

UNIVERSIDAD AUTÓNOMA DE SAN LUIS POTOSÍ

FACULTAD DE INGENIERÍA
ÁREA MECÁNICA Y ELÉCTRICA
CENTRO DE INVESTIGACIÓN Y ESTUDIOS
DE POSGRADO



Tesis que presenta:

Ing. Miguel Ángel Rodríguez y Domínguez Kessler

Para recibir el título de:

Maestro en Ingeniería Mecánica

“Cámaras virtuales para el control basado en visión de robots industriales”

Asesor:

Dr. Emilio Jorge González Galván

SAN LUIS POTOSÍ, S. L. P. MÉXICO

Febrero de 2013

Resumen

La presente tesis, describe el desarrollo de un sistema de adquisición de imágenes basado en cámaras virtuales, cuya aplicación es proyectada como medio de captura de información visual, para el control por visión de robots industriales mediante una conocida metodología llamada *Manipulación de Espacio de Cámara (CSM, por sus siglas en inglés)*. Dicho sistema de adquisición, aprovecha la conjunción de una cámara de video con un serie de espejos, que al interferir con el campo visual del sensor, producen observadores virtuales mejor conocidos como cámaras virtuales; siendo estos equivalentes a como si se contara con múltiples sensores en un sistema de adquisición de imágenes.

En un primer capítulo se hace un estudio del estado del arte de las cámaras virtuales, ofreciéndose un espectro general de sus aplicaciones en diversos campos, que van desde la reconstrucción tridimensional de objetos, hasta el control por visión de robots manipuladores. En este capítulo también se plantean los objetivos y las motivaciones del presente trabajo de investigación.

El segundo capítulo del presente proyecto de investigación, trata explícitamente de la metodología *CSM*, ahondando en los algoritmos necesarios para llevar a cabo el control por visión de un robot industrial, los cuales son aplicados posteriormente para la validación experimental del sistema.

El tercer capítulo, muestra a detalle el proceso de desarrollo de un sistema de cámaras virtuales, atendiendo a ciertos lineamientos de la metodología de *CSM*, sin dejar de lado las limitaciones que puede traer consigo el sistema. Este capítulo cubre además, ciertos tópicos complementarios como lo es el procesamiento necesario imágenes provenientes del sistema de cámaras virtuales, para extraer de ellas información útil para un sistema de control por visión de robots industriales *CSM*. En este capítulo también se retoman algunos algoritmos del capítulo anterior, detallan-

do ciertos aspectos adicionales necesarios para llevar a cabo el control de un robot.

El cuarto capítulo reporta los resultados experimentales, de las pruebas de validación del sistema de cámaras virtuales desarrollado a ser aplicado al control de un manipulador mediante la metodología de *CSM*, dando pie a la discusión de los resultados obtenidos, siendo estos positivos en un panorama general, demostrándose la viabilidad del uso de cámaras virtuales, en estos.

Las conclusiones del presente trabajo de investigación se muestran en una sección adicional, en la que se plantea además, trabajo futuro para posteriores trabajos de investigación.

Finalmente, en apéndices complementarias se ofrece información de apoyo para los capítulos de esta tesis, así como algunas rutinas computacionales para reproducir algo del trabajo aquí realizado.

Agradecimientos

De todo corazón a mis padres Adán e Ida, por su ejemplar labor en mi formación, por todas las grandes oportunidades que me han brindado en la vida, por el amor, el respeto y el cariño que siempre me han dado.

A mis hermanos Margarita, Marcos, Peter y Theresia, que siempre han sido para mi modelos a seguir y grandes consejeros.

A mi novia Sara, por todo el amor, la alegría, el apoyo, y la motivación que como un motor en ella encuentro.

A mi asesor el Dr. Emilio, por el honor que para mi significa la oportunidad de haber sido su alumno, por toda la labor verdaderamente profesional y de gran calidad como guía en este proyecto de investigación, por todo el apoyo, la disposición y los excelentes recursos que me fueron brindados.

A mis sinodales el Dr. Mauro, el Dr. Antonio y la Dra. Nancy, por su labor de refuerzo y retroalimentación imprescindible para concluir este proyecto de investigación, por su disposición y experiencia.

A mis grandes amigos Francisco, David, Joel, Alberto y Diana, por la amistad sincera que de ellos siempre he tenido.

A la UASLP por ser una institución de enseñanza firme, de propósitos constantes y uno de los grandes pilares de mi formación.

Al CIEP-FI y a todo su personal, por la lucha constante por mantener una oferta educativa de alto nivel, por la calidad de sus instalaciones y por los gratos recuerdos que conservo.

A mis compañeros y amigos de la maestría, en especial a Reynado, Ares, Aarón, Cesar y Liz por su amistad, compañía, ayuda y buenos momentos.

Al CONACYT, porque sin él nada de esto sería posible.

A todos muchísimas gracias!!!

Índice general

Introducción	1
1. Antecedentes y Alcances del Proyecto	4
1.1. Antecedentes	7
1.2. Proyectos Precedentes	11
1.3. Objetivos y del Proyecto	17
2. El Método de Manipulación de Espacio de Cámara	21
2.1. Cinemática Directa del Manipulador	24
2.2. Modelo de Cámara de Perspectiva	33
2.3. Modelo de Cámara de Perspectiva Recursivo	35
2.4. Estimación de Parámetros de Cámara	40
2.4.1. Estimación de los Parámetros de Cámara 1..6	41
2.4.2. Estimación del Parámetro de Cámara 7	45
2.5. Algoritmo de Estimación de Posición Tridimensional de Marcas Visuales	46
2.6. Algoritmo de Estimación de la Configuración del Manipulador	48
3. Determinación de Parámetros de Visión para Cámaras Virtuales	51
3.1. Diseño, Simulación y Construcción de un Sistema de Cámaras Vir- tuales Compatible con la Metodología de CSM	52
3.1.1. Diseño del Sistema de Cámaras Virtuales	54
3.1.2. Simulación del Sistema de Cámaras Virtuales	57
3.1.3. Construcción del Sistema de Cámaras Virtuales	67
3.2. Equivalencia del Sistema de Cámaras Virtuales Diseñado con un Sis- tema de Adquisición de Múltiples Sensores	69
3.3. Procesamiento de Imágenes	73
3.3.1. Separación de Imágenes	74
3.3.2. Detección de Marcas Artificiales	76

3.3.3.	Diagrama de Flujo del procesamiento de imágenes	85
3.4.	Maniobra de Posicionamiento	86
3.4.1.	Ejecución de Trayectoria Preplaneada	86
3.4.2.	Definición del Objetivo de una Maniobra de Posicionamiento y Caracterización de Superficies	88
3.4.3.	Ejecución de Maniobra de Posicionamiento y Refinamiento de Parámetros de Visión	92
3.4.4.	Control de Orientación del Manipulador	97
4.	Resultados Experimentales	105
4.1.	Equipo Disponible en el Laboratorio	107
4.2.	Determinación de la Relación “mm/px” por Cámara	107
4.3.	Experimentos con Maniobras de Posicionamiento	114
4.3.1.	Primera Secuencia Experimental de Pruebas de Posicionamiento	120
4.3.2.	Segunda Secuencia Experimental de Pruebas de Posicionamiento	122
4.3.3.	Tercera Secuencia Experimental de Pruebas de Posicionamiento	124
4.3.4.	Cuarta Secuencia Experimental de Pruebas de Posicionamiento	126
4.3.5.	Quinta Secuencia Experimental de Pruebas de Posicionamiento	128
4.3.6.	Sexta Secuencia Experimental de Pruebas de Posicionamiento	130
4.3.7.	Discusión de Resultados de Experimentos con Maniobras de Posicionamiento	132
4.4.	Prueba de detección de diferencias de relieve en una superficie	137
4.4.1.	Discusión	141
	Conclusiones y Trabajo Futuro	142
	Bibliografía	146
	Appendices	151
A.	Rutinas de Simulación de Sistema de Cámaras Virtuales	151
A.1.	Rutina Principal	151
A.2.	Subrutina para Creación de Espejos	156
A.3.	Subrutina para Creación de Campo Visual	161
A.4.	Subrutina para el Análisis del Eje Focal del Sistema	169
A.5.	Otras subrutinas	171
B.	Intersección entre una Recta y un Plano y Vector Reflejado Resultante	178
B.1.	Cálculo del Punto de Intersección entre una Recta y un Plano	178
B.2.	Cálculo del Vector Reflejado Resultante	180

C.	Rutinas para el Procesamiento de Imágenes Utilizando OpenCV . . .	183
C.1.	La clase CProcesamiento3	184
C.1.1.	Procesamiento3.h	184
C.1.2.	Procesamiento3.cpp	187
D.	Interfaz Gráfica para Validación Experimental	196
D.1.	Entorno General	196

Índice de figuras

1.1.	Bosquejos de cámaras virtuales con elementos reflexivos planos [1].	8
1.2.	Ejemplo de cámaras virtuales en un sistema catadióptrico; a la izquierda el bosquejo del sistema, a la derecha un prototipo [2].	9
1.3.	Ejemplo de mapas de profundidad; a la izquierda imágenes originales de un sistema estereoscópico, a la derecha mapas de profundidad [1].	10
1.4.	Bosquejo de un sistema de control de un robot no-holonómico, empleando cámaras fijas [3].	12
1.5.	Sistema automatizado de ensamble de componentes electrónicos, basado en un control por visión CSM [4].	14
1.6.	Prototipo de vehículo de exploración del planeta marte llamado “FIDO”, controlado mediante MCSM [5].	16
1.7.	Robot Fanuc M16ib/20T; robot de amplio espacio de trabajo, controlado con CSM mediante multiples cámaras [6].	17
2.1.	Esquema representativo de un sistema de control por visión basado en CSM	22
2.2.	Diagrama cinemático de robot industrial Fanuc M16-iB/20T	26
2.3.	Representación del efecto de paralelogramo en el robot industrial Fanuc M16-iB/20T.	29
2.4.	Esquema de un sistema de visión por computadora, aplicado de forma práctica.	33
2.5.	Esquema de proyección de perspectiva modelada a través de modelo de cámara de orificio.	35
2.6.	Esquema de tarea de posicionamiento de un manipulador industrial en donde se representa la colocación del rin de un neumático en el eje de un automóvil.	36
2.7.	Algoritmo de estimación de parametros de cámara $P_{1,6}$	45
2.8.	Esquema de algoritmo para cálculo del parámetro de cámara P_7	46

2.9. Ejemplificación de empate de marcas visuales artificiales y marcas visuales objetivo, durante algoritmo de estimación de configuración interna del manipulador.	50
3.1. Comportamiento de la variancia de la estimación de la posición tridimensional de un objetivo en <i>CSM</i> , con respecto del ángulo entre ejes focales de dos cámaras.	54
3.2. Bosquejos de sistemas de cámaras virtuales encontrados en la literatura; a) Con dos espejos; b) Con tres espejos; c) Con cuatro espejos; d) Con cinco espejos.	55
3.3. Reflexión de un rayo incidente en un medio reflejante.	58
3.4. Análisis de óptico-geométrico de sistema de cámaras virtuales, compuesto de 4 espejos y una sola cámara real.	58
3.5. Ángulos de orientación del espejo lateral 1.	62
3.6. Ángulos de orientación de espejo lateral 2.	63
3.7. Simulación de sistema de cámaras virtuales en Matlab [®]	65
3.8. Simulación de sistema de cámaras virtuales en diferentes configuraciones; a) Con objetivo en $\mathbf{P}\{640, 3660, -1800\}_0$; b) Con objetivo en $\mathbf{P}\{640, 0, -1800\}_0$	66
3.9. Soportes de espejos de sistema de cámaras virtuales	68
3.10. Ejemplo de imagen adquirida mediante el sistema de cámaras virtuales diseñado; a) Vista de perspectiva izquierda; b) Vista de perspectiva derecha.	68
3.11. Equivalencia entre cámaras virtuales, del sistema catadióptrico de adquisición diseñado, y cámaras reales	69
3.12. Representación de la disminución de la densidad de resolución de las capturas.	70
3.13. Relación de aspecto entre la proyección en espacio de cámara de un cuerpo lejano y un cuerpo cercado de mismas dimensiones.	71
3.14. Niveles de un sistema de visión.	73
3.15. Proceso de separación de imágenes.	75
3.16. Ejemplo de diferenciación de imágenes.	77
3.17. Etapas del procesamiento de imágenes, necesario para el aislamiento y cálculo de los centroides de múltiples marcas visuales	78
3.18. Ejemplos de marcas visuales cuyo centroide es calculado mediante diferentes métodos	79
3.19. Ejemplificación del algoritmo de Freeman para la búsqueda de contornos	83

3.20. Cambio de origen de coordenadas para los centroides de las marcas visuales.	84
3.21. Algoritmo de general procesamiento de imágenes	85
3.22. Trayectoria preplaneada	88
3.23. Definición del objetivo de maniobra.	90
3.24. Aproximación de un plano a marcas visuales	92
3.25. LEDs sobre cuerpo prismático para el refinamiento de los parámetros de cámara.	93
3.26. Cuerpo prismático hexagonal del manipulador.	94
3.27. Definición de vectores unitarios sobre las caras del cuerpo hexagonal de herramienta del manipulador.	95
3.28. Traslado de marcas visuales hasta herramienta del manipulador, para conformar información artificial.	102
4.1. Equipo disponible en el laboratorio de robótica de la UASLP-FI; a) Robot Fanuc M16iB/20T; b) Cámara Ueye mod. UI-1450-CM; c), e) Espejos de medidas 1000 × 1000 mm; d) Espejos de medidas 300 × 300 mm; f) Herramienta del robot, con cuerpo prismático hexagonal.	108
4.2. Plano construido para experimentos de cuantificación de resolución específica.	109
4.3. Soporte de emisor láser para simulación del eje óptico de cada cámara virtual.	110
4.4. Tablero de madera montado y alineado con haz de luz láser, para simular un plano perpendicular al eje óptico de la cámara.	111
4.5. Secuencia repetitiva de movimientos de robot Fanuc M16-iB/20T para medición de segmento de cuadrícula sobre tablero de madera.	112
4.6. Gráfica de correlación entre distancia a objetivo (D) y resolución específica promedio (R_p).	113
4.7. Gráfica de resultados por experimento del valor de resolución específica.	114
4.8. Experimento de maniobra de posicionamiento.	115
4.9. Plano para localización de objetivo en pruebas de posicionamiento.	116
4.10. Instrumentos de medición empleados para cuantificar el error de posición y orientación en una maniobra de posicionamiento; a) Vernier; b) Compás; c) Escuadra universal	117
4.11. Orden de rotaciones registradas en el plano de pruebas para cada secuencia experimental	118

4.12. Referencias para la medición del orientación de la herramienta del manipulador en el objetivo	119
4.13. Gráfica del error tridimensional por experimento, de primera secuencia experimental.	120
4.14. Gráfica del error de posición por eje coordenado, en maniobras de posicionamiento de primera secuencia experimental.	121
4.15. Gráfica del error de orientación angular, en maniobras de posicionamiento de primera secuencia experimental.	121
4.16. Gráfica del error tridimensional por experimento, de segunda secuencia experimental.	122
4.17. Gráfica del error de posición por eje coordenado, en maniobras de posicionamiento de segunda secuencia experimental.	123
4.18. Gráfica del error de orientación angular, en maniobras de posicionamiento de segunda secuencia experimental.	123
4.19. Gráfica del error tridimensional por experimento, de tercera secuencia experimental.	124
4.20. Gráfica del error de posición por eje coordenado, en maniobras de posicionamiento de tercera secuencia experimental.	125
4.21. Gráfica del error de orientación angular, en maniobras de posicionamiento de tercera secuencia experimental.	125
4.22. Gráfica del error tridimensional por experimento, de cuarta secuencia experimental.	126
4.23. Gráfica del error de posición por eje coordenado, en maniobras de posicionamiento de cuarta secuencia experimental.	127
4.24. Gráfica del error de orientación angular, en maniobras de posicionamiento de cuarta secuencia experimental.	127
4.25. Gráfica del error tridimensional por experimento, de quinta secuencia experimental.	128
4.26. Gráfica del error de posición por eje coordenado, en maniobras de posicionamiento de quinta secuencia experimental.	129
4.27. Gráfica del error de orientación angular, en maniobras de posicionamiento de quinta secuencia experimental.	129
4.28. Gráfica del error tridimensional por experimento, de sexta secuencia experimental.	130
4.29. Gráfica del error de posición por eje coordenado, en maniobras de posicionamiento de sexta secuencia experimental.	131

4.30. Gráfica del error de orientación angular, en maniobras de posicionamiento de sexta secuencia experimental.	131
4.31. Distorsión óptica en espejo del sistema de cámaras virtuales.	135
4.32. Comparación entre proyecciones en espacio de cámara, de las marcas visuales objetivo y las marcas artificiales sobre la herramienta del manipulado; a la izquierda cámara 1; a la derecha cámara 2.	136
4.33. Representación de cuerpo de geometría escalonada	138
4.34. Cuerpo escalonado construido para determinar la sensibilidad del modelo de perspectiva recursivo.	138
4.35. Maniobra de posicionamiento para actualización de parámetros de cámara a fin de asegurar validez de barrido en cuerpo escalonado; a) Maniobra de posicionamiento; b) Ubicación de cuerpo escalonado. . .	139
4.36. a) Nivelación de cuerpo escalonado; b) Barrido con haz láser de cuerpo escalonado y sentido del barrido.	140
4.37. Coordenadas sobre plano “ZY”, de posición estimada tridimensionalmente de puntos sobre cuerpo escalonado; las coordenadas estimadas están dadas con respecto al origen de la base del robot.	140
D.1. Entorno general de la interfaz gráfica	197

Índice de tablas

2.1. Diagrama cinemático de robot Fanuc M16-iB/20T	25
4.1. Resultados experimentales de pruebas de determinación de relación <i>mm/px</i>	113
4.2. Datos de primera secuencia experimental	120
4.3. Datos de primera secuencia experimental.	120
4.4. Datos de segunda secuencia experimental	122
4.5. Resultados condensados de segunda secuencia experimental	122
4.6. Datos de tercera secuencia experimental	124
4.7. Resultados condensados de tercera secuencia experimental	124
4.8. Datos de cuarta secuencia experimental	126
4.9. Resultados condensados de cuarta secuencia experimental	126
4.10. Datos de quinta secuencia experimental	128
4.11. Resultados condensados de quinta secuencia experimental	128
4.12. Datos de sexta secuencia experimental	130
4.13. Resultados condensados de sexta secuencia experimental	130
4.14. Agrupamiento de resultados de secuencias experimentales.	133
4.15. Errores de empate en espacio de cámara de marcas visuales objetivo y marcas visuales artificiales	136
4.16. Análisis del error de elevación promedio de estimación tridimensional de marcas sobre cuerpo escalonado	141

Introducción

La visión, siendo uno de los sentidos más poderosos del ser humano, ha sido estudiado extensamente desde muchos puntos de vista; tales como el anatómico, el neurológico y como sistema óptico, a fin de comprender los procesos relacionados con la capacidad del hombre para reconocer su entorno. Sin embargo, debido a la complejidad de todos los fenómenos involucrados, resulta ser bastante ambicioso comprender a plenitud la forma como el hombre percibe su entorno a través de este sentido.

Una rama reciente de la computación, conocida como visión artificial o visión por computadora, tiene como objetivo analizar, procesar y comprender la información proveniente de imágenes captadas a partir de cámaras de video. Esta rama de la computación ha tenido gran auge dentro del campo de la robótica desde hace ya varios años, gracias al nacimiento del *control por visión de autómatas*, el que, partiendo de información visual del entorno, es capaz de realizar tareas y tomar decisiones.

El presente trabajo de tesis gira en torno a una metodología de control por visión de robots, basada en teoría de estimación, conocida como Manipulación de Espacio de Cámara (*CSM* por sus siglas en inglés). Dicha metodología está conformada de una serie de modelos matemáticos, los cuales describen la correlación existente entre la posición de un objeto en el espacio y su proyección en imagen, posibilitando a un robot aproximarse a un objetivo con precisión. La presente metodología ha sido probada con éxito en robots holonómicos y no-holonómicos, como manipuladores industriales y robots móviles, respectivamente.

En la metodología de *CSM*, se emplean cámaras de video para adquirir información visual del entorno. El comportamiento de cada uno de estos sensores, se describe mediante un modelo de cámara propio, compuesto por una serie de parámetros los cuales, además de definir la posición tridimensional de cada cámara de video,

respecto de un sistema de coordenadas de referencia, también son capaces de predecir la aparición de la proyección de un punto en pantalla, del cual es conocida su posición tridimensional referenciada a dicho sistema de coordenadas.

Dentro de la metodología de *CSM* es posible además, recrear la posición tridimensional de un punto a partir de sus coordenadas en pantalla o espacio de cámara, pero bajo la restricción de ser conocidas dichas coordenadas en el marco de referencia dos o más sensores, cuyo modelo de cámara es conocido. Sin embargo, una sola cámara de video puede ser dotada con la misma funcionalidad de dos o más sensores, al ser acoplada a un sistema de espejos, dando origen así a la generación de una serie de *cámaras u observadores virtuales*.

Los esfuerzos realizados en el presente trabajo de investigación estuvieron encaminados a validar la metodología de *CSM*, haciendo uso de cámaras virtuales como una alternativa, en vez del uso de varios sensores optoelectrónicos de costo generalmente elevado estando de por medio, como restricción a la eficacia del método, la magnitud de la resolución de una sola cámara, además de posibles aberraciones ópticas derivadas del uso de espejos de bajo costo. Como se mostrará en capítulos subsecuentes, pese al cúmulo de factores adversos, fue posible lograr exitosamente una eficacia cercana a la ya alcanzada con los métodos estándares que utilizan múltiples cámaras, que no fueron aquí aplicados.

El presente trabajo de investigación está conformado por cuatro capítulos. Los primeros tres capítulos están estructurados sistemáticamente, partiendo de un panorama global, antecedentes del tema y conceptos generales, hasta tratar con cuestiones involucradas meramente con la metodología *CSM* y el caso particular aquí tratado de *cámaras virtuales*. Así, el primer capítulo de esta tesis es un compendio referenciado de los antecedentes de la metodología de *Manipulación de Espacio de Cámara* haciendo mención además, de algunas otras metodologías de objetivos semejantes basadas en otros enfoques. También se citan, proyectos anteriores de significativa aportación para este proyecto de investigación. Finalmente se definen los propósitos que se persiguen al tratar con cámaras virtuales y el control basado en visión de robots industriales.

El segundo capítulo de tiene, como objetivo, abordar el tema de la metodología de *CSM*. La presentación abarca desde la descripción del modelo de control por visión utilizado, hasta la explicación de los algoritmos necesarios para aprovechar la información devuelta por el modelo y utilizarla eficazmente para la realización de

una tarea de posicionamiento en un manipulador.

Para tratar específicamente el tema de cámaras virtuales, en el tercer capítulo se definen y describen los sistemas de cámaras virtuales; asimismo, se revelan los fundamentos para la selección de un sistema aplicable al método de *CSM*. En adición, se presentan los procedimientos para el procesamiento de imágenes previo y necesario al control por visión utilizado en el presente trabajo.

El cuarto capítulo se centra en exponer pruebas que demuestran la viabilidad del uso de cámaras virtuales en *CSM*, a partir del procesamiento estadístico de los resultados de los experimentos de validación. También, se revela experimentalmente la relación intrínseca que existe entre la precisión del método y la resolución con la que se realiza la adquisición de información por un sensor optoelectrónico.

Finalmente, se sintetizan las conclusiones a las que se ha llegado con el presente trabajo de tesis. Se exponen también propuestas para el que podría ser trabajo futuro de esta investigación.

Una serie de apéndices complementarios hacen mención de algunas herramientas computacionales, utilizadas para materializar los algoritmos de control por visión de manipuladores industriales contemplados en *CSM*, añadiéndose las rutinas computacionales creadas para satisfacer experimentalmente el presente proyecto de investigación; no sin pasar por alto, un manual detallado para el uso de una interfaz gráfica, creada con la finalidad de reproducir los experimentos realizados para validar el uso de cámaras virtuales.

Capítulo 1

Antecedentes y Alcances del Proyecto

La Robótica, como rama multidisciplinaria de la tecnología, constituye un muy amplio y prometedor campo de investigación en constante avance, gracias al sinnúmero de posibles aplicaciones desarrollables; por ejemplo, sistemas robóticos industriales en celdas de manufactura de alta producción, sistemas automáticos de navegación para vehículos, sistemas robóticos quirúrgicos, robots militares, robots humanoides etc. Todas estas innovaciones, encaminadas a sustituir tareas llevadas a cabo por el hombre, conllevan un proceso lento pero firme de integración en la sociedad, motivado por la aparición de necesidades emergentes de simplificación de la vida cotidiana.

Los sistemas robóticos industriales se encuentran constituidos por una serie de elementos eléctricos, electrónicos y mecánicos (manipulador y controlador) que, comandados por un software, dotan de relativa autonomía a un robot para realizar tareas diversas. No obstante, generalmente es un técnico calificado con ciertos conocimientos de robótica, quien se encarga de programar la secuencia de movimientos que deberá de ejecutar el autómata, para que éste cumpla con sus labores. La vía de programación convencional de un robot industrial es una “Terminal de Enseñanza” (*en inglés: Teach Pendant*), conformada de una pantalla (que muestra una interfaz gráfica) y un teclado digital; se utiliza para controlar todas funciones del robot, generar programas con diversos comandos de movimiento y ejecutar operaciones específicas.

La terminal de enseñanza, pese a que es la vía más común de programación de los

manipuladores industriales, está sujeta a limitantes que afectan la simplicidad con la cual el sistema puede ser programado y reprogramado. Algunas de estas barreras son que las interfaces gráficas de las terminales no son del todo intuitivas, además de que un programa realizado por esta vía restringe al robot a movimientos concretos, muy dependientes de un entorno bien definido.

Para dotar de una mayor flexibilidad a un robot industrial y aumentar su autonomía, tanto en términos de programación como de operación, han surgido diversos enfoques de control de robots en el campo de la Robótica. Algunos de estos enfoques de control están basados en la medición de fuerzas o torques ejercidos en un manipulador [7, 8]. Con la información resultante de estas mediciones, un operador, que ejerce una fuerza directamente sobre las juntas de un autómatas o bien sobre un sensor de fuerza colocado en el efector final del mismo, puede manipular manualmente la configuración de dicho artefacto y registrando posiciones sucesivas, puede programar una tarea completa que puede ser ejecutada múltiples veces. Por otro lado, un robot es capaz, a su vez, de afrontar ciertas variaciones del entorno, al medirse las fuerzas resultantes del contacto o colisión con objetos.

Otros enfoques de control, orientados a incrementar la versatilidad de los manipuladores industriales, son aquellos basados en visión por computadora, donde la premisa fundamental es la generación de algoritmos en los cuales, partiendo de información visual proveniente de cámaras de video, se pretende reconocer el espacio de acción del robot. De esta forma, un autómatas programado a realizar una tarea repetitiva, puede enfrentar casos en los cuales sus objetivos cambien de posición entre cada ciclo, o bien mantener la precisión con la cual una tarea es realizada, ejecutando los ajustes necesarios según las posibles variaciones del entorno.

Como ejemplos a estos enfoques de control de robots, basados en visión por computadora, se tienen dos metodologías; una denominada como *Visual Servoing (VS)* y la otra conocida como *Manipulación de Espacio de Cámara (CSM por sus siglas en ingles)*, siendo esta última la utilizada para el presente trabajo de investigación. La metodología de Visual Servoing alude a un sistema de control de lazo cerrado del movimiento o posición de un robot, partiendo de información captada por una cámara [9, 10], siendo fundamental, para preservar la estabilidad del sistema, una alta velocidad de muestreo y cómputo dentro del controlador [11]. Visual Servoing tiene a su vez diversas vertientes, sin embargo, no serán tema de estudio para el presente, trabajo de investigación.

La metodología de *Manipulación de Espacio de Cámara* está conformada por una serie de modelos matemáticos, que describen la correlación existente entre la posición tridimensional de un objeto y su ubicación en espacio de cámara. Los parámetros correspondientes a dichos modelos matemáticos se determinan a partir de muestreos previos a una maniobra de un robot. La información recabada en tales muestreos parte de la observación, mediante cámaras (por lo menos dos), de un objeto colocado sobre un manipulador, que al mismo tiempo, es trasladado a través de una trayectoria bien definida en función de coordenadas del mencionado autómat. Es decir, durante la ejecución de una *trayectoria preplaneada* se registran las coordenadas tridimensionales y en espacio de cámara de dicho objeto. Posteriormente, partiendo de la información previamente registrada, se procede a la parametrización (construcción) de los modelos matemáticos de visión de *CSM*; donde el fin último es la capacidad de controlar un robot para que alcance un objetivo, que ha sido definido únicamente a partir de información en espacio de cámara. Puesto que los modelos de *CSM* tienen únicamente validez local en una región del espacio, dichos modelos son actualizados de forma sucesiva, al tiempo en el que se ejecuta una maniobra.

Dentro de las bondades que ofrece el método de *Manipulación de Espacio de Cámara* destacan, la no dependencia de la velocidad de cómputo para la estabilidad del sistema, el relativamente bajo costo computacional de los modelos *CSM* en contra parte con *VS*, la convergencia de los parámetros pertenecientes a los modelos *CSM* y, principalmente, que este método en particular no requiere de la calibración de todo un sistema de visión en el sentido estricto; sino que se trata de un método libre de calibración, basado en modelos matemáticos sencillos con validez local, cuyos parámetros se actualizan constantemente.

Debido a que la metodología de *Manipulación de Espacio de Cámara* requiere de cuando menos dos vistas distintas en espacio de cámara de una misma región del espacio, habitualmente se han desarrollado sistemas de control de posicionamiento de manipuladores industriales, haciendo uso de por lo menos dos cámaras de video. No obstante, es posible dotar de mayor funcionalidad a una sola cámara de video, proporcionándole la capacidad de captar dos perspectivas diferentes de una mismo objetivo, empleando sistemas de espejos; dando lugar así, a lo que se conoce como *Cámaras Virtuales*, es decir, observadores virtuales generados al desviar el eje óptico de una cámara por medio de espejos. Dado que para llevar a cabo tareas precisas, es intrínseco contar con una relación elevada entre la resolución de cámara (*número de píxeles - px*) y el espacio de trabajo cubierto por dicho sensor (*milímetros*), el uso

de cámaras virtuales podría demeritar la precisión con la cual se llevan a cabo maniobras en un manipulador. Sin embargo, gracias a los avances recientes en sensores digitales de video y a la creciente resolución con la cual estos se encuentran disponibles en el mercado, el uso de sistemas de espejos en lugar de múltiples cámaras, es una alternativa económica tanto en el sentido monetario, como en ahorro de coste computacional por ser menor la cantidad de dispositivos periféricos conectados a un ordenador.

En el presente capítulo se brindará un espectro general de los antecedentes a este trabajo de investigación, además de una síntesis del acervo existente en la materia, gracias a proyectos previos. Finalmente se darán a conocer los alcances y contribuciones que han desprendido los esfuerzos realizados en concretar la presente investigación.

1.1. Antecedentes

La metodología de *Manipulación de Espacio de Cámara* es un enfoque de control por visión aplicable a robots industriales, que surgió en la segunda mitad de la década de los 80's [12]. Dicha metodología esta basada en modelos de cámara a partir de los cuales es posible construir la relación existente entre espacio bidimensional, en coordenadas de imagen, y espacio tridimensional, respecto de un sistema de coordenadas fijo, por ejemplo, en la base de un robot industrial. Algunos de estos modelos son el *modelo de cámara de orificio*, también conocido como *modelo de perspectiva*, en donde se tiene noción de la profundidad a la cual están dispuestos los objetos dentro de una imagen; y el *modelo de cámara ortográfico*, que es una simplificación del modelo anterior, en donde no se tiene noción de la diferencia entre la lejanía a la cual se encuentran los elementos que conforman una imagen [13].

Los diferentes modelos de visión que conforman la metodología *CSM* tienen la capacidad de estimar fácilmente la posición tridimensional de un objeto, a partir de información en espacio de imagen; sin embargo, son necesarias por lo menos dos perspectivas distintas de una misma escena para recrear dicha tarea. El uso de un sistema de espejos acoplado con una sola cámara, puede suplir el uso de múltiples cámaras de video, para captar una misma región del espacio desde ángulos distintos, dando origen así, a observadores virtuales o *cámaras virtuales*, (Figura 1.1).

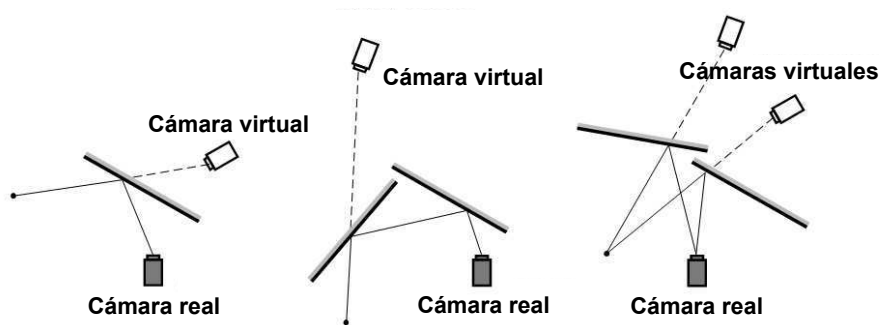


Figura 1.1: Bosquejos de cámaras virtuales con elementos reflexivos planos [1].

La capacidad de un espejo aunado a una cámara de video, de generar lo que se conoce como cámara virtual u observador virtual, se debe a la cualidad de la superficie del espejo de reflejar la totalidad de la luz incidente. En óptica dicha superficie es considerada como reflectora y el comportamiento de la propagación de la luz que incide en ella, es descrito por la *ley de reflexión* [14]. En el presente trabajo de investigación, a los espejos empleados se les denominara también como elementos reflexivos, reflectores o reflejantes.

Aquellos medios a través de los cuales la luz se propaga, son llamados en óptica medios transparentes y pueden ser, como por ejemplo, el vacío, el aire y el vidrio. La colindancia entre las superficies de dos medios transparentes distintos se conoce como interfaz, y el paso de un rayo de luz a través de la interfaz que divide a estos dos medios, da lugar a lo que se conoce como *fenómeno de refracción*. Dicho fenómeno, es descrito por la *ley de refracción*, también llamada *ley de Snell* [14], y a grandes rasgos consiste en el cambio de dirección de un haz de luz que pasa de un medio transparente a otro, donde ambos medios poseen diferentes índices de refracción, la cual es un propiedad inherente de los medios transparentes. Cuando la luz pasa a través de un lente, como por ejemplo el de una lupa, el fenómeno de refracción está presente por lo que a estos elementos se les conoce también como elementos refractivos, refractores o refractantes.

Un *sistema óptico* es un conjunto de superficies que separan medios de distintos índices de refracción [15], a través de los cuales se propaga la luz. Otra definición de un sistema óptico es la disposición diversa, de superficies reflectoras y refractoras, cercanas a una fuente luminosa [14]. Si las superficies son solamente refractantes, el sistema óptico se llama *dióptrico*; si las superficies del sistema son únicamente reflejantes, como lo es la superficie de un espejo, el sistema óptico recibe el nombre de *catóptrico*. Si el sistema óptico esta constituido de superficies tanto refractivas

como reflectivas, el sistema se denomina *catadióptrico*. Es importante señalar que los términos dióptrico y catóptrico, se refieren únicamente a la presencia de elementos refractivos (ej. lentes) y reflectivos (ej. espejos), sin importar la geometría de los mismos.

Como ya fue mencionado, aquellos sistemas ópticos que consisten de elementos refractivos o lentes, y elementos reflexivos o espejos, ya sea planos o bien curvos, son conocidos como sistemas catadióptricos. En el sentido estricto, el uso de cámaras de video y espejos como sistema óptico es por lo tanto, un sistema catadióptrico, ya que las cámaras de video emplean generalmente en su construcción una serie de lentes, para enfocar adecuadamente las imágenes. Si la finalidad de un sistema catadióptrico, es la de generar múltiples observadores virtuales o cámaras virtuales, dicho sistema óptico es denominado también, *sistema de cámaras virtuales* (Figura 1.2).

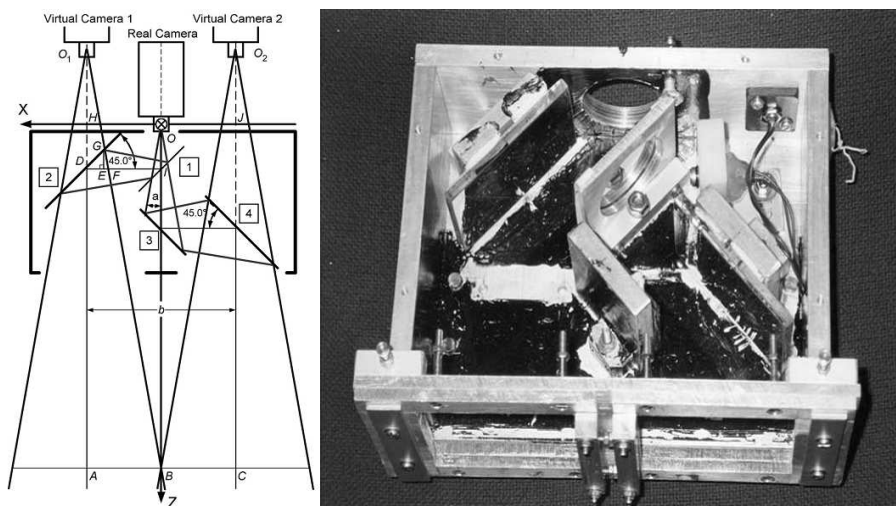


Figura 1.2: Ejemplo de cámaras virtuales en un sistema catadióptrico; a la izquierda el bosquejo del sistema, a la derecha un prototipo [2].

Una aplicación de los sistemas de cámaras virtuales es la de su empleo como sistemas *estereoscópicos*. Los sistemas estereoscópicos son sistemas ópticos que tienen la finalidad de obtener varias vistas distintas de una misma escena de interés simultáneamente. Convencionalmente, estos sistemas emplean múltiples cámaras de video, donde su calibración, posición y alineación es cuidadosamente controlada. Para el evitar el uso de múltiples cámaras, un solo sensor opto-electrónico, acoplado a varios espejos, puede ser dotado de la misma funcionalidad de un sistema estereoscópico con múltiples cámaras, dando origen así a un sistema catadióptrico estereoscópico o simplemente catadióptrico estéreo. Existe en la literatura un amplio

acervo sobre el estudio de sistemas catadióptricos estéreo [16, 17, 2, 1, 18], donde se ofrece un extenso conocimiento sobre el diseño, uso y justificación de tales sistemas. Dentro del potencial de los sistemas estereoscópicos, se encuentra la capacidad de reconstruir *mapas de profundidad* de una escena [19]; es decir, gráficos similares a los mapas de curvas de nivel utilizados en topografía, pero en este caso sobre imágenes e incluso video. A partir de los mapas de profundidad, es posible tener conocimiento preciso de la lejanía hasta el observador, a la cual se encuentran los diferentes objetos contenidos en una imagen, siendo posible diferenciar distancias inferiores a una pulgada y superiores a varias millas [20]. La cantidad de aplicaciones para los mapas de profundidad (Figura 1.3) van desde el desarrollo de sistemas de navegación para robots [21], donde no se requiere demasiada precisión, hasta sistemas más sensibles, útiles por ejemplo, en el control de manipuladores industriales. No obstante, la ejecución de algoritmos computacionales para el cálculo de mapas de profundidad, exige de la calibración del sistema de visión empleado. Puesto que los sistemas estereoscópicos convencionales constan de múltiples cámaras de video, la calibración suele hacerse para cada una de las cámaras involucradas sin embargo, para el caso de cámaras virtuales, en donde exista únicamente una sola cámara real, la calibración suele simplificarse.

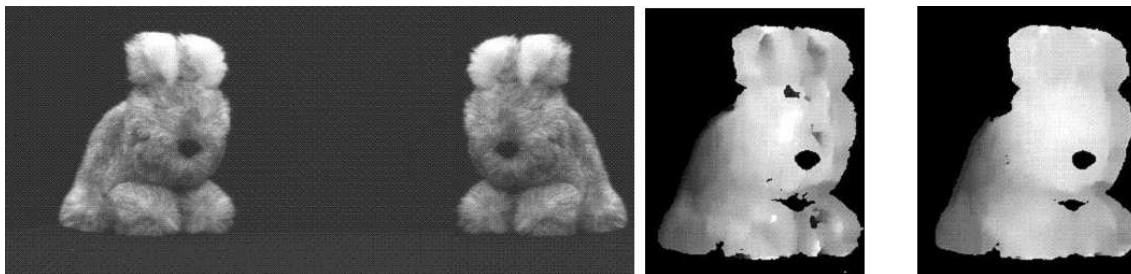


Figura 1.3: Ejemplo de mapas de profundidad; a la izquierda imágenes originales de un sistema estereoscópico, a la derecha mapas de profundidad [1].

Una vertiente más en donde se han empleado sistemas catadióptricos estereoscópicos, ha sido la invención de algoritmos para la reconstrucción tridimensional de objetos sobre una plataforma tipo CAD. Algunos de estos algoritmos se basan en la identificación de la silueta de un objeto [22] y a partir de la rotación del mismo o simplemente con la obtención de varias vistas con diferentes perspectivas de dicho objeto, se logra la generación de un modelo tridimensional. Otros algoritmos se basan en la proyección de un haz de luz estructurado sobre la superficie de un cuerpo, en donde se suele proyectar un patrón ya sea en dos o más colores, conformado por un arreglo de líneas paralelas; cuando dicho haz incide sobre la superficie del

cuerpo, el arreglo de líneas se curva y es captado por varias cámaras. Las imágenes resultantes son procesadas para generar una reconstrucción tridimensional del cuerpo [23]. Finalmente, un enfoque adicional, consiste en capturar un vista perimetral completa de un cuerpo a partir de una secuencia de imágenes, donde la posición relativa de los observadores al objetivo es similar al contorno de un cilindro respecto de su eje. La secuencia de imágenes generada es procesada a partir de algoritmos de empare de imágenes, para posteriormente extraer información que permita generar un reconstrucción tridimensional del cuerpo [24].

La metodología de control por visión *Visual Servoing*, ha arrojado estudios del control de manipuladores industriales, basando la adquisición de imágenes en cámaras virtuales [25]; obteniendo cierto grado de éxito en el guiado de un robot hasta un objetivo, registrando errores en el posicionamiento de entre tres y cinco milímetros. En cuanto a la metodología *CSM*, solo hasta el presente estudio se han tenido antecedentes del uso de cámaras virtuales para la validación del método. Sin embargo, en experimentos de posicionamiento convencionales en los que se utiliza un sistema de adquisición de imágenes de múltiples cámaras, la metodología de *Manipulación de Espacio de Cámara* arroja resultados con una precisión de aproximadamente de un milímetro de error e inferior [26, 27, 6].

1.2. Proyectos Precedentes

El método de *Manipulación de Espacio de Cámara*, desde su invención en la segunda mitad de la década de los 80's, ha sido estudiado ampliamente, arrojando gran cantidad de trabajos en donde se ha demostrado la funcionalidad, efectividad y eficiencia de este enfoque. Desde su creación, *CSM* ha conservado las nociones que la caracterizan, es decir, una metodología libre de calibración basada en un modelo de visión sencillo de validez local, cuyos parámetros son re-estimados constantemente en el transcurso de una maniobra y no existiendo en ningún momento dependencia del tiempo para garantizar la estabilidad del modelo de visión en cuestión [27]. Por otra parte, para el caso de robots industriales, *CSM* aprovecha la extraordinaria repetitibilidad con la que cuentan estos dispositivos, absorbiendo además los errores derivados de la imprecisión que, en contraparte, los caracteriza.

Partiendo de una revisión bibliográfica extensa, hasta el momento *CSM* no cuenta con estudios sobre un sistema de visión que emplee cámaras virtuales en su adqui-

sición. No obstante, a continuación se hará mención de algunos proyectos descritos en la literatura que, basados en *CSM*, demuestran la funcionalidad del método.

Como primer antecedente se tiene la introducción de la metodología de *CSM* en el año de 1987, por S. B. Skaar y W. H. Brockman [12], a través de un artículo titulado como “*Camera Space Manipulation*”. En dicho trabajo se proponen las ideas fundamentales de esta metodología como un enfoque de control por visión libre de calibración, basado en teoría de estimación y en donde las cámaras de video empleadas permanecen inmóviles a lo largo de toda una maniobra de control; siendo necesarias un número de cámaras mínimo de dos ($N_c = 2$). En dicho compendio se expone además el espectro de aplicaciones posibles para *CSM*, que van desde brazos robóticos industriales hasta robots móviles.

La propuesta de un modelo de control para robots no-holonómicos (*descrita en [3]*), como lo son vehículos con ruedas, se materializa unos años más tarde. En este planteamiento, las cámaras de video utilizadas, no se encuentran sobre el vehículo controlado, sino que éstas están fijas sobre un soporte y cubren un determinado campo visual por donde el robot móvil es controlado (Figura 1.4).

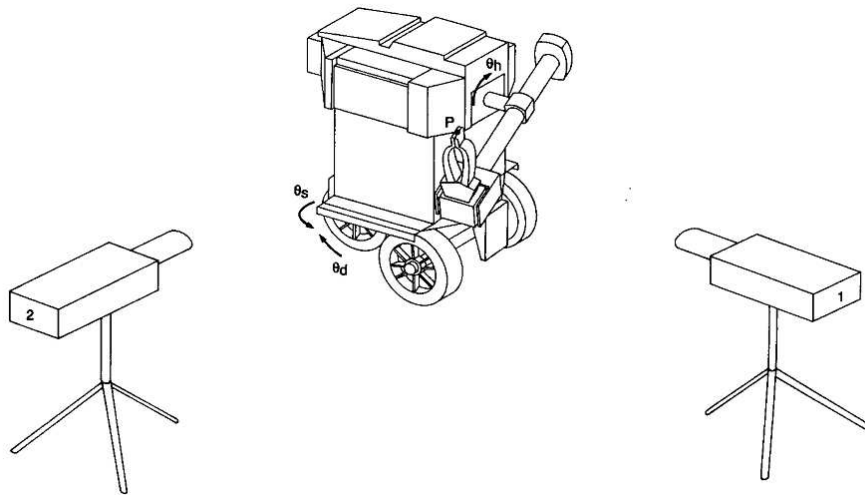


Figura 1.4: Bosquejo de un sistema de control de un robot no-holonómico, empleando cámaras fijas [3].

El primer modelo de visión empleado por *CSM*, fue el conocido como modelo ortográfico de cámara [13]. Dicho modelo es una simplificación del modelo de cámara de perspectiva, al considerar el límite asintótico cuando la distancia a lo largo del

eje óptico de un sensor, desde este mismo hasta un objetivo, es considerablemente grande respecto de la amplitud de la zona de trabajo que contiene a dicho objetivo, con esta medida el efecto de perspectiva, o la noción de profundidad de una imagen, queda suprimida. No obstante las consideraciones tomadas, el modelo ortográfico de cámara al ser empleado en *CSM*, desprende resultados de buena apreciación en cuanto a precisión en el control de manipuladores. Una de las aplicaciones desarrolladas fue la expuesta en [28], donde se presenta el control del brazo robótico de un prototipo para un vehículo autónomo proyectado para hacer exploración en suelo marciano. Otra aplicación desarrollada fue la creación de un sistema automático de corte por plasma (en inglés “*plasma spraying*”) de placas metálicas, controlado por visión [29], capaz de enfrentar retos como lo son las variaciones dimensionales, resultantes de la dilatación térmica del material. Un trabajo más que ejemplifica la funcionalidad del control por visión mediante *CSM* con el modelo ortográfico, es el propuesto para un sistema automatizado de pintura, donde un sistema de visión se emplea para la programación de trayectorias en un robot manipulador que posee una boquilla de automatización para pintura (“*spray painting*”) [30]. Finalmente, una aplicación relativamente reciente del modelo ortográfico en *CSM*, es la de un sistema automatizado de ensamble de componentes electrónicos, siendo relevante el nivel de precisión requerido en la ejecución de tareas de este tipo [31, 4] (Figura 1.5).

El desarrollo del modelo ortográfico de cámara, bajo los preceptos de *CSM*, involucra la estimación de seis parámetros visión por cada cámara involucrada, bajo un proceso de minimización, donde dichos parámetros son agrupados en $2 \times N_c$ expresiones no-lineales descriptivas del modelo de visión mencionado. Para mejorar la precisión y eficiencia de cálculo de este modelo matemático, se introdujeron dos avances significativos en 1996 descritos en [32]. El primero de estos progresos es la agrupación de las expresiones matemáticas para el cálculo de los parámetros de cámara, en matrices de momentos, lo que reduce en gran medida el costo computacional de los algoritmos de *CSM* puesto que, independientemente de la cantidad de información tomada para una estimación determinada, las matrices de momentos al tener naturaleza simétrica y poseer además términos constituidos por sumatorias, evitan el crecimiento de la memoria dedicada al procesamiento requerido por un ordenador, al mantenerse fijas las dimensiones de los términos matriciales implicados. El segundo de estos adelantos es el desarrollo un enfoque alternativo conocido como “*flattening*”, el cual es un proceso correctivo de los parámetros de cámara para hacerlos más consistentes con el modelo ortográfico. A grandes rasgos, la técnica de *flattening* requiere del

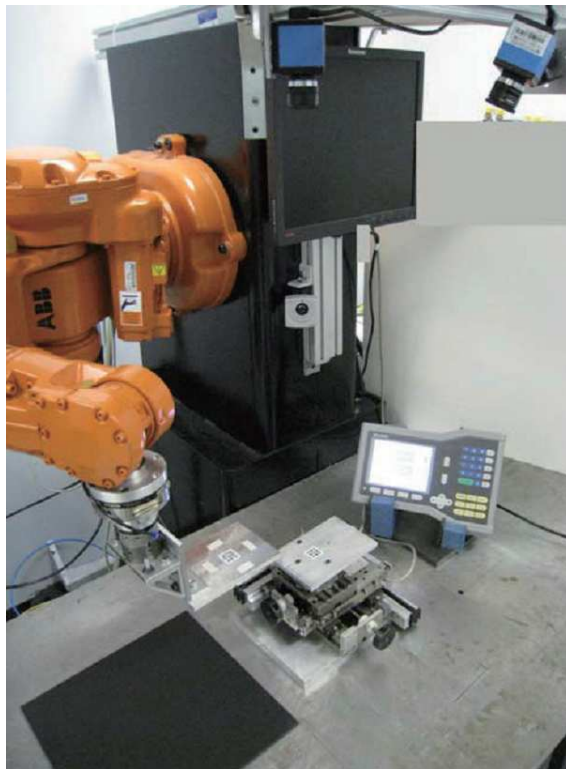


Figura 1.5: Sistema automatizado de ensamble de componentes electrónicos, basado en un control por visión CSM [4].

conocimiento aproximado de la distancia a la cual se encuentra un punto cercano al objetivo de final de maniobra, respecto de los observadores (cámaras); con esta información, se refinan los parámetros de visión del modelo ortográfico de cámara para acrecentar la precisión en el control de un robot.

El desarrollo de tareas de control por visión a través de *CSM* involucra ciertos procesos previos a su ejecución. Estos procesos son la obtención de parámetros iniciales de cámara mediante una trayectoria preplaneada, la asignación de un objetivo de la maniobra y la caracterización de una superficie. Estos últimos dos eventos constituyen avances sustanciales dentro del enfoque de *CSM*, ya que contribuyen en versatilidad con el método. Bajo la premisa de que un operador, con pocos conocimientos de robótica, sea capaz de programar una tarea en un robot industrial, se creó un sistema de selección rápida de objetivos, controlado a través de una interfaz gráfica de usuario (*GUI*). En dicha plataforma el operario, mediante un ratón, designa los objetivos a alcanzar por un robot, en el espacio de imagen de una de las cámaras participantes; la gama de puntos objetivo seleccionables ha sido trazada previamente mediante una base giratoria automática (*en inglés "pan/tilt"*), con un emisor láser,

que proyecta haces de luz en dirección de ciertos blancos de interés [33, 34]. Estos puntos luminiscente son captados por los sensores opto-electrónicos asociados con la escena de interés, completando así el número de perspectivas necesario ($N_c \geq 2$), para la reconstrucción tridimensional de un punto.

La caracterización de una superficie es un proceso necesario para definir la orientación a la cual se desea que arribe el efector final de un robot a un objetivo. Dicho proceso consiste en estimar la posición tridimensional de una serie de marcas visuales artificiales, dispuestas sobre una superficie arbitraria, de modo que la información generada permita hacer una reconstrucción topográfica de la superficie a tratar. Normalmente las marcas visuales empleadas son haces de luz láser, pues tienen la ventaja de no dejar una marca permanente sobre el lugar donde inciden. Puesto que el proceso de caracterización de una superficie requiere de una multiplicidad de marcas artificiales sobre la zona a tratar, un emisor láser de luz estructurada, como por ejemplo, en forma de un arreglo matricial de puntos, es una alternativa para agilizar el proceso de caracterización superficial. No obstante, puesto que es necesario mapear cada una de las marcas visuales captadas, para cada uno de los espacios de cámara presentes (N_c), han sido realizados trabajos anteriores con propuestas a la solución de este problema no trivial [34, 35]; derivando en aportaciones tales como la estimación de la ecuación de la superficie caracterizada y la definición de trayectorias a seguir por un robot industrial sobre superficies arbitrarias.

En el método de *CSM*, el control de un manipulador industrial está limitado por el campo visual cubierto por el conjunto de cámaras empleado; debido a que durante el tiempo en que se efectúa una operación de control, los sensores ópticos deben permanecer fijos, si un manipulador industrial posee un espacio de trabajo mayor al espacio de cámara captado, surge la imposibilidad de aprovechar todo el potencial del autómat controlado. Para atacar este problema, se desarrolló un procedimiento en el cual un par de cámaras colocadas sobre bases giratorias “*pan/tilt*”, logran hacer cobertura de un espacio de trabajo mayor. Dicho procedimiento encuentra una forma de re-estimar los parámetros de cada cámara de video utilizada, para que conserven su validez, tras un redireccionamiento en aras de cubrir un espacio de trabajo distinto [36].

Una variante del método de *Manipulación de Espacio de Cámara* se reporta en 2002, para el control de autómatas móviles. En este enfoque de *CSM*, las cámaras empleadas se encuentran sobre una plataforma móvil o un vehículo, es decir, las

cámaras no permanece inmóviles a lo largo de una tarea de control, sino que se desplazan junto con el dispositivo comandado. Esta aportación ha recibido el nombre de “*Mobile Camera Space Manipulation*” (MCSM), y como aplicaciones a este nuevo enfoque, se enuncian el de la conducción de un prototipo de robot móvil para la exploración del planeta Marte [5] (Figura 1.6), y el desarrollo de un sistema para el control de un montacargas automatizado [37].



Figura 1.6: Prototipo de vehículo de exploración del planeta Marte llamado “FIDO”, controlado mediante MCSM [5].

Una última aportación que será descrita brevemente en este apartado, es la del *Modelo de Perspectiva Recursivo* creado bajo el enfoque de CSM, el cual es el modelo utilizado para el presente trabajo de investigación. Desarrollado a partir del modelo de perspectiva (modelo de cámara de orificio), el modelo de perspectiva recursivo es un modelo de control por visión que, a diferencia del modelo cámara ortográfico, considera el efecto de perspectiva en una imagen. El modelo de perspectiva recursivo está conformado por siete parámetros de visión, siendo el séptimo parámetro adicional, una característica intrínseca de la cámara de video empleada, igual al valor recíproco negativo de su distancia focal. Funcionalmente, el modelo de perspectiva recursivo arroja resultados similares a los obtenidos por el modelo ortográfico de cámara con *flattening*; sin embargo, dicho proceso de *flattening* no es requerido en el modelo de perspectiva recursivo sino que, a cambio, es necesaria únicamente una sola estimación inicial del séptimo parámetro antes de iniciar una tarea de control [26, 6]; proceso que es basado en un ciclo de estimaciones de los seis parámetros restantes, hasta alcanzar un mínimo de error. Otra forma de conocer la magnitud

del séptimo parámetro de visión del modelo de perspectiva recursivo, es haciendo uso de alguno de los métodos de calibración de cámara existentes, como por ejemplo el descrito por Zhengyou Zhang en [38], aunque esto va en contra de la filosofía de *CSM*, como un método libre de calibración, ciertamente el nombre del modelo de perspectiva recursivo justifica el empleo de un recurso adicional de calibración para un único parámetro de visión.

Algunos proyectos partidarios del modelo de perspectiva recursivo son, por ejemplo, la implementación del control por visión en un robot de gran espacio de trabajo, empleando más de dos cámaras de video para cubrir un campo visual extenso [6] (Figura 1.7); y un proyecto en donde, a partir de cámaras sobre bases giratorias *pan/tilt*, se realiza un control efectivo de un manipulador en un campo de acción similarmente extenso [26].



Figura 1.7: Robot Fanuc M16ib/20T; robot de amplio espacio de trabajo, controlado con *CSM* mediante múltiples cámaras [6].

1.3. Objetivos y del Proyecto

El presente trabajo de investigación tiene como objetivo principal demostrar la viabilidad del uso de un sistema de cámaras virtuales como medio de adquisición de imágenes, en la ejecución de una metodología de control por visión de robots, conocida como *Manipulación de Espacio de Cámara (CSM)*; dentro de dicha metodología se utilizará un modelo de visión, denominado *Modelo de Perspectiva Recursivo* [6],

que ha sido ya exitoso en el control de manipuladores industriales.

Para llevar a cabo la presente investigación, se contó con las instalaciones y el equipo disponible dentro del Laboratorio de Robótica, ubicado en las instalaciones de la Facultad de Ingeniería de la Universidad Autónoma de San Luis Potosí.

Puesto que la metodología *CSM* necesita cuando menos un número de dos cámaras de video ($N_c = 2$), dirigidas hacia una misma región de interés para obtener un par de perspectivas distintas de dicha ubicación. El uso de cámaras virtuales se justifica al dotar de mayor funcionalidad a una sola cámara de video, pues de esta forma, una sola imagen obtenida puede contener la información de dos vistas diferentes de un objetivo; las cuales son resultantes de la manipulación mediante espejos del eje óptico del sensor optoelectrónico utilizado.

De entre las ventajas del uso de cámaras virtuales se encuentran las referidas a la disminución del costo computacional debido a la reducción de periféricos conectados a un ordenador, por otra parte, la programación de algoritmos computacionales dedicados a adquisición de imágenes se facilita puesto que no es necesario sincronizar varias cámaras de video. En aquellos modelos de visión no exentos de calibración, el cálculo de los parámetros intrínsecos [38, 39] de cámara se simplifica y todas cámaras virtuales son calibradas simultáneamente con los mismos parámetros puesto que hay un único sensor real [1, 19].

Por mencionar una serie de ventajas más del uso de cámaras virtuales, se enuncia la económica, puesto que al reducir la cantidad de sensores opto-eléctricos en un sistema de visión, eventualmente el costo global de implementación se reduce notablemente, debido al costo generalmente elevado de las cámaras de video digitales con las características deseadas como lo es la alta resolución y la calidad de imagen (bajo ruido entre capturas). Un factor más a favor del uso de cámaras virtuales, es la versatilidad que se les es posible conceder en su diseño, ya que es posible desarrollar sistemas muy compactos, reconfigurables, económicos, duraderos y de bajo costo por mantenimiento.

La utilización de cámaras virtuales, por otro lado, conlleva ciertas restricciones que comprometen la precisión con la cual un sistema de visión es capaz de generar la homografía [38] o el mapeo entre coordenadas de espacio de cámara y coordenadas tridimensionales de los objetos contenidos en un escena. En la metodología *CSM*, la precisión con la que se realiza una maniobra de posicionamiento en un robot,

depende de la relación existente entre las dimensiones del espacio de trabajo cubierto por una cámara y la resolución que el sensor puede ofrecer en sus capturas, es decir, milímetros (mm) lineales cubiertos sobre los correspondientes pixeles (px) en imagen (mm/px). De la relación anterior, se desean valores pequeños. Al utilizar espejos para incrementar el número de perspectivas que una sola cámara puede adquirir, muy posiblemente aumentará la magnitud de la relación (mm/px), lo que influirá con un decremento significativo en la precisión del sistema de visión.

Otras restricciones de la utilización de sistemas catadióptricos se derivan de posibles aberraciones ópticas presentes en los elementos reflectivos (espejos) y refractivos (lentes). Para el caso particular del presente estudio, puesto que los únicos lentes involucrados en el proceso de adquisición de imágenes se encuentran en la cámara de video empleada, se considera que la mayoría de las aberraciones ópticas del sistema proceden de los espejos utilizados; siendo que éstos idealmente se consideran planos, por el contrario, su geometría es en realidad ligeramente irregular. De esto también se puede inferir que, a mayor cantidad de espejos utilizados para manipular el eje focal de una cámara, aumentará el número de aberraciones ópticas.

A pesar de las limitantes del uso de cámaras virtuales, existen suficientes argumentos a favor para incursionar en su estudio, empleando la metodología de *Manipulación de Espacio de Cámara*. A continuación se enuncian los objetivos complementarios de esta investigación:

- Diseño y construcción de un sistema de visión basado en cámaras virtuales.
- Puesta en operación del sistema de visión concebido, empleando la metodología de *Manipulación de Espacio de Cámara*.
- Cuantificación del error obtenido en pruebas de posicionamiento utilizando como dispositivo controlado, un manipulador industrial de seis grados de libertad, a fin de justificar la viabilidad del sistema.
- Cuantificación de la sensibilidad del sistema de visión, a los cambios de relieve sobre una superficie.
- Programación de todos los algoritmos necesarios para la implementación del sistema de cámaras virtuales y la respectiva ejecución de pruebas experimentales.

- Desarrollo de una interfaz gráfica para hacer más amigable el control por visión del manipulador, así como las pruebas experimentales.

Capítulo 2

El Método de Manipulación de Espacio de Cámara

La metodología de *Manipulación de Espacio de Cámara (CSM)*, es un enfoque de control por visión de robots, libre de calibración; basado en la teoría de estimación. Dicha metodología surge en la segunda mitad de la década de los 80's [12], como una alternativa a la metodología de control por visión de lazo cerrado conocida como *Visual Servoing*; basada en la retroalimentación del *error visual* y en el cálculo de un jacobiano, que correlaciona la configuración interna de un robot con la información procedente del espacio de cámara [11]. La metodología de *Manipulación de Espacio de Cámara* ha demostrado ser un enfoque robusto y funcional para el control de autómatas a un costo computacional relativamente bajo.

A grandes rasgos, el control por visión de robots mediante de *Manipulación de Espacio de Cámara*, esta basado en la parametrización o reconstrucción de modelos matemáticos que tratan de describir la correlación existente, entre la coordenadas tridimensionales $\{X_i, Y_i, Z_i\}$ de una serie de puntos ($\{i = 1..n\}$), con la respectiva proyección de cada uno de los puntos $\{X_c^i, Y_c^i\}$, en algún espacio de cámara “ c ”. Dicha correlación o mapeo de coordenadas, llamado también “*homografía*” [38], puede ser realizado de forma bidireccional, siempre y cuando, el número de cámaras utilizado sea mayor o igual a dos; $\{X_c^i, Y_c^i\} \Leftrightarrow \{X_i, Y_i, Z_i\}$ si y solo si $c = 1..Nc$ donde $Nc \geq 2$.

A continuación se presenta un bosquejo representativo de un sistema de control por visión de un brazo robótico industrial, mediante el método de *Manipulación de*

Espacio de Cámara:

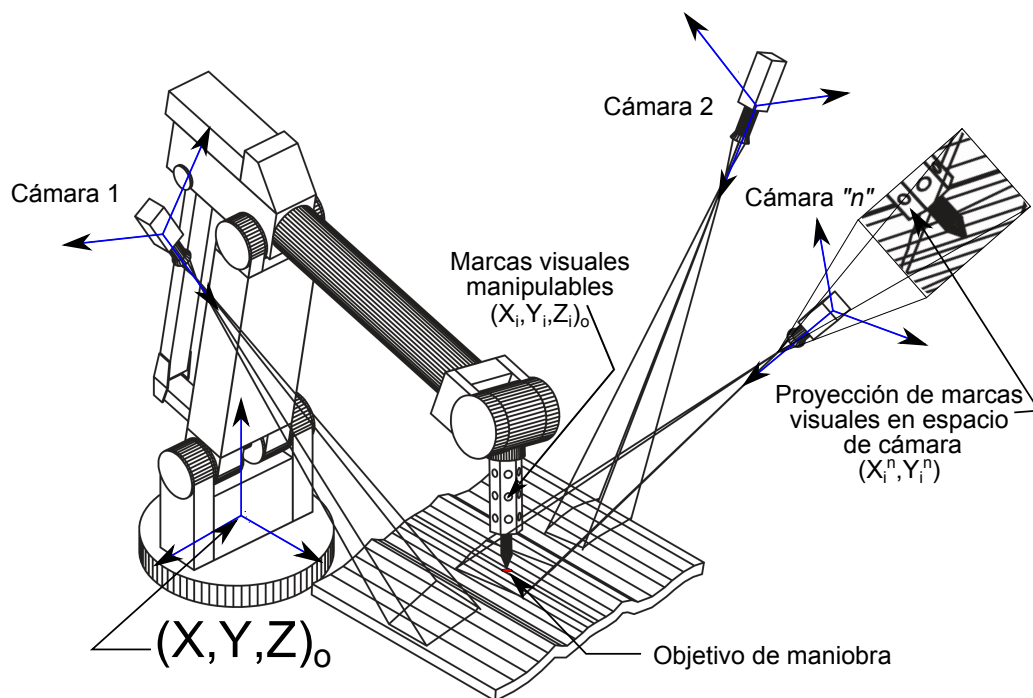


Figura 2.1: Esquema representativo de un sistema de control por visión basado en CSM

Antes de entrar en materia con la metodología CSM, se enuncian a continuación, una serie de características fundamentales:

- El método de *Manipulación de Espacio de Cámara* es un método libre de calibración, puesto que no es necesario el uso de un procedimiento estricto de calibración al inicio de una tarea de control de un robot; ya que los modelos de visión empleados en CSM son por naturaleza simples y no se requieren del cálculo de parámetros de cámara tales como la distorsión radial y tangencial en lentes [39]. Tampoco es necesario tener conocimiento de la ubicación espacial de las cámaras respecto de algún sistema de referencia.
- La metodología de control por visión de CSM está basada en teoría de estimación, siendo un método de cálculo, el método de estimación por mínimos cuadrados.
- Los modelos de visión empleados en CSM, son modelos matemáticos de validez local, cuyos parámetros son solo funcionales para realizar mapeos de puntos en determinada región del espacio físico y el espacio de cámara.

- Los parámetros de cámara incluidos en los modelos de visión empleados en *CSM*, son estimados inicialmente antes de una tarea de control de un robot, partiendo del muestreo de marcas en espacio de cámara, que además, son de posición tridimensional conocida y asociada a un sistema de coordenadas (en manipuladores industriales, normalmente en la base del robot); dicho proceso es conocido como “*Trayectoria Preplaneada*”.
- Durante el desplazamiento de un robot sometido a una tarea de control mediante *CSM*, los parámetros del modelo de visión utilizado en cuestión, son re-estimados constantemente a partir de información visual y física de marcas de referencia, geométricamente definidas y adheridas al robot controlado. Esto hecho con la finalidad de asegurar la validez local del modelo de visión empleado.
- El número mínimo de cámaras (N_c) requerido por la metodología *CSM* es de dos; aunque esta restricción se atenúa cuando se conoce la geometría del cuerpo no-manipulable u objetivo de maniobra [40].
- Durante toda la maniobra de control de un robot, las cámaras relacionadas con la escena de interés deben permanecer fijas; a menos de que se emplee la metodología derivada de *CSM* conocida como *MCSM* [5], o que se empleen bases giratorias *pan/tilt*, con sensores que permitan cuantificar ángulos de rotación y seguir el procedimiento descrito en [36].

La solución a los modelos de control por visión tratados por la metodología *CSM*, involucran la estimación de una serie de parámetros, asociados a cada una de las cámaras o sensores, dedicados a realizar la cobertura visual de determinada región del espacio de trabajo de un robot. En todo proceso de estimación, para determinar el valor de los parámetros pertenecientes a un modelo físico, hace falta conocer previamente un cúmulo de información resultante de mediciones realizadas al fenómeno físico que trata de ser descrito. Las mediciones realizadas al fenómeno en cuestión, son magnitudes físicas que guardan una relación intrínseca entre sí, de modo que al haber un nexo de causalidad, la propuesta de un modelo matemático descriptivo del proceso, es factible.

El muestreo previo a la parametrización de los modelos *CSM*, ocurre durante la ejecución de lo que se conoce como una *trayectoria preplaneada*. Dicho evento está constituido por una secuencia de movimientos pre-programados a realizar por el robot que será controlado más tarde por visión. El robot en cuestión, ya sea no-holonómico o del tipo holonómico, posee una o un conjunto de marcas o distintivos visuales, de

posición geoméricamente definida respecto de un sistema de coordenadas, los cuales sirven como elementos de referencia para un número de sensores opto-electrónicos. El robot, al proceder con la secuencia de movimientos que se le ha ordenado, lleva consigo en su recorrido a las marcas visuales de las que ha sido provisto. Dichas marcas son captadas, una a la vez, por las cámaras asociadas a la subsiguiente maniobra de control por visión; al realizarse la captura, las imágenes resultantes son procesadas y la ubicación de la proyección en espacio de cámara, de las referencias visuales, es entonces registrada. Puesto que la localización física de cada marca en cuestión también es conocida; ambos conjuntos de datos, tanto las coordenadas en imagen, como coordenadas físicas, son almacenados manteniendo afinidad de acuerdo al orden de detección.

En el presente trabajo de investigación, se ha utilizado un robot industrial del tipo antropomórfico; marca Fanuc, modelo M16-iB/20T. Éste posee seis grados de libertad, de los cuales, uno de ellos es resultante de una junta cinemática prismática y los cinco restantes, proceden de juntas cinemáticas de revoluta. A cada par o junta cinemática, según su naturaleza, se le puede asignar una determinada magnitud angular o lineal, que influye directamente en la configuración cinemática que habrá de adoptar el manipulador industrial, entendiéndose por configuración a la posición y orientación del efector final o herramienta del autómatas en cuestión. El modelo matemático que describe la relación existente entre las magnitudes de junta de un manipulador y la configuración cinemática del mismo, recibe comúnmente el nombre de “*cinemática directa*”. En la sección siguiente, se expondrá la cinemática directa del robot industrial, que para los fines del presente estudio fue utilizado, además de una breve explicación de su obtención y desarrollo.

2.1. Cinemática Directa del Manipulador

Para los fines del presente de investigación, se ha utilizado un robot industrial de la marca *Fanuc*, modelo *M16-ib/20T*, el cual posee una configuración del tipo antropomórfica [13], es decir, análoga a la forma de un brazo humano. Los robots industriales de este tipo están constituidos de una serie de elementos mecánicos rígidos llamados eslabones, los cuales se encuentran unidos entre sí, mediante lo que se conoce como juntas o pares cinemáticos. Dichas uniones permiten algún tipo de movimiento relativo entre eslabones adyacentes, y en conjunto, pares cinemáticos y

eslabonamientos conforman lo que se denomina “cadena cinemática”.

Existen dos tipos de cadenas cinemáticas; si el número de eslabonamientos pertenecientes a la cadena, logran en algún punto cerrar un lazo, en el cual el eslabón “final” queda conectado al “inicial” mediante algún tipo de junta, la cadena cinemática es *cerrada*. De forma contraria, una cadena cinemática cuyos eslabones componen un lazo abierto, recibe el nombre de cadena cinemática “abierta”. El manipulador industrial con el que se cuenta, entra dentro de esta última clasificación. Cabe destacar que, en los robots industriales de cadena cinemática abierta, el primer eslabón de la cadena cinemática es fijo y dicho eslabón lo constituye la base del mismo manipulador.

Según el tipo de movimiento relativo permisible por una junta entre dos eslabones, los pares cinemáticos atienden cierta clasificación. Para efectos del presente trabajo de investigación, los únicos pares cinemáticos mencionados serán los llamados de primer orden, que son el *prismático* y el de *revoluta*.

Un atributo inherente de toda cadena cinemática es el número de grados de libertad, es decir, la cantidad de posibles movimientos realizables por los eslabones que integran la cadena. El robot industrial con el cual se cuenta, posee seis grados de libertad, distribuidos en seis juntas de las cuales, una es del tipo prismática y las cinco juntas restantes son del tipo de revoluta. A partir del conocimiento de la geometría robot, se puede generar lo que se conoce como un *diagrama cinemático*, es decir, un gráfico en el cual se muestra la secuencia de eslabonamientos de una cadena cinemática, atendiendo a la descripción del tipo de cada par cinemático. A continuación se presenta el diagrama cinemático del robot industrial utilizado:

Eslabón	Segmento	Dimensión (mm)
1	J1	variable
2	a	770
3	b	100
3	c	740
4	d	100

Tabla 2.1: Elementos del diagrama cinemático del robot industrial Fanuc M16-iB/20T

La construcción del modelo matemático conocido como cinemática directa de un

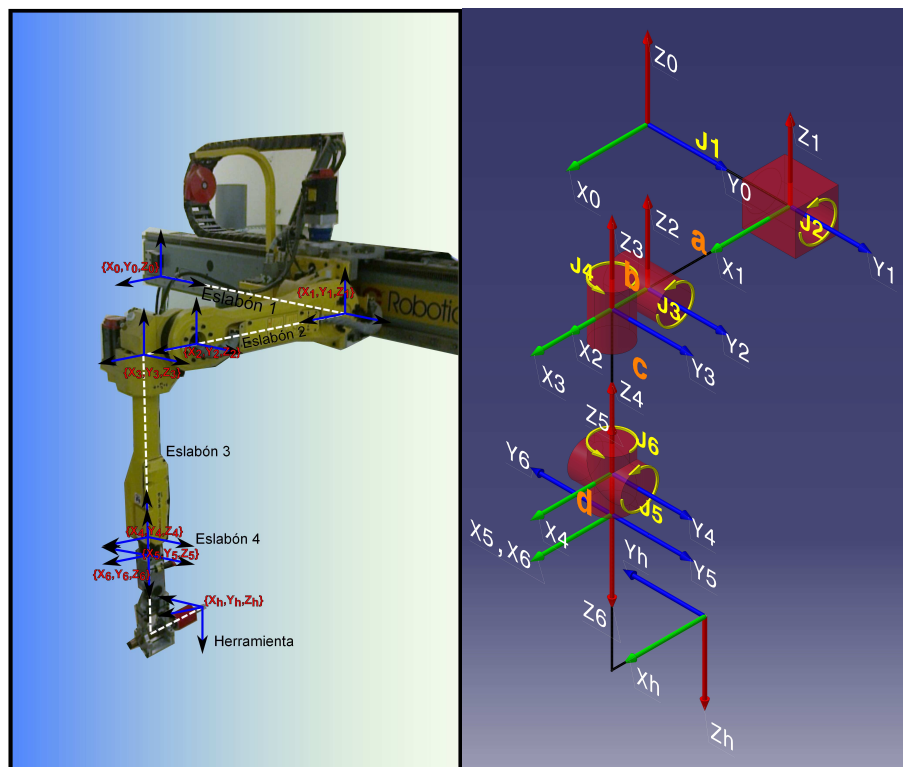


Figura 2.2: Diagrama cinemático de robot industrial Fanuc M16-iB/20T

robot, relaciona los valores asociados a las magnitudes de cada par cinemático, con la ubicación y orientación física del efector final del autómatas en cuestión, con base a la definición previa de un sistema coordenado de referencia. Para generar la correlación entre la configuración interna de un manipulador con la respectiva localización de su efector final o herramienta, se definen una serie de matrices de transformación homogéneas, de rotación-traslación, en donde se representa tanto la geometría del manipulador como su configuración interna. Efectuando una secuencia de productos sucesivos entre cada una de las matrices de transformación definidas, se obtiene una matriz resultante que contiene toda la información concerniente a la posición y orientación de la herramienta de robot.

La definición de la cinemática directa del manipulador se realiza a partir de matrices rotación-traslación respecto de un sistema actual de coordenadas [13]. Este proceso consiste de una secuencia de productos entre matrices homogéneas de rotación y traslación. La obtención de la cinemática directa de un manipulador es semejante a definir un punto sobre el extremo de la herramienta del manipulador, considerando su origen de coordenadas como el centro de la placa donde dicha herramienta es adherida al robot; y posteriormente trasladar el origen de coordenadas mencionado

hasta la base fija del autómeta.

Al tratar con un sistema de coordenadas cartesiano, las matrices de rotación y traslación homogéneas se definen de modo tal que una rotación o traslación se realiza alrededor o a lo largo de uno de los ejes del sistema coordenado cartesiano respectivamente. Si los ejes del sistema coordenado cartesiano (“↑”) son llamados “ X, Y, Z ”, las matrices de rotación (“ $R_{\uparrow, \star}$ ”) y traslación (“ $T_{\uparrow, \star}$ ”) homogéneas, se definen como:

$$\mathbf{R}_x, \theta = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{R}_y, \theta = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}_z, \theta = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}_x, a = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{T}_y, b = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{T}_z, c = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

De acuerdo con el diagrama cinemático de la Figura 2.2, puede deducirse la cinemática directa del robot Fanuc M16-iB/20T empleado en este proyecto, es deducible fácilmente. Esto se consigue a través de seis matrices de traslación y de cinco matrices de rotación. Es importante señalar que una característica relevante de los robots de la marca Fanuc, es un efecto llamado “*paralelogramo*”, derivado del modo con el cual los controladores de esta marca de robots dirigen las juntas de estos manipuladores. Dicho efecto será explicado en breve. A continuación se presentará la definición de cada una de las matrices homogéneas que constituyen la cinemática directa del robot. La primera matriz de transformación se deduce a partir de una traslación por sobre el eje “ Y_0 ” del sistema coordenado ubicado en el origen del manipulador (X_0, Y_0, Z_0). La traslación resultante es igual al valor de “ $J1$ ”, o bien, de la junta prismática número uno (Ec. 2.1) y da origen al sistema de coordenadas uno (X_1, Y_1, Z_1).

$$\mathbf{H}_{0,1} = \mathbf{T}_{y,J1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & J1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

El efecto de *paralelogramo* resulta de la forma con la cual, el marco de referencia número dos, es definido. Dicho efecto consiste en que, no importando el valor de la junta “ $J2$ ”, el eslabón número tres del robot conservará su orientación respecto del sistema coordenado uno (X_1, Y_1, Z_1) (Figura 2.3). Partiendo del diagrama cinemático del manipulador (Figura 2.2), el sistema coordenado dos (X_2, Y_2, Z_2) es determinado a partir de una sola traslación sobre el plano Z_1, X_1 . Dicha traslación depende de la longitud “ a ” del eslabón 2 del manipulador y del valor de la junta “ $J2$ ”.

$$\mathbf{H}_{1,2} = \mathbf{T}_{zx,J2} = \begin{bmatrix} 1 & 0 & 0 & a \cdot \cos(J2) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & a \cdot \sin(J2) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

El sistema coordenado número tres se obtiene a partir de una traslación a lo largo de segmento “ b ” del diagrama cinemático (Figura 2.2) y de una rotación “ $J3$ ”, alrededor del eje “ $Y2$ ” del sistema coordenado dos (X_2, Y_2, Z_2). Debido a que el sistema coordenado 2 posee la misma orientación del sistema coordenado 1 y 0, se podría argumentar que la rotación “ $J3$ ”, se realiza con respecto de alguno de estos dos marcos de referencia.

$$\mathbf{T}_{zx,b} = \begin{bmatrix} 1 & 0 & 0 & b \cdot \cos(J3) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & b \cdot \sin(J3) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

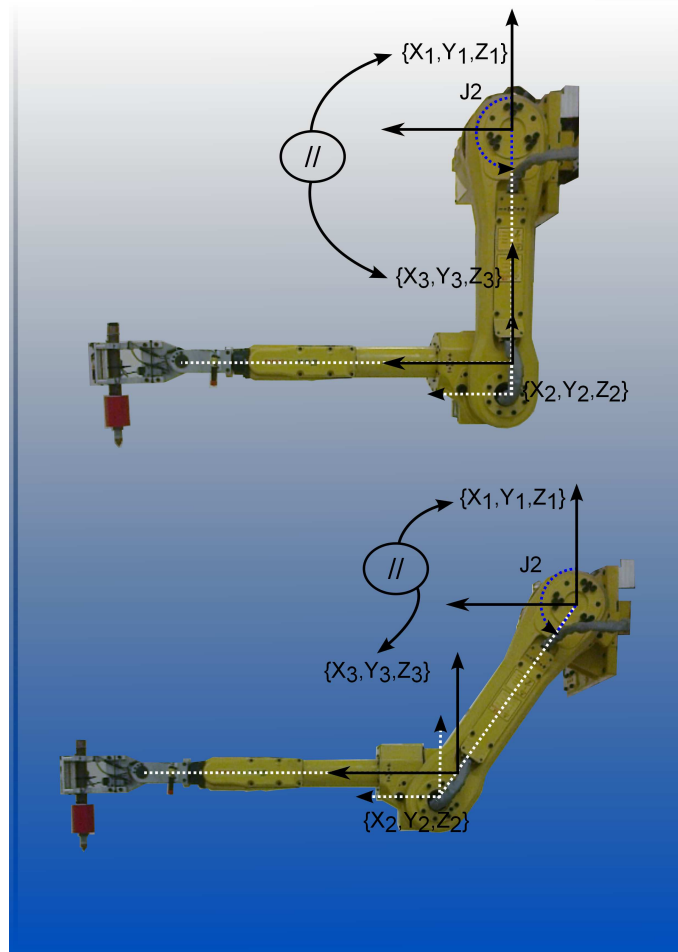


Figura 2.3: Representación del efecto de paralelogramo en el robot industrial Fanuc M16-iB/20T.

$$\mathbf{R}_{y,J3} = \begin{bmatrix} \cos(J3) & 0 & \sin(J3) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(J3) & 0 & \cos(J3) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{H}_{2,3} = \mathbf{T}_{zx,b} * \mathbf{R}_{y,J3} \quad (2.3)$$

La cuarta matriz de transformación homogénea de la cinemática directa del robot, se obtiene de multiplicar una matriz de rotación seguida de una de traslación. La rotación se realiza alrededor del eje “Z” del sistema coordinado tres del diagrama cinemático, con valor igual de la junta “J4” y la traslación a lo largo de este mismo eje en una magnitud igual al valor negativo del segmento “c”.

$$\mathbf{R}_{z,J4} = \begin{bmatrix} \cos(J4) & -\sin(J4) & 0 & 0 \\ \sin(J4) & \cos(J4) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}_{z,-c} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{H}_{3,4} = \mathbf{R}_{z,J4} * \mathbf{T}_{z,c} \quad (2.4)$$

La quinta matriz de transformación homogénea perteneciente a la cinemática directa del manipulador se logra mediante una rotación de magnitud “ $J5$ ” alrededor del eje “ $Y4$ ”, seguida de una traslación igual al valor negativo del segmento “ d ” sobre el eje “ Z ”, dando lugar al quinto sistema coordenado X_5, Y_5, Z_5 del diagrama cinemático (Figura 2.2).

$$\mathbf{R}_{y,J5} = \begin{bmatrix} \cos(J5) & 0 & \sin(J5) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(J4) & 0 & \cos(J5) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}_{z,-d} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{H}_{4,5} = \mathbf{R}_{y,J5} * \mathbf{T}_{z,-d} \quad (2.5)$$

Una sexta matriz de transformación tiene la función de generar la rotación de la última junta del manipulador (“ $J6$ ”), además de adecuar la orientación del eje coordenado en la punta de la herramienta. La siguiente matriz de transformación

homogénea se calcula a partir de dos matrices de rotación y da origen al sistema coordinado seis (X_6, Y_6, Z_6) .

$$\mathbf{R}_{z,J6} = \begin{bmatrix} \cos(J6) & -\sin(J6) & 0 & 0 \\ \sin(J6) & \cos(J6) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}_{x,\pi} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\pi) & -\sin(\pi) & 0 \\ 0 & \sin(\pi) & \cos(\pi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{H}_{5,6} = \mathbf{R}_{z,J6} * \mathbf{R}_{x,\pi} \quad (2.6)$$

Una última matriz de traslación representa las dimensiones de la herramienta adherida al manipulador, llevando al último sistema coordinado (X_h, Y_h, Z_h) , hasta su punta.

$$\mathbf{H}_{6,7} = \mathbf{T}_h = \begin{bmatrix} 1 & 0 & 0 & h_x \\ 0 & 1 & 0 & h_y \\ 0 & 0 & 1 & h_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

Finalmente, la cinemática directa del manipulador se define a través de la siguiente expresión que involucra a todas las matrices de transformación descritas hace un momento (Ecs. 2.1..2.7).

$$\mathbf{H}(\Theta) = \mathbf{H}_{0,1} * \mathbf{H}_{1,2} * \mathbf{H}_{2,3} * \mathbf{H}_{3,4} * \mathbf{H}_{4,5} * \mathbf{H}_{5,6} * \mathbf{H}_{6,7} \quad (2.8)$$

A partir de la ecuación 2.8, se obtiene una matriz $\mathbf{H}(\Theta)$ dependiente de la magnitud del valor de las juntas del manipulador ($\Theta = \{J1 \dots J6\}$), de dimensiones 4×4 , cuyos elementos contienen toda la información concerniente a la posición y orientación de

la herramienta del robot. La posición de la punta de la herramienta del manipulador está definida mediante los primeros tres elementos de la cuarta columna de la matriz H , donde cada elemento representa las componentes de un vector de posición (\vec{R}_{xyz}) referenciado al origen del manipulador:

$$\vec{r}_{xyz} = \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} = \begin{bmatrix} H_{1,4} \\ H_{2,4} \\ H_{3,4} \end{bmatrix} \quad (2.9)$$

La orientación de la herramienta del robot se define de igual forma con respecto al origen de coordenadas del mismo, mediante una secuencia rotaciones alrededor de cada uno de los ejes de dicho sistema *fijo* de coordenadas. Al igualar los términos de la matriz \mathbf{H} a los términos de una matriz \mathbf{R} de rotación, compuesta de tres rotaciones sucesivas alrededor de los ejes X_0 - Y_0 - Z_0 , del origen fijo de coordenadas del robot, es decir:

$$\mathbf{H}(\Theta)_{i,j} = \mathbf{R}_{i,j}$$

donde:

$$\mathbf{R} = R_{Z_0, R \circlearrowleft} * R_{Y_0, P \circlearrowleft} * R_{X_0, W \circlearrowleft}$$

Los ángulos de rotación resultantes reciben el nombre de ángulo de oscilación ($W \circlearrowleft$), cabeceo ($P \circlearrowleft$) y balanceo ($R \circlearrowleft$), siendo alrededor de los ejes X_0 , Y_0 y Z_0 respectivamente y su definición se muestra a continuación.

$$W \circlearrowleft = \text{atan2}(H_{3,2}, H_{3,3}) \quad (2.10)$$

$$P \circlearrowleft = \text{atan2}(-H_{3,1}, \sqrt{H_{3,2}^2 + H_{3,3}^2}) \quad (2.11)$$

$$R \circlearrowleft = \text{atan2}(H_{2,1}, H_{1,1}) \quad (2.12)$$

2.2. Modelo de Cámara de Perspectiva

Los sistemas de visión por computadora tienen la finalidad de procesar, interpretar y dar respuesta a la información visual procedente de cámaras de video. Dependiendo del tipo de aplicación, deriva el tipo de respuesta del sistema; y el procesamiento de la información contenida en imágenes, consiste de su extracción y abstracción selectiva, ya que una sola imagen posee una enorme cantidad de información procedente de la captura de una escena. Para el caso de robots industriales, los sistemas de visión por computadora son capaces de dotarles de flexibilidad y cierto nivel de inteligencia, encaminada a tomar decisión atendiendo a las características del entorno.

Un sistema de visión por computadora aplicado de manera práctica consta de ciertas etapas en cuanto al manejo de información procedente de uno o bien un conjunto de sensores opto-electrónicos (Figura 2.4).

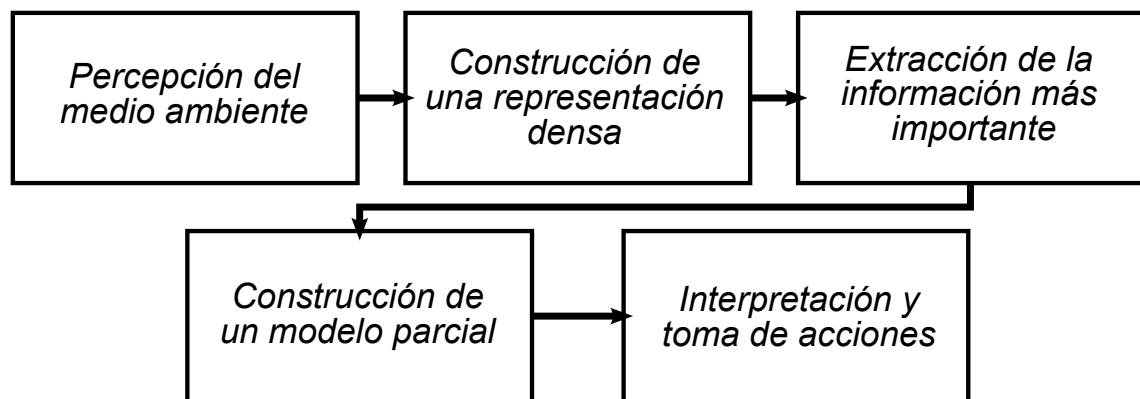


Figura 2.4: Esquema de un sistema de visión por computadora, aplicado de forma práctica.

Los modelos de cámara o de visión son modelos matemáticos cuyo propósito es generar la correlación o mapeo, entre espacio físico tridimensional y espacio de cámara o de imagen. El modelo de cámara de perspectiva, también llamado, modelo de cámara de orificio, es un modelo de visión capaz de generar un mapeo preciso entre las coordenadas tridimensionales de un punto X_i, Y_i, Z_i y las coordenadas de dicho punto en espacio de imagen x_{ci}, y_{ci} , a través de una proyección de perspectiva [13]. Dicho mapeo se modela de la siguiente manera:

$$x_{c,i} = f \frac{X_i}{Z_i}, \quad (2.13a)$$

$$y_{c,i} = f \frac{Y_i}{Z_i} \quad (2.13b)$$

Donde f es igual al valor de la distancia focal de la cámara modelada, dato cuyo valor es intrínseco del diseño del sensor y de la lente o conjunto de lentes empleadas para su construcción.

Las coordenadas del punto en el espacio X_i, Y_i, Z_i , pueden estar referidas a su vez a un sistema de coordenadas arbitrario, mediante la siguiente transformación que involucra matriz de rotación arbitraria, expresada en cuaterniones de Euler:

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} = \begin{bmatrix} e_1^2 + e_2^2 - e_3^2 - e_4^2 & 2(e_2e_3 + e_1e_4) & 2(e_2e_4 - e_1e_3) & X_0 \\ 2(e_2e_3 - e_1e_4) & e_1^2 - e_2^2 + e_3^2 - e_4^2 & 2(e_3e_4 + e_1e_2) & Y_0 \\ 2(e_2e_4 + e_1e_3) & 2(e_3e_4 - e_1e_2) & e_1^2 - e_2^2 - e_3^2 + e_4^2 & Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \quad (2.14)$$

Donde como condición necesaria se tiene:

$$e_1^2 + e_2^2 + e_3^2 + e_4^2 = 1$$

El modelo de perspectiva se representa esquemáticamente a continuación (Figura 2.5), ejemplificando la proyección de un punto en espacio de cámara que posee coordenadas tridimensionales físicas referidas tanto a un sistema de coordenadas arbitrario, como al sistema de coordenadas propio del sensor opto-electrónico de captura.

Es importante aclarar que el modelo de cámara de orificio es un modelo sencillo, en donde se idealiza la existencia de lentes perfectas en la cámara; es decir, se asume que no existen aberraciones ópticas. Existen modelos de cámara más complejos, que parten del modelo de cámara de orificio [38, 39], en los que si se considera la presencia de aberraciones ópticas en las lentes de las cámaras, que son modeladas

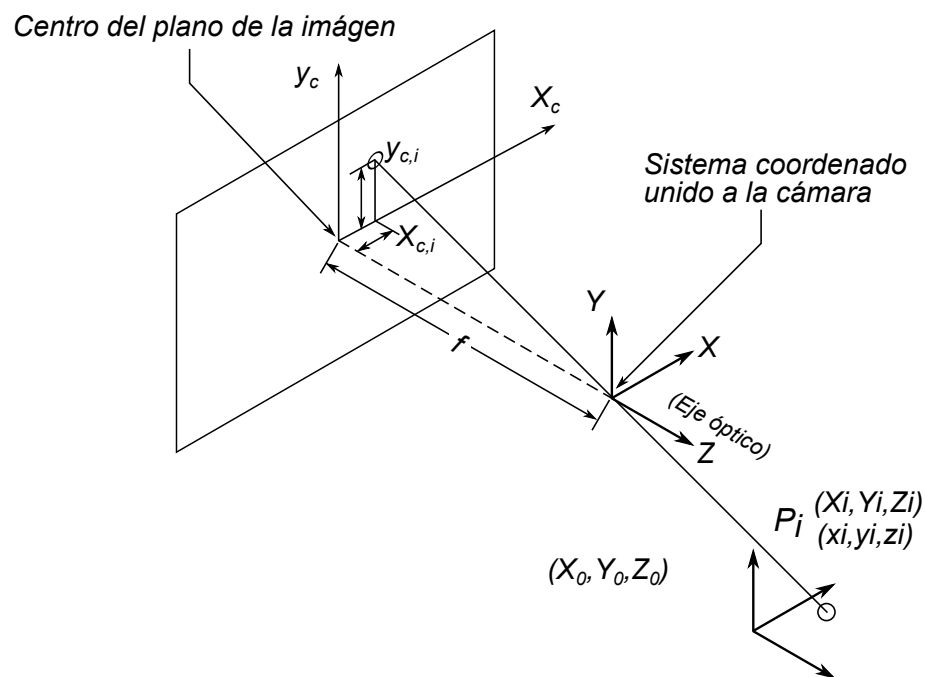


Figura 2.5: Esquema de proyección de perspectiva modelada a través de modelo de cámara de orificio.

como distorsiones radiales y tangenciales de una imagen. La aplicación de estos modelos más complejos es usual en sistemas de visión basados en calibración.

2.3. Modelo de Cámara de Perspectiva Recursivo

Dentro de la metodología *CSM*, han sido desarrollados varios modelos de control de robots industriales por visión, cuyo desempeño en pruebas experimentales ha sido exitoso. Concretamente, las tareas designadas a manipuladores industriales a partir del control por visión *CSM*, se basan en el posicionamiento del efector final de estos dispositivos, desde una localización inicial, hasta un objetivo designado en espacio de cámara, teniendo en todo momento dominio del modo con el cual se realiza la aproximación del robot a su objetivo; es decir, se es capaz de controlar el trayecto y la orientación del efector final del autómatas. Así pues, por medio del control por visión de *CSM*, se posibilita la ejecución de movimientos para la consumación de tareas no triviales por un robot industrial; en donde el control del efector final del robot puede involucrar el uso de una herramienta para un fin determinado, como la manipulación o modificación de un objeto. Por mencionar un ejemplo, sería la puesta del rin de

un neumático en el eje de un automóvil, que involucra la coincidencia de una serie de pernos con los barrenos de un rin.

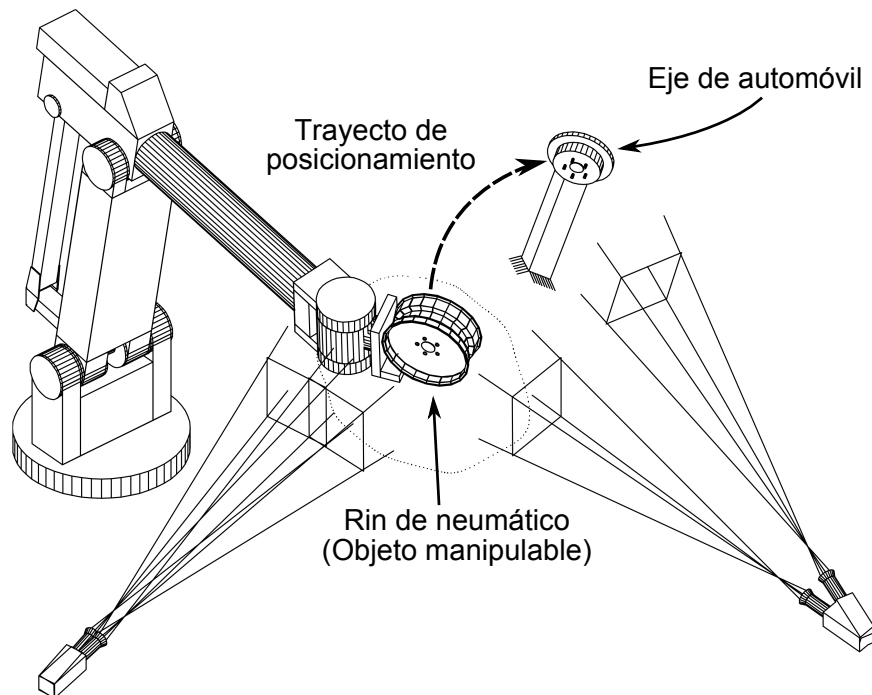


Figura 2.6: Esquema de tarea de posicionamiento de un manipulador industrial en donde se representa la colocación del rin de un neumático en el eje de un automóvil.

Concretamente dentro de la metodología *CSM*, en cuanto al control de manipuladores industriales, existen cuatro modelos de control por visión desarrollados hasta el momento; estos modelos son el modelo de cámara ortográfica, el modelo de cámara ortográfica con corrección por *flattening*, el modelo lineal de cámara y el modelo de perspectiva recursivo. Los primeros tres modelos son descritos en las siguientes citas bibliográficas [29, 33, 41] y corresponde a la presente sección la descripción del modelo de perspectiva recursivo.

Aplicando la matriz de rotación arbitraria descrita en la ecuación 2.14, al modelo de cámara de orificio descrito en las ecuaciones 2.13a y 2.13b, se obtiene la siguiente ecuación, que describe la proyección de perspectiva de un punto del espacio cuyas

coordenadas están dadas respecto de un sistema de coordenadas arbitrario:

$$x_{c,i} = f \frac{R_{11}x_i + R_{12}y_i + R_{13}z_i + X_0}{R_{31}x_i + R_{32}y_i + R_{33}z_i + Z_0} \quad (2.15a)$$

$$y_{c,i} = f \frac{R_{21}x_i + R_{22}y_i + R_{23}z_i + Y_0}{R_{31}x_i + R_{32}y_i + R_{33}z_i + Z_0} \quad (2.15b)$$

Donde:

$$R_{11} = e_1^2 + e_2^2 - e_3^2 - e_4^2$$

$$R_{12} = 2(e_2e_3 + e_1e_4)$$

$$R_{13} = 2(e_2e_4 - e_1e_3)$$

$$R_{21} = 2(e_2e_3 - e_1e_4)$$

$$R_{22} = e_1^2 - e_2^2 + e_3^2 - e_4^2$$

$$R_{23} = 2(e_3e_4 + e_1e_2)$$

$$R_{31} = 2(e_2e_4 + e_1e_3)$$

$$R_{32} = 2(e_3e_4 - e_1e_2)$$

$$R_{33} = e_1^2 - e_2^2 - e_3^2 + e_4^2$$

Desarrollando las ecuaciones 2.15a y 2.15b, se puede determinar el valor de las proyecciones de perspectiva $X_{c,i}$ y $Y_{c,i}$ a partir de la suma de dos componentes referidas a una proyección ortográfica de cámara y a una corrección de perspectiva. Esto se muestra a continuación:

$$x_{c,i} = f \frac{\overbrace{R_{11}x_i + R_{12}y_i + R_{13}z_i + X_0}^{\text{proyección ortográfica}}}{Z_0} - \frac{\overbrace{R_{31}x_i x_{c,i} + R_{32}y_i x_{c,i} + R_{33}z_i x_{c,i}}^{\text{corrección de perspectiva}}}{Z_0} \quad (2.17a)$$

$$y_{c,i} = f \frac{\overbrace{R_{21}x_i + R_{22}y_i + R_{23}z_i + Y_0}^{\text{proyección ortográfica}}}{Z_0} - \frac{\overbrace{R_{31}x_i y_{c,i} + R_{32}y_i y_{c,i} + R_{33}z_i y_{c,i}}^{\text{corrección de perspectiva}}}{Z_0} \quad (2.17b)$$

Siendo que los elementos de $R_{i,j}$ del par de ecuaciones anteriores (2.17a, 2.17b), están compuestos de los cuaterniones de euler (e_1, e_2, e_3, e_4) , pertenecientes a la matriz

de rotación arbitraria descrita en la ecuación 2.14, se puede proponer la siguiente agrupación de términos:

$$P_i = \frac{f}{Z_0} e_i^2 \quad ; \quad i = 1, \dots, 4 \quad (2.18a)$$

$$P_5 = f \frac{X_0}{Z_0} \quad (2.18b)$$

$$P_6 = f \frac{Y_0}{Z_0} \quad (2.18c)$$

$$P_7 = -\frac{1}{f} \quad (2.18d)$$

Resultando el siguiente par de ecuaciones:

$$\begin{aligned} x_{c,i} = & (P_1^2 + P_2^2 - P_3^2 - P_4^2)x_i + 2(P_2P_3 + P_1P_4)y_i + 2(P_2P_4 - P_1P_3)z_i + P_5 + \\ & 2P_7(P_2P_4 + P_1P_3)x_ix_{c,i} + 2P_7(P_3P_4 - P_1P_2)y_ix_{c,i} + P_7(P_1^2 - P_2^2 - P_3^2 + P_4^2)z_ix_{c,i} \end{aligned} \quad (2.19a)$$

$$\begin{aligned} y_{c,i} = & 2(P_2P_3 - P_1P_4)x_i + (P_1^2 - P_2^2 + P_3^2 - P_4^2)y_i + 2(P_3P_4 + P_1P_2)z_i + P_6 + \\ & 2P_7(P_2P_4 + P_1P_3)x_iy_{c,i} + 2P_7(P_3P_4 - P_1P_2)y_iy_{c,i} + P_7(P_1^2 - P_2^2 - P_3^2 + P_4^2)z_iy_{c,i} \end{aligned} \quad (2.19b)$$

Agrupando todos los términos $P_1 \dots P_7$, se puede construir una versión más adecuada de las ecuaciones 2.19. La forma de agrupar dichos términos es como sigue:

$$b_1(\mathbf{P}) = P_1^2 + P_2^2 - P_3^2 - P_4^2 \quad (2.20a)$$

$$b_2(\mathbf{P}) = 2(P_2P_3 + P_1P_4) \quad (2.20b)$$

$$b_3(\mathbf{P}) = 2(P_2P_4 - P_1P_3) \quad (2.20c)$$

$$b_4(\mathbf{P}) = P_5 \quad (2.20d)$$

$$b_5(\mathbf{P}) = 2(P_2P_3 - P_1P_4) \quad (2.20e)$$

$$b_6(\mathbf{P}) = P_1^2 - P_2^2 + P_3^2 - P_4^2 \quad (2.20f)$$

$$b_7(\mathbf{P}) = 2(P_3P_4 + P_1P_2) \quad (2.20g)$$

$$b_8(\mathbf{P}) = P_6 \quad (2.20h)$$

$$b_9(\mathbf{P}) = 2P_7(P_2P_4 + P_1P_3) \quad (2.20i)$$

$$b_{10}(\mathbf{P}) = 2P_7(P_3P_4 - P_1P_2) \quad (2.20j)$$

$$b_{11}(\mathbf{P}) = P_7(P_1^2 - P_2^2 - P_3^2 + P_4^2) \quad (2.20k)$$

Al sustituir las expresiones 2.20a . . . 2.20k, dentro de las ecuaciones 2.19a y 2.19b, se obtienen las ecuaciones del modelo de perspectiva recursivo (Ecs. 2.21), dadas en función de las coordenadas tridimensionales de un punto (x_i, y_i, z_i) y de su proyección ortográfica en espacio de cámara $(x_{c,i}, y_{c,i})$. Los valores $P_1 \dots P_7$ son los parámetros del modelo y son conocidos como *parámetros de cámara*. Dichas ecuaciones se presentan a continuación.

$$f_x(x_i, y_i, z_i, x_{c,i}; \mathbf{P}) = \mathcal{O}_x(x_i, y_i, z_i; \mathbf{P}) + \mathcal{P}_x(x_i, y_i, z_i, x_{c,i}; \mathbf{P}) \quad (2.21a)$$

$$f_y(x_i, y_i, z_i, y_{c,i}; \mathbf{P}) = \mathcal{O}_y(x_i, y_i, z_i; \mathbf{P}) + \mathcal{P}_y(x_i, y_i, z_i, y_{c,i}; \mathbf{P}) \quad (2.21b)$$

Donde:

$$\mathcal{O}_x(x_i, y_i, z_i, \mathbf{P}) = b_1(\mathbf{P})x_i + b_2(\mathbf{P})y_i + b_3(\mathbf{P})z_i + b_4(\mathbf{P}) \quad (2.22a)$$

$$\mathcal{P}_x(x_i, y_i, z_i, x_{c,i}, \mathbf{P}) = b_9(\mathbf{P})x_ix_{c,i} + b_{10}(\mathbf{P})y_ix_{c,i} + b_{11}(\mathbf{P})z_ix_{c,i} \quad (2.22b)$$

$$\mathcal{O}_y(x_i, y_i, z_i, \mathbf{P}) = b_5(\mathbf{P})x_i + b_6(\mathbf{P})y_i + b_7(\mathbf{P})z_i + b_8(\mathbf{P}) \quad (2.22c)$$

$$\mathcal{P}_y(x_i, y_i, z_i, y_{c,i}, \mathbf{P}) = b_9(\mathbf{P})x_iy_{c,i} + b_{10}(\mathbf{P})y_iy_{c,i} + b_{11}(\mathbf{P})z_iy_{c,i} \quad (2.22d)$$

En las expresiones 2.21a y 2.21b del modelo de perspectiva recursivo, es importante notar que se han agrupado los términos del lado derecho de estas expresiones en dos

componentes; la primera ($\mathcal{O}_{x,y}(x_i, y_i, z_i, \mathbf{P})$) corresponde con la proyección ortográfica de un punto en espacio de cámara, y la segunda componente, representa una corrección a dicha proyección ortográfica, mediante una corrección de perspectiva ($\mathcal{P}_{x,y}(x_i, y_i, z_i, \{x_{c,i}, y_{c,i}\}, \mathbf{P})$).

2.4. Estimación de Parámetros de Cámara

En la sección anterior se presentaron las ecuaciones 2.21a y 2.21b, las cuales representan el modelo de perspectiva recursivo. Este par de ecuaciones está compuesto por una serie de parámetros de cámara ($P_1 \dots P_7$), cuya determinación es en esencia, la construcción del modelo de cámara de perspectiva. El cálculo de estos parámetros se realiza a través de uno de los procesos que caracterizan a la metodología de *CSM*, que es un proceso de estimación por minimización de una función de error $J(\mathbf{P})$, mejor conocido como proceso de *corrección diferencial por mínimos cuadrados* [13]. Dicho proceso es un proceso de estimación no lineal caracterizado por ser iterativo, en donde los parámetros del modelo son calculados a partir de valores iniciales propuestos que, mediante aproximaciones diferenciales sucesivas buscan satisfacer una condición de mínimo error. La función a minimizar que define la condición de error, es desarrollada a partir del modelo mismo de perspectiva recursivo y está expresada como sigue:

$$J(\mathbf{P}) = \sum_{i=1}^n \{ [x_{c,i} - f_x(x_i, y_i, z_i, x_{c,i}; \mathbf{P})]^2 + [y_{c,i} - f_y(x_i, y_i, z_i, y_{c,i}; \mathbf{P})]^2 \} W_i \quad (2.23)$$

La minimización de la ecuación 2.23, mediante el proceso de *corrección diferencial por mínimos cuadrados*, requiere de ciertas consideraciones debido a problemas de inestabilidad numérica, al tratar de obtener simultáneamente los siete parámetros de cámara [27]. Se necesitan proponer valores iniciales cercanos de P_i a la solución, para lograr convergencia en el curso del proceso iterativo.

Dada la naturaleza del séptimo parámetro de cámara igual al recíproco negativo de la distancia focal de la cámara (f), y siendo éste un parámetro intrínseco de cámara, se sugieren dos métodos para la recuperación de los siete parámetros de cámara. El primero de ellos propone la obtención del valor $P_7 = -1/f$, a partir de alguno de

los procesos de calibración de cámaras actualmente existentes, en los que el dato de la distancia focal de la cámara, es una variable calculable [38, 39]. Posteriormente, una vez conocido del valor de P_7 , partiendo de la ecuación 2.23 es posible estimar los seis parámetros de cámara restantes sin problemas de convergencia, inclusive sin proponer valor iniciales de $P_1 \dots P_6$, cercanos a su solución.

El segundo método para obtener los valores de P_i , consiste en proponer un rango de valores de P_7 para los cuales han de ser estimados los valores de $P_1 \dots P_6$; en cada nueva estimación de estos seis parámetros de cámara, se calcula el valor de $J(\mathbf{P})$ mediante la ecuación 2.23; donde $J(\mathbf{P})$ es igual al error medio cuadrático existente, entre una serie de muestras obtenidas de marcas visuales en espacio de cámara, y la proyección de perspectiva dada por el modelo de perspectiva recursivo, de dichas marcas visuales a partir de su localización tridimensional. Cada valor de $J(\mathbf{P})$ es asociado al valor de P_7 con el cual fue calculado y, tras evaluar todos los valores de P_7 , aquel que minimice de la mejor forma a la magnitud de $J(\mathbf{P})$, es elegido.

De los dos procedimientos para la obtención del valor de P_7 mencionados, el presente trabajo de investigación hace uso del segundo método, dándose una explicación más detenida en la sección 2.4.2. Es propósito de la siguiente sección es explicar del proceso de obtención de los primeros seis parámetros de visión $P_1 \dots P_6$.

2.4.1. Estimación de los Parámetros de Cámara 1..6

La obtención de los primeros seis parámetros de cámara, es realizada mediante un procesos de estimación conocido como *corrección diferencial por mínimos cuadrados*. En este proceso de estimación, se minimiza la función dada en la ecuación 2.23, la cual es dependiente de \mathbf{P} , donde $\mathbf{P} = [P_1 \dots P_6]^T$, ya que el parámetro P_7 , se considera constante.

El procedimiento de estimación de *corrección diferencial por mínimos cuadrados*, es un método iterativo de aproximaciones sucesivas. En el caso particular del modelo de perspectiva recursivo, en donde se desea estimar el valor solución, de las componentes del vector \mathbf{P} cuando la función de escalar de error $J \rightarrow 0$, el valor de las aproximaciones sucesivas $\Delta\mathbf{P}$ de \mathbf{P} , es dado mediante la condición necesaria para la minimización de las aproximaciones lineales, de la función J :

$$\Delta(P) = [A^T W A]^{-1} [A^T W \Delta Y] \quad (2.24)$$

Donde $[A^T W A]$ representa una matrix no singular de dimensiones 6×6 y $[A^T W \Delta Y]$, es un vector de 6×1 . Su obtención se menciona a continuación:

$$[A^T W A] = [\partial \mathbf{B}]^T [\mathbf{M}_A] [\partial \mathbf{B}] \quad (2.25)$$

$$[A^T W \Delta Y] = [\partial \mathbf{B}]^T [\mathbf{M}_B] [\mathbf{B}] \quad (2.26)$$

Los elementos de $[A^T W A]$ y $[A^T W \Delta Y]$ se definen a continuación:

$$[\partial \mathbf{B}] = \begin{bmatrix} \frac{\partial b_1}{\partial P_1} & \cdots & \frac{\partial b_1}{\partial P_6} \\ \vdots & \ddots & \vdots \\ \frac{\partial b_{11}}{\partial P_1} & \cdots & \frac{\partial b_{11}}{\partial P_6} \end{bmatrix} \quad (2.27)$$

$$[\mathbf{B}] = \begin{bmatrix} 1 \\ -b_1 \\ \vdots \\ -b_{11} \end{bmatrix} \quad (2.28)$$

El vector \mathbf{B} y los elementos diferenciales del jacobiano $\partial \mathbf{B}$, están conformados por cada una de las expresiones $b_1 \dots b_{11}$ dependientes de los parámetros de visión $P_1 \dots P_7$, presentadas en las ecuaciones 2.20a...2.20k.

Siguiendo el mismo paradigma de matrices de momentos presentado en [32], pero aplicado al modelo de perspectiva recursivo, es posible realizar una compresión eficiente de la información proveniente de n muestras $\{x_{c,i}, y_{c,i}\}$ en espacio de cámara, con su correspondiente localización en espacio tridimensional $\{x_i, y_i, z_i\}$, en matrices de dimensiones invariantes de acuerdo al número de muestras. Las matrices \mathbf{M}_A y \mathbf{M}_B agrupan el total de matrices de momentos, como se expone a continuación:

$$[\mathbf{M}_A] = \begin{bmatrix} M_1 & \vdots & 0 & \vdots & M_2 \\ \cdots & \cdot & \cdots & \cdot & \cdots \\ 0 & \vdots & M_1 & \vdots & M_3 \\ \cdots & \cdot & \cdots & \cdot & \cdots \\ M_2^T & \vdots & M_2^T & \vdots & M_4 \end{bmatrix} \quad (2.29)$$

$$[\mathbf{M}_B] = \begin{bmatrix} M_x & \vdots & M_1 & \vdots & 0 & \vdots & M_2 \\ \cdots & \cdot & \cdots & \cdot & \cdots & \cdot & \cdots \\ M_y & \vdots & 0 & \vdots & M_1 & \vdots & M_3 \\ \cdots & \cdot & \cdots & \cdot & \cdots & \cdot & \cdots \\ M_{xy} & \vdots & M_2^T & \vdots & M_3^T & \vdots & M_4 \end{bmatrix} \quad (2.30)$$

Teniendo que:

$$\mathbf{M}_1 = \begin{bmatrix} \sum x_i^2 W_i & \sum x_i y_i W_i & \sum x_i z_i W_i & \sum x_i W_i \\ & \sum y_i^2 W_i & \sum y_i z_i W_i & \sum y_i W_i \\ & & \sum z_i^2 W_i & \sum z_i W_i \\ \text{simétrica} & & & \sum W_i \end{bmatrix} \quad (2.31)$$

$$\mathbf{M}_2 = \begin{bmatrix} \sum x_i^2 x_{c,i} W_i & \sum x_i y_i x_{c,i} W_i & \sum x_i z_i x_{c,i} W_i \\ \sum x_i y_i x_{c,i} W_i & \sum y_i^2 x_{c,i} W_i & \sum y_i z_i x_{c,i} W_i \\ \sum x_i z_i x_{c,i} W_i & \sum y_i z_i x_{c,i} W_i & \sum z_i^2 x_{c,i} W_i \\ \sum x_i x_{c,i} W_i & \sum y_i x_{c,i} W_i & \sum z_i x_{c,i} W_i \end{bmatrix} \quad (2.32)$$

$$(2.33)$$

$$\mathbf{M}_3 = \begin{bmatrix} \sum x_i^2 y_{c,i} W_i & \sum x_i y_i y_{c,i} W_i & \sum x_i z_i y_{c,i} W_i \\ \sum x_i y_i y_{c,i} W_i & \sum y_i^2 y_{c,i} W_i & \sum y_i z_i y_{c,i} W_i \\ \sum x_i z_i y_{c,i} W_i & \sum y_i z_i y_{c,i} W_i & \sum z_i^2 y_{c,i} W_i \\ \sum x_i y_{c,i} W_i & \sum y_i y_{c,i} W_i & \sum z_i y_{c,i} W_i \end{bmatrix} \quad (2.34)$$

$$\mathbf{M}_4 = \begin{bmatrix} M_{411} & M_{412} & M_{413} \\ & M_{422} & M_{423} \\ \text{simétrica} & & M_{433} \end{bmatrix} \quad (2.35)$$

En donde:

$$M_{411} = \sum x_i^2(x_{c,i}^2 + y_{c,i}^2)W_i \quad (2.36)$$

$$M_{412} = \sum x_i y_i(x_{c,i}^2 + y_{c,i}^2)W_i \quad (2.37)$$

$$M_{413} = \sum x_i z_i(x_{c,i}^2 + y_{c,i}^2)W_i \quad (2.38)$$

$$M_{422} = \sum y_i^2(x_{c,i}^2 + y_{c,i}^2)W_i \quad (2.39)$$

$$M_{423} = \sum y_i z_i(x_{c,i}^2 + y_{c,i}^2)W_i \quad (2.40)$$

$$M_{433} = \sum z_i^2(x_{c,i}^2 + y_{c,i}^2)W_i \quad (2.41)$$

$$M_x = \begin{bmatrix} \sum x_i x_{c,i} W_i \\ \sum y_i x_{c,i} W_i \\ \sum z_i x_{c,i} W_i \\ \sum x_{c,i} W_i \end{bmatrix} \quad (2.42)$$

$$M_y = \begin{bmatrix} \sum x_i y_{c,i} W_i \\ \sum y_i y_{c,i} W_i \\ \sum z_i y_{c,i} W_i \\ \sum y_{c,i} W_i \end{bmatrix} \quad (2.43)$$

$$M_{xy} = \begin{bmatrix} \sum x_i(x_{c,i}^2 + y_{c,i}^2)W_i \\ \sum y_i(x_{c,i}^2 + y_{c,i}^2)W_i \\ \sum z_i(x_{c,i}^2 + y_{c,i}^2)W_i \end{bmatrix} \quad (2.44)$$

Al analizar los términos de las matrices 2.29 y 2.30, se puede notar que todos ellos están compuestos por sumatorias, que tienen la función de agrupar el total de muestras procedentes de espacio de cámara $\{x_{c,i}, y_{c,i}\}$, con su respectiva posición en espacio tridimensional $\{x_i, y_i, z_i\}$. No importando el número de muestras, tomadas para la estimación de un conjunto específico de parámetros de cámara, las matrices 2.29 y 2.30 conservarán sus dimensiones originales.

Una vez conocido el significado de cada uno de los términos de la ecuación 2.24, para calcular el valor de las aproximaciones diferenciales lineales $\Delta(\mathbf{P})$, del proceso de *corrección diferencial por mínimos cuadrados*, se procede en breve a presentar

esquemáticamente el algoritmo para cumplir con dicho proceso de estimación. El algoritmo presentado es iterativo y cada iteración cumple la función de calcular un nuevo valor de $\Delta\mathbf{P}$, para aproximar los valores de $P_{1..6}$ a cumplir con la condición de mínimo error $J(\mathbf{P})$. El mecanismo de paro del algoritmo consiste en la evaluación de la magnitud de la norma del vector de $\Delta\mathbf{P}$; cuando dicha norma es menor a una tolerancia propuesta, se dice que los valores de \mathbf{P} ya no tendrán un cambio significativo de una iteración a otra y por tanto, se ha alcanzado una solución estable.

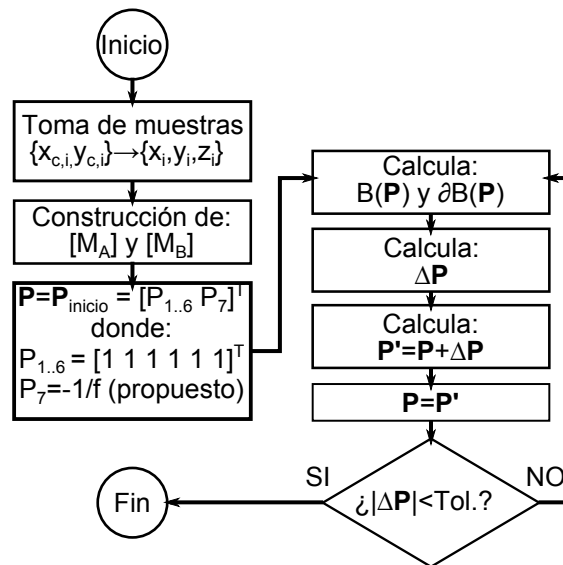


Figura 2.7: Algoritmo de estimación de parámetros de cámara $P_{1..6}$.

2.4.2. Estimación del Parámetro de Cámara 7

Una vez conocido el proceso de determinación de los primeros seis parámetros de visión ($P_1 \dots P_6$) al mantener el parámetro P_7 constante, es posible calcular el valor óptimo del parámetro P_7 atendiendo a la magnitud del error $J(\mathbf{P})$, definido en la ecuación 2.23. Si se propone un rango de valores de distancias focales (f), donde $f \ll 0$, para el cual se han de determinar los valores de $P_1 \dots P_6$, a partir de un conjunto de “ n ” muestras $\{x_{c,i}, y_{c,i}, x_i, y_i, z_i\}_{i=1..n}$; sabiendo que $P_7 = -1/f$, se dice que existe un único valor de f para el cual, la magnitud de la función escalar $J(\mathbf{P})$, alcanzar su valor mínimo.

El algoritmo para el cálculo del parámetro P_7 se muestra a continuación:

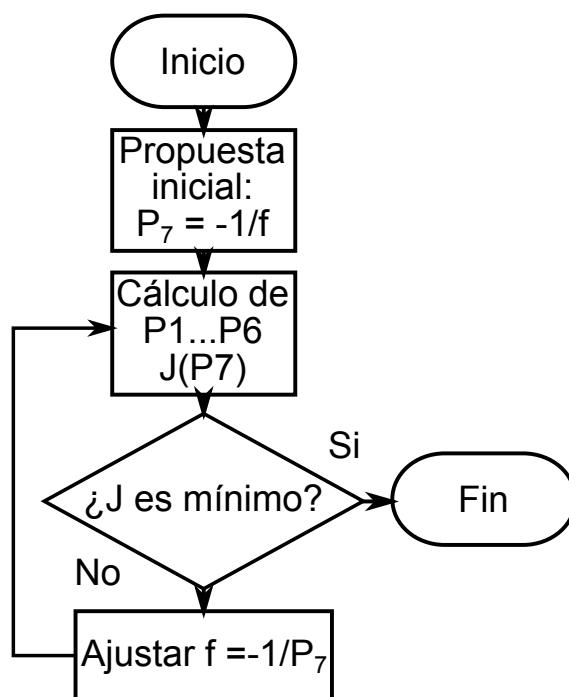


Figura 2.8: Esquema de algoritmo para cálculo del parámetro de cámara P_7 .

2.5. Algoritmo de Estimación de Posición Tridimensional de Marcas Visuales

Al tener conocimiento de los siete parámetros de cámara de un sensor, mediante el modelo de perspectiva recursivo propuesto en las ecuaciones 2.21a y 2.21b y dadas las coordenadas de un punto en espacio tridimensional (x, y, z) , es posible calcular la proyección de dicho punto, en el espacio de cámara del sensor en cuestión (x_c, y_c) . No obstante, si se deseara partir de espacio de cámara para determinar la posición tridimensional de un punto, el conocimiento de los parámetros de un solo modelo de cámara es insuficiente para concretar dicha tarea. Para llevar a cabo el mapeo de coordenadas, de espacio de cámara a espacio tridimensional, hace falta tener conocimiento de los parámetros del modelo de cámara de por lo menos dos sensores distintos ($N_c \geq 2$), los cuales mediante la cobertura de un punto tridimensional en común, generan un par de proyecciones del mismo punto en su respectivo espacio de cámara.

La importancia de la determinación de la posición tridimensional de un punto, a partir del conocimiento de por lo menos un par de proyecciones del mismo en es-

pacio de cámara, radica en el hecho, de que a partir de la información de dicha posición tridimensional, es posible construir algoritmos más complejos, como el proceso conocido como caracterización de superficies, que consiste en la estimación de los parámetros de una ecuación topológica dada, a partir de una nube de puntos tridimensionales.

El proceso de determinación de la posición tridimensional de una marca visual, es similar a un método de triangulación en el que, a partir de las coordenadas en espacio de cámara de un punto, teniendo por lo menos dos cámaras, se deduce la posición tridimensional de dicho punto. No obstante, dicho método es extensible al caso en el que se conocen las proyecciones del punto a partir de “ N_c ” espacios cámaras ($N_c \geq 1$). Haciendo uso de teoría de estimación, mediante un proceso de estimación lineal, al minimizar la siguiente expresión se puede determinar la posición tridimensional de un punto visto por de N_c cámaras:

$$\phi = \sum_{i=1}^{N_c} [x_c^{(i)} - f_x(x, y, z, x_c^{(i)}; (P)^{(i)})]^2 + [y_c^{(i)} - f_y(x, y, z, y_c^{(i)}; (P)^{(i)})]^2 \quad (2.45)$$

Donde la condición necesaria para la minimización de ϕ arroja la siguiente solución:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = A^{-1}B \quad (2.46)$$

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{12} & A_{22} & A_{23} \\ A_{13} & A_{23} & A_{33} \end{bmatrix}, \quad B = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix} \quad (2.47)$$

Cada término de A y de B es definido a continuación:

$$\begin{aligned}
A_{11} &= \sum_{i=1}^{N_c} [(b_1^{(i)} + b_9^{(i)} x_c^{(i)})^2 + (b_5^{(i)} + b_9^{(i)} y_c^{(i)})^2] \\
A_{12} &= \sum_{i=1}^{N_c} [(b_1^{(i)} + b_9^{(i)} x_c^{(i)})(b_2^{(i)} + b_{10}^{(i)} x_c^{(i)}) + (b_5^{(i)} + b_9^{(i)} y_c^{(i)})(b_6^{(i)} + b_{10}^{(i)} y_c^{(i)})] \\
A_{13} &= \sum_{i=1}^{N_c} [(b_1^{(i)} + b_9^{(i)} x_c^{(i)})(b_3^{(i)} + b_{11}^{(i)} x_c^{(i)}) + (b_5^{(i)} + b_9^{(i)} y_c^{(i)})(b_7^{(i)} + b_{11}^{(i)} y_c^{(i)})] \\
A_{22} &= \sum_{i=1}^{N_c} [(b_2^{(i)} + b_{10}^{(i)} x_c^{(i)})^2 + (b_6^{(i)} + b_{10}^{(i)} y_c^{(i)})^2] \\
A_{23} &= \sum_{i=1}^{N_c} [(b_2^{(i)} + b_{10}^{(i)} x_c^{(i)})(b_3^{(i)} + b_{11}^{(i)} x_c^{(i)}) + (b_6^{(i)} + b_{10}^{(i)} y_c^{(i)})(b_7^{(i)} + b_{11}^{(i)} y_c^{(i)})] \\
A_{33} &= \sum_{i=1}^{N_c} [(b_3^{(i)} + b_{11}^{(i)} x_c^{(i)})^2 + (b_7^{(i)} + b_{11}^{(i)} y_c^{(i)})^2] \\
B_1 &= \sum_{i=1}^{N_c} [(x_c^{(i)} - b_4^{(i)})(b_1^{(i)} + b_9^{(i)} x_c^{(i)}) + ((y_c^{(i)} - b_8^{(i)})(b_5^{(i)} + b_9^{(i)} y_c^{(i)})] \\
B_2 &= \sum_{i=1}^{N_c} [(x_c^{(i)} - b_4^{(i)})(b_2^{(i)} + b_{10}^{(i)} x_c^{(i)}) + ((y_c^{(i)} - b_8^{(i)})(b_6^{(i)} + b_{10}^{(i)} y_c^{(i)})] \\
B_3 &= \sum_{i=1}^{N_c} [(x_c^{(i)} - b_4^{(i)})(b_3^{(i)} + b_{11}^{(i)} x_c^{(i)}) + ((y_c^{(i)} - b_8^{(i)})(b_7^{(i)} + b_{11}^{(i)} y_c^{(i)})]
\end{aligned}$$

En cada término de A y B , las expresiones $b_1^{(i)} \dots b_{11}^{(i)}$ engloban los parámetros $P_1 \dots P_7$ de la “ i -ésima” cámara ($i = 1 \dots N_c$).

2.6. Algoritmo de Estimación de la Configuración del Manipulador

La metodología *CSM* engloba un conjunto de procedimientos con los que es posible realizar el control de la posición de un robot, a partir de información procedente de

cámaras de video; en la sección actual, se expone el algoritmo fundamental con el cual, a partir de la información proveniente de la ubicación de un objetivo en espacio de cámara, se puede controlar al brazo de robot industrial a alcanzar dicho objetivo.

La estimación de la configuración interna del manipulador se calcula mediante un proceso de *corrección diferencial por mínimos cuadrados*. La función a minimizar representa el empate en espacio de cámara de un conjunto de “ $N_t \times N_c$ ” puntos artificiales, con el mismo número de puntos objetivo vistos desde “ N_c ” cámaras ($N_c \geq 2$):

$$\gamma(\Theta) = \sum_{i=1}^{N_c} \sum_{j=1}^{N_t(i)} \{ [x_{t_i}^{(j)} - f_x(r_{x_i}^{(j)}(\Theta), r_{y_i}^{(j)}(\Theta), r_{z_i}^{(j)}(\Theta), x_{t_i}^{(j)}; \mathbf{P}^{(i)})]^2 + [y_{t_i}^{(j)} - f_y(r_{x_i}^{(j)}(\Theta), r_{y_i}^{(j)}(\Theta), r_{z_i}^{(j)}(\Theta), y_{t_i}^{(j)}; \mathbf{P}^{(i)})]^2 \} \quad (2.48)$$

A grandes rasgos, los puntos objetivo ($x_{t_i}^{(j)}, y_{t_i}^{(j)}$) se encuentran inmóviles en espacio de cámara, mientras que los puntos artificiales ($f_x(r_{xyz_i}^{(j)}(\Theta); \mathbf{P}^{(i)}), f_y(r_{xyz_i}^{(j)}(\Theta); \mathbf{P}^{(i)})$), son generados bajo el supuesto de que se encuentran sobre la herramienta del manipulador, siendo su localización en espacio de cámara dependiente de $\mathbf{P} \times N_c$ parámetros del modelo de visión y de $\Theta = [J_1 \dots J_6]^T$ valores de junta del modelo cinemático del manipulador. Al minimizar la función de error propuesta en la ecuación 2.48, los parámetros a estimar de la función son los valores de $\Theta = [J_1 \dots J_6]^T$, y dicha solución se obtiene cuando se ha encontrado una configuración del manipulador para la cual, los puntos artificiales sobre su herramienta empatan con los puntos objetivo, en espacio de cámara.

La generación de los puntos objetivo en espacio de cámara, en la práctica, se realiza mediante la emisión de un haz de luz láser hacia una posición de “destino” para el efector final de manipulador; siendo sencillo, de esta forma, designar múltiples puntos objetivo, sin dejar marca permanente sobre una superficie.

La información artificial generada sobre la herramienta del robot, se construye a partir de la estimación de la posición tridimensional de los puntos objetivo (ecuación 2.45); al trasladar dichas coordenadas tridimensionales al extremo de la herramienta del robot, se pueden proyectar nuevamente estos puntos a espacio de cámara,

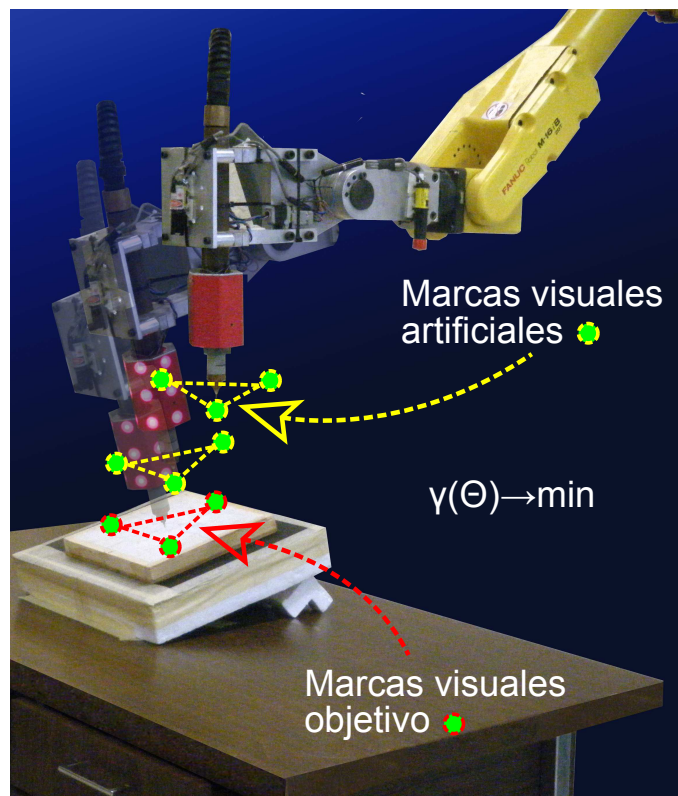


Figura 2.9: Ejemplificación de emparejamiento de marcas visuales artificiales y marcas visuales objetivo, durante algoritmo de estimación de configuración interna del manipulador.

generándose así la proyección de un cúmulo de marcas visuales artificiales, cuya posición coincide con la región en donde se encuentra la herramienta del robot en espacio de cámara.

Capítulo 3

Determinación de Parámetros de Visión para Cámaras Virtuales

El capítulo anterior trata explícitamente de la metodología de *Manipulación de Espacio de Cámara*, ofreciéndose una serie de tópicos en los cuales se detalla el modo de empleo de la metodología, bajo la propuesta de un modelo de visión, a partir del cual se pretende realizar el control de un manipulador industrial. El modelo de visión expuesto en el capítulo anterior, es el modelo de *perspectiva recursivo*, el cual ha demostrado ser robusto y preciso en proyectos precedentes [26, 27, 6].

El objetivo del presente capítulo es, retomar los temas vistos en el capítulo 2, y aplicarlos al caso particular en el que un sistema de control por visión *CSM* de un robot industrial, basa la adquisición de imágenes, en un sistema de cámaras virtuales. La forma de afrontar este caso particular de estudio, lleva a la descripción de aspectos tales como el diseño del sistema de adquisición de la información visual, el procesamiento de imágenes posterior necesario y los procedimientos teóricos para el desarrollo de pruebas experimentales, con las que el sistema de cámaras virtuales, ha de validar su funcionalidad.

3.1. Diseño, Simulación y Construcción de un Sistema de Cámaras Virtuales Compatible con la Metodología de CSM

Un sistema de cámaras virtuales es la conjunción de una o varias cámaras de video, con elementos reflexivos (espejos) ya sea planos o curvos [16, 17, 2, 1, 18], que al interferir con el eje óptico de alguna de las cámaras involucradas, permiten la modificación del campo visual de dicho sensor; ya sea reorientando su eje óptico o aumentando las dimensiones del campo visual cubierto. La manipulación del campo visual de una cámara mediante espejos, también puede aprovecharse para multiplicar el número de perspectivas de una escena adquiribles por un sensor opto-electrónico, dotándolo así de mayor funcionalidad.

La idea de crear un sistema de cámaras virtuales, para emplearlo como medio de adquisición de imágenes para el control por visión de un robot industrial, sugiere ciertos lineamientos, de acuerdo a las características de la metodología *CSM*; no obstante muchas de las facultades de estos sistemas de captura de imágenes son aprovechables. Algunos de los lineamientos que deben ser tomados en cuenta para el diseño de un sistema de adquisición de imágenes, utilizado como plataforma para un sistema de control por visión *CSM*, son los siguiente:

- La metodología de *CSM* exige cuando menos, un mínimo de dos cámaras ($N_c \geq 2$) o perspectivas distintas de la misma escena o espacio de acción, dentro del cual será controlado el autómata.
- La precisión con la cual se efectúa el control de un robot, mediante la metodología de *CSM*, es sensible a la razón existente entre la resolución (*pixeles*) de las cámaras empleadas y las dimensiones del campo visual asociado (*mm*); incrementando la precisión del sistema al utilizarse sensores de alta resolución con un campo visual poco extendido.
- Los modelos de visión actualmente existentes en *CSM*, son el modelo de cámara ortográfico y el modelo cámara orificio, cuyo campo de acción esta restringido a cámaras de video con valores altos de distancia focal [12, 36, 41], es decir, de reducida apertura visual; esto para evitar efectos de distorsión en imágenes, derivadas del uso de cámaras con lentes de *corta focal* [14].

- Al hablar en términos de aberraciones ópticas, aquellas conocidas como *distorsiones* ya sea en espejos o lentes, por su clasificación dentro del grupo de aberraciones monocromáticas o de Seidel [14], no son modelables ni corregibles por los modelos de visión *CSM*; no obstante, distorsiones en imágenes razonablemente bajas son aceptables ya que la metodología *CSM*, en su conformación, emplea modelos de visión sencillos, cuyos parámetros son actualizados con frecuencia para asegurar su validez a través de la localidad por la cual se desplaza el efector final del manipulador controlado. De lo anterior se deduce que un sistema de cámaras virtuales diseñado para efectos de un sistema de control por visión *CSM*, puede emplear por el momento únicamente espejos planos.
- Dimensiones del campo de trabajo del robot a controlar por visión. En el capítulo 4 se describe el equipo disponible en el laboratorio de robótica del CIEP-FI, UASLP, para los fines del presente trabajo de investigación; entre los artículos disponibles, se encuentra el robot a controlar mediante *CSM*, cuya marca y modelo es *FANUC, M-16iB/20T*. Dicho manipulador industrial está caracterizado por tener un amplio espacio de trabajo, debido a las dimensiones de sus eslabones, por lo que para asegurar una relación adecuada entre la resolución de las cámaras virtuales y el tamaño del campo de visión cubierto, los espejos que componen el sistema óptico de adquisición deben de ser reorientables para tener cobertura de diferentes regiones del espacio de trabajo del robot. En el presente estudio se contemplan únicamente observadores virtuales que permanecerán inmóviles durante todo el transcurso de una maniobra de control por *CSM*, y solo hasta que dicha tarea culmina, se pueden reorientar los espejos del sistema de cámaras virtuales; considerando nuevos parámetros de visión iniciales, para cada cámara reorientada.
- De acuerdo a los resultados de una serie de experimentos reportados en [42], existe un ángulo óptimo entre los ejes focales de las cámaras de video involucradas en un sistema de control de robots por visión *CSM*, para el cual la varianza de la estimación de la posición tridimensional de un objetivo es mínima; dicho ángulo entre ejes focales es de 90 y tiene un margen de ± 30 . Un gráfico donde se muestra el comportamiento la varianza de la estimación de la posición tridimensional de un objetivo, en función del ángulo entre dos cámaras de video, y expresando la varianza a través de la traza de la matriz de covarianza que agrupa las respectivas varianzas de la estimación, se muestra

en la Figura 3.1. Al tratar en el presente proyecto con un sistema de adquisición de imágenes basado en cámaras virtuales, aplicado a la metodología de *CSM*, el ángulo entre los ejes focales de cada observador virtual, debe de ser considerado.

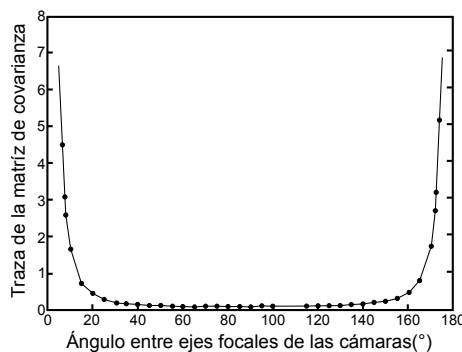


Figura 3.1: Comportamiento de la variancia de la estimación de la posición tridimensional de un objetivo en *CSM*, con respecto del ángulo entre ejes focales de dos cámaras; (véase [42]).

3.1.1. Diseño del Sistema de Cámaras Virtuales

En resumen, de lo sugerido en los puntos tratados en la sección anterior, se desea construir un sistema de cámaras virtuales compatible con la metodología *CSM*, el cual debe emplear espejos planos en su construcción para ofrecer por lo menos dos perspectivas distintas de una misma región de interés; y este debe ser reconfigurable o bien que sus espejos sean reorientables para tener cobertura de diversas zonas del área de trabajo del robot y a su vez elegir un ángulo adecuando entre los ejes focales de las cámaras virtuales.

Para diseñar el sistema de cámaras virtuales, se atenderá el caso más simple en el que únicamente se emplean dos observadores para realizar la adquisición de imágenes necesaria en *CSM*. Por lo tanto, se propone el uso de una sola cámara de video, que con la ayuda de espejos logre capturar el par de perspectivas necesario.

Dado que en la literatura existe un basto número de sistemas de cámaras virtuales propuesto, el diseño del sistema aquí empleado partirá de alguna configuración existente. La determinación de las dimensiones específicas del diseño, será dada mediante la simulación computacional de este sistema catadióptrico.

A continuación se muestran algunos de los sistemas de cámaras virtuales encontra-

dos en la literatura [16]; de ellos, ha sido elegido uno solo, que por su arquitectura presenta características deseables para el desarrollo de un sistema de control por visión de robots mediante *CSM*. Los sistemas catadióptricos siguientes, son clasificados de acuerdo al número de espejos de los que se componen, y todos tienen en común el uso de una sola cámara de video.

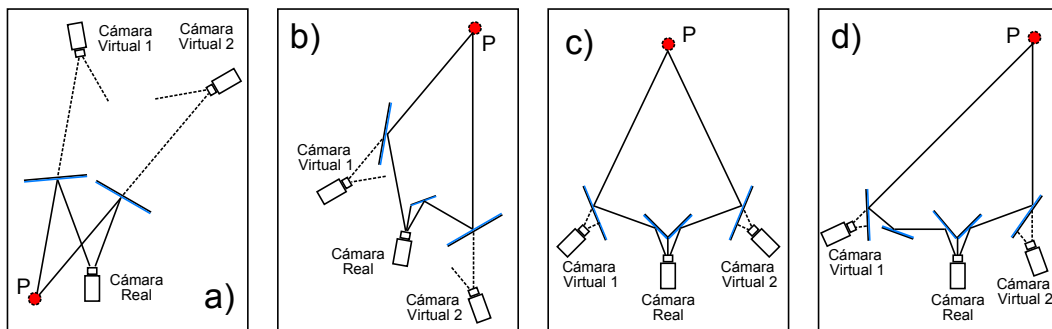


Figura 3.2: Bosquejos de sistemas de cámaras virtuales encontrados en la literatura; a) Con dos espejos; b) Con tres espejos; c) Con cuatro espejos; d) Con cinco espejos.

El primer sistema de cámaras virtuales presentado en la figura 3.2a, por contar solo con dos espejos, parece ser un modelo sencillo de cámara virtual, útil para ser aplicado en *CSM*, no obstante, presenta algunos inconvenientes:

- El control del ángulo entre los ejes ópticos de las cámaras virtuales es limitado, ya que para este fin solo es posible controlar el ángulo de rotación de los espejos alrededor de su eje vertical.
- Las imágenes adquiridas, requieren de ser invertidas en su procesamiento.

El segundo sistema catadióptrico presentado en la figura 3.2b, es un sistema de adquisición el cual cuenta con tres espejos. El control del eje óptico entre su par de cámaras virtuales, es más flexible que el sistema presentado en la figura 3.2a, no obstante, presenta los siguientes inconvenientes:

- Su arquitectura es asimétrica lo cual pudiera no ser deseable, dado que la longitud de cada eje óptico de los observadores virtuales podría variar ampliamente, lo que pudiera dar pie a dejar fuera de foco, o sin enfoque posible, a alguna de las dos cámaras virtuales de la zona de interés.

- Para lograr una configuración óptima para la captura de una escena pudiera ser necesario modificar el ángulo de todos los espejos existentes.

El tercer sistema de adquisición, mostrado en la figura 3.2c, está compuesto de cuatro espejos, y su arquitectura pudiera ser o no simétrica. Respecto de los anteriores sistemas, en éste en particular se puede controlar fácilmente el ángulo existente entre los ejes ópticos de las cámaras virtuales, y se puede capturar la escena de un objetivo, únicamente reorientando los espejos laterales. Como desventaja tendría:

- Constructivamente es un sistema de adquisición de imágenes más complejo que los anteriores.

El cuarto sistema de cámaras virtuales expuesto en la figura 3.2d, consta de cinco espejos. Este sistema es aun más complejo que los anteriores y es asimétrico. El ángulo entre los ejes ópticos de las cámaras virtuales existentes puede ser modificado fácilmente, no obstante se encuentran las siguientes desventajas:

- Puede presentar la misma desventaja de arquitectura asimétrica que el sistema de adquisición de la figura 3.2b.
- El número de espejos existentes, convierte en no trivial al problema de orientar los observadores virtuales, hacia una escena de interés.

Antes de hacer mención del sistema de adquisición seleccionado de entre los cuatro presentados, hay que notar un aspecto en común entre ellos. Puesto que al estar estos sistemas acoplados a un solo sensor real de captura de video, las imágenes captadas por dicho sensor tendrán que ser separadas a fin de aislar cada una de las escenas captadas, para su futuro procesamiento, sin embargo, la separación de imágenes es vertical para todos los distintos sistemas de adquisición. Otro tipo de separación de imágenes no fue considerada, ya que la cámara de video con la que se cuenta (Ueye UI-1450-CM), ofrece una captura natural de imagen de 1600×1200 pixeles y pensar en otra forma de división de la imagen, por ejemplo en el sentido horizontal, limitaría en buena medida el rango vertical de las capturas.

El sistema de cámaras virtuales seleccionado es el de la figura 3.2c; aunque dicho sistema compuesto de cuatro espejos, es más complejo que su par de predecesores de

dos y tres espejos, cuenta con ciertas características geométricas favorables, como simetría y que el control de los ejes ópticos de cada cámara virtual puede ser realizado simplemente con reorientar los espejos laterales.

3.1.2. Simulación del Sistema de Cámaras Virtuales

La simulación computacional del sistema catadióptrico ilustrado en la figura 3.2c, se realizó mediante el conocido *software* de MATLAB[®]. Mediante este software, a partir de rutinas propias del mismo, es fácil programar las relaciones matemáticas presentes en la óptica del sistema de observadores virtuales, a fin de generar un gráfico tridimensional que permita percibir claramente su comportamiento de acuerdo a la inclusión de ciertos parámetros, derivados de los siguientes aspectos:

- Dimensiones del campo de trabajo del robot (FANUC M-16iB/20T).
- Espacio disponible en el laboratorio de robótica del CIEP-FI, UASLP ($6,4 \times 8,37 \times 3,53$ m).
- Control apropiado de los ángulos entre ejes focales de los observadores virtuales, considerándose deseable un ángulo de 90 ± 30 grados [42].
- Apertura del campo visual de la cámara de video empleada.

Para facilitar el desarrollo de los algoritmos presentes en la simulación del sistema de cámaras virtuales, se realizó un análisis del bosquejo de la figura 3.2c, con la finalidad de comprender la óptica del sistema conformado básicamente de elementos reflexivos planos. La ley que rige la óptica de los espejos planos es la ley de “reflexión” y establece que “*el ángulo de incidencia de un rayo de luz sobre la superficie de un medio reflejante, es igual al ángulo de reflexión del rayo reflejado; donde el rayo incidente, la normal a la superficie reflejante y el rayo reflejado, se encuentran en un mismo plano, denominado plano de incidencia*” [14]; es decir $\theta_i = \theta_r$ (figura 3.3).

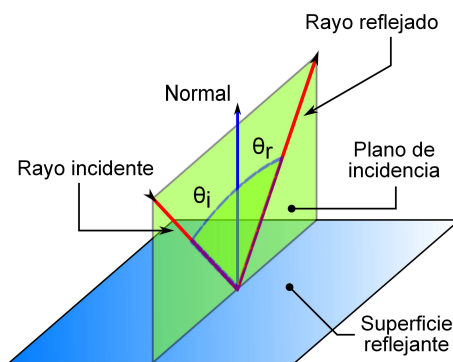


Figura 3.3: Reflexión de un rayo incidente en un medio reflejante.

A partir del sistema de cámaras virtuales de la figura 3.2c, se realizó un análisis óptico-geométrico, el cual es válido para el caso aquí tratado. Este tipo clásico de análisis, sirve para modelar la interacción de haces de luz visible, con elementos reflectivos y refractivos, mediante la aplicación de las leyes de la óptica de la reflexión y la refracción (*ley de Snell*) [14, 15]. El análisis óptico-geométrico realizado, está limitado al estudio de la interacción del campo visual de la cámara real con los espejos del sistema, en tanto que la óptica interna del sistema de lentes de la cámara es ignorado. Dicho análisis se presenta en la siguiente figura (Figura 3.4).

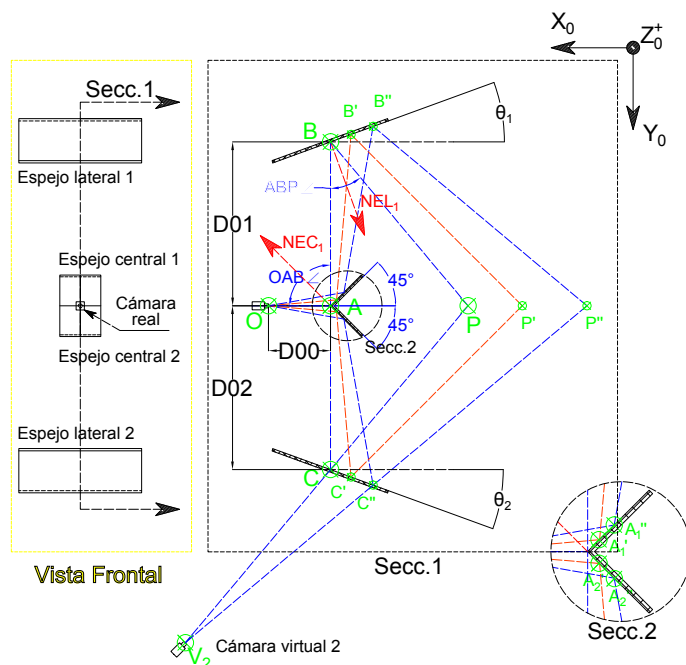


Figura 3.4: Análisis de óptico-geométrico de sistema de cámaras virtuales, compuesto de 4 espejos y una sola cámara real.

Para simplificar el análisis óptico geométrico de la figura 3.4, se hace una consideración importante en el diseño de dicho sistema catadióptrico, la cual consiste en mantener fijo el ángulo de los espejos centrales, de esta forma el enfoque de un objetivo viene dado casi exclusivamente por la manipulación de la orientación de los espejos laterales. Observando la figura 3.4, el ángulo de rotación de los espejos centrales es dado alrededor de un eje coincidente con el punto A y paralelo al eje Z_0 , en una magnitud de ± 45 en referencia del eje X_0 del sistema coordenado $\{X_0, Y_0, Z_0\}$.

En el diagrama de la (figura 3.4) se pueden apreciar todos los elementos principales de un sistema catadióptrico, como el presentado en la figura 3.2c. Dada la simetría de componentes hacia arriba y hacia abajo del diagrama, algunos elementos fueron omitidos. El objetivo principal del análisis óptico-geométrico de dicho diagrama se centra en la deducción de la orientación de los espejos laterales, dada la localización de un punto objetivo \mathbf{P} ; luego de conocer dicho comportamiento, se puede modelar un entorno similar al existente en el laboratorio de robótica del CIEP-FI (UASLP), es decir, considerando los aspectos mencionados al inicio de esta sección.

Como primera fase del análisis se deben reconocer los elementos del diagrama de la figura 3.2c:

- *Cámara real*: Representación del único sensor real del sistema catadióptrico.
- *Espejos centrales 1 y 2*: Elementos reflexivos, empleados para redireccionar el eje óptico de la cámara real en una primera etapa.
- *Espejos laterales 1 y 2*: Elementos reflexivos, empleados para redireccionar el eje óptico de la cámara real en una segunda etapa.
- *Origen del robot*: Representación del origen de coordenadas del robot, como elemento de referencia para los puntos del diagrama; dicho elemento se encuentra fuera del recuadro de la *sección 1* (Sec. 1), por lo que su localización no coincide con el plano de corte que da origen a esta sección.
- *Cámara virtual 2*: Representación del observador virtual generado detrás del espejo lateral 2.

También se pueden notar una serie de cotas, puntos, segmentos y rectas:

- Punto \mathbf{O} : Origen de cámara real.

- Punto **A**: Origen del vértice convexo, formado por los espejos centrales 1 y 2.
- Punto **B**: Punto coincidente con el eje de rotación vertical del espejo lateral 1.
- Punto **C**: Punto coincidente con el eje de rotación vertical del espejo lateral 2.
- Punto **P**: Punto objetivo para análisis óptico-geométrico. Intersección de los segmentos BP, CP , pertenecientes al límite interior del campo visual de las cámaras virtuales.
- Punto **P'**: Intersección de los segmentos $B'P', C'P'$, pertenecientes al eje central del campo visual de las cámaras virtuales.
- Punto **P''**: Intersección de los segmentos $B''P'', C''P''$, pertenecientes al límite exterior del campo visual de las cámaras virtuales.
- Punto A'_1, A'_2, A''_1, A''_2 : Puntos adicionales sobre espejos centrales 1 y 2.
- Punto B', B'' : Puntos adicionales sobre espejo lateral 1.
- Punto C', C'' : Puntos adicionales sobre espejo lateral 2.
- Recta NEC_1 : Normal a espejo central 1.
- Recta NEL_1 : Normal a espejo lateral 1.
- Cota **D00**: Distancia desde cámara real hasta origen de espejos centrales en **A**.
- Cota **D01**: Distancia desde origen de espejo central en (**A**) hasta origen de espejo lateral 1 en (**B**).
- Cota **D02**: Distancia desde origen de espejo central en (**A**) hasta origen de espejo lateral 2 en (**C**).
- Segmentos $\overline{OA}, \overline{AB}, \overline{BP}, \overline{PC}, \overline{CA}, \overline{AO}$: Límite interior del campo visual del sistema de catadióptrico.
- Segmentos $\overline{OA''}, \overline{A''B''}, \overline{B''P''}, \overline{P''C''}, \overline{C''A''}, \overline{A''O}$: Límite exterior del campo visual del sistema de catadióptrico.
- Segmentos $\overline{OA'_1}, \overline{A'_1B'}, \overline{B'P'}, \overline{P'C'}, \overline{C'A'_2}, \overline{A'_2O}$: Eje central del campo visual del sistema de catadióptrico.

- Ángulo $\mathbf{OAB}\angle$: Ángulo entre segmentos de rayo incidente \overline{OA} y rayo reflejado \overline{AB}
- Ángulo $\mathbf{ABP}\angle$: Ángulo entre segmentos de rayo incidente \overline{AB} y rayo reflejado \overline{BP}

Suponga que la localización de los puntos $\mathbf{O}, \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{P}$ es conocida, y dicha información de su posición es dada en coordenadas tridimensionales referenciadas al origen de robot $\{X_0, Y_0, Z_0\}$; de modo que el punto \mathbf{O} poseería una terna de coordenadas representada como $\mathbf{O}\{X_O, Y_O, Z_O\}_0$, el punto \mathbf{A} de igual forma un conjunto de coordenadas $\mathbf{A}\{X_A, Y_A, Z_A\}_0$, y así sucesivamente con todos los demás puntos $\mathbf{B}, \mathbf{C}, \mathbf{P}$. De este modo, todos los segmentos de recta posibles entre los puntos $\mathbf{O}, \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{P}$, serían deducibles.

La segunda parte del análisis consiste en determinar la orientación de los espejos laterales. La orientación de los espejos laterales debe ser congruente con la ley de reflexión; es decir, las normales de dichos espejos son vectores sobre el plano de incidencia de los rayos reflejados, cuya orientación es tal que su separación angular de ambos rayos, incidente y reflejado es $\theta_i = \theta_r$.

Considerando el caso del espejo lateral 1, la normal a este, designada como $\overrightarrow{NEL_1}$, será igual al vector normalizado, coplanar e intermedio, a las normas $\|\overrightarrow{AB}\|$ y $\|\overrightarrow{BP}\|$, de los vectores \overrightarrow{AB} y \overrightarrow{BP} , por tanto:

$$\overrightarrow{NEL_1} = \|\vec{S}\| \quad (3.1)$$

donde:

$$\vec{S} = \frac{(\|\overrightarrow{AB}\| + \|\overrightarrow{BP}\|)}{2}$$

De modo que la ecuación del plano que representase al espejo lateral 1, sería igual al plano fuera del origen, cuya normal fuera dada por el vector unitario $\overrightarrow{NEL_1}$ ($|\overrightarrow{NEL_1}| = 1$), es decir, de la forma:

$$a(x - h) + b(y - k) + c(z - j) = 0 \quad (3.2)$$

donde:

$$\begin{aligned} h &= \mathbf{A}_x & a &= \mathbf{NEL}_{1,x} \\ k &= \mathbf{A}_y & b &= \mathbf{NEL}_{1,y} \\ j &= \mathbf{A}_z & c &= \mathbf{NEL}_{1,z} \end{aligned}$$

De lo anterior se puede deducir fácilmente la orientación del espejo lateral 1, a partir de las componentes del vector \overrightarrow{NEL}_1 . Los ángulos de orientación del espejo lateral 1, se muestran en la figura 3.5.

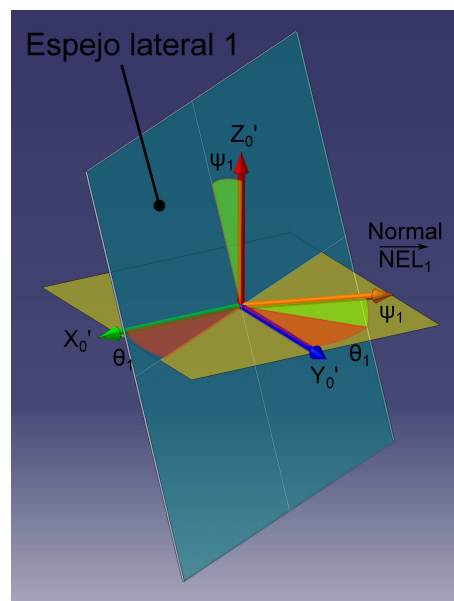


Figura 3.5: Ángulos de orientación del espejo lateral 1.

En la figura anterior, también se muestran el vector normal al espejo \overrightarrow{NEL}_1 , el punto \mathbf{A} del cual surge este vector y un sistema coordenado $\{X'_0, Y'_0, Z'_0\}$, cuyo origen está dado por las coordenadas del punto \mathbf{A} y conserva la misma orientación del sistema coordenado del robot $\{X_0, Y_0, Z_0\}$. Los ángulos θ_1 y ψ_1 , sirven para definir la orientación del espejo, y son medidos a partir de las componentes del vector \overrightarrow{NEL}_1 , dadas en función del sistema coordenado $\{X'_0, Y'_0, Z'_0\}$.

Los ángulos de orientación del espejo lateral 1, θ_1 y ψ_1 , están dados mediante las siguientes expresiones:

$$\theta_1 = \arccos(\|\overrightarrow{NEL}_{1,XY}\| \cdot \hat{U}_y) \quad (3.3)$$

$$\psi_1 = \arccos(\|\overrightarrow{NEL_{1,XY}}\| \cdot \overrightarrow{NEL_1}) \quad (3.4)$$

donde:

$$\overrightarrow{NEL_{1,XY}} = \begin{bmatrix} NEL_{1,x} \\ NEL_{1,y} \\ 0 \end{bmatrix} \quad \hat{U}_y = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Análogamente, la orientación del espejo lateral 2 puede estar descrita por un par de ángulos llamados θ_2 y ψ_2 , definidos a partir del conocimiento de un vector unitario normal a este espejo el cual, análogo al vector normal del espejo lateral 1 ($\overrightarrow{NEL_{21}}$), lleva el nombre de $\overrightarrow{NEL_2}$. Si el espejo lateral 2 adopta una orientación referenciada a un sistema de coordenadas $\{X_0'', Y_0'', Z_0''\}$, paralelo al sistema de coordenadas $\{X_0, Y_0, Z_0\}$ ubicado en la base del robot, el espejo lateral 2 puede ser representado de acuerdo a la figura 3.6.

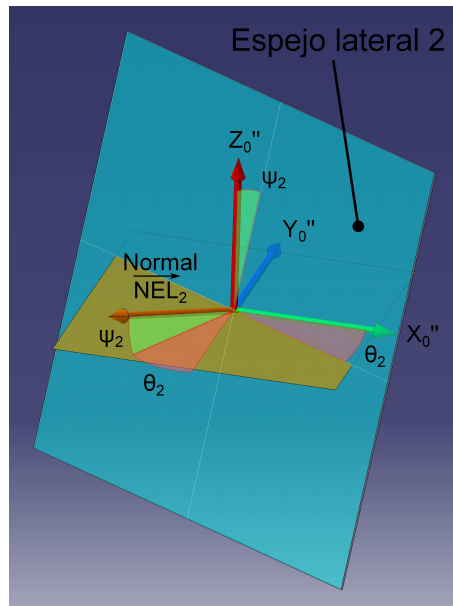


Figura 3.6: Ángulos de orientación de espejo lateral 2.

Los ángulos de orientación del espejo lateral 2, pueden ser calculados como sigue:

$$\theta_2 = \arccos(\|\overrightarrow{NEL_{2,XY}}\| \cdot -\hat{U}_y) \quad (3.5)$$

$$\psi_2 = \arccos(\|\overrightarrow{NEL_{2,XY}}\| \cdot \overrightarrow{NEL_2}) \quad (3.6)$$

donde:

$$\overrightarrow{NEL_{2,XY}} = \begin{bmatrix} NEL_{2,x} \\ NEL_{2,y} \\ 0 \end{bmatrix}$$

Las rutinas computacionales generadas en MATLAB[®], para generar la totalidad del campo visual de sistema catadióptrico de estudio, simulan la reflexión de cada uno de los segmentos de dicho campo visual en los planos correspondientes a los espejos del sistema. Para concretar esta tarea, se parte del conjunto de vectores, que representan los límites del campo visual directamente procedentes de la cámara real. El destino de este primer grupo de vectores, es incidir contra el par de espejos centrales de orientación conocida; es aquí donde es necesario conocer el punto de intersección de cada vector, con el respectivo plano del espejo, al que incide. Posteriormente, acorde a la ley de reflexión para cada vector incidente en los espejos, existe a su vez un vector reflejado. Esto ocurre de igual forma, para cada vector reflejado de los espejos centrales hacia los espejos laterales.

La representación del sistema de cámaras virtuales de la figura 3.4, es únicamente un corte transversal del sistema; en contraparte, en la simulación realizada se describe el comportamiento de dicho sistema de adquisición en tres dimensiones, por lo que la verdadera cantidad de vectores existente, perteneciente a los límites del campo visual, es mucho mayor. En el apéndice B se muestra el modo de obtener el punto de intersección entre un vector y un plano, y a su vez, la forma de obtener el vector reflejado.

A continuación se muestra el gráfico tridimensional representativo de la simulación del sistema de cámaras virtuales de estudio; todas las rutinas programadas para esta tarea, se presentan en el apéndice A.

Simulación de Sistema de Cámaras Virtuales de Cuatro Espejos

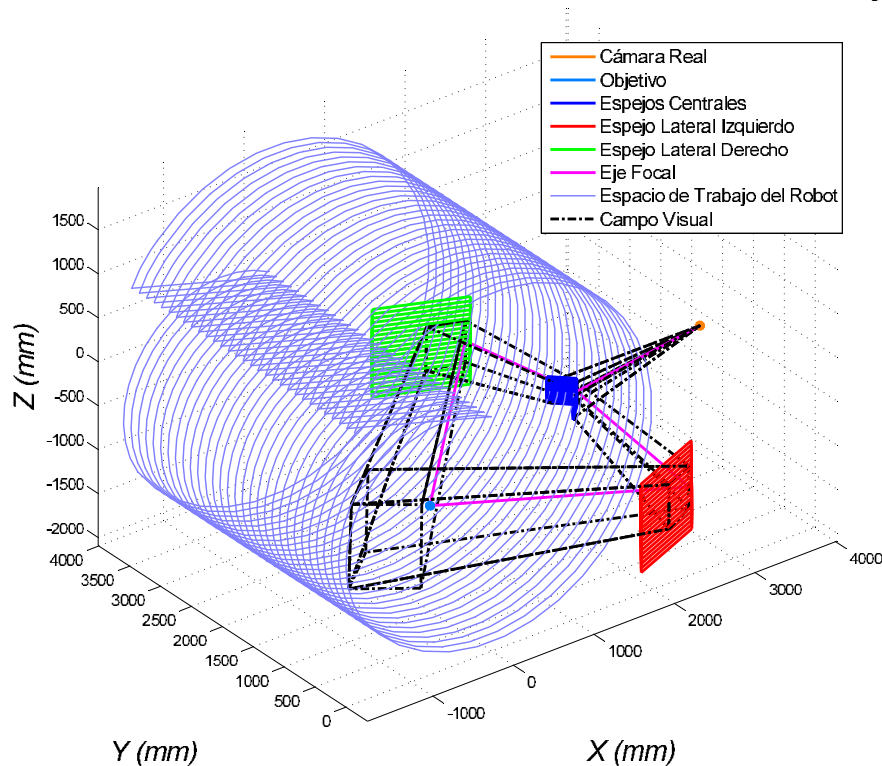


Figura 3.7: Simulación de sistema de cámaras virtuales en Matlab [®].

El dimensionamiento final de los elementos del sistema de cámaras virtuales, se realiza mediante varias simulaciones simultáneas en donde el objetivo \mathbf{P} se coloca en diversas localizaciones dentro del campo de trabajo del robot. Al posicionar dicho punto \mathbf{P} en ubicaciones a los extremos del espacio de trabajo del robot, los espejos laterales se orientan automáticamente para enfocar a dicho punto; sin embargo, entre más extrema sea la posición del punto, por las características del sistema óptico, el espejo lateral más cercano a este punto estará orientado mayormente sesgado a los rayos del campo visual procedentes del espejo central, demandándose así la máxima anchura de dicho espejo para lograr total reflexión del campo visual incidente. La altura del espejo, se determina de forma similar a su anchura, al considerar la máxima elevación o depresión del punto \mathbf{P} , sin embargo, grandes elevaciones no son consideradas puesto que quedan fuera del alcance físico real con el que es posible colocar un objetivo de pruebas.

El dimensionamiento de los espejos centrales del robot se realizó mediante la propuesta de un valor de $D00 = 1,5 m$, para un determinado ángulo de apertura visual de la cámara. El ángulo de apertura visual de la cámara está determinado por su lente (marca INFAIMON modelo OPT- MY31795MC), el cual es de enfoque manual variable, siendo su rango de apertura visual de 2.8 a 22 grados y su formato de imagen de 1/2 pulgada. Los ángulos de apertura visual horizontal y vertical para el lente de la cámara considerados, son de 12 y 9 grados respectivamente.

La localización de los espejos laterales respecto del vértice de los espejos centrales es dada mediante las dimensiones $D01$ y $D02$; cuyos valores son determinados paralelamente al eje Y_0 de la base del robot, desde el punto **A**, en el vértice de los espejos centrales, hasta los puntos **B** y **C** en los ejes de rotación vertical de los espejos laterales, para las distancias $D01$ y $D02$ respectivamente. El valor de $D01$ y $D02$ empleado en la simulación es de $1,8 m$. A continuación se muestra la simulación del sistema óptico de cámaras virtuales para diferentes configuraciones del objetivo **P**.

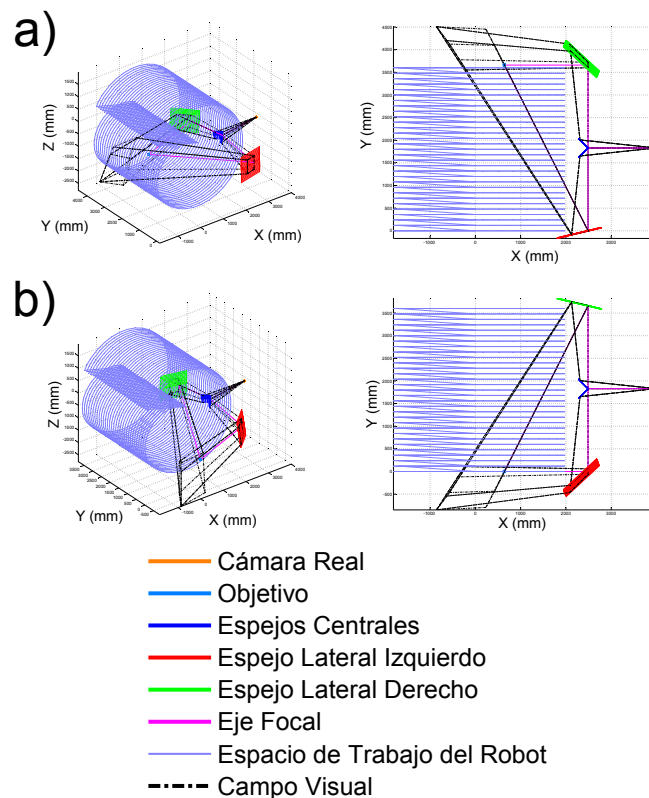


Figura 3.8: Simulación de sistema de cámaras virtuales en diferentes configuraciones; a) Con objetivo en $\mathbf{P}\{640, 3660, -1800\}_0$; b) Con objetivo en $\mathbf{P}\{640, 0, -1800\}_0$.

3.1.3. Construcción del Sistema de Cámaras Virtuales

Con base a la simulación computacional de la sección anterior, se definieron las dimensiones de los elementos del sistema de cámaras virtuales. Las cuales son dadas a continuación:

	Espejo central 1	Espejo central 2	Espejo lateral 1	Espejo lateral 2
Ancho (mm)	300	300	1000	1000
Alto (mm)	300	300	1000	1000

Las dimensiones correspondientes a los espejos laterales, fueron extendidas para facilitar la orientación de los espejos laterales sin perder cobertura del campo visual procedente de los espejos centrales.

La construcción de los elementos reflexivos del sistema, consistió en la fabricación de tres soportes independientes para asegurar la posición de láminas de espejo comerciales. Dos de los soportes, fueron destinados para los espejos laterales y uno adicional para mantener en posición al par de espejos centrales de orientación fija. El posicionamiento de la única cámara de video, se realizó mediante un tripié. El empleo de soportes independientes tiene la finalidad de darle libertad al sistema para ser reconfigurado.

Los soportes fueron fabricados (Figura 3.9) con perfiles cuadrados de madera de 1 – 5/16 pulgadas, lográndose así suficiente robustez y bajo peso, como para manipular y orientar adecuadamente los espejos. Para orientar a los espejos laterales sobre su eje horizontal, puesto que no son requeridos ángulos de rotación demasiado grandes, dicha tarea se logra calzando la base de cada soporte, los cuales posee suficiente estabilidad como para evitar su caída.

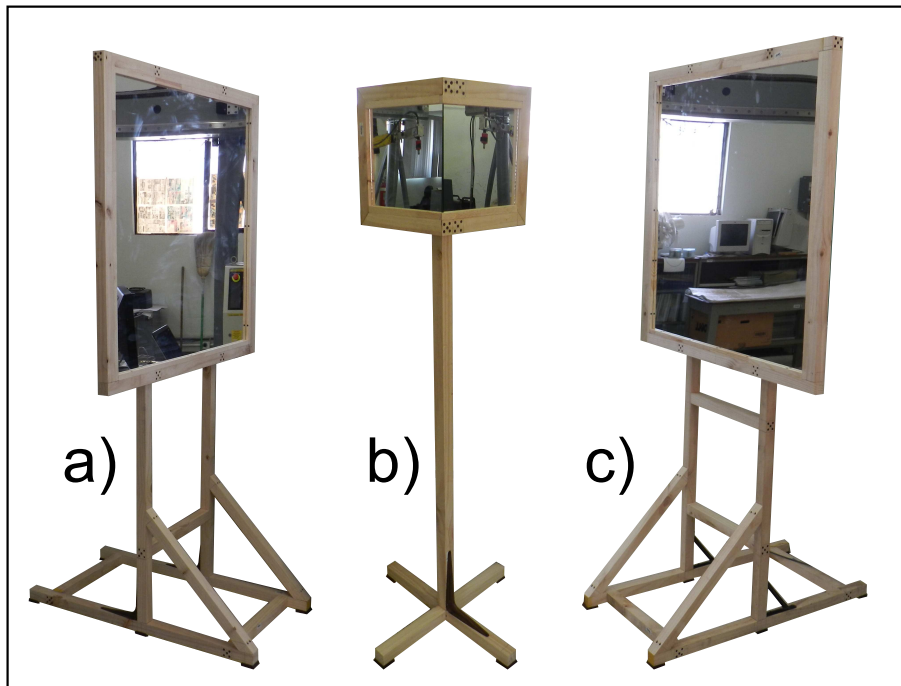


Figura 3.9: Soportes de espejos de sistema de cámaras virtuales; a) Soporte de espejo lateral derecho; b) Soporte de espejos centrales; c) Soporte de espejo lateral izquierdo.

Una imagen típica adquirida mediante el sistema adquisición diseñado es la siguiente:

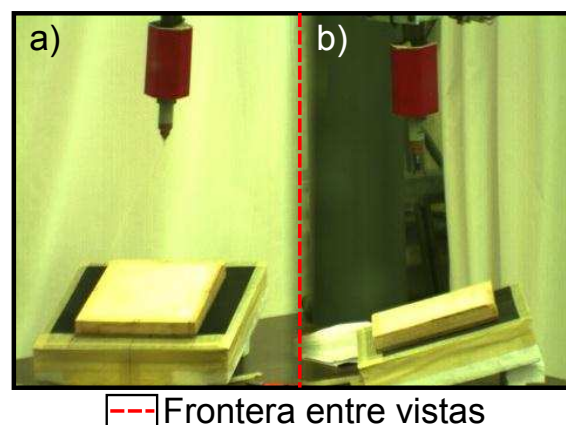


Figura 3.10: Ejemplo de imagen adquirida mediante el sistema de cámaras virtuales diseñado; a) Vista de perspectiva izquierda; b) Vista de perspectiva derecha.

3.2. Equivalencia del Sistema de Cámaras Virtuales Diseñado con un Sistema de Adquisición de Múltiples Sensores

En la sección 3.1, se describe el desarrollo de un sistema de adquisición de imágenes, el cual hace uso de espejos y un solo sensor opto-electrónico para generar un par de cámaras virtuales, con las que es posible obtener la información necesaria requerida por los algoritmos que constituyen la metodología de control por visión *CSM*. Ciertamente el sistema de cámaras virtuales diseñado es equivalente a un sistema adquisición basado puramente en cámaras reales; sin embargo, la equivalencia entre estos exige la mención de las restricción del sistema de cámaras virtuales diseñado, las cuales son enunciadas a continuación.

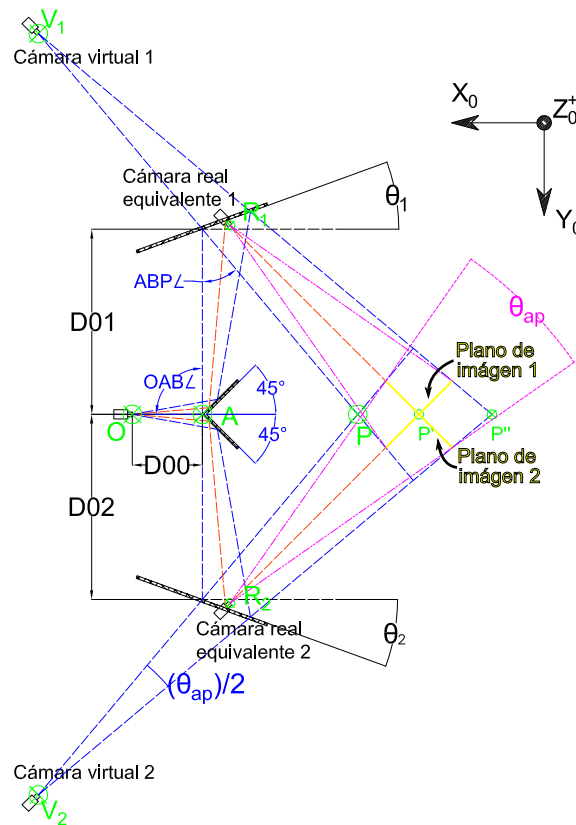


Figura 3.11: Equivalencia entre cámaras virtuales, del sistema catadióptrico de adquisición diseñado, y cámaras reales.

- El ángulo de apertura visual de la cámara real empleada θ_{ap} , es dividido por el par de cámaras virtuales resultantes, teniendo estas la mitad del ángulo de

apertura visual de la cámara real ($\theta_{ap}/2$); de modo que, una cámara virtual resultante de dividir por la mitad el campo visual de una cámara real, para tener la misma cobertura de dicho sensor, necesita ser generada al doble de distancia de un plano de imagen dado que la requerida por el sensor real (Figura 3.11). Afortunadamente las cámaras virtuales son simplemente representaciones de la óptica de un sistema, compuesto de elementos reflexivos y un sensor real; sin embargo, la mencionada limitante tiene influencia sobre las dimensiones del sistema de espejos.

- La resolución de la cámara real utilizada es dividida entre el par de cámaras virtuales resultantes; por lo que la captura de una misma escena empleando cámaras virtuales, será realizada con la mitad de la resolución que al hacerlo directamente con sensores reales de las mismas características que el único empleado en el sistema de cámaras virtuales (Figure 3.12).

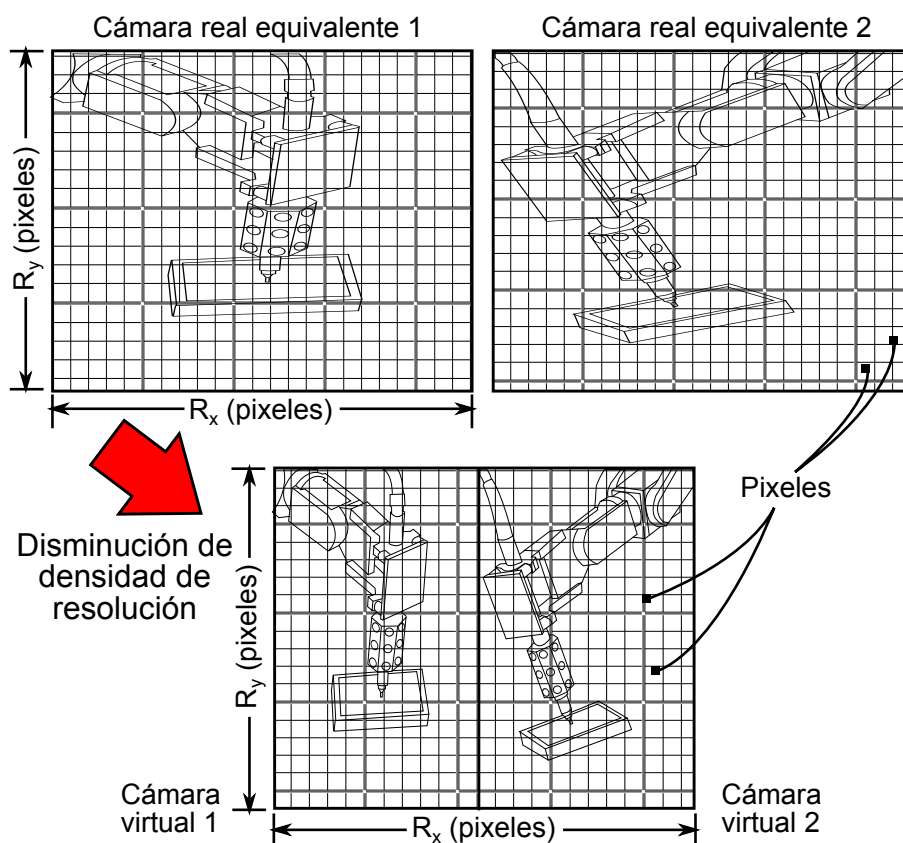


Figura 3.12: Representación de la disminución de la densidad de resolución de las capturas.

De la segunda restricción mencionada del sistema de cámaras virtuales, es necesario abordar un concepto relevante, el cual consiste en la razón existente entre la resolución de la cámara empleada (píxeles) y las dimensiones del campo visual (mm,cm ...) cubierto, ubicado a cierta lejanía del observador; dicho concepto será aquí llamado *densidad de resolución*, y es expresando como el cociente de dividir la cantidad de píxeles que representan la dimensión horizontal o vertical de una imagen captada por una cámara, por la magnitud de la dimensión física real horizontal o vertical de la escena captada, definidas estas dimensiones sobre un plano perpendicular al eje óptico de la cámara.

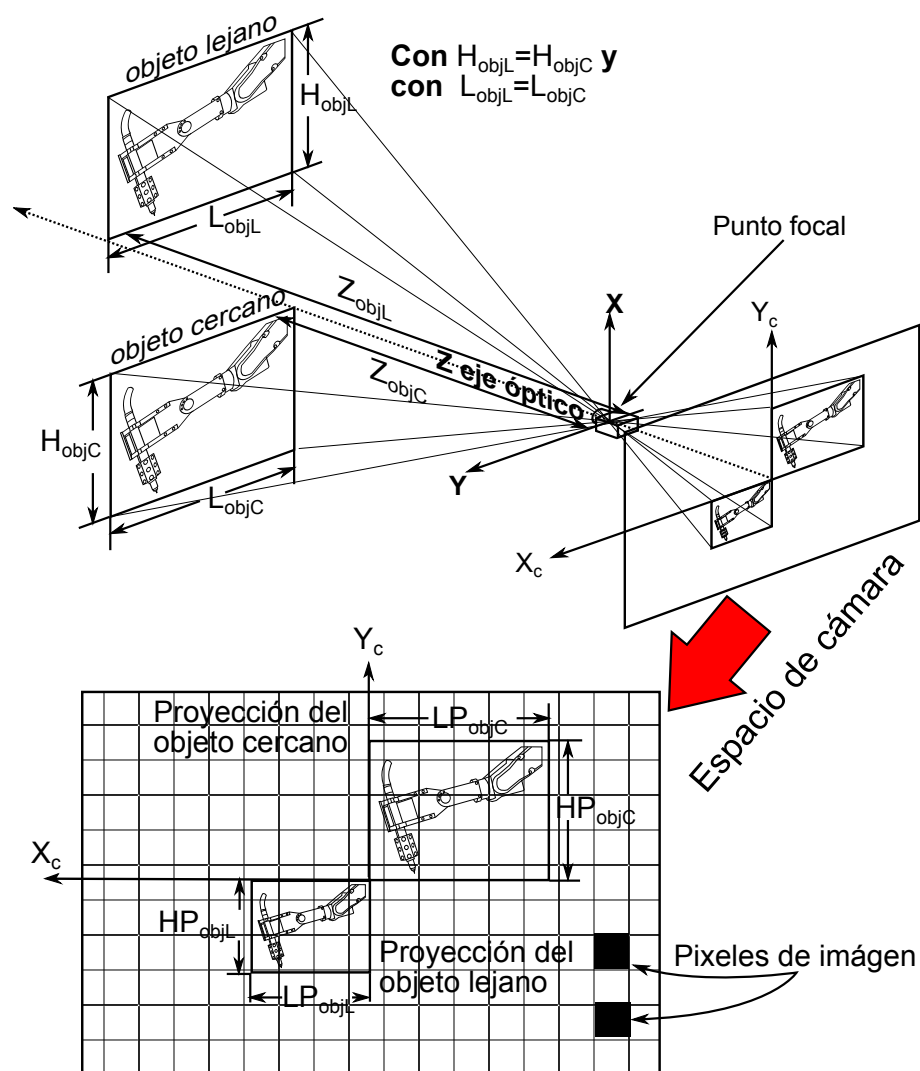


Figura 3.13: Relación de aspecto entre la proyección en espacio de cámara de un cuerpo lejano y un cuerpo cercano de mismas dimensiones.

De acuerdo a la figura 3.13, la resolución específica de un cuerpo se podría definir de la siguiente manera:

$$\text{Densidad de resolución} = \frac{LP_{obj}}{L_{obj}} \quad (3.7)$$

$$(3.8)$$

$$\text{Densidad de resolución} = \frac{HP_{obj}}{H_{obj}}$$

Donde LP_{obj} y HP_{obj} están dados en pixeles y representa la longitud y altura respectivamente de una imagen que contiene a la escena de interés, y donde L_{obj} y H_{obj} , están dados en unidades de longitud (mm,cm ...) y representan la longitud y altura física real de la escena interés; donde las dimensiones de dicha escena son tomadas sobre un plano perpendicular al eje óptico de la cámara. Cabe mencionar que el concepto de densidad de resolución es variable y depende de la lejanía a la cual se encuentran de la cámara los objetos en la escena, siendo mayor entre más cercanos sean los objetos.

Como valor recíproco de la densidad de resolución, se propone además el concepto de *resolución específica*, de forma complementaria.

$$\text{Resolución específica} = \frac{1}{\text{Densidad de resolución}} \quad (3.9)$$

$$(3.10)$$

El concepto de densidad de resolución, debe ser entendido como un valor de magnitud variable y solo es válido para los objetos captados, que se encuentren sobre un mismo plano perpendicular al eje óptico de la cámara. No obstante, dicho valor es relevante cuando se realiza una maniobra de control por visión *CSM*, puesto que a una densidad de resolución elevada, la representación de un objetivo en espacio de cámara será densa, y esto permite un mapeo de coordenadas a espacio tridimensional, más sensible a pequeños desplazamientos del objetivo en espacio de cámara.

Como última restricción del sistema de adquisición de cámaras virtuales, se encuentra que, si se compara a dicho sistema con un sistema de adquisición que emplee

puramente sensores reales, de las mismas características al utilizado en el sistema de cámaras virtuales; en el caso en el que ambos sistemas, deseen captar una escena de dimensiones semejantes, el sistema de cámaras virtuales tendrá una densidad de resolución menor en sus capturas. Sin embargo, el sistema de cámaras virtuales puede ofrecer aun buenos resultados, al ser aplicado a un sistema de control por visión CSM (Figura 3.12).

3.3. Procesamiento de Imágenes

En la sección 2.2 del capítulo anterior, se presenta un esquema de las diferentes etapas que constituyen un sistema de control por visión de forma práctica (figura 2.4); dichas etapas pueden ser clasificadas a su vez en tres niveles, nombrados como visión de bajo, mediano y alto nivel.

El procesamiento de imágenes requerido por el sistema de control por visión CSM aquí empleado, tiene como finalidad extraer de las imágenes adquiridas, la información de espacio de cámara útil para el modelo de visión empleado. No obstante, debido a que el sistema de adquisición emplea una única cámara real, una sola imagen posee la información de dos perspectivas distintas, relativas a un mismo punto de captura; por ello, antes de la extracción de la información de la imagen adquirida, ésta debe de ser separada por zonas, de acuerdo a las áreas que contienen la visualización de cada perspectiva.

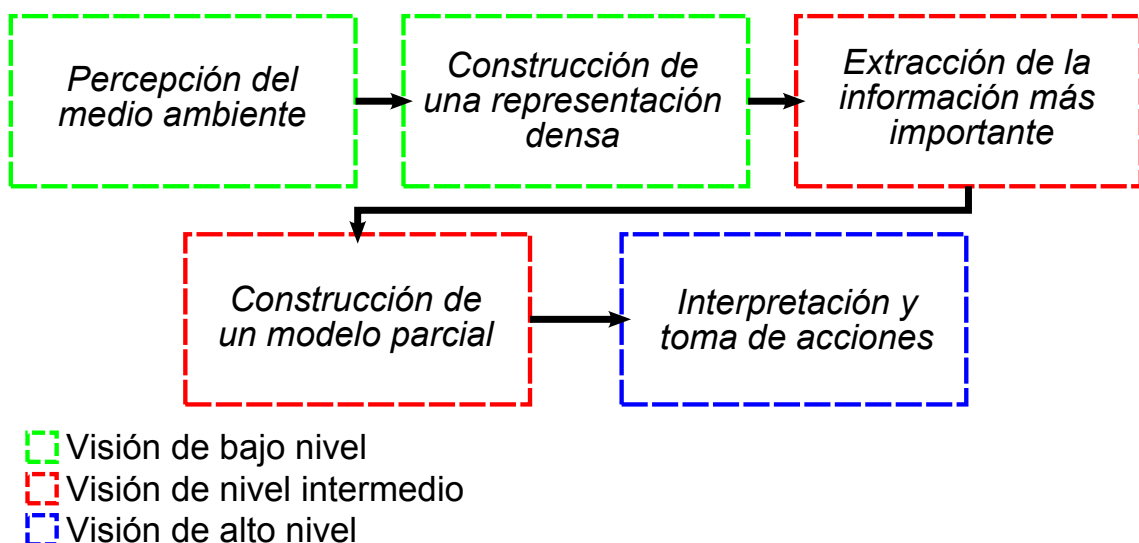


Figura 3.14: Niveles de un sistema de visión.

El procesamiento de imágenes descrito en esta sección, es clasificado como de nivel intermedio de acuerdo al diagrama de la figura 3.14, y esta dado en dos pasos, el primero de ellos llamado *separación de imágenes*, y el segundo nombrado como *detección de marcas artificiales*.

Para realizar el procesamiento de imágenes de forma práctica, se escribió una serie de rutinas computacionales en el lenguaje de programación *C++*, con las que se aprovechó el uso de una biblioteca multiplataforma de código abierto, específicas para el desarrollo de sistemas de visión artificial, llamada *OpenCV* [43, 44]. Esta biblioteca, desarrollada inicialmente por *Intel*®, se encuentra disponible para su descarga gratuita en línea desde 1999 y está conformada más de 500 funciones optimizadas, con las que es posible realizar múltiples aplicaciones de visión robótica y visión artificial que implican procesamiento de imágenes. Más información sobre *OpenCV* está disponible en el sitio web “<http://opencv.willowgarage.com/wiki/>”.

En las secciones siguientes se detalla la forma en la que fue realizado el procesamiento de imágenes requerido previo al control por visión del manipulador industrial mediante *CSM*.

3.3.1. Separación de Imágenes

El tipo de imágenes adquiridas por el sistema de cámaras virtuales, diseñado en la sección anterior, son similares a la que se presenta en la figura 3.10; a la izquierda y a la derecha de la imagen, se tiene una perspectiva diferente del área de trabajo del robot, manteniendo en escena una serie de elementos en común, como lo es la herramienta del manipulador, y una superficie en donde se colocan los objetivos de una maniobra. Al centro de la imagen captada, se puede notar una zona de transición o fronteriza, que es en realidad resultante del vértice que existe entre cada espejo central; esta zona fronteriza separa el par de perspectivas contenidas en la imagen, en regiones de forma aproximadamente rectangular por la verticalidad del vértice central.

Aprovechando la geometría rectangular del par de perspectivas captadas por el sistema catadióptico de adquisición de imágenes, el procedimiento para separar dichas perspectivas es simple. En primera instancia se generan dos copias de la imagen inicialmente captada, donde cada una de las copias representará cada perspectiva diferente. Posteriormente, en cada una de las copias realizadas, con la ayuda de la

biblioteca de *OpenCV*, se dibuja un rectángulo negro sobre la región externa a la perspectiva de interés; para trazar dicho rectángulo solo es necesario conocer el lugar en el cual se encuentra la frontera entre ambas perspectivas, lo cual puede ser decidido por un usuario con la ayuda del ratón de la computadora y una interfaz gráfica.

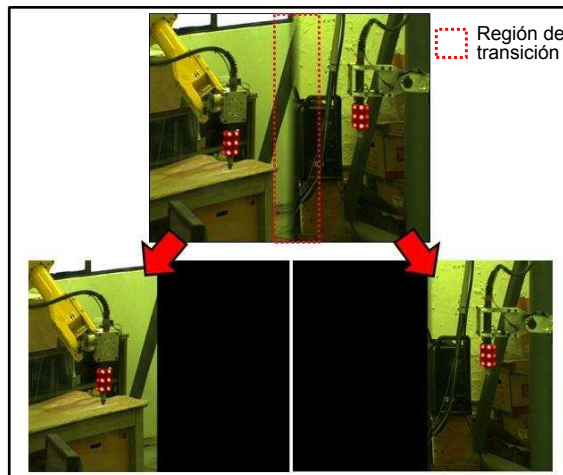


Figura 3.15: Proceso de separación de imágenes.

La representación de una imagen en un ordenador, es similar a la de una gran matriz, en donde cada pixel en la imagen representa algún valor en un elemento de la matriz. Si la imagen es a blanco y negro, dicha matriz posee dimensiones iguales a la resolución de la imagen (*Filas* \times *Columnas*); si la imagen es a color, por ejemplo de tres canales en formato rojo-verde-azul (*RGB por sus siglas en inglés*), la matriz poseerá el triple de columnas que las dadas por la resolución de la imagen, debido a que cada pixel de la imagen a color es representado por la combinación de tres sub-pixeles de colores rojo, verde y azul, donde a cada uno de los tonos de estos colores, le corresponde un número dentro de su representación matricial. El intervalo de valores posibles para los números en la matriz, que representan los pixeles de una imagen, está definido por la “*profundidad de color*” de la imagen, dado en bits; si la imagen es de 8 bits de profundidad, cada pixel o sub-pixel (si la imagen es a color), posee un valor cuyo rango numérico es de $\{0 \dots (2^8 - 1)\}$ o bien de $\{0 \dots 255\}$, donde el cero representa la ausencia de color (negro), y el 255 la mayor cantidad de color. Una de las ventajas del uso de la biblioteca *OpenCV*, es que permite trabajar con imágenes como si se tratase de matrices, siendo posible realizar una gran cantidad de operaciones basadas en álgebra lineal.

Durante el proceso de separación de imágenes, el tamaño de las copias de la imagen original captada no cambia y el hecho de dibujar un rectángulo negro para delimitar cada perspectiva de interés, equivale a sustituir por cero al valor de los píxeles en la zona del rectángulo; de esta forma, aquellos píxeles nulificados no generan influencia en el procesamiento posterior.

La función en las librerías de *OpenCV*, con la cual es posible dibujar un rectángulo negro en un imagen, es llamada `cvRect` [43], cuyos datos de entrada son la imagen en donde se hará el trazo, la esquina superior izquierda e inferior izquierda del rectángulo, y su color en el formato de la imagen correspondiente (*RGB* si es a color).

3.3.2. Detección de Marcas Artificiales

Como se mencionó anteriormente en el capítulo 2, la metodología de *CSM* provee de modelos de visión que, basados en una serie de parámetros de cámara, son capaces de generar el mapeo de coordenadas de un punto en espacio tridimensional, con sus respectivas coordenadas en espacio de cámara o espacio de imagen. No obstante, durante el transcurso de una maniobra de control por visión con *CSM*, los parámetros de los modelos de visión empleados deben de ser enriquecidos, lo cual se logra con la toma de muestras desde espacio de cámara, de una serie de marcas visuales de posición tridimensional conocida; donde la toma de estas muestras consiste del aislamiento de cada marca visual y del cálculo de las coordenadas de sus centroides dado en píxeles. Las marcas visuales artificiales en el caso práctico, son *LEDs* de alta intensidad adheridos a las caras de un cuerpo prismático hexagonal, ubicado en la herramienta del manipulador; el encendido de los *LEDs* es controlado, de acuerdo al momento en el que se desea que éstos sean visibles. Para definir el objetivo de una maniobra de interés, también se emplean marcas visuales, sin embargo, la fuente luminosa es distinta a los *LEDs*, puesto que en este caso se emplea un emisor láser para generar proyecciones de puntos luminiscentes, visibles mediante un sensor opto-electrónico.

El proceso de detección de marcas visuales se basa en lo que se conoce como diferenciación de imágenes. Si una imagen es considerada como una matriz cuyos elementos numéricos representan los píxeles de la imagen, entonces dos imágenes de dimensiones iguales pueden ser restadas una de otra, de forma similar a la resta de dos

matrices; dando origen a una nueva imagen cuyos pixeles son el residuo de la *diferenciación* de los pixeles de las dos imágenes iniciales (Figura 3.16). Si las dos imágenes inicialmente restadas, provienen de una cámara que se ha mantenido inmóvil entre tomas, preservándose los elementos de la escena captada, en condiciones ideales o en ausencia de ruido entre capturas, la imagen resultante sería totalmente negra; si algunos elementos de la escena cambiaron de posición entre cada captura, entonces la imagen resultante será negra evidenciando, con manchas de colores, aquellos lugares donde la primera imagen difiere de la segunda.

$$|[\text{IMG}_1 - \text{IMG}_2]| = |[\text{IMG}_3]|$$

Figura 3.16: Ejemplo de diferenciación de imágenes.

Puesto que la visibilidad de las marcas visuales existentes en escena es controlable, ya sea al encender o apagar *LEDs* o un emisor láser, el método de diferenciación de imágenes es útil como un primer paso para detectar el total de las marcas. Al diferenciar dos imágenes de una misma escena, en donde exista una primera toma con marcas visuales presentes (Figura 3.17a) y una segunda con marcas ausentes (Figura 3.17b), se obtendrá una imagen residual que contendrá puramente marcas visuales aisladas en un fondo negro u oscuro. Sin embargo, al aplicar esta técnica en la práctica, es necesario algún método de filtrado para eliminar el ruido entre capturas, ya que dos tomas simultáneas de una cámara, aunque estas provengan de una misma escena cuyos elementos y luminosidad hayan permanecido estáticos, poseen entre si ligeras variaciones, que son notorias al comparar pixel por pixel de una y otra captura (Figura 3.17c). El tipo de filtro que fue empleado en la práctica, es similar a un filtro de banda, es decir, un filtro cuyos umbrales inferior y superior definen un rango de valores para la selección de pixeles válidos; sin embargo, dicho filtro únicamente ofrece una salida binaria, es decir, el valor de los pixeles descartados será cero y aquellos pixeles aceptados toman el valor de 1, o más específicamente de 255 por tratarse de imágenes de 8 bits de profundidad. Debido a que las imágenes procesadas provenientes de la cámara de video son a color (*RGB*), se emplearon tres

filtros simultáneamente, uno por cada canal o color de la imagen, resultando una imagen en blanco y negro de un solo canal, cuyos píxeles toman el valor de 0 o 255 (Figura 3.17d). El filtro utilizado se tomó de una de las funciones de la biblioteca de *OpenCV* y se encuentra como `cvInRangeS` [43].

Tras aislar las marcas visuales de todos los elementos del “ambiente” de una captura, la información útil de dichas marcas es la concerniente a su posición en espacio de cámara, es decir, las coordenadas de sus centroides (Figura 3.17e). Existen varios métodos para calcular el centroide de una marca visual cuando ésta ha sido aislada en pantalla, aquí se mencionarán tres de ellos, los cuales son similares al cálculo del centro de masa de un cuerpo en dos dimensiones, en donde el cuerpo en cuestión está conformado por sub-elementos, que no son otra cosa que píxeles de intensidad y posición conocida.

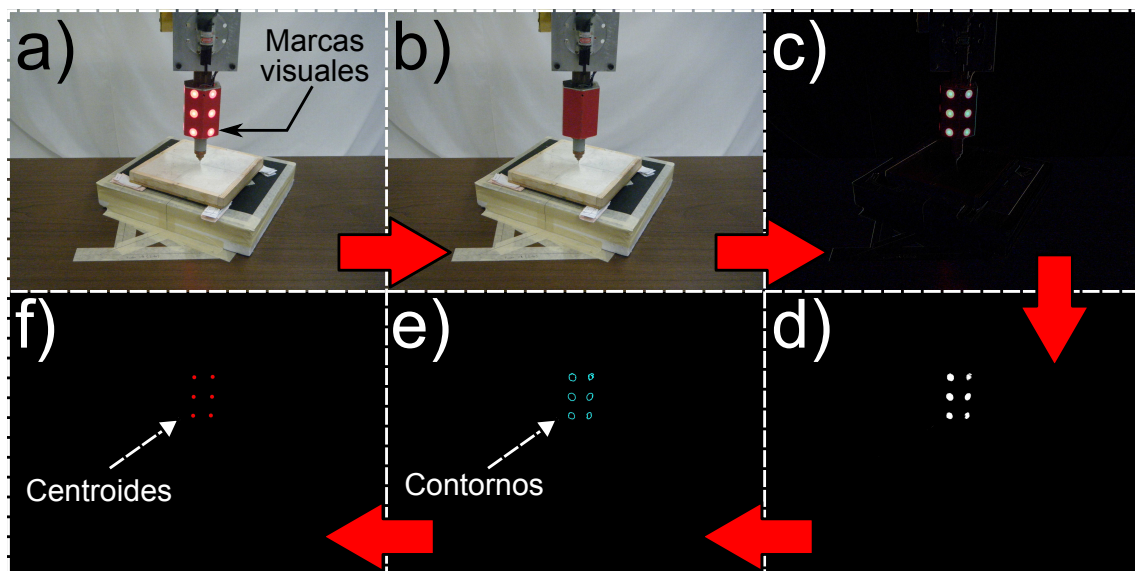


Figura 3.17: Etapas del procesamiento de imágenes, necesario para el aislamiento y cálculo de los centroides de múltiples marcas visuales; a) imagen con marcas visuales visibles; b) imagen con marcas visuales ausentes; c) imagen diferenciada con ruido; d) imagen filtrada; e) imagen con contornos de marcas visuales; f) imagen con centroides de marcas visuales.

El primer método para el cálculo del centroide de una marca visual, es similar al cálculo del centro de masa de un cuerpo de densidad uniforme en dos dimensiones, es decir, todos los píxeles que conforman la marca en la imagen se consideran de la misma intensidad. El segundo método es similar al cálculo del centro de masa de un

cuerpo bidimensional de densidad no uniforme, es decir, cada pixel en la marca visual aporta su verdadera intensidad al cálculo del centroide. Finalmente, el tercer método para obtener el centroide de una marca visual consiste en calcular el centro de masa de la marca en cuestión, empleando únicamente los pixeles del contorno de la marca, donde la intensidad de éstos es considerada constante. Este último método ha sido elegido, por ser adecuado para las características del filtrado de la imagen, además de que la biblioteca de *OpenCV* tiene herramientas disponibles para el cómputo de contornos en una imagen.

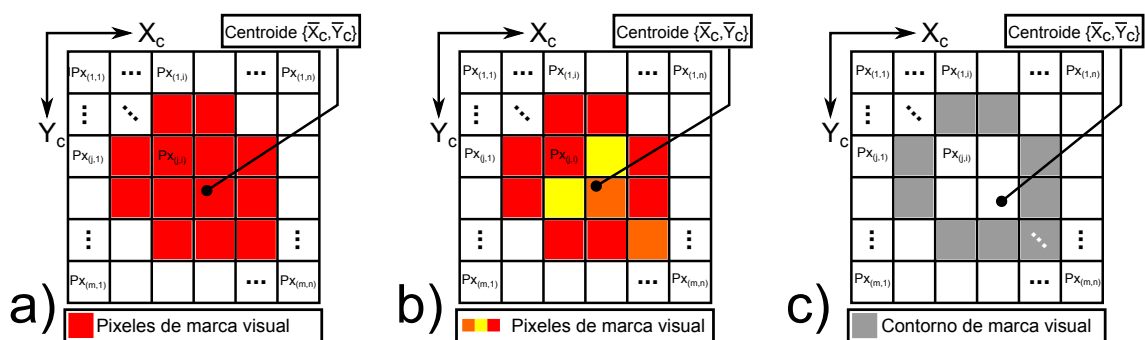


Figura 3.18: Ejemplos de marcas visuales cuyo centroide es calculado mediante diferentes métodos; a) Centroides de marca con pixeles de intensidad uniforme; a) Centroides de marca con pixeles de intensidad no uniforme; a) Centroides de marca por contorno;

En la figura 3.18 se ejemplifican marcas visuales cuyos pixeles son dados de acuerdo al método de cálculo de su centroide; únicamente en este ejemplo, los pixeles en color blanco representan pixeles cuya intensidad es cero.

A continuación se muestran los procedimientos para el cálculo de las coordenadas de los centroides de marcas visuales $\{\bar{X}_c, \bar{Y}_c\}$, atendiendo a diferentes enfoques. Las fórmulas presentadas (Ecuaciones 3.11,3.12,3.13), son útiles en el caso de haber una sola marca visual disponible en pantalla; si más marcas visuales estuviesen presentes, cada marca visual debe ser aislada de las demás marcas en pantalla para calcular su centroide.

- Enfoque con pixeles de intensidad constante:

$$\begin{aligned}\bar{X}_c &= \frac{\sum_{j=1}^m \sum_{i=1}^n (Px_{i,j})}{(m)(n)} \\ \bar{Y}_c &= \frac{\sum_{j=1}^m \sum_{i=1}^n (Py_{i,j})}{(m)(n)}\end{aligned}\quad (3.11)$$

donde:

$Px_{i,j}$ = Posición en “X” del i-j-ésimo pixel

$Py_{i,j}$ = Posición en “Y” del i-j-ésimo pixel

- Enfoque considerando la intensidad de cada pixel:

$$\begin{aligned}\bar{X}_c &= \frac{\sum_{j=1}^m \sum_{i=1}^n (Px_{i,j} I_{i,j})}{\sum_{j=1}^m \sum_{i=1}^n (I_{i,j})} \\ \bar{Y}_c &= \frac{\sum_{j=1}^m \sum_{i=1}^n (Py_{i,j} I_{i,j})}{\sum_{j=1}^m \sum_{i=1}^n (I_{i,j})}\end{aligned}\quad (3.12)$$

donde:

$I_{i,j}$ = Intensidad del i-j-ésimo pixel

- Enfoque considerando el contorno de la marca:

$$\begin{aligned}\bar{X}_c &= \frac{\sum_{i=1}^N (Px_i^{cont})}{(N)} \\ \bar{Y}_c &= \frac{\sum_{i=1}^N (Py_i^{cont})}{(N)}\end{aligned}\quad (3.13)$$

donde:

N = Número de pixeles de contorno

Px_i^{cont} = Posición en “X” de i-ésimo pixel del contorno

Py_i^{cont} = Posición en “Y” de i-ésimo pixel del contorno

En caso del cálculo del centroide de una marca visual a partir de su contorno, la ecuación 3.13 es válida cuando se consideran los puntos pertenecientes exclusivamente al contorno de la marca. Para encontrar el contorno de las marcas visuales de interés, se empleó una función de *OpenCV* conocida como `cvFindContours` [43], la cual es capaz de calcular todos los contornos presentes en una imagen ya sea a blanco y negro o a color. La característica principal de un contorno es que este constituye el perímetro cerrado de alguna geometría dentro de la imagen. Al emplear la función `cvFindContours` de *OpenCV*, los contornos en la imagen son calculados y separados en bloques de datos, cuyos elementos están ordenados en estructuras compuestas de un par de variables, cada una como si se tratase de las coordenadas bidimensionales de un punto; de este modo, para cada contorno encontrado representado por un bloque de datos, es sencillo el cálculo de su centroide mediante la ecuación 3.13, considerando los elementos o puntos que componen a dicho contorno.

El algoritmo con el cual se calculan los contornos de las marcas visuales mediante la biblioteca de *OpenCV*, es un algoritmo clásico de visión artificial, conocido como algoritmo de *código de cadena de Freeman* [45], el cual fue ideado en 1961 por el Dr. Herbert Freeman. Este algoritmo primeramente, aprovecha la existencia de otros algoritmos para la búsqueda de *bordes* en imágenes, considerándose como borde a una región donde existe una variación significativa en la magnitud de los píxeles de una vecindad. Cuando los bordes en una imagen han sido encontrados, el algoritmo de *código de cadena de Freeman* procede a definir todos los contornos posibles a partir de los bordes calculados.

Los criterios para la localización de bordes en una imagen, se basan en casos particulares de lo que se conoce comúnmente en matemáticas como *convolución*, que es básicamente el cálculo de una función resultante del traslape de otras dos funciones; en visión por computadora, una de estas funciones es la imagen a procesar, la segunda función se conoce como *kernel de convolución*, y la función de resultante, es la imagen transformada a partir de la definición del *kernel* [45]. Los algoritmos para la búsqueda de bordes, también se conocen como filtros de bordes y ejemplos de estos algoritmos son los filtros basados en gradientes, como el filtro de *Sobel* y *Scharr*, y los filtros basados en laplacianos como el filtro simple de *Laplace* y el llamado *detector de bordes de Canny*. El filtro de Canny, (desarrollado por John F. Canny en 1986 [46]), es el filtro empleado por la función `cvFindContours` de *OpenCV* para la búsqueda de contornos.

El algoritmo de *código de cadena de Freeman*, se basa en la decodificación del contorno de una geometría a partir de una cadena de base octal, es decir, compuesta solo de dígitos del cero al siete. Suponga que se han encontrado todos los bordes de una imagen, mediante alguno de los filtros mencionados en el párrafo anterior; la imagen resultante es por lo regular una imagen binaria, cuyo fondo es negro y los bordes se presentan como múltiples líneas de color blanco. El algoritmo de Freeman, plantea el análisis de los bordes de la imagen, como si se tratase del viaje de un cursor exclusivamente por sobre aquellos pixeles que componen los bordes de la imagen; dicho cursor comienza su recorrido a través de un borde en una posición o pixel inicial, si el cursor finaliza su recorrido en el mismo punto en el que comenzó, entonces se ha descubierto un contorno, de otra forma, se trata únicamente de un borde. La dirección de navegación del cursor se da mediante el análisis de los ocho pixeles circundantes a este, en un orden definido, ya sea en contra o a favor de las manecillas del reloj; cada pixel vecino es examinado y el primero de ellos en pertenecer al borde o ser de color blanco, definirá la nueva posición del cursor. El cursor puede desplazarse en ocho posibles direcciones, a cada una de estas direcciones le es asignado un número del cero al siete, de modo que cuando la posición el cursor cambia, el trayecto seguido es registrado y codificado en una secuencia de números del cero al siete. El largo de la secuencia es igual al número de pixeles que componen al contorno desde que este inicia, hasta que el cursor vuelve a su punto de partida. El algoritmo de *código de cadena de Freeman* se ejemplifica a continuación (Figura 3.19).

Una vez que se han encontrado los contornos de todas las marcas visuales en la imagen, es posible que a pesar del proceso de filtrado de color anterior, hayan pasado aún a la imagen filtrada partículas que no correspondan a las marcas visuales de interés, pero que de igual forma, hayan sido procesadas por la función de *OpenCV* de `cvFindContours` como si estas fueran marcas visuales. Por fortuna, la cantidad de pixeles de las partículas o manchas “*fugitivas*” es bastante inferior al de las marcas visuales reales, lo que se traduce en que estas poseen contornos de un reducido número de elementos, permitiendo así lidiar con el problema al seleccionar únicamente contornos de un mínimo dado de elementos, para la etapa posterior de cálculo de centroides.

Todos lo algoritmos para procesamiento de imágenes realizado, se resumen en un diagrama de flujo en la siguiente sección.

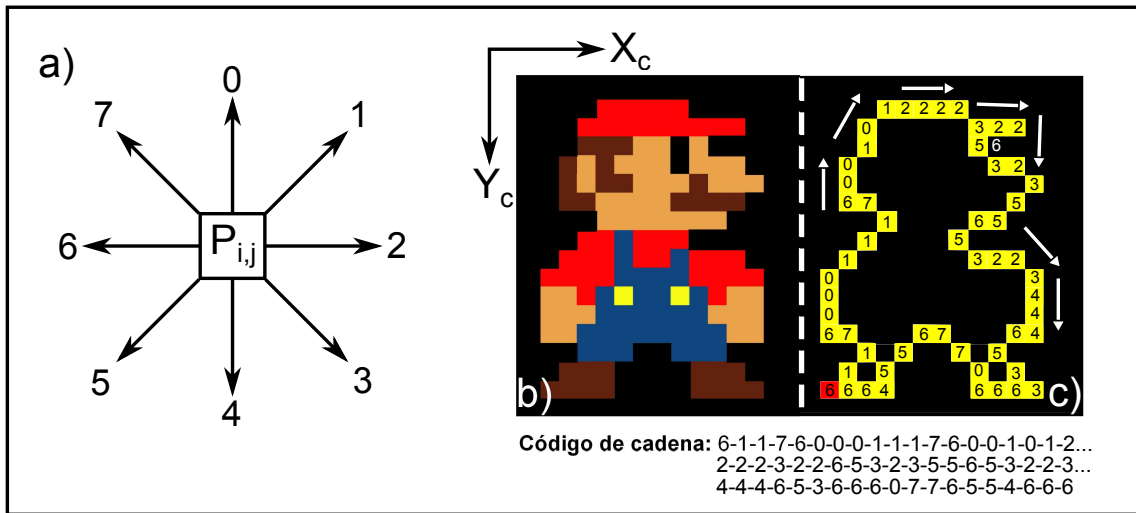


Figura 3.19: Ejemplificación del algoritmo de Freeman para la búsqueda de contornos; a) Numeración y direcciones del cursor; b) Imagen inicial; c) Contorno de imagen encontrado a través del algoritmo de Freeman.

Cuando los centroides de las marcas visuales han sido encontrados, es importante aclarar que su origen de referencia es la esquina superior izquierda de la imagen captada; no obstante, este no es un origen de coordenadas coherente con la definición de los modelos de control por visión de *CSM*, los cuales consideran al origen de coordenadas en un espacio de cámara dado como el centro de la imagen, por lo que las coordenadas de todos los centroides calculados en algún instante, deben de ser redefinidas con respecto al centro de la imagen captada por la cámara virtual en cuestión. Si la posición y orientación del nuevo origen de coordenadas de la imagen es conocido, respecto del antiguo origen de coordenadas, entonces es posible redefinir fácilmente las coordenadas de un centroide, ahora con respecto del nuevo sistema coordenadas, mediante una matriz homogénea de rotación-traslación. La posición del nuevo sistema de coordenadas, está dada al centro de la región visible de cada imagen producto de la separación de perspectiva, mientras que la orientación está definida por un sistema de coordenadas de mano derecha, cuyos ejes “X” y “Y”, son los ejes horizontal y vertical respectivamente (Figura 3.20). La forma de calcular la nueva posición y orientación de los centroides es la siguiente:

$$\begin{bmatrix} \bar{X}_c^{nuevo} \\ \bar{Y}_c^{nuevo} \\ 1 \end{bmatrix} = \mathbb{H}_{nuevo \rightarrow viejo} \begin{bmatrix} \bar{X}_c^{viejo} \\ \bar{Y}_c^{viejo} \\ 1 \end{bmatrix} \tag{3.14}$$

donde:

$$H_{nuevo \rightarrow viejo} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -PSCx^{viejo} \\ 0 & 1 & -PSCy^{viejo} \\ 0 & 0 & 1 \end{bmatrix}$$

con:

$PSCx^{viejo}$ = Posición en "X" del nuevo sistema de coordenadas respecto del viejo

$PSCy^{viejo}$ = Posición en "Y" del nuevo sistema de coordenadas respecto del viejo

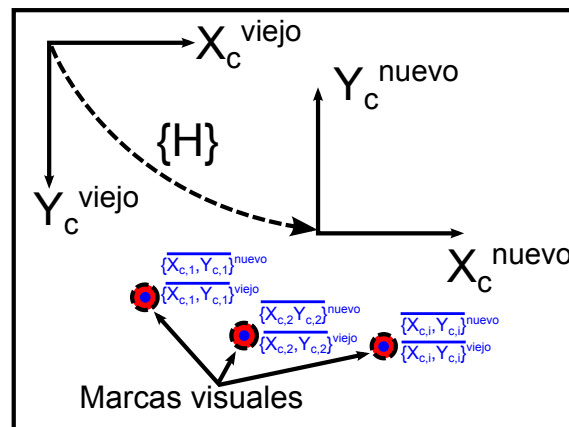


Figura 3.20: Cambio de origen de coordenadas para los centroides de las marcas visuales.

3.3.3. Diagrama de Flujo del procesamiento de imágenes

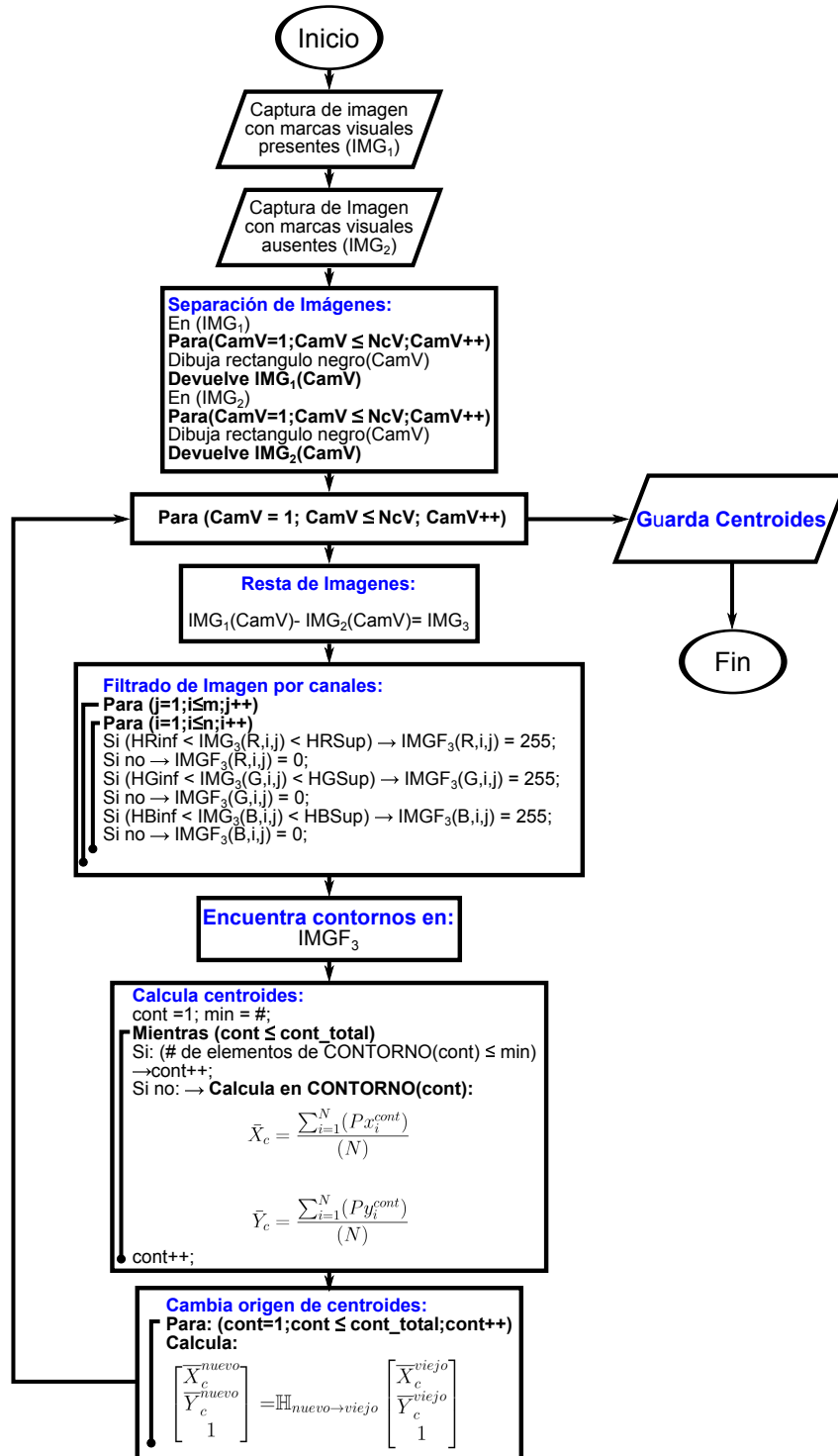


Figura 3.21: Algoritmo de general procesamiento de imágenes, requerido para el sistema de cámara virtuales diseñado e implementado en un sistema de control CSM

3.4. Maniobra de Posicionamiento

En las secciones anteriores del presente capítulo, se relató lo concerniente al proceso de concepción de un sistema óptico el cual, basado en espejos, pueda permitirle a una sola cámara de video la adquisición de dos perspectivas distintas de una misma escena del espacio, aprovechando la generación de observadores virtuales a partir de los espejos del sistema óptico. Para dicho sistema óptico, estrictamente del tipo catadióptrico, también se desarrolló un proceso para el análisis de imágenes, el cual tiene como finalidad adecuar y extraer información de las capturas del sistema óptico, de modo que la información obtenida sea útil para un sistema de control de robots industriales mediante *CSM*.

La forma de validar la funcionalidad de un sistema de adquisición de información visual basado en cámaras virtuales, aplicado al control por visión *CSM* de un robot industrial, es mediante la ejecución de un experimento típico conocido como *maniobra de posicionamiento*. Una maniobra de posicionamiento es una tarea relativamente simple en la que se le ordena a un robot industrial a alcanzar un objetivo designado, donde la información acerca de la posición del objetivo proviene exclusivamente de espacio de cámara. Durante el proceso de ejecución de la maniobra, se controla tanto la posición como la orientación del efector final del manipulador o de su herramienta, siendo imprescindibles todos los algoritmos descritos en el capítulo 2 de esta tesis.

En los apartados de la presente sección, se describirán cada una de las etapas necesarias para el desempeño de una maniobra de posicionamiento efectuada por manipulador industrial, cuyo control este basado en la metodología de control por visión de *CSM*.

3.4.1. Ejecución de Trayectoria Preplaneada

El modelo de visión empleado en el presente proyecto de investigación mediante el enfoque de la metodología de *CSM*, es el modelo de perspectiva recursivo. Dicho modelo construye la correlación existente entre las coordenadas tridimensionales un punto ($\{X_i, Y_i, Z_i\}$) y sus coordenadas de espacio de cámara ($\{X_{c,i}, Y_{c,i}\}$), mediante el modelado de una proyección de perspectiva, que es función siete parámetros pertenecientes al modelo de cámara. Donde las coordenadas tridimensionales del punto

de interés, están dadas con referencia a un sistema coordinado arbitrario, que puede ser el origen de coordenadas de un manipulador.

Para comenzar una tarea específica de control por visión mediante *CSM*, de cualquier clase de dispositivo, es esencial el conocimiento de los parámetros iniciales de cámara del modelo de visión empleado; siendo el número total de parámetros requeridos igual al número de parámetros que constituyen el modelo, por el número de sensores u observadores (N_c) involucrados en la tarea de control. En otras palabras, es necesario conocer N_c modelos de cámara (uno por cámara u observador), compuestos cada uno de siete parámetros por tratarse del modelo de perspectiva recursivo. Cuando el dispositivo a controlar por visión mediante *CSM* es un robot industrial, son necesarios cuando menos dos cámaras u observadores ($N_c \geq 2$), ya que la reconstrucción de la posición tridimensional de un punto a partir de información de espacio de cámara, requiere de por lo menos dos perspectivas diferentes del mismo punto.

La forma de conocer los parámetros iniciales de cámara de cada uno de los observadores, en este caso cámaras virtuales, es mediante un muestreo previo de marcas visuales en espacio de cámara, cuya posición tridimensional también es conocida. En dicho muestreo, el manipulador industrial a controlar ejecuta un rutina de movimientos, en los que una sola marca visual presente en la herramienta del robot, es captada en espacio de cámara. Puesto que la posición de la punta herramienta del manipulador es conocida en todo momento mediante su cinemática directa, por definición también lo es cualquier marca u objeto adherido a la herramienta, siempre y cuando la posición de éste también esté definida con respecto a la punta de la herramienta. Durante la secuencia de movimientos pre-programados, tanto la posición física ($\{X_i, Y_i, Z_i\}$) como la posición en espacio de cámara ($\{X_{c,i}, Y_{c,i}\}$) de la marca visual es almacenada, sumando un número total de muestras igual a la cantidad de movimientos ejecutados en la secuencia. A dicha secuencia de movimientos del manipulador se le conoce como *trayectoria preplaneada*, y su importancia radica en que gracias a ésta, se obtiene la información suficiente para una primera estimación de los parámetros de cámara de cada observador, ya sea virtual o real según el caso. La ejecución de trayectorias pre-planeadas puede ser individual para cada cámara virtual o conjunta, siempre y cuando para ambas cámaras sea visible la marca visual de interés.

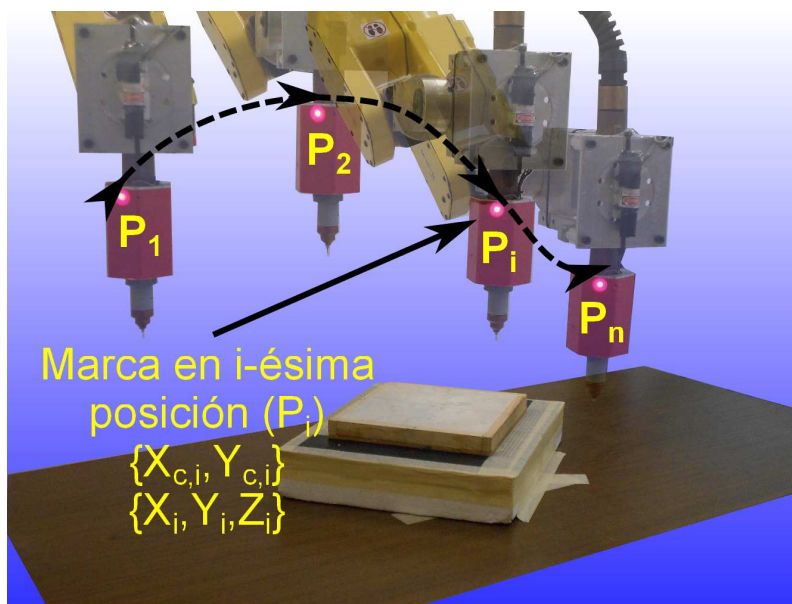


Figura 3.22: Trayectoria preplaneada

La razón por la cual durante una trayectoria preplaneada se muestrea una sola marca visual a la vez, es debido a que hasta este punto, no hay forma de correlacionar cada marca visual en espacio de cámara, con su respectiva posición tridimensional.

Toda información recopilada durante una trayectoria preplaneada es almacenada en las matrices de momentos $[M_A]$ y $[M_B]$, descritas en la sección 2.4.1 del capítulo anterior, para posteriormente calcular los parámetros iniciales de cámara del modelo de perspectiva recursivo.

3.4.2. Definición del Objetivo de una Maniobra de Posicionamiento y Caracterización de Superficies

La metodología de control por visión de *CSM* es capaz de dotar de flexibilidad y autonomía a un robot industrial, aprovechando la información procedente de capturas realizadas por cámaras de video, es decir, de espacio de cámara. La mencionada flexibilidad se refiere a la facilidad con la cual el dispositivo controlado puede ser re-programado para realizar determinadas tareas, o bien, para desplazarse hasta un objetivo. Mediante *CSM*, los objetivos de una maniobra se designan sencillamente desde espacio de cámara, simplemente con seleccionar un punto de interés dentro del espacio de trabajo del robot, que sea visto por un mínimo de dos cámaras de

video ($N_c \geq 2$). El punto de interés puede ser alguna característica distintiva de un objeto, o la proyección de un haz de luz láser sobre una superficie.

El proceso de definición del objetivo de una maniobra de posicionamiento de un robot por *CSM*, se compone de dos importantes procedimientos. El primero de ellos es el procesamiento de imágenes necesario para aislar el objetivo de todos los elementos presentes en la imagen, y calcular las coordenadas en espacio de cámara de su centroide (Sección 3.3.2); hecho esto para un mínimo de dos cámaras.

El segundo paso para la designación de un objetivo de maniobra consiste en la estimación de la posición tridimensional de éste (Sección 2.5), con base en su información, procedente de los distintos espacios de cámara. Como las coordenadas tridimensionales del objetivo son estimadas con respecto a la base del manipulador, en efecto se podría ordenar al manipulador a arribar simplemente al punto con una orientación propuesta; dependiendo de la validez de los parámetros de cámara, se obtendría una aceptable aproximación al objetivo. Sin embargo, la metodología de *CSM* provee de las herramientas suficientes para la generación de rutinas de control por visión de robots industriales, con las que es posible controlar tanto la posición como la orientación del manipulador; al realizar la designación de los objetivos de una maniobra, si se desea controlar la orientación del manipulador, el objetivo planteado debe estar definido con por lo menos tres puntos. La necesidad de definir la posición del objetivo mediante tres puntos, se debe al número de grados de libertad del manipulador (seis en total); al conocer las coordenadas tridimensionales de un solo punto del objetivo, solo se puede definir la posición a la cual arribará el manipulador. Mediante dos puntos, se puede definir además de la posición, la orientación de dos ángulos de rotación de la herramienta del robot. Tres puntos no colineales pueden determinar por completo los seis grados de libertad del robot.

Una de las características fundamentales de la metodología de control por visión de *CSM*, es el empleo de la teoría de estimación en muchos de los algoritmos aplicados para el control de robots. Uno de estos algoritmos es el algoritmo de estimación de la configuración interna del manipulador, el cual es planeado en la sección 2.6 del capítulo anterior. La función de este algoritmo es la de estimar el valor de las juntas del robot, de tal forma que en espacio de cámara “*empaten*” dos tipos de información; el primero proviene de las coordenadas en espacio de cámara del objetivo. El segundo es llamado información “*artificial*” y su origen es el traslado de los puntos objetivo originales en espacio de imagen, hasta su ubicación en espacio

de cámara, pero sobre la herramienta del manipulador. La información artificial se calcula a partir de las coordenadas tridimensionales de los puntos objetivo y de la cinemática directa del robot (Sección 2.1); al conocer la coordenadas tridimensionales de los puntos objetivo, éstos son trasladados primeramente hasta el origen del robot. Posteriormente, mediante la cinemática directa del mismo, dichos puntos son llevados hasta la punta de la herramienta. Finalmente, aprovechando las ecuaciones del modelo de perspectiva recursivo (Ecs 2.21), los puntos tridimensionales en la herramienta son nuevamente proyectados en espacio de cámara, para generarse así lo que se conoce como información artificial.

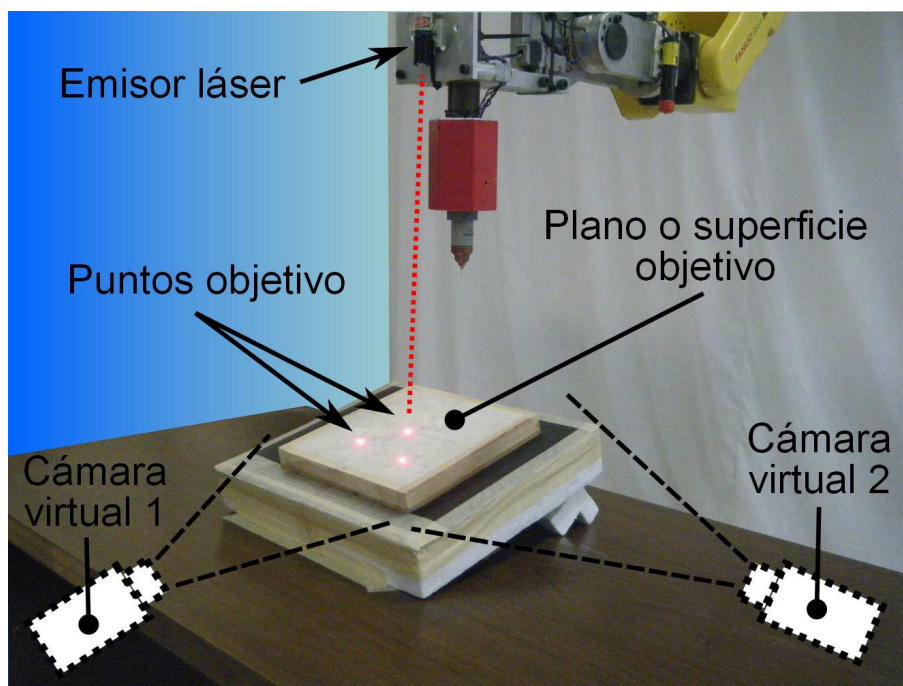


Figura 3.23: Definición del objetivo de maniobra.

El algoritmo de la estimación de la configuración interna del manipulador, requiere de por lo menos tres puntos para asegurar la definición del total de grados de libertad del robot. En este algoritmo, al empear información en espacio de cámara tanto real como artificial, si un número inferior a tres puntos es dado, se tendrá como resultado infinitas soluciones en el proceso de estimación, y la convergencia no será alcanzada.

Cuando se desea que el robot manipulador interactúe de determinada manera con una superficie en donde se encuentra el objetivo de maniobra, surge la necesidad de abstraer las características geométricas fundamentales de dicha superficie; este proceso es conocido como *caracterización de superficie* y es similar a la definición

de la ecuación topológica de ésta, a partir de las coordenadas tridimensionales de varios puntos en ella.

El proceso de caracterización de una superficie, mediante *CSM*, parte de la recopilación de las coordenadas de varias puntos en espacio de cámara, los cuales se encuentran sobre la superficie a caracterizar con $N_c \geq 2$. Posteriormente, mediante el algoritmo de la sección 2.5, se calcula la posición tridimensional de todos los puntos dados en espacio de imagen. La última parte del proceso de caracterización de la superficie, consiste en estimar los parámetros de una ecuación topológica, que describa de la mejor manera a la superficie de interés. La ecuación topológica de una superficie puede ser de múltiples formas y órdenes, según el tipo de superficie. En el presente proyecto de investigación, únicamente se empleó el proceso de caracterización de una superficie de primer orden, es decir, de la ecuación del plano, sin embargo, en proyectos precedentes [35, 47] se han empleado ecuaciones de segundo orden para caracterizar superficies curvas.

El proceso de estimación de la ecuación del plano parte de la minimización de la siguiente función escalar:

$$\varphi = \sum_{i=1}^m (AX_i + BY_i + CZ_i - 1)^2 \quad (3.15)$$

Donde las condiciones necesarias son:

$$\frac{\partial \varphi}{\partial A}; \quad \frac{\partial \varphi}{\partial B}; \quad \frac{\partial \varphi}{\partial C}; \quad (3.16)$$

Las cuales producen la siguiente solución de A, B, C :

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = \mathbf{P}^{-1} \mathbf{S} \quad (3.17)$$

Considerando a \mathbf{P} como no singular, entonces:

$$\mathbf{P} = \begin{bmatrix} \sum X_i^2 & \sum X_i Y_i & \sum X_i Z_i \\ & \sum Y_i^2 & \sum Y_i Z_i \\ \text{simétrica} & & \sum Z_i^2 \end{bmatrix} \quad (3.18)$$

$$\mathbf{S} = \begin{bmatrix} \sum X_i \\ \sum Y_i \\ \sum Z_i \end{bmatrix} \quad (3.19)$$

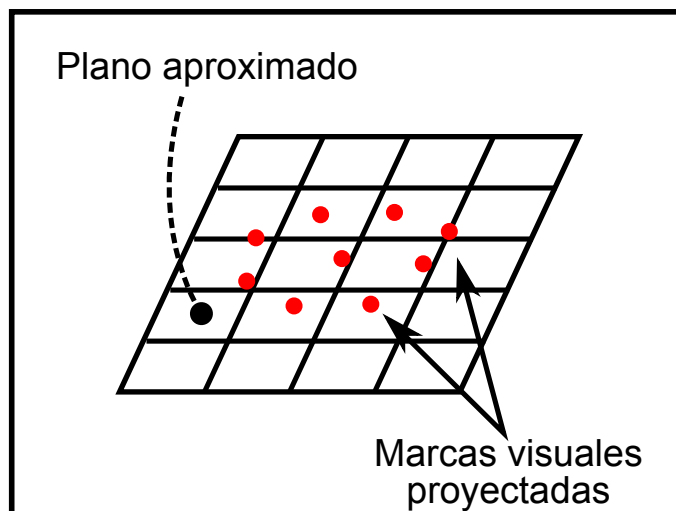


Figura 3.24: Aproximación de un plano a marcas visuales

Los puntos empleados para caracterizar la superficie del plano donde se encuentra el objetivo de maniobra, pueden ser empleados como los mismo puntos objetivo, con los que se calcula la información artificial útil para el algoritmo de estimación de juntas del manipulador. Al caracterizar la superficie en la cual arribará la herramienta del manipulador, se puede aprovechar a la ecuación topológica calculada para definir algoritmos que permitan al robot llegar a su objetivo siempre con una orientación deseada.

3.4.3. Ejecución de Maniobra de Posicionamiento y Refinamiento de Parámetros de Visión

Al ejecutar una maniobra de posicionamiento de un manipulador industrial mediante la metodología de *CSM*, se han estimado previamente un conjunto de parámetros de cámara, pertenecientes a las cámaras (en este caso virtuales) involucradas con la captura de la escena donde se encuentra el objetivo de maniobra, mediante una trayectoria pre-planeada. También se ha definido un conjunto de puntos objetivo en espacio de cámara, útiles para la estimación de la configuración interna del manipulador. Sin embargo, por la sencillez de los modelos de cámara empleados en *CSM*,

el conjunto de parámetros que constituyen el modelo de cada observador empleado, son solo válidos en una pequeña región del espacio; lo que exige su re-estimación sucesiva, al aproximarse el robot a su objetivo durante una maniobra de posicionamiento. Dicha re-estimación de los parámetros de cámara durante el transcurso de una maniobra de posicionamiento, recibe el nombre de *refinamiento de parámetros de cámara o de visión*.

La estructura basada en matrices de momentos del algoritmo para la estimación de parámetros de cámara de la sección 2.4, permite fácilmente la adición de información para re-estimar de los parámetros de cámara en todo momento, al agregar a las sumatorias de los elementos de las matrices \mathbf{M}_A , \mathbf{M}_B , nuevas muestras de las coordenadas de puntos, tanto de espacio físico ($\{X_i, Y_i, Z_i\}$), como de espacio de cámara ($\{X_{c,i}, Y_{c,i}\}$). La clave del éxito para el refinamiento de los parámetros de cámara, es el aumento del peso W_i asignado a cada nuevo conjunto de muestras adquirido, las cuales provienen de la captura de *LEDs* ubicados en un cuerpo prismático hexagonal, de geometría definida, colocado cerca de la punta de la herramienta del manipulador. Cuando transcurre una maniobra de posicionamiento, los parámetros de cámara son re-estimados de tal forma que siempre se le da un mayor peso a la estimación más actual, que es la cercana a la localidad por la que se está desplazando el efector final del robot; asegurando así el aumento de la precisión del manipulador, conforme éste se aproxima a su objetivo.

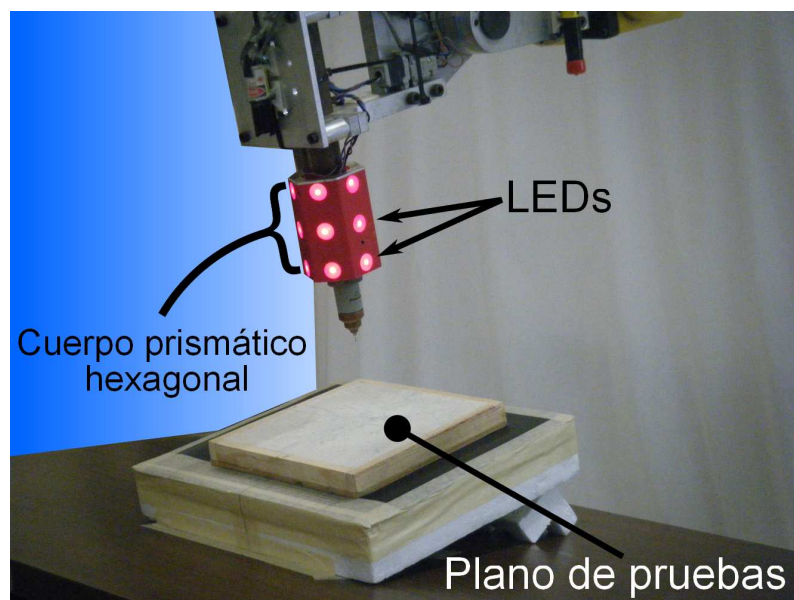
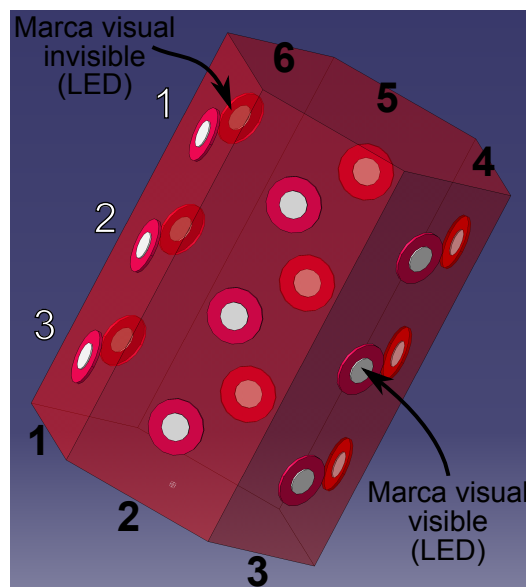


Figura 3.25: LEDs sobre cuerpo prismático para el refinamiento de los parámetros de cámara.

El cuerpo hexagonal que contiene a las marcas visuales, útiles para la refinación de los parámetros de cámara, posee en total 18 *LEDs* de alta intensidad; tres en cada una de sus caras. La geometría del cuerpo hexagonal y la distribución de los *LEDs* se muestran a continuación (Figura 3.26).



1..6: Caras del cuerpo prismático hexagonal
1..3: LEDs de cada cara del cuerpo prismático

Figura 3.26: Cuerpo prismático hexagonal del manipulador.

Hay un par de dificultades a enfrentar al refinar los parámetros de cámara de cada uno de los observados involucrados en un maniobra de posicionamiento; una de ellas es la de correlacionar cada marca visual captada en los N_c espacios de cámara con su respectiva posición tridimensional, ya que en este punto, serán captadas múltiples marcas visuales a la vez. Este problema se resuelve gracias a que una estimación previa de los parámetros de cámara es conocida, puesto que si la posición tridimensional de cada marca visual (*LED*) se define con la ayuda de la cinemática directa del robot, entonces es posible predecir la proyección en espacio de cámara donde aparecerán dichas marcas (Ecs 2.21). Al comparar las marcas visuales reales captadas por cada cámara (virtual), con la predicción de su aparición mediante el modelo de cámara (perspectiva recursivo), aquellas marcas reales que estén más cercanas a cierta predicción, pueden ser correlacionadas con las coordenadas tridimensionales con las que fue calculada la predicción dada.

El segundo problema a enfrentar es, para cuales de los *LEDs* o marcas visuales, hacer la predicción de su posición en espacio de cámara; ya que no todos los *LEDs*

son visibles al mismo tiempo, ni tampoco son captados por las mismas cámaras a la vez. El contratiempo se resuelve mediante el cálculo de un par de vectores, uno de ellos de orientación y sentido igual al del eje óptico del observador en cuestión; el otro de orientación igual a la normal saliente del plano, correspondiente a la cara del cuerpo prismático hexagonal en donde se encuentra la marca visual. El producto escalar de ambos vectores, define la condición de visibilidad para una marca visual dada. El procedimiento se muestra en breve.

Para definir el vector normal a la cara donde se encuentra cada marca visual, se parte de la cinemática directa del manipulador y de la definición de seis vectores unitarios normales, asociados a cada cara del cuerpo hexagonal, cuyo sistema de coordenadas de referencia, es el sistema coordenado de la herramienta. La definición de estos vectores unitarios se muestra a partir de la figura 3.27.

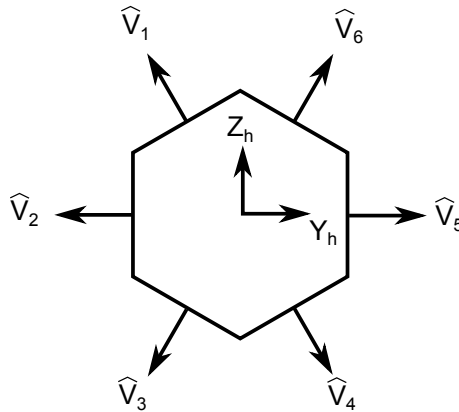


Figura 3.27: Definición de vectores unitarios sobre las caras del cuerpo hexagonal de herramienta del manipulador.

Los valores de los vectores unitarios $\hat{V}_{k=1..6}$ son:

$$\hat{V}_1 = \begin{bmatrix} 0 \\ -\frac{1}{2} \\ \frac{\sqrt{3}}{2} \\ 0 \end{bmatrix} \quad \hat{V}_2 = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \end{bmatrix} \quad \hat{V}_3 = \begin{bmatrix} 0 \\ -\frac{1}{2} \\ -\frac{\sqrt{3}}{2} \\ 0 \end{bmatrix}$$

$$\hat{V}_4 = \begin{bmatrix} 0 \\ \frac{1}{2} \\ -\frac{\sqrt{3}}{2} \\ 0 \end{bmatrix} \quad \hat{V}_5 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad \hat{V}_6 = \begin{bmatrix} 0 \\ \frac{1}{2} \\ \frac{\sqrt{3}}{2} \\ 0 \end{bmatrix}$$

A partir de la ecuación 2.8 de la cinemática directa el manipulador, la dirección de cada uno de los vectores asociados a los LEDs del cuerpo hexagonal, se puede definir como:

$$\vec{V}_k = \mathbf{H}(\Theta) * \hat{V}_k \quad (3.20)$$

La forma de definir el vector asociado al eje óptico de cada cámara virtual ($\overrightarrow{CV\hat{O}}$), se puede realizar a partir de una matriz de rotación arbitraria (Ec. 2.14). Esta matriz puede relacionar fácilmente las coordenadas tridimensionales de un observador virtual con respecto a la base del manipulador, una vez que se ha precisado el valor sus parámetros de cámara.

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} e_1^2 + e_2^2 - e_3^2 - e_4^2 & 2(e_2e_3 + e_1e_4) & 2(e_2e_4 - e_1e_3) & X_0 \\ 2(e_2e_3 - e_1e_4) & e_1^2 - e_2^2 + e_3^2 - e_4^2 & 2(e_3e_4 + e_1e_2) & Y_0 \\ 2(e_2e_4 + e_1e_3) & 2(e_3e_4 - e_1e_2) & e_1^2 - e_2^2 - e_3^2 + e_4^2 & Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{R}_{\text{arb}}} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \quad (3.21)$$

Donde $\{x_i, y_i, z_i\}$ son coordenadas referidas al sistema coordenado del manipulador y $\{X, Y, Z\}$ son coordenadas referidas a al origen de la cámara. Por otro lado, la definición del sistema coordenado de la cámara en términos del origen del manipulador es:

$$\begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} = \mathbf{R}_{\text{arb}}^{-1} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (3.22)$$

El vector asociado al eje óptico de la cámara, de acuerdo a la definición del modelo de perspectiva recursivo, solo puede tener una componen en el sentido de $-Z$, puesto que éste va dirigido hacia la escena de interés donde se encuentra la marca visual, por lo que el vector resultante es:

$$\overrightarrow{CV\dot{O}} = R_{arb}^{-1} \begin{bmatrix} 0 \\ 0 \\ -1 \\ 0 \end{bmatrix} = \begin{bmatrix} -2(e_2e_4 + e_1e_3) \\ -2(e_3e_4 - e_1e_2) \\ -e_1^2 + e_2^2 + e_3^2 - e_4^2 \\ 0 \end{bmatrix} \quad (3.23)$$

La condición de visibilidad de una marca visual dada, consiste en que el ángulo existente entre ambos vectores \vec{V}_k y $\overrightarrow{CV\dot{O}}$, debe estar en un rango de 90° a 270° , por lo tanto, el producto escalar de estos debe ser menor a cero.

$$\vec{V}_k \cdot \overrightarrow{CV\dot{O}} < 0 \quad (3.24)$$

3.4.4. Control de Orientación del Manipulador

El control de la orientación de un manipulador es esencial en el desempeño de tareas de un robot, donde se requiere que la herramienta de éste, arribe con determinada orientación a un objetivo. El algoritmo de estimación de la configuración interna del manipulador (Sección 2.6), provisto por la metodología de control por visión *CSM*, es capaz de controlar la orientación de la herramienta del robot a partir de la información de puntos objetivo, captados sobre una determinada superficie. En la presente sección se resolverá el problema en el caso en el que se desee hacer que la herramienta del manipulador llegue normal a una superficie aproximada a un plano; para esta tarea, se desarrollará el algoritmo de cálculo de las juntas del manipulador, pero solo de forma convencional sin emplear el enfoque de matrices de momentos.

Retomando la ecuación dada en la sección 2.6, para la estimación de las configuración interna de un manipulador, en base a la información de las coordenadas de un objetivo en espacio de cámara:

$$\gamma(\Theta) = \sum_{i=1}^{N_c} \sum_{j=1}^{N_t(i)} \{ [x_{t_i}^{(j)} - f_x(r_{x_i}^{(j)}(\Theta), r_{y_i}^{(j)}(\Theta), r_{z_i}^{(j)}(\Theta), x_{t_i}^{(j)}; \mathbf{P}^{(i)})]^2 + [y_{t_i}^{(j)} - f_y(r_{x_i}^{(j)}(\Theta), r_{y_i}^{(j)}(\Theta), r_{z_i}^{(j)}(\Theta), y_{t_i}^{(j)}; \mathbf{P}^{(i)})]^2 \} \quad (3.25)$$

Se puede observar que el proceso de minimización tiene como objetivo estimar el valor óptimo del vector Θ , para el cual una serie de muestras en espacio de cámara ($\{x_{t_i}^{(j)}, y_{t_i}^{(j)}\}$), empata con una versión de estas provista por el modelo de cámara de perspectiva recursivo ($\{f_x(r_{xyz_i}^{(j)}(\Theta); \mathbf{P}^{(i)}), f_y(r_{xyz_i}^{(j)}(\Theta); \mathbf{P}^{(i)})\}$). Las muestras dadas en espacio de cámara proceden de los puntos de un objetivo de maniobra; mientras que las marcas provistas por el modelo de cámara de perspectiva reursivo, corresponden con la llamada versión artificial de los puntos objetivo, cuando estos se encuentran situados sobre la herramienta del manipulador. Note que las muestras artificiales dadas en $\{f_x(r_{xyz_i}^{(j)}(\Theta); \mathbf{P}^{(i)})\}$ y en $\{f_y(r_{xyz_i}^{(j)}(\Theta); \mathbf{P}^{(i)})\}$, también son función de las muestras en espacio de cámara de los puntos objetivo.

El proceso de minimización de la ecuación 3.25, es el proceso iterativo de *corrección diferencial por mínimos cuadrados* [13], y consiste en el cálculo de una serie de incrementos $\Delta \mathbf{X}$, que son aproximaciones sucesivas a la solución óptima (Θ^{opt}) de los términos del vector Θ , partiendo de un vector inicial $\Theta^{inicial}$. Si el proceso de minimización es convergente, entonces la magnitud de los incrementos $\Delta \mathbf{X}$, irá disminuyendo con cada iteración del proceso, por lo que es posible determinar el valor de los términos del vector Θ^{opt} , a partir de cierta magnitud alcanzada de los incrementos $\Delta \mathbf{X}$, definida por una tolerancia (TOL); es decir, que $\Theta \approx \Theta^{opt}$ cuando $\Delta \mathbf{X} \leq TOL$.

La forma de calcular los incrementos $\Delta \mathbf{X}$, se presenta a continuación:

$$\Delta \mathbf{X} = (\mathbf{A}^T \mathbf{W} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{W} \Delta \mathbf{Y} \quad (3.26)$$

Donde \mathbf{W} es igual a una matriz de pesos, útil para darle un valor relativo a cada una de las muestras tomadas en espacio de cámara; sin embargo, en este proceso el peso relativo de cada muestras es considera constante. El término de $\Delta \mathbf{Y}$ es igual a un vector compuesto de los valores residuales de restar a cada una de las muestras

de los puntos objetivo tomadas en espacio de cámara ($\{x_{t_i}^{(j)}, y_{t_i}^{(j)}\}$), de la información artificial respectiva a éstas, resultante de trasladar los puntos objetivo, hasta la herramienta del manipulador ($\{f_x(r_{xy_{z_i}}^{(j)}(\Theta); \mathbf{P}^{(i)})\}, \{f_y(r_{xy_{z_i}}^{(j)}(\Theta); \mathbf{P}^{(i)})\}$). Finalmente, \mathbf{A} representa una matriz no singular de derivadas parciales, resultante de derivar las ecuaciones del modelo de perspectiva recursivo, por cada una de las variables asociadas a los valores de las juntas del manipulador ($\Theta = \{J1 \dots J6\}$). El vector $\Delta \mathbf{Y}$ se desarrolla en la ecuación 3.27

$$\Delta \mathbf{Y} = \begin{bmatrix} x_{t_{i=1}}^{j=1} - f_x(r_{x_{i=1}}^{(j=1)}(\Theta), r_{y_{i=1}}^{(j=1)}(\Theta), r_{z_{i=1}}^{(j=1)}(\Theta), x_{t_{i=1}}^{(j=1)}; \mathbf{P}^{(i=1)}) \\ y_{t_{i=1}}^{j=1} - f_y(r_{x_{i=1}}^{(j=1)}(\Theta), r_{y_{i=1}}^{(j=1)}(\Theta), r_{z_{i=1}}^{(j=1)}(\Theta), y_{t_{i=1}}^{(j=1)}; \mathbf{P}^{(i=1)}) \\ \vdots \\ x_{t_{i=1}}^{N_t} - f_x(r_{x_{i=1}}^{(N_t)}(\Theta), r_{y_{i=1}}^{(N_t)}(\Theta), r_{z_{i=1}}^{(N_t)}(\Theta), x_{t_{i=1}}^{(N_t)}; \mathbf{P}^{(i=1)}) \\ y_{t_{i=1}}^{N_t} - f_y(r_{x_{i=1}}^{(N_t)}(\Theta), r_{y_{i=1}}^{(N_t)}(\Theta), r_{z_{i=1}}^{(N_t)}(\Theta), y_{t_{i=1}}^{(N_t)}; \mathbf{P}^{(i=1)}) \\ \vdots \\ \dots \\ \vdots \\ x_{t_{N_c}}^{j=1} - f_x(r_{x_{N_c}}^{(j=1)}(\Theta), r_{y_{N_c}}^{(j=1)}(\Theta), r_{z_{N_c}}^{(j=1)}(\Theta), x_{t_{N_c}}^{(j=1)}; \mathbf{P}^{(N_c)}) \\ y_{t_{N_c}}^{j=1} - f_y(r_{x_{N_c}}^{(j=1)}(\Theta), r_{y_{N_c}}^{(j=1)}(\Theta), r_{z_{N_c}}^{(j=1)}(\Theta), y_{t_{N_c}}^{(j=1)}; \mathbf{P}^{(N_c)}) \\ \vdots \\ x_{t_{N_c}}^{N_t} - f_x(r_{x_{N_c}}^{(N_t)}(\Theta), r_{y_{N_c}}^{(N_t)}(\Theta), r_{z_{N_c}}^{(N_t)}(\Theta), x_{t_{N_c}}^{(N_t)}; \mathbf{P}^{(N_c)}) \\ y_{t_{N_c}}^{N_t} - f_y(r_{x_{N_c}}^{(N_t)}(\Theta), r_{y_{N_c}}^{(N_t)}(\Theta), r_{z_{N_c}}^{(N_t)}(\Theta), y_{t_{N_c}}^{(N_t)}; \mathbf{P}^{(N_c)}) \end{bmatrix} \quad (3.27)$$

Donde el valor de “ i ” se refiere a la i -ésima cámara (virtual) y “ N_c ” al número total de cámaras ($N_c \geq 2$). El valor de “ j ” esta asociado a la j -ésima marca visual objetivo y a la representación artificial de esta misma, mediante el modelo de perspectiva recursivo. El valor de “ N_t ”, es el número total de marcas visuales objetivo utilizadas para cada uno de los “ i ” espacios de cámara.

El número de renglones del vector $\Delta \mathbf{Y}$, varia dependiendo del número de marcas visuales objetivo y del número de cámaras involucradas en una maniobra de posicionamiento, siendo igual a $2 \times N_t \times N_c$.

La matrix \mathbf{A} de derivadas parciales está compuesta de los siguientes elementos:

$$\mathbf{A} = \left[\begin{array}{ccc}
\frac{\partial f_x(r_{x_{i=1}}^{(j=1)}(\Theta), r_{y_{i=1}}^{(j=1)}(\Theta), r_{z_{i=1}}^{(j=1)}(\Theta), x_{t_{i=1}}^{(j=1)}; \mathbf{P}^{(i=1)})}{\partial J1} & \dots & \frac{\partial f_x(r_{x_{i=1}}^{(j=1)}(\Theta), r_{y_{i=1}}^{(j=1)}(\Theta), r_{z_{i=1}}^{(j=1)}(\Theta), x_{t_{i=1}}^{(j=1)}; \mathbf{P}^{(i=1)})}{\partial J6} \\
\frac{\partial f_y(r_{x_{i=1}}^{(j=1)}(\Theta), r_{y_{i=1}}^{(j=1)}(\Theta), r_{z_{i=1}}^{(j=1)}(\Theta), y_{t_{i=1}}^{(j=1)}; \mathbf{P}^{(i=1)})}{\partial J1} & \dots & \frac{\partial f_y(r_{x_{i=1}}^{(j=1)}(\Theta), r_{y_{i=1}}^{(j=1)}(\Theta), r_{z_{i=1}}^{(j=1)}(\Theta), y_{t_{i=1}}^{(j=1)}; \mathbf{P}^{(i=1)})}{\partial J6} \\
\vdots & \ddots & \vdots \\
\frac{\partial f_x(r_{x_{i=1}}^{(N_t)}(\Theta), r_{y_{i=1}}^{(N_t)}(\Theta), r_{z_{i=1}}^{(N_t)}(\Theta), x_{t_{i=1}}^{(N_t)}; \mathbf{P}^{(i=1)})}{\partial J1} & \dots & \frac{\partial f_x(r_{x_{i=1}}^{(N_t)}(\Theta), r_{y_{i=1}}^{(N_t)}(\Theta), r_{z_{i=1}}^{(N_t)}(\Theta), x_{t_{i=1}}^{(N_t)}; \mathbf{P}^{(i=1)})}{\partial J6} \\
\frac{\partial f_y(r_{x_{i=1}}^{(N_t)}(\Theta), r_{y_{i=1}}^{(N_t)}(\Theta), r_{z_{i=1}}^{(N_t)}(\Theta), y_{t_{i=1}}^{(N_t)}; \mathbf{P}^{(i=1)})}{\partial J1} & \dots & \frac{\partial f_y(r_{x_{i=1}}^{(N_t)}(\Theta), r_{y_{i=1}}^{(N_t)}(\Theta), r_{z_{i=1}}^{(N_t)}(\Theta), y_{t_{i=1}}^{(N_t)}; \mathbf{P}^{(i=1)})}{\partial J6} \\
\vdots & \ddots & \vdots \\
\frac{\partial f_x(r_{x_{N_c}}^{(j=1)}(\Theta), r_{y_{N_c}}^{(j=1)}(\Theta), r_{z_{N_c}}^{(j=1)}(\Theta), x_{t_{N_c}}^{(j=1)}; \mathbf{P}^{(N_c)})}{\partial J1} & \dots & \frac{\partial f_x(r_{x_{N_c}}^{(j=1)}(\Theta), r_{y_{N_c}}^{(j=1)}(\Theta), r_{z_{N_c}}^{(j=1)}(\Theta), x_{t_{N_c}}^{(j=1)}; \mathbf{P}^{(N_c)})}{\partial J6} \\
\frac{\partial f_y(r_{x_{N_c}}^{(j=1)}(\Theta), r_{y_{N_c}}^{(j=1)}(\Theta), r_{z_{N_c}}^{(j=1)}(\Theta), y_{t_{N_c}}^{(j=1)}; \mathbf{P}^{(N_c)})}{\partial J1} & \dots & \frac{\partial f_y(r_{x_{N_c}}^{(j=1)}(\Theta), r_{y_{N_c}}^{(j=1)}(\Theta), r_{z_{N_c}}^{(j=1)}(\Theta), y_{t_{N_c}}^{(j=1)}; \mathbf{P}^{(N_c)})}{\partial J6} \\
\vdots & \ddots & \vdots \\
\frac{\partial f_x(r_{x_{N_c}}^{(N_t)}(\Theta), r_{y_{N_c}}^{(N_t)}(\Theta), r_{z_{N_c}}^{(N_t)}(\Theta), x_{t_{N_c}}^{(N_t)}; \mathbf{P}^{(N_c)})}{\partial J1} & \dots & \frac{\partial f_x(r_{x_{N_c}}^{(N_t)}(\Theta), r_{y_{N_c}}^{(N_t)}(\Theta), r_{z_{N_c}}^{(N_t)}(\Theta), x_{t_{N_c}}^{(N_t)}; \mathbf{P}^{(N_c)})}{\partial J6} \\
\frac{\partial f_y(r_{x_{N_c}}^{(N_t)}(\Theta), r_{y_{N_c}}^{(N_t)}(\Theta), r_{z_{N_c}}^{(N_t)}(\Theta), y_{t_{N_c}}^{(N_t)}; \mathbf{P}^{(N_c)})}{\partial J1} & \dots & \frac{\partial f_y(r_{x_{N_c}}^{(N_t)}(\Theta), r_{y_{N_c}}^{(N_t)}(\Theta), r_{z_{N_c}}^{(N_t)}(\Theta), y_{t_{N_c}}^{(N_t)}; \mathbf{P}^{(N_c)})}{\partial J6}
\end{array} \right] \quad (3.28)$$

El control de la orientación del robot está dado al manipular la información artificial en espacio de cámara, de las marcas sobre la punta del manipulador. Para generar la información artificial como se mencionó en la sección 3.4.2, se hace uso de la reconstrucción tridimensional de los puntos objetivo, los cuales son trasladados a la punta de la herramienta del robot mediante su cinemática directa, para ser luego proyectados nuevamente en espacio de cámara mediante el modelo de visión empleado. La clave para el control de la orientación del manipulador, es la de agregar una matriz de rotación extra, a la cinemática directa del robot ($\mathbf{H}(\Theta)$), con la cual se decide la orientación de los puntos artificiales sobre la herramienta del robot. Si se desarrollan las ecuaciones del modelo de perspectiva recursivo utilizadas en \mathbf{A} y $\Delta\mathbf{Y}$, para el caso de la i -ésima cámara y de la j -ésima marca objetivo, entonces se tiene lo siguiente:

$$\begin{aligned}
f_x(r_{x_i}^{(j)}(\Theta), r_{y_i}^{(j)}(\Theta), r_{z_i}^{(j)}(\Theta), x_{t_i}^{(j)}; \mathbf{P}^{(i)}) &= b_1(\mathbf{P})r_{x_i} + b_2(\mathbf{P})r_{y_i} + b_3(\mathbf{P})r_{z_i} + b_4(\mathbf{P}) \dots \\
&+ b_9(\mathbf{P})r_{x_i}x_{t_i}^{(j)} + b_{10}(\mathbf{P})r_{y_i}x_{t_i}^{(j)} + b_{11}(\mathbf{P})r_{z_i}x_{t_i}^{(j)} \quad (3.29)
\end{aligned}$$

$$\begin{aligned}
f_y(r_{x_i}^{(j)}(\Theta), r_{y_i}^{(j)}(\Theta), r_{z_i}^{(j)}(\Theta), y_{t_i}^{(j)}; \mathbf{P}^{(i)}) &= b_5(\mathbf{P})r_{x_i} + b_6(\mathbf{P})r_{y_i} + b_7(\mathbf{P})r_{z_i} + b_8(\mathbf{P}) \dots \\
&+ b_9(\mathbf{P})r_{x_i}y_{t_i}^{(j)} + b_{10}(\mathbf{P})r_{y_i}y_{t_i}^{(j)} + b_{11}(\mathbf{P})r_{z_i}y_{t_i}^{(j)} \quad (3.30)
\end{aligned}$$

Los valores de $r_{x_i}^j, r_{y_i}^j, r_{z_i}^j$, corresponden con las coordenadas de la posición tridimensional de la j -ésima marca artificial captada por la i -ésima cámara, y dependen de la cinemática directa del robot ($\mathbf{H}(\Theta)$); la orientación de dichas coordenadas tridimensionales se puede manipular mediante una matriz de rotación arbitraria, para que el algoritmo de estimación de la configuración interna del manipulador estime valores de juntas, con las que el robot alcance su objetivo con una orientación específica. Esto se debe a que durante el empate de marcas en espacio de cámara, realizado al minimizar la función de la ecuación 3.25, se motivará la búsqueda de una configuración de Θ , en la cual los puntos objetivo empaten con la información artificial rotada.

Para trasladar un cúmulo de marcas artificiales hasta la herramienta del robot, una vez que se conocen las coordenadas tridimensionales de éstas, primeramente se debe de elegir, de entre estas marcas, alguna que sea considerada como el *punto exacto objetivo*; es decir, la marca que coincidirá con la punta de la herramienta del robot. Posteriormente, todas las marcas restantes deben de ser referenciadas a la marca elegida, tal cual, como si ésta fuera su origen de coordenadas; una vez hecho esto, mediante la cinemática directa del robot, es posible fácilmente trasladar todas las marcas artificiales hasta la herramienta del robot.

El traslado de una j -ésima marca artificial ($\{x^j, y^j, z^j\}$), siendo conocida la posición del punto objetivo exacto ($\{x^{obj}, y^{obj}, z^{obj}\}$) y la cinemática directa del manipulador ($\mathbf{H}(\Theta)$), es dada a continuación:

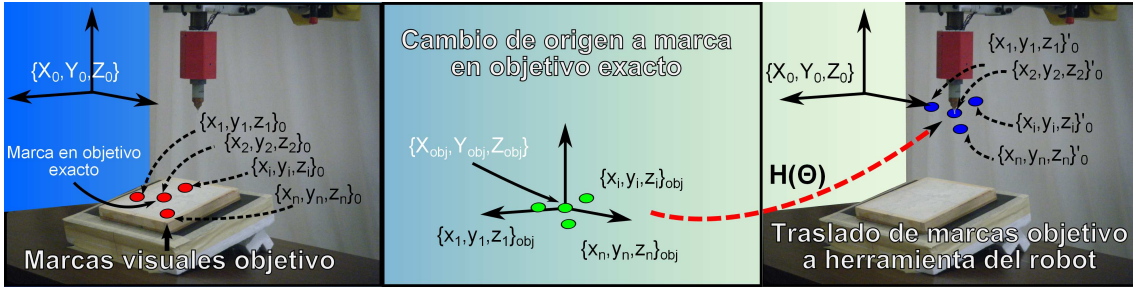


Figura 3.28: Traslado de marcas visuales hasta herramienta del manipulador, para conformar información artificial.

$$\begin{bmatrix} r_x^j \\ r_y^j \\ r_z^j \\ 1 \end{bmatrix} = \mathbf{H}(\Theta) * \begin{bmatrix} 1 & 0 & 0 & -x^{obj} \\ 0 & 1 & 0 & -y^{obj} \\ 0 & 0 & 1 & -z^{obj} \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x^j \\ y^j \\ z^j \\ 1 \end{bmatrix} \quad (3.31)$$

Si los puntos artificiales en la herramienta del robot ($\{r_x^j, r_y^j, r_z^j\}$), no sufrieron ninguna modificación en cuanto a su orientación en el proceso de traslado, durante la maniobra de posicionamiento en curso, el manipulador llegará hasta su objetivo con la misma orientación de su herramienta a como comenzó. El proceso para definir una matriz de rotación adicional a la cinemática directa del robot, con la que este puede llegar a un objetivo plano de forma ortogonal, se presenta en breve.

Si la matriz de rotación adicional es llamada \mathbf{S} , y su forma es la de una matriz de rotación arbitraria, entonces:

$$\mathbf{S} = \begin{bmatrix} e_1^2 + e_2^2 - e_3^2 - e_4^2 & 2(e_2e_3 + e_1e_4) & 2(e_2e_4 - e_1e_3) & 0 \\ 2(e_2e_3 - e_1e_4) & e_1^2 - e_2^2 + e_3^2 - e_4^2 & 2(e_3e_4 + e_1e_2) & 0 \\ 2(e_2e_4 + e_1e_3) & 2(e_3e_4 - e_1e_2) & e_1^2 - e_2^2 - e_3^2 + e_4^2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.32)$$

Donde los cuaterniones de la matriz, están definidos en función de un vector unitario de rotación arbitrario $\hat{\lambda}_{\text{arb}}$ y un ángulo de rotación θ , de la siguiente manera:

$$e_1 = -\cos(-\theta/2) \quad (3.33)$$

$$e_2 = \hat{\lambda}_{\text{arb}_x} \text{sen}(-\theta/2) \quad (3.34)$$

$$e_3 = \hat{\lambda}_{\text{arb}_y} \text{sen}(-\theta/2) \quad (3.35)$$

$$e_4 = \hat{\lambda}_{\text{arb}_z} \text{sen}(-\theta/2) \quad (3.36)$$

Donde $\hat{\lambda}_{\text{arb}_x}$, $\hat{\lambda}_{\text{arb}_y}$, $\hat{\lambda}_{\text{arb}_z}$ son las componentes de vector unitario $\hat{\lambda}_{\text{arb}}$. Si se presenta el caso particular en el cual, el eje Z_h , del sistema coordenado ubicado al extremo de la herramienta X_h, Y_h, Z_h , coincide con la orientación del eje Z del último eslabón del manipulador, se puede definir la magnitud del ángulo θ , en base al vector normal \mathbf{N} derivado de la ecuación del plano característica de la superficie en donde se encuentran los puntos objetivo (Ec 3.15). Dicho ángulo es igual al arco cuyo coseno es el producto punto del vector normalizado $\|\mathbf{N}\|$ con un vector unitario en el sentido del eje Z_0 del origen de coordenadas del manipulador, entonces:

$$\theta = \|\mathbf{N}\| \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad (3.37)$$

El vector arbitrario $\hat{\lambda}_{\text{arb}}$, se define como el producto vectorial entre el vector normalizado $\|\mathbf{N}\|$ y el vector unitario paralelo al eje Z_0 :

$$\hat{\lambda}_{\text{arb}} = \|\mathbf{N}\| \times \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad (3.38)$$

Finalmente la forma de obtener la información artificial de los puntos tridimensionales sobre la herramienta del robot con la cual, en una maniobra de posicionamiento, el efector final del robot se oriente ortogonal a la superficie de un plano, es la siguiente:

$$\begin{bmatrix} r_x^j \\ r_y^j \\ r_z^j \\ 1 \end{bmatrix} = \mathbf{H}(\Theta) * \begin{bmatrix} 1 & 0 & 0 & -x^{obj} \\ 0 & 1 & 0 & -y^{obj} \\ 0 & 0 & 1 & -z^{obj} \\ 0 & 0 & 0 & 1 \end{bmatrix} * \mathbf{S} * \begin{bmatrix} x^j \\ y^j \\ z^j \\ 1 \end{bmatrix} \quad (3.39)$$

Capítulo 4

Resultados Experimentales

Después de un análisis teórico extenso realizado en los capítulos anteriores, concierne a la metodología de *Manipulación de Espacio de Cámara* aplicada al caso particular en el que se ha de concretar el control por visión de un manipulador industrial, utilizando un sistema de cámaras virtuales como medio de adquisición de imágenes; el presente capítulo está dedicado a reportar la información derivada de la validación experimental. Además, este capítulo está ocupado en realizar un análisis interpretativo y simultáneamente inferencial, para detectar las causas y los significados de los resultados obtenidos a partir de la labor experimental.

La metodología de *CSM* es ampliamente dependiente de la “*densidad de resolución*”, con la cual se realiza la toma de imágenes a las escenas de interés [26, 33]. Por *densidad de resolución*, se entiende al cociente resultante de dividir la cantidad de píxeles (px) que representan linealmente la dimensión longitudinal de una imagen, por la longitud física real (cm, mm) de la escena sobre un plano perpendicular al eje óptico de una cámara, con la que la imagen fue captada. La influencia de la densidad de resolución en el control por visión de robots en *CSM*, radica en que entre más “densa” sea la representación en espacio de cámara del área de trabajo de un robot, el mapeo entre espacio físico tridimensional y espacio de imagen por algún modelo de cámara será localmente más sensible, lo que posibilita distinguir diferencias de desplazamientos más pequeños vistos en espacio de cámara, lo que se traduce en el posicionamiento de un robot, con mayor precisión en un objetivo.

Dada la relevancia del valor de la *densidad de resolución* con la cual se realizan las capturas de imágenes, durante una maniobra de control por visión de un mani-

pulador; el primer apartado sobre pruebas experimentales realizadas está dedicado a describir el método experimental para obtener dicho valor. No obstante, para propósitos del presente trabajo de investigación, el valor de *densidad de resolución* será expresado en términos de la “*resolución específica*” (mm/px) o valor recíproco de la *densidad de resolución* ($px/mm, px/cm$).

La forma de verificar la eficacia de la metodología de *CSM*, sobre un sistema de visión determinado, es mediante lo que se conoce como “*maniobra de posicionamiento*”, que consiste en la designación de un objetivo en espacio de imagen. Posteriormente, un proceso que involucra el conocimiento de los parámetros de cada cámara en uso, cámaras virtuales en este caso, es utilizado para estimar la configuración interna del manipulador controlado, de modo tal que se satisfaga una condición de aproximación de la herramienta del manipulador al objetivo en espacio de cámara. Finalmente, el robot controlado es ordenado a adoptar la configuración estimada y así arribar a su objetivo. Durante el proceso de aproximación al objetivo de maniobra, los parámetros de cada cámara virtual son re-estimados y enriquecidos con nueva información procedente del espacio de imagen captado; esto con el fin de asegurar la validez local del modelo de visión empleado.

El modo de cuantificar el éxito de una maniobra de posicionamiento es mediante la medición de la proximidad lineal de la herramienta del robot controlado al objetivo designado. La segunda sección de este capítulo muestra los resultados experimentales obtenidos en maniobras de posicionamiento realizadas con el sistema de cámaras virtuales desarrollado; haciendo mención de la relación mm/px , a la interpretación de los resultados obtenidos.

El último apartado de este capítulo consiste en determinar la capacidad del sistema de cámaras virtuales, para detectar cambios de relieve en una superficie. Esto con el objetivo de dar pie a trabajo futuro, en el cual se plantea el empleo de la metodología de *CSM* y un sistema de adquisición de imágenes basado en cámaras virtuales, no en un sistema de control por visión de robots industriales, sino como parte de un sistema de medición de espesores, o de caracterización de superficies basado en visión; que halla aplicación en métodos alternativos de restauración de obras.

Antes de comenzar con la descripción de los experimentos realizados para la validación del presente proyecto de investigación, se procede a describir el equipo disponible en el Laboratorio de Robótica, de la Facultad de Ingeniería de la UASLP.

4.1. Equipo Disponible en el Laboratorio

Para llevar a cabo la presente investigación, se contó con las instalaciones y el equipo disponible dentro del Laboratorio de Robótica (Figura 4.1), ubicado en las instalaciones de la Facultad de Ingeniería de la Universidad Autónoma de San Luis Potosí. El instrumental empleado fue un ordenador portátil, una cámara de video USB de la marca *Ueye*, modelo UI-1450-CM, capaz de brindar una resolución máxima de 1600×1200 pixeles a una tasa de 16 fotogramas por segundo. Un robot industrial marca *Fanuc*, modelo M-16iB/20T con un controlador R-30iA, el cual posee seis grados de libertad, uno de ellos sobre una única junta prismática de considerable amplitud (3660 mm); dos grados de libertad en juntas de revoluta para la manipulación de los eslabones rígidos de mayor tamaño (770 y 740 mm de longitud) y tres grados de libertad más, en una muñeca que precede al último eslabón del robot (100 mm de longitud) donde se acopla la herramienta. Específicamente, dicho instrumento está conformado por un cuerpo de aluminio, que sostiene una antorcha de corte de plasma modificada en su extremo; la cual ha sido adicionada con un cuerpo prismático hexagonal hueco, de geometría definida, cuyo eje central está alineado con el eje longitudinal de la antorcha. El prisma hexagonal contiene tres LEDs color rojo de alta intensidad en cada una de sus caras, empleados como referencias visuales para la generación de información útil en los procesos de estimación en *CSM*. Un dispositivo más, colocado sobre el herramental del manipulador, es un láser marca *Lasiris* color rojo (635 nm) y es utilizado en la designación del objetivo de una maniobra. Para generar el sistema de cámaras virtuales necesario, se emplearon cuatro espejos convencionales (planos) con espesor de tres mm , dos de ellos con dimensiones de $300 \times 300 \text{ mm}$ y dos más con dimensiones de $1000 \times 1000 \text{ mm}$; de este sistema se da una explicación más detallada en el Capítulo tres.

4.2. Determinación de la Relación “mm/px” por Cámara

La relación “ mm/px ” o *resolución específica*, es un dato de importancia considerable en la obtención de resultados favorables o eficaces al realizar maniobras de posicionamiento, siendo deseable el manejo de valores pequeños de *resolución específica*; en

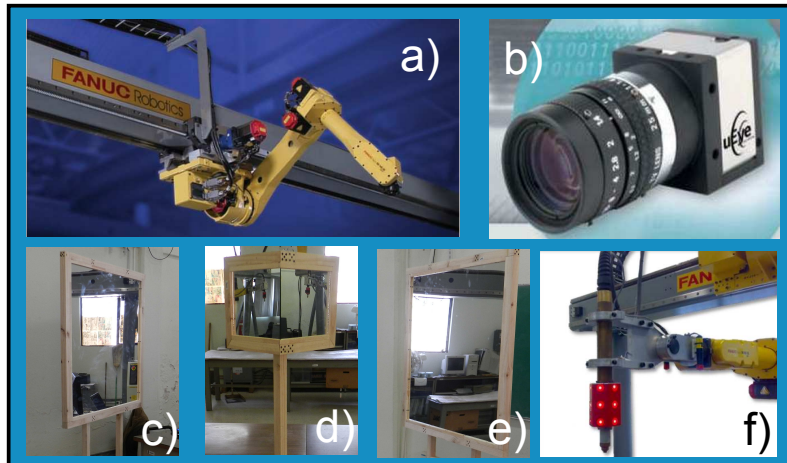


Figura 4.1: Equipo disponible en el laboratorio de robótica de la UASLP-FI; a) Robot Fanuc M16iB/20T; b) Cámara Ueye mod. UI-1450-CM; c), e) Espejos de medidas $1000 \times 1000 \text{ mm}$; d) Espejos de medidas $300 \times 300 \text{ mm}$; f) Herramienta del robot, con cuerpo prismático hexagonal.

CSM inferiores a la unidad, cuando la *resolución específica* es expresada en mm/px , dado que es deseable (según la aplicación), mantener el error en pruebas de posicionamiento en el orden de milímetros o fracciones de milímetros. Como ya se ha mencionado anteriormente, al utilizar cámaras virtuales como una alternativa al uso de múltiples sensores opto-electrónicos, decrece la *densidad de resolución* o aumenta la *resolución específica* con la cual se realiza la captura de imágenes necesaria para el flujo de información en un sistema de control por visión. Para generar una correcta interpretación de los resultados experimentales en pruebas de posicionamiento, el conocimiento del valor de *resolución específica* con el cual se realiza una secuencia de experimentos es indispensable. No obstante, otros factores relacionados son de gran importancia, como por ejemplo, la orientación del objetivo de una maniobra respecto del eje óptico de los observadores virtuales; ya que un objetivo ubicado en un plano orientado de forma sesgada respecto del eje focal de una cámara, será captado con una *resolución específica* alta y por ende, la aproximación a él en una maniobra de posicionamiento será menos precisa.

Experimentalmente, el modo de obtener la *resolución específica* de una escena de interés para pruebas experimentales, es mediante la medición en espacio de imagen, de la cantidad de pixeles que longitudinalmente conforman un “objeto” de dimensiones conocidas. El objeto se ha colocado sobre un plano cuya normal es paralela al eje óptico de alguna de las cámaras virtuales empleadas; la ubicación física de este plano es cercana al objetivo de las maniobras de posicionamiento posteriormente realizadas. El experimento se realiza por cada una de las cámaras involucradas en

un pruebas de posicionamiento y se realizan un total de 30 mediciones en espacio de imagen del objeto para encontrar una relación de (mm/px) válida.

A continuación se describe paso a paso el proceso para determinar la resolución específica de una escena donde se desarrollarán pruebas de posicionamiento:

1. **Construcción del plano y del objeto caracterizado:** Para representar el plano sobre el cual será colocado el objeto caracterizado, se utiliza un tablero de madera, razonablemente plano o perceptible como tal (Figura 4.2); el objeto caracterizado no es otra cosa que una hoja de cuadrícula milimétrica adherida al plano empleado. De la hoja milimétrica es considerado únicamente un segmento lineal de 50 mm, para efectuar la medición correspondiente en espacio de cámara. Para sujetar y orientar el plano que contiene la cuadrícula milimétrica, se ha utilizado un tornillo de banco portátil, el cual posee una rótula ajustable, que permite orientar a preferencia cualquier objeto sujetado.

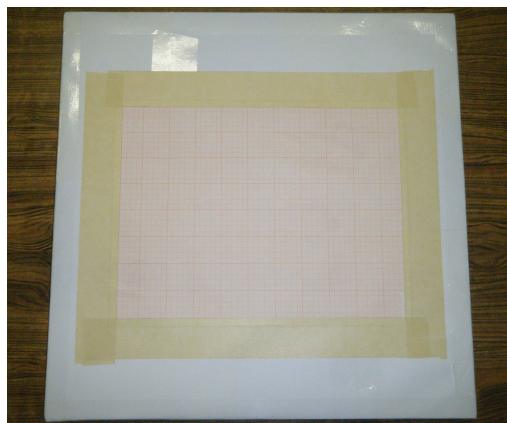


Figura 4.2: Plano construido para experimentos de cuantificación de resolución específica.

2. **Alineación del plano físico con el eje óptico de la cámara:** Este procedimiento se realiza manualmente. Para simular el eje óptico de la cámara se ha construido un dispositivo adherible a ella, que sirve de soporte para un emisor láser (Figura 4.3).

El emisor láser es sujetado con su eje longitudinal paralelo al eje focal de la cámara, de modo que al disparar el haz de luz procedente del láser, su trayecto será similar al del eje óptico de la cámara. Puesto que el trayecto del eje óptico de cada cámara virtual, resulta de la desviación mediante espejos del eje óptico

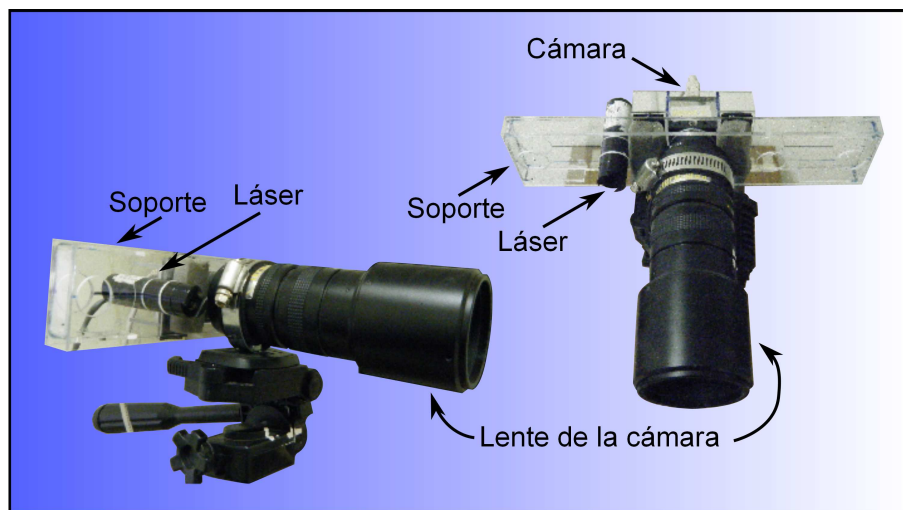


Figura 4.3: Soporte de emisor láser para simulación del eje óptico de cada cámara virtual.

de la cámara real, el haz láser emitido sufrirá los mismos cambios de dirección, hasta llegar al plano que deberá ser alineado normal a él. Para alinear dicho plano, es necesario hacer lateralmente visible el haz láser de referencia, además de adherir a dicho plano de análisis, algún elemento de referencia que posibilite su alineación de forma perpendicular al haz de luz láser. El problema que es resuelto de la forma siguiente, mediante un procedimiento eficaz, aunque no del todo limpio. Para hacer visible el haz del láser, se dispersa algún tipo de polvo fino como yeso pulverizado (sulfato cálcico bihidrato $CaSO_4 \cdot 2H_2O$: gis para pizarrón), lo cual genera una nube de partículas, que posibilitan ver el haz cuando hay poca iluminación en el recinto de pruebas experimentales. Una vez que el haz es visible, se ha utilizado una prensa para escuadrar, con la que se ha fijado una placa de aluminio al plano cuadrículado. La placa a su vez, contiene el trazo de una línea perpendicular al plano, que es utilizada como referencia de alineación.

3. **Medición en espacio de cámara de objeto en escena:** La medición en espacio de cámara del segmento sobre la cuadrícula milimétrica, se realiza de forma automática, aprovechando la disponibilidad del robot Fanuc M16-iB/20T y la clase de procesamiento de imágenes programada para la interfaz gráfica (*GUI*) desarrollada (*Apéndice C*). Dado que el autómatas empleado cuenta con un emisor láser adicionado a su herramienta, un par de movimientos programados en la interfaz gráfica son ejecutados por el robot cíclicamente, en los que

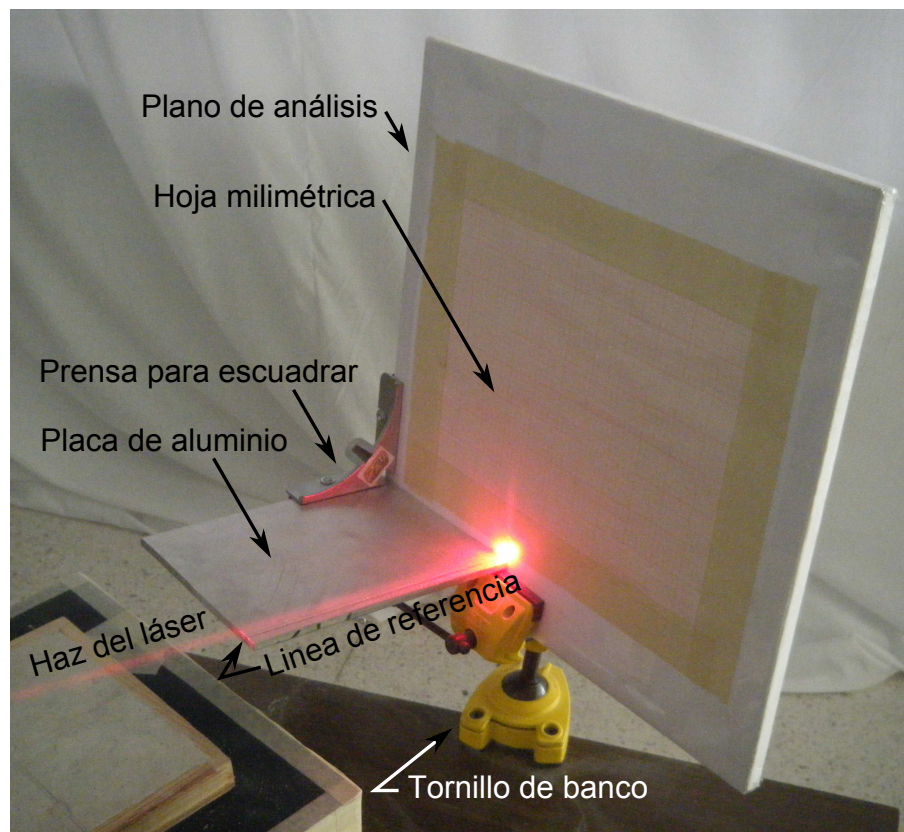


Figura 4.4: Tablero de madera montado y alineado con haz de luz láser, para simular un plano perpendicular al eje óptico de la cámara.

el láser sobre el efector final es dirigido hacia la cuadrícula del plano. El punto luminiscente es proyectado al inicio y al final del segmento sobre la cuadrícula de referencia (Figura 4.5). Entre cada movimiento del autómata, la cámara para la cual se realiza la cuantificación de la *resolución específica*, realiza coordinadamente las capturas en donde el punto láser está al inicio y al final del segmento de referencia. Finalmente se ubican las coordenadas en imagen de los centroides de cada par de marcas luminiscentes ($\{x_{c,i}, y_{c,i}\} \rightarrow i = 1 \dots 2$) y se calcula la distancia respectiva entre ellas. La distancia calculada es dada en píxeles y su valor es utilizado como divisor de la magnitud del segmento de cuadrícula para obtener la razón igual a la *resolución específica*. A continuación se muestra la expresión matemática para el cálculo de la resolución específica:

$$R_p = \frac{50}{\sqrt{(x_{c,1} - x_{c,2})^2 + (y_{c,1} - y_{c,2})^2}} \quad (4.1)$$

Donde las coordenadas $x_{c,i}, y_{c,i}$ con $i = 1, 2$, son dadas en pixeles.

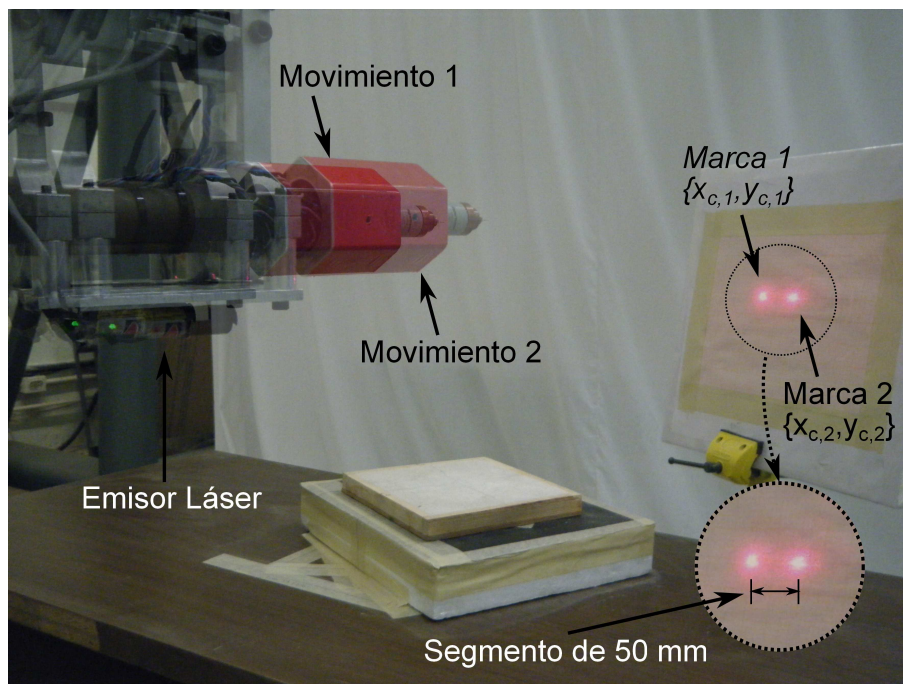


Figura 4.5: Secuencia repetitiva de movimientos de robot Fanuc M16-iB/20T para medición de segmento de cuadrícula sobre tablero de madera.

Teóricamente, al realizar la captura de una escena mediante un sensor opto-electrónico, el tamaño del campo visual cubierto es directamente proporcional a la *resolución específica* de la captura; así como la distancia a la cual se encuentran los elementos de una toma. Es decir, entre más lejos estén dichos objetos, la relación mm/px será mayor. Por ello, un dato a considerar en cada secuencia de experimentos para cuantificar el valor de “resolución específica”, es la distancia desde el observador (cámara real) hasta el plano que contiene la cuadrícula de referencia, siguiendo la respectiva trayectoria del eje óptico de la cámara virtual en cuestión. Esta medición es tomada de forma aproximada, utilizando un cordón de nylon y un flexómetro.

A continuación se presentan los resultados experimentales obtenidos a partir de cuatro secuencias de experimentos realizados, cada una compuesta de 30 ensayos. Entre cada secuencia experimental, el observador (cámara virtual) ha sido colocado a diferentes distancias del plano cuadrículado, manteniendo el *zoom* de la cámara real fijo.

Sec. Exp.	Res. E. (mm/px)	Dev. Std.	D. al Obj. (mm)	Cám. Virtual
1	0.57	0.0081	6610	1
2	0.56	0.0078	6550	2
3	0.67	0.0039	7656	1
4	0.65	0.0116	7640	2

Tabla 4.1: Resultados experimentales de pruebas de determinación de relación mm/px ; en columna 1, se enumera cada secuencia experimental; columna 2, muestra la resolución específica promedio; columna 3, muestra la desviación estándar correspondiente; columna 4, registra la distancia al objetivo; columna 5, señala la cámara virtual empleada.

Los resultados experimentales expuestos en la tabla 4.1, han sido también graficados a continuación, con el fin de apreciar alguna correlación en los datos.

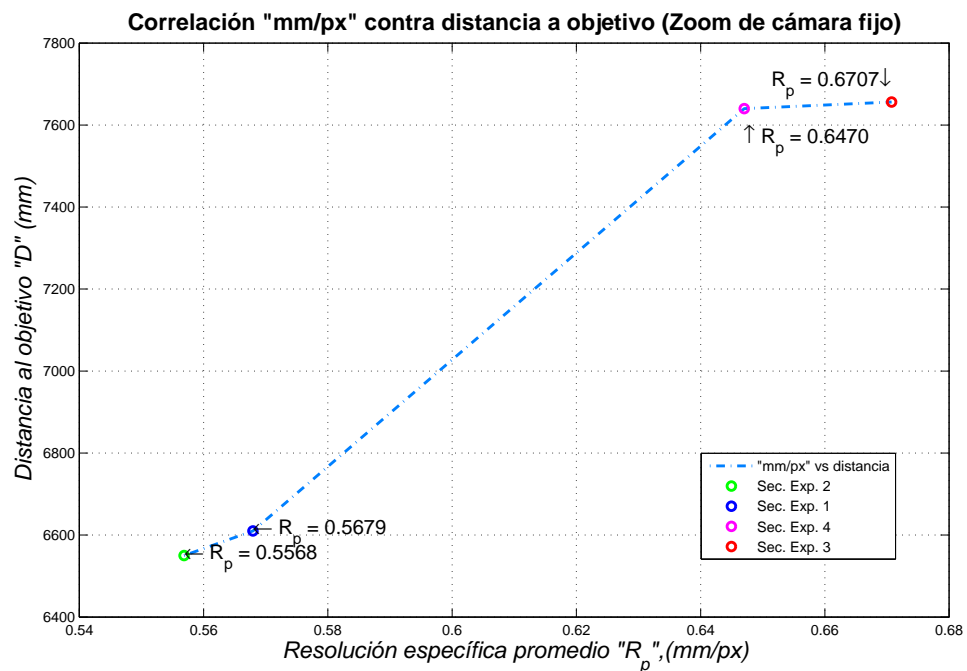


Figura 4.6: Gráfica de correlación entre distancia a objetivo (D) y resolución específica promedio (R_p).

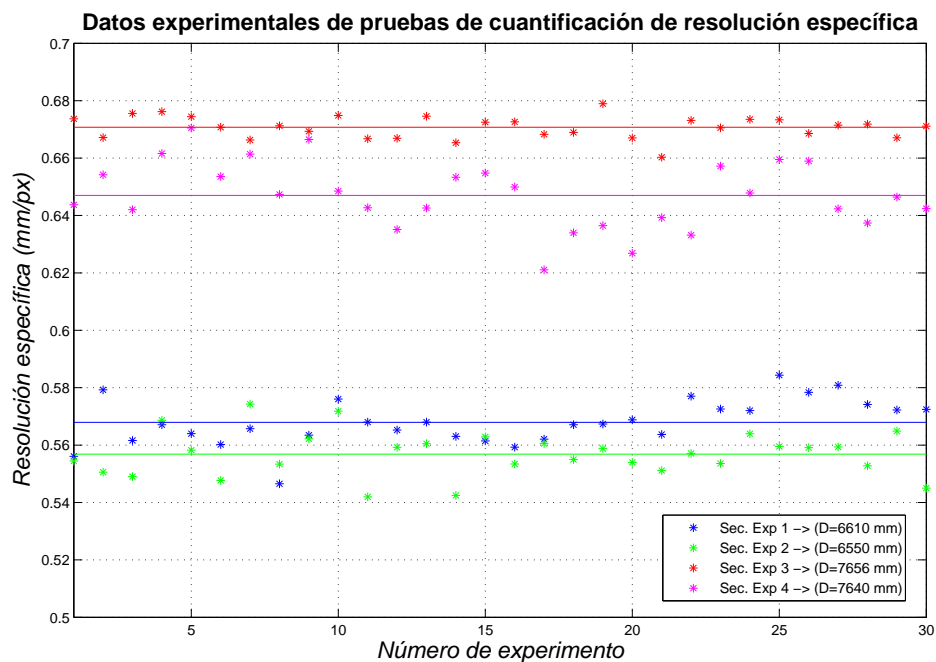


Figura 4.7: Gráfica de resultados por experimento del valor de resolución específica.

De los resultados experimentales anteriores, se observa una fuerte correlación entre los valores de resolución específica (mm/px) calculados y la distancia desde el observador (mm) a la cual se encuentra el plano de análisis. La correlación encontrada es de proporcionalidad directa entre ambos parámetros. Lo que comprueba experimentalmente que a mayor área física cubierta por un espacio de cámara dado, habrá una *resolución específica* mayor, o bien, una menor *densidad de resolución*.

4.3. Experimentos con Maniobras de Posicionamiento

La maniobra de posicionamiento de un manipulador, es el experimento típico para la validación de la eficacia del método de *Manipulación de Espacio de Cámara*, aplicado al caso en el que se ha de controlar un manipulador industrial. La forma de realizar esta validación experimental no difiere mucho al sustituir cámaras reales por cámaras virtuales. Únicamente, son necesarios procesos previos para el procesamiento de imágenes de modo que una sola imagen contenedora de múltiples vistas, pueda ser separada en cada una de ellas. El resto del sistema de visión es desarrollado de forma

similar a uno que emplee varios sensores de video.

En la sección 3.4 del presente estudio, se relata todo lo concerniente a los algoritmos y al flujo de información, necesario para la ejecución de una tarea de posicionamiento por un manipulador industrial, mediante la metodología de control por visión de CSM; empleando en la adquisición de información visual un sistema de cámaras virtuales.

Los experimentos de pruebas de posicionamiento fueron organizados en seis secuencias experimentales, cada una constituida de un total de 30 ensayos. Cada secuencia de experimentos se ha realizado bajo ciertas condiciones particulares; éstas son la resolución específica (Res. Esp.) de cada cámara virtual involucrada en la prueba (C.V.), la distancia aproximada hasta el objetivo (Dist. a Obj.) desde estos observadores, la ubicación del objetivo respecto de la base del manipulador (Loc. de Obj.) y la orientación del plano de pruebas sobre el cual se designa dicho objetivo (Áng. de plano).

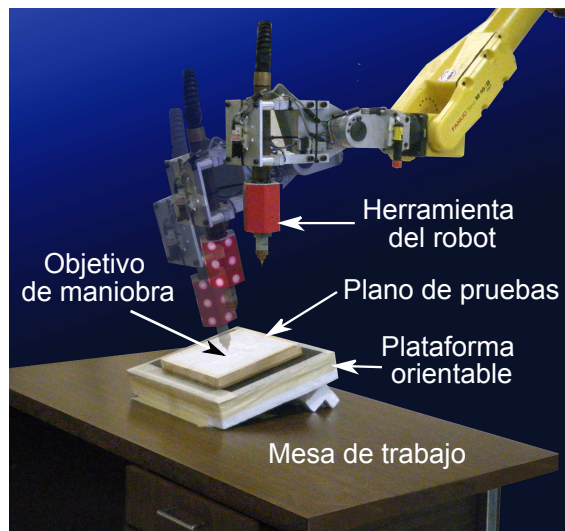


Figura 4.8: Experimento de maniobra de posicionamiento.

El plano de pruebas sobre el cual se define el objetivo de una maniobra de posicionamiento del manipulador, es una caja construida en madera (Figura 4.9), cuyas dimensiones de largo, ancho y altura son $21\text{cm} \times 21\text{cm} \times 2,5\text{cm}$ respectivamente; el interior de la caja se encuentra lleno de plastilina y los cantos superiores de cada lado de la caja permiten servir de guía para un rodillo, utilizado para aplanar la plastilina y aproximarla lo más posible a la geometría de una superficie plana. De entre las ventajas del uso de un material flexible como la plastilina, para confor-

mar una superficie de pruebas, es que es posible crear y borrar trazos con facilidad, para definir y re-definir un objetivo de maniobra. Otra ventaja es la capacidad de la superficie para absorber una posible colisión del efector final del manipulador, inminente tras un fallo en el flujo de información durante una prueba de posicionamiento; situación que es poco probable, no obstante posible, al detectarse marcas visuales erróneas durante el lapso en el que se realiza el procesamiento de imágenes.

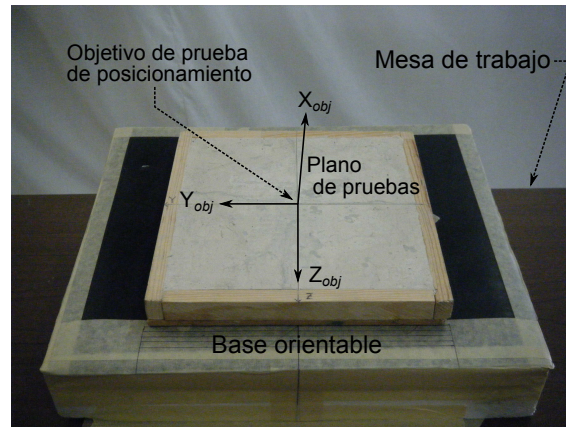


Figura 4.9: Plano para localización de objetivo en pruebas de posicionamiento.

Los datos registrados de cada prueba son el error de posición tridimensional del extremo de la herramienta del manipulador, y el error de orientación de la misma, respecto de la normal al plano donde se encuentra el objetivo de maniobra. El error de posición tridimensional es determinado mediante la definición de un sistema coordenado $\{X_{obj}, Y_{obj}, Z_{obj}\}$, cuyo origen es el objetivo de la prueba, y su orientación es acorde al plano donde se encuentra dicho objetivo. Cada eje coordenado de este sistema, define una dirección sobre la cual se realiza la medición del error de posición. En total se realizan tres mediciones previas al cálculo del error de posición tridimensional, estas mediciones se realizan sobre los ejes $X_{obj}, Y_{obj}, Z_{obj}$ y determinan al respectiva desviación del efector final de la herramienta del manipulador, del objetivo de maniobra. Los errores de posición sobre cada eje coordenado $X_{obj}, Y_{obj}, Z_{obj}$ serán abreviados como E_{Px}, E_{Py}, E_{Pz} por motivos de practicidad. El error de posición tridimensional (E_{Pxyz}) se calcula mediante la norma euclidiana de los errores de posición E_{Px}, E_{Py}, E_{Pz} , como sigue:

$$E_{Pxyz} = \sqrt{E_{Px}^2 + E_{Py}^2 + E_{Pz}^2} \quad (4.2)$$

El instrumento de medición empleado para la cuantificación de los errores de posición

es un vernier (calibrador o pie de rey) y adicional a este, se emplea un compás para trasladar medidas, para los casos en los que la falta de espacio impide obtener una medición directa con el vernier.

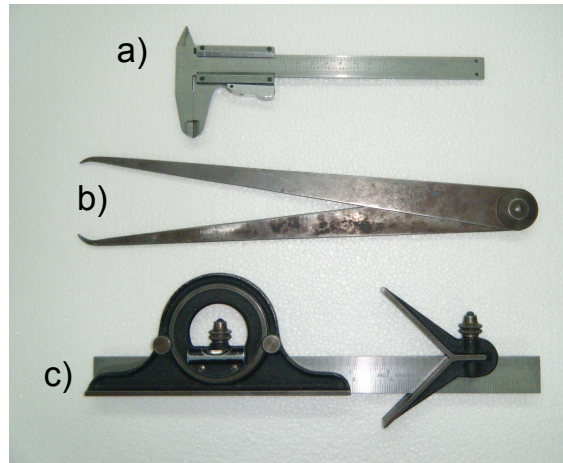


Figura 4.10: Instrumentos de medición empleados para cuantificar el error de posición y orientación en una maniobra de posicionamiento; a) Vernier; b) Compás; c) Escuadra universal

Al final del total de experimentos de cada secuencia experimental, a partir de los errores de posición obtenidos $(E_{Px}, E_{Py}, E_{Pz}, E_{Pxyz})$, se calcula el error absoluto promedio de estos $(\bar{E}_{Px}, \bar{E}_{Py}, \bar{E}_{Pz}, \bar{E}_{Pxyz})$ y la respectiva desviación estándar poblacional (σ) .

$$\bar{E}_{Px} = \sum_{i=1}^{30} \left(\frac{E_{Px}^{(i)}}{30} \right) \quad (4.3)$$

$$\bar{E}_{Py} = \sum_{i=1}^{30} \left(\frac{E_{Py}^{(i)}}{30} \right) \quad (4.4)$$

$$\bar{E}_{Pz} = \sum_{i=1}^{30} \left(\frac{E_{Pz}^{(i)}}{30} \right) \quad (4.5)$$

$$\bar{E}_{Pxyz} = \sum_{i=1}^{30} \left(\frac{E_{Pxyz}^{(i)}}{30} \right) \quad (4.6)$$

$$\sigma = \left(\frac{1}{n-1} \sum_{i=1}^{n=30} (E_P - \bar{E}_P)^2 \right)^{1/2} \quad (4.7)$$

Para controlar la orientación de la herramienta del manipulador, se ha desarrollado una subrutina incluida en el algoritmo de estimación de la configuración interna del manipulador (Sección 3.4.4) con el cual, dada una orientación arbitraria del plano de pruebas, la herramienta del manipulador al final de una maniobra de posicionamiento llegará con su eje longitudinal de forma paralela a la normal del plano de pruebas. El método para comprobar la eficacia de esta subrutina, es proponiendo orientaciones arbitrarias del plano de pruebas; no obstante, es importante guardar registro del orden y sentido de las rotaciones aplicadas al plano entre cada secuencia experimental. Dado que en la ubicación en donde se ha trazado el objetivo de maniobra, ha sido definido un sistema de coordenadas $X_{obj}, Y_{obj}, Z_{obj}$, dicho sistema sirve de referencia para rotar el plano de pruebas. El estado natural de dicho sistema de coordenado, es siendo su eje coordenado " X_{obj} " paralelo al eje Z_0 del origen de coordenadas del robot y siendo su eje " Y_{obj} " paralelo y de sentido contrario al eje Y_0 en el origen de coordenadas del robot. La secuencia de rotaciones empleadas para el plano de pruebas son, una rotación φ_1 alrededor del eje X_{obj} , seguida de una rotación negativa φ_2 alrededor del eje Y'_{obj} (Figura 4.11).

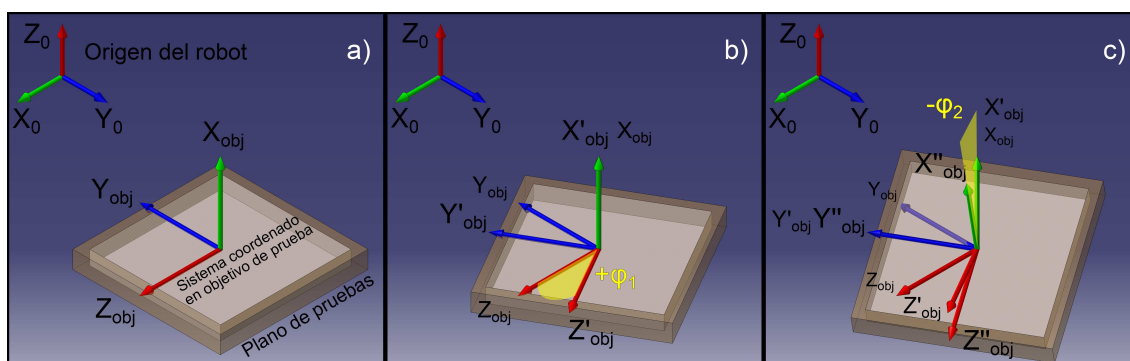


Figura 4.11: Orden de rotaciones registradas en el plano de pruebas para cada secuencia experimental; a): orientación original del plano de pruebas; b) primera rotación, siendo alrededor del eje X_{obj} en un ángulo $+\varphi_1$; c) segunda rotación, siendo alrededor del eje Y'_{obj} en un ángulo $-\varphi_2$.

El error de orientación es medido con la ayuda de una escuadra universal, y la referencia de medición es la normal al plano donde esta situado el objetivo. Puesto que en cada prueba de posicionamiento, idealmente el eje longitudinal de la herramienta

del manipulador debería arribar a su objetivo de forma paralela al eje normal al plano de pruebas; el ángulo formado entre estos dos ejes sirve para definir el error de orientación. Sin embargo, en total son medidos dos ángulos, cada uno de los cuales están definidos alrededor de los ejes coordenados Z_{obj} y Y_{obj} , que forman un plano paralelo a la superficie de pruebas y que pertenecen al sistema coordenado de referencia, con origen en el objetivo. El error de orientación medido sobre el eje Z_{obj} recibe el nombre de α y el error de orientación medido sobre el eje Y_{obj} recibe el nombre de β .

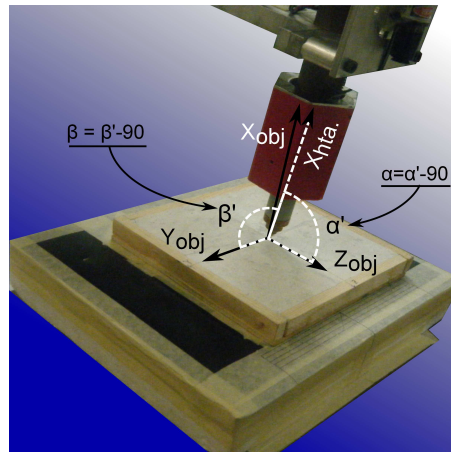


Figura 4.12: Referencias para la medición del orientación de la herramienta del manipulador en el objetivo

Al final de cada secuencia experimental se calculan los errores de posición promedio $\bar{\alpha}$ y $\bar{\beta}$ con la respectiva desviación estándar poblacional σ .

$$\bar{\alpha} = \sum_{i=1}^{30} \left(\frac{\alpha^{(i)}}{30} \right) \quad (4.8)$$

$$\bar{\beta} = \sum_{i=1}^{30} \left(\frac{\beta^{(i)}}{30} \right) \quad (4.9)$$

A continuación se presentan los resultados obtenidos en cada una de las seis secuencias de pruebas de posicionamiento.

4.3.1. Primera Secuencia Experimental de Pruebas de Posicionamiento

Las condiciones bajo las cuales se realizó la primera secuencia experimental son las siguientes:

Datos de primera secuencia experimental								
Loc. de Obj. (mm)			Dist. a Obj. (mm)		Res. Esp. (mm/px)		Áng. de plano. (°)	
X	Y	Z	C.V. 1	C.V. 2	C.V. 1	C.V. 2	φ_1	φ_2
636.2	1774.4	-1521.1	6610	6550	0.57	0.56	30	17.5

Tabla 4.2: Datos de primera secuencia experimental

Los resultados condensados de la primera secuencia experimental son los siguientes:

Resultados de primera secuencia experimental					
Errores de posición promedio (mm)				Errores de orientación promedio (°)	
\bar{E}_{Px}	\bar{E}_{Py}	\bar{E}_{Pz}	\bar{E}_{Pxyz}	$\bar{\alpha}$	$\bar{\beta}$
0.3	1.2	3.2	3.4	4.4	4.6

Tabla 4.3: Datos de primera secuencia experimental.

Los resultados experimentales se presentan ahora de forma gráfica:

Error de posición tridimensional en maniobras de posicionamiento de secuencia experimental..1

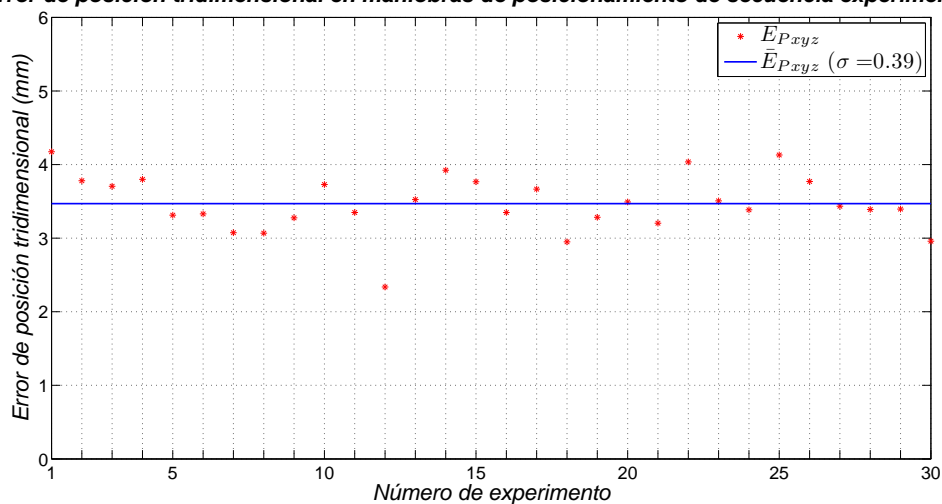
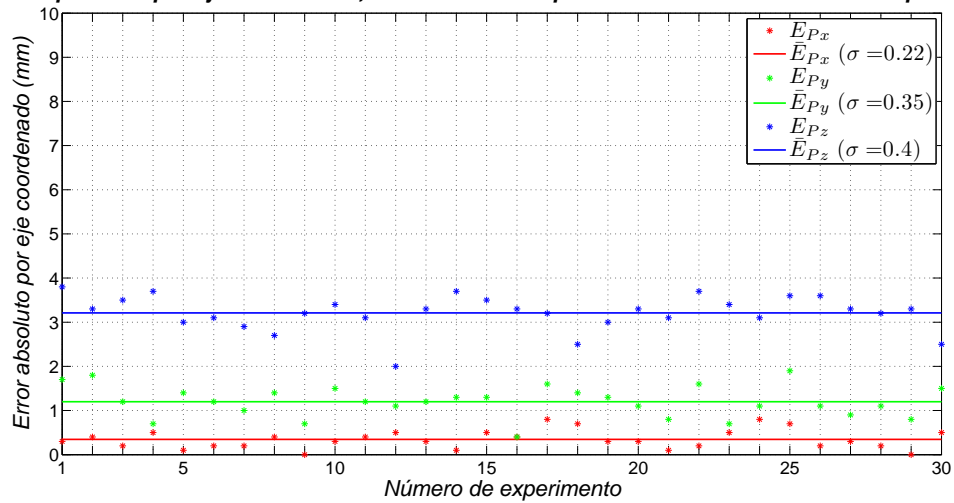
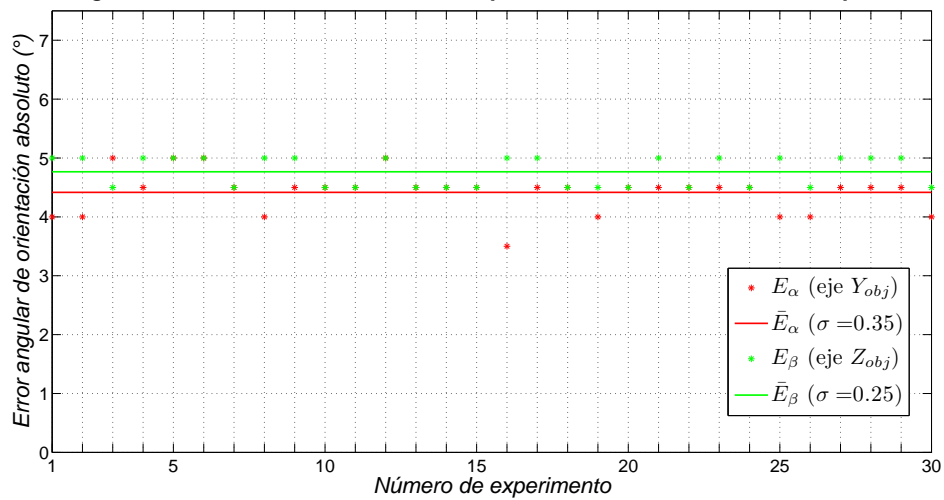


Figura 4.13: Gráfica del error tridimensional por experimento, de primera secuencia experimental.

Error de posición por eje coordenado, en maniobras de posicionamiento de secuencia experimental..1**Figura 4.14:** Gráfica del error de posición por eje coordenado, en maniobras de posicionamiento de primera secuencia experimental.**Error angular de orientación en maniobras de posicionamiento de secuencia experimental..1****Figura 4.15:** Gráfica del error de orientación angular, en maniobras de posicionamiento de primera secuencia experimental.

4.3.2. Segunda Secuencia Experimental de Pruebas de Posicionamiento

Las condiciones bajo las cuales se realizó la segunda secuencia experimental son las siguientes:

Datos de segunda secuencia experimental								
Loc. de Obj. (mm)			Dist. a Obj.(mm)		Res. Esp.(mm/px)		Áng. de plano. (°)	
X	Y	Z	C.V. 1	C.V. 2	C.V. 1	C.V. 2	φ_1	φ_2
694.6	1812.0	-1520.9	6610	6550	0.57	0.56	0	16.5

Tabla 4.4: Datos de primera secuencia experimental

Los resultados condensados de la segunda secuencia experimental son los siguientes:

Resultados de segunda secuencia experimental					
Errores de posición promedio (mm)				Errores de orientación promedio (°)	
\bar{E}_{Px}	\bar{E}_{Py}	\bar{E}_{Pz}	\bar{E}_{Pxyz}	$\bar{\alpha}$	$\bar{\beta}$
0.3	0.7	2.7	2.8	2.3	6.3

Tabla 4.5: Resultados condensados de segunda secuencia experimental

Los resultados experimentales se presentan ahora de forma gráfica:

Error de posición tridimensional en maniobras de posicionamiento de secuencia experimental..2

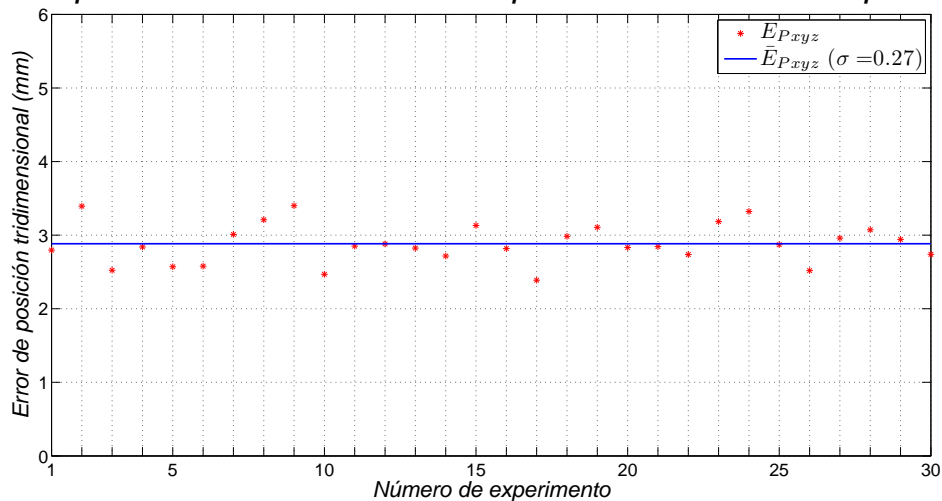


Figura 4.16: Gráfica del error tridimensional por experimento, de segunda secuencia experimental.

Error de posición por eje coordenado, en maniobras de posicionamiento de secuencia experimental..2

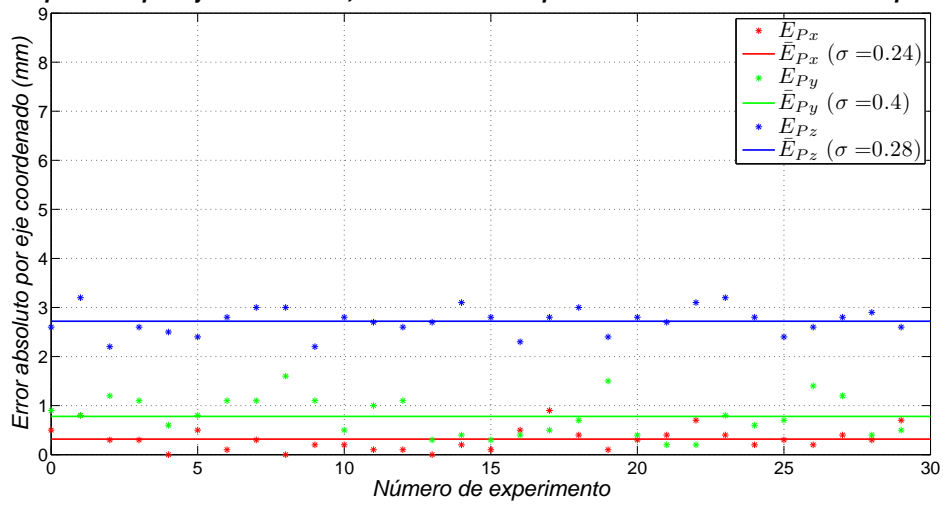


Figura 4.17: Gráfica del error de posición por eje coordenado, en maniobras de posicionamiento de segunda secuencia experimental.

Error angular de orientación en maniobras de posicionamiento de secuencia experimental..2

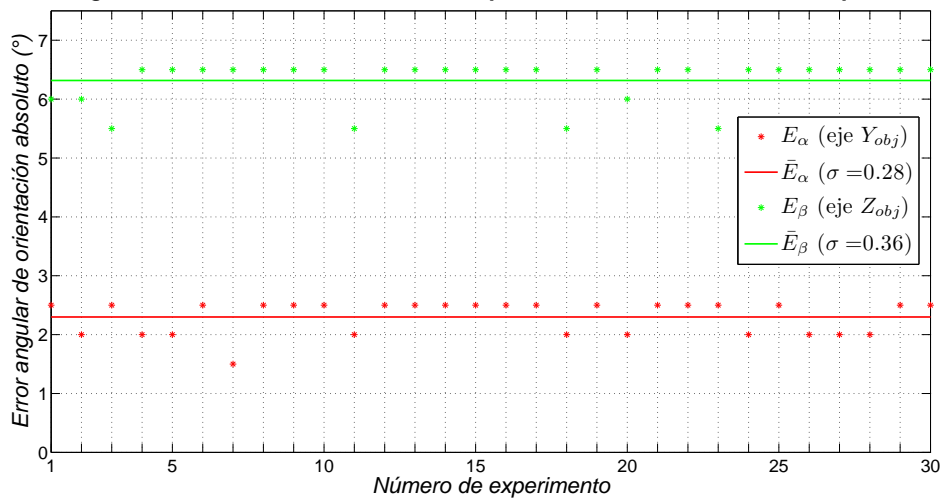


Figura 4.18: Gráfica del error de orientación angular, en maniobras de posicionamiento de segunda secuencia experimental.

4.3.3. Tercera Secuencia Experimental de Pruebas de Posicionamiento

Las condiciones bajo las cuales se realizó la tercera secuencia experimental son las siguientes:

Datos de tercera secuencia experimental								
Loc. de Obj. (mm)			Dist. a Obj. (mm)		Res. Esp.(mm/px)		Áng. de plano. (°)	
X	Y	Z	C.V. 1	C.V. 2	C.V. 1	C.V. 2	φ_1	φ_2
650.9	1911.7	-1522.5	6610	6550	0.57	0.56	-30	16.5

Tabla 4.6: Datos de primera secuencia experimental

Los resultados condensados de la tercera secuencia experimental son los siguientes:

Resultados de tercera secuencia experimental					
Errores de posición promedio (mm)				Errores de orientación promedio (°)	
\bar{E}_{Px}	\bar{E}_{Py}	\bar{E}_{Pz}	\bar{E}_{Pxyz}	$\bar{\alpha}$	$\bar{\beta}$
0.2	3.1	1.2	3.4	1.2	3.9

Tabla 4.7: Resultados condesados de tercera secuencia experimental

Los resultados experimentales se presentan ahora de forma gráfica:

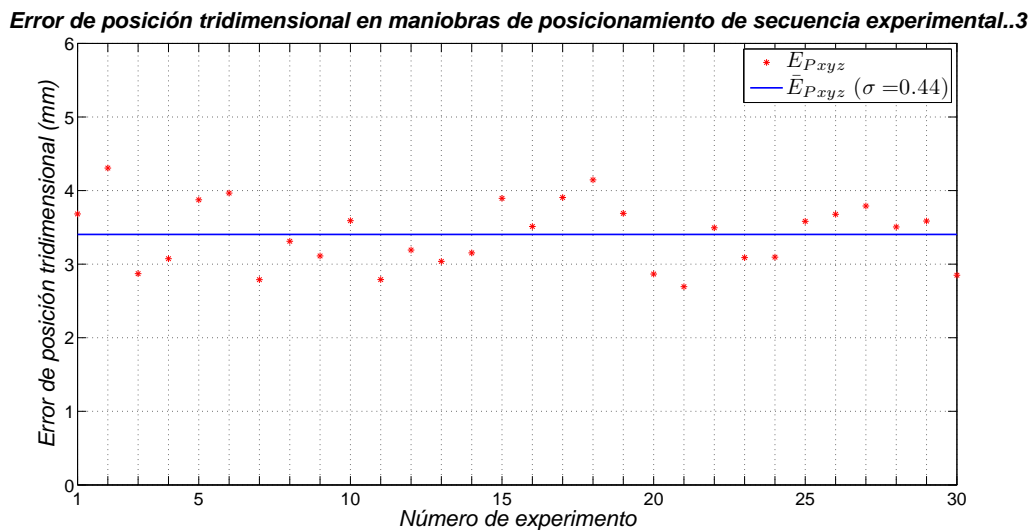
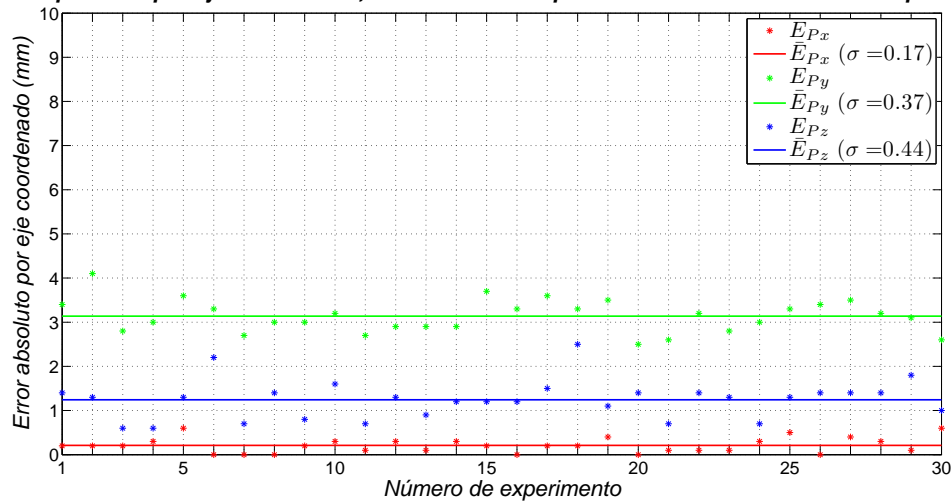
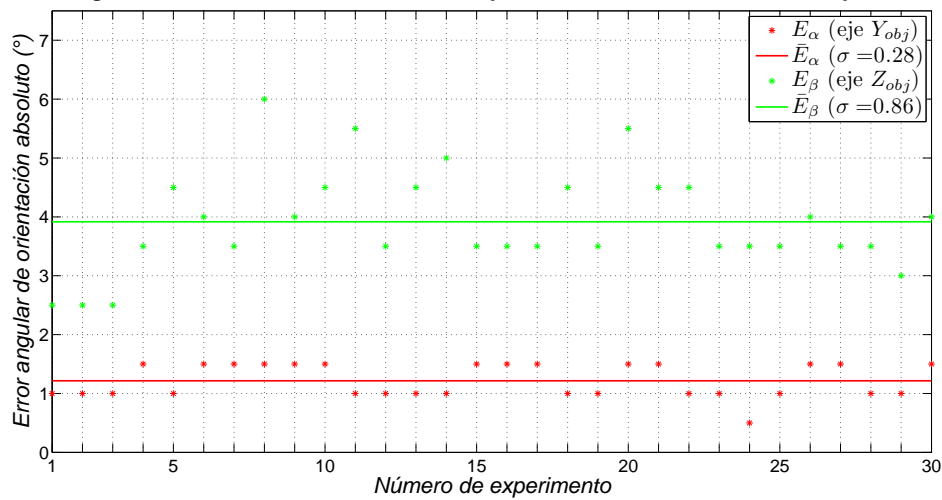


Figura 4.19: Gráfica del error tridimensional por experimento, de tercera secuencia experimental.

Error de posición por eje coordenado, en maniobras de posicionamiento de secuencia experimental..3**Figura 4.20:** Gráfica del error de posición por eje coordenado, en maniobras de posicionamiento de tercera secuencia experimental.**Error angular de orientación en maniobras de posicionamiento de secuencia experimental..3****Figura 4.21:** Gráfica del error de orientación angular, en maniobras de posicionamiento de tercera secuencia experimental.

4.3.4. Cuarta Secuencia Experimental de Pruebas de Posicionamiento

Las condiciones bajo las cuales se realizó la cuarta secuencia experimental son las siguientes:

Datos de cuarta secuencia experimental								
Loc. de Obj. (mm)			Dist. a Obj. (mm)		Res. Esp. (mm/px)		Áng. de plano. (°)	
X	Y	Z	C.V. 1	C.V. 2	C.V. 1	C.V. 2	φ_1	φ_2
641.6	1794.7	-1524.3	7656	7640	0.67	0.65	30	17.5

Tabla 4.8: Datos de primera secuencia experimental

Los resultados condensados de la cuarta secuencia experimental son los siguientes:

Resultados de cuarta secuencia experimental					
Errores de posición promedio (mm)				Errores de orientación promedio (°)	
\bar{E}_{Px}	\bar{E}_{Py}	\bar{E}_{Pz}	\bar{E}_{Pxyz}	$\bar{\alpha}$	$\bar{\beta}$
1.6	1.3	1.8	2.9	3.9	3.5

Tabla 4.9: Resultados condensados de cuarta secuencia experimental

Los resultados experimentales se presentan ahora de forma gráfica:

Error de posición tridimensional en maniobras de posicionamiento de secuencia experimental..4

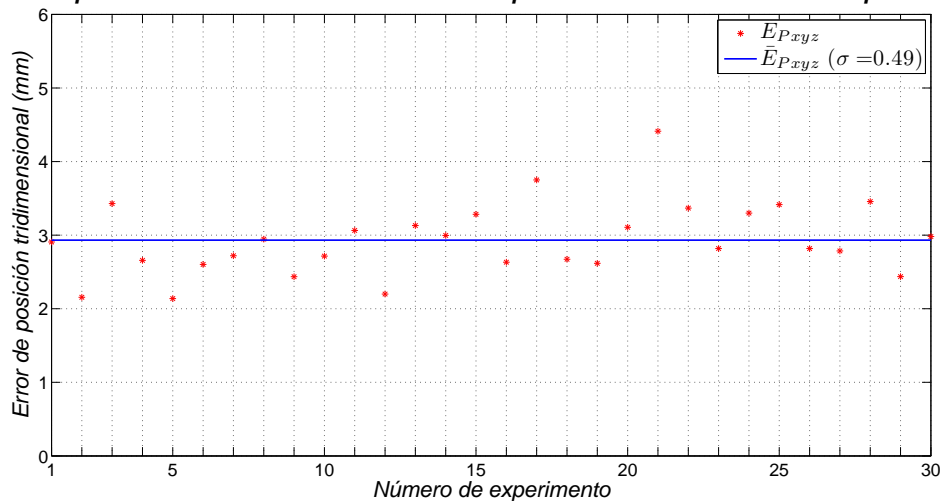
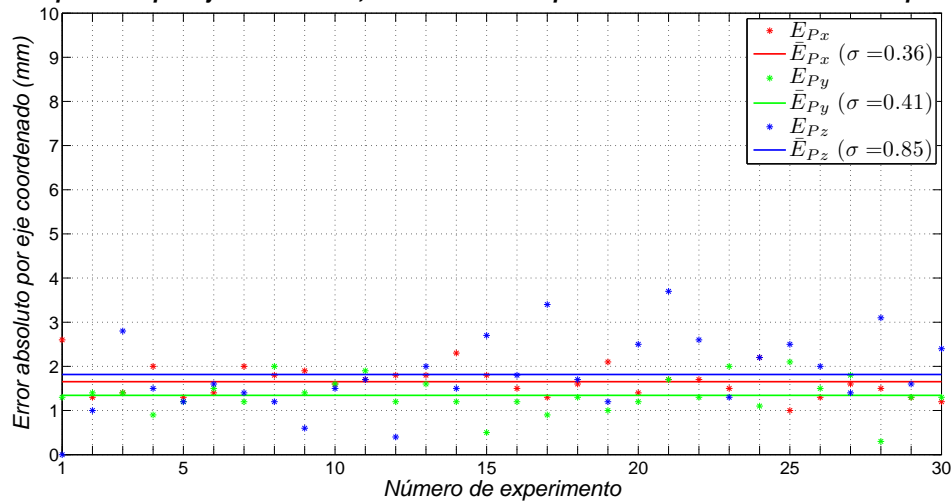
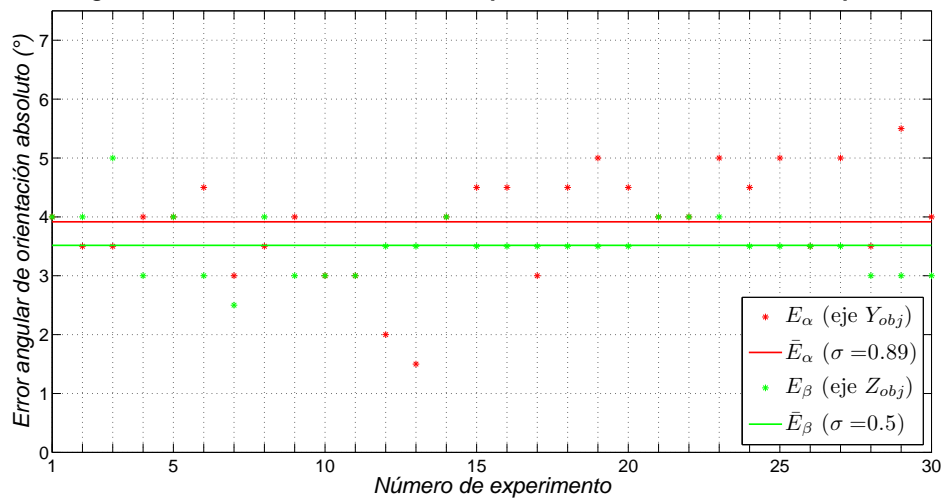


Figura 4.22: Gráfica del error tridimensional por experimento, de cuarta secuencia experimental.

Error de posición por eje coordenado, en maniobras de posicionamiento de secuencia experimental..4**Figura 4.23:** Gráfica del error de posición por eje coordenado, en maniobras de posicionamiento de cuarta secuencia experimental.**Error angular de orientación en maniobras de posicionamiento de secuencia experimental..4****Figura 4.24:** Gráfica del error de orientación angular, en maniobras de posicionamiento de cuarta secuencia experimental.

4.3.5. Quinta Secuencia Experimental de Pruebas de Posicionamiento

Las condiciones bajo las cuales se realizó la quinta secuencia experimental son las siguientes:

Datos de quinta secuencia experimental								
Loc. de Obj.(mm)			Dist. a Obj. (mm)		Res. Esp.(mm/px)		Áng. de plano. (°)	
<i>X</i>	<i>Y</i>	<i>Z</i>	C.V. 1	C.V. 2	C.V. 1	C.V. 2	φ_1	φ_2
713	1830.3	-1522.5	7656	7640	0.67	0.65	0	17.5

Tabla 4.10: Datos de primera secuencia experimental

Los resultados condensados de la quinta secuencia experimental son los siguientes:

Resultados de quinta secuencia experimental					
Errores de posición promedio (mm)				Errores de orientación promedio (°)	
\bar{E}_{Px}	\bar{E}_{Py}	\bar{E}_{Pz}	\bar{E}_{Pxyz}	$\bar{\alpha}$	$\bar{\beta}$
0.6	0.4	3.9	4.0	2.5	6.8

Tabla 4.11: Resultados condensados de quinta secuencia experimental

Los resultados experimentales se presentan ahora de forma gráfica:

Error de posición tridimensional en maniobras de posicionamiento de secuencia experimental..5

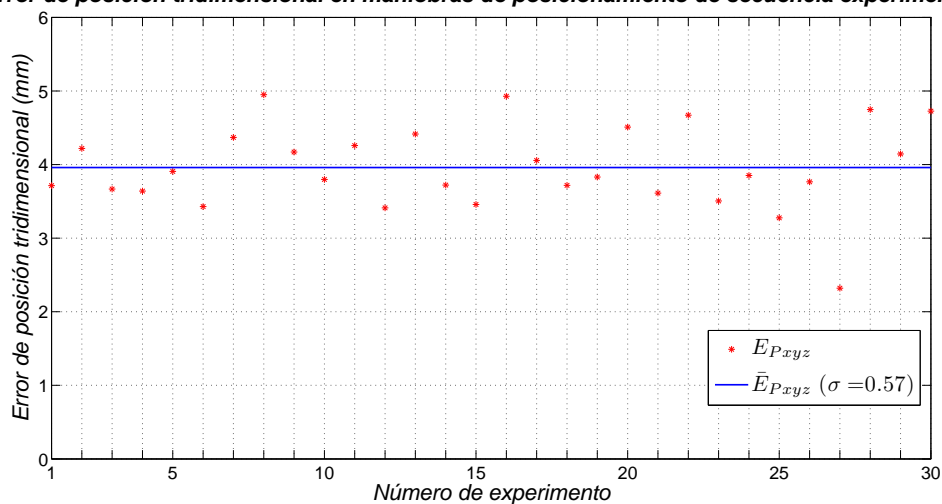


Figura 4.25: Gráfica del error tridimensional por experimento, de quinta secuencia experimental.

Error de posición por eje coordenado, en maniobras de posicionamiento de secuencia experimental..5

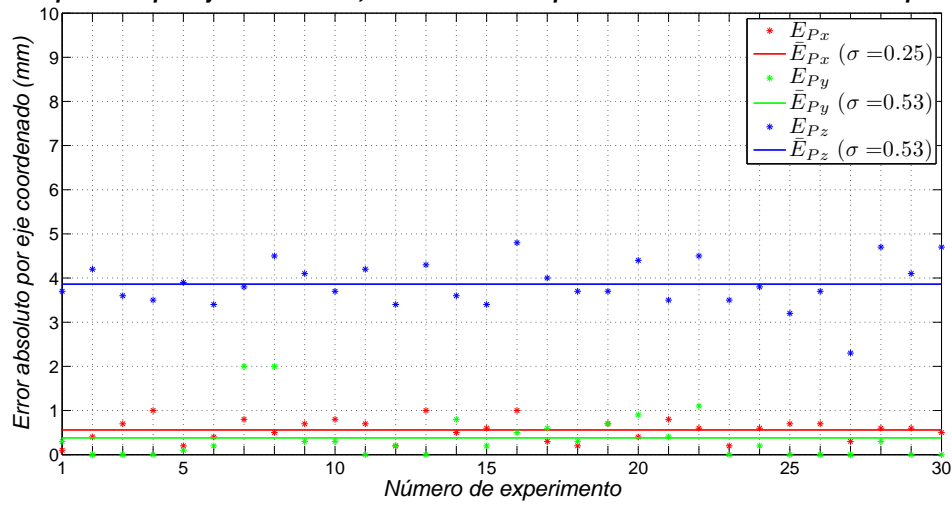


Figura 4.26: Gráfica del error de posición por eje coordenado, en maniobras de posicionamiento de quinta secuencia experimental.

Error angular de orientación en maniobras de posicionamiento de secuencia experimental..5

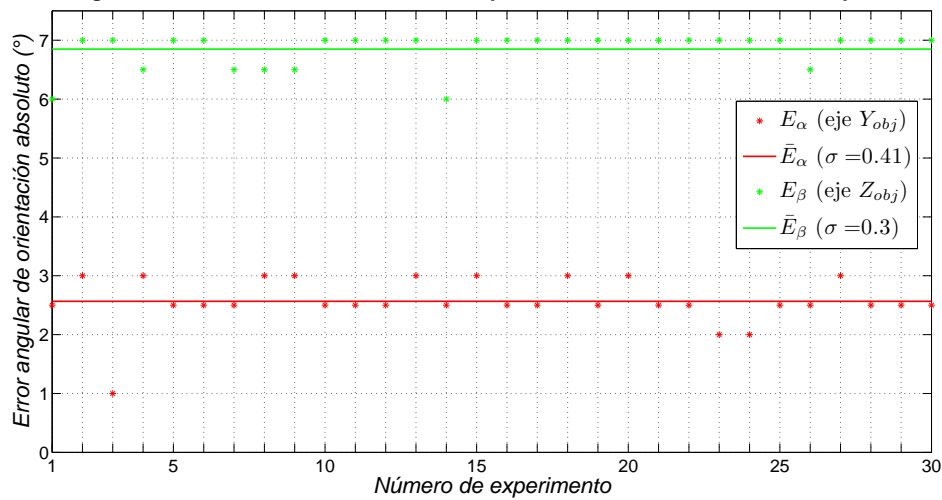


Figura 4.27: Gráfica del error de orientación angular, en maniobras de posicionamiento de quinta secuencia experimental.

4.3.6. Sexta Secuencia Experimental de Pruebas de Posicionamiento

Las condiciones bajo las cuales se realizó la sexta secuencia experimental son las siguientes:

Datos de sexta secuencia experimental								
Loc. de Obj. (mm)			Dist. a Obj. (mm)		Res. Esp. (mm/px)		Áng. de plano. (°)	
X	Y	Z	C.V. 1	C.V. 2	C.V. 1	C.V. 2	φ_1	φ_2
653.2	1908.7	-1524	7656	7640	0.67	0.65	-30	17.5

Tabla 4.12: Datos de primera secuencia experimental

Los resultados condensados de la sexta secuencia experimental son los siguientes:

Resultados de sexta secuencia experimental					
Errores de posición promedio (mm)				Errores de orientación promedio (°)	
\bar{E}_{Px}	\bar{E}_{Py}	\bar{E}_{Pz}	\bar{E}_{Pxyz}	$\bar{\alpha}$	$\bar{\beta}$
0.5	2.3	3.9	4.6	0.8	4.1

Tabla 4.13: Resultados condensados de sexta secuencia experimental

Los resultados experimentales se presentan ahora de forma gráfica:

Error de posición tridimensional en maniobras de posicionamiento de secuencia experimental..6

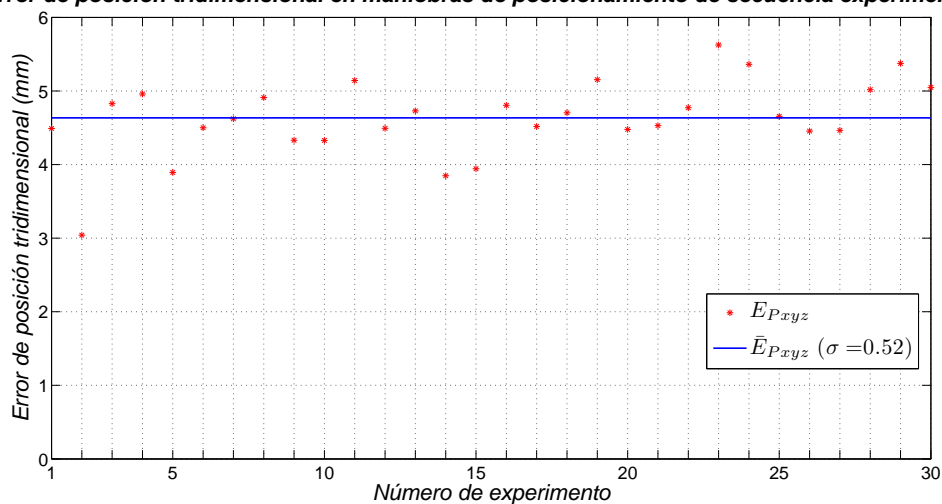
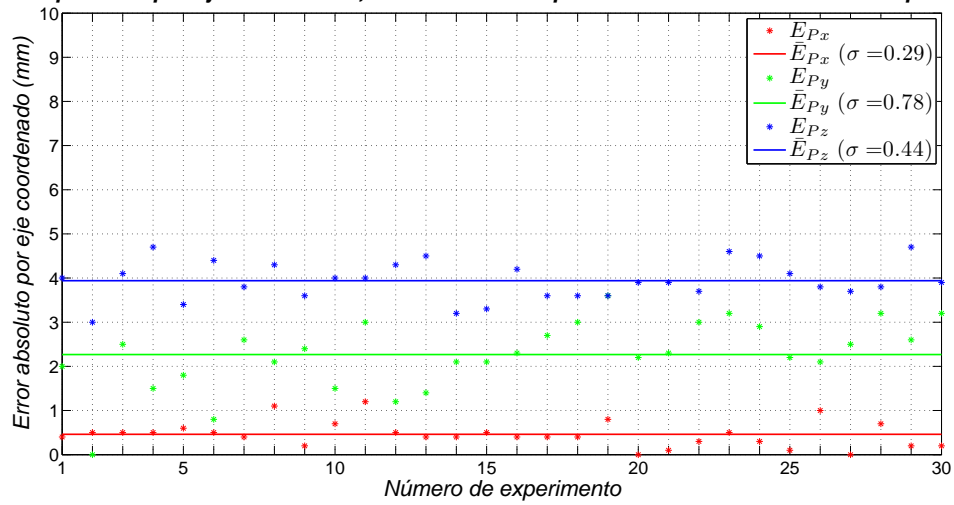
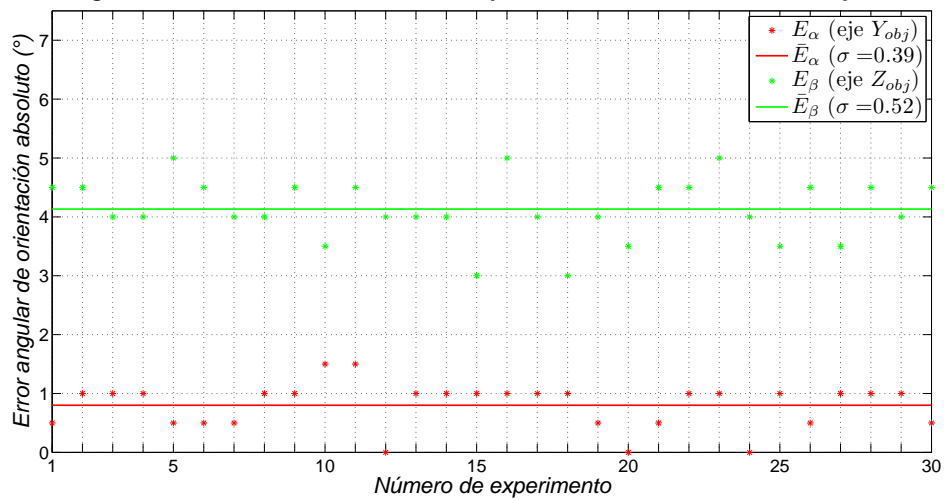


Figura 4.28: Gráfica del error tridimensional por experimento, de sexta secuencia experimental.

Error de posición por eje coordenado, en maniobras de posicionamiento de secuencia experimental..6**Figura 4.29:** Gráfica del error de posición por eje coordenado, en maniobras de posicionamiento de sexta secuencia experimental.**Error angular de orientación en maniobras de posicionamiento de secuencia experimental..6****Figura 4.30:** Gráfica del error de orientación angular, en maniobras de posicionamiento de sexta secuencia experimental.

4.3.7. Discusión de Resultados de Experimentos con Maniobras de Posicionamiento

Del total de secuencias experimentales de pruebas de posicionamiento, se pueden verificar algunas correlaciones entre las condiciones bajo las cuales se realizan los experimentos y los resultados arrojados respecto al error de posición y al error de orientación. Las condiciones de cada secuencia de pruebas generan cierta pauta para organizar dichas secuencias en dos grupos, cada uno con diferentes resoluciones específicas. El primer grupo de pruebas está constituido por las tres primeras secuencias experimentales y se segundo grupo por las tres secuencias restantes. Estos dos grupos se diferencian el uno del otro por las resoluciones específicas a las que han sido condicionados. El primer grupo de pruebas emplea una resolución específica menor a la del segundo. No obstante, ambos grupos de secuencias experimentales guardan cierta semejanza el uno del otro, que radica en la orientaciones elegidas del plano de pruebas para cada secuencia experimental individual; la primera secuencia de experimentos del primer grupo, guarda una orientación del plano de pruebas similar a la primera serie de experimental del segundo grupo, así como la segunda secuencia de experimentos del primer grupo, con la segunda serie de pruebas del segundo grupo y así sucesivamente.

Es importante aclarar que la configuración del sistema catadióptrico en todas las secuencias experimentales, se a procurado de forma simétrica, es decir, las distancias entre los espejos centrales y los laterales (D01,D02), son similares entre sí, por lo que entre cada grupo de secuencias experimentales, las resoluciones específicas por cada cámara son similares.

Para buscar alguna correlación entre los datos resultantes de las pruebas y sus respectivas condiciones experimentales, es necesario generar supuestos iniciales, para prever alguna tendencia en los resultados. Dichos supuestos se presentan a continuación:

1. El primer supuesto consiste en que, la resolución específica medida sobre un plano cercano al lugar en donde se lleva a cabo el control de un manipulador, tiene influencia en la precisión con la cual dicho manipulador llegará a su objetivo. A mayor resolución específica medida en la zona cercana a la maniobra, menor será la precisión con la que el robot es controlado y este llega a su objetivo.

2. Un sistema de cámaras virtuales que emplee espejos convencionales planos en su fabricación, podría ser ópticamente más aberrante que un sistema de adquisición de imágenes formado puramente de los mismos sensores opto-electrónicos, utilizados directamente. Las aberraciones ópticas presentes en el sistema de cámaras virtuales, pueden afectar la precisión con la cual el sistema de control por visión *CSM* se desempeña.
3. Puesto que el control de la orientación del manipulador industrial mediante *CSM*, parte de la caracterización de la superficie correspondiente al plano de pruebas, y dado que para dicho efecto es necesario predecir la posición la tridimensional de una serie de puntos a partir sus respectivas proyecciones en espacio de cámara, la resolución específica de los sensores involucrados, debe ser también un factor relacionado al control preciso de la orientación de la herramienta del robot.

Los resultados experimentales anteriores, se condensan a continuación en una tabla comparativa en donde se agrupan las secuencias experimentales de acuerdo su resolución específica:

Clasificación de secuencias experimentales								
Grupo 1 (Res. Esp. C.V. 1.: 0.57 mm/px, Res. Esp. C.V. 2.: 0.56 mm/px)					Grupo 2 (Res. Esp. C.V. 2.: 0.67 mm/px, Res. Esp. C.V. 2.: 0.65 mm/px)			
	Sec. Exp. 1	Sec. Exp. 2	Sec. Exp. 3	Prom.	Sec. Exp. 4	Sec. Exp. 5	Sec. Exp. 6	Prom.
φ_1 (°)	30	0	-30	-	30	0	-30	-
φ_2 (°)	17.5	16.5	16.5	-	17.5	17.5	17.5	-
\bar{E}_{P_x} (mm)	0.3	0.3	0.2	0.26	1.6	0.6	0.5	0.9
\bar{E}_{P_y} (mm)	1.2	0.7	3.1	1.67	1.3	0.4	2.3	1.34
\bar{E}_{P_z} (mm)	3.2	2.7	1.2	2.36	1.8	3.9	3.9	3.2
$\bar{E}_{P_{xyz}}$ (mm)	3.4	2.8	3.4	3.2	2.9	4.0	4.6	3.8
$\bar{\alpha}$ (°)	4.4	2.3	1.2	2.6	3.9	2.5	0.8	2.4
$\bar{\beta}$ (°)	4.6	6.3	3.9	4.9	3.5	6.8	4.1	4.8

Tabla 4.14: Agrupamiento de resultados de secuencias experimentales.

A partir de la tabla anterior, se pueden discutir las hipótesis propuestas dadas para los resultados experimentales obtenidos de las series de experimentos con maniobras de posicionamiento; los razonamientos concluyentes se enumeran a continuación:

1. El primer supuesto se considera como válido. Es decir, a un valor más pequeño de resolución específica (mm/px), existe un incremento en la precisión con la cual se efectúan las maniobras de posicionamiento. Al considerar los promedios globales del error de posicionamiento tridimensional \bar{E}_{Pxyz} de ambos grupos de secuencias experimentales, se observa un menor error en el grupo número uno, el cual posee en sus cámaras una menor magnitud de resolución específica. Al analizar las secuencias experimentales de cada grupo individualmente, se puede observar que la única secuencia experimental que no cumple con el supuesto, es la secuencia experimental número cuatro; no obstante, es la única secuencia de experimentos que registra un alza significativa del error \bar{E}_{Px} , arrojando un valor de $1.6 mm$, que es un valor muy superior al valor próximo en magnitud registrado, perteneciente a la secuencia experimental cinco, con un valor de $0.6 mm$; por ello el primer supuesto se sigue considerando como válido, dado que los demás errores \bar{E}_{Py} y \bar{E}_{Pz} , de la secuencia experimental número cuatro, se mantienen en valores intermedios.
2. Tratando el supuesto número dos, si bien por el momento no se tiene un procedimiento concreto para cuantificar las aberraciones ópticas presentes en los espejos, se tiene certeza de que están en dichos elementos reflexivos, ya que al observar el reflejo de algún objeto en éstos, se puede distinguir la distorsión del objeto reflejado a simple vista, siendo más notoria dicha distorsión cuando se observa el reflejo de un objeto desde un punto de observación distante a la superficie del espejo. Un ejemplo de esta distorsión directamente perceptible, se ofrece en la figura 4.31.



Figura 4.31: Distorsión óptica en espejo del sistema de cámaras virtuales.

En trabajos anteriores [26, 27], el error reportado en maniobras de posicionamiento va de 0.6 a 0.8 mm de desviación tridimensional, del efector final del manipulador de su objetivo. No obstante, en dichos trabajos, se utiliza una cámara individual para cada perspectiva cubierta, del área de trabajo del robot, manejando individualmente una resolución de 1600×1200 píxeles. Mediante el sistema de cámaras virtuales aquí empleado, el error de posición tridimensional, en maniobras de posicionamiento ronda de entre 3.2 a 3.8 mm ; si bien la resolución del sistema de cámaras virtuales es comparativamente menor, de 800×1200 en cada perspectiva del objetivo, se cree que una parte del error se debe a la distorsión de las imágenes captadas, debido a los espejos. Una forma correcta de hacer la comparación de la precisión del sistema de cámaras virtuales aquí desarrollado, y los sistemas de adquisición de imágenes empleados en otros trabajos, sería bajo conocimiento de la resolución específica (mm/px) de los trabajos anteriores, dato que no es reportado; no obstante, se cree que dicho valor de resolución específica es mayor al menos en los trabajos precedentes en donde se emplea el modelo de perspectiva recursivo.

Una forma más para atribuir una parte del error de posicionamiento tridimensional, a las aberraciones ópticas presentes en los elementos reflexivos del sistema de cámaras virtuales, es revisando el desempeño del algoritmo de estimación de juntas. Dicho algoritmo minimiza una función de error, en la cual se estima la mejor configuración posible del manipulador, con la que es posible empatar una serie de marcas visuales en espacio de cámara pertenecientes al objetivo de maniobra, con un número igual de marcas artificiales generadas sobre el extremo de la herramienta del robot. El algoritmo de estimación

de juntas es iterativo, y emplea parámetros de visión actualizados, con cada movimiento que realiza el robot para alcanzar su objetivo. Al finalizar una maniobra de posicionamiento se esperaría que la información en espacio de cámara tanto del objetivo, como la artificial lograra un empate adecuado y con poco error; si dicho empate de marcas es adecuado, y aun así se aprecia un nivel considerable de error de posicionamiento del robot en una maniobra, se puede inferir que el sistema de adquisición de imágenes influye con cierto grado de error con causales que pueden deberse a aberraciones ópticas.

A continuación se muestra el error promedio entre la proyección de las marcas objetivo en espacio de cámara y las marcas artificiales sobre la herramienta del autómatas, al finalizar una maniobra de posicionamiento.

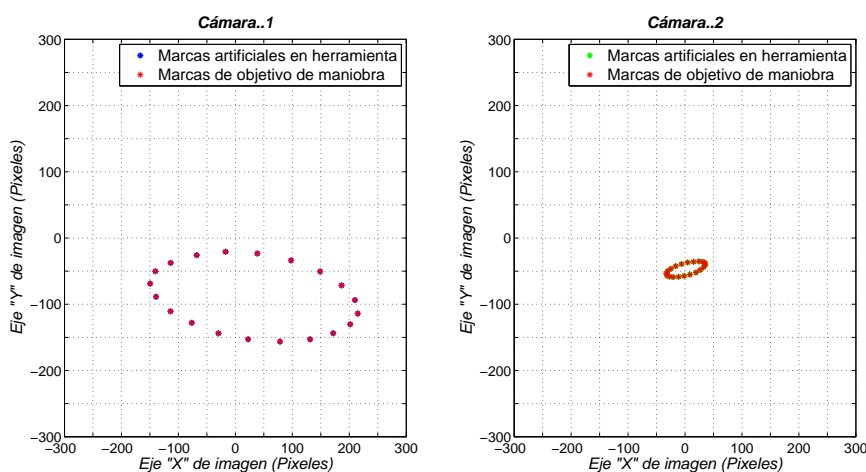


Figura 4.32: Comparación entre proyecciones en espacio de cámara, de las marcas visuales objetivo y las marcas artificiales sobre la herramienta del manipulado; a la izquierda cámara 1; a la derecha cámara 2.

Del gráfico anterior, los resultados experimentales son los siguientes:

Error de empate de marcas visuales objetivo y artificiales en espacio de cámara, al finalizar maniobra de posicionamiento			
Error de empate en cámara 1 (px)	Des. std.	Error de empate en cámara 2 (px)	Des. std.
0.07	0.04	0.07	0.005

Tabla 4.15: Errores de empate en espacio de cámara de marcas visuales objetivo y marcas visuales artificiales.

Como se puede observar en los resultados de la Tabla 4.14, los errores de empate en espacio de cámara son inferiores a una décima de pixel, por lo que se considera que el modelo de cámara recursivo funciona adecuadamente y que parte del error de posición en las maniobras de posicionamiento, es debido a aberraciones ópticas en el sistema de visión.

3. Tratando con el supuesto número tres, no se observa una disminución del error de orientación, con el incremento de la densidad de resolución o la disminución del valor de resolución específica; dado que en los resultados obtenidos respecto a la orientación del efector final del manipulado se observa cierta similitud entre el grupo el primer y el segundo grupo de secuencias experimentales. No obstante, parece ser que la orientación del plano de pruebas entre cada secuencia experimental, afecta en gran medida el valor del error angular, ya que al observar los valores de $\bar{\alpha}$ y $\bar{\beta}$ entre cada secuencia experimental, aquellos experimentos de ambos grupos de pruebas, en donde se mantuvieron valores similares de φ_1 y φ_2 , guardan una cierta similitud de acuerdo a la proporción existente entre las magnitudes de $\bar{\alpha}$ y $\bar{\beta}$. Por ende se deduce que se puede relacionar dicha tendencia, con el sesgo existente entre la orientación del eje focal de cada cámara virtual y la orientación del plano de pruebas.

4.4. Prueba de detección de diferencias de relieve en una superficie

El sistema de cámaras virtuales desarrollado como método alternativo de adquisición de imágenes para un sistema de control por visión *CSM* de robots, de acuerdo a los resultados experimentales de la sección anterior, ha demostrado ser funcional, pudiendo ser viable su uso dependiendo de las características deseadas para una aplicación dada.

En la presente sección se demuestra el procedimiento seguido en una prueba, cuyo objetivo es encontrar en que medida, el sistema de cámaras virtuales desarrollado aunado a un sistema de control por visión *CSM*, es capaz de cuantificar diferencias de elevación de puntos sobre una superficie, debidas a los relieves de la misma; donde los resultados obtenidos pueden dar pie a futuras aplicaciones.

La tarea de medir las diferencias de elevación de puntos sobre una superficie, parte

del barrido de este mediante un haz de luz láser, de un cuerpo de geometría escalonada (Figura 4.33), el cual esta constituido de varios planos paralelos, cuya elevación es ascendente respecto de un nivel de referencia.

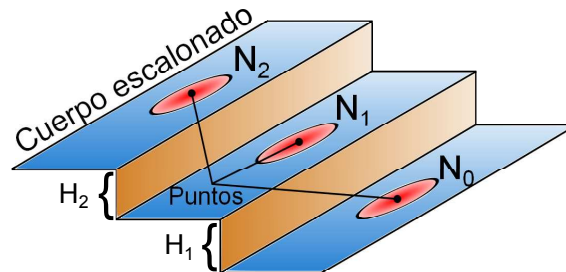


Figura 4.33: Representación de cuerpo de geometría escalonada

Dicho cuerpo escalonado es una placa plana de aluminio, sobre la cual se han adherido capas de cinta adherente, una encima de otra. Con cada nueva capa de cinta, se construye algo similar a una escalera, en donde la diferencia de elevación entre escalones es equivalente al espesor de la cinta (0.15 - 0.40 *mm* aproximadamente). El objeto escalonado construido, es caracterizado mediante el algoritmo de estimación de la posición tridimensional de puntos en espacio de cámara ($N_c = 2$), proyectándose una marca luminosa por medio de un láser en el sentido ascendente de la geometría escalonada; es decir, un punto láser por escalón, los cuales poseen un ancho de 10 *mm* cada uno.

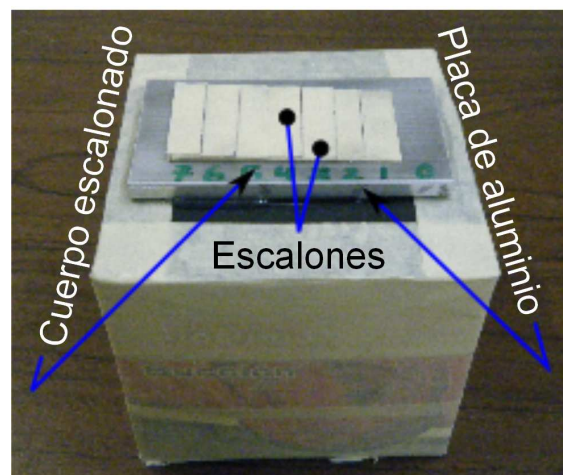


Figura 4.34: Cuerpo escalonado construido para determinar la sensibilidad del modelo de perspectiva recursivo.

Los parámetros de cámara empleados en el algoritmo de estimación tridimensional,

son parámetros cercanos a la región sobre la cual será colocado el objeto a ser caracterizado; los parámetros mencionados se obtienen de una maniobra de posicionamiento del robot, efectuada cerca de la zona donde se colocará el cuerpo escalonado. Una vez que se cuenta con parámetros de cámara locales válidos, se efectúa la caracterización del cuerpo escalonado, mediante la proyección de varios puntos láser, sucesivamente en la dirección del aumento de elevación del cuerpo escalonado. Cada punto luminiscente captado en espacio de cámara es almacenado y procesado para obtener su mapeo respectivo de coordenadas en espacio tridimensional, referidas a la base del robot. Finalmente, los puntos tridimensionales calculados, son analizados en busca de concordancia entre su elevación estimada desde un nivel de referencia, con la elevación real que se espera; donde el nivel de referencia de medición es el nivel inferior al ras de la placa de aluminio.

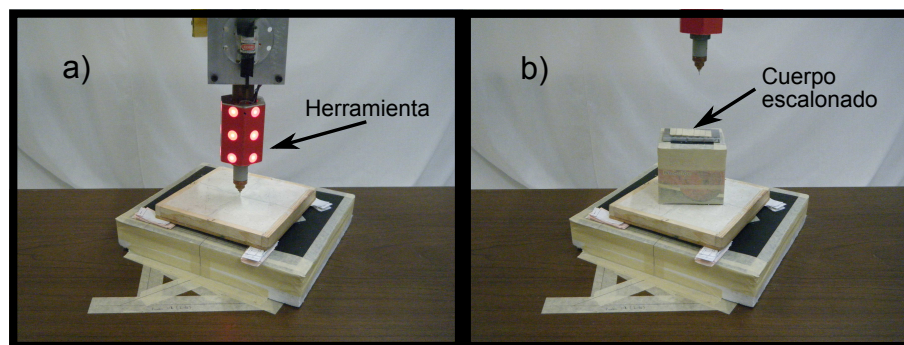


Figura 4.35: Maniobra de posicionamiento para actualización de parámetros de cámara a fin de asegurar validez de barrido en cuerpo escalonado; a) Maniobra de posicionamiento; b) Ubicación de cuerpo escalonado.

El número total de barridos realizados al cuerpo escalonado es de 30, para emitir un correcto análisis estadístico de los datos. En suma, hay un número de siete escalones con elevaciones diferentes, más el nivel al ras a la placa de aluminio. Ésta ha de ser nivelada con respecto a la horizontal, antes de comenzar el experimento. La resolución específica utilizada para esta prueba es de 6610 y 6650 mm/px para la cámara uno y la cámara dos respectivamente.

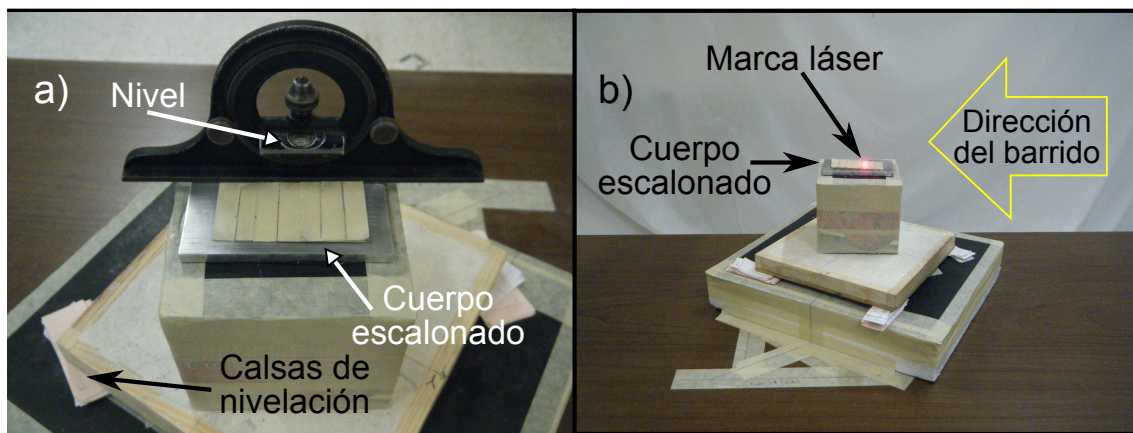


Figura 4.36: a) Nivelación de cuerpo escalonado; b) Barrido con haz láser de cuerpo escalonado y sentido del barrido.

La prueba se realiza de forma automática y se aprovecha al manipulador como dispositivo de orientación del láser que incidirá sobre cada nivel del cuerpo escalonado. Los resultados experimentales obtenidos de las pruebas de barrido se muestran a continuación.

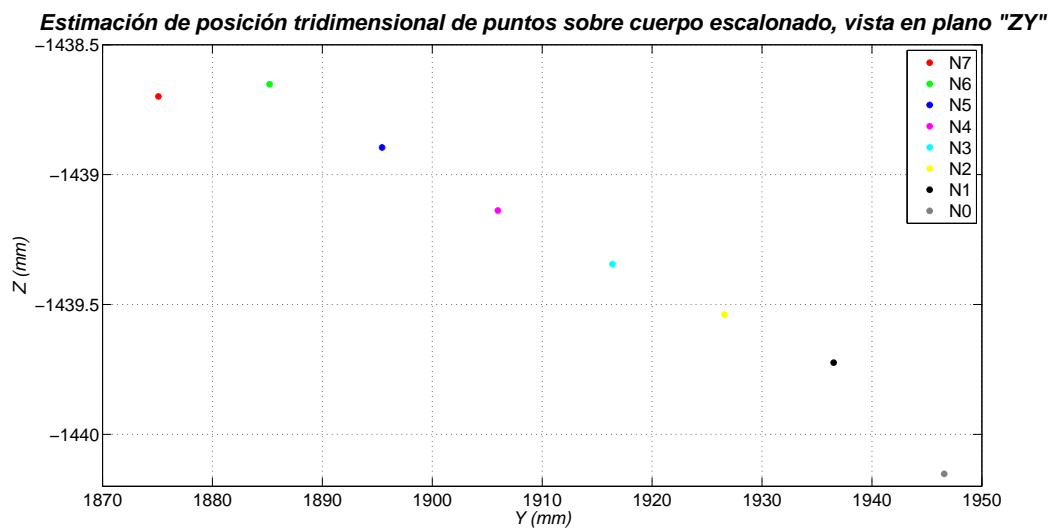


Figura 4.37: Coordenadas sobre plano "ZY", de posición estimada tridimensionalmente de puntos sobre cuerpo escalonado; las coordenadas estimadas están dadas con respecto al origen de la base del robot.

Del gráfico anterior, los resultados experimentales son lo siguientes:

Análisis del error de elevación promedio, sobre eje "Z", de estimación tridimensional de marcas sobre cuerpo escalonado							
Marca	N1	N2	N3	N4	N5	N6	N7
Elevación de referencia (mm)*	0.4	0.55	0.9	1.15	1.4	1.7	2.05
Elevación estimada (mm)*	0.42	0.61	0.80	1.01	1.25	1.50	1.45
Porcentaje de error	2.8 %	6.3 %	9.2 %	13.6 %	14.3 %	19.9 %	59.6 %

Tabla 4.16: Análisis del error de elevación promedio de estimación tridimensional de marcas sobre cuerpo escalonado. La *elevación de las marcas es medida con base al nivel inferior del cuerpo escalonado.

4.4.1. Discusión

De los resultados experimentales generados para la determinación de la sensibilidad del modelo de perspectiva recursivo se tienen los siguientes razonamientos:

1. Se puede concluir que el sistema de control por visión basado en cámaras virtuales desarrollado, es capaz de detectar cambios de elevación de una marca visual sobre una superficie con relieves, del orden de 0.40 mm, obteniéndose un error del 2.8 %.
2. Se observa que conforme aumenta el nivel de elevación de la marca proyectada sobre el cuerpo escalonado, el valor de la estimación tridimensional diverge ascendentemente con respecto de las mediciones de elevación de referencia hechas al cuerpo escalonado.
3. La marca "N7" arroja un comportamiento no ascendente en elevación, lo cual se presenta posiblemente debido al alejamiento de la zona de validez de los parámetros de visión empleados.

Conclusiones y Trabajo Futuro

El motivo del presente capítulo es hacer mención de las conclusiones derivadas de la presente tesis y en particular de las obtenidas con base en las discusiones planteadas en el capítulo 4, realizadas para cada una de las validaciones experimentales. Por otro lado, también se hace énfasis en lo que puede ser considerado como trabajo futuro, para la línea de investigación trazada por la metodología de control por visión de robots de *Manipulación de Espacio de Cámara*.

Conclusiones

Con base en la validación experimental realizada mediante maniobras de posicionamiento, tomando en cuenta el valor de la *resolución específica* por captura de cada cámara virtual, se emiten las siguientes conclusiones:

- Se controló exitosamente a un robot industrial, mediante la metodología de control por visión de *CSM*, basando la adquisición de imágenes en un sistema de cámaras virtuales.
- La precisión con la cual se efectúa una maniobra de posicionamiento de un robot manipulador está fuertemente relacionada con la resolución específica (mm/px), a la cual los sensores opto-electrónicos, en este caso cámaras u observadores virtuales, trabajan. A menor resolución específica (mm/px), o a mayor densidad de resolución (px/mm), mayor será la precisión con la cual se efectúa una maniobra de posicionamiento dada.
- Un sistema óptico, basado en elementos reflexivos planos para la generación de cámaras u observadores virtuales, es útil como medio de adquisición de imágenes para un sistema de control por visión de robots *CSM*, no obstante

el desempeño del sistema, puede estar comprometido en cuanto a la precisión con la cual un robot puede aproximarse a un objetivo, debido a la disminución de la densidad de resolución en los observadores virtuales y a las aberraciones ópticas presentes en los elementos reflexivos. Si dichos elementos son de espesor delgado y dimensiones considerables, la deflexión en ellos puede provocar un fenómeno conocido como distorsión óptica, lo que llevará al sistema óptico de adquisición a obtener capturas poco fieles al entorno. Como medida preventiva a la distorsión óptica en espejos planos, se puede optar por el uso de espejos cuyo espesor impida su deflexión debida a sus dimensiones. En cuanto a la disminución de la densidad de resolución en los observadores virtuales, puesto que no se conoce la relación exacta de esta en la disminución de la precisión con la que un robot es controlado, simplemente se recomienda el uso de sensores de alta resolución.

- El error de posición medido en los experimentos de maniobras de posicionamiento, no es suficientemente grande como para descartar la posibilidad del uso de un sistema de cámaras virtuales, en un sistema robótico de control. Aplicaciones para el sistema podrían ser, por ejemplo, las de carga y descarga de material o productos por un brazo robot industrial en un entorno móvil, ofreciéndose un aceptable desempeño en cuanto a precisión.
- Los experimentos realizados para la detección de diferencias de relieve sobre una superficie, muestran resultados de error de elevación bastante aceptables en la estimación de la posición tridimensional de marcas láser sobre un cuerpo escalonado. Se observa además que, conforme se acrecenta la elevación con respecto del nivel inferior, el error de estimación aumenta, pudiendo ser debido a su acumulación, o bien, a las características de la región de validez de los parámetros de cámara empleados.

Trabajo Futuro

Dentro de toda línea seguida en un trabajo investigación, es importante señalar los resultados obtenidos, a fin de encontrar las posibles coyunturas que den pie a nuevos descubrimientos e inventos, o bien que sirvan para reajustar el rumbo de la investigación, mediante el perfeccionamiento de la técnica. Del presente trabajo de investigación se plantearon ciertos puntos, útiles para el desarrollo de nuevos

proyectos y el crecimiento en aportaciones a la metodología de control por visión de *CSM*.

- Partiendo de la mecánica de diseño de un sistema cámaras virtuales, presentada en el capítulo 2, se puede generar un sistema catadióptrico más compacto, cuyos espejos tuviesen un espesor mayor, de esta manera la distorsión óptica se minimizaría.
- Un sistema de cámaras virtuales cuyos espejos fuesen móviles podría ser una alternativa al uso de sistemas de adquisición de imágenes con cámaras móviles, por lo que la propuesta de dicho sistema catadióptrico sería interesante.
- En el presente proyecto de investigación se exploró la integración de la biblioteca de procesamiento de imágenes de *OpenCV*, con una interfaz gráfica creada en lenguaje Visual C++ que emplea las librerías de *MFC*; aprovechando esto, se podría invertir en el desarrollo de un sistema de procesamiento de imágenes más complejo y robusto, para brindarle mayor agilidad a un robot industrial controlado por *CSM*.
- Aprovechando los resultados de las pruebas de detección de diferencias de relieve en una superficie, se podría desarrollar un sistema poco invasivo basado en *CSM*, para el barrido y reconstrucción tridimensional de superficies blandas.

Aportaciones

Con base a los objetivos del proyecto enunciados en el capítulo 1, se presentan las siguientes aportaciones:

- Se desarrolló un sistema de control de manipuladores industriales, basado en la metodología de control por visión de *CSM* cuyo sistema de adquisición de imágenes, se basó en la generación de observadores virtuales o cámaras virtuales, a partir de una sola cámara y múltiples espejos planos. La funcionalidad de dicho sistema de adquisición fue validada experimentalmente a través del modelo de perspectiva recursivo.

-
- Se creó una interfaz gráfica para la ejecución de maniobras simples de control por visión de un manipulador industrial, la cual se integró a la biblioteca de procesamiento de imágenes de OpenCV.
 - Se desarrolló un sistema para el diseño de la geometría del sistema de adquisición de cámaras virtuales, el cual aunque esta dado para el caso simple de solo dos observadores virtuales, éste contiene las bases para el desarrollo de sistemas ópticos más complejos, con mayor cantidad de observadores virtuales.
 - Se validó una técnica no invasiva, basada en la metodología de control por visión *CSM*, para la medición de espesores o diferencias de relieves en superficies; la cual fue producto de la búsqueda de un modo para determinar la sensibilidad del modelo de perspectiva recursivo.
 - Se desarrolló una técnica para la medición de la resolución específica, con la cual una cámara de video realiza capturas a una distancia dada medida sobre su eje óptico.

Bibliografía

- [1] J. Gluckman y S. Nayar, “Rectified catadioptric stereo sensors,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, pp. 224–236, feb 2002.
- [2] T. Pachidis y J. Lygouras, “Pseudostereo-vision system: A monocular stereo-vision system as a sensor for real-time robot applications,” *IEEE Transactions on Instrumentation and Measurement*, vol. 56, pp. 2547–2560, December 2007.
- [3] S. Skaar, I. Yalda-Mooshabad, y W. Brockman, “Nonholonomic camera-space manipulation,” *Robotics and Automation, IEEE Transactions on*, vol. 8, pp. 464–479, aug 1992.
- [4] B. Zhang, J. Wang, G. Rossano, C. Martinez, y S. Kock, “Vision-guided robot alignment for scalable, flexible assembly automation,” in *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*, pp. 944–951, dec. 2011.
- [5] M. Seelinger, J. Yoder, E. Baumgartner, y S. Skaar, “High-precision visual control of mobile manipulators,” *Robotics and Automation, IEEE Transactions on*, vol. 18, pp. 957–965, dec 2002.
- [6] E. Gonzalez-Galvan, A. Loreda-Flores, L. Raygoza, J. Palos, y S. Skaar, “Precise and robust large-shape reproduction using uncalibrated vision,” in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pp. 2799–2803, may 2009.
- [7] G. Magnani y P. Rocco, “Impedance control for elastic joints industrial manipulators,” *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 488–498, 2004.

-
- [8] C. Hun, K. Park, K. S. Hoon, K. Igo, y N. Hogan, “Stochastic estimation of human arm impedance using robots with nonlinear frictions: An experimental validation,” *IEEE/ASME Transactions on Mechatronics*, vol. PP, no. 99, pp. 1–12, 2011.
- [9] K. Hashimoto, “A review on vision-based control of robot manipulators,” *VSP and Robotics Society of Japan*, vol. 17, no. 10, pp. 969–991, 2003.
- [10] A. Gangloff y M. Manthelin, “High-speed visual servoing of a 6-d.o.f manipulator using multivariable predictive control,” *VSP and Robotics Society of Japan*, vol. 17, no. 10, pp. 993–1021, 2003.
- [11] M. Seelinger, S. Skaar, y M. Robinson, “An alternative approach for image-plane control of robots,” in *The confluence of vision and control* (D. Kriegman, G. Hager, y A. Morse, eds.), vol. 237 of *Lecture Notes in Control and Information Sciences*, pp. 41–65, Springer London, 1998.
- [12] S. Skaar, W. Brockman, y R. Hanson, “Camera space manipulation,” *International Journal of Robotics Research*, vol. 6, no. 4, pp. 20–32, 1987.
- [13] E. Gonzalez-Galván, *Notas sobre Teoría de Estimación Aplicada a la Robótica*. Centro de Investigación y Estudios de Posgrado de la Facultad de Ingeniería-UASLP, 2010.
- [14] E. Hetch, *Optics*. Addison Wesley, 4 ed., 2002.
- [15] J. Casas, *Optica*. Facultad de Ciencias de la Universidad de Zaragoza (España), 5 ed., 1985.
- [16] J. Gluckman y S. Nayar, “Rectified catadioptric stereo sensors,” in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 380–387, IEEE, 2000.
- [17] M. Inaba, T. Hara, y H. Inoue, “A stereo viewer based on a single camera with view-control mechanisms,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, pp. 1857–1865, July 1993.
- [18] K.-H. Tan, H. Hua, y N. Ahuja, “Multiview panoramic cameras using a mirror pyramid,” in *Omnidirectional Vision, 2002. Proceedings. Third Workshop on*, pp. 87 – 93, 2002.

-
- [19] J. Gluckman y S. Nayar, “Planar catadioptric stereo: geometry and calibration,” in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 1, pp. 2 vol. (xxiii+637+663), 1999.
- [20] A. Goshtasby y W. A. Gruver, “Design of a single-lens stereo camera system.,” *Pattern Recognition*, vol. 26, no. 6, pp. 923–937, 1993.
- [21] E. Grosso y M. Tistarelli, “Active/dynamic stereo vision,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 17, pp. 868 –879, sep 1995.
- [22] B. Hu, “It’s all done with mirrors: Calibration-and-correspondence-free 3d reconstruction.,” in *CRV*, pp. 148–154, IEEE Computer Society, 2009.
- [23] D. Lanman, D. E. Crispell, y G. Taubin, “Surround structured lighting for full object scanning.,” in *3DIM*, pp. 107–116, IEEE Computer Society, 2007.
- [24] S. A. Nene y S. K. Nayar, “Stereo with mirrors,” in *Proceedings of the Sixth International Conference on Computer Vision, ICCV ’98*, (Washington, DC, USA), pp. 1087–, IEEE Computer Society, 1998.
- [25] C. Kulpate, R. Paranjape, y M. Mehrandezh, “Precise 3d positioning of a robotic arm using a single camera and a flat mirror,” *International Journal of Optomechatronics*, vol. 2, no. 3, pp. 205–232, 2008.
- [26] E. González-Galván, C. Chávez, I. Bonilla, M. Mendoza, L. Raygoza, A. Loredoflores, y B. Zhang, “Precise industrial robot positioning and path-tracking over large surfaces using non-calibrated vision,” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 5160 –5166, may 2011.
- [27] L. Raygoza-Pérez, E. González-Galván, L.-F. A., J. Pastor, y E. Baumgartner, “An enabling vision-based approach for non-calibrated, robot-positioning task,” *International Review of Automatic Control*, vol. 3, no. 6, p. 710, 2010.
- [28] E. Baumgartner y P. Schenker, “Autonomous image-plane robot control for martian lander operations,” in *Robotics and Automation, 1996. Proceedings, 1996 IEEE International Conference on*, vol. 1, pp. 726 –731 vol.1, apr 1996.
- [29] M. Seelinger, E. González-Galván, M. Robinson, y S. Skaar, “Towards a robotic plasma spraying operation using vision,” *Robotics Automation Magazine, IEEE*, vol. 5, pp. 33 – 38,49, dec 1998.

-
- [30] *Spray painting of a general three-dimensional surface*, vol. 3, 2000.
- [31] B. Zhang, J. Wang, G. Rossano, y C. Martinez, “Vision-guided robotic assembly using uncalibrated vision,” in *Mechatronics and Automation (ICMA), 2011 International Conference on*, pp. 1384 –1389, aug. 2011.
- [32] E. González-Galván y S. Skaar, “Efficient camera-space manipulation using moments,” in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, vol. 4, pp. 3407 –3412 vol.4, apr 1996.
- [33] E. J. González-Galván, S. R. Cruz-Ramírez, M. J. Seelinger, y J. J. Cervantes-Sánchez, “An efficient multi-camera, multi-target scheme for the three-dimensional control of robots using uncalibrated vision,” *Robotics and Computer Integrated Manufacturing*, vol. 19, pp. 387–400, 2003.
- [34] S. Cruz-Ramírez, “Desarrollo de un algoritmo de control basado en visión para la ejecución de tareas robotizadas sobre superficies arbitrarias curvas,” Master’s thesis, Facultad de Ingeniería-Ciep, UASLP, 2002.
- [35] E. González-Galván, A. Loredó-Flores, F. Pazos-Flores, y J. Cervantes-Sanchez, “An optimal path-tracking algorithm for unstructured environments based on uncalibrated vision,” in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 2547 – 2552, april 2005.
- [36] E. González-Galván, F. Pazos-Flores, S. Skaar, y A. Cárdenas-Galindo, “Camera pan/tilt to eliminate the workspace-size/pixel-resolution tradeoff with camera-space manipulation,” *Robotics and Computer Integrated Manufacturing*, vol. 18, pp. 95–104, 2003.
- [37] M. Seelinger y J. Yoder, “Automatic pallet engagement by a vision guided forklift,” in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 4068 – 4073, april 2005.
- [38] Z. Zhang, “Flexible camera calibration by viewing a plane from unknown orientations,” in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 1, pp. 666 –673 vol.1, 1999.
- [39] J. Heikkila y O. Silven, “A four-step camera calibration procedure with implicit image correction,” in *Computer Vision and Pattern Recognition, 1997.*

- Proceedings., 1997 IEEE Computer Society Conference on*, pp. 1106 –1112, jun 1997.
- [40] J. Viramontes-Reyna y G.-G. E.J., “Visión para posicionar un manipulador utilizando un sistema de visión mono-ocular,” *Memorias de 6to Congreso Nacional de Mecatrónica, Instituto Tecnológico de San Luis Potosí, Asociación Mexicana de Mecatrónica A.C.*, vol. 6, Noviembre 2007.
- [41] J. Rendón-Mancha, A. Cárdenas, M. García, E. González-Galván, y B. Lara, “Robot positioning using camera-space manipulation with a linear camera model,” *Robotics, IEEE Transactions on*, vol. 26, pp. 726 –733, aug. 2010.
- [42] W. Chen, U. Korde, y S. Skaar, “Position control experiments using vision,” *The International Journal of Robotics Research*, vol. 13, no. 3, pp. 199–208, 1994.
- [43] G. Bradski y A. Kaehler, *Learning OpenCV*. O’Reilly Media, Inc., 1 ed., 2008.
- [44] R. Laganière, *OpenCV 2 Computer Vision, Application Programming Cookbook*. Packt publishing Ltd., 2011.
- [45] W. Wathen-Dunn y A. F. C. R. L. U. D. S. Laboratory, *Models for the perception of speech and visual form: proceedings of a symposium*. M.I.T. Press, 1967.
- [46] J. Canny, “A computational approach to edge detection,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-8, pp. 679 –698, nov. 1986.
- [47] E. J. González-Galván, A. Loredó-Flores, J. Jesús Cervantes-Sánchez, L. Antonio Aguilera-Cortés, y S. B. Skaar, “An optimal path-generation algorithm for manufacturing of arbitrarily curved surfaces using uncalibrated vision,” *Robot. Comput.-Integr. Manuf.*, vol. 24, pp. 77–91, Feb. 2008.

Apéndice A

Rutinas de Simulación de Sistema de Cámaras Virtuales

A.1. Rutina Principal

En esta rutina se definen las variables de la simulación y se imprimen los elementos de la simulación en un gráfico tridimensional, donde dichos elementos son construidos a partir de otras subrutinas. Las variables de entrada son:

- Posición del objetivo respecto del sistema origen de robot $\{X_0, Y_0, Z_0\}$ (PObj).
- Posición de la cámara respecto del sistema origen de robot $\{X_0, Y_0, Z_0\}$ (PCam).
- Ángulo de rotación de la cámara respecto de su eje propio “Y”, paralelo al eje Y_0 de la base del robot (RCamY).
- Distancias **D00**, **D01**, **D02**, medidas paralelamente a los ejes X_0 y Y_0 del sistema de la base del robot, cuyo valor indica la posición de los espejos centrales y laterales con respecto de la cámara real (D00, D01, D02).
- Apertura del campo visual de la cámara real (AP).
- Paso ángulo y longitudinal para la impresión del espacio de trabajo del robot (paso1, paso2).
- Ángulos de orientación del plano de pruebas φ_1, φ_2 (phi1, phi2).

```

1
2 %% Simulación Cámara Virtual.-- Segunda Versión.
3 % Corrección de acuerdo a primera versión.
4 %% Convenciones:
5 %1.-El origen de todo el gráfico es el origen del robot Fanuc ↔
      M16iB/20T
6 %2.-
7
8 %% Definición de Variables:
9 clear;clc;
10 PObj = [636;3660/2;-1500.17]; %Localización del objetivo de ↔
      maniobra. (XYZ)[636;3660;-1521.17]
11 PCam = [3988;3660/2;-629.53]; %Localización de cámara real. ↔
      (XYZ)[3988;1913;-629.53]
12 RCamY = -8.0;%Rotación sobre el eje Y del eje focal de la ↔
      Cámara Real
13 D00 = 60*25.4; %Distancia desde cámara real a espejos centrales↔
      60*25.4
14 D01 = 71.25*25.4; %Distancia desde espejos centrales hasta ↔
      espejo izquierdo 74.25*25.4
15 D02 = 71.25*25.4; %Distancia desde espejos centrales hasta ↔
      espejo derecho      71*25.4
16 AP = [12;8];      %Apertura visual de cámara [XY,XZ] ↔
      [20;10]
17 paso1 = 120;      %Separación longitudinal de puntos del campo de ↔
      trabajo del robot
18 paso2 = 10;      %Separación angular de puntos del campo de ↔
      trabajo del robot
19 phi1 = 30;      %Ángulo phi1 de plano objetivo. ↔
      30
20 phi2 = -17.5;    %Ángulo phi2 de plano objetivo. ↔
      -17.5
21
22 %% Funciones de generación de objetos
23
24 [CAM,E00,E01,E02,OBJ,FOCAL] = CV_ESPEJOS(PObj,PCam,RCamY,D00,↔
      D01,D02);
25

```

```

26 [VC01, VCI01, VCD01, VCI02, VCD02] = CV_CAMPO(E00, E01, E02, FOCAL, AP)↔
    ;
27
28 [VV, PP]=CV_GRAFICA_ESP(paso1, paso2, 0, 0, 0);
29
30 [AngF, AngF01ZY, AngF01XY, AngF02ZY, AngF02XY, PLANO] = ↔
    ANALISIS_FOCAL(FOCAL, phi1, phi2)
31 %% Impresiones
32 f1 = figure(1)
33 %Impresión de cámara 1
34 plot3(CAM(1,:), CAM(2,:), CAM(3,:), '-', 'LineWidth', 2, 'Color'↔
    , [1, 0.5, 0]) %Cámara
35 hold on
36 %Impresión de objetivo
37 plot3(OBJ(1,:), OBJ(2,:), OBJ(3,:), '-', 'LineWidth', 2, 'Color'↔
    , [0, 0.5, 1]) %Objetivo
38 hold on
39 %Impresión de espejo central
40 plot3(E00(1,:), E00(2,:), E00(3,:), '-', 'LineWidth', 2, 'Color'↔
    , [0, 0, 1]) %espejo central
41 hold on
42 %Impresión de espejo lateral izquierdo
43 plot3(E01(1,:), E01(2,:), E01(3,:), '-', 'LineWidth', 2, 'Color'↔
    , [1, 0, 0]) %espejo lateral izquierdo
44 hold on
45 %Impresión de espejo lateral derecho
46 plot3(E02(1,:), E02(2,:), E02(3,:), '-', 'LineWidth', 2, 'Color'↔
    , [0, 1, 0]) %espejo lateral derecho
47 hold on
48 %Impresión de eje focal
49 plot3(FOCAL(1,:), FOCAL(2,:), FOCAL(3,:), '-', 'LineWidth', 2, 'Color'↔
    , [1, 0, 1]) %Eje focal
50 hold on
51 %Impresión de espacio de trabajo del robot
52 plot3(VV(:, 1), VV(:, 2), VV(:, 3), '-', 'LineWidth', 1, 'Color'↔
    , [0.5, 0.5, 1])
53 hold on
54 %Impresión de proyección de espacio de trabajo del robot

```

```

55 %plot3(PP(:,1),PP(:,2),PP(:,3),'-','LineWidth',1,'Color'←
    ',[0,0,0.5])
56 %hold on
57 %Impresión de campo visual entre cámara y espejo central
58 plot3(VCO1(1,:),VCO1(2,:),VCO1(3,),'-.','LineWidth',2,'Color'←
    ',[0,0,0])
59 hold on
60 %Impresión de campo visual entre espejo central y espejo ←
    lateral izquierdo
61 plot3(VCI01(1,:),VCI01(2,:),VCI01(3,),'-.','LineWidth',2,'←
    Color',[0,0,0])
62 hold on
63 %Impresión de campo visual entre espejo central y espejo ←
    lateral derecho
64 plot3(VCD01(1,:),VCD01(2,:),VCD01(3,),'-.','LineWidth',2,'←
    Color',[0,0,0])
65 hold on
66 %Impresión de campo visual entre espejo lateral izquierdo y ←
    objetivo
67 plot3(VCI02(1,:),VCI02(2,:),VCI02(3,),'-.','LineWidth',2,'←
    Color',[0,0,0])
68 hold on
69 %Impresión de campo visual entre espejo lateral derecho y ←
    objetivo
70 plot3(VCD02(1,:),VCD02(2,:),VCD02(3,),'-.','LineWidth',2,'←
    Color',[0,0,0])
71 hold on
72 %Impresión de plano objetivo
73 %plot3(PLANO(1,:),PLANO(2,:),PLANO(3,),'-','LineWidth',1,'←
    Color',[1,0,0.5]) %Objetivo
74 %hold on
75
76 titulo = strcat('Simulación de Sistema de Cámaras Virtuales de ←
    Cuatro Espejos')
77 title(titulo,'FontSize',20,'FontName','Arial','FontWeight','←
    bold','FontAngle','italic');
78 leg = legend('Cámara Real','Objetivo','Espejos Centrales','←
    Espejo Lateral Izquierdo','Espejo Lateral Derecho','Eje ←

```

```
        Focal', 'Espacio de Trabajo del Robot', 'Campo Visual')
79 set(leg, 'interpreter', 'none', 'FontSize', 12)
80 grid on
81 axis equal
82 xlabel('X (mm)', 'FontSize', 18, 'FontName', 'Arial', 'FontWeight', '↔
    light', 'FontAngle', 'italic')
83 ylabel('Y (mm)', 'FontSize', 18, 'FontName', 'Arial', 'FontWeight', '↔
    light', 'FontAngle', 'italic')
84 zlabel('Z (mm)', 'FontSize', 18, 'FontName', 'Arial', 'FontWeight', '↔
    light', 'FontAngle', 'italic')
```

A.2. Subrutina para Creación de Espejos

Esta subrutina, se encarga de la construcción de varios elementos del sistema de cámaras virtuales; entre ellos, el sistema de espejos ya orientados, la cámara real, el objetivo visual del sistema y los ejes ópticos del sistema.

```

1  %%FUNCION DE IMPRESIÓN DE CÁMARA, ESPEJOS Y OBJETIVO.
2  %%POR MARDK 10/12/2012
3  function [CAM, E00, E01, E02, OBJ, FOCAL] = CV_ESPEJOS (PObj, PCam, ←
      RCamY, D00, D01, D02)
4  %% Creación de un cúbo
5  Dc = 25;
6  %   [ 1   2   3   4   1   5   6   2   6   7   3   7   8   4   ←
      8   5
7  CX = [-Dc -Dc +Dc +Dc -Dc -Dc -Dc -Dc -Dc +Dc +Dc +Dc +Dc +Dc ←
      Dc -Dc];
8  CY = [-Dc +Dc +Dc -Dc -Dc -Dc +Dc +Dc +Dc +Dc +Dc +Dc -Dc -Dc ←
      Dc -Dc];
9  CZ = [+Dc +Dc +Dc +Dc +Dc -Dc -Dc +Dc -Dc -Dc +Dc -Dc -Dc +Dc ←
      Dc -Dc];
10 UU = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1];
11 %% Creación de cámara
12 CUBO = [CX;CY;CZ;UU];
13 TCAM = [eye(4,3) [PCam;1]];
14 RCAMy = [cosd(RCamY) 0 sind(RCamY) 0;0 1 0 0;-sind(RCamY) 0 ←
      cosd(RCamY) 0;0 0 0 1];
15 CAM = TCAM*RCAMy*CUBO; %%—————Posición y orientación de←
      cámara real
16 %% Creación de objetivo
17 TOBJ = [eye(4,3) [PObj;1]];
18 OBJ = TOBJ*CUBO;          % Posición de objetivo
19 %% Creación de espejo central
20 Xi = []; Yi = []; Zi = []; U0 = []; XI = []; YI = []; ZI = []; UI = []; XD = []; YD = []; ZD ←
      = []; UD = [];
21 TE00 = [eye(4,3) [-D00;0;D00*tand(RCamY);1]];
22 PE00 = TCAM*TE00; %%—————Posición de ←
      espejo central

```



```

23 L1 = 300;
24 L2 = 300;
25
26 for i=0:0.1:1
27 E1 = PE00*[1 0 0 0;0 1 0 0;0 0 1 -L2/2*i;0 0 0 1];
28 E2 = PE00*[1 0 0 0;0 1 0 0;0 0 1 +L2/2*i;0 0 0 1];
29 E3 = PE00*[1 0 0 -L1*cosd(45);0 1 0 -L1*sind(45);0 0 1 -L2/2*i↔
    ;0 0 0 1];
30 E4 = PE00*[1 0 0 -L1*cosd(45);0 1 0 -L1*sind(45);0 0 1 +L2/2*i↔
    ;0 0 0 1];
31 E5 = PE00*[1 0 0 -L1*cosd(45);0 1 0 +L1*sind(45);0 0 1 -L2/2*i↔
    ;0 0 0 1];
32 E6 = PE00*[1 0 0 -L1*cosd(45);0 1 0 +L1*sind(45);0 0 1 +L2/2*i↔
    ;0 0 0 1];
33
34 Xi = [E3(1,4) E1(1,4) E5(1,4) E6(1,4) E2(1,4) E4(1,4) E3(1,4) ↔
    Xi];
35 Yi = [E3(2,4) E1(2,4) E5(2,4) E6(2,4) E2(2,4) E4(2,4) E3(2,4) ↔
    Yi];
36 Zi = [E3(3,4) E1(3,4) E5(3,4) E6(3,4) E2(3,4) E4(3,4) E3(3,4) ↔
    Zi];
37 U0 = [1 1 1 1 1 1 1 U0];
38 end
39
40 E00 = [Xi;Yi;Zi;U0];
41
42 %% Creación de espejo izquierdo
43 YVec = [0;1;0]; %Normal a espejos izquierdo en posición inicial
44
45 TE01 = [eye(4,3) [0;-D01;D01*tand(RCamY);1]];
46 PE01 = PE00*TE01; %—————Posición de espejo ↔
    lateral izquierdo
47
48 VLC1 = [PE00(1,4)-PE01(1,4);PE00(2,4)-PE01(2,4);PE00(3,4)-PE01↔
    (3,4)];
49 VL01 = [TOBJ(1,4)-PE01(1,4);TOBJ(2,4)-PE01(2,4);TOBJ(3,4)-PE01↔
    (3,4)];
50

```

```

51 VLC1u = VLC1./norm(VLC1); %Vector unitario de espejo izquierdo ↔
    a espejo central
52 VL01u = VL01./norm(VL01); %Vector unitario de espejo izquierdo ↔
    a objetivo
53
54 VM01 = (VLC1u+VL01u)./2
55 VM01u = VM01./norm(VM01); %Vector unitario medio entre VLC1u y ↔
    VLO1u
56
57 Ang1 = acosd(dot(YVec,VM01u));
58 VRarb1 = cross(YVec,VM01u)./norm(cross(YVec,VM01u));
59
60 e1 = -cosd(Ang1/2);
61 e2 = VRarb1(1)*sind(Ang1/2);
62 e3 = VRarb1(2)*sind(Ang1/2);
63 e4 = VRarb1(3)*sind(Ang1/2);
64
65 RArb1 = [e1^2+e2^2-e3^2-e4^2 2*(e2*e3+e1*e4) 2*(e2*e4-e1*e3) 0;
66          2*(e2*e3-e1*e4) e1^2-e2^2+e3^2-e4^2 2*(e1*e2+e3*e4) 0;
67          2*(e1*e3+e2*e4) 2*(e3*e4-e1*e2) e1^2-e2^2-e3^2+e4^2 0;
68          0 0 0 1];
69
70 for i=0:0.1:1
71 I1 = PE01*RArb1*[1 0 0 -950;0 1 0 0;0 0 1 500*i;0 0 0 1];
72 I2 = PE01*RArb1*[1 0 0 +50;0 1 0 0;0 0 1 500*i;0 0 0 1];
73 I3 = PE01*RArb1*[1 0 0 +50;0 1 0 0;0 0 1 -500*i;0 0 0 1];
74 I4 = PE01*RArb1*[1 0 0 -950;0 1 0 0;0 0 1 -500*i;0 0 0 1];
75
76 XI = [I1(1,4) I2(1,4) I3(1,4) I4(1,4) I1(1,4) XI];
77 YI = [I1(2,4) I2(2,4) I3(2,4) I4(2,4) I1(2,4) YI];
78 ZI = [I1(3,4) I2(3,4) I3(3,4) I4(3,4) I1(3,4) ZI];
79 UI = [1 1 1 1 1 UI];
80 end
81
82
83 E01 = [XI;YI;ZI;UI];
84 %% Creación de espejos derecho
85 YVec = [0;-1;0];

```

```

86
87 TE02 = [eye(4,3) [0;+D02;D02*tand(RCamY);1]];
88 PE02 = PE00*TE02;
89
90 VLC2 = [PE00(1,4)-PE02(1,4);PE00(2,4)-PE02(2,4);PE00(3,4)-PE02↔
(3,4)];
91 VL02 = [TOBJ(1,4)-PE02(1,4);TOBJ(2,4)-PE02(2,4);TOBJ(3,4)-PE02↔
(3,4)];
92
93 VLC2u = VLC2./norm(VLC2); %Vector unitario de espejo derecho a ↔
espejo central
94 VL02u = VL02./norm(VL02); %Vector unitario de espejo derecho a ↔
objetivo
95
96 VM02 = (VLC2u+VL02u)./2;
97 VM02u = VM02./norm(VM02);
98 Ang2 = acosd(dot(YVec,VM02u))
99 VRarb2 = cross(YVec,VM02u)./norm(cross(YVec,VM02u));
100
101 e1 = -cosd(Ang2/2);
102 e2 = VRarb2(1)*sind(Ang2/2);
103 e3 = VRarb2(2)*sind(Ang2/2);
104 e4 = VRarb2(3)*sind(Ang2/2);
105
106 RArb2 = [e1^2+e2^2-e3^2-e4^2 2*(e2*e3+e1*e4) 2*(e2*e4-e1*e3) 0;
107          2*(e2*e3-e1*e4) e1^2-e2^2+e3^2-e4^2 2*(e1*e2+e3*e4) 0;
108          2*(e1*e3+e2*e4) 2*(e3*e4-e1*e2) e1^2-e2^2-e3^2+e4^2 0;
109          0 0 0 1];
110
111
112 for i=0:0.1:1
113 D1 = PE02*RArb2*[1 0 0 -950;0 1 0 0;0 0 1 500*i;0 0 0 1];
114 D2 = PE02*RArb2*[1 0 0 +50;0 1 0 0;0 0 1 500*i;0 0 0 1];
115 D3 = PE02*RArb2*[1 0 0 +50;0 1 0 0;0 0 1 -500*i;0 0 0 1];
116 D4 = PE02*RArb2*[1 0 0 -950;0 1 0 0;0 0 1 -500*i;0 0 0 1];
117
118
119 XD = [D1(1,4) D2(1,4) D3(1,4) D4(1,4) D1(1,4) XD];

```

A.2. SUBROUTINA PARA CREACIÓN DE ESPEJOS

```
120 YD = [D1(2,4) D2(2,4) D3(2,4) D4(2,4) D1(2,4) YD];
121 ZD = [D1(3,4) D2(3,4) D3(3,4) D4(3,4) D1(3,4) ZD];
122 UD = [1 1 1 1 1 UD];
123 end
124
125
126 E02 = [XD;YD;ZD;UD];
127
128
129 %% Creación de ejes focales
130 FX = [TCAM(1,4) PE00(1,4) PE01(1,4) TOBJ(1,4) PE02(1,4) PE00↔
        (1,4) TCAM(1,4)];
131 FY = [TCAM(2,4) PE00(2,4) PE01(2,4) TOBJ(2,4) PE02(2,4) PE00↔
        (2,4) TCAM(2,4)];
132 FZ = [TCAM(3,4) PE00(3,4) PE01(3,4) TOBJ(3,4) PE02(3,4) PE00↔
        (3,4) TCAM(3,4)];
133 UF = [1 1 1 1 1 1 1];
134
135 FOCAL = [FX;FY;FZ;UF];
136 end
```

A.3. Subrutina para Creación de Campo Visual

Esta subrutina se encarga de la construcción de el campo visual del sistema de catadióptrico, aplicando la ley de refracción a cada vector incidente en algún espejo.

```

1 %% Algoritmo para determinar el campo visual de la cámara ←
   virtual.
2 % Por MARDK 10/12/2012
3 function [VC01, VCI01, VCD01, VCI02, VCD02] = CV_CAMPO (E00, E01, E02, ←
   FOCAL, AP)
4 %% Generación de primer campo visual.
5 AP = AP./2;
6 % Orden de E00 E3(1,4) E1(1,4) E5(1,4) E6(1,4) E2(1,4) E4(1,4) ←
   E3(1,4)
7
8 %% Generación de puntos de plano de espejo 01 (Izquierdo)
9 PAIZ = [E00(1,2); E00(2,2); E00(3,2)]; % Punto 1
10 PBIZ = [E00(1,1); E00(2,1); E00(3,1)]; % Punto 3
11 PCIZ = [E00(1,6); E00(2,6); E00(3,6)]; % Punto 4
12
13 %% Generación de puntos de plano de espejo 02 (Derecho)
14 PADR = [E00(1,5); E00(2,5); E00(3,5)]; % Punto 2
15 PBDR = [E00(1,4); E00(2,4); E00(3,4)]; % Punro 6
16 PCDR = [E00(1,3); E00(2,3); E00(3,3)]; % Punto 5
17
18 %% Generación de puntos de plano de espejo 01 (Lateral Izquierdo)
19 PALIZ = [E01(1,1); E01(2,1); E01(3,1)]; % Punto 1
20 PBLIZ = [E01(1,2); E01(2,2); E01(3,2)]; % Punto 3
21 PCLIZ = [E01(1,3); E01(2,3); E01(3,3)]; % Punto 3
22
23 %% Generación de puntos de plano de espejo 02 (Lateral Derecho)
24 PALDR = [E02(1,1); E02(2,1); E02(3,1)]; % Punto 1
25 PBLDR = [E02(1,2); E02(2,2); E02(3,2)]; % Punro 2
26 PCLDR = [E02(1,3); E02(2,3); E02(3,3)]; % Punto 3
27
28 %% Generación de puntos de vector1
29 Val = 100;

```

```

30 Vc01 = rotaY(-AP(2))*rotaZ(0)*[FOCAL(1,2)-FOCAL(1,1);FOCAL(2,2)↔
    -FOCAL(2,1);FOCAL(3,2)-FOCAL(3,1);1];
31 Vc01A = [FOCAL(1,1);FOCAL(2,1);FOCAL(3,1)];
32 Vc01B = [FOCAL(1,1)+Vc01(1);FOCAL(2,1)+Vc01(2);FOCAL(3,1)+Vc01↔
    (3)];
33
34 [intx,inty,intz,REB001] = CV_INTERSECCION02(PAIZ,PBIZ,PCIZ,↔
    Vc01A,Vc01B);
35 Vc01C = [intx;inty;intz];
36
37 %Generación de puntos de vector2
38
39 Vc02 = rotaY(+AP(2))*rotaZ(0)*[FOCAL(1,2)-FOCAL(1,1);FOCAL(2,2)↔
    -FOCAL(2,1);FOCAL(3,2)-FOCAL(3,1);1];
40 Vc02A = [FOCAL(1,1);FOCAL(2,1);FOCAL(3,1)];
41 Vc02B = [FOCAL(1,1)+Vc02(1);FOCAL(2,1)+Vc02(2);FOCAL(3,1)+Vc02↔
    (3)];
42
43 [intx,inty,intz,REB002] = CV_INTERSECCION02(PADR,PBDR,PCDR,↔
    Vc02A,Vc02B);
44 Vc02C = [intx;inty;intz];
45
46 %Generación de puntos de vector3
47
48 Vc03 = rotaY(-AP(2))*rotaZ(AP(1))*[FOCAL(1,2)-FOCAL(1,1);FOCAL↔
    (2,2)-FOCAL(2,1);FOCAL(3,2)-FOCAL(3,1);1];
49 Vc03A = [FOCAL(1,1);FOCAL(2,1);FOCAL(3,1)];
50 Vc03B = [FOCAL(1,1)+Vc03(1);FOCAL(2,1)+Vc03(2);FOCAL(3,1)+Vc03↔
    (3)];
51
52 [intx,inty,intz,REB003] = CV_INTERSECCION02(PAIZ,PBIZ,PCIZ,↔
    Vc03A,Vc03B);
53 Vc03C = [intx;inty;intz];
54 %Generación de puntos de vector4
55
56 Vc04 = rotaY(+AP(2))*rotaZ(AP(1))*[FOCAL(1,2)-FOCAL(1,1);FOCAL↔
    (2,2)-FOCAL(2,1);FOCAL(3,2)-FOCAL(3,1);1];
57 Vc04A = [FOCAL(1,1);FOCAL(2,1);FOCAL(3,1)];

```

```

58 Vc04B = [FOCAL(1,1)+Vc04(1);FOCAL(2,1)+Vc04(2);FOCAL(3,1)+Vc04(↵
    (3)];
59
60 [intx,inty,intz,REBO04] = CV_INTERSECCION02(PAIZ,PBIZ,PCIZ,↵
    Vc04A,Vc04B);
61 Vc04C = [intx;inty;intz];
62 %Generación de puntos de vector5
63
64 Vc05 = rotaY(-AP(2))*rotaZ(-AP(1))*[FOCAL(1,2)-FOCAL(1,1);FOCAL(↵
    (2,2)-FOCAL(2,1);FOCAL(3,2)-FOCAL(3,1);1];
65 Vc05A = [FOCAL(1,1);FOCAL(2,1);FOCAL(3,1)];
66 Vc05B = [FOCAL(1,1)+Vc05(1);FOCAL(2,1)+Vc05(2);FOCAL(3,1)+Vc05(↵
    (3)];
67
68 [intx,inty,intz,REBO05] = CV_INTERSECCION02(PADR,PBDR,PCDR,↵
    Vc05A,Vc05B);
69 Vc05C = [intx;inty;intz];
70 %Generación de puntos de vector6
71
72 Vc06 = rotaY(+AP(2))*rotaZ(-AP(1))*[FOCAL(1,2)-FOCAL(1,1);FOCAL(↵
    (2,2)-FOCAL(2,1);FOCAL(3,2)-FOCAL(3,1);1];
73 Vc06A = [FOCAL(1,1);FOCAL(2,1);FOCAL(3,1)];
74 Vc06B = [FOCAL(1,1)+Vc06(1);FOCAL(2,1)+Vc06(2);FOCAL(3,1)+Vc06(↵
    (3)];
75
76 [intx,inty,intz,REBO06] = CV_INTERSECCION02(PADR,PBDR,PCDR,↵
    Vc06A,Vc06B);
77 Vc06C = [intx;inty;intz];
78
79 VC01 = [Vc01A Vc01C Vc03C Vc03A Vc04C Vc03C Vc03A Vc02C Vc04C ↵
    Vc04A Vc06C Vc02C Vc02A Vc05C Vc06C Vc06A Vc01C Vc05C Vc05A↵
    ];
80
81 %Vectores extras, reflexión de vector 1 (segunda)
82 Vc07 = rotaY(-AP(2))*rotaZ(0)*[FOCAL(1,2)-FOCAL(1,1);FOCAL(2,2)↵
    -FOCAL(2,1);FOCAL(3,2)-FOCAL(3,1);1];
83 Vc07A = [FOCAL(1,1);FOCAL(2,1);FOCAL(3,1)];

```

```

84 Vc07B = [ FOCAL(1,1)+Vc07(1); FOCAL(2,1)+Vc07(2); FOCAL(3)+Vc07(3)↔
      ];
85
86 [intx,inty,intz,REB007] = CV_INTERSECCION02(PADR,PBDR,PCDR,↔
      Vc07A,Vc07B);
87 Vc07C = [intx;inty;intz];
88
89 %Vectores extras, reflexión de vector 2 (segunda)
90 Vc08 = rotaY(+AP(2))*rotaZ(0)*[FOCAL(1,2)-FOCAL(1,1); FOCAL(2,2)↔
      -FOCAL(2,1); FOCAL(3,2)-FOCAL(3,1); 1];
91 Vc08A = [FOCAL(1,1); FOCAL(2,1); FOCAL(3,1)];
92 Vc08B = [FOCAL(1,1)+Vc08(1); FOCAL(2,1)+Vc08(2); FOCAL(3,1)+Vc08↔
      (3)];
93
94 [intx,inty,intz,REB008] = CV_INTERSECCION02(PAIZ,PBIZ,PCIZ,↔
      Vc08A,Vc08B);
95 Vc08C = [intx;inty;intz];
96
97 %%Generación de campo visual entre espejo central y espejo ↔
      izquierdo.
98 %A partir de vector 1:
99 L = 100;
100
101 Vc01Cb = [Vc01C(1)+REB001(1)*L; Vc01C(2)+REB001(2)*L; Vc01C(3)+↔
      REB001(3)*L];
102 [intx,inty,intz,REB001L] = CV_INTERSECCION02(PALIZ,PBLIZ,PCLIZ,↔
      Vc01C,Vc01Cb);
103 Vc01Cs = [intx;inty;intz];
104 Vc01Ct = [Vc01Cs(1)+REB001L(1)*L; Vc01Cs(2)+REB001L(2)*L; Vc01Cs↔
      (3)+REB001L(3)*L];
105 Vc01IZ = [Vc01C(1) Vc01Cs(1); Vc01C(2) Vc01Cs(2); Vc01C(3) Vc01Cs↔
      (3)];
106
107 Vc08Cb = [Vc08C(1)+REB008(1)*L; Vc08C(2)+REB008(2)*L; Vc08C(3)+↔
      REB008(3)*L];
108 [intx,inty,intz,REB008L] = CV_INTERSECCION02(PALIZ,PBLIZ,PCLIZ,↔
      Vc08C,Vc08Cb);
109 Vc08Cs = [intx;inty;intz];

```



```

110 Vc08Ct = [Vc08Cs(1)+REB008L(1)*L;Vc08Cs(2)+REB008L(2)*L;Vc08Cs↔
      (3)+REB008L(3)*L];
111 Vc08IZ = [Vc08C(1) Vc08Cs(1);Vc08C(2) Vc08Cs(2);Vc08C(3) Vc08Cs↔
      (3)];
112
113 Vc04Cb = [Vc04C(1)+REB004(1)*L;Vc04C(2)+REB004(2)*L;Vc04C(3)+↔
      REB004(3)*L];
114 [intx,inty,intz,REB004L] = CV_INTERSECCION02(PALIZ,PBLIZ,PCLIZ,↔
      Vc04C,Vc04Cb);
115 Vc04Cs = [intx;inty;intz];
116 Vc04Ct = [Vc04Cs(1)+REB004L(1)*L;Vc04Cs(2)+REB004L(2)*L;Vc04Cs↔
      (3)+REB004L(3)*L];
117 Vc04IZ = [Vc04C(1) Vc04Cs(1);Vc04C(2) Vc04Cs(2);Vc04C(3) Vc04Cs↔
      (3)];
118
119 Vc03Cb = [Vc03C(1)+REB003(1)*L;Vc03C(2)+REB003(2)*L;Vc03C(3)+↔
      REB003(3)*L];
120 [intx,inty,intz,REB003L] = CV_INTERSECCION02(PALIZ,PBLIZ,PCLIZ,↔
      Vc03C,Vc03Cb);
121 Vc03Cs = [intx;inty;intz];
122 Vc03Ct = [Vc03Cs(1)+REB003L(1)*L;Vc03Cs(2)+REB003L(2)*L;Vc03Cs↔
      (3)+REB003L(3)*L];
123 Vc03IZ = [Vc03C(1) Vc03Cs(1);Vc03C(2) Vc03Cs(2);Vc03C(3) Vc03Cs↔
      (3)];
124
125 VCI01 = [Vc01IZ Vc08Cs Vc08C Vc01C Vc03IZ Vc01Cs Vc08Cs Vc04Cs ↔
      Vc04C Vc03IZ Vc04Cs];
126
127 %%Generación de campo visual entre espejo central y espejo ↔
      derecho.
128
129 Vc07Cb = [Vc07C(1)+REB007(1)*L;Vc07C(2)+REB007(2)*L;Vc07C(3)+↔
      REB007(3)*L];
130 [intx,inty,intz,REB007L] = CV_INTERSECCION02(PALDR,PBLDR,PCLDR,↔
      Vc07C,Vc07Cb);
131 Vc07Cs = [intx;inty;intz];
132 Vc07Ct = [Vc07Cs(1)+REB007L(1)*L;Vc07Cs(2)+REB007L(2)*L;Vc07Cs↔
      (3)+REB007L(3)*L];

```

A.3. SUBROUTINA PARA CREACIÓN DE CAMPO VISUAL

```
133 Vc07DR = [Vc07C(1) Vc07Cs(1);Vc07C(2) Vc07Cs(2);Vc07C(3) Vc07Cs↔
      (3)];
134
135 Vc02Cb = [Vc02C(1)+REB002(1)*L;Vc02C(2)+REB002(2)*L;Vc02C(3)+↔
      REB002(3)*L];
136 [intx,inty,intz,REB002L] = CV_INTERSECCION02(PALDR,PBLDR,PCLDR,↔
      Vc02C,Vc02Cb);
137 Vc02Cs = [intx;inty;intz];
138 Vc02Ct = [Vc02Cs(1)+REB002L(1)*L;Vc02Cs(2)+REB002L(2)*L;Vc02Cs↔
      (3)+REB002L(3)*L];
139 Vc02DR = [Vc02C(1) Vc02Cs(1);Vc02C(2) Vc02Cs(2);Vc02C(3) Vc02Cs↔
      (3)];
140
141 Vc05Cb = [Vc05C(1)+REB005(1)*L;Vc05C(2)+REB005(2)*L;Vc05C(3)+↔
      REB005(3)*L];
142 [intx,inty,intz,REB005L] = CV_INTERSECCION02(PALDR,PBLDR,PCLDR,↔
      Vc05C,Vc05Cb);
143 Vc05Cs = [intx;inty;intz];
144 Vc05Ct = [Vc05Cs(1)+REB005L(1)*L;Vc05Cs(2)+REB005L(2)*L;Vc05Cs↔
      (3)+REB005L(3)*L];
145 Vc05DR = [Vc05C(1) Vc05Cs(1);Vc05C(2) Vc05Cs(2);Vc05C(3) Vc05Cs↔
      (3)];
146
147 Vc06Cb = [Vc06C(1)+REB006(1)*L;Vc06C(2)+REB006(2)*L;Vc06C(3)+↔
      REB006(3)*L];
148 [intx,inty,intz,REB006L] = CV_INTERSECCION02(PALDR,PBLDR,PCLDR,↔
      Vc06C,Vc06Cb);
149 Vc06Cs = [intx;inty;intz];
150 Vc06Ct = [Vc06Cs(1)+REB006L(1)*L;Vc06Cs(2)+REB006L(2)*L;Vc06Cs↔
      (3)+REB006L(3)*L];
151 Vc06DR = [Vc06C(1) Vc06Cs(1);Vc06C(2) Vc06Cs(2);Vc06C(3) Vc06Cs↔
      (3)];
152
153 VCD01 = [Vc07DR Vc05Cs Vc05C Vc07C Vc02DR Vc07Cs Vc02Cs Vc06Cs ↔
      Vc06C Vc05DR Vc06Cs];
154 %VCD01 = [Vc07DR Vc05Cs Vc05C Vc07C Vc02C]% Vc07Cs]
155 %%Generación de planos borde de campo visual.
156
```

```

157 PAIZOP = [Vc03Cs(1);Vc03Cs(2);Vc03Cs(3)]; %
158 PBIZOP = [Vc04Cs(1);Vc04Cs(2);Vc04Cs(3)]; %
159 PCIZOP = [Vc03Ct(1);Vc03Ct(2);Vc03Ct(3)]; %
160
161 PADROP = [Vc05Cs(1);Vc05Cs(2);Vc05Cs(3)]; %
162 PBDROP = [Vc06Cs(1);Vc06Cs(2);Vc06Cs(3)]; %
163 PCDROP = [Vc05Ct(1);Vc05Ct(2);Vc05Ct(3)]; %
164
165 %%Generación de campo visual de espejo lateral izquierdo
166
167 %Vector 1 salida del lateral
168 [intx,inty,intz]= CV_INTERSECCION(PADROP,PBDROP,PCDROP,Vc01Cs,↔
    Vc01Ct);
169 Vc01Cv = [intx;inty;intz];
170 %Vector 8 salida del lateral
171 [intx,inty,intz] = CV_INTERSECCION(PADROP,PBDROP,PCDROP,Vc08Cs,↔
    Vc08Ct);
172 Vc08Cv = [intx;inty;intz];
173 %Vector 4 salida del lateral
174 [intx,inty,intz] = CV_INTERSECCION(PADROP,PBDROP,PCDROP,Vc04Cs,↔
    Vc04Ct);
175 Vc04Cv = [intx;inty;intz];
176 %Vector 3 salida del lateral
177 [intx,inty,intz] = CV_INTERSECCION(PADROP,PBDROP,PCDROP,Vc03Cs,↔
    Vc03Ct);
178 Vc03Cv = [intx;inty;intz];
179
180 VCI02 = [Vc01Cs Vc08Cs Vc04Cs Vc03Cs Vc01Cs Vc01Cv Vc08Cv ↔
    Vc08Cs Vc08Cv Vc04Cv Vc04Cs Vc04Cv Vc03Cv Vc03Cs Vc03Cv ↔
    Vc01Cv]
181 %%Generación de campo visual de espejo lateral Derecho
182
183 %Vector 2 salida del lateral
184 [intx,inty,intz]= CV_INTERSECCION(PAIZOP,PBIZOP,PCIZOP,Vc02Cs,↔
    Vc02Ct);
185 Vc02Cv = [intx;inty;intz];
186 %Vector 7 salida del lateral

```

```
187 [intx,inty,intz] = CV_INTERSECCION(PAIZOP,PBIZOP,PCIZOP,Vc07Cs,↔
    Vc07Ct);
188 Vc07Cv = [intx;inty;intz];
189 %Vector 5 salida del lateral
190 [intx,inty,intz] = CV_INTERSECCION(PAIZOP,PBIZOP,PCIZOP,Vc05Cs,↔
    Vc05Ct);
191 Vc05Cv = [intx;inty;intz];
192 %Vector 6 salida del lateral
193 [intx,inty,intz] = CV_INTERSECCION(PAIZOP,PBIZOP,PCIZOP,Vc06Cs,↔
    Vc06Ct);
194 Vc06Cv = [intx;inty;intz];
195
196 VCD02 = [Vc07Cs Vc05Cs Vc06Cs Vc02Cs Vc07Cs Vc07Cv Vc05Cv ↔
    Vc05Cs Vc05Cv Vc06Cv Vc06Cs Vc06Cv Vc02Cv Vc02Cs Vc02Cv ↔
    Vc07Cv];
197 %VCD02 = [Vc07Cs Vc05Cs Vc06Cs Vc02Cs Vc07Cs Vc07Cv];
198
199 end
```

A.4. Subrutina para el Análisis del Eje Focal del Sistema

Esta subrutina tiene la finalidad de calcular el ángulo entre los ejes focales provenientes de los espejos laterales del sistema, además de calcular su ángulo de incidencia con respecto a planos ZY y XY , paralelos al origen de coordenadas del robot ($\{X_0, Y_0, Z_0\}$). En esta subrutina se construye además la representación del plano objetivo del sistema, debidamente orientado.

```

1  %Función que analisis relación entre ejes focales de cámara ↔
    virtual 01 y
2  %02. Fecha de creación 12/01/13. Por Ing Miguel A. Rdz y Dmz ↔
    Kessler
3  function [AngF, AngF01ZY, AngF01XY, AngF02ZY, AngF02XY, PLANO] = ↔
    ANALISIS_FOCAL(FOCAL, Phi1, Phi2)
4  %Copia de vectores finales:
5  VCV01 = [FOCAL(1,4) - FOCAL(1,3); FOCAL(2,4) - FOCAL(2,3); FOCAL↔
    (3,4) - FOCAL(3,3)];
6  VCV02 = [FOCAL(1,4) - FOCAL(1,5); FOCAL(2,4) - FOCAL(2,5); FOCAL↔
    (3,4) - FOCAL(3,5)];
7  %Calculo de angulo entre ejes focales de cámaras virtuales
8  VCV01u = VCV01./norm(VCV01);
9  VCV02u = VCV02./norm(VCV02);
10 AngF = acosd(dot(VCV01u, VCV02u));
11 %Calculo de angulos entre ejes focales y planos "ZY" y "XY".
12 VCV01ZY = [0; VCV01u(2); VCV01u(3)]./norm([0; VCV01u(2); VCV01u(3)↔
    ]);
13 VCV01XY = [VCV01u(1); VCV01u(2); 0]./norm([VCV01u(1); VCV01u(2)↔
    ;0]);
14 AngF01ZY = acosd(dot(VCV01u, VCV01ZY));
15 AngF01XY = acosd(dot(VCV01u, VCV01XY));
16 VCV02ZY = [0; VCV02u(2); VCV02u(3)]./norm([0; VCV02u(2); VCV02u(3)↔
    ]);
17 VCV02XY = [VCV02u(1); VCV02u(2); 0]./norm([VCV02u(1); VCV02u(2)↔
    ;0]);
18 AngF02ZY = acosd(dot(VCV02u, VCV02ZY));
19 AngF02XY = acosd(dot(VCV02u, VCV02XY));

```

```
20 %Creación de plano de pruebas.
21 FLine = [-105;-105;0;1];
22 OLine = FLine;
23 X = 0; Y = 0; Z = 0;
24
25 for(i=2:1:22)
26 if(mod(i,2) == 0)
27 FLine = [eye(4,3) [0;210;0;1]]*FLine;
28 OLine = [OLine FLine];
29 FLine = [eye(4,3) [10;0;0;1]]*FLine;
30 OLine = [OLine FLine];
31 else
32 FLine = [eye(4,3) [0;-210;0;1]]*FLine;
33 OLine = [OLine FLine];
34 FLine = [eye(4,3) [10;0;0;1]]*FLine;
35 OLine = [OLine FLine];
36 end
37 % if(i == 2 || i==22)
38 %X = 0; Y = 0; Z = 0;
39 % FLine = [eye(4,3) [X;Y;Z;1]]*FLine;
40 % else
41 %X = 10; Y = 0; Z = 0;
42 % FLine = [eye(4,3) [X;Y;Z;1]]*FLine;
43 % end
44 end
45 trans = [eye(4,3) [FOCAL(1,4);FOCAL(2,4);FOCAL(3,4);1]];
46 PLANO = trans*rotaz(Phi1)*rotay(-Phi2)*OLine;
47 end
```

A.5. Otras subrutinas

Cinemática Directa del Manipulador (loc_efector):

```

1 function [X1 , Y1 , Z1 , X2 , Y2 , Z2 , X3 , Y3 , Z3 , X4 , Y4 , Z4 , X5 , Y5 , Z5 , X6 , Y6 , Z6 ←
    , X7 , Y7 , Z7 , X8 , Y8 , Z8]=loc_efector (J , Pox , Poy , Poz)
2 %Comprobación de Método CDMAE.. Cinemática directa de robot ←
    Fancu Mate
3 %M-16iB
4
5 %ANGULOS DE LAS JUNTAS
6 teta2 = J(2)*pi/180;
7 teta3 = J(3)*pi/180;
8 teta4 = J(4)*pi/180;
9 teta5 = J(5)*pi/180;
10 teta6 = J(6)*pi/180;
11
12 %DESPLAZAMIENTO DE JUNTA PRISMATICA
13 D1 = J(1) ;
14
15 A2 = 770;
16 A3 = 100;
17 A4 = 740;
18 A5 = 100;
19
20 %MATRICES DE ROTACIÓN
21
22 Po =[1 0 0 Pox;0 1 0 Poy;0 0 1 Poz;0 0 0 1];
23 A = [1 0 0 0;0 1 0 D1;0 0 1 0;0 0 0 1];
24 B = [1 0 0 A2*cos(-teta2);0 1 0 0;0 0 1 A2*sin(-teta2);0 0 0 ←
    1];
25 C = [cos(teta3) 0 -sin(teta3) A3*cos(teta3);0 1 0 0;sin(teta3) ←
    0 cos(teta3) A3*sin(teta3);0 0 0 1];
26 D = [cos(teta4) -sin(teta4) 0 0;sin(teta4) cos(teta4) 0 0;0 0 1 ←
    -A4;0 0 0 1];
27 E = [cos(teta5) 0 -sin(teta5) sin(teta5)*A5;0 1 0 0;sin(teta5) ←
    0 cos(teta5) cos(teta5)*-A5;0 0 0 1];

```

```

28 F = [cos(teta6) -sin(teta6) 0 0;sin(teta6) cos(teta6) 0 0;0 0 1↵
      0;0 0 0 1];
29 G = [1 0 0 0;0 cos(pi) -sin(pi) 0;0 sin(pi) cos(pi) 0;0 0 0 1];
30
31 %SOLUCIÓN A CINEMATICA DIRECTA DEL MANIPULADOR HASTA SU EXTREMO↵
      SIN
32 %HERRAMIENTA
33
34 %SIS0 = A*B*C*D*E*F;
35
36 %VECTOR DE CALIBRACIÓN DE HERRAMIENTA
37 %SIS8 = [-261.6 1.4 379.6 1]';
38 SISA = [1 0 0 -284.6;0 1 0 0;0 0 1 0;0 0 0 1];
39 SISB = [1 0 0 0;0 1 0 1.3;0 0 1 0;0 0 0 1];
40 SISC = [1 0 0 0;0 1 0 0;0 0 1 366.2;0 0 0 1];
41 SISD = [0 0 0 1]';
42
43 %SOLUCIÓN A CINEMATICA DIRECTA DEL MANIPULADOR HASTA EL EXTREMO↵
      DE SU
44 %HERRAMIENTA
45
46
47 SIS1 = Po*A;
48 SIS2 = SIS1*B;
49 SIS3 = SIS2*C;
50 SIS4 = SIS3*D;
51 SIS5 = SIS4*E;
52 %SIS0 = SIS5*F*G*SIS8;
53 SIS6 = SIS5*F*G*SISA;
54 SIS7 = SIS6*SISB;
55 SIS8 = SIS7*SISC;
56
57
58 X1=SIS1(1,4);X2=SIS2(1,4);X3=SIS3(1,4);X4=SIS4(1,4);X5=SIS5↵
      (1,4);X6 = SIS6(1,4);X7 = SIS7(1,4);X8 = SIS8(1,4);%Xf=SIS0↵
      (1);
59 Y1=SIS1(2,4);Y2=SIS2(2,4);Y3=SIS3(2,4);Y4=SIS4(2,4);Y5=SIS5↵
      (2,4);Y6 = SIS6(2,4);Y7 = SIS7(2,4);Y8 = SIS8(2,4);%Xf=SIS0↵

```



```

        (2);
60 Z1=SIS1(3,4);Z2=SIS2(3,4);Z3=SIS3(3,4);Z4=SIS4(3,4);Z5=SIS5↔
        (3,4);Z6 = SIS6(3,4);Z7 = SIS7(3,4);Z8 = SIS8(3,4);%zf=SIS0↔
        (3);
61 end

```

Intersección entre un vector y un plano (CV_INTERSECCION):

```

1 %%.....FUNCION LOCALIZA PUNTO DE INTERSECCIÓN ENTRE UN PLANO Y ↔
  UNA RECTA...
2 %.....CREADA POR: ING. MIGUEL ANGEL RODRIGUEZ Y DOMINGUEZ ↔
  KESSLER.....
3 %.....FECHA: ↔
  07.03.2012.....
4
5 function [intx,inty,intz] = CV_INTERSECCION(PA,PB,PC,PR1,PR2)
6
7 %INTERCAMBIO DE VARIABLES.
8 a1 = PA(1);a2 = PA(2);a3 = PA(3);b1 = PB(1);b2 = PB(2);b3 = PB↔
  (3);c1 = PC(1);c2 = PC(2);c3 = PC(3);
9 gx = PR1(1); gy = PR1(2); gz = PR1(3); fx = PR2(1); fy = PR2(2)↔
  ; fz = PR2(3);
10
11 ux = (gx-fx);
12 uy = (gy-fy);
13 uz = (gz-fz);
14
15 A = +det([a2 a3 1;b2 b3 1;c2 c3 1]);
16 B = -det([a1 a3 1;b1 b3 1;c1 c3 1]);
17 C = +det([a1 a2 1;b1 b2 1;c1 c2 1]);
18 D = -det([a1 a2 a3;b1 b2 b3;c1 c2 c3]);
19
20 %A = +(a2*(b3-c3)-a3*(b2-c2)+(b2*c3-c2*b3));
21 %B = -(a1*(b3-c3)-a3*(b1-c1)+(b1*c3-c1*b3));
22 %C = +(a1*(b2-c2)-a2*(b1-c1)+(b1*c2-c1*b2));
23 %D = -(a1*(b2*c3-c2*b3)-a2*(b1*c3-c1*b3)+a3*(b1*c2-c1*b2));
24
25 L = -(A*fx+B*fy+C*fz+D)/(A*ux+B*uy+C*uz);

```

```

26
27 intx = ux*L+fx;
28 inty = uy*L+fy;
29 intz = uz*L+fz;
30
31 %L = -(A*ux+B*uy+C*uz)/(A*fx+B*fy+C*fz+D);
32 %
33 %intx = ux/L+fx;
34 %inty = uy/L+fy;
35 %intz = uz/L+fz;
36
37 end

```

Intersección entre un vector y un plano, y vector resultante de aplicar la ley de reflexión (CV_INTERSECCION02):

```

1 %%.....FUNCION LOCALIZA PUNTO DE INTERSECCIÓN ENTRE UN PLANO Y ←
  UNA RECTA...
2 %.....CREADA POR: ING. MIGUEL ANGEL RODRIGUEZ Y DOMINGUEZ ←
  KESSLER.....
3 %.....FECHA: ←
  07.03.2012.....
4
5 function [intx,inty,intz,REBO] = CV_INTERSECCION02(PA,PB,PC,PR1←
  ,PR2)
6
7 %INTERCAMBIO DE VARIABLES.
8 a1 = PA(1);a2 = PA(2);a3 = PA(3);b1 = PB(1);b2 = PB(2);b3 = PB←
  (3);c1 = PC(1);c2 = PC(2);c3 = PC(3);
9 gx = PR1(1); gy = PR1(2); gz = PR1(3); fx = PR2(1); fy = PR2(2)←
  ; fz = PR2(3);
10
11 ux = (gx-fx);
12 uy = (gy-fy);
13 uz = (gz-fz);
14
15 A = +det([a2 a3 1;b2 b3 1;c2 c3 1]);
16 B = -det([a1 a3 1;b1 b3 1;c1 c3 1]);

```

```

17 C = +det([a1 a2 1;b1 b2 1;c1 c2 1]);
18 D = -det([a1 a2 a3;b1 b2 b3;c1 c2 c3]);
19
20 %A = +(a2*(b3-c3)-a3*(b2-c2)+(b2*c3-c2*b3));
21 %B = -(a1*(b3-c3)-a3*(b1-c1)+(b1*c3-c1*b3));
22 %C = +(a1*(b2-c2)-a2*(b1-c1)+(b1*c2-c1*b2));
23 %D = -(a1*(b2*c3-c2*b3)-a2*(b1*c3-c1*b3)+a3*(b1*c2-c1*b2));
24
25 L = -(A*fx+B*fy+C*fz+D)/(A*ux+B*uy+C*uz);
26
27 intx = ux*L+fx;
28 inty = uy*L+fy;
29 intz = uz*L+fz;
30
31 %L = -(A*ux+B*uy+C*uz)/(A*fx+B*fy+C*fz+D);
32 %
33 %intx = ux/L+fx;
34 %inty = uy/L+fy;
35 %intz = uz/L+fz;
36
37 % Construcción de ecuación del plano con "d = 1"
38 Max = [a1 a2 a3;b1 b2 b3;c1 c2 c3];
39 dd = det(Max);
40 AA = -(det([1;1;1] Max(:,2) Max(:,3)]))/dd;
41 BB = -(det([Max(:,1) 1;1;1] Max(:,3)]))/dd;
42 CC = -(det([Max(:,1) Max(:,2) 1;1;1]))/dd;
43
44 %Vector normal al plano.
45 NP = [AA;BB;CC];
46 NPu = NP./norm(NP);
47 Ray = [ux;uy;uz]./norm([ux;uy;uz]);
48
49 Ang = acosd(dot(Ray, NPu));
50 if(Ang <= 90)
51     NPu = -NPu;
52
53 else
54     NPu = +NPu;

```

```

55
56 end
57
58 Ang = acosd(dot(Ray,NPu))
59 Arb = cross(Ray,NPu)./norm(cross(Ray,NPu));
60
61
62 e1 = -cosd(2*Ang/2);           %Doble de Ang
63 e2 = Arb(1)*sind(2*Ang/2);    %Doble de Ang
64 e3 = Arb(2)*sind(2*Ang/2);    %Doble de Ang
65 e4 = Arb(3)*sind(2*Ang/2);    %Doble de Ang
66
67 RotArb = [e1^2+e2^2-e3^2-e4^2  2*(e2*e3+e1*e4)  2*(e2*e4-e1*e3)  ←
            0;
            2*(e2*e3-e1*e4)  e1^2-e2^2+e3^2-e4^2  2*(e1*e2+e3*e4)  ←
            0;
            2*(e1*e3+e2*e4)  2*(e3*e4-e1*e2)  e1^2-e2^2-e3^2+e4^2  ←
            0;
            0 0 0 1];
70
71
72 REBO = RotArb * [-Ray;1];
73
74 end

```

Matriz de rotación sobre el eje X (rotaX):

```

1 %%.....FUNCION DE MATRIZ DE ROTACIÓN SOBRE EL EJE "X"←
  " .....
2 %.....CREADA POR: ING. MIGUEL ANGEL RODRIGUEZ Y DOMINGUEZ ←
  KESSLER.....
3 %.....FECHA: ←
  07.03.2012.....
4
5 function [Matrix] = rotaX(A)
6 Matrix = [1 0 0 0; 0 cosd(A) -sind(A) 0;0 sind(A) cosd(A) 0;0 0←
            0 1];
7 end

```

Matriz de rotación sobre el eje Y (rotaY):

```
1 %%.....FUNCION DE MATRIZ DE ROTACIÓN SOBRE EL EJE "Y↔
  " .....
2 %.....CREADA POR: ING. MIGUEL ANGEL RODRIGUEZ Y DOMINGUEZ ↔
  KESSLER.....
3 %.....FECHA: ↔
  07.03.2012.....
4
5 function [Matrix] = rotaY(A)
6 Matrix = [cosd(A) 0 sind(A) 0; 0 1 0 0;-sind(A) 0 cosd(A) 0;0 0↔
  0 1];
7 end
```

Matriz de rotación sobre el eje Z (rotaZ):

```
1 %%.....FUNCION DE MATRIZ DE ROTACIÓN SOBRE EL EJE "Z↔
  .....
2 %.....CREADA POR: ING. MIGUEL ANGEL RODRIGUEZ Y DOMINGUEZ ↔
  KESSLER.....
3 %.....FECHA: ↔
  07.03.2012.....
4
5 function [Matrix] = rotaZ(A)
6 Matrix = [cosd(A) -sind(A) 0 0; sind(A) cosd(A) 0 0;0 0 1 0;0 0↔
  0 1];
7 end
```

Apéndice B

Intersección entre una Recta y un Plano y Vector Reflejado Resultante

La determinación del punto de intersección entre una recta y un plano, además de la obtención del vector reflejado resultante, es uno de los cálculos frecuentemente realizados por la simulación del sistema óptico de cámaras virtuales presentada en la apéndice A, al generar la representación de los rayos que constituyen el campo de visual del sistema, cuando éstos chocan con los varios elementos reflexivos presentes. Dicho procedimiento de cálculo es descrito en la presente sección.

B.1. Cálculo del Punto de Intersección entre una Recta y un Plano

Considere la situación en la que las ecuaciones de un plano Π_1 y una recta \mathbf{R}_1 son conocidas. Si la ecuación del plano está dada en su forma cartesiana y la ecuación de la recta esta expresada de forma vectorial, entonces se tendrá lo siguiente:

$$\Pi_1 : A_1x + B_1y + C_1z = D_1 \quad (\text{B.1})$$

$$\mathbf{R}_1 : (x, y, z) = P_1 + t \vec{v}_1 \quad (\text{B.2})$$

Donde la recta \mathbf{R}_1 pasa por los puntos $P_1 = (p_x, p_y, p_z)$ y $Q_1 = (q_x, q_y, q_z)$, y $\vec{v}_1 = (v_{1x}, v_{1y}, v_{1z})$ es un vector paralelo a esta misma y resultante de $Q_1 - P_1 = \overrightarrow{P_1Q_1} = \vec{v}_1$. La forma de obtener la ecuación del plano Π_1 puede ser a través de tres puntos conocidos a, b, c del plano Π_1 , mediante la solución de los determinantes siguientes:

$$A_1 = \det \begin{pmatrix} a_y & a_z & 1 \\ b_y & b_z & 1 \\ c_y & c_z & 1 \end{pmatrix} \quad (\text{B.3})$$

$$B_1 = \det \begin{pmatrix} a_x & a_z & 1 \\ b_x & b_z & 1 \\ c_x & c_z & 1 \end{pmatrix} \quad (\text{B.4})$$

$$C_1 = \det \begin{pmatrix} a_x & a_y & 1 \\ b_x & b_y & 1 \\ c_x & c_y & 1 \end{pmatrix} \quad (\text{B.5})$$

$$D_1 = \det \begin{pmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \end{pmatrix} \quad (\text{B.6})$$

Si la ecuación de la recta \mathbf{R}_1 es convertida a sus formas paramétricas entonces se tiene:

$$\mathbf{R}_1 : \begin{cases} x(t) = p_x + tv_{1x} \\ y(t) = p_y + tv_{1y} \\ z(t) = p_z + tv_{1z} \end{cases} \quad (\text{B.7})$$

Al sustituir cada ecuación paramétrica de la recta \mathbf{R}_1 en el plano Π_1 , se puede resolver para “t”, entonces se tiene que:

$$t = -\frac{A_1 p_x + B_1 p_y + C_1 p_z + D_1}{A_1 v_{1x} + B_1 v_{1y} + C_1 v_{1z}} \quad (\text{B.8})$$

Al sustituir “t” proveniente de la ecuación B.8, en las ecuaciones paramétricas de la recta \mathbf{R}_1 (Ecs. B.7), se pueden encontrar las coordenadas $\{x(t), y(t), z(t)\}$ del punto de intersección de la recta \mathbf{R}_1 con el plano $\mathbf{\Pi}_1$; este punto será llamado \mathbf{T}_1 .

B.2. Cálculo del Vector Reflejado Resultante

En la sección anterior se mostró el procedimiento para calcular el punto de intersección \mathbf{T}_1 entre una recta \mathbf{R}_1 y un plano $\mathbf{\Pi}_1$. Para calcular ahora un vector con una dirección igual a la de un rayo reflejado por un espejo, considerando al rayo como una recta y al espejo como un plano, simplemente hay partir de un vector incidente al plano, paralelo al rayo incidente y generar un rotación del mismo sobre el plano de incidencia acorde a la ley óptica de la reflexión [14, 15]. Retomando los conceptos de la sección anterior, si la ecuación del plano $\mathbf{\Pi}_1$, correspondiente al plano que describe la superficie de un espejo, y la ecuación de la recta \mathbf{R}_1 asociada a un rayo incidente, son conocidas; entonces, tanto la normal al espejo, como el vector paralelo al rayo incidente, también son conocidos, entonces:

$$\text{Si: } \mathbf{\Pi}_1 : A_1x + B_1y + C_1z = D_1$$

Entonces, un vector normal al plano es:

$$\vec{N}_1 = A_1\hat{i} + B_1\hat{j} + C_1\hat{k}; \quad (\text{B.9})$$

$$\text{Si: } \mathbf{R}_1 : (x, y, z) = P_1 + t\vec{v}_1$$

Entonces, un vector paralelo al rayo incidente es:

$$\vec{v}_1 = v_{1x}\hat{i} + v_{1y}\hat{j} + v_{1z}\hat{k} \quad (\text{B.10})$$

Si el ángulo entre el vector incidente \vec{v}_1 y el vector normal \vec{N}_1 es llamado θ_1 , por la ley de reflexión, un ángulo θ_2 medido entre la normal \vec{N}_1 y un vector reflejando \vec{v}_2 , debe ser igual al ángulo de incidencia θ_1 ; entonces $\theta_1 = \theta_2$. La forma de calcular el ángulo de incidencia θ_1 es la siguiente:

$$\theta_1 = \arccos(\|\vec{N}_1\| \cdot \|\vec{v}_1\|) \quad (\text{B.11})$$

Donde $\|\vec{N}_1\|$ es vector normalizado de \vec{N}_1 y $\|\vec{v}_1\|$ es el vector normalizado \vec{v}_1 .

El vector reflejado \vec{v}_2 es igual a una rotación de vector \vec{v}_1 en un ángulo $2\theta_1$, sobre el plano de incidencia; lo que equivale a una rotación arbitraria del vector \vec{v}_1 alrededor de un vector arbitrario de rotación \vec{v}_{arb} , mediante una matriz homogénea de rotación arbitraria \mathbf{H}_{arb} . La forma de obtención del vector \vec{v}_{arb} es la siguiente:

$$\vec{v}_{arb} = \|\vec{N}_1\| \times \|\vec{v}_1\| \quad (\text{B.12})$$

Es importante señalar que, para asegurar que el vector del rayo reflejado \vec{v}_2 esté dado en el mismo lado del plano del espejo que el vector \vec{v}_1 y que su dirección sea la adecuada, la normal al plano del espejo \vec{N}_1 deberá tener cierto sentido antes de calcular los valores de θ_1 y \vec{v}_{arb} , mediante las ecuaciones B.11 y B.12:

$$\begin{aligned} \text{Considerando: } \theta &= \arccos(\|\vec{N}_1\| \cdot \|\vec{v}_1\|) \\ \text{Sí: } (\theta \leq 90) &\rightarrow \vec{N}_1 = \vec{N}_1 \\ \text{Si nó: } &\rightarrow \vec{N}_1 = -\vec{N}_1 \end{aligned}$$

La matriz homogénea de rotación arbitraria \mathbf{H}_{arb} , es en tanto:

$$\mathbf{H}_{arb} = \begin{bmatrix} e_1^2 + e_2^2 - e_3^2 - e_4^2 & 2(e_2e_3 + e_1e_4) & 2(e_2e_4 - e_1e_3) \\ 2(e_2e_3 - e_1e_4) & e_1^2 - e_2^2 + e_3^2 - e_4^2 & 2(e_3e_4 + e_1e_2) \\ 2(e_2e_4 + e_1e_3) & 2(e_3e_4 - e_1e_2) & e_1^2 - e_2^2 - e_3^2 + e_4^2 \end{bmatrix} \quad (\text{B.13})$$

Donde:

$$e_1 = -\cos(\theta_1) \quad (\text{B.14})$$

$$e_2 = v_{arb_x} \text{sen}(-\theta_1) \quad (\text{B.15})$$

$$e_3 = v_{arb_y} \text{sen}(-\theta_1) \quad (\text{B.16})$$

$$e_4 = v_{arb_z} \text{sen}(-\theta_1) \quad (\text{B.17})$$

Finalmente el vector paralelo al rayo reflejado es igual a:

$$\vec{v}_2 = \mathbf{H}_{\text{arb}} * \vec{v}_1 \quad (\text{B.18})$$

La recta \mathbf{R}_2 que es coincidente con el rayo reflejado, es igual a:

$$\mathbf{R}_2 : (x, y, z) = T_1 + t\vec{v}_2 \quad (\text{B.19})$$

Apéndice C

Rutinas para el Procesamiento de Imágenes Utilizando OpenCV

Las rutinas de código necesarias para realizar el procesamiento de imágenes necesario y descrito en la sección 3.3, se muestran a continuación. El código fue englobado en una clase escrita en lenguaje C++ que lleva por nombre `Procesamiento3`, que hace uso de la biblioteca de *OpenCV*.

Las funciones adjuntas a continuación permiten la ejecución de las siguientes acciones:

- Separación de imágenes (`Procesamiento3::processIt`).
- Resta de imágenes (`Procesamiento3::processIt`).
- Filtrado de imagen (`Procesamiento3::processIt`).
- Búsqueda de contornos en imagen (`Procesamiento3::processIt`).
- Sección de contornos por número mínimo de elementos (`Procesamiento3::processIt`).
- Cálculo de centroides de contornos (`Procesamiento3::processIt`).
- Impresión de contornos en imagen (`Procesamiento3::processIt`).
- Cambio de origen de coordenadas de centroide (`Procesamiento3::ChangeImgCenter`).
- Retorno de imagen procesada (`Procesamiento3::getIMG01_3`).

El código aquí presentado puede ser reutilizado en todo momento, con la debida configuración de un proyecto ya sea para aplicación de consola o *MFC* (*Microsoft Foundation Class, por sus siglas en ingles*); más información al respecto puede ser encontrada en el sitio web “<http://opencv.willowgarage.com/wiki/>”.

C.1. La clase CProcesamiento3

C.1.1. Procesamiento3.h

```
1
2 #if !defined PROCESAMIENTO3
3 #define PROCESAMIENTO3
4 // #pragma once
5 #include "cv.h"
6 #include "highgui.h"
7 #include "cxcore.h"
8 #include "math.h"
9
10 class CProcesamiento3
11 {
12 private:
13     private:
14     IplImage* IMG01; //entrada 1 (bgra) ... Leds encendidos // 4↔
15         Canales
16     IplImage* IMG02; //entrada 2 (bgra) ... Leds apagados // 4↔
17         Canales
18     IplImage* IMG01_3; //entrada 1 (bgr) ... Leds apagados ↔
19         // 3 Canales
20     IplImage* IMG02_3; //entrada 2 (bgr) ... Leds apagados ↔
21         // 3 Canales
22     IplImage* IMG03; //Imagen Restada (bgr) // 3 ↔
23         Canales
24     IplImage* IMG04; //Imagen Filtrada (Binaria) // 1 ↔
25         Canal
26     IplImage* IMG05; //Salida con nuevos circulos coloreados
```

```

21 //IplImage* IMG05; //Imagen de Salida
22
23 CvMemStorage* MemContornos; //Memoria de contornos
24 CvMemStorage* MemCentroides;
25
26 CvSeq* SeqContornos; //Secuencia de contornos
27 CvSeq* SeqCentroides; //Secuencia contenedora de centroides. ←
    Estructura: CvPoint + Index
28
29 int Nc; // Numero instantaneo de contornos encontrados (Por ←
    muestreo)
30 int X,Y,Reg,n;
31 float A,B,C,D;
32 CvPoint* Point; // Cada punto que pertece al contorno de una ←
    marca.
33
34 CvPoint2D32f PtNwCentro; // Centroide corregido de cada marca←
    individual tras efectura conversión a origen en centro de←
    imagen
35 CvPoint2D32f PtOldCentro; // Retorno a viejo centro de ←
    imagen
36
37 public:
38 Procesamiento3();
39 ~Procesamiento3();
40
41 ////////////////////////////////////////////////////////////////////←
42
43 ////////////////////////////////////////////////////////////////////←
44 CvPoint PSI; //Esquina superior izquierda de rectangulo negro←
    -> para el procesamiento se asignaran por referencia
45 CvPoint PID; //Esquina inferior derecha de rectangulo negro ←
    -> para el procesamiento se asignaran por referencia
46
47 int Bmin, Bmax, Gmin, Gmax, Rmin, Rmax; //Componentes de ←
    estructuras cvScalar de funcion cvInRangeS()

```

```

48     int Dim; //Dimensión del contorno válido
49     float XcNew,YcNew;
50
51     ////////////////////////////////////////←
52     ->Regresor de Arreglo Contenedor de Contornos
53     CvSeq* GetSeqCentroides();
54
55     ////////////////////////////////////////←
56     ->Regresor de Numero de Contornos
57     int getNc();
58
59     ////////////////////////////////////////←
60     ->Regresor de imagen 01
61     IplImage* getIMG01_3();
62
63     ////////////////////////////////////////←
64     ->Regresor de imagen 02
65     //IplImage* getIMG02()
66     //{
67     //  return IMG02;
68     //}
69
70     ////////////////////////////////////////←
71     ->Regresor de imagen 03
72     //IplImage* getIMG03()
73     //{
74     //  return IMG03;
75     //}
76
77     ////////////////////////////////////////←
78     ->Regresor de imagen 04
79     //IplImage* getIMG04()
80     //{
81     //  return IMG04;
82     //}
83
84     ////////////////////////////////////////←
85     bool justInitialized(IplImage *Image01,IplImage *Image02);
86
87     ////////////////////////////////////////←

```

```

78     void InitializeThen(IplImage *Image01,IplImage *Image02);
79
80     //////////////////////////////////////<
81     void processIt(IplImage *Image01,IplImage *Image02);
82
83     //////////////////////////////////////<
84     //void processThisEntry(IplImage *Image01,IplImage *Image02);<
85     // <- podria ser aqui
86     //////////////////////////////////////<
87     void release();
88     //////////////////////////////////////
89     void processThisEntry(IplImage *Image01,IplImage *Image02);
90     //////////////////////////////////////
91     CvPoint2D32f ChangeImgCenter(float Xc,float Yc,float Xf,float<
92     Yf);
93     //////////////////////////////////////
94     CvPoint2D32f OldImgCenter(float Xc,float Yc,float Xf,float Yf<
95     );
96     //////////////////////////////////////
97     void SetDim(int Enter_Dim);
98     //////////////////////////////////////
99     IplImage* getNewColored();
100    //////////////////////////////////////
101    void setNewColored(CvPoint2D32f XYcNew); //->Dibujar circulos<
102    de prediccion de modelo.
103    //////////////////////////////////////
104    IplImage* getNewColored2(CvPoint2D32f PosHta,CvPoint2D32f <
105    PosObj); //-> Admite un imagen de 4 canales y regresa <
106    una imagen de 3 canales con hta y objetivo dibujados
107
108 };
109 #endif

```

C.1.2. Procesamiento3.cpp


```

28
29  }
30  //////////////////////////////////////
31  void Procesamiento3::InitializeThen(IplImage *Image01, IplImage ←
    *Image02)
32  {
33      if(IMG01 != NULL)
34          {cvReleaseImage(&IMG01);}
35      if(IMG02 != NULL)
36          {cvReleaseImage(&IMG02);}
37      if(IMG01_3 != NULL)
38          {cvReleaseImage(&IMG01_3);}
39      if(IMG02_3 != NULL)
40          {cvReleaseImage(&IMG02_3);}
41      if(IMG03 != NULL)
42          {cvReleaseImage(&IMG03);}
43      if(IMG04 != NULL)
44          {cvReleaseImage(&IMG04);}
45      if(IMG05 != NULL)
46          {cvReleaseImage(&IMG05);}
47
48      //IMG01 = cvCreateImageheader(cvSize(Image01->width, Image01←
          ->height), Image01->depth, Image01->nChannels);
49      //IMG02 = cvCreateImageheader(cvSize(Image01->width, Image01←
          ->height), Image01->depth, Image01->nChannels);
50      IMG01_3 = cvCreateImage(cvSize(Image01->width, Image01->←
          height), Image01->depth, Image01->nChannels-1);
51      IMG02_3 = cvCreateImage(cvSize(Image01->width, Image01->←
          height), Image01->depth, Image01->nChannels-1);
52      IMG03 = cvCreateImage(cvSize(Image01->width, Image01->height)←
          , Image01->depth, Image01->nChannels-1);
53      IMG04 = cvCreateImage(cvSize(Image01->width, Image01->height)←
          , Image01->depth, Image01->nChannels-3);
54      //IMG05 = cvCreateImage(cvSize(Image01->width, Image01->←
          height), Image01->depth, Image01->nChannels-3);
55
56      MemContornos = cvCreateMemStorage(0); //64Kb por default

```

```

57     MemCentroides = cvCreateMemStorage(0);
58
59     SeqContornos = 0;
60     SeqCentroides = 0;
61 }
62 //////////////////////////////////////
63 void Procesamiento3::processIt(IplImage *Image01,IplImage *↵
        Image02)
64 {
65
66 //Copia inicial de imagenes de entrada:
67     IMG01 = cvCloneImage(Image01); //->Leds encendido
68     IMG02 = cvCloneImage(Image02); //->Leds apagados
69
70     cvCvtColor(IMG01,IMG01_3,CV_BGRA2BGR);
71     cvCvtColor(IMG02,IMG02_3,CV_BGRA2BGR);
72
73     cvReleaseImage(&IMG01);
74     cvReleaseImage(&IMG02);
75
76     cvRectangle(IMG01_3,cvPoint(PSI.x,PSI.y),cvPoint(PID.x,PID.y)↵
        ,cvScalar(0,0,0),-1,8,0);
77     cvRectangle(IMG02_3,cvPoint(PSI.x,PSI.y),cvPoint(PID.x,PID.y)↵
        ,cvScalar(0,0,0),-1,8,0);
78
79 //Resta de Imagenes
80     cvSub(IMG01_3,IMG02_3,IMG03);
81
82     cvInRangeS(IMG03,cvScalar(Bmin,Gmin,Rmin),cvScalar(Bmax,↵
        Gmax,Rmax),IMG04);
83
84     Nc = cvFindContours(IMG04,MemContornos,&SeqContornos,sizeof(↵
        CvContour),CV_RETR_CCOMP/*CV_RETR_LIST*/,↵
        CV_CHAIN_APPROX_NONE,cvPoint(0,0)); //<-Modificiacion ↵
        28/09/2012 Solo serán detectados contornos exteriores
85
86     n=0;A=0;B=0;

```

```

87     //CvSeq *CopySeq = 0;
88
89     CvSeqWriter writer;
90     cvStartWriteSeq(CV_32FC2, sizeof(CvSeq), sizeof(CvPoint2D32f)↵
        ,MemCentroides,&writer);
91     //for(CopySeq = SeqContornos; CopySeq != NULL; CopySeq = ↵
        CopySeq->h_next)
92     for(SeqContornos; SeqContornos != NULL; SeqContornos = ↵
        SeqContornos->h_next) //"->h_next": Navegación unicamente ↵
        por sobre el primer nivel del arbol de jerarquia de datos,↵
        esto es: "CV_RETR_CCOMP" Solo se toman en cuenta ↵
        contornos exteriores.
93     {
94
95     n = n + 1;
96     X = 0; Y = 0; Reg = 0;
97     if ((SeqContornos->total-1) >= Dim) //-> Solo aquellos ↵
        contornos compuestos por mas de 15 puntos serán válidos
98     {
99     for(int i = 0; i <= (SeqContornos->total-1); i++) //->Aqui se ↵
        calculan los centroides de las marcas
100    {
101    Point = CV_GET_SEQ_ELEM(CvPoint, SeqContornos, i);
102    X = X + Point->x;
103    Y = Y + Point->y;
104    Reg = Reg + 1;
105    }
106
107    A = (float)X/Reg;
108    B = (float)Y/Reg;
109    //C = X/Reg;
110    //D = Y/Reg;
111
112    CV_WRITE_SEQ_ELEM(cvPoint2D32f(A,B), writer); //---> Se graba ↵
        secuencia de centroides
113
114    cvCircle(IMG01_3, cvPoint(cvRound(A), cvRound(B)), 10, cvScalar↵
        (255,0,0), 3, 8, 0); //<-Aqui iba IMG05

```

```
115     }
116 }
117 SeqCentroides = cvEndWriteSeq(&writer);
118 }
119 //////////////////////////////////////
120 void Procesamiento3::release()
121 {
122     //FUNCION DEL DESTRUCTOR
123
124     if (IMG01_3!=NULL)
125 cvReleaseImage(&IMG01_3);
126     if (IMG02_3!=NULL)
127 cvReleaseImage(&IMG02_3);
128     if (IMG03!=NULL)
129 cvReleaseImage(&IMG03);
130     if (IMG04!=NULL)
131 cvReleaseImage(&IMG04);
132     if (IMG05!=NULL)
133 cvReleaseImage(&IMG05);
134
135     if (MemContornos != NULL) //Modificacion 08/02/2012: se ←
        puso el "if"
136 cvClearMemStorage(MemContornos);
137     if (MemCentroides != NULL)
138 cvClearMemStorage(MemCentroides);
139
140 }
141 //////////////////////////////////////
142 void Procesamiento3::processThisEntry(IplImage *Image01, ←
    IplImage *Image02)
143 {
144     if (!justInitialized(Image01, Image02))
145     {
146         InitializeThen(Image01, Image02);
147     }
148     processIt (Image01, Image02);
```

```
149  }
150  ///////////////////////////////////////////////////////////////////

151  IplImage* Procesamiento3::getIMG01_3()
152  {
153      return IMG01_3;
154  }
155  ///////////////////////////////////////////////////////////////////

156  CvSeq* Procesamiento3::GetSeqCentroides()
157  {
158      return SeqCentroides;
159  }
160  ///////////////////////////////////////////////////////////////////

161  int Procesamiento3::getNc()
162  {
163      return Nc;
164  }
165  ///////////////////////////////////////////////////////////////////

166  CvPoint2D32f Procesamiento3::ChangeImgCenter(float Xc,float ↵
    Yc,float Xf,float Yf) //(Xc y Yc), coordenadas de marca ↵
    encontrada,(Xf y Yf) distancia a la cual se ha configurado↵
    el centro de imagen
167  { //Cambio de definición de sistema coordenado de imagen ↵
    10/02/2013
168      PtNwCentro.x = Xc - Xf; /* PtNwCentro.x = Xf - Xc; */
169      PtNwCentro.y = Yf - Yc; /* PtNwCentro.y = Yf - Yc; */
170      return PtNwCentro;
171  }
172  ///////////////////////////////////////////////////////////////////

173  void Procesamiento3::SetDim(int Enter_Dim)
174  {
175      Dim = Enter_Dim;
176  }
177  ///////////////////////////////////////////////////////////////////
```

```
178   IplImage* Procesamiento3::getNewColored()
179   {
180       return IMG01_3;
181       //return IMG03;
182       //return IMG05;
183   }
184   ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////

185   void Procesamiento3::setNewColored(CvPoint2D32f XYcNew)
186   {
187       //IMG05 = cvCloneImage(IMG01_3); //→IMG05 declara aqui ←
           mismo su encabezado. Se destruye hasta el destructor.
188       //cvCircle(IMG05, cvPoint(cvRound(XYcNew.x), cvRound(XYcNew.y)) ←
           ,10, cvScalar(0,255,0),3,8,0);
189       //cvCircle(IMG03, cvPoint(cvRound(XYcNew.x), cvRound(XYcNew.y)) ←
           ,10, cvScalar(0,255,0),3,8,0);
190       cvCircle(IMG01_3, cvPoint(cvRound(XYcNew.x), cvRound(XYcNew.y)) ←
           ,10, cvScalar(0,255,0),3,8,0);
191   }
192   ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////

193   CvPoint2D32f Procesamiento3::OldImgCenter(float Xc, float Yc, ←
           float Xf, float Yf) //Regresa un punto a coordenadas ←
           originales de pantalla (esquina superior izquierda = ←
           oringen)
194   { //Cambio de definición de sistema coordenado de imagen ←
           10/02/2013
195       PtOldCentro.x = Xf+Xc; //Xf, Yf = Centro de imagen—— ←
           PtOldCentro.x = Xf - Xc;
196       PtOldCentro.y = Yf-Yc; ///Xc, Yc = Coordenadas del punto ←
           —— PtOldCentro.y = Yf - Yc;
197
198       return PtOldCentro;
199   }
200   ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////

201   IplImage* Procesamiento3::getNewColored2(CvPoint2D32f PosHta, ←
```

```
        CvPoint2D32f PosObj)
202  {
203
204      cvCircle(IMG01_3, cvPoint( cvRound(PosHta.x), cvRound(PosHta.y)↔
          ),10, cvScalar(0,255,0),3,8,0);
205  cvCircle(IMG01_3, cvPoint( cvRound(PosObj.x), cvRound(PosObj.y)↔
          ),10, cvScalar(255,0,0),3,8,0);
206  return IMG01_3;
207
208  }
209  //////////////////////////////////////
```

Apéndice D

Interfaz Gráfica para Validación Experimental

D.1. Entorno General

La interfaz gráfica de usuario desarrollada, está basada en el lenguaje programación de visual C++ y en las bibliotecas de MFC (*Microsoft Foundation Class, por sus siglas en inglés*); dicha interfaz fue creada para controlar por visión al robot industrial de la marca Fanuc, modelo M-16iB/20T, a través de la metodología de CSM. La interfaz también hace uso de la biblioteca de procesamiento de imágenes de *OpenCV* y el nombre dado a esta intefaz es *CamV_CRobot*. Las funciones realizables por medio de la interfaz gráfica creada, son todas las asociadas a la realización de los experimentos con maniobras de posicionamiento dado un objetivo en espacio de cámara; las funciones asociadas al cálculo de la resolución específica de una escena, además de las funciones requeridas para la medición de la sensibilidad del modelo de perspectiva recursivo. El entorno general de la interfaz gráfica es el siguiente:

A continuación se detalla la función de cada botón, selector y barras de desplazamiento:

- **INICIO:** Botón que inicial comunicación con robot (Fanuc modelo M-16iB/20T).
- **FIN:** Botón que termina comunicación con robot (Fanuc modelo M-16iB/20T).
- **Captura!:** Botón que realiza capturas de prueba ya sea en modo simple o mo-

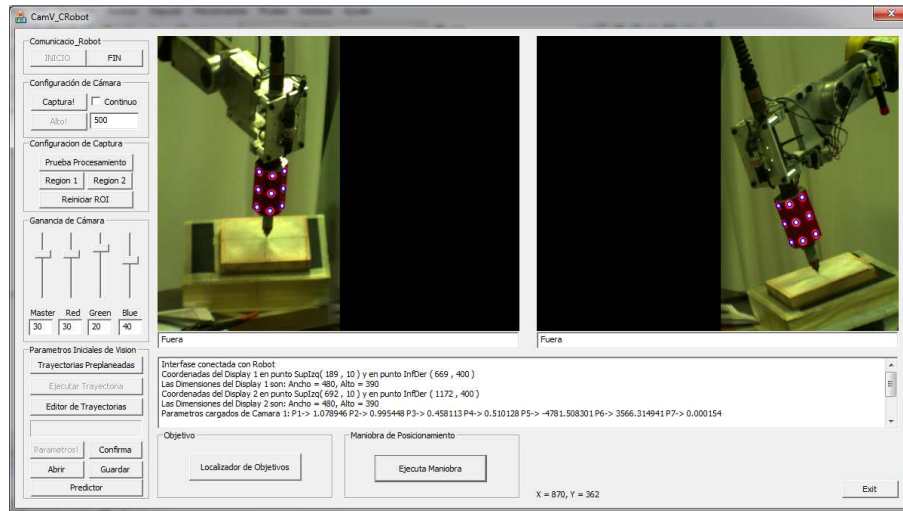


Figura D.1: Entorno general de la interfaz gráfica

do continuo; en el modo continuo se puede controlar el tiempo en milisegundos entre cada captura.

- **Continuo:** Sector de habilitación de modo continuo tanto de capturas como de pruebas de procesamiento de imágenes.
- **Alto!:** Detiene capturas en modo continuo.
- **Prueba procesamiento:** Botón que realiza una captura y muestra el procesamiento de imagen en pantalla para ambas cámaras virtuales.
- **Región 1:** Botón para configurar el área de la cámara virtual 1 mediante el ratón.
- **Región 2:** Botón para configurar el área de la cámara virtual 2 mediante el ratón.
- **Reiniciar ROI:** Botón que reinicia la configuración de cámara virtual 1 y cámara virtual 2.
- **Ganancia de Cámara:** Selectores de nombre “Master”, “Red”, “Green”, “Blue”, para controlar los parámetros de ganancia de cámara.
- **Trayectorias Treplaneadas:** Botón que abre el entorno de carga de trayectorias preplaneadas en formato “.TXT”.
- **Ejecutar Trayectoria:** Botón que ejecuta la trayectoria preplaneada cargada.

- **Editor de Trayectorias:** Botón que abre el entorno de generación manual de trayectorias preplaneadas.
- **Parámetros:** Botón que calcula los parámetros iniciales de cámara una vez que ha terminado una trayectoria preplaneada.
- **Confirma:** Botón que confirma la validez de los parámetros iniciales de cámara, al re-calcular las proyecciones en espacio de cámara de los puntos tridimensionales de las muestras provenientes de la trayectoria preplaneada.
- **Abrir:** Botón que abre y carga parámetros de cámara de una corrida anterior.
- **Guardar:** Botón que guarda parámetros de cámara una corrida en formato “.TXT”.
- **Predictor:** Botón que abre entorno de predicción de aparición de puntos en cámara.
- **Localizador de Objetivos:** Botón que abre entorno selección de objetivo en espacio de cámara.
- **Ejecuta Maniobra:** Botón que abre entorno de ejecución de trayectorias preplaneadas una vez se tienen parámetros de cámara iniciales.