



UNIVERSIDAD AUTÓNOMA DE SAN LUIS POTOSÍ

Facultad de Ingeniería

Centro de Investigación y Estudios de Posgrado

**“UN ALGORITMO BASADO EN VISIÓN PARA LA
DETECCIÓN Y MANIPULACIÓN DE PIEZAS EN
UNA CELDA DE MANUFACTURA”**

TESIS

QUE PARA OBTENER EL GRADO DE

MAESTRO EN INGENIERÍA MECÁNICA

Presenta:

ING. GABRIEL FERNANDO GARCÍA CEDILLO

Asesor:

DR. JUAN ANTONIO CÁRDENAS GALINDO



UNIVERSIDAD AUTÓNOMA
DE SAN LUIS POTOSÍ

21 de octubre de 2010

**ING. GABRIEL FERNANDO GARCÍA CEDILLO
P R E S E N T E. –**

En atención a su solicitud de Temario, presentada por el **Dr. Juan Antonio Cárdenas Galindo** Asesor de la Tesis que desarrollará Usted, con el objeto de obtener el Grado de **Maestría en Ingeniería Mecánica**. Me es grato comunicarle que en la Sesión de Consejo Técnico Consultivo celebrada el día 21 de octubre del presente año, fue aprobado el Temario propuesto:

TEMARIO:

**“UN ALGORITMO BASADO EN VISIÓN PARA LA DETECCIÓN Y
MANIPULACIÓN DE PIEZAS EN UNA CELDA DE MANUFACTURA”**

INTRODUCCIÓN.

1. SISTEMA DE VISIÓN POR COMPUTADORA.
2. MÉTODO DE CONTROL BASADO EN VISIÓN PARA ROBOTS.
3. DETECCIÓN AUTOMÁTICA DE PIEZAS EN UNA BANDA TRANSPORTADORA POR VISIÓN COMPUTACIONAL.
4. IMPLEMENTACIÓN Y PRUEBAS EXPERIMENTALES.

CONCLUSIONES Y RESULTADOS.

BIBLIOGRAFÍA.

APÉNDICE.

“MODOS ET CUNCTARUM RERUM MENSURAS AUDEBO”

A T E N T A M E N T E


**ING. ARMANDO VIRAMONTES ALDANA
DIRECTOR**


UNIVERSIDAD AUTÓNOMA
DE SAN LUIS POTOSÍ
FACULTAD DE INGENIERÍA
DIRECCIÓN

*bcb



**FACULTAD
DE INGENIERÍA**

Av. Manuel Nava 8
Zona Universitaria • CP 78290
San Luis Potosí, S.L.P.
tel. (444) 826 2330 al39
fax (444) 826 2336
www.uaslp.mx

Resumen

En la actualidad, se ha hecho evidente la idea de contar con celdas de manufactura inteligentes que utilicen robots industriales, los cuales cuentan con una serie de sensores para garantizar la ubicación, posición y dimensión de las piezas de trabajo con referencia al robot.

En la mayoría de los casos los sensores más comúnmente usados no facilitan la realización de una tarea diferente, ya que un cambio de producto en la línea de producción requerirá de una reubicación de los sensores utilizados; evidentemente esto elevaría los costos y reduciría la flexibilidad de la celda. Un punto crítico para que los robots realicen este tipo de tareas depende mucho de los algoritmos con los que se hayan programado a los robots, una alternativa viable a este tipo de situaciones son los algoritmos basados en visión. Los sistemas de visión en robots minimizan el tiempo requerido para cambiar una línea de producción con diferente producto en tiempo real. “Con los sistemas de visión, sólo tendría que ejecutarse un programa diferente al robot, sin tener que modificar las instalaciones de la línea” [7].

La motivación para la realización del presente trabajo de tesis, es la implementación de un sistema que no requiera de la ayuda, ni de la calibración por parte del usuario para su operación, sino que sea totalmente autónomo y flexible tanto en el funcionamiento

como en la detección de la pieza. De tal forma que el sistema de visión sea capaz de detectar superficies planas de piezas con geometría regular con diferentes alturas y realizar tareas de manipulación tales como carga o descarga de una estiba.

Para el desarrollo de este trabajo se utilizó un robot industrial de la marca MOTO-MAN modelo HP3 de 6 grados de libertad con capacidad de cargar objetos de hasta 3 kg para ejecutar tareas de estiba de piezas, de tal forma que con este sistema el robot identifique y estibe de manera automática la pieza mediante sistemas de visión. Se busca desarrollar este sistema con un enfoque innovador y flexible como una propuesta alternativa al proceso rígido tradicional de “enseñanza-aprendizaje” de programación de robots con la validación y como plataforma de pruebas para un nuevo modelo de visión basado en la Metodología de Manipulación en Espacio de Cámara utilizando un Modelo Lineal de Cámara .

Dedicatoria

A mis padres Raquel y Gabriel.

A Lourdes.

Agradecimientos

A mis padres Raquel y Gabriel por su confianza y apoyo incondicional que me han dado a lo largo de mi vida.

A Lourdes por su cariño y comprensión.

A mi hermano Humberto por su apoyo.

A mi asesor de tesis Dr. Antonio por sus consejos y apoyo en la realización de este trabajo.

A mi coasesor Dr. Juan Manuel por sus valiosas observaciones.

A todos los profesores que participaron en la formación de mis estudios de posgrado.

A mis amigos y compañeros de generación Juan Carlos y Jorge, por su amistad.

Al CONACYT por otorgarme la beca para mis estudios.

Índice general

| | |
|--|-----------|
| Introducción | 1 |
| Motivación | 1 |
| Estado del Arte | 3 |
| Organización de la tesis | 5 |
| 1. Sistemas de visión por computadora | 6 |
| 1.1. Adquisición de imágenes | 7 |
| 1.2. Análisis de imágenes | 9 |
| 1.3. Procesamiento de imágenes | 12 |
| 1.3.1. Operaciones individuales | 13 |
| 1.4. Extracción de bordes | 17 |
| 1.4.1. Filtros morfológicos | 19 |
| 1.5. Descripción de figuras geométricas en imágenes | 21 |
| 1.6. Modelos de cámara | 23 |
| 1.6.1. Modelo de cámara de orificio | 23 |
| 1.6.2. Modelo de cámara ortográfico | 24 |
| 2. Método de Control basado en Visión para Robots | 27 |
| 2.1. Robots industriales | 27 |
| 2.1.1. Robots manipuladores | 28 |

| | | |
|-----------|---|-----------|
| 2.1.2. | Modelo cinemático directo de un robot manipulador | 29 |
| 2.1.3. | Modelo cinemático del robot Motoman HP3 | 30 |
| 2.2. | Método de Manipulación en Espacio de Cámara | 31 |
| 2.2.1. | Parámetros de visión | 35 |
| 2.2.2. | Flattening | 39 |
| 2.3. | Método de manipulación de espacio de cámara con modelo lineal de cámara (LCM-CSM) | 40 |
| 2.3.1. | Matriz de proyección | 40 |
| 2.4. | Determinación de la configuración interna del manipulador | 45 |
| 2.4.1. | Posicionamiento del robot con modelo de cámara ortográfico | 46 |
| 2.4.2. | Posicionamiento del robot con modelo de cámara lineal | 48 |
| 2.4.3. | Orientación | 50 |
| 3. | Detección automática de piezas en una banda transportadora por visión computacional | 53 |
| 3.1. | Implementación de técnicas de análisis y procesamiento de imágenes para visión computacional | 53 |
| 3.1.1. | Obtención de coordenadas de marcas visuales | 55 |
| 3.2. | Metodología para determinar posición y orientación de las piezas de trabajo | 58 |
| 4. | Implementación y pruebas experimentales | 67 |
| 4.1. | Hardware de las plataformas de pruebas | 68 |
| 4.1.1. | Celda de Manufactura Integrada por Computadora (CIM) | 68 |
| 4.1.2. | Controlador Lógico Programable (PLC) | 69 |
| 4.1.3. | Robot manipulador | 71 |
| 4.1.4. | Marcas visuales | 72 |

| | |
|--|------------|
| 4.1.5. Cámaras digitales | 73 |
| 4.1.6. Unidad Pan-Tilt (PTU) | 74 |
| 4.1.7. Unidad de láser estructurado | 75 |
| 4.1.8. Computadora de control | 76 |
| 4.2. Software | 77 |
| 4.2.1. Software de control de señales de entrada/salida | 77 |
| 4.2.2. Software de procesamiento de información | 78 |
| 4.3. Implementación | 79 |
| 4.3.1. Detección y estiba de envases de producto | 79 |
| 4.3.2. Detección y estiba de piezas en una celda de manufactura | 81 |
| Conclusiones y resultados | 86 |
| Resultados de la implementación de detección y estiba de envases de producto | 87 |
| Resultados de la implementación de detección y estiba de piezas en una celda de manufactura | 90 |
| Conclusiones | 96 |
| Trabajo a futuro | 100 |
| Bibliografía | 102 |
| Apéndices | 107 |
| Descripción de funciones de los algoritmos programados | 107 |
| A1.1. Funciones de OpenCV | 107 |
| A1.1.1. Detección de líneas | 110 |
| A1.2. Funciones realizadas para las implementaciones | 110 |
| A1.2.1. robot.h | 111 |
| A1.2.2. camara.h | 112 |

| | |
|---|------------|
| A1.2.3.pantilt.h | 113 |
| A1.2.4.loc_centroides_imagen.h | 114 |
| A1.2.5.encuentra_puntos_pieza.h | 115 |
| A1.2.6.calcula_config_robot.h | 116 |
| Interfaz Gráfica de Usuario de detección de líneas | 118 |
| Planos de diseño del soporte de marcas visuales | 120 |

Índice de figuras

| | |
|---|----|
| 1. Robots industriales | 2 |
| 1.1. Adquisición de una imagen digital | 7 |
| 1.2. Convención de ejes utilizada para la representación de imágenes | 9 |
| 1.3. Etapas del proceso de visión artificial | 10 |
| 1.4. Funciones de punto y vecindad | 13 |
| 1.5. Operación individual para una imagen | 14 |
| 1.6. Transformación de una imagen por medio del operador umbral | 15 |
| 1.7. Representación simbólica de la transformación de dos imágenes en una | 16 |
| 1.8. Diagrama del concepto de primera y segunda derivada para la extracción de bordes | 18 |
| 1.9. Imagen obtenida con el procesamiento del algoritmo de Canny | 20 |
| 1.10. Descripción de líneas en una imagen por el método de la transformada de Hough | 22 |
| 1.11. Geometría de la cámara de orificio | 23 |
| 1.12. Proyección de perspectiva por proyección ortográfica | 25 |
| 2.1. Robot manipulador Motoman HP3 | 31 |
| 2.2. Sistema de referencia unido a la cámara | 33 |
| 2.3. Diagrama de flujo del cálculo de los parámetros de visión | 38 |
| 2.4. Muñeca del robot con configuración esférica | 46 |
| 2.5. Diagrama de flujo del cálculo del posicionamiento del robot | 52 |

| | |
|---|----|
| 3.1. Proceso de sustracción de imágenes | 56 |
| 3.2. Resultado de la aplicación de los filtros de canny y dilatación | 57 |
| 3.3. Resultado del algoritmo de localización del centroide de una marca visual del robot | 57 |
| 3.4. Captura de imágenes durante el desplazamiento de la proyección de la línea de luz láser sobre la pieza | 59 |
| 3.5. Procesamiento de imágenes mediante el algoritmo de Canny | 60 |
| 3.6. Resultado del procesamiento de imágenes mediante el algoritmo de detección de líneas | 61 |
| 3.7. Procesamiento de imagen. Adquisición de línea láser y coordenada de los puntos de inicio y fin de línea. | 62 |
| 3.8. Resultado de la aplicación de la función de OpenCV <code>cvMinAreaRect2</code> | 63 |
| 3.9. Estimación del plano que describe la posición y orientación de la superficie de la pieza de trabajo visto en el plano XZ. | 64 |
| 3.10. Estimación del plano que describe la posición y orientación de la superficie de la pieza de trabajo visto en los tres ejes XYZ. | 65 |
| 3.11. Diagrama de flujo del algoritmo de detección de superficies planas en piezas | 66 |
| 4.1. Banda transportadora de la Celda Inteligente de Manufactura (CIM) | 70 |
| 4.2. Diagrama de conexión del PLC de control de la estación de paro del robot. | 71 |
| 4.3. Sistema del robot manipulador | 72 |
| 4.4. Diseño de la estructura de las marcas visuales | 73 |
| 4.5. Cámara digital | 74 |
| 4.6. Unidad Pan-Tilt | 75 |
| 4.7. Unidad de láser estructurado con cabezal de línea | 75 |

| | |
|--|-----|
| 4.8. Diagrama de conexión de señales de control entre la PC y el equipo de las implementaciones | 76 |
| 4.9. Implementación de detección y estiba de envases de producto en el laboratorio de Mecatrónica | 80 |
| 4.10. Implementación de detección y estiba de piezas en CIM | 82 |
| 1. Estudio de error de posicionamiento del robot realizada en el patrón cuadrulado. | 88 |
| 2. Análisis de error de posicionamiento del robot para 20 muestras . . . | 90 |
| 3. Análisis de error de posicionamiento del robot para 100 muestras . . . | 91 |
| 4. Adquisición de envase inadecuada derivada del error de posicionamiento. | 91 |
| 5. Etapas de la implementación de detección y estiba de envases de producto. | 92 |
| 6. Error de estimación de la ubicación del centro de la superficie de la pieza. | 94 |
| 7. Error de estimación de la orientación de la superficie de la pieza. . . . | 95 |
| 8. Proceso de convergencia de las juntas del robot mediante el algoritmo de posicionamiento y orientación. | 96 |
| 9. Etapas de la implementación de detección y estiba de piezas en la banda transportadora de la CIM. | 97 |
| A2.1. Interfaz Gráfica de Usuario de detección de líneas. | 119 |
| A3.1. Ensamble del soporte de marcas visuales. | 121 |

Introducción

Motivación

El término Robot evoca imágenes de películas del género de la ciencia ficción, los cuales son capaces de realizar las mismas actividades cotidianas que realiza una persona, pero con mayor destreza y precisión. Es cierto que existe un gran avance en el desarrollo del campo de la robótica, es decir, existen robots que son usados en accidentes nucleares, exploración de volcanes y viajes espaciales, pero aun así, no son capaces de realizar completamente la cantidad de actividades realizadas por una persona. Mas sin embargo los robots se han convertido en instrumentos para realizar labores en situaciones peligrosas o en lugares inalcanzables para los humanos.

La robótica tuvo su comienzo cuando George Devol realizó las primeras patentes para los robots en 1954, y poco después en 1956 fundó la primera compañía robótica de manufactura “Unimation”. Después de dos décadas de esfuerzos para mejorar los diseños de los robots, la robótica ha experimentado una aplicación muy extendida en la industria la cual comienza a fines de la década de 1970. Hoy en día, los robots en la industria son bien conocidos por su alta velocidad, destreza, resistencia y precisión y poder realizar una amplia gama de tareas industriales, tales como el paletizado de material, estampado, soldadura, montaje, pintura, inspección, etc. Debido a la

aplicación de los robots en la automatización de la producción industrial, es que se ha podido elevar la productividad, llevar a cabo infinidad de tareas de riesgo, sustituir la mano de obra de personas y reducir costos de producción, entre otras ventajas.

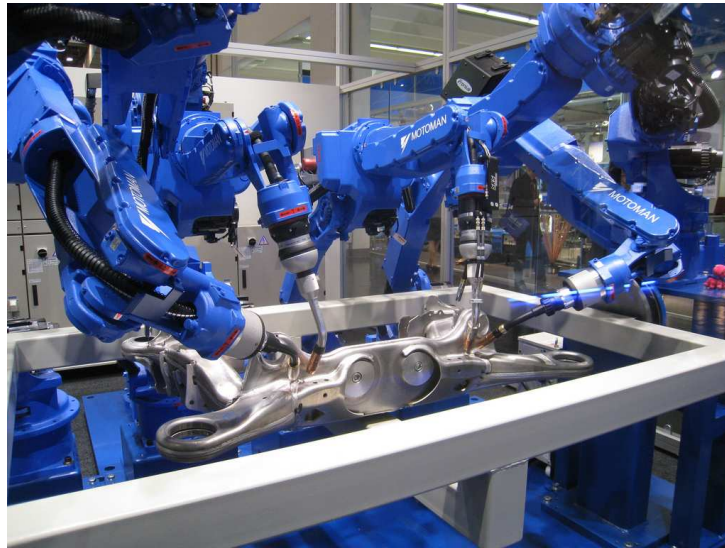


Figura 1: Robots industriales

En la actualidad, se ha hecho evidente la necesidad de contar con celdas de manufactura inteligentes que utilicen robots industriales, los cuales cuentan con una serie de sensores para garantizar la ubicación, posición y dimensión de las piezas de trabajo con referencia al robot, [1], [2].

En la mayoría de los casos los sensores que garantizan la posición de la pieza de trabajo no facilitan la flexibilidad para reprogramar una tarea diferente, ya que un cambio de producto en la línea de producción requerirá de una reubicación de los sensores utilizados, [3], [4]; evidentemente esto eleva los costos y reduce la flexibilidad de la celda. Para que los robots tengan flexibilidad para realizar este tipo de tareas se requiere que los algoritmos con los que se hayan programado los robots sean adap-

tables al cambio del entorno, una alternativa viable para este tipo de situaciones son los algoritmos que hacen uso de sensores de visión. Los sistemas de visión de robots minimizan y flexibilizan el tiempo requerido para cambiar una línea de producción en tiempo real.

La motivación para la realización del presente trabajo, es la implementación de un sistema que no requiera de la ayuda, ni de la calibración por parte del usuario para su operación, sino que sea totalmente autónomo y flexible tanto en el funcionamiento como en la detección de la pieza. Se espera tal forma que el sistema de visión sea capaz de detectar superficies planas de piezas con geometría de poliedro regular con diferentes alturas y realizar tareas de manipulación tales como carga o descarga de una estiba.

Estado del Arte

Durante la revisión bibliográfica se encontró que Mana Saedan y cols.[5] reportan un trabajo de un sistema basado en un modelo de cámara de orificio (“pinhole”) para localizar un objeto (rígido) en el espacio y verificar su posición mediante un robot manipulador industrial de 7 grados de libertad. En este trabajo estimaron la posición del objeto en coordenadas tridimensionales a partir de la relación entre el objeto y las cámaras, obteniendo un valor de error aceptable [5].

M. Peña y cols.[6] reportan un trabajo realizado en el laboratorio de Manufactura Inteligente de CIATEQ, en el que mediante un sistema de visión le permite a un robot manipulador de 6 grados de libertad obtener información de la localización de objetos para tareas de ensamble. El sistema utiliza una sola cámara con la cual se aplican filtros a las imágenes adquiridas, de tal forma que se pueda identificar

la forma de la pieza a localizar. En este trabajo M. Peña y cols. obtuvieron como resultado la localización de las piezas en una línea de producción con la restricción de conocer previamente la altura de los objetos. M. Peña y cols. llevaron a cabo los experimentos utilizando superficies geométricas y dimensionales básicas como cuadrados, círculos, rectángulos, triángulos y polígonos.

Un trabajo parcialmente relacionado con el presente es el de R. Lougheed y R. Sampson[8], en el que mediante un escáner 3D láser, una cámara de video y un robot manipulador se detectaron piezas (ventiladores de aire caliente) de tamaño establecido en un contenedor. El algoritmo utilizado esta basado en visión y además tiene las siguientes características: no requiere de calibración, discretiza las imágenes para localizar el segmento de la pieza que va a ser tomada, la pieza a manipular la adquiere en cualquier posición en un instante dado.

En el Laboratorio de Robótica (LR) de la Facultad de Ingeniería de la Universidad Autónoma de San Luis Potosí se han implementado técnicas de control basadas en visión artificial aplicadas a robots industriales. Uno de los trabajos realizados ha sido el de Norma Covarrubias [1] quien desarrolló un algoritmo utilizando el método de Manipulación de Espacio de Cámara (CSM) para automatizar la tarea de identificación de la superficie de trabajo de un robot manipulador industrial.

Zhang [32] desarrolló su tesis doctoral acerca de la robustez del método de CSM y su implementación en aplicaciones de los robots en la industria para tareas, tales como, montaje de piezas cubo-flecha con tolerancias muy cerradas (aproximadamente de 0.5 mm), la estiba de piezas sobre tarimas para ser manipuladas y organizadas en estantes; así también como trabajos de acabado de superficies, que involucra el seguimiento de trayectorias sobre superficies planas de las piezas de trabajo, obteniendo un acabado de superficie. Zhang reporta un error de ± 0.1 mm en acabados

sobre tablas de madera.

Organización de la tesis

Este trabajo se organizó de tal forma que, en el capítulo 1 se realiza una revisión de los fundamentos teóricos del proyecto. En el capítulo 2 se describen los fundamentos y los algoritmos de visión usados para detectar la posición y orientación de las piezas de trabajo. En el capítulo 3 se explica el método utilizado para la manipulación y estiba de las piezas de trabajo. En el capítulo 4 se muestra la implementación y pruebas experimentales obtenidas. Finalmente en el capítulo 5 se presentan los resultados de este trabajo así como las correspondientes conclusiones.

Sistemas de visión por computadora

Visión por computadora es la habilidad de una computadora por la cual puede “*ver*”. Esta disciplina puede ser definida como el proceso de extracción de información del mundo a partir de imágenes utilizando como herramienta una computadora. La entrada de información a un sistema de visión es a partir de una o varias imágenes bidimensionales obtenidas por un elemento de adquisición o cámara. Los sistemas de visión por computadora son diseñados para reducir o eliminar la necesidad de la supervisión humana de una tarea específica, lo cual se busca que reduzca los costos de mano de obra. Las aplicaciones en general de los sistemas de visión son en áreas tales como: medición, inspección y orientación de objetos. Los sistemas de medición determinan las dimensiones de un objeto de acuerdo con el campo de visión de cámara. Los sistemas de inspección determinan si un objeto coincide con las descripciones predeterminadas. Los sistemas de orientación permiten a una computadora tomar ciertas decisiones de acción basadas en la posición de rasgos distintivos de los objetos en espacio de cámara.

1.1. Adquisición de imágenes

La adquisición de imágenes digitales requiere de dos elementos básicos, el primero un dispositivo físico sensible a una determinada banda del espectro electromagnético (tal como rayos X, ultravioleta, infrarrojo, sonar, escáner de láser) y que produce una señal eléctrica de salida proporcional al nivel de energía incidente en un instante de tiempo; el segundo elemento denominado digitalizador, es un dispositivo para convertir la señal eléctrica de salida continua del primer dispositivo físico en forma digital.

Como se muestra en la figura 1.1, una imagen es adquirida por una cámara digital de la siguiente manera: la luz entra a la cámara por medio del lente; la luz es captada por un sensor fotosensible conocido como dispositivo de acoplamiento de carga ó CCD (Charged Couple Device), el cual tiene una superficie en forma de rejilla o arreglo matricial, compuesto de miles de fotómetros que traduce la cantidad de luz (fotones) en electrones (cargas eléctricas), esta señal eléctrica es transformada por un conversor analógico-digital (ADC) en información binaria (bytes).

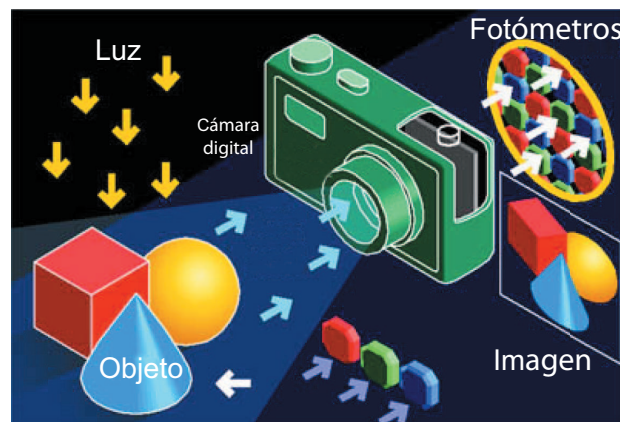


Figura 1.1: Adquisición de una imagen digital

Ya que se obtuvo información de la imagen, un digitalizador de video (*Frame Grabber*) captura cuadros individuales, en el que los fotogramas de vídeo se convierten a formato digital y posteriormente se visualizan, almacenan o transmiten directamente a otros dispositivos. Una imagen monocromática sólo contiene información de brillo (o nivel de gris) que se asigna a cada elemento de imagen o pixel (abreviatura del inglés *picture elements*) correspondiente. El valor de cada pixel es cuantificado a código binario entero positivo (el brillo no puede ser negativo). El número de niveles de cuantificación está determinado por la relación $L = 2^B$, donde B representa el número de bits necesarios para cada muestra y L el número de niveles de gris. Para tener una imagen de buena calidad para el ojo humano es necesario tener como mínimo 64 niveles de gris [13]. Para una apreciación fina se usa una cuantificación estándar a 8 bits, esto es, 256 niveles de gris. A una imagen cuantificada de esta manera se le conoce como imagen en tonos de gris. La cuantificación a 8 bits (1 byte) se dice estándar porque es el mínimo número direccionable directamente por la mayoría de los microprocesadores. Una imagen binaria es una imagen monocromática cuantificada a 1 bit por pixel, esto es, dos niveles, blanco y negro.

Una imagen digital es un arreglo bidimensional $f(x, y)$ que ha sido identificado en coordenadas (x, y) y en la cantidad de brillo f , como se muestra en la figura 1.2. Esta imagen puede considerarse como una matriz cuyos índices del renglón y columna identifican un punto en la imagen y el correspondiente valor del elemento de la matriz que identifica el valor del pixel.

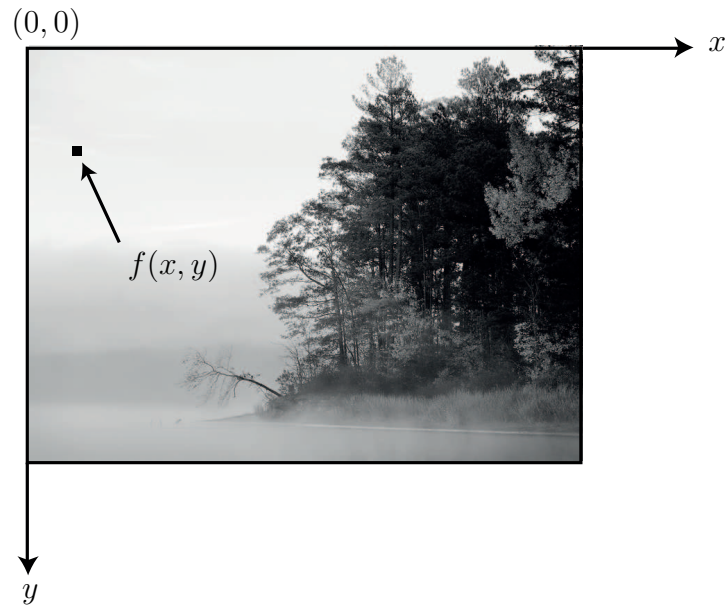


Figura 1.2: Convención de ejes utilizada para la representación de imágenes digitales

1.2. Análisis de imágenes

A diferencia del estudio de los mecanismos de la visión humana, el procesamiento y análisis de imágenes digitales nacieron en el momento en que se disponen de recursos tecnológicos para captar y manipular, en forma de matrices de valores, gran cantidad de información. Históricamente, la primera vez que se hizo uso de las técnicas de procesamiento de imágenes fue en los años veinte, cuando se transmitieron imágenes de fotografías periodísticas por cable submarino entre Londres y New York, [13]. La introducción del sistema de transmisión de imágenes por cable a través del Atlántico, conocido como sistema Bartlante, redujo el tiempo de transmisión de más de una semana a menos de tres horas. Inicialmente, el sistema era capaz de codificar imágenes en 5 niveles de gris distintos, capacidad que se vio aumentada a 15 niveles hacia 1929.

Las etapas del proceso de visión por computadora se suelen clasificar en tres niveles de información: bajo, medio y alto. La información de bajo nivel está dada por

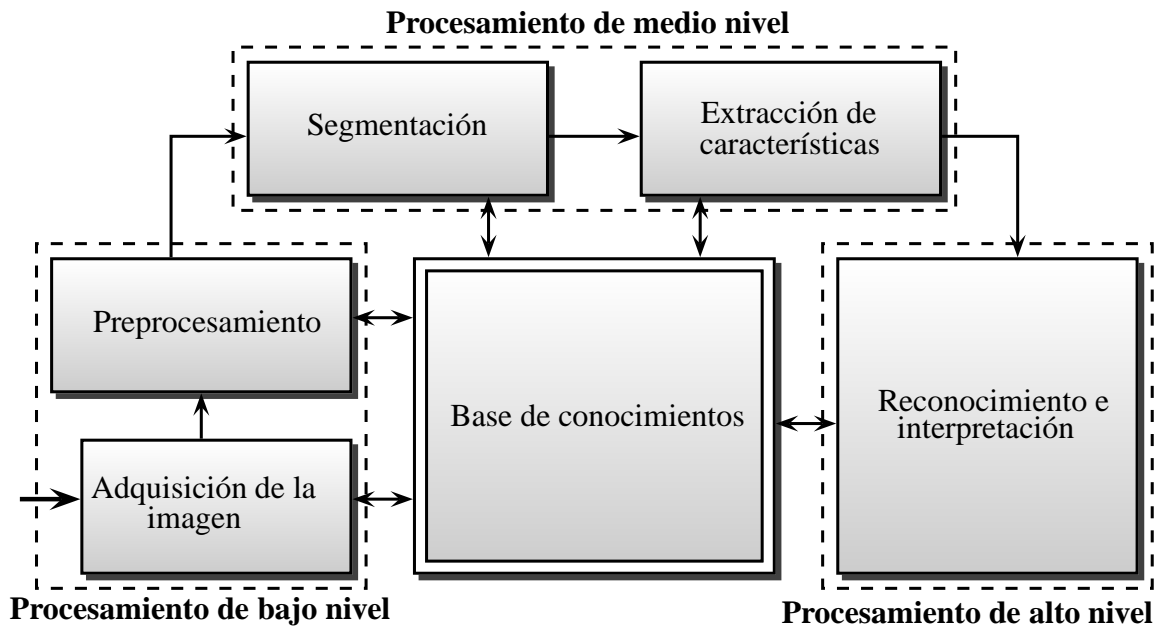


Figura 1.3: Etapas del proceso de visión artificial

las etapas de adquisición de imágenes y preprocesamiento, las de medio nivel son las de segmentación y extracción de las características y las de alto nivel con las etapas de reconocimiento e interpretación. En la figura 1.3 se presenta un diagrama de flujo de las distintas etapas de una aplicación artificial y su clasificación en los distintos niveles de información.

No necesariamente deben cubrirse todas las etapas en una implementación concreta. Por otro lado, aunque el diagrama indique un encadenamiento entre una etapa y otra, no es real esta simplificación; se hace para facilitar la comprensión, en la puesta en práctica siempre se encuentra retroalimentación entre las distintas etapas.

A continuación, se describen de forma breve las etapas del proceso de visión por computadora:

- *Adquisición de la imagen.* En esta etapa se capturan las imágenes digitales utilizando digitalizadores de video o cámaras digitales.
- *Preprocesamiento.* Su función es mejorar la imagen. El preprocesamiento típicamente trata con técnicas de eliminación de ruido y realce de características de la imagen, como pueden ser bordes o áreas.
- *Segmentación.* Permite separar la imagen en los diferentes elementos o áreas de interés.
- *Extracción de características.* Se obtiene una representación formal de los elementos segmentados en la etapa anterior.
- *Reconocimiento e interpretación.* Es la etapa donde se asignan etiquetas de acuerdo con la información que se adquirió de los elementos discretizados de la etapa anterior. Es decir, se reconoce con precisión los objetos de interés mediante técnicas de clasificación, como redes neuronales, sistemas expertos, lógica difusa o reconstrucción tridimensional.
- *Base de conocimientos.* La base de conocimientos mantiene toda la información que se tiene acerca del proceso de visión en cuestión. Este conocimiento puede ser simple, teniendo solo las regiones de la imagen donde la información de interés se conoce que está localizada, con lo que se limita la búsqueda que conduce a esa información. Además de guiar la operación de cada etapa del proceso, la base de conocimientos también controla la interacción entre las demás etapas.

Es importante destacar que los resultados del análisis de imágenes puede verse en la salida de cualquier etapa de la figura 1.3 y que no todas las aplicaciones del procesamiento de imágenes requieren de la complejidad de interacciones como se mostró en esta figura. No todos los módulos son siempre necesarios, por ejemplo,

el realce de imágenes para la interpretación visual humana rara vez va más allá de la etapa de preprocesamiento. En general, las funciones de procesamiento que incluyen reconocimiento e interpretación están asociadas con aplicaciones de análisis de la imagen en la cual el objetivo es extraer información de una imagen en forma automática o semiautomática, como es el caso del presente trabajo.

1.3. Procesamiento de imágenes

Aunque la distinción entre procesamiento y análisis de imágenes digitales no es obvio de forma inmediata, el procesamiento de imágenes puede ser visto como una transformación de imagen en otra, es decir a partir de una imagen se obtiene otra imagen modificada. Por otro lado, el análisis es una transformación de una imagen en algo distinto, como sería el caso cuando se realiza la extracción de información tridimensional a partir de dos o más imágenes.

El objetivo del procesamiento de imágenes es hacer el análisis posterior más simple y más fiable. En esta etapa son típicas las transformaciones basadas en la aplicación de una función sobre el valor del nivel de gris de los píxeles como se mencionó en la sección 1.1 y aquellas otras en las que el valor de la intensidad de los píxeles se modifica en función de los valores de las intensidades de los píxeles vecinos.

El procesamiento de imágenes en el sistema de visión puede considerarse desde 2 perspectivas [13] y [20]:

1. Alteración pixel a pixel de los datos en una escala global (individuales).
2. Operaciones basadas en múltiples puntos (vecindad).

La generación de un nuevo pixel en una nueva imagen será una función del valor de cada pixel en su localización individual o bien de los valores de los píxeles en la

vecindad de un pixel dado, como se muestra en la figura 1.4:

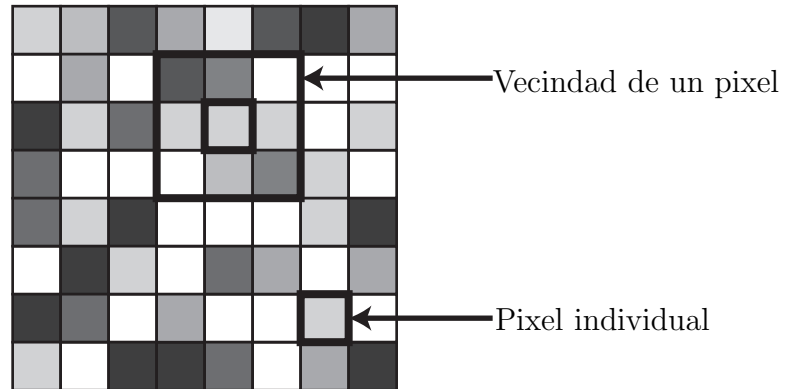


Figura 1.4: Funciones de punto y vecindad

1.3.1. Operaciones individuales

Las operaciones individuales implican la generación de una nueva imagen modificando el valor del pixel en una simple localización basándose en una regla global aplicada a cada localización de la imagen original. El proceso consiste en obtener el valor del pixel de una localización dada en la imagen, modificándolo por una operación lineal o no lineal y colocando el valor del nuevo pixel en la correspondiente localización de la nueva imagen. El proceso se repite para todas y cada una de las localizaciones de los pixeles en la imagen original.

Como se aprecia en la figura 1.5, el operador f se aplica a cada pixel en la imagen o sección de la imagen y la salida depende únicamente de la magnitud del correspondiente pixel de entrada; la salida es independiente de los pixeles adyacentes. La función transforma el valor del nivel de gris de cada pixel en la imagen y el nuevo valor se obtiene a través de la siguiente ecuación:

$$q(x, y) = f(p(x, y)) \quad (1.1)$$

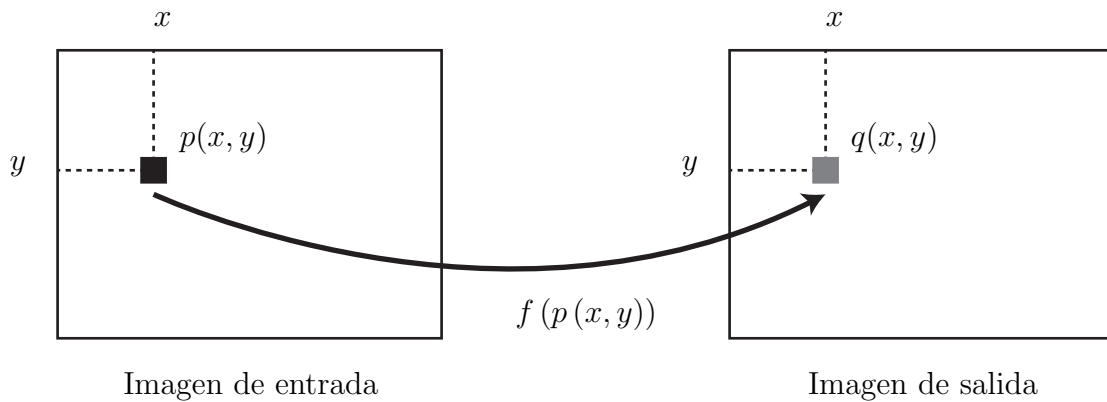


Figura 1.5: Operación individual para una imagen

Entre las funciones más comunes utilizadas en el campo de la visión por computadora destacan: operador umbral y transformaciones de dos imágenes punto a punto, las cuales se discuten en las siguientes subsecciones.

Operador umbral

Esta clase de transformación crea una imagen de salida binaria a partir de una imagen en escala de grises, donde el nivel de transición está dado por un parámetro de entrada p_1 . La función de transformación es la siguiente:

$$q = \begin{cases} 0 & \rightarrow p \leq p_1 \\ 255 & \rightarrow p > p_1 \end{cases} \quad (1.2)$$

En la figura 1.6 se muestra la imagen obtenida al aplicar la transformación según la ecuación 1.2 a la imagen de la figura 1.6 con valor de umbral $p_1 = 185$.

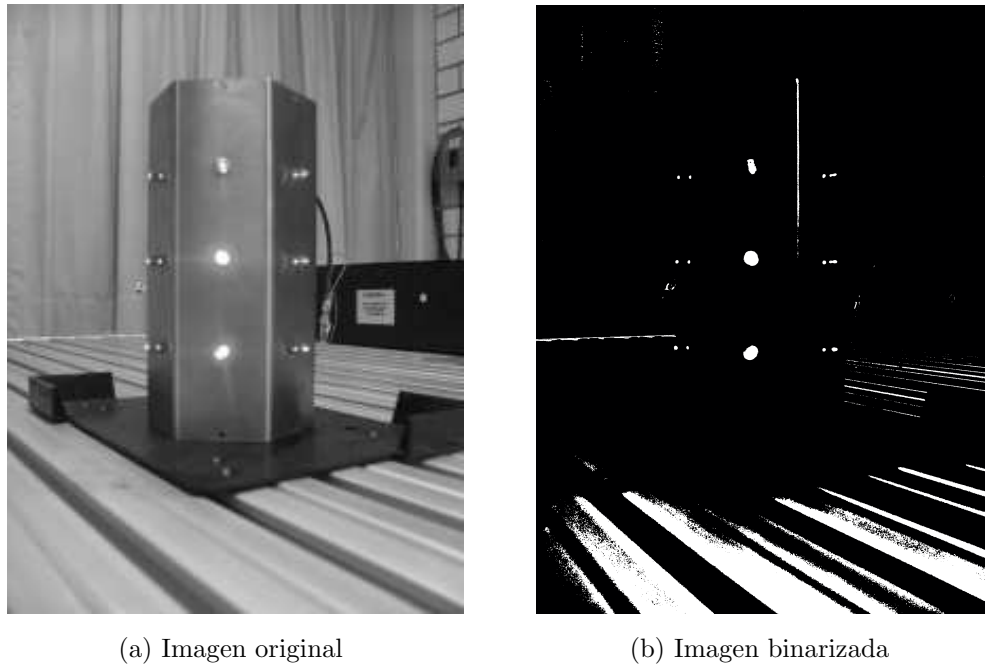


Figura 1.6: Transformación de una imagen por medio del operador umbral

Transformaciones de dos imágenes punto a punto

Éstas utilizan la información contenida en la misma localización (posición de los píxeles) de dos imágenes de entrada a y b para crear una nueva imagen c . La dimensión de las imágenes es la misma. La función de la transformación f_D puede ser lineal o no. Esta función se aplica a todos los pares de píxeles en las imágenes de entrada. Esto es, la información en una localización de píxel de una imagen se combina con la información de la correspondiente localización de la segunda imagen para obtener el valor de la tercera imagen. La función característica está dada por la siguiente ecuación:

$$c_{x,y} = f_D(a_{x,y}, b_{x,y}) \quad (1.3)$$

donde f_D es una función de dos variables y los subíndices x e y varían de 0 a M y de 0 a N respectivamente (donde M y N corresponden a las dimensiones de la imagen). La función f_D puede ser adición, sustracción, multiplicación, división, exponenciación, máximo o cualquier otra función que se pueda definir. La función deberá tener un factor de escala apropiado k para mantener los valores de salida dentro del rango adecuado, así como para evitar desbordamientos o valores negativos.

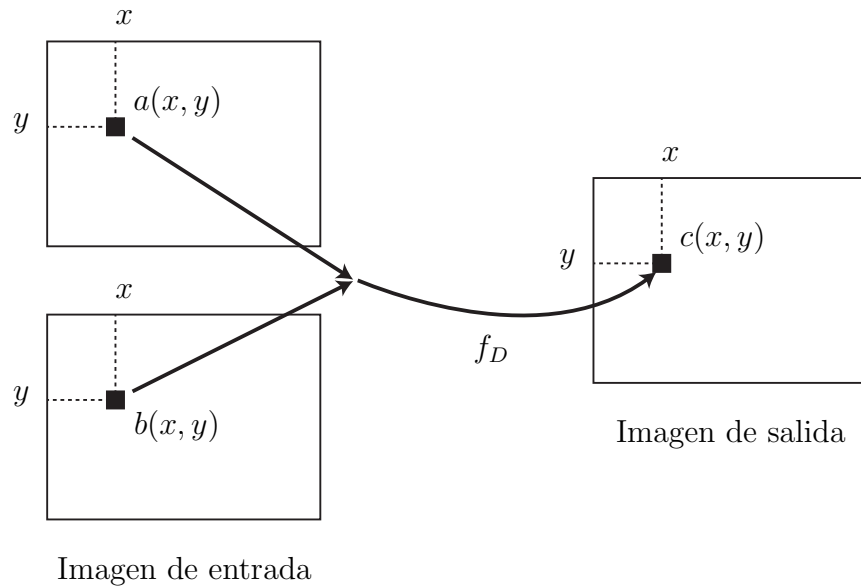


Figura 1.7: Representación simbólica de la transformación de dos imágenes en una

La *sustracción de imágenes* es una técnica útil para detectar el cambio producido en dos imágenes que han sido captadas en dos instantes de tiempo diferente, mientras que la cámara con la cual fueron capturadas se mantuvo en una posición fija. Puesto que en el procesamiento de imágenes se utilizan valores positivos, es necesario definir la salida de tal manera que produzcan todos los valores con este signo. Esto implica

un reescalado donde el valor más negativo que le corresponde es 0 y el máximo con 255. En este caso la relación esta dada por:

$$c_{x,y} = k(a_{x,y} - b_{x,y}) \quad (1.4)$$

donde k es una función no lineal definido de modo tal que el valor mínimo que toma $c_{x,y}$ es 0 y 255 el máximo.

1.4. Extracción de bordes

Se considera como *bordes* a los pixeles alrededor de los cuales la imagen presenta una brusca variación en los niveles de gris. El objetivo consiste en, dada una imagen que puede o no estar corrompida por ruido, localizar los bordes más probables generados por elementos de la escena y no por ruido.

El término borde se refiere a cadenas conectadas de puntos o pixeles de borde, esto es, fragmentos de contorno. Esto no impide que la imagen también pueda contener puntos aislados que presentan un alto contraste en los niveles de gris. Existen varias razones por los cuales existe un interés en la extracción de bordes. Los contornos de los objetos sólidos de la escena, las marcas en las superficies, las sombras, todas generan bordes. Además las líneas de las imágenes, las curvas y contornos son elementos básicos para muchas aplicaciones tales como estereopsis, calibración, movimiento o reconocimiento.

La detección de bordes es una parte fundamental de la mayoría de los sistemas de visión puesto que el éxito de las etapas siguientes de procesamiento depende fuertemente de la fiabilidad de las características, en este caso bordes. La filosofía básica de muchos algoritmos de detección de bordes es el cómputo del cálculo de derivadas

locales (primera o segunda).

En la figura 1.8 se puede observar que los bordes (transición de oscuro a claro o viceversa) se modelan como una rampa en lugar de hacerlo como un cambio brusco de intensidad, debido a que en la imagen original suelen estar borrosos como resultado del muestreo.

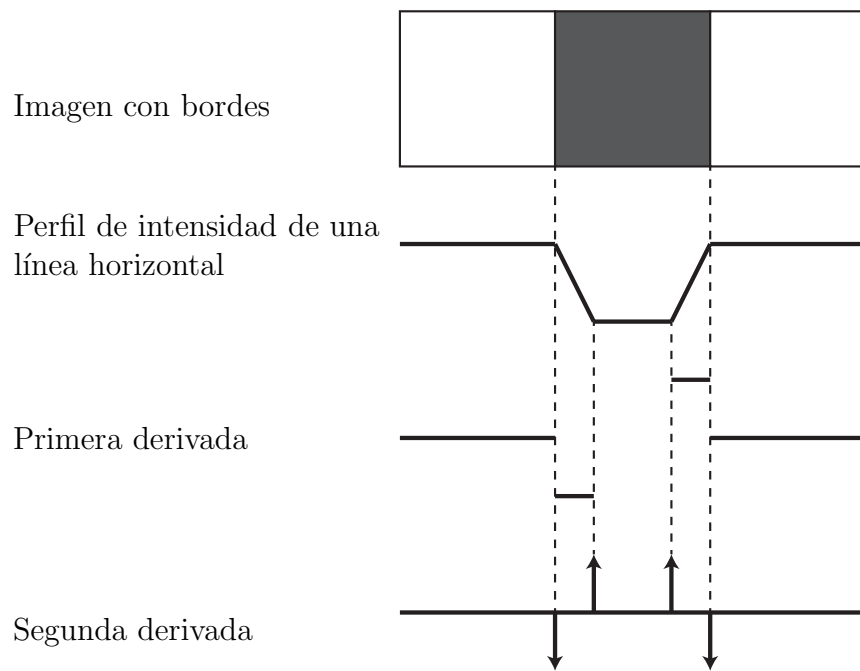


Figura 1.8: Diagrama del concepto de primera y segunda derivada para la extracción de bordes

La primera derivada es cero en todas las regiones de intensidad constante y tiene un valor constante en toda la transición de intensidad. La segunda derivada, en cambio, es cero en todos los puntos, excepto en el comienzo y final de una transición de intensidad. Por tanto, un cambio de intensidad se manifiesta como un cambio brusco en la primera derivada y presenta un paso por cero, es decir se produce un cambio

de signo en su valor, en la segunda derivada.

Basándose en estas observaciones y en los conceptos ilustrados en la figura, 1.8 es evidente que el valor de la primera derivada puede utilizarse para detectar la presencia de un borde así como el signo en la segunda derivada.

De entre los operadores más utilizados basados en la primera derivada se encuentran los de *gradiente*, una de sus principales aplicaciones se debe a Canny [14], [15] y [16], quien utiliza el concepto de gradiente en su algoritmo, en el que utiliza cuatro filtros para detectar bordes en horizontal, vertical y diagonal de la imagen, este algoritmo no sólo extrae los bordes sino que propone un método para cerrar contornos como se observa en la figura 1.9. El concepto del gradiente ha originado diversos métodos para su aplicación en la extracción de bordes, como lo son los operadores de Sobel, Prewitt, Roberts, Kirsch, Robinson, Frei-Chen, como resultado de estos se obtiene el valor para la primera derivada en la dirección horizontal (G_y) y la dirección vertical (G_x), a partir de éste, se pueden determinar el gradiente de borde G y la dirección ϕ mediante las siguientes ecuaciones.

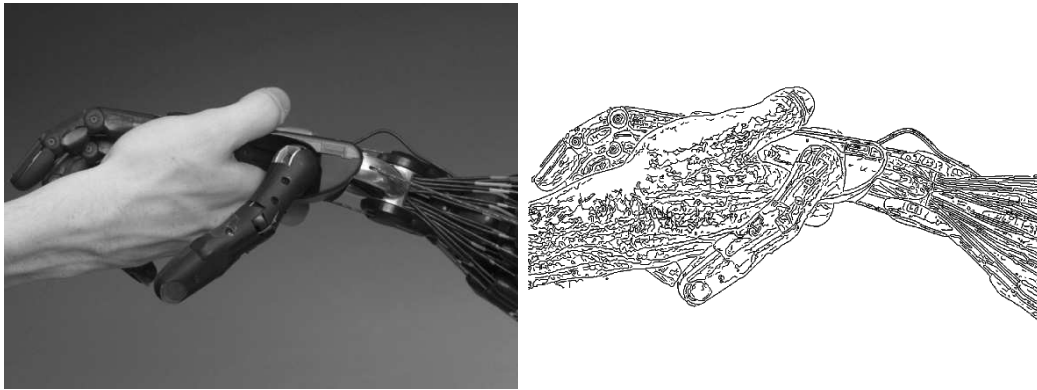
$$G[f(x, y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} f(x, y) \\ \frac{\partial}{\partial y} f(x, y) \end{bmatrix}, \quad (1.5)$$

$$|G| = \sqrt{G_x^2 + G_y^2} = |G_x| + |G_y|,$$

$$\phi(x, y) = \tan^{-1} \frac{G_y}{G_x}$$

1.4.1. Filtros morfológicos

La morfología matemática es un método no lineal de procesamiento de imágenes binarias basadas en formas. Su principal objetivo es la cuantificación de estructu-



(a) Imagen original

(b) Imagen resultado

Figura 1.9: Imagen obtenida con el procesamiento del algoritmo de Canny

ras geométricas. Las principales operaciones morfológicas son la dilatación, erosión, apertura y cierre. El operador de dilatación es comúnmente conocido como “relleno”, “expansión” o “crecimiento”. Puede ser usado para rellenar “huecos” de tamaño igual o menor que el elemento estructurante (también conocido como operación basada en múltiples puntos o vecindad) con la que se opera la dilatación, por tal motivo es muy común utilizarlo en conjunción de técnicas de extracción de bordes, ya que aumenta las probabilidades de localizar bordes cerrados en la imagen. En la ecuación 1.6 se representa este operador morfológico.

$$C = A \oplus B \quad (1.6)$$

donde A es la imagen de origen, B el elemento estructurante y C la imagen resultante. Entonces, si el pixel de la imagen A no es cero, cada pixel que cae en la estructura B es añadido a la imagen C aplicando el operador “or”.

1.5. Descripción de figuras geométricas en imágenes

En la actualidad se ha planteado la problemática sobre la extracción de figuras a partir de una imagen arbitraria. La cuestión sobre cómo resuelven este problema las personas ha intrigado durante años a los psicólogos, particularmente a los de la escuela Gestalt. Sin embargo, no ha surgido una teoría clara que resuelva esta cuestión, ya que existen ilusiones ópticas en las cuales la figura y el fondo aparecen alternantes. Por tal motivo, no existe un algoritmo que de forma general pueda extraer las figuras de las imágenes. Existe una variedad de técnicas que son útiles en situaciones particulares, como por ejemplo la extracción de bordes, técnicas de binarización, sustracción de imágenes, metodologías de descripción de líneas o la combinación de varias de éstas.

Existe una metodología para la descripción de líneas muy útil en la descripción de figuras conocida como transformada de Hough [18], su objetivo es encontrar la ecuación de una línea que pasa por un conjunto de n puntos en el plano xy . Una línea que une una secuencia de puntos o píxeles puede expresarse de la forma $y = mx + b$ en la que se evalúan los parámetros m y b . En base a esto, la línea se puede representar como un punto (b, m) en el espacio de la imagen. Sin embargo, esto se enfrenta al problema de que las líneas verticales dan lugar a valores infinitos de los parámetros m y b . Por razones de cómputo, es conveniente utilizar diferentes parámetros, como lo son r y θ , para las líneas de la transformada de Hough.

El parámetro r representa la distancia entre la línea y el origen, mientras que θ es el ángulo del vector r . Por tanto, la ecuación de la recta puede escribirse como:

$$y = \left(-\frac{\cos \theta}{\sin \theta} \right) x + \left(\frac{r}{\sin \theta} \right) \quad (1.7)$$

reorganizando la ecuación obtenemos:

$$r = x \cos \theta + y \sin \theta \quad (1.8)$$

Esto corresponde a una curva sinusoidal en el plano (r, θ) , la cual es única en un punto. Si las curvas correspondientes a dos puntos están superpuestas, la ubicación (en el espacio de Hough) donde se intersectan corresponde a una línea (en el espacio de la imagen original) que pasa a través de ambos puntos. De manera más general, un conjunto de puntos que forman una línea recta producirán curvas sinusoidales que se intersectan. Un ejemplo de la aplicación de esta metodología puede observarse en la figura 1.10.

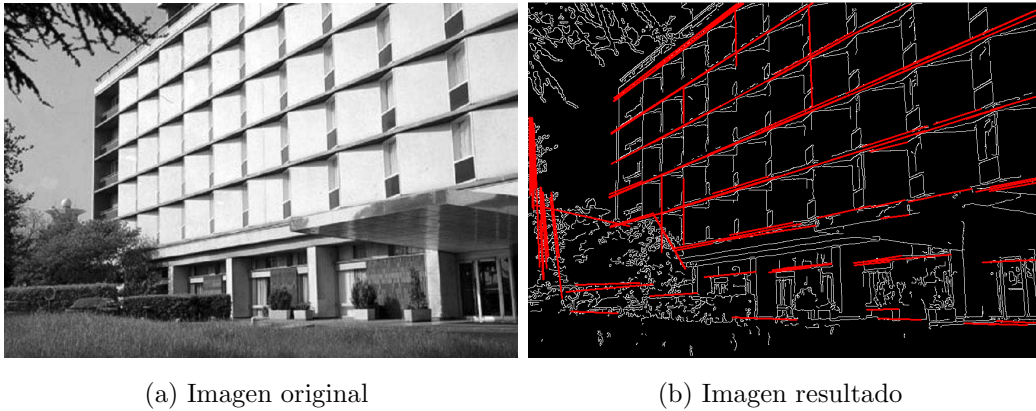


Figura 1.10: Descripción de líneas en una imagen por el método de la transformada de Hough

1.6. Modelos de cámara

Con el objetivo de poder deducir la posición y/o orientación de un objeto en una imagen, es necesario establecer la posición de la cámara y orientación en el espacio con respecto a algún sistema de coordenadas de referencia, denominado como *sistema de coordenadas del mundo*. Además de esto, es preciso establecer un *modelo de cámara* y algún método o proceso para encontrar los distintos parámetros del modelo. La elección del modelo de cámara depende en gran parte de la aplicación en particular del sistema de visión. A continuación se presentan los modelos de cámara de orificio y ortográfico, de los cuales el primero forma parte de la metodología modificada de CSM, al cual se le denomina LCM-CSM, mientras que el método de CSM esta basado en un modelo de cámara ortográfico.

1.6.1. Modelo de cámara de orificio

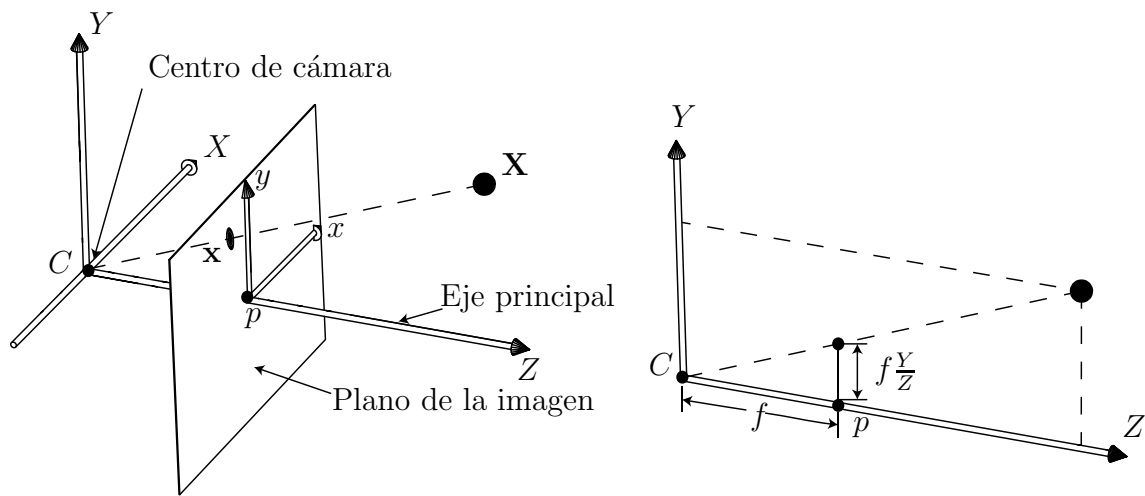


Figura 1.11: Geometría de la cámara de orificio

En el “*modelo de cámara de orificio (pinhole)*”, un punto en el espacio con coordenadas $\mathbf{X} = (X, Y, Z)$ se representa por un punto en la imagen, donde la línea que une el punto \mathbf{X} con el centro de proyección incide en el plano de la imagen (ver figura 1.11). Por triángulos semejantes se observa que el punto (X, Y, Z) es “*mapeado*” al punto $(f \frac{X}{Z}, f \frac{Y}{Z}, f)$ en el plano de la imagen.

De acuerdo con la relación anterior que describe el mapeo de proyección central del mundo al sistema de coordenadas de la imagen es la siguiente:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \mapsto \begin{bmatrix} f \frac{X}{Z} \\ f \frac{Y}{Z} \end{bmatrix} \quad (1.9)$$

Esto representa un mapeo de un espacio tridimensional Euclidiano \mathbb{R}^3 a un espacio bidimensional Euclidiano \mathbb{R}^2 .

El centro de proyección se llama *centro de cámara* (punto C de la figura 1.11), también conocido como *centro óptico* o *punto focal*. La línea que parte del centro óptico y es perpendicular al plano de la imagen se le conoce como *eje principal* o *rayo principal*, y el punto donde el eje principal coincide con el plano de la imagen se denomina *punto principal*. El plano que pasa a través del centro de la cámara y que es paralelo al plano de imagen se conoce como *plano principal de la cámara* (plano XY). A este modelo también se le denomina “*modelo de perspectiva*”.

1.6.2. Modelo de cámara ortográfica

Una simplificación al modelo de perspectiva ó modelo de cámara de orificio, ocurre cuando se realiza una proyección ortográfica de un punto en el espacio \mathbf{X} sobre un plano por un rayo paralelo ortogonal a este plano. Considere la figura 1.12 en lo que se refiere a la geometría de la proyección ortográfica, se tiene que $X = x$ e $y = Y$

donde (X, Y) y (x, y) denotan las coordenadas del punto en el espacio y de la imagen respectivamente.

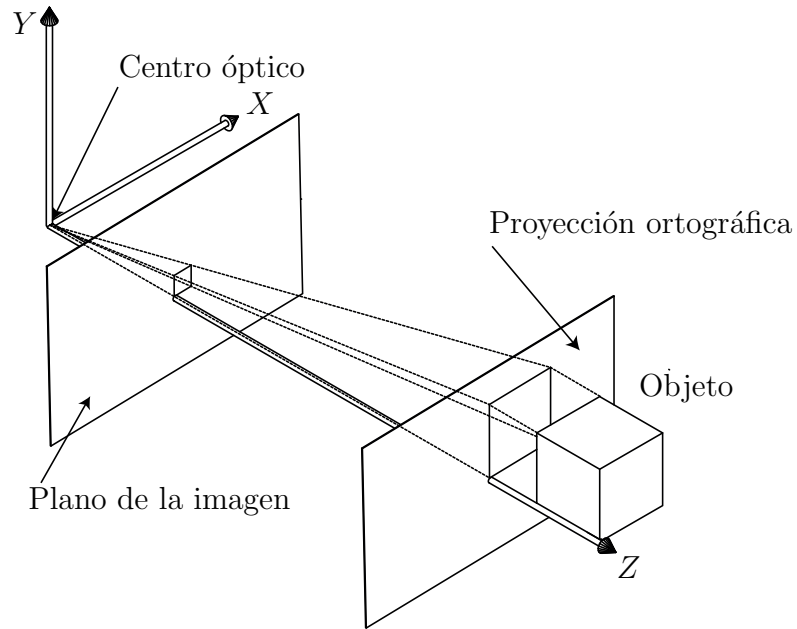


Figura 1.12: Aproximación de la proyección de perspectiva por proyección ortográfica.

Considerar la geometría de la cámara de orificio o proyección de perspectiva de la figura 1.11. A medida que el objeto se aleja del centro óptico a lo largo del eje Z , la imagen se hace más pequeña. Y lo que es más importante, el factor de reducción f/Z en la relación (1.9) se hace menos sensible a Z . Esto es, $f/(Z + \Delta Z)$ tiende a tomar un valor próximo a f/Z a medida que $Z/\Delta Z$ tiende a ser grande, esto significa que la distancia a la que se encuentra un objeto Z es grande con relación al rango de distancias en el propio objeto ΔZ . Esta aproximación de f/Z en lugar de $f/(Z + \Delta Z)$ podría llevar a creer que, siempre que la distancia a la que se encuentra un objeto es grande con relación al rango de distancias en el propio objeto, la proyección de perspectiva puede aproximarse por proyección ortográfica hasta un factor de escala

f/Z . No obstante, esta hipótesis es incorrecta [19]. Para que dicha hipótesis tenga cierta consistencia, es decir, para que la proyección de perspectiva pueda aproximarse por proyección ortográfica hasta un factor de escala uniforme se han de cumplir dos condiciones:

1. El objeto debe situarse próximo al eje principal.
2. Las dimensiones del objeto deben ser pequeñas.

Los términos *próximo* y *pequeño* son con respecto a la distancia del objeto a partir del centro de proyección. La figura 1.12 ilustra gráficamente la aproximación de proyección de perspectiva como un proceso de dos pasos: proyección ortográfica sobre un plano próximo al objeto y paralelo al plano de la imagen y luego proyección de perspectiva sobre el plano de la imagen. Lo último es equivalente a un escalado uniforme.

En el presente capítulo se expusieron los antecedentes de los sistemas de visión y se describieron algunas técnicas de procesamiento y análisis de imágenes que permiten obtener información del entorno a partir de las imágenes adquiridas. Se dio una descripción de dos modelos de cámara comúnmente utilizados en los sistemas de visión, los cuales son la base para el método de control basado en visión, Manipulación en Espacio de Cámara o CSM, y el cual se describe en el siguiente capítulo.

Método de Control basado en Visión para Robots

En el presente capítulo se describen los principales fundamentos del método de control basado en visión para robots, utilizado en este trabajo. Debido a que en este trabajo se utilizó un robot del tipo manipulador para la estiba de piezas, se presentarán los antecedentes de los robots del tipo serial. Otro aspecto importante presentado en este capítulo son los sistemas de visión por computadora y el análisis de imágenes digitales, es decir los procedimientos por los cuales se extrae la información necesaria de las imágenes digitales y su proceso para adquirir puntos de referencia para el robot. Finalmente, se describen algunas de las principales metodologías utilizadas para controlar la posición y orientación de la herramienta colocada en el efector final del robot manipulador mediante visión computacional.

2.1. Robots industriales

El término **robot** fue introducido por primera vez en 1920 por el checo Karel Capek y se deriva de la palabra “*robot*” que en checo significa trabajo. Desde entonces el término ha sido aplicado a todo aquel dispositivo mecánico que opera con algún

grado de autonomía, generalmente controlado por computadora.

Una definición oficial del término robot viene de la Asociación de Industriales en Robótica (RIA por sus siglas en inglés): *Un robot es un manipulador multifuncional reprogramable diseñado para mover material, partes, herramientas o dispositivos especializados por medio de movimientos variables programados, para el desarrollo de una variedad de tareas*,[\[9\]](#),[\[10\]](#).

2.1.1. Robots manipuladores

Un robot manipulador es un brazo mecánico articulado, controlado por computadora y capaz de realizar una variedad de tareas, tales como desplazar objetos en un espacio de trabajo determinado. Un robot de este tipo está compuesto por eslabones conectados por juntas (uniones o articulaciones) para formar una cadena cinemática abierta, la herramienta con la cual el robot puede realizar distintas tareas (tales como el paletizado de material, estampar, soldar, pintar, etc.) se monta sobre la última articulación del robot y se denomina comúnmente como efector final.

A las juntas de un robot manipulador se les puede clasificar del tipo de: de rotación (revoluta) o lineal (prismática). Una junta de revoluta es como una bisagra y permite un giro relativo entre dos eslabones. Una junta prismática permite un movimiento en línea recta entre dos eslabones. La cantidad de juntas con las que cuenta determina el número de grados de libertad del mismo. Una de las configuraciones más comunes en los robots manipuladores son aquellos que cuentan con seis grados de libertad, de las cuales tres son usadas para el posicionamiento y las otras tres para la orientación del efector final del robot,[\[11\]](#).

2.1.2. Modelo cinemático directo de un robot manipulador

El modelo cinemático directo de los robots establece la relación entre la posición y la orientación del marco de referencia del efector final con respecto al de la base. Éste se basa en transformaciones entre sistemas de referencia, es decir, la relación que existe entre los ángulos de giro de cada junta y la posición y orientación del marco de referencia del efector final. Para determinar el modelo cinemático es necesario definir un marco de referencia fijo en cada eslabón del manipulador. Una de las convenciones usadas para la selección sistemática de los marcos de referencia asignados a cada eslabón es la representación de Denavit y Hartenberg [12].

En esta convención, cada transformación homogénea A_i se representa por un producto de cuatro transformaciones básicas como se muestra en la ecuación (2.1).

$$A_i = Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,\alpha_i} Rot_{x,\alpha_i} \quad (2.1)$$

Usando la notación Rot_{z,θ_i} para representar a la matriz de rotación del ángulo θ_i alrededor del eje z y donde $Trans_{z,d_i}$ representa la matriz de translación de una distancia d_i a lo largo del eje z . La ecuación (2.1) puede ser representada en forma matricial como sigue:

$$A_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

donde:

a_i representa longitud;

α_i se refiere a un ángulo;

d_i es el desplazamiento de una junta prismática;

θ_i corresponde al desplazamiento de una junta de revoluta.

Los parámetros a_i , d_i , θ_i y α_i están relacionados al eslabón i . La matriz A_i es función de una sola variable y tres de los cuatro parámetros son constantes para un eslabón determinado. En el caso de una junta o articulación de revoluta, θ_i es variable, mientras que d_i es variable para una junta de tipo prismática.

2.1.3. Modelo cinemático del robot Motoman HP3

En el laboratorio de Mecatrónica de la Facultad de Ingeniería de la Universidad Autónoma de San Luis Potosí se cuenta con un robot manipulador industrial de la marca Motoman del modelo HP3, ver figura 2.1. El modelo cinemático del robot está definido por:

$$H = A_1 \cdot A_2 \cdot A_3 \cdot A_4 \cdot A_5 \cdot A_6 \quad (2.3)$$

en esta expresión las matrices de transformación para cada junta del modelo cinemático del robot son:

$$A_1 = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_2 = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & 100 \\ 0 & 0 & 1 & 0 \\ -\sin \theta_2 & -\cos \theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & 0 \\ -\sin \theta_3 & -\cos \theta_3 & 0 & -290 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_4 = \begin{bmatrix} 0 & 0 & -1 & 300 \\ \sin \theta_4 & \cos \theta_4 & 0 & 85 \\ \cos \theta_4 & -\sin \theta_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_5 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ \sin \theta_5 & \cos \theta_5 & 0 & 0 \\ -\cos \theta_5 & \sin \theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_6 = \begin{bmatrix} 0 & 0 & -1 & 90 \\ \sin \theta_6 & \cos \theta_6 & 0 & 0 \\ \cos \theta_6 & -\sin \theta_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

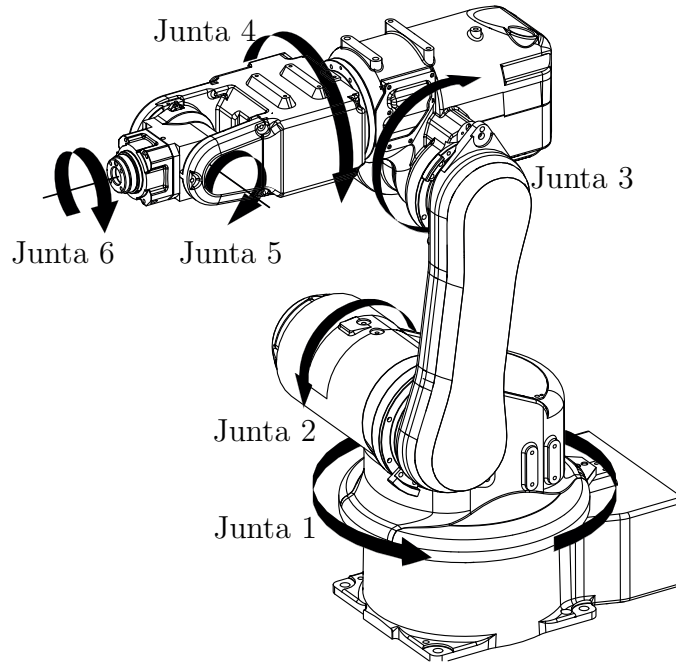


Figura 2.1: Robot manipulador Motoman HP3

2.2. Método de Manipulación en Espacio de Cámara

El Método de Manipulación de Espacio de Cámara CSM (Camera Space Manipulation) es una técnica basada en visión por computadora, el cual se caracteriza por

ser un método libre de calibración. El sistema de visión permite que el robot pueda ser posicionado y orientado con un alto nivel de precisión y robustez con respecto a una pieza de trabajo, [21], [22]. Debido a esta característica resulta ventajoso en la solución al problema de colocación para distintas operaciones industriales como manufactura, ensamble y estiba de materiales.

El método de manipulación de espacio de cámara (CSM) deriva su exactitud y robustez a partir de que lleva a cabo un control total entre el robot y el plano de la imagen bidimensional de referencia de cada cámara participante, [1], [11], [23]-[26]. Este método de control basado en visión requiere de al menos dos cámaras colocadas en diferente posición una respecto a la otra. CSM se basa en una relación algebraica no lineal entre la configuración interna del robot y su localización en espacio de cámara, esto último se logra a través de la detección de marcas visuales colocadas en el robot; esta relación algebraica se realiza por medio del modelo cinemático del robot y un modelo ortográfico de cámara. Esta relación algebraica está definida por medio de seis “*parámetros de visión*” \mathbf{C} , los cuales se estiman por medio de un proceso de estimación no lineal.

Considerando el “*modelo de cámara de orificio (pin-hole)*” como la relación no lineal entre la ubicación física y la respuesta correspondiente en espacio de cámara de una marca visual unida a un objeto manipulable determinado, de acuerdo con la relación (1.9) podemos reescribir:

$$x_{ci} = f \frac{X_i}{Z_i}, \quad y_{ci} = f \frac{Y_i}{Z_i} \quad (2.5)$$

donde f representa la distancia focal de la cámara, (X_i, Y_i, Z_i) las coordenadas tridimensionales y (x_{ci}, y_{ci}) las coordenadas de la imagen de acuerdo con la figura 2.2.

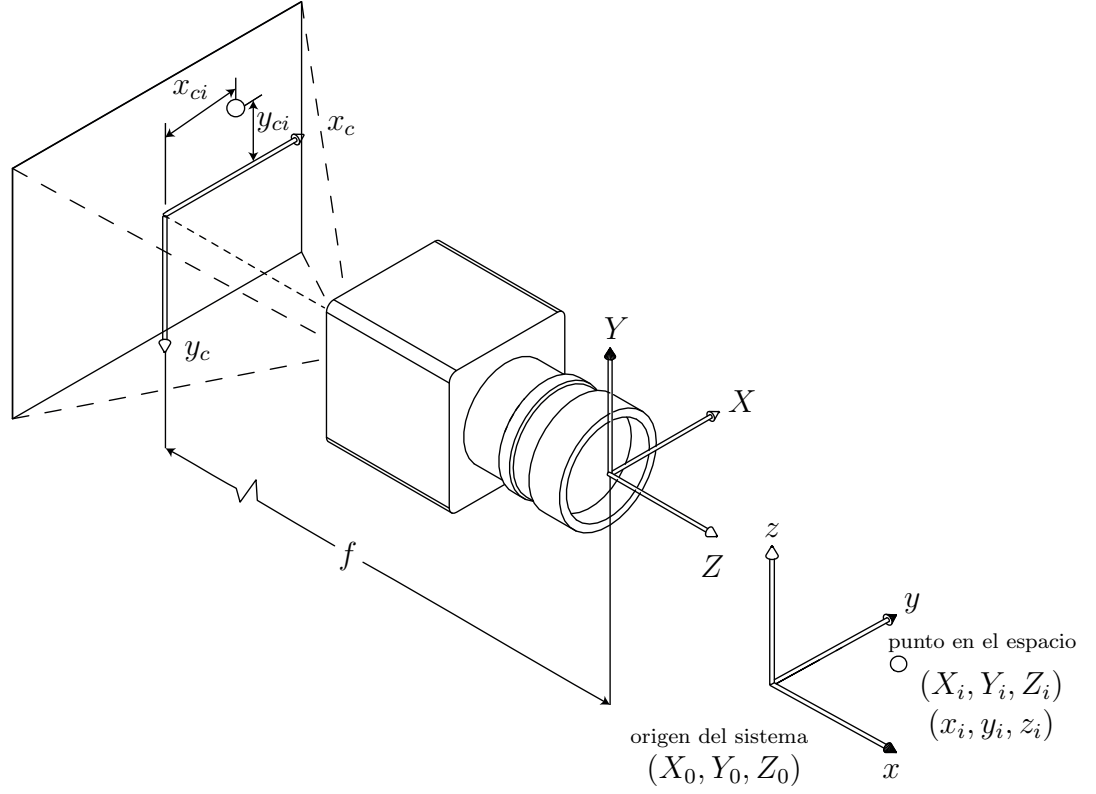


Figura 2.2: Sistema de referencia unido a la cámara

Se puede usar la siguiente expresión para relacionar los puntos de un sistema coordenado unido a la cámara referidos a otro sistema de coordenadas xyz :

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} = \begin{bmatrix} e_1^2 + e_2^2 - e_3^2 - e_4^2 & 2(e_2e_3 + e_1e_4) & 2(e_2e_4 - e_1e_3) & X_0 \\ 2(e_2e_3 - e_1e_4) & e_1^2 - e_2^2 + e_3^2 - e_4^2 & 2(e_3e_4 + e_1e_2) & Y_0 \\ 2(e_2e_4 + e_1e_3) & 2(e_3e_4 - e_1e_2) & e_1^2 - e_2^2 - e_3^2 + e_4^2 & Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \quad (2.6)$$

Donde $e_1 \dots e_4$ representan los cuatro parámetros de Euler, los cuales satisfacen la restricción $e_1^2 + e_2^2 + e_3^2 + e_4^2 = 1$; (X_0, Y_0, Z_0) son el origen tridimensional del sistema coordenado XYZ .

De acuerdo con las consideraciones antes mencionadas en la sección 1.6.2, se puede considerar el modelo de cámara ortográfico como una simplificación del modelo de perspectiva, esto es cuando Z_0 es mucho mayor que las cantidades de x_i , y_i , z_i , X_0 , y Y_0 , como se muestra en la figura 2.2. Reescribiendo la ecuación (2.5) obtenemos:

$$\begin{aligned} x_{ci} &= f \frac{X_i}{Z_i} = \frac{N_x}{D + \frac{1}{\epsilon}} = \Psi_x(\epsilon) \\ y_{ci} &= f \frac{Y_i}{Z_i} = \frac{N_y}{D + \frac{1}{\epsilon}} = \Psi_y(\epsilon) \end{aligned} \quad (2.7)$$

donde,

$$\begin{aligned} N_x &= f [(e_0^2 + e_1^2 - e_2^2 - e_3^2) x_i + 2(e_1 e_2 + e_0 e_3) y_i + 2(e_1 e_3 - e_0 e_2) z_i + X_0] \\ N_y &= f [2(e_1 e_2 - e_0 e_3) x_i + (e_0^2 - e_1^2 + e_2^2 - e_3^2) y_i + 2(e_2 e_3 + e_0 e_1) z_i + Y_0] \\ D &= [2(e_1 e_3 + e_0 e_2) x_i + 2(e_2 e_3 - e_0 e_1) y_i + (e_0^2 - e_1^2 - e_2^2 + e_3^2) z_i] \\ \epsilon &= \frac{1}{Z_0} \end{aligned}$$

Expandiendo el valor de Ψ_x usando una serie de Taylor para $\epsilon = 0$ se obtiene la siguiente expresión:

$$\begin{aligned} \Psi_x(\epsilon) &= \Psi_x|_{\epsilon=0} + \left. \frac{\partial \Psi_x}{\partial \epsilon} \right|_{\epsilon=0} \epsilon + \left. \frac{\partial^2 \Psi_x}{\partial \epsilon^2} \right|_{\epsilon=0} \frac{\epsilon^2}{2} + \dots \\ &= 0 + N_x \epsilon - N_x D \epsilon^2 + \dots \end{aligned}$$

El procedimiento anterior se puede repetir para y_c . Si se consideran únicamente la aproximación de primer orden, entonces $\frac{f}{Z_0}$ se puede combinar con el resto de los parámetros para formar un nuevo vector de parámetros $\mathbf{C} = [C_1, C_2, C_3, C_4, C_5, C_6]^T$. La aproximación resultante del modelo de cámara de orificio es:

$$x_{ci} = (C_1^2 + C_2^2 - C_3^2 - C_4^2) x_i + 2(C_2C_3 + C_1C_4) y_i + 2(C_2C_4 - C_1C_3) z_i + C_5$$

$$\equiv f_x(x, y, z; \mathbf{C})$$

$$y_{ci} = 2(C_2C_3 - C_1C_4) x_i + (C_1^2 - C_2^2 + C_3^2 - C_4^2) y_i + 2(C_3C_4 + C_1C_2) z_i + C_6$$

$$\equiv f_y(x, y, z; \mathbf{C})$$

Donde: $e_i \approx \frac{C_i}{\|C\|}$, $\|C\| = (C_1^2 + C_2^2 + C_3^2 + C_4^2)^{1/2}$, $i = 1, \dots, 4$

Estas ecuaciones definen al modelo de cámara ortográfica, mientras que los parámetros C_1, \dots, C_6 representan los parámetros de visión los cuales definen la relación entre la localización física de puntos tridimensionales y su correspondiente localización en el plano de la imagen.

Para el modelo de cámara ortográfico se puede observar que existe una relación de proporcionalidad que incluye un factor de escala y un factor de conversión de unidades usadas para la medición de la posición física de los puntos en un espacio tridimensional (X_i, Y_i) a las usadas para medir la posición de los puntos proyectados en el plano de la imagen (x_c, y_c) .

2.2.1. Parámetros de visión

Para determinar los seis parámetros de visión contenidos en el vector $\mathbf{C} = [C_1, C_2, C_3, C_4, C_5, C_6]^T$ se requiere minimizar la siguiente función de manera independiente para cada cámara participante:

$$J(\mathbf{C}) = \sum_{i=1}^m \{ [x_{ci} - f_x(x_i, y_i, z_i; \mathbf{C})]^2 + [y_{ci} - f_y(x_i, y_i, z_i; \mathbf{C})]^2 \} W_i \quad (2.8)$$

donde:

m : Número de mediciones realizadas.

(x_{ci}, y_{ci}) : Coordenadas de las señales visuales en el plano de la imagen.

(x_i, y_i, z_i) : Localización física de las marcas visuales obtenidas a partir del modelo cinemático del robot.

W_i : Peso relativo que se le da a cada medición.

Las expresiones $f_x(x_i, y_i, z_i, \mathbf{C})$ y $f_y(x_i, y_i, z_i, \mathbf{C})$ son obtenidas del modelo ortográfico de cámara y está definidas como:

$$f_x(x_i, y_i, z_i; \mathbf{C}) \equiv b_1(\mathbf{C})x_i + b_2(\mathbf{C})y_i + b_3(\mathbf{C})z_i + b_4(\mathbf{C}) \quad (2.9)$$

$$f_y(x_i, y_i, z_i; \mathbf{C}) \equiv b_5(\mathbf{C})x_i + b_6(\mathbf{C})y_i + b_7(\mathbf{C})z_i + b_8(\mathbf{C}) \quad (2.10)$$

donde b_1, \dots, b_8 agrupa a una expresión no lineal que depende de los parámetros de visión como sigue:

$$\begin{aligned} b_1(\mathbf{C}) &= C_1^2 + C_2^2 - C_3^2 - C_4^2 & b_5(\mathbf{C}) &= 2(C_2C_3 - C_1C_4) \\ b_2(\mathbf{C}) &= 2(C_2C_3 + C_1C_4) & b_6(\mathbf{C}) &= C_1^2 - C_2^2 + C_3^2 - C_4^2 \\ b_3(\mathbf{C}) &= 2(C_2C_4 - C_1C_3) & b_7(\mathbf{C}) &= 2(C_3C_4 - C_1C_2) \\ b_4(\mathbf{C}) &= C_5 & b_8(\mathbf{C}) &= C_6 \end{aligned} \quad (2.11)$$

Una definición de $J(\mathbf{C})$ en forma matricial es:

$$J(\mathbf{C}) = \Delta \mathbf{Y}^T \mathbf{W} \Delta \mathbf{Y} \quad (2.12)$$

donde el vector de residuos $\Delta \mathbf{Y}$ esta definida por:

$$\Delta \mathbf{Y} = \begin{bmatrix} x_{c1} - f_x(x_1, y_1, z_1; \mathbf{C}) \\ y_{c1} - f_y(x_1, y_1, z_1; \mathbf{C}) \\ \vdots \\ x_{cm} - f_x(x_m, y_m, z_m; \mathbf{C}) \\ y_{cm} - f_y(x_m, y_m, z_m; \mathbf{C}) \end{bmatrix} \quad (2.13)$$

Debido a que los parámetros de visión son no lineales, el proceso de estimación es iterativo y se deriva a partir del procedimiento de estimación no lineal de corrección de mínimos cuadrados, éste requiere de un valor inicial para los parámetros de visión \mathbf{C}^c , subsecuentemente se aplica un incremento $\Delta \mathbf{C} = [\Delta C_1, \dots, \Delta C_6]^T$ al linealizar el vector $\Delta \mathbf{Y}$ usando una expansión en series de Taylor.

$$\Delta \mathbf{Y}^P \approx \Delta \mathbf{Y}^c - \mathbf{A} \Delta \mathbf{C} \quad (2.14)$$

Donde:

$\Delta \mathbf{Y}^P$: vector de predicción de los residuos.

$\Delta \mathbf{Y}^c$: vector de residuos actuales obtenidos al sustituir \mathbf{C}^c en la ecuación 2.13.

La matriz de derivadas parciales \mathbf{A} de $2m \times 6$ está representada por:

$$\mathbf{A} = \begin{bmatrix} \frac{\partial f_x(x_1, y_1, z_1; C)}{\partial C_1} & \dots & \frac{\partial f_x(x_1, y_1, z_1; C)}{\partial C_6} \\ \frac{\partial f_y(x_1, y_1, z_1; C)}{\partial C_1} & \dots & \frac{\partial f_y(x_1, y_1, z_1; C)}{\partial C_6} \\ \vdots & & \vdots \\ \frac{\partial f_x(x_m, y_m, z_m; C)}{\partial C_1} & \dots & \frac{\partial f_x(x_m, y_m, z_m; C)}{\partial C_6} \\ \frac{\partial f_y(x_m, y_m, z_m; C)}{\partial C_1} & \dots & \frac{\partial f_y(x_m, y_m, z_m; C)}{\partial C_6} \end{bmatrix} \quad (2.15)$$

La corrección $\Delta \mathbf{C}$ es obtenida al minimizar:

$$J'(\Delta \mathbf{C}) = \Delta \mathbf{Y}^{pT} \mathbf{W} \Delta \mathbf{Y}^p \quad (2.16)$$

La condición necesaria para la minimización de la función anterior de acuerdo al

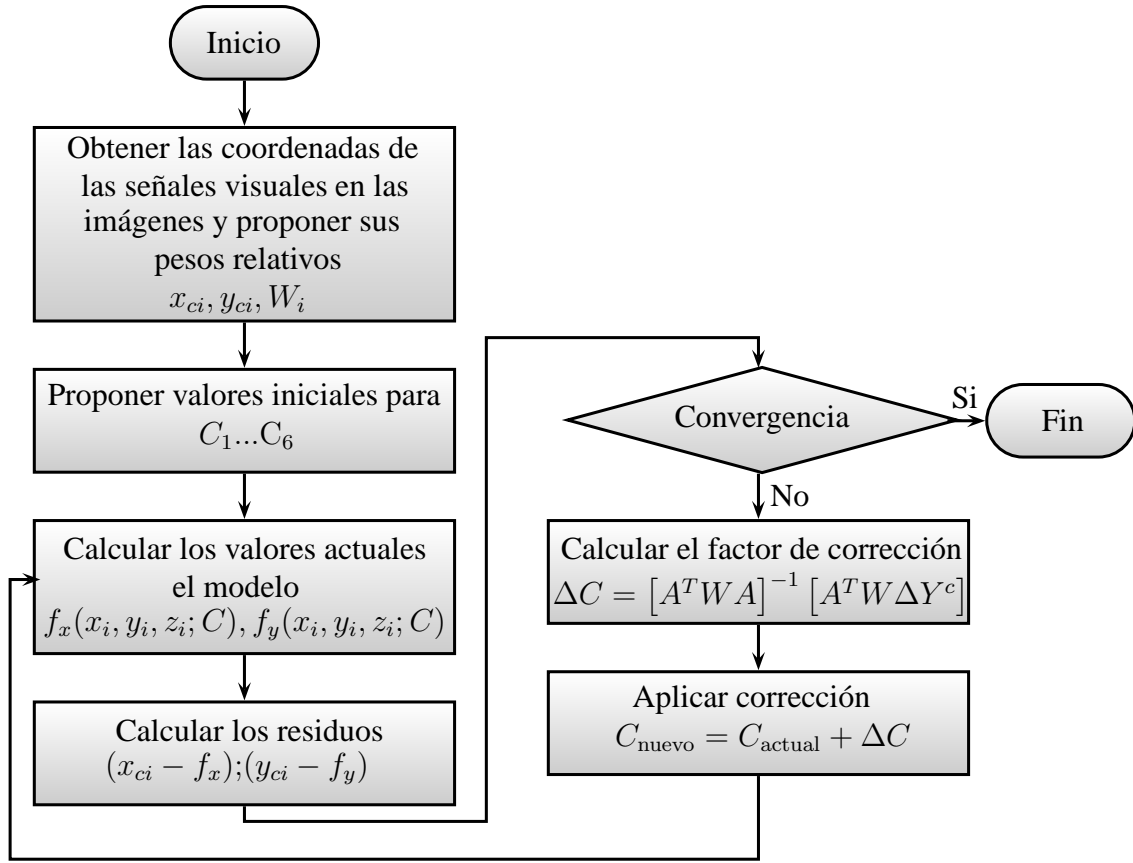


Figura 2.3: Diagrama de flujo del cálculo de los parámetros de visión

procedimiento de mínimos cuadrados y la introducción de la ecuación 2.14, lleva a la siguiente expresión para resolver ΔC :

$$\Delta C = [A^T W A]^{-1} [A^T W \Delta Y^c] \quad (2.17)$$

en la que se supone que la matriz $[A^T W A]$ no es singular. Y donde C se calcula iterativamente y finaliza hasta que exista convergencia (fig. 2.3) mediante:

$$C_{nuevo} = C_{actual} + \Delta C \quad (2.18)$$

2.2.2. Flattening

El procedimiento de *flattening* es una medida para incrementar la precisión al método mostrado anteriormente, éste cuenta con las ventajas de ser numéricamente estable y fácil de programar, sin embargo, debido a que se basa en el modelo ortográfico de cámara se ve limitado a un espacio de trabajo reducido por el efecto de perspectiva. El procedimiento consiste en modificar las muestras obtenidas en el plano de la imagen, de tal manera que sean más consistentes con nuestro modelo de estimación.

Considerando el sistema coordenado unido a la cámara, como el que muestra la figura 2.2, en el que el origen de este sistema coordenado está localizado en el punto focal de la cámara, con esto se pueden ubicar varios puntos en el espacio físico en coordenadas (X_i, Y_i, Z_i) y sus correspondientes en espacio cámara (x_{ci}, y_{ci}) . A partir de las muestras detectadas originalmente (x_{ci}, y_{ci}) se obtienen muestras modificadas (x_{fi}, y_{fi}) mediante:

$$x_{fi} = x_{ci} \frac{Z_i}{Z_r} \quad ; \quad y_{fi} = y_{ci} \frac{Z_i}{Z_r} \quad (2.19)$$

De la expresión (2.6) se obtiene:

$$Z_r = 2(e_2 e_4 + e_1 e_3) x_i + 2(e_3 e_4 + e_1 e_2) y_i + 2(e_1^2 - e_2^2 - e_3^2 + e_4^2) z_i + Z_0 \quad (2.20)$$

Donde Z_0 representa la distancia desde el origen del sistema coordenado unido a la cámara al origen del sistema coordenado desde el que se mide la posición física (x_i, y_i, z_i) , medida a lo largo del eje óptico \mathbf{Z} de la cámara. Mientras que Z_r representa la coordenada a lo largo del eje \mathbf{Z} de una marca visual cercana a la posición física donde finalizará la maniobra, la cual será determinada cuando se tenga disponible una buena estimación de los parámetros de visión para cada cámara.

2.3. Método de manipulación de espacio de cámara con modelo lineal de cámara (LCM-CSM)

Como se mencionó anteriormente sobre el método de manipulación de espacio de cámara se basa en la suposición de un modelo de cámara ortográfico, existe una modificación a este método que se fundamenta en un modelo lineal de cámara, [26]. El modelo de cámara de orificio es una transformación homogénea que existe entre las coordenadas tridimensionales de un punto en el espacio \mathbf{X} y su proyección en el plano de una imagen \mathbf{x} , [27], [16], [29], [28], [31].

2.3.1. Matriz de proyección

Si las coordenadas espaciales y de la imagen de los puntos son representadas mediante vectores aumentados, entonces la proyección central puede ser representada como un mapeo lineal entre las coordenadas homogéneas de dichos sistemas coordenados. Entonces, la ecuación 1.9 puede ser representada de la siguiente manera:

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (2.21)$$

Ésta expresión puede ser expresada de manera simplificada como $\text{diag}(f, f, 1) [I | 0]$ donde: $\text{diag}(f, f, 1)$ es una matriz diagonal y $[I | 0]$ representa una matriz dividida en un bloque de 3×3 (matriz identidad) y un vector columna, en este caso con todos los elementos cero.

Denominando \mathbf{X} a las coordenadas del mundo representadas por el vector $(X, Y, Z, 1)^T$, \mathbf{x} a los puntos en la imagen representadas por un vector de 3 elementos y P a la matriz de 3×4 elementos conocida como *matriz de proyección de cámara*.

Lo cual se puede expresar en forma compacta como:

$$\mathbf{x} = P\mathbf{X} \quad (2.22)$$

Lo anterior define la matriz para el modelo de cámara de orificio con proyección central siendo:

$$P = \text{diag}(f, f, 1) [I | 0] \quad (2.23)$$

Asumiendo que se tiene un número determinado de correspondencias de puntos $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ entre el sistema coordinado del mundo y la imagen. A partir de esos puntos es posible calcular la matriz de transformación homogénea P entre dichos sistemas, tal que $\mathbf{x}_i = P\mathbf{X}_i$ para todo punto i . De esta igualdad se deriva la siguiente relación en forma matricial:

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} \quad (2.24)$$

de lo anterior podemos obtener que:

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11}X_i + p_{12}Y_i + p_{13}Z_i + p_{14} \\ p_{21}X_i + p_{22}Y_i + p_{23}Z_i + p_{24} \\ p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34} \end{bmatrix} \quad (2.25)$$

dividendo entre el último renglón

$$x_i = \frac{p_{11}X_i + p_{12}Y_i + p_{13}Z_i + p_{14}}{p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}} \quad y_i = \frac{p_{21}X_i + p_{22}Y_i + p_{23}Z_i + p_{24}}{p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}} \quad (2.26)$$

Reacomodando los términos se llega a las siguientes expresiones:

$$x_i (p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}) = p_{11}X_i + p_{12}Y_i + p_{13}Z_i + p_{14}$$

$$y_i (p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}) = p_{21}X_i + p_{22}Y_i + p_{23}Z_i + p_{24}$$

Las cuales en forma matricial pueden ser representadas como un sistema lineal de la siguiente manera:

$$\begin{bmatrix} X & Y & Z & 1 & 0 & 0 & 0 & 0 & -x_iX_i & -x_iY_i & -x_iZ_i & -x_i \\ 0 & 0 & 0 & 0 & X & Y & Z & 1 & -y_iX_i & -y_iY_i & -y_iZ_i & -y_i \end{bmatrix} P = 0 \quad (2.27)$$

Donde P es un vector de 12 elementos de la siguiente forma:

$$P = [p_{11}, p_{12}, p_{13}, p_{14}, p_{21}, p_{22}, p_{23}, p_{24}, p_{31}, p_{32}, p_{33}, p_{34}]^T$$

Se hace la consideración de que el último elemento del vector P representa un factor de escala, el cual puede ser elegido como $p_{34} = 1$ [17], por lo que el siguiente sistema lineal es válido:

$$\begin{bmatrix} X & Y & Z & 1 & 0 & 0 & 0 & 0 & -x_iX_i & -x_iY_i & -x_iZ_i \\ 0 & 0 & 0 & 0 & X & Y & Z & 1 & -y_iX_i & -y_iY_i & -y_iZ_i \end{bmatrix} P = \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (2.28)$$

Se utiliza el método de mínimos cuadrados en el que se encuentra un vector P tal que minimice $\|AP - b\|$. La solución lineal entonces estará dada por $P = [A^T A]^{-1} A^T b$, donde la matriz A y el vector b están definidos como:

$$A = \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -x_1X_1 & -x_1Y_1 & -x_1Z_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -y_1X_1 & -y_1Y_1 & -y_1Z_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -x_iX_i & -x_iY_i & -x_iZ_i \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -y_iX_i & -y_iY_i & -y_iZ_i \end{bmatrix} \quad (2.29)$$

$$b = \begin{bmatrix} x_1 \\ y_1 \\ \vdots \\ x_i \\ y_i \end{bmatrix} \quad (2.30)$$

Estimación óptima de la matriz de proyección P

El método de estimación óptima de los parámetros P se obtiene al incorporar el mayor número de muestras al sistema, idealmente un número infinito. Ya que un número infinito de muestras no es posible tenerlos en la práctica, se buscó establecer un número lo suficientemente grande, en este caso 100, y para ello es que se considera en este trabajo la realización de 2 estimaciones, uno con 20 muestras y otro con 100 muestras y realizar una comparativa de los resultados, los cuales se presentan en la sección 4.3.2.

Para la estimación de la matriz de proyección P , se requieren las coordenadas en el plano de imagen (x_c, y_c) de las marcas visuales, además de su correspondiente vector de posición (X, Y, Z) referido al marco de referencia de la base del robot a través del modelo cinemático. Las coordenadas en espacio de cámara de las marcas visuales se obtienen a partir de un proceso, en el que las cámaras digitales adquieren imágenes al momento que se enciende y se apaga un led colocado en el efector final del robot. Éstas 2 se sustraen una de la otra obteniendo una nueva imagen en la que la marca visual es más notoria, para facilitar la obtención de las coordenadas se realiza un tratamiento digital de la imagen utilizando filtros de Canny, dilatación y de detección de bordes, ver sección 3.1.1.

Con la ubicación del centroide de la marca visual \mathbf{x}_i se procede a almacenar las

coordenadas (x_c, y_c) y su correspondiente \mathbf{X}_i del vector de posición (X, Y, Z) , en los que para cada \mathbf{x}_i se posiciona y orienta el robot manipulador en una serie de poses predefinidas (*preplan*), una vez realizado esto se obtiene la solución para el vector P mediante el método de mínimos cuadrados para $P = [A^T A]^{-1} A^T b$, en donde A y b fueron definidos en las ecuaciones (2.29) y (2.30).

Una vez que se obtiene la matriz de proyección P , ésta permite estimar el mapeo de un punto en el espacio 3D y su correspondiente en la imagen. Este punto debe ser observado por más de una cámara ya que para resolver para un punto en el espacio con coordenadas (X, Y, Z) se requiere conocer por lo menos de $1\frac{1}{2}$ par de coordenadas en espacio de cámara (x_c, y_c) .

Para la estimación del punto 3D mediante el método de minimización de los mínimos cuadrados se tiene que:

$$X = [A^T A]^{-1} A^T b \quad (2.31)$$

donde:

$$A = \begin{bmatrix} p_{11}^1 - x_{ci}^1 p_{31}^1 & p_{12}^1 - x_{ci}^1 p_{32}^1 & p_{13}^1 - x_{ci}^1 p_{33}^1 \\ p_{11}^1 - y_{ci}^1 p_{31}^1 & p_{12}^1 - y_{ci}^1 p_{32}^1 & p_{13}^1 - y_{ci}^1 p_{33}^1 \\ \vdots & \vdots & \vdots \\ p_{11}^n - x_{ci}^n p_{31}^n & p_{12}^n - x_{ci}^n p_{32}^n & p_{13}^n - x_{ci}^n p_{33}^n \\ p_{11}^n - y_{ci}^n p_{31}^n & p_{12}^n - y_{ci}^n p_{32}^n & p_{13}^n - y_{ci}^n p_{33}^n \end{bmatrix} \quad (2.32)$$

$$b = \begin{bmatrix} x_{ci}^1 - p_{14}^1 \\ y_{ci}^1 - p_{24}^1 \\ \vdots \\ x_{ci}^n - p_{14}^n \\ y_{ci}^n - p_{24}^n \end{bmatrix} \quad (2.33)$$

2.4. Determinación de la configuración interna del manipulador

Para que el robot posicione y oriente su efector final de una posición inicial hacia una posición objetivo se requiere determinar la configuración de los ángulos de las articulaciones que permitan realizar la maniobra.

Existe un caso particular en el que es posible desacoplar el procedimiento del cálculo del posicionamiento del manipulador en dos etapas, las cuales consistirían en los cálculos de la posición y orientación, dicho proceso se conoce como desacoplamiento cinemático.

Este desacoplamiento es posible cuando los ejes de las tres últimas articulaciones del robot se intersectan en un punto, esto es conocido como “*muñeca de configuración esférica*” (fig 2.4).

Cuando esto es válido, es posible relacionar las tres primeras articulaciones $(\theta_1, \theta_2, \theta_3)$ con la posición del manipulador y las tres últimas $(\theta_4, \theta_5, \theta_6)$ con la orientación del efector final, ambas etapas pueden ser calculadas de manera independiente, lo cual reduce en complejidad el cálculo de la cinemática inversa.

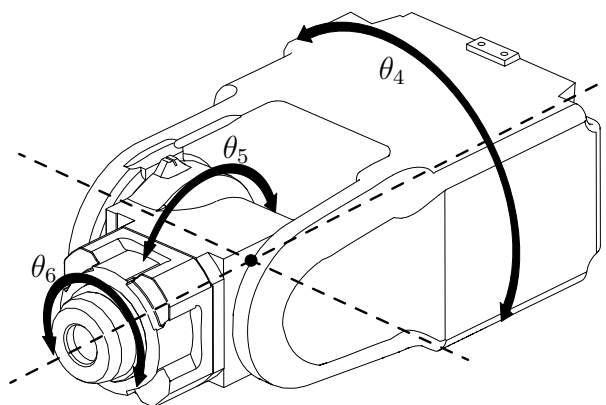


Figura 2.4: Muñeca del robot con configuración esférica

El proceso de estimación de la configuración del robot utilizado en este trabajo se realizó en dos etapas estas son: el posicionamiento y la orientación del efector final del robot. La primera etapa se determina mediante el uso de las tres primeras articulaciones para el posicionamiento del efector final del robot, mientras que en la segunda etapa se determina la orientación del mismo por medio de las tres restantes articulaciones (las cuales corresponden a las de la muñeca de configuración esférica como se muestra en la figura 2.4).

2.4.1. Posicionamiento del robot con modelo de cámara ortográfico

La configuración interna del manipulador requerida para obtener la posición deseada usando un punto objetivo se realiza al minimizar la siguiente función γ considerando las coordenadas angulares $(\theta_1, \theta_2, \theta_3)$, que se incluyen dentro del vector Θ :

$$\begin{aligned} \gamma(\Theta) = & \sum_{i=1}^{n_c} [x_{ci} - f_x(r_x(\Theta), r_y(\Theta), r_z(\Theta); \mathbf{C}^i)]^2 \\ & + [y_{ci} - f_y(r_x(\Theta), r_y(\Theta), r_z(\Theta); \mathbf{C}^i)]^2 \end{aligned} \quad (2.34)$$

Donde n_c es el número de cámaras participantes, (x_{ci}, y_{ci}) corresponden a las coordenadas en las imágenes en la $i^{\text{ésima}}$ cámara y su correspondiente localización física como $(r_x(\Theta), r_y(\Theta), r_z(\Theta))$, la cual depende de la configuración angular del robot contenida en Θ . Las funciones $f_x(r_x, r_y, r_z, \mathbf{C}^i)$ y $f_y(r_x, r_y, r_z, \mathbf{C}^i)$ son obtenidas del modelo ortográfico de cámara como se describe en las ecs. (2.9) y (2.10).

Para obtener la minimización de la función γ se realiza mediante:

$$\Delta\Theta = [\mathbf{A}^T \mathbf{A}]^{-1} [\mathbf{A}^T \Delta\mathbf{Y}] \quad (2.35)$$

Donde las matrices \mathbf{A} y $\Delta\mathbf{Y}$ están definidas como:

$$A = \begin{bmatrix} \left(\frac{\partial H}{\partial \theta_1}\right)^T \begin{bmatrix} b_1^1 \\ \vdots \\ b_4^1 \end{bmatrix} & \left(\frac{\partial H}{\partial \theta_2}\right)^T \begin{bmatrix} b_1^1 \\ \vdots \\ b_4^1 \end{bmatrix} & \left(\frac{\partial H}{\partial \theta_3}\right)^T \begin{bmatrix} b_1^1 \\ \vdots \\ b_4^1 \end{bmatrix} \\ \left(\frac{\partial H}{\partial \theta_1}\right)^T \begin{bmatrix} b_5^1 \\ \vdots \\ b_8^1 \end{bmatrix} & \left(\frac{\partial H}{\partial \theta_2}\right)^T \begin{bmatrix} b_5^1 \\ \vdots \\ b_8^1 \end{bmatrix} & \left(\frac{\partial H}{\partial \theta_3}\right)^T \begin{bmatrix} b_5^1 \\ \vdots \\ b_8^1 \end{bmatrix} \\ \vdots & \vdots & \vdots \\ \left(\frac{\partial H}{\partial \theta_1}\right)^T \begin{bmatrix} b_1^m \\ \vdots \\ b_4^m \end{bmatrix} & \left(\frac{\partial H}{\partial \theta_2}\right)^T \begin{bmatrix} b_1^m \\ \vdots \\ b_4^m \end{bmatrix} & \left(\frac{\partial H}{\partial \theta_3}\right)^T \begin{bmatrix} b_1^m \\ \vdots \\ b_4^m \end{bmatrix} \\ \left(\frac{\partial H}{\partial \theta_1}\right)^T \begin{bmatrix} b_5^m \\ \vdots \\ b_8^m \end{bmatrix} & \left(\frac{\partial H}{\partial \theta_2}\right)^T \begin{bmatrix} b_5^m \\ \vdots \\ b_8^m \end{bmatrix} & \left(\frac{\partial H}{\partial \theta_3}\right)^T \begin{bmatrix} b_5^m \\ \vdots \\ b_8^m \end{bmatrix} \end{bmatrix} \quad (2.36)$$

donde:

H : matriz de la cinemática directa del robot, definido en la sección 2.1.3.

$b_1 \dots b_8$: expresiones que dependen de los parámetros de visión para las m cámaras participantes, definidos en la ecuación (2.11).

$$\Delta Y = \begin{bmatrix} x_c^1 - [b_1^1 r_x(\Theta) + b_2^1 r_y(\Theta) + b_3^1 r_z(\Theta) + b_4^1] \\ y_c^1 - [b_5^1 r_x(\Theta) + b_6^1 r_y(\Theta) + b_7^1 r_z(\Theta) + b_8^1] \\ \vdots \\ x_c^m - [b_1^m r_x(\Theta) + b_2^m r_y(\Theta) + b_3^m r_z(\Theta) + b_4^m] \\ y_c^m - [b_5^m r_x(\Theta) + b_6^m r_y(\Theta) + b_7^m r_z(\Theta) + b_8^m] \end{bmatrix} \quad (2.37)$$

donde $(r_x(\Theta), r_y(\Theta), r_z(\Theta))$ son las coordenadas físicas en (X, Y, Z) de la ubicación del robot de acuerdo a la configuración angular actual.

$\Delta\Theta$ deja de calcularse en el momento en el que se minimiza la función:

$$J = \Delta Y^T \Delta Y \quad (2.38)$$

2.4.2. Posicionamiento del robot con modelo de cámara lineal

Tal como se mencionó anteriormente, para obtener la posición deseada usando un punto objetivo se realiza mediante la minimización de la función γ en la ecuación (2.34), la cual depende de los seis parámetros de visión contenidos en $\mathbf{C} = [C_1, \dots, C_6]$, en esta ocasión existe la consideración de que dependa de los once coeficientes de $P = [p_{11}, \dots, p_{33}, 1]$. Por lo que podemos reescribir la ecuación como:

$$\begin{aligned} \gamma(\Theta) = & \sum_{i=1}^{n_c} [x_{ci} - p_x(r_x(\Theta), r_y(\Theta), r_z(\Theta); P^i)]^2 \\ & + [y_{ci} - p_y(r_x(\Theta), r_y(\Theta), r_z(\Theta); P^i)]^2 \end{aligned} \quad (2.39)$$

donde:

n_c : número de cámaras participantes.

(x_{ci}, y_{ci}) : coordenadas en las imágenes del punto objetivo

$(r_x(\Theta), r_y(\Theta), r_z(\Theta))$: coordenada en el espacio del punto objetivo de acuerdo con la configuración angular del robot.

p_x, p_y : estimaciones actuales de la matriz de proyección P .

Para obtener la minimización de la función γ se realiza mediante el método de mínimos cuadrados descrito por la ecuación 2.35. Donde las matrices \mathbf{A} y $\Delta\mathbf{Y}$ están definidas como:

$$A = \begin{bmatrix} \left(\frac{\partial H}{\partial \theta_1} \right)^T \begin{bmatrix} p_{11}^1 \\ \vdots \\ p_{14}^1 \end{bmatrix} - x_c^1 \left\{ \left(\frac{\partial H}{\partial \theta_1} \right)^T \begin{bmatrix} p_{31}^1 \\ \vdots \\ 1 \end{bmatrix} \right\} & \cdots & \left(\frac{\partial H}{\partial \theta_3} \right)^T \begin{bmatrix} p_{11}^1 \\ \vdots \\ p_{14}^1 \end{bmatrix} - x_c^1 \left\{ \left(\frac{\partial H}{\partial \theta_3} \right)^T \begin{bmatrix} p_{31}^1 \\ \vdots \\ 1 \end{bmatrix} \right\} \\ \left(\frac{\partial H}{\partial \theta_1} \right)^T \begin{bmatrix} p_{21}^1 \\ \vdots \\ p_{24}^1 \end{bmatrix} - y_c^1 \left\{ \left(\frac{\partial H}{\partial \theta_1} \right)^T \begin{bmatrix} p_{31}^1 \\ \vdots \\ 1 \end{bmatrix} \right\} & \cdots & \left(\frac{\partial H}{\partial \theta_3} \right)^T \begin{bmatrix} p_{21}^1 \\ \vdots \\ p_{24}^1 \end{bmatrix} - y_c^1 \left\{ \left(\frac{\partial H}{\partial \theta_3} \right)^T \begin{bmatrix} p_{31}^1 \\ \vdots \\ 1 \end{bmatrix} \right\} \\ \vdots & \vdots & \vdots \\ \left(\frac{\partial H}{\partial \theta_1} \right)^T \begin{bmatrix} p_{11}^m \\ \vdots \\ p_{14}^m \end{bmatrix} - x_c^m \left\{ \left(\frac{\partial H}{\partial \theta_1} \right)^T \begin{bmatrix} p_{31}^m \\ \vdots \\ 1 \end{bmatrix} \right\} & \cdots & \left(\frac{\partial H}{\partial \theta_3} \right)^T \begin{bmatrix} p_{11}^m \\ \vdots \\ p_{14}^m \end{bmatrix} - x_c^m \left\{ \left(\frac{\partial H}{\partial \theta_3} \right)^T \begin{bmatrix} p_{31}^m \\ \vdots \\ 1 \end{bmatrix} \right\} \\ \left(\frac{\partial H}{\partial \theta_1} \right)^T \begin{bmatrix} p_{21}^m \\ \vdots \\ p_{24}^m \end{bmatrix} - y_c^m \left\{ \left(\frac{\partial H}{\partial \theta_1} \right)^T \begin{bmatrix} p_{31}^m \\ \vdots \\ 1 \end{bmatrix} \right\} & \cdots & \left(\frac{\partial H}{\partial \theta_3} \right)^T \begin{bmatrix} p_{21}^m \\ \vdots \\ p_{24}^m \end{bmatrix} - y_c^m \left\{ \left(\frac{\partial H}{\partial \theta_3} \right)^T \begin{bmatrix} p_{31}^m \\ \vdots \\ 1 \end{bmatrix} \right\} \end{bmatrix} \quad (2.40)$$

donde:

m : número de cámaras participantes.

H : matriz de la cinemática directa del robot, definido en la sección 2.1.3.

(x_{ci}, y_{ci}) : representan las coordenadas del punto objetivo en las imágenes para cada p^m elemento de la matriz de proyección P .

$$\Delta Y = \begin{bmatrix} x_c^1 [p_{31}^1 r_x(\Theta) + p_{32}^1 r_y(\Theta) + p_{33}^1 r_z(\Theta) + 1] - [p_{11}^1 r_x(\Theta) + p_{12}^1 r_y(\Theta) + p_{13}^1 r_z(\Theta) + p_{14}^1] \\ y_c^1 [p_{31}^1 r_x(\Theta) + p_{32}^1 r_y(\Theta) + p_{33}^1 r_z(\Theta) + 1] - [p_{21}^1 r_x(\Theta) + p_{22}^1 r_y(\Theta) + p_{23}^1 r_z(\Theta) + p_{24}^1] \\ \vdots \\ x_c^m [p_{31}^m r_x(\Theta) + p_{32}^m r_y(\Theta) + p_{33}^m r_z(\Theta) + 1] - [p_{11}^m r_x(\Theta) + p_{12}^m r_y(\Theta) + p_{13}^m r_z(\Theta) + p_{14}^m] \\ y_c^m [p_{31}^m r_x(\Theta) + p_{32}^m r_y(\Theta) + p_{33}^m r_z(\Theta) + 1] - [p_{21}^m r_x(\Theta) + p_{22}^m r_y(\Theta) + p_{23}^m r_z(\Theta) + p_{24}^m] \end{bmatrix} \quad (2.41)$$

donde $(r_x(\Theta), r_y(\Theta), r_z(\Theta))$ son las coordenadas físicas en (X, Y, Z) del punto objetivo, que depende del modelo cinemático directo del manipulador.

Al obtener el mínimo de la función γ de manera iterativa lo que se busca es encontrar una solución válida en la que las juntas del robot contenidas en el vector Θ converjan en la configuración de las articulaciones $\theta_1, \theta_2, \theta_3$ que permitan obtener la posición del robot en el punto objetivo, dejando como fijos los valores de las articulaciones $\theta_4, \theta_5, \theta_6$.

2.4.3. Orientación

Para la segunda etapa, esto es, el cálculo de la orientación para los dos distintos modelos de cámara, se requiere tener de un punto objetivo en común para las cámaras participantes el cual deberá hacer uso de un vector auxiliar perpendicular al plano de trabajo. Éste deberá obtenerse realizando un producto vectorial expresado por la siguiente expresión:

$$n = \frac{\overline{BC} \times \overline{BA}}{|\overline{BC}| |\overline{BA}|} \quad (2.42)$$

La orientación se obtiene mediante la solución de la siguiente expresión:

$$f(\theta_4, \theta_5) \cdot \hat{n} = -1 \quad (2.43)$$

Donde:

$f(\theta_4, \theta_5)$: vector de orientación del efector final del robot, el cual depende de las articulaciones 4 y 5.

\hat{n} : vector normal al plano de trabajo.

En la figura 2.5, se presenta un diagrama de flujo que describe los cálculos del proceso de posicionamiento del manipulador.

En el presente capítulo se presentaron los antecedentes teóricos de la robots manipuladores y los fundamentos para la obtención del modelo cinemático directo de los mismos, haciendo una descripción de las matrices de transformación que permiten obtener la configuración del robot Motoman. Se hace también una descripción del Método de Manipulación en Espacio de Cámara (CSM) tanto basado en el modelo ortográfico y corrección por flattening y con el modelo de cámara lineal. En el siguiente capítulo se describe una metodología fundamentada en LCM-CSM, mediante la cual se obtiene de manera automática la posición y orientación de piezas de trabajo con geometría regular y la cual permite detectar piezas transportadas en la banda de una celda de Manufactura Integrada por Computadora (CIM).

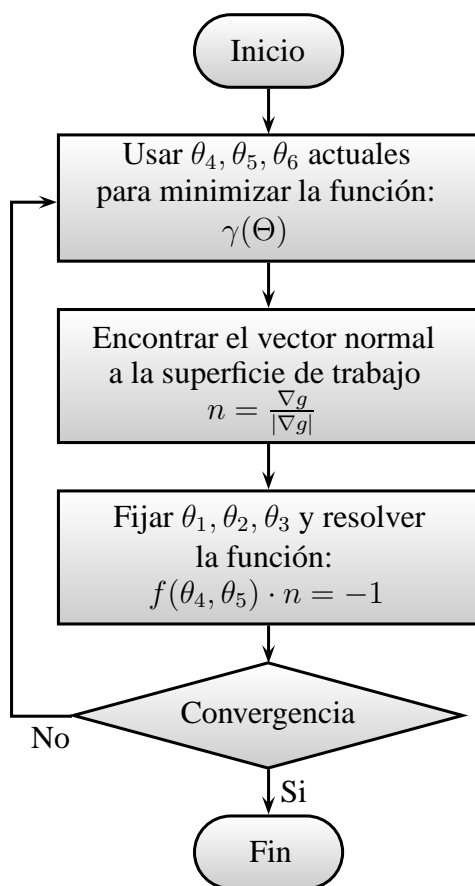


Figura 2.5: Diagrama de flujo del cálculo del posicionamiento del robot

Detección automática de piezas en una banda transportadora por visión computacional

Una vez definido los antecedentes teóricos de la robótica y la visión por computadora, se prosigue a explicar la metodología que se utilizó como base en la implementación de LCM-CSM realizada en el presente trabajo. Con esta metodología se determina la localización y la orientación de una pieza en una banda transportadora, referenciada al sistema coordenado del robot manipulador.

3.1. Implementación de técnicas de análisis y procesamiento de imágenes para visión computacional

Entre los objetivos que se planteó para este proyecto fue el proporcionar un entorno de desarrollo eficiente y fácil de utilizar. Esto se logró, desarrollando la programación en lenguaje C y C++ y aprovechando además las capacidades que proveen los procesadores multi núcleo y las librerías de *OpenCV* y *Numerical Recipes*.

El lenguaje de programación `C++` fue creado en los años 80 por Bjarne Stroustrup basado en el lenguaje `C`. Este es un lenguaje orientado a algoritmos y a objetos al que se le añadieron características y cualidades de las que carecía el lenguaje `C`. En la actualidad, el `C++` es un lenguaje versátil, potente y general [35]. Su éxito entre los programadores lo ha llevado a ocupar el primer puesto como herramienta de desarrollo de aplicaciones, como es el caso de las bases de datos, procesadores de texto, juegos y de la misma forma es muy usado en aplicaciones científicas (para experimentos informáticos, físicos, químicos, matemáticos, entre otros, parte de ellos conocidos como modelos y simuladores) e industriales (robótica, cibernética, sistemas de información).

Un punto importante para desarrollar eficientemente programas, es el uso de librerías que facilitan el tiempo de programación y que optimicen el procesamiento de cómputo. De entre las librerías especializadas en el campo de la visión por computadora se encuentra *OpenCV* (Open source Computer Vision library) [36] desarrollada originalmente por Intel® y bajo la publicación de licencia BSD que permite ser utilizada libremente para propósitos comerciales y de investigación con las condiciones en ella expresadas. *OpenCV* es una librería multiplataforma, existiendo en versiones para GNU/Linux, Mac OS X® y Windows®. Contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de visión, como reconocimiento de objetos, calibración de cámaras, visión estereoscópica y visión robótica.

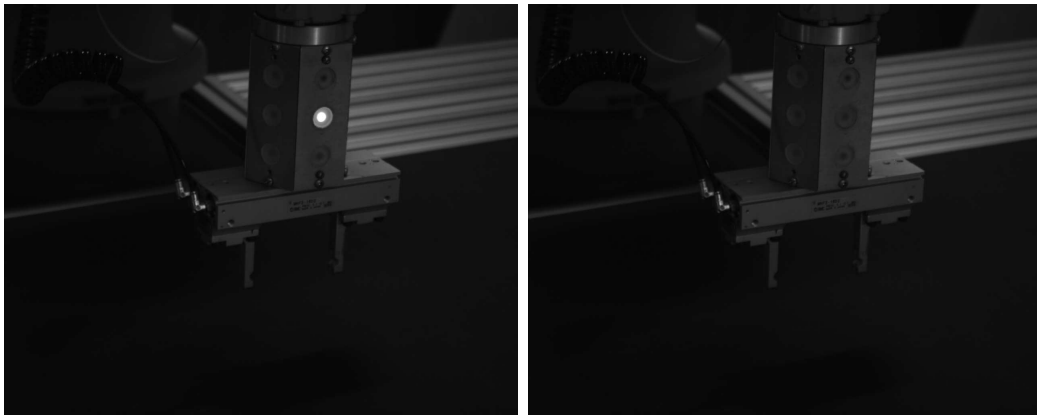
Otra librería muy utilizada en la bibliografía especializada es la que proporciona la literatura de “*Numerical Recipes in C*” [37] la cual contiene una amplia gama de algoritmos de análisis numérico y métodos matemáticos, con implementaciones que garanticen portabilidad y eficiencia en la ejecución.

Es por estas características que se decidió utilizar el lenguaje C++ y las librerías de *OpenCV* y *Numerical Recipes* para el desarrollo del presente trabajo.

3.1.1. Obtención de coordenadas de marcas visuales

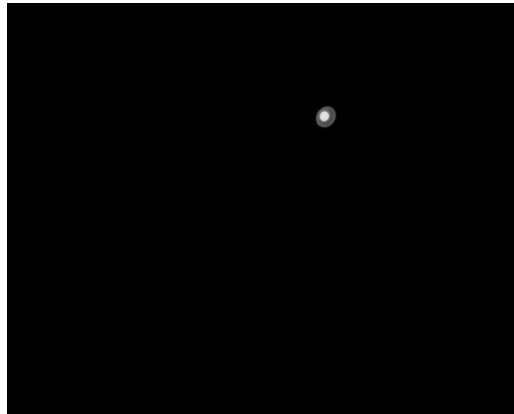
Para adquirir el centroide de las marcas visuales, es necesario encender y apagar un led en el efector final del robot que se ocupa como una marca visual distintiva. Se adquiere una imagen con el led prendido y otra con el led apagado. Estas 2 imágenes (o matrices) se sustraen para obtener una nueva imagen figura 3.1, haciendo más notoria la ubicación del led como la marca visual deseada.

Esta imagen es previamente tratada digitalmente, primeramente con un filtro basado en el algoritmo de Canny ,[14], [15], (sección 1.4) y utilizando la función *OpenCV: cvCanny*, de esta forma se realzan los bordes como se muestra en la figura 3.2 (a). Posteriormente para garantizar que existieran bordes cerrados en la imagen se aplicó un filtro morfológico conocido como dilatación (sección 1.4.1), función de *OpenCV cvDilate*, (figura 3.2 (b)) y por último por medio de la función *cvFindContours* se detectan y almacenan las coordenadas (x, y) pixel que conforman cada uno de los contornos cerrados en la imagen. Con esta información se obtiene la ubicación del centroide (x_c, y_c) . El resultado de la localización del centroide se puede observar en la figura 3.3, en donde se interpreta por medio de una cruz la posición del centroide de la marca visual localizado en la imagen, y en donde el contorno exterior localizado forma parte de los puntos en consideración para obtener este centroide .



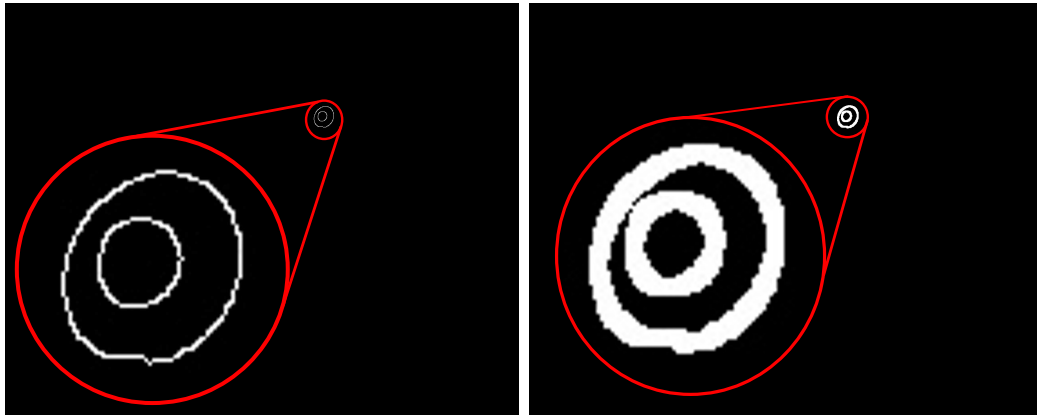
(a) Imagen con el led encendido

(b) Imagen con el led apagado



(c) Imagen resultado de la resta

Figura 3.1: Proceso de sustracción de imágenes



(a) Filtro de canny

(b) Filtro de dilatación

Figura 3.2: Resultado de la aplicación de los filtros de canny y dilatación

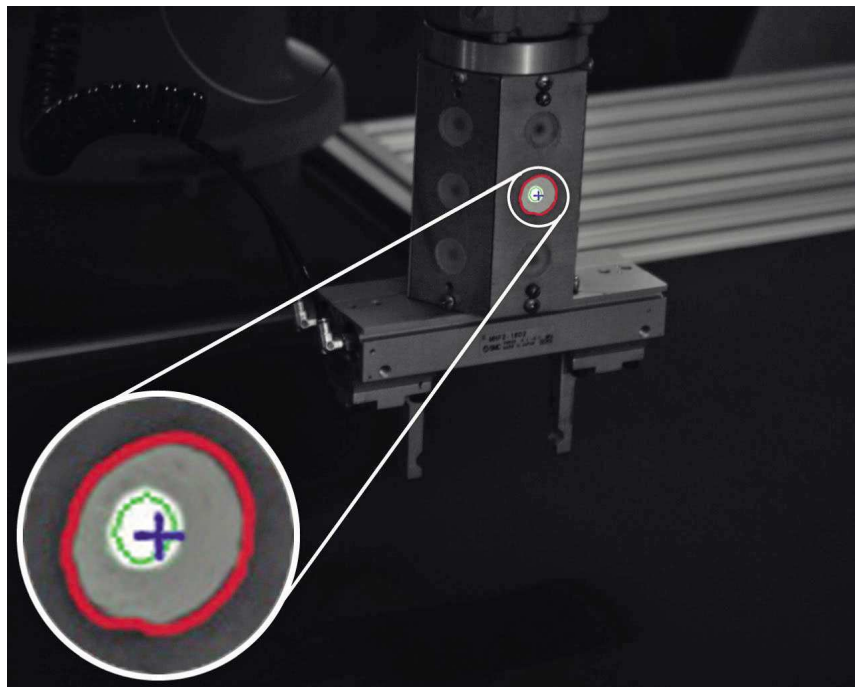


Figura 3.3: Resultado del algoritmo de localización del centroide de una marca visual del robot

3.2. Metodología para determinar posición y orientación de las piezas de trabajo

Uno de los problemas más frecuentes en los sistemas de visión computacional es el de determinar las formas y figuras en las imágenes, las cuales al ojo humano son de forma muy evidente pero que para una computadora requiere de varios procesamientos que vayan filtrando la información, de tal manera que se pueda determinar los contornos que describan la figura buscada.

Para facilitar la detección de los bordes de piezas de trabajo con geometría regular, se utilizó iluminación estructurada con sistemas de luz láser. En el caso particular del presente trabajo se optó por utilizar un patrón de luz láser que proyectara una línea. Viendo la forma como cambia este patrón sobre el objeto iluminado se puede determinar las características que describen la geometría del objeto buscado. Conociendo las coordenadas de los puntos que describen los bordes superiores de la pieza es posible estimar sus coordenadas tridimensionales y ajustar un plano que describa la superficie. Con esta información es entonces que se estima la configuración de las articulaciones del robot para poder orientar y posicionar el efector final, de tal forma que pueda tomar el objeto y estibararlo. A continuación se presentan los pasos de la metodología que se implementó en el presente trabajo para lograr lo anterior:

- a) **Estimación de la matriz de proyección.** En esta etapa se realiza la metodología que se mencionó en la sección 2.3.1, debe tomarse en cuenta que este paso se hace una sola vez, siempre y cuando no se hayan cambiado de posición las cámaras ni el robot. Es posible además estimar los parámetros de la matriz de proyección, de tal forma que se realice una actualización al momento que el robot se aproxime a la pieza de trabajo, lo cual es en lo que se basa la metodología de CSM, con lo que se espera obtener mejores resultados de aproximación al punto

objetivo.

- b) **Ajuste de las condiciones de cámara.** Esta es una interfaz gráfica de usuario (GUI) que monitorea en tiempo real las líneas discretizadas en las imágenes adquiridas por las cámaras de la pieza de trabajo y que corresponden a sus bordes, el propósito de ésta es la de encontrar la configuración adecuada que encuentre las líneas buscadas, es decir, que para cada imagen capturada en un instante se compensé la cantidad de luz que recibe por medio del ajuste del obturador de la lente. Esta interface permite especificar al usuario parámetros tales como valores de umbral para el ligado de bordes y para establecer el inicio del segmento de orillas, el valor de longitud de línea, la distancia máxima entre segmentos de línea y que pertenezcan a la misma línea, así como los límites superior e inferior de los ángulos θ_l de las líneas deseadas. Esta GUI se muestra en el apéndice A2. Una vez que se ajustan todos estos parámetros en la GUI de forma manual estos se graban en un archivo que es usado en la etapa del procesamiento de las imágenes.
- c) **Adquisición de imágenes.** Se realiza la adquisición de imágenes mientras el PTU desplaza de forma horizontal la línea proyectada por la unidad láser a lo largo de la pieza, de manera que todas las imágenes que son adquiridas se almacenan para posteriormente ser procesadas, figura 3.4.

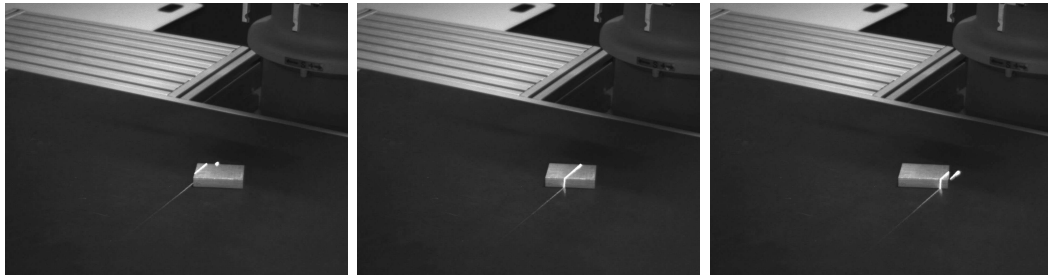


Figura 3.4: Captura de imágenes durante el desplazamiento de la proyección de la línea de luz láser sobre la pieza

- d) **Procesamiento de las imágenes.** Se procesan todas las imágenes adquiridas en el paso anterior de una en una, en las que se discretizan líneas según los parámetros de configuración que fueron ajustados con la GUI en la etapa de ajuste de las condiciones de cámara. Los datos que extrae de esta etapa son las coordenadas (x_c, y_c) pixel de los puntos de inicio y fin de la línea encontrada, los cuales corresponden con las esquinas la cara superior de la pieza, figuras 3.5 y 3.6.

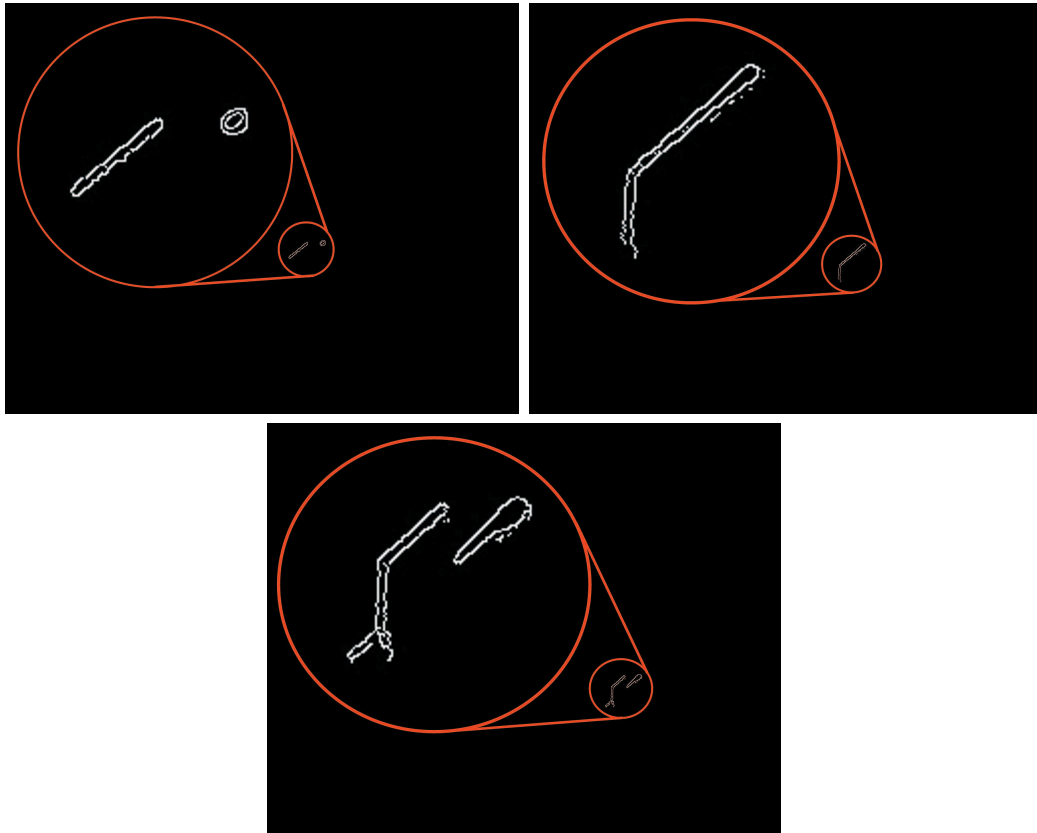


Figura 3.5: Procesamiento de imágenes mediante el algoritmo de Canny

- e) **Reconstrucción 3D de puntos.** Una vez que se tienen las coordenadas de los puntos de inicio y fin de línea para cada imagen en las diferentes instancias

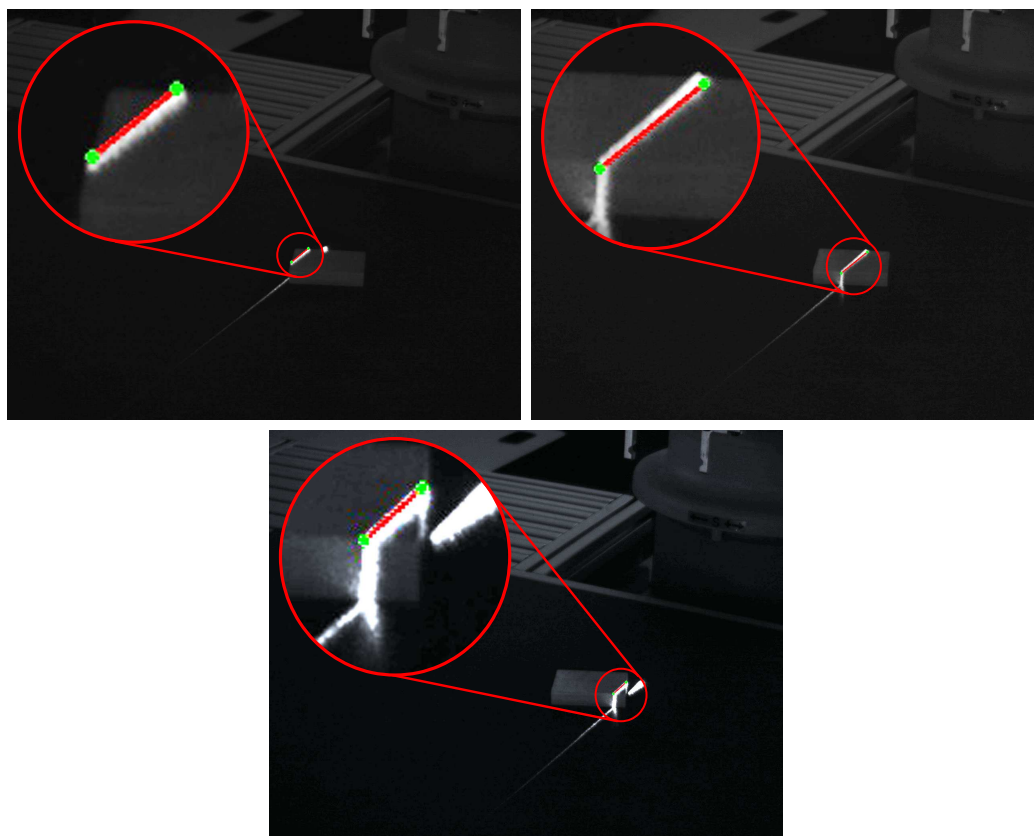


Figura 3.6: Resultado del procesamiento de imágenes mediante el algoritmo de detección de líneas

adquiridas, se procede a determinar las coordenadas espaciales correspondientes por cada par de imágenes $\mathbf{X}_i \longleftrightarrow \mathbf{x}_i$ por medio de la solución de $\mathbf{x} = P\mathbf{X}$.

- f) **Obtención de la superficie plana de la pieza.** En esta última etapa se cuenta con una nube de puntos en el espacio y lo que se busca es encontrar un plano rectangular que mejor se ajuste a estos datos. Bajo la suposición de que la superficie de la mesa de trabajo se encuentra paralela al plano XY de referencia (robot) y por lo tanto la cara superior de la pieza también es paralela, se establecen los parámetros de la ecuación del plano:

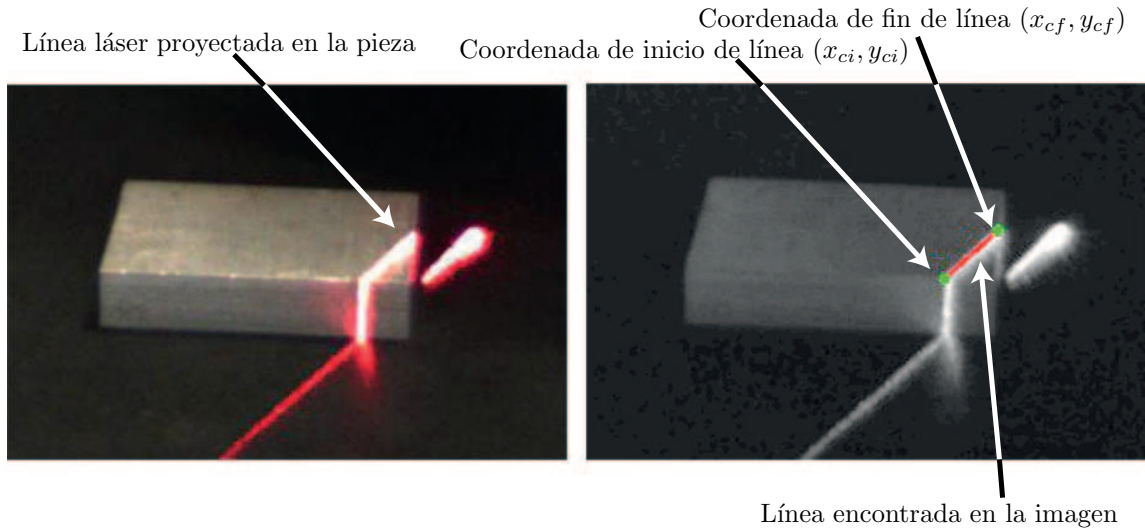


Figura 3.7: Procesamiento de imagen. Adquisición de línea láser y coordenada de los puntos de inicio y fin de línea.

$$Ax + By + Cz + D = 0 \quad (3.1)$$

donde la condición necesaria para determinar un plano que sea paralelo al plano XY , es que los coeficientes A , B y D sean iguales con 0 y de ello que se establezca $z = C$, por lo tanto para determinar el coeficiente C de un plano que mejor se ajuste a los puntos, se promedia la coordenada Z de cada uno de los puntos del inciso e). Una vez estimado el plano en el que se encuentra la superficie de la pieza, lo siguiente es obtener el tamaño y la orientación de la superficie de la pieza que mejor se adapte a las coordenadas (x, y) de los puntos, lo cual se obtiene mediante la función de *OpenCV* `cvMinAreaRect2`, esta función determina un rectángulo delimitado por el área mínima del conjunto de puntos bidimensionales contenidos en una envolvente convexa o cierre convexo (convex hull), [38], [39]. Con esta función se obtienen las dimensiones del área limitada por los bordes que conforman un superficie rectangular y el ángulo que existe entre los bordes y el

eje horizontal del sistema de referencia del robot, como se muestra en la figura 3.8.

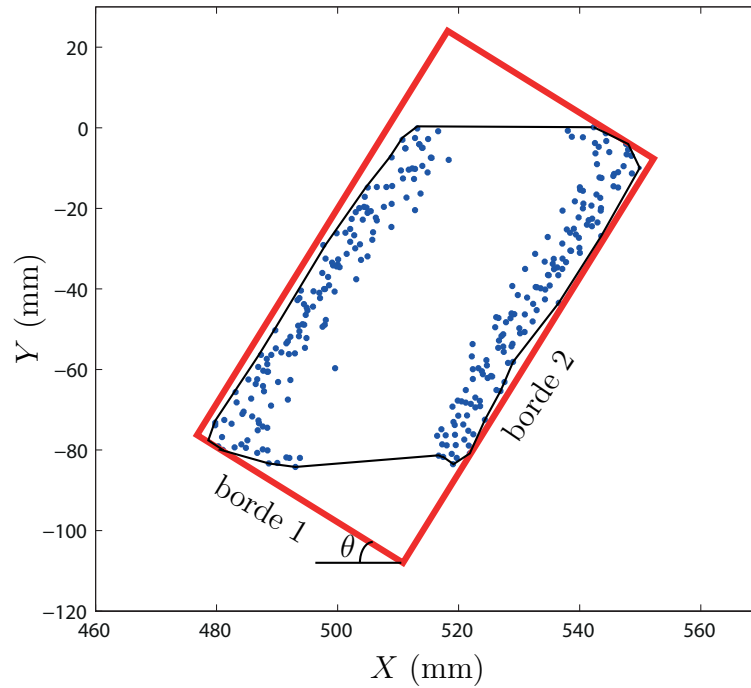


Figura 3.8: Resultado de la aplicación de la función de OpenCV `cvMinAreaRect2`

Los pasos descritos en esta metodología se presentan a manera de diagrama de flujo en la figura 3.11.

En el presente capítulo se ha descrito la metodología utilizada para determinar la posición y orientación con respecto al robot manipulador de una pieza de geometría regular. Para determinar la ubicación de la pieza es necesario obtener la información de la cara superior, esto se obtuvo por medio de la adquisición de imágenes consecutivas al momento que se desplaza la proyección de la línea de luz láser sobre la pieza (figura 3.4), posteriormente estas imágenes fueron procesadas para obtener la línea que mejor se ajuste a la proyectada sobre la superficie de la pieza y con esto obtener los puntos de inicio y fin de la línea en espacio de cámara (figura 3.6), los

cuales corresponden con las esquinas de la cara superior de la pieza. Estos puntos en las imágenes son estimados tridimensionalmente para obtener una nube de puntos en el espacio 3D. A estos puntos se ajusta un plano paralelo al plano XY (figuras 3.10 y 3.9), y además se ajustan los puntos a este plano un rectángulo, a partir de este ajuste se determinan las: longitudes de los lados, inclinación con respecto al eje horizontal y las coordenadas del centro del rectángulo. Una descripción general del algoritmo de detección de superficies planas en piezas se presenta en la figura 3.11. En el siguiente capítulo se describe el sistema de visión empleado para validar de manera experimental la metodología presentada en este capítulo.

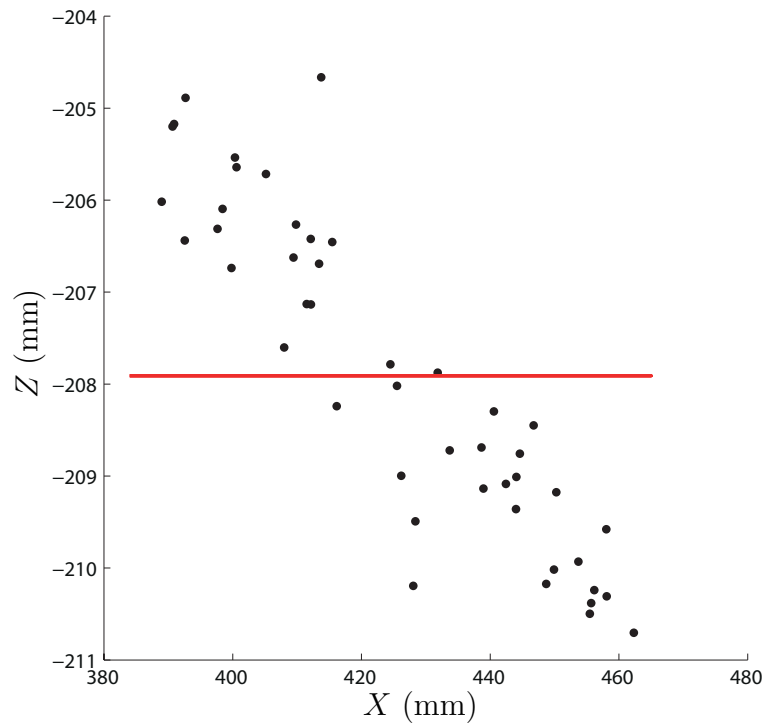


Figura 3.9: Estimación del plano que describe la posición y orientación de la superficie de la pieza de trabajo visto en el plano XZ.

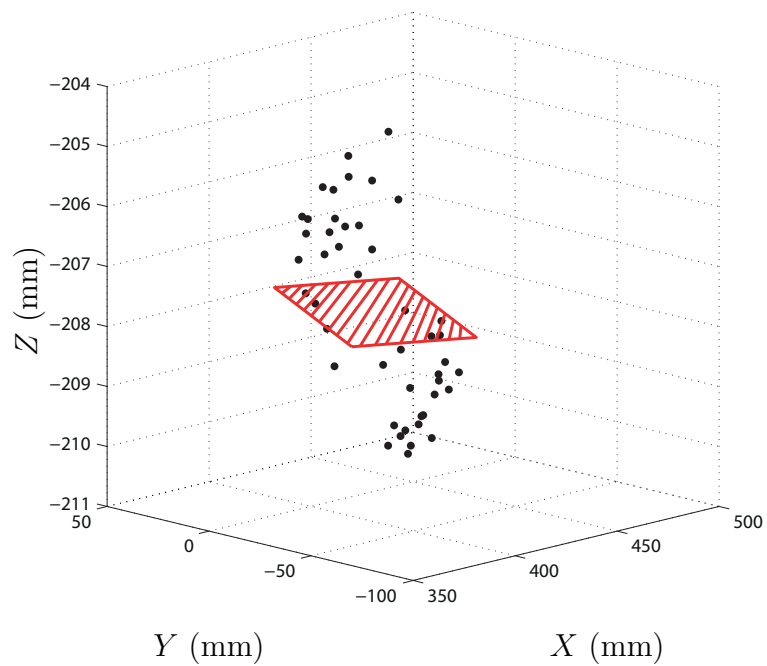


Figura 3.10: Estimación del plano que describe la posición y orientación de la superficie de la pieza de trabajo visto en los tres ejes XYZ.

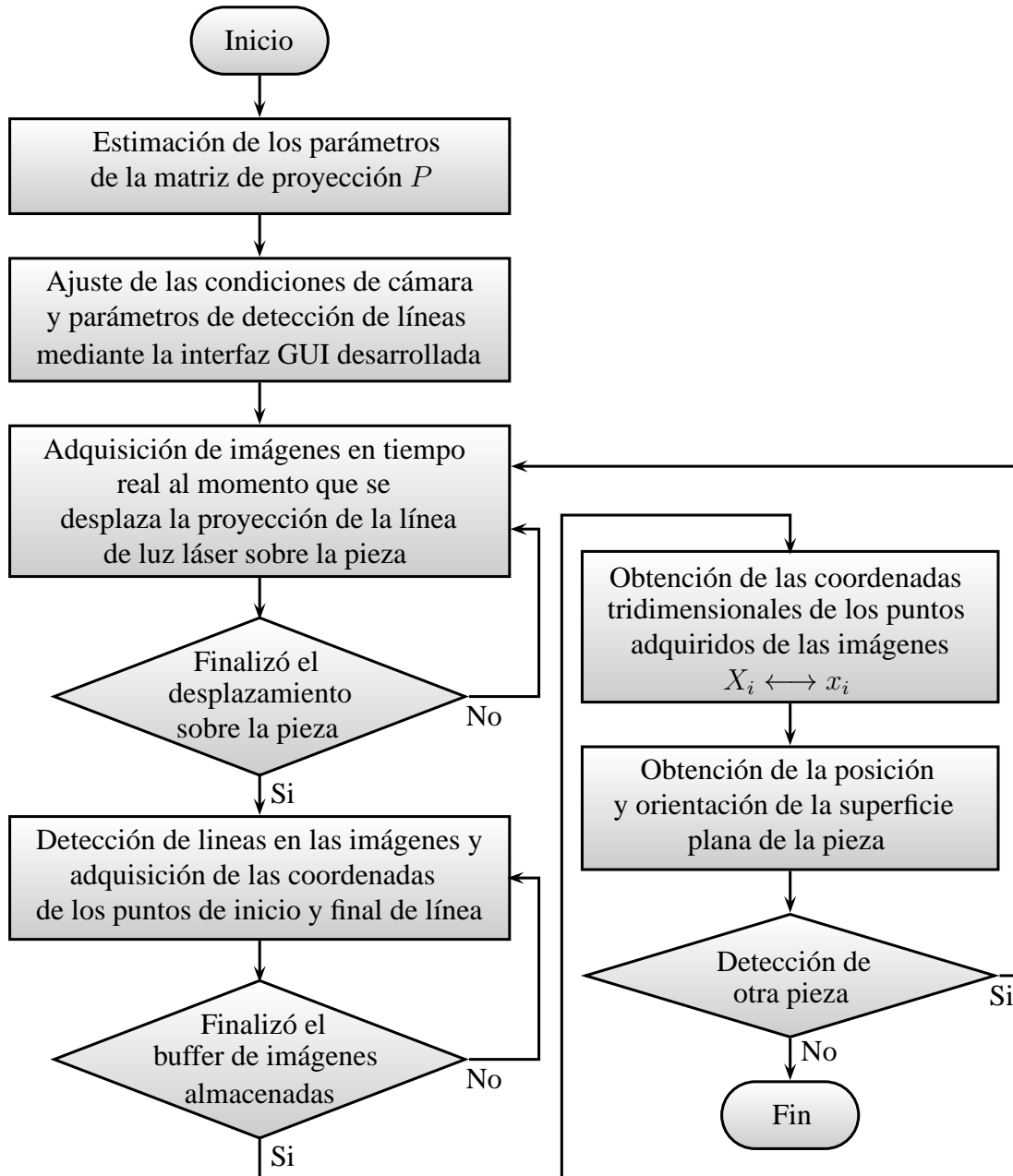


Figura 3.11: Diagrama de flujo del algoritmo de detección de superficies planas en piezas

Implementación y pruebas experimentales

En el capítulo anterior se definió los algoritmos que se emplearon en el presente trabajo, con el propósito de obtener la posición y la orientación de las piezas mediante visión computacional. Estos algoritmos de visión son combinados con los de control del robot manipulador y con los del control de la celda de manufactura para obtener un sistema de detección y manipulación automático de piezas.

En su primera etapa, este sistema de robótica y visión se implementó en el laboratorio de Mecatrónica de la Facultad de Ingeniería de la Universidad Autónoma de San Luis Potosí (UASLP). El objetivo de esta implementación fue probar la metodología LCM-CSM y obtener una aplicación que fuera capaz de detectar envases de productos y estibarlos a un contenedor. La plataforma se conforma por una computadora de control que se utilizó: a) para procesar la información proveniente de las cámaras digitales, b) dirigir el robot manipulador a la configuración adecuada para estibar la pieza, c) controlar el encendido y apagado de los leds del efector final, d) controlar la unidad de luz láser estructurado y e) controlar la posición de la unidad PTU en la que se encuentra montada el láser y que proyecta un punto de luz estruc-

turada sobre la pieza a estibar.

En la segunda etapa de este proyecto se desarrolló una implementación en la celda de Manufactura Integrada por Computadora (CIM) con el que se cuenta en el laboratorio de Control Numérico por Computadora (CNC-CIM) de la misma institución. Con esta aplicación se realizaron pruebas de precisión de detección y estiba de piezas en una banda transportadora. El hardware de esta plataforma es similar a la plataforma para estiba de envases, las diferencias residen que esta última cuenta con una banda transportadora, controlado por medio de un Controlador Lógico Programable (PLC) externo con el que se mandan las acciones de conmutación entre el robot y la CIM, otra diferencia con respecto a la primera implementación, es la integración de un patrón de línea en la unidad de luz láser estructurado. Este patrón de línea se utiliza para detectar por visión computacional los bordes de la pieza y encontrar su posición y ubicación con respecto al sistema referenciado a la base del robot.

4.1. Hardware de las plataformas de pruebas

El hardware con el que se realizaron las 2 implementaciones antes citadas se describen a continuación.

4.1.1. Celda de Manufactura Integrada por Computadora (CIM)

Una celda de manufactura integrada por computadora (CIM) es un sistema compuesto por varias máquinas y subsistemas enlazados por un sistema de transporte y control común, correspondientes a diferentes piezas o productos, sin necesidad de cambiar los equipos del sistema. La CIM del laboratorio de CNC/CIM está formada por: un torno de control numérico, una fresadora de control numérico, un robot

manipulador tipo serie de 6 grados de libertad, una banda transportadora de lazo cerrado, una estación de almacenamiento automático y una computadora que coordina las labores de mecanizado, almacenamiento y transporte de las piezas de trabajo. El equipo con el que se realizó el presente trabajo fue conformado por: el robot manipulador y la banda transportadora. A continuación se presentan las características de la banda transportadora, las características del robot manipulador serán mencionadas en la sección [4.1.3](#).

La banda transportadora es un sistema de lazo cerrado, que consta de movimiento continuo para transportar piezas en las tarimas de transporte (pallets). Los pallets son detenidos en las estaciones de la banda transportadora donde las piezas son cargadas o descargadas por el robot. Las dimensiones de la banda transportadora son de 1.4×6.24 mts. con una longitud total de 15 mts. Ésta es controlada por un motor de VAC220/380, 3 fases y de 0.75 hp. Cuenta con 6 estaciones de paro, mediante las cuales se monitorea y controla el flujo de los pallets a lo largo de la banda transportadora. Cada estación de paro esta conformada por: 2 sensores magnéticos para la detección del arribo del pallet, además de 4 sensores magnéticos para la identificación y monitoreo de los pallets dentro de la banda transportadora y un pistón neumático para detener y liberar el pallet. En la figura [4.1](#) se puede observar las características de la banda transportadora, como lo son la estación de paro del robot y la estación temporal de descarga de piezas.

4.1.2. Controlador Lógico Programable (PLC)

La banda transportadora cuenta con 6 estaciones de paro, en una de éstas se realiza la detección y descarga de la pieza por parte del robot. Cada estación de paro cuenta con un sistema de identificación y monitoreo de pallets, manipulado por un controlador lógico programable (PLC) y conectados a una red Profibus DP a

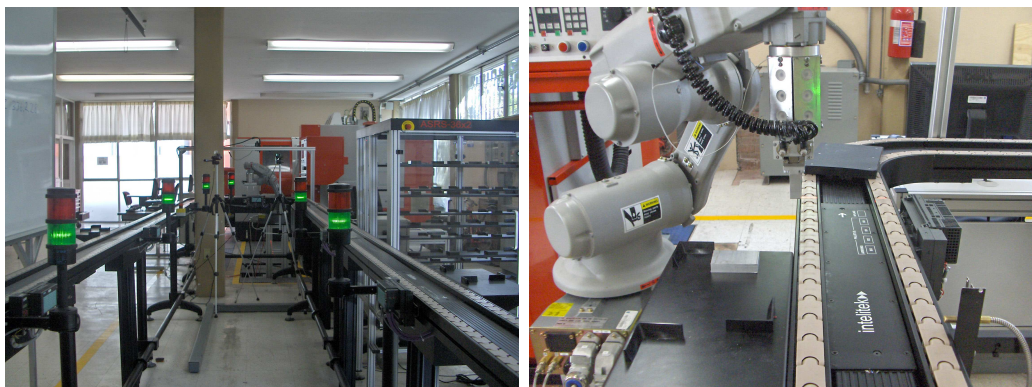


Figura 4.1: Banda transportadora de la Celda Inteligente de Manufactura (CIM)

un PLC central encargado de llevar acabo el monitoreo de los pallets en la banda transportadora. Se optó por añadir un PLC de la marca Siemens® modelo LOGO! a la estación de paro donde se encuentra el robot, para la conmutación de tareas entre la banda transportadora y el robot. El diagrama de conexión del PLC adaptado a la estación de paro del robot se puede observar en la figura 4.2.

Como se observa en la figura 4.2, el sensor fotoeléctrico S1 se activa al detectar la presencia del pallet que se encuentra en la banda transportadora. Este sensor energiza un relevador de control CR1, el cual a su vez activa CR2 quien se encarga de energizar el solenoide SOL-01 para que el pistón extienda y detenga el pallet. Simultáneamente se energiza CR3 el cual le da la señal al robot para que realice su maniobra. Una vez terminada la maniobra del robot, éste da una señal mediante el relevador CR4, el cual abre el circuito de CR2, originando que este se apague y el solenoide SOL-01 se desenergice para regresar el pistón a su posición original, permitiéndole al pallet continuar su movimiento en la banda transportadora.

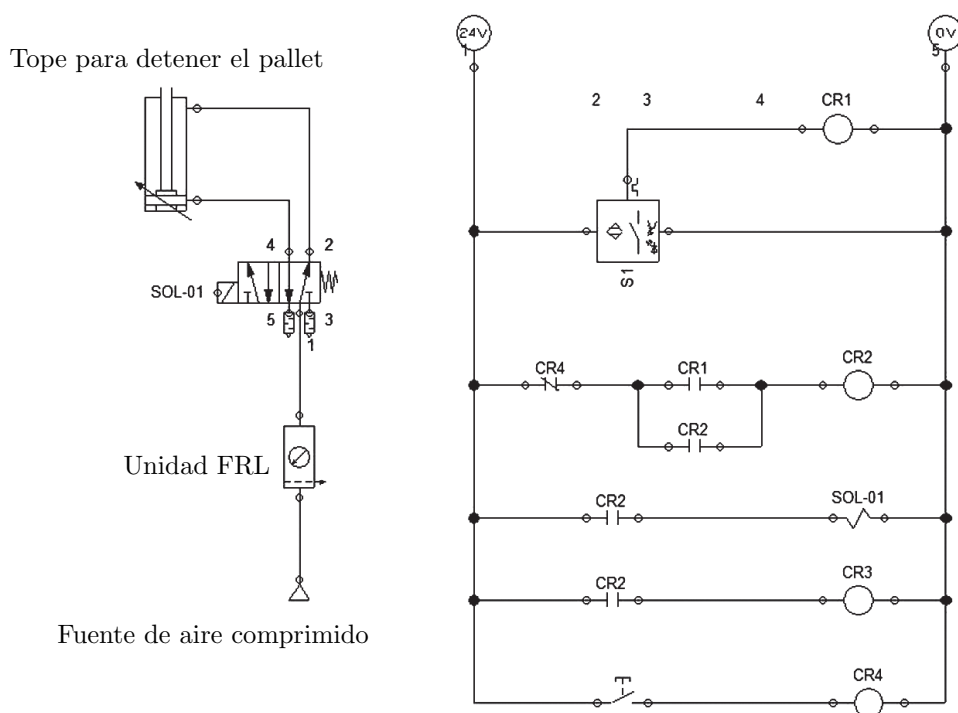


Figura 4.2: Diagrama de conexión del PLC de control de la estación de paro del robot.

4.1.3. Robot manipulador

Para ambas implementaciones se utilizaron robots industriales de la compañía Yaskawa marca MOTOMAN y modelo HP3 de 6 grados de libertad, con un alcance vertical de 1163 mm y 701 mm en horizontal, con una capacidad de carga útil de 3 kg. y una precisión de repetibilidad de ± 0.03 mm, comandado por un controlador de la serie NXC100. Este controlador cuenta con una interfaz de control “Teach-Pendant” con ambiente Windows CE[®], que se comunica a una computadora a través de los puertos de comunicación serial RS232C y Ethernet (100 Mbps).

Para la estiba de piezas se utilizó un actuador neumático (gripper) de la marca SMC[®] modelo MHF2-16D2, acoplado al último eslabón del robot, con dos dedos pa-

ralelos y una apertura de 64 mm. Éste se activa por medio de una válvula de solenoide conectada a una interfaz de señales entrada/salida comandadas por el controlador NXC100.



Figura 4.3: Sistema del robot manipulador

4.1.4. Marcas visuales

Como se mencionó en la sección 2.3.1, las marcas visuales son necesarias para ubicar la posición del efector final en espacio de cámara y realizar la estimación de parámetros de la matriz de proyección P . Estas marcas visuales se establecieron mediante leds ensamblados a un contenedor de aluminio, el cual se diseñó y fabricó con el propósito de servir como estructura de soporte de los circuitos y leds, además de servir como unión entre el último eslabón del robot y el actuador neumático. El diseño se realizó con el propósito de utilizarlo en futuras aplicaciones del método de

CSM, en las que son necesarias varias marcas visuales. En nuestro caso, por facilidad de programación del algoritmo, se recurrió al uso de un led como la marca visual para la iniciación del método aplicado. Se muestra en la figura 4.4 el diseño del soporte de las marcas visuales fabricado para el presente trabajo. En la figura 4.8 puede observarse un esquema de conexión para el control del encendido y apagado de las marcas visuales mediante la PC.



Figura 4.4: Diseño de la estructura de las marcas visuales

4.1.5. Cámaras digitales

La adquisición de imágenes se realiza por medio de 2 cámaras digitales marca uEye® modelo UI-1640-C, con una resolución de 1280×1024 pixeles, con un tamaño de sensor de $1/3''$, que permite obtener una captura de 25 cuadros por segundo (fps) en su máxima resolución, con comunicación con la computadora a través del puerto usb. Estas cámaras cuentan con lentes marca Infaimon modelo INFW38013MCN, las cuales cuentan con tornillos de fijación para el ajuste del foco y el iris, con una distancia focal de 8 mm y ángulos de visión horizontal y vertical de 58×45 grados respectivamente. En la figura 4.5 se muestra la cámara digital.

Estas cámaras son encargadas de la adquisición de las imágenes para el procesamiento de la información visual del sistema. Se encuentran montadas sobre trípodes que les permiten posicionarse y orientarse, de tal manera que sea posible visualizar el área donde se encuentran las piezas de trabajo y la marca visual que tiene el robot manipulador.



Figura 4.5: Cámara digital

4.1.6. Unidad Pan-Tilt (PTU)

Este dispositivo es un mecanismo de dos grados de libertad, de la marca Directed Perception[®] modelo D46-17.5, que dispone de una resolución de 0.012857° , una velocidad máxima de $300^\circ/\text{seg}$ y una capacidad de carga de 1.81 kg. En esta unidad se encuentra montada la unidad láser con la que se proyecta la línea de haz de luz sobre la pieza, permitiendo que se desplace sobre el objeto a detectar, adquiriendo las imágenes de los cambios de la línea en la pieza. La unidad PTU permite un control de programación por medio de una computadora, a través de los puertos de comunicación RS485 y RS232, en el cual es posible controlar la posición, velocidad y aceleración del mismo. Este dispositivo se muestra en la figura 4.6.



Figura 4.6: Unidad Pan-Tilt

4.1.7. Unidad de láser estructurado

Se trata de un sistema de luz láser estructurado utilizado para iluminar un objeto con un patrón de luz específico (punto, línea, etc.) y para detectar la forma como cambia este patrón sobre el objeto. La unidad láser utilizada en este proyecto es marca Lasiris[®] modelo SNF, conformado por 2 partes, el generador láser propiamente dicho y el cabezal con el patrón de una línea. Los cabezales son intercambiables y ajustables para obtener el ancho de línea deseado a la distancia en que se encuentre. El generador láser requiere de una alimentación de 4.8 a 6.5 VDC, el cual se tiene conectado a la caja de señales I/O del que se controla el encendido y apagado a través del controlador del robot, como se muestra en la figura 4.7.

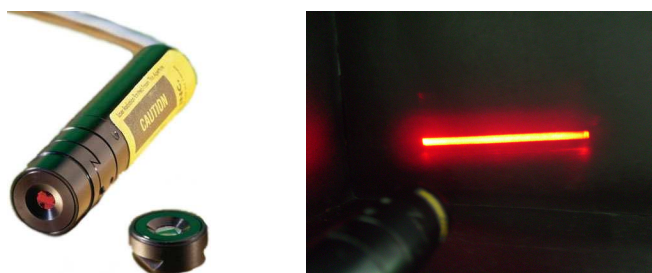


Figura 4.7: Unidad de láser estructurado con cabezal de línea

4.1.8. Computadora de control

La computadora encargada del control del sistema de visión y manejo de las señales de I/O de la celda de manufactura cuenta con las siguientes características: procesador Intel® Core2 Duo, a 3.33 GHz, y 3 GB de memoria RAM.

En la figura 4.8 puede observarse el diagrama de señales de control, realizado entre la computadora y los distintos hardware utilizados para las 2 implementaciones; cabe destacar que para la implementación de detección y estiba de envases de producto no se utilizó el PLC indicado en el diagrama.

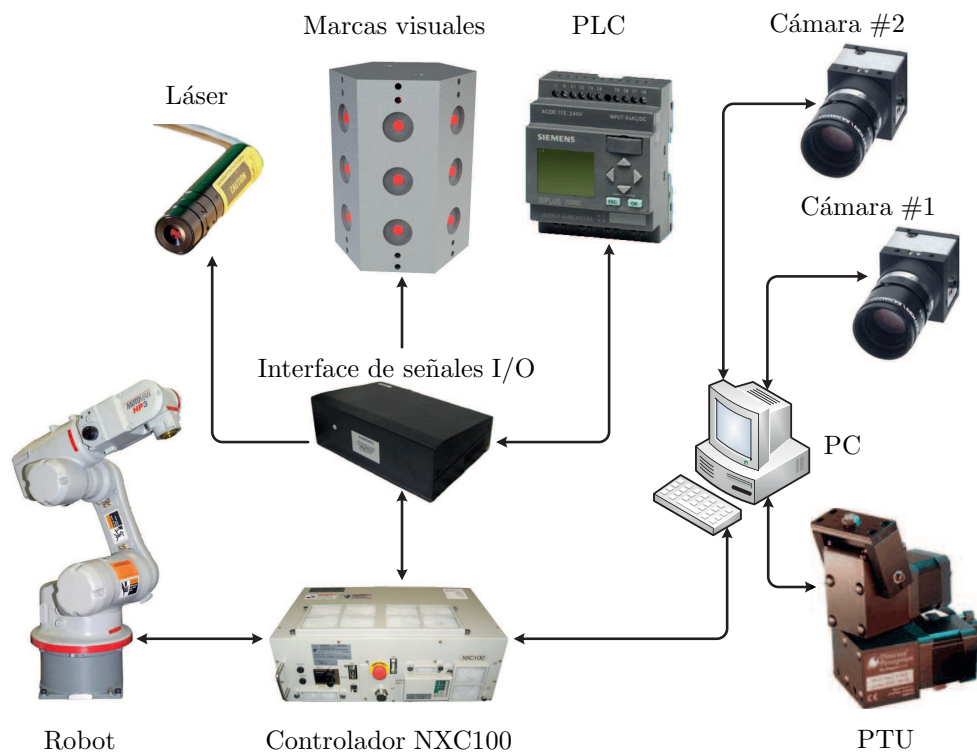


Figura 4.8: Diagrama de conexión de señales de control entre la PC y el equipo de las implementaciones

4.2. Software

El software con el que se realizó las implementaciones en el laboratorio de Mecatrónica y CNC/CIM, fue desarrollado en lenguaje de programación C++ en el entorno de desarrollo integrado de Microsoft Visual Studio para el sistema operativo Windows[®] XP SP3, con integración de las librerías de *OpenCV*, *Numerical Recipes*, *Motocom32* (funciones de transmisión del control del robot), *uEye API* (para la comunicación de las cámaras digitales).

En general, el software desarrollado se puede dividir en 2 secciones: (a) software de control de señales de entrada/salida (I/O), el cual se encarga de recibir y enviar señales de los distintos dispositivos de hardware antes mencionados y (b) software de procesamiento de información, en el que se programaron los algoritmos para el tratamiento de imágenes provenientes de las cámaras y del procesamiento de algoritmos.

4.2.1. Software de control de señales de entrada/salida

Para la transmisión de datos entre los distintos dispositivos de hardware y la computadora se programaron distintas funciones y librerías, las cuales se describen a continuación de acuerdo al hardware controlado.

Robot manipulador . En este se utilizaron las librerías de programación *Motocom32* propias del robot, para el control del robot y de las señales I/O. Éstas se encuentran conectadas a una tarjeta de adquisición seguida del controlador *NXC100* del robot. Las funciones de control de las señales de entrada permiten la conmutación de encendido y apagado, para el sistema de marcas visuales (leds), la unidad láser y el control del relevador *CR4* del PLC incorporado a la estación de paro del robot de la banda transportadora.

Cámaras digitales . Éstas cuentan con las librerías de programación uEye API nativas de las cámaras, en las que es posible controlar la adquisición de imágenes. En el caso del presente trabajo, las imágenes adquiridas son almacenadas en una localidad de memoria (memoria dinámica), ésta es asignada a una estructura de *OpenCV* del tipo imagen `IplImage`, con la cual es posible trabajar con las demás funciones de la librería de *OpenCV* y las cuales se describen en el apéndice A1.

Unidad Pan-Tilt . Este dispositivo se comunica con la PC a través del puerto serial. La comunicación nativa del PTU es por medio del programa HyperTerminal, en el cual es posible enviar y recibir información, esto mediante comandos en código ASCII. Se realizaron funciones mediante protocolos de intercambio de datos por el puerto serial, con los cuales es posible establecer comunicación, usando el listado de comandos propios de la unidad.

4.2.2. Software de procesamiento de información

El software para procesamiento de información comprende los algoritmos utilizados para el manejo de la datos. Una descripción general de acuerdo al tipo de información procesada se presenta a continuación.

Algoritmos de procesamiento de imágenes . Estos algoritmos procesan la información procedente de las imágenes, entre las operaciones realizadas por estos algoritmos se encuentran: filtros y diferenciación de imágenes, extracción de bordes, obtención de centroides de marcas visuales, detección de líneas y descripción de formas. Estas operaciones se realizaron con ayuda de las funciones de *OpenCV* y *Numerical Recipes*.

Algoritmo de estimación de parámetros . Éste describe la técnica de estimación de parámetros de la matriz de proyección P , como se mencionó en la

sección 2.3.1.

Algoritmo de reconstrucción de superficies planas . Con éste se establece la posición y orientación de una superficie plana, el cual se realiza a partir del procesamiento de imágenes. Este algoritmo se describe en la sección 3.2.

Algoritmo LCM-CSM . Este algoritmo consta de la metodología de manipulación de espacio de cámara con modelo lineal de cámara, la cual permite posicionar y orientar el robot con respecto a una pieza de trabajo, como se mencionó en la sección 2.3.

Una descripción específica de las distintas funciones mencionadas se pueden encontrar en el apéndice A1.

4.3. Implementación

En el presente trabajo se realizaron 2 implementaciones, las cuales tienen su base en la metodología de LCM-CSM. Éstas fueron realizadas en los laboratorios de la Facultad de Ingeniería en la Universidad Autónoma de San Luis Potosí. En ambos casos se contó con el robot manipulador Motoman descrito en la sección 4.1.3; a continuación se presenta una descripción de estas 2 implementaciones realizadas.

4.3.1. Detección y estiba de envases de producto

La detección y estiba de envases producto se realizó en el laboratorio de Mecatrónica de la Facultad de Ingeniería de la UASLP. El hardware utilizado para esta implementación fue: robot manipulador (sección 4.1.3), 2 cámaras digitales (sección 4.1.5), la unidad láser con cabezal de proyección de un punto (sección 4.1.7), la unidad Pan-Tilt (sección 4.1.6) y una computadora (sección 4.1.8), figura 4.9. Esta implementación requiere que se ejecute una serie de poses preplaneadas, éstas se refieren como el “*preplan*”, y se utilizan para obtener la inicialización en la estimación

de los parámetros de la matriz de proyección P utilizadas en LCM-CSM (sección 2.3.1). A continuación se enumeran los pasos realizados en la implementación.

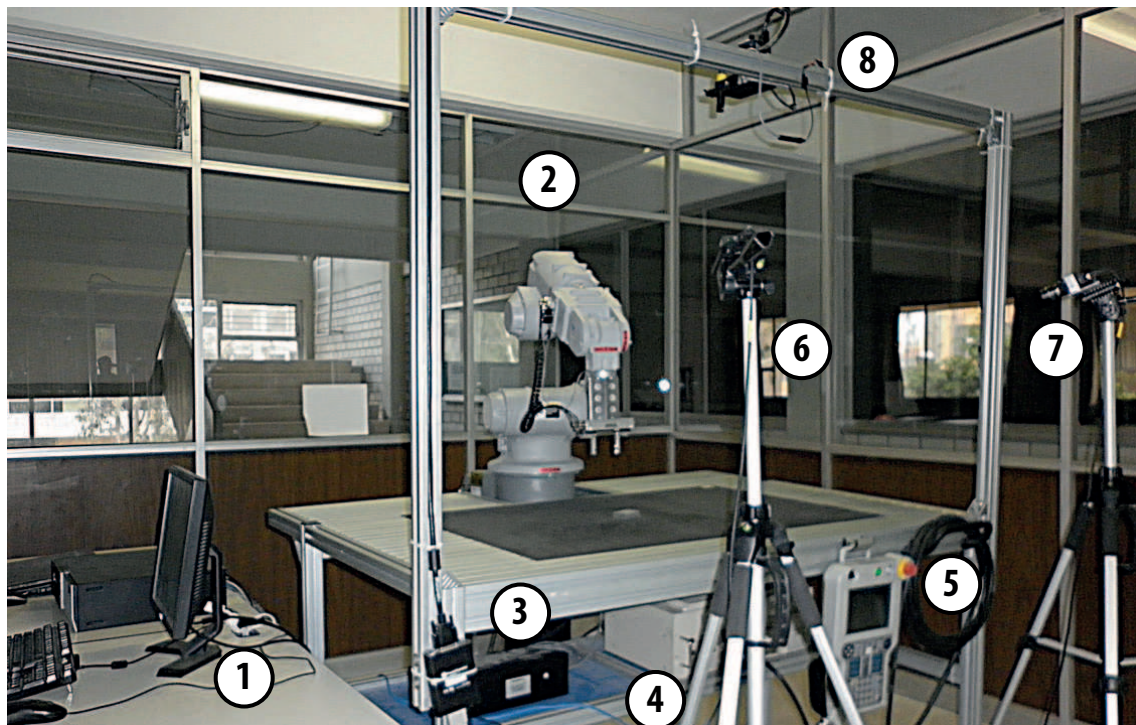


Figura 4.9: Implementación de detección y estiba de envases de producto en el laboratorio de Mecatrónica; (1) PC de control, (2) Robot, (3) Interface de señales I/O, (4) Controlador NXC100, (5) Teach-Pendant, (6) y (7) Cámaras digitales, (8) Unidad láser montada en PTU.

Paso 1: Proyección del láser sobre el envase . Se coloca la proyección del punto de haz de luz láser sobre la tapa del envase, mediante la unidad Pan-Tilt en la que se encuentra montado el láser. El posicionamiento se realiza de forma manual, hasta obtener la proyección del punto láser centrado en la tapa del envase.

Paso 2: Captura de imágenes . Se captura una imagen por cada cámara con el

láser encendido y otra apagado.

Paso 3: Obtención del centroide del láser . Se determinan las coordenadas del centroide (x_c^i, y_c^i) de la mancha del láser en las imágenes, como se mencionó en la sección 3.1.1.

Paso 4: Reconstrucción 3D . Se obtiene el vector de posición (X, Y, Z) del punto láser mediante la solución de $\mathbf{x} = P\mathbf{X}$.

Paso 5: Posicionamiento del robot . Con la coordenada (X, Y, Z) obtenida, se comanda el robot para que alcance una posición superior a la obtenida, con la orientación de la muñeca del robot perpendicular al envase de producto.

Paso 6: Estiba del producto . Se acciona el gripper y se posiciona el robot en uno de los espacios de un arreglo matricial predefinidos en el contenedor, para finalmente soltar el envase y colocarse en una posición de inicio para el siguiente envase.

4.3.2. Detección y estiba de piezas en una celda de manufactura

Una segunda implementación fue realizada en el laboratorio de CNC-CIM. En ésta, el hardware utilizado fue: el robot manipulador, 2 cámaras digitales, la unidad láser con cabezal de proyección de una línea, la unidad Pan-Tilt, la banda transportadora de la CIM y una computadora personal; la descripción de estos se menciona en la sección 4.1, figura 4.10. De forma igual que la aplicación anterior, ésta requiere se realice previamente la estimación de parámetros de la matriz de proyección P y se calcula una vez para cualquier número de veces que se vaya a realizar la implementación, siempre y cuando no se cambie orientación y/o posición de las cámaras. La metodología seguida para esta implementación se presenta a continuación:

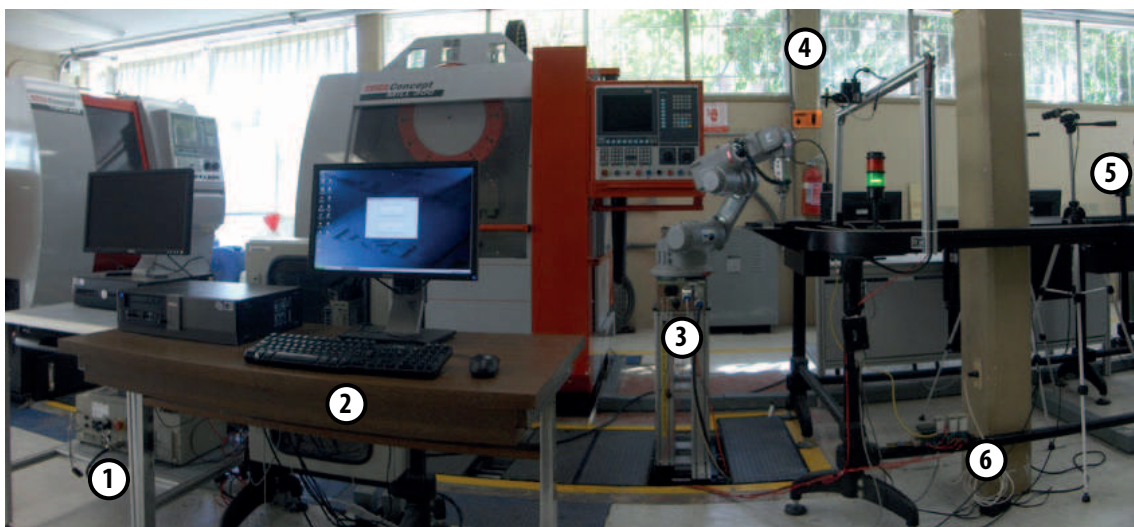


Figura 4.10: Implementación de detección y estiba de piezas en CIM; (1) Controlador NXC100, (2) PC de control, (3) Robot adaptado con soporte de marcas visuales y gripper, (4) Unidad láser montada sobre PTU, (5) Cámaras digitales, (6) PLC externo.

Paso 1: Colocación de la pieza en uno de los “*pallets*” sobre la banda transportadora, la cual se encuentra en movimiento.

Paso 2: Detección del pallet por medio del sensor fotoeléctrico, en la estación de paro de carga/descarga de piezas del robot.

Paso 3: Energización del solenoide por medio del PLC para que el pistón extienda y detenga el pallet.

Paso 4: Una vez detenido el pallet, el PLC energiza un relevador de control el cual cierra un circuito con la caja de señales I/O, éste manda una señal digital al controlador NXC100 del robot, la cual es interpretada como la señal #1 de “entrada universal” por el robot.

Paso 5: Una vez que el robot detecta como activada la “entrada universal” #1, se

inicia la adquisición de imágenes, al momento que se hace un barrido con la línea láser sobre la pieza.

Paso 6: Las imágenes adquiridas son procesadas; este procesamiento consiste en ajustar una línea en la imagen a la línea proyectada sobre la superficie de la pieza. Además de obtener las coordenadas en la imagen de los puntos de inicio y fin de línea.

Paso 7: Las coordenadas obtenidas de los puntos en las imágenes, se mapea por medio de la matriz de proyección P , obteniendo las coordenadas 3D de los puntos con respecto al robot. En la figura 3.7 se observa la línea encontrada en la imagen y sus coordenadas.

Paso 8: Esta serie de puntos en el espacio, son ajustados a un plano del cual se determine su posición y orientación.

Paso 9: Con las coordenadas de la posición y orientación de la superficie de la pieza, se comanda el robot para que se posicione para tomar la pieza, se sujeta la pieza activando la señal de “salida universal” #1 del robot para cerrar el gripper. Por último se posiciona al robot sobre una de las estaciones temporales del CIM y se desactiva la señal del gripper para liberar la pieza.

Paso 10: Al finalizar la maniobra de estiba de la pieza por el robot, éste activa la señal de “salida de universal” #2 activando un relevador de la caja de señales I/O, el cual a su vez manda una señal al relevador CR4 del PLC de la banda transportadora (figura 4.2), originando que el pistón regrese a su posición original y permitiéndole continuar al pallet su movimiento en la banda transportadora.

En esta implementación se utilizó una banda transportadora para trasladar piezas, la cual se controlaba por la computadora central a través de un PLC. La banda

transportadora contaba ya con un sistema de detección e identificación de los pallets en cada una de las estaciones de paro; el sistema de detección del pallet cuenta con 2 sensores magnéticos para el arribo a la estación de paro, mientras que la identificación se realiza por medio de 4 sensores magnéticos que detectan la combinación de imanes (mediante código binario) ubicados en la parte inferior de cada pallet. Cada estación de paro es controlada por un PLC, los cuales se encuentran conectados por medio de una red Profibus DP al PLC central encargado de llevar a cabo el monitoreo de los pallets en las diferentes estaciones de paro de la banda transportadora. Para la estación de paro ubicada en el área de descarga del robot se realizó una modificación, a la cual se desactivó el sistema de detección e identificación de los pallets de fábrica, agregando un PLC Logo! (descrito en la sección 4.1.2) el cual controla el arribo de los pallets a la estación deteniéndolos por un pistón extendido propio de la estación. Este es activado al momento que el pallet se aproxima a la estación, detectado por un sensor fotoeléctrico externo. Una vez detectado el pallet por el PLC Logo!, éste manda una señal a la interfaz de señales I/O conectada al robot para el inicio de maniobra.

Al momento que se detecta la señal de entrada al robot de inicio de maniobra, se realiza la adquisición de imágenes del barrido que realiza la línea de luz láser sobre la pieza, Estas imágenes son procesadas para estimar los parámetros de la línea que mejor se ajuste a la proyectada sobre la pieza, de ahí se obtienen las coordenadas de inicio y fin de línea en la imagen, ver figura 3.7. A partir de las coordenadas adquiridas de los puntos en las imágenes, se obtienen los puntos correspondientes en coordenadas 3D (X, Y, Z) con respecto al robot mediante la solución de $\mathbf{x} = P\mathbf{X}$. Estos puntos son ajustados a un plano del cual se determina su posición y orientación; este algoritmo se describió en la sección 3.2. En las figuras 3.10 y 3.9 se observa el resultado de la estimación del plano que describe la posición y orientación de la

superficie de la pieza.

En el presente capítulo se presentó la descripción del sistema de visión utilizado, así como los algoritmos que describen esta metodología, la cual se validó a través de dos implementaciones realizadas. En este capítulo se presentaron características y especificaciones tanto del hardware como del software utilizado en el presente trabajo. Las conclusiones y resultados de estas implementaciones se mencionan en el siguiente capítulo.

Conclusiones y resultados

En el presente capítulo se presenta una discusión de los resultados obtenidos con las dos implementaciones. Estas implementaciones se realizaron con el fin de obtener una validación de la metodología del LCM-CSM. Como se dijo anteriormente, este método parte de la propuesta de utilizar la metodología de Manipulación de Espacio de Cámara (CSM) con un Modelo Lineal de Cámara (LCM). Además, se realizaron una serie de experimentos para establecer la precisión de la metodología implementada, la cual se hizo a través de la estimación de la posición y orientación de las piezas de trabajo contra datos conocidos previamente en los experimentos.

La primera implementación fue un algoritmo de control basado en visión para la detección y estiba de envases de producto. Ésta se realizó como una “*prueba de concepto*” o “*PoC*”, en otros términos, es una implementación realizada con el propósito de validar la metodología de LCM-CSM. En esta implementación se controla únicamente la posición del robot con respecto de la pieza de trabajo, haciendo incidir el láser puntual sobre la superficie de la pieza de manera manual. Por ello se buscó un envase con tapa de geometría circular, en la cual el problema principal reside en comandar al robot a la posición de donde se encuentra la tapa del envase, estableciendo como la orientación de la muñeca perpendicular al envase sin importar la orientación al plano horizontal.

Una segunda implementación se realizó en una celda de manufactura, sobre un algoritmo de visión que permite detectar y estibar piezas regulares en una banda transportadora. Esta implementación determina de forma automática la posición y orientación de una pieza con respecto al robot, de tal forma que el algoritmo sea capaz de determinar la orientación adecuada para que el robot sea capaz sujetar la pieza mediante el gripper.

Resultados de la implementación de detección y estiba de envases de producto.

Para la realización de la metodología que permite detectar la posición de las piezas, es necesario contar con la matriz de proyección P . Ésta se evalúa mediante el algoritmo de estimación de los parámetros de P descrito en la sección 2.3.1; es importante mencionar que P es válida mientras las condiciones de posición y orientación de las cámaras no hayan sido modificadas con respecto al robot. Los resultados obtenidos del algoritmo de estimación de parámetros P de las 2 cámaras, para una muestra de 100 configuraciones “preplan” fueron:

Cámara 1

$$P_1 = \begin{bmatrix} -1.5588 & 1.94480 & -0.11 & 1317.1920 \\ 0.5074 & 0.3621 & -2.3793 & -243.1766 \\ -0.0006 & -0.0002 & -0.0002 & 1 \end{bmatrix}$$

Cámara 2

$$P_2 = \begin{bmatrix} 0.7640 & 2.5194 & -0.0785 & 102.5530 \\ 0.6232 & -0.2101 & -2.5222 & -282.2620 \\ -0.0005 & 0.0003 & -0.0002 & 1 \end{bmatrix}$$

De manera preliminar a la implementación de la detección y estiba de envases de producto, se realizó un análisis del error que se lograba en el control de la posición, el cual consistió en colocar una cuadrícula sobre la superficie de trabajo del robot. Se programaron 50 posiciones en la unidad Pan-Tilt, las cuales permiten hacer incidir el punto láser en los nodos (intersección en la cuadrícula) de interés. En cada una de estas posiciones del Pan-Tilt se comandó llegar el robot por medio del algoritmo descrito en la sección 4.3.1. Éste deberá hacer coincidir un marcador colocado en el efector final del robot con un punto de interés establecido en una cuadrícula. Las pruebas se realizaron para dos estimaciones de matrices de proyección, con 20 y 100 muestras. En las figuras 2 y 3 se muestran los resultados obtenidos para los 2 casos de estudio de error en los que se observa que, para el caso de 20 muestras, se obtuvo un error promedio de 5.9 mm con una desviación estándar de 2.5 mm, mientras que para el caso de 100 muestras se obtuvo un error promedio de 4.6 mm con una desviación estándar de 0.9 mm. Cabe mencionar que con el aumento en el número de muestras se mejora la precisión del sistema.



Figura 1: Estudio de error de posicionamiento del robot realizada en el patrón cuadrículado.

Para que el algoritmo descrito en la sección 3 fuera capaz de localizar los objetos con respecto al marco de referencia del robot, es necesario especificar el punto objetivo en la pieza de trabajo. Para ello se proyectó un haz de luz láser puntual que se buscaba incidir en el centro de la tapa superior del envase, de esta manera se tiene una marca visual de la cual se obtiene su localización en espacio de cámara. En esta primera implementación se direccionaba la unidad Pan-Tilt que posicionaba el punto de láser de manera manual. Una vez posicionado el punto láser sobre la tapa del envase, se inició el proceso de estimación del vector 3D de posición de esta marca visual con respecto al sistema de coordenadas del robot. Con esto el robot evalúa la coordenada (X, Y, Z) para tomar el envase y posteriormente colocarlo en el contenedor en una posición preestablecida. El tiempo total de ciclo para estibar cada envase fue de 50 segundos, la mayor cantidad de tiempo se consumía al momento que el usuario ubicaba manualmente el láser puntual para que incidiera sobre el centro geométrico de la tapa del envase (aproximadamente 30 seg.), la maniobra requerida por el robot para tomar y estibar el envase fue de aproximadamente 20 segundos. El tiempo de adquisición de la información visual y procesamiento de los datos no requería más de un par de segundos.

En esta implementación se presentó un error de posicionamiento debido a la selección que el usuario hacía para ubicar el centro de la tapa con el láser, ya que existe una diferencia entre la ubicación donde el usuario proyecta el láser y el centro geométrico de la circunferencia de la tapa del envase; además de este error de posicionamiento, debido a la destreza del usuario para ubicar el centro geométrico de la tapa, se cuenta con un error intrínseco al cálculo derivado de la estimación de la matriz de proyección P y la falta de actualización de parámetros, fundamental en CSM. Si la suma de estos errores son superiores al ancho del dedo del gripper

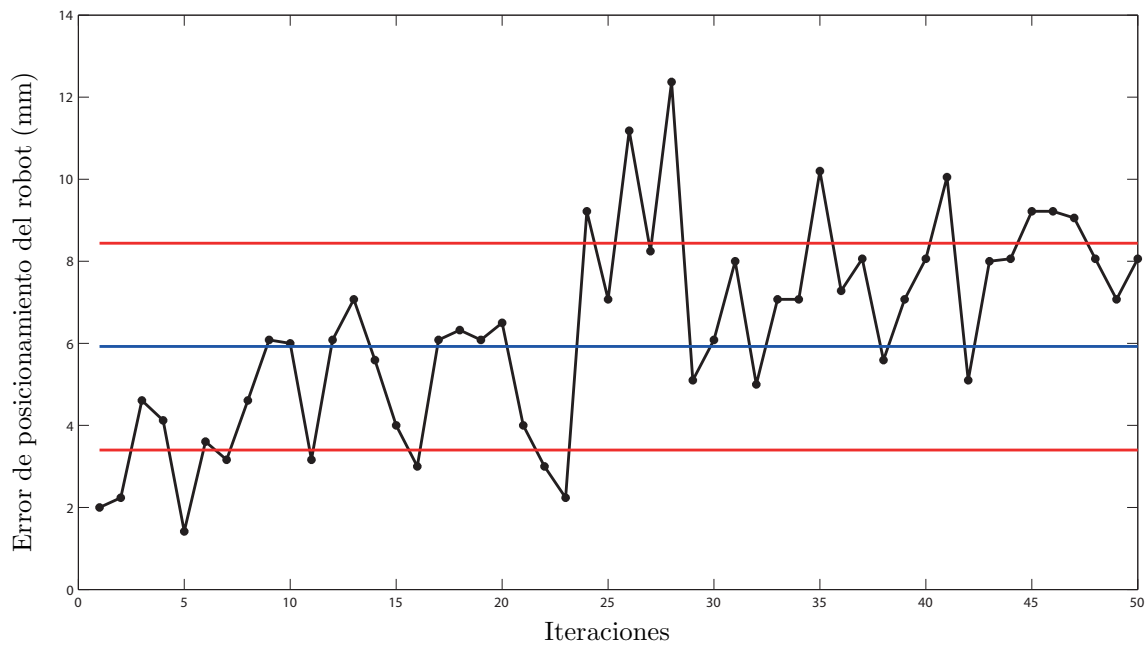


Figura 2: Análisis de error de posicionamiento del robot para 20 muestras en la estimación de los parámetros de la matriz de proyección.

neumático de 10 mm en el plano XY y de 25 mm en el eje Z , que es la altura a la que puede tomar el actuador los envases, provoca que no se tome con exactitud el objeto y lo coloque en una mala ubicación en el contenedor o que inclusive sea una operación fallida en su totalidad, situación la cual no sucedió en ninguna ocasión. Una descripción de las etapas de esta implementación se hace en la figura 5, además puede observarse en la figura 4 que el envase es sujetado de forma inadecuada debido al error de posicionamiento del robot generado a partir de las consideraciones antes citadas.

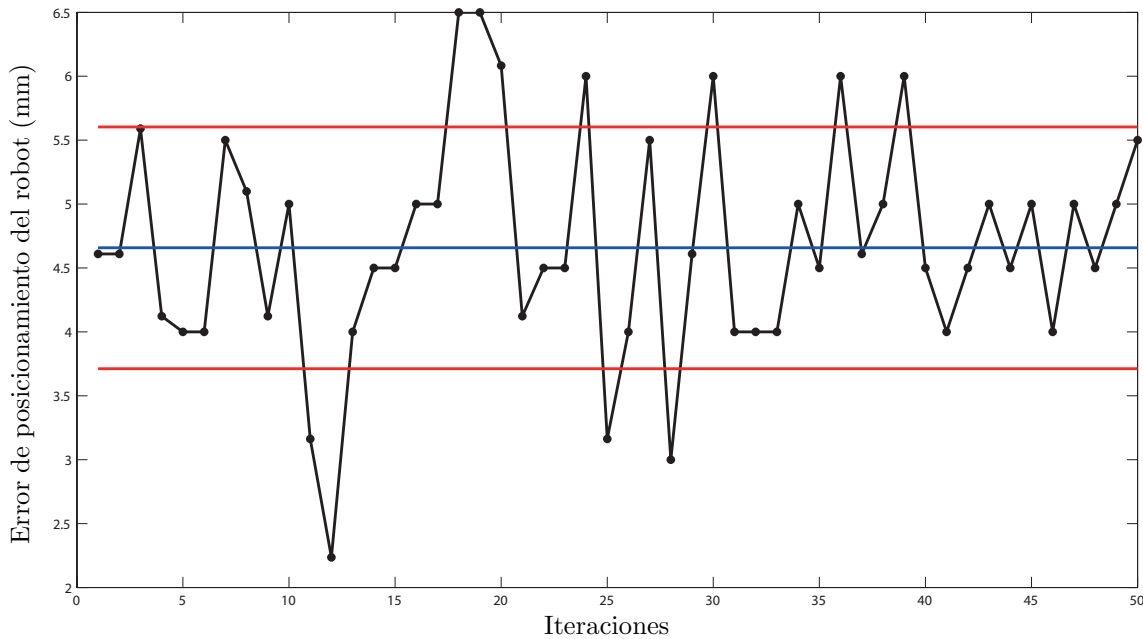


Figura 3: Análisis de error de posicionamiento del robot para 100 muestras en la estimación de los parámetros de la matriz de proyección.

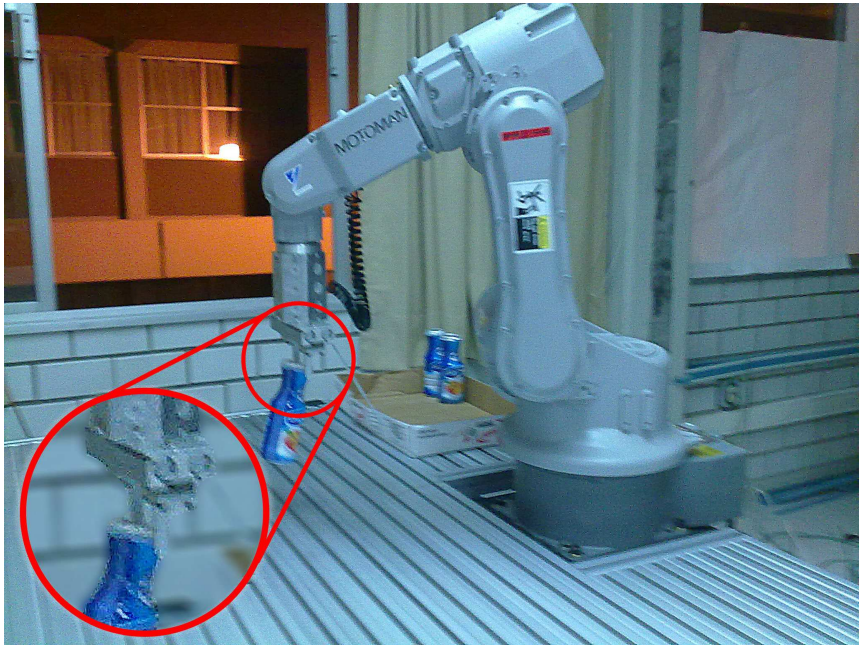


Figura 4: Adquisición de envase inadecuada derivada del error de posicionamiento.

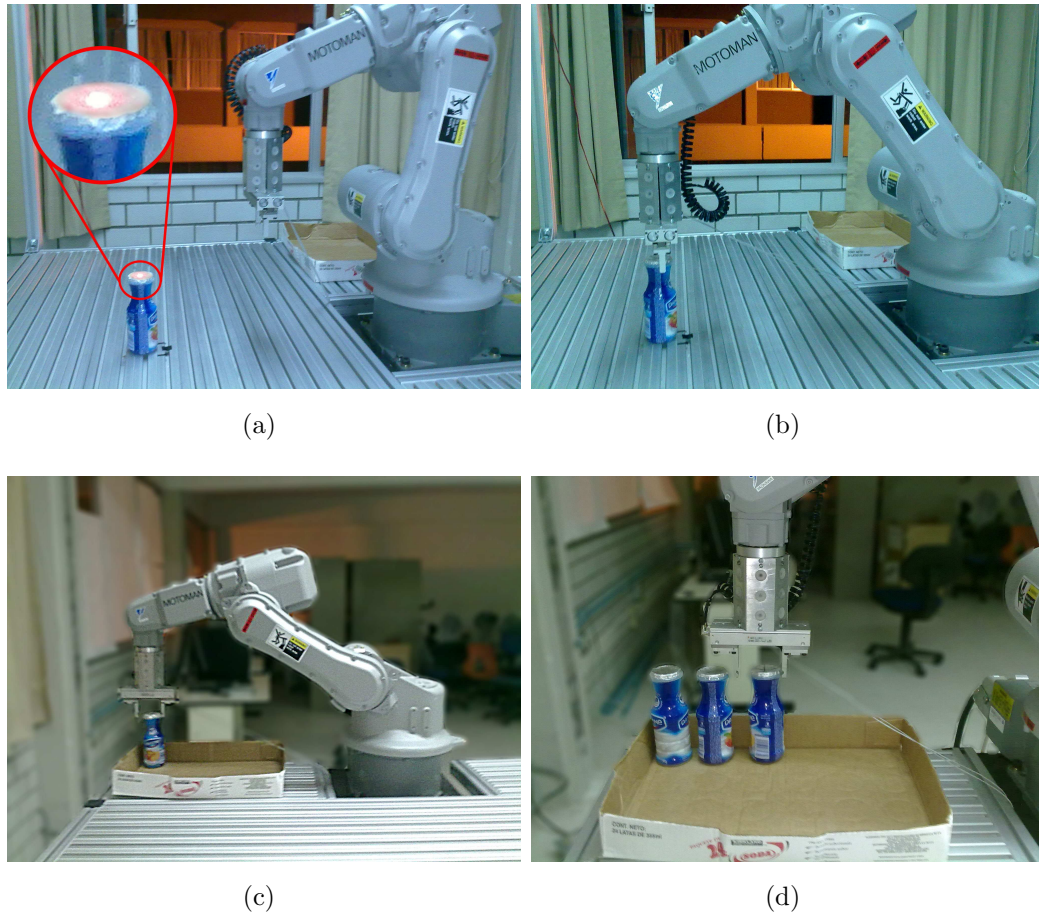


Figura 5: Etapas de la implementación de detección y estiba de envases de producto realizada en el laboratorio de Mecatrónica; (a) ubicación por el usuario del punto láser en el centro geométrico de la tapa del envase y adquisición del centroide en las imágenes, (b) posicionamiento por medio del algoritmo de visión y sujeción del envase mediante el actuador del robot, (c) traslado del envase al contenedor, (d) disposición de los envases en el contenedor.

Resultados de la implementación de detección y estiba de piezas en una celda de manufactura.

En esta segunda implementación, realizada en el laboratorio de CNC-CIM de la UASLP, se realizó un algoritmo basado en visión para la detección y estiba totalmente automática de piezas en una banda transportadora. De igual forma que la implementación anterior, se requiere de haber estimado la matriz de proyección P mediante el preplan.

Esta serie de puntos 3D de los bordes de la superficie de la pieza calculados, cuentan con un error derivado de la adquisición de las imágenes en el momento que se desplaza la proyección de línea láser sobre la pieza, debido a que el algoritmo realizado es del tipo secuencial, lo que significa que existe un retraso en la adquisición de las imágenes por parte de ambas cámaras (figura 3.6), ya que estas no toman la imagen de manera simultánea e instantánea; además, existe otro error proveniente del ajuste de la línea en la imagen derivado de la fluctuación de la reflectancia de la línea láser sobre la superficie de la pieza; es decir, que si se mantiene en una posición la proyección de línea de láser sobre la pieza y se procesan las imágenes para adquirir la línea, los parámetros de la misma cambian. Es decir, la pendiente y la ordenada al origen sufren variaciones entre una y otra iteración.

A partir de los errores en el algoritmo de detección de superficies de piezas, se decidió realizar un análisis para determinar cuál de los factores mencionados en el párrafo anterior influía más en el resultado y fuera más determinante para el error en la ubicación y orientación de la superficie. Este experimento consistió en colocar la pieza de trabajo, mediante el robot, en una posición y orientación conocida. Una vez contando con esta información, se inicia el algoritmo de detección de la superficie

de la pieza y se almacenó la información de la posición y orientación de la superficie de la pieza calculada con respecto al robot. A partir de un conjunto de 100 experimentos con piezas regulares de diversas dimensiones, se obtuvo un error promedio de 4.1 mm, con una desviación estándar de 1.5 mm para posición, para la orientación se obtuvo un error promedio de 0.9 grados, con una desviación estándar de 1.7 grados. Estos errores de posicionamiento y orientación son debidos a la probabilidad de asertividad en que el algoritmo detecta las líneas proyectadas en la pieza, es decir, que con la velocidad de la unidad Pan-Tilt de $0.6^\circ/\text{seg}$ equivalente a una velocidad de 4 mm/seg (de acuerdo a la velocidad de desplazamiento de la línea láser sobre la pieza) es posible detectar líneas al momento que se desplazó la línea láser sobre la pieza, obteniendo esos errores de posicionamiento y orientación de la pieza. En las figuras 6 y 7 se puede observar el resultado del estudio de sensibilidad para una estimación de la matriz de proyección P de 100 muestras.

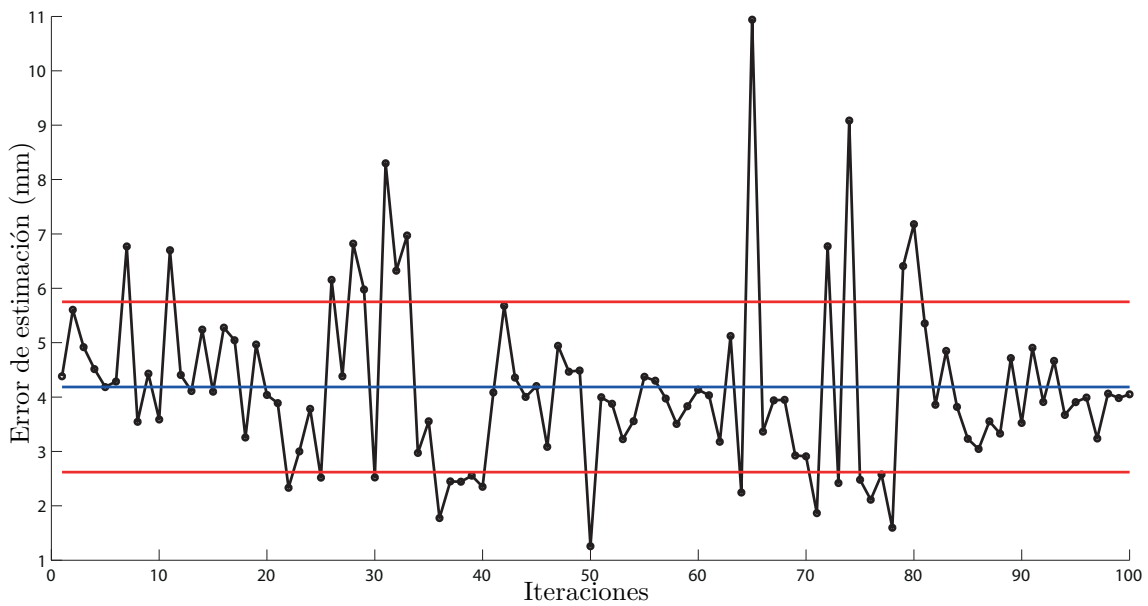


Figura 6: Error de estimación de la ubicación del centro de la superficie de la pieza.

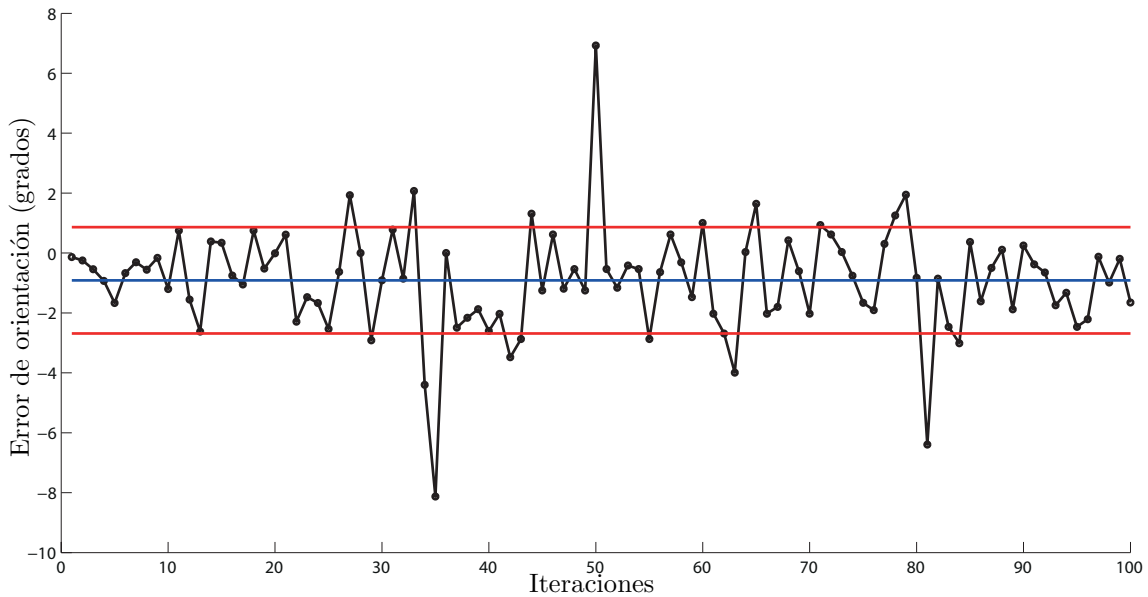


Figura 7: Error de estimación de la orientación de la superficie de la pieza.

Con la información de la posición y orientación de la pieza con respecto al robot, se realiza el cálculo para determinar la configuración de las juntas del robot que permitan tomar la pieza de trabajo. Estos cálculos se ejecutan de forma iterativa y mediante la cinemática del robot, hasta que se converge en el punto objetivo en espacio de cámara. Éste es el centro de la superficie de la pieza del cual se estimó su coordenada 3D del algoritmo de detección de superficies, de éste se obtiene su coordenada en espacio de cámara mediante la solución de $\mathbf{x} = P\mathbf{X}$, mientras que la orientación del robot está dada por el ángulo de inclinación de la pieza. La descripción del algoritmo de posicionamiento del robot se puede observar en el diagrama de flujo de la figura 2.5. En la figura 8 se describe el proceso de convergencia de las juntas del robot.

Con la configuración de las juntas del robot que permiten tomar la pieza detectada, se comanda el robot para que se posicione y sujete la pieza. Una vez esto se

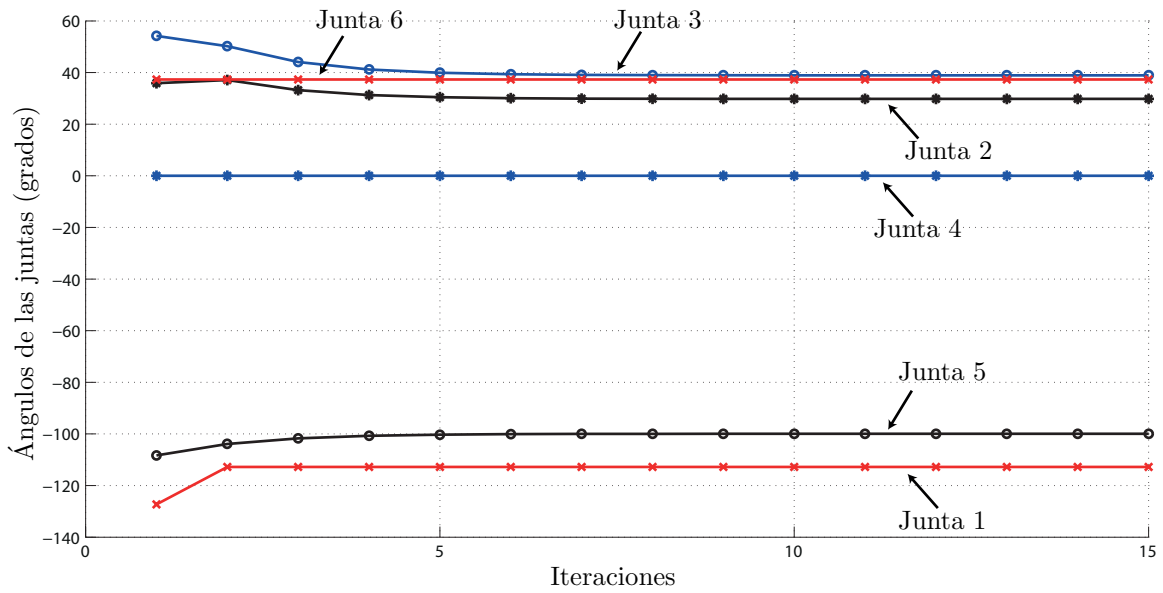


Figura 8: Proceso de convergencia de las juntas del robot mediante el algoritmo de posicionamiento y orientación.

coloca la pieza en una de las estaciones de descarga de piezas. Al finalizar la maniobra el robot, mediante la interfaz de señales I/O, manda una señal al PLC Logo! para desenergizar el solenoide del actuador de la banda, permitiendo el paso al pallet y en espera del siguiente pallet con otra pieza de trabajo. El ciclo de tiempo total desde la detección del pallet a la estación de paro del robot hasta la liberación del mismo fue de 1 minuto con 10 segundos. Los diferentes etapas de la implementación se describen en la figura 9.

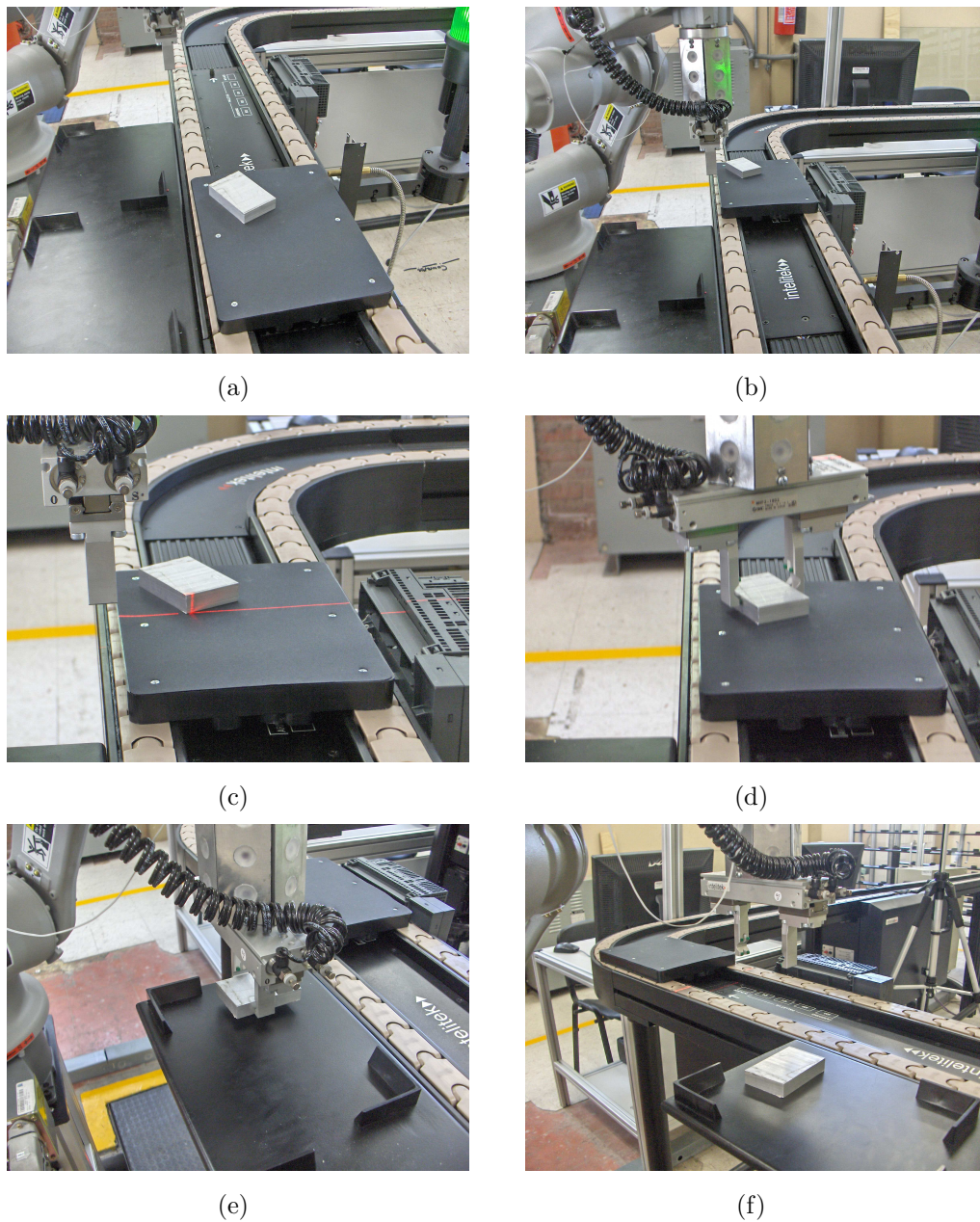


Figura 9: Etapas de la implementación de detección y estiba de piezas en la banda transportadora de la CIM; (a) detección de arribo del pallet a la estación de paro mediante el sensor fotoeléctrico, (b) detención del pallet mediante el pistón de la estación de paro, (c) adquisición de imágenes al momento que se hace el barrido de la línea láser sobre la pieza, (d) una vez finalizado el algoritmo de detección de superficies, se comanda el robot para posicionarse con la orientación adecuada para sujetar la pieza mediante el efector final, (e) colocación de la pieza la estación de descarga de la banda, (f) se libera el pallet de la estación de paro.

Conclusiones

Con el desarrollo del presente trabajo se llegó a la conclusión que, con la metodología de CSM-LCM aplicada en 2 implementaciones diferentes, es posible estimar la posición y estiba de objetos, con la suficiente precisión para el caso de envases de producto que cuentan con un radio de 42 mm en la tapa del mismo, así como para la detección de posición y orientación de piezas de distintas dimensiones de: $100 \times 43 \times 38$ mm, $40 \times 60 \times 12$ mm, $63 \times 52 \times 21$ mm, teniendo en cuenta la apertura posible del efector final de 64 mm.

En la segunda implementación fue posible describir una superficie plana en el espacio mediante las técnicas descritas en la sección 4.3.2, con lo cual es factible detectar la posición y orientación de la pieza de trabajo. Con esta información y utilizando la metodología de LCM-CSM, se determinó la configuración de las juntas del robot que permitan posicionarse y orientarse para sujetar y estibar la pieza. Es importante mencionar que se realizaron los experimentos para distintas dimensiones y materiales como lo son el aluminio y la madera, los cuales tienen distintas propiedades de reflectividad de la luz, obteniendo resultados en la detección de los cuales se presentaron en la sección 4.3.2. Con esto se comprueba la flexibilidad del algoritmo desarrollado, lo cual es una de las características deseadas en las celdas de manufactura inteligentes que utilizan robots.

Existe un problema en la mayoría de los sistemas de visión, el cual está relacionado con la iluminación del ambiente y la reflexión del haz de luz láser proyectado sobre la pieza, es decir, cuando la luz incide sobre un objeto, varía según sea la reflectancia de la superficie y la inclinación de los rayos de luz incidentes. Esto afecta en la adquisición de imágenes, dando como consecuencia una fluctuación en los resultados del procesamiento de las imágenes, como se mencionó en la sección de resultados

de la implementación de detección y estiba de piezas en una celda de manufactura. El efecto de la incidencia de la luz láser sobre la pieza y el procesamiento de las imágenes para adquirir la línea, da como resultado una variación en los parámetros de la pendiente y la ordenada en cada iteración; es posible reducir este error mediante un muestreo más fino, es decir, adquiriendo un mayor número de muestras de las imágenes, esto es posible reduciendo la velocidad de la unidad Pan-Tilt, con lo que el proceso de adquisición de líneas se realizaría más lento pero más preciso en la adquisición de la pieza; es importante mencionar que la utilidad del uso del obturador de la lente de la cámara, ya que esta permite controlar la cantidad de luz que pase por la lente, permitiendo ver con nitidez la iluminación que irradia de los leds y de la proyección del láser. En el caso de la adquisición de los centroides de los leds, el efecto de iluminación no es tan notorio, debido a la cercanía de las cámaras. Pero por el contrario, la adquisición de la línea que se proyecta de la luz láser sobre la superficie de la pieza, requiere de un ajuste de varios parámetros (longitud de línea, rango de ángulos de línea esperado, valores de umbral) que permitan ver de forma adecuada la línea buscada en cuestión. Para esto se realizó una interfaz GUI donde se pueden ajustar los parámetros, como se muestra en el apéndice A2.

Cabe mencionar que existen ventajas para el uso del método de reconstrucción 3D, entre las que destacan: operaciones algebraicas elementales, metodología sencilla de programar, además que los algoritmos requieren de un bajo costo de procesamiento.

En este trabajo se realizó la implementación de un sistema libre de calibración por parte del usuario para su operación, además de autonomía y flexibilidad tanto en el funcionamiento como en la detección de la pieza. De tal forma que el sistema de visión fue capaz de detectar superficies planas de piezas con geometría regular con

diferentes alturas y realizar tareas de manipulación, tales como descarga de la pieza en una estación de paro.

Trabajo a futuro

Como trabajo a futuro se considera aplicar el método de estimación de parámetros de la matriz de proyección P , al inicio y durante el proceso de iterativo de aproximación, lo cual se espera incida directamente en la precisión de la estimación de las coordenadas 3D de los puntos objetivos.

Realizar el proceso de estimación de la posición y orientación de la pieza de trabajo en espacio de cámara, es decir que los puntos que son adquiridos de las líneas proyectadas sobre la pieza y que son detectadas en las imágenes no se determina su coordenadas física, sino que el análisis del ajuste del plano de la superficie de la pieza se realiza en coordenadas de las imágenes.

Elaborar un algoritmo de detección de líneas, a partir del cual sea posible obtener la línea proyectada sobre la pieza en las imágenes, de manera que requiera de un menor procesamiento.

En la implementación de la estiba de envases, es posible automatizar el proceso de localización de la tapa de los envases de los productos, agregando una tercer cámara digital colocada en la parte superior de la superficie de trabajo, en la cual pueda observarse la silueta de la tapa geoméricamente como una circunferencia, a partir de esto se puede posicionar el láser puntual en la tapa del envase y realizar, de manera automática, el algoritmo propuesto de detección y estiba de envases de producto.

Es posible determinar un plano que describa una superficie de la pieza de traba-

jo, sin importar la forma bidimensional de la misma. Con esto es posible contener todos estos puntos a través de la técnica de la envolvente convexa y el algoritmo descrito para determinar el centro de la figura. Utilizando una ventosa neumática como herramienta del robot, puede ser capaz de tomar la pieza, con lo que es posible detectar y estibar un número mayor configuraciones de piezas.

El uso de algoritmos de programación en paralelo permitirían optimizar el uso de memoria de los procesadores multinúcleo, además de ser una opción viable para reducir el error de cálculo derivado de la adquisición de imágenes en el momento que se proyecta la línea láser sobre la pieza, para la detección y orientación de las piezas de trabajo con mayor precisión.

Como se mencionó en la sección de conclusiones, es posible reducir el error de posicionamiento mediante la adquisición de un mayor número de imágenes, lo que permitiría tener un muestreo mayor de líneas procesadas en las imágenes. Esta adquisición se realizaría de dos formas: una reduciendo la velocidad de la unidad Pan-Tilt y otra reduciendo la resolución con las que son adquiridas las imágenes. Es posible además realizar la adquisición de la posición y orientación de la pieza, mediante la proyección de un arreglo matricial de puntos de luz láser sobre la pieza de trabajo.

Bibliografía

- [1] Covarrubias González N.Y., de la Peña Contreras F.A., Cárdenas Galindo J.A., González Galván E. J., “*Un Algoritmo de Visión de Máquina para la Definición Automática de Superficies de Trabajo Planas para Tareas Repetitivas de un Robot Manipulador Industrial*”, VII Congreso Mexicano de Robótica, COMRob. México. 2004.
- [2] G. Reinhart, Jochen Werner, F. Lange, “*Robot based system for the automation of flow assembly lines*”, WGP, Prod. Eng. Res. Devel. (2009) 3:121-126.
- [3] I. López-Juárez, “*On-Line Learning for Robotic Assembly using artificial neural networks and contact force sensing*”, PhD. Thesis, Nottingham Trent University, Inglaterra, 2000.
- [4] I. López-Juárez, M. Howarth. “*Knowledge Acquisition and Learning in Unstructured Robotic Assembly Environments*”, The International Journal of Information Sciences, 2002.
- [5] Mana Saedan, Marcelo H. Ang Jr., “*3D Vision-Based Control On An Industrial Robot*”, IASTED International Conference on Robotics and Applications, Nov 19-22, 2001, Florida, USA, pp. 152-157.

- [6] M. Peña, I López Juárez, J. Corona, K. Ordaz., “*Visión para Robots en Tareas de Ensamble*”, IFAC, 2004.
- [7] http://www.robotics.org/content-detail.cfm/Industrial-Robotics-Feature-Article/Robots-Use-3D-Vision-to-Improve-System-Costs-and-Product-Quality/content_id/1993, Nov 2010.
- [8] Robert M. Loughheed, Robert E. Sampson., “*3-D Imaging Systems and High-Speed Processing for Robot Control*”, Machine Vision and Applications (1988) 1:41-57.
- [9] Mikell P. Groover, Mitchell Weiss, Roger N. Nagel, Nicholas G. Odrey . “*Robotica Industrial. Tecnología, programación y aplicaciones*”, McGrawHill, 1990.
- [10] Aníbal Ollero Baturone. “*Robótica. Manipuladores y robots móviles*”, Alfaomega, marcombo, 2001.
- [11] González Galván, E., “*Introducción al Curso de Robótica y Visión por Computadora. Notas del curso impartido en el CIEP-FI*”, 2009.
- [12] Denavit J., Hartenberg R, “*A kinematic notation for lower-pair mechanisms based on matrices*”, ASME Journal of Applied Mechanics, pp 215-221, 1955.
- [13] Rafael C. Gonzalez, Richard E. Woods, “*Digital Image Processing*”, 3rd Edition, Prentice Hall, 2008.
- [14] J. Canny, “*A Computational Approach to Edge Detection*”, IEEE Trans. Patt. Anal. Machine Intell. 8,679-698, 1986.
- [15] J. Canny, “*A Computational Approach to Edge Detection*”, In Fischler, M.A. and Firschein, O. (Eds.). Readings in Computer, 1987.

- [16] Trucco E. and Verri A., *“Introductory Techniques for 3-D Computer Vision”*, Prentice-Hall, Uppers Saddle River:NJ, 1998.
- [17] Faugeras O., *“Three-dimensional computer vision: a geometric viewpoint”*, MIT Press, 1993.
- [18] Hough P.V.C., *“Method and Means for Recognizing Complex Patterns”*, U.S. Patent 3,069,654, 1962.
- [19] Gonzalo Pajares, Jesus M. de la Cruz, *“Visión por Computador: Imágenes digitales y aplicaciones”*, Alfaomega, 2002.
- [20] Galbiati, L.J., *“Machine Vision and Digital Image Processing Fundamentals”*, Prentice Hall, 1990.
- [21] Seelinger M. *“Point & Click Camera-Space manipulation, mobile camera-space manipulation, and some fundamental issues regarding the control of robots using vision”*, University of Notre Dame, Notre Dame IN, April 1999. Ph.D. Thesis.
- [22] Huadong Sam Chen, *“Error Analysis of Camera-Space Manipulation”*, University of Notre Dame, Notre Dame IN, August 2007. Ph.D. Thesis.
- [23] S.B. Skaar, W. H. Brockman and R. Hason. *“Camera Space Manipulation”*, International Journal of Robotic Research, 6(4):20-32, 1987.
- [24] S. B. Skaar, W. H. Brockman, and W. S. Jang, *“Three dimensional camera space manipulation”*, Int. J. Robot. Res., vol. 9, no. 4, pp. 22–39, 1990.
- [25] Skaar SB, Seelinger MJ, Robinson ML, Gonzalez Galvan EJ. *“Means and method of robot control relative to an arbitrary surface using camera-space manipulation”*, United States Patent #6,304,050.2001

- [26] Marco A. García, Antonio Cárdenas, Juan M. Rendón and Mauro Maya Méndez, “*Una Plataforma de Control Basado en Visión para la Rehabilitación de Robots Manipuladores de Tipo Industrial*”, *Computación y Sistemas* Vol. 12 No. 4, 2009, pp 409-420 ISSN 1405-5546.
- [27] Richard Hartley, Andrew Zisserman, “*Multiple Vision Geometry in computer vision*”, Cambridge University press, second edition, 2003.
- [28] Z. Zhang. “*A flexible new technique for camera calibration*”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1330-1334, 2000.
- [29] Ferrer Gutierrez Vicente, “*Calibración de una cámara por seguimiento de características en una secuencia de imágenes*”, Tesis de Maestria, IPICYT, México, 2004.
- [30] Nassir Navab. “*SVD abstract*” en la documentacion “*3D Computer Vision Script Draft*”, Technische Univertitat Munchen, 2006, pp. 48-51.
- [31] Moons Theo, “*A Guided Tour Through Multiview Relations*”, Katholieke Mercierlaan 94, 3001 Leuven, Belgium, 2000.
- [32] Biao Zhang, “*Three-dimensional laser-assisted image analysis for robotic surface operation with camera-space manipulation*”, PhD. Thesis, Notre Dame, Indiana, 2007.
- [33] Eddie A. Sobrado, “*Sistema de visión artificial para el reconocimiento y manipulación de objetos utilizando un brazo robot*”, Tesis de Maestria, Pontificia Universidad Católica del Perú, Lima, 2003.
- [34] Julio Molleda, “*Técnicas de visión por computador para la reconstrucción en tiempo real de la forma 3d de productos laminados*”, Tesis doctoral, Universidad de Oviedo, Gijón, 2008.

- [35] Brian W. Kernighan, Dennis M. Ritchie, Néstor Gómez Muñoz, *“El lenguaje de programación C”*, 2da Edición, Pearson Educación, 1991.
- [36] <http://opencv.willowgarage.com>, Oct 2009.
- [37] William H. Press, Saul Teukolsky, William Vetterling and Brian Flannery, *“Numerical Recipes in C. The Art of Scientific Computing.”*, 2nd Edition, Cambridge University Press, 1992.
- [38] M.I. Shamos, *“Computational geometry”*, PhD thesis, Yale University, 1978.
- [39] Godfried T. Toussaint, *“Solving geometric problems with the rotating calipers”*, Proc. MELECON '83, Athens, 1983.

Descripción de funciones de los algoritmos programados

Este apéndice se presentan las funciones de los algoritmos del lenguaje de programación de C++, utilizados para la realización del presente trabajo.

A1.1. Funciones de OpenCV

En esta se presentan las funciones más importantes utilizadas para la realización de las implementaciones presentadas anteriormente.

Canny

Implementa el algoritmo de Canny para la detección de bordes en las imágenes.

```
void cvCanny(const CvArr* image, CvArr* edges, double threshold1,  
double threshold2, int aperture_size=3)
```

- `image` - Imagen de entrada en escala de grises.
- `edges` - Imagen donde se almacena los bordes detectados por la función.

- `threshold1` - Primer valor de umbral.
- `threshold2` - Segundo valor de umbral.
- `aperture_size` - Parámetro del operador de Sobel.

Para las variables `threshold1` y `threshold2`, el valor menor de estas se utiliza para definir la unión de borde mientras que el otro valor se utiliza para encontrar los segmentos iniciales de los bordes mejor definidos.

Dilatación

Dilata una imagen mediante el uso de un elemento de estructura.

```
void cvDilate(const CvArr* src, CvArr* dst, IplConvKernel*
element=NULL, int iterations=1)
```

- `src` - Imagen de entrada.
- `dst` - Imagen de salida.
- `element` - Elemento estructurado utilizado para la dilatación. Si el parámetro es NULL, se utiliza una estructura rectangular de 3×3 .
- `iterations` - Número de iteraciones que se aplica la dilatación.

Detección de contornos

Encuentra los contornos de una imagen binaria.

```
int cvFindContours(CvArr* image, CvMemStorage* storage, CvSeq**
first_contour, int header_size = sizeof(CvContour), int
mode = CV_RETR_LIST, int method = CV_CHAIN_APPROX_SIMPLE, CvPoint
offset = cvPoint(0, 0))
```

- `image` - Imagen de entrada.
- `storage` - Variable que almacena los contornos encontrados.
- `first_contour` - Parámetro de salida donde se almacena el apuntador al primer contorno encontrado.
- `mode` - Forma en el que son agrupados los contornos encontrados.
- `method` - Método de aproximación de los contornos.
- `offset` - Desplazamiento de las coordenadas de los contornos encontrados.

Detección de contornos

Encuentra los contornos de una imagen binaria.

```
int cvFindContours(CvArr* image, CvMemStorage* storage, CvSeq**  
first_contour, int header_size = sizeof(CvContour), int mode =  
CV_RETR_LIST, int method=CV_CHAIN_APPROX_SIMPLE, CvPoint offset =  
cvPoint(0, 0))
```

- `image` - Imagen de entrada.
- `storage` - Variable que almacena los contornos encontrados.
- `first_contour` - Parámetro de salida donde se almacena el apuntador al primer contorno encontrado.
- `mode` - Forma en el que son agrupados los contornos encontrados.
- `method` - Método de aproximación de los contornos.
- `offset` - Desplazamiento de las coordenadas de los contornos encontrados.

A1.1.1. Detección de líneas

Detecta líneas en una imagen binaria utilizando la transformada de Hough.

```
CvSeq* cvHoughLines2(CvArr* image, void* storage, int method, double rho, double theta, int threshold, double param1=0, double param2=0)
```

- `image` - Imagen binaria de entrada.
- `storage` - Variable que almacena las líneas encontrados.
- `method` - Método utilizado de las diferentes variantes de la transformada de Hough (método estándar, probabilístico y variante del método estándar).
- `rho` - Distancia de relación entre la unidad y la resolución en pixeles .
- `theta` - Resolución del ángulo en radianes.
- `threshold` - Parámetro de umbral. En el que una línea es devuelta por la función si el valor obtenido de la línea es mayor que el parámetro de umbral.
- `param1` - Parámetro 1 de acuerdo con el método utilizado.
- `param2` - Parámetro 2 de acuerdo con el método utilizado.

A1.2. Funciones realizadas para las implementaciones

En esta sección se describen las funciones que se realizaron para las implementaciones citadas en el presente trabajo. A continuación se muestran de acuerdo al archivo de encabezado de la definición de estas funciones, cada una de estas se agruparon de tal forma que describieran el control de uno de los dispositivos utilizados o de un algoritmo de cálculo.

A1.2.1. robot.h

Funciones de control del robot manipulador Motoman HP3.

Inicialización del robot

Esta realiza la inicialización de los parámetros de comunicación del robot, devolviendo: 0 si la inicialización fue correcta, 1 si se realizó un error en la comunicación, 2 si el control del robot no se encuentra en modo “remoto”.

```
short IniRobot(short nCid, HWND hWnd)
```

- nCid - Numero de identificación del robot.
- hWnd - Manejador de ventana, para uso interno de las funciones propias del robot.

Finalización del robot

Realiza el proceso de desconexión de comunicación entre la computadora y el robot.

```
int DescRobot(short nCid)
```

- nCid - Numero de identificación del robot.

Espera que finalice la maniobra del robot

Monitorea la posición actual del robot hasta que este llegue a la posición o configuración objetivo.

```
int EsperaRobot(short nCid, double* pos_objetivo, bool config_XYZ)
```

- nCid - Numero de identificación del robot.
- pos_objetivo - Posición o configuración objetivo.
- config_XYZ - Determina si la posición objetivo introducida esta en coordenadas cartesianas (X, Y, Z) o en configuración de las juntas del robot en pulsos.

A1.2.2. camara.h

Funciones de control de las cámaras digitales uEye.

Inicialización de la cámara

Realiza la inicialización de los parámetros de la cámara. Esta función regresa el manejador de la cámara inicializada.

```
HIDS IniCamara(HIDS hCam, char** pcImgMem, int* id, int NoImgBuff)
```

- hCam - Manejador de cámara.
- pcImgMem - Guarda las direcciones de memoria donde se almacenan las imágenes.
- id - Números de identificación de las direcciones de memoria donde se almacenan las imágenes.
- NoImgBuff - Tamaño del buffer de imágenes a almacenar.

Captura de imagen

Adquiere una imagen y la almacena en la dirección de memoria indicada.

```
IplImage* CapImagen(HIDS hCam, char* pcImgMem, int id)
```

- hCam - Manejador de cámara.
- pcImgMem - Dirección de memoria donde se almacena la imagen.
- id - Número de identificación de la dirección de memoria donde se almacena la imagen capturada.

Manejo de imagen capturada

Permite establecer la imagen en cuestion en la memoria activa para su uso.

```
IplImage* CargarImagen(HIDS hCam, char* pcImgMem, int id)
```

- `hCam` - Manejador de cámara.
- `pcImgMem` - Dirección de memoria donde se almaceno la imagen.
- `id` - Número de identificación de la dirección de memoria donde se almaceno la imagen capturada.

Finalización de la cámara

Realiza la finalización de los parámetros de la cámara.

```
BOOL DescCamara(HIDS hCam, char** pcImgMem, int* id, int NoImgBuff)
```

- `hCam` - Manejador de cámara.
- `pcImgMem` - Dirección de memoria donde se almacenaron las imágenes.
- `id` - Número de identificación de las direcciones de memoria donde se almacenaron las imágenes capturadas.
- `NoImgBuff` - Tamaño del buffer de las imágenes capturadas.

A1.2.3. `pan tilt.h`

Funciones de control de la unidad Pan-Tilt.

Inicialización del Pan-Tilt

Realiza la inicialización de los parámetros del puerto serial para entablar comunicación con el Pan-Tilt. Retornando la variable del manejador de comunicaciones.

```
HANDLE IniPantilt(HANDLE hPantilt)
```

- `hPantilt` - Manejador de comunicaciones.

Control de posición y velocidad

Permite comandar operaciones de control básicas al Pan-Tilt, estas son el posicionamiento y la velocidad de los ejes Pan y Tilt.

```
BOOL operacion( HANDLE hPantilt, char eje, char propiedad_cinematica,
int valor)
```

- `hPantilt` - Manejador de comunicaciones.
- `eje` - Este se trata si la acción a realizar será del Pan o del Tilt.
- `propiedad_cinematica` - Operación a realizar.
- `valor` - Valor de la operación a realizar.

Obtención de la posición actual

Adquiere la posición actual de la unidad de los ejes Pan y Tilt.

```
int posicion_actual(HANDLE hPantilt, char eje)
```

- `hPantilt` - Manejador de comunicaciones.
- `eje` - Este se trata si la acción a monitorear es del eje Pan o del Tilt.

Finalización de comunicación

Realiza la terminación de comunicaciones por el puerto serial.

```
int DescPantilt(HANDLE hPantilt)
```

- `hPantilt` - Manejador de comunicaciones.

A1.2.4. `loc_centroides_imagen.h`

Función de detección de las coordenadas del o los centroides de las marcas visuales en la imagen.

```
int localiza_centroide(short nCid, BOOL enciende_todos_los_leds ,
```

```
float* xc1 , float* yc1, float* xc2 , float* yc2, char  
*muestra_imagen = "no")
```

- nCid - Numero de identificación del robot.
- enciende_todos_los_leds - Si este es verdadero enciende todas las marcas visuales del efector final del robot, de caso contrario solo enciende una marca visual.
- xc1 - Obtiene la(s) coordenada(s) x_c en pixeles del o los centroides en la imagen de la cámara 1.
- yc1 - Obtiene la(s) coordenada(s) y_c en pixeles del o los centroides en la imagen de la cámara 1.
- xc2 - Obtiene la(s) coordenada(s) x_c en pixeles del o los centroides en la imagen de la cámara 2.
- yc2 - Obtiene la(s) coordenada(s) y_c en pixeles del o los centroides en la imagen de la cámara 2.
- muestra_imagen - Establece si se muestran o no los resultados del procesamiento de las imágenes en ventanas.

A1.2.5. encuentra_puntos_pieza.h

Permite encontrar las coordenadas de los puntos de inicio y fin de la línea que mejor se ajuste a la línea láser proyectada sobre la pieza.

```
int encuentra_puntos_pieza( short nCid, int pos_ini_pantilt, int  
pos_fin_pantilt, float *xc_1, float *yc_1, float *xc_2, float *yc_2,  
int *num_puntos_encontrados, bool guarda_imagenes = false)
```

- nCid - Numero de identificación del robot.

- `pos_ini_pantilt` - Posición de inicio para el barrido de la pieza con el láser.
- `pos_fin_pantilt` - Posición final del barrido de la pieza con el láser.
- `xc_1` - Guarda las coordenadas x_c en pixeles del punto de la línea en la imagen de la cámara 1.
- `yc_1` - Guarda las coordenadas y_c en pixeles del punto de la línea en la imagen de la cámara 1.
- `xc_2` - Guarda las coordenadas x_c en pixeles del punto de la línea en la imagen de la cámara 2.
- `yc_2` - Guarda las coordenadas y_c en pixeles del punto de la línea en la imagen de la cámara 2.
- `num_puntos_encontrados` - Cantidad de puntos encontrados en las imágenes.
- `guarda_imagenes` - Permite guardar las imágenes procesadas en disco para su interpretación posterior.

A1.2.6. `calcula_config_robot.h`

Realiza los cálculos para encontrar la configuración del robot pulsos de cada junta que permita posicionarse y orientarse en una coordenada XYZ y orientación calculada previamente por la función.

```
int calcula_config_robot( double *config_objetivo_robot_XYZ, double
*config_actual_robot_pulsos, float dist_herramienta, bool
escribe_datos )
```

- `config_objetivo_robot_XYZ` - Configuración objetivo en coordenadas cartesianas XYZ del robot.

Apéndice A1. Descripción de funciones de los algoritmos programados

- `config_actual_robot_pulsos` - Configuración actual del robot en pulsos de las juntas del robot.
- `dist_herramienta` - Distancia en el eje Z desde el último eslabón del robot hasta la punta del efector final.
- `escribe_datos` - Guarda los resultados de convergencia de las juntas del robot al punto objetivo.

Interfaz Gráfica de Usuario de detección de líneas

Se desarrolló una Interfaz Gráfica de Usuario (GUI), la cual permite detectar y mostrar en tiempo real líneas que se ajustan a la proyectada sobre la pieza. Esta GUI permite modificar los parámetros de: umbral #1, umbral # 2, longitud de línea, distancia entre segmentos, theta mínima y theta máxima, mediante una barra de desplazamiento horizontal, en la cual el valor modificado realiza un nuevo procesamiento mostrando el resultado del cambio de la variable.

Los parámetros de “umbral #1” y “umbral # 2” se refieren a los valores de umbral `threshold1` y `threshold2` de la función de *OpenCV* `cvCanny`; las variables de “longitud de línea” y “distancia entre segmentos” se utilizan en la función de *OpenCV* `cvHoughLines2` estos parámetros corresponden al método `CV_Hough_PROBABILISTIC` de la misma función, la primer variable determinar la longitud de línea mínima a encontrar, mientras que la segunda variable permite determinar la distancia máxima que pueda existir entre segmentos que conformen a una línea; por último los parámetros de “theta mínima” y “theta máxima” corresponden a las variables del algoritmo programado para detectar líneas que se encuentren en ese rango de ángulos. En la

figura A2.1 puede observarse las 2 ventanas de la GUI en la cual en una de ellas se encuentran los parámetros que pueden ser modificados y donde se muestra el resultado de las líneas encontradas, mientras que en la otra ventana se observa el resultado de la función `cvCanny`.

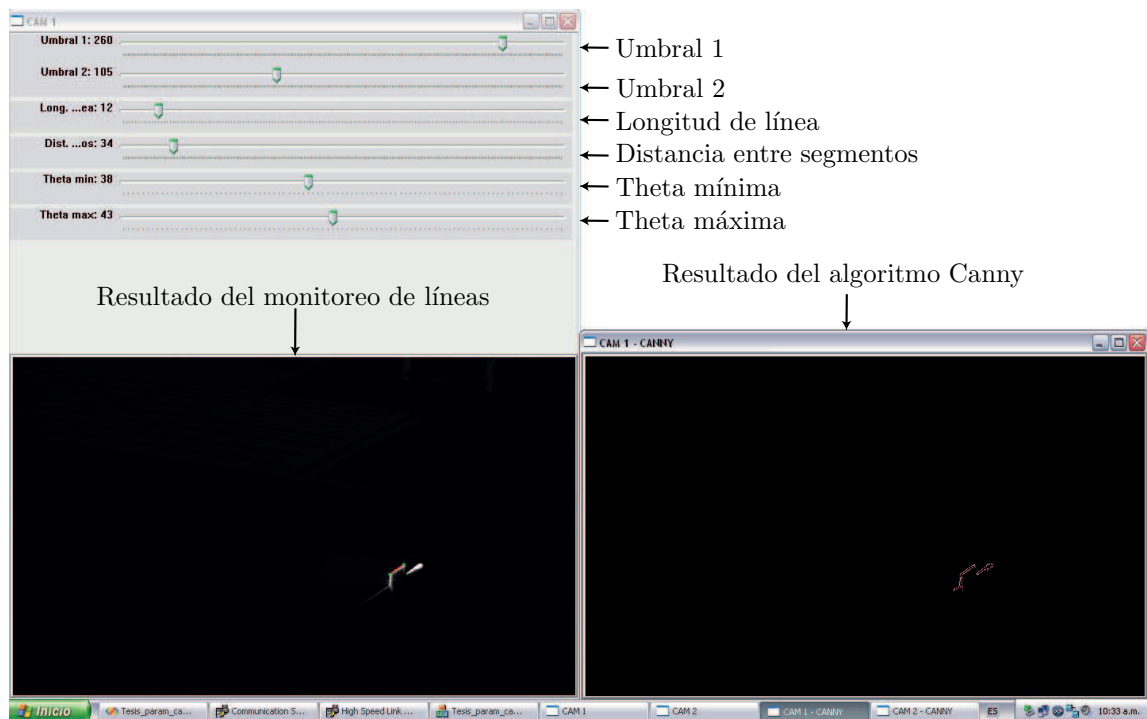


Figura A2.1: Interfaz Gráfica de Usuario de detección de líneas.

Planos de diseño del soporte de marcas visuales

En el presente apéndice se muestran los planos del diseño realizado para el soporte de marcas visuales, el cual fue ensamblado entre el último eslabón del robot y el actuador neumático (gripper). El diseño fue realizado para fabricarse en aluminio, esto debido a que se requiere que el soporte se ligero por la restricción de carga útil del robot de 3 kg.; el soporte esta diseñado de 2 tipos de placas formando un prisma hexagonal como se muestra en la figura [A3.1](#), el ensamble de este se realiza por medio de tornillos. A continuación se presentan los planos de las piezas.

Apéndice A3. Planos de diseño del soporte de marcas visuales

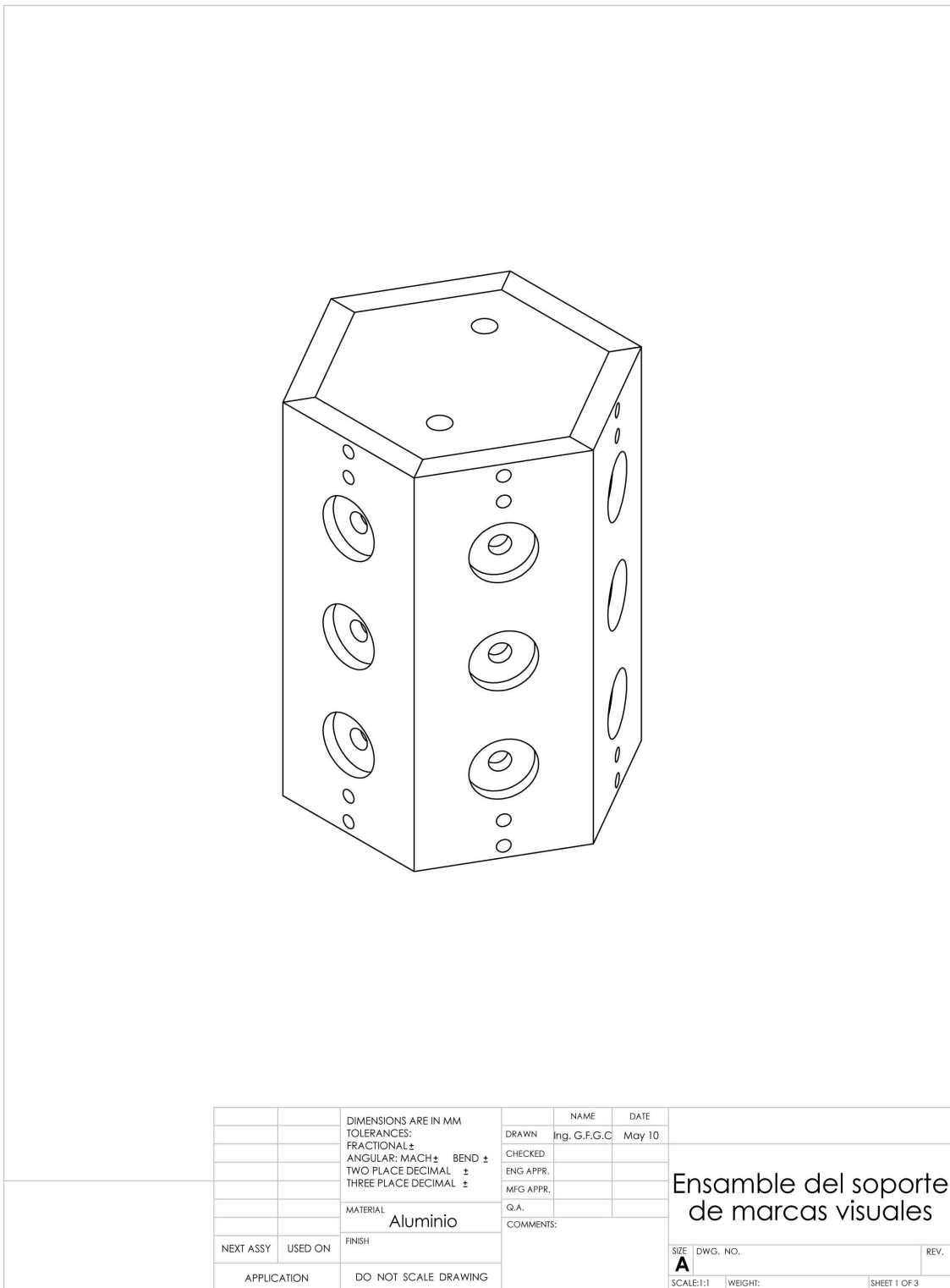
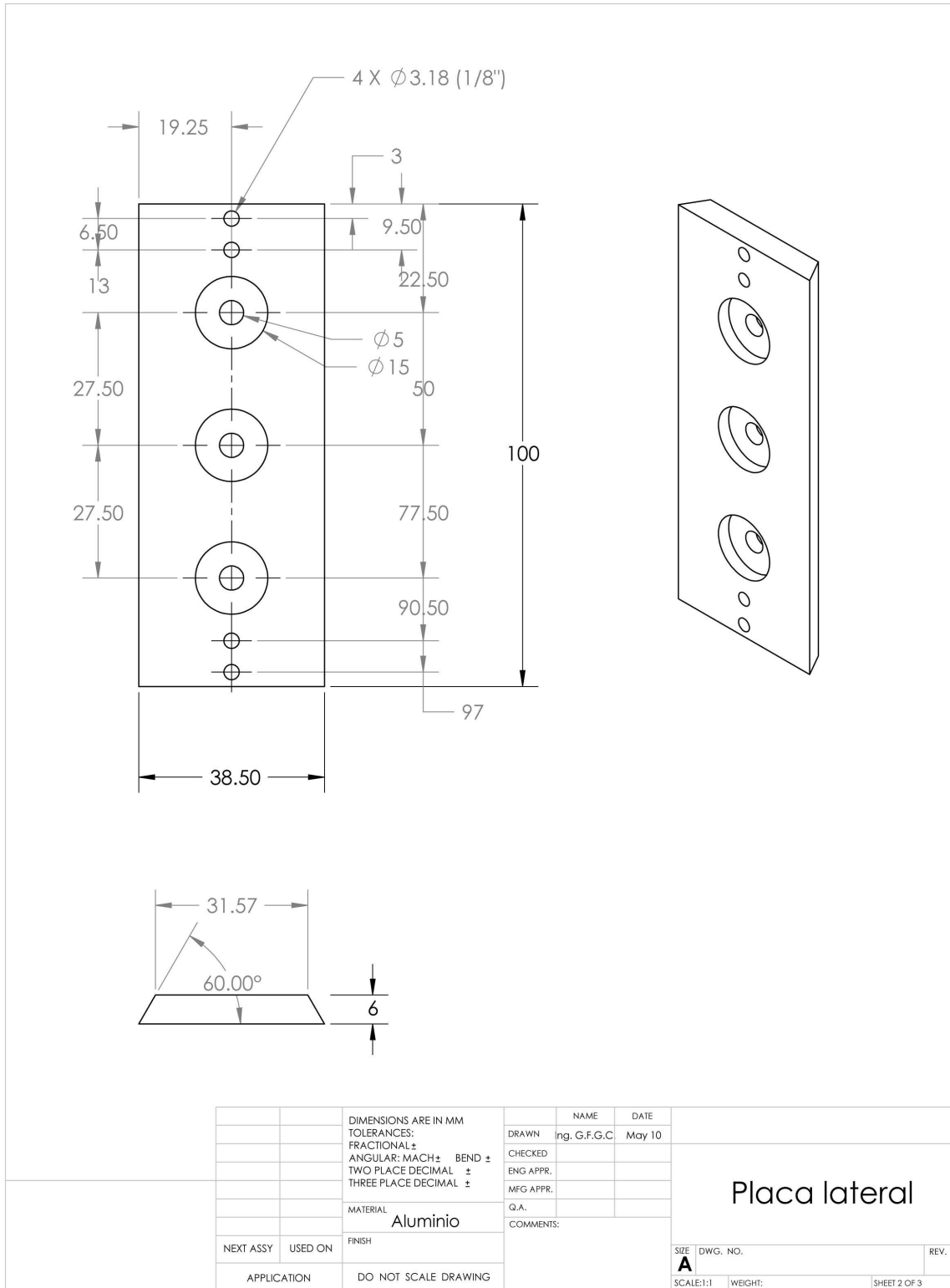


Figura A3.1: Ensamble del soporte de marcas visuales.

Apéndice A3. Planos de diseño del soporte de marcas visuales



Apéndice A3. Planos de diseño del soporte de marcas visuales

