

APLICACIÓN DEL RECONOCIMIENTO DE IMÁGENES PARA LA
MANIPULACIÓN ROBÓTICA DE OBJETOS EN MOVIMIENTO

ANDREA BUILES GÓMEZ
ESTEBAN PALACIO VELÁSQUEZ

UNIVERSIDAD PONTIFICIA BOLIVARIANA
ESCUELA DE INGENIERÍAS
FACULTAD DE INGENIERÍA MECÁNICA
MEDELLÍN

2015

APLICACIÓN DEL RECONOCIMIENTO DE IMÁGENES PARA LA
MANIPULACIÓN ROBÓTICA DE OBJETOS EN MOVIMIENTO

ANDREA BUILES GÓMEZ
ESTEBAN PALACIO VELÁSQUEZ

Trabajo de grado para optar al título de Ingeniero Mecánico

Director

Julio César Correa Rodríguez

Doctor of Philosophy in Mechanical Engineering

UNIVERSIDAD PONTIFICIA BOLIVARIANA

ESCUELA DE INGENIERÍAS

FACULTAD DE INGENIERÍA MECÁNICA

MEDELLÍN
2015

Nota de aceptación

Firma

Nombre:

Presidente del jurado

Firma

Nombre:

Jurado

Firma

Nombre:

Jurado

Medellín, Abril 17 de 2015

DEDICATORIA

En mayor medida a mis padres y mi hermano, por entenderme y apoyarme siempre, por estar disponibles siempre para mí y por esforzarse por verme crecer, a mi gran amiga y compañera Andrea que cada día me motivó a mejorar un poco más no solo como estudiante sino como persona, a mi linda Mayi que me alegró desde ese primer bonito día y por último a Dios por darme la gran oportunidad de estar aquí.

Esteban Palacio Velásquez

A mi madre, por enseñarme el valor de soñar; a mi padre, por enseñarme el valor de materializar mis sueños; y al coautor, por brindarme un gran trabajo en equipo, una excelente amistad y un increíble crecimiento personal.

Andrea Builes Gómez

AGRADECIMIENTOS

Queremos darle las gracias a todos y cada uno de los profesores que nos dieron lo mejor de sí, por sus grandes enseñanzas de vida, por su dedicación a la docencia y por su buena energía. Le queremos dar infinitas gracias a los doctores Julio César Correa y Jorge Mario Londoño por darnos esta gran oportunidad, por ser personas tan comprometidas, excelentes guías y por poner tanto conocimiento en nuestras manos, de verdad muchas gracias.

CONTENIDO

1.	INTRODUCCIÓN A LOS SISTEMAS DE VISIÓN ARTIFICIAL	16
1.1	Conceptos básicos	16
1.2	Posicionamiento de la cámara	20
1.3	<i>Toolbox</i> de <i>Matlab</i> para procesamiento de imágenes	20
2.	INTRODUCCIÓN A LA TEORÍA DE MANIPULADORES ROBÓTICOS	22
2.1	Robots industriales	23
2.1.1	Conceptos básicos de mecanismos y manipuladores robóticos.	23
2.2	Clasificación de los manipuladores	24
2.3	Análisis de posición directo e inverso	26
2.4	Lenguaje de programación	26
2.5	Sensores adicionales usados en robótica	27
3.	CALIBRACIÓN	29
3.1	¿Qué es calibración de cámara y por qué se debe realizar?	30
3.2	Parámetros intrínsecos y extrínsecos	30
3.3	Tipos de calibración	31
3.3.1	Calibración basada en un objeto de referencia 3D	31
3.3.2	Calibración basada en un plano 2D	32
3.3.3	Calibración basada en una línea 1D	33

3.3.4	Auto-calibración.....	33
3.4	Proceso común de calibración basada en un plano 2d	34
3.5	<i>Toolbox</i> de calbración de cámara de <i>Matlab</i> ®.	35
4.	SIMULACIÓN.....	36
4.1	Calibración de cámara.....	37
4.2	Obtención de centroides por medio del procesamiento de imágenes	38
4.2.1	Conversión de pixeles a unidades de longitud real.	42
4.3	Análisis cinemático de posición inverso para manipulador robótico serial.....	43
4.4	Simulación en <i>Solid Edge</i> ®.	45
4.4.1	Resultado de la simulación.....	49
5.	TRABAJO FUTURO.....	52
5.1	Estudio del lenguaje de programación del robot.....	53
5.2	Determinación de las condiciones óptimas de interconexión; Error! Marcador no definido.	
5.3	Creación del algoritmo de integración	54
5.4	Ejecución y análisis de resultados.....	55

LISTA DE FIGURAS

Figura 1: Etapas de un sistema de visión artificial. Realización de los autores.....	16
Figura 2: Binarización de una imagen. (a) Imagen a color. (b) Imagen binaria.	17
Figura 3: Ejemplo de segmentación. (a) Imagen original. (b) Imagen segmentada por regiones.	18
Figura 4: orden común de ejecución de las tareas de un sistema de visión artificial.....	20
Figura 5: Resumen capítulo 2.	22
Figura 6: Ejemplos de efectores finales. (a) Pinza de vacío. (b) Pinza neumática. (c) Pinza hidráulica. (d) Pinza servo-eléctrica.....	24
Figura 7: Semejanza brazo robótico serial con brazo humano.	25
Figura 8: Ejemplo de robot paralelo.....	26
Figura 9: Ejemplo robot híbrido.....	26
Figura 10: resumen capítulo 3.....	29
Figura 11: Objeto usado para la calibración 3D.....	32
Figura 12: Patrón usado para la calibración 2D.....	32
Figura 13: Objeto usado para la calibración 1D.....	33
Figura 14: Puntos característicos detectados en la imagen. Elaboración de los autores con ayuda de <i>Matlab</i> ®.....	34
Figura 15: Menú principal del <i>Camera Calibrator App</i> de <i>Matlab</i> ®.....	35
Figura 16: resumen capítulo 5.....	36

Figura 17: cámara <i>web</i> Logitech HD C615.	37
Figura 18: Hoja patrón recomendada por <i>Matlab</i> ® para la calibración de cámara.	38
Figura 19: Objetos creados para procesamiento de video.....	39
Figura 20: Pre-procesamiento de las imágenes tomadas. (a) Imagen original, (b) imagen en escala de grises, (c) imagen binarizada.	40
Figura 21: Extracción de bordes de regiones continuas.....	40
Figura 22: Recorte de la imagen alrededor de su <i>bounding box</i>	41
Figura 23: Identificación de centroides para diferentes geometrías. (a) Triángulo, (b) rectángulo 1, (c) rectángulo 2, (d) círculo, (e) círculo de la figura (d) a escala, (f) vista frontal de la figura (e).	41
Figura 24: Imagen patrón. (a) Imagen original, (b) <i>zoom</i> del origen.....	43
Figura 25: Origen de la banda transportador y el manipulador robótica. (a) Vista frontal, (b) vista superior. Realización de los autores a partir de <i>Solid Edge</i> ®.	44
Figura 26: Modelo del manipulador <i>KUKA KR 6</i> en <i>Solid Edge</i> ®.....	46
Figura 27: Modelos de la banda transportadora (a) y la cámara <i>web</i> (b). Imagen tomada por los autores desde <i>Solid Edge</i> ®.	46
Figura 28: Entorno de simulación con todos los modelos. Imagen tomada por los autores desde <i>Solid Edge</i> ®.	47
Figura 29: Menú del ambiente <i>Motion</i> de <i>Solid Edge</i> ®. Tomado del <i>software</i>	48
Figura 30: Procedimiento para importar los resultados de visión artificial a <i>Solid Edge</i> ®. Realización de los autores.	49
Figura 31: Geometría utilizada. Triángulo rectángulo.....	50

Figura 32: Trayectoria seguida por el manipulador <i>KUKA KR 6</i> a partir de los resultados del sistema de visión artificial. (a) Vista superior, (b) vista frontal, (c) vista lateral.	51
Figura 33: Resumen capítulo 5.	52
Figura 34. Interfaz software KRC.	54

LISTA DE TABLAS

Tabla 1: Especificaciones técnicas de la cámara web Logitech HD C615. Tomado de [29].	37
--	----

GLOSARIO

CALIBRACIÓN: en un proceso de visión artificial, la calibración geométrica de un cámara consiste en obtener ciertos parámetros de la misma, que sirven para recuperar información del mundo real (3D) y realizar una correcta representación de éste, a partir de una imagen (2D) [1].

MANIPULADOR INDUSTRIAL: es un manipulador automáticamente controlado, reprogramable y multipropósito [2], “diseñado para mover materiales, partes, herramientas o dispositivos especiales a través movimientos variables programados, para el desarrollo de tareas diversas” [3].

PROCESAMIENTO DE IMÁGENES: es el conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorarlas en algunos aspectos que faciliten la obtención posterior de información.

SIMULACION: “La simulación es el proceso de diseñar un modelo de un sistema real y llevar a término experiencias con él, con la finalidad de comprender el comportamiento del sistema o evaluar nuevas estrategias -dentro de los límites impuestos por un cierto criterio o un conjunto de ellos - para el funcionamiento del sistema”[4].

SISTEMA DE VISION ARTIFICIAL: sistema que pretende reproducir artificialmente el sentido de la vista mediante el procesamiento e interpretación de imágenes adquiridas por medio de algún dispositivo de adquisición, y utilizando para ello las prestaciones de los ordenadores [5].

RESUMEN

En este proyecto se simula la integración entre un sistema de visión artificial con un manipulador robótico serial, con el fin de que éste seleccione y posicione cuerpos con formas geométricas simples que se encuentran sobre una banda transportadora en movimiento. Para ello, se desarrolló un código en el *software Matlab*® capaz de realizar las tareas de procesamiento de imágenes necesarias para extraer el centroide del objeto deseado sobre la banda transportadora, y posteriormente realizar un análisis de posición inverso, donde se genera la trayectoria que debe seguir el manipulador para posicionar la figura. Este código presenta como principales características la posibilidad de elegir entre figuras como triángulos, rectángulos, círculos, entre otras; y que el centroide hallado se encuentra en medidas de longitud real (i.g. milímetros).

INTRODUCCIÓN

Debido al crecimiento acelerado de la tecnología y por ende de los procesos industriales automatizados, es de gran importancia para la Universidad comenzar a abordar temas como la robótica y la visión artificial, pues son dos campos relativamente nuevos que serán parte fundamental en las próximas tecnologías y desarrollos.

Aunque la integración de manipuladores robóticos con sistemas de visión artificial para actuar sobre objetos en movimiento ha sido profundizada en otros países, en Colombia aún hay un gran campo por explorar. Es por lo anterior que se ha decidido comenzar a trabajar en este tema, buscando generar unas bases importantes y empezar a construir dicho conocimiento en la universidad, facilitando así próximos trabajos que quieran continuar con la investigación.

El trabajo entonces busca abordar los dos temas ya mencionados, integrarlos y generar resultados que puedan ser punto de partida para futuros procesos industriales. En sí el trabajo simula la integración de un manipulador robótico con un sistema de visión artificial, el cual debe ser capaz de identificar un objeto sobre una banda transportadora en movimiento y reposicionarlo en algún otro lugar.

Para lo anterior, el trabajo se divide en varios capítulos, los cuales deberán ser estudiados uno a uno para poder lograr el objetivo del mismo. Para facilitar el trabajo del lector, se presentarán mapas conceptuales al inicio de cada capítulo que muestran, en términos generales, el desarrollo del mismo. Inicialmente se presentan los conceptos básicos tanto de los sistemas de visión artificial como de los manipuladores robóticos, esto se presentara en los capítulos 1 y 2 respectivamente. Más adelante se presenta en el capítulo 2, toda la teoría referente a la calibración del sistema de adquisición de imágenes y luego de esto se muestran los resultados y análisis obtenidos en el capítulo número 4. En el último capítulo, se plantean recomendaciones acerca de los pasos a seguir, para la futura continuación de este trabajo.

Cabe aclarar que aunque esto ya puede haber sido desarrollado en otro lugar, es importante ser partícipes del desarrollo, es necesario el hacer para poder entender y más adelante, ser capaces de crear.

1. INTRODUCCIÓN A LOS SISTEMAS DE VISIÓN ARTIFICIAL

Según Dykinson [5], la visión artificial se define como “un área multidisciplinar que pretende, en cierta medida, reproducir artificialmente el sentido de la vista mediante el procesamiento e interpretación de las imágenes, captadas con distintos tipos de sensores (fundamentalmente, cámaras), y utilizando para ello las prestaciones de los ordenadores.”

1.1 CONCEPTOS BÁSICOS

Es importante mostrar el proceso necesario a seguir para generar datos precisos sobre posiciones de objetos por medio del uso de la visión artificial o procesamiento de imágenes. Para ello se muestra en la Figura 1 un diagrama en el que visualizan las diferentes etapas y el orden de ejecución de las tareas de un sistema de visión.

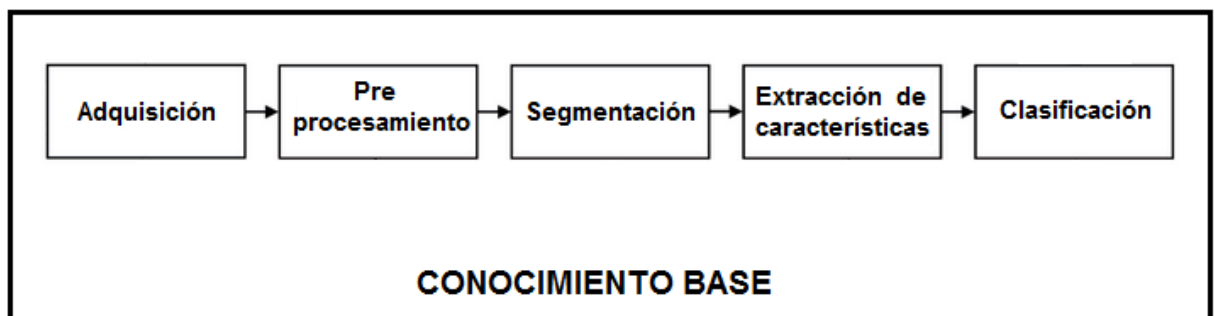


Figura 1: Etapas de un sistema de visión artificial. Realización de los autores.

A continuación se observará en mayor detalle cada una de las etapas:

Se comienza con la etapa conocida como *adquisición*. En ésta, se busca obtener las imágenes que contienen los objetos a analizar, por medio de dispositivos como cámaras, escáneres, videograbadoras, etc., los cuales puedan enviar la imagen adquirida al computador en forma de información, que será usada posteriormente para continuar el proceso.

Luego de tener la información de las imágenes, se puede realizar un pre-procesamiento en caso de que éstas lo requieran. Éste se compone de varias operaciones que pueden ser aplicadas o no, dependiendo del estado de la imagen. En los casos en los que las imágenes se adquieren en espacios controlados (iluminación adecuada que mejore el contraste, sin partículas visibles en el aire, y demás fenómenos que afecten la imagen), un pre-procesamiento riguroso no es necesario, o en un caso extremo, no sería necesaria esta etapa en absoluto. Esto sería positivo para la agilización del procedimiento.

Entre las principales operaciones de *pre-procesamiento* para el mejoramiento de la imagen se encuentran las siguientes:

Binarización: en esta operación se busca pasar de una imagen en escala de grises a una imagen en la que cada uno de los píxeles que la conforman, solo toman valores de 0 ó 1. Dicho de otro modo, la imagen queda completamente en blanco y negro (ver Figura 2).

Mejoramiento del contraste: “Mejoramiento de la iluminación y/o del color de la imagen para que los objetos en ésta se distingan fácilmente” [5], esta operación busca, por medio cambios en la iluminación, mejorar la diferencia entre el objeto y el resto de la imagen.



(a)



(b)

Figura 2: Binarización de una imagen. (a) Imagen a color. (b) Imagen binaria. Tomado de [6].

Además de éstas, existen más operaciones como: reducción de ruido, enfoque, desenfoque, agudizado, etc., pero las que realmente se usarán en este trabajo se exponen en el capítulo 4.

Comprendidas y utilizadas las técnicas de procesamiento de imágenes, y por lo tanto teniendo imágenes más fáciles de analizar, se procede con la siguiente etapa, en la que se debe realizar una *segmentación*. Según Fernández [7] “en Visión Artificial la etapa de segmentación determina los bordes o fronteras que delimitan la forma de un objeto captado por una cámara para un fin determinado”. Es decir, que es necesario generar unas fronteras entre los objetos que deben ser analizados y la demás partes de la imagen que no aportan nada al proceso (ver Figura 3).

Para la segmentación se cuenta también con una gran cantidad de técnicas. Tomando la clasificación presentada en [8] se encuentran:

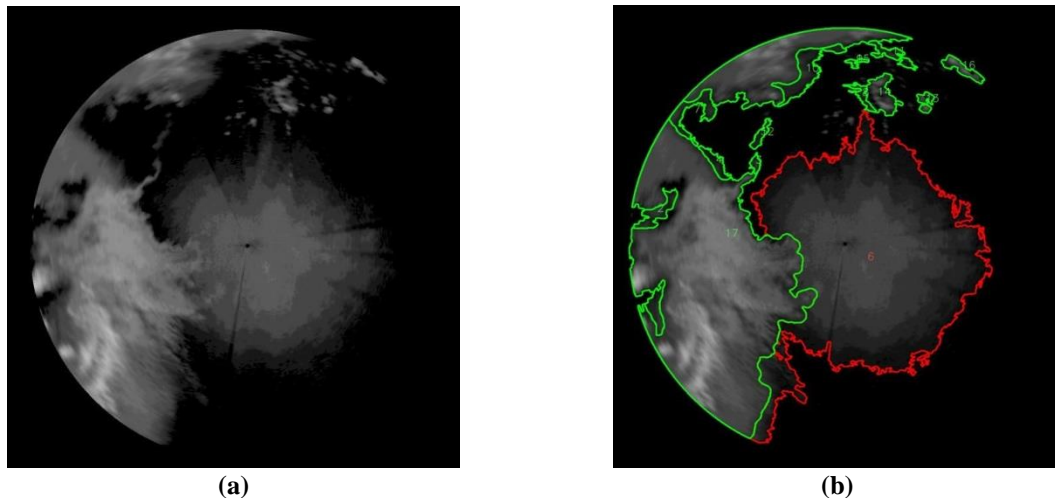


Figura 3: Ejemplo de segmentación. (a) Imagen original. (b) Imagen segmentada por regiones. Tomado de [9].

Métodos basados en regiones: se basa en el criterio de conectividad y cercanía entre un pixel y sus pixeles vecinos. Dos de las operaciones que mejor se adaptan en este grupo son: “región creciente” (*region growing*) y “dividir y combinar” (*split and merge*).

Otros métodos de segmentación basados en textura, bordes y movimiento, etc., son agrupados en otra clasificación. Los métodos usados para el trabajo se exponen en el capítulo 4.

Teniendo ya separados los objetos de una imagen, es de gran importancia conocer las características que éstos tienen y los diferencian de los demás. Por lo mismo, se realiza la siguiente etapa que se encarga de la *extracción de características*. Es bueno entonces conocer algunas de las propiedades, entre muchas, se destaca:

Invariancias: las características de los objetos pueden ser invariantes, dependiendo de la aplicación que se vaya a trabajar. Algunas de las características invariantes más utilizadas son los invariantes RST (*rotation, scaling and translation*). Esta propiedad permite que cualquier tipo de objeto pueda ser identificado sin importar la posición, la orientación y el grado de acercamiento que se tenga en la imagen.

Por último, es necesario realizar una clasificación adecuada de los objetos encontrados en la imagen, por medio de los valores obtenidos de la extracción de características, con el fin de seleccionar el elemento de estudio y así, tener la información requerida para cualquier operación posterior. Para esta etapa, también existen varias técnicas que permiten realizar dicha clasificación de diferentes formas, pero obteniendo resultados similares. Entre ellas se resalta el *reconocimiento de patrones* o *clasificación de patrones*. El objetivo principal de esta técnica, se basa en determinar la clase o el objeto que se tiene en la imagen, por medio de una comparación con patrones que representan el objeto o la clase adecuada para el análisis y que son previamente elegidos.

En la Figura 4 muestra el orden común de ejecución de las tareas de un sistema de visión artificial.

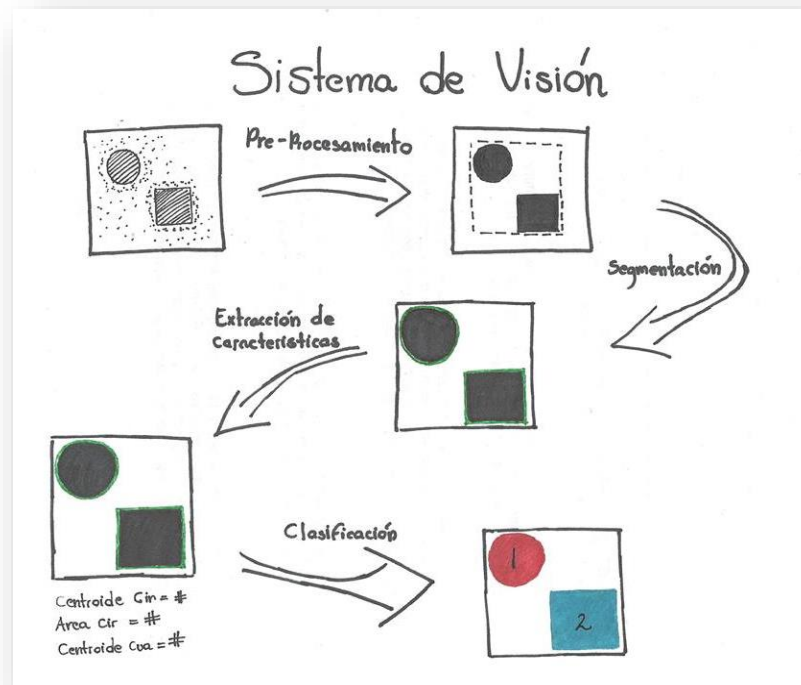


Figura 4: orden común de ejecución de las tareas de un sistema de visión artificial.

1.2 POSICIONAMIENTO DE LA CÁMARA

La cámara encargada de tomar las imágenes de los objetos puede tener dos ubicaciones con respecto al robot. Una de éstas es "hand-in-eye", donde la cámara se ubica en un soporte fijo externo al robot, y la posición del objeto se representa por medio de un marco de referencia fijo o absoluto. La otra ubicación es "eye-in-hand", donde la cámara se ubica en el efector final del manipulador (junto con la herramienta). Según esto la posición de un objeto será determinado por medio de una marco de referencia móvil o relativo.

1.3 TOOLBOX DE MATLAB PARA PROCESAMIENTO DE IMÁGENES

Matlab, software matemático ampliamente usado dentro de la ingeniería, posee una serie de herramientas llamadas *Toolboxes*. Dentro de ellas, se encuentra una enfocada al procesamiento de imágenes conocida como *Image Processing Toolbox (IPT)*. Este toolbox

ofrece una agrupación de funciones que aumentan la capacidad básica de *Matlab* y están relacionadas con el procesamiento de señales e imágenes, para permitir operaciones especializadas en el campo en cuestión [8]. Entre las funciones que se encuentran en este entorno se destacan algunos grupos como: transformaciones espaciales, filtros lineales y de diseño, transformaciones matemáticas, operaciones morfológicas, etc., las cuales hacen parte del proceso de visión artificial explicado anteriormente.

2. INTRODUCCIÓN A LA TEORÍA DE MANIPULADORES ROBÓTICOS

Los manipuladores robóticos o robots industriales hacen parte importante del desarrollo de la industria manufacturera actual, en especial en la automatización de procesos. Es mucho lo que se puede estudiar acerca de estas máquinas, sin embargo, en este capítulo se hará una breve introducción en este tema, con el fin de dar a conocer al lector los conceptos básicos que sean necesarios durante la lectura del presente trabajo.

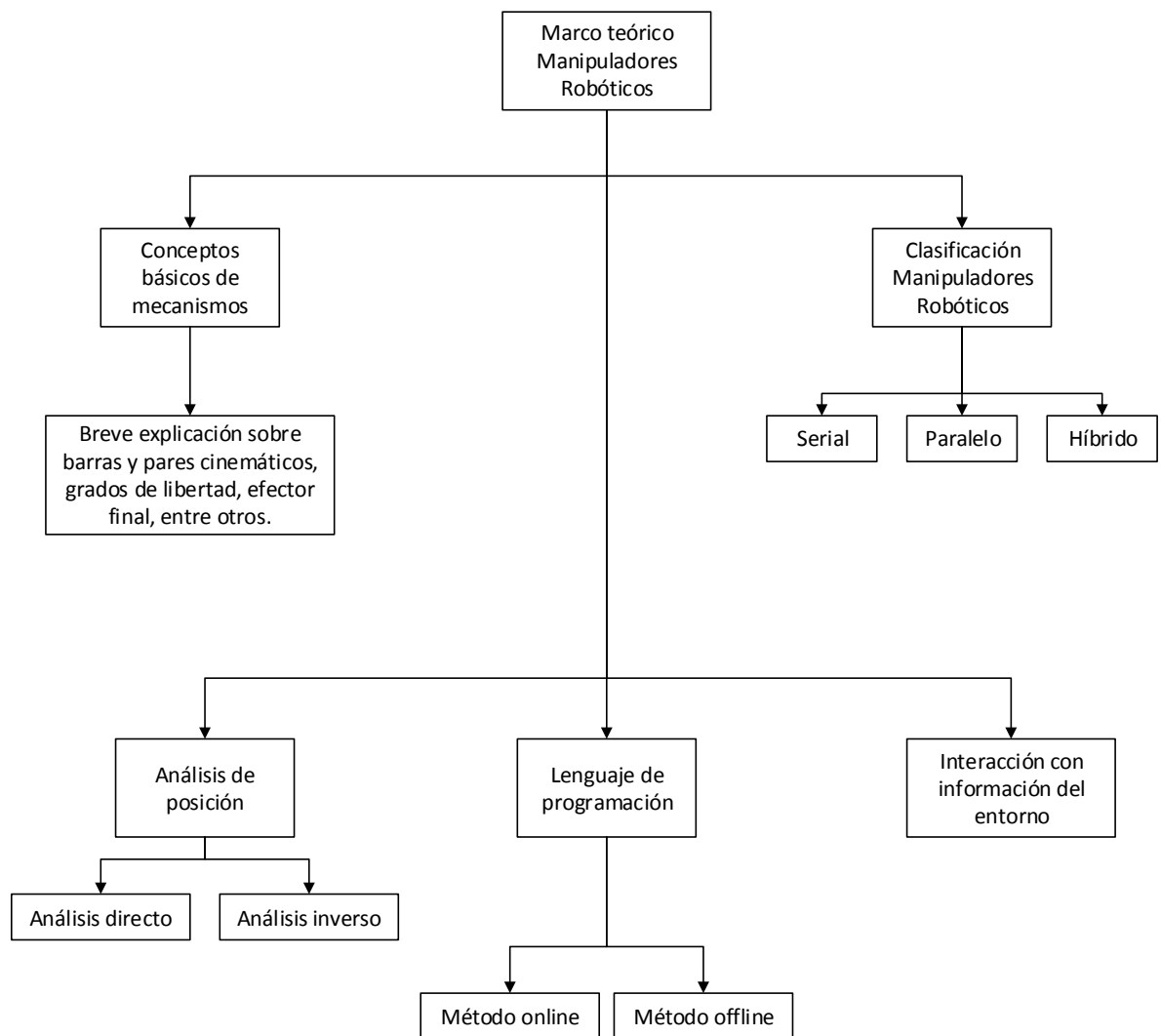


Figura 5: Resumen capítulo 2.

2.1 ROBOTS INDUSTRIALES

Según la Organización Internacional de Normalización ISO, un robot industrial es un manipulador automáticamente controlado, reprogramable y multipropósito [2], “diseñado para mover materiales, partes, herramientas o dispositivos especiales a través movimientos variables programados, para el desarrollo de tareas diversas” [3].

Para poder clasificar los distintos manipuladores robóticos es necesario introducir una serie de conceptos básicos de mecanismos.

2.1.1 Conceptos básicos de mecanismos y manipuladores robóticos.

Un mecanismo está hecho de varias *barras* o eslabones, conectados entre sí por medio de *articulaciones* o *pares cinemáticos* [3], [10]. A este arreglo se le considera una cadena cinemática. Las barras están unidas de tal forma que no existe movimiento relativo entre ellas. Por lo anterior, se consideran como cuerpos rígidos, incluso si se tiene, por ejemplo, dos poleas sobre un mismo eje [10].

Las articulaciones o pares cinemáticos permiten ciertos movimientos entre las barras que conectan. Se consideran seis tipos de pares en la teoría de mecanismos y manipuladores robóticos, siendo los pares de revoluta (*R*) y prismáticos (*P*) los más usados generalmente. Un par *R* permite la rotación relativa entre dos barras sobre un mismo eje, por ejemplo, las bisagras de un portón. Un par *P* permite el deslizamiento relativo entre dos barras sobre un eje, por ejemplo, un pistón que se desplaza dentro de un cilindro.

Los *grados de libertad* (DOF, *degrees of freedom*), son el número de variables independientes o mínimas para describir completamente la posición o configuración de un mecanismo [11]. El *efector final* de un manipulador es la herramienta instalada al extremo libre de éste, usada para agarrar, levantar y manipular piezas. Herramientas más especializadas, como pistolas de aerografía, electrodos de soldadura o herramientas de corte, también son considerados efectores finales (ver Figura 6).



(a)



(b)



(c)



(d)

Figura 6: Ejemplos de efectores finales. (a) Pinza de vacío. (b) Pinza neumática. (c) Pinza hidráulica. (d) Pinza servo-eléctrica. Tomado de [12] y editado.

Los *actuadores* proporcionan movimiento al manipulador y por ende al efector final. En robótica, los actuadores más usados son eléctricos. Por ejemplo, motores paso a paso (*stepper motors*) o servomotores (*servomotors*). Los últimos permiten un mayor control de posición y velocidad [13].

2.2 CLASIFICACIÓN DE LOS MANIPULADORES.

Los manipuladores robóticos se pueden clasificar según diversos parámetros, como sus grados de libertad, aplicaciones, estructura cinemática, entre otros. Para este proyecto, se clasificarán principalmente según su estructura cinemática, esto es: serial, paralelo o híbrido.

Se considera que un manipulador es *serial* si su estructura cinemática toma forma de una cadena de lazo abierto. Es decir, cada barra está conectada a otra por uno y solo un camino, usando generalmente articulaciones tipo R y P. Uno de los extremos del manipulador, la base, está fijado a la tierra, y el otro, el efector final, está libre para maniobrar en el espacio. Este tipo de robot industrial también es llamado brazo robótico, por su similitud con un brazo humano (ver Figura 7).

Un manipulador *paralelo* posee una estructura de cadena de lazo cerrado. Es decir, una barra está conectada a otra por dos o más caminos. Consiste en una plataforma móvil conectada a una base por medio de varias “extremidades” (ver Figura 8). Cada una de las extremidades es controlada por un actuador independiente, que está cerca o sobre de la base del robot. Usualmente, por cada extremidad hay un DOF [10].

Dado a que la carga es distribuida en cada una de las extremidades, el manipulador paralelo tiene mayor capacidad de carga, rigidez y menor inercia que un manipulador serial.

Un *manipulador híbrido* es aquel que tiene configuración tanto serial como paralelo (ver Figura 9).

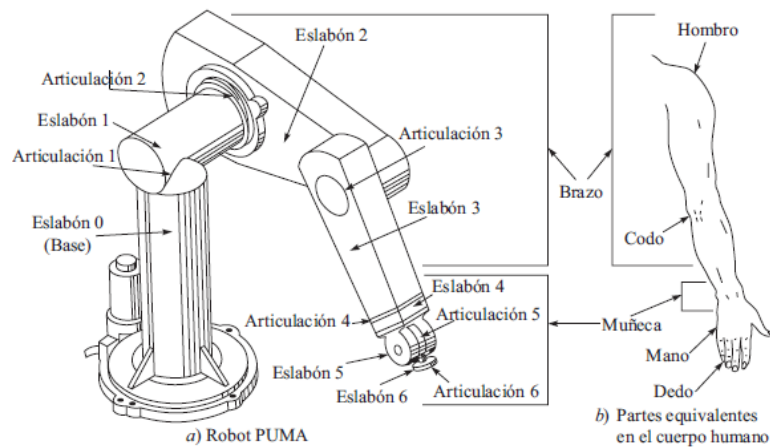


Figura 7: Semejanza brazo robótico serial con brazo humano. Tomado de [3].

2.3 ANÁLISIS DE POSICIÓN DIRECTO E INVERSO

El *análisis de posición* permite indicar la posición del efector final con respecto a la base del robot. Se identifican dos problemas típicos: *análisis de posición directo* o *cinemática directa*, y *análisis de posición inverso* o *cinemática inversa*. En el primer caso, se conoce la ubicación de las articulaciones y se desea saber la del efector final. En el segundo, se conoce la posición del efector final y se desea saber la posición de las articulaciones.

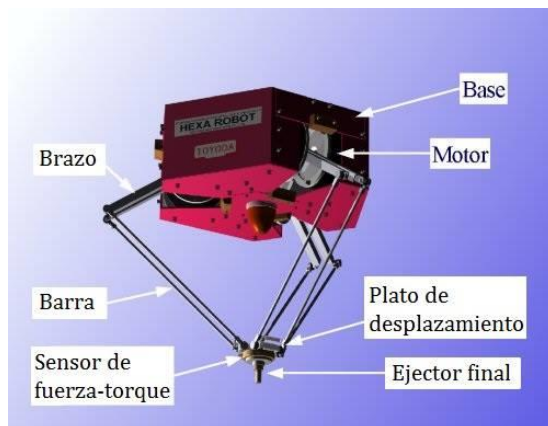


Figura 8: Ejemplo de robot paralelo. Tomado de [14].



Figura 9: Ejemplo robot híbrido. Tomado de [15].

Ambos tipos de análisis son especiales para cada tipo de robot, y en particular el grado de dificultad del análisis inverso, varía dependiendo del tipo de robot. En el presente proyecto, se realiza un estudio más detallado en el análisis de posición inverso para un manipulador serial, pues el robot a utilizar es el *KUKA KR 6* de la Universidad Pontificia Bolivariana.

2.4 LENGUAJE DE PROGRAMACIÓN

Una de las grandes ventajas de los robots industriales es que son reprogramables, y por lo tanto flexibles. El lenguaje de programación es el que permite la comunicación entre el ser humano y el manipulador, y es a través de éste que se le indica al robot la tarea que debe realizar.

Dentro de los distintos métodos de programación, se reconocen los métodos online y offline [3]. En la programación online, el usuario guía el robot a través de puntos que

conforman una trayectoria deseada, haciendo que el robot la memorice y la reproduzca posteriormente de manera autónoma. Es un método muy usado en la práctica, pues es de fácil acceso a cualquier usuario y de programación sencilla en comparación con los métodos offline. Sin embargo, esto implica que el robot se detenga para poder enseñarle los nuevos movimientos, lo que representa pérdidas dentro de un contexto productivo. A este método también se le conoce como “enseñar mostrando” [11].

En el método *offline*, la programación del robots se hace por medio de algoritmos y permite el uso de datos provenientes de simulaciones CAD [3]. Se clasifica principalmente en dos categorías: *programación orientada al robot* o *lenguajes de programación de robots (RPL, robot programming language)*, en donde cada línea de programación indica explícitamente el movimiento y la tarea del robot; y la *programación basada en las tareas*, en donde al robot se le indican puntos a donde se desea llegar o la tarea que debe realizar, más no se le indica la trayectoria, dejando que ésta sea elegida por el mismo con la ayuda de una base de datos (modelos de entorno). Los lenguajes de programación pueden ser los usados en computación, como C, C++, M, etc., o más especializados como KRL para manipuladores KUKA [16].

El método *offline* no implica que el manipulador se detenga durante los períodos de producción, pues la programación se hace con anterioridad y puede ser simulada para verificar que sea correcta. Permite mayor precisión, pero implica mayor conocimiento y no cualquier persona puede realizarlo. Además, dependiendo si el lenguaje de programación es especializado o no, involucra mayor inversión económica en software.

2.5 SENSORES ADICIONALES USADOS EN ROBÓTICA

Como lo es para los seres humanos, los robots necesitan medir diferentes variables que se presentan en el ambiente, con el fin de realizar diferentes movimientos de manera efectiva. Es indispensable para los robots tener sensores que les permitan detectar obstáculos, conocer sus propios límites de movimiento o reconocer algunos cambios, por ejemplo,

pesos adicionales en el sistema. Además, deben ser capaces de generar movimientos con velocidades y aceleraciones controladas, previniendo daños a sí mismos.

Entre los sensores usados en los robots se tienen de proximidad, de posición, de visión, etc., que ayudan a realizar tareas como detección de objetos. En el proyecto, se utiliza un sensor óptico para el sistema de visión, o más específicamente, una cámara *web*. En el capítulo 4 se abordará el tema con más detalle.

3. CALIBRACIÓN

El proceso de calibración de cámara, es un tema que aparentemente no es de mayor importancia, sin embargo, es un paso que debe ser ejecutado cuidadosamente en una aplicación de visión artificial. A continuación se presentan aspectos como: qué es calibrar una cámara y su importancia, qué parámetros se tienen en cuenta, qué tipo de calibraciones existen, entre otros.

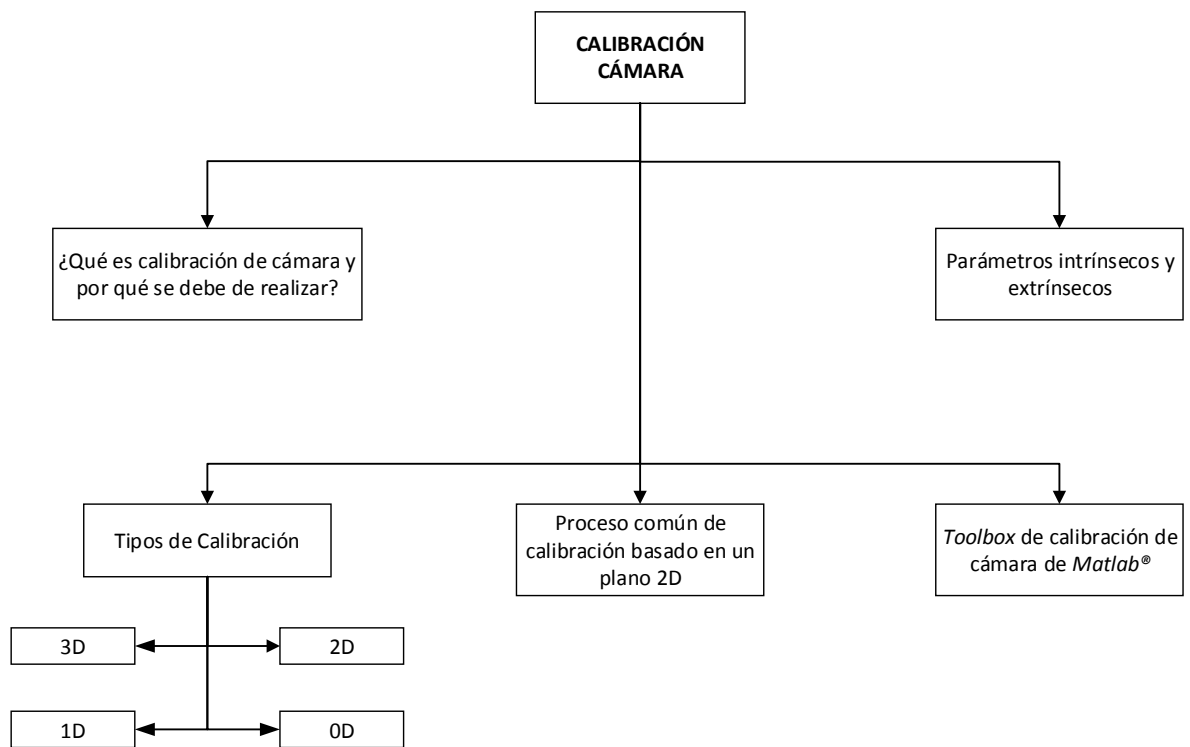


Figura 10: resumen capítulo 3.

3.1 ¿QUÉ ES CALIBRACIÓN DE CÁMARA Y POR QUÉ SE DEBE REALIZAR?

En un proceso de visión artificial, la calibración geométrica¹ de un cámara consiste en obtener ciertos parámetros de la misma, que sirven para recuperar información del mundo real (3D) y realizar una correcta representación de éste, a partir de una imagen (2D) [1]. Estos parámetros son conocidos como intrínsecos y extrínsecos, de los cuales se hablará más adelante.

Una de las ventajas de calibrar, es poder obtener una relación entre las medidas de longitud reales (e.g. metros o pulgadas) y las medidas digitales (e.g. pixeles). Otra ventaja significativa, es poder determinar la posición y orientación de un objeto en el espacio con respecto al plano de la cámara, o viceversa. Estos aspectos son cruciales en aplicaciones como [17] fotogrametría, que es realizar mediciones a partir de fotografías [18]; y en *stereo visión* (visión estéreo), “proceso de extraer información 3D a partir de múltiples imágenes de una escena” [19].

¿Se puede entonces trabajar sin calibrar? esto depende de la aplicación. Si el objetivo es interpretar el mundo real a partir de imágenes, como es el caso del presente trabajo, entonces la respuesta es no. Es importante notar que incluso, en ocasiones, es necesario repetir la calibración. Por ejemplo, cuando hubo un cambio de posición de la cámara, una variación de enfoque, o un aumento o disminución del zoom del dispositivo. Si no se recalibra, los resultados podrían tener un error significativo que conllevan a resultados indeseados.

3.2 PARÁMETROS INTRÍNSECOS Y EXTRÍNSECOS

Los parámetros intrínsecos y extrínsecos son aquellos que se hallan durante un proceso de calibración. Los primeros, también llamados “parámetros internos”, son característicos de la cámara. Algunos de ellos son:

¹ También existe la calibración radiométrica, de la cual no se trata en el presente documento porque está más allá de los objetivos y alcances del mismo.

- Longitud focal: “distancia entre el centro óptico de lente de la cámara y el sensor de imagen de la cámara, cuando el lente es enfocado al infinito.”[20]. En otras palabras, este parámetro se relaciona con el enfoque de la cámara, y contiene qué tanto de la escena real será capturada y qué tan grandes serán los elementos (ángulo de visión y magnificación).
- Factores de escala: relación entre las dimensiones físicas de un objeto en el mundo real a las correspondientes en una imagen [21].
- Coeficiente de oblicuidad: describe la pérdida de ortogonalidad entre los ejes de la imagen [22].
- Distorsiones: defectos que se presentan en ciertas ocasiones, debido a imperfecciones en el lente de la cámara [23]. Existen dos tipos: radial y tangencial. Las distorsiones son parámetros importantes si se trabaja con cámaras *web*.

El segundo tipo de parámetros, también llamados “parámetros externos”, corresponden a la orientación y posición de la cámara.

Dependiendo del tipo de calibración que se realice, los extrínsecos pueden ser necesarios o no, mientras que los intrínsecos siempre serán necesarios [24].

3.3 TIPOS DE CALIBRACIÓN

Diferentes técnicas de calibración son desarrolladas día a día a nivel mundial. Varios autores han realizado una breve clasificación en sus publicaciones ([17], [22], [25] por citar algunos). *Zhengyou Zhang* hace una clasificación genérica y clara [24], y es la que se usa a continuación de forma general en el presente trabajo.

3.3.1 Calibración basada en un objeto de referencia 3D

Proceso en el que se usa como objeto de referencia dos o tres planos ortogonales, en los cuales ha impreso un patrón, como se muestra en la Figura 11. Las coordenadas del patrón

son conocidas con mucha precisión respecto a un sistema coordenado. Presenta como desventaja que la configuración para calibrar debe ser muy elaborada.

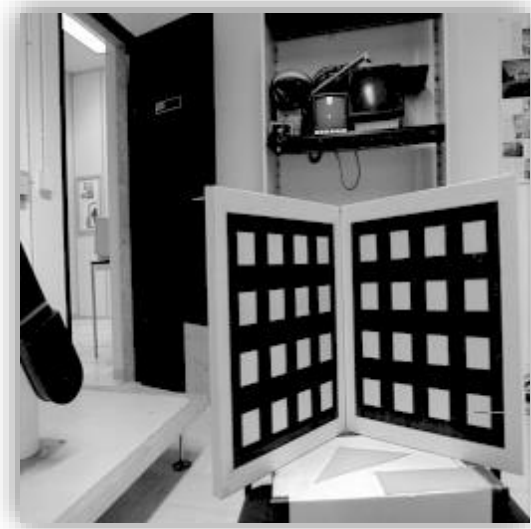


Figura 11: Objeto usado para la calibración 3D. Tomado de [24].

3.3.2 Calibración basada en un plano 2D

Esta técnica requiere un patrón en dos dimensiones como el mostrado en la Figura 12, ubicado en diferentes posiciones y orientaciones. Presenta como ventaja que el proceso es fácil de realizar y la precisión obtenida es bastante buena. Es de notar que éste es el que se usa en el *Toolbox* de *Matlab*®, y en este documento.

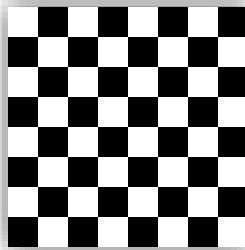


Figura 12: Patrón usado para la calibración 2D. Tomado de [26].

3.3.3 Calibración basada en una línea 1D

Técnica en la que el objeto de calibración es un conjunto de puntos colineales, como se observa en la Figura 13, en donde la calibración se hace a partir de la observación de la línea que forman los puntos en movimiento, desde un punto fijo. A mayor número de puntos, mejor será el resultado de calibración.

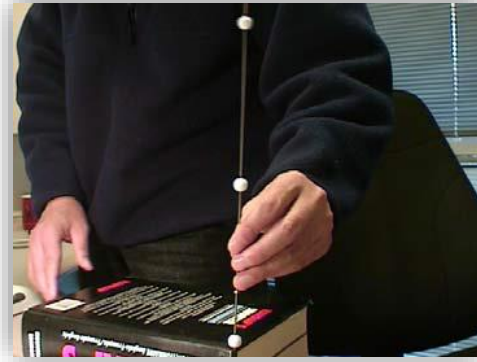


Figura 13: Objeto usado para la calibración 1D. Tomado de [9].

3.3.4 Auto-calibración

También puede conocerse como un acercamiento 0D. Técnica que usa la correspondencia entre puntos de una imagen, a partir del movimiento de la cámara en un escenario estático. Este método solo requiere de los parámetros intrínsecos de la cámara, y no necesita objeto de calibración a diferencia de las técnicas anteriores. Presenta como desventaja que requiere mucha computación matemática, ya que se deben determinar un gran número de parámetros.

Entre las técnicas mencionadas, solo se explicará en mayor detalle la segunda: calibración basada en un plano 2D, pues es la que le concierne a este trabajo. Es de notar que no se harán tratamientos matemáticos, pues estos están fuera de los objetivos del documento. En caso de requerir mayor información en tal aspecto, se recomienda referirse a [9] y [12].

3.4 PROCESO COMÚN DE CALIBRACIÓN BASADA EN UN PLANO 2D

Según *Zhang*, el proceso recomendado de calibración para esta técnica es el siguiente:

1. Imprimir un patrón de referencia (e.g. Figura 12), y fijarlo a una superficie plana. Imperfecciones sobre la superficie pueden afectar la calibración.
2. Tomar varias imágenes del patrón en posiciones y orientaciones diferentes, o moviendo la cámara.
3. Detectar los puntos característicos en las imágenes (ver Figura 14).
4. Determinar los parámetros intrínsecos y extrínsecos que sean necesarios para la calibración. Esto se hace con procedimientos analíticos.
5. Estimar los coeficientes de distorsión radial. También con métodos analíticos.
6. Refinar los parámetros hallados.

Otros procedimientos dentro de esta categoría están siendo investigados, por ejemplo, el uso de esferas para calibrar en lugar de planos, dado a que una esfera es simétrica, y su silueta es la misma, sin importar la orientación [25].

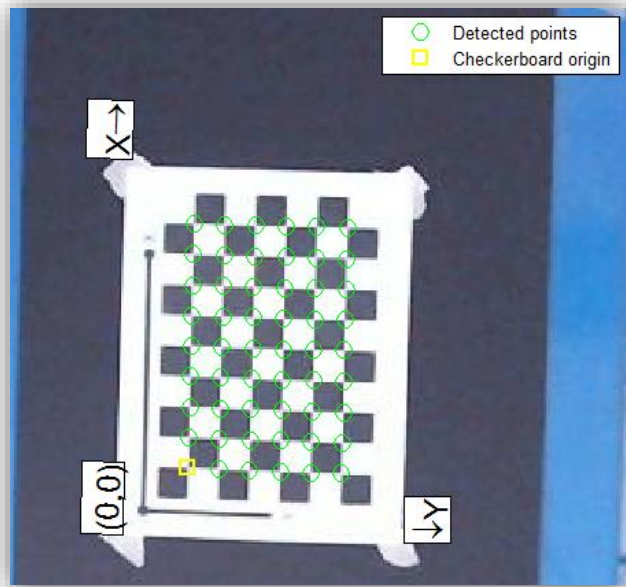


Figura 14: Puntos característicos detectados en la imagen. Elaboración de los autores con ayuda de *Matlab*®.

3.5 TOOLBOX DE CALBRACIÓN DE CÁMARA DE MATLAB®.

Matlab® posee una aplicación especial para calibrar cámaras dentro de su *toolbox Computer Vision System* (Sistema de Visión Artificial) llamada *Camera Calibrator* (calibrador de cámara). Los pasos para utilizar esta herramienta son en esencia los mismos mencionados en la sección anterior, con la ventaja de que el usuario solo tiene que realizar los pasos 1 y 2, y en ciertas ocasiones el 6, y el *toolbox* se encarga de realizar todos los cálculos necesarios para la calibración. Esta aplicación también ofrece la posibilidad de elegir opciones como calcular el coeficiente de oblicuidad o elegir coeficientes de distorsión radial y tangencial (ver Figura 15).

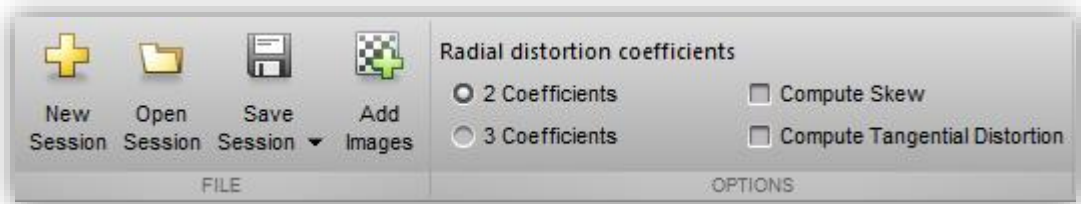


Figura 15: Menú principal del *Camera Calibrator App* de *Matlab*®. Tomado de [28].

Una vez calibrada la cámara, el *software* ofrece la posibilidad de guardar los parámetros resultantes para un uso futuro, o simplemente almacenarlos temporalmente en el *workspace* del programa.

4. SIMULACIÓN

En este capítulo se mostrará el procedimiento y los resultados obtenidos durante el trabajo. El experimento tiene como objetivo reconocer el centroide de objetos sobre una banda transportadora en movimiento por medio del procesamiento de imágenes, y que dichos centroides sirvan de información para la trayectoria que deba seguir el manipulador elegido. Posteriormente, el seguimiento de la trayectoria, y otros movimientos que debe realizar el robot serán simulados en un *software CAD*

El capítulo cuenta con las siguientes etapas: calibración de cámara, obtención de centroides por medio del procesamiento de imágenes, análisis de posición inverso para el robot, y por último, simulación.

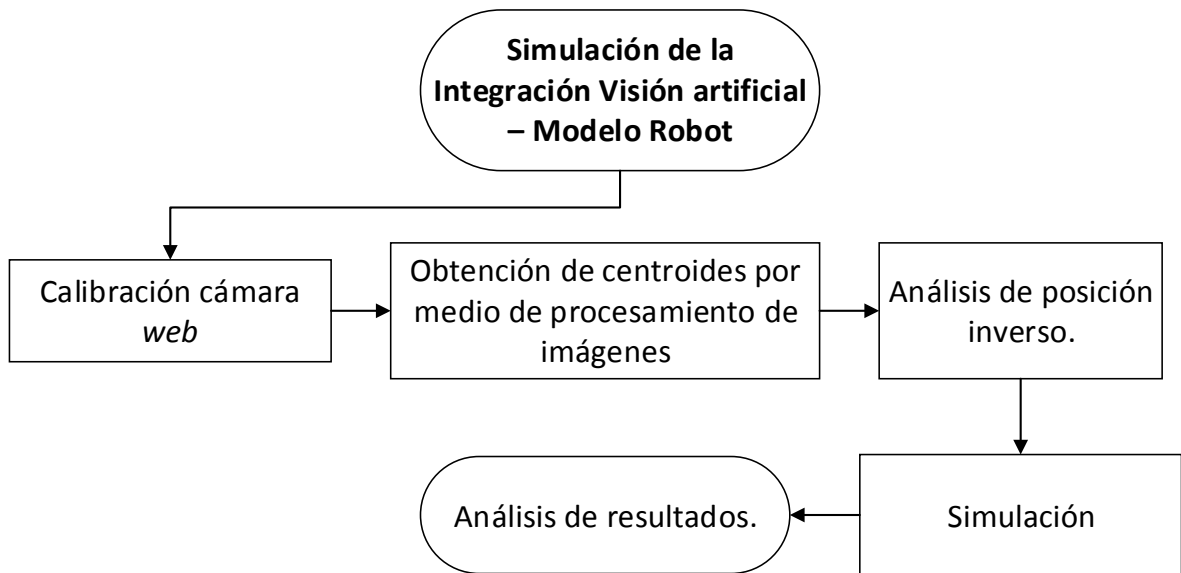


Figura 16: resumen capítulo 5.

4.1 CALIBRACIÓN DE CÁMARA

Como se mencionó en el capítulo 3, la calibración de cámara es un paso importante en algunas aplicaciones de visión artificial. Sabiendo esto, se selecciona una cámara que pueda cumplir con las tareas básicas de adquisición y procesamiento de imágenes, y que permita una fácil comunicación con el *software Matlab*®.

El dispositivo usado es la cámara *web Logitech HD C615* (ver Figura 17). Esta cámara cumple con las necesidades básicas mencionadas, y su resolución es aceptable para el trabajo. Los datos técnicos generales de ésta se muestran en la Tabla 1. Si el lector requiere conocer más información de la misma, puede referirse a [29].



Figura 17: cámara *web Logitech HD C615*. Tomado de [30].

Tabla 1: Especificaciones técnicas de la cámara *web Logitech HD C615*. Tomado de [29].

Especificaciones generales	
Tipo de conexión	USB 2.0.
Tipo de lente y sensor	Vidrio.
Tipo de enfoque	Automático.
Resolución óptica	Verdadera = 2MP, Mejorada con <i>software</i> = 8MP.
Campo de visión (FOV)	74°.
Longitud focal	4.4 mm ²
Dimensiones (L x W x H)	40.4 mm x 68.5 mm x 34.4 mm.
Peso	102.8 gramos.

² Medida basa en [41].

Para calibrar la cámara *web* se usa el *toolbox Camera Calibration* de *Matlab*®. El patrón utilizado es el que recomienda el *software* (ver Figura 18), impreso en una hoja tamaño carta, donde el lado de cada cuadrado mide 23 mm; la calibración se hace siguiendo los pasos básicos explicados en la sección 3.4 y 3.5.

Como resultado de las operaciones realizadas por el *toolbox*, se obtiene una estructura llamada *cameraParams* que contiene los parámetros intrínsecos de la cámara (los extrínsecos se calculan más adelante), que permitirán la corrección de las imágenes que se obtengan en pasos posteriores. Esta estructura se exporta al *Workspace* de *Matlab*® y se guarda como un archivo **.m*, para poder ser utilizado nuevamente. Es de recordar, que si la cámara cambia de posición, orientación o de *zoom*, deberá ser re-calibrada.

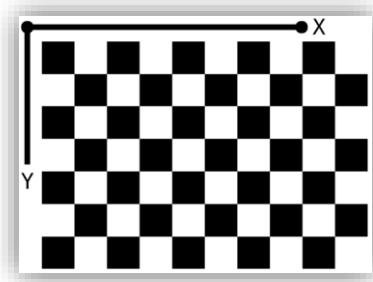


Figura 18: Hoja patrón recomendada por *Matlab*® para la calibración de cámara. Tomado de [28].

4.2 OBTENCIÓN DE CENTROIDES POR MEDIO DEL PROCESAMIENTO DE IMÁGENES

A continuación, se describe el procedimiento de procesamiento de imágenes y visión artificial, que involucra pasos como la inicialización del dispositivo de adquisición, obtención de propiedades de objetos en la imagen, transformación de coordenadas del mundo 2D a 3D, entre otras, hasta que se obtienen las coordenadas reales de los centroides. En el trabajo, la función que encargada de esto es nombrada como `Fn_ImageProcessing`.

Uno de los primeros pasos a realizar en un *script* que involucre procesamiento de video en *Matlab*®, es crear objetos del sistema y definir ciertas propiedades en éstos. En el trabajo, se crean objetos para la cámara *web*, que involucra propiedades como la región de interés o

la resolución deseada de la cámara; reproductor de video, texto y fondo del texto. Estos objetos permiten visualizar el resultado del centroide hallado en tiempo real, como se muestra en la Figura 19.

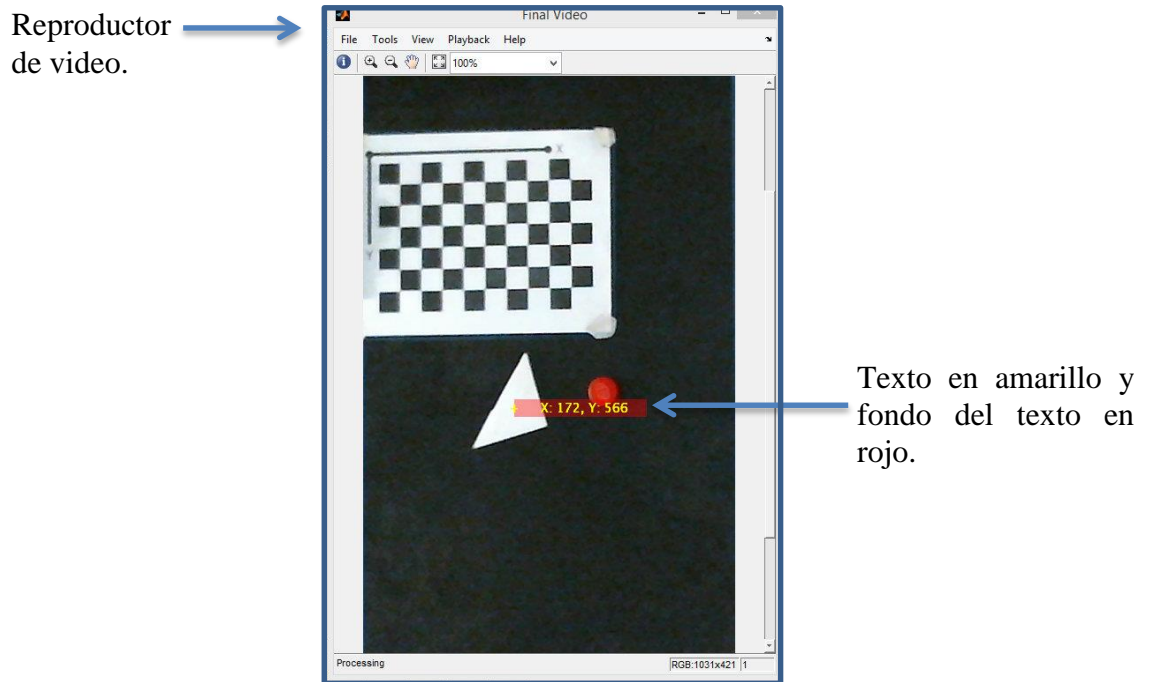
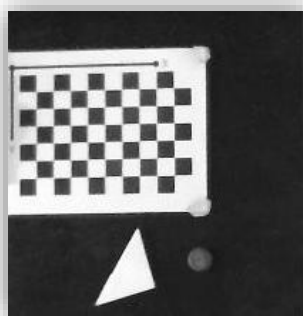


Figura 19: Objetos creados para procesamiento de video.

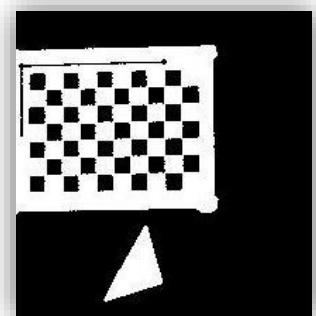
Luego se procede con la obtención de las imágenes deseadas (e.g. Figura 20a), las cuales son posteriormente pre-procesadas para facilitar la detección de centroides de los objetos en la misma. Este pre-procesamiento involucra la conversión de la imagen a escala de grises y luego a una binaria, como se muestra en la Figura 20.



(a)



(b)



(c)

Figura 20: Pre-procesamiento de las imágenes tomadas. (a) Imagen original, (b) imagen en escala de grises, (c) imagen binarizada.

Después de tener las imágenes depuradas, se procede con la segmentación, donde se extraen los bordes de las regiones continuas detectadas por medio de la función `bwboundaries` del *IPT* de *Matlab*® (ver Figura 21). A dichas regiones se les realiza una extracción de características por medio de la función `regionprops` del *IPT* de *Matlab*®, por ejemplo: área, cuadro delimitador (*bounding box*), o centroide en dirección x , y , siendo ésta última la buscada para generar la trayectoria que seguirá el robot en la simulación.

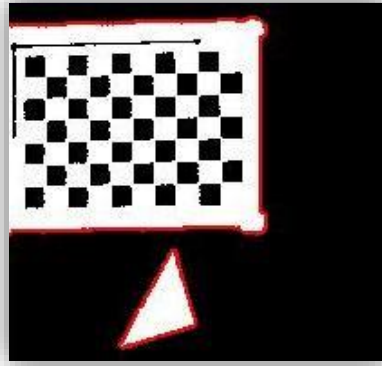


Figura 21: Extracción de bordes de regiones continuas.

Teniendo esto, se continúa con la fase de clasificación: determinar cuál de las figuras en la imagen es la que se desea posicionar con el manipulador. Para esto se realizan los siguientes pasos:

- Recorte de la imagen original en cada una de las figuras que se encuentran en ella, con la ayuda del *bounding box*, como se muestra en la Figura 22.
- Cálculo de momentos invariantes de las figuras recortadas, por medio de la función `feature_vec` [31]. Estos momentos fueron propuestos por primera vez por *Hu* en 1992 [8], y son característicos de cada imagen e invariantes a la rotación, traslación y escala de la figura.

- Comparación del primer momento invariante obtenido con un valor F previamente definido. Este valor corresponde al primer momento invariante de cada figura geométrica, y por lo tanto es diferente para todas.



Figura 22: Recorte de la imagen alrededor de su *bounding box*.

Es de notar que el código, gracias a los momentos invariantes, es capaz de clasificar entre diferentes geometrías y a diferente escala, como se muestra en la Figura 23.

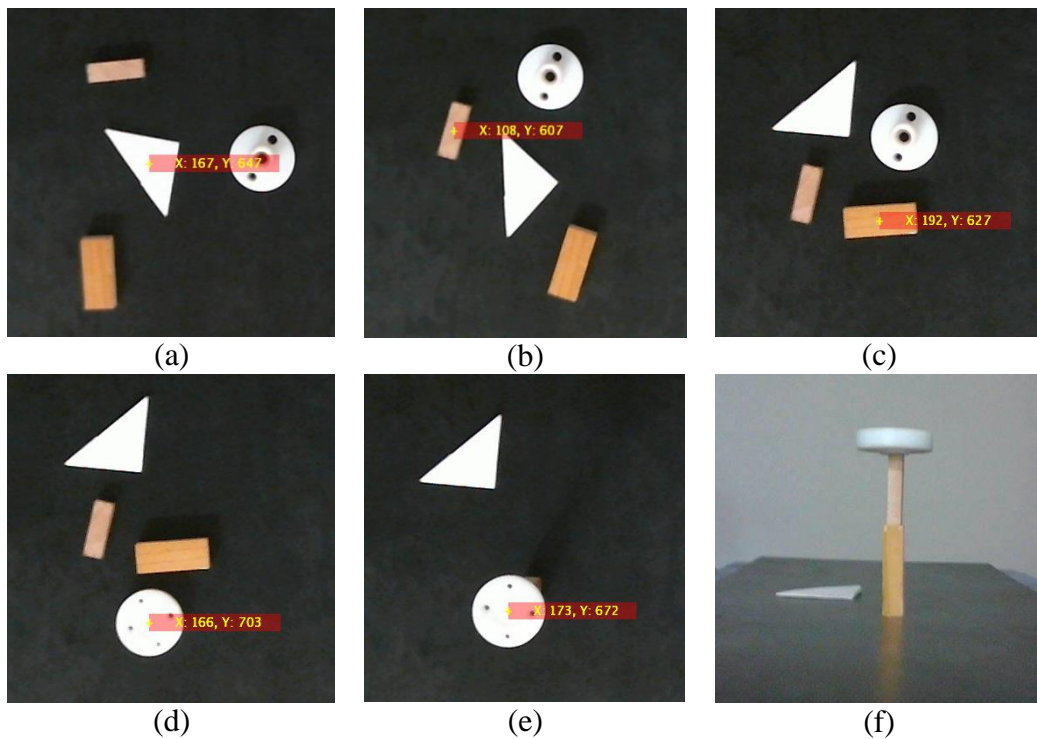


Figura 23: Identificación de centroides para diferentes geometrías. (a) Triángulo, (b) rectángulo 1, (c) rectángulo 2, (d) círculo, (e) círculo de la figura (d) a escala, (f) vista frontal de la figura (e).

4.2.1 Conversión de píxeles a unidades de longitud real.

Si el procedimiento anterior es completado con éxito, se puede entonces proceder con la conversión de píxeles a coordenadas reales. Para esto, se crea la función `Fn_WorldPoints`, encargada de hacer el cambio de coordenadas por medio de una matriz de transformación, que se genera teniendo en cuenta los parámetros intrínsecos de la cámara, establecidos en el proceso de la calibración, y los extrínsecos, calculados al interior de la función.

Dentro de esta función, se destacan los siguientes aspectos:

- Es aquí donde se cargan los parámetros de calibración *cameraParams.m* mencionados en la sección capítulo 4.1.
- También se carga la imagen-patrón a utilizar. Ésta no es la hoja patrón que recomienda Matlab® en sí (Figura 18), sino una imagen del escenario en el que se está trabajando, el cual contenga tal hoja. En la Figura 24(a) se muestra un ejemplo de una imagen-patrón usada durante el proyecto. El fondo negro corresponde a la cinta de la banda transportadora.
- Se define la longitud de cada lado de un cuadrado del patrón, en milímetros. En este caso, 23 mm.
- La función retorna las coordenadas de los centroides hallados en medidas reales, cuyo origen es el que se muestra marcado con una X roja en la Figura 24b.

Por último, las coordenadas retornadas por `Fn_WorldPoints` son almacenadas durante cada iteración en la variable `centroid`, lista para ser enviada a la trayectoria que debe seguir el robot en la simulación.

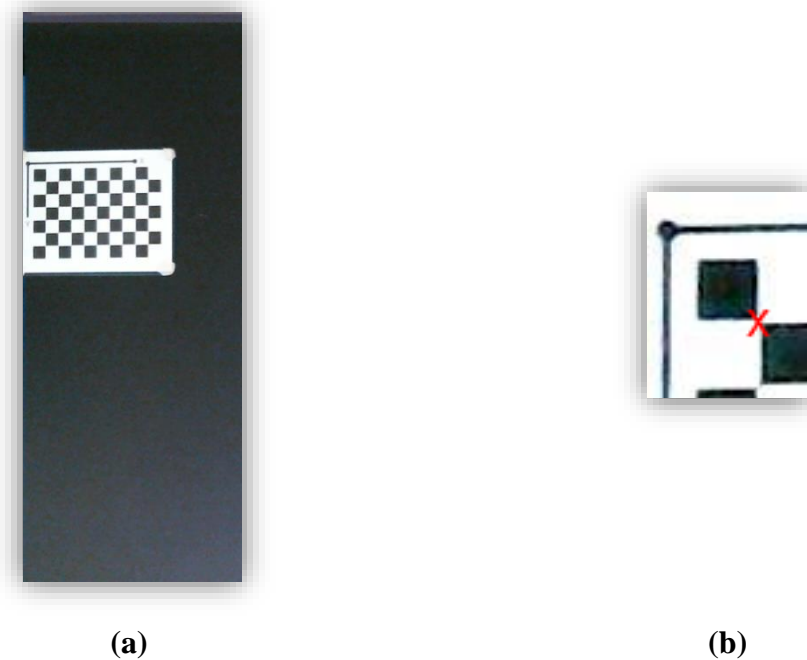


Figura 24: Imagen patrón. (a) Imagen original, (b) zoom del origen.

Es de notar que durante este procedimiento se realizaron todos los pasos básicos que caracterizan las tareas de un sistema de visión artificial: pre-procesamiento, segmentación, extracción de características y clasificación (ver Figura 4, capítulo 1).

4.3 ANÁLISIS CINEMÁTICO DE POSICIÓN INVERSO PARA MANIPULADOR ROBÓTICO SERIAL

En la sección anterior se realizó todo lo concerniente a visión artificial, lo cual es la primera fase de la integración a simular. El paso a seguir corresponde a la utilización de los resultados obtenidos para generar la trayectoria del robot en la simulación, por medio de un análisis de posición inverso.

Es importante recordar que, al realizar un análisis inverso, se obtiene la posición de las articulaciones (de los motores) del robot a partir del punto al que se quiere llegar en el

espacio. Este análisis se realiza en el código principal de este proyecto, llamado `A_MainCode`. Al interior del código, se sigue el mismo procedimiento analítico explicado en el trabajo de [10], [11] y [32], aplicable al manipulador serial *KUKA KR6* que se encuentra en las instalaciones de la Universidad Pontificia Bolivariana. Se decide trabajar con éste pues, entre los robots con que cuenta la Universidad, es el que permite establecer una integración real con mayor prontitud.

Aunque los centroides ya están en medidas reales, su origen no corresponde al del robot, así que se debe sumar la distancia existente entre ambos orígenes; en la Figura 25 se puede apreciar mejor esta distancia, siendo el origen del manipulador O_R y el de la imagen-patrón O_p . Este último varía según la posición que tenga la hoja patrón sobre la banda transportadora.

Posteriormente, estos centroides respecto al sistema coordenado del robot se colocan como parte de la trayectoria que debe seguir el mismo, en la variable `Ptool_F_Matriz`.

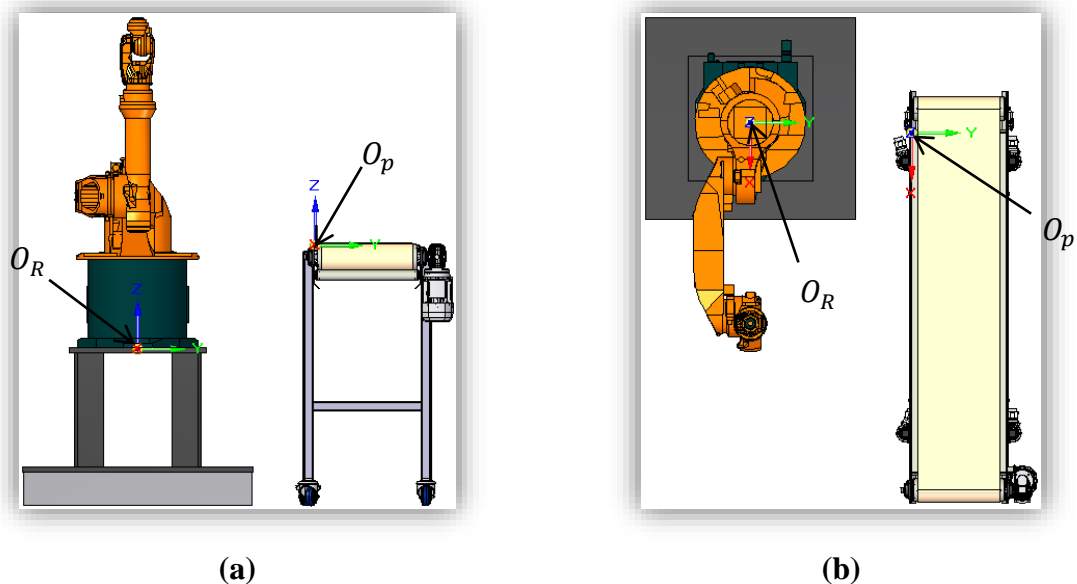


Figura 25: Origen de la banda transportador y el manipulador robótica. (a) Vista frontal, (b) vista superior. Realización de los autores a partir de *Solid Edge*®.

Para proceder con el análisis inverso, se utiliza la función `Fn_An1Inv`, que, además de la trayectoria a seguir, necesita otros parámetros (e.g. la posición de la herramienta con respecto al efector final) cuyos valores no es necesario modificar, así que no se entrará en mayor detalle

Una vez seleccionada la solución deseada, se almacena junto con un vector de tiempo en 6 variables, llamadas `DatosRev` (e.g. `DatosRev1`, `DatosRev2`, etc), cada una correspondiente a la posición que debe tomar las seis articulaciones del manipulador en un tiempo determinado. Luego, se exportan los `DatosRev` en 6 archivos `*.csv`. Para esto, se utiliza la función `csvwrite`, de *Matlab*®.

4.4 SIMULACIÓN EN *SOLID EDGE*®.

Solid Edge® es un programa CAD (*Computer-Aided Design*: Diseño asistido por computador) desarrollado por *Siemens PLM Software*, que permite realizar las actividades involucradas en un proceso de desarrollo de productos [34]. Entre éstas, se encuentra el modelado 3D y simulaciones a partir de los mismos, útiles para este proyecto. Por otro lado, la empresa *KUKA Roboter GmbH*, ofrece en su página web el modelo CAD del manipulador *KR 6* [35], también utilizado durante el trabajo. El modelo del robot es el mostrado en la Figura 26.

Como el objetivo es realizar la simulación de la trayectoria que debe seguir el robot, a partir de los datos obtenidos por un sistema de visión artificial y un objeto sobre una banda transportadora en movimiento, se adiciona el modelo de una banda y la cámara elegida. El modelo de la banda es realizado por [36], y de la cámara es realizado por [37] (Figura 27a y Figura 27b respectivamente), ambos compartidos en la *web*.

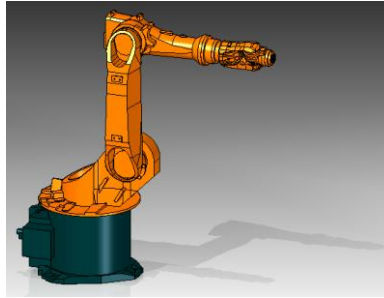
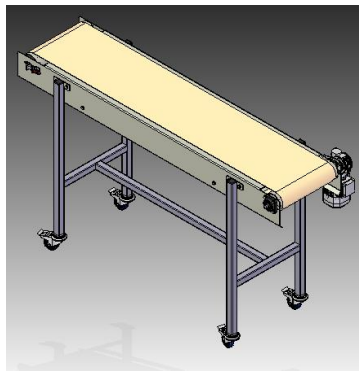
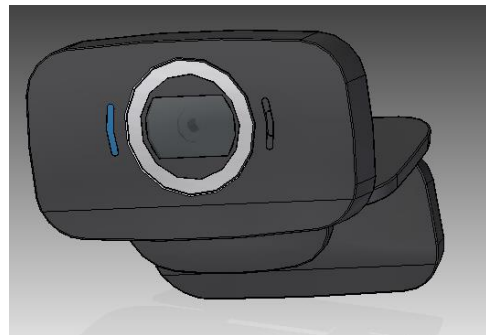


Figura 26: Modelo del manipulador KUKA KR 6 en Solid Edge®.



(a)



(b)

Figura 27: Modelos de la banda transportadora (a) y la cámara web (b). Imagen tomada por los autores desde Solid Edge®.

Para crear un ambiente más cercano al que existe en laboratorio de la Universidad, se crean las bases del robot y la cámara. El entorno de simulación es el mostrado en la Figura 28.

La simulación se debe llevar a cabo en el ambiente *Motion* de Solid Edge®, para poder exportar la trayectoria que debe seguir el robot en la simulación. El modelo del manipulador presenta relaciones de movimiento entre sus barras, que representan el movimiento o posición que deben tener los motores del mismo en la vida real. De acuerdo con lo visto en el la sección 2.1.1, estas relaciones son de tipo R (*revolute* en Solid Edge®). El menú de *Motion* con las relaciones anteriores luce como se muestra en la Figura 29, y es en éstas en donde se importan los resultados procesados a partir de la información generada por el sistema de visión artificial.

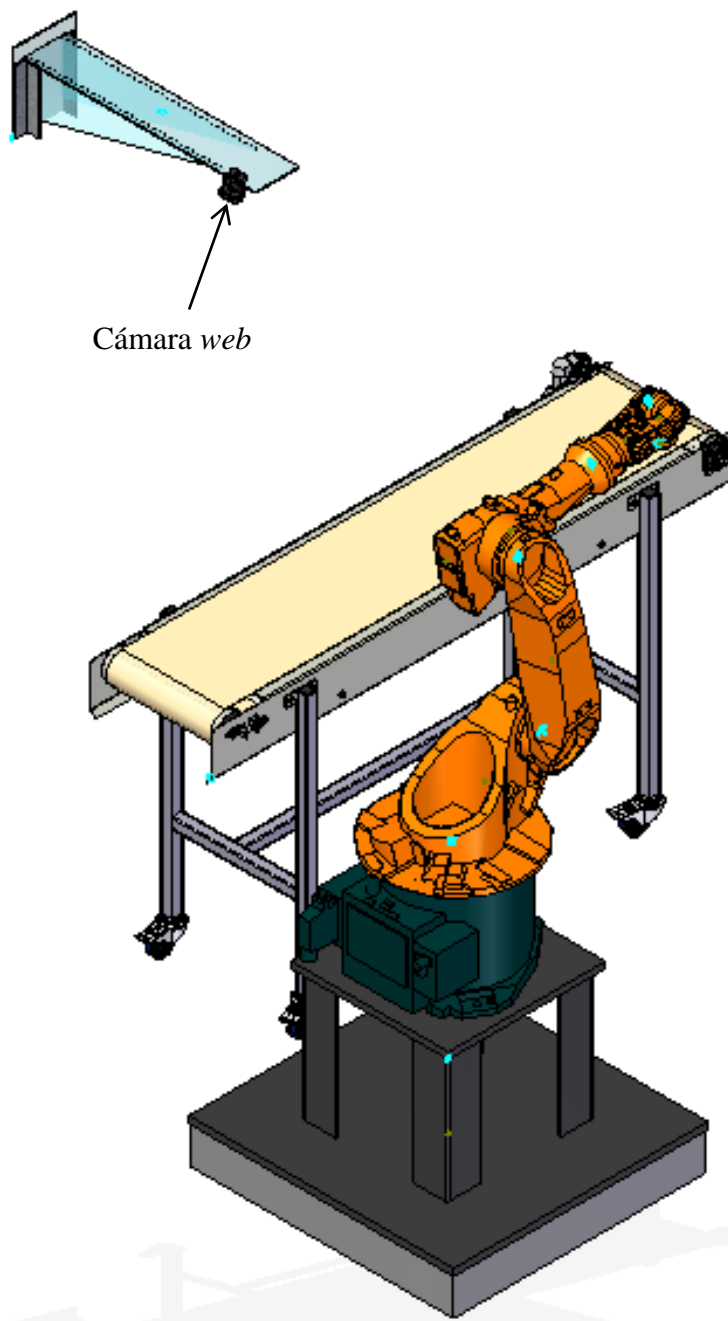


Figura 28: Entorno de simulación con todos los modelos. Imagen tomada por los autores desde *Solid Edge*®.

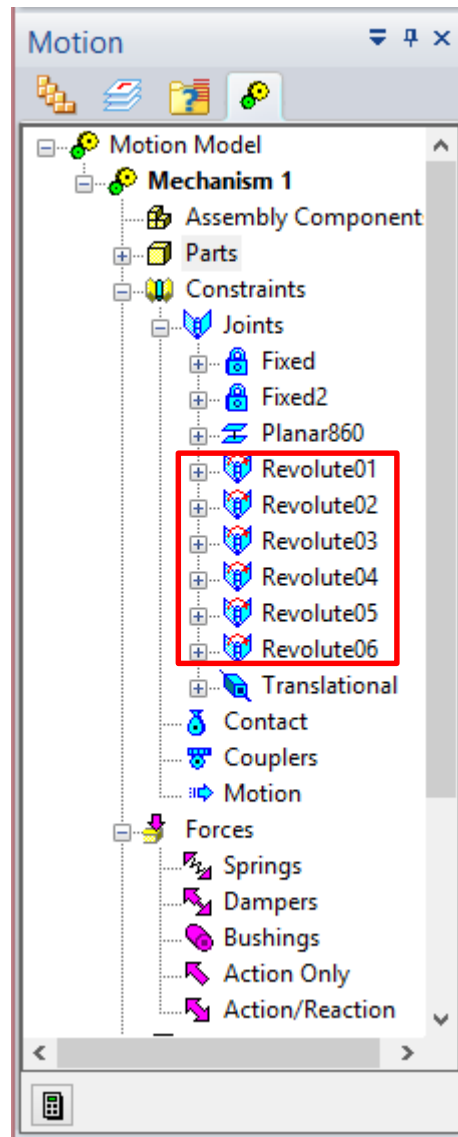


Figura 29: Menú del ambiente *Motion* de *Solid Edge*®. Tomado del *software*.

El procedimiento para importar los resultados a *Solid Edge*® es el siguiente:

1. Ingresar a las propiedades de cada relación de revoluta.
2. En la pestaña *Motion*, elegir:
 - a. Movimiento alrededor del eje Z.
 - b. Tipo de movimiento desplazamiento.
 - c. Función tipo *Spline*.
 - d. Presionar “*Load From File*” (Cargar Desde Archivo).

- e. Del paso anterior se abre una ventana para importar el archivo deseado. Allí se elige el archivo *.csv correspondiente (e.g. *DatosRev1.csv*), generado con anterioridad.
- f. Confirmar.

La Figura 30 muestra un resumen gráfico de lo anterior. Con esto hecho, se puede realizar la simulación final.

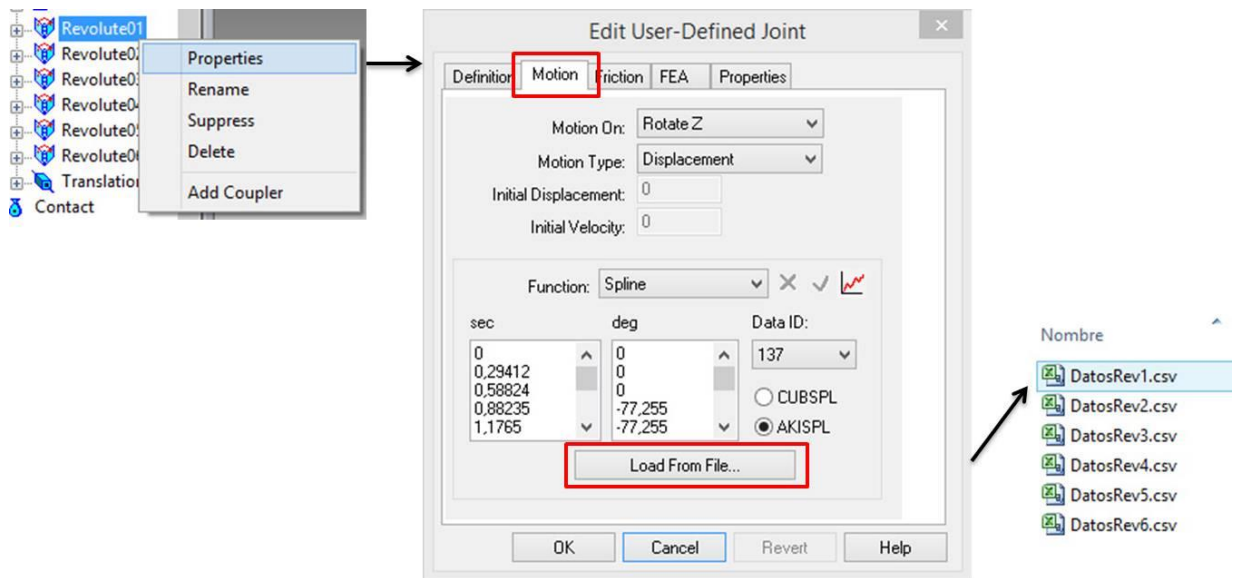


Figura 30: Procedimiento para importar los resultados de visión artificial a Solid Edge®. Realización de los autores.

4.4.1 Resultado de la simulación

A continuación se muestran los resultados obtenidos al realizar lo explicado durante el capítulo. La figura geometría utilizada en este caso, se muestra en la Figura 31.

Design Simulation Technologies, Inc [38], ha desarrollado un complemento para *Motion*, el cual permite observar la trayectoria que sigue el robot en la simulación. En la Figura 32, se

muestra el resultado, además, un video que ilustra mejor lo anterior se encuentra en el siguiente enlace:

<http://youtu.be/JqgvXYJgP9o>

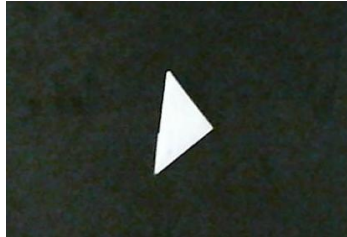
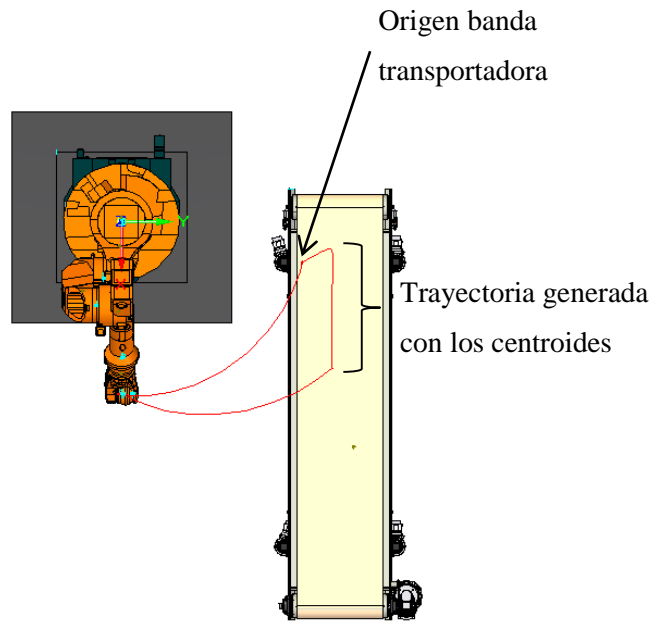


Figura 31: Geometría utilizada. Triángulo rectángulo.

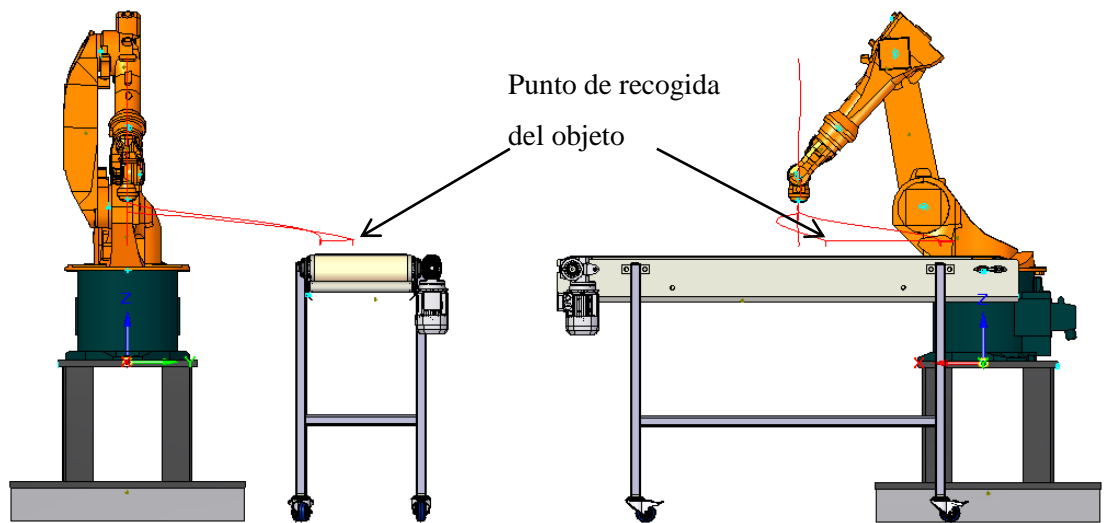
Es importante realizar las siguientes observaciones:

- Cuando el manipulador se dirige a la banda transportadora, sigue una trayectoria que impide que se choque contra el sistema de visión o la banda transportadora.
- Al dirigirse a la banda transportadora, primero se posiciona en el origen de ésta, y luego sigue la trayectoria generada a partir de los centroides.
- Cuando finaliza el recorrido de los centroides, el manipulador baja en Z para recoger la figura, y cuando la toma, se dirige a otra posición para depositar la figura, también siguiendo una trayectoria que impide se choque.

Con esto, finaliza la simulación de la integración de un sistema de visión artificial con un manipulador robótico, seleccionando y posicionando objetos de contornos muy simples que se encuentran sobre una banda transportadora en movimiento.



(a)



(b)

(c)

Figura 32: Trayectoria seguida por el manipulador *KUKA KR 6* a partir de los resultados del sistema de visión artificial. (a) Vista superior, (b) vista frontal, (c) vista lateral.

5. TRABAJO FUTURO

En este capítulo se tiene como objetivo mostrar los posibles pasos a seguir, partiendo del trabajo realizado, para llegar a una integración completa entre el sistema de visión artificial y el manipulador robótico con el que cuenta la Universidad Pontificia Bolivariana (*KUKA KR6*). Para dicha integración será necesario completar algunos pasos fundamentales: Determinación de las condiciones óptimas de interconexión, estudio del lenguaje de programación del robot, creación del algoritmo de integración, ejecución y análisis de resultados y por último las posibles mejoras que puedan optimizar algunos aspectos de la interconexión o del proceso como tal.

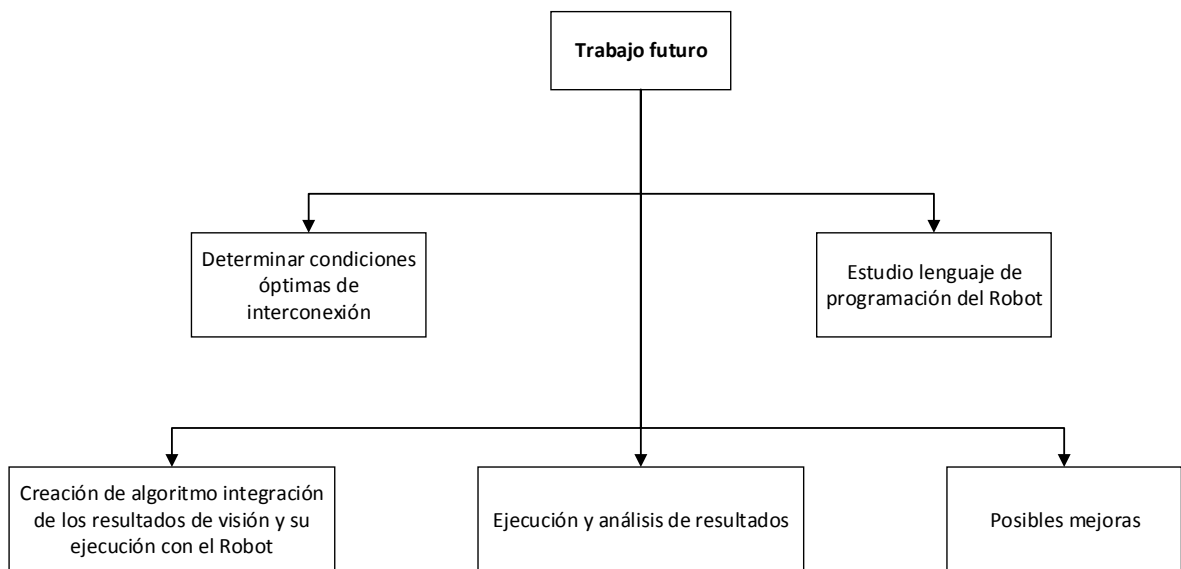


Figura 33: Resumen capítulo 5.

5.1 ESTUDIO DEL LENGUAJE DE PROGRAMACIÓN DEL ROBOT

Un paso fundamental, además de conocer el lenguaje de programación del robot con que se va a trabajar, es saber manejarlo a un nivel avanzado, ya que este será determinante a la hora de generar la interconexión entre los resultados obtenidos del proceso de visión artificial y el robot. El robot *KUKA* con el que cuenta la universidad tiene como lenguaje de programación el *KUKA SYSTEM SOFTWARE (KSS)*.

Será entonces necesario abordar dicho lenguaje desde varios aspectos. Inicialmente se deberá conocer el modo como se leen o se escriben los diferentes *scripts*, se deberá también conocer en qué formato y en dónde deben ser guardados. Por otro lado se deberá estudiar con detenimiento la estructura característica del lenguaje y los parámetros que se definen automáticamente, los cuales son parte esencial de la comunicación: lenguaje de programación – robot. Por último se deberán estudiar las funciones principales que contiene dicho lenguaje y que posiblemente puedan ser de gran ayuda para agilizar el proceso de la generación del algoritmo final. En este punto se deberá dedicar una gran parte del tiempo a la investigación o entendimiento de las funciones relacionadas con los tipos de comunicación, pues estos serán determinantes en la integración mencionada anteriormente (e.g. *CWRITE*, *CREAD*). En la Figura 34 se muestra una parte de la interfaz máquina-usuario que tiene el manipulador robótico y en la cual se deberán editar y ejecutar los diferentes programas que deberán ser realizados.

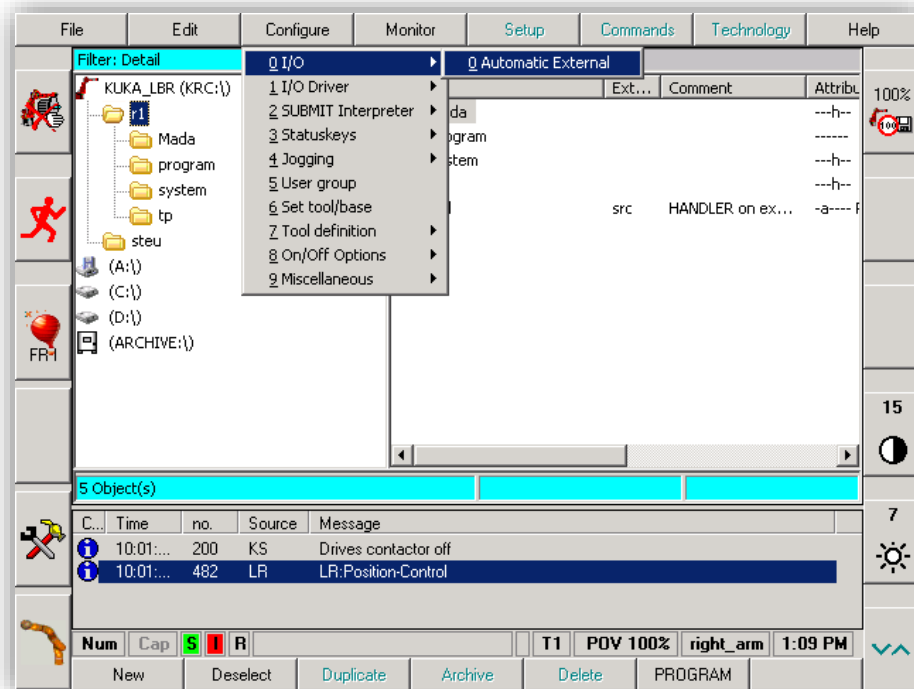


Figura 34. Interfaz software KRC. Tomado de [39].

Aunque no es necesario, es importante tener en cuenta que en un futuro se podrían realizar dichas comunicaciones por medio de otros tipos que permiten una mayor velocidad y facilidad de conexión. Entre estos podemos destacar la comunicación vía Ethernet, la cual podría ser una alternativa óptima que facilitaría la comunicación y por ende la velocidad de ejecución del programa.

5.2 CREACIÓN DEL ALGORITMO DE INTEGRACIÓN

En este paso se debe tomar lo realizado en los dos pasos anteriores, con el fin de generar el algoritmo de integración. Es importante decir que para esto se deben realizar dos códigos relacionados: uno en el *software* encargado del sistema de visión artificial y otro en el *software* del robot, estos se complementan ya que deben estar en la capacidad de compartir información en cualquier momento, siendo esto uno de los puntos cruciales para la obtención de resultados precisos. Lo anterior lleva entonces a pensar en el cómo deben

realizarse dichos códigos. Para esto se han considerado los siguientes aspectos que podrían ser punto de partida teniendo en cuenta las especificaciones ya mencionadas (se presentan a continuación los aspectos fundamentales, únicamente para la parte de la comunicación).

- Código para el *software* encargado del sistema de visión artificial.
- Definición de paquetes de datos que se deben enviar al manipulador.
- Sección que abra el puerto de comunicación.
- Sección que permita la comunicación serial, pudiendo enviar un paquete de datos si ya se terminó de adquirir.
- Sección que permita determinar si los datos están llegando de manera adecuada al manipulador y proteja al sistema de posibles errores en la información.
- Sección final que cierre el puerto de comunicación.

Aunque estos no son estrictamente los pasos a seguir, si son unos aspectos importantes que pueden conducir hacia el objetivo establecido, por ello deben ser tomados como un sugerencia para el procedimiento futuro y está abierto a cambios, dependiendo de lo que se vaya descubriendo en el transcurso del trabajo.

5.3 EJECUCIÓN Y ANÁLISIS DE RESULTADOS

Uno de los puntos fundamentales que se deben mirar en el trabajo futuro es implementar los conceptos expuestos, de modo que el robot sea capaz de sujetar el objeto que se encuentra en movimiento y pueda depositarlo en el lugar ya establecido a partir de la información suministrada por el reconocimiento de imágenes. A continuación se dan algunas ideas de las posibles pruebas que se pueden realizar para la generación de análisis.

- Modificar las condiciones del entorno de trabajo. Algunos factores que se podrían modificar y estudiar son:
 - ✓ Cambios en el método de detección de figuras.
 - ✓ Cambios en la iluminación del espacio de trabajo.
 - ✓ Cambios en el color de las figuras de trabajo.

- El trabajo hasta ahora se ha centrado en el cumplimiento de una tarea sin tener en cuenta consideraciones de tiempo. Este es un aspecto importante que debe estudiarse y que incluye la velocidad de procesamiento y transmisión de datos, así como la respuesta mecánica del robot y la máxima velocidad a la que puede operar la banda considerando los tiempos de procesamiento de datos y de respuesta del robot.

Además de estas pruebas se pueden realizar otras que se podrán ir generando a medida que se vaya trabajando y se definan nuevos retos para el trabajo.

CONCLUSIONES

La simulación de la integración de un sistema de visión artificial con un manipulador robótico serial, para tratar objetos geométricos sobre una banda transportadora en movimiento fue realizada en este trabajo. Allí, se logró detección de los objetos y sus propiedades por medio de una cámara *web* y técnicas comunes de procesamiento de imágenes, dando como resultado satisfactorio la ubicación del centroide de los mismos respecto al origen del manipulador en coordenadas reales (i.g. mm). Estos centroides permiten saber la trayectoria que debe realizar el manipulador en la simulación para seguir el objeto a lo largo de la banda, tomarlo y posicionarlo en un lugar diferente previamente establecido. Lo anterior se realiza con un análisis inverso de posición para manipuladores seriales, del cual se genera la información necesaria para realizar la simulación en un programa *CAD*.

El sistema de visión generado se encuentra en la capacidad de detectar cualquier tipo de figura sin importar su escala su posición u orientación, siempre y cuando se conozca el primer momento invariante de la misma. Un reproductor de video permite observar el centroide encontrado, verificando que éste fue hallado con éxito.

Este proyecto es una etapa inicial en cuanto al uso de la visión artificial en conjunto con robots industriales, y servirá de base para futuros desarrollos.

La interacción de los manipuladores robóticos y la visión artificial abre posibilidades muy interesantes para la automatización de diversos procesos industriales.

RECOMENDACIONES

La realización de este trabajo es el resultado de lo aprendido sobre procesamiento de imágenes y manipuladores robóticos. Aunque se tomaron decisiones para poder realizar esto, nunca se comprobó que fueran las más eficientes, lo cual deja un buen campo de investigación en el tema y posibles mejoras.

Si se quiere agilizar el proceso de visión artificial, se recomienda obviar el reproductor de video que muestra el centroide, que es netamente una verificación de que el centroide fue hallado exitosamente. También se recomienda explorar otros lenguajes de programación para procesamiento de imágenes, como lo puede ser el *OpenCV (Open Source Computer Vision)* que podrían también agilizar este procedimiento.

El trabajo cuenta con una parte real (visión artificial) y otra simulada (la trayectoria sigue el robot). Sería positivo que a futuro se realice la integración con un manipulador real y acercarse más a tareas útiles a nivel industrial. El capítulo 5 deja un lineamiento básico de cómo se podría realizar esto.

BIBLIOGRAFÍA

- [1] P. Joshi, “Understanding Camera Calibration,” 2014. [Online]. Available: <http://prateekvjoshi.com/2014/05/31/understanding-camera-calibration/>. [Accessed: 14-Oct-2014].
- [2] ISO, “ISO 8373:2012,” 2012. [Online]. Available: <https://www.iso.org/obp/ui/#iso:std:iso:8373:ed-2:v1:en>.
- [3] Subir Kumar Saha, *Introducción a la Robótica*, Primera. Mexico: Mc Graw Hill, 2010.
- [4] R. Shannon and J. D. Johannes, “Systems Simulation: The Art and Science,” *IEEE Trans. Syst. Man. Cybern.*, vol. 6, no. 10, pp. 723–724, 1976.
- [5] *Aplicaciones de la visión artificial y la biometría informática*. Librería-Editorial Dykinson, 2005.
- [6] C. Pérez Sarmiento, “Binarizacion de Imagenes ~ site - informatica.” .
- [7] J. Á. Fernández Muñoz, “Estudio Comparativo de las Técnicas de Procesamiento Digital de Imágenes,” Universidad Politecnica de Madrid, 1999.
- [8] O. Marques, *Practical Image and Video Processing Using MATLAB®*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2011.
- [9] A. Grossman, “Forecast Blog: Cleaning Radar Images using Neural Nets & Computer Vision.” [Online]. Available: <http://blog.forecast.io/cleaning-radar-images-using-neural-nets-computer-vision/>. [Accessed: 31-Mar-2014].
- [10] L.-W. Tsai, *Robot analysis: the mechanics of serial and parallel manipulators*. John Wiley & Sons, Inc., 1999.
- [11] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, Third. EEUU: Pearson Prentice Hall, 2005.
- [12] C. Bernier, “Challenges of Dexterous End Effectors,” 2013. [Online]. Available: <http://blog.robotiq.com/bid/66051/Challenges-of-Dexterous-End-Effectors>. [Accessed: 23-Feb-2014].
- [13] D. Roza Osorio, “Anteproyecto: Desarrollo de una aplicación para el análisis cinemático de un prototipo de un manipulador serial de seis grados de libertad,” Universidad Pontificia Bolivariana, 2013.

- [14] Design of Intelligent Machines Laboratory, “High-Speed Parallel Robot HEXA.” [Online]. Available: <http://www.space.mech.tohoku.ac.jp/research/hexa/hexa-e.html>. [Accessed: 23-Feb-2014].
- [15] Modular robotic & Robot locomotion Group, “Parallel Robot Design,” 2004. [Online]. Available: <http://155.69.254.10/users/risc/www/para-intro.html>. [Accessed: 23-Feb-2014].
- [16] KUKA Roboter GmbH, “KUKA Robot Training and Education.” [Online]. Available: [http://www.kuka-robotics.com/australia/en/college/cs/csSeminarDetail/?SEM_ID=5F\\$\\$\\$7W1\\$\\$\\$CS\\$---13](http://www.kuka-robotics.com/australia/en/college/cs/csSeminarDetail/?SEM_ID=5F$$$7W1$$$CS$---13). [Accessed: 03-Mar-2014].
- [17] D. Li, G. Wen, B. Wei Hui, S. Qiu, and W. Wang, “Cross-ratio invariant based line scan camera geometric calibration with static linear data,” *Opt. Lasers Eng.*, vol. 62, pp. 119–125, Nov. 2014.
- [18] “Photogrammetry.” [Online]. Available: <http://www.photogrammetry.com/>. [Accessed: 14-Oct-2014].
- [19] “Stereo Vision with MATLAB and Simulink.” [Online]. Available: <http://www.mathworks.com/discovery/stereo-vision.html?refresh=true>. [Accessed: 14-Oct-2014].
- [20] Jmeyer, “What is focal length: definition, comparison, every question answered | Digital Camera World,” 2012. [Online]. Available: <http://www.digitalcameraworld.com/2012/09/07/what-is-focal-length-definition-comparison-every-question-answered/>. [Accessed: 14-Oct-2014].
- [21] A. Badali and Y. Zhang, “Scale factor in digital cameras,” *Photonics ...*, p. 10, 2005.
- [22] M. D. E. Proyección, “Técnicas de calibrado de cámaras.”
- [23] “Lens distortion: everything every photographer must know | Digital Camera World.” [Online]. Available: <http://www.digitalcameraworld.com/2012/10/12/lens-distortion-everything-every-photographer-must-know/>. [Accessed: 14-Oct-2014].
- [24] Z. Zhang, “Camera calibration,” in *Emerging topics in computer vision*, 1st ed., Prentice Hall, 2005, pp. 5–43.
- [25] J. Sun, X. Chen, Z. Gong, Z. Liu, and Y. Zhao, “Accurate camera calibration with distortion models using sphere images,” *Opt. Laser Technol.*, vol. 65, pp. 83–87, Jan. 2015.

- [26] D. Maljovec, "Canny's edge detection scheme," 2010. [Online]. Available: <http://www.cs.utah.edu/~maljovec/CS6640/project4/images/checker32-seg.gif>. [Accessed: 14-Oct-2014].
- [27] G. Medioni and S. Kang, *Emerging topics in computer vision*, 1st ed. Prentice Hall, 2004.
- [28] I. The MathWorks, "Matlab." 2014.
- [29] "HD Webcam C615." [Online]. Available: https://support.logitech.com/en_ca/product/8041#download. [Accessed: 28-Oct-2014].
- [30] Logitech and Nextway S.A., "Logitech HD Webcam C615," 2011. [Online]. Available: <http://www.nextway.ch/webcams/logitech-hd-webcam-c615-p-16374.html>. [Accessed: 28-Oct-2014].
- [31] MATLABAcademy, "The Seven Invariant Moments," 2011. [Online]. Available: [http://www.mathworks.com/matlabcentral/fileexchange/33975-the-seven-invariant-moments/content/Invariant Moments/feature_vec.m](http://www.mathworks.com/matlabcentral/fileexchange/33975-the-seven-invariant-moments/content/Invariant%20Moments/feature_vec.m). [Accessed: 28-Oct-2014].
- [32] D. Roza Osorio, "Desarrollo de una aplicación para el análisis cinemático de un prototipo de un manipulador serial de seis grados de libertad," Universidad Pontificia Bolivariana, 2013.
- [33] J. César, "Curso: ANÁLISIS DE MANIPULADORES ROBÓTICOS." 2014.
- [34] Siemens Product Lifecycle Management Software Inc, "Solid Edge: Siemens PLM Software." 2014.
- [35] KUKA Robotics Corp, "KUKA KR 6-2," 2014. [Online]. Available: [http://www.kuka-robotics.com/usa/en/downloads/search/?sc_META_01=Product&type=current&rs_META_10=KR 6-2](http://www.kuka-robotics.com/usa/en/downloads/search/?sc_META_01=Product&type=current&rs_META_10=KR%206-2). [Accessed: 03-Nov-2014].
- [36] Honza, "Easy made Belt Conveyor," *GRABCAD*, 2014. [Online]. Available: <https://grabcad.com/library/easy-made-belt-conveyor-1>. [Accessed: 03-Nov-2014].
- [37] Rose, "Logitech HD Webcam C615," *GRABCAD*, 2013. [Online]. Available: <https://grabcad.com/library/logitech-hd-webcam-c615-1>. [Accessed: 03-Nov-2014].
- [38] I. Design Simulation Technologies, "Dynamic Designer Motion," 2014. [Online]. Available: <https://www.design-simulation.com/DDM/SolidEdge/ddm/index.php>. [Accessed: 03-Nov-2014].

- [39] "Information about the KUKA lightweight robot." [Online]. Available: http://toychest.ai.uni-bremen.de/wiki/_media/projects:kuka_lwr:aut-ext-open.png. [Accessed: 04-Nov-2014].
- [40] I. The MathWorks, "Installing the Support Packages for Image Acquisition Toolbox Adaptors - MATLAB & Simulink," *R2014a Documentation*. [Online]. Available: <http://www.mathworks.com/help/imaq/installing-the-support-packages-for-image-acquisition-toolbox-adaptors.html>. [Accessed: 24-Aug-2014].
- [41] Logi Nu, "logitech c615 webcam focal length and sensor dimen... - Page 2 - Logitech Forums," 2014. [Online]. Available: <http://forums.logitech.com/t5/Webcams/logitech-c615-webcam-focal-length-and-sensor-dimension/td-p/1189763/page/2>. [Accessed: 28-Oct-2014].

ANEXOS

SOBRE ADAPTADORES DE CÁMARA WEB EN MATLAB R2014A®.

Un adaptador de cámara en *Matlab*®, es el que permite que el software reconozca con éxito la cámara, como un dispositivo de entrada. En versiones anteriores del *Image Acquisition Toolbox*™ dichos adaptadores estaban incluidos durante la instalación de *Matlab*®. Sin embargo, en la versión R2014a, cada adaptador está disponible separadamente en el Soporte de Instalador de Paquetes (*Support Package Installer*) [40]. Si esto no se realiza, entonces no se puede hacer uso de la cámara *web*, y por lo tanto, nada de lo aquí presentado.

Para acceder al soporte, se puede proceder de dos formas:

- En el *Command Window* de *Matlab*®, escribir: `supportPackageInstaller`
- Click en Home > Resources > Add-Ons > Get Hardware Support Packages.

Allí, es importante que el paquete *OS Generic Video Interface* esté instalado (si se está usando *Microsoft Windows*™ como sistema operativo), para que reconozca la cámara *web* del computador u otras cámaras *web* con conexión USB; fuera de éste, también se necesita instalar el *USB Webcams Package*. Con este par de paquetes es suficiente para poder realizar los procedimientos de procesamiento de imágenes y video desde *Matlab*®.