



UNIVERSIDAD AUTÓNOMA DE SAN LUÍS POTOSÍ

**FACULTAD DE INGENIERÍA
CENTRO DE INVESTIGACIÓN Y ESTUDIOS DE POSGRADO**

**“ESTRATEGIA DE CONTROL BASADA EN VISIÓN
PARA ROBOTS MANIPULADORES CON UN MODELO
LINEAL DE CÁMARA Y EL MÉTODO DE
MANIPULACIÓN EN ESPACIO DE CÁMARA”**

T E S I S

QUE PARA OBTENER EL GRADO DE:

**DOCTOR EN INGENIERÍA ELÉCTRICA
OPCIÓN: CONTROL AUTOMÁTICO**

P R E S E N T A

M.I. MARCO ALONSO GARCÍA ROMERO

ASESOR:

DR. JUAN ANTONIO CÁRDENAS GALINDO

San Luis Potosí, S.L.P.

Julio 2016





UNIVERSIDAD AUTÓNOMA DE SAN LUÍS POTOSÍ



FACULTAD DE INGENIERÍA
CENTRO DE INVESTIGACIÓN Y ESTUDIOS DE POSGRADO

“ESTRATEGIA DE CONTROL BASADA EN VISIÓN PARA
ROBOTS MANIPULADORES CON UN MODELO LINEAL
DE CÁMARA Y EL MÉTODO DE MANIPULACIÓN EN
ESPACIO DE CÁMARA”

Presenta:

M.I. Marco Alonso García Romero

Comité

Dr. Juan Antonio Cárdenas Galindo
(Asesor de tesis)

Dr. Carlos Daniel García Beltrán

Dr. Emilio Jorge González Galván

Dr. Raúl Eduardo Balderas Navarro

Dr. Ciro Alberto Núñez Gutierrez

A mi esposa Elvira y mi hijo Marco Tadeo

Agradecimientos

A Dios por darme la oportunidad de concluir esta etapa tan importante en mi vida.

A mis padres María del Refugio y Jaime porque me han apoyado de manera incondicional durante todas las etapas de mi vida.

A mi esposa Elvira y mi hijo Marco Tadeo que han sido mi motor, y que sin ellos no hubiera podido culminar este trabajo.

A mis hermanos César e Iván que siempre han estado cuando los he necesitado.

A mi asesor el Dr. Juan Antonio Cárdenas Galindo por su paciencia y apoyo y por ser un excelente guía durante el desarrollo de esta tesis.

A los miembros del comité revisor: Dr. Emilio Jorge González Galván, Dr. Raúl Balderas Navarro, Dr. Ciro Alberto Núñez Gutiérrez, Dr. Carlos García Beltrán y Dr. Mauro Maya Méndez por sus atinadas observaciones que contribuyeron a mejorar este trabajo.

A la Sra. Chela por ayudarme siempre en trámites importantes y por brindarme su amistad.

A mis compañeros y amigos Raúl, Matus, Priscila, Eder, Efraín, por su apoyo y por compartir tantos momentos tan especiales.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por otorgarme la beca de estudios.

Resumen

En este trabajo se presenta una metodología de control basado en visión para robots manipuladores fundamentada en el método de manipulación en espacio de cámara (CSM por sus siglas en inglés). El método CSM emplea seis parámetros de visión que permiten obtener una aproximación al modelo ortográfico de cámara, los cuales posteriormente son actualizados acorde al modelo de perspectiva mediante un proceso conocido como *flattening*. Dichos parámetros son obtenidos mediante un proceso iterativo de estimación no lineal. En esta tesis se propone sustituir esos seis parámetros por un modelo lineal de cámara que consta de once parámetros. Lo anterior permite obtener el modelo de cámara de orificio en un solo proceso, de una manera más rápida y, al ser un modelo lineal, no presenta problemas de convergencia en la solución, además de que permite obtener el mismo error de posicionamiento del robot que el método original.

Se presenta también un análisis de sensibilidad a errores en los datos, que permite visualizar como es el comportamiento del modelo cuando se presentan errores aleatorios en la captura de la información visual. Se incluyen además resultados experimentales en tres diferentes plataformas de prueba que muestran la eficacia de la metodología presentada en este trabajo.

Índice general

Índice de figuras	xv
Índice de cuadros	xxi
Introducción	1
1 Motivación y estado del Arte	5
1.1 Contexto y estado del arte	6
1.2 El método de manipulación en espacio de cámara	10
1.2.1 Modelo ortográfico	11
1.2.2 Corrección por <i>flattening</i>	12
1.2.3 Algoritmo CSM	13
1.2.4 Configuración angular del manipulador	14
1.3 Motivación	15
1.4 Objetivos y alcances	15
1.5 Organización de la tesis	16
2 Desarrollo de un control basado en visión de un manipulador robótico utilizando un modelo lineal de cámara	17
2.1 El modelo de cámara de orificio	18
2.1.1 Parámetros intrínsecos	19
2.1.2 Parámetros extrínsecos	22
2.2 Metodología para la estimación de los parámetros de visión	24
2.2.1 Sistema de ecuaciones lineales	24
2.2.2 Descomposición en valores singulares	26
2.3 Correspondencia entre los parámetros del modelo lineal y el modelo de CSM	28

2.4	Metodología CSM con un modelo lineal de cámara	31
2.4.1	Proceso de estimación de las matrices de proyección	31
2.4.2	Estimación de la configuración angular del manipulador	34
2.4.3	Caracterización de la superficie	35
3	Análisis de sensibilidad del modelo lineal de cámara y la metodología de manipulación en espacio de cámara	41
3.1	Plataforma de pruebas para la simulación	41
3.1.1	Modelo de Cámara	42
3.1.2	Modelo cinemático nominal	44
3.1.3	Validación de la plataforma de simulación	44
3.2	Sensibilidad del modelo de cámara a errores presentes en las mediciones .	47
3.2.1	Simulación de errores con una distribución normal	47
3.2.2	Simulación de errores con una distribución uniforme	57
3.3	Sensibilidad de posicionamiento del robot a errores presentes en las mediciones	58
3.3.1	Simulación de errores con una distribución normal	60
3.3.2	Simulación de errores con una distribución uniforme	64
3.4	Variación del número de muestras y su efecto en el error de posicionamiento del manipulador	67
3.5	Efecto de la resolución en la detección de los datos en espacio de cámara .	70
3.6	Efecto de las muestras tomadas durante la maniobra del manipulador . . .	72
3.6.1	Mayor resolución en las detecciones en espacio de cámara	73
4	Resultados Experimentales	75
4.1	Plataformas de pruebas	75
4.1.1	Plataforma 1 basada en un robot PUMA 761	76
4.1.2	Plataforma 2 basada en un robot Motoman HP3	76
4.1.3	Plataforma 3 basada en un robot paralelo tipo delta	78
4.2	Resultados experimentales en la plataforma 1	79
4.2.1	Errores en el modelo de cámara	80
4.2.2	Píxeles cuadrados vs rectangulares	81
4.2.3	Normalización de las muestras	83
4.2.4	Errores debidos al número de muestras	84

4.2.5	Cálculo de matrices de proyección de cámara (mínimos cuadrados vs DVS)	85
4.2.6	Costo computacional de los modelos	87
4.2.7	Error de posicionamiento del Robot	88
4.3	Resultados exprimentales en la plataforma 2	90
4.3.1	Interfaz gráfica intuitiva para la manipulación de objetos	90
4.3.2	Detección automática de arandelas por medio de visión computacional	92
4.3.3	Descripción general del algoritmo de detección de objetos	93
4.3.4	Trazado del controrno	94
4.3.5	Dirección de búsqueda del contorno	94
4.3.6	Resultados	96
4.4	Resultados experimentales en la plataforma 3	97
Conclusiones y trabajo futuro		101
Anexos		107
A Modelo Cinemático directo del manipulador		107
Bibliografía		111

Índice de figuras

1.1	Sistema coordenados de referencia del método CSM.	13
2.1	Geometría de la cámara de orificio. C representa el centro de la cámara, p el punto principal. El centro de la cámara se localiza en el origen del sistema coordenado.	18
2.2	Proyección de un punto tridimensional sobre el plano YZ del sistema coordenado tridimensional.	19
2.3	Transformación Euclidiana entre el sistema coordenado tridimensional y el sistema coordenado unido a la cámara.	22
2.4	Proceso para la obtención de la matriz de proyección de cámara.	32
2.5	Marcas visuales distintivas colocadas en el efector final del manipulador. .	33
2.6	Sistemas coordenados involucrados en el método CSM.	33
2.7	Laser matricial proyectado sobre la superficie de trabajo del robot.	36
2.8	Caracterización de la superficie de trabajo del robot a partir de las matrices de proyección de cámara.	38
2.9	Proceso para calcular la posición y orientación del manipulador	39
3.1	Proceso para generar información a partir de la plataforma de simulación.	44
3.2	Datos generados en la plataforma de simulación vs proyecciones usando trayectoria pre-planeada para cámara 1 y 2 respectivamente.	46
3.3	Diferentes configuraciones de datos utilizadas para el análisis de sensibilidad del modelo lineal.	48
3.4	Mediciones vs proyecciones aplicando un error con una distribución normal con una media de cero y una desviación estándar de (a)1 pixel, (b)5 píxeles.	51

3.5	Muestras contenidas en 4 planos utilizados para el análisis de sensibilidad en la plataforma de simulación (1828 datos).	52
3.6	Errores obtenidos aplicando una simulación de Montecarlo al modelo lineal de cámara empleando 1828 datos, a los cuales se les agregó ruido aleatorio utilizando una distribución normal con una media de cero y una desviación estándar de 1 pixel.	52
3.7	Magnitud del error de estimación correspondiente al plano 1 de la plataforma de simulación.	53
3.8	Magnitud del error de estimación correspondiente al plano 2 de la plataforma de simulación.	54
3.9	Magnitud del error de estimación correspondiente al plano 3 de la plataforma de simulación.	55
3.10	Magnitud del error de estimación correspondiente al plano 4 de la plataforma de simulación.	56
3.11	Error de estimación del modelo de cámara lineal utilizando datos simulados a los que se les inyectó error aleatorio con una distribución normal, con una media de cero píxeles y una desviación estándar de 0.2, 0.5, 1 y 2 píxeles.	57
3.12	Desviación estándar del modelo de cámara lineal utilizando datos simulados a los que se les inyectó error aleatorio con una distribución normal, con una media de cero píxeles y una desviación estándar de 0.2, 0.5, 1 y 2 píxeles.	58
3.13	Error aleatorio inyectado a los datos, correspondiente a una distribución normal con una media de cero y una desviación estándar de 1 pixel. . . .	59
3.14	Error aleatorio generado, correspondiente a una distribución uniforme con valores comprendidos entre 1 y -1 píxeles.	59
3.15	Errores promedio obtenidos aplicando una simulación de Montecarlo al modelo lineal de cámara utilizando 1828 puntos, a los que les fue agregado ruido aleatorio con una distribución uniforme con valores comprendidos entre -1 y 1 píxel.	60
3.16	Error de estimación del modelo de cámara lineal utilizando datos simulados a los que se les inyectó error aleatorio con una distribución uniforme con valores comprendidos entre -0.2 y 0.2 ; -0.5 y 0.5 ; -1 y 1 ; -2 y 2 píxeles.	61

3.17	Desviación estándar del modelo de cámara lineal utilizando datos simulados a los que se les inyectó error aleatorio con una distribución uniforme con valores comprendidos entre -0.2 y 0.2 ; -0.5 y 0.5 ; -1 y 1 ; -2 y 2 píxeles.	62
3.18	Errores promedio de posicionamiento del robot utilizando datos simulados a los que se les inyectó error aleatorio con una distribución normal y una media de cero píxeles para una desviación estándar de 0.2 , 0.5 , 1 y 2 píxeles.	63
3.19	Errores de posicionamiento del robot utilizando datos simulados. Se emplearon 1828 datos a lo que se les inyectó error aleatorio con una media de cero y una desviación estándar de 1 pixel.	64
3.20	Diagrama de caja del error de posicionamiento del robot utilizando ruido inyectado con una distribución normal.	65
3.21	Histograma del error de posicionamiento del robot utilizando ruido inyectado con una distribución normal.	65
3.22	Errores promedio de posicionamiento del robot utilizando datos simulados a los que se les inyectó error aleatorio con una distribución uniforme con valores comprendidos entre -0.2 y 0.2 ; -0.5 y 0.5 ; -1 y 1 ; -2 y 2 píxeles.	66
3.23	Errores de posicionamiento del robot utilizando datos simulados. Se emplearon 1828 datos a lo que se les inyectó error con una distribución uniforme de -1 a 1 pixel.	67
3.24	Diagrama de caja del error de posicionamiento del robot utilizando ruido inyectado con una distribución uniforme.	68
3.25	Histograma del error de posicionamiento del robot utilizando ruido inyectado con una distribución uniforme.	68
3.26	Análisis del error del modelo lineal a la variación en el número de muestras cercanas al punto objetivo para el caso de 315 datos.	69
3.27	Análisis del error del modelo lineal a la variación en el número de muestras cercanas al punto objetivo para el caso de 1828 datos.	70
3.28	Análisis de sensibilidad al número de muestras aplicando un error aleatorio con una magnitud de -0.25 a 0.25 píxeles para el caso de 300 datos. .	71
3.29	Análisis de sensibilidad al número de muestras aplicando un error aleatorio con una magnitud de -0.25 a 0.25 píxeles para el caso de 1800 datos.	72
3.30	Efecto de las muestras tomadas durante la maniobra de posicionamiento. .	73

3.31	Efecto de una mayor resolución en la detección de las muestras tomadas durante la maniobra de posicionamiento.	74
4.1	Robot PUMA 761 con su computadora de control	76
4.2	Arquitectura del hardware de la plataforma de control basado en visión para el robot PUMA 761.	77
4.3	Plataforma de pruebas experimentales utilizada para validar los resultados obtenidos. (1) Robot Motoman HP3, (2) Unidad pan-tilt donde está montado el láser, (3) Cámaras uEye montadas en tripies.	77
4.4	Detalle de los componentes de la plataforma de visión. (a) Robot Motoman HP3. (b) Unidad pan-tilt.	78
4.5	Plataforma de pruebas basada en un robot paralelo tipo delta Parallax LKF-2040	79
4.6	Comportamiento del error en función de la distancia entre el origen del sistema coordenado 3D y y el centroide del área que abarcan los datos usados para estimar el modelo lineal empleando muestras locales.	84
4.7	Error de posicionamiento del robot vs número de muestras.	86
4.8	Valores promedio de error de estimación del modelo de cámara. Método 1: Mínimos cuadrados. Método 2: Descomposición en valores singulares.	87
4.9	Error de posicionamiento del robot para 100 maniobras utilizando el método OF-CSM	89
4.10	Error de posicionamiento del robot para 100 maniobras utilizando el método LCM-CSM	89
4.11	Interfaz Gráfica Intuitiva para la selección manual de puntos objetivo	91
4.12	Ampliación del área de imagen para la selección del punto objetivo.	91
4.13	Selección múltiple de puntos objetivo. a) Selección múltiple. b) Ajuste de líneas	92
4.14	Arandela de 1 mm de espesor, utilizada para la tarea de posicionamiento del robot.	92
4.15	Partes principales que definen el algoritmo de detección de objetos basado en rastreo de contornos.	93
4.16	Búsqueda del contorno de un objeto en la imagen	94
4.17	Índices asignados a los valores colindantes de un punto que compone un contorno.	95

4.18	Detección de arandelas de diferentes diámetros colocadas sobre la superficie de trabajo del robot.	96
4.19	Detección de objetos con geometrías distintas.	96
4.20	Resultado de 10 maniobras para tomar una arandela colocada de manera arbitraria.	97
4.21	Ejemplo de la maniobra de posicionamiento del robot para tomar una arandela de 1mm de espesor, usando la plataforma de control basado en visión.	98
4.22	Error de posicionamiento del robot para 50 maniobras	98
A.1	Robot manipulador motoman HP3	109

Índice de cuadros

3.1	Datos arrojados del análisis de sensibilidad a errores en las proyecciones en espacio de cámara del modelo lineal.	50
4.1	Errores en el modelo de cámara	81
4.2	Muestras locales versus no locales en píxeles rectangulares	82
4.3	Muestras locales versus no locales en píxeles cuadrados	82
4.4	Error promedio y desviación estándar utilizando muestras locales normalizadas	83
4.5	Comparativa de tiempos requeridos para calcular el modelo de cámara usando tres diferentes metodologías	88
A.1	Parámetros <i>Denavit-Hartenberg</i> para el Robot Motoman HP3	109

Introducción

En la actualidad el uso de visión computacional se ha extendido de una manera notable, esto en gran medida impulsado por los avances tecnológicos en sensores y el desarrollo de nuevas tecnologías que han permitido abaratar la producción de estos dispositivos. En este sentido, una imagen puede proporcionar una enorme cantidad de información, sin embargo sigue siendo un reto el poder aislar la información de interés y poder interpretarla acorde a la tarea que se desea realizar.

Tradicionalmente los robots manipuladores se han utilizado en tareas repetitivas donde el operador programa una maniobra y el robot se encarga de repetirla constantemente. El añadir un sensor de visión permite aprovechar las capacidades de los robots de ser reprogramados en tiempo real, además de que los vuelve más versátiles en el sentido de que pueden adaptarse a entornos cambiantes.

Una de las principales ventajas de utilizar un sistema robótico de control basado en visión es que proporciona un sistema adaptable a la posición de la pieza de trabajo. Al ser un sistema adaptable elimina la necesidad de que la pieza de trabajo permanezca fija o en una posición conocida respecto al origen del robot, aunado a que explota las capacidades de los robots de ser reprogramados de manera automática durante una maniobra.

El método de manipulación en espacio de cámara es una técnica que usa visión por computadora para el control de movimientos de robots manipuladores, una de sus características principales es que no requiere de un proceso de calibración ni del manipulador ni del sistema de visión. Este método está basado en la idea de que la relación tridimensional espacial entre el efector final del manipulador y la pieza de trabajo puede ser representado por una relación geométrica entre dos cuerpos tal y como se presenta en el plano de imagen en dos cámaras distintas. La relación entre el posicionamiento en el espacio real y las imágenes tomadas está representada por seis parámetros de visión, los cuales son obtenidos mediante un proceso de estimación no lineal. Estos parámetros definen una relación algebraica entre la configuración interna del robot y la localización correspondiente en el

espacio de la imagen, de un cierto número de señales visuales, las cuales están colocadas sobre el efector final del robot, y están relacionadas por medio del modelo cinemático nominal del manipulador y un modelo de cámara. Para realizar una maniobra de posicionamiento mediante este método es necesario el uso de dos o más cámaras para lograr el correcto posicionamiento del manipulador respecto a la pieza de trabajo.

El algoritmo del método de manipulación en espacio de cámara utiliza dos etapas para calcular el modelo de cámara de orificio. La primera etapa consiste en obtener una estimación de un modelo ortográfico de cámara, y la segunda consiste en actualizar los datos para añadir el efecto de perspectiva al modelo, este proceso se conoce como *flattening*. La primera etapa, al utilizar un modelo ortográfico, tiene como requerimiento que las cámaras deben estar alejadas del espacio de trabajo del robot, con el fin de minimizar el efecto de perspectiva. La segunda etapa (*flattening*) requiere obtener una medición física de la distancia entre el centro de la cámara y el origen del robot, para posteriormente con esa información poder modificar los datos acorde al modelo de cámara de orificio. En ambas etapas se utiliza un algoritmo iterativo de estimación no lineal para calcular seis parámetros que definen el modelo de cámara, los cuales se van actualizando durante el desarrollo de la maniobra del robot.

En este trabajo se plantea el problema de eliminar las dos restricciones antes mencionadas que presenta el método CSM, que son la necesidad de alejar las cámaras del espacio de trabajo, y el poder incluir dentro de los parámetros del modelo la distancia entre el centro de la cámara y el origen del robot y así eliminar el requerimiento de realizar la medición de manera física. Además se plantea la problemática de poder obtener los parámetros del modelo de cámara en una sola iteración, de manera que se evite cualquier problema de convergencia que pudiera surgir en el cálculo de la solución, y permita obtener la unicidad en el proceso de estimación del modelo.

En [García, 2007] se presenta una propuesta de solución a la problemática antes mencionada, donde se sustituye el modelo ortográfico por el modelo de cámara de orificio. Este modelo no requiere que las cámaras estén alejadas, ni tampoco la medición física de ninguno de los parámetros, ya que toda la información está incluida en el modelo. Dicha solución permite obtener los parámetros de visión en un solo proceso lo que hace innecesaria la corrección por *flattening*. Esta metodología presenta como desventaja que, al tener once grados de libertad, es muy difícil lograr la convergencia del modelo a una solución. Para evitar ese problema se utiliza la solución de un sistema lineal, que sirve como condición inicial para un proceso de estimación no lineal denominado corrección diferencial por mínimos cuadrados, el cual permite optimizar el cálculo del modelo.

Por lo anteriormente descrito, en este trabajo se plantea la hipótesis de que bajo ciertas condiciones, una aproximación lineal del modelo de cámara es suficiente para lograr el posicionamiento del robot de manera correcta. Dicha aproximación inicial se va refinando conforme se va actualizando la posición del robot durante la maniobra, tal como se plantea en la metodología CSM. De esta manera se elimina el proceso de optimización posterior y hace que el cálculo del modelo sea más fácil y rápido en su implementación.

Gracias a la aportación presentada en este trabajo es posible establecer un esquema adecuado que permite sustituir el modelo no lineal de cámara utilizado en CSM, por un modelo lineal que incluye todos los parámetros involucrados en el proceso de formación de la imagen. Esto elimina la necesidad de realizar el proceso de *flattening*, por lo que no se requiere realizar ninguna medición física. Debido a que el modelo lineal incluye directamente el efecto de perspectiva, se elimina el requerimiento de que las cámaras estén alejadas. Otra de las ventajas del modelo propuesto es que se evita cualquier problema de convergencia que pudiera existir en la solución, además de que el proceso de estimación de los parámetros de visión es más sencillo de implementar y más rápido en su ejecución.

Motivación y estado del Arte

En robótica, desde el punto de vista del procesamiento de la información, se involucran funciones de control de movimientos, percepción y planificación. El sistema de control involucra tanto bucles de retroalimentación de la información obtenida por los sensores internos, como de los sensores externos. La configuración de la estructura mecánica articulada se mide a través de los sensores internos, que permiten obtener información de posición, velocidad, fuerza, par. Por otro lado los sensores externos proporcionan sentidos al robot. La información que arrojan es utilizada por el sistema de percepción para realizar una representación del entorno. Estos sistemas de percepción sensorial permiten que un robot pueda adaptar su comportamiento de manera automática en función de las variaciones que se producen en su entorno.

El desarrollo de sistemas de percepción sensorial surge a partir de los progresos tecnológicos en sensores, tales como los de visión, tacto, audición e incluso olfato. Sin embargo, la percepción no sólo involucra la captación de la información sensorial, sino también su tratamiento e interpretación. Por lo tanto es necesario realizar una abstracción a partir de un cierto conocimiento previo del entorno. Por lo anterior es claro que la complejidad de la percepción artificial depende de lo estructurado que esté dicho entorno [[Baturone, 2005](#)].

En los siguientes párrafos se realiza una mención sobre los sistemas que existen en la actualidad, enfocado a la finalidad de la investigación realizada en este trabajo de tesis

que es el control de un robot manipulador utilizando como sensores externos sistemas de visión computacional.

1.1 Contexto y estado del arte

El control de robots mediante sensores de visión es una de las áreas que ha tomado mayor interés dentro del campo de la robótica. Este interés ha sido propulsado principalmente por los avances en la tecnología de los componentes electrónicos, y por el desarrollo de nuevos algoritmos que han permitido que esta tecnología sea viable de utilizarse en procesos industriales. Por tal motivo ha sido posible el desarrollo de algunas técnicas que han ido evolucionando a través del tiempo. Las técnicas más populares de control basado en visión para robots manipuladores son Teleoperación [[Ma et al., 2000](#)], Servo Visual (VS por sus siglas en inglés) [[Hutchinson et al., 1996](#)], Manipulación en Espacio de Cámara (CSM, por sus siglas en inglés) [[Skaar et al., 1987](#)] y Técnicas de Calibración [[Hartley and Zisserman, 2004](#)].

La teleoperación hace posible el control vía remota de un robot manipulador utilizando como retroalimentación la información visual del operador [[Passenberg et al., 2010](#)]. Un sistema de teleoperación se compone básicamente de un robot maestro gobernado por una o varias personas y un robot remoto o esclavo cuyo cometido es interactuar con el entorno de una manera que resulte útil en el desempeño de tareas concretas [[Soyguder and Abut, 2016](#)] [[Khurshid et al., 2016](#)]. De manera general, un sistema teleoperado es aquel que permite controlar un robot esclavo que se ubica en una zona remota a través del manejo de un robot maestro comandado por un operador. El control se realiza mediante cámaras de video que transmiten la imagen al operador. La teleoperación toma ventaja de la capacidad de percepción del sentido de la vista humano. La principal limitante de estos sistemas es el tiempo de retraso asociado a la transmisión de video en tiempo real, ya que si estos retrasos son significativos puede resultar muy complicado el completar la maniobra del robot. Para lidiar con ese problema se han propuesto algunas soluciones en: [[Park et al., 2011](#)], [[Sun et al., 2014](#)], [[Sun et al., 2016](#)] y [[Martínez et al., 2016](#)]. También se han desarrollado algunos trabajos de teleoperación semi-autónoma que tratan de compensar los problemas que ocasionan los retardos en la transmisión de la información [[Liu and Chopra, 2013](#)][[Zhai and Xia, 2016](#)]. La aplicación más representativa de los robots teleoperados es en telecirugías.

El control basado en Servo Visual, es una técnica ampliamente utilizada para el control de robots por medio de visión, se define como el control de la posición y orientación

del efector final de un robot mediante la información extraída de una o varias cámaras que pueden estar orientadas hacia el área de trabajo [Cai et al., 2016] o bien montadas en el efector del robot [Hajiloo et al., 2016], [Duflot et al., 2016]. Esta técnica es basada en la percepción visual del robot y de la pieza de trabajo, la diferencia entre la posición actual de la pieza de trabajo y la posición deseada se reduce mediante una ley de control [Hashimoto, 2003] [Zacharia et al., 2009] [Ma and Su, 2015] [Ourak et al., 2016]. La metodología VS puede ser dividida en dos áreas principales: basado en posición y basado en imagen.

La técnica de control VS basado en posición emplea un marco de referencia, el cual está definido como la posición y orientación relativa entre el objeto manipulable y el efector final del robot. La información de posición y orientación es obtenida a través de imágenes captadas de una o dos cámara situada en la herramienta del robot, o por medio de triangulación usando dos o más cámaras fijas [Westmore and Wilson, 1991] [Parsapour et al., 2013] [Burger et al., 2015]. La eficacia de esta metodología depende en gran medida en la robustez y velocidad de la detección del objeto. Debido a lo anterior, esta técnica es bastante sensible al ruido presente en las imágenes. Por otro lado, VS basado en imagen define su marco de referencia en el plano de la imagen, generalmente se extraen algunas características de los objetos presentes en las imágenes. Tales características pueden ser tamaños de regiones, líneas, centros, esquinas o segmentos que pueden ser procesados previamente. Se calcula la distancia entre la característica actual y la característica de referencia, y posteriormente, la maniobra del robot se realiza al reducir esa distancia [Burlacu et al., 2011] [Hwang et al., 2015] [Hajiloo et al., 2016].

El control basado en Servo Visual requiere la estimación en tiempo real del jacobiano de la imagen para la actualización de datos en tiempo real, si existe un retraso significativo en la actualización de los datos, la ejecución del método se interrumpe. Para solucionar este problema se han propuesto algunas técnicas para el cálculo del jacobiano [Nadi et al., 2014] [Qing and Yue, 2015].

El método de Manipulación en Espacio de Cámara, a diferencia de VS, utiliza un sistema de control en lazo abierto y no requiere de un proceso de calibración de las cámaras [Skaar et al., 1990] [Gonzalez-Galvan et al., 2011] [Liu et al., 2014b]. Utiliza una relación algebraica entre el espacio de coordenadas tridimensional del robot y el espacio de imagen, esta relación algebraica está definida por 6 parámetros que definen un modelo ortográfico de cámara [González-Galván et al., 2008]. El proceso de posicionamiento del robot se realiza en dos etapas, primero se obtienen los parámetros correspondientes al modelo ortográfico utilizando unas marcas visuales distintivas colocadas en

1. MOTIVACIÓN Y ESTADO DEL ARTE

la herramienta del robot. Posteriormente, debido a que el modelo ortográfico de cámara es una aproximación del modelo real, se realiza una operación denominada *flattening* [Gonzalez-Galvan et al., 1999] [Zhang et al., 2011], que representa una corrección de los datos acorde al modelo de cámara de orificio. Al emplear un modelo ortográfico, uno de los requerimientos del sistema es que la distancia del centro de la cámara a la pieza de trabajo sea bastante mayor que la distancia que comprende la superficie de trabajo del robot[Skaar et al., 1987]. Lo anterior implica que un mínimo de dos cámaras deben estar alejadas de la superficie donde se encuentra la pieza de trabajo. En [Gonzalez-Galvan et al., 2009] se propone una metodología basada en un modelo no lineal en la que se utilizan siete parámetros de visión para incluir directamente el efecto de perspectiva en el modelo de cámara. Los siete parámetros del modelo son obtenidos mediante un proceso iterativo de estimación no lineal. En [Bonilla et al., 2010] y [Bonilla et al., 2012] se utiliza esta misma técnica en combinación con un método de control por impedancia para el seguimiento de trayectorias.

Por otro lado, las Técnicas de Calibración consisten en encontrar una serie de parámetros que definen una relación entre puntos localizados en una imagen y su correspondencia en el espacio tridimensional. En general el objetivo de la calibración es obtener los parámetros internos de la cámara, tales como distancia focal, punto central del plano de imagen, posición y orientación con respecto al origen del sistema coordinado y factores de distorsión. Esto se realiza con la finalidad de poder extraer información métrica a partir de las imágenes. A la relación que contiene los parámetros involucrados en el proceso de formación de la imagen se le denominada matriz de cámara [Hartley and Zisserman, 2004]. Para obtener los parámetros de cámara normalmente se utilizan patrones de calibración que permiten obtener una correspondencia de puntos entre el espacio tridimensional y el espacio de imagen. Estos patrones tienen la característica que sus bordes, centros o esquinas son fácilmente detectables en la imagen [Wang et al., 2012] [Liu et al., 2016a]. Para ubicar un objeto en el mundo real se establece un sistema coordinado de referencia denominado marco de referencia del mundo, y de igual manera se define un marco de referencia de la imagen. Los procesos de calibración y posicionamiento del robot se realizan de manera separada, es decir, primero se localizan las coordenadas de la pieza de trabajo a partir de su ubicación en las imágenes, y posteriormente se calcula la posición requerida del robot empleando la cinemática inversa del manipulador.

Dentro de las técnicas de calibración, la reconstrucción tridimensional es una de las áreas de visión computacional que ha recibido bastante atención recientemente. Exis-

ten diversos métodos para estimar una estructura tridimensional a partir de estructuras bidimensionales [Winston and Horn, 1975], esto incluye la obtención de formas a partir de sombras [Horn and Brooks, 1989], [Yang et al., 2015], [Wang and Cheng, 2016], profundidad a través de enfoques [Ens and Lawrence, 1993], [Moeller et al., 2015] y visión estéreo [Dhond and Aggarwal, 1991], [Li et al., 2015a]. Normalmente en aplicaciones que requieren mayor precisión y velocidad se utiliza visión estéreo.

La reconstrucción tridimensional basada en visión estéreo consiste en estimar la estructura tridimensional de los objetos a partir de dos o más imágenes [Akkoul et al., 2014] [Sharma et al., 2015] [Liu et al., 2016c]. Los métodos clásicos de visión estéreo utilizan una serie de correspondencias de puntos entre las imágenes, que posteriormente son utilizadas para obtener las coordenadas tridimensionales del objeto por medio de una triangulación [Hartley and Sturm, 1997] [Otero and Sánchez, 2015] [de Villiers et al., 2015]. La triangulación se realiza utilizando un modelo matemático de la cámara. El modelo de cámara permite realizar una transformación entre los puntos ubicados en la imagen, y su correspondencia con los puntos ubicados en un eje coordenado tridimensional. El proceso de obtención del modelo de cámara es denominado calibración [Tsai, 1996] [Wu and Zhu, 2015]. Es necesario realizar una calibración para cada cámara involucrada en el sistema de visión.

Los algoritmos de calibración también han ido evolucionando a través del tiempo, algunas soluciones para el problema de calibración se presentan en [Wong et al., 2003] y en [Guillemaut et al., 2005]. También se han propuesto algunas metodologías robustas con el fin de realizar una reconstrucción tridimensional precisa: [Rahman and Krouglicof, 2012] [Liu et al., 2014a] [Song et al., 2016]. La mayoría de los métodos de calibración utilizan patrones de calibración para calcular el modelo de cámara [Sorgi and Bushnevskiy, 2015] [Liu et al., 2016b]. Sin embargo, también existen otras metodologías que no requieren el uso de dichos patrones, tales como: [Faugeras et al., 1992], [Quan and Triggs, 2000] y [Herrera et al., 2016].

Las primeras propuestas para calcular los parámetros relacionados con la calibración vienen dadas por los métodos clásicos de calibración. Estos métodos emplean patrones de puntos conocidos que son fácilmente identificables en las imágenes [Faugeras, 1993] [Batista et al., 1998] [Zhang, 2000]. La desventaja de estas metodologías es que dependen de un objeto externo y requieren de un proceso manual para la selección de las correspondencias de puntos.

Las técnicas que no dependen de un patrón de calibración se denominan métodos de auto-calibración [Maybank and Faugeras, 1990] [Faugeras et al., 1992] [Hartley, 1994].

1. MOTIVACIÓN Y ESTADO DEL ARTE

Estos métodos emplean correspondencias de puntos entre imágenes tomadas de distintas vistas de un mismo escenario, algunos ejemplos de estos métodos se reportan en: [Li et al., 2015b], [Xu and Deng, 2015], [Batteoui et al., 2016] y [Boudine et al., 2016]. La principal desventaja de estas metodologías es que dependen en gran medida de la precisión en la obtención de la correspondencia de puntos, ya que la presencia de falsos positivos reduce la eficacia de la calibración.

Tal como se mencionó anteriormente, el método CSM es una metodología que no requiere de una calibración precisa de las cámaras, ni del manipulador; tampoco requiere una retroalimentación en tiempo real de la información visual. Por tal motivo este método es viable de ser utilizado en maniobras de posicionamiento donde la ubicación de la pieza no manipulable permanece fija durante la maniobra del robot. Esta maniobra de posicionamiento es la más utilizada en procesos de manufactura industrial. En la siguiente sección se describe de manera general el método de manipulación en espacio de cámara, que es la metodología de control basado en visión que se aborda en este trabajo de tesis.

1.2 El método de manipulación en espacio de cámara

El método de manipulación en espacio de cámara es un método de control basado en visión que, como ya se mencionó anteriormente, difiere de los métodos de calibración y servo visual, ya que representa una estrategia de control en lazo abierto, basado en técnicas de estimación.

Como se mencionó anteriormente, el método CSM original [Skaar et al., 1990], consta de dos etapas para el proceso de obtención de los parámetros de visión. la primera etapa consiste en la obtención de la estimación del modelo matemático de la cámara. En esta primera etapa, CSM emplea un modelo ortográfico [Skaar et al., 1987], este modelo de cámara es una aproximación al modelo real, y tiene la ventaja que es más fácil de calcular. Sin embargo tiene la restricción de que las cámaras deben estar alejadas del espacio de trabajo del robot, esto para reducir el efecto de perspectiva [Gonzalez-Galvan et al., 1999]. La segunda etapa de CSM consiste en realizar una corrección de los datos acorde al modelo de cámara de orificio para tomar en cuenta el efecto de perspectiva. Este proceso es conocido como *flattening*. En los siguientes párrafos se describen de manera general estas dos etapas y se presentan las ecuaciones que las definen.

1.2.1 Modelo ortográfico

A través del modelo matemático de cámara de orificio es posible realizar un mapeo entre un sistema de coordenadas tridimensional del mundo real y el plano de una imagen bidimensional obtenido mediante dicha cámara. Debido a esto es posible obtener de una manera muy precisa la proyección de un punto tridimensional (X_i, Y_i, Z_i) sobre el plano de imagen (x_{ci}, y_{ci}) . Esto se puede realizar utilizando una proyección en perspectiva, la cual está definida mediante las siguientes ecuaciones:

$$x_{ci} = f \frac{X_i}{Z_i}, \quad y_{ci} = f \frac{Y_i}{Z_i} \quad (1.1)$$

donde f representa la distancia focal de la cámara, (X, Y, Z) las coordenadas tridimensionales del i -ésimo punto y (x, y) son las coordenadas del mismo punto proyectadas sobre el plano de imagen, tal como se muestra en la figura 1.1.

La ecuación que relaciona un sistema coordenado unido a la cámara referido a otro sistema de coordenadas arbitrario se muestra en la siguiente expresión:

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} = \begin{bmatrix} e_1^2 + e_2^2 - e_3^2 - e_4^2 & 2(e_2e_3 + e_1e_4) & 2(e_2e_4 - e_1e_3) & X_0 \\ 2(e_2e_3 - e_1e_4) & e_1^2 - e_2^2 + e_3^2 - e_4^2 & 2(e_3e_4 + e_1e_2) & Y_0 \\ 2(e_2e_4 + e_1e_3) & 2(e_3e_4 - e_1e_2) & e_1^2 - e_2^2 - e_3^2 + e_4^2 & Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \quad (1.2)$$

donde $e_1 \dots e_4$ representan los cuatro parámetros de Euler, que satisfacen la restricción $e_1^2 + e_2^2 + e_3^2 + e_4^2 = 1$, las coordenadas (X_0, Y_0, Z_0) localizan el origen de un sistema coordenado XYZ relativo a otro sistema de coordenadas xyz .

Cuando la distancia Z_0 es mucho mayor que los valores x_i, y_i, z_i, X_0, Y_0 , es posible utilizar una simplificación del modelo de cámara de orificio. A esta simplificación se le conoce como modelo ortográfico de cámara [Skaar et al., 1987]. El modelo ortográfico de cámara puede ser definido mediante las siguientes ecuaciones:

$$\begin{aligned} f_x(R_X(\Theta), R_Y(\Theta), R_Z(\Theta), C) &= x_{ci} = (C_1^2 + C_2^2 - C_3^2 - C_4^2) x_i \\ &\quad + 2(C_2C_3 + C_1C_4)y_i + 2(C_2C_4 - C_1C_3)z_i + C_5 \\ f_y(R_X(\Theta), R_Y(\Theta), R_Z(\Theta), C) &= y_{ci} = 2(C_2C_3 - C_1C_4)x_i \\ &\quad + (C_1^2 - C_2^2 + C_3^2 - C_4^2)y_i + 2(C_3C_4 + C_1C_2)z_i + C_6 \end{aligned} \quad (1.3)$$

donde las funciones f_x, f_y definen al modelo ortográfico de cámara, mientras que los valores $C_1 \dots C_6$ representan los 6 parámetros de visión, los cuales representan la relación

1. MOTIVACIÓN Y ESTADO DEL ARTE

entre la localización física de los puntos tridimensionales y su correspondiente localización en el plano de la imagen. Las coordenadas $R_X(\Theta)$, $R_Y(\Theta)$, $R_Z(\Theta)$ representan la posición de las marcas visuales colocadas en el manipulador referidas al sistema coordenado tridimensional, que a su vez dependen de los valores de las articulaciones del robot, $\Theta = (\theta_1, \theta_2, \dots, \theta_n)$, y que están contenidas en el modelo cinemático nominal del manipulador (Apéndice A), y x_{ci}, y_{ci} son las coordenadas en píxeles de la i -ésima marca visual en el plano de imagen.

Los seis parámetros de visión son estimados a partir de un número n de correspondencias de puntos en el espacio tridimensional y el espacio de imagen para cada una de las m cámaras participantes en el sistema de control por visión. Los parámetros de visión son obtenidos mediante la minimización del funcional $J(C)$ definido en la ecuación 1.4.

$$J(C) = \sum_{k=1}^m \left[\sum_{i=1}^n \left[(x_{ci}^k - f_x^k(R_{X_i}(\Theta), R_{Y_i}(\Theta), R_{Z_i}(\Theta); C))^2 + (y_{ci}^k - f_y^k(R_{X_i}(\Theta), R_{Y_i}(\Theta), R_{Z_i}(\Theta); C))^2 \right] \right] \quad (1.4)$$

El modelo ortográfico tiene la ventaja que es fácil de implementar, sin embargo, no toma en cuenta el efecto de perspectiva, por ello se utiliza posteriormente una actualización de los datos a través de un proceso conocido como *flattening*, tal como se describe en seguida.

1.2.2 Corrección por *flattening*

Para obtener información acorde al modelo de perspectiva se realiza una corrección de las muestras mediante las siguientes expresiones:

$$x_{ci}^* = \frac{Z_i}{Z_r} x_{ci}, \quad y_{ci}^* = \frac{Z_i}{Z_r} y_{ci} \quad (1.5)$$

a esta corrección de los datos se le conoce como proceso de *flattening*. De la ecuación 1.2 se puede obtener la siguiente expresión:

$$Z_i = Z_0 + 2(e_2 e_4 + e_1 e_3) x_i + 2(e_3 e_4 + e_1 e_2) y_i + 2(e_1^2 - e_2^2 - e_3^2 + e_4^2) z_i$$

donde Z_0 representa la distancia entre el origen del eje coordenado y el punto focal de la cámara. Los parámetros de *Euler* pueden ser aproximados mediante la siguiente expresión:

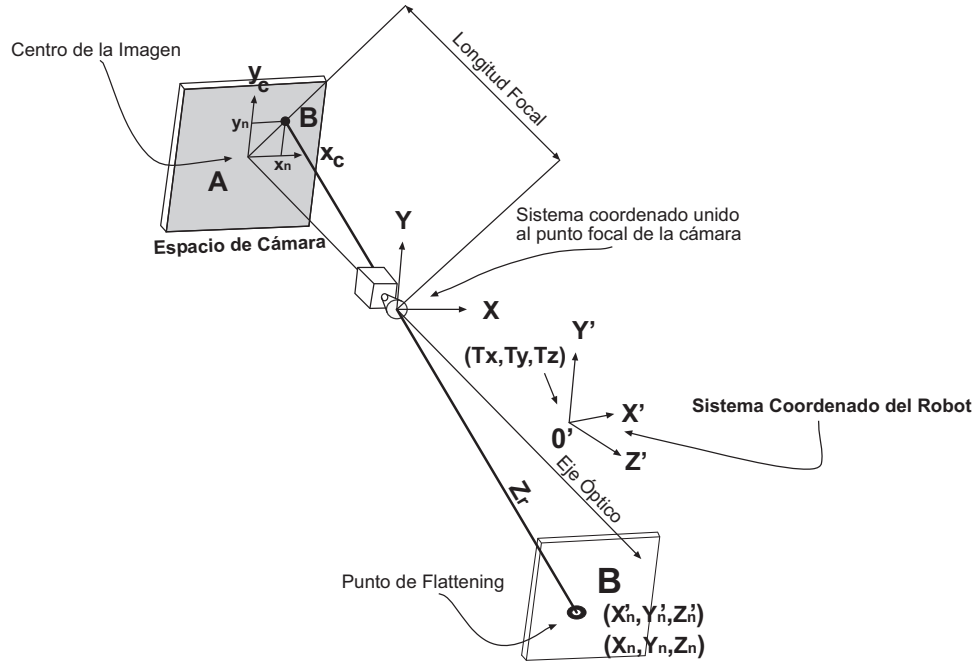


Figura 1.1: Sistema coordenados de referencia del método CSM.

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

el valor Z_r de la expresión 1.5 representa la distancia entre el eje óptico de la cámara y una marca visual de referencia, como se muestra en la figura 1.1. Este parámetro es calculado al final de la maniobra, cuando se tiene una buena estimación de los parámetros de visión cercanos al punto objetivo. Luego de obtener los datos corregidos mediante el proceso de *flattening*, se realiza nuevamente la estimación de los parámetros de visión, pero ahora aplicando como valores iniciales los obtenidos mediante el modelo ortográfico.

1.2.3 Algoritmo CSM

El algoritmo para la obtención de los parámetros de visión puede ser resumido en dos etapas:

1. Etapa Ortográfica

Estimar los parámetros C a través de la minimización de la ecuación 1.4 para un mínimo de dos cámaras.

2. Etapa de Flattening

1. MOTIVACIÓN Y ESTADO DEL ARTE

- (a) Fijar un valor aproximado para Z_r , que es conocida como la distancia de *Flat-tening* (fig. 1.1)).
- (b) Estimar Z_i a partir de un conjunto de muestras utilizando la ecuación 1.2.
- (c) Calcular las coordenadas x_{ci}^*, y_{ci}^* corregidas de los puntos a través de las ecuaciones 1.5.
- (d) Con los datos corregidos recalcular los valores C utilizando la expresión 1.4.
- (e) Si no se logra la convergencia, ir al paso (b).

Una de las principales características del método CSM es que no requiere de un proceso de calibración, debido a esto es necesario actualizar los seis parámetros de visión localmente y de manera iterativa durante la maniobra de posicionamiento. En cada iteración se estiman los parámetros de visión utilizando la información más reciente y se actualiza el cálculo de la configuración angular del robot para posicionarse sobre el punto objetivo. A continuación se describe el proceso para realizar el cálculo de la configuración articular del robot utilizado en CSM.

1.2.4 Configuración angular del manipulador

Una vez estimados los seis parámetros de visión del modelo, es posible calcular los valores de los ángulos del robot para poder alcanzar el punto objetivo de la maniobra. El cálculo de la posición del robot se obtiene al establecer los valores Θ de cada una de las articulaciones. Los valores de estos ángulos pueden ser obtenidos al minimizar el funcional $\Gamma(\Theta)$ definido en la ecuación 1.6:

$$\Gamma(\Theta) = \sum_{j=1}^{nc} \left[\left[x_t^j - f_x(R_x^j(\Theta), R_y^j(\Theta), R_z^j(\Theta); C^j) \right]^2 + \left[y_t^j - f_y(R_x^j(\Theta), R_y^j(\Theta), R_z^j(\Theta); C^j) \right]^2 \right] \quad (1.6)$$

donde nc representa el número de cámaras participantes; (x_t^j, y_t^j) son las coordenadas del punto objetivo en la j -ésima cámara; (R_x^j, R_y^j, R_z^j) son las coordenadas tridimensionales del punto objetivo; f_x, f_y representan el modelo ortográfico de cámara definido en las expresiones 1.3.

1.3 Motivación

El presente trabajo está enfocado en la metodología CSM, que es la técnica de control por visión que se ha venido desarrollando con éxito en la Universidad de Notre Dame, EU, y actualmente se utiliza para fines de investigación en la Facultad de Ingeniería de la Universidad Autónoma de San Luis Potosí. CSM es una técnica que usa visión por computadora que en este caso se aplica para el control de movimientos, y que, como se ha venido mencionando, no requiere de un proceso de calibración ni del manipulador ni del sistema de visión. Esta metodología ha sido muy utilizada para realizar la tarea de ubicar un objeto manipulable en una posición respecto a una pieza de trabajo denominada objeto no manipulable. La importancia de este método radica en que dicha tarea de posicionamiento es básica en las tareas más comúnmente realizadas por robots en la industria, como son los procesos de manufactura, manipulación de materiales y operaciones de ensamble.

1.4 Objetivos y alcances

El objetivo principal del presente trabajo está enfocado al desarrollo y la implementación de un nuevo método basado en visión para el control de un brazo robótico, fundamentado en el método de manipulación en espacio de cámara. Tal como se mencionó anteriormente, el método CSM requiere de dos etapas para obtener los parámetros involucrados en el sistema de control por visión. Dichos parámetros son obtenidos a partir de un proceso iterativo de estimación no lineal. Uno de los objetivos proyectados para el presente trabajo es el plantear un esquema para obtener los parámetros de visión en un solo proceso. Otro de los objetivos es el proponer un modelo de cámara simplificado que no presente problemas de convergencia y que facilite la implementación de la metodología CSM, además de proporcionar resultados similares a los obtenidos por la metodología original en el error final de posicionamiento del manipulador.

Dentro del trabajo de investigación se plantearon los siguientes objetivos particulares:

- Proponer y analizar un modelo simplificado de cámara para el método de manipulación en espacio de cámara.
- Desarrollar un esquema innovador de control por visión basado en un modelo lineal y la metodología CSM, con la finalidad de obtener la unicidad de la solución de los parámetros de visión.

1. MOTIVACIÓN Y ESTADO DEL ARTE

- Analizar los errores involucrados en el sistema de visión propuesto mediante el análisis del modelo de cámara y del modelo cinemático del manipulador de manera independiente.
- Realizar una validación experimental del esquema propuesto.
- Realizar una comparativa de las ventajas y desventajas de la metodología propuesta versus el método CSM original.

1.5 Organización de la tesis

En este capítulo se presentaron los antecedentes y alcances del presente trabajo, se realizó una breve introducción del método de manipulación en espacio de cámara, así como las etapas que involucra dicha metodología. En el siguiente capítulo se describe de manera general el modelo matemático de cámara de orificio. Se definen los parámetros que componen dicho modelo de cámara y dos metodologías para estimar los parámetros involucrados en el proceso de formación de la imagen. Se propone una metodología basada en CSM, que sustituye el modelo ortográfico por el modelo de cámara de orificio, cuyos parámetros son calculados mediante la solución de un sistema lineal. En el capítulo 3 se presenta un análisis de sensibilidad de la metodología propuesta en este trabajo empleando datos simulados y se realiza una validación a través de datos experimentales. En el capítulo 4 se presenta los resultados experimentales obtenidos para la metodología propuesta en este trabajo y se describe el software desarrollado que permite de una manera intuitiva la definición de una maniobra en espacio de cámara para el posicionamiento del manipulador, también se describe la implementación de un algoritmo para la detección automática y manipulación de arandelas. Finalmente se presenta la contribución de este trabajo, así como las conclusiones y el trabajo a futuro.

Desarrollo de un control basado en visión de un manipulador robótico utilizando un modelo lineal de cámara

En este capítulo se presenta una metodología para el posicionamiento de un robot manipulador usando visión computacional. Esta metodología está fundamentada en el método de manipulación en espacio de cámara. Se propone la sustitución del modelo ortográfico y el proceso de *flattening*, usados en la metodología original, por el modelo de cámara de orificio. Esto con la finalidad de obtener los parámetros de visión en una sola etapa. Además se presentan dos metodologías basadas en la solución de un sistema de ecuaciones lineales para estimar los parámetros del modelo de cámara de orificio.

Dentro de los sistemas de visión una parte importante es el modelo matemático de la cámara, ya que es el que permite realizar una transformación entre puntos ubicados en el espacio de cámara y su correspondencia en el espacio tridimensional. A continuación se describe de manera general el modelo de cámara de orificio.

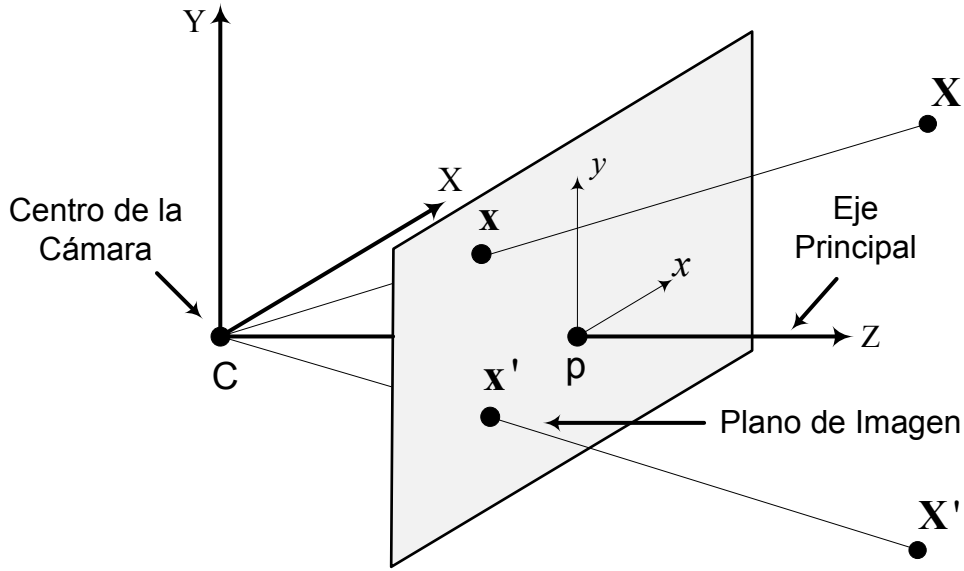


Figura 2.1: Geometría de la cámara de orificio. C representa el centro de la cámara, p el punto principal. El centro de la cámara se localiza en el origen del sistema coordenado.

2.1 El modelo de cámara de orificio

El modelo de cámara de orificio define una relación geométrica entre puntos situados en el espacio tridimensional y su proyección correspondiente en el plano de imagen, también denominado espacio de cámara. Cuando se emplea un modelo de cámara de orificio, al mapeo de puntos entre el espacio tridimensional y el espacio bidimensional de la imagen se le denomina proyección de perspectiva. El centro de la proyección en perspectiva, que es el punto donde todas las proyecciones se intersectan, se le conoce como centro óptico o centro de la cámara y la línea que es perpendicular al plano de imagen que pasa a lo largo del centro óptico se denomina eje principal o eje focal, tal como se presenta en la figura 2.1. Al punto que intersecta el plano de imagen con el eje principal se le denomina punto principal. El modelo de cámara de orificio que representa una proyección de perspectiva de puntos en el espacio tridimensional sobre el plano de imagen se describe con mayor detalle en los siguientes párrafos.

Si se define $\mathbf{X} = (X, Y, Z)^T \in \mathbb{R}^3$ como un punto en el espacio tridimensional, el cual es mapeado en un punto $\mathbf{x} = (x, y)^T \in \mathbb{R}^2$ en el plano de imagen. Para simplificar el modelo, el plano de imagen es proyectado a lo largo del eje Z frente al centro de la cámara, entre el eje óptico y el punto $\mathbf{X}$ tal como se muestra en la figura 2.1.

Acorde a la figura 2.1, existen dos sistemas coordenados en el modelo: el sistema

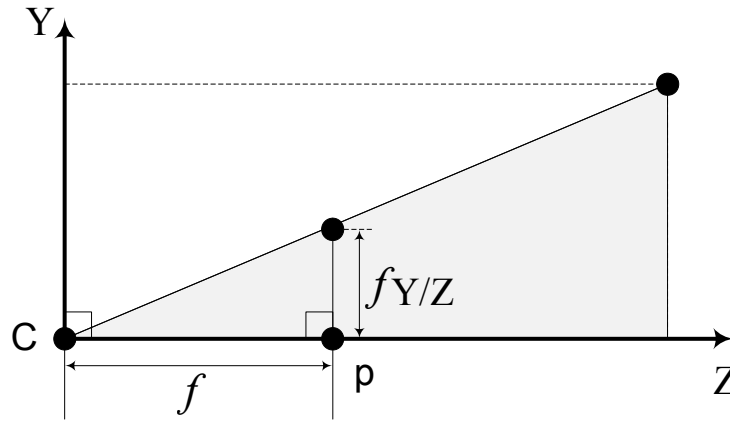


Figura 2.2: Proyección de un punto tridimensional sobre el plano YZ del sistema coordenado tridimensional.

coordenado unido a la cámara XYZ y el sistema coordenado de la imagen denotado por xy . Las coordenadas del punto proyectado $\mathbf{X} = (X, Y, Z)^T$ y $\mathbf{x} = (x, y)^T$ son arbitrarias. En la figura 2.2 se muestra la proyección del punto sobre el plano YZ.

El modelo de cámara de orificio está integrado por dos tipos de parámetros, los parámetros intrínsecos, que son los valores que posee la cámara de acuerdo a su fabricación, tales como distancia focal, coordenadas del punto principal y distorsión. Y los parámetros extrínsecos, que definen la posición y orientación del sistema coordenado unido a la cámara con respecto a un sistema coordenado definido. En los siguientes párrafos se describen las ecuaciones que definen el modelo de cámara de orificio así como los parámetros que lo componen.

2.1.1 Parámetros intrínsecos

Los parámetros intrínsecos o parámetros internos de la cámara están relacionados directamente con el proceso de formación de la imagen. Estos parámetros establecen las características geométricas y ópticas del sensor de visión. Los parámetros intrínsecos definen la proyección de perspectiva a través de la distancia focal, la relación entre el sistema coordenado del plano de imagen y las coordenadas en píxeles y la distorsión geométrica debida al uso de lentes. A continuación se describen de manera general los elementos que componen los parámetros internos de la cámara.

De acuerdo a la figura 2.2, el parámetro f que es denominado distancia focal, representa la distancia entre el centro óptico de la cámara y el plano de imagen. Mediante la aplicación de triángulos semejantes se pueden obtener las siguientes equivalencias:

2. DESARROLLO DE UN CONTROL BASADO EN VISIÓN

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} fX/Z \\ fY/Z \end{bmatrix} \quad (2.1)$$

Coordenadas homogéneas

Las coordenadas tridimensionales y sus proyecciones en el plano de imagen pueden ser representadas mediante coordenadas homogéneas, para el caso de las proyecciones en espacio de cámara se pueden representar de la forma que aparecen en la ecuación 2.2.

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} \mapsto \begin{bmatrix} \lambda x \\ \lambda y \\ \lambda \end{bmatrix} = \lambda \tilde{\mathbf{x}} \quad (2.2)$$

Se puede calcular un punto en el espacio de cámara a través de sus coordenadas homogéneas mediante la siguiente expresión:

$$\tilde{\mathbf{x}} = \begin{bmatrix} x \\ y \\ \lambda \end{bmatrix} = \begin{bmatrix} x/\lambda \\ y/\lambda \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} x/\lambda \\ y/\lambda \end{bmatrix} = \mathbf{x} \quad (2.3)$$

de acuerdo a la expresión 2.2 los puntos ubicados en el espacio tridimensional y sus proyecciones sobre el plano de imagen se pueden representar mediante coordenadas homogéneas de la manera que aparece en las siguientes expresiones:

$$\mathbf{X} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \tilde{\mathbf{X}}; \quad \mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \tilde{\mathbf{x}} \quad (2.4)$$

entonces, haciendo uso de coordenadas homogéneas, la expresión 2.1 se puede escribir de la siguiente manera:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} fX/Z \\ fY/Z \end{bmatrix} \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \lambda \begin{bmatrix} fX/Z \\ fY/Z \\ 1 \end{bmatrix} = \lambda \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} \quad (2.5)$$

la igualdad anterior puede ser expresada en forma matricial, tal como aparece en la siguiente ecuación:

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.6)$$

donde λ representa el factor de escala homogéneo. La ecuación 2.6 puede ser escrita de manera reducida como $\text{diag}(f, f, 1) [I | 0]$, donde $\text{diag}(f, f, 1)$ representa una matriz

diagonal y $[I|0]$ representa una matriz dividida en dos bloques, el primero representa una matriz identidad de 3×3 y el segundo bloque un vector columna con todos sus elementos igual a cero. La expresión 2.6 se puede escribir de manera compacta como aparece en la siguiente expresión:

$$\mathbf{x} = \mathbf{P}\mathbf{X} \quad (2.7)$$

donde el vector $\mathbf{X}$ representa las coordenadas de un punto en el espacio tridimensional, el vector $\mathbf{x}$ representa la proyección del punto en coordenadas de la imagen y $\mathbf{P}$ define la matriz homogénea de 3×4 elementos, denominada matriz de proyección de cámara. La expresión 2.7 define la matriz para el modelo de cámara de orificio con proyección central siendo de la siguiente manera:

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

Punto principal

En la ecuación 2.6 se asume que el origen del sistema coordenado de la imagen está localizado en el centro de la misma, sin embargo, en la práctica es muy común que ese no sea el caso. Por lo tanto, en general existe un mapeo de la forma:

$$(X, Y, Z)^T \mapsto (fX/Z + c_x, fY/Z + c_y)^T$$

de esta manera, para tomar en cuenta esa diferencia se añade una compensación de distancia (c_x, c_y) , para el eje x y eje y de la imagen. La expresión obtenida se muestra a continuación:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \lambda \begin{bmatrix} fX/Z + c_x \\ fY/Z + c_y \\ 1 \end{bmatrix} = \lambda \begin{bmatrix} f & 0 & c_x & 0 \\ 0 & f & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.8)$$

Píxeles por unidad de distancia

Debido a que los parámetros f, c_x, c_y son manejados en unidad de longitud, es necesario introducir los factores m_x, m_y para compensar los valores en píxeles de los parámetros en espacio de cámara. Dichos factores están definidos como el número de píxeles por unidad de longitud en dirección x e y respectivamente. De esta manera, a partir de la ecuación 2.8 se obtiene la siguiente expresión:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \lambda \begin{bmatrix} \alpha_x & 0 & p_x & 0 \\ 0 & \alpha_y & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.9)$$

2. DESARROLLO DE UN CONTROL BASADO EN VISIÓN

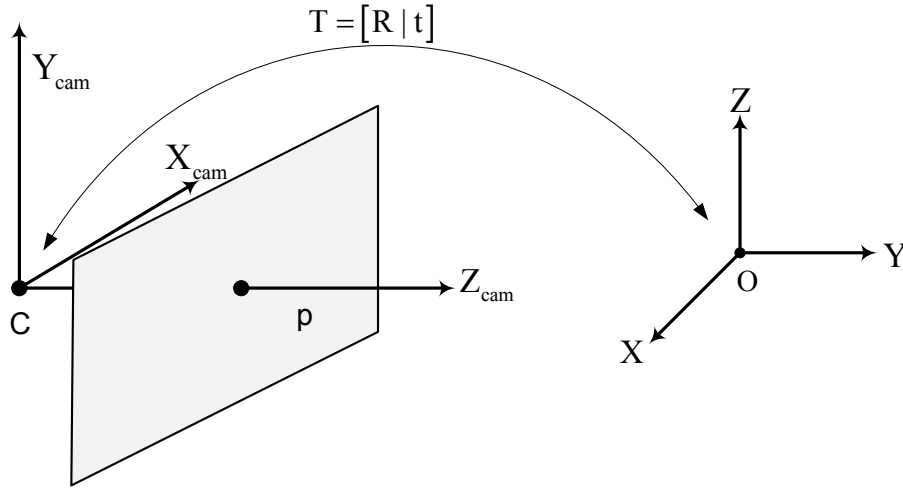


Figura 2.3: Transformación Euclidiana entre el sistema coordenado tridimensional y el sistema coordenado unido a la cámara.

donde:

$$\begin{aligned}\alpha_x &= m_x f, & p_x &= m_x c_x \\ \alpha_y &= m_y f, & p_y &= m_y c_y\end{aligned}$$

la sub matriz de 3×3 de la expresión 2.9 es denominada matriz de calibración K y contiene los parámetros intrínsecos de la cámara, tal como aparece en la siguiente ecuación:

$$K = \begin{bmatrix} \alpha_x & s & p_x \\ 0 & \alpha_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.10)$$

En la expresión 2.10 se introdujo el parámetro s que representa un factor de distorsión, el cual es cero si los ejes del sensor son ortogonales que es el caso más común.

2.1.2 Parámetros extrínsecos

Los parámetros extrínsecos o parámetros externos de la cámara definen la relación que existe entre el sistema coordenado unido a la cámara y un sistema coordenado tridimensional definido, el cual es denominado sistema coordenado del mundo. Los parámetros extrínsecos están conformados por seis elementos, de los cuales tres corresponden a los ángulos de rotación sobre cada uno de los ejes y los otros tres integran un vector de traslación, tal como se describe a continuación.

Rotación y traslación de la cámara

Para tomar en cuenta la posición y orientación del eje coordenado unido a la cámara

con relación al sistema coordenado tridimensional también denominado marco de referencia del mundo, se utiliza una transformación de coordenadas que incluye una matriz de rotación y un vector de traslación (figura 2.3) como se denota a continuación:

$$T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}$$

donde r_{ij} representan los elementos de la matriz $\mathbf{R}$ de rotación y t_{ij} son los elementos del vector $\mathbf{t}$ de traslación.

Los nueve elementos de la matriz de rotación $\mathbf{R}$ no representan nueve parámetros independientes, ya que la transformación de rotación entre dos sistemas coordenados puede ser representada por un máximo de tres grado de libertad. Por lo tanto se requieren sólo tres parámetros para definir una rotación entre dos ejes coordenados. De acuerdo a lo anterior la matriz de rotación posee las siguientes restricciones:

$$\begin{aligned} \sum r_{ij}^2 &= 1 & j \in \{1, 2, 3\} \\ r_{1i}r_{1j} + r_{2i}r_{2j} + r_{3i}r_{3j} &= 0 & i \neq j \end{aligned}$$

la primera restricción proviene del hecho de que las columnas de una matriz de rotación son vectores unitarios. La segunda restricción es debida a que las columnas de dicha matriz son ortogonales entre sí. De tal manera que ambas restricciones definen seis ecuaciones independientes con nueve incógnitas. Lo anterior implica que sólo existen tres grados de libertad en la matriz de rotación.

De esta forma se puede expresar de manera general la ecuación 2.9 como sigue:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \lambda \begin{bmatrix} \alpha_k & s & p_x \\ 0 & \alpha_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.11)$$

La expresión 2.11 representa el modelo de cámara de orificio que contiene once grados de libertad, que corresponden a 5 parámetros intrínsecos y 6 parámetros extrínsecos, de los cuales 3 corresponden a la matriz de rotación y 3 al vector de traslación. Estos parámetros están contenidos de manera implícita en la matriz de proyección $\mathbf{P}$, mostrada en la expresión 2.7. De manera compacta, la ecuación que define la matriz de proyección se puede expresar de la siguiente manera:

$$\mathbf{P} = \mathbf{K} [\mathbf{R} | \mathbf{t}] .$$

2. DESARROLLO DE UN CONTROL BASADO EN VISIÓN

Tal como se mencionó anteriormente, en la matriz de proyección están contenidos de manera implícita los once parámetros involucrados en el proceso de formación de la imagen, para poder calcularlos es necesario obtener una estimación de dicha matriz de proyección a partir de una serie de correspondencias de puntos entre el espacio 3D y las imágenes. En los siguientes párrafos se describe de manera general la metodología empleada para estimar los parámetros de visión contenidos en la matriz de proyección de cámara.

2.2 Metodología para la estimación de los parámetros de visión

Tal como se describió anteriormente, el método CSM original utiliza dos etapas para obtener los parámetros de visión. Estas dos etapas están compuestas por la estimación de los parámetros del modelo ortográfico y posteriormente la corrección de las observaciones acorde al modelo de perspectiva mediante *flattening*. Una de los objetivos del presente trabajo es el proponer una metodología para calcular los parámetros de visión en una sola etapa, y eliminar *flattening* en el proceso de estimación del modelo de cámara.

A continuación se describen dos metodologías para estimar la matriz de proyección que define el modelo de cámara de orificio. Ambas estrategias permiten obtener los parámetros del modelo de cámara en un sólo proceso. La primera metodología se basa en una solución lineal de ecuaciones y la segunda metodología utiliza descomposición en valores singulares para estimar los parámetros de la matriz de proyección.

2.2.1 Sistema de ecuaciones lineales

Como se mencionó al principio de este documento, una de las propuestas para el desarrollo de esta tesis fue el sustituir el modelo ortográfico de cámara utilizado en el método CSM original, por el modelo de cámara de orificio. Esto con la finalidad de obtener los parámetros de visión en una sola etapa y eliminar el proceso denominado *flattening*. A continuación se describe una manera de estimar la matriz de proyección de cámara usando la solución de un sistema de ecuaciones lineales. Esta metodología es denominada *Direct Linear Transformation* [Hartley and Zisserman, 2004] (DLT por sus siglas en inglés). DLT es una técnica de calibración de cámaras comúnmente utilizada para la reconstrucción tridimensional de ambientes.

De acuerdo al modelo de cámara de orificio definido en la ecuación 2.7, la relación entre las coordenadas (x, y) en espacio de imagen de un punto y su posición física (X, Y, Z) en el espacio tridimensional está representada por las siguientes funciones:

$$\begin{aligned} f_x(X, Y, Z; P) &= x_i = \frac{p_{11}X + p_{12}Y + p_{13}Z + p_{14}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}} \\ f_y(X, Y, Z; P) &= y_i = \frac{p_{21}X + p_{22}Y + p_{23}Z + p_{24}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}} \end{aligned} \quad (2.12)$$

donde p_{ij} representan los elementos de la matriz P , denominada matriz de proyección de cámara. Reacomodando los términos de la expresión 2.12 se obtienen las siguientes ecuaciones:

$$\begin{aligned} x_i (p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}) &= p_{11}X_i + p_{12}Y_i + p_{13}Z_i + p_{14} \\ y_i (p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}) &= p_{21}X_i + p_{22}Y_i + p_{23}Z_i + p_{24} \end{aligned} \quad (2.13)$$

expresando la ecuación 2.13 en forma matricial se obtiene:

$$\begin{bmatrix} X & Y & Z & 1 & 0 & 0 & 0 & 0 & -xX & -xY & -xZ & -x \\ 0 & 0 & 0 & 0 & X & Y & Z & 1 & -yX & -yY & -yZ & -y \end{bmatrix} P = 0 \quad (2.14)$$

la ecuación anterior representa un sistema de ecuaciones lineales con 12 incógnitas. Donde el vector, p tiene la forma:

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

La matriz de proyección contiene de manera implícita los parámetros del modelo matemático de la cámara como ya se mencionó anteriormente y puede representarse de la siguiente manera:

$$\begin{bmatrix} \lambda x_i \\ \lambda y_i \\ \lambda \end{bmatrix} = \begin{bmatrix} \alpha_x & s & p_x \\ & \alpha_y & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} \quad (2.15)$$

La ecuación 2.15 como ya se mencionó anteriormente tiene 11 grados de libertad. Al multiplicar las matrices de parámetros intrínsecos y extrínsecos de la expresión 2.15 se obtiene la matriz P que de igual manera tiene 11 grados de libertad dado que es una matriz homogénea [Hartley and Zisserman, 2004].

De acuerdo a lo anterior, considerando que el último elemento de la matriz P representa un factor de escala, el cual puede ser elegido como $p_{34} = 1$, se puede obtener el siguiente sistema de ecuaciones:

$$\begin{bmatrix} X & Y & Z & 1 & 0 & 0 & 0 & 0 & -xX & -xY & -xZ \\ 0 & 0 & 0 & 0 & X & Y & Z & 1 & -yX & -yY & -yZ \end{bmatrix} \tilde{P} = \begin{bmatrix} x \\ y \end{bmatrix} \quad (2.16)$$

2. DESARROLLO DE UN CONTROL BASADO EN VISIÓN

donde $\tilde{\mathbf{p}}$ está formado por los primeros 11 elementos del vector $\mathbf{p}$. La matriz de proyección $\mathbf{P}$ contiene 12 elementos; ignorando la escala entonces tiene 11 grados de libertad, por lo cual es necesario un mínimo de 11 ecuaciones para poder resolver este sistema. Debido a que cada correspondencia entre el sistema coordenado tridimensional y el espacio de imagen proporciona dos ecuaciones, entonces se requieren $5\frac{1}{2}$ puntos para resolverlo. El $\frac{1}{2}$ indica que sólo es necesario una ecuación del sexto punto, es decir, sólo es necesario conocer la coordenada x o y de dicho punto.

En la práctica es recomendable utilizar más de 6 puntos para lograr una mejor estimación de la matriz de proyección, esto genera un sistema de ecuaciones sobredeterminado que contiene 11 incógnitas y n ecuaciones. De acuerdo a la expresión 2.16, se tiene un sistema de ecuaciones de la forma $\mathbf{A}\tilde{\mathbf{p}} = \mathbf{b}$, para solucionar el sistema se puede utilizar el método de mínimos cuadrados para calcular un vector que minimice $\|\mathbf{A}\tilde{\mathbf{p}} - \mathbf{b}\|$ [Larson et al., 2009]. La solución lineal entonces estará dada por la siguiente expresión:

$$\tilde{\mathbf{p}} = [\mathbf{A}^T \mathbf{A}]^{-1} \mathbf{A}^T \mathbf{b} \quad (2.17)$$

Los componentes de la matriz de proyección de cámara se obtienen solucionando el sistema de ecuaciones 2.16. Este proceso se realiza de manera independiente para un mínimo de 2 cámaras que integran el sistema de control basado en visión.

2.2.2 Descomposición en valores singulares

Existe una manera alternativa para calcular la matriz de proyección de cámara $\mathbf{P}$, que aparece en la ecuación 2.7. Esta solución se basa en la descomposición en valores singulares (DVS)[Press et al., 2007], tal como se describe a continuación.

Si se tiene un número determinado de correspondencias de puntos $X_i \leftrightarrow x_i$ entre el sistema coordenado tridimensional y la imagen respectivamente. A partir de esos puntos es posible calcular la transformación entre dichos sistemas, denominada matriz $\mathbf{P}$ de calibración que define el modelo de cámara de orificio, tal que $\mathbf{x}_i = \mathbf{P}\mathbf{X}_i$ para todo punto i . Dado que la ecuación anterior involucra vectores homogéneos, puede darse el caso que los vectores correspondientes a ambos términos de la ecuación tengan la misma dirección pero diferente magnitud [Hartley and Zisserman, 2004]. Es ese sentido, la ecuación que define el modelo de cámara de orificio puede ser definida usando el producto vectorial de la siguiente manera:

$$\mathbf{x}_i \times \mathbf{P}\mathbf{X}_i = 0 \quad (2.18)$$

Si se define P^{jT} como la j -ésima fila de la matriz P , se puede obtener la siguiente expresión:

$$PX_i = \begin{bmatrix} P^{1T}X_i \\ P^{2T}X_i \\ P^{3T}X_i \end{bmatrix}$$

Realizando el producto cruz, y sustituyendo el vector $x_i = (x_c, y_c, w_c)^T$ se obtiene lo siguiente:

$$x_i \times PX_i = \begin{bmatrix} y_c P^{3T}X_i - w_c P^{2T}X_i \\ w_c P^{1T}X_i - x_c P^{3T}X_i \\ x_c P^{2T}X_i - y_c P^{1T}X_i \end{bmatrix} \quad (2.19)$$

Dado que $P^{jT}X_i = X_i^T P^j$ para $j = 1 \dots 3$, y tomando $w_c = 1$, la ecuación 2.19 puede ser escrita de forma matricial de la siguiente manera:

$$\begin{bmatrix} 0^T & -X_i^T & y_c X_i^T \\ X_i^T & 0^T & -x_c X_i^T \\ -y_c X_i^T & x_c X_i^T & 0^T \end{bmatrix} \begin{bmatrix} P^1 \\ P^2 \\ P^3 \end{bmatrix} = 0 \quad (2.20)$$

donde P es un vector de 12 elementos, que componen la matriz de proyección de cámara. El sistema anterior contiene 3 ecuaciones, sin embargo sólo dos de ellas son linealmente independientes, debido a que la tercera fila es obtenida de la suma de x_c veces la primera fila y y_c veces la segunda. Por tal motivo, la expresión 2.20 se puede expresar como aparece en la igualdad 2.21.

$$\begin{bmatrix} 0^T & -X_i^T & y_c X_i^T \\ X_i^T & 0^T & -x_c X_i^T \end{bmatrix} \begin{bmatrix} P^1 \\ P^2 \\ P^3 \end{bmatrix} = 0 \quad (2.21)$$

Finalmente de manera explícita la ecuación que permite calcular la matriz de proyección de cámara a partir de la descomposición en valores singulares queda definida en la ecuación 2.22.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & -X_i & -Y_i & -Z_i & -1 & y_c X_i & y_c Y_i \\ X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -x_c X_i & -x_c Y_i \\ & & & & y_c Z_i & y_c & & & & \\ & & & & -x_c Z_i & -x_c & & & & \end{bmatrix} \begin{bmatrix} P^1 \\ P^2 \\ P^3 \end{bmatrix} = 0 \quad (2.22)$$

La ecuación anterior representa un sistema de la forma $AP = 0$. Donde A es una matriz de dimensión $2i \times 12$, siendo i el número de puntos utilizados para calcular la matriz de proyección. La solución de la ecuación 2.22 es obtenida mediante la descomposición en

2. DESARROLLO DE UN CONTROL BASADO EN VISIÓN

valores singulares de la matriz A , y viene dada por el vector que corresponde al valor singular más pequeño.

En las secciones anteriores se describió el modelo de cámara de orificio, así como dos metodologías para estimar las matrices de proyección de cámara a partir de correspondencias de puntos. Ambas metodologías utilizan la solución de un sistema de ecuaciones lineales. El usar un sistema lineal implica que teniendo un número suficiente de muestras, es posible calcular los parámetros involucrados en el proceso de formación de la imagen en una sola iteración y evitando así cualquier problema de convergencia en la solución. Lo anterior indica que la solución de los parámetros de visión sería de fácil obtención y computacionalmente más rápida. Por estas razones, se propone la sustitución del modelo tradicionalmente usado en la metodología CSM original, por el modelo lineal de cámara descrito anteriormente. Esto permite estimar los parámetros de visión en una sola etapa, haciendo innecesario el proceso de *flattening*, además de que elimina también la necesidad de tener una medición física aproximada de la distancia Z_r (fig. 1.1) [Skaar et al., 1990].

En los siguientes párrafos se presenta un análisis de la equivalencia de los parámetros del modelo CSM original, descrito en la sección 1.2, y el modelo lineal de cámara descrito anteriormente.

2.3 Correspondencia entre los parámetros del modelo lineal y el modelo de CSM

Una de las propuestas de este trabajo de tesis es sustituir el modelo ortográfico usado en CSM por el modelo de cámara de orificio. En los siguientes párrafos se presenta un análisis de la correspondencia de los parámetros que componen tanto el modelo lineal, el cual será denominado como LCM-CSM (*Linear Camera Model - Camera Space Manipulation*, por sus siglas en inglés), como del modelo usado en la metodología CSM original, el cual será nombrado como OF-CSM (*Orthographic + Flattening - Camera Space Manipulation*, por sus siglas en inglés). Se realiza además una descripción general de los parámetros que los componen y la correspondencia entre ambos modelos.

Modelo OF-CSM

Para el caso del modelo OF-CSM el mapeo que se realiza en esta metodología está representado por un conjunto de parámetros de visión expresados por $C = [C_1, C_2 \cdots C_6]^T$

2.3 Correspondencia entre los parámetros del modelo lineal y el modelo de CSM

y está definido por el siguiente conjunto de ecuaciones no lineales:

$$\begin{aligned} x_i &= C_5 + (C_1^2 + C_2^2 - C_3^2 - C_4^2)R_{X_i}(\Theta) + 2(C_2C_3 + C_1C_4)R_{Y_i}(\Theta) \\ &\quad + 2(C_2C_4 - C_1C_3)R_{Z_i}(\Theta) \\ y_i &= C_6 + 2(C_2C_3 - C_1C_4)R_{X_i}(\Theta) + (C_1^2 - C_2^2 + C_3^2 - C_4^2)R_{Y_i}(\Theta) \\ &\quad + 2(C_3C_4 + C_1C_2)R_{Z_i}(\Theta) \end{aligned} \quad (2.23)$$

donde (x_i, y_i) representan las coordenadas en espacio de imagen del centro de la i -ésima marca visual localizada en el efector final del robot. El vector $R_{X_i}(\Theta)$, $R_{Y_i}(\Theta)$, $R_{Z_i}(\Theta)$ representa la localización de cada una de las marcas visuales relativa al origen del sistema coordenado del mundo, el cual coincide con el origen del modelo cinemático del robot.

El modelo ortográfico mostrado en la ecuación 2.23 puede ser representado en una forma matricial tal como se muestra a continuación:

$$\begin{bmatrix} x \\ y \end{bmatrix} = m \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} mr_{11}X' + mr_{12}Y' + mr_{13}Z' + mT_x \\ mr_{21}X' + mr_{22}Y' + mr_{23}Z' + mT_y \end{bmatrix} \quad (2.24)$$

El modelo de la ecuación 2.24 está compuesto por un factor de escala, una proyección ortográfica con respecto al plano Z , una matriz de rotación tridimensional y un vector de traslación. Con respecto a la ecuación, los elementos correspondientes son:

$$\begin{aligned} mr_{11} &= C_1^2 + C_2^2 - C_3^2 - C_4^2 & mr_{12} &= 2(C_2C_3 + C_1C_4) & mr_{13} &= 2(C_2C_4 - C_1C_3) \\ mr_{21} &= 2(C_2C_3 - C_1C_4) & mr_{22} &= C_1^2 - C_2^2 + C_3^2 - C_4^2 & mr_{23} &= 2(C_3C_4 + C_1C_2) \\ X' &= R_{X_i}(\Theta) & Y' &= R_{Y_i}(\Theta) & Z' &= R_{Z_i}(\Theta) \\ mT_x &= C_5 & mT_y &= C_6 \end{aligned}$$

donde m representa un factor de escala, el cual permite modelar únicamente píxeles cuadrados. $r_{11}, r_{12} \dots r_{23}$ son los elementos de la matriz de rotación, la cual tiene 3 grados de libertad, T_x y T_y son componentes del vector de traslación. De tal forma que la matriz que representa el modelo ortográfico contiene el mismo número de grados de libertad que los parámetros de visión $C_1, C_2 \dots C_6$.

La segunda etapa de la metodología OF-CSM, denominada *flattening* es realizada con el propósito de incrementar la precisión del modelo ortográfico. La finalidad de esta etapa es incluir el efecto de perspectiva en el modelo de cámara.

El modelo que incluye el efecto de perspectiva puede ser descrito mediante las siguientes ecuaciones:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} Z_r m & & \\ & Z_r m & \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} \quad (2.25)$$

2. DESARROLLO DE UN CONTROL BASADO EN VISIÓN

donde $[X, Y, Z]^T$ representa las coordenadas de un punto con respecto al sistema coordenado de la cámara. Las ecuaciones 2.25 pueden ser escritas de manera explícita de la siguiente manera:

$$\begin{aligned} x &= Z_r \frac{X}{Z} = Z_r m \frac{r_{11}X' + r_{12}Y' + r_{13}Z' + T_x}{r_{31}X' + r_{32}Y' + r_{33}Z' + T_z} \\ y &= Z_r \frac{Y}{Z} = Z_r m \frac{r_{21}X' + r_{22}Y' + r_{23}Z' + T_y}{r_{31}X' + r_{32}Y' + r_{33}Z' + T_z}. \end{aligned} \quad (2.26)$$

Las ecuaciones 2.26 corresponden al modelo matemático empleado en el método CSM original. Si se incluye Z_r , que es la distancia de *flattening* que permite tomar en cuenta el efecto de perspectiva, entonces el modelo OF-CSM tiene un total de 7 parámetros. Una cosa a destacar en esta metodología es que aunque la medición del parámetro Z_r no sea muy exacta, la estimación final del modelo es muy precisa debido a que el parámetro m mostrado en la ecuación 2.25 absorbe los errores [González-Galván and Seelinger, 1999], [González-Galván et al., 1997].

Modelo LCM-CSM

En este caso la relación entre puntos ubicados en el espacio tridimensional y su proyección sobre el plano de imagen está definido por la siguiente expresión:

$$\begin{bmatrix} \rho x \\ \rho y \\ \rho \end{bmatrix} = \begin{bmatrix} \alpha_x & s & p_x \\ & \alpha_y & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} \quad (2.27)$$

donde $(x, y)^T$ representa las coordenadas del punto en la imagen, ρ es un factor de escala, $(X, Y, Z)^T$ son las coordenadas de un punto ubicado con respecto al sistema coordenado tridimensional, $\alpha_x = m_x f$, $\alpha_y = m_y f$, f es la distancia focal, m_x, m_y representan la relación de aspecto de la medida del ancho y alto del pixel con respecto a las unidades del sistema coordenado tridimensional, p_x, p_y son las coordenadas del punto principal de la imagen y s representa un factor de distorsión.

Como ya se mencionó anteriormente, la ecuación 2.27 posee 11 grados de libertad. Para realizar una comparación de este modelo con el modelo Ortográfico + *flattening* se asumen los siguientes valores: $s = p_x = p_y = 0$.

Realizando la multiplicación en la ecuación 2.27 se obtienen las siguientes expresiones:

$$\begin{bmatrix} \rho x \\ \rho y \\ \rho \end{bmatrix} \begin{bmatrix} \alpha_x r_{11} & \alpha_x r_{12} & \alpha_x r_{13} & \alpha_x T_x \\ \alpha_y r_{21} & \alpha_y r_{22} & \alpha_y r_{23} & \alpha_y T_y \\ r_{31} & r_{32} & r_{33} & T_z \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix}$$

Dividiendo entre la tercera fila de la ecuación anterior se obtienen las siguiente igualdades:

$$x = \alpha_x \frac{r_{11}X' + r_{12}Y' + r_{13}Z' + T_x}{r_{31}X' + r_{32}Y' + r_{33}Z' + T_z}; \quad y = \alpha_y \frac{r_{21}X' + r_{22}Y' + r_{23}Z' + T_y}{r_{31}X' + r_{32}Y' + r_{33}Z' + T_z} \quad (2.28)$$

Comparando la ecuación 2.26 con la ecuación 2.28, se puede observar que ambos modelos son similares cuando $\alpha_x = \alpha_y$, es decir, cuando los píxeles son cuadrados. En la sección de resultados se muestra una simulación usando datos sintéticos que demuestra esta aseveración.

En los párrafos anteriores se demostró que existe una equivalencia entre parámetros del modelo OF-CSM y LCM-CSM. En la siguiente sección se describirá la metodología completa para el posicionamiento del robot, utilizando el modelo LCM-CSM.

2.4 Metodología CSM con un modelo lineal de cámara

En esta sección se describe de manera general la metodología de control basado en visión propuesta en este trabajo, que es una variante de la metodología CSM original. Esta nueva metodología, como ya se mencionó anteriormente, emplea el modelo de cámara de orificio y la solución de un sistema de ecuaciones lineales para obtener los parámetros de visión en una sola iteración, esto conlleva algunas ventajas como la simplicidad en la implementación del método, no presenta problemas de convergencia en la solución y requiere de un menor costo computacional, tal como se mostrará más adelante.

Las etapas principales que se emplean en la metodología LCM-CSM para el posicionamiento del robot manipulador son: La estimación de la matrices de proyección de cámara, caracterización de la superficie de trabajo y el cálculo de la configuración angular del manipulador. A continuación se describen dichas etapas que son cruciales para el correcto posicionamiento del robot.

2.4.1 Proceso de estimación de las matrices de proyección

El proceso para obtener una estimación de las matrices de cámara se muestra en la figura 2.4. A continuación se describe con más detalle cada una de las etapas.

2. DESARROLLO DE UN CONTROL BASADO EN VISIÓN

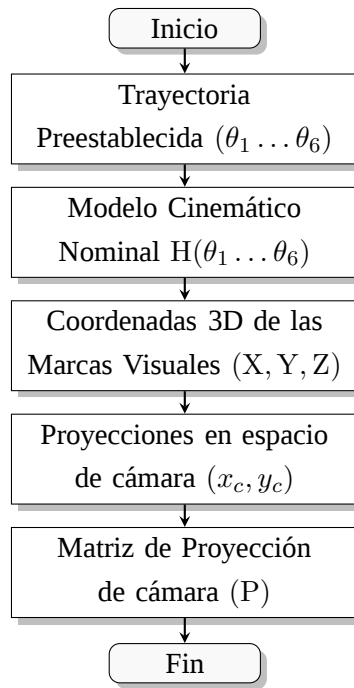


Figura 2.4: Proceso para la obtención de la matriz de proyección de cámara.

Para estimar las matrices de proyección que definen el modelo de cámara de orificio se utilizan una serie de marcas visuales distintivas colocadas en el efector final del robot (Fig. 2.5). La primera etapa consiste en realizar una trayectoria preestablecida. Dicha trayectoria consiste en programar una serie de movimientos del robot. En cada movimiento se toma información de la posición tridimensional de una de las marcas visuales del robot y de su posición en cada una de las cámaras del sistema de visión. Esa información posteriormente será utilizada para calcular una estimación inicial de las matrices de proyección de cámara.

Posteriormente, las coordenadas en espacio de cámara son detectadas para cada una de las marcas visuales y su posición tridimensional es obtenida a través del modelo cinemático directo del manipulador. Esta información es utilizada para generar una serie de puntos de correspondencia entre el espacio de cámara y el espacio tridimensional. A partir de dichos puntos es posible obtener una estimación de los parámetros de visión contenidos en la matriz de proyección de cámara definida en la ecuación 2.7. Lo anterior se describe con más detalle en los siguientes párrafos.

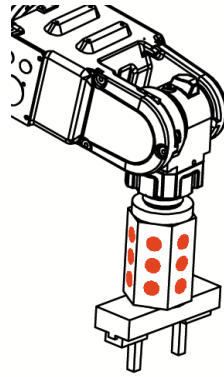


Figura 2.5: Marcas visuales distintivas colocadas en el efector final del manipulador.

A partir del modelo cinemático nominal del robot, es posible obtener las coordenadas tridimensionales de cada una de las marcas visuales colocadas en el efector final del robot referidas al origen (X_0, Y_0, Z_0) situado en la base del manipulador (Fig. 2.6). Las coordenadas (X_i, Y_i, Z_i) de la i -ésima marca visual pueden ser obtenidas mediante la siguiente igualdad:

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} = H(\Theta) \begin{bmatrix} x_t \\ y_t \\ z_t \\ 1 \end{bmatrix}$$

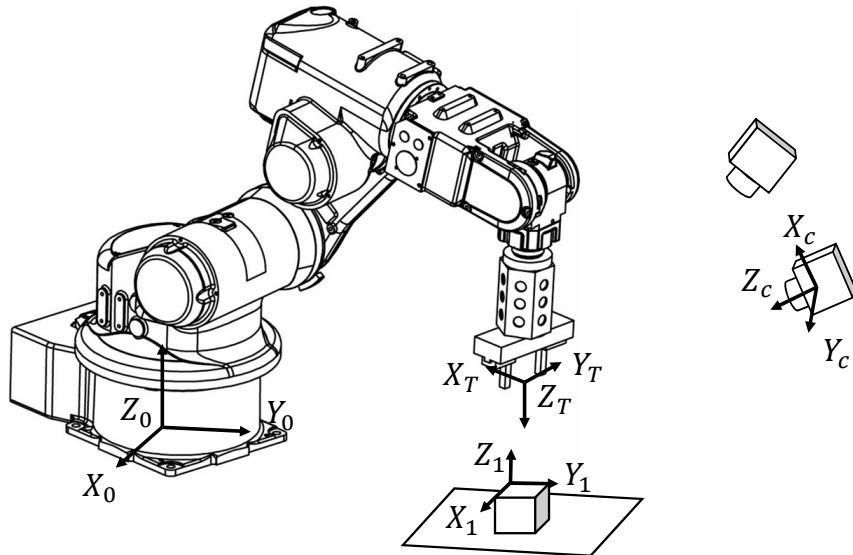


Figura 2.6: Sistemas coordenados involucrados en el método CSM.

2. DESARROLLO DE UN CONTROL BASADO EN VISIÓN

donde (X_i, Y_i, Z_i) representan las coordenadas de la i -ésima marca visual, con respecto al sistema coordenado unido a la base del robot, H es la matriz homogénea que define el modelo cinemático directo del robot, y (x_t, y_t, z_t) son las coordenadas de la i -ésima marca visual referida al eje coordenado situado al final de la herramienta del robot (Fig. 2.6).

Posteriormente a partir de la información de la posición tridimensional de las marcas visuales y su correspondiente detección en la imagen, es posible calcular las matrices de proyección utilizando la metodología descrita en la sección 2.2.

Las matrices de proyección de cámara son actualizadas constantemente durante la maniobra, para ello se toma información actual de la posición del robot, a través de las marcas visuales, conforme el robot se acerca al punto objetivo.

Teniendo la información de las matrices de proyección y la información del punto objetivo en espacio de cámara, es posible calcular los valores angulares del robot necesarios para lograr la posición y orientación requerida en la maniobra, tal como se describe en la siguiente sección.

2.4.2 Estimación de la configuración angular del manipulador

El cálculo de la configuración angular del robot, también denominada cinemática inversa, consiste en obtener los valores requeridos de las juntas del robot para posicionar el efector final del robot sobre un punto objetivo definido y con una orientación dada.

Para calcular la configuración angular del manipulador requerida para posicionar el efector final sobre un punto objetivo, se utilizan las matrices de proyección de cada una de las cámaras y la información del punto objetivo definido en la imagen.

El cálculo de la configuración angular del robot implica dos etapas principales. La primera consiste en obtener la posición y la segunda en lograr la correcta orientación de la herramienta con respecto a la superficie de posicionamiento.

Los parámetros de la matriz de proyección de cámara son actualizados continuamente durante el desarrollo de la maniobra. Los parámetros más recientes son utilizados para estimar la posición que el robot debe adoptar para llegar al objetivo. El cálculo de la configuración de las juntas del robot requeridas se realiza mediante la minimización del siguiente funcional:

$$\Gamma(\Theta) = \sum_{j=1}^{nc} \left\{ \left[x^j - f_x(r_X^j(\Theta), r_Y^j(\Theta), r_Z^j(\Theta); \mathbf{P}^j) \right]^2 + \left[y^j - f_y(r_X^j(\Theta), r_Y^j(\Theta), r_Z^j(\Theta); \mathbf{P}^j) \right]^2 \right\} \quad (2.29)$$

donde las funciones $f_x(\dots)$ y $f_y(\dots)$ corresponden a las expresiones que definen un modelo de cámara de orificio, mostradas en las ecuaciones 2.12; nc es el número total de cámaras utilizadas durante la maniobra, (x^j, y^j) representan las coordenadas en píxeles del punto objetivo en el que se desea posicionar el robot, proyectadas en la j -ésima cámara. Las coordenadas físicas de dicho punto objetivo están representadas por (r_X^j, r_Y^j, r_Z^j) , las cuales dependen de los valores de la configuración angular del manipulador contenidos en el vector Θ y que son obtenidos a partir del modelo cinemático nominal del robot.

Durante la maniobra de posicionamiento, conforme el robot se va acercado al objetivo, se realiza de manera iterativa una actualización de la información. En cada iteración se obtienen muestras de la posición en espacio de cámara del robot, a través de la detección de las marcas visuales. Con esa información y con las coordenadas físicas de dichas marcas, definidas a través del modelo cinemático nominal, las matrices de proyección son actualizadas P y se realiza la estimación del punto objetivo en espacio de cámara (f_x, f_y) . De igual manera, la configuración angular del robot Θ necesaria para posicionarse sobre el objetivo, definida en la expresión 2.29, es calculada en cada iteración utilizando la información más reciente.

En algunas aplicaciones es deseable, además de lograr un posicionamiento del robot en un punto, hacerlo de tal manera que el efector final se coloque de forma perpendicular a la superficie, sobre todo en tareas de manipulación de objetos. Para esto es necesario caracterizar dicha superficie de trabajo. En la siguiente sección se describe una metodología que permite caracterizar la superficie de trabajo del robot utilizando información visual.

2.4.3 Caracterización de la superficie

Para realizar una correcta caracterización es necesario tener información de un número determinado de puntos tridimensionales sobre el espacio de trabajo. Esta información sirve para calcular una función que se ajuste a dichos puntos. Para este propósito se hace uso de un láser matricial el cual proyecta sobre la superficie de trabajo una matriz de puntos laser tal como se muestra en la figura 2.7.

Para poder estimar las coordenadas tridimensionales de cada uno de los puntos proyectados sobre el área que se va a caracterizar, se requieren las coordenadas del punto en cuestión proyectadas en al menos dos cámaras del sistema de visión, con esa información es posible realizar una reconstrucción 3D del punto a partir de la información visual. Para poder conocer las coordenadas de un punto en dos o más cámaras es necesario conocer su correspondencia, es decir, saber cuales son las coordenadas de ese punto proyectadas

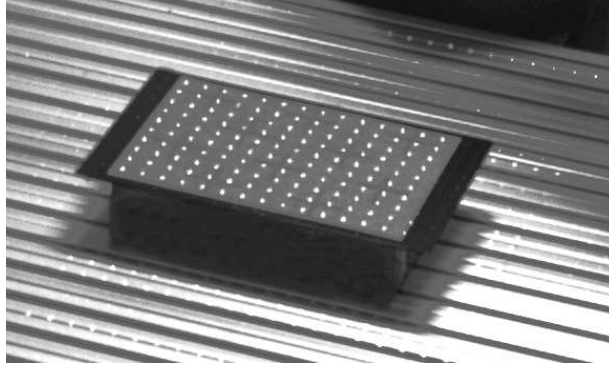


Figura 2.7: Laser matricial proyectado sobre la superficie de trabajo del robot.

en las demás cámaras del sistema de visión. A continuación se describe una metodología utilizada para obtener la correspondencia de dichos puntos.

Correspondencia de puntos

Tal como se mencionó anteriormente, con la finalidad de poder obtener las coordenadas tridimensionales de un punto sobre el área de trabajo, es necesario conocer sus coordenadas en el plano de al menos dos cámaras. Posteriormente se calcula una función de mapeo de puntos entre ambas cámaras. Para el caso en que la superficie a caracterizar está definida por un plano, dicha función puede estar representada por una transformación proyectiva u homografía, la cual puede ser obtenida mediante la siguiente expresión:

$$\begin{bmatrix} x & y & 1 & 0 & 0 & 0 & -x'x & -x'y \\ 0 & 0 & 0 & x & y & 1 & -y'x & -y'y \end{bmatrix} [\mathbf{h}] = \begin{bmatrix} x' \\ y' \end{bmatrix} \quad (2.30)$$

donde: $\mathbf{h} = [h_{11} \ h_{12} \ h_{13} \ h_{21} \ h_{22} \ h_{23} \ h_{31} \ h_{32}]^T$; h_{ij} representan los elementos de la matrix de homografía H , el último elemento de la matrix de homografía es considerado un factor de escala de tal forma que se puede seleccionar como $h_{33} = 1$ [Hartley and Zisserman, 2004].

Para resolver el sistema anterior se requieren al menos 4 correspondencias iniciales. La estimación de un punto en la cámara 1, proyectado en la cámara 2 se obtiene mediante la siguiente expresión:

$$\mathbf{x}_1 = \mathbf{H}\mathbf{x}_2$$

en donde $\mathbf{x}_1$, $\mathbf{x}_2$ son las coordenadas en píxeles de un punto localizado en la cámara 1 y 2 respectivamente y $\mathbf{H}$ es la matriz de 3×3 que representa la transformación proyectiva entre ambos planos de imagen.

Existe un error entre los datos estimados y los datos detectados en la imagen debido a que el modelo matemático que se emplea es lineal, pero en la realidad el modelo de transformación es no lineal a causa de la distorsión de las lentes, por esta razón es necesario encontrar el valor más cercano a cada estimación y asignarle el valor real obtenido de la detección en la imagen. La distancia entre el punto detectado y el estimado se calcula con la expresión:

$$r = \sqrt{(x_p - x_d)^2 + (y_p - y_d)^2}$$

donde r es la distancia entre el punto estimado (x_e, y_e) , y el punto candidato detectado (x_d, y_d) .

Luego de obtener las correspondencias de los puntos proyectados sobre la superficie a caracterizar, es posible estimar sus coordenadas tridimensionales, a través de las matrices de proyección, tal como se menciona a continuación:

Coordenadas tridimensionales

El cálculo de las coordenadas tridimensionales de cada punto se realiza a partir de las coordenadas en píxeles del punto en cuestión y las matrices de calibración actuales. Este objetivo se logra a partir de la minimización del funcional que aparece en la expresión 2.31:

$$J(X_i, Y_i, Z_i) = \sum_{j=1}^{nc} \left\{ [x_i^j - f_x(X_i, Y_i, Z_i; P^j)]^2 + [y_i^j - f_y(X_i, Y_i, Z_i; P^j)]^2 \right\} \quad (2.31)$$

donde: f_x, f_y son las funciones que representan la matriz de proyección definidas en las expresiones 2.12, P^j es la matriz proyección de cámara actual de la j -ésima cámara, x_i^j, y_i^j son las coordenadas en píxeles del punto en cuestión en la j -ésima cámara, y X_i, Y_i, Z_i representan las coordenadas tridimensionales del punto. Una aproximación lineal se puede obtener mediante la siguiente expresión:

$$\begin{bmatrix} x_i p^{3T} - p^{1T} \\ y_i p^{3T} - p^{2T} \\ x'_i p'^{3T} - p'^{1T} \\ y'_i p'^{3T} - p'^{2T} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} x_i p_{34} - p_{14} \\ y_i p_{34} - p_{24} \\ x'_i p'_{34} - p'_{14} \\ y'_i p'_{34} - p'_{24} \end{bmatrix}$$

El sistema de ecuaciones anterior se soluciona para los valores X_i, Y_i, Z_i ; p^{iT} representa la i -ésima fila de la matriz de proyección $\mathbf{P}$; (x_i, y_i) y (x'_i, y'_i) son las mediciones de los puntos para la cámara 1 y 2 respectivamente. En la figura 2.8 se muestra un ejemplo de puntos obtenidos para caracterizar la superficie de trabajo.

2. DESARROLLO DE UN CONTROL BASADO EN VISIÓN

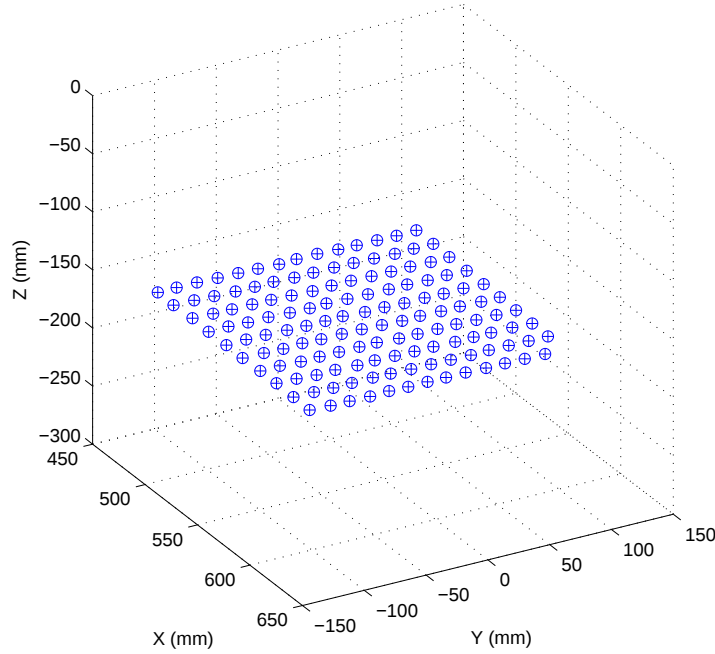


Figura 2.8: Caracterización de la superficie de trabajo del robot a partir de las matrices de proyección de cámara.

Una vez calculadas las coordenadas 3D de los puntos proyectados sobre la superficie, es posible calcular una ecuación que se ajuste a dichos puntos. En este caso se utiliza la ecuación de una superficie plana, como se describe a continuación.

Para lograr una correcta orientación del efector final del robot se requiere conocer la orientación del plano de trabajo. Teniendo un número n determinado de puntos tridimensionales ($n \geq 3$) sobre el plano de trabajo es posible calcular la ecuación que define dicho plano. La ecuación paramétrica de una superficie plana está definida como sigue:

$$g = a_1X + a_2Y + a_3Z = 1 \quad (2.32)$$

donde los parámetros a definen un plano en el espacio con respecto al sistema coordenado del robot. Para conocer la orientación del plano calculado es necesario calcular un vector perpendicular a dicho plano. Este vector se obtiene a partir de la siguiente expresión:

$$n = \frac{\nabla g}{|\nabla g|}$$

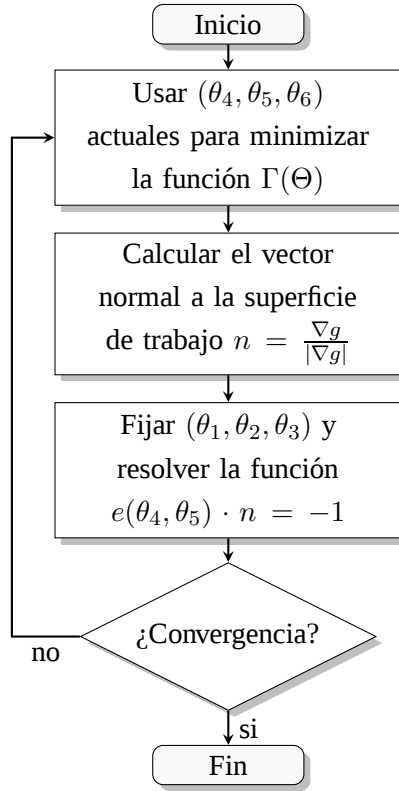


Figura 2.9: Proceso para calcular la posición y orientación del manipulador

donde n representa el vector normal a la superficie y g es la ecuación que define una superficie plana (ecuación 2.32). Finalmente la orientación del manipulador perpendicular a la superficie se obtiene solucionando la siguiente expresión:

$$e(\theta_4, \theta_5) \cdot n = -1$$

donde e es una función definida por el modelo cinemático nominal del robot y que depende de las articulaciones 4 y 5 del manipulador, que definen la orientación del mismo, la articulación 6 permanece fija durante la maniobra.

El proceso completo para lograr la posición y orientación del efector final utilizando el método CSM-Lineal se muestra en la figura 2.9.

En este capítulo se describió la estrategia de control por visión basada en un modelo lineal, en el siguiente capítulo se describe un análisis en simulación de las variables involucradas en dicha metodología.

Análisis de sensibilidad del modelo lineal de cámara y la metodología de manipulación en espacio de cámara

En este capítulo se presenta un análisis de sensibilidad de la metodología LCM-CSM propuesta en este trabajo utilizando datos simulados. Se presentan una serie de simulaciones utilizando datos a los que se les inyecta un error aleatorio, simulando los errores presentes en las detecciones de la información visual. Posteriormente con esa información se trata de reconstruir el modelo original y finalmente se mide la diferencia entre ambos. Esto con la finalidad de medir la sensibilidad del modelo a errores presentes en las mediciones. A continuación se describe de manera general la plataforma de simulación utilizada para dicho análisis.

3.1 Plataforma de pruebas para la simulación

Con la finalidad de realizar un análisis más detallado de los factores que influyen en el error de posicionamiento del robot, se propuso realizar una simulación del sistema de control basado en visión con el que ya se cuenta en el laboratorio de mecatrónica de la

3. ANÁLISIS DE SENSIBILIDAD

facultad de ingeniería de la Universidad Autónoma de San Luis Potosí, dicha plataforma se describe con más detalle en la sección de plataforma de pruebas experimentales. Posteriormente la plataforma real será utilizada para validar experimentalmente los resultados obtenidos en simulación.

La simulación del sistema consistió en dos etapas principales:

- Simulación de la posición tridimensional del robot y de las marcas visuales colocadas en el efector final del mismo.
- Simulación de las proyecciones de dichas marcas visuales sobre las imágenes de las cámaras utilizadas en el sistema de visión.

Para simular la posición tridimensional del manipulador y de las marcas visuales se hace uso del modelo cinemático nominal del robot. Este modelo nos permite conocer la posición y orientación del robot con respecto a un sistema coordinado de referencia, a partir de los valores angulares de las juntas. En el caso de las proyecciones de las marcas visuales en el plano de imagen se requiere conocer los parámetros de calibración de las cámaras y su posición y orientación con respecto a un sistema coordinado conocido.

Esta simulación del sistema real nos permitirá analizar los parámetros del sistema de visión de una manera independiente, cosa que no es posible usando el sistema real. A continuación se describe la obtención de las matrices de proyección de cámara utilizadas para generar los datos simulados.

3.1.1 Modelo de Cámara

Para el caso del modelo de cámara, se asume que los datos proyectados son acorde a un modelo de cámara de orificio. Para obtener el modelo de cada una de las dos cámaras del sistema de visión se realizó un proceso de calibración utilizando un total de 1800 correspondencias de puntos entre el espacio tridimensional y su proyección sobre las imágenes generadas por ambas cámaras. Con esa información se calcularon las matrices de proyección de cámara asociadas a cada sensor de visión.

Las matrices obtenidas para cada una de las cámaras del sistema de visión son:

$$P_1 = \begin{bmatrix} 0.2297 & 1.0886 & -0.0667 & 466.1899 \\ 0.3479 & -0.1193 & -1.0392 & 232.2408 \\ -0.0003 & 0.0001 & -0.0001 & 1 \end{bmatrix}$$

$$P_2 = \begin{bmatrix} -0.5313 & 1.0152 & -0.0485 & 866.3306 \\ 0.3373 & 0.1477 & -1.0726 & 247.3176 \\ -0.0003 & -0.0001 & -0.0001 & 1 \end{bmatrix}$$

A partir de las matrices de cámara anteriores, se pueden obtener los parámetros involucrados en el proceso de formación de las imágenes, a partir de un proceso descomposición RQ , tal como se describe en [\[Hartley and Zisserman, 2004\]](#).

Los valores de los parámetros obtenidos para el caso de la cámara 1, acorde a la expresión 2.7 se muestran a continuación:

$$K_1 = \begin{bmatrix} 3305.1703 & 2.4820 & 594.5468 \\ 0 & 3309.0586 & 282.9653 \\ 0 & 0 & 1 \end{bmatrix} \quad t_1 = \begin{bmatrix} 2612.8520 \\ -905.9403 \\ 1202.2634 \end{bmatrix}$$

$$R_1 = \begin{bmatrix} 0.3615 & 0.9322 & 0.0140 \\ 0.3892 & -0.1372 & -0.9108 \\ -0.8472 & 0.3347 & -0.4124 \end{bmatrix}$$

Los valores obtenidos para la cámara 2 son los siguientes:

$$K_2 = \begin{bmatrix} 3520.2979 & -5.8208 & 685.0832 \\ 0 & 3528.4334 & 355.2335 \\ 0 & 0 & 1 \end{bmatrix} \quad t_2 = \begin{bmatrix} 2810.5618 \\ 675.2714 \\ 1207.4311 \end{bmatrix}$$

$$R_2 = \begin{bmatrix} -0.3017 & 0.9527 & 0.0363 \\ 0.3866 & 0.1571 & -0.9087 \\ -0.8715 & -0.2601 & -0.4158 \end{bmatrix}$$

Para completar la simulación se requiere generar puntos tridimensionales para establecer la posición física de las marcas visuales con respecto al sistema coordenado unido a la base del robot. Para este fin se puede utilizar el modelo cinemático nominal del manipulador.

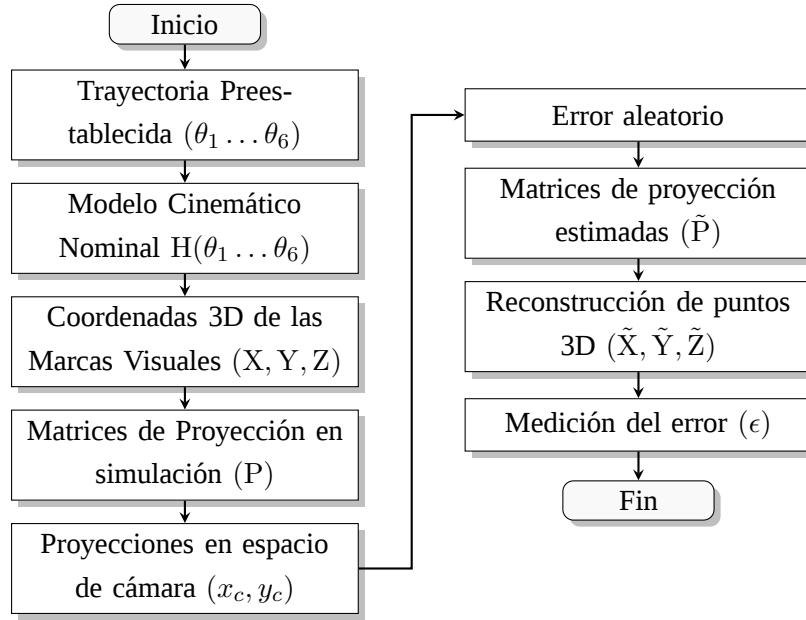


Figura 3.1: Proceso para generar información a partir de la plataforma de simulación.

3.1.2 Modelo cinemático nominal

En el caso de la simulación del robot manipulador se utilizó el modelo cinemático nominal del robot descrito en el Apéndice A, el cual fue obtenido a partir de las especificaciones del fabricante. Tal como ya se mencionó anteriormente el modelo cinemático permite generar la posición tridimensional del efector final del robot, a partir de los valores angulares de las juntas del robot. Con la información de una serie de puntos tridimensionales y su correspondencia en el plano de imagen obtenido a través del modelo de cámara es posible generar una simulación del sistema de visión real. Para posteriormente poder analizar el efecto de los errores presentes en los datos mediante la adición de error aleatorio, con el fin de simular errores presentes en las mediciones en espacio de cámara.

3.1.3 Validación de la plataforma de simulación

Con la finalidad de validar la plataforma de simulación se realizó una prueba simulando una trayectoria pre-establecida del robot. Esta trayectoria pre-establecida consistió en colocar el robot en 34 posiciones diferentes proporcionando los valores angulares del robot. Estas posiciones fueron definidas de manera aleatoria de manera que los puntos cubran un área amplia del espacio de imagen en cada una de las cámaras del sistema de visión.

A través del modelo cinemático nominal se calculan las coordenadas tridimensionales de una marca visual colocada en el efector final, cuya posición es conocida con respecto al origen coordinado del robot. Posteriormente utilizando las matrices de proyección calculadas para la plataforma de simulación, se realiza el cálculo de las proyecciones en espacio de cámara de la marca visual utilizando sus coordenadas tridimensionales empleando las expresiones 2.12. Lo anterior genera 34 correspondencias de puntos entre el espacio 3D del robot y el espacio 2D de las imágenes.

A partir de las 34 correspondencias obtenidas se hizo una re-estimación de la matriz de proyección de cada una de las cámaras empleando la ecuación 2.7. Finalmente se vuelven a calcular las proyecciones en espacio de cámara de los 34 puntos con las nuevas matrices. El error se define como la distancia entre el punto generado con la plataforma de visión y el punto estimado usando las nuevas matrices de proyección. Los resultados obtenidos se muestran en la figura 3.2. El error promedio obtenidos para ambas cámaras fue de cero píxeles con una desviación estándar de cero.

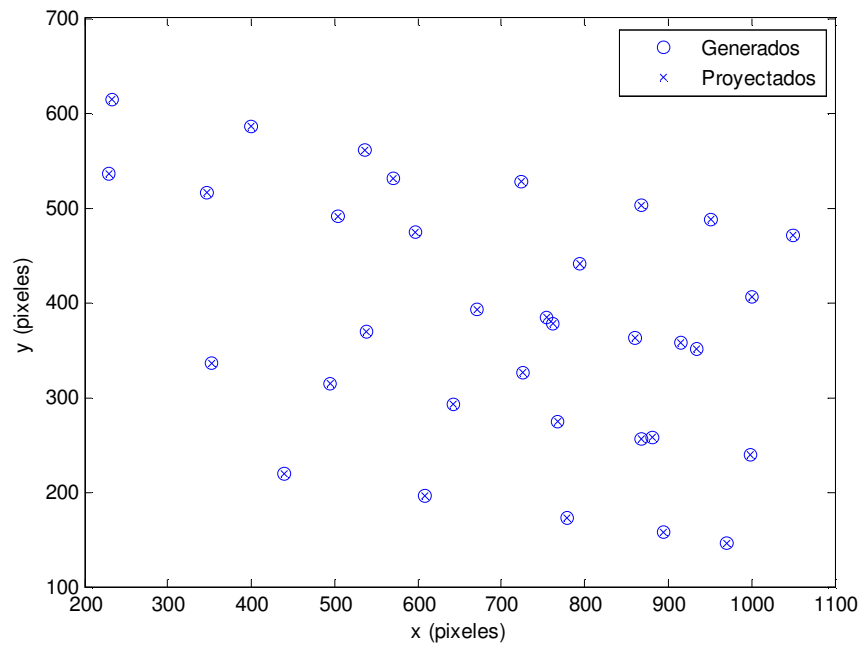
Tal como se muestra en la fig. 3.2, los datos generados en la plataforma de simulación y las proyecciones obtenidas a través de la trayectoria pre-planeada coinciden. Esto verifica que los parámetros obtenidos para la plataforma de simulación son correctos.

Una vez validada la plataforma de simulación, es posible agregar error aleatorio a los datos generados con la finalidad de simular errores en las mediciones en espacio de cámara y analizar la sensibilidad del sistema de visión con respecto a esos errores. Con este fin se realizó un análisis de sensibilidad tomando en cuenta los siguientes aspectos:

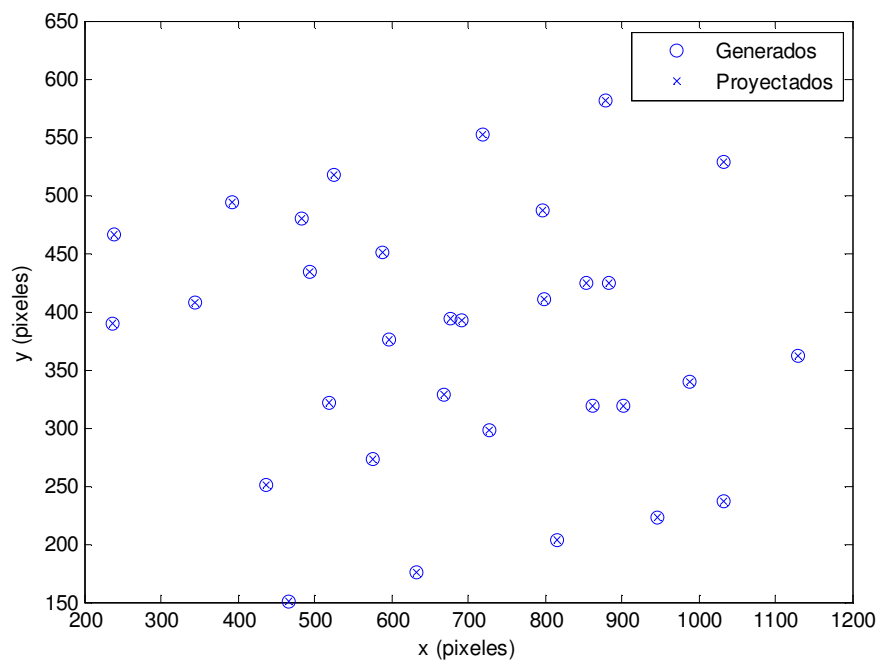
- Sensibilidad a los errores aleatorios presentes en las mediciones en espacio de cámara.
- Variación en el número de muestras utilizadas para calcular el modelo de cámara.
- Efecto de la resolución de las cámaras en la detección de la información visual.
- Influencia de la información visual actualizada durante la maniobra del manipulador.

En los siguientes párrafos se describen cada una de las simulaciones mencionadas anteriormente.

3. ANÁLISIS DE SENSIBILIDAD



(a) Cámara 1



(b) Cámara 2

Figura 3.2: Datos generados en la plataforma de simulación vs proyecciones usando trayectoria pre-planeada para cámara 1 y 2 respectivamente.

3.2 Sensibilidad del modelo de cámara a errores presentes en las mediciones

Para analizar el efecto de los errores en las mediciones en espacio de cámara sobre el cálculo del modelo lineal, se realizó una simulación de Montecarlo. Esta simulación fue dividida en las siguientes tareas:

- Se tomaron una serie de puntos tridimensionales directamente del robot.
- Se buscó abarcar la mayor cantidad de espacio de trabajo del manipulador, con la restricción de que fuera visible en las dos cámaras de la plataforma de visión.
- Se utilizaron los 34 puntos de la trayectoria pre-planeada, y se añadieron 3 configuraciones más. Para las otras 3 configuraciones se generaron 4 planos dentro del área de trabajo del robot. En estos planos se modificó la distancia entre cada punto con la finalidad de aumentar la densidad de información en cada plano, por lo tanto se generaron cuatro configuraciones con 34, 92, 315 y 1828 puntos. Las configuraciones de puntos utilizadas se muestran en la figura 3.3.

Para cada una de las cuatro configuraciones de datos se realizó el siguiente proceso:

- Se calcularon las proyecciones del modelo lineal a partir de la expresión 2.12 usando la plataforma de simulación.
- Se realizó el cálculo del modelo lineal a partir de los datos obtenidos y se verificó que el error de estimación acorde a las ecuaciones 2.12 fuera igual a cero, con la finalidad de validar que los datos simulados eran correctos.
- Se inyectó ruido aleatorio a los datos simulando errores en las mediciones en espacio de cámara. Para esto se asumieron dos tipos de errores, con una distribución normal y con una distribución uniforme.

3.2.1 Simulación de errores con una distribución normal

Para realizar la simulación en cada iteración se utilizaron $(n - 1)$ puntos, donde n es el número total de muestras utilizado para calcular el modelo lineal a partir de la expresión 2.7, posteriormente se realizó la estimación del punto n , usando las ecuaciones 2.12. Este proceso se realizó para cada uno de los puntos muestra. El error fue definido como la

3. ANÁLISIS DE SENSIBILIDAD

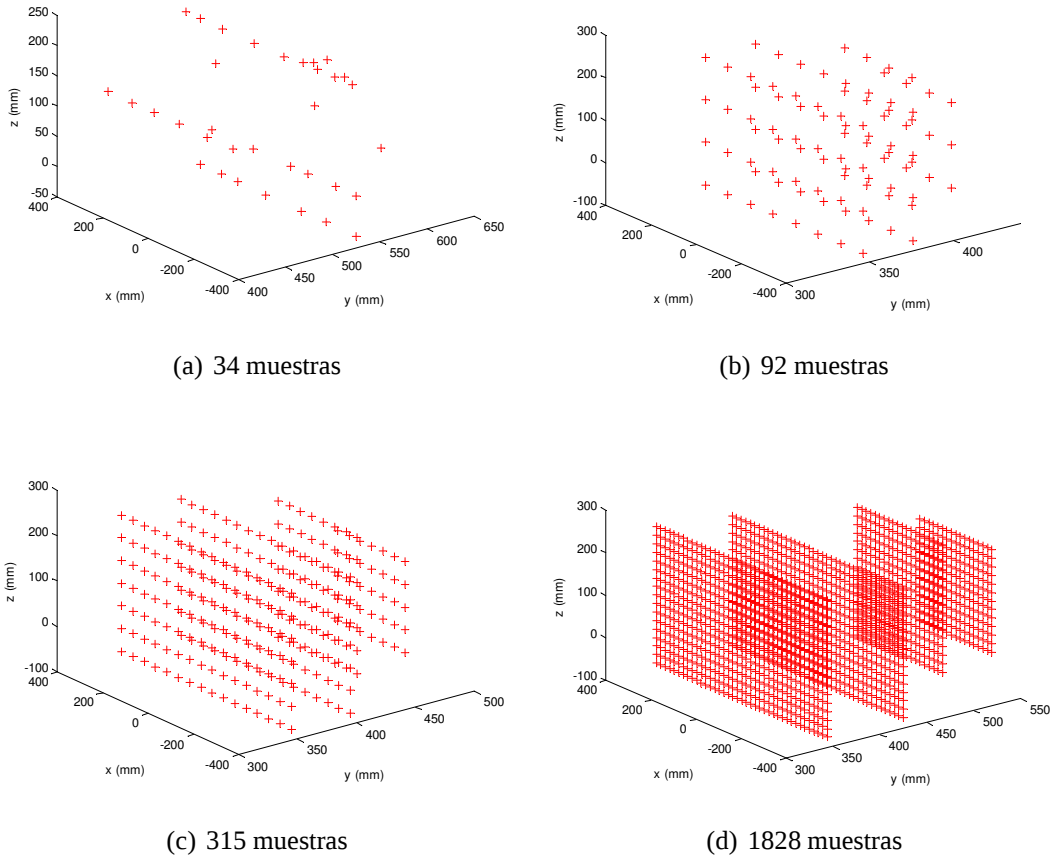


Figura 3.3: Diferentes configuraciones de datos utilizadas para el análisis de sensibilidad del modelo lineal.

distancia entre el punto estimado usando los datos con ruido aleatorio y el punto generado directamente a partir de la plataforma de simulación descrita en la sección 3.1.

Para generar el ruido aleatorio con una distribución normal se utilizó la expresión:

$$y = f(x|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

donde μ es la media y σ la desviación estándar.

En el cuadro 3.1 se muestran los resultados de la simulación mencionada anteriormente. Como se puede observar se realizaron 5 procesos para cada una de las cuatro configuraciones de datos. En cada proceso se agregó ruido aleatorio con una media de cero y una desviación estándar de 0.2, 0.5, 1, 2 y 5 píxeles de manera independiente en cada caso.

3.2 Sensibilidad del modelo de cámara a errores presentes en las mediciones

Tal como se aprecia en el cuadro 3.1, el mejor caso se presenta cuando la magnitud del error agregado es más pequeño. Como era de esperarse, el peor caso se presenta cuando el error inyectado tiene una desviación estándar de 5 píxeles y el modelo no puede reconstruir los puntos de manera correcta. En la columna de error promedio se puede observar que conforme aumenta el número de puntos usados para calcular el modelo lineal, el error promedio disminuye en todos los casos. Como ya se había mencionado anteriormente, esto es atribuible a que al emplear mayor cantidad de información los errores aleatorios se compensan y tienden a cero. Cabe mencionar que al aumentar el número de muestras, además de que el valor promedio del error disminuye, también disminuyen los valores máximos del error y llegan a ser menores que los valores máximos del ruido inyectado.

En la figura 3.6 se muestra la distribución de los errores en las proyecciones de cámara para uno de los casos realizados en la simulación descrita anteriormente y que aparece de manera numérica en el cuadro 3.1. En este caso se muestra la prueba en la que se emplearon 1828 datos tal como aparece en la figura 3.5, y se agregó a los datos un error aleatorio con una media de cero y una desviación estándar de 1 pixel. En la figura se graficó el error promedio de cada uno de los puntos para un total de 100 iteraciones. En esta figura se puede observar la separación de los puntos que corresponden a cada uno de los cuatro planos utilizados para calcular el modelo de cámara. Además se puede observar que los valores mínimos correspondientes a cada uno de los planos se localizan en el centro de dichos planos, esto es debido a que en el centro se tiene información en todas direcciones del plano y además de forma simétrica. Finalmente como era de esperarse, los errores más pequeños se encuentran ubicados aproximadamente en el centro de todo el conjunto de muestras, esto debido a que en esa área se tiene información simétrica en todas direcciones para el cálculo del modelo de cámara. Lo anterior se puede visualizar más claramente en las figuras 3.7, 3.8, 3.9, 3.10, que corresponden a los errores estimados de los planos 1, 2, 3 y 4 respectivamente de la plataforma de simulación (fig. 3.5)

En la figuras 3.11 y 3.12 se presentan de manera gráfica los resultados mostrados en el cuadro 3.1. En la figura 3.11 se puede observar que a mayor magnitud del error inyectado se obtiene un error mayor en las estimaciones del modelo lineal, sin embargo, en todos los casos conforme el número de muestras se incrementa, los valores promedio del error disminuyen y eventualmente llegarán a ser cero.

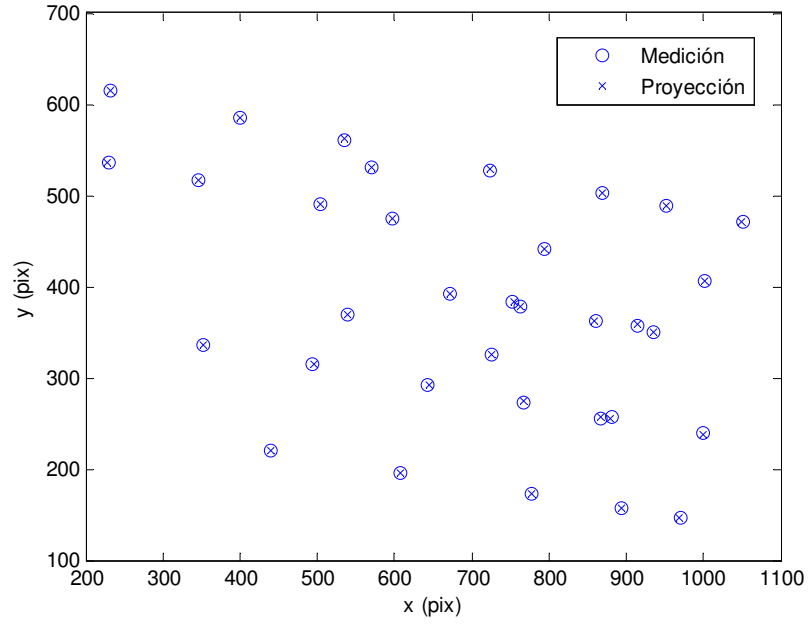
En la figura 3.13 se pueden observar los puntos que corresponden al error inyectado para el caso donde se utilizaron 1828 muestras. En este caso se grafica el error que representa una distribución normal con una media de cero y una desviación estándar de 1

3. ANÁLISIS DE SENSIBILIDAD

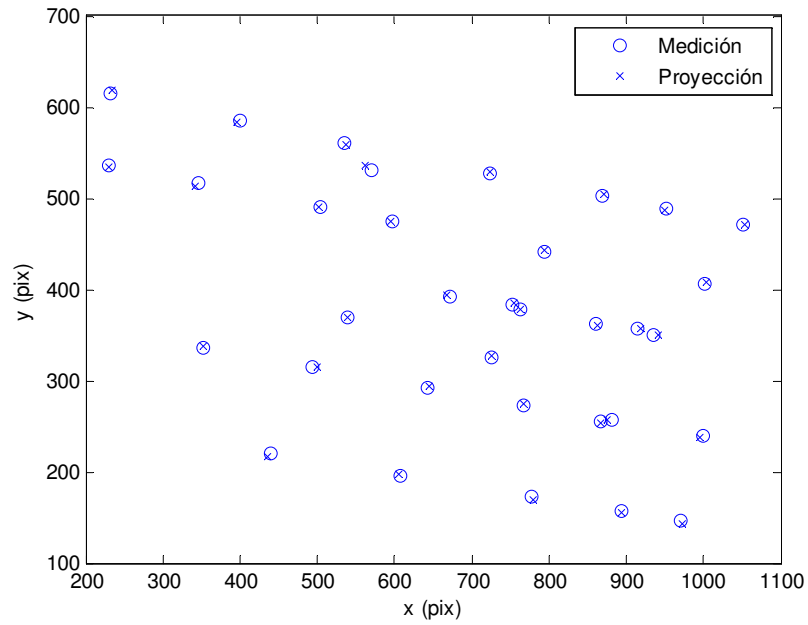
	Ruido Inyectado (pix)			Error Obtenido (pix)			
n	σ_r	min	max	min	max	promedio	σ_e
34	0.2	-0.42	0.61	0.0626	0.1898	0.1097	0.0587
92		-0.46	0.54	0.0322	0.1164	0.0678	0.0364
315		-0.53	0.56	0.0107	0.0777	0.0327	0.0173
1828		-0.75	0.75	0.0051	0.0297	0.0134	0.0056
34	0.5	-1.13	1.01	0.169	0.4664	0.275	0.151
92		-1.33	1.51	0.0843	0.2641	0.1564	0.083
315		-1.67	1.68	0.034	0.1733	0.0821	0.0419
1828		-1.55	1.57	0.0112	0.0823	0.0345	0.0177
34	1	-1.96	1.93	0.3179	0.9437	0.5621	0.3001
92		-2.47	2.04	0.1668	0.5403	0.3197	0.1689
315		-2.79	2.50	0.0647	0.3621	0.172	0.0857
1828		-3.65	3.83	0.0251	0.1875	0.0769	0.039
34	2	-3.43	3.17	0.6312	2.0821	1.1322	0.6194
92		-4.30	4.15	0.3502	1.3761	0.701	0.3695
315		-5.60	5.14	0.1403	1.0118	0.3878	0.1918
1828		-6.89	7.00	0.0553	0.6572	0.2036	0.0912
34	5	-11.56	11.05	1.5945	11.0507	3.3791	1.7256
92		-11.63	12.32	0.8126	7.0174	2.6309	1.13
315		-16.01	15.97	0.3926	5.1575	1.628	0.6092
1828		-19.50	18.78	0.2112	3.432	1.0228	0.5848

Cuadro 3.1: Datos arrojados del análisis de sensibilidad a errores en las proyecciones en espacio de cámara del modelo lineal.

3.2 Sensibilidad del modelo de cámara a errores presentes en las mediciones



(a) $\sigma_r = 1$ píxel



(b) $\sigma_r = 5$ píxeles

Figura 3.4: Mediciones vs proyecciones aplicando un error con una distribución normal con una media de cero y una desviación estándar de (a)1 píxel, (b)5 píxeles.

3. ANÁLISIS DE SENSIBILIDAD

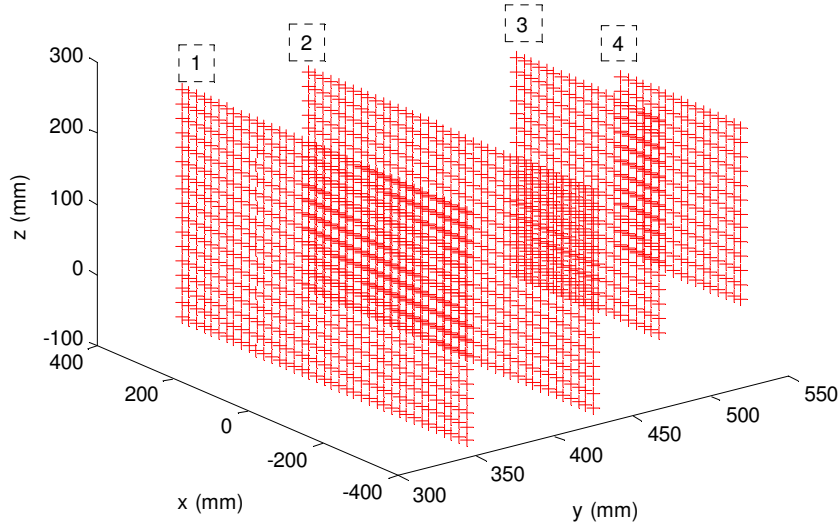


Figura 3.5: Muestras contenidas en 4 planos utilizados para el análisis de sensibilidad en la plataforma de simulación (1828 datos).

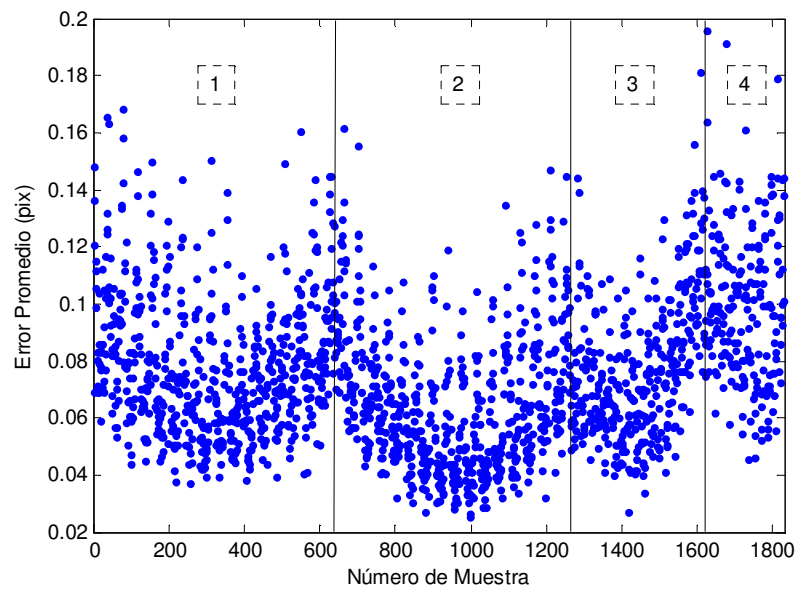
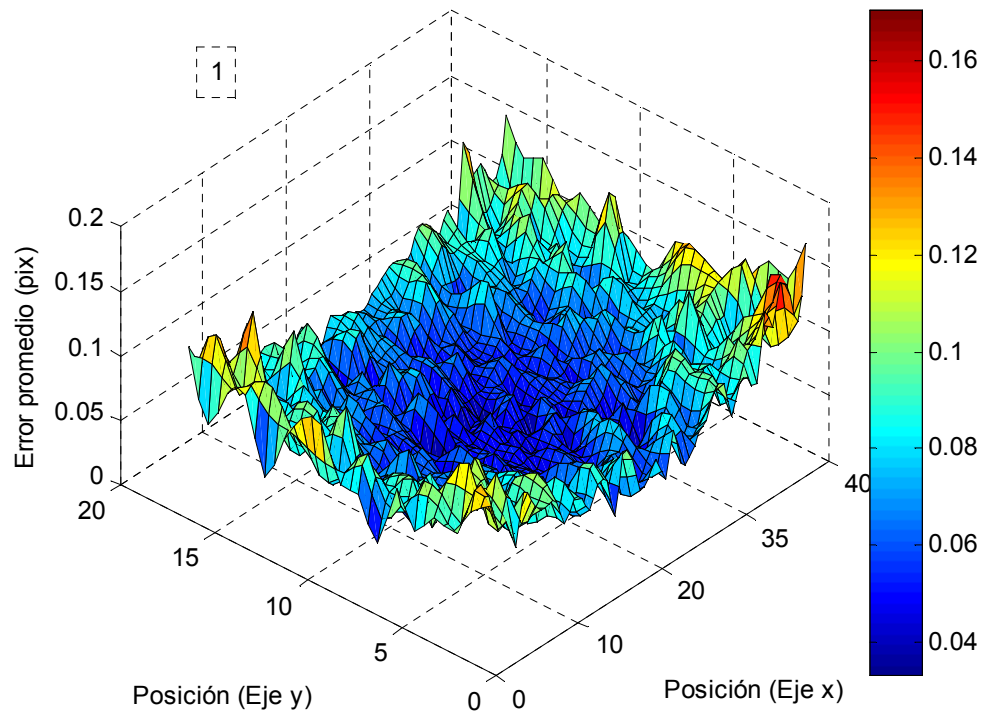
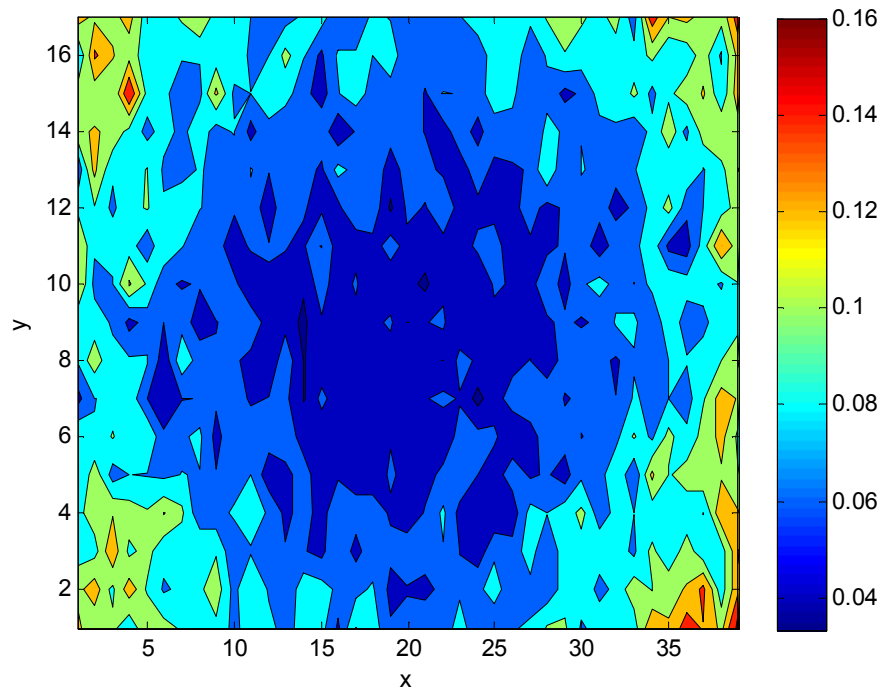


Figura 3.6: Errores obtenidos aplicando una simulación de Montecarlo al modelo lineal de cámara empleando 1828 datos, a los cuales se les agregó ruido aleatorio utilizando una distribución normal con una media de cero y una desviación estándar de 1 pixel.



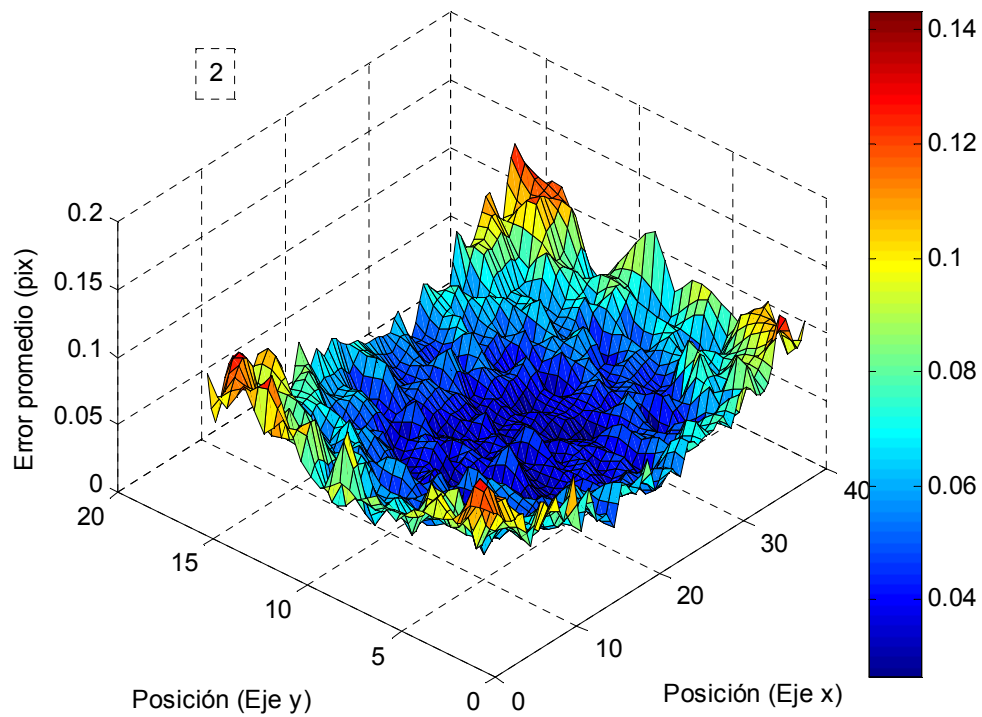
(a) 3D



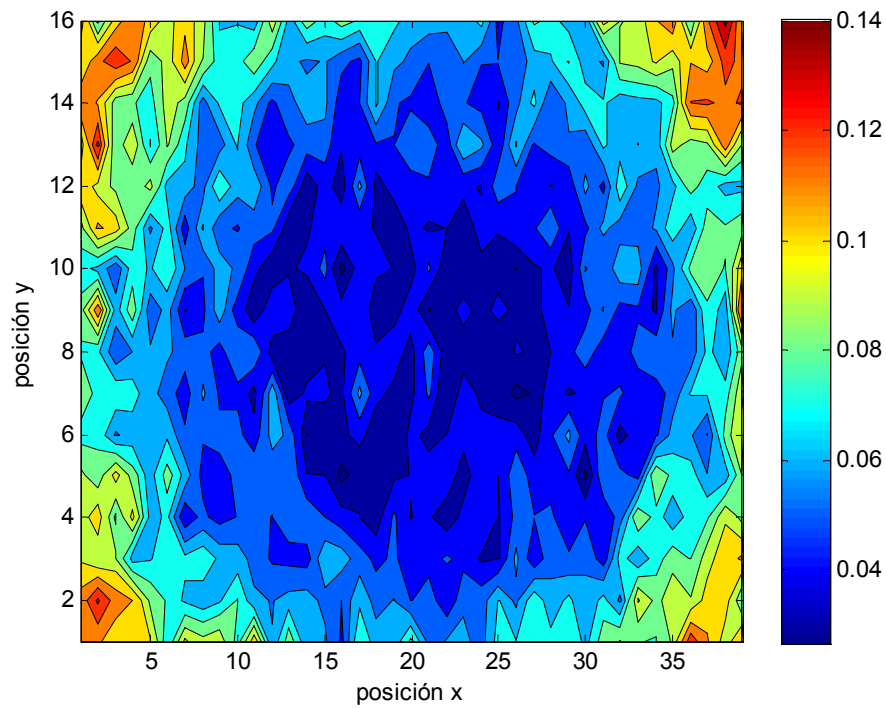
(b) 2D

Figura 3.7: Magnitud del error de estimación correspondiente al plano 1 de la plataforma de simulación.

3. ANÁLISIS DE SENSIBILIDAD

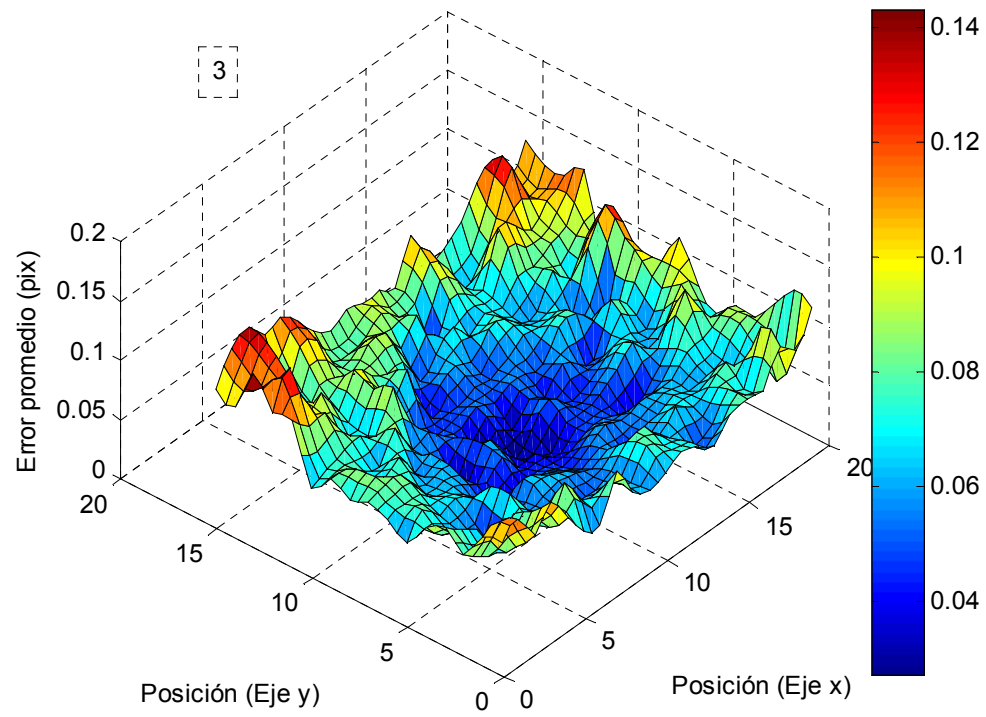


(a) 3D

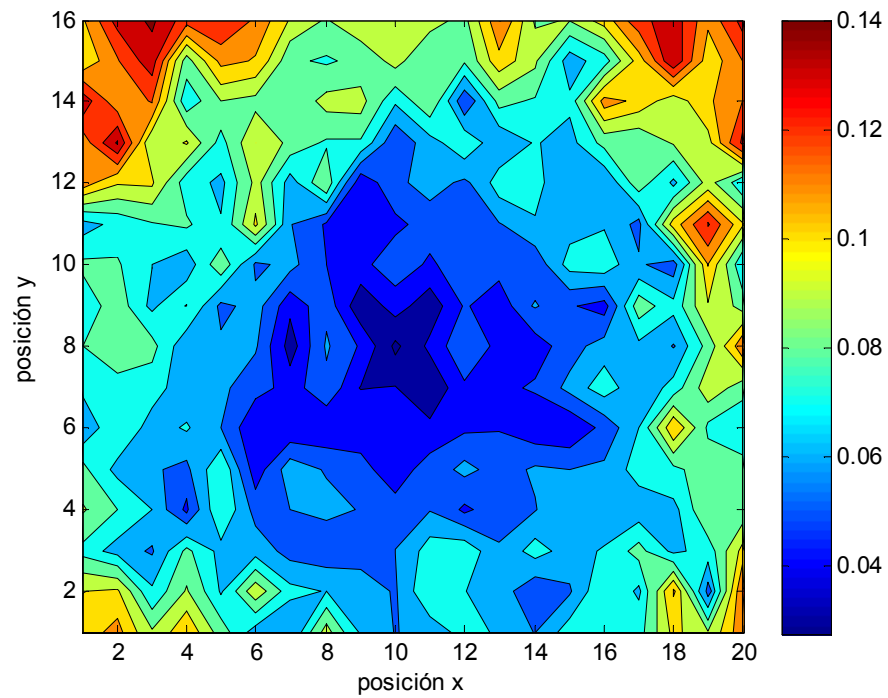


(b) 2D

Figura 3.8: Magnitud del error de estimación correspondiente al plano 2 de la plataforma de simulación.



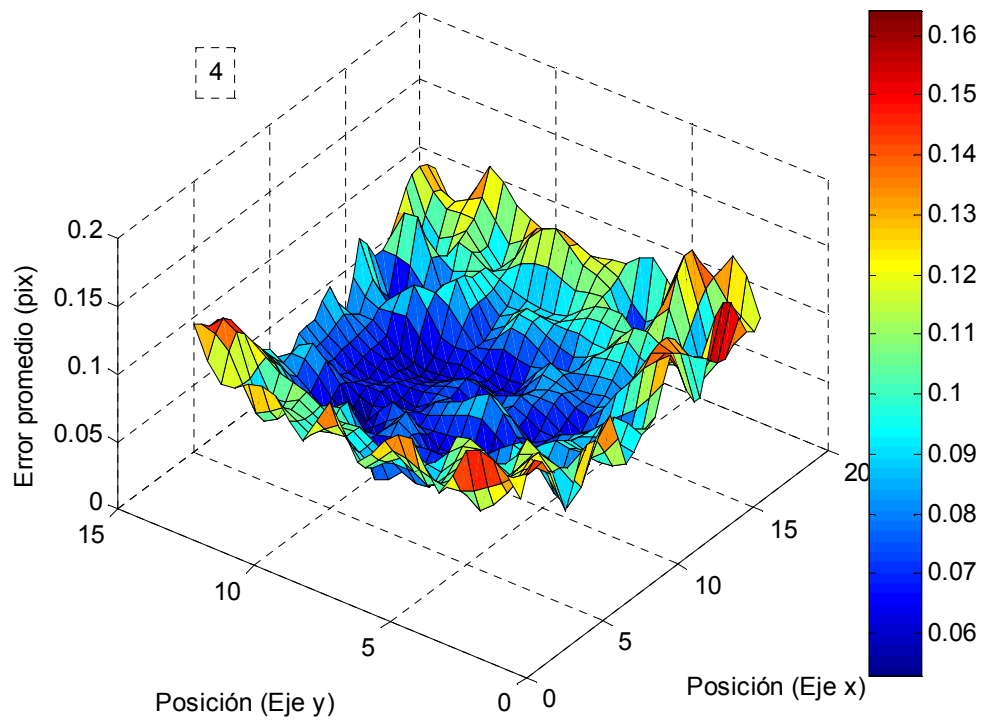
(a) 3D



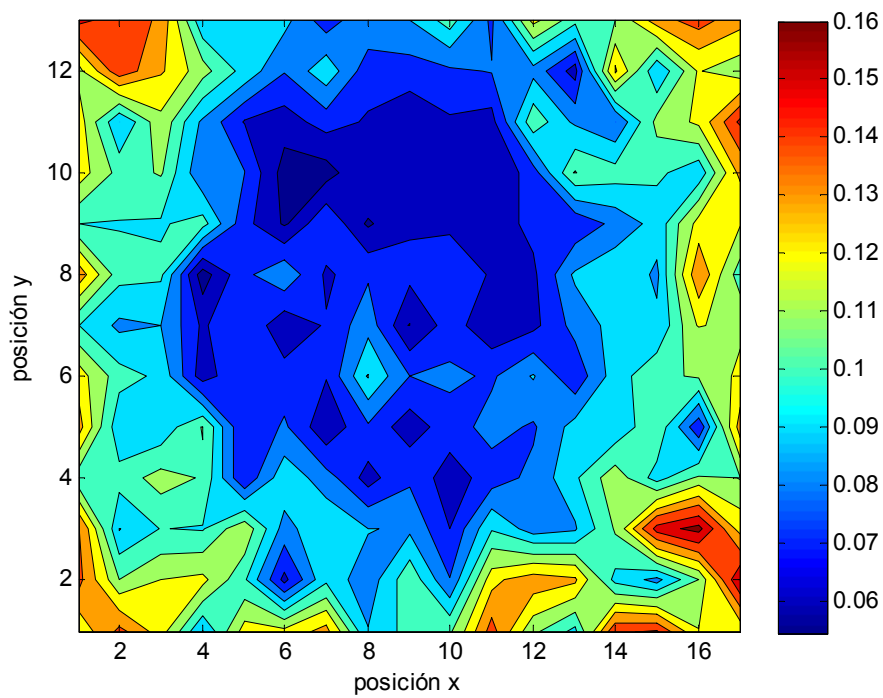
(b) 2D

Figura 3.9: Magnitud del error de estimación correspondiente al plano 3 de la plataforma de simulación.

3. ANÁLISIS DE SENSIBILIDAD



(a) 3D



(b) 2D

Figura 3.10: Magnitud del error de estimación correspondiente al plano 4 de la plataforma de simulación.

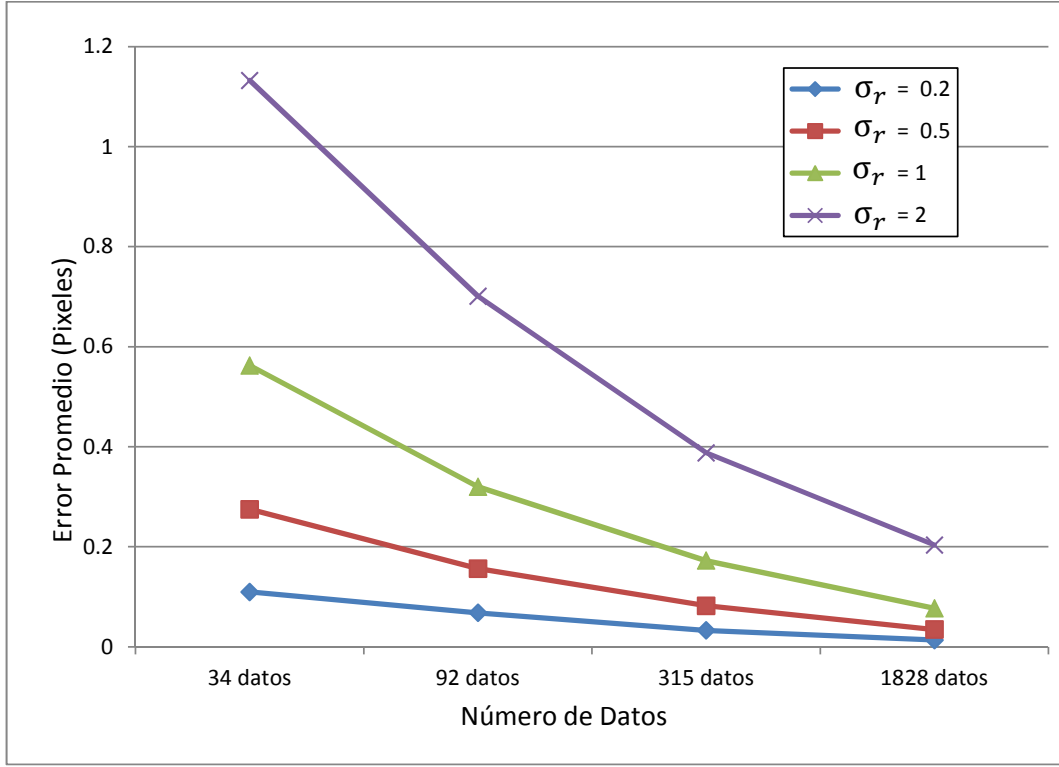


Figura 3.11: Error de estimación del modelo de cámara lineal utilizando datos simulados a los que se les inyectó error aleatorio con una distribución normal, con una media de cero píxeles y una desviación estándar de 0.2, 0.5, 1 y 2 píxeles.

pixel, se puede apreciar en la figura que la magnitud de los errores van desde -3.5 a 3.5 píxeles.

3.2.2 Simulación de errores con una distribución uniforme

Para este caso el ruido inyectado a los datos se generó de tal manera que representara una distribución uniforme, tal como se muestra en la figura 3.14. Se utilizaron cuatro configuraciones de error aleatorio generado, con valores comprendidos entre -0.2 y 0.2 , -0.5 y 0.5 , -1 y 1 , y finalmente con valores de -2 y 2 píxeles. Para generar el error aleatorio con una distribución uniforme con valores comprendidos entre a y b se utilizó la siguiente expresión:

$$y = f(x|a, b) = \frac{x - a}{b - a}.$$

En la figura 3.15 se muestra la distribución de los errores promedio obtenidos para el caso en el que se utilizaron 1828 datos. En la figura 3.16 se muestran los errores prome-

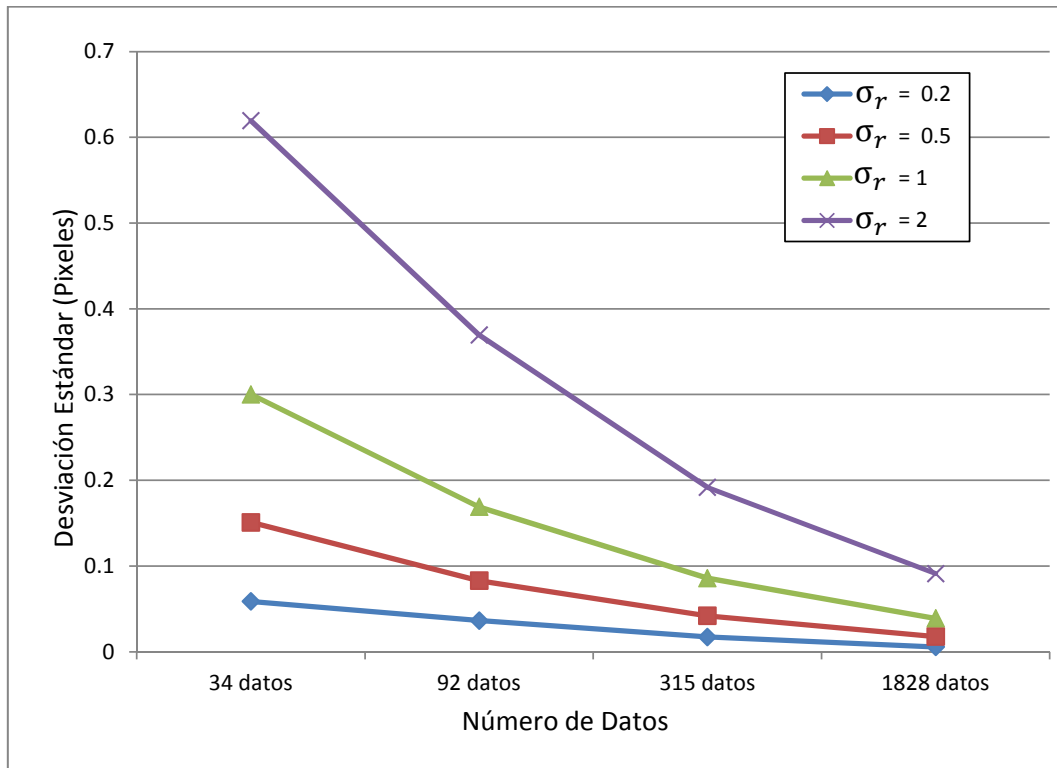


Figura 3.12: Desviación estándar del modelo de cámara lineal utilizando datos simulados a los que se les inyectó error aleatorio con una distribución normal, con una media de cero píxeles y una desviación estándar de 0.2, 0.5, 1 y 2 píxeles.

dio obtenidos y en la figura 3.17 se muestra la desviación estándar. Se observa que el comportamiento es similar al presentado anteriormente cuando se aplicó error con una distribución normal.

3.3 Sensibilidad de posicionamiento del robot a errores presentes en las mediciones

En esta sección se presentan los resultados en simulación del error de posicionamiento del robot, con respecto a errores presentes en las mediciones en espacio de cámara. Primeramente se analiza simulando errores en las mediciones con una distribución normal y posteriormente con una distribución uniforme.

3.3 Sensibilidad de posicionamiento del robot a errores presentes en las mediciones

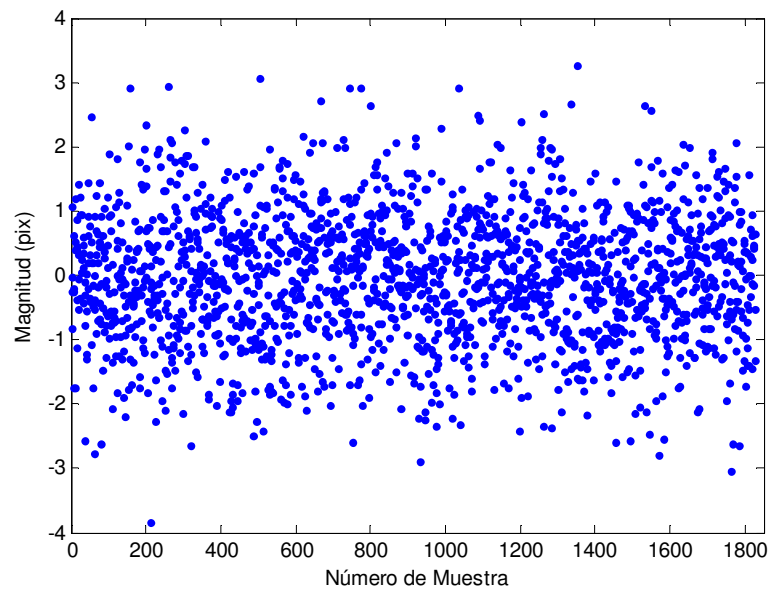


Figura 3.13: Error aleatorio inyectado a los datos, correspondiente a una distribución normal con una media de cero y una desviación estándar de 1 pixel.

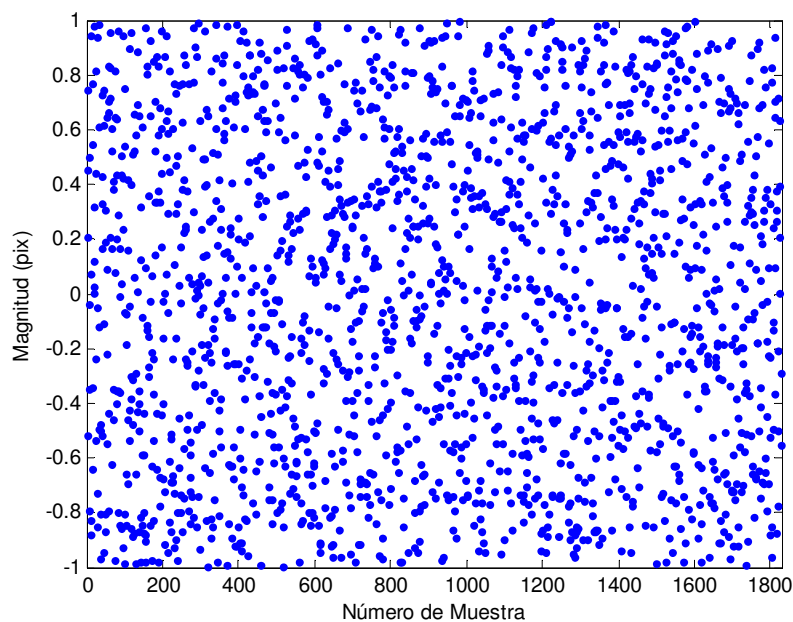


Figura 3.14: Error aleatorio generado, correspondiente a una distribución uniforme con valores comprendidos entre 1 y -1 píxeles.

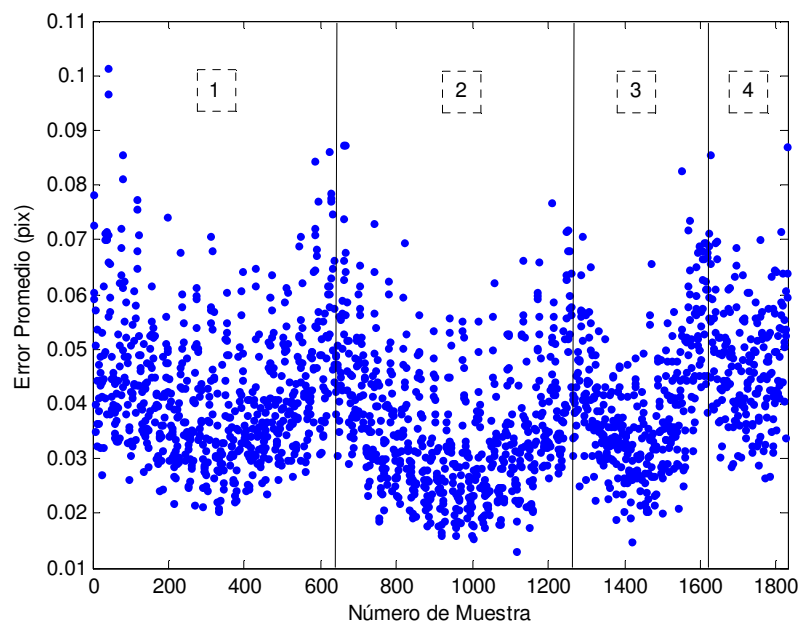


Figura 3.15: Errores promedio obtenidos aplicando una simulación de Montecarlo al modelo lineal de cámara utilizando 1828 puntos, a los que les fue agregado ruido aleatorio con una distribución uniforme con valores comprendidos entre -1 y 1 píxel.

3.3.1 Simulación de errores con una distribución normal

Para medir el efecto de los errores presentes en el modelo de cámara sobre el error de posicionamiento final del robot, se realizó un análisis de Montecarlo usando la plataforma de simulación descrita al inicio de esta sección. A continuación se define de manera general este análisis.

Descripción de la simulación

Para este caso se realizó una simulación para medir el efecto de los errores aleatorios presentes en el modelo de cámara, sobre el error de posicionamiento 3D del manipulador (error en mm). De igual manera que la simulación anterior, se emplearon las cuatro configuraciones de datos generadas para la plataforma de simulación. De manera análoga a lo mostrado anteriormente, se añadió ruido aleatorio a los datos simulados para las dos cámaras del sistema de visión y posteriormente se calcularon los valores angulares del robot para posicionarse sobre un punto objetivo definido de manera arbitraria. A continuación se detallan las etapas de la simulación:

- Se generaron las proyecciones del modelo de cámara lineal para las cuatro configu-

3.3 Sensibilidad de posicionamiento del robot a errores presentes en las mediciones

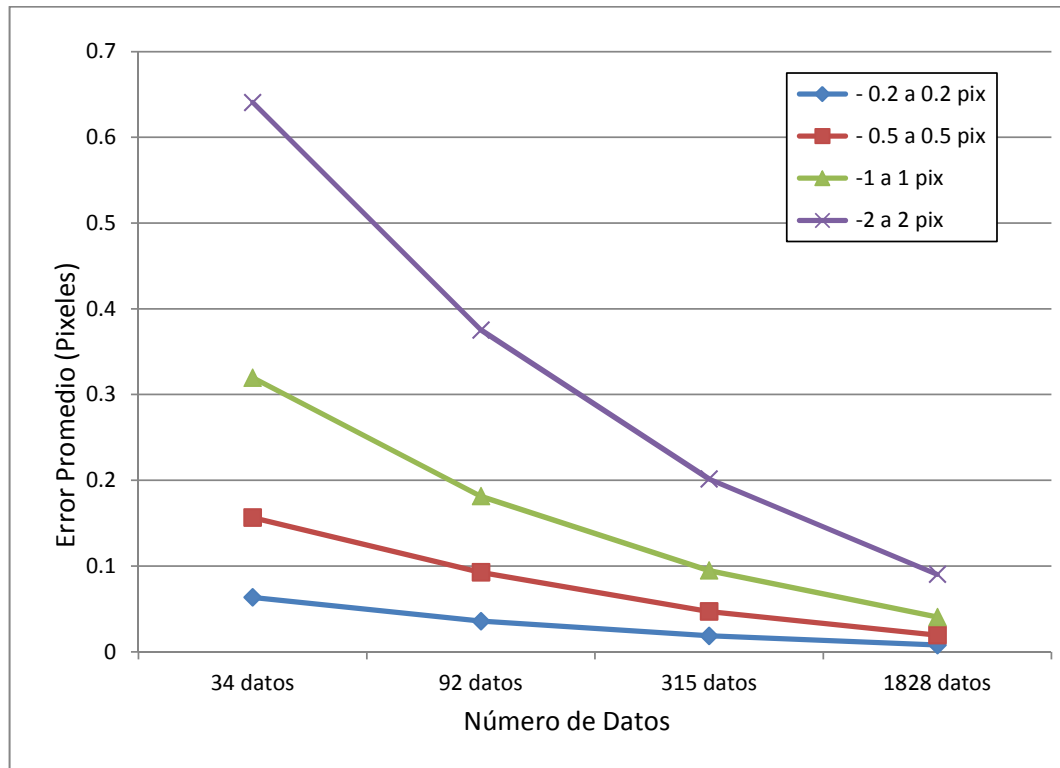


Figura 3.16: Error de estimación del modelo de cámara lineal utilizando datos simulados a los que se les inyectó error aleatorio con una distribución uniforme con valores comprendidos entre -0.2 y 0.2 ; -0.5 y 0.5 ; -1 y 1 ; -2 y 2 píxeles.

raciones de datos mencionadas anteriormente. Esto se realizó de manera independiente para las dos cámaras de la plataforma de visión.

- A las proyecciones generadas se les añadió error aleatorio de tipo gaussiano con una media de cero y una desviación estándar de 0.2 , 0.5 , 1 y 2 píxeles, simulando errores en la detección de la información en espacio de cámara.
- Se calculó el modelo lineal utilizando los datos con error para cada una de las cuatro configuraciones de puntos utilizando $(n - 1)$ puntos, siendo n el número total de datos. Esto se realizó de manera independiente para cada una de las cámaras de la plataforma de simulación.
- Posteriormente se calcularon las configuraciones angulares del robot para posicionarse sobre el punto objetivo n . Este proceso se realizó de manera iterativa para cada uno de los puntos utilizados en cada una de las cuatro configuraciones de datos mostrados en la figura 3.5.

3. ANÁLISIS DE SENSIBILIDAD

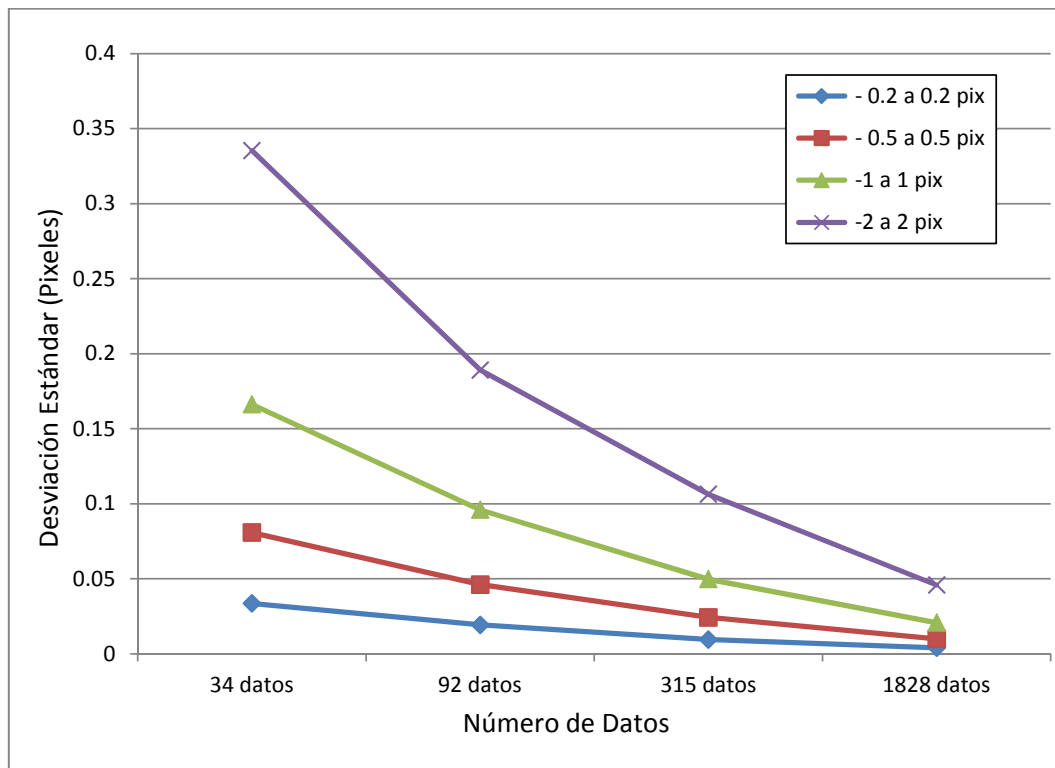


Figura 3.17: Desviación estándar del modelo de cámara lineal utilizando datos simulados a los que se les inyectó error aleatorio con una distribución uniforme con valores comprendidos entre -0.2 y 0.2 ; -0.5 y 0.5 ; -1 y 1 ; -2 y 2 píxeles.

- Finalmente se definió el error como la distancia euclidiana entre el punto 3D proyectado a través de la plataforma de simulación y el punto calculado a partir de los datos que incluyen error aleatorio.

En la figura 3.13 se pueden observar los puntos que corresponden al error inyectado para el caso donde se utilizaron 1828 muestras. En este caso se grafica el error que representa una distribución normal con una media de cero y una desviación estándar de 1 pixel, se puede apreciar en la figura que la magnitud de los errores está comprendida entre -4 y 4 píxeles.

En la figura 3.18 se muestran de manera gráfica los errores promedio de posicionamiento del robot obtenidos en la simulación de Montecarlo. Se puede apreciar que el valor promedio máximo del error es de aproximadamente 1 mm y se presenta al utilizar 34 puntos y una desviación estándar de 2 píxeles en el error inyectado. En esta figura se puede observar que conforme incrementa el número de datos para calcular el modelo de cámara, el error de posicionamiento tiende a cero. También se puede ver en la figura que

3.3 Sensibilidad de posicionamiento del robot a errores presentes en las mediciones

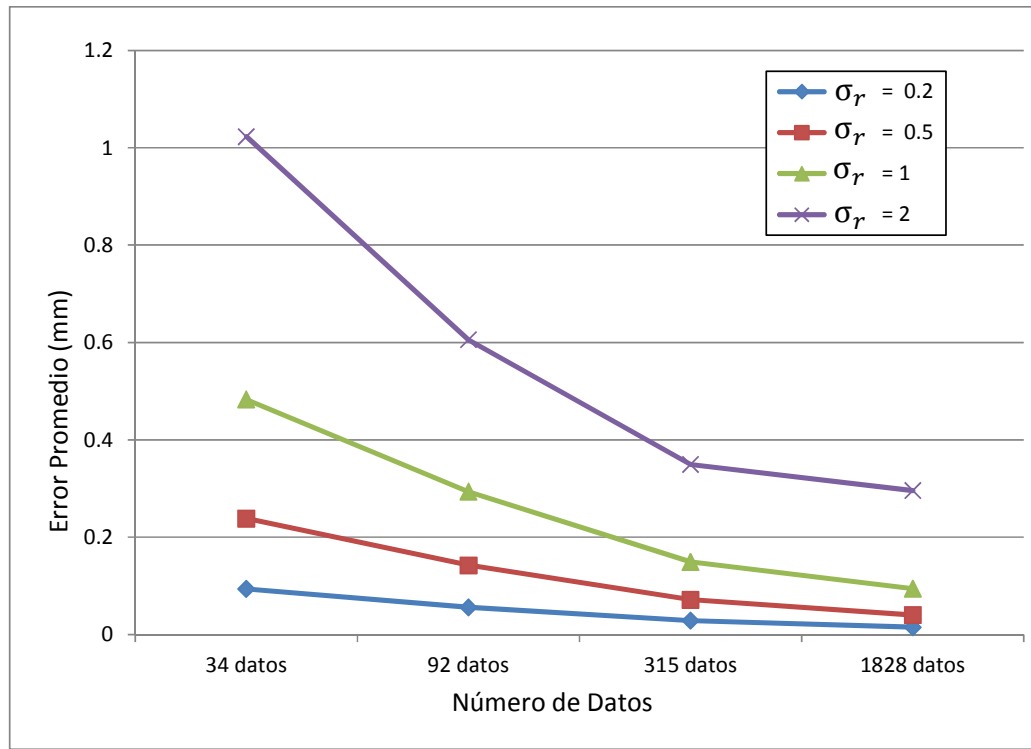


Figura 3.18: Errores promedio de posicionamiento del robot utilizando datos simulados a los que se les inyectó error aleatorio con una distribución normal y una media de cero píxeles para una desviación estándar de 0.2, 0.5, 1 y 2 píxeles.

la magnitud del error inyectado, que corresponde al error en la detección de las marcas visuales, está ligado también con el error de posicionamiento del robot, esto es, si se reduce el error en la detección de las marcas visuales es posible obtener mejores resultados en el posicionamiento del robot.

En la figura 3.19 se muestra la distribución de los errores en cada uno de los ejes coordenados para el caso de 1828 datos, a los que se les agregó ruido aleatorio con una media de cero y una desviación estándar de 1 pixel. También se muestra la métrica del error. Se puede apreciar en la figura 3.19(d), que el error de posicionamiento es menor a 0.5 mm en todos los casos.

La figura 3.20 presenta el diagrama de caja de los errores calculados. Se puede apreciar que en los ejes X, Y, Z el promedio de error es prácticamente cero. En la figura 3.21, se muestra el histograma obtenido en la simulación.

3. ANÁLISIS DE SENSIBILIDAD

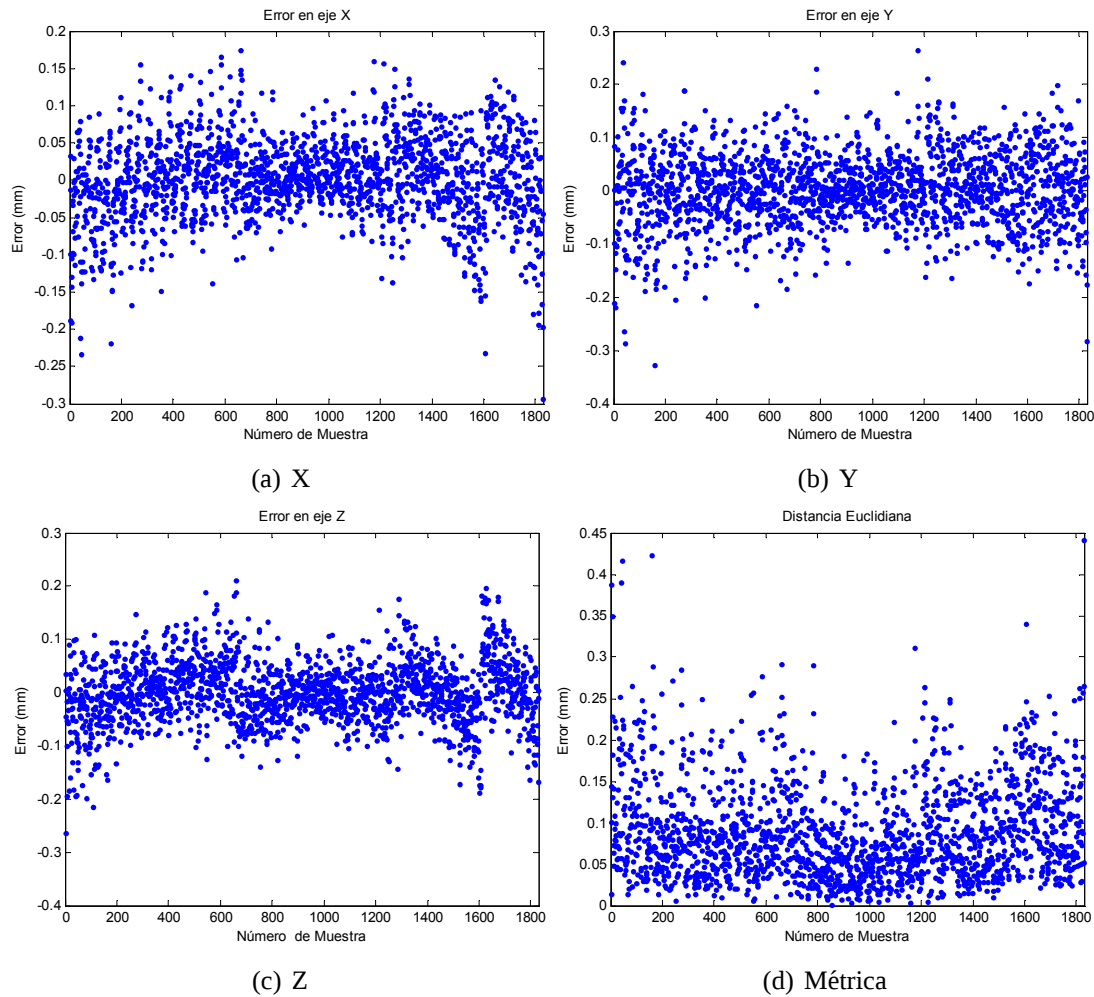


Figura 3.19: Errores de posicionamiento del robot utilizando datos simulados. Se emplearon 1828 datos a lo que se les inyectó error aleatorio con una media de cero y una desviación estándar de 1 pixel.

3.3.2 Simulación de errores con una distribución uniforme

Descripción de la simulación

Finalmente para completar este análisis se repitió la misma simulación descrita anteriormente, en esta ocasión variando la configuración del error aleatorio utilizado. Para este caso se generó un error con una distribución uniforme. En la figura 3.14 se grafica el error aleatorio inyectado a los datos simulados con valores entre -1 y 1 pixel, para el caso de 1828 datos. En la figura 3.22 se muestran los resultados obtenidos de esta simulación de Montecarlo, se puede observar que el comportamiento del error es similar al mostrado en la figura 3.18.

3.3 Sensibilidad de posicionamiento del robot a errores presentes en las mediciones

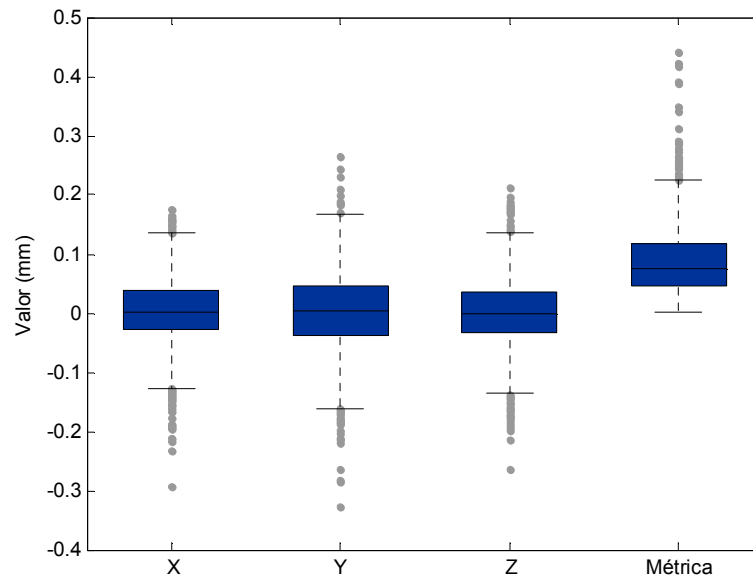


Figura 3.20: Diagrama de caja del error de posicionamiento del robot utilizando ruido inyectado con una distribución normal.

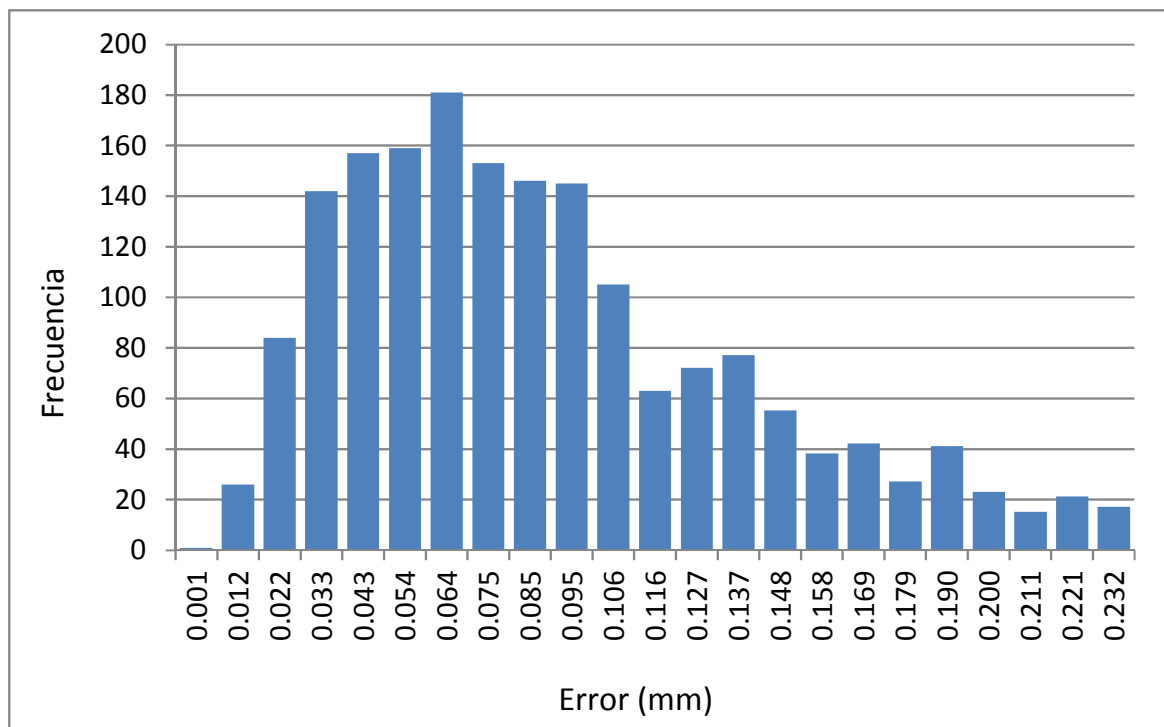


Figura 3.21: Histograma del error de posicionamiento del robot utilizando ruido inyectado con una distribución normal.

3. ANÁLISIS DE SENSIBILIDAD

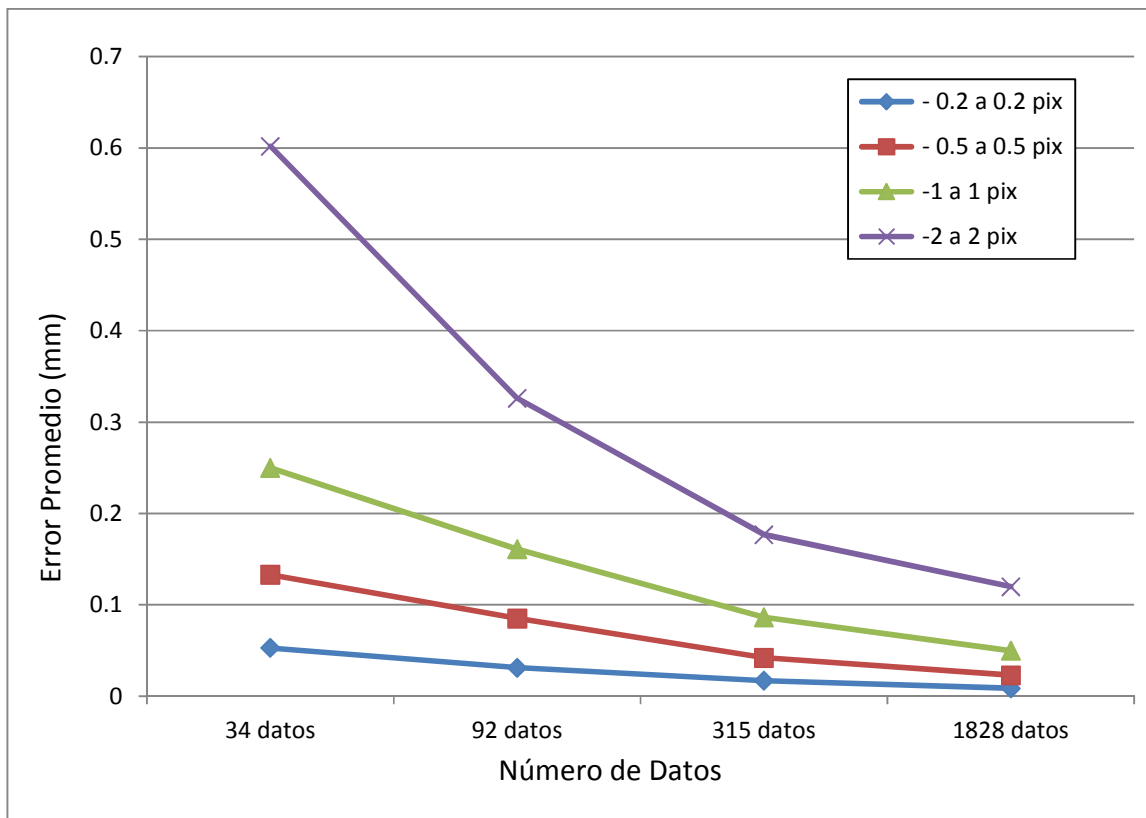


Figura 3.22: Errores promedio de posicionamiento del robot utilizando datos simulados a los que se les inyectó error aleatorio con una distribución uniforme con valores comprendidos entre -0.2 y 0.2 ; -0.5 y 0.5 ; -1 y 1 ; -2 y 2 píxeles.

En la figura 3.23 se grafica la distribución de los errores en cada uno de los ejes coordenados para el caso de 1828 datos, a los que se les agregó ruido aleatorio con una distribución uniforme con valores entre -1 y 1 pixel. De igual manera se muestra la métrica del error obtenida para cada uno de los puntos calculados. Se puede apreciar en la gráfica que el error de posicionamiento es menor a 0.3 mm en todos los casos y el comportamiento es similar al observado cuando se inyectó ruido con una distribución gaussiana. En la figura 3.24 se muestra el diagrama de caja de los errores calculados. Se puede apreciar que, al igual que en la distribución normal, en los ejes X, Y, Z el promedio de error es prácticamente cero. Lo cual indica que la distribución de los errores presentes en las mediciones son afectan en el error de posicionamiento del robot si se tienen los suficientes datos para le cálculo del modelo. En la figura 3.25, se muestra el histograma obtenido para esta misma configuración.

3.4 Variación del número de muestras y su efecto en el error de posicionamiento del manipulador

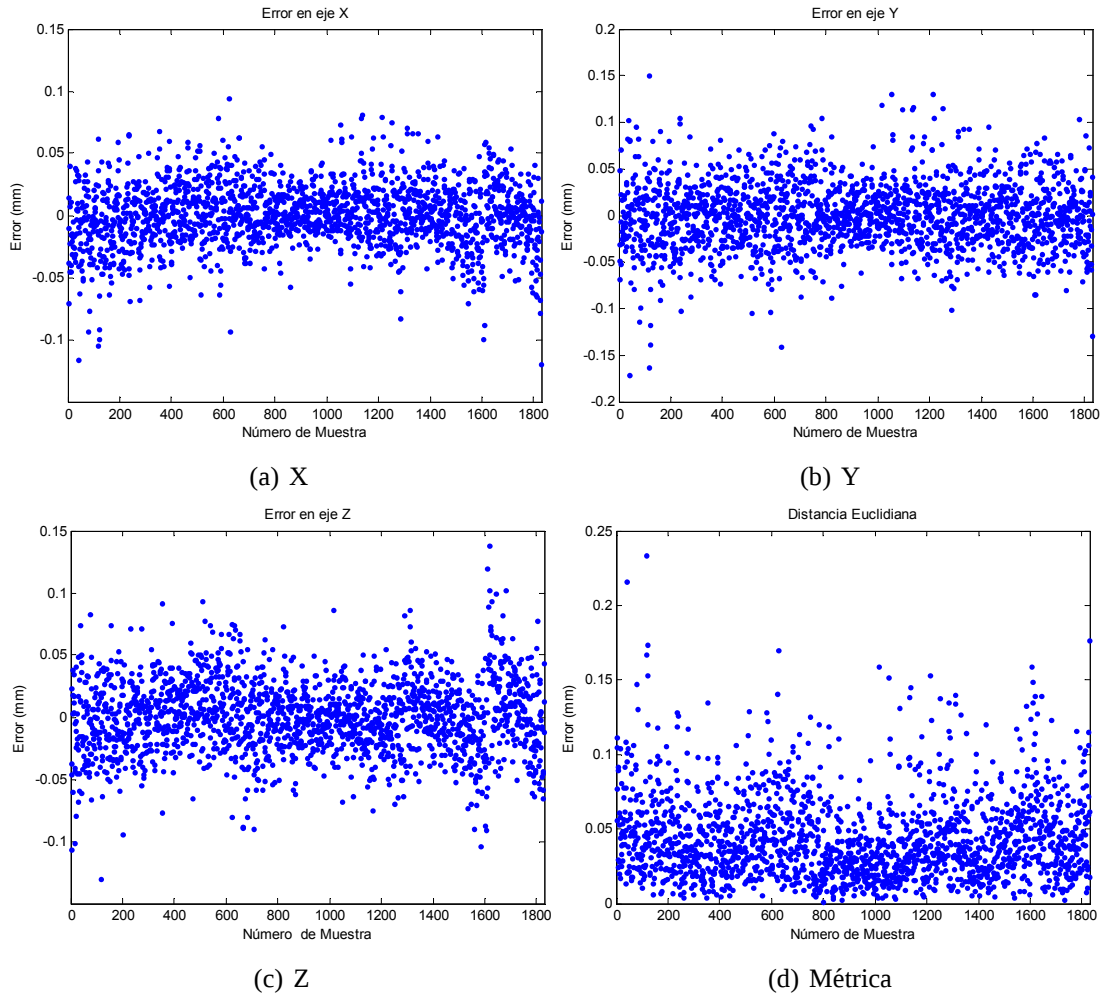


Figura 3.23: Errores de posicionamiento del robot utilizando datos simulados. Se emplearon 1828 datos a lo que se les inyectó error con una distribución uniforme de -1 a 1 pixel.

3.4 Variación del número de muestras y su efecto en el error de posicionamiento del manipulador

En [Chen, 2007] se reportaron resultados de un análisis del modelo CSM original. Esos resultados fueron utilizados como punto de comparación para validar la metodología propuesta en este trabajo.

Para este análisis se inyectó a los datos un error aleatorio con una distribución uniforme y una magnitud entre -0.5 y 0.5 píxeles tal como aparece en [Chen, 2007]. Posteriormente se realizó una simulación de Montecarlo. En cada iteración se seleccionó un punto objetivo arbitrario, se hizo variar el número de muestras cercanas a dicho punto objetivo

3. ANÁLISIS DE SENSIBILIDAD

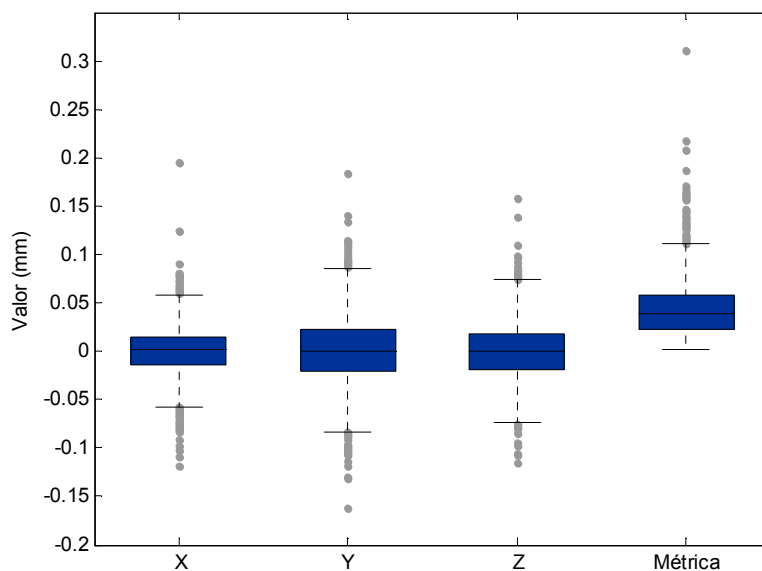


Figura 3.24: Diagrama de caja del error de posicionamiento del robot utilizando ruido inyectado con una distribución uniforme.

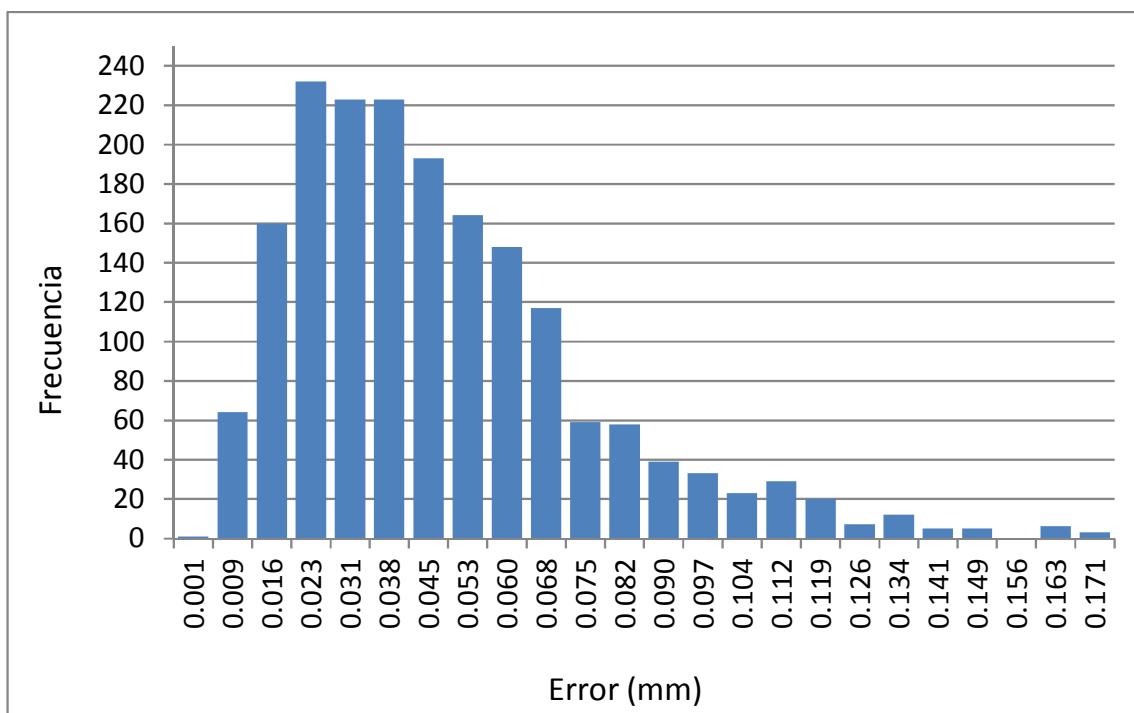


Figura 3.25: Histograma del error de posicionamiento del robot utilizando ruido inyectado con una distribución uniforme.

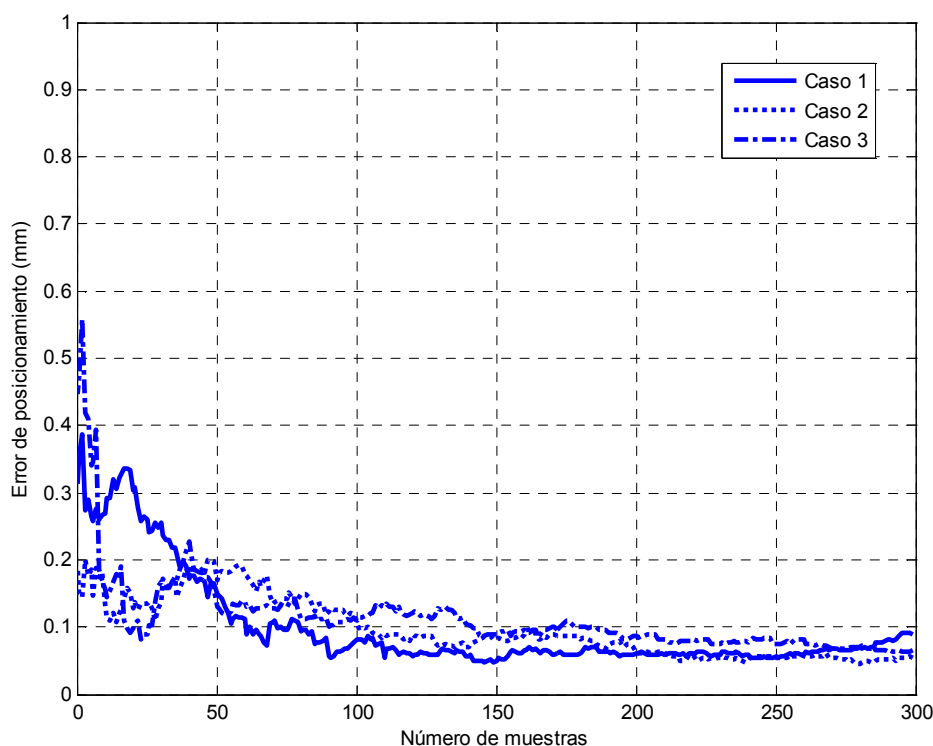


Figura 3.26: Análisis del error del modelo lineal a la variación en el número de muestras cercanas al punto objetivo para el caso de 315 datos.

y se calcularon las matrices de proyección de las cámaras así como la estimación de las coordenadas angulares del robot para posicionarse sobre ese punto.

Tal como se puede observar en la figura 3.26 conforme se utilizan una mayor cantidad de datos, el error de posicionamiento del manipulador decrece. Un comportamiento similar sucede en el trabajo anteriormente citado. La estimación inicial a partir de la trayectoria pre-planeada proporciona una estimación con un error de posicionamiento menor a 0.6 mm.

En la figura 3.26 se presentan tres simulaciones obtenidas del análisis de Montecarlo descrito anteriormente, esto con la finalidad de mostrar que en cada corrida, debido a los datos aleatorios, los resultados son diferentes; sin embargo conforme el número de datos incrementa, los tres casos convergen al mismo valor aproximadamente. Tal como se puede observar en la figura, el error de posicionamiento del robot decrece de 0.55 mm a menos de 0.1 mm para el caso de 315 muestras. En la figura 3.27 se muestra la misma simulación utilizando 1828 datos para el cálculo del modelo, y se puede apreciar el mismo efecto en la reducción del error de posicionamiento. En la figura 3.26 se puede apreciar que utilizando únicamente la información de la trayectoria pre-planeada inicial

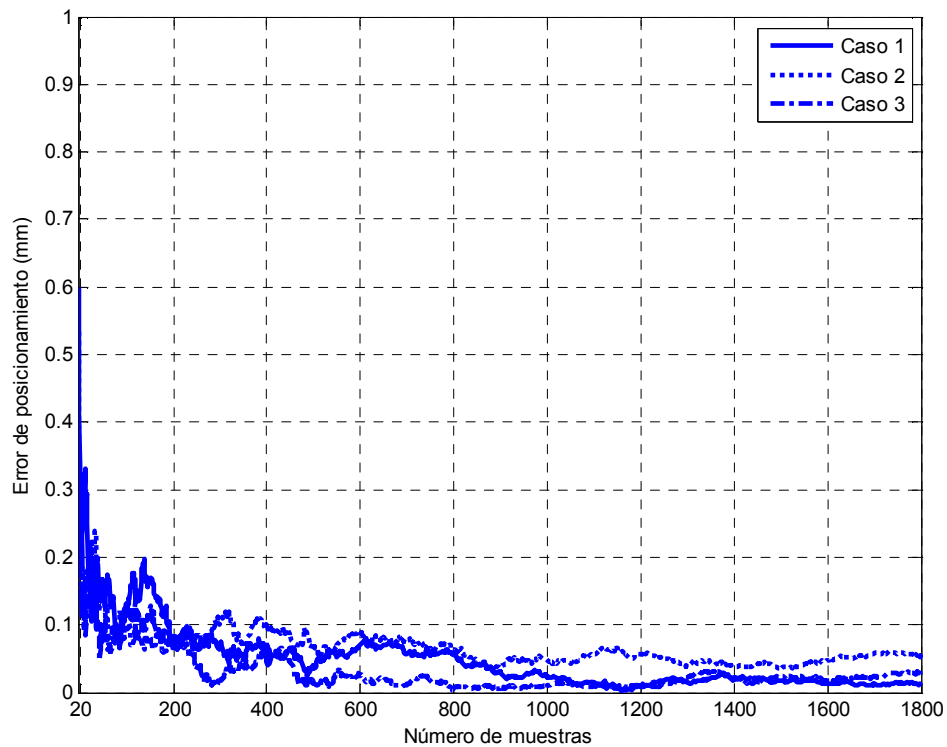


Figura 3.27: Análisis del error del modelo lineal a la variación en el número de muestras cercanas al punto objetivo para el caso de 1828 datos.

con 34 muestras esparcidas en toda el área de imagen, se obtiene una buena estimación del punto objetivo (del orden de 0.5 mm), al agregar el efecto de 50 muestras cercanas tomadas durante la maniobra de posicionamiento, el error decrece a 0.2 mm, y a partir de 200 datos el error se estabiliza por debajo de 0.1 mm.

3.5 Efecto de la resolución en la detección de los datos en espacio de cámara

De acuerdo los algoritmos de detección de marcas visuales utilizados en el método CSM, como ya se mencionó anteriormente, es posible obtener una resolución en la detección de marcas visuales con una precisión menor a 1 píxel. En este contexto, se propuso realizar una simulación para medir tal efecto sobre el error de posicionamiento del robot. Con esta finalidad, se repitió el análisis descrito anteriormente, pero en este caso, se modificó la magnitud del error inyectado, para esta prueba se añadió a los datos un error con una magnitud de entre -0.25 y 0.25 píxeles para simular una mayor resolución en la detección

3.5 Efecto de la resolución en la detección de los datos en espacio de cámara

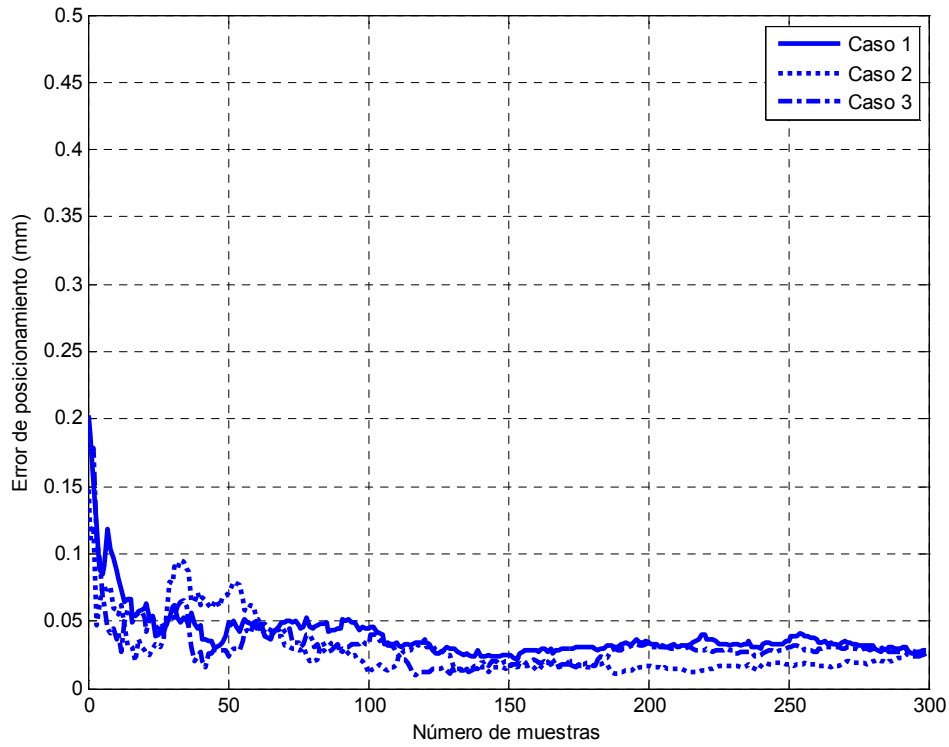


Figura 3.28: Análisis de sensibilidad al número de muestras aplicando un error aleatorio con una magnitud de -0.25 a 0.25 píxeles para el caso de 300 datos.

de las marcas visuales en espacio de cámara. En las figuras 3.28 y 3.29 se muestran los resultados obtenidos de este análisis para el caso de 315 y 1828 datos.

Tal como se muestra en las figuras 3.28 y 3.29, al reducir el error en la detección de las marcas visuales es posible disminuir el error de posicionamiento del manipulador, una manera de lograr una mayor precisión en la detección de las marcas visuales es aumentando la resolución de las cámaras. Aunque como se ha mencionado anteriormente, al utilizar mayor cantidad de información, los errores aleatorios presentes en las mediciones en espacio de cámara se compensan y el error de posicionamiento disminuye. Es decir, existe un compromiso en el número de datos utilizados para calcular el modelo, y la precisión en la detección de los mismos.

En la práctica, la plataforma de visión real requiere menor cantidad de datos para realizar una maniobra, esto es debido a que conforme se realiza la maniobra, se van adquiriendo nuevas muestras de la posición del manipulador conforme el robot se acerca al punto objetivo. Estas muestras son obtenidas mediante las marcas visuales colocadas en el efector final del mismo. Para analizar el efecto de las muestras cercanas se realizó una simulación del error de posicionamiento del robot variando el número de muestras tomadas

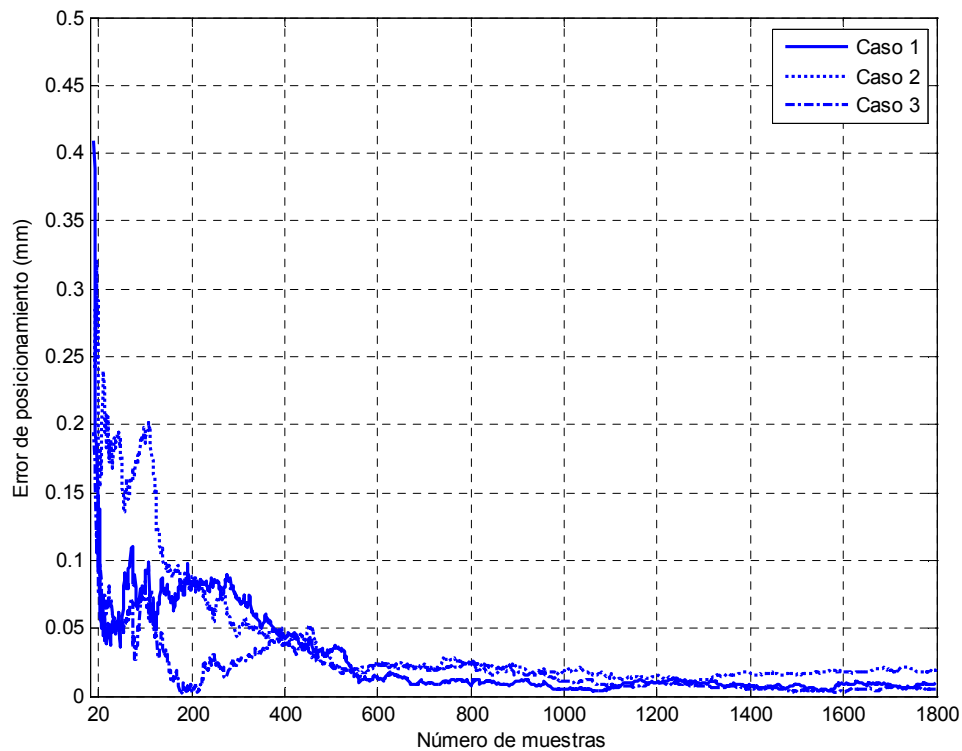


Figura 3.29: Análisis de sensibilidad al número de muestras aplicando un error aleatorio con una magnitud de -0.25 a 0.25 píxeles para el caso de 1800 datos.

durante la maniobra del manipulador. Tal como se describe a continuación.

3.6 Efecto de las muestras tomadas durante la maniobra del manipulador

Como ya se ha mencionado anteriormente, una de las características de la metodología CSM es que conforme se realiza la maniobra de posicionamiento del manipulador, se va actualizando de manera continua la posición del robot en espacio de cámara. Esa actualización constante permite obtener una mejor estimación del punto objetivo al que se desea llevar el robot.

Para el caso de esta simulación se agregó a los datos el efecto de las muestras cercanas tomadas durante la maniobra de posicionamiento del robot. Como primer paso se tomaron 34 posiciones del robot definidas previamente, esparcidas en todo el plano de imagen y tratando de abarcar la mayor cantidad de espacio de cámara posible. Posteriormente se generó un punto objetivo arbitrario y se generó una trayectoria simulada del

3.6 Efecto de las muestras tomadas durante la maniobra del manipulador

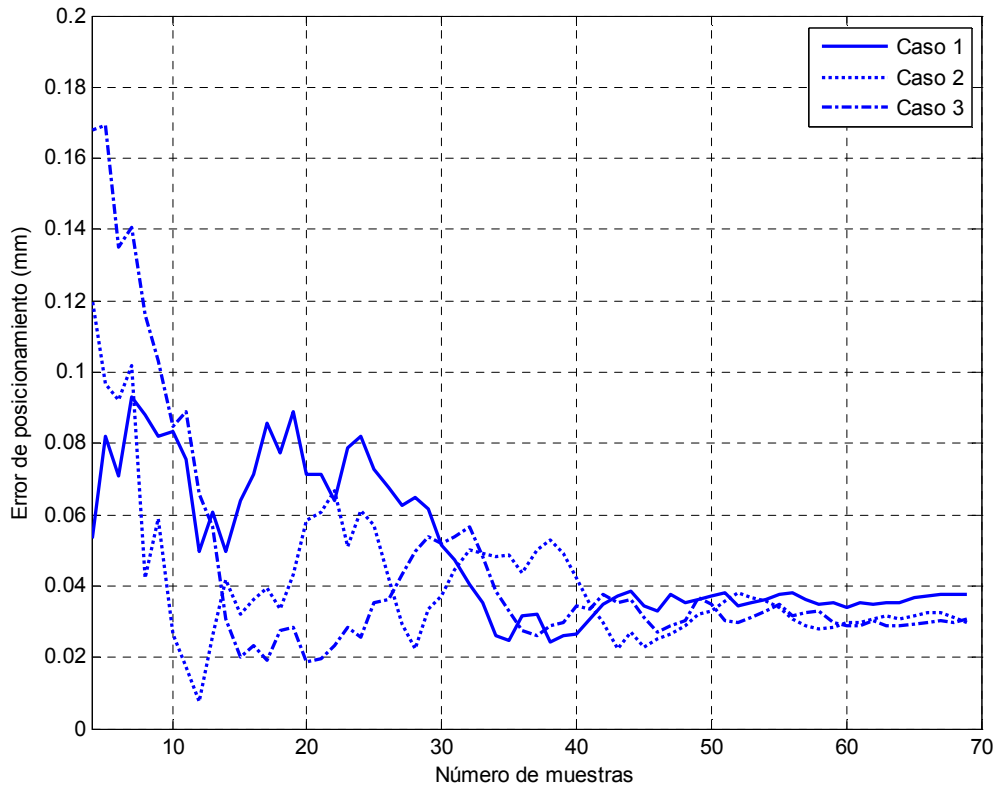


Figura 3.30: Efecto de las muestras tomadas durante la maniobra de posicionamiento.

manipulador para posicionarse sobre dicho punto objetivo. En la trayectoria simulada se fueron generando muestras actualizadas de la posición del manipulador. Posteriormente se inyectó a estos datos un error aleatorio con una distribución uniforme de valores comprendidos entre -0.5 y 0.5 píxeles. Finalmente se realizó una simulación de Montecarlo y los resultados obtenidos aparecen en la figura 3.30.

3.6.1 Mayor resolución en las detecciones en espacio de cámara

La simulación anterior fue realizada nuevamente, en esta ocasión reduciendo la magnitud de los errores añadidos a las muestras. En este caso se añadió un error con una distribución uniforme de valores comprendidos entre -0.25 y 0.25 píxeles simulando una mayor resolución en las detecciones en espacio de cámara. En la figura 3.31 se muestran los resultados de la simulación. De acuerdo a dicha figura, al disminuir el error en la detección de la información en espacio de cámara, el error de posicionamiento del robot disminuye y requiere de un menor número de muestras para estabilizarse en un valor.

3. ANÁLISIS DE SENSIBILIDAD

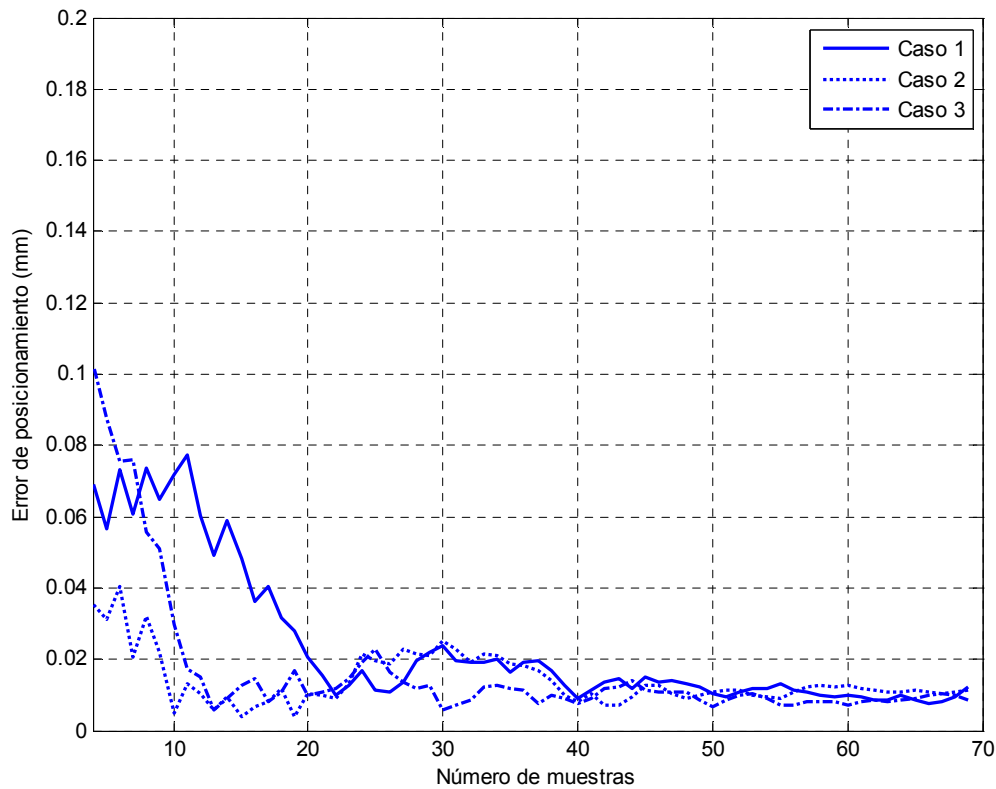


Figura 3.31: Efecto de una mayor resolución en la detección de las muestras tomadas durante la maniobra de posicionamiento.

En este capítulo se presentó un análisis de sensibilidad a partir de datos simulados del modelo LCM-CSM. Se analizó el comportamiento del modelo bajo diferentes esquemas de errores presentes en la mediciones, utilizando mayor o menor cantidad de información, así como el efecto de las muestras tomadas durante la maniobra. En el siguiente capítulo se describen las diferentes plataformas de visión utilizadas para la validación de la metodología propuesta en este trabajo, así como los resultados experimentales obtenidos.

Resultados Experimentales

En este capítulo se presentan los resultados experimentales obtenidos a través de una serie de plataformas de pruebas que fueron implementadas y que serán descritas en las siguientes secciones. Además se presenta una validación del análisis de sensibilidad mostrado en el capítulo anterior, utilizando información real tomada de una de las plataformas mencionadas. También se presenta una descripción general de la interfaz gráfica intuitiva diseñada para la manipulación de objetos por medio de visión computacional, que fue desarrollada para validar los resultados obtenidos durante el desarrollo del presente trabajo de tesis. Además se describe de manera general un algoritmo para la detección de objetos que fue implementado con la finalidad de realizar la selección de puntos objetivo en espacio de cámara de una manera automática.

4.1 Plataformas de pruebas

Para la validación de la investigación realizada en el presente trabajo, se utilizaron tres plataformas de pruebas que fueron acondicionadas para dicho propósito. La primera plataforma es basada en un robot PUMA 761, la segunda utiliza un robot Motoman HP3 y la tercera un robot paralelo tipo delta. A continuación se describe de manera general los componentes de cada una de las plataformas.



Figura 4.1: Robot PUMA 761 con su computadora de control

4.1.1 Plataforma 1 basada en un robot PUMA 761

La plataforma de control por visión en la que se realizaron las pruebas iniciales de validación está conformada por un robot PUMA 761 y equipo de visión convencional (fig. 4.1). El sistema de visión está conformado por una computadora de control Pentium IV corriendo un sistema operativo Linux, que es la encargada de procesar toda la información obtenida a través de las cámaras usando un frame grabber PCI Data Translation y de enviar las acciones necesarias al manipulador por medio de una tarjeta controladora Galil modelo DMC-1860. Esta plataforma consta además de dos cámaras Sony SSCM383 utilizadas como cámaras de control y una cámara uEye usada para la selección del punto objetivo (fig. 4.2). Todos los algoritmos fueron programados en lenguaje C++ bajo sistema operativo Linux distribución Fedora.

4.1.2 Plataforma 2 basada en un robot Motoman HP3

Con la finalidad de validar experimentalmente los resultados obtenidos en el presente trabajo de tesis se implementó una nueva plataforma de control basado en visión. Dicha plataforma está conformada por un robot Motoman HP3 de 6 grados de libertad, un par de cámaras uEye con una resolución de 1280×1024 píxeles, una unidad de dos grados de libertad bajo control computacional, donde se encuentra colocado el láser utilizado para la selección del punto objetivo y una computadora con un procesador CoreDuo corriendo el sistema operativo Windows XP. Los algoritmos fueron programados en lenguaje c, utilizando el compilador de Visual Studio 2008 y se utilizaron las librerías gráficas de OpenCV. En las figuras 4.3 y 4.4 se muestra la plataforma descrita anteriormente.

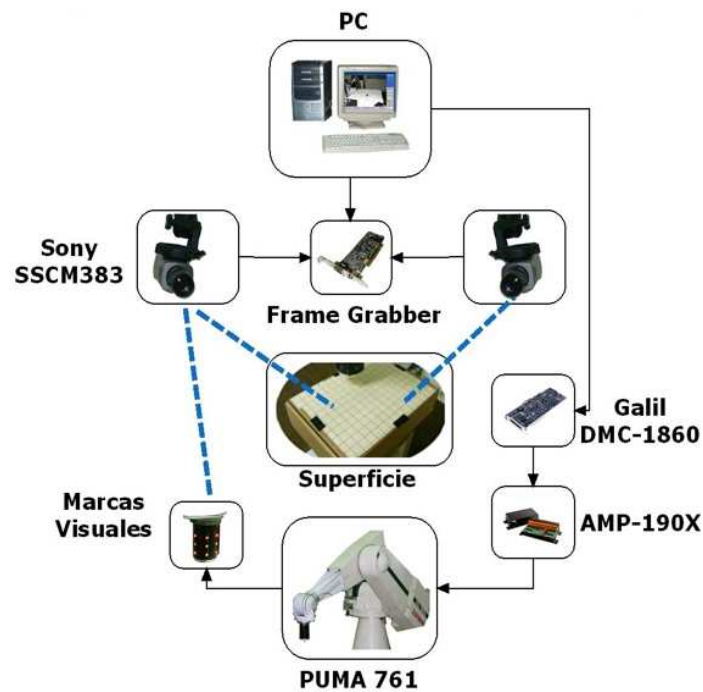


Figura 4.2: Arquitectura del hardware de la plataforma de control basado en visión para el robot PUMA 761.



Figura 4.3: Plataforma de pruebas experimentales utilizada para validar los resultados obtenidos. (1) Robot Motoman HP3, (2) Unidad pan-tilt donde está montado el láser, (3) Cámaras uEye montadas en tripies.

4. RESULTADOS EXPERIMENTALES

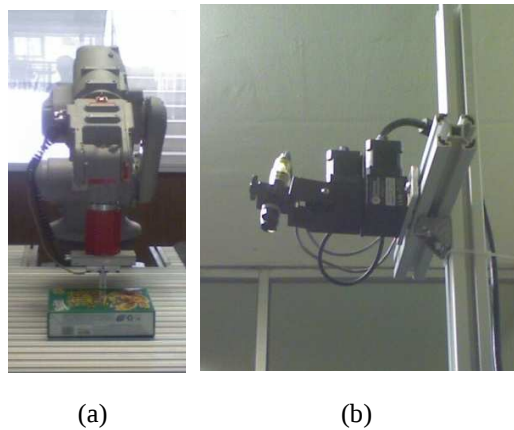


Figura 4.4: Detalle de los componentes de la plataforma de visión. (a) Robot Motoman HP3. (b) Unidad pan-tilt.

Con relación a la plataforma utilizada inicialmente basada en el robot PUMA 761, la nueva plataforma presenta una serie de ventajas que se describen a continuación:

- La nueva plataforma utiliza un robot manipulador de última generación, como lo es el Motoman HP3, que permite obtener mayor precisión y velocidad en las tareas de posicionamiento.
- Se añadió al manipulador un nuevo aditamento para colocar las marcas visuales. Este aditamento permite una mejor detección de los elementos visuales en las imágenes que el utilizado en la primera plataforma. Dicho aditamento permitió además caracterizar de una manera más precisa la ubicación tridimensional de las marcas visuales colocadas en el efector final del manipulador.
- Para esta plataforma de visión se utilizaron cámaras de mayor resolución y se optimizó el algoritmo de detección de centroides de las marcas visuales en la imagen, esto para reducir el error en la detección de las marcas visuales.

4.1.3 Plataforma 3 basada en un robot paralelo tipo delta

En este caso se implementó un sistema de control basado en visión en un robot paralelo tipo delta de 3 grados de libertad. El sistema se compone de dos cámaras CCD montadas en tripiés, un robot delta parallax LFK-2040, y un controlador basado en una PC(4.5). La PC ejecuta los algoritmos de control basado en visión, así como los algoritmos de control del robot. La morfología original del robot paralelo se ha mantenido. Únicamente

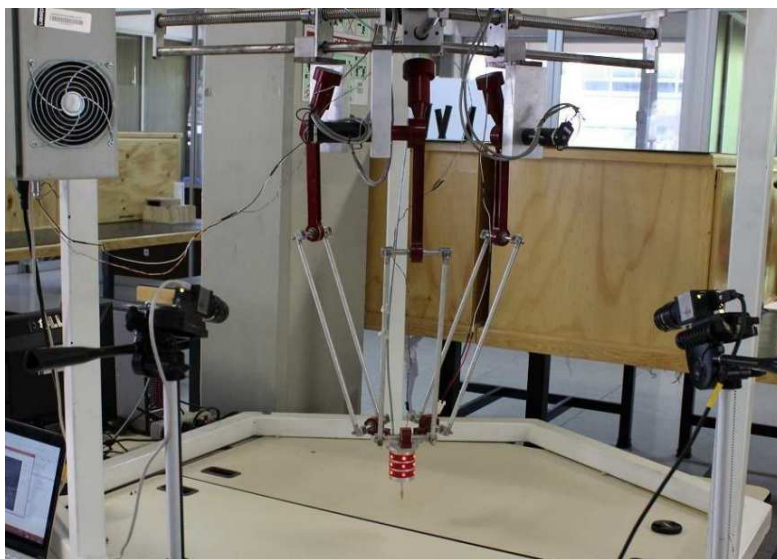


Figura 4.5: Plataforma de pruebas basada en un robot paralelo tipo delta Parallax LKF-2040

se agregó al extremo del efector final una estructura cilíndrica que contiene una matriz de leds de 6×3 . Para el control implementado en la PC se utiliza una tarjeta controladora de movimiento PIC-SERVO SC. Las cámaras CCD utilizadas son de la marca uEye con una resolución de 640×480 y sensor CCD 1/3" modelo Ui-1640-C con unos lentes de 12 mm modelo 1639 VCS. El sistema operativo utilizado es Windows 7.

4.2 Resultados experimentales en la plataforma 1

En esta plataforma se realizó un análisis de sensibilidad tanto de la metodología OF-CSM como LCM-CSM utilizando información real tomada directamente del robot y del sistema de visión de la plataforma, además de una comparación en el error de posicionamiento final del robot en ambos métodos.

En la metodología de manipulación en espacio de cámara existen tres principales fuentes de error que afectan en el posicionamiento del robot, estas fuentes son: errores en la estimación del modelo de cámara, errores en el modelo cinemático del robot, y errores debidos a la resolución finita de las cámaras [Chen, 2007]. Para el análisis de sensibilidad presentado en los siguientes párrafos se tomó en cuenta únicamente el error en la estimación del modelo de cámara, ya que la principal aportación de este trabajo de tesis es precisamente la sustitución del modelo de cámara originalmente utilizado en CSM. Para el caso de los errores en el modelo cinemático se ha demostrado que a través de la infor-

4. RESULTADOS EXPERIMENTALES

mación visual es posible compensar dichos errores tal como se presenta en [Chen, 2007]. Además de los errores mencionados anteriormente, existe un error debido a la selección del punto objetivo en pantalla por parte del operador, pero para este análisis se asume que es cero.

Para el análisis de sensibilidad, se planteó realizar un estudio de los factores principales que influyen en la estimación del modelo de cámara que se propuso en este trabajo, tales como errores en la detección de la información visual y el número de datos utilizados para la estimación del modelo. También se analizaron otros factores para realizar una comparación del modelo de cámara tradicionalmente utilizado en CSM con el modelo propuesto en este trabajo de tesis, tales como la utilización de sensores que proyectan píxeles cuadrados y rectangulares.

4.2.1 Errores en el modelo de cámara

Con la finalidad de analizar los efectos de los errores en las detecciones de las imágenes se propuso realizar un análisis de sensibilidad del sistema. Este análisis consistió en obtener una serie de datos de la plataforma de visión y utilizarlos para calcular el modelo de cámara tanto para el modelo Ortográfico + *Flattening* (OF-CSM) como para el modelo lineal propuesto en este trabajo (LCM-CSM). Esto con la finalidad de realizar una comparativa de los resultados obtenidos por ambos modelos. Para este caso se utilizaron 100 datos para calcular ambos modelos de cámara. Cada dato contiene las coordenadas (X,Y,Z) de una marca visual colocada en el efector final del robot y sus coordenadas (x,y) proyectadas sobre la imagen en píxeles. Este análisis constó de dos etapas principales: La primera consistió en obtener el modelo de cámara empleando $n - 1$ datos, y a partir del modelo calculado estimar el punto n , denominado punto objetivo. Este proceso se realizó de manera iterativa para cada uno de los 100 datos.

En la segunda etapa se realizó una simulación de Montecarlo empleando el mismo proceso que la primera etapa pero agregando error aleatorio a los datos de la imagen con una media de cero y una desviación estándar de 1 pixel, esto con la finalidad de ver la sensibilidad a los errores presentes en los datos. Ambas etapas se realizaron para el Modelo LCM-CSM y para el Modelo OF-CSM, los resultados obtenidos se muestran en el cuadro 4.1

De acuerdo a los resultados presentados en la tabla 4.1 se puede concluir que el modelo LCM-CSM realiza una mejor reconstrucción de la información visual en esta plataforma,

Error de estimación obtenido				
Datos	$\sigma_e = 0$		$\sigma_e = 1$	
Modelo	LCM-CSM	OF-CSM	LCM-CSM	OF-CSM
Error(pix)	0.1416	1.237	0.5778	1.5089
Desviación	0.0817	1.4014	0.29	1.6713

Cuadro 4.1: Errores en el modelo de cámara

ya que el error de estimación del modelo es menor. En cuanto a la sensibilidad a los errores en los datos, los dos modelos presentan prácticamente los mismos resultados. Cabe mencionar que las cámaras utilizadas en la plataforma 1, con la que se realizó el análisis anterior, contienen sensores que proyectan píxeles rectangulares con una relación de aspecto de 0.8. A continuación se presenta un análisis del efecto de los píxeles rectangulares en ambos modelos.

4.2.2 Píxeles cuadrados vs rectangulares

Estas pruebas fueron realizadas en dos etapas, la primera fue utilizando una cámara sony modelo SSCM383 y la segunda etapa usando una cámara uEye de la marca IDS modelo UI-2210-M, ambas con una resolución de 640×480 píxeles. Esto con la finalidad de medir los errores del modelo cuando los píxeles son cuadrados y cuando son rectangulares. Las cámaras Sony generan píxeles rectangulares con una relación de aspecto de 0.8, y las cámaras uEye proyectan píxeles cuadrados.

Descripción de las pruebas

Para cada conjunto de datos se calcularon ambos modelos de cámara usando $n - 1$ muestras, posteriormente se calculó la estimación del punto n a partir de cada modelo. El error se definió como la distancia entre el punto medido y el punto estimado, para el caso de píxeles cuadrados y posteriormente en píxeles rectangulares. Tal como se describe a continuación:

Prueba 1. Se calcularon ambos modelos usando 100 conjuntos de 9 muestras en cada caso con píxeles rectangulares, a manera de simular el número máximo de marcas que puede visualizar la cámara a partir del aditamento colocado en el efector final del robot. Ambos modelos fueron calculados usando muestras locales, y muestras distribuidas en toda la imagen. Los valores promedio del error calculado ϵ y la desviación estándar σ obtenidos en cada experimento se muestran en el cuadro 4.2.

4. RESULTADOS EXPERIMENTALES

	Locales	No Locales
OF-CSM	$\epsilon = 0.89$ $\sigma = 0.51$	$\epsilon = 27.87$ $\sigma = 26.28$
LCM-CSM	$\epsilon = 7.70$ $\sigma = 11.99$	$\epsilon = 0.88$ $\sigma = 0.48$

Cuadro 4.2: Muestras locales versus no locales en píxeles rectangulares

	Locales	No Locales
OF-CSM	$\epsilon = 0.37$ $\sigma = 0.21$	$\epsilon = 1.70$ $\sigma = 1.17$
LCM-CSM	$\epsilon = 0.58$ $\sigma = 0.43$	$\epsilon = 0.44$ $\sigma = 0.28$

Cuadro 4.3: Muestras locales versus no locales en píxeles cuadrados

Prueba 2. En esta prueba se realizó exactamente la misma metodología que la prueba anterior pero empleando datos obtenidos de las cámaras que proyectan píxeles cuadrados. Los resultados obtenidos se muestran en el cuadro 4.3, los peores casos en cada modelo se muestran resaltados.

De acuerdo a los datos presentes en los cuadros 4.2 y 4.3 se puede observar que el modelo OF-CSM presenta una mejor estimación de los puntos cuando los datos están concentrados en un área local; el peor caso se presenta cuando se utilizan píxeles rectangulares y se emplean muestras no locales. Este error es mayor debido a que el efecto de perspectiva en las muestras no locales es más representativo, aunado a que dicho modelo no tiene en cuenta el efecto de los píxeles rectangulares.

Para el caso del modelo LCM-CSM, se obtiene una buena estimación en muestras no locales, mientras que para muestras que abarcan un área pequeña de la imagen, se tiene un error mayor. Este error es debido a que los datos locales no proporcionan suficiente información para modelar el efecto de perspectiva. En la siguiente sección se presenta una técnica utilizada para reducir la magnitud de dicho error.

	Locales
LCM-CSM	$\epsilon = 0.65$ $\sigma = 0.28$

Cuadro 4.4: Error promedio y desviación estándar utilizando muestras locales normalizadas

4.2.3 Normalización de las muestras

A partir de los datos obtenidos en las pruebas anteriores se concluye que el peor caso para el modelo lineal es cuando se utiliza información local. Este error puede ser reducido si se realiza una normalización de las muestras locales [Hartley and Zisserman, 2004]. Esta normalización de los datos se puede obtener a partir de la expresión 4.1.

$$\tilde{X}_i = UX_i \quad (4.1)$$

Donde $\tilde{X}_i$ representa el vector de coordenadas 3D normalizadas del punto X_i y U representa la matriz de transformación que traslada el origen del sistema coordenado tridimensional a un punto cercano a las muestras locales. En este caso la matriz de transformación está definida por el modelo cinemático nominal del robot. Con las muestras normalizadas se calcula la matriz de cámara $\tilde{P}$ acorde a la expresión 2.16. Finalmente se aplica una denormalización de la matriz de cámara aplicando la expresión 4.2.

$$P = \tilde{P}U \quad (4.2)$$

Aplicando la normalización descrita anteriormente a los datos, se repitió la *prueba 1*, definida en la sección anterior, para reconstruir los datos mostrados en el cuadro 4.2. Los resultados obtenidos se muestran en el cuadro 4.4.

De acuerdo a la tabla 4.4 se puede concluir que el proceso de normalización de los datos reduce considerablemente el error de estimación del modelo de cámara lineal cuando se emplea información concentrada en un área pequeña de la imagen.

Continuando con este análisis, a partir de muestras normalizadas se realizó una simulación del error de estimación del modelo LCM-CSM en función de la distancia entre el origen del sistema tridimensional y el centroide del área que abarcan las muestras locales. Dicha distancia fue modificada a partir de la matriz de transformación U que define el modelo cinemático del robot. El error promedio calculado en función de la distancia del origen con respecto al centroide de las muestras se grafica en la figura 4.6.

4. RESULTADOS EXPERIMENTALES

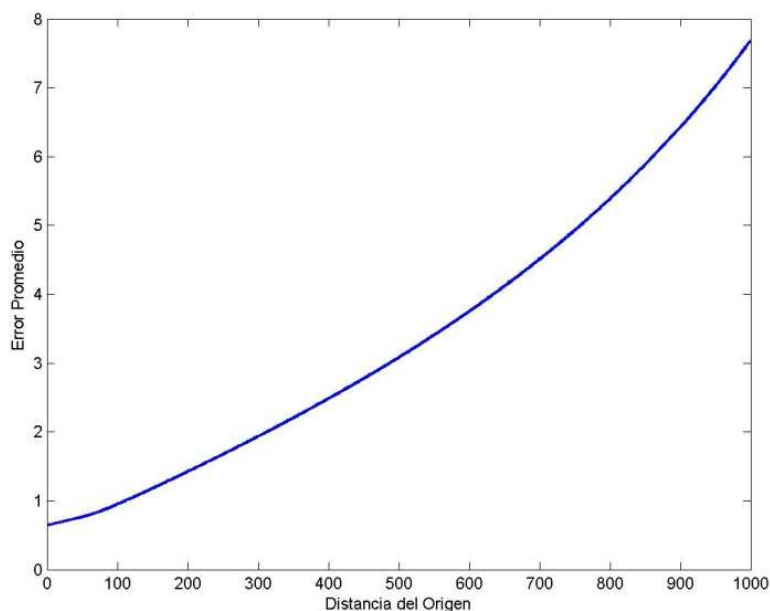


Figura 4.6: Comportamiento del error en función de la distancia entre el origen del sistema coordenado 3D y el centroide del área que abarcan los datos usados para estimar el modelo lineal empleando muestras locales.

En la figura 4.6 se puede apreciar claramente que conforme las muestras locales son alejadas del origen el error de estimación del modelo de cámara crece.

4.2.4 Errores debidos al número de muestras

Para medir el efecto del número de muestras utilizadas para la estimación del modelo de cámara, se realizó un análisis de sensibilidad de los algoritmos de posicionamiento del robot empleando el modelo LCM-CSM, esto con la finalidad de ver cuál es el resultado de utilizar mayor o menor cantidad de muestras del robot durante la maniobra de posicionamiento.

Esta prueba consistió en tres etapas principales: La primera fue la obtención de las muestras del manipulador. La segunda consistió en el cálculo de la configuración angular del robot para posicionarlo sobre un punto definido de manera arbitraria y la tercera está relacionada con la medición del error. Los resultados fueron obtenidos a partir de muestras experimentales tomadas de la plataforma de control basado en visión. A continuación se describen dichas etapas:

1. **Toma de muestras.** Se realizaron 160 desplazamientos del robot, los cuales cubrieron aproximadamente el 80 % del área de imagen. En cada movimiento se almacenaron los siguientes datos:
 - La posición tridimensional de una marca visual colocada en el efector final del robot, obtenida a partir del modelo cinemático.
 - Las coordenadas en píxeles de la marca visual correspondiente detectada en ambas cámaras del sistema de visión.
2. **Cálculo de las juntas del robot.** En esta etapa se realizó el proceso definido en la expresión 2.29 para determinar la configuración angular del robot que es requerida para posicionarse sobre un punto objetivo definido en espacio de cámara.
3. **Medición del error.** El error δ se define como la distancia Euclidiana entre el punto objetivo definido (X_d, Y_d, Z_d) y el punto estimado a través de la información visual (X_e, Y_e, Z_e) . Dicho error se puede obtener mediante la siguiente expresión:

$$\delta = \sqrt{(X_d - X_e)^2 + (Y_d - Y_e)^2 + (Z_d - Z_e)^2}$$

Posteriormente se realizó un proceso iterativo, en cada iteración se redujo en 1 el número de muestras utilizadas para calcular el modelo de cada una de las cámaras, y posteriormente se calcularon los valores de las juntas del robot para posicionarse sobre un punto objetivo definido de manera arbitraria. Los resultados aparecen graficados en la figura 4.7.

De acuerdo a la figura 4.7 se puede concluir que el error de posicionamiento se reduce conforme el número de muestras utilizadas aumenta, esto es debido a que al utilizar un mayor número de muestras los errores aleatorios presentes en las mediciones se compensan. Si se incrementa el número de muestras el error tiende a cero.

4.2.5 Cálculo de matrices de proyección de cámara (mínimos cuadrados vs DVS)

En la sección 2.2 se abordaron dos metodologías que son utilizadas para el cálculo de las matrices de proyección de cámara definida en la ecuación 2.7. En los siguientes párrafos se presenta una comparativa entre dichas metodologías.

4. RESULTADOS EXPERIMENTALES

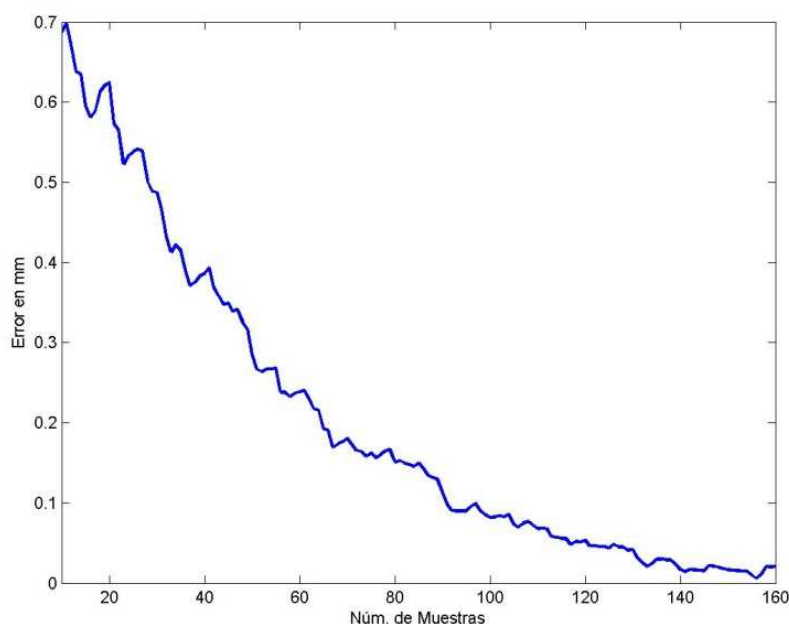


Figura 4.7: Error de posicionamiento del robot vs número de muestras.

Para realizar una comparación entre los dos métodos, se realizó un análisis empleando 200 datos experimentales obtenidos de la plataforma 1 de visión, dichos datos se utilizaron para realizar el cálculo del modelo de cámara empleando las metodologías como se describe a continuación.

Tal como se mencionó anteriormente, la primer metodología para el cálculo de las matrices de proyección de cámara es denominada mínimos cuadrados, la cual permite obtener la matriz de cámara mediante la solución de un sistema de ecuaciones lineales. La segunda metodología, tal como se describió en la sección 2.2, es basada en una descomposición en valores singulares (DVS).

Se realizó una simulación de Montecarlo para cada una de las metodologías. En cada iteración se inyectó ruido a los datos con una distribución normal, una media de cero y una desviación estándar variable de 0.5 a 2 píxeles. En la figura 4.8 se grafica el error promedio obtenido para cada uno de los métodos para 100 iteraciones en cada caso.

Tal como se puede apreciar en la figura 4.8, el método basado en DVS presenta menos sensibilidad a errores en las mediciones. Cuando el ruido es menor (del orden de 0.5 píxeles), ambos métodos prácticamente presentan los mismos resultados, pero la diferencia resulta significativa conforme los valores incrementan. Esta mayor sensibilidad al ruido del método de mínimos cuadrados se atribuye a que al momento de calcular la pseudo inversa (ec. 2.17), si la cantidad de datos es elevada, se obtiene una matriz con valores

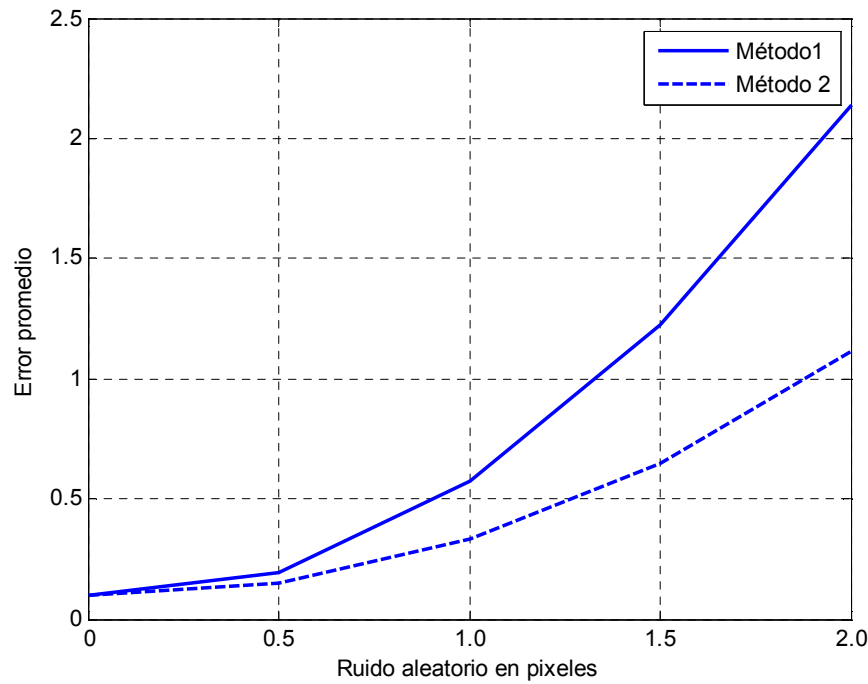


Figura 4.8: Valores promedio de error de estimación del modelo de cámara. Método 1: Mínimos cuadrados. Método 2: Descomposición en valores singulares.

muy pequeños. Esto puede provocar pérdida de información al momento de reconstruir el modelo.

4.2.6 Costo computacional de los modelos

Una de las ventajas de utilizar un modelo lineal en la metodología CSM, es que reduce el tiempo requerido para el cálculo de los parámetros de visión. A continuación se presenta un análisis del tiempo requerido para la obtención del modelo de cámara empleando tres estrategias de cálculo.

En el cuadro 4.5 se muestra una comparativa de los tiempos requeridos para calcular el modelo de cámara utilizando las dos estrategias presentadas en este trabajo y comparando los resultados obtenidos con la estrategia no lineal reportada en [García, 2007]. Para cada modelo se utilizaron las cuatro configuraciones de datos descritas en la sección 3.2. La metodología del modelo no lineal se describe brevemente a continuación:

Esta metodología emplea el modelo de cámara de orificio, para sustituir el modelo ortográfico y el proceso de flattening utilizados en la metodología CSM original. Los parámetros contenidos en el modelo de cámara de orificio son calculados mediante un

4. RESULTADOS EXPERIMENTALES

No. de Muestras	Tiempo (segundos)		
	Mínimos cuadrados	DVS	No lineal
34	0.0004	0.003 (7.5 veces)	0.263 (657.5 veces)
92	0.0005	0.012 (24 veces)	0.531 (1062 veces)
315	0.0014	0.139 (99.3 veces)	1.647 (1176.4 veces)
1828	0.0048	3.981 (829.4 veces)	10.329 (2151.9 veces)

Cuadro 4.5: Comparativa de tiempos requeridos para calcular el modelo de cámara usando tres diferentes metodologías

proceso iterativo de estimación no lineal. El proceso de estimación no lineal se realiza a través de un proceso denominado corrección diferencial de mínimos cuadrados, que calcula un factor de corrección de los parámetros del modelo de cámara en cada iteración del algoritmo, el proceso finaliza cuando se logra la convergencia en los valores estimados. Las condiciones iniciales para el cálculo de este proceso no lineal son obtenidas mediante la solución de un sistema de ecuaciones lineales.

Tal como se puede observar en el cuadro 4.5, el método de mínimos cuadrados requiere menor tiempo de cálculo en el modelo de cámara. Como es de esperarse, el tiempo es más representativo conforme el número de datos utilizados para calcular el modelo se incrementa.

4.2.7 Error de posicionamiento del Robot

Finalmente para realizar una comparativa entre el modelo OF-CSM y LCM-CSM, se realizaron una serie de maniobras de posicionamiento del robot, empleando ambos modelos. En cada maniobra se hizo llegar el robot a un punto objetivo predefinido y se midió la distancia entre el efector final del robot y el punto previamente seleccionado. Los resultados obtenidos para un total de 100 maniobras de posicionamiento se muestran en las figuras 4.9 y 4.10. Para el caso del modelo OF-CSM el error promedio obtenido fue de 0.97 mm con una desviación estándar de 0.43 mm. El modelo LCM-CSM obtuvo un error promedio de 0.88 mm con una desviación estándar de 0.38 mm. Lo cual demuestra que con ambos modelos se obtienen prácticamente los mismos resultados en el posicionamiento del robot.

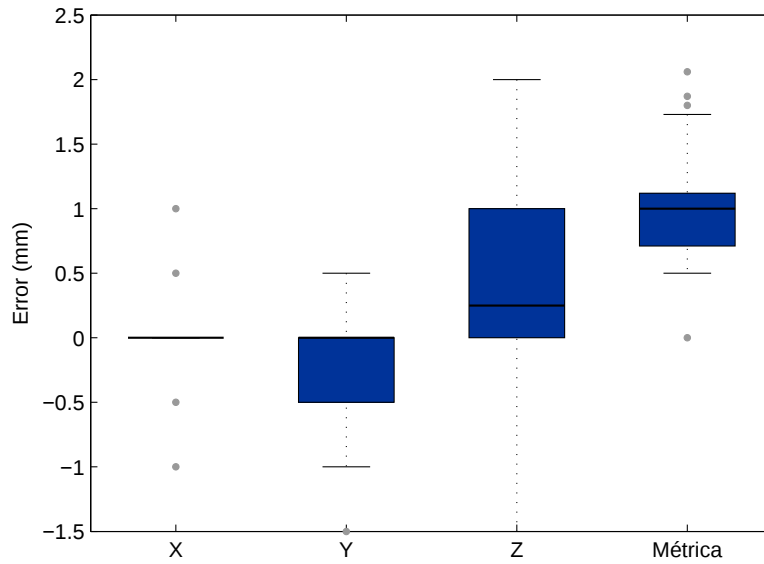


Figura 4.9: Error de posicionamiento del robot para 100 maniobras utilizando el método OF-CSM

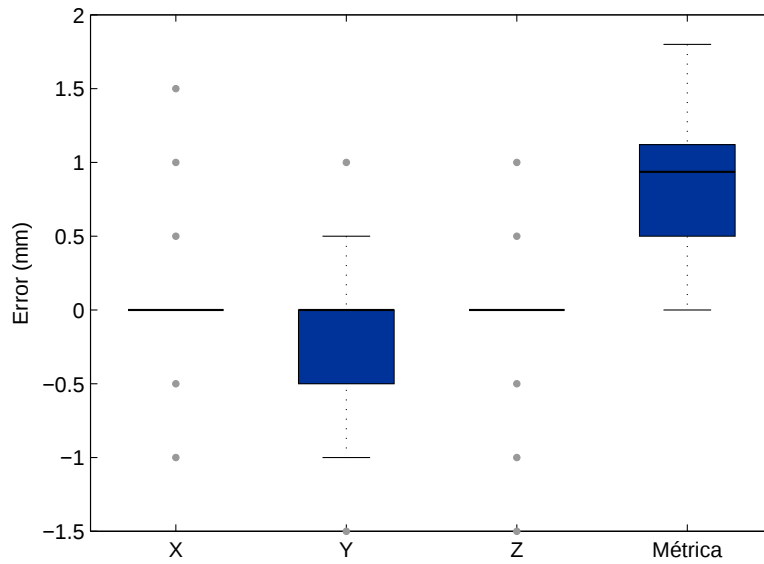


Figura 4.10: Error de posicionamiento del robot para 100 maniobras utilizando el método LCM-CSM

4.3 Resultados experimentales en la plataforma 2

Para esta plataforma se desarrolló una interfaz gráfica que permite la selección de puntos objetivo de una manera intuitiva. Para la selección de los puntos objetivos se utilizó una técnica denominada *Point and Click*, que consiste en que el operador selecciona de manera gráfica con el ratón de la computadora el punto objetivo donde se requiere que se posicione el manipulador. También se implementó un algoritmo de detección automática de objetos para facilitar la selección de objetivos en la imagen, tal como se describe a continuación.

4.3.1 Interfaz gráfica intuitiva para la manipulación de objetos

Con la finalidad de facilitar la selección de puntos objetivo para el sistema de control basado en visión se diseñó una interfaz gráfica de usuario (GUI por sus siglas en inglés). Dicha interfaz facilita al operador de la maniobra definir con facilidad el punto donde se desea que se posicione el robot. Esto se realiza simplemente dando un click con el ratón en el punto designado y posteriormente mediante un botón situado en la GUI se indica el inicio de la maniobra de posicionamiento. En la figura 4.11 se muestra la interfaz gráfica para la selección manual de puntos objetivo en espacio de cámara. La parte de selección gráfica de la GUI consta de dos ventanas, la ventana de mayor tamaño es la que permite visualizar toda la superficie de trabajo del robot y permite además la selección del objetivo por parte del operador mediante un click del ratón; y otra ventana más pequeña que permite realizar una ampliación del área de donde se está desplazando el puntero del ratón. Esta ampliación o *zoom*, permite al operador posicionarse de una manera más precisa en el centro del objeto que se desea manipular. Esta manera de seleccionar el punto es muy sencilla pero implica un error que está relacionado directamente con la habilidad del operador para seleccionar el punto correctamente, tal como se describirá más adelante.

La GUI se divide en tres partes principales: La parte de caracterización de la superficie de trabajo, la selección del punto objetivo y el control de la maniobra del manipulador.

Tal como se mencionó anteriormente, una de las problemáticas más comunes que se presentan a la hora de la selección de puntos objetivo en espacio de cámara, al utilizar la técnica *point and click*, es que depende en gran medida de la habilidad del operador para seleccionar el punto de manera precisa. Existen algunas soluciones que se han propuesto para reducir el error en la selección del punto objetivo [Chen, 2007]. Una de ellas consiste en realizar una ampliación del área de imagen cercana al punto objetivo (fig. 4.12).

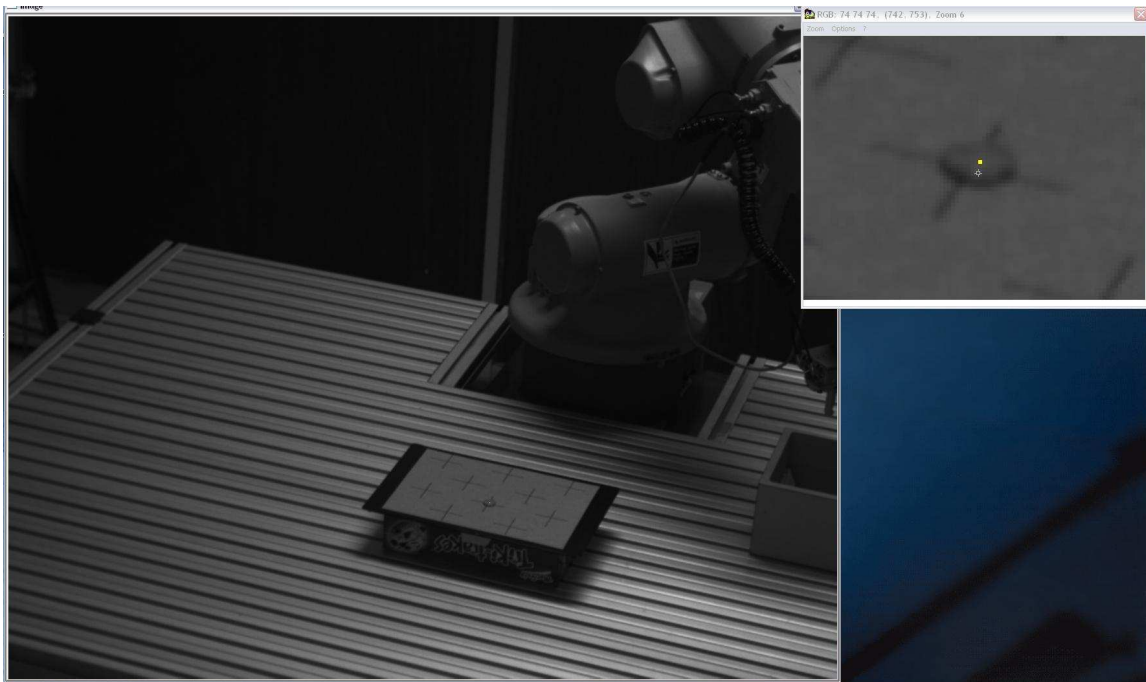


Figura 4.11: Interfaz Gráfica Intuitiva para la selección manual de puntos objetivo

Otra solución que se ha propuesto es la selección múltiple de puntos sobre una línea y posteriormente realizar un ajuste de una recta sobre dichos puntos (fig. 4.13), esto con la finalidad de promediar los errores presentes en cada punto y así reducir el error de apreciación por parte del operador.

Para el presente trabajo de tesis se propuso definir una tarea del manipulador con el fin de validar los algoritmos de control basado en visión implementados. Dicha tarea consistió en la manipulación de un objeto delgado que pueda mostrar la precisión de los algoritmos implementados. Para este caso se optó por utilizar una arandela de 1 mm de

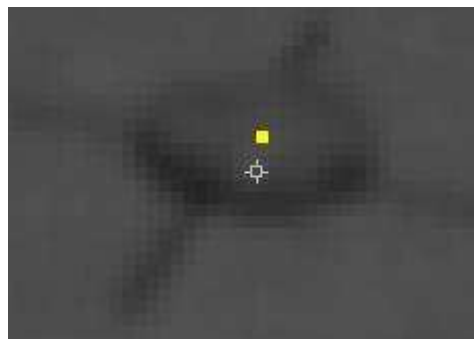


Figura 4.12: Ampliación del área de imagen para la selección del punto objetivo.

4. RESULTADOS EXPERIMENTALES

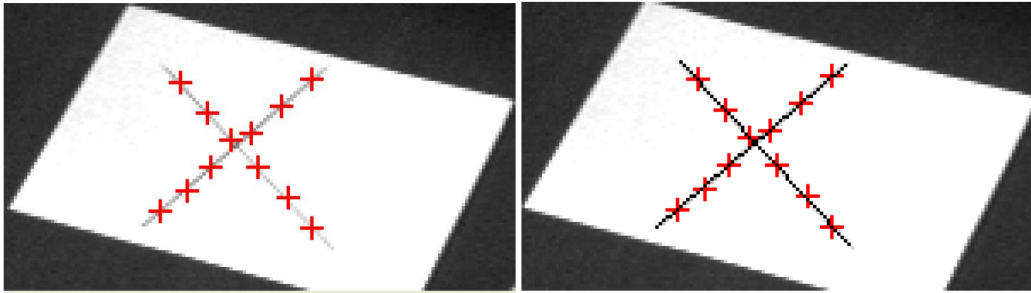


Figura 4.13: Selección múltiple de puntos objetivo. a) Selección múltiple. b) Ajuste de líneas

espesor. Tal como se muestra en la figura 4.14.

Luego de seleccionar la tarea de posicionamiento se propuso utilizar un algoritmo de detección automática de objetos, para realizar la selección de puntos objetivo de manera autónoma, y así reducir los errores al momento de la selección manual. Dicho algoritmo se describe a continuación.



Figura 4.14: Arandela de 1 mm de espesor, utilizada para la tarea de posicionamiento del robot.

4.3.2 Detección automática de arandelas por medio de visión computacional

Tal como se mencionó anteriormente, la maniobra para validar los algoritmos de control basado en visión consiste en que el robot manipulador pueda tomar de manera automática una serie de arandelas colocadas sobre una superficie con inclinación arbitraria. Para realizar la tarea antes mencionada se implementó un algoritmo para la detección automática

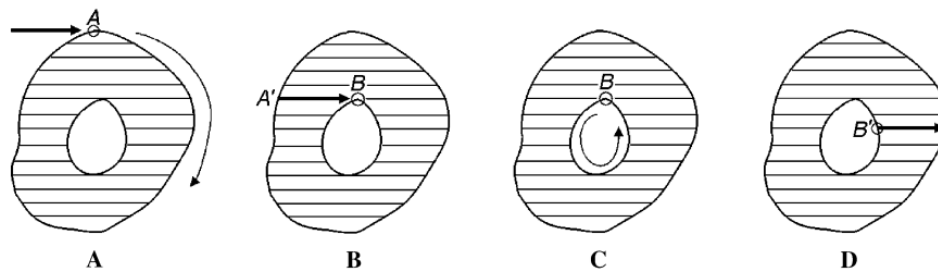


Figura 4.15: Partes principales que definen el algoritmo de detección de objetos basado en rastreo de contornos.

de manchas u objetos presentes en la imagen, el cual está basado en técnicas de rastreo de contorno, tal como se describe a continuación.

4.3.3 Descripción general del algoritmo de detección de objetos

El algoritmo implementado se denomina etiquetado de objetos utilizando una técnica de rastreo de contorno [Chang et al., 2004]. Consiste básicamente en realizar un escaneo de los píxeles sobre una imagen binarizada buscando píxeles que correspondan al contorno de un objeto. Este escaneo se realiza de arriba hacia abajo y de izquierda a derecha en toda la imagen. De manera general, este algoritmo se puede resumir en 4 puntos principales, tal como aparecen en la figura 4.15.

1. Cuando se encuentra un punto que corresponde a un contorno externo por primera vez (Punto A de la figura 4.15-A), se realiza un trazado del contorno del objeto hasta retornar nuevamente al punto A. Posteriormente se le asigna una etiqueta al punto A y a todos los demás puntos que correspondan a ese contorno.
2. Cuando es encontrado un punto A' que corresponde a una etiqueta ya asignada (Fig 4.15-B), se realiza un escaneo sobre la misma línea para encontrar los puntos que corresponden al mismo objeto y se les asigna la etiqueta A'.
3. Cuando se encuentra un punto de contorno interno la primera vez (Fig. 4.15-C), se asigna al punto B la misma etiqueta tiene el contorno externo del mismo objeto. Posteriormente se realiza un trazado del contorno interno que contiene el punto B y se le asigna la misma etiqueta del punto B a todos los demás puntos que forman dicho contorno.

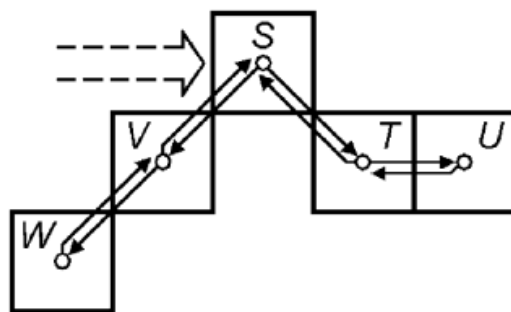


Figura 4.16: Búsqueda del contorno de un objeto en la imagen

4. Cuando en el escaneo se encuentra un punto de contorno interno ya etiquetado B' (Fig. 4.15-D), se sigue la línea de escaneo para localizar los puntos subsecuentes que corresponden al mismo objeto, y se les asigna la misma etiqueta que al punto B'.

4.3.4 Trazado del contorno

La finalidad del trazado del contorno es localizar un contorno interno o externo sobre un punto dado S. El primer paso para encontrar un contorno es determinar si el punto en cuestión es un punto aislado, en tal caso se finaliza la búsqueda del contorno. En caso contrario se asigna al punto que sigue al punto S con la etiqueta T, tal como se muestra en la figura 4.16, posteriormente se continua con la búsqueda del contorno hasta que se cumplan las dos condiciones siguientes: (a) el final del contorno es el punto S y (b) el punto que sigue en el contorno es el punto T. Tal como aparece en la figura 4.16, cuando el punto S es el punto de inicio del contorno y T es el punto que le sigue, el contorno trazado es "STUTSVWVS".

4.3.5 Dirección de búsqueda del contorno

Para un punto P que corresponde a un contorno, el objetivo de la búsqueda es verificar cuál de los 8 puntos cercanos al punto P corresponden al contorno formado por P. Para esto se le asigna un índice a cada uno de los 8 puntos colindantes, tal como aparece en la figura 4.17-A. La búsqueda se realiza en sentido horario desde un punto inicial, el cual es determinado de la siguiente manera.

Si el punto P en cuestión es el punto inicial de un contorno externo, la posición inicial se fija como 7, es decir, la esquina superior derecha. Si el punto corresponde a un contorno

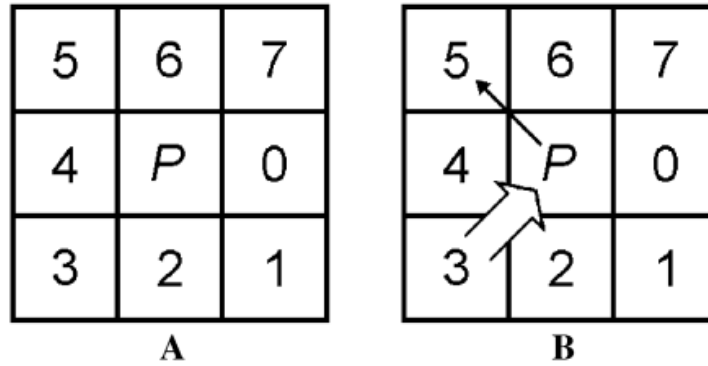


Figura 4.17: Índices asignados a los valores colindantes de un punto que compone un contorno.

interno, el punto inicial es fijado en 3, es decir, la esquina inferior izquierda, dado que el punto debajo de *P* es un pixel blanco y el punto 3 corresponde al inicio de la trayectoria en sentido horario. Por otro lado, si el punto previo del contorno existe, y se ubica en la posición 3, entonces el punto inicial se fija en la posición 5, debido a que la posición 4 ya debió haber sido analizada anteriormente por el algoritmo (Fig. 4.17-B). En general cuando el punto *P* no representa el punto inicial de un contorno, su posición inicial de búsqueda se define como $d + 2$, donde d es la posición de punto previo en el contorno. Una vez que el punto inicial se ha determinado, se realiza una búsqueda en dirección de las manecillas del reloj para localizar el siguiente pixel negro, si no existe este pixel dentro del círculo de búsqueda, el punto *P* se identifica como un punto aislado. Finalmente, los puntos asignados a la misma etiqueta corresponden a un objeto detectado en la imagen. El algoritmo descrito anteriormente, fue implementado en la plataforma de control basado en visión que se ha venido desarrollando en el trabajo de tesis, en la siguiente sección se muestran los resultados obtenidos en la detección de las arandelas.

En la figura 4.18 se muestran los resultados de la aplicación del algoritmo de detección de objetos, se puede observar en la figura, cinco arandelas colocadas sobre la superficie de trabajo del manipulador. En la imagen de la derecha se pueden apreciar resaltados en color rojo los objetos detectados por el algoritmo.

Para validar los alcances del algoritmo de detección de objetos, se realizó una prueba utilizando elementos con distintas geometrías, los resultados obtenidos se muestran en la figura 4.19. Tal como se puede apreciar en la figura, la detección de los objetos se realiza de manera adecuada. Una de las características de este algoritmo es que permite clasificar los objetos de acuerdo a su tamaño, a través de la información del número de

4. RESULTADOS EXPERIMENTALES

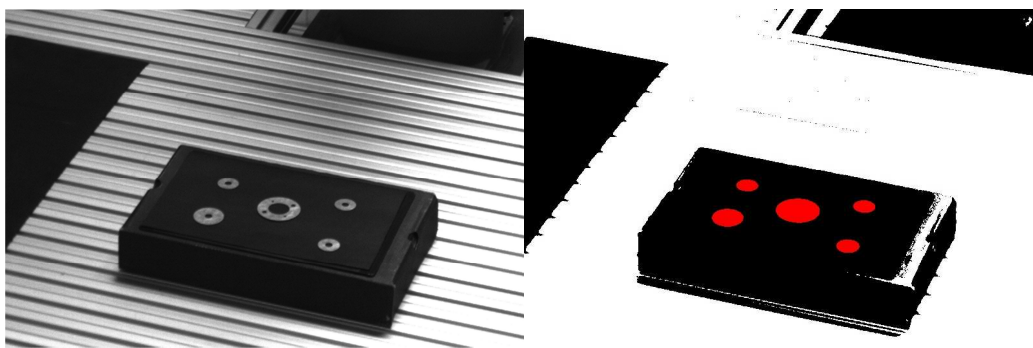


Figura 4.18: Detección de arandelas de diferentes diámetros colocadas sobre la superficie de trabajo del robot.

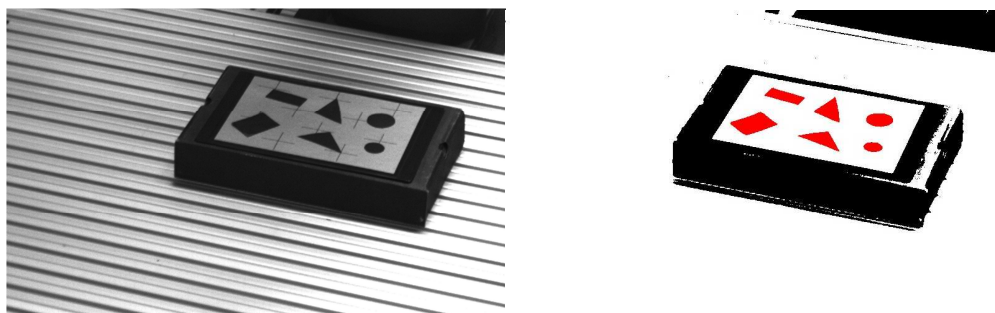


Figura 4.19: Detección de objetos con geometrías distintas.

píxeles etiquetados en cada figura. Esto permite restringir la detección de los objetos en la imagen de acuerdo al tamaño del objeto que se desea manipular.

4.3.6 Resultados

Para validar los resultados obtenidos, se realizó la implementación de los algoritmos del método LCM-CSM en la plataforma 2. Posteriormente se propuso una tarea preliminar de posicionamiento donde se pudieran validar dichos algoritmos. La tarea propuesta, tal como ya se mencionó antes, consistió en manipular una arandela de 1 mm de espesor que aparece indicada en la figura 4.14. Por medio de la interfaz de usuario desarrollada, el operador selecciona con un click sobre el centro de la arandela e indica en la interfaz mediante un botón, el inicio de la maniobra. Para esta tarea, se utilizaron 34 muestras iniciales esparcidas en toda el área de imagen, durante la maniobra de posicionamiento se tomaron muestras cercanas al punto objetivo a una distancia de 100, 50, 20 y 10 mm para actualizar las matrices de cámara y la información de la caracterización de la superficie de trabajo.

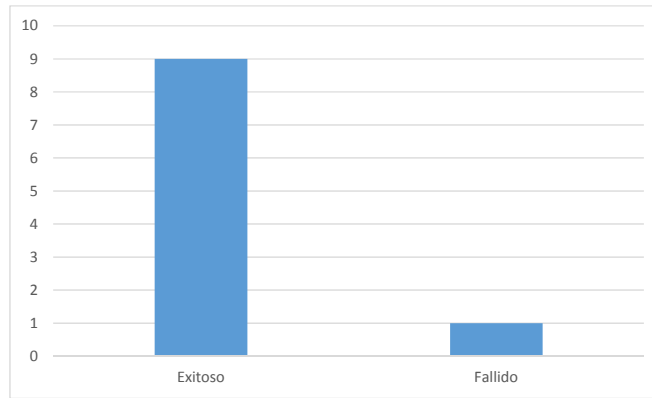


Figura 4.20: Resultado de 10 maniobras para tomar una arandela colocada de manera arbitraria.

En la figura 4.21 se muestran 4 imágenes de una tarea de posicionamiento del robot. Como se mencionó anteriormente, el robot se detiene para actualizar la información visual en cuatro ocasiones conforme se va acercando al punto objetivo. El tiempo que tardó el robot en tomar la arandela y colocarla en un contenedor después de recibir la orden de inicio de maniobra fue de 20 segundos aproximadamente.

Para validar esta plataforma se corrieron una serie de experimentos que consistieron en tomar una arandela colocada sobre una superficie con inclinación arbitraria. De dichos experimentos se documentaron un total de 10 casos, de los cuales 9 fueron exitosos y 1 fallido, tal como se muestra en la figura 4.20.

De acuerdo a las pruebas realizadas se pudo verificar que el error de posicionamiento en la plataforma 2 de control basado en visión es menor a 1 mm que corresponde con el espesor del objeto que se está manipulando. Sin embargo, como es común en esta metodología, el error de posicionamiento del robot depende en gran medida de la habilidad del operador para seleccionar el centro del objeto. Dicho error es eliminado al utilizar el algoritmo de detección automática de objetos.

4.4 Resultados experimentales en la plataforma 3

Con la finalidad de validar la versatilidad de la metodología LCM-CSM se implementó utilizando la plataforma 3 antes descrita, basada en un robot paralelo tipo delta de 3 grados de libertad. En este caso se programó una tarea punto a punto, donde se definió un punto objetivo que el robot debía alcanzar. Dicha tarea se repitió para un total de 50 experimentos. Los resultados obtenidos se muestran en la figura 4.22. Los experimentos arrojaron

4. RESULTADOS EXPERIMENTALES

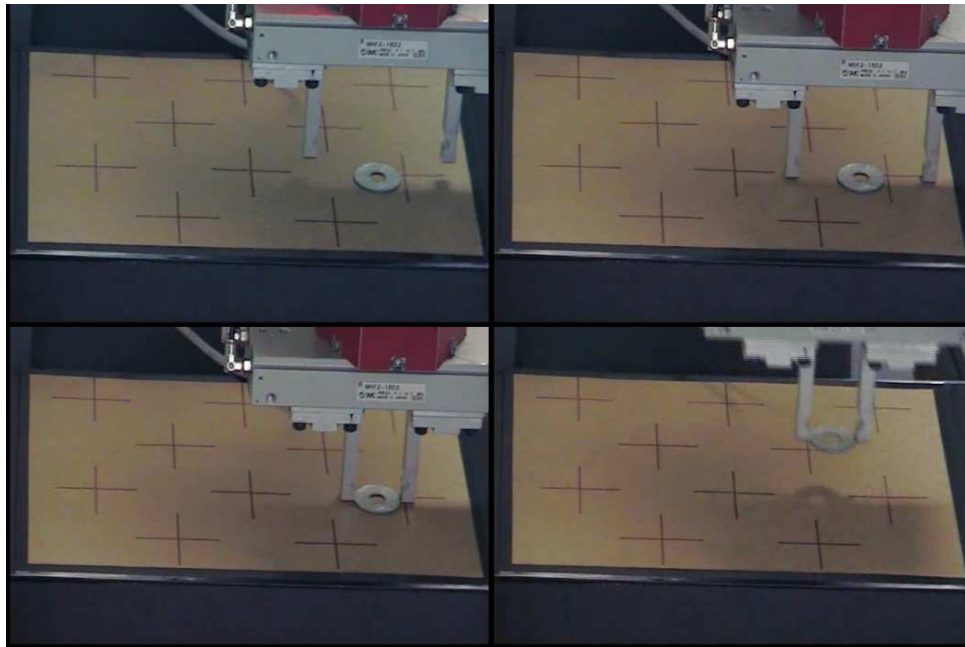


Figura 4.21: Ejemplo de la maniobra de posicionamiento del robot para tomar una arandela de 1mm de espesor, usando la plataforma de control basado en visión.

un error promedio de 1.7 mm con una desviación estándar de 0.6 mm, lo cual es aceptable considerando que se trata de un prototipo didáctico.

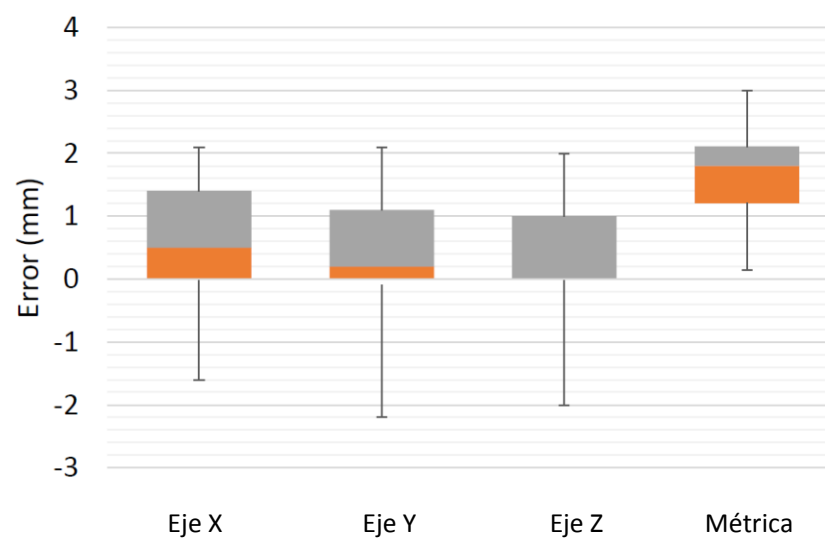


Figura 4.22: Error de posicionamiento del robot para 50 maniobras

En este capítulo se describieron las diferentes plataformas utilizadas para validar el presente trabajo de tesis. Se demostró que la metodología propuesta en este trabajo es viable de ser utilizada para el control de robot manipuladores obteniendo errores de posicionamiento del robot que van del orden de los 2 mm, hasta menos de 1 mm, dependiendo de las características de la plataforma utilizada. En el siguiente capítulo se describen las conclusiones finales del trabajo y el trabajo que queda pendiente para un futuro.

Conclusiones y trabajo futuro

Las conclusiones obtenidas a partir del trabajo de tesis desarrollado se pueden resumir de la siguiente manera:

En este trabajo de tesis se propuso una modificación a la metodología CSM original, la cual consistió en reemplazar el conjunto de parámetros de visión obtenidos a partir de un modelo ortográfico de cámara, por un conjunto de parámetros lineales establecidos a partir del modelo de cámara de orificio.

Uno de los requerimientos básicos para lograr la convergencia a partir del modelo ortográfico, consiste en que la distancia entre el punto focal de la cámara y el origen del sistema coordinado unido a la base del robot, debe ser lo suficientemente grande para despreciar el efecto de perspectiva. Posteriormente se utiliza el proceso de *flattening* como medida de optimización del algoritmo. Sin embargo, este proceso requiere una estimación inicial aproximada del parámetro Z_r que debe ser medido manualmente.

La modificación propuesta en este trabajo presenta diversas ventajas. La primera ventaja consiste en que los parámetros de visión son obtenidos a partir de un modelo lineal, lo que permite obtener un mínimo global en la solución. Al mismo tiempo, la linealidad del modelo hace que sea más sencillo de implementar. La segunda ventaja deriva del hecho de que el método propuesto es basado en el modelo de cámara de orificio, el cual modela directamente el efecto de perspectiva. Lo anterior permite el cálculo de los parámetros de visión en un solo proceso, a diferencia del método original que requiere de dos procesos. Estas dos ventajas mencionadas permiten que el algoritmo LCM-CSM tenga un menor costo computacional.

Se pudo demostrar que existe una relación directa entre los parámetros de los modelos OF-CSM, y LCM-CSM. Sin embargo, la propuesta de este trabajo permite modelar algunos parámetros adicionales tales como distorsión y píxeles rectangulares.

En el análisis de sensibilidad del modelo de cámara lineal, se pudo determinar que el peor caso se presenta al utilizar muestras concentradas en un área muy pequeña del plano

CONCLUSIONES

de imagen, esto debido principalmente a que no se tiene suficiente información respecto al efecto de perspectiva, sin embargo, el error de estimación se reduce considerablemente al realizar un proceso de normalización de las muestras. También se pudo concluir que al utilizar mayor cantidad de información visual, el error en el cálculo del modelo de cámara se reduce, esto debido principalmente a que se compensan los errores aleatorios presentes en las mediciones. Otro de los resultados obtenidos a través de este análisis es que la resolución de las imágenes utilizadas también repercute en el error de posicionamiento final del robot.

Tanto el método OF-LCM como LCM-CSM fueron implementados utilizando un robot industrial de 6 grados de libertad. De acuerdo a los resultados experimentales obtenidos, se puede concluir que ambas metodologías no son estadísticamente diferenciables con respecto al error de posicionamiento.

La metodología propuesta en este trabajo pudo ser implementada con éxito usando diferentes plataformas, utilizando diferentes dispositivos de imagen y empleando robots manipuladores con diferentes configuraciones. Cabe mencionar que el trabajo reportado en este trabajo de tesis ha servido como base para posteriores implementaciones que han sido objeto de estudio.

Se pudo implementar con éxito una técnica de detección automática de objetos para la selección de puntos objetivo, conformando una plataforma de control basado en visión semi-automática, y logrando un error de posicionamiento menor a 1 mm. A través de la implementación, se pudo demostrar que al utilizar un algoritmo de detección de objetos es posible reducir los errores de selección de puntos objetivo y así mejorar la precisión de posicionamiento del manipulador.

Se puede concluir que el algoritmo utilizado para la detección de objetos, en general, presenta buenos resultados, en algunos casos se presentan falsos positivos en la detección de objetos, sin embargo estos falsos positivos se pudieron eliminar restringiendo el área de búsqueda únicamente al contorno que forma la superficie de trabajo donde se colocan los objetos. Esto se realiza de manera automática al momento de caracterizar la superficie de trabajo a partir de la información visual, utilizando los algoritmos de la metodología de control basado en visión que se ha venido desarrollando en este trabajo de tesis.

Actividades realizadas

De acuerdo a los objetivos cubiertos durante el trabajo de investigación desarrollado se pueden mencionar las siguientes actividades realizadas:

1. Se pudo establecer un método que no presenta problemas de convergencia para obtener los parámetros de visión de CSM.
2. Modificación a la metodología CSM original, que permite obtener los parámetros de visión en una sola iteración. Esto implica que el algoritmo sea más fácil de implementar y computacionalmente más rápido.
3. Desarrollo e implementación de un análisis de sensibilidad del modelo propuesto en este trabajo que permitió obtener un esquema adecuado para la implementación experimental del método LCM-CSM.
4. Evaluación de la metodología propuesta en simulación y de manera experimental en diversas plataformas de hardware:
 - (a) **Robot PUMA.** Se realizó la implementación del método LCM-CSM utilizando como plataforma un robot PUMA 761, cuyo principal problema es que presenta juego mecánico en las articulaciones, lo que lo hace poco preciso en su posicionamiento. Se pudo demostrar que el método propuesto en este trabajo puede compensar dicho juego mecánico a través de la retroalimentación visual, obteniendo un error de posicionamiento del efector final menor a 2 mm.
 - (b) **Motoman HP3.** Se realizó también la implementación en un robot de última generación como lo es el HP3 de Motoman. En esta plataforma se pudieron lograr errores de posicionamiento menores a 1 mm.
 - (c) **Robot Paralelo tipo Delta.** Se pudo validar experimentalmente la metodología LCM-CSM en una plataforma de visión utilizando un robot paralelo tipo Delta, logrando un error de posicionamiento del orden de 2 mm, lo cual resulta razonable tratándose de un prototipo académico. Lo anterior demuestra la versatilidad de la metodología propuesta.
5. Desarrollo de una interfaz gráfica intuitiva para la programación de maniobras del robot en tareas de manipulación de objetos. Esta plataforma fue desarrollada en lenguaje C++, bajo el entorno de programación de Visual Studio y sistema operativo Windows XP.
6. Implementación de un algoritmo de detección de objetos para la selección automática de puntos objetivo en espacio de cámara. Este algoritmo facilita la programación

de la maniobra del robot manipulador y permite detectar múltiples objetos al mismo tiempo y de manera automática. Esto permite definir diversas maniobras de posicionamiento del robot con una sola captura. Como limitante, se requiere que los objetos permanezcan fijos durante la maniobra de posicionamiento, lo cual es un requerimiento también de la metodología CSM.

Publicaciones

Los resultados reportados en este trabajo de tesis generaron las siguientes publicaciones:

1. Juan Manuel Rendón, Antonio Cárdenas, **Marco Alonso García Romero**, Emilio J. González-Galván, Bruno Lara “*Robot Positioning using Camera-Space Manipulation with a Linear Camera Model*” IEEE Transactions on Robotics. Vol. 26 No. 4. pp. 726-733. ISSN 1552-3098. (2010)
2. **Marco A. García**, Antonio Cárdenas, Juan M. Rendón, Mauro Maya Méndez “*Una Plataforma de Control Basado en Visión Para la Rehabilitación de Robots Manipuladores de Tipo Industrial*” Revista Computación y Sistemas del IPN, Vo. 12, No. 4. ISSN: 1405-5546. (2009).
3. Enrique Coronado, Mauro Maya, Antonio Cárdenas, **Marco García** “*Control de un Robot Paralelo Tipo Delta basado en Manipulación en Espacio de Cámara Lineal*” Congreso Nacional de Control Automático 2013, Ensenada, Baja California.

Trabajo a futuro

A partir de los resultados obtenidos durante el desarrollo de esta tesis se plantean los siguientes objetivos a desarrollar como trabajo futuro:

- Desarrollo de un software robusto con interfaz intuitiva que pueda ser implementado en cualquier plataforma de control basado en visión.
- Implementación de un sistema de control basado en visión de teleoperación asistida, donde se permita la definición de tareas de posicionamiento del robot a través de internet.
- Desarrollo de una interfaz gráfica para dispositivos móviles y tabletas electrónicas que permita la selección, ejecución de tareas y monitoreo de la maniobra del manipulador.

- Implementación de una arquitectura de control basado en visión de bajo costo y código abierto que permita su implementación que cualquier plataforma de hardware.
- Reducir el costo de implementación utilizando cámaras web de menor costo, ya que la robustez del método permite lidiar con errores de distorsión.
- Implementación de la metodología desarrollada en este trabajo de tesis empleando sistemas embebidos basados en FPGA y DSP.
- Desarrollo de un software de control basado en visión intuitivo que pueda utilizarse en cualquier sistema de cómputo independientemente de su arquitectura o sistema operativo.
- Realizar una comparativa de algoritmos de optimización para el cálculo de los valores articulares del manipulador y analizar otras estrategias como redes neuronales o algoritmos genéticos para mejorar la precisión y la velocidad del algoritmo.
- Diseñar e implementar una estrategia para manipular objetos sobre superficies arbitrarias.
- Ampliar el algoritmo de detección de objetos para manipular elementos con geometrías arbitrarias.
- Diseñar un sistema de clasificación de objetos de acuerdo a su forma o tamaño por medio de visión.



Modelo Cinemático directo del manipulador

A continuación se describe de manera general el modelo cinemático directo y se obtienen las matrices de transformación que definen dicho modelo para el caso del robot Motoman HP3, que corresponde a la plataforma 2.

El modelo cinemático del robot representa la relación que existe entre los valores de las variables asociadas a las articulaciones del robot y la posición y orientación de un sistema de referencia unido al robot. Este sistema de referencia se define acorde a la tarea que se pretende realizar. En un robot manipulador, es común que este sistema de referencia se asocie al efector final del robot, esto permite estudiar la relación que existe entre las variables articulares y la posición y orientación del efector final del robot con relación a la base del mismo. Por lo tanto en el modelo cinemático directo se involucra esencialmente el estudio de la relación entre las variables articulares y el espacio de trabajo del robot, que suele ser un espacio cartesiano.

El modelo cinemático directo viene dado por una función que permite expresar la posición y orientación del sistema de referencia en el espacio cartesiano p en términos de las variables articulares θ :

$$p = \varphi(\theta) \tag{A.1}$$

donde φ es un conjunto de funciones no lineales.

Para obtener el modelo cinemático es necesario definir un marco coordinado de referencia fijo asociado a cada articulación, posteriormente es posible obtener la matriz de transformación asociada a cada eslabón. Una de las convenciones más utilizadas para la selección de manera sistemática de los sistemas coordinados unidos a cada articulación es la representación de *Denavit-Hartenberg*

De acuerdo a la convención *Denavit-Hartenberg* (DH), un robot articulado puede describirse mediante cuatro magnitudes asociadas a cada junta del robot, una de las magnitudes corresponde a la variable de la articulación y las otras tres representan parámetros fijos que dependen de cómo está constituido el manipulador tal como se describe a continuación.

En la convención DH las transformaciones homogéneas entre cada sistema coordinado puede ser representada mediante cuatro transformaciones básicas representadas mediante la siguiente expresión:

$$T_i = Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i} \quad (A.2)$$

donde Rot_{z,θ_i} representa una matriz de rotación del ángulo θ_i alrededor del eje z y $Trans_{z,d_i}$ define la matriz de traslación de una distancia d_i a lo largo del eje z . La ecuación A.2 se puede representar en forma matricial como aparece en la expresión A.3:

$$T_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (A.3)$$

donde la variable de una articulación i de rotación es representada mediante el ángulo θ_i y una prismática mediante el desplazamiento d_i . Los otros dos parámetros representan la distancia a_i entre el eje de la articulación $i - 1$ y el eje de la articulación i y el ángulo α_i entre estos dos ejes. Los cuatro elementos anteriores son denominados parámetros de *Denavit-Hartenberg*.

A partir de las matrices asociadas a cada articulación, puede obtenerse la transformación compuesta T_i que relaciona la localización del sistema coordinado i con el sistema origen, que es función de las i variables de las articulaciones: $\theta_1, \theta_2, \dots, \theta_n$. De esta manera el modelo cinemático directo del manipulador queda definido mediante la siguiente expresión:

$$H = \varphi(\theta) = T_1 \cdot T_2 \dots T_i \quad (A.4)$$

La aplicación de la ecuación A.4 permite estimar la posición y orientación del efector final del manipulador a partir de los valores de las variables articulares.

Junta	θ_i	α_i	a_i	d_i
T_1	0	-90	100	0
T_2	90	180	370	0
T_3	0	-90	85	0
T_4	0	90	0	-380
T_5	0	90	0	0
T_6	-180	180	0	-90

Cuadro A.1: Parámetros *Denavit-Hartenberg* para el Robot Motoman HP3

En el cuadro A.1 se presentan los parámetros DH obtenidos para el robot Motoman, modelo HP3 (figura A.1).

El modelo cinemático del robot es obtenido mediante la expresión:

$$H = T_1 \cdot T_2 \cdot T_3 \cdot T_4 \cdot T_5 \cdot T_6$$

donde la matriz T_i representa la matriz de transformación asociada a la junta i del manipulador.

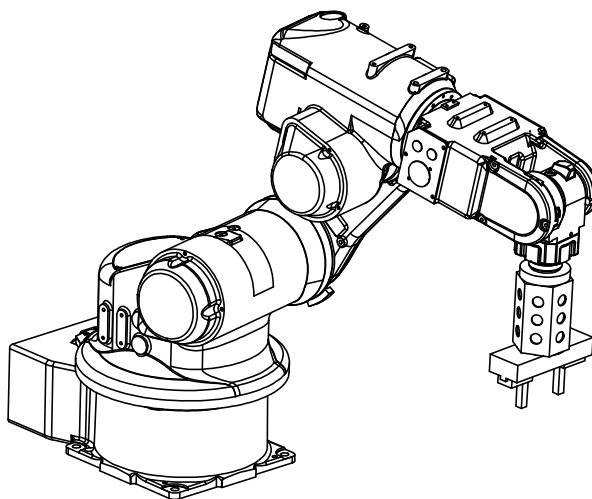


Figura A.1: Robot manipulador motoman HP3

Bibliografía

- [Akkoul et al., 2014] Akkoul, S., Hafiane, A., Leconge, R., Harrar, K., Lespessailles, E., and Jennane, R. (2014). 3d reconstruction method of the proximal femur and shape correction. In *2014 4th International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6. [9](#)
- [Batista et al., 1998] Batista, J., Araújo, H., and de Almeida, A. T. (1998). Iterative multi-step explicit camera calibration. In *ICCV*, pages 709–714. [9](#)
- [Batteoui et al., 2016] Batteoui, I. E., Satori, K., and Saaidi, A. (2016). A stereo planar self-calibration method of a camera with variable intrinsic parameters. In *2016 13th International Conference on Computer Graphics, Imaging and Visualization (CGiV)*, pages 6–11. [10](#)
- [Baturone, 2005] Baturone, A. O. (2005). *Robótica: manipuladores y robots móviles*. Marcombo ISBN: 9788426713131. [5](#)
- [Bonilla et al., 2010] Bonilla, I., Gonzalez-Galvan, E., Chavez-Olivares, C., Mendoza, M., Loredó-Flores, A., Reyes, F., and Zhang, B. (2010). A vision-based, impedance control strategy for industrial robot manipulators. In *Automation Science and Engineering (CASE), 2010 IEEE Conference on*, pages 216–221. [8](#)
- [Bonilla et al., 2012] Bonilla, I., Mendoza, M., Gonzalez-Galván, E. J., Chavez-Olivares, C., Loredó-Flores, A., and Reyes, F. (2012). Path-tracking maneuvers with industrial robot manipulators using uncalibrated vision and impedance control. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):1716–1729. [8](#)

BIBLIOGRAFÍA

- [Boudine et al., 2016] Boudine, B., Kramm, S., Akkad, N. E., Bensrhair, A., Saaidi, A., and Satori, K. (2016). A flexible technique based on fundamental matrix for camera self-calibration with variable intrinsic parameters from two views. *Journal of Visual Communication and Image Representation*, 39:40 – 50. [10](#)
- [Burger et al., 2015] Burger, W., Dean-Leon, E., and Cheng, G. (2015). Robust second order sliding mode control for 6d position based visual servoing with a redundant mobile manipulator. In *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pages 1127–1132. [7](#)
- [Burlacu et al., 2011] Burlacu, A., Copot, C., Cervera, E., and Lazar, C. (2011). Real-time visual predictive control of manipulation systems. In *Advanced Robotics (ICAR), 2011 15th International Conference on*, pages 383–388. [7](#)
- [Cai et al., 2016] Cai, C., Somani, N., and Knoll, A. (2016). Orthogonal image features for visual servoing of a 6-dof manipulator with uncalibrated stereo cameras. *IEEE Transactions on Robotics*, 32(2):452–461. [7](#)
- [Chang et al., 2004] Chang, F., Chen, C.-J., and Lu, C.-J. (2004). A linear-time component-labeling algorithm using contour tracing technique. *Computer Vision and Image Understanding*, 93(2):206 – 220. [93](#)
- [Chen, 2007] Chen, H. S. (2007). *Error Analysis of Camera-Space Manipulation*. PHD Thesis, University of Notre Dame. [67](#), [79](#), [80](#), [90](#)
- [de Villiers et al., 2015] de Villiers, J., Jermy, R., and Nicolls, F. (2015). A study on the effect of different image centres on stereo triangulation accuracy. In *Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech), 2015*, pages 24–29. [9](#)
- [Dhond and Aggarwal, 1991] Dhond, U. R. and Aggarwal, K. K. (1991). Structure from stereo: A review. In Iyengar, S. S. and Elfes, A., editors, *Autonomous Mobile Robots: Perception, Mapping, and Navigation (Vol. 1)*, pages 25–46. IEEE Computer Society Press, Los Alamitos, CA. [9](#)
- [Duflot et al., 2016] Duflot, L. A., Krupa, A., Tamadazte, B., and Andreff, N. (2016). Towards ultrasound-based visual servoing using shearlet coefficients. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3420–3425. [7](#)

- [Ens and Lawrence, 1993] Ens, J. and Lawrence, P. (1993). An investigation of methods for determining depth from focus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(2):97–108. [9](#)
- [Faugeras, 1993] Faugeras, O. (1993). *Three-dimensional computer vision, a geometric viewpoint*. MIT Press. [9](#)
- [Faugeras et al., 1992] Faugeras, O. D., Luong, Q.-T., and Maybank, S. J. (1992). Camera self-calibration: Theory and experiments. In *European Conference on Computer Vision*, pages 321–334. [9](#)
- [García, 2007] García, M. (2007). *Implementación de un método de calibración aplicado al control basado en visión de un robot manipulador y comparativa experimental con el método CSM*. Tesis de Maestría, Universidad Autónoma de San Luis Potosí. [2](#), [87](#)
- [Gonzalez-Galvan et al., 2011] Gonzalez-Galvan, E., Chavez, C., Bonilla, I., Mendoza, M., Raygoza, L., Lored-Flores, A., and Zhang, B. (2011). Precise industrial robot positioning and path-tracking over large surfaces using non-calibrated vision. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 5160–5166. [7](#)
- [Gonzalez-Galvan et al., 2009] Gonzalez-Galvan, E., Lored-Flores, A., Raygoza, L., Palos, J., and Skaar, S. (2009). Precise and robust large-shape reproduction using uncalibrated vision. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 2799–2803. [8](#)
- [González-Galván et al., 2008] González-Galván, E. J., Lored-Flores, A., Jesús Cervantes-Sánchez, J., Antonio Aguilera-Cortés, L., and Skaar, S. B. (2008). An optimal path-generation algorithm for manufacturing of arbitrarily curved surfaces using uncalibrated vision. *Robot. Comput.-Integr. Manuf.*, 24(1):77–91. [7](#)
- [González-Galván and Seelinger, 1999] González-Galván, E. J. and Seelinger, M. J. (1999). Efficient camera-space target disposition in a matrix of moments structure using camera-space manipulation. *I. J. Robotic Res.*, 18(8):809–818. [30](#)
- [González-Galván et al., 1997] González-Galván, E. J., Skaar, S. B., Korde, U. A., and Chen, W. (1997). Application of a precision-enhancing measure in 3d rigid-body positioning using camera-space manipulation. *I. J. Robotic Res.*, 16(2):240–257. [30](#)

- [Gonzalez-Galvan et al., 1999] Gonzalez-Galvan, E. J., Skaar, S. B., and Seelinger, M. J. (1999). Efficient camera-space manipulation. *International Journal of Robotics Research*, 18(8):809–818. [8](#), [10](#)
- [Guillemaut et al., 2005] Guillemaut, J.-Y., Aguado, A. S., and Illingworth, J. (2005). Using points at infinity for parameter decoupling in camera calibration. *Transactions on Pattern Analysis and Machine Intelligence*, 27(2):265–270. [9](#)
- [Hajiloo et al., 2016] Hajiloo, A., Keshmiri, M., Xie, W. F., and Wang, T. T. (2016). Robust online model predictive control for a constrained image-based visual servoing. *IEEE Transactions on Industrial Electronics*, 63(4):2242–2250. [7](#)
- [Hartley, 1994] Hartley, R. I. (1994). Self-calibration from multiple views with a rotating camera. In Eklundh, J.-O., editor, *ECCV*, volume 800 of *Lecture Notes in Computer Science*, pages 471–478. Springer. [9](#)
- [Hartley and Sturm, 1997] Hartley, R. I. and Sturm, P. (1997). Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157. [9](#)
- [Hartley and Zisserman, 2004] Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition. [6](#), [8](#), [24](#), [25](#), [26](#), [36](#), [43](#), [83](#)
- [Hashimoto, 2003] Hashimoto, K. (2003). A review on vision-based control of robot manipulators. *Advanced Robotics*, 17(10):969–991. [7](#)
- [Herrera et al., 2016] Herrera, D., Kannala, C. J., and Heikkila, J. (2016). Forget the checkerboard: Practical self-calibration using a planar scene. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. [9](#)
- [Horn and Brooks, 1989] Horn, B. K. P. and Brooks, M. J., editors (1989). *Shape from shading*. MIT Press, Cambridge, MA, USA. [9](#)
- [Hutchinson et al., 1996] Hutchinson, S., Hager, G., and Corke, P. (1996). A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–670. [6](#)
- [Hwang et al., 2015] Hwang, K. S., Chung, M. H., and Jiang, W. C. (2015). Image based visual servoing using proportional controller with compensator. In *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on*, pages 347–352. [7](#)

- [Khurshid et al., 2016] Khurshid, R., Fitter, N., Fedalei, E., and Kuchenbecker, K. (2016). Effects of grip-force, contact, and acceleration feedback on a teleoperated pick-and-place task. *IEEE Transactions on Haptics*, PP(99):1–1. [6](#)
- [Larson et al., 2009] Larson, R., Edwards, B., and Falvo, D. (2009). *Elementary Linear Algebra*. Houghton Mifflin Harcourt Publishing Company, ISBN: 9780618783762, 6 edition. [26](#)
- [Li et al., 2015a] Li, J., Duan, P., and Wang, J. (2015a). Binocular stereo vision calibration experiment based on essential matrix. In *Computer and Communications (ICCC), 2015 IEEE International Conference on*, pages 250–254. [9](#)
- [Li et al., 2015b] Li, M., Zheng, D., Zhang, R., Yin, J., and Tian, X. (2015b). Overview of 3d reconstruction methods based on multi-view. In *2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics*, volume 2, pages 145–148. [10](#)
- [Liu et al., 2014a] Liu, J., Li, Y., and Chen, S. (2014a). Robust camera calibration by optimal localization of spatial control points. *IEEE Transactions on Instrumentation and Measurement*, 63(12):3076–3087. [9](#)
- [Liu et al., 2016a] Liu, R., Zhang, H., Lu, J., and Sun, Y. (2016a). Extrinsic parameters and focal length calibration using rotation-symmetric patterns. *IET Image Processing*, 10(3):189–197. [8](#)
- [Liu et al., 2016b] Liu, Y., Liu, S., Cao, Y., and Wang, Z. (2016b). Automatic chessboard corner detection method. *IET Image Processing*, 10(1):16–23. [9](#)
- [Liu et al., 2014b] Liu, Y., Shi, D., and Skaar, S. B. (2014b). Robust industrial robot real-time positioning system using vw-camera-space manipulation method. *Industrial Robot: An International Journal*, 41(1):70–81. [7](#)
- [Liu and Chopra, 2013] Liu, Y.-C. and Chopra, N. (2013). Control of semi-autonomous teleoperation system with time delays. *Automatica*, 49(6):1553 – 1565. [6](#)
- [Liu et al., 2016c] Liu, Z., Huang, J., Bu, S., Han, J., Tang, X., and Li, X. (2016c). Template deformation-based 3-d reconstruction of full human body scans from low-cost depth cameras. *IEEE Transactions on Cybernetics*, PP(99):1–14. [9](#)

- [Ma et al., 2000] Ma, Q., Cardenas, A., Goodwine, B., and Skaar, S. B. (2000). Supervisory control of a mobile robot using point-and-click mobile camera-space manipulation. In Sánchez, B., II, R. H., Soriano, M., and Tiako, P., editors, *Proceedings of the World Multiconference on Systemics, cybernetics and Informatics*, volume IX, pages 38–43. [6](#)
- [Ma and Su, 2015] Ma, Z. and Su, J. (2015). Robust uncalibrated visual servoing control based on disturbance observer. *{ISA} Transactions*, 59:193 – 204. [7](#)
- [Martínez et al., 2016] Martínez, C. A. L., van de Molengraft, R., Weiland, S., and Steinbuch, M. (2016). Switching robust control for bilateral teleoperation. *IEEE Transactions on Control Systems Technology*, 24(1):172–188. [6](#)
- [Maybank and Faugeras, 1990] Maybank, S. and Faugeras, O. D. (1990). A theory of self-calibration of a moving camera. *International Journal of Computer Vision*, 8(2):123–151. [9](#)
- [Moeller et al., 2015] Moeller, M., Benning, M., Schonlieb, C., and Cremers, D. (2015). Variational depth from focus reconstruction. *IEEE Transactions on Image Processing*, 24(12):5369–5378. [9](#)
- [Nadi et al., 2014] Nadi, F., Derhami, V., and Rezaeian, M. (2014). Visual servoing control of robot manipulator with jacobian matrix estimation. In *Robotics and Mechatronics (ICRoM), 2014 Second RSI/ISM International Conference on*, pages 405–409. [7](#)
- [Otero and Sánchez, 2015] Otero, J. and Sánchez, L. (2015). Local iterative {DLT} soft-computing vs. interval-valued stereo calibration and triangulation with uncertainty bounding in 3d reconstruction. *Neurocomputing*, 167:44 – 51. [9](#)
- [Ourak et al., 2016] Ourak, M., Tamadazte, B., Lehmann, O., and Andreff, N. (2016). Wavelets-based 6 dof visual servoing. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3414–3419. [7](#)
- [Park et al., 2011] Park, S., Seo, C., Kim, J.-P., and Ryu, J. (2011). Robustly stable rate-mode bilateral teleoperation using an energy-bounding approach. *Mechatronics*, 21(1):176 – 184. [6](#)

- [Parsapour et al., 2013] Parsapour, M., RayatDoost, S., and Taghirad, H. D. (2013). Position based sliding mode control for visual servoing system. In *Robotics and Mechatronics (ICRoM), 2013 First RSI/ISM International Conference on*, pages 337–342. [7](#)
- [Passenberg et al., 2010] Passenberg, C., Peer, A., and Buss, M. (2010). A survey of environment, operator, and task-adapted controllers for teleoperation systems. *Mechatronics*, 20(7):787 – 801. [6](#)
- [Press et al., 2007] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007). *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 3 edition. [26](#)
- [Qing and Yue, 2015] Qing, L. S. and Yue, L. S. (2015). Online-estimation of image jacobian based on adaptive kalman filter. In *Control Conference (CCC), 2015 34th Chinese*, pages 6016–6019. [7](#)
- [Quan and Triggs, 2000] Quan, L. and Triggs, B. (2000). A unification of autocalibration methods. In *Asian Conference on Computer Vision*, pages 917–922. [9](#)
- [Rahman and Krouglicof, 2012] Rahman, T. and Krouglicof, N. (2012). An efficient camera calibration technique offering robustness and accuracy over a wide range of lens distortion. *Image Processing, IEEE Transactions on*, 21(2):626–637. [9](#)
- [Sharma et al., 2015] Sharma, A., Chaudhary, S., Roy, S. D., and Chand, P. (2015). Three dimensional reconstruction of cylindrical pellet using stereo vision. In *2015 Fifth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*, pages 1–4. [9](#)
- [Skaar et al., 1987] Skaar, S. B., Brockman, W. H., and Hanson, R. (1987). Camera space manipulation. *International Journal of Robotics Research*, 6(4):20–32. [6](#), [8](#), [10](#), [11](#)
- [Skaar et al., 1990] Skaar, S. B., Brockman, W. H., and Jang, W. S. (1990). Three dimensional camera space manipulation. *International Journal of Robotics Research*, 9(4):22–39. [7](#), [10](#), [28](#)
- [Song et al., 2016] Song, H., Choi, W., and Kim, H. (2016). Robust vision-based relative-localization approach using an rgb-depth camera and lidar sensor fusion. *IEEE Transactions on Industrial Electronics*, 63(6):3725–3736. [9](#)

- [Sorgi and Bushnevskiy, 2015] Sorgi, L. and Bushnevskiy, A. (2015). Color-coded pattern for non metric camera calibration. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 4878–4882. [9](#)
- [Soyguder and Abut, 2016] Soyguder, S. and Abut, T. (2016). Haptic industrial robot control with variable time delayed bilateral teleoperation. *Industrial Robot: An International Journal*, 43(4):null. [6](#)
- [Sun et al., 2014] Sun, D., Naghdy, F., and Du, H. (2014). Application of wave-variable control to bilateral teleoperation systems: A survey. *Annual Reviews in Control*, 38(1):12 – 31. [6](#)
- [Sun et al., 2016] Sun, D., Naghdy, F., and Du, H. (2016). A novel approach for stability and transparency control of nonlinear bilateral teleoperation system with time delays. *Control Engineering Practice*, 47:15 – 27. [6](#)
- [Tsai, 1996] Tsai, R. Y. (1996). An efficient and accurate camera calibration technique for 3d machine vision. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 364–374, Miami Beach, FL. [9](#)
- [Wang et al., 2012] Wang, D., Bai, Y., and Zhao, J. (2012). Robot manipulator calibration using neural network and a camera-based measurement system. *Transactions of the Institute of Measurement and Control*, 34(1):105–121. [8](#)
- [Wang and Cheng, 2016] Wang, G. and Cheng, J. (2016). Three-dimensional reconstruction of hybrid surfaces using perspective shape from shading. *Optik - International Journal for Light and Electron Optics*, 127(19):7740 – 7751. [9](#)
- [Westmore and Wilson, 1991] Westmore, D. and Wilson, W. J. (1991). Direct dynamic control of a robot using an end-point mounted camera and kalman filter position estimation. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pages 2376–2384 vol.3. [7](#)
- [Winston and Horn, 1975] Winston, P. and Horn, B. (1975). *The psychology of computer vision*. McGraw-Hill computer science series. McGraw-Hill. [9](#)
- [Wong et al., 2003] Wong, K.-Y. K., Mendonça, P. R., and Cipolla, R. (2003). Camera calibration from surfaces of revolution. *Transactions on Pattern Analysis and Machine Intelligence*, 25(2):147–161. [9](#)

- [Wu and Zhu, 2015] Wu, L. and Zhu, B. (2015). Binocular stereovision camera calibration. In *2015 IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 2638–2642. [9](#)
- [Xu and Deng, 2015] Xu, J. and Deng, F. (2015). A camera self-calibration method based on ios-pso. In *Chinese Automation Congress (CAC), 2015*, pages 489–494. [10](#)
- [Yang et al., 2015] Yang, L., Zhang, N., Ren, Y., and Hu, H. (2015). Linear perspective shape-from-shading method with two images. *Journal of Systems Engineering and Electronics*, 26(5):1080–1087. [9](#)
- [Zacharia et al., 2009] Zacharia, P., Aspragathos, N., Mariolis, I., and Dermatas, E. (2009). A robotic system based on fuzzy visual servoing for handling flexible sheets lying on a table. *Industrial Robot: An International Journal*, 36(5):489–496. [7](#)
- [Zhai and Xia, 2016] Zhai, D. H. and Xia, Y. (2016). Adaptive control of semi-autonomous teleoperation system with asymmetric time-varying delays and input uncertainties. *IEEE Transactions on Cybernetics*, PP(99):1–13. [6](#)
- [Zhang et al., 2011] Zhang, B., Wang, J., Rossano, G., and Martinez, C. (2011). Vision-guided robotic assembly using uncalibrated vision. In *Mechatronics and Automation (ICMA), 2011 International Conference on*, pages 1384–1389. [8](#)
- [Zhang, 2000] Zhang, Z. (2000). A flexible new technique for camera calibration. *Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334. [9](#)