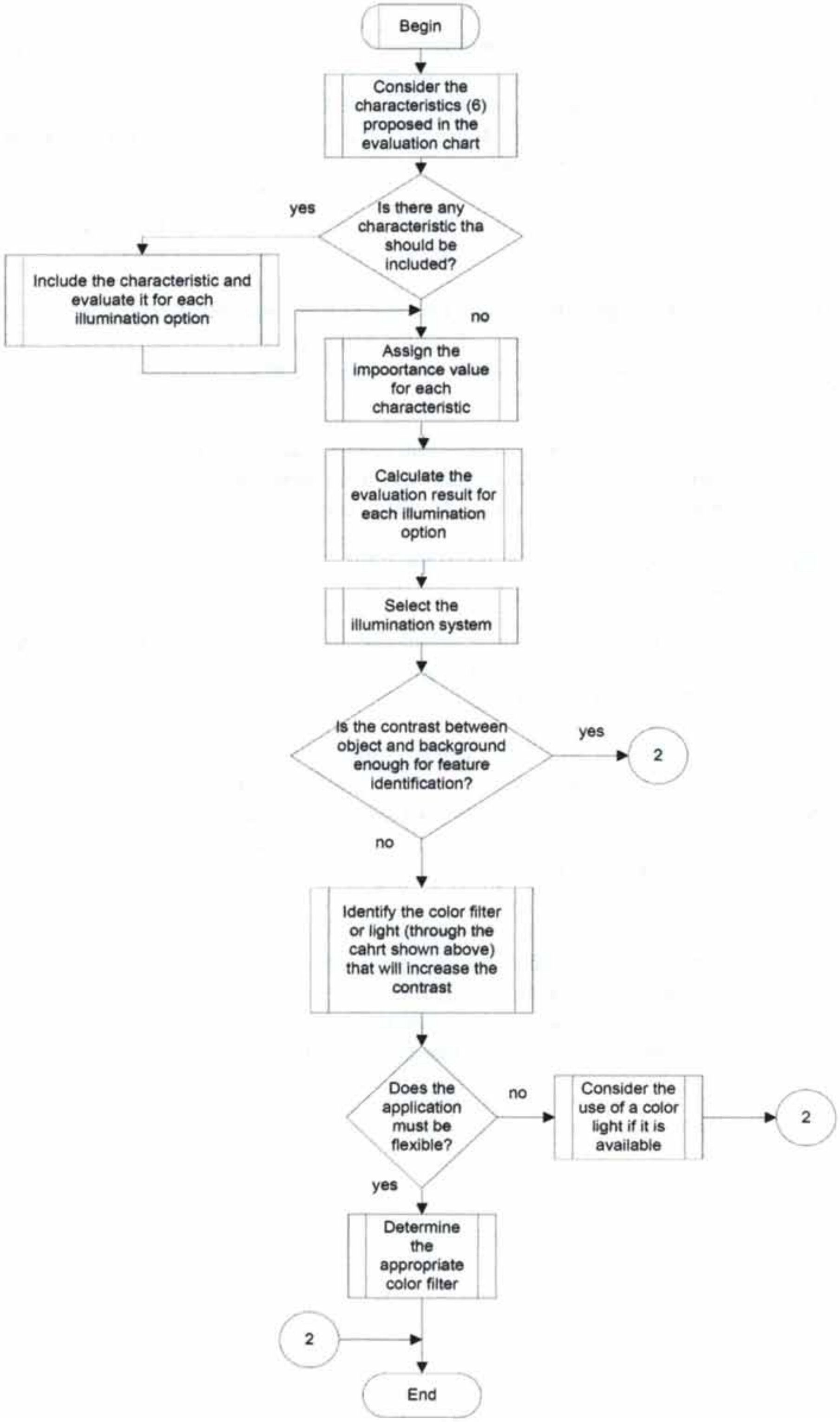
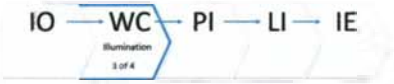


Figure 4.11: Selection process for illumination type



Illumination technique

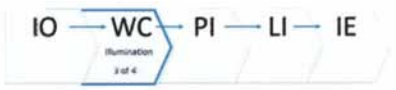
Once that the illumination type has been selected, another important step is to define the illumination technique, as it has been explained in Chapter II, there are a lot of techniques and the selection of one will be determinant on the success or not of the illumination system, even if a correct illumination type has been already chosen, following it is presented the methodology to select an appropriate technique for the vision system.

It is necessary to define that this part of the methodology is based on the next 4 parameters:

- **Number of orthogonal planes to be analyzed.-** As all the parameters involved in this section of the methodology, the number of planes will depend on the application nature, in this methodology is being considered only orthogonal planes (when all planes are perpendicular to each other).
- **Object material characteristics.-** Every material has a lot of properties and characteristics, but in this methodology the features of interest are related with the type of surface and its optical behavior (reflective, translucent, refractive, etc.).
- **Mounting conditions.-** refers to al dimensional parameters that could limit the use of certain illumination technique.
- **Application objective.-** as it has been explained in chapter II, the use of vision systems is intended for many applications, each one requires a different image analysis, so also requires a certain and specific illumination system.

The selection of the illumination technique will depend on the principal parameters shown above, in this methodology it is presented an analysis to define a recommended technique in base with Valiente (2010) and Perez (2010), it is important to clarify that recommending a technique for a certain application does not mean that the others techniques are not useful or appropriate in that application, it is presented, in base with the bibliographic research, the best and recommended option for the application.

The first step to select the illumination technique consists on the assignment of a priority for each of the 4 parameters, this assignment is as variable as applications of vision system can be, that is why is very difficult to evaluate or propose a evaluation factor, once that the designer has made this step, he must organize the charts shown below this paragraph in an importance order.



Number of orthogonal planes to be analyzed

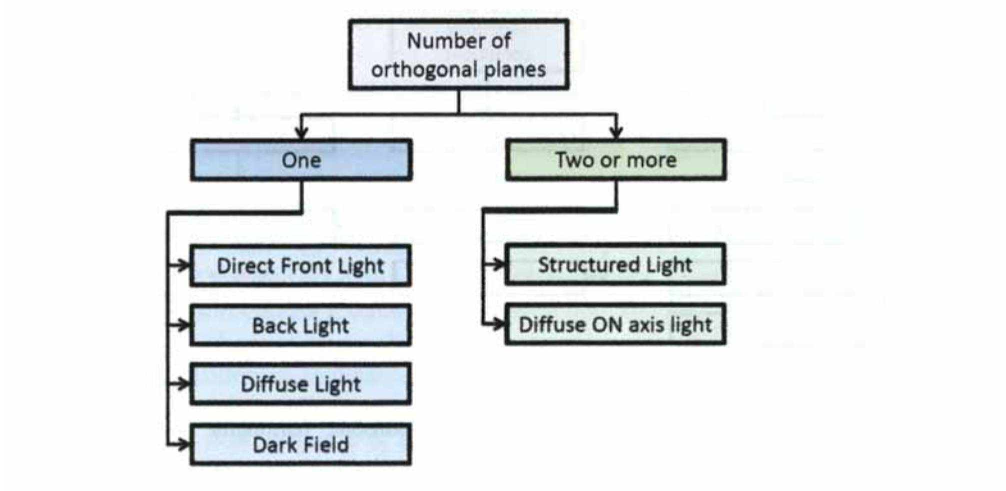


Figure 4.12: Options for "orthogonal planes" parameter

Object material characteristics

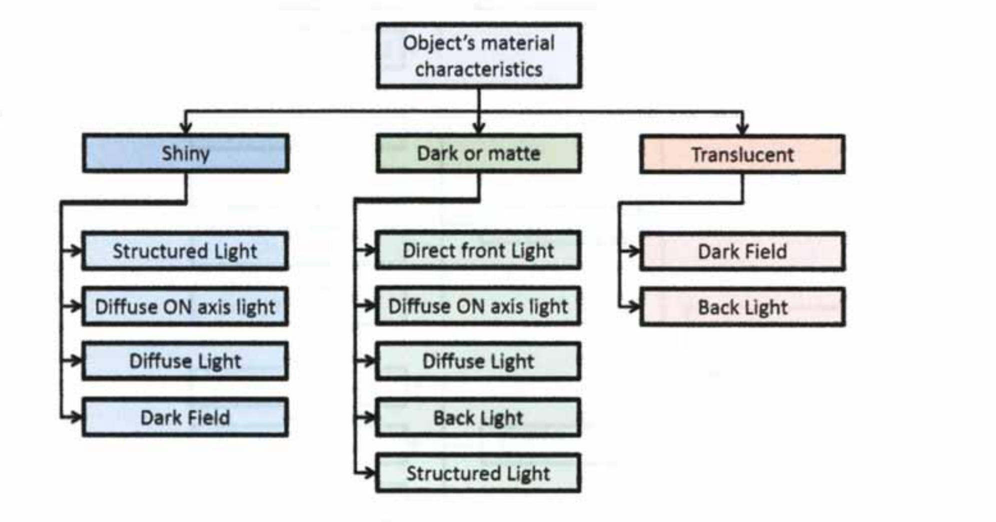
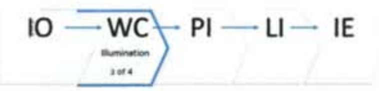


Figure 4.13: Options for "material characteristics" parameter



Mounting conditions

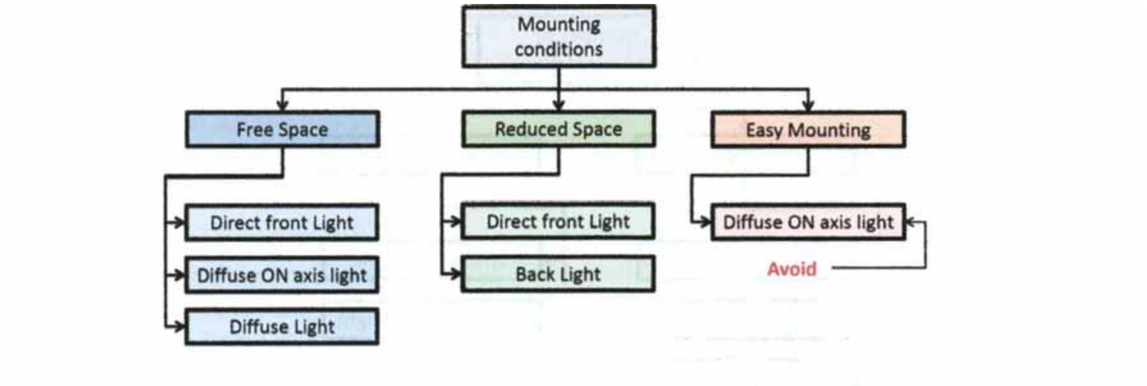


Figure 4.14: Options for "Mounting conditions" parameter

Application objective

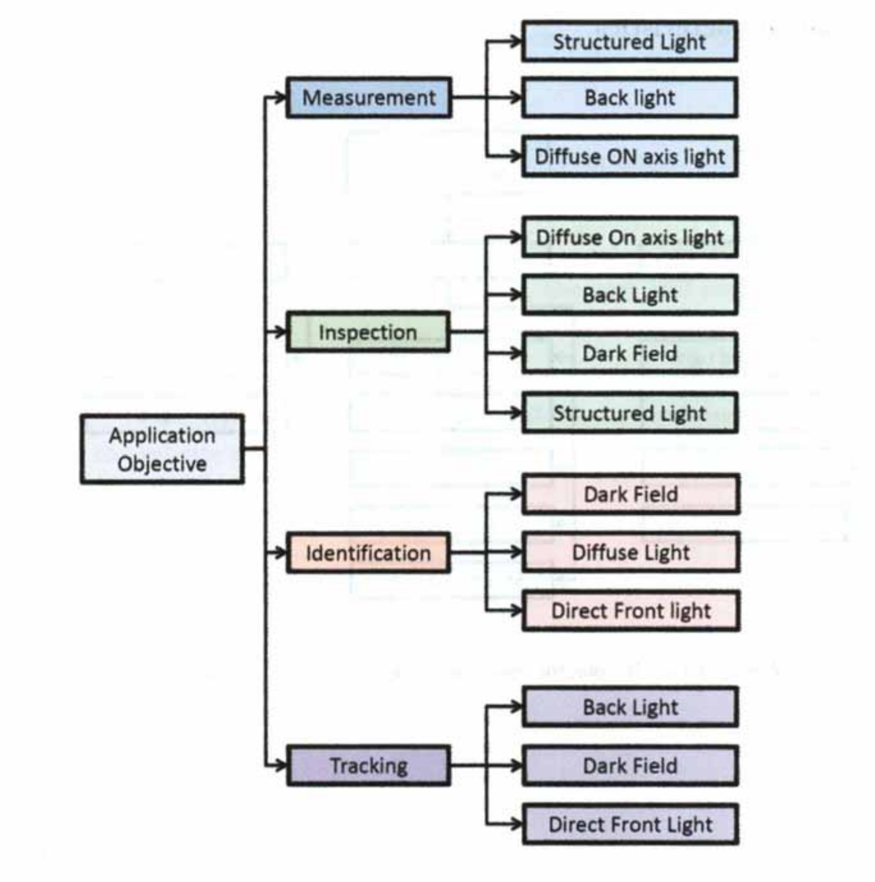
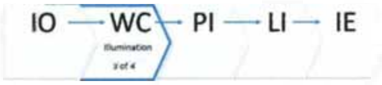


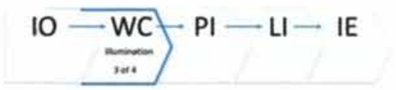
Figure 4.15: Options for "Application objective" parameter



As it is seen, there is a chart for each parameter, these charts help the designer to choose a technique depending on his specific application, for example if the most important parameter for the designer is the number of planes to be analyzed (2 planes), he must look for this option in the appropriate chart, now it can be inferred that the best options for this analysis are "structured light" and "diffuse light".

The following step is to evaluate the application on the next chart (remind that the charts must be ordered) and select the adequate technique, the designer must repeat this process for each chart (4 in total), and then he must have selected the final technique. This part of the methodology is represented in the flow diagram of figure 4.17

There will be experiments where the parameters could have almost the "same" importance and it will be difficult to order them, in this case an alternative process is proposed, where the designer must evaluate all the charts and choose a technique for each one, the first option is when the selected technique is the same in all the requirements, in this case the selection will be unanimous. But, if in the selection per chart different techniques are chosen, the designer must look for a technique that accomplish according to his application at least three of the four parameters. This alternative process is represented in the flow diagram of Figure 4.18



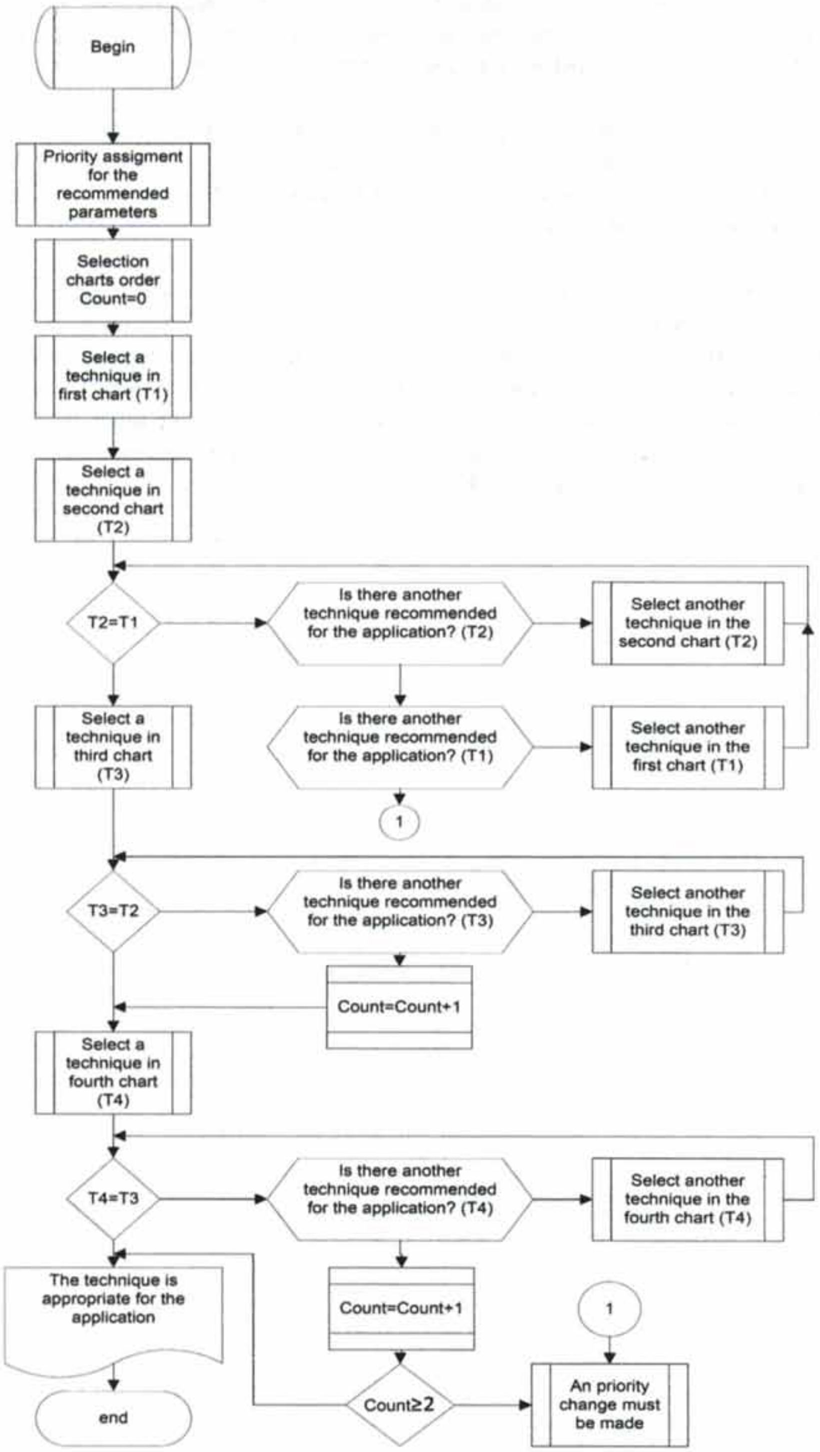


Figure 4.16: Process to determine the illumination technique

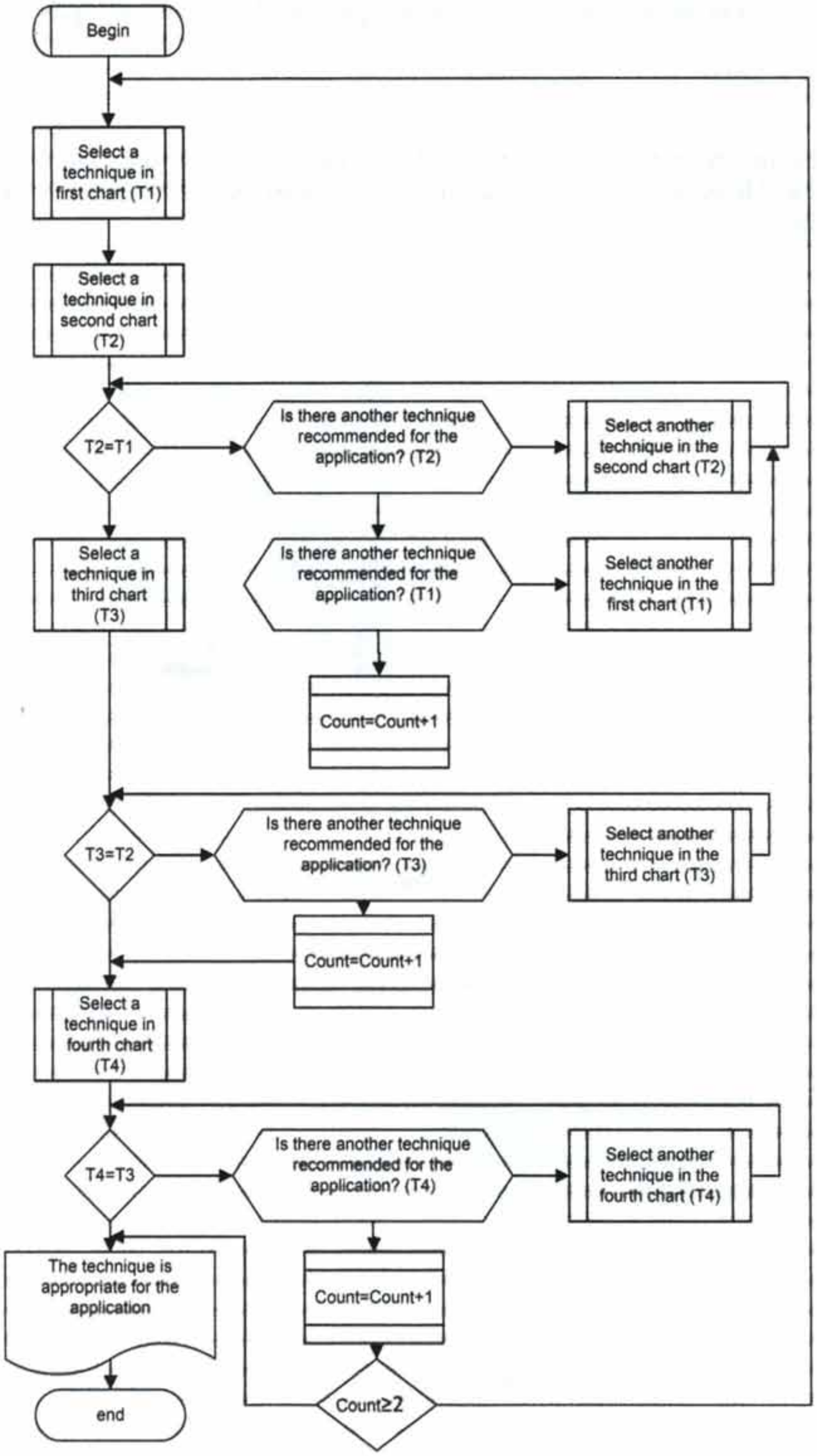
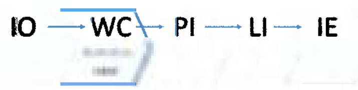


Figure 4.17: Alternative process to determine the illumination technique



In order to facilitate the application of the presented methodology, it has been developed an application software for the Illumination type and technique selection, where de input data includes:

For technique selection:

- Importance factors (from 1 to 4) for each one of the 4 parameters (Orthogonal planes, Material Characteristics, Mounting conditions and Application objective) that determine the integration requirements.

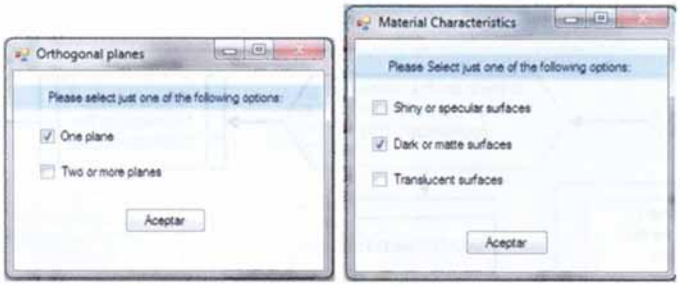

Illumination Technique Selection Software (ITSS)

Importance	Parameters
1	Orthogonal Planes
3	Material Characteristics
4	Mounting Conditions
2	Aplication Objective

Recommended techniques:

Figure 4.18: ITTSSoftware importance assignment

- Selection of just one option of the ones presented for each one of the 4 parameters.


Orthogonal planes

Please select just one of the following options:

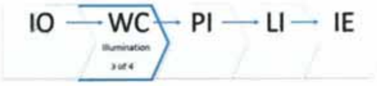
☑ One plane
☐ Two or more planes

Material Characteristics

Please Select just one of the following options:

☐ Shiny or specular surfaces
☑ Dark or matte surfaces
☐ Translucent surfaces

Figure 4.19: Option selection for each parameter



For light type:

- Importance factor for each one of the 6 parameters (Light intensity, Lifetime, Response time, Refrigeration, Cost, and Design Flexibility) that constraint the light type selection.

Assign an importance factor for each parameter factor should be 1,6 or 9 the press Go"

The recommended light type for your application is the one with higher ranking

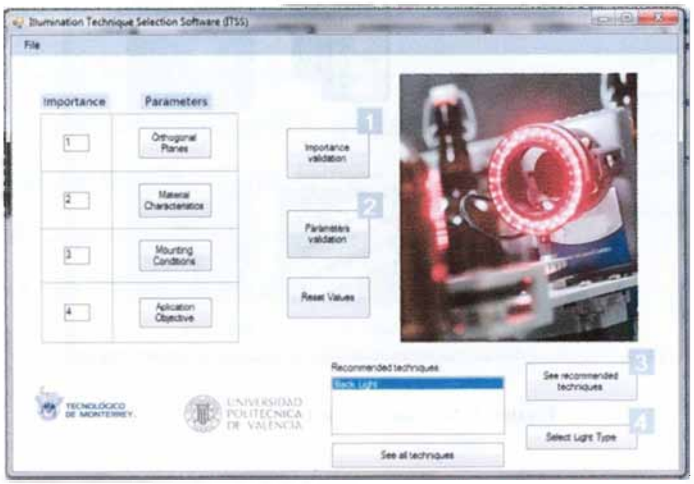
Light intensity: 0
Lifetime: 0
Response time: 0
Refrigeration: 0
Cost: 0
Design Flexibility: 0

Figure 4.20: Light type selection

And as output data the software throws:

For technique selection:

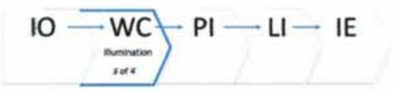
- The illumination technique(s) suitable for the application.



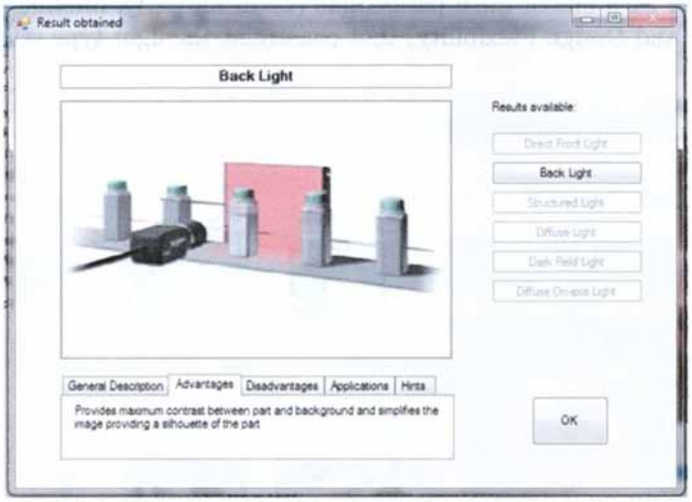
Orthogonal Planes: 1
Material Characteristics: 2
Mounting Conditions: 3
Aplication Objective: 4

Recommended techniques: Back Light

Figure 4.21: Illumination technique result



- General description, Advantages, Disadvantages, Applications, and Hints for the result.

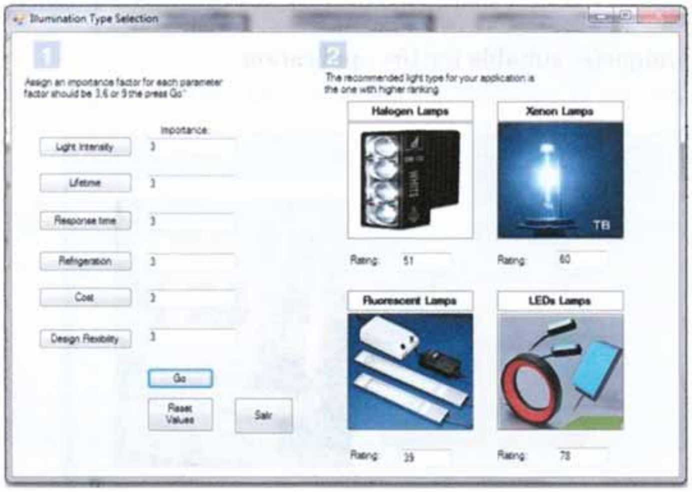


Provides maximum contrast between part and background and simplifies the image providing a silhouette of the part

Figure 4.22: Information provided by ITTSS

For light type:

- The light type that is recommended for the working conditions and the relative score of each one of the light types considered.



Assign an importance factor for each parameter factor should be 3,6 or 9 the press Go"

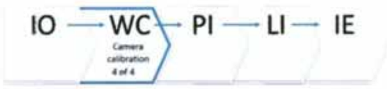
Light Intensity 3
Lifetime 3
Response time 3
Refrigeration 3
Cost 3
Design Flexibility 3

The recommended light type for your application is the one with higher ranking

Halogen Lamps Rating: 51
Xenon Lamps Rating: 60
Fluorescent Lamps Rating: 35
LEDs Lamps Rating: 78

Figure 4.23: Light type result

More detailed information about the software is presented in Appendix A.



4.2.4 Camera Calibration

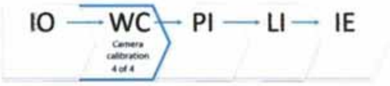
As a result of the research in camera calibration process (see section 2.1.2) there are some software packages or modules which make this procedure in an automatic and reliable way, for example Mathworks released a "Camera Calibration Toolbox" for Matlab which bases it's performance in Zhang's method, this toolbox also includes a Correspondence with other calibration methods notation as **Heikkil and Reg Willson's**, another software is "GML Camera Calibration toolbox" developed by "Graphics and Media Lab in Moscow State University", this software is Operative System independent and the results of intrinsic parameters are easily obtained, different method's notation are provided and supports two detection and two calibration methods, "OpenCV" is a library of programming developed by Intel it is cross-platform and provides methods and classes to calibrate a camera to finally get the intrinsic parameters of the camera, Cognex also provides a module in it's software "Insight" which allows to the user calibrate the camera and apply the adequate correction to the image obtained by the system. Usually the main information of every calibration procedure consists on:

- Focal Length of the camera
- Distortion parameter
- Camera matrix
- Pixel error
- Principal point

As well the extrinsic parameters: Rotation Matrix(R) and Translation Vector (t) are needed in order to transform the 3D camera coordinates to 3D world coordinates and viceversa.

In base with the calibration procedure proposed by Sanchez (2008) and with the research of V.Vezhnevets and A.Velizhev (2005) the use of a toolbox to calibrate the camera and the following hints for camera calibration are proposed:

- Capture more than 25 photos of the chessboard
- All the corners should be visible in every photo
- Locate the chessboard pattern along the camera field of view
- Make photos from the positions shown in Figure 4.25
- Maintain the tilt angle of the camera constant
- If available select the Zhang's calibration method



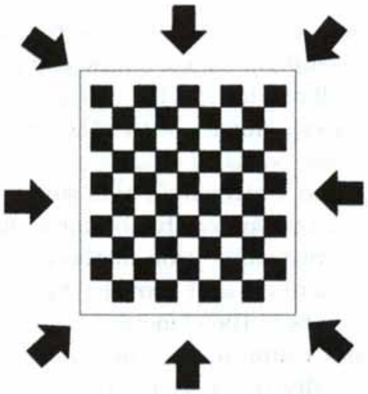


Figure 4.24: Suggested positions to make photos for calibration process

Speed of response of the vision system

The speed of response of the vision devices represented in frames per second (fps) is one of the most important features in the utilization of a vision system, nowadays this speed has increased up to 1000 fps Kagami(2010), this speed allows the user to acquire a great quantity of images in order to detect changes in the scene, in this research this parameter (speed of response) is not considered due the velocity of the robot movements, as this movements are very slow (in reference with the cameras capability), the speed of response will not be a determinant consideration in the integration of VRT.

4.3 Physic Integration

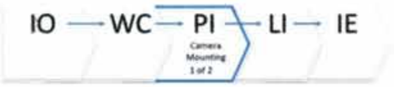
4.3.1 Camera Mounting

When integrating VRT, there are a lot of configurations available regarding to camera mounting, as it has been mentioned before, there is a primary classification that includes eye-in-hand mounting and external mounting, as the experimental setups of previous researches (Pauli, 2001; Muis, 2005; Pea-Cabrera, 2005; Motai, 2008; Shin, 2009; Hernandez, 2010; and others) show that there are many subcategories inside the two mentioned above. The objective of this part of the research is to provide a method to identify the subconfiguration that is more suitable for certain application. There is a need to mention that this method most probably apply when environment restrictions do not constraint the integration in a high level.

A proposed classification for the camera mounting configurations is presented next:

1. **Eye in hand.-** The camera is mounted as near as possible to the robot's end-effector
    (a) **Fixed.-** The camera is permanently fixed through mechanic fasteners.
    (b) **Interchangeable.-** The camera is fixed through pneumatic or electric fasteners which allow the automatic mounting/dismounting operation.
2. **External.-** The camera is mounted any place except in the robot.
    (a) **Eye to hand.-** The camera placement allows to obtain the end-effector's location and pose.
        i. **Fixed.-** The camera is permanently fixed in a certain location.
        ii. **Articulated.-** The camera is mounted to a fixture which has more than one degree of freedom.
    (b) **Eye to object.-** The camera placement allows to obtain the object's location and pose.
        i. **Fixed.-** The camera is permanently fixed in a certain location.
        ii. **Articulated.-** The camera is mounted to a fixture which has more than one degree of freedom.
    (c) **Panoramic.-** The camera placement allows to obtain information about all devices or objects included in the image.
        i. **Fixed.-** The camera is permanently fixed in a certain location.
        ii. **Articulated.-** The camera is mounted to a fixture which has more than one degree of freedom.

In base with the proposed classification a method to determine which configuration should be used is presented as follows:



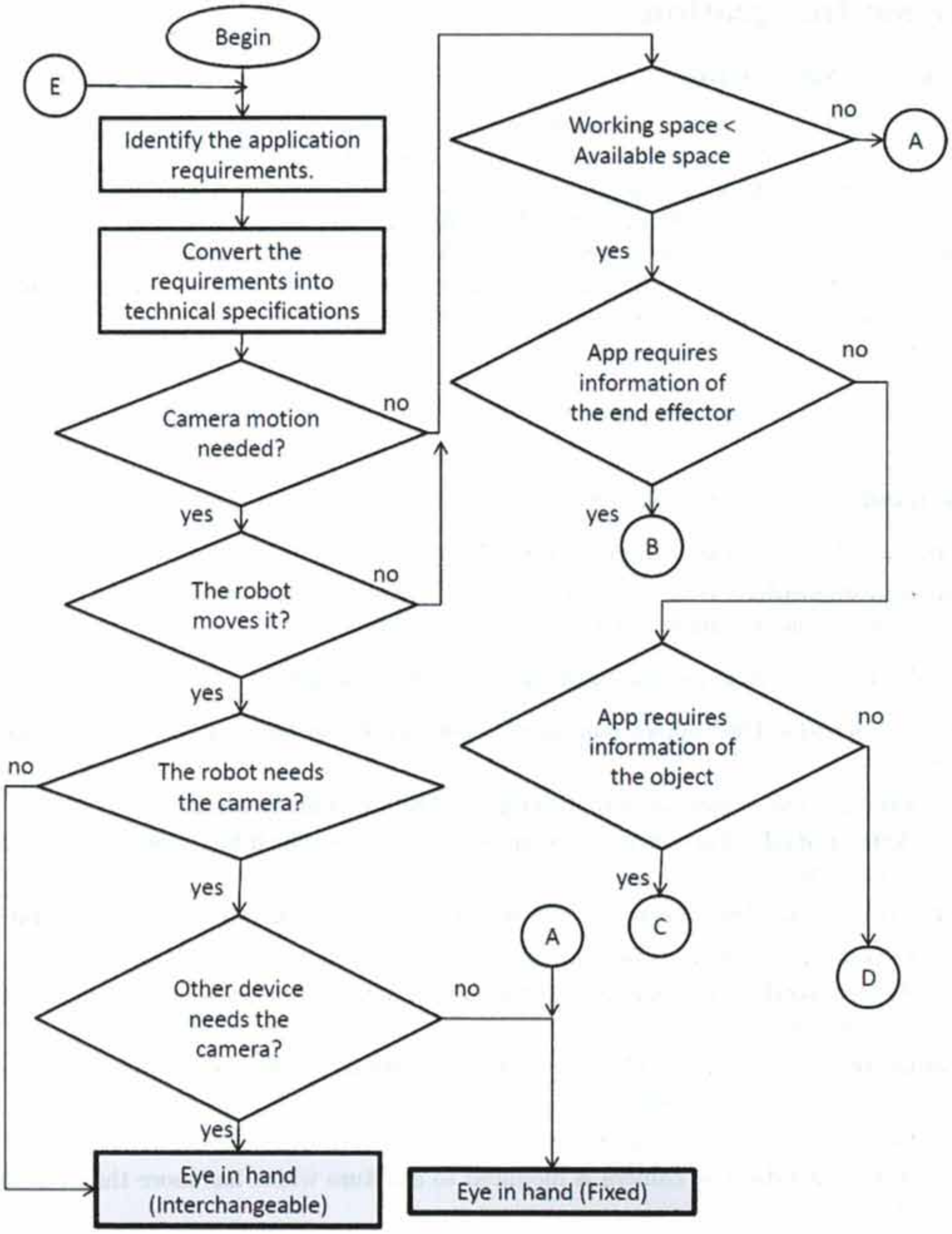
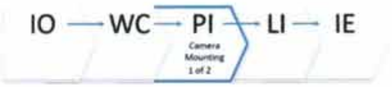


Figure 4.25: Process to obtain the mounting configuration



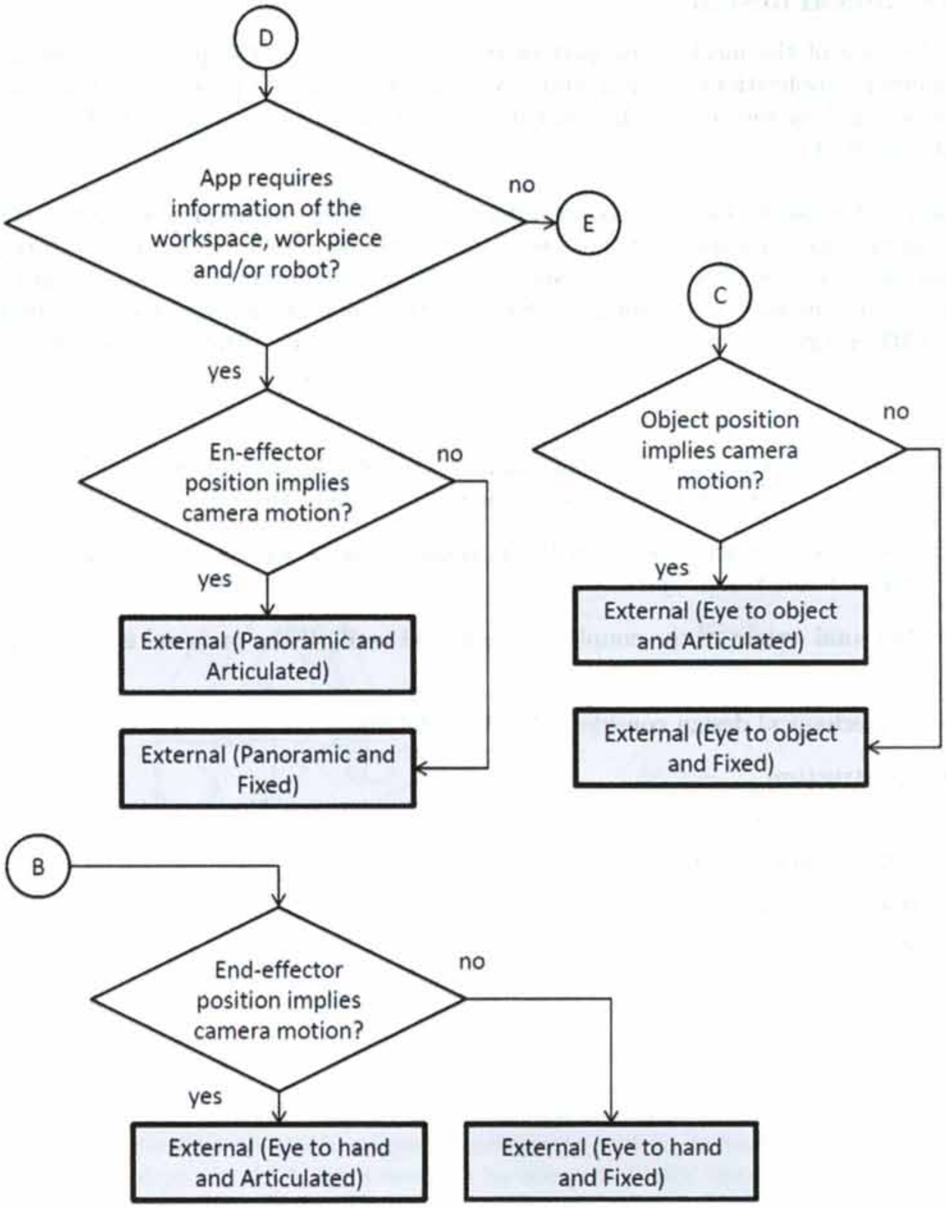
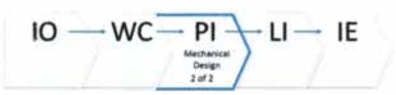


Figure 4.26: Process to obtain the mounting configuration (cont.)

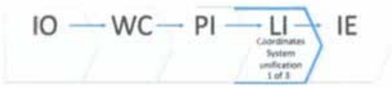


4.3.2 Mechanical design

This is probably one of the most vague part of the methodology in the perspective of the great quantity of different applications, devices and environment conditions possible. So the presented guidelines are in somehow too general, however it is important to remind this part of the VRT and the main hints involved.

This section is directed to the mechanical connections between the main components of a VRT system, for example: the attachment of the camera to the fixture which will fix it to a certain location, or the mechanism that allows the automatic mounting/dismounting of a eye in hand camera configuration, etc. In the work of Hernandez (2010), a eye in hand (interchangeable) configuration was used in a VRT integration, in order to provide the interchangeable condition the following hints are proposed:

- Use pneumatic actuators to “grab” the camera, consider the usage of a small(as much as possible) actuators that may be able to handle the payload of the camera
- Remind that a high security factor in the payload estimation should be included (Due the fragility and cost of a Vision system)
- Consider the total weight of the coupling system and verify if the proposal is viable (Robot’s features)
- As in every mechanical design consider DFMA guidelines:
  - Module structure
  - Complexity reduction
  - Identify a main element
  - Restrict the mounting axis
  - Facilitate unions



4.4 Logic Integration

4.4.1 Coordinates System Unification

In order to detect and identify the object of interest, there are a lot of algorithms implemented in the software of the vision systems, and once that the calibration procedure has been done, the interpretation of the available information is the next step in consideration.

This step of the methodology considers the following procedure:

1. **Obtain the intrinsic parameters of the camera** (Usually available after the calibration process has been realized).
2. **Obtain the position and orientation of each of the objects of interest** (This information is obtained by the use of the vision system software).
3. **Transform the object's localization to world's coordinates system of the robot.**

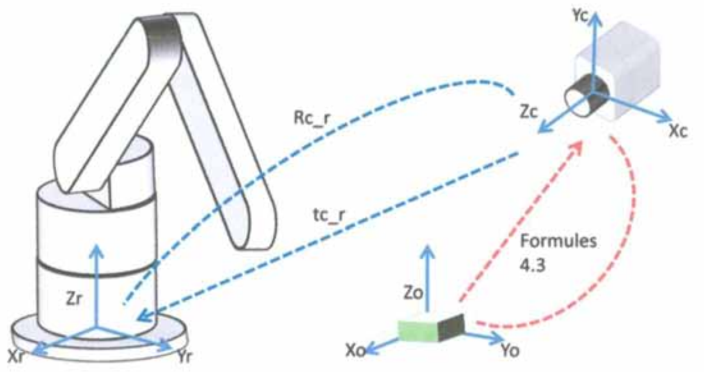


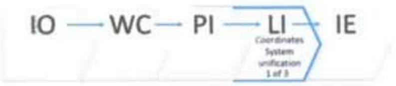
Figure 4.27: Coordinates Systems of the scene

In order to perform the transformation the first step is to transform the object's coordinates from image coordinates system to camera's coordinate system, in base with Burrus (2010) the 3D point referred to certain image point can be computed with the next formulas:

$$X_c = \frac{(x - u_0) * z}{\alpha_x}, \ Y_c = \frac{(y - v_0) * z}{\alpha_y} \ and \ Z_c = z \tag{4.4}$$

Once that the object's location in reference with the camera's coordinate system is obtained the next step is to get the object's location in reference with the world's coordinate system of the robot, to perform this task a Rotation and Translation transformation is needed, the transformation is represented by:

$$P_r = R * [P_c] + t \tag{4.5}$$



$$P_r = \begin{pmatrix} X_r \\ Y_r \\ Z_r \end{pmatrix} = (R) \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \tag{4.6}$$

This transformation also can be represented as:

$$\begin{pmatrix} X_r \\ Y_r \\ Z_r \\ 1 \end{pmatrix} = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} \tag{4.7}$$

The rotation matrix R and the translation vector t are unknown, but this parameters can be obtained bye the following procedure:

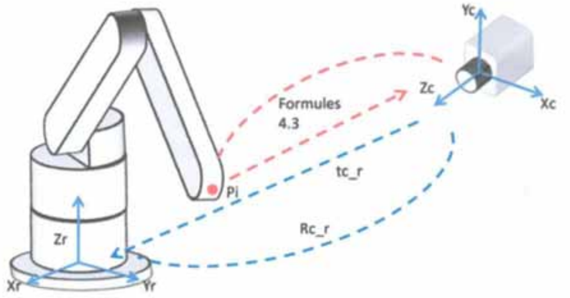


Figure 4.28: Rotation matrix and translation vector calculation

(a) Fix the camera in the working position of the application.

(b) Place a mark in the Tool Center Point of the robot. (see figure 4.29)

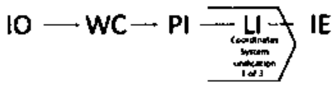
(c) Move the robot to a known position $Pr(Xr, Yr, Zr)$

(d) Identify the mark (which corresponds to the "Tool Center Point") with the camera, and get the coordinates of this point in reference with the image coordinates system $p(x, y, z)$.

(e) Transform the localization coordinates of the mark detected into a new localization in reference with Camera's coordinates system $Pc(Xc, Yc, Zc)$ (Equations 4.3).

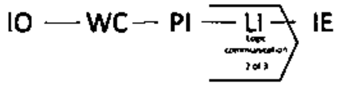
(f) Compute the Rotation matrix R and the translation vector t.

In order to compute this parameters the location of points referenced to the camera and to the robot is needed, each point data corresponds to three equations (x,y,z), and as ${}^{c}R_r$ and ${}^{c}t_r$ have 12 unknown variables, the number of needed points to get ${}^{c}R_r$ and ${}^{c}t_r$ is $12/3 = 4$ points resulting the following equation system:



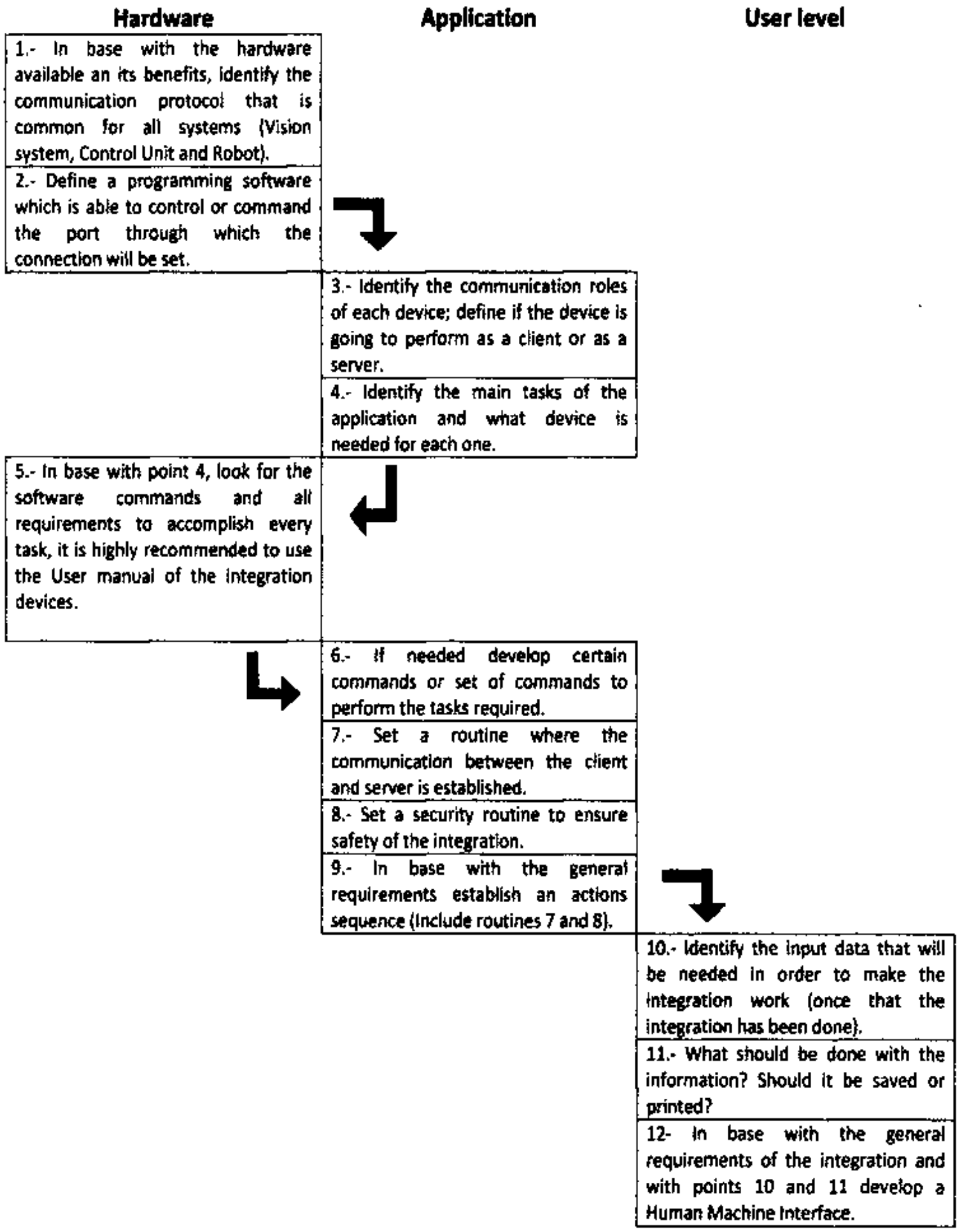
$$
\begin{aligned}
X_{R1} &= X_{C1}R_{XX} + Y_{C1}R_{XY} + Z_{C1}R_{XZ} + t_X \quad (4.8)\\
Y_{R1} &= X_{C1}R_{YX} + Y_{C1}R_{YY} + Z_{C1}R_{YZ} + t_Y \quad (4.9)\\
Z_{R1} &= X_{C1}R_{ZX} + Y_{C1}R_{ZY} + Z_{C1}R_{ZZ} + t_Z \quad (4.10)\\
X_{R2} &= X_{C2}R_{XX} + Y_{C2}R_{XY} + Z_{C2}R_{XZ} + t_X \quad (4.11)\\
Y_{R2} &= X_{C2}R_{YX} + Y_{C2}R_{YY} + Z_{C2}R_{YZ} + t_Y \quad (4.12)\\
Z_{R2} &= X_{C2}R_{ZX} + Y_{C2}R_{ZY} + Z_{C2}R_{ZZ} + t_Z \quad (4.13)\\
X_{R3} &= X_{C3}R_{XX} + Y_{C3}R_{XY} + Z_{C3}R_{XZ} + t_X \quad (4.14)\\
Y_{R3} &= X_{C3}R_{YX} + Y_{C3}R_{YY} + Z_{C3}R_{YZ} + t_Y \quad (4.15)\\
Z_{R3} &= X_{C3}R_{ZX} + Y_{C3}R_{ZY} + Z_{C3}R_{ZZ} + t_Z \quad (4.16)\\
X_{R4} &= X_{C4}R_{XX} + Y_{C4}R_{XY} + Z_{C4}R_{XZ} + t_X \quad (4.17)\\
Y_{R4} &= X_{C4}R_{YX} + Y_{C4}R_{YY} + Z_{C4}R_{YZ} + t_Y \quad (4.18)\\
Z_{R4} &= X_{C4}R_{ZX} + Y_{C4}R_{ZY} + Z_{C4}R_{ZZ} + t_Z \quad (4.19)
\end{aligned}
$$

The resulting equations system can be solved by any numeric method for example Gauss-Jordan elimination or Newton-Raphson method, by the end of the solution the 12 variables values will be known consequently the rotation Matrix and the translation vector could be built.



4.4.2 Logic communication

As it is known the development or application of software (programming level), is very variable in function with the user or programmer, the advantages that many of the devices(logic) have, is that generally there are a lot of logic structures that in certain arrangement are able to obtain the same result. As it is impossible to establish a generic rule to establish communication between Vision System - Control Unit - Robot, a list of common tasks that should be done in order to integrate the VRT is proposed as follows:



1.- In base with the hardware available an its benefits, identify the communication protocol that is common for all systems (Vision system, Control Unit and Robot).

2.- Define a programming software which is able to control or command the port through which the connection will be set.

3.- Identify the communication roles of each device; define if the device is going to perform as a client or as a server.

4.- Identify the main tasks of the application and what device is needed for each one.

5.- In base with point 4, look for the software commands and all requirements to accomplish every task, it is highly recommended to use the User manual of the integration devices.

6.- If needed develop certain commands or set of commands to perform the tasks required.

7.- Set a routine where the communication between the client and server is established.

8.- Set a security routine to ensure safety of the integration.

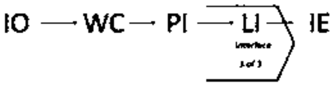
9.- In base with the general requirements establish an actions sequence (Include routines 7 and 8).

10.- Identify the input data that will be needed in order to make the integration work (once that the integration has been done).

11.- What should be done with the information? Should it be saved or printed?

12- In base with the general requirements of the integration and with points 10 and 11 develop a Human Machine Interface.

Figure 4.29: Tasks to be realized in order to establish the logic sequence

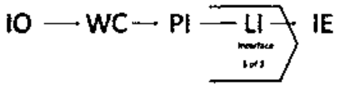


4.4.3 Interface

In order to satisfy the components for a HMI according to ISO 9241 (Flores,2005) and in base with Betsabe (2007), Haley (2005) and Polzer (2006) there are certain features that a HMI must include:

- Consider an hierarchical and sequential accommodation of controls and indicators, according to the relevance and use supposed.
- Consider a visual indicator to show that the HMI is working (real time or simulation)
- It is recommended to group variables of the controls in order to organize the information that will be shown in the display.
- Consider a spot to store information that could be used by posterior users.
- Consider at all time the user environment and define standards
  - Use high contrast for primary data (process variables).
  - Secondary or support data should be smaller.
  - Alarms should be easy to recognize using bright colors.
  - Be consistent in the use of fonts, symbols, abbreviations, etc.

Betsabe (2007) proposes a flow diagram to follow when designing and implementing a HMI, the next flow diagram is a modification of it by leading it to the VRT integration:



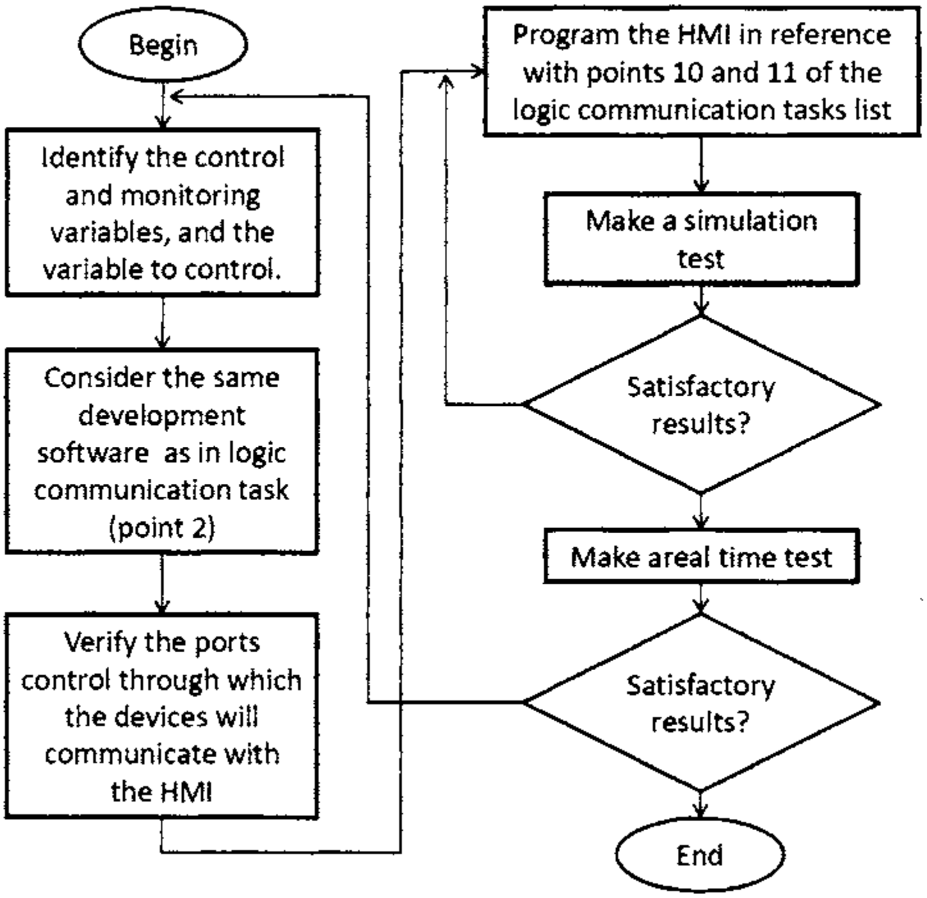


Figure 4.30: Flow diagram to design a HMI

4.5 Evaluation

There are as many as evaluation process as applications of integrated VRT, following the main purpose of this research and in base with the literature review, the most common and **generic** evaluation methods are going to be presented next:

4.5.1 Positioning error

Pauli (2001) proposes an alternative calibration system, and evaluates the positioning error in millimeters in base with the number of training samples, the next figure shows the results table from his research:

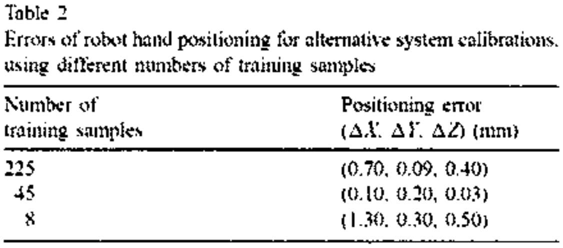


Table 2
Errors of robot hand positioning for alternative system calibrations, using different numbers of training samples

Number of training samples	Positioning error (ΔX, ΔY, ΔZ) (mm)
225	(0.70, 0.09, 0.40)
45	(0.10, 0.20, 0.03)
8	(1.30, 0.30, 0.50)

Figure 4.31: Positioning error evaluation chart

As it can be inferred the positioning evaluation consists in the measure of the difference between the coordinates of the robot desired and the real robot coordinates. This difference should be monitored as the same time that another variable is modified (in the Pauli (2001) example, the number of training samples is the variable modified).

Pena–Cabrera (2005) evaluates the poses obtained by his method and compares them with the error limits, the next figure shows the chart presented in his research:

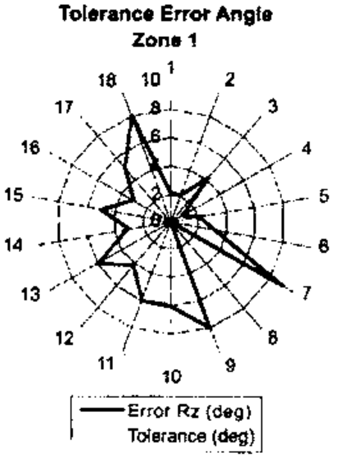
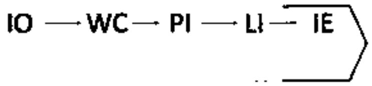


Figure 4.32: Pose error evaluation chart



4.5.2 Efficiency (number of correct cycles)

Another way to evaluate this kind of systems is the efficiency in the number of correct cycles perspective. This method has not to evaluate the whole cycle necessarily, it can also evaluate just certain task(s) of it. Balakrishnan (2000) evaluates the efficiency of his proposal in base with the number of correct assemblies, the results table is shown next:

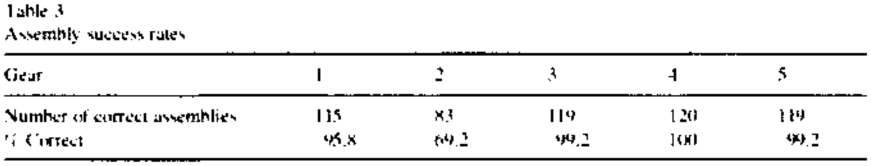


Table 3
Assembly success rates

Gear	1	2	3	4	5
Number of correct assemblies	115	83	119	120	119
% Correct	95.8	69.2	99.2	100	99.2

Figure 4.33: Efficiency evaluation chart

The efficiency can be also represented in percentage terms, this indicator can be easily obtained by the next formula:

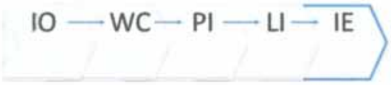
$$efficiency\,(\%) = \left(\frac{\#\,of\ correct\ tasks}{\#\,of\ total\ tasks\ monitored}\right)(100) \tag{4.20}$$

4.5.3 Connection time

Suarez (2009) evaluates the time of connection of robot–control unit module and of camera–control unit module, this evaluation considers a histogram chart building, the variable that is plotted is the connection time and it is obtained through the next formula:

$$Connection\ time\ =\ Confirmation\ time\ -\ Ask\ time \tag{4.21}$$

The following chart is presented by Suarez (2009) where the connection time is evaluated:



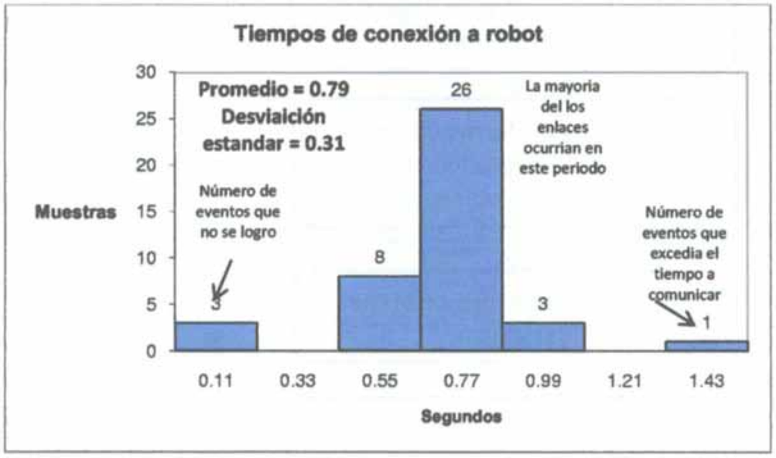


Figure 4.34: Connection time evaluation chart

4.5.4 General consideration

The evaluation of the VRT system is highly determined by the application demands, usually the evaluation will be focused to measure how much is the system accomplishing the initial requirements (Number of assemblies, number of pieces detected and handled, number of errors in the process, accuracy of the motion, welding or painting quality, etc.) but in a general form all this factors will be defined in the accuracy of the robot location in base with vision information, so the connection time, number of correct tasks, positioning error and pose error are the main variables that should be considered in order to evaluate the system because these variables are the ones that directly represent the accuracy of the system.

4.6 Methodology resume

With the intention of summarizing or representing in just one chart all the methodology previously developed, the most important steps or guidelines of the VRT integration were included in the next diagram:

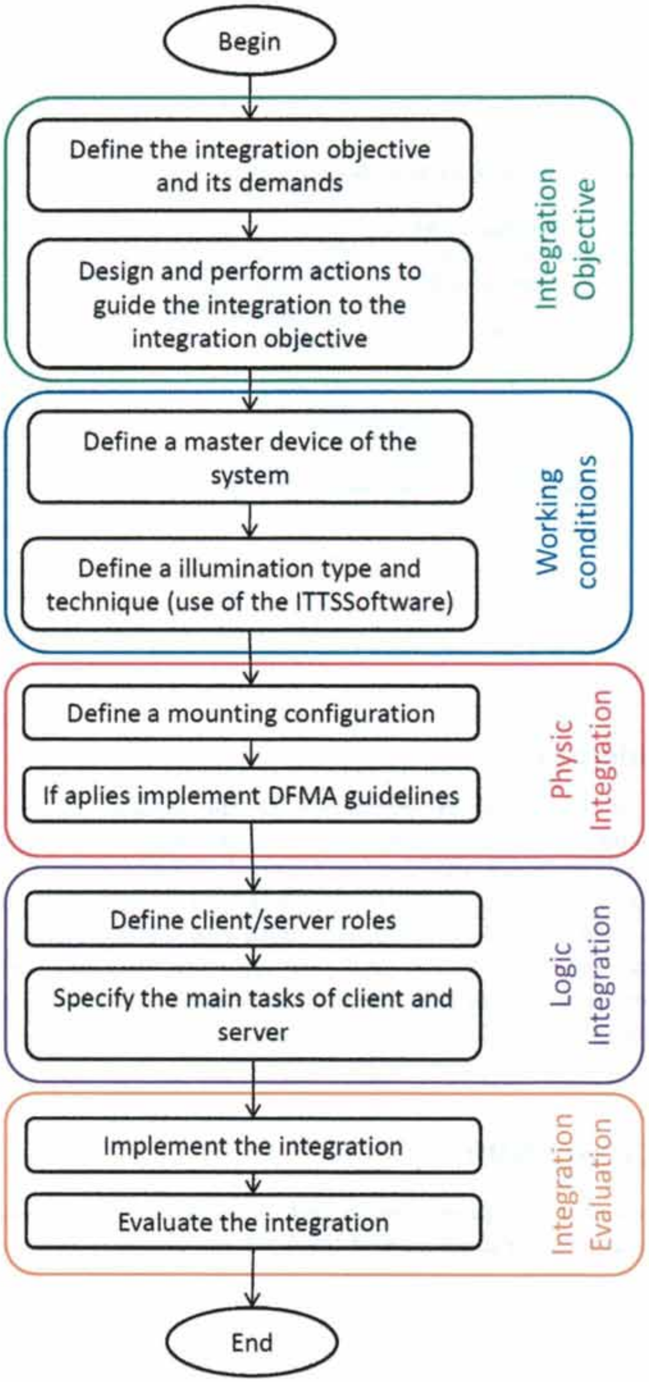


Figure 4.35: Resume of the Methodology proposed

The following charts describe in a general form the performance of the methodology, where each of the steps considered is analyzed and its inputs and outputs are identified, in order to consider the methodology application as successful, the obtention of all the outputs is necessary.

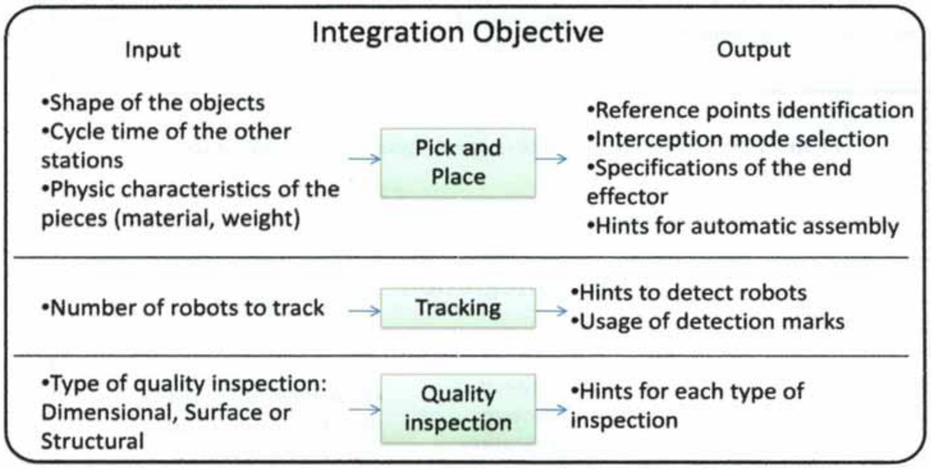


Figure 4.36: Inputs and outputs of the "Integration Objective" stage

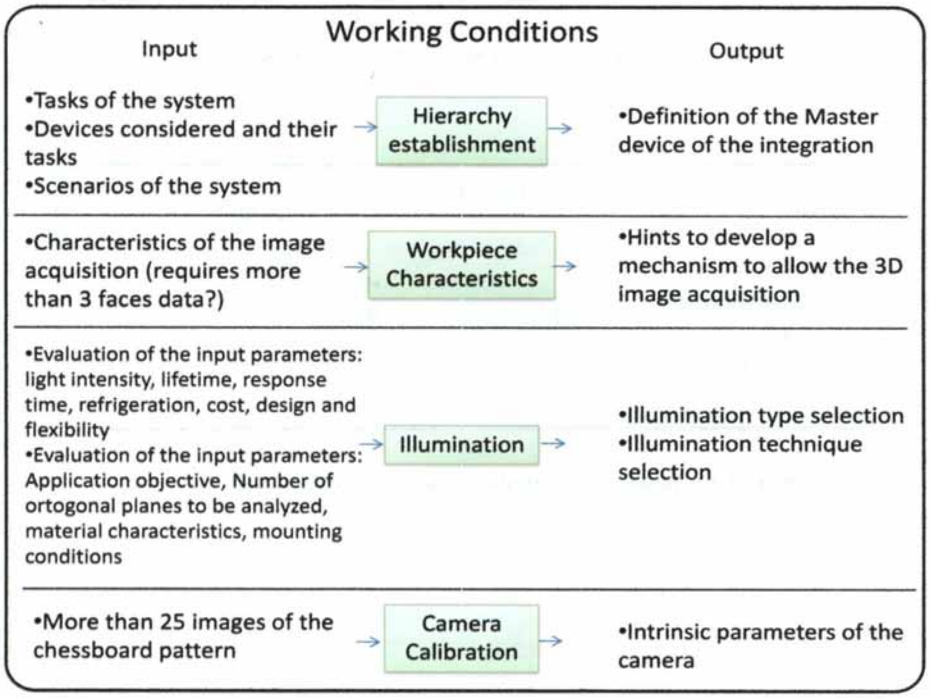


Figure 4.37: Inputs and outputs of the "Working Conditions" stage

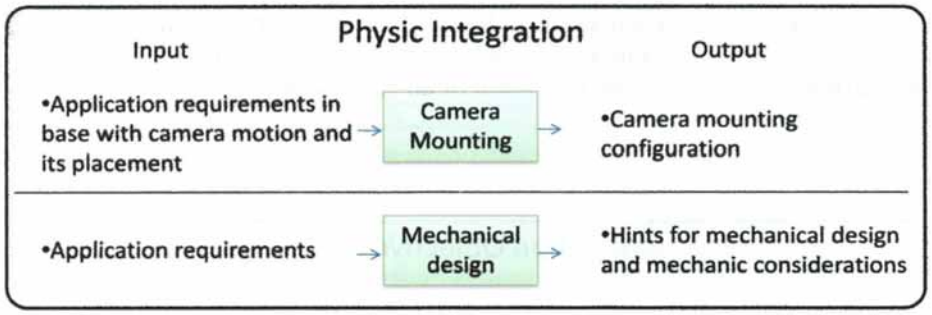


Figure 4.38: Inputs and outputs of the "Physic Integration" stage

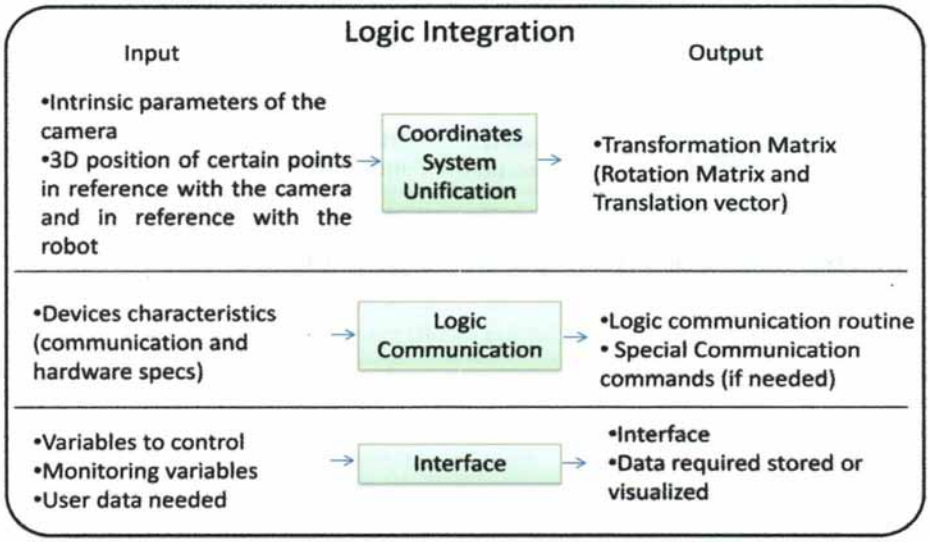


Figure 4.39: Inputs and outputs of the "Logic Integration" stage

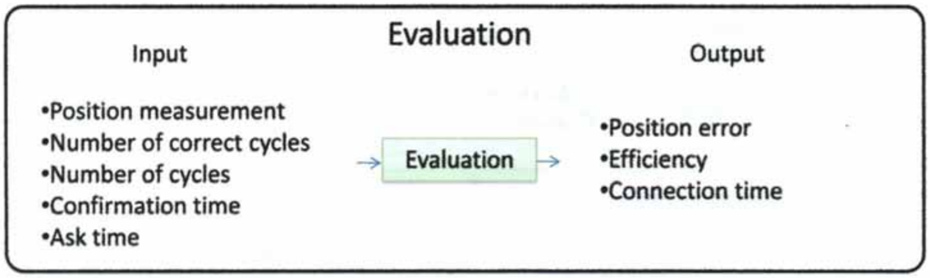


Figure 4.40: Inputs and outputs of the "Evaluation" stage

Chapter 5

Case of study: Implementation of a Integrated VRT system

5.1 General description

In order to validate the methodology proposed, a VRT integration will be set, as it has been said a lot of manufacturing process can be accomplished by the integration of VRT, in this specific experiment a packaging process will be implemented, for this purpose the packaging operation is considered as a Pick and Place operation, this consideration is based in the following statements: Pick and Place operations (assembly, packaging and material handling) consist on the execution of the following tasks: detect objects, identify objects, locate objects, grasp objects and dispose objects in a desired position. So in base with the last idea, the implementation will be set in base with the pick and place recommendations and hints.

As it has been mentioned in section 3.1 the main components of a integrated robotic vision system are three: Robotic device, Vision System and a Control Unit, in this experiment the following devices are considered:

Robotic device

Figure 5.1: Robotic arm FANUC LR Mate 200iB

The robotic device used in this experiment is a FANUC^{TM}robotic arm of LR Mate 200iB serie, and its features are:

- 6 axes
- 5 Kgs of payload
- H- Reach of 700mm
- Repeatibility of 0.04mm

The control unit for this robot is FANUC^{TM}System R-J3iB Mate which features are:

- Multi-processor architecture (separate motion and communication) with realtime
- Built-in Ethernet
- 100 Base-TX/10 Base-T with RJ-45 connector
- Three RS-232 ports (one can be configured as RS-422)

Vision System

Figure 5.2: XBOX Kinect Camera

The vision system considered in this research is a non-industrial 3D camera (Kinect from Microsoft), this camera is compound by:

- VGA camera.- also called RGB camera has an output video of 30 frames per second with 24bpp image format in which every pixel is represented by a value of Red, Green and Blue color, each value has a 8-bit size (from 0 to 255). The resolution of the image provided by the camera is 640x480 pixels.(Crawford,2010)
- Depth sensor.- the output of this sensor is in VGA resolution (640x480 pixels) with 11-bit depth values, so there are 2,048 levels of sensitivity in order to get the depth of the image, this sensor is the combination of an infrared laser projector and a monochrome CMOS sensor, the CMOS sensor also works at 30 frames per second.(Crawford,2010)

The Kinect camera supports communication via USB port.

Control Unit

Figure 5.3: Sony vaio VGN-CS320J

The control unit utilized is a Personal Computer, and its features are:

- Processor: Intel(R) Core (TM)2Duo at 2.1 GHz
- 4GB of Installed memory (RAM)
- 64 bit Operating System

5.2 Experimental set–up

5.2.1 Objective of the VRT Integration

The objective of the case of study is to implement a VRT integration in order to perform a Packaging Operation (From now on: "PO"), this objective implies the use of a vision system to determine the localization (orientation and position) of the objects to be packaged, once this information is obtained the next step consists on the generation of certain robot tasks that should be transferred to the robot controller and executed by the robotic arm.

5.2.2 Description of the Packaging Operation

The "PO" of this research remains in the placement of 9 pieces in their packaging spot, the orientation and position of this figures is undefined, the vision system must identify this items and provide to the control unit the 3D coordinates of each piece, the next figure shows the pieces available for this operation:

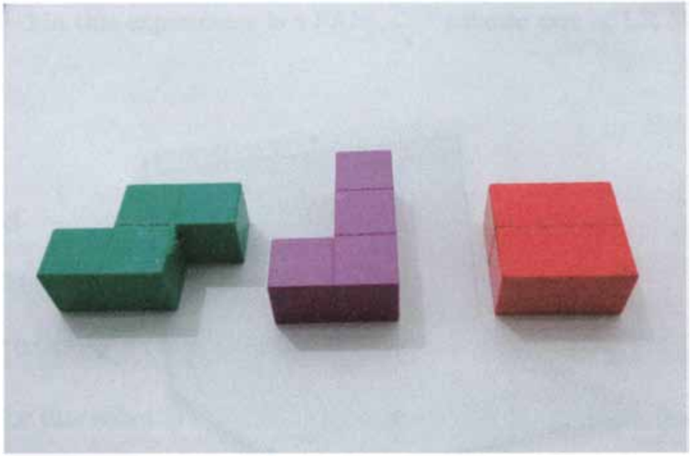


Figure 5.4: Objects to be packaged

From now on the pieces will be identified by a figure type, for red pieces: Type 1, for purple pieces: Type 2 and for green pieces: Type 3.

In order to perform a correct package an end effector is needed, a two-finger gripper is selected due the shape of the objects to be grasped, as all the objects are compound by squared units, the grasping of this kind of borders do not need any special fingers or another type of end-effector.

5.3 Integration development

5.3.1 Integration Objective

As it has been mentioned before, the integration objective remains on a packaging operation (considered as a “Pick and Place” operation), so the next considerations are leaded to this manufacturing operation.

Scope of the integration

The mentioned PO considers:

- The package of 7 pieces which lay on a base facing to it always in the same configuration. The detection and identification of the pieces was implemented to work when the piece lays in a horizontal and vertical form, but the limitation of the illumination devices available and the existing shadows when the pieces lay vertically on the base did not allow the utilization of this two configurations.
- The package of 7 pieces divided in two sub-packages (Package 1: 2 red pieces, 1 purple piece and 1 green piece, Package 2: 1 red piece, 1 purple piece and 1 green piece), this split is needed due the size of the gripper, and the working zone of the robot, if the 7 pieces lay on the base, collision between the gripper and some piece(s) is very probable, and due the location of the robot, objects can not be placed far away from the base because it will be a position not reachable for the robot or it will go through singularity configurations.

Workpiece Detection

The workpieces that are considered in the integration are the following:

- 3 pieces Type 1 (Red)
- 2 pieces Type 2 (Purple)
- 2 pieces Type 3 (Green)

In base with the methodology proposed the first step is to identify a reference point, in this experiment the number of reference points needed is just one due the 3D mapping that the camera provides, so in order to detect the workpieces the following reference points are selected:

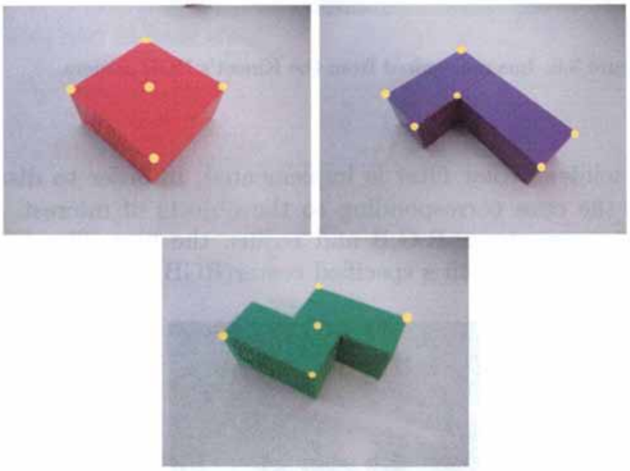


Figure 5.5: Reference points identification

Due the rectangular characteristics of the objects the reference points selected correspond to the corners and centers of gravity of each piece, in that way this reference points will also provide information for further steps of the methodology.

Workpiece Identification

There is not any commercial software for Kinect camera manipulation, so the communication with kinect camera is implemented by the use of programming libraries (DLL'S) the library used is "freenect.dll", this library includes all code necessary to activate, initialize and communicate with the Kinect camera. This library is OS independent and there is an application file which is compatible with C, C++, C#, VB.Net, Java, and Python.(Open Kinect, 2011)

In base with the previous knowledge of the C# platform and knowing that the Vision Frameworks are usually compatible with .NET framework, this language is chosen in order to develop the application which will allow the communication with Kinect hardware.

The workpiece(s) identification is implemented via well-known algorithms in machine-vision, in order to apply those algorithms a computer-vision Framework is needed, the framework used is AForge.NET which "is an open source for NET framework designed for developers and researchers in the fields of Computer Vision and Artificial Intelligence - image processing, neural networks, genetic algorithms, fuzzy logic, machine learning, robotics, etc." (Kirillov, 2011)the process followed to identify the object and its coordinates is:

1. **Camera image.-** In order to perform a workpiece identification, the main resource is the image of the scene which contains de object of interest, this image was obtained through the usage of libraries which allows the utilization of the Kinect Camera with a PC, more information about the programming and image processing is provided in the section 5.3.1

Figure 5.6: Image acquired from the Kinect's RGB camera

2. **Color filter.-** An Euclidean color filter is implemented, in order to dismiss all the objects which colors are not the ones corresponding to the objects of interest. This type of filter can be adjusted by 4 parameters: R,G,B and radius, the filter filters pixel which color is inside/outside of a RGB sphere with a specified center(RGB values) and radius.

Figure 5.7: Result image of the Euclidean filter

3. **Blobs identification.-** Once that the image has been processed the objects can be identified by the use of separated objects (Blobs), this identification can be configured in order to detect only the blobs that accomplish certain dimension specifications.

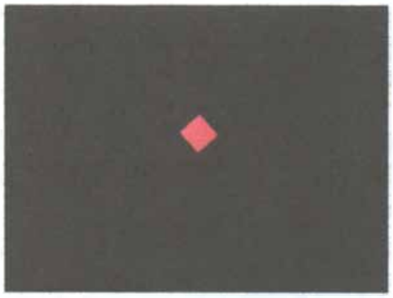


Figure 5.8: Images of each blob detected

Picking workpiece up

1. **2D data (x,y and θ)**

The AFORGE framework in its BlobsCounter class provides the localization(x,y) of all blobs detected, this Center of gravity corresponds to the objects localization (in pixels) in the image coordinate system.

(a) **For squared and rectangular objects(Type 1)**

In order to get the orientation of squared objects, the use of an algorithm which detects quadrilateral corners is implemented, and once that the position (x,y) of each of the corners is obtained, and in base with:

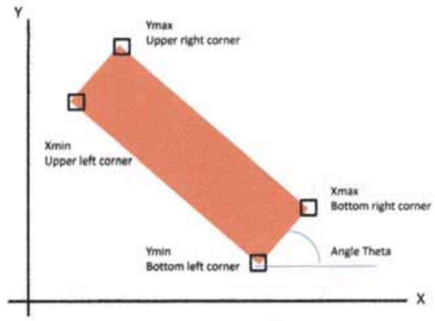


Figure 5.9: Orientation of a rectangular object

The following procedure is done:

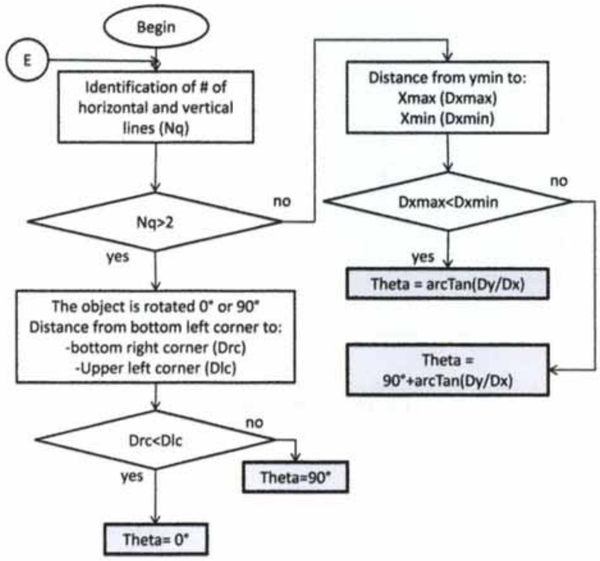


Figure 5.10: Rotation angle computation for squared pieces

The next figure shows an image example that is used as input data for the Rotation angle computation procedure.

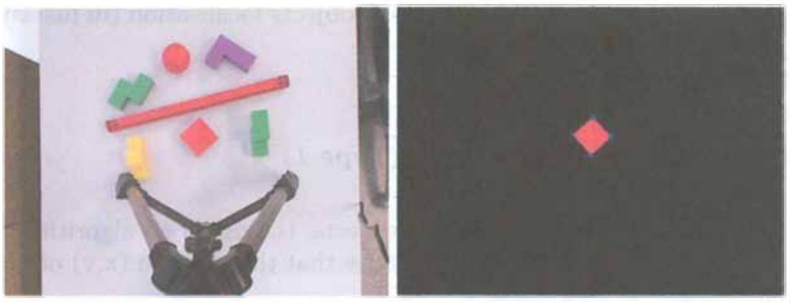


Figure 5.11: RGB image (left) and Input image(right) for red pieces

(b) **For non-squared pieces(Type2)**

The main features of this type of figures are the corners with maximum distance to the center of gravity, with these two corners the angle can be obtained, due the shape of the figure, the angle obtained with these corners will be the same for quadrant I and III, also if the figure is in quadrant II and IV, in order to distinguish this difference the center of gravity is considered as the orientation feature.Considering:

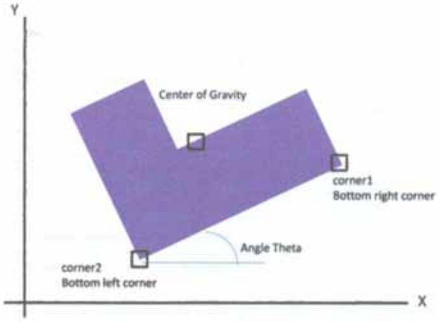


Figure 5.12: Rotation angle

The following procedure is done:

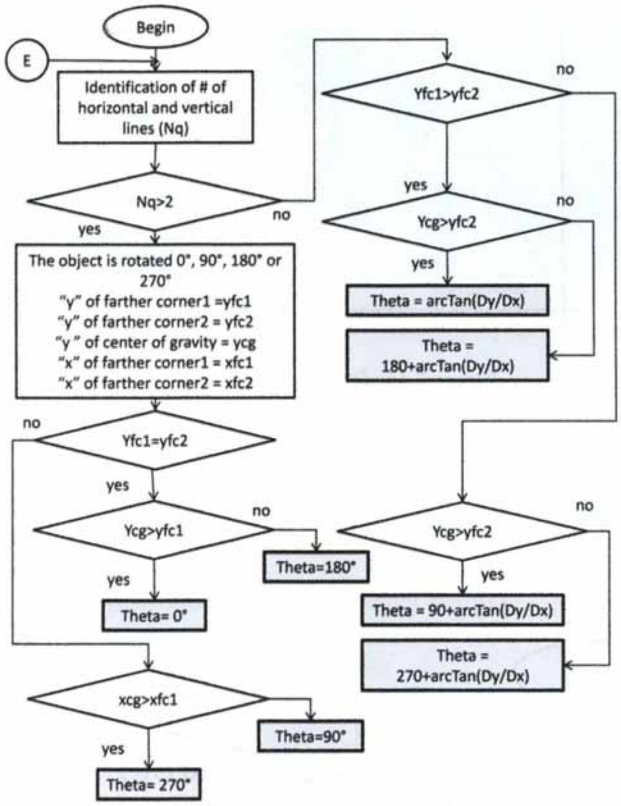


Figure 5.13: Rotation angle computation for purple pieces

The next figure shows images examples that are used as input data for the Rotation angle computation procedure.

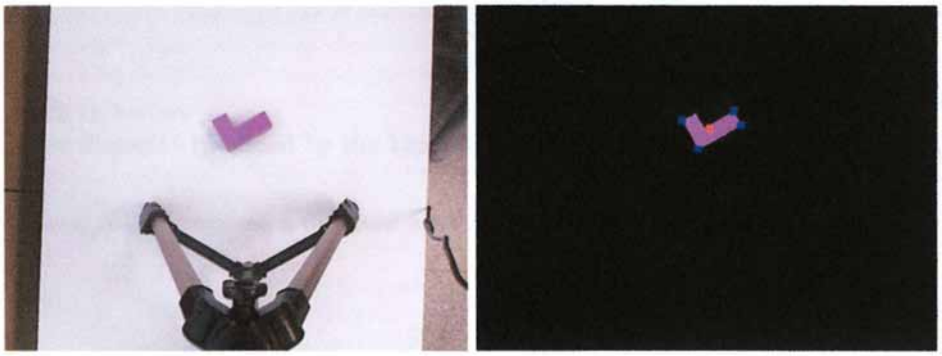


Figure 5.14: RGB images (left) and input images(right) for purple pieces

Green figures(Type 3)
The main features of this type of figure are the corners with maximum distance to the center of gravity, with these two corners it can be inferred in which quadrant is the figure (I or II), in order to compute the angle one of this corner is utilized an the other one is the one of the right side of the center of gravity,with these analysis and considering:

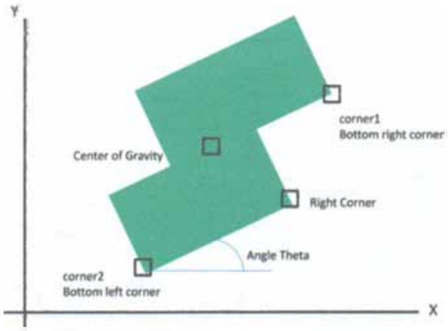


Figure 5.15: Rotation angle

The following procedure is done:

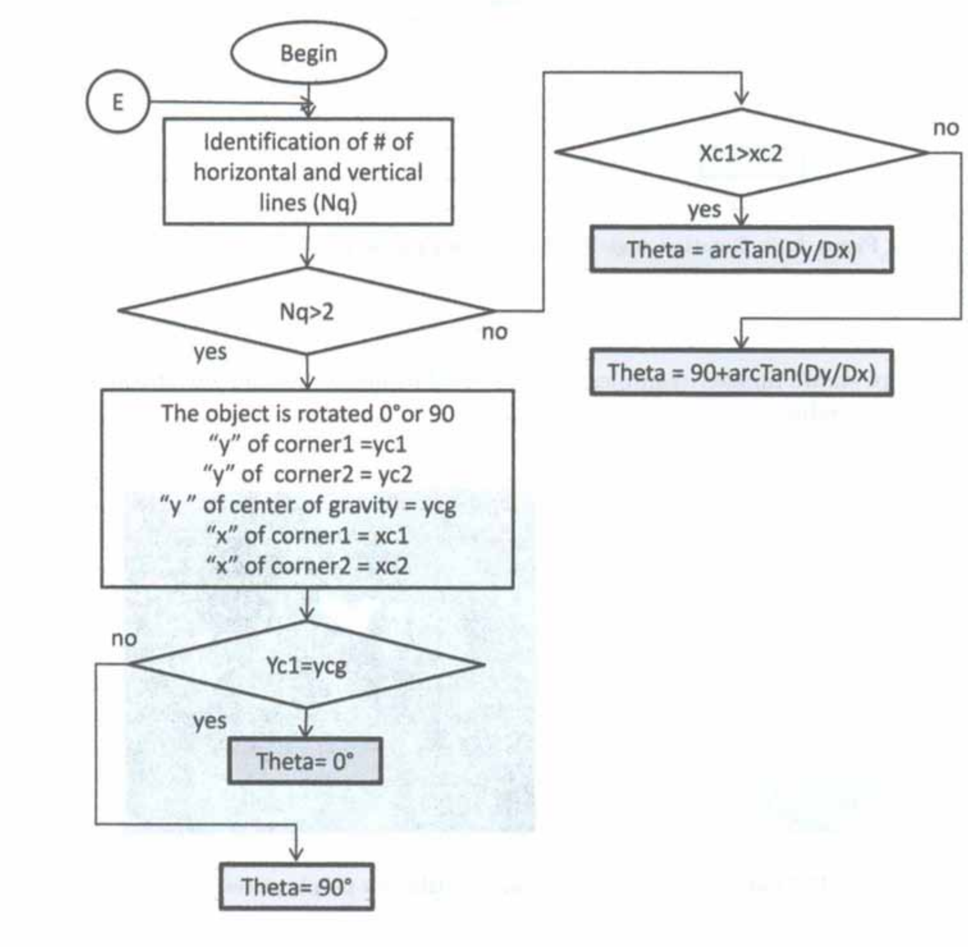


Figure 5.16: Rotation angle computation for green pieces

The next figure shows an image example that is used as input data for the Rotation angle computation procedure.

Figure 5.17: RGB image(left) and input image(right) for green pieces

2. **3D data (z)**

As it has been said, the kinect software provides a 11-bit value for the depth of each one of the 640x480 pixels of the image, this value also called disparity does not represents a depth in a unity of any Unities System. There are some researchers that have used the kinect camera and a opensource library (like freenect) for research projects and in base with experimental data have gotten approximations to represent the depth value in Centimeters or meters, for example, Burrus(2010) proposes the next formula:

$$D = \frac{100}{-0.00307 * Rd + 3.33} \tag{5.1}$$

Where:
D= depth in centimeters
Rd= Raw disparity provided by the kinect (11-bit value)

Magnenat(2010) proposes the next approximation:

$$D = 0.1236 * tan(\frac{Rd}{2842.5} + 1.1863) \tag{5.2}$$

Where:
D= depth in meters
Rd= Raw disparity provided by the kinect (11-bit value)

Also Crock(2011) proposes a different approximation to the depth value:

$$D = \frac{34800}{1091.5 - Rd} \tag{5.3}$$

Where:
D= depth in centimeters
Rd= Raw disparity provided by the kinect (11-bit value)

In order to verify that the value provided by kinect and that the formulas are consistent with real dimensions, the measurement of the depth of certain pixel is performed, a line with marks each 10 centimeters was adhered to the ground, the camera was located at the beginning of the line (0 cms) and a box was moved along the line at the same time that the depth of certain pixel was calculated.

The pixel used in this experiment was the pixel in (320,480) coordinates.

The object was moved every mark and centered in the image in order to make sure that the depth value belongs to the box position.The following images shows this experiment:

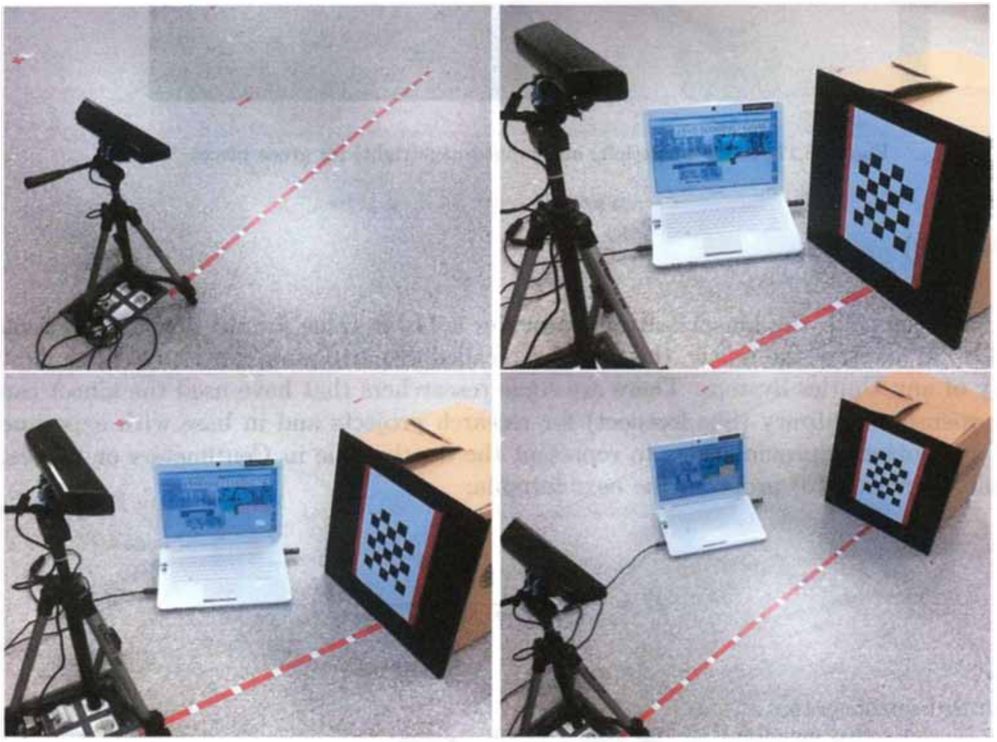


Figure 5.18: Verification of depth calculation

The approximation used in this research is the one of Sthepane Magnenat (2010) based on the own experiments this is the formula which provides more accurate results.

5.3.2 Working conditions

Hierarchy Establishment

Following the procedure proposed in section 4.2.1:

1. The main task of the integration consists on the package of the Product.
2. The device which performs the main task is the robotic arm.
3. Scenarios of the integration
    - Communication between PC and Robot.
    - Communication between PC and Vision System.
    - Objects localization.
    - Path Generation.
    - Objects grasping.
    - Path execution.

4. for:
    - Robot there are 2 scenarios.
    - Vision system there are 1 scenarios.
    - Control Unit there are 3 scenarios
5. The main task is not performed by the device which performs the major quantity of tasks. But as the control unit can not be the master device the fact that the Master Device is the Robot is inherent.

As a conclusion it can be inferred that the Robotic Arm is the Master device, so the integration type of this integration is a Robot assisted by Machine Vision (In base with Espinosa (2006)).

That fact means that the Robot performs the main task (package) but it needs information from the vision system(objects localization) in order to successfully execute the main task.

Workpiece Characteristics

As it can be seen in the figures 5.11, 5.14 and 5.17 the shape of the objects is mostly prismatic, and as every type of figure has de same color and shape they can be easily identified, in base with this considerations any mechanism is needed in order to relocate the camera or the objects.

Illumination

In this part of the section the Illumination Type and Technique Selection Software (ITTSS), for the first step the input data is:

The Parameters ordered by importance:

1. **Orthogonal Planes.-** This is the most important parameter due the use of a 3D camera the image analysis remains on more than one orthogonal plane.
2. **Application Objective.-** This parameter is the second in importance, because the light system should provide the adequate environment for the packaging operation.
3. **Mounting Conditions.-** As there is enough space for the integration this parameter has less importance than others.
4. **Material Characteristics.-** In base with the objects of the package presented in section 5.2.2 it can be inferred that the material Characteristics is not a high-importance parameter due the usage of objects which surface is not translucent or excessively shiny.

Options selected for each parameter:

1. **Orthogonal Planes.-** Two or more orthogonal planes
2. **Application Objective.-** Objects identification (which is the base of the packaging operation).
3. **Mounting Conditions.-** Free space.
4. **Material Characteristics.-** Shiny Objects.

The result for Illumination technique is Diffuse-Lighting:

Importance from 1 to 4	Parameters
1	Orthogonal Planes
4	Material Characteristics
3	Mounting Conditions
2	Aplication Objective

Recommended techniques: Diffuse Light

Diffuse Light

It is based on the impact of light on the piece but in different distributed directions, this illumination type is used almost in the same situations as the direct light but with soft or bright surfaces

Figure 5.19: Illumination technique result

For Light Type selection the Input data and the results are:

Assign an importance factor for each parameter factor should be 3,6 or 9 the press Go"

	Importance
Light intensity	5
Lifetime	5
Response time	3
Refrigeration	5
Cost	9
Design Flexibility	6

The recommended light type for your application is the one with higher ranking

Halogen Lamps
Rating: 153

Xenon Lamps
Rating: 180

Fluorescent Lamps
Rating: 39

LEDs Lamps
Best option! 210

Figure 5.20: Light type result

Camera Calibration

Each of the cameras in kinect hardware (RGB and DEPTH camera) has different characteristics so the images do not match exactly, in order to correct this gap the calibration of each camera is needed, for this process the GML Calibration Tool Box was utilized.

1. **For RGB camera:**

   In order to perform this process a chessboard pattern was utilized with squares of 30x30 mms, the images of the pattern were taken from the suggested positions in Figure 4.25, the next images show the image acquisition:

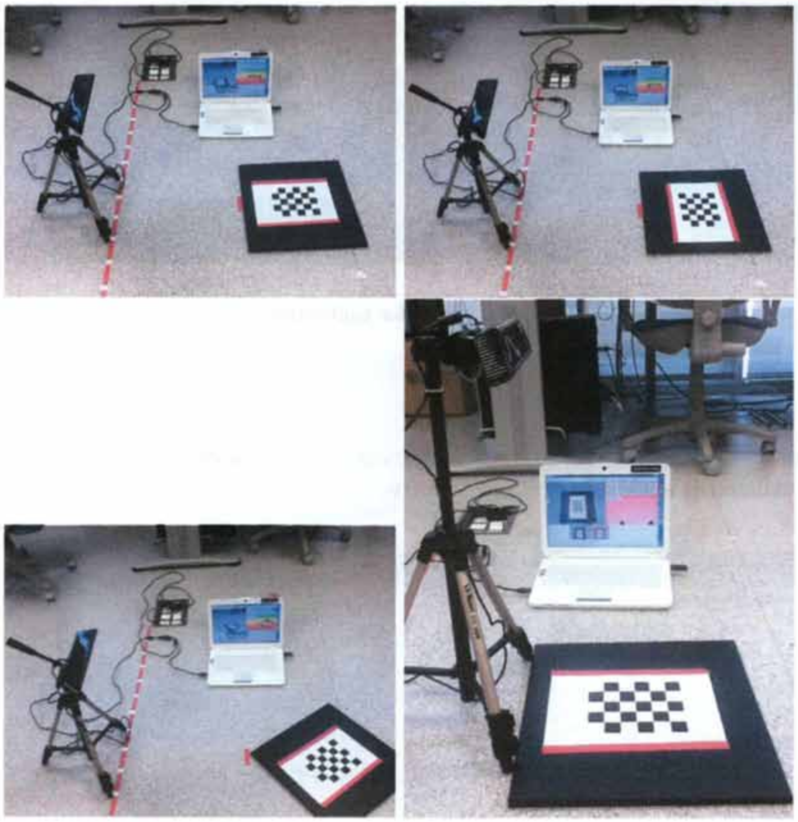


Figure 5.21: Images acquisition

Some of the images acquired from different position are presented:

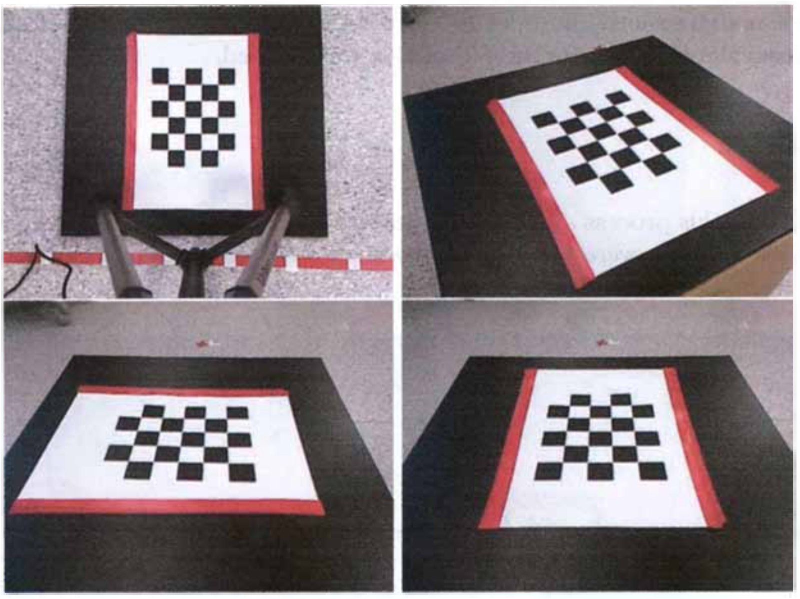
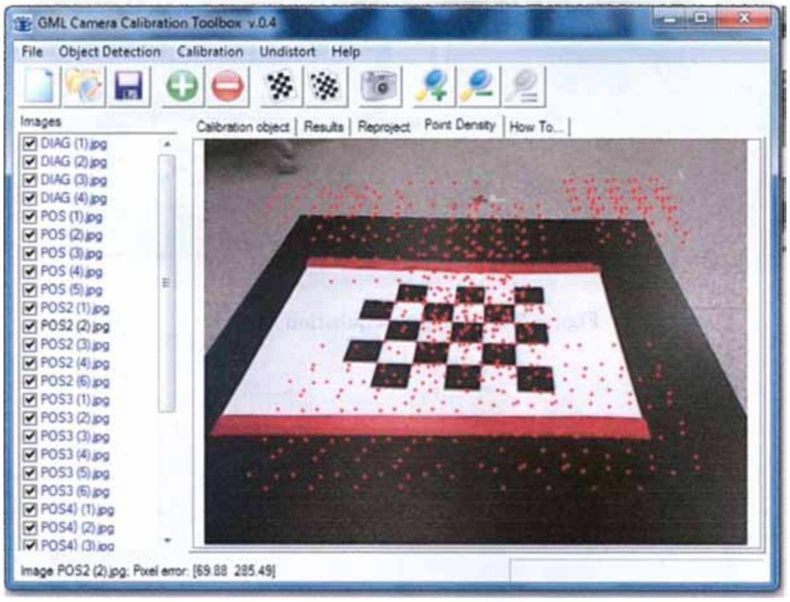


Figure 5.22: Images acquired for calibration process

There are 33 images in total, these images were loaded in the GML interface, the Point Density of the images utilized is shown in the next figure:



☑ DIAG (1).jpg
☑ DIAG (2).jpg
☑ DIAG (3).jpg
☑ DIAG (4).jpg
☑ POS (1).jpg
☑ POS (2).jpg
☑ POS (3).jpg
☑ POS (4).jpg
☑ POS (5).jpg
☑ POS2 (1).jpg
☑ POS2 (2).jpg
☑ POS2 (3).jpg
☑ POS2 (4).jpg
☑ POS2 (6).jpg
☑ POS3 (1).jpg
☑ POS3 (2).jpg
☑ POS3 (3).jpg
☑ POS3 (4).jpg
☑ POS3 (5).jpg
☑ POS3 (6).jpg
☑ POS4) (1).jpg
☑ POS4) (2).jpg
☑ POS4) (3).jpg

Image POS2 (2).jpg; Pixel error: [69.88 285.49]

Figure 5.23: Point Density of the image set

The corner detection and the calibration process were executed, obtaining the following camera matrix:

$$A_{RGB} = \left\{ \begin{array}{ccc} \alpha_x & \gamma & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{array} \right\} = \left\{ \begin{array}{ccc} 529.21508098293293 & 0 & 328.94272028759258 \\ 0 & 525.56393630057437 & 267.48068171871557 \\ 0 & 0 & 1 \end{array} \right\} \tag{5.4}$$

2. **For IR camera:**
   As the IR camera is the one that computes the depth of the image, this camera also has to be calibrated, in order to perform the same process done with the RGB camera, the use of an halogen lamp is needed in order to modify the image acquired by the IR camera, this correction is shown in the next figures:

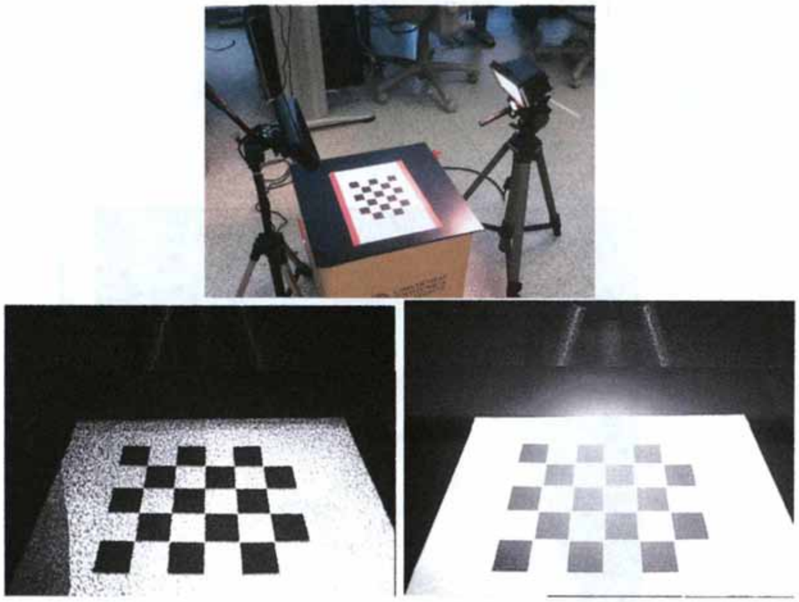


Figure 5.24: IR image modification

Once that the images obtained by the IR camera are suitable to be processed by a Calibration Tool Box, the detection and calibration process is executed, the image acquisition was performed as similar as possible as the RGB image acquisition, some of the images loaded in the GML tool Box are presented:

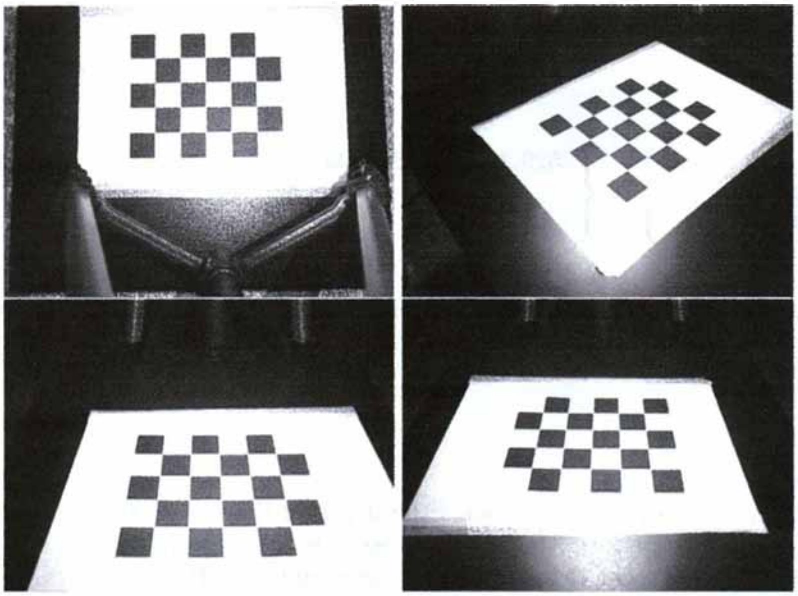
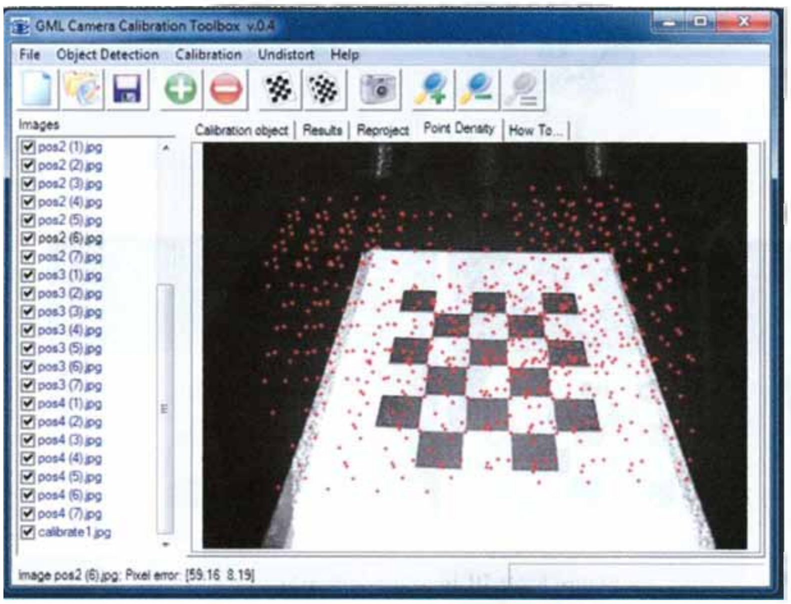


Figure 5.25: Images acquired for calibration process

The Point density is presented in the next figure:



```
Image pos2 (6).jpg: Pixel error: [59.16 8.19]
```

Figure 5.26: Point density of the image set

As a result of the calibration process the following matrix is obtained:

$$A_{IR} = \left\{ \begin{array}{ccc} 594.21434211923247 & 0 & 339.30780975300314 \\ 0 & 591.04053696870778 & 242.73913761751615 \\ 0 & 0 & 1 \end{array} \right\} \tag{5.5}$$

5.3.3 Physic Integration

Camera Mounting

In base with the procedure shown in Figures 4.26 an 4.27, the following data is generated:

Application requirements:
The camera should be able to:

- Identify the objects of interest.
- Get the localization of the object in 3D (x,y,z and θ).

The robot should be able to:

- Receive and execute the path generated in base with the camera's information.

Following the procedure:

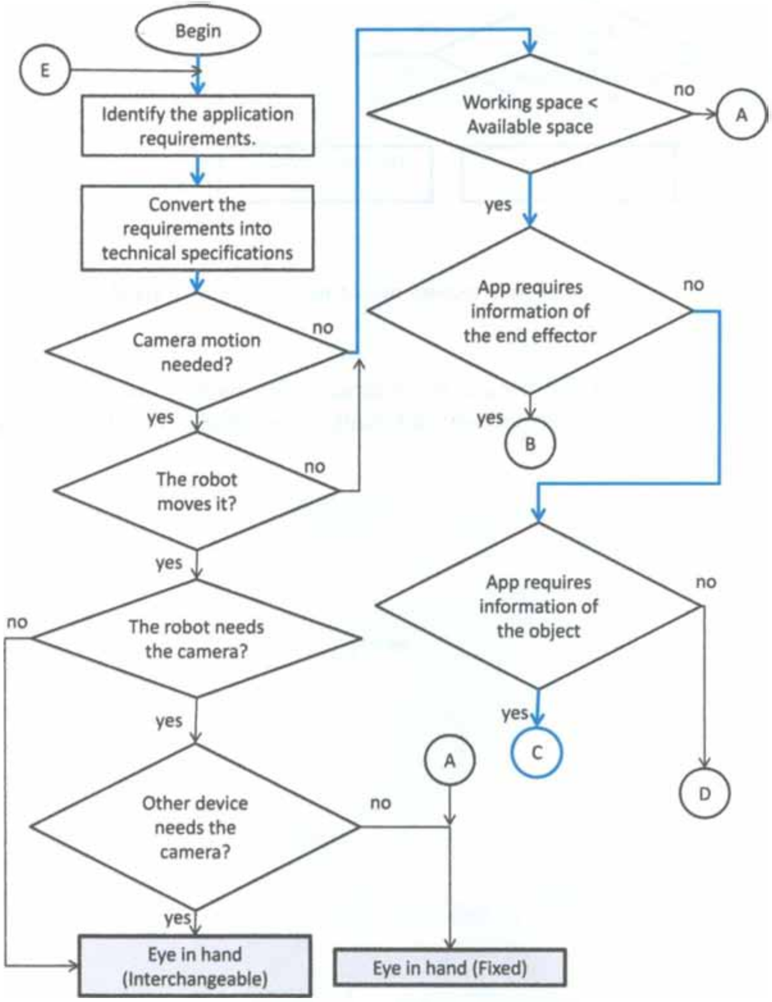


Figure 5.27: Selection of the camera mounting configuration

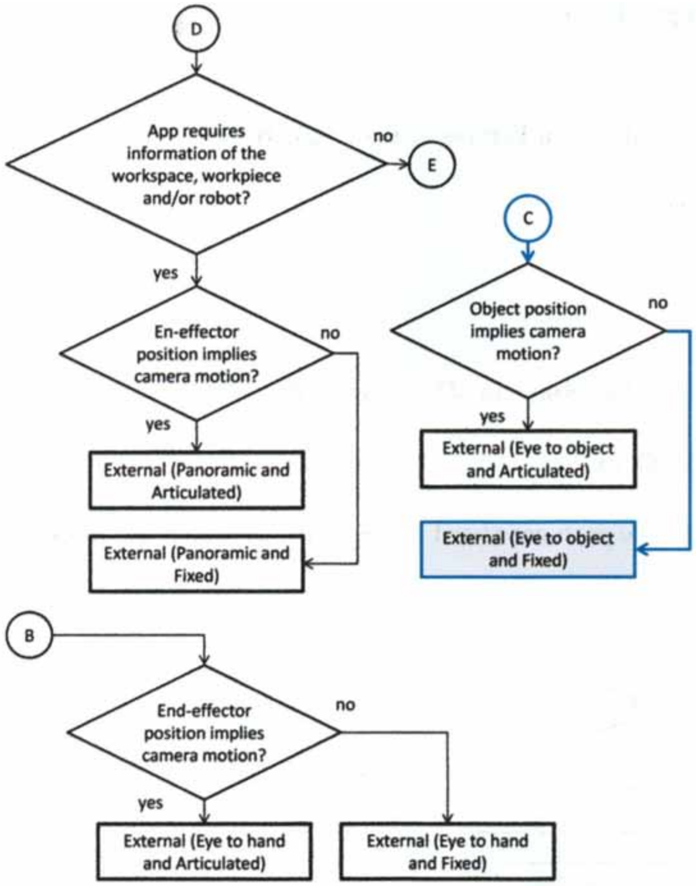


Figure 5.28: Selection of the camera mounting configuration (cont.)

The result obtained is that the best camera mounting configuration for this application is External-eye to object and fixed to a previously determined position. The next image shows this mounting:

Figure 5.29: Camera Mounting Implementation

Mechanical Design

In this VRT integration the mechanic design was reduced to the camera mounting, and as there is a wall mount for the kinect hardware available it was used in the following configuration:

Figure 5.30: Mechanical mounting of the camera

5.3.4 Logic Integration

RGB and IR images matching

As it has been mentioned before the images of the RGB and IR camera do not match, this is caused by the different placement of the cameras and the small differences between them, this situation is shown in the next figures:

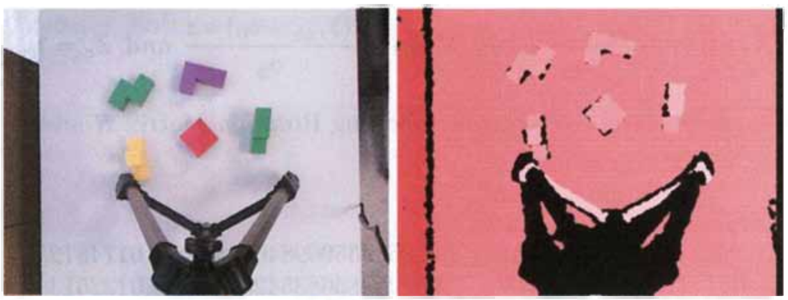


Figure 5.31: RGB(left) and Depth(right) image provided by Kinect

Figure 5.32: Images Overlap

Once that one object is detected, its Center of Gravity is found and saved, but this data only contains the position in x and y axis, in order to obtain the z value the method described in 5.3.1

specifically the use of Equation 5.2 is needed, the Raw Disparity value (*Rd*) belongs to a certain pixel, this pixel should match with the Center of Gravity one (x,y), but as the figure 5.26 shows the look for the *Rd* in the depth image will be incorrect if any correction procedure is done.

In order to find the IR pixel which corresponds to the RGB pixel, the following procedure is done:

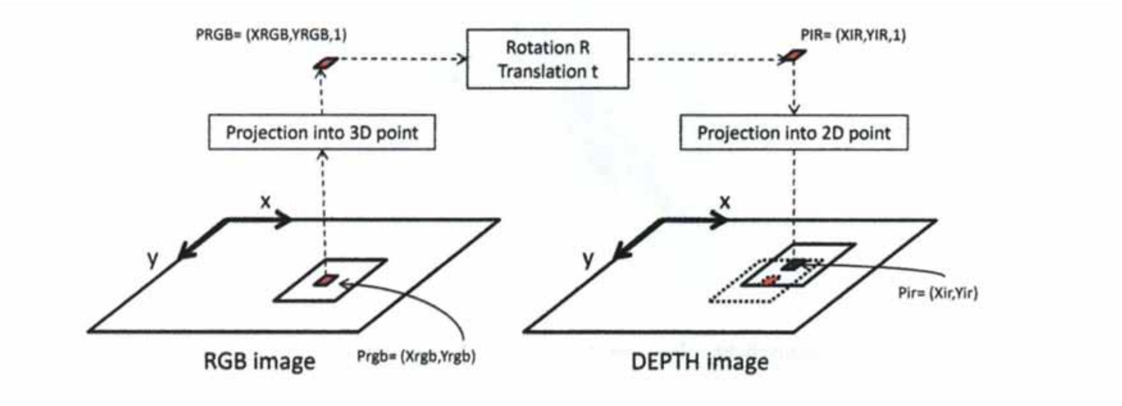


Figure 5.33: Images Gap

1. Find the Center of Gravity of the object detected.
2. Project this point(pixel) into 3D space.

$$X_{RGB} = \frac{(X_{rgb} - u_0) * z}{\alpha_x}, \; Y_{RGB} = \frac{(Y_{rgb} - v_0) * z}{\alpha_y} \; and \; Z_c = 1 \tag{5.6}$$

3. Apply the transformation in base with the following Rotation Matrix R and the translation vector t

$$^{RGB}R_{IR} = \left\{ \begin{array}{ccc} .99984628826577793 & .0012635359098409581 & -.017487233004436643 \\ -.0014779096108364480 & .99992385683542895 & -.012251380107679535 \\ .017470421412464927 & .012275341476520762 & .99977202419716948 \end{array} \right\} \tag{5.7}$$

$$^{RGB}t_{IR} = \left\{ \begin{array}{c} .03 \\ -.01 \\ 0 \end{array} \right\} \tag{5.8}$$

Parameters R and t were empirically obtained by Burrus(2010) and have worked fine for Fisher(2010) research. The coordinates transformation is:

$$\begin{pmatrix} X_{IR} \\ Y_{IR} \\ Z_{IR} \\ 1 \end{pmatrix} = \begin{pmatrix} ^{RGB}R_{IR} & ^{RGB}t_{IR} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} X_{RGB} \\ Y_{RGB} \\ Z_{RGB} \\ 1 \end{pmatrix} \tag{5.9}$$

4. Project the new 3D point into the IR image

$$X_{ir} = \frac{(X_{IR}) * \alpha_x}{z = 1} + u_0, \; Y_{ir} = \frac{(Y_{IR}) * \alpha_y}{z = 1} + v0 \tag{5.10}$$

5. Get the Raw Disparity Rd of the point(pixel) projected.

$$Z_{ir} = D = 0.1236 * tan(\frac{Rd}{2842.5} + 1.1863) \tag{5.11}$$

The implementation of this procedure is shown in the next figure, where the Center of Gravity $p(x_{rgb}, y_{rgb})$ is the pixel identified with AFORGE blobcounter class (see section 5.3.1), this pixel is marked in the figure below with a red square. If the pixel's coordinates are translated to the Depth image this point will not correspond with the center of gravity of the object and in some cases it will address to a another object's pixel, once that the calibration procedure has been executed the pixel's address that will be considered is the one provided by the coordinates transformation, this pixel is marked in the image below with a blue square. Through this procedure the data obtained of the object's localization (x,y,z) does not include any mismatching error.

Figure 5.34: RGB to IR transformation

The following pictures show the image transformation of every pixel of the depth image provided by the kinect hardware:

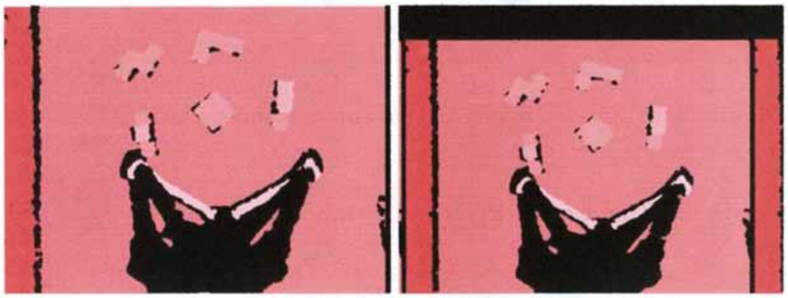


Figure 5.35: Original depth image (left) and calibrated depth image(right)

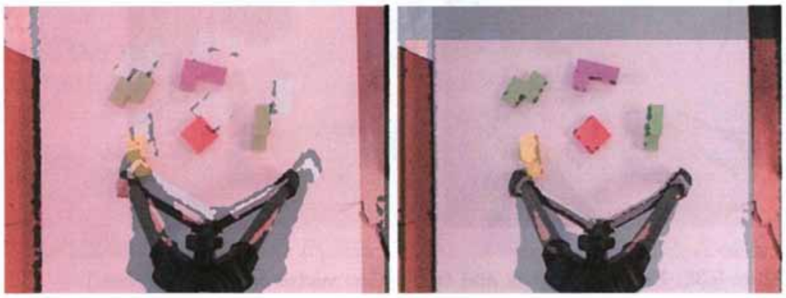


Figure 5.36: Images overlap before calibration(left) and after calibration(right)

Coordinate System Unification

1. **Camera's intrinsic parameters.-** In order to obtain the Camera matrix the GML Calibration Tool Box is utilized (see section 5.3.2) the cameras matrixes are:

$$A_{RGB} = \left\{ \begin{array}{ccc} 529.21508098293293 & 0 & 328.94272028759258 \\ 0 & 525.56393630057437 & 267.48068171871557 \\ 0 & 0 & 1 \end{array} \right\} \tag{5.12}$$

$$A_{IR} = \left\{ \begin{array}{ccc} 594.21434211923247 & 0 & 339.30780975300314 \\ 0 & 591.04053696870778 & 242.73913761751615 \\ 0 & 0 & 1 \end{array} \right\} \tag{5.13}$$

2. Rotation Matrix and Translation Vector computation

   (a) Fix the camera in the working position of the application.

Figure 5.37: Rotation matrix and translation vector calculation

   (b) Place a mark in the Tool Center Point of the robot.

Figure 5.38: Rotation matrix and translation vector calculation (cont.)

   (c) Move the robot to a known position $Pr(Xr, Yr, Zr)$

(d) Identify the mark (which corresponds to the "Tool Center Point") with the camera, and get the coordinates of this point in reference with the image coordinates system $p(x, y, z)$. The TCP utilized in this research is the 3D point located in the middle of the fingers of the gripper, in order to configure this TCP in the control unit of the robot the procedure presented on the supplier manual was executed (This procedure is shown in Appendix C). In order to get the $p(x, y, z)$ data, a known-object was located in the new TCP (previously set) as it is shown in the next figure:

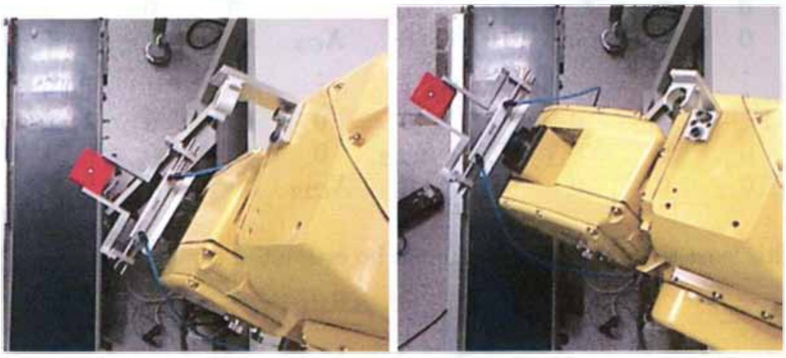


Figure 5.39: Tool Center Point location

With this, the center of gravity of the Red Object corresponds to the TCP of the robot. The transformation parameters which describe the relationship between the original TCP and the new TCP are obtained from the Teach Pendant of the Robot and presented next:

$$\begin{Bmatrix} tx \\ ty \\ tz \end{Bmatrix} = \begin{Bmatrix} -2.3 \\ 1.2 \\ 129.9 \end{Bmatrix} \tag{5.14}$$

$$\begin{Bmatrix} w \\ p \\ r \end{Bmatrix} = \begin{Bmatrix} 123.4 \\ -75.9 \\ 72.0 \end{Bmatrix} \tag{5.15}$$

(e) Transform the localization coordinates of the mark detected into a new localization in reference with Camera's coordinates system $Pc(Xc, Yc, Zc)$ (Equations 4.3).

(f) Compute the Rotation matrix R and the translation vector t.

The computation requires the $P(X_R, Y_R, Z_R)$ and $P(X_C, Y_C, Z_C)$ of at least four points, this data is obtained repeating the steps (d),(e) and (f) one time for each point. The first approach of this step considered 12 points in the 3D space in base with the data available and considering least squares optimization, the solution of the equation system is of the $Ax = b$ type and can be represented as follows:

$$\begin{bmatrix} X_{C1} & Y_{C1} & Z_{C1} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & X_{C1} & Y_{C1} & Z_{C1} & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & X_{C1} & Y_{C1} & Z_{C1} & 0 & 0 & 1 \\ X_{C2} & Y_{C2} & Z_{C2} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & X_{C2} & Y_{C2} & Z_{C2} & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & X_{C2} & Y_{C2} & Z_{C2} & 0 & 0 & 1 \\ X_{C3} & Y_{C3} & Z_{C3} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & X_{C3} & Y_{C3} & Z_{C3} & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & X_{C3} & Y_{C3} & Z_{C3} & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_{C12} & Y_{C12} & Z_{C12} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & X_{C12} & Y_{C12} & Z_{C12} & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & X_{C12} & Y_{C12} & Z_{C12} & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{XX} \\ R_{XY} \\ R_{XZ} \\ R_{YX} \\ R_{YY} \\ R_{YZ} \\ R_{ZX} \\ R_{ZY} \\ R_{ZZ} \\ t_X \\ t_Y \\ t_Z \end{bmatrix} = \begin{bmatrix} X_{R1} \\ X_{R2} \\ X_{R3} \\ X_{R4} \\ X_{R5} \\ X_{R6} \\ X_{R7} \\ X_{R8} \\ X_{R9} \\ X_{R10} \\ X_{R11} \\ X_{R12} \end{bmatrix} \tag{5.16}$$

In base with least squares the solution can be estimated as:

$$A^T A x' = A^T b \quad \therefore \quad x' = (A^T A)^{-1} A^T b \tag{5.17}$$

This computation was performed in MATLAB, and the result for x' is:

$$x' = \begin{bmatrix} -0.0156 \\ 0.9924 \\ -0.0753 \\ 1.0028 \\ 0.0191 \\ -0.0077 \\ 0.0153 \\ -0.0337 \\ -0.9311 \\ -0.0923 \\ -0.5727 \\ 0.5278 \end{bmatrix} \quad \therefore \quad {}^{C}R_{R} = \begin{bmatrix} -0.0156 & 0.9924 & -0.0753 \\ 1.0028 & 0.0191 & -0.0077 \\ 0.0153 & -0.0337 & -0.9426 \end{bmatrix} \text{ and } {}^{C}t_{R} = \begin{bmatrix} -0.0923 \\ -0.5727 \\ 0.5278 \end{bmatrix} \tag{5.18}$$

The mean squared error of this solution is $e = 0.0062mts$

In order to optimize this solution the layoffs points were excluded from the matrix computation, obtaining the following results for six 3D points:

$${}^{C}R_{R} = \begin{bmatrix} -0.0004 & 1.0026 & -0.0627 \\ 1.0076 & 0.0203 & -0.0146 \\ -0.0052 & -0.0445 & -0.9311 \end{bmatrix} \text{ and } {}^{C}t_{R} = \begin{bmatrix} -0.1025 \\ -0.5670 \\ 0.5168 \end{bmatrix} \tag{5.19}$$

Obtaining a mean squared error of $e = 0.00087mts$

Logic Communication

1. **Communication protocol selection**

In base with features described in section 5.1.1, the following protocols are selected:

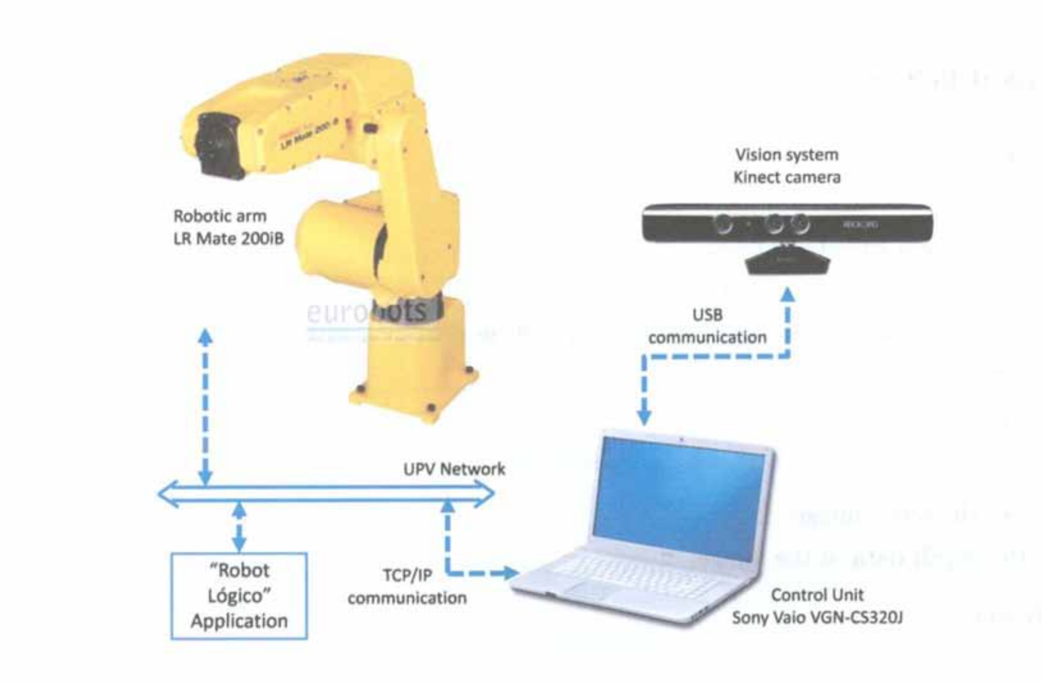


Figure 5.40: Communication Protocols in VRT system

2. **Programming software selection**

As mentioned above the protocols that need to be used in order to communicate the Control Unit with the Robot and the camera are the Universal Serial Data (USB) and the Ethernet protocol at application level (TCP/IP), the use of the USB port can be enabled by the use of a dynamic library, on the other hand, Microsoft Visual Studio supports the TCP client and TCP server classes which can be implemented in order to establish a TCP/IP communication.

In order to continue working with the same platform in which the communication with kinect has been done, the language visual C# is utilized, the main advantages provided by this platform are:

- Use of unsafe code which is necessary to access to the depth data provided by the Kinect Camera.
- Use of pointers which are variables that read and store memory addresses of certain objects.
- Use of dynamic libraries which allow the USB port manipulation.
- Use of TCP client and server classes to establish the TCP/IP communication.
- This language is compatible with .NET Framework which is also compatible with Computer Vision frameworks like Aforge and OpenCV.

3. **Communication roles identification.**

   The Control unit (PC) will perform as a client and both devices the robot and the camera will perform as servers, the servers will be listening to any connection, this connection will be initialized by the Client and will ask for certain information depending on the task that will be executed.

4. **Main task definition.**

   For the control unit:

   - Ask connection with the camera
   - Ask connection with the robot
   - Process image and obtain objects data (x, y, z, θ) in reference with the coordinate system of the robot

   For the Camera:

   - Acquire the scene image.
   - Get the depth data of the image.

   For the Robot:

   - Go to home position.
   - Grasp objects to be packaged.
   - Execute routine.

5. **Commands identification**

   The following figure shows the way that the integration is intended to interact, in base with the tasks or orders shown the identification and/or generation of the adequate commands of each device system is needed.

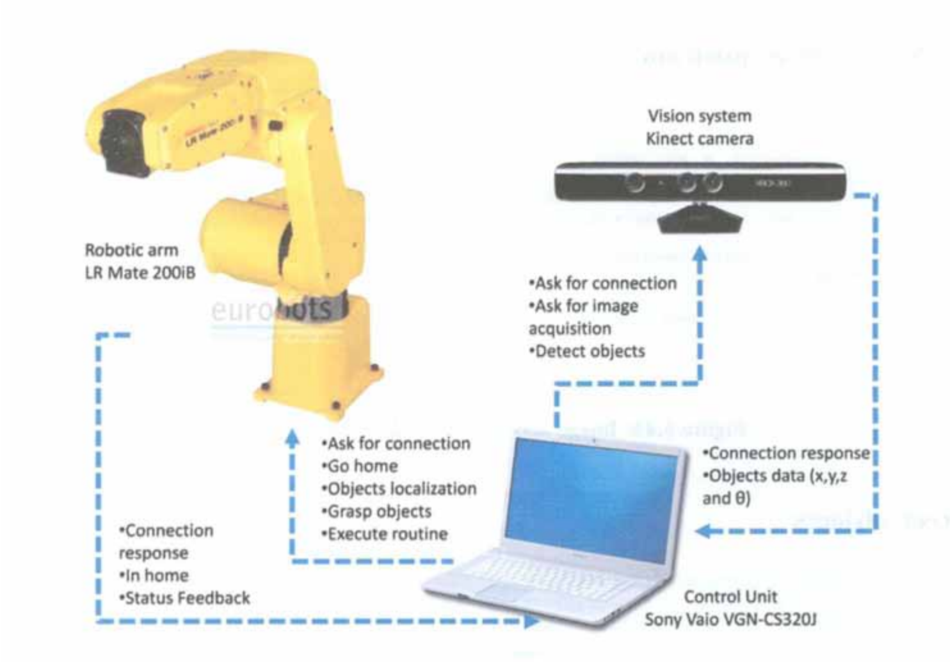
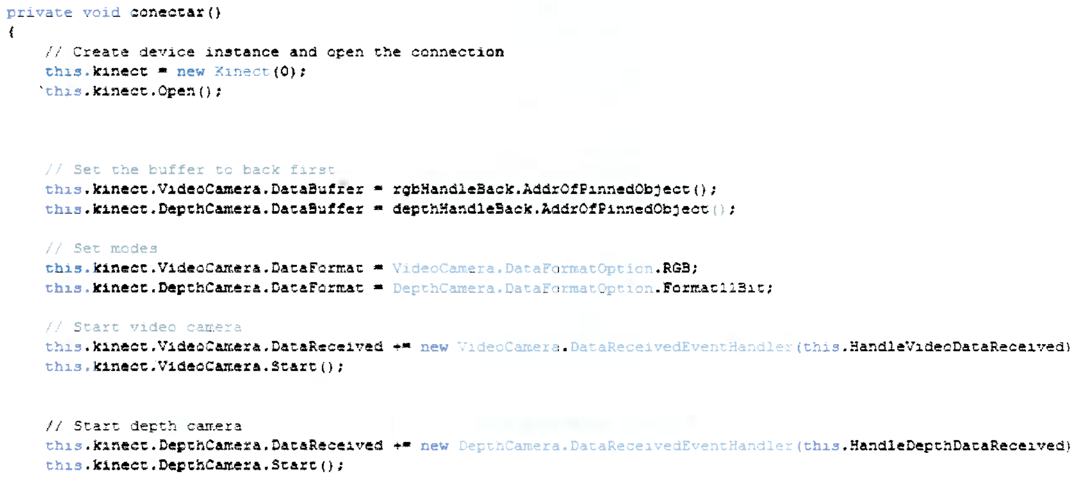


Figure 5.41: Data transfer between the elements of the VRT

In base with the last figure the following methods were developed in base with commands supported by the “freenect.dll” and the “robot system”:

Communication between the Vision System and the Control Unit

- **Ask for connection**



```
private void conectar()
{
    // Create device instance and open the connection
    this.kinect = new Kinect(0);
    this.kinect.Open();



    // Set the buffer to back first
    this.kinect.VideoCamera.DataBuffer = rgbHandleBack.AddrOfPinnedObject();
    this.kinect.DepthCamera.DataBuffer = depthHandleBack.AddrOfPinnedObject();

    // Set modes
    this.kinect.VideoCamera.DataFormat = VideoCamera.DataFormatOption.RGB;
    this.kinect.DepthCamera.DataFormat = DepthCamera.DataFormatOption.Format11Bit;

    // Start video camera
    this.kinect.VideoCamera.DataReceived += new VideoCamera.DataReceivedEventHandler(this.HandleVideoDataReceived)
    this.kinect.VideoCamera.Start();


    // Start depth camera
    this.kinect.DepthCamera.DataReceived += new DepthCamera.DataReceivedEventHandler(this.HandleDepthDataReceived)
    this.kinect.DepthCamera.Start();
```

Figure 5.42: Function to connect the Kinect device with the PC

- **Ask for image acquisition**

```
public Bitmap copiarimagen()
{
    Rectangle rect = new Rectangle(0, 0, 640, 480);
    this.bmp = new Bitmap(640, 480);
    BitmapData data = bmp.LockBits(rect, ImageLockMode.WriteOnly, System.Drawing.Imaging.PixelFormat.Format24bppRgb);
    GL.ReadPixels(0, 0, 640, 480, OpenTK.Graphics.OpenGL.PixelFormat.Bgr, PixelType.UnsignedByte, data.Scan0);
    bmp.UnlockBits(data);
    bmp.RotateFlip(RotateFlipType.RotateNoneFlipY);
    pictureBox3.Image = bmp;

    return bmp;

}
```

Figure 5.43: Image acquisition of the scene

- **Detect objects**

```
//blobcounter configuration
this.blobsimage = ima_gray;//image should be grayscale
int miw = Convert.ToInt16(trackBar9.Value);
int maw = Convert.ToInt16(trackBar10.Value);
int mih = Convert.ToInt16(trackBar11.Value);
int mah = Convert.ToInt16(trackBar12.Value);
this.getblobs = new BlobCounter();
this.getblobs.MinWidth = miw;
this.getblobs.MaxWidth = maw;
this.getblobs.MinHeight = mih;
this.getblobs.MaxHeight = mah;
getblobs.FilterBlobs = true;
getblobs.ObjectsOrder = ObjectsOrder.Area;
getblobs.ProcessImage(blobsimage);
int nofblobs = getblobs.ObjectsCount;
//this.cornersinfo = new IntPoint[nofblobs][];
this.blobsinfo = new IntPoint[nofblobs][];
this.angulos = new double[nofblobs];
this.cogs = new IntPoint[nofblobs];
this.cogscalib = new IntPoint[cogs.Count()];
this.prof = new double[nofblobs];
label28.Text= Convert.ToString(nofblobs);

//arrays initialization (blobs and images)
Blob[] blobs = getblobs.GetObjectsInformation();
Bitmap[] blobsimagesarray = new Bitmap[nofblobs];

//fill the images array
for (int i = 0; i < nofblobs; i++)
{
    getblobs.ExtractBlobsImage(blobsimage, blobs[i], true);
    blobsimagesarray[i] = blobs[i].Image.ToManagedImage();
}
```

Figure 5.44: Objects detection implementation

Communication between the Robotic arm and the Control Unit

In order to establish this communication, a previously developed application was utilized, this application is named "*Robot Logico*" (Logic Robot), the detailed explanation can be found in the Olmos (2009) research, in resume this application receives XML instructions in a determined protocol, and if this instructions are correct it converts the set of instructions in a new set of commands but in the native language of the robot (KAREL programming), once that the programm is built, the application send it to the control unit of the robot in order to execute the motion required.

- **Go to home position** A known home position was defined and implemented in a C# method, the next code belongs to this instruction:

```
public void homefigure()
{

    string text = "<instruction name=\"MOVEJ\" time=\"1\"><frame name=\"NAME\" x=\"-83\" y=\" -430\" z=\"-160\" alfa=\"-5
    string canal = "INSTRUCTION_FANUC";
    byte[] bytes = new byte[text.Length];
    System.Text.ASCIIEncoding encoding1 = new System.Text.ASCIIEncoding();
    bytes = encoding1.GetBytes(text);

    try
    {
        RTSCoreAdapter.Singleton.Write(canal, bytes, 0, RTSCoreAdapter.Singleton.TicksNow());
    }
    catch (Upv.Disca.FSA.FSACreateAdapterException)
    {

    }
```

Figure 5.45: Home position method

- **Objects localization** Once that the objects data is obtained in base with the information provided by the Kinect camera, a method is called in order to grasp each one of the objects, this method is presented in the next figure:

```
public void graspobject(double[][] coord, int numero)
{
    double[][] coorde= new double[this.TETRIS.Count()][];
    coorde = coord;
    //verify that the coordinates are ina the safe working zone
    if ((coord[numero][0] < 0.041 & coord[numero][0] > -0.400) & (coord[numero][1] < -0.300 & coord[numero][1] > -0.640)
    {
        coorde[numero][0] = coord[numero][0] * 1000;
        coorde[numero][1] = coord[numero][1] * 1000;
        coorde[numero][2] = (coord[numero][2]+.150) * 1000;

        double t = 0;

        switch (numero)
        {
            case 0:
                t = coord[numero][3] + 22.0001;
                break;
            case 1:
                t=coord[numero][3] + 22.0001;
                break;
            case 2:
                t = coord[numero][3] + 22.0001;
                break;
            case 3:
                t = coord[numero][3] + 22.0001+90;
                break;
            case 4:
                t=coord[numero][3] + 22.0001+90;
                break;
            default:
                break;
        }
  string x = Convert.ToString(coord[numero][0]);
  string y = Convert.ToString(coord[numero][1]);
  string z = Convert.ToString(coord[numero][2]);
  string theta = Convert.ToString(t);

  char[] delimit = new char[] { '.' };
  string[] cadenas = x.Split(delimit);
  string[] cadenas2 = y.Split(delimit);
  string[] cadenas3= z.Split(delimit);
  string[] angulo = theta.Split(delimit);

  string xfanuc = cadenas[0] + "," + cadenas[1];
  string yfanuc = cadenas2[0] + "," + cadenas2[1];
  string zfanuc = cadenas3[0] + "," + cadenas3[1];
  string tfanuc = angulo[0] + "," + angulo[1];

  string text = "<instruction name=\"MOVEJ\" time=\"1\"><frame name=\"NAME\" x=\""+xfanuc+"\" y=\""+yfanuc+"\" z=\""
  string canal = "INSTRUCTION_FANUC";
  byte[] bytes = new byte[text.Length];
  System.Text.ASCIIEncoding encoding1 = new System.Text.ASCIIEncoding();
  bytes = encoding1.GetBytes(text);

  try
  {
      RTSCoreAdapter.Singleton.Write(canal, bytes, 0, RTSCoreAdapter.Singleton.TicksNow());
  }
  catch (Upv.Disca.FSA.FSACreateAdapterException)
  {

  }
```

Figure 5.46: Grasp Objects method

- **Objects Package** Once that the object has been grasped the robot executes a routine where it leaves the object in a certain position, depending on the type of figure that is being processed.

```
public void leavefigure(int figure)
{
    string text;
    switch (figure)
    {
        case 0:
            text = "<instruction name=\"MOVEJ\" time=\"1\"><frame name=\"NAME\" x=\"-496,8\" y=\"-240\" z=\"-230\" alfa=\"
            break;
        case 1:
            text = "<instruction name=\"MOVEJ\" time=\"1\"><frame name=\"NAME\" x=\"-496,8\" y=\"-240\" z=\"-200\" alfa=\"
            break;
        case 2:
            text = "<instruction name=\"MOVEJ\" time=\"1\"><frame name=\"NAME\" x=\"-496,8\" y=\"-200,7\" z=\"-170\" alfa=
            break;
        case 3:
            text = "<instruction name=\"MOVEJ\" time=\"1\"><frame name=\"NAME\" x=\"-459\" y=\"-264\" z=\"-203\" alfa=\"-4
            break;
        case 4:
            text = "<instruction name=\"MOVEJ\" time=\"1\"><frame name=\"NAME\" x=\"-540\" y=\"-264\" z=\"-203\" alfa=\"-4
            break;
        default:
            text="error";
            break;
    }

    string canal = "INSTRUCTION_FANUC";
    byte[] bytes = new byte[text.Length];
    System.Text.ASCIIEncoding encoding1 = new System.Text.ASCIIEncoding();
    bytes = encoding1.GetBytes(text);

    try
    {
        RTSCoreAdapter.Singleton.Write(canal, bytes, 0, RTSCoreAdapter.Singleton.TicksNow());
    }
    catch (Upv.Disca.FSA.FSACreateAdapterException)
    {
    }
}
```

Figure 5.47: Objects assembly method

6. Communication sequence

The following diagram shows the ideal communication sequence for this integration:

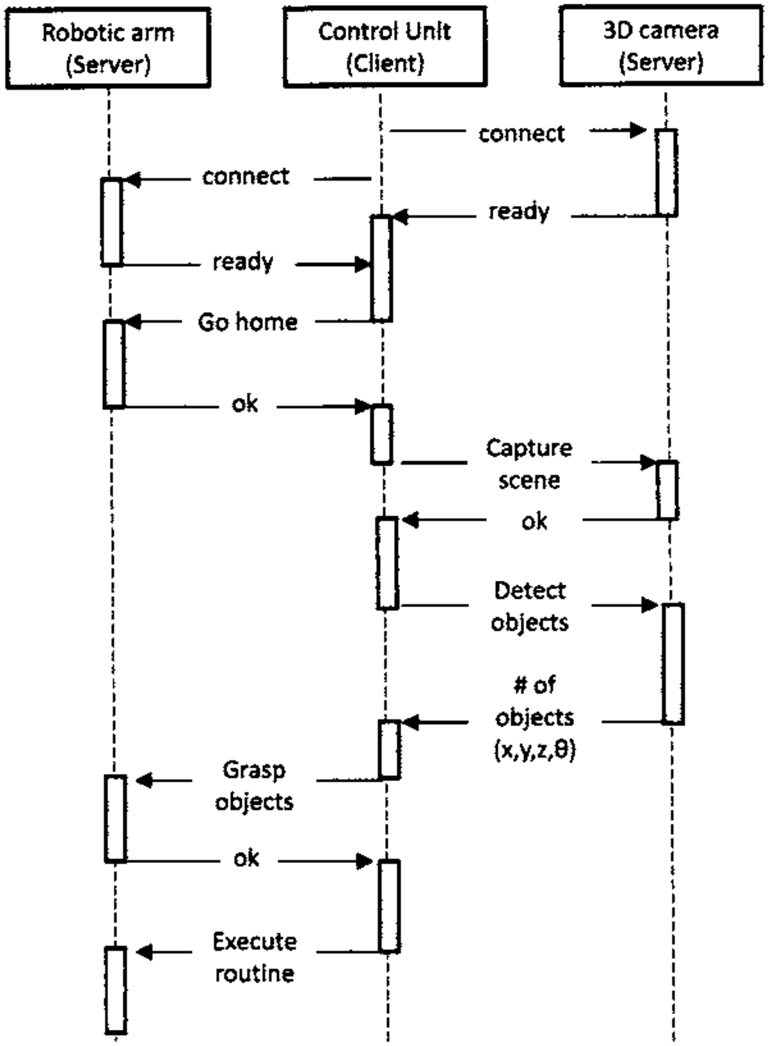


Figure 5.48: Ideal communication sequence in the VRT integration

As it has been mentioned the "Robot Logico" application was used in order to perform the integration, as this application does not support a function to indicate if the robot has reached the desired position, the communication sequence had to be changed, instead of sending a sign when the robot is in the desired position, the integration wait a certain number of seconds before a new motion instruction is sent, this change is shown in the next figure:

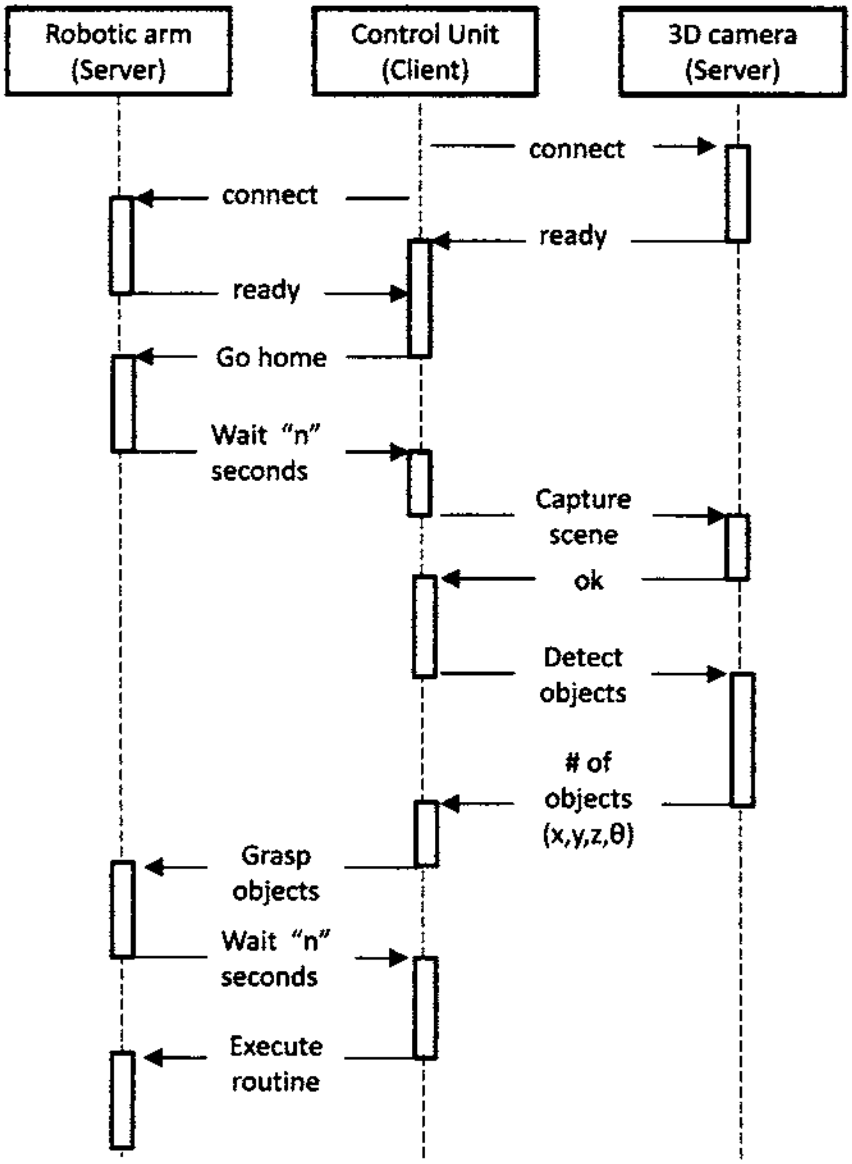
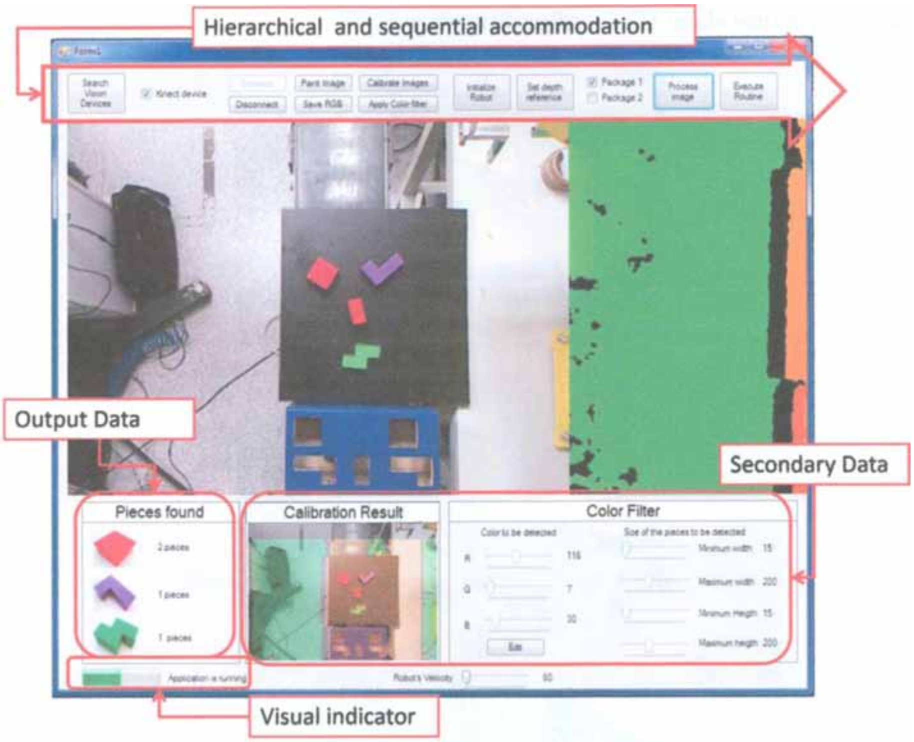


Figure 5.49: Communication sequence implemented in the VRT integration

Interface

The interface was designed in C# platform, this platform allow to create controls and configure their methods and functions, the next image shows the interface designed for this application:



2 pieces
1 pieces
1 pieces

Color to be detected
R 116
G 7
B 30

Size of the pieces to be detected
Minimum width 15
Maximum width 200
Minimum Heigth 15
Maximum heigth 200

Application is running
Robot's Velocity 50

Figure 5.50: Interface: Package Operation

As the figures 5.49 shows, the hints provided in the methodology were applied:

- The main commands to operate the interface are ordered in a sequential and hierarchical form, in resume the controls allow to: Connect to the kinect cameras, process the image, acquire information needed for the package execution, execute the operation in real time.
- The visual indicator that is enabled when the application is running.
- The secondary data is included and located away form the primary data.
- The output data is considered and shown in the interface as a preliminary result of the procedure, once that this output data (number and types of pieces detected) is obtained, the package operation can or can not be executed.

For more details regarding the interface design and performance refer to Appendix B.

5.3.5 Evaluation

Positioning error

The positioning error in this application refers to the calibration error which is present in the estimation of the Transformation matrix between the camera's coordinate system and the robot's coordinate system, this error is the same as the mean squared error obtained in the calibration process, the error estimated is:

Objects localization variation

Another consideration refers to the variation on the 3D information provided by the vision system, in order to evaluate it, the acquisition of the 3D location was performed executing the application developed 10 times and acquiring the 3D data 5 times in each execution, the 3D in each acquisition is the 3D position in reference with the camera of a certain object that was static through all the process. The following picture show this configuration:

Figure 5.51: Acquisition of 3D data of one object

The 3D data was stored and analyzed with MATLAB obtaining the following results:

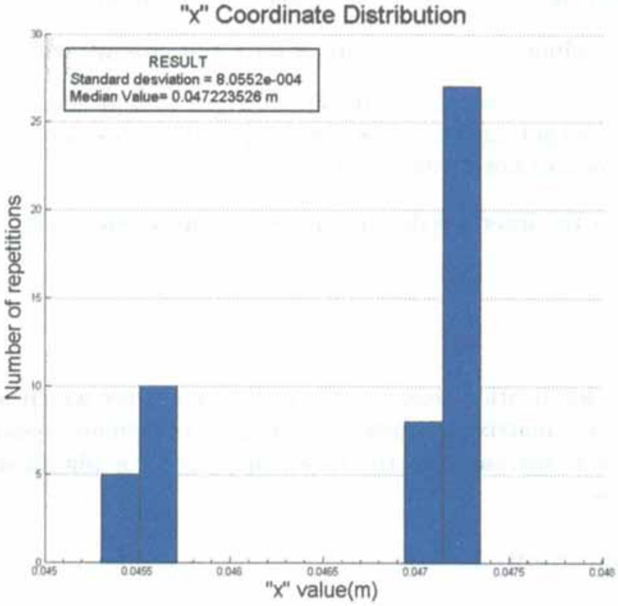


Figure 5.52: "x" coordinate distribution(50 acquisitions)

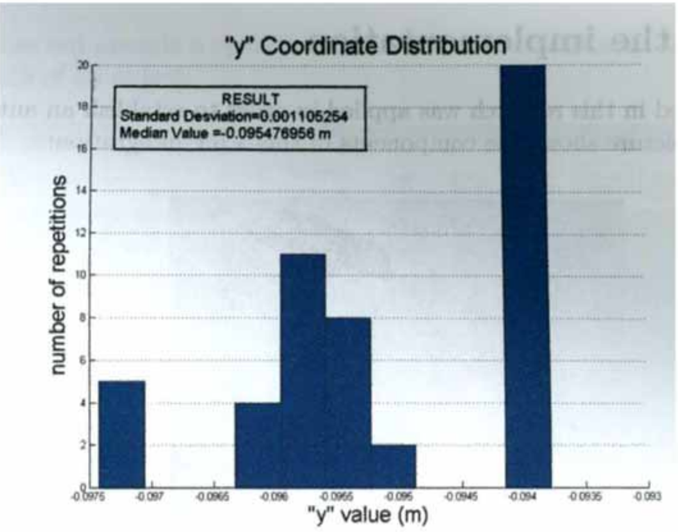


Figure 5.53: "y" coordinate distribution(50 acquisitions)

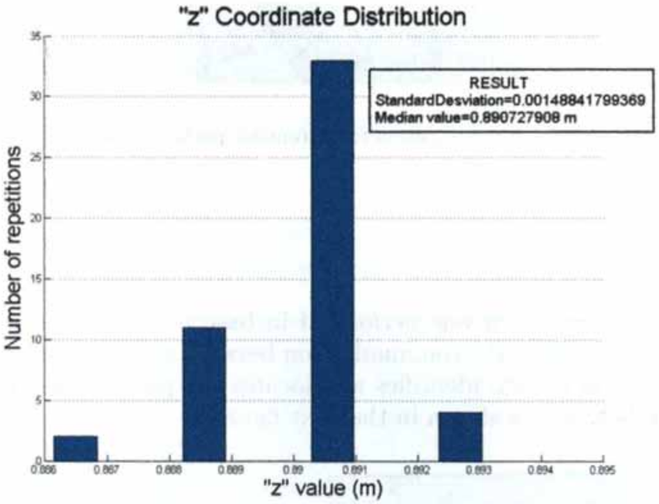


Figure 5.54: "z" coordinate distribution(50 acquisitions)

Efficiency

In order to evaluate the efficiency of the system a cycle is considered as the execution of the following tasks: Objects detection, objects grasping and objects approximation to the desired position, a successful cycle is considered when this three task are satisfactory executed.

$$efficiency\,(\%) = \left(\frac{\#\,of\ correct\ tasks}{\#\,of\ total\ tasks\ monitored}\right)(100) = \frac{9}{10}\,(100) = 90\% \qquad \text{(5.2}$$

There is another task that could be considered in this estimation and it is if the figure fits or not in its package spot, this task is excluded from the estimation due the fact that the position of the package pallet in reference with the base of the robot is not controllable in this research, if this position is fixed then this task can be included in the efficiency computation.

5.4 Results of the implementation

The methodology proposed in this research was applied in order to establish an automatic Package operation, the following picture shows the components of this VRT integration:

Figure 5.55: VRT integration for automatic package operation

The results obtained are:

- An automatic package operation was performed in base with the execution of the interface developed, this interface allows the communication between the three main components of this VRT system. The vision system identifies and locates the pieces of interest and provides its orientation to the robot, as it s shown in the next figure.

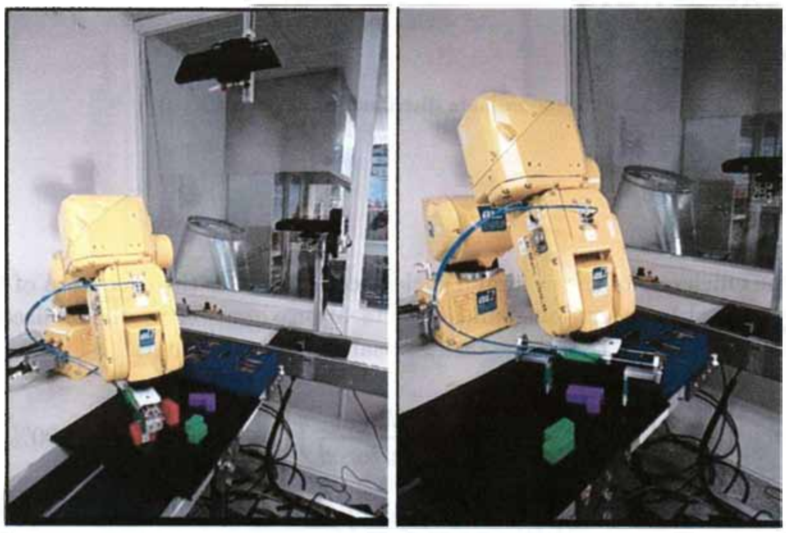


Figure 5.56: Robot grasping a piece in base with 3D coordinates provided by the kinect camera

- The system does not execute a operation when the secure statements are not satisfied, as the maximum depth of an object:

☑ Kinect device

☐ Package 1

☑ Package 2

The location provided by the camera is not reachable for the working zone

1 pieces

1 pieces

1 pieces

Color to be detected

R 116

G 7

B 30

Size of the pieces to be detected

Minimum width 15

Maximum width 200

Minimum Heigh 15

Maximum heigh 200

Application is running

Robot's Velocity 350

Figure 5.57: Error message: the location provided by the camera is not inside the working zone

- The system is able to decide if the execution of a package operation should be done or not in base with the type and number of pieces detected in the scene:

Figure 5.58: Interface behavior when the number or type of pieces are not correct for the package requested

- The VRT integration is able to perform a package operation even though the orientation and position of the objects to be packaged is unknown for the robot.

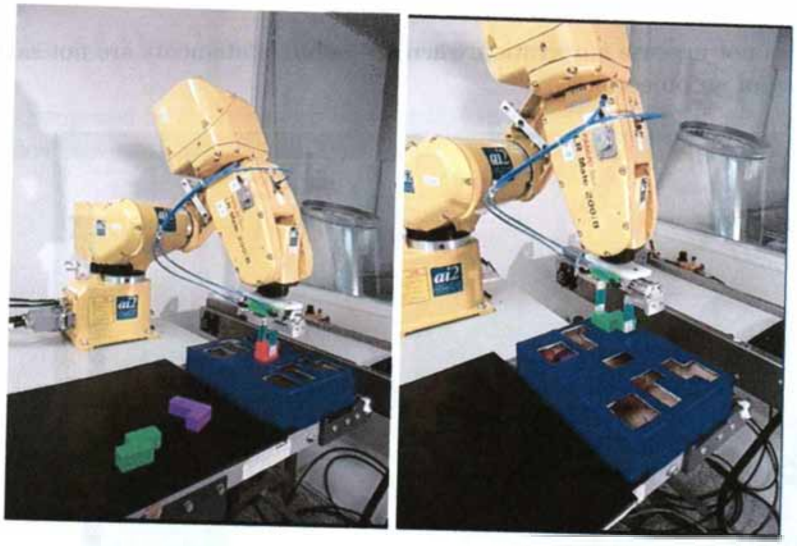


Figure 5.59: Successful package operation

Chapter 6

Conclusions and Future work

6.1 Conclusions

In this research a generic methodology to integrate Vision and Robot Technologies is presented, this methodology is intended to provide generic rules and steps that "must" be followed in order to establish a integration between these two technologies, The methodology established is compound by five principal steps which are divided in sub-steps according to the nature of the topic.

The methodology includes hints and recommendations for the most important manufacturing processes where this integration is applied, even though, this methodology is flexible to be adequate to another process, preserving the majority of its applicability, also the structure of the methodology allows to focus it into a certain process of interest, this focusing action will considerably change only one of the five stages proposed.

The methodology presented was validated through its application in a case of study, the methodology evaluation is considered as satisfactory because its application allowed the adequate and successful integration of a vision system and a robotic device which performed an automatic packaging process.

The application of this methodology can be considered as the first step in order to establish a standardization of this type of integration, where certain activities and parameters can be easily defined for the majority of the processes or operations that involve the vision robot integration.

The main limitation of the methodology presented is:

- Due the great diversity of manufacturing processes that include VRT the methodology part regarding to the application objective can not be applied in every kind of process, and including all the processes is not an option, so the three principal applications were considered in order to establish this part of the methodology.

6.2 Future Work

- One of the potential research lines is to implement all the steps proposed in the methodology in a knowledge-based software application in order to facilitate the VRT integration to the user.
- In "Integration Objective" stage more detailed information can be included in order to "ensure" that most of the VRT applications are included in the methodology.
- In "Illumination" and "Camera Mounting" steps, information of commercial devices can be added, including suppliers information and specifications. With this not only a illumination

technique, light type or a mounting configuration will be suggested, also a set of commercial devices that fulfill the requirements could be available.

Bibliography

[Arai et al., 2000] Arai, T., Aiyama, Y., Maeda, Y., Sugi, M., and Ota, J. (2000). Agile assemly system by plug and produce. *Annals of the CIRP*, 49:1–4.

[Balakrishnan et al., 2000] Balakrishnan, S., Popplewell, N., and M., T. (2000). Intelligent robotic assembly. *Computers & Industrial Engineering*, 38:467–478.

[Benhabib, 2003] Benhabib, B. (2003). *Manufacturing: Design, Production, Automation and Integration.* Marcel Dekker.

[Bone and Capson, 2003] Bone, G. and Capson, D. (2003). Vision-guided fixtureless assembly of automotive components. *Robotics and Computer Integrated Manufacturing*, 19:79–87.

[Boothroyd, 2005] Boothroyd, G. (2005). *Assembly Automation and Product Design.* CRC Press.

[Brosnan and Sun, 2003] Brosnan, T. and Sun, D. (2003). Improving quality inspection of food products by computer vision-a review. *Journal of Food Engineering.*

[Bruccoleri et al., 2006] Bruccoleri, M., Pasek, Z. J., and Koren, Y. (2006). Operation management in reconfigurable manufacturing systems: Reconfiguration for error handling. *Production economics*, 100:87–100.

[Burrus, 2010] Burrus, N. (2010). Kinect calibration, obtained from url: http://nicolas.burrus.name/index.php/research/kinectcalibration.

[Conolly, 2008] Conolly, C. (2008). Artificial intelligence and robotic hand-eye coordiantion. *Industrial robot*, 35(6):496–503.

[Conolly, 2009] Conolly, C. (2009). Machine vision advances and applications. *Assembly Automation*, 29(2):106–111.

[Crawford, 2010] Crawford, S. (2010). How microsoft kinect works, obtained from url: http://electronics.howstuffworks.com/microsoft-kinect.htm.

[Crock, 2011] Crock, N. (2011). Kinect depth vs. actual distance, obtained from url: http://mathnathan.com/2011/02/03/depthvsdistance.

[Cuevas, 2005] Cuevas, R. (2005). Frovis: A fexible robot vision system. Master's thesis, Instituto Tecnologico y de Estudios Superiores de Monterrey.

[Espinosa Perez, 2006] Espinosa Perez, P. (2006). Desarrollo de una metodologia para la integracion de sistemas robot-vision. Master's thesis, Instituto Tecnologico y de Estudios Superiores de Monterrey.

[Fabel, 1997] Fabel, G. (1997). Machine vision systems looking better all the time.

[Fisher, 2010] Fisher, M. (2010). Kinect, obtained from url: http://graphics.stanford.edu/ mdfisher/kinect.html.

[Flores, 2005] Flores, B. (2005). Model to integrate mobility technologies for material-handling automation in manufacturing systems. Master's thesis, Instituto Tecnologico y de Estudios Superiores de Monterrey.

[Fu et al., 2010] Fu, K., Gonzalez, R., and Lee, C. (2010). *Robotics: control,sensing,vision and Intelligence.* McGraw-Hill.

[Gonzalez et al., 2003] Gonzalez, E., Cruz, S., Seelinger, M., and Cervantes, J. (2003). An efficient multi-camera, multi-target scheme for the three-dimensional control of robots using uncalibrated vision. *Robotics and Computer Integrated Manufacturing*, 19:387–400.

[Gonzalez, 2007] Gonzalez, M. (2007). Desarrollo de una herramienta computacional para la implementacin de estrategias de control. Master's thesis, Instituto Tecnologico y de Estudios Superiores de Monterrey.

[Gottfried, 2008] Gottfried, B. (2008). Qualitative similarity measuresthe case of two-dimensional outlines. *Computer Vision and Image Understanding*, 110:117–133.

[Groover, 2008] Groover, M. (2008). *Automation, production systems, and computer-integrated manufacturing.* Prentice Hall.

[Haley and Kuehl, 2005] Haley, R. and Kuehl, S. (2005). Four steps for hmi design. *Control Engineering*, 52:110.

[Hardin, 2009] Hardin, W. (2009). Future manufacturing trends increase the demand for machine vision. *Machine vision Online.*

[Hernandez Campos, 2010] Hernandez Campos, D. (2010). Vision robot assistance in manufacturing cell.

[Hunsicker et al., 1994] Hunsicker, R., Patten, J., Ledford, A., Ferman, C., and Allen, M. (1994). Automatic vision inspection and measurement system for external screw threads. *Journal of Manufacturing Systems*, 13:370–383.

[Hunt, 2002] Hunt, M. (2002). *Machine Vision Applications in Industrial Inspection X.* Proceedings of SPIE.

[Janannathan, 1997] Janannathan, S. (1997). Automatic inspection of wave soldered joints using neural networks. *Journal of Manufacturing Systems*, 16:389–398.

[Jiang and Chen, 2001] Jiang, B. and Chen, T. C. (2001). Machine vision inspection for the protrusion rate of a diamond tool. *Journal of Manufacturing Systems*, 20:357–362.

[Jóźviak et al., 2010] Jóźviak, L., Nedjah, N., and Figueroa, M. (2010). Modern development methods and tools for embedded reconfigurable systems: A survey. *INTEGRATION,the VLSI journal*, 43:1–33.

[Kagami, 2010] Kagami, S. (2010). High-speed vision systems and projectors for real-time perception of the world. *IEEE.*

[Kinect, 2011] Kinect, O. (2011). Main page.

[Kruger et al., 2009] Kruger, J., LIEN, T., and Verl, A. (2009). Cooperation of human and machines in assembly lines. *CIRP Annals - Manufacturing Technology*, 58:628–646.

[Langmoen, 1983] Langmoen, T. (1983). Analysing products with respect to flexible assemblyautomation. *Proceedings of the 15th CIRP International Seminar on Manufacturing Systems*, pages 277–288.

[Lanzetta et al., 1999] Lanzetta, M., Santochi, M., and G., T. (1999). Computer-aided visual inspection in assembly. *Annals of the CIRP*, 48.

[López Peláez and Kyriakou, 2008] López Peláez, A. and Kyriakou, D. (2008). Robots, genes and bytes: technology development and social changes towards the year 2020. *Technological Forecasting and Social Change*, 75:1176–1201.

[Magnenat, 2011] Magnenat, S. (2011). Imaging information, obtained from url: http://openkinect.org/wiki/imaginginformation.

[Malamas et al., 2003] Malamas, E. N., Euripides, P., Zervakis, M., Petit, L., and Legat, J. D. (2003). A survey on industrial vision systems, applications and tools. *Image and Vision Computing*, 21:171–188.

[McKerrow, 1991] McKerrow, P. J. (1991). *Introduction to Robotics.* ADDISON WESLEY.

[Miller et al., 1996] Miller, W., Jaskot, J., McCoy, B., and Schiller, E. (1996). A distributed system for 100 _ inspection of aluminum sheet products. In *International Conference on Signal Processing Applications and Technology.*

[Olmos, 2009] Olmos, E., B. M. S. A. (2009). Programacion de robots a traves de xml. *XXX Jornadas de Automatica.*

[Pauli et al., 2001] Pauli, J., Schmidt, A., and Sommer, G. (2001). Vision-based integrated system for object inspection and handling. *Robotics and Autonomous Systems*, 37:297–309.

[Pena et al., 2005] Pena, M., Lopez, I., Rios, R., and Corona, J. (2005). Machine vision approach for robotic assembly. *Assembly Automation*, 25:204–216.

[Perez, 2010] Perez, I. (2010). Curso basico insight.

[Polzer and Stauffer, 2006] Polzer, K. and Stauffer, T. (2006). Ease of use in engineering. *Control Engineering*, 53.

[Rampersad, 1994] Rampersad, H. (1994). *Integrated and simultaneous design for robotic assembly.* Wiley.

[Sang et al., 2009] Sang, I., Sang, N., Rodney, R., and Moon, S. (2009). A minimum-time algorithm for intercepting and object on a conveyor belt. *Industrial Robot: An International Journal*, 36:127–137.

[Scholz-Reiter and Freitag, 2007] Scholz-Reiter, B. and Freitag, M. (2007). Autonomous processes in assembly systems. *Annals of the CIRP*, 56:712–729.

[Shin et al., 2002] Shin, D., Han, C., and Shik Moon, Y. (2002). Ball stud inspection system using machine vision. *ISA transactions*, 41:3–11.

[Sitte and Winzer, 2007] Sitte, J. and Winzer, P. (2007). Methodic design of robot vision systems. *International Conference on Mechatronics and Automation.*

[Steger et al., 2008] Steger, C., Ulrich, M., and Wiedemann, C. (2008). *Machine Vision Algorithms and Applications.* WILEY-VCH.

[Suarez Corona, 2009] Suarez Corona, V. (2009). Imvr: Robot guiado por vision en tiempo real en celda flexible de manufactura. Master's thesis, Instituto Tecnologico y de Estudios Superiores de Monterrey.

[Tsai et al., 1998] Tsai, D., Chen, J., and J.F., C. (1998). A vision system for surface roughness assessment using neural networks. *International Journal of Advanced Manufacturing Technology*, 14:412–422.

[Tsai and Hsieh, 1999] Tsai, D. and Hsieh, C. (1999). Automated surface inspection for directional textures. *Image and Vision Computing*, 18:49–62.

[Valiente et al.,] Valiente, J., Andreu, G., and Rodas, A. Iluminacion.

[Watanabe et al., 2005] Watanabe, A., Sakakibara, S., Ban, K., Yamada, M., Shen, G., and Arai, T. (2005). Autonomous visual measurement for accurate setting of workpieces in robotic cells. *Annals of the CIRP*, 54:13–16.

Appendix A

Illumination Type and Technique Selection Software (ITTSS)

The next figure shows the principal window of ITTSS:

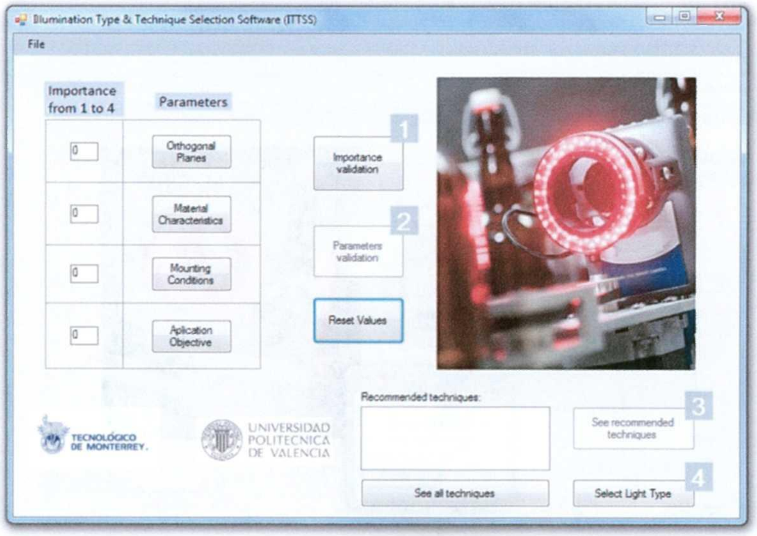
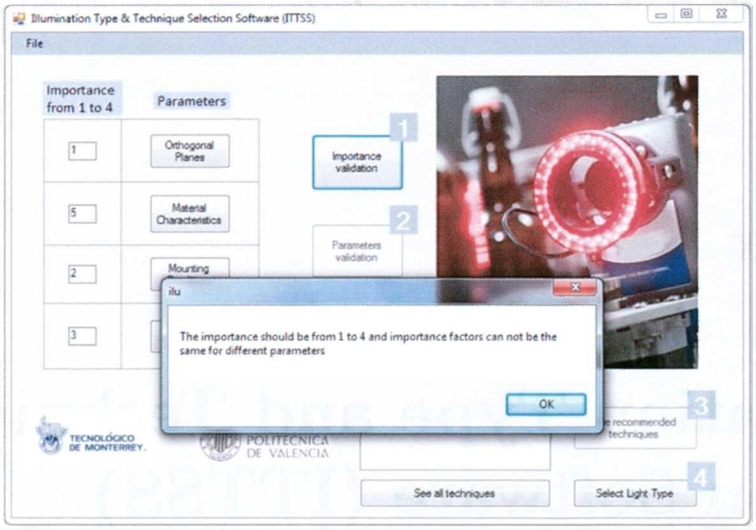


Figure A.1: Illumination Type and Technique Selection Software

The following images show the performance of the software, the first step is to assign importance factor to each one of the four parameters available. If the user assigns an incorrect importance factor (from 1 to 4), the software will pop an information window:



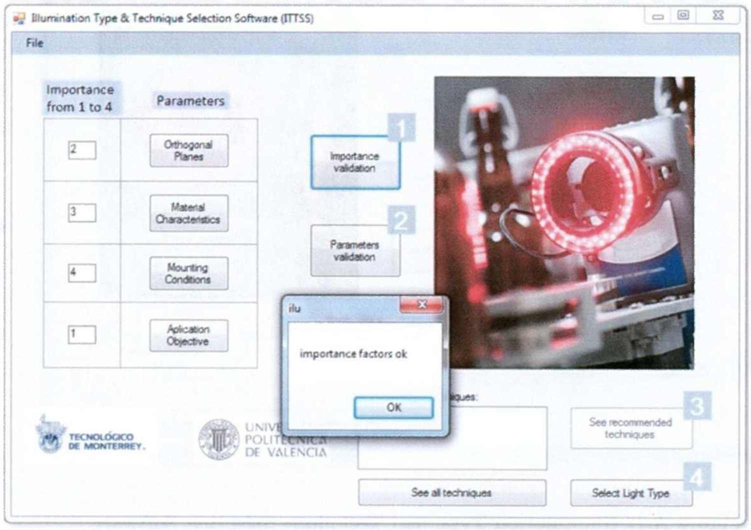
Importance from 1 to 4	Parameters
1	Orthogonal Planes
5	Material Characteristics
2	Mounting
3	

ilu

The importance should be from 1 to 4 and importance factors can not be the same for different parameters

Figure A.2: Importance assignment error

When the importance factors are ok, the system will let it know, and the "Parameters validation" button will be enabled:



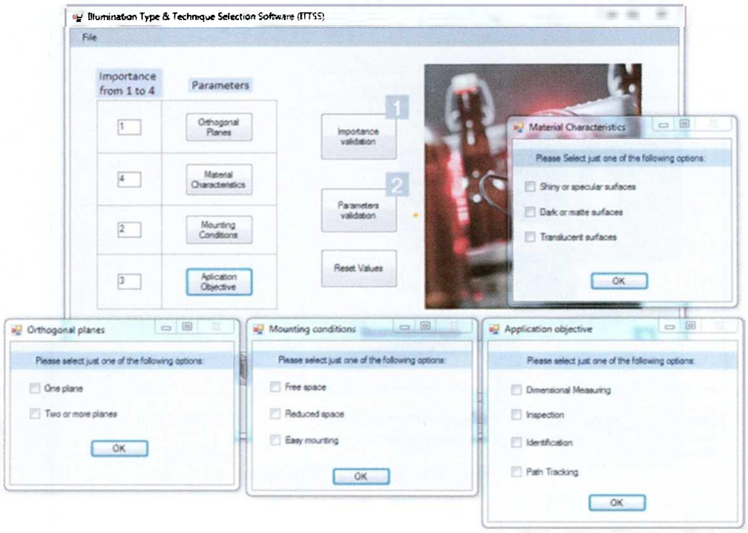
Importance from 1 to 4	Parameters
2	Orthogonal Planes
3	Material Characteristics
4	Mounting Conditions
1	Aplication Objective

ilu

importance factors ok

Figure A.3: Importance assignment ok

The following step is to select one of the available option for each parameter, the next figure shows this situation:



Importance from 1 to 4	Parameters
1	Orthogonal Planes
4	Material Characteristics
2	Mounting Conditions
3	Aplication Objective

Material Characteristics

Please Select just one of the following options:

☐ Shiny or specular surfaces
☐ Dark or matte surfaces
☐ Translucent surfaces

Orthogonal planes

Please select just one of the following options:

☐ One plane
☐ Two or more planes

Mounting conditions

Please select just one of the following options:

☐ Free space
☐ Reduced space
☐ Easy mounting

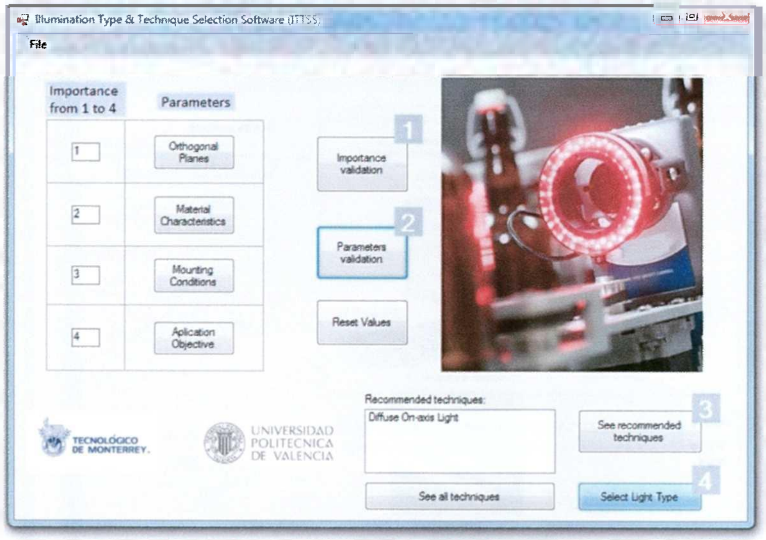
Application objective

Please select just one of the following options:

☐ Dimensional Measuring
☐ Inspection
☐ Identification
☐ Path Tracking

Figure A.4: Selection windows

Once that the options have been selected, and by clicking on the "Parameters Validation" button, a listbox containing the recommended techniques will be filled, also the "See recommended techniques" button will be enabled.



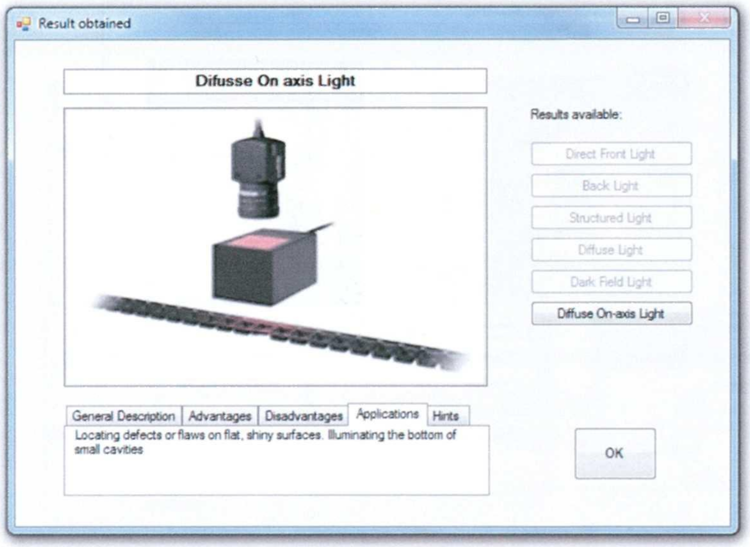
Importance from 1 to 4	Parameters
1	Orthogonal Planes
2	Material Characteristics
3	Mounting Conditions
4	Aplication Objective

Recommended techniques:

Diffuse On-axis Light

Figure A.5: Result

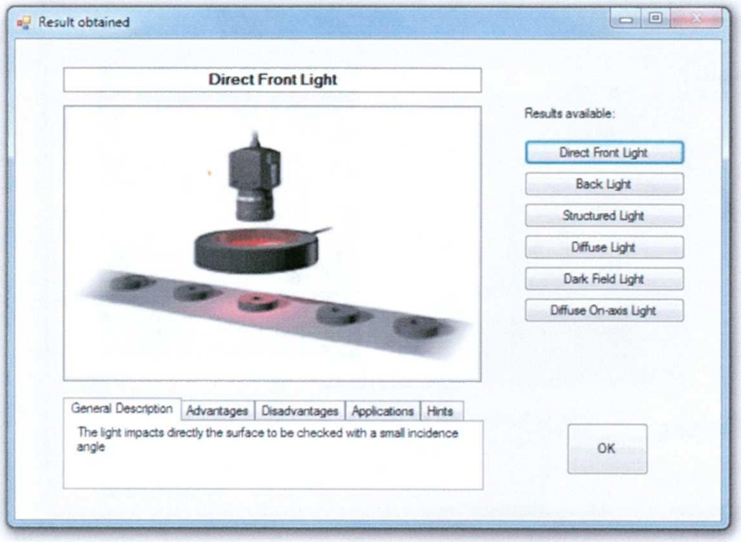
If the user clicks on the "See recommended techniques" button, a new window will pop up, in this instance the user will be able to select only the techniques recommended by ITTSS, in the left side of the window is a picture showing the configuration of the technique and a text box containing relevant information about the technique.



Locating defects or flaws on flat, shiny surfaces. Illuminating the bottom of small cavities

Figure A.6: Detailed information about the technique

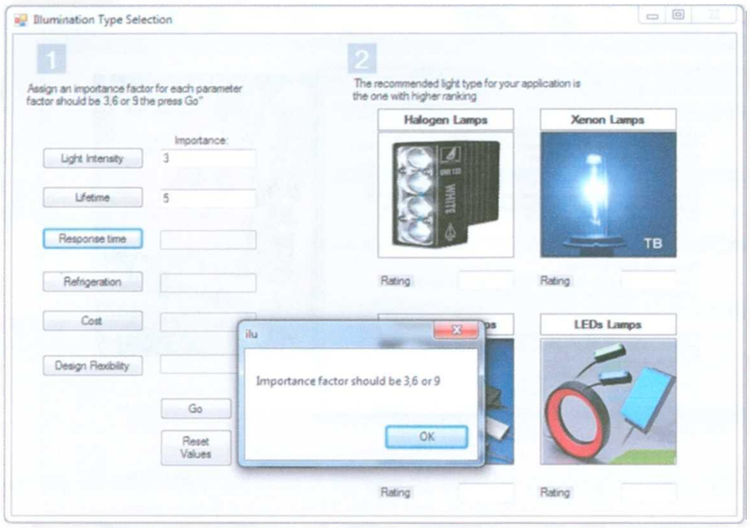
If the user wants to check all the techniques out, there is a "see all techniques" button, which will show a window where all techniques and their information will be available.



The light impacts directly the surface to be checked with a small incidence angle

Figure A.7: All techniques information window

In order to select the light type, the user should click on "Select Light type" button, and a new window will appear, in this window a importance factor should be set for each one of the 6 parameters shown, (if the user assigns a different factor the system could not continue)



Illumination Type Selection

1

Assign an importance factor for each parameter factor should be 3,6 or 9 the press Go"

Importance:

Light Intensity: 3

Lifetime: 5

Response time:

Refrigeration:

Cost:

Design Flexibility:

2

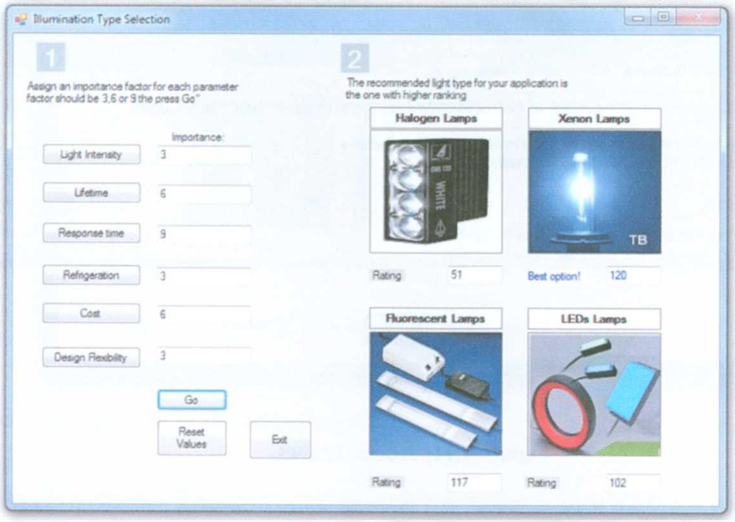
The recommended light type for your application is the one with higher ranking

ilu

Importance factor should be 3,6 or 9

Figure A.8: Importance assignment error

Once that the importance factors have been assigned and by clicking the "Go" button, a score for each light type will be shown under every Picture, the best score will be in blue color:



Illumination Type Selection

1

Assign an importance factor for each parameter factor should be 3,6 or 9 the press Go"

Importance:

Light Intensity: 3

Lifetime: 6

Response time: 9

Refrigeration: 3

Cost: 6

Design Flexibility: 3

2

The recommended light type for your application is the one with higher ranking

Halogen Lamps — Rating: 51

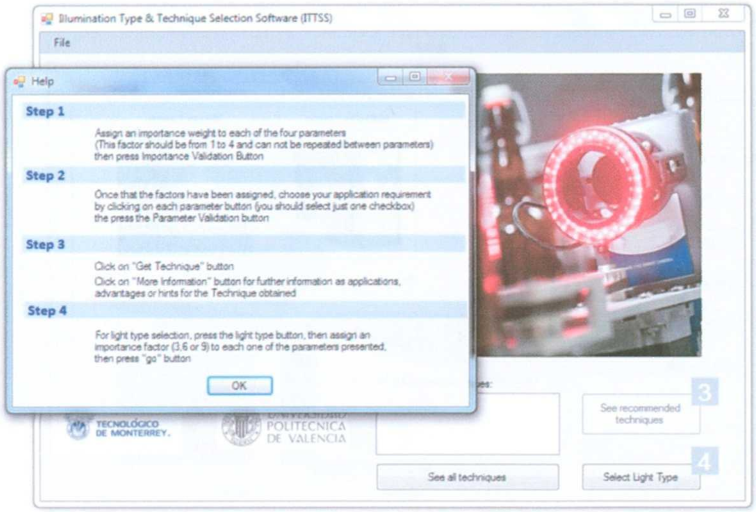
Xenon Lamps — Best option!: 120

Fluorescent Lamps — Rating: 117

LEDs Lamps — Rating: 102

Figure A.9: Result for Light Type

The next figures show additional information of the ITTSS:



Step 1

Assign an importance weight to each of the four parameters
(This factor should be from 1 to 4 and can not be repeated between parameters)
then press Importance Validation Button

Step 2

Once that the factors have been assigned, choose your application requirement
by clicking on each parameter button (you should select just one checkbox)
the press the Parameter Validation button

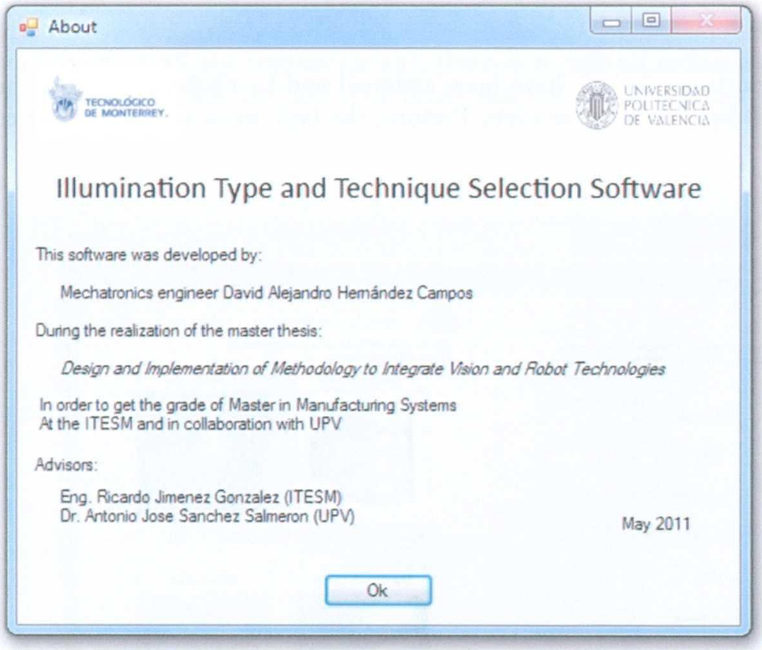
Step 3

Click on "Get Technique" button
Click on "More Information" button for further information as applications,
advantages or hints for the Technique obtained

Step 4

For light type selection, press the light type button, then assign an
importance factor (3,6 or 9) to each one of the parameters presented,
then press "go" button

Figure A.10: Help window



Illumination Type and Technique Selection Software

This software was developed by:

Mechatronics engineer David Alejandro Hernández Campos

During the realization of the master thesis:

Design and Implementation of Methodology to Integrate Vision and Robot Technologies

In order to get the grade of Master in Manufacturing Systems
At the ITESM and in collaboration with UPV

Advisors:

Eng. Ricardo Jimenez Gonzalez (ITESM)
Dr. Antonio Jose Sanchez Salmeron (UPV)

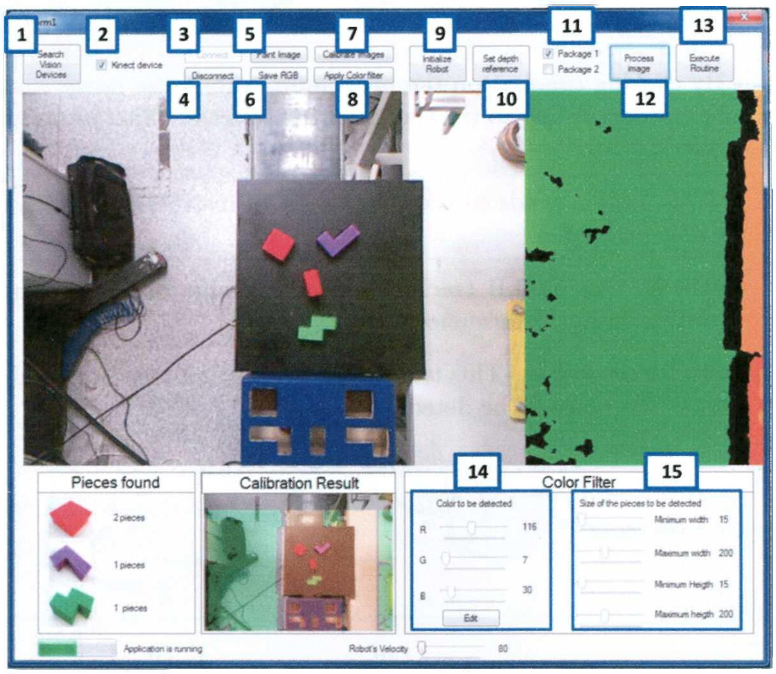
May 2011

Figure A.11: About ITTSS

Appendix B

Application User Interface

The following image shows the user interface designed for the packaging operation.



2 pieces
1 pieces
1 pieces

R 116
G 7
B 30

Minimum width 15
Maximum width 200
Minimum Heigth 15
Maximum heigth 200

Robot's Velocity 80

Figure B.1: About ITTSS (cont.)

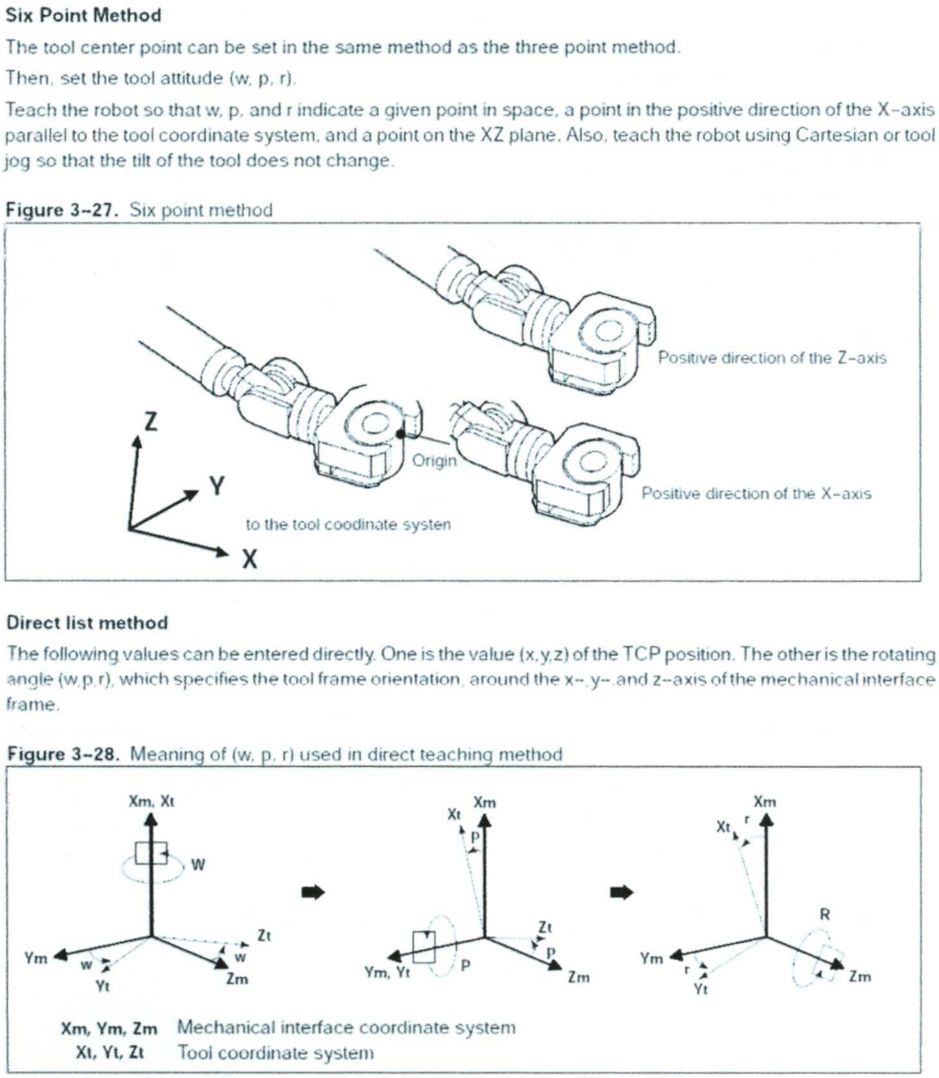
The functions of the controls is described next:

1. **Search Vision Devices.-** This button executes a routine where through a DLL all the devices connected to the USB ports are identified, and indicates if there is any kinect device connected to the PC.
2. **Kinect Device.-** Through this checkbox the kinect device connected is selected as the vision system in the application.
3. **Connect.-**Clicking this button creates a new instance of the kinect class, starts and configure the video and depth cameras and creates an updating thread.
4. **Disconnect**This button closes the connection previously opened and abort the thread created in the "connect" button

5. **Paint Image.-**Creates an image corresponding to the kinect data, and displays it in the GL control added in the interface.
6. **Save RGB.-**Saves in a predetermined directory a jpg image corresponding to the data captured by the RGB camera.
7. **Calibrate images.-** Perform a calibration where the process described in section 44 is executed and displays in the "Calibration Result box" the overlap between RGB and calibrated depth images.
8. **Apply color filter.-** Applies an Euclidean color filter (see section 44), using the values of the R,G,B trackbars and store the image resulting in a determined path.
9. **Initialize robot.-** This button send a set of instructions to the robot in order to configure its velocity, the Tool Center Point and the Home position.
10. **Set depth reference.-**This button executes an estimation of the surface where de objects lay in order to store it as reference for further calculations.
11. **Package 1 and 2.-**This checkboxes select which of the two possible packages should be executed.
12. **Process Image.-** By clicking this button the programm process the image and looks for the type and number of pieces according to the checkbox(12) selected. This procedure displays its result in the "Pieces found box".
13. **Execute routine.-**This button sends to the robot a set of instructions which correspond to the objects grasping and packaging.
14. **Color to be detected.-** The R,G,B trackbars sets the value of RGB channels that the Euclidean filter will utilize in its performance.
15. **Size of the pieces to be detected.-**This trackbars set the minimum and maximum of width and height (in pixels) of the blobs to be detected.

Appendix C

Tool Center Point Configuration



Six Point Method

The tool center point can be set in the same method as the three point method.

Then, set the tool attitude (w, p, r).

Teach the robot so that w, p, and r indicate a given point in space, a point in the positive direction of the X-axis parallel to the tool coordinate system, and a point on the XZ plane. Also, teach the robot using Cartesian or tool jog so that the tilt of the tool does not change.

Figure 3-27. Six point method

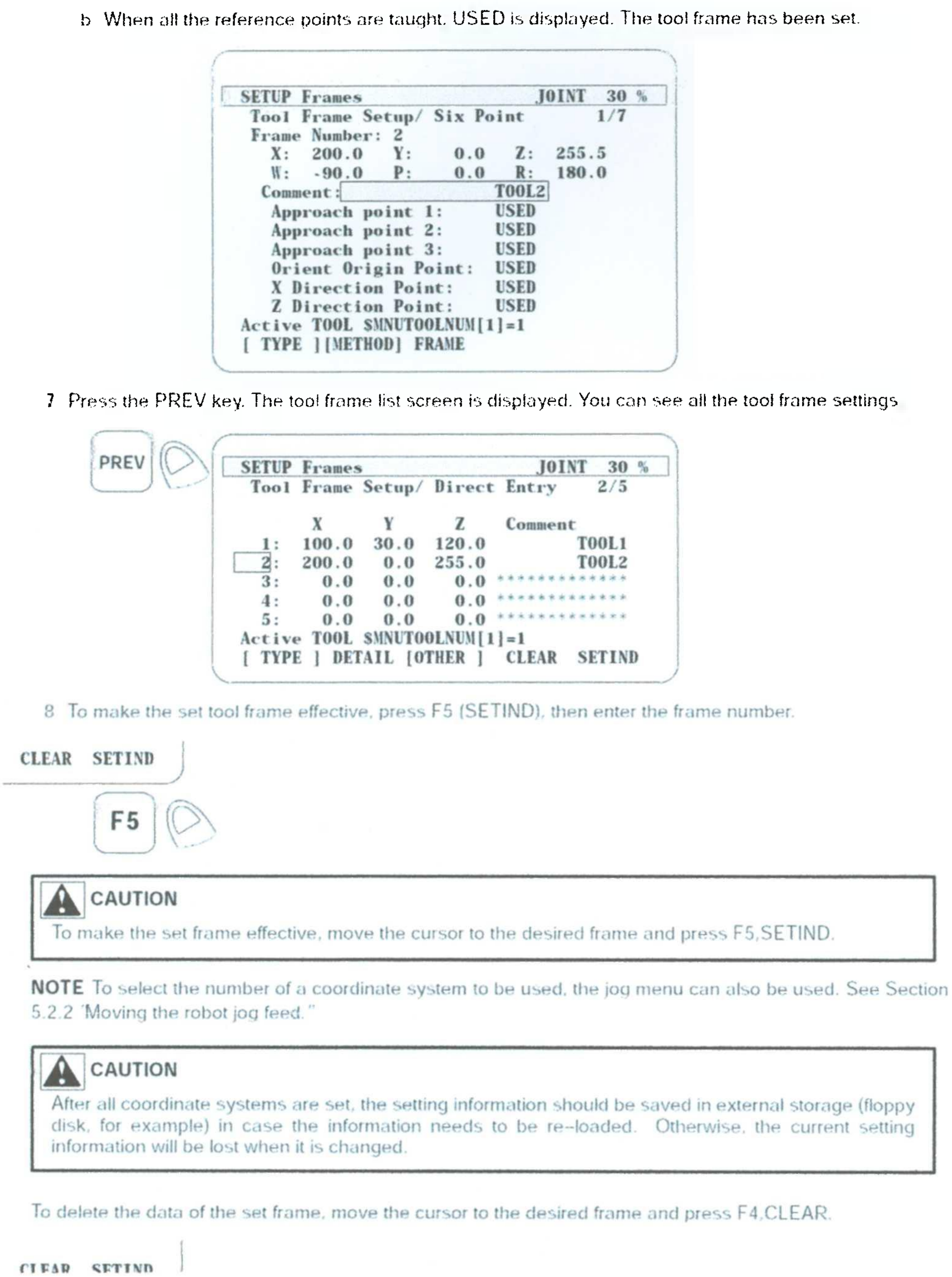
Direct list method

The following values can be entered directly. One is the value (x,y,z) of the TCP position. The other is the rotating angle (w,p,r), which specifies the tool frame orientation, around the x-, y-, and z-axis of the mechanical interface frame.

Figure 3-28. Meaning of (w, p, r) used in direct teaching method

Xm, Ym, Zm Mechanical interface coordinate system
Xt, Yt, Zt Tool coordinate system

Figure C.1: TCP Configuration



b When all the reference points are taught, USED is displayed. The tool frame has been set.

```
SETUP Frames                          JOINT  30 %
  Tool Frame Setup/ Six Point          1/7
  Frame Number: 2
    X:  200.0   Y:     0.0   Z:  255.5
    W:  -90.0   P:     0.0   R:  180.0
   Comment:                TOOL2
    Approach point 1:      USED
    Approach point 2:      USED
    Approach point 3:      USED
    Orient Origin Point:   USED
    X Direction Point:     USED
    Z Direction Point:     USED
Active TOOL $MNUTOOLNUM[1]=1
[ TYPE ][METHOD] FRAME
```

7 Press the PREV key. The tool frame list screen is displayed. You can see all the tool frame settings

PREV

```
SETUP Frames                          JOINT  30 %
  Tool Frame Setup/ Direct Entry      2/5

         X       Y       Z      Comment
   1:  100.0   30.0   120.0          TOOL1
   2:  200.0    0.0   255.0          TOOL2
   3:    0.0    0.0     0.0 *************
   4:    0.0    0.0     0.0 *************
   5:    0.0    0.0     0.0 *************
Active TOOL $MNUTOOLNUM[1]=1
[ TYPE ] DETAIL [OTHER ]  CLEAR  SETIND
```

8 To make the set tool frame effective, press F5 (SETIND), then enter the frame number.

CLEAR SETIND

F5

CAUTION

To make the set frame effective, move the cursor to the desired frame and press F5,SETIND.

NOTE To select the number of a coordinate system to be used, the jog menu can also be used. See Section 5.2.2 'Moving the robot jog feed.''

CAUTION

After all coordinate systems are set, the setting information should be saved in external storage (floppy disk, for example) in case the information needs to be re-loaded. Otherwise, the current setting information will be lost when it is changed.

To delete the data of the set frame, move the cursor to the desired frame and press F4,CLEAR.

CLEAR SETIND

Figure C.3: TCP Configuration (cont.)