

CENTRO DE INVESTIGACIONES  
EN ÓPTICA, A.C.

# "IMPLEMENTACIÓN DE UN CONTROL VISUAL PARA ROBOTS MÓVILES CON RUEDAS"



Como requisito para obtener el grado en:  
Maestro en Optomecatrónica

**Asesor:**

**Dr. J Ascención Guerrero Viramontes**

**Co-asesor:**

**Dr. Héctor Manuel Becerra Fermín**

**Presenta:**

**Ing. Oscar Salvador Torres Muñoz**

29 Octubre de 2014  
León, Guanajuato, México

## RESUMEN

En la actualidad las investigaciones en el área de la robótica móvil han tomado gran importancia debido al gran uso de robots en diversas aplicaciones y tareas como en procesos de automatización, transporte de materiales, exploración de zonas de riesgo, asistencia a personas discapacitadas etc; para ello es necesario desarrollar sistemas de control capaces de llevar a cabo la navegación autónoma de robots en forma eficiente y robusta.

Con el fin de lograr la navegación autónoma es necesario que el robot interactúe con el medio ambiente; para ello hoy en día se dispone de una amplia variedad de sensores, entre los cuales destacan los sistemas de visión, dada la importante ventaja de la medición sin contacto, su versatilidad, bajo costo y la rica información visual que proporcionan. Sin embargo, es importante mencionar que el uso adecuado y completo de la vasta información proporcionada por este tipo de sistemas, aun representa un reto en la comunidad científica.

En esta tesis se presentan dos esquemas de control visual para robots móviles con ruedas, utilizando un sistema de visión montado sobre el robot, de tal manera que el problema se enfrenta utilizando únicamente información visual, sin recurrir a otro tipo de sensor, o información proporcionada por la odometría.

Los esquemas de control presentados se basan en la estimación de una restricción geométrica impuesta entre un conjunto de imágenes de una misma escena. En particular se trabaja con el tensor trifocal porque es la restricción geométrica más general y además permite desarrollar sistemas de control capaces de funcionar en cualquier tipo de escenas y no únicamente en escenarios planos como en el caso de la homografía. Además reducen los problemas de mal condicionamiento que puede presentar la geometría epipolar.

## AGRADECIMIENTOS.

Le agradezco a Dios por permitirme continuar con mis estudios brindándome salud física y mental, y por estar presente en cada momento de mi vida, ayudándome a tomar las decisiones que más me han convenido.

Gracias a mi familia por todo su apoyo y cariño, lo cual me a ayudado no solo en esta etapa de mi formación, si no en cada etapa de mi vida.

Gracias a mi asesor y coasesores por su gran ayuda durante el desarrollo de mi tesis.

Gracias al CONACYT por apoyarme para cursar un posgrado de calidad.,

Gracias al CIO y al CIMAT por ayudarme a crecer como profesionista y como persona, por brindarme un espacio y el equipo necesario para desarrollar mi proyecto de Tesis.

# Índice general

<b>1. INTRODUCCIÓN</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Planteamiento del problema . . . . .	2
1.3. Objetivos . . . . .	3
1.3.1. Objetivo general . . . . .	3
1.3.2. Objetivos particulares . . . . .	3
1.4. Estructura de la tesis . . . . .	3
<b>2. Estado del arte</b>	<b>5</b>
2.1. Métodos del control visual . . . . .	5
2.2. Geometría de múltiples vistas . . . . .	6
<b>3. Fundamento teórico</b>	<b>7</b>
3.1. Control Visual . . . . .	7
3.2. Geometría epipolar . . . . .	8
3.3. Tensor trifocal . . . . .	9
3.3.1. Bases geométricas para el tensor trifocal . . . . .	10
3.3.2. Notación tensorial . . . . .	12
3.3.3. Tensor trifocal a partir de las posiciones de cámara . . . . .	12
3.4. Descriptores y detectores . . . . .	15
<b>4. Metodología</b>	<b>17</b>
4.1. Extracción de las características visuales . . . . .	17
4.2. Calibración de cámara. . . . .	18
4.3. Estimación del tensor trifocal . . . . .	20
4.3.1. Transferencia de puntos utilizando el TT . . . . .	22
4.3.2. Número de muestreos necesarios en RANSAC . . . . .	22
4.4. Rastreo de puntos . . . . .	23
4.5. Implementación . . . . .	24
<b>5. Controlador con primitivas de movimiento simples</b>	<b>26</b>
5.1. Definición del control para cada etapa . . . . .	28
5.1.1. Control directo con elementos del TT . . . . .	28
5.1.2. Control para seguimiento de trayectoria . . . . .	29
5.2. Resultados . . . . .	30
5.2.1. Evaluación general del tensor trifocal . . . . .	30
5.2.2. Pruebas con un simulador dinámico . . . . .	31
5.2.2.1. Control directo con elementos del TT . . . . .	34
5.2.2.2. Control para seguimiento de trayectoria. . . . .	42
5.2.3. Pruebas con imágenes reales . . . . .	48
5.3. Conclusiones . . . . .	53

## ÍNDICE GENERAL

<b>6. Control visual con control super-twisting</b>	<b>55</b>
6.1. Modelo cinemático de un robot móvil . . . . .	55
6.2. Planteamiento de la imagen virtual . . . . .	56
6.3. Esquema con una sola ley de control usando el TT . . . . .	58
6.4. Resultados . . . . .	60
6.5. Conclusiones . . . . .	66
<b>Bibliografía</b>	<b>66</b>

# Índice de figuras

1.1.	Esquema general de control visual. . . . .	2
1.2.	Tarea de posicionamiento para un esquema de control visual basado en imagen. . . . .	3
3.1.	Geometría epipolar. . . . .	9
3.2.	Una línea $L$ en el espacio 3D es proyectada como una tripleta de líneas correspondientes $(l, l', l'')$ en tres vistas. Inversamente los planos proyectados hacia atrás por cada una de las líneas correspondientes se cortan en una misma línea en el espacio 3D. . . . .	10
3.3.	Geometría entre tres ubicaciones de cámaras sobre un plano; a) con respecto a un sistema de referencia global; b) con respecto a un sistema de referencia local. . . . .	13
4.1.	Imágenes tomadas a diferentes distancias y orientaciones con respecto a un marco de referencia asociado al centro óptico de la cámara, para formar un sistema de ecuaciones con la finalidad de obtener los parámetros intrínsecos de la cámara. . . . .	19
4.2.	Robots móviles. a) Robot real. Robot simulado en Webots. . . . .	25
5.1.	Etapas del control visual basado en el tensor trifocal, utilizando primitivas de movimiento simples. . . . .	27
5.2.	Trayectoria de referencia utilizada en las etapas de control. . . . .	29
5.3.	Evolución de los elementos no nulos del tensor trifocal para el caso de un movimiento plano. La línea roja continua y la línea azul punteada corresponden a los elementos estimados a partir información visual e información métrica respectivamente. . . . .	32
5.4.	Análisis del comportamiento de los elementos del tensor trifocal visual (línea roja continua) que se seleccionaron como salidas del sistema (robot-cámara) a lo largo de una trayectoria en contraste con el TT métrico (línea azul punteada). a) Puntos de Harris. b) Puntos SURF. c) Puntos SIFT. . . . .	33
5.5.	Configuración del experimento donde el robot móvil debe avanzar para alcanzar su objetivo. a) Ubicación inicial del robot (superior) e imagen observada en dicho momento (inferior). b) Ubicación destino del robot (superior) e imagen observada. . . . .	35
5.6.	Evolución de las salidas del sistema. a) Elemento utilizado en la ley de control durante la primera etapa. b) Elemento empleado en la segunda etapa de control. c) Elemento utilizado en la tercera etapa de control. . . . .	36
5.7.	Resultados visuales. a) Imagen observada por el robot al termino de su trayectoria. b) Imagen correspondiente a la ubicación objetivo. . . . .	36
5.8.	Evolución de los valores RMS a lo largo de la trayectoria del robot. a) Valores RMS de la distancia en $x$ entre puntos en el plano imagen. b) Valores RMS de la distancia en $y$ entre puntos. c) Valores RMS de la distancia euclídiana entre puntos en la imagen. . . . .	37
5.9.	Evolución de la ubicación del robot. a) Posición lateral. b) Posición longitudinal. c) Orientación. . . . .	38
5.10.	Velocidades del robot. a) Velocidad de traslación. b) Velocidad de rotación. . . . .	38
5.11.	Movimiento de los puntos en la imagen. Los puntos verdes indican la posición inicial; En azul se representa la trayectoria de los puntos en la imagen. Los puntos amarillos representan la posición alcanzada al final de la trayectoria del robot móvil. Y los puntos rojos corresponden a la posición objetivo. . . . .	39
5.12.	Configuración del experimento donde el robot móvil debe retroceder para alcanzar su objetivo. a) Ubicación inicial del robot (superior) e imagen observada en dicho momento (inferior). b) Ubicación destino del robot (superior) e imagen observada. . . . .	40
5.13.	Resultados visuales. a) Imagen observada por el robot al termino de su trayectoria. b) Imagen correspondiente a la ubicación objetivo. . . . .	40

## ÍNDICE DE FIGURAS

5.14. Evolución del valor RMS de la distancia euclidiana entre puntos en el plano de la imagen. . . . .	41
5.15. Velocidades del robot. a) Velocidad de traslación. b) Velocidad de rotación. . . . .	41
5.16. Movimiento de los puntos en la imagen. Los puntos verdes indican la posición inicial. En azul se representa la trayectoria de los puntos en la imagen. Los puntos amarillos representan la posición alcanzada al final de la trayectoria del robot móvil. Y los puntos rojos corresponden a la posición objetivo. . . . .	42
5.17. Resultados visuales. a) Imagen observada por el robot al termino de su trayectoria. b) Imagen correspondiente a la ubicación objetivo. . . . .	43
5.18. Evolución de las salidas del sistema. a) Elemento utilizado en la ley de control durante la primera etapa. b) Elemento empleado en la segunda etapa de control. c) Elemento utilizado en la tercera etapa de control. . . . .	44
5.19. Evolución de la ubicación del robot. a) Posición lateral. b) Posición longitudinal. c) Orientación. . . . .	44
5.20. Evolución de los valores RMS a lo largo de la trayectoria del robot. a) Valores RMS de la distancia en $x$ entre puntos en el plano imagen. b) Valores RMS de la distancia en $y$ . c) Valores RMS de la distancia euclidiana entre puntos. . . . .	45
5.21. Velocidades del robot. a) Velocidad de traslación. b) Velocidad de rotación. . . . .	45
5.22. Evolución de las funciones de error del sistema. a) función de error de la primera etapa del control. b) función de error de la segunda etapa del control. c) Función de error de la tercera etapa del control. . . . .	46
5.23. Movimiento de los puntos en la imagen. Los puntos verdes indican la posición inicial. En azul se representa la trayectoria de los puntos en la imagen. Los puntos amarillos representan la posición alcanzada al final de la trayectoria del robot móvil. Y los puntos rojos corresponden a la posición objetivo. . . . .	46
5.24. Resultados visuales. a) Imagen observada por el robot al termino de su trayectoria. b) Imagen correspondiente a la ubicación objetivo. . . . .	47
5.25. Evolución del valor RMS de la distancia euclidiana entre puntos en el plano de la imagen. . . . .	47
5.26. Velocidades del robot. a) Velocidad de traslación; b) Velocidad de rotación. . . . .	48
5.27. Movimiento de los puntos en la imagen. Los puntos verdes indican la posición inicial. En azul se representa la trayectoria de los puntos en la imagen. Los puntos amarillos representan la posición alcanzada al final de la trayectoria del robot móvil. Y los puntos rojos corresponden a la posición objetivo. . . . .	48
5.28. Experimento con imágenes reales. a) Imagen correspondiente al punto de partida en la trayectoria evaluada. b) Imagen final. . . . .	49
5.29. Elementos del TT estimados al emplear puntos seleccionados con SIFT. a) Elemento $T_{221}$ . b) Elemento $T_{223}$ . . . . .	50
5.30. Elementos del TT estimados al emplear puntos seleccionados con SURF. a) Elemento $T_{221}$ . b) Elemento $T_{223}$ . . . . .	51
5.31. Elementos del TT estimados al emplear puntos seleccionados con Harris. a) Elemento $T_{221}$ . b) Elemento $T_{223}$ . . . . .	52
6.1. Definición del esquema del robot móvil. . . . .	56
6.2. Esquema de la imagen virtual. Las 3 posiciones de cámara inicial se encuentran sobre un mismo plano. . . . .	57
6.3. Imágenes generadas transfiriendo puntos con el tensor trifocal, utilizando imágenes reales. Los puntos azules y verdes representan la imagen original y virtual respectivamente. a) Con desplazamiento vertical $t_y = 0.2$ . b) Sin desplazamiento vertical. . . . .	60
6.4. Imágenes generadas transfiriendo puntos con el tensor trifocal utilizando imágenes de Webots. Los puntos azules y verdes representan la imagen original y virtual respectivamente. a) Con desplazamiento vertical $t_y = 0.2$ . b) Sin desplazamiento vertical. . . . .	61
6.5. Resultados de simulaciones en Matlab. a) Trayectorias seguida por el robot móvil desde diferentes ubicaciones en el plano x-z. b) Evolución de la posición y orientación del robot. . . . .	64
6.6. Resultados de simulaciones en Matlab. a) Velocidades calculadas. b) Evolución de las salidas del sistema (mediciones visuales). . . . .	65

# Índice de tablas

5.1. Elementos del tensor trifocal calculados a partir de información métrica e información visual, donde las ubicaciones de cámara están determinadas por $C_1 = [-2, -11, 0]$ , $C_2 = [-5, -9, 0]$ y $C_3 = [0, 0, 0]$ . . . .	31
6.1. Pruebas realizadas para estimar el tensor trifocal con diferentes niveles de ruido en la imagen. $TT$ y $\overline{TT}$ representan el tensor calculado con las 3 imágenes originales y el estimado introduciendo la imagen virtual respectivamente. . . . .	62



# Capítulo 1

## INTRODUCCIÓN

### 1.1. Motivación

Las investigaciones en la robótica móvil han cobrado gran importancia en la actualidad. Continúan incrementándose los trabajos desarrollados en torno a esta área, esto motivado por el gran número de tareas en las cuales pueden participar los robots móviles, como pueden ser en procesos industriales, actividades domésticas, apoyo a visitas guiadas, asistencia a personas discapacitadas, exploración de la superficie de otros planetas o zonas de riesgo, e incluso en la asistencia de cirugías, etc. Sin embargo, a pesar de que la robótica móvil es un campo maduro en la investigación aún es un área en desarrollo en la que hay mucho por innovar, y así contribuir a desarrollar mejores esquemas de control que permitan la ejecución de tareas cada vez más complejas.

Hoy en día, la visión por computadora se acepta como uno de los mejores medios de sensado para la navegación autónoma de robots, razón por la cual se utiliza ampliamente en esquemas de control como los que se presentan en esta tesis. Los esfuerzos de las investigaciones se han centrado en el uso de la visión robótica utilizando sistemas de visión monoculares. Esos sistemas a pesar de no proporcionar información de la profundidad a la cual se encuentra la escena por no ser propiamente sensores 3D, proveen gran información visual. Dicha información se puede procesar con un costo computacional relativamente bajo en comparación con el pasado, gracias a los avances computacionales, lo cual motiva a incursionar en este campo del conocimiento el cual alguna vez fue un área difícil de abordar.

Durante la última década, se han logrado grandes avances en la comprensión y en el modelado matemático de la geometría de múltiples vistas aplicadas a la visión por computadora, permitiendo desarrollar tareas que antes parecían mucho más complejas. Sin embargo, los conocimientos vastos de esta área no se han explotado al cien por ciento en el control visual de robots, impulsando el desarrollo de nuevos esquemas de control con la finalidad de dotar al robot con las herramientas y habilidades necesarias para efectuar tareas específicas.

A pesar de los grandes adelantos que se han obtenido en los últimos años, la visión robótica sigue siendo un área en desarrollo, y un campo de oportunidades para la innovación. En efecto la comunidad científica aún está lejos de poder desarrollar un sistema de visión con la percepción visual humana; pues es una disciplina bastante compleja que guarda fuertes conexiones con las matemáticas, la física, las ciencias de la computación, las ciencias neurológicas y la psicología de la percepción.

Las innumerables aplicaciones de la visión por computadora la convierten en una área que promueve la generación de nuevos conocimientos, y aún más en conjunto con el control visual de robots móviles. Recientemente esta tecnología se comienza a incorporar poco a poco en la vida cotidiana de las personas. Un ejemplo claro es la industria automotriz que ha iniciado a incluir algunas funciones de control visual, y no hay duda alguna que en los próximos años esta disciplina experimentará grandes avances que impactarán fuertemente en la sociedad humana.

## 1.2. Planteamiento del problema

El control visual o visual servo-control, es un proceso que minimiza una tarea especificada visualmente mediante un sistema de visión para controlar el movimiento de un robot. En la Figura 1.1, se muestra un esquema general de control visual.

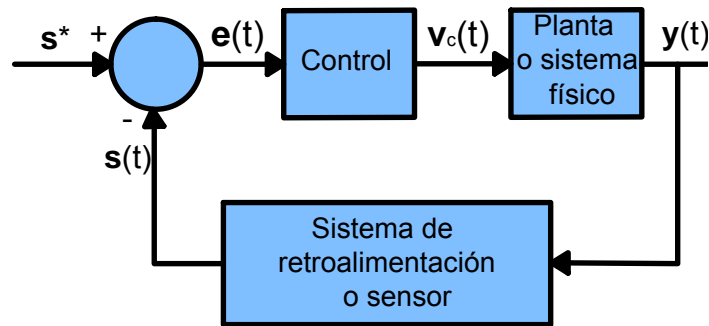


Figura 1.1: Esquema general de control visual.

donde  $s^*$  es la referencia visual en el esquema de control,  $s(t)$  las mediciones actuales de las características visuales,  $V_c(t)$  las velocidades del sistema (robot-cámara),  $e(t)$  el error del sistema, y  $y(t)$  la ubicación actual del robot.

Los esquemas de control visual que se presentan en esta tesis son basados en imagen, debido a que solo utilizan información visual extraída de un conjunto de imágenes, sin recurrir a otro tipo de información 3D.

El objetivo de un esquema de control visual basado en imagen (IBVS) para robots móviles como los que se presentan en esta tesis, consiste en minimizar una función de tarea que permita conducir de manera autónoma un robot, desde una ubicación inicial, hasta una ubicación destino mediante información visual de retro-alimentación, usando una cámara como sensor.

Es importante mencionar que durante la navegación del robot móvil, jamás se debe perder del campo de vista el objetivo, de lo contrario resultaría imposible cumplir con el objetivo de la tarea de posicionamiento.

La Figura 1.2, muestra de manera gráfica la tarea de posicionamiento de un esquema de control visual basado en imagen.

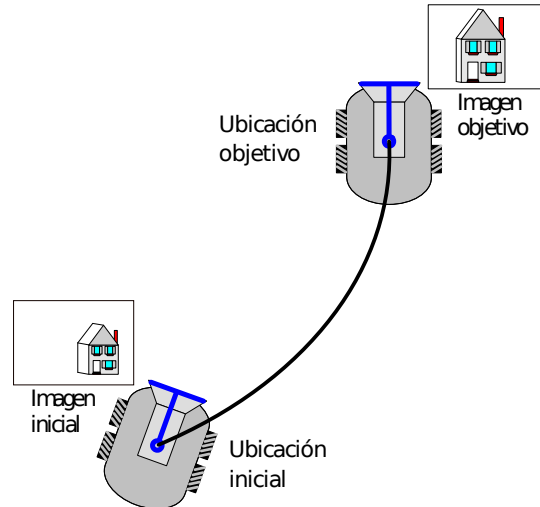


Figura 1.2: Tarea de posicionamiento para un esquema de control visual basado en imagen.

## 1.3. Objetivos

### 1.3.1. Objetivo general

El objetivo principal de la tesis consiste en la implementación de un control visual para robots móviles con ruedas basado en las restricciones geométricas inducidas por el tensor trifocal, para lo cual es necesario integrar conceptos formales de la teoría del control y de la visión por computadora; logrando así un esquema de control que permita la navegación autónoma de un robot móvil.

### 1.3.2. Objetivos particulares

1. Implementar una estimación robusta del tensor trifocal.
2. Comparar el desempeño de la estimación del tensor trifocal utilizando diferentes detectores y descriptores de características visuales.
3. Desarrollar un programa en lenguaje C ++ para llevar a cabo el control visual de un robot móvil.
4. Implementar en software de simulación el control visual basado en retro-alimentación del tensor trifocal y evaluarlo en un ambiente sintético.
5. Familiarizarse con la forma de programación de los robots móviles del tipo Pioneer.
6. Diseñar, implementar y evaluar diferentes experimentos usando una plataforma robótica real.

## 1.4. Estructura de la tesis

En esta sección, se describe la estructura y el contenido de la tesis. En el capítulo 2, se presenta el estado del arte del control visual de robots móviles, donde se muestran algunas de las principales aportaciones que se han desarrollado en el área durante los últimos años.

En el capítulo 3, se presentan los conocimientos teóricos básicos para el entendimiento y la implementación de los esquemas de control que se presentan en esta tesis. En el capítulo 4, se detalla la metodología seguida para el desarrollo de la tesis.

En el capítulo 5, se presenta un esquema de control visual utilizando el tensor trifocal basado en primitivas de movimiento simple, los resultados obtenidos de su implementación y experimentación, tanto en simulación como en casos reales. El capítulo 6 está dedicado a un esquema de control visual basado en el tensor trifocal donde se introduce el uso de una imagen virtual.

# Capítulo 2

## Estado del arte

El control visual (visual servoing) de robots se refiere al uso de información obtenida por medio de la visión por computadora para controlar el movimiento de un robot móvil.

El control visual de robots es un área en desarrollo, donde en los últimos años se han presentado avances y contribuciones muy importante; muchos de esos trabajos se centran en planteamientos generales para una cámara que se mueve con 6 grados de libertad [13, 27], los cuales no son aplicables directamente en robots móviles con restricciones no holonómicas. Para el caso de los robots móviles, los cuales son de interés particular en esta tesis, también se han presentados contribuciones recientemente [1, 6, 15]. Una revisión detallada de los trabajos desarrollados en los últimos años en el campo de la visión por computadora para la navegación de robots móviles se presenta en [14].

### 2.1. Métodos del control visual

Los enfoques en el campo del control visual se pueden clasificar de acuerdo a varios criterios. Un método por ejemplo se puede clasificar dependiendo de las restricciones del movimiento del robot. Otros criterios de clasificación se basan en el número de cámaras utilizadas, o en la configuración de los sistemas de visión.

La información visual se puede obtener utilizando una cámara que se monta sobre el robot, o en alguna ubicación en el área de trabajo de tal manera que se encuentre observando el robot, aunque se pueden presentar otras configuraciones donde se involucre el uso de más de una cámara. En base a este criterio se pueden presentar las siguientes configuraciones:

- **Configuración eye-in-hand:** la cámara se monta sobre el efector final del robot. Aquí se presenta una relación normalmente constante entre la posición de la cámara, y la posición del efector final del robot.
- **Configuración eye-to-hand:** la cámara se fija en algún lugar en el espacio de trabajo. Para este caso la imagen objetivo observada por la cámara es independiente del movimiento del robot, a menos que el objetivo sea en si el efector final del robot.

Una de las clasificaciones más populares para los métodos de control visual, se basa en la forma en la que se utiliza la información extraída de las imágenes para alcanzar el objetivo. En base a esto se tienen los siguientes esquemas:

- **Control visual basado en posición:** La posición del robot se recupera con respecto al ambiente de trabajo o con respecto al objetivo. Un ejemplo de este tipo esquema es [26], donde se implementa

un sistema de control de posición remota para un robot móvil. Las imágenes capturadas se procesan con un algoritmo a través del cual se determina la posición y orientación del robot, y finalmente se envían los comandos de control desde una PC al robot móvil utilizando módulos para comunicación inalámbrica.

Otro ejemplo de éste tipo de esquema utilizado en robots móviles es [11]; donde el objetivo consiste en controlar la posición de un robot móvil utilizando un filtro de Kalman, en combinación con técnicas de lógica difusa. La fusión de ambas técnicas se realiza con la finalidad de poder controlar el robot en un terreno desconocido.

- **Control visual basado en imagen:** el control visual utiliza información 2D de extraída de la imagen sin necesidad de recuperar la posición del robot en el espacio de trabajo, o cualquier otra información 3D. Un ejemplo de este enfoque se presenta en [30]. En ese trabajo se presenta un método para la navegación de robots móviles en un espacio 3D, donde el control del movimiento se llevaba a cabo mediante la recuperación de la postura relativa del robot con respecto a la deseada. En el enfoque que se presenta no se requiere conocer información 3D del medio ambiente, pero es necesario conocer los parámetros intrínsecos de la cámara. El objetivo consiste en alcanzar una posición y orientación determinada por una imagen previamente capturada.

En torno a este enfoque se han desarrollado varios trabajos donde se presentan alternativas para hacer frente a la dependencia del conocimiento de la profundidad de las características visuales [33, 25].

- **Control híbrido:** toma las ventajas de ambos enfoques evitando sus inconvenientes. Un ejemplo de esta clase de esquema se presenta en [15]. Allí mediante la comparación entre puntos correspondientes a partir de dos imágenes diferentes, se deriva una transformación que relaciona la posición y orientación actual del robot, con la posición y orientación deseada. Se utilizan técnicas basadas en Lyapunov para construir una estimación adaptativa con la finalidad compensar la falta de información de la profundidad de la escena.

Un par de tutoriales que muestran los fundamentos teóricos básicos y una introducción del control visual del robots se pueden encontrar en [8], mientras que una descripción de conceptos más avanzados y algunos enfoques recientes como el control visual híbrido que combina las ventajas de los esquemas de control visual basados en imagen (IBVS), y basados en posición (PBVS) se presentan en [9].

## 2.2. Geometría de múltiples vistas

Los esquemas de control visual que se muestran en esta tesis corresponde al tipo IBVS, donde el error se evalúa directamente de la posición de las características visuales en la imagen.

Las características visuales más utilizadas en los enfoques clásicos de IBVS son puntos de interés en las imágenes; debido a que son las características más simples que se pueden extraer. Sin embargo, en los últimos años se le ha dado importante atención al uso de otro tipo de primitivas geométricas (por ejemplo líneas), ya que en muchos casos el uso de puntos no resulta suficiente para definir por completo la escena contemplada.

Las restricciones geométricas entre imágenes se han utilizado en varios esquemas de control, algunos de ellos basados en la homografía o las restricciones geométricas inducidas por la geometría epipolar [34, 28, 5, 15]; donde comúnmente se presentan singularidades o degeneraciones que impiden cumplir satisfactoriamente con la función de tarea utilizando una sola ley de control.

Otros trabajos se basan las restricciones inducidas por el tensor trifocal [23, 16]. Este tipo de restricciones son las que se prefirieron para los esquemas de control presentados en esta tesis, ya que son más robustas y mejor definidas que las inducidas por geometría epipolar o la homografía.

# Capítulo 3

## Fundamento teórico

En este capítulo se presenta la teoría de los temas necesarios para el desarrollo de la tesis. Se abarcan principios básicos de control visual. Se describe ampliamente el tema del tensor trifocal en el cual se basan los esquemas de control visual que se presentan; y como antecedente se incluye de manera breve otra restricción geométrica ampliamente utilizada en el control visual de robots, como es la geometría epipolar.

Con la finalidad de estimar el tensor trifocal se pueden utilizar varios tipos de características visuales, como es el caso de los puntos de interés en las imágenes procesadas. Para la selección de los puntos a usar se realizaron pruebas con diferentes descriptores y detectores de puntos, los cuales se describen brevemente.

### 3.1. Control Visual

El control visual se refiere al uso de información extraída con ayuda de la visión por computadora para controlar el movimiento de un robot [8]. El objetivo de todos los esquemas de control visual consiste en minimizar un error  $\mathbf{e}(t)$ , el cual se puede definir de manera general como:

$$\mathbf{e}(t) = \mathbf{s}(\mathbf{m}(t), \mathbf{a}) - \mathbf{s}^* . \quad (3.1)$$

En la ecuación anterior  $\mathbf{m}(t)$ , representa un conjunto de mediciones en la imagen (por ejemplo las coordenadas de un conjunto de puntos en la imagen, o las coordenadas del centroide de un objeto en la imagen). Estas mediciones en la imagen se utilizan para estimar un vector  $\kappa$  de características visuales  $\mathbf{s}(\mathbf{m}(t), \mathbf{a})$ , donde  $\mathbf{a}$  representa conocimiento adicional sobre el sistema (por ejemplo los parámetros intrínsecos de la cámara). El vector  $\mathbf{s}^*$  representa los valores deseados para las características visuales, como pueden ser las coordenadas de los puntos de interés en la imagen.

Los esquemas de control visual difieren principalmente en la forma que se define  $\mathbf{s}$ . Existen esquemas donde  $\mathbf{s}$  consiste en un conjunto de características que se extraen directamente a partir información de la imagen; en otros esquemas  $\mathbf{s}$  consiste en un conjunto de parámetros 3D, que se deben estimar a partir de la información extraída de la imagen.

Una vez seleccionado el vector  $\mathbf{s}$ , el siguiente paso consiste en diseñar el esquema de control; uno de los enfoques más directos se basa en diseñar un controlador de velocidad. Para diseñar un controlador de velocidad, es necesario conocer la relación de las variaciones en el tiempo de  $\mathbf{s}$  con la velocidad de desplazamiento de la cámara. Denotando la velocidad espacial de una cámara como  $\mathbf{v}_c = (v_c, \omega_c)$ , donde  $v_c$  y  $\omega_c$  representan la velocidad lineal instantánea, y la velocidad angular instantánea del origen del marco de referencia de la cámara respectivamente; la relación entre  $\mathbf{v}_c$  y  $\mathbf{s}$  está dada por:

$$\dot{\mathbf{s}} = \mathbf{L}_s \mathbf{v}_c, \quad (3.2)$$

donde  $\dot{\mathbf{s}}$  es una matriz de dimensiones  $\kappa \times 6$ , a la cual se le conoce como la matriz de interacción relacionada con  $\mathbf{s}$ . A partir de las ecuaciones 3.1 y 3.2 se puede obtener la relación entre la velocidad de la cámara y la variación en el tiempo del error:

$$\dot{\mathbf{e}} = \mathbf{L}_e \mathbf{v}_c, \quad (3.3)$$

donde  $\mathbf{L}_e = \mathbf{L}_s$ .  $\mathbf{V}_c$  se considera como el vector de entradas del sistema. Si se desea asegurar que el error decrezca en forma exponencial (por ejemplo  $\mathbf{e} = e^{-\lambda t} \mathbf{e}_0$ ), se obtiene:

$$\mathbf{V}_c = -\lambda \mathbf{L}_e^+ \mathbf{e}, \quad (3.4)$$

donde  $\mathbf{L}_e^+$  de dimensiones  $6 \times \kappa$ , es la pseudo-inversa de la matriz  $\mathbf{L}_e$ , la cual se obtiene como  $\mathbf{L}_e^+ = (\mathbf{L}_e^T \mathbf{L}_e)^{-1} \mathbf{L}_e^T$ , cuando  $\mathbf{L}_e$  es de rango completo. Cuando  $\kappa = 6$  (caso en el que se tiene una matriz cuadrada) si el  $\det(\mathbf{L}_e) \neq 0$ , es posible invertir  $\mathbf{L}_e$ . Con lo anterior la ley de control se define como:

$$\mathbf{V}_c = -\lambda \mathbf{L}_e^{-1} \mathbf{L}_e \mathbf{e}. \quad (3.5)$$

## 3.2. Geometría epipolar

La geometría epipolar es la geometría proyectiva intrínseca entre dos vistas. Ésta es independiente de la escena y sólo depende de los parámetros intrínsecos de la cámara y del movimiento entre las dos vistas de la misma escena.

Dadas dos imágenes de la misma escena capturadas desde diferentes ubicaciones, la geometría epipolar es esencialmente la geometría dada por la intersección de los planos de cada imagen con una familia de planos que contienen la línea base. La línea base es la línea que une los centros ópticos de las cámaras.

Un punto  $\mathbf{X}$  en el espacio 3D se proyecta en dos imágenes como los puntos correspondientes  $\mathbf{x}$  y  $\mathbf{x}'$  tal como se aprecia en la Figura 3.1, siendo estos puntos y los centros de cámara ( $\mathbf{C}$  y  $\mathbf{C}'$ ) coplanares; donde se observa que los rayos proyectados por  $\mathbf{x}$  y  $\mathbf{x}'$  cortan en  $\mathbf{X}$  definiendo el plano  $\pi$ .



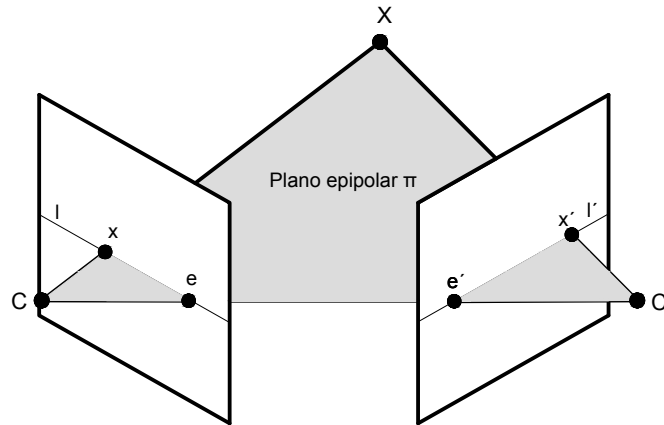


Figura 3.1: Geometría epipolar.

Suponiendo que se conoce la posición exacta de  $\mathbf{x}$  en la primera imagen, y se desea obtener  $\mathbf{x}'$ , con lo descrito anteriormente se sabe que el rayo proyectado por el punto desconocido debe caer en el plano  $\pi$ , y por lo tanto  $\mathbf{x}'$  debe encontrarse en la segunda imagen sobre la línea de intersección ( $l'$ ) del rayo proyectado por  $\mathbf{x}$  con el plano  $\pi$ .

Las intersecciones de las líneas que unen los centros ópticos de cada cámara con el plano de la imagen se denominan epipolos ( $\mathbf{e}_1$  y  $\mathbf{e}_2$ ). Cualquier plano que contenga la línea epipolar se denomina plano epipolar. La intersección entre un plano epipolar y el plano de la imagen se conoce como línea epipolar.

La representación algebraica de la geometría epipolar se hace a través de lo que se denomina como matriz fundamental. En otras palabras la matriz fundamental es la transformación proyectiva de un punto  $\mathbf{x}$  en la primera imagen a una línea  $l'$  en la segunda imagen.

La matriz fundamental es de dimensiones  $3 \times 3$  y al representar un mapeo de un espacio proyectivo bidimensional a uno unidimensional, debe ser de rango 2. La relación establecida por la matriz fundamental se puede expresar como:

$$\mathbf{F}\mathbf{x} \propto l' \quad (3.6)$$

Una condición que debe satisfacer la matriz fundamental para cualquier par de puntos correspondientes en dos vistas es la llamada restricción epipolar, la cual se expresa como:

$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0 \quad (3.7)$$

Por lo anterior, el conocimiento de un número suficiente de correspondencias de puntos ( $\mathbf{x}, \mathbf{x}'$ ) permite recuperar los parámetros de  $\mathbf{F}$  [18].

### 3.3. Tensor trifocal

El tensor trifocal (TT) juega un papel análogo en tres vistas al jugado por la matriz fundamental en dos vistas. Éste encapsula todas las relaciones geométricas entre tres vistas independientes de una misma

escena.

El tensor trifocal depende únicamente del movimiento entre vistas de una misma escena, y de los parámetros intrínsecos de la cámara. Se define por las matrices de cámara de cada una de las vistas. Sin embargo, se puede estimar a partir de correspondencias entre las imágenes sin necesidad de conocimiento alguno sobre el movimiento relativo entre cámaras.

El TT consiste en un conjunto de 3 matrices de  $3 \times 3$ , es decir cuenta con 27 elementos de los cuales únicamente 18 representan grados de libertad independientes. .

### 3.3.1. Bases geométricas para el tensor trifocal

Existen varios caminos en los que el tensor trifocal puede ser planteado y definido, pero en este apartado se toma como punto de partida la relación de incidencia entre 3 líneas correspondientes.

Supongamos que una línea en el espacio 3D, es proyectada en 3 vistas como se aprecia en la Figura 3.2. Los planos proyectados hacia atrás por las líneas en cada una de las imágenes deben unirse en una sola línea en el espacio. Dicha línea 3D es la que se proyecta en cada una de las vistas. Dado que en general tres planos arbitrarios en el espacio no se unen en una única línea, esta condición de incidencia geométrica proporciona una verdadera restricción entre un conjunto de líneas.

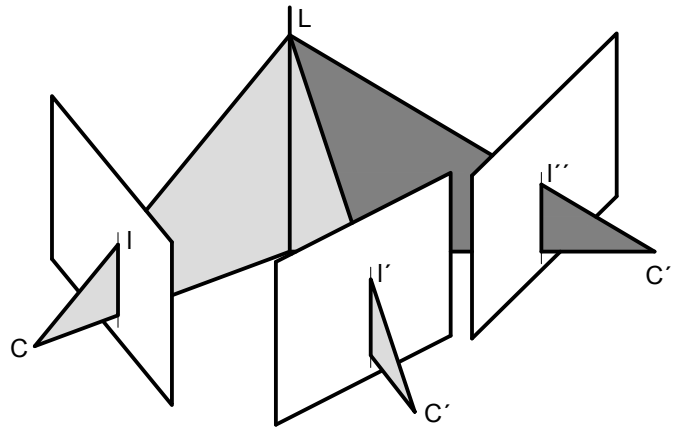


Figura 3.2: Una línea  $L$  en el espacio 3D es proyectada como una tripleta de líneas correspondientes ( $l, l', l''$ ) en tres vistas. Inversamente los planos proyectados hacia atrás por cada una de las líneas correspondientes se cortan en una misma línea en el espacio 3D.

Denotamos a un conjunto de líneas correspondientes como  $l_i, l'_i$  y  $l''_i$ , y las matrices de cámara para las 3 vistas como  $\mathbf{P} = [\mathbf{I}|0]$ ,  $\mathbf{P}' = [\mathbf{A}|\mathbf{a}_4]$  y  $\mathbf{P}'' = [\mathbf{B}|\mathbf{b}_4]$  respectivamente, donde  $\mathbf{A}$  y  $\mathbf{B}$  son matrices de  $3 \times 3$ , y los vectores  $\mathbf{a}_i$  y  $\mathbf{b}_i$  representa la  $i$ -ésima columna de su respectiva matriz de cámara para  $i = 1 \dots 4$ . De hecho,  $\mathbf{a}_4$  y  $\mathbf{b}_4$  son los epipolos en la segunda y tercera vista obtenidos con respecto a la primera imagen.

Cada una de las 3 líneas corresponde a la proyección de un plano en el espacio como se muestra en la Figura 3.2. dichos planos son:

$$\pi = \mathbf{P}^T \mathbf{l} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad (3.8)$$

$$\pi' = \mathbf{P}'^T \mathbf{l}' = \begin{pmatrix} \mathbf{A}^T \mathbf{l}' \\ \mathbf{a}_4^T \mathbf{l}' \end{pmatrix}, \quad (3.9)$$

$$\pi'' = \mathbf{P}''^T \mathbf{l}'' = \begin{pmatrix} \mathbf{B}^T \mathbf{l}'' \\ \mathbf{b}_4^T \mathbf{l}'' \end{pmatrix}. \quad (3.10)$$

Como las tres líneas en las imágenes son derivadas a partir de una sola línea en el espacio, se establece que estos tres planos no son independientes y deben cortarse en la línea 3D que los origina. Esta restricción de intersección puede ser expresada algebraicamente por el requerimiento de que la matriz  $\mathbf{M}$  de  $4 \times 3$ ,  $\mathbf{M} = [\pi, \pi', \pi'']$  sea de rango 2. Esto se puede ver de la siguiente manera: puntos en la línea de intersección de dos planos se pueden representar como  $\mathbf{X} = \alpha X_1 + \beta X_2$ , con  $X_1$  y  $X_2$  linealmente independientes. Puesto que todos los puntos a lo largo de la línea de intersección caen dentro de cada uno de los planos y  $\pi^T \mathbf{X} = \pi'^T \mathbf{X} = \pi''^T \mathbf{X} = 0$  se puede deducir que  $\mathbf{M}^T \mathbf{X} = \mathbf{M}^T X_1 = \mathbf{M}^T X_2$ .

Esta restricción de intersección induce una restricción entre las líneas proyectadas en cada una de las imágenes  $(\mathbf{l}, \mathbf{l}', \mathbf{l}'')$ . Como la matriz  $\mathbf{M}$  tiene rango 2, existe una dependencia lineal entre las columnas  $\mathbf{m}_i$ .

$$\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3] = \begin{bmatrix} 1 & \mathbf{A}^T \mathbf{l}' & \mathbf{B}^T \mathbf{l}'' \\ 0 & \mathbf{a}_4^T \mathbf{l}' & \mathbf{b}_4^T \mathbf{l}'' \end{bmatrix}. \quad (3.11)$$

La relación lineal se puede escribir como  $\mathbf{m}_1 = \alpha \mathbf{m}_2 + \beta \mathbf{m}_3$ . Como el elemento inferior izquierdo de la matriz  $\mathbf{M}$  es cero, se tiene que  $\alpha = k \mathbf{b}_4^T \mathbf{l}''$  y  $\beta = -k \mathbf{a}_4^T \mathbf{l}'$  para algún escalar  $k$ . Aplicando lo anterior a cada uno de los 3 vectores de la parte superior de cada columna se tiene:

$$\mathbf{l} = (\mathbf{b}_4^T \mathbf{l}'') (\mathbf{A}^T \mathbf{l}') - (\mathbf{a}_4^T \mathbf{l}') (\mathbf{B}^T \mathbf{l}'') = \mathbf{l}''^T (\mathbf{b}_4) (\mathbf{A}^T \mathbf{l}') - \mathbf{l}'^T (\mathbf{a}_4) (\mathbf{B}^T \mathbf{l}''). \quad (3.12)$$

Por lo tanto, la  $i$ -ésima coordenada  $l_i$  de  $\mathbf{l}$  se puede escribir como:

$$l_i = \mathbf{l}''^T (\mathbf{b}_4 \mathbf{a}_i^T) \mathbf{l}' - \mathbf{l}'^T (\mathbf{a}_4 \mathbf{b}_i^T) \mathbf{l}'' = \mathbf{l}'^T (\mathbf{a}_i \mathbf{b}_4^T) \mathbf{l}'' - \mathbf{l}''^T (\mathbf{a}_4 \mathbf{b}_i^T) \mathbf{l}'. \quad (3.13)$$

Introduciendo la notación:

$$\mathbf{T}_i = \mathbf{a}_i \mathbf{b}_4^T - \mathbf{a}_4^T \mathbf{b}_i^T, \quad (3.14)$$

la relación de incidencia entre las líneas se puede escribir como:

$$l_i = \mathbf{l}'^T \mathbf{T}_i \mathbf{l}''. \quad (3.15)$$

El tensor trifocal en notación matricial está constituido por el conjunto de matrices  $[\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3]$ . Denotando las tres matrices  $[\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3]$  de una manera más breve  $[\mathbf{T}_i]$ , la última relación se puede expresar como:

$$\mathbf{l}^T = \mathbf{l}'^T [\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3] \mathbf{l}'', \quad (3.16)$$

donde  $\mathbf{l}^T = \mathbf{l}'^T [\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3] \mathbf{l}''$  representa al vector  $[\mathbf{l}'^T \mathbf{T}_1 \mathbf{l}'', \mathbf{l}'^T \mathbf{T}_2 \mathbf{l}'', \mathbf{l}'^T \mathbf{T}_3 \mathbf{l}'']$ , y por analogía se tiene una relación similar para  $\mathbf{l}'^T = \mathbf{l}^T [\mathbf{T}_i'] \mathbf{l}''$  y  $\mathbf{l}''^T = \mathbf{l}^T [\mathbf{T}_i''] \mathbf{l}'$ . Los tres tensores  $[\mathbf{T}_i]$ ,  $[\mathbf{T}_i']$  y  $[\mathbf{T}_i'']$  existen pero son distintos entre sí. De echo aunque los tres tensores pueden ser calculados a partir de cualquiera de ellos, no existe una relación sencilla y directa entre cada uno. Por lo tanto, existen tres tensores trifocales diferentes para cada tripleta de vistas de una misma escena.

### 3.3.2. Notación tensorial

Puntos y líneas en las imágenes son representados en coordenadas homogéneas mediante un vector columna de  $3 \times 1$  y un vector fila de  $1 \times 3$  respectivamente. Por ejemplo  $\mathbf{x} = (\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3)^T$  y  $\mathbf{l} = (\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3)$ . De manera similar, la *ji-ésima* entrada de una matriz  $A$  se denota como  $\mathbf{a}_j^i$ , donde  $i$  representa el índice contravariante (fila), y  $j$  el índice covariante (columna). Podemos apreciar que el hecho de que se repita un mismo número en la posición del índice covariante y contravariante en dos términos consecutivos representa una sumatoria en el rango 1...3. En notación tensorial la ecuación  $\mathbf{x}' = \mathbf{A}\mathbf{x}$ , es equivalente a la sumatoria  $\mathbf{x}'^i = \sum_j \mathbf{a}_j^i \mathbf{x}^j$ , lo cual se puede escribir como  $\mathbf{x}'^i = \mathbf{a}_j^i \mathbf{x}^j$ .

Partiendo de la definición del tensor trifocal dada por la ecuación 3.14, y utilizando la notación tensorial se obtiene:

$$\mathbf{T}_i^{jk} = \mathbf{a}_4^j \mathbf{b}_4^k - \mathbf{a}_4^k \mathbf{b}_4^j. \quad (3.17)$$

La posición de los índices en  $\mathbf{T}_i^{jk}$  (2 contravariantes y un covariante) están determinados por la posición de los índices en el lado derecho de la ecuación. En notación tensorial, la relación de incidencia expresada en la ecuación 3.16 se puede denotar como:

$$\mathbf{l}_i = \mathbf{l}'_j \mathbf{l}''_k \mathbf{T}_i^{jk}. \quad (3.18)$$

Se puede notar que al multiplicar tensores no importa el orden de las entradas, a diferencia del caso de la notación matricial. Por ejemplo el lado derecho de la ecuación 3.18 es igual a:

$$\mathbf{l}'_j \mathbf{l}''_k \mathbf{T}_i^{jk} = \sum_{j,k} \mathbf{l}'_j \mathbf{l}''_k \mathbf{T}_i^{jk} = \sum_{j,k} \mathbf{l}'_j \mathbf{T}_i^{jk} \mathbf{l}''_k = \mathbf{l}'_j \mathbf{T}_i^{jk} \mathbf{l}''_k. \quad (3.19)$$

Un par de tensores genéricos particularmente importantes son  $\epsilon_{ijk}$  y su contraparte  $\epsilon^{ijk}$ . Este tensor es utilizado para representar el producto vectorial. Por ejemplo la línea que une dos puntos  $\mathbf{x}^i$  y  $\mathbf{y}^j$  es igual al producto cruz  $\mathbf{x}^i \mathbf{y}^j \epsilon_{ijk}$ , y la matriz antisimétrica  $[\mathbf{x}]_{\times}$  se escribe como  $\mathbf{x}^i \epsilon_{irs}$  en notación tensorial. Con lo anterior resulta sencillo entender las expresiones que envuelven las relaciones de incidencia del tensor trifocal.

Correspondencia (línea-línea-línea):

$$\mathbf{l}_r (\epsilon^{ris}) \mathbf{l}'_j \mathbf{l}''_k \mathbf{T}_i^{jk} = 0^s. \quad (3.20)$$

Correspondencia (punto-punto-punto):

$$\mathbf{x}^i (\mathbf{x}'^j \epsilon_{jpr}) (\mathbf{x}''^k \epsilon_{kqs}) \mathbf{T}_i^{pq} = 0_{rs}. \quad (3.21)$$

Las formas de las relaciones establecidas por el tensor trifocal entre diferentes características visuales, se pueden entender con mayor facilidad si se observa que los tres índices  $i$ ,  $j$ , y  $k$ , en  $\mathbf{T}_i^{jk}$ , corresponden a entidades en la primera, segunda y tercera vista respectivamente.

Existen otras relaciones de incidencia dadas por el tensor trifocal, sin embargo, la correspondencia punto-punto-punto es la de interés para el presente trabajo. Cabe mencionar que se ha preferido el uso de puntos de interés en lugar de líneas en esta tesis debido a la robustez con la que éstos pueden ser detectados. Como se describirá en la sección 3.3, existen detectores-descriptores de puntos de interés que permiten establecer correspondencias de puntos de manera eficiente [17].

### 3.3.3. Tensor trifocal a partir de las posiciones de cámara

Aunque el tensor trifocal puede ser obtenido a partir de correspondencias (de puntos o líneas) entre tres vistas como se describió en la sección anterior, aquí se presenta un modo de obtener las expresiones geométricas del tensor a partir de las posiciones de cámara, para un caso en particular donde el movimiento

relativo entre cámaras se restringe a un plano (caso común en robótica).

Un punto en el espacio 3D se puede representar como:

$$\mathbf{X} = (X, Y, Z, 1)^T \quad (3.22)$$

Por otro lado, los vectores que describen la proyección del punto en tres imágenes tomadas desde diferentes ubicaciones, pueden ser representados utilizando coordenadas homogéneas de la siguiente manera:

$$\mathbf{u} = (u^1, u^2, u^3)^T, \mathbf{v} = (v^1, v^2, v^3)^T, \mathbf{w} = (w^1, w^2, w^3)^T. \quad (3.23)$$

Partiendo de que las 3 imágenes son tomadas con la misma cámara calibrada, y utilizando el modelo pinhole de cámara, la proyección del punto 3D en cada una de las imágenes está dada por:

$$\lambda_1 \mathbf{u} = \mathbf{P}_1 \mathbf{X}, \lambda_2 \mathbf{v} = \mathbf{P}_2 \mathbf{X}, \lambda_3 \mathbf{w} = \mathbf{P}_3 \mathbf{X}, \quad (3.24)$$

donde  $\lambda_1, \lambda_2$  y  $\lambda_3$  son factores de escala,  $\mathbf{P}_1, \mathbf{P}_2$  y  $\mathbf{P}_3$  son las matrices de proyección de cámara las cuales en un sistema de referencia común se pueden representar como:

$$\mathbf{P}_1 = [\mathbf{R}_1 | \mathbf{t}_1], \mathbf{P}_2 = [\mathbf{R}_2 | \mathbf{t}_2], \mathbf{P}_3 = [\mathbf{R}_3 | \mathbf{t}_3]. \quad (3.25)$$

Considerando un sistema de referencia global con origen en el centro óptico de la tercera posición tal como se muestra en la Figura 3.3, las posiciones de cámara están dadas por  $\mathbf{C}_1 = (x_1, y_1, z_1)$ ,  $\mathbf{C}_2 = (x_2, y_2, z_2)$  y  $\mathbf{C}_3 = (x_3, y_3, z_3)$ , cada una con sus respectivas orientaciones  $\phi_1, \phi_2$  y  $\phi_3$ , donde este último término vale cero, al haber definido el sistema de referencia con respecto a la tercera cámara.

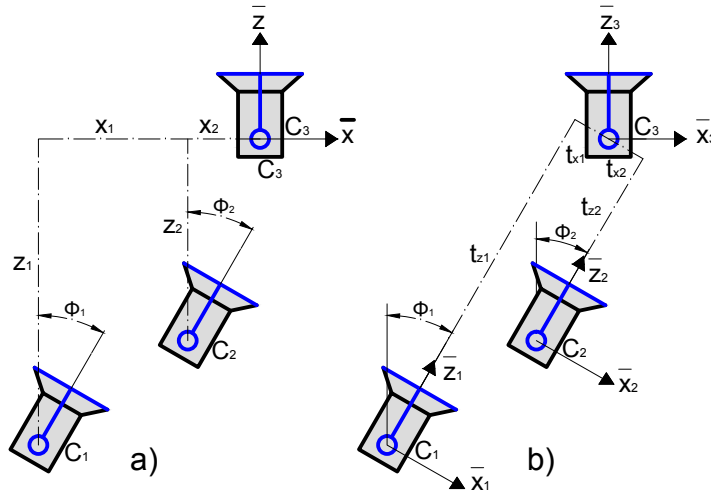


Figura 3.3: Geometría entre tres ubicaciones de cámaras sobre un plano; a) con respecto a un sistema de referencia global; b) con respecto a un sistema de referencia local.

Las matrices de rotación de cámara son:

$$\mathbf{R}_1 = \begin{bmatrix} \cos(\phi_1) & 0 & \sin(\phi_1) \\ 0 & 1 & 0 \\ -\sin(\phi_1) & 0 & \cos(\phi_1) \end{bmatrix}, \quad (3.26)$$

$$\mathbf{R}_2 = \begin{bmatrix} \cos(\phi_2) & 0 & \sin(\phi_2) \\ 0 & 1 & 0 \\ -\sin(\phi_2) & 0 & \cos(\phi_2) \end{bmatrix}, \quad (3.27)$$

$$\mathbf{R}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.28)$$

El desplazamiento de cámaras con respecto a un sistema de referencia local se puede expresar como:

$$\mathbf{t}_1 = \begin{pmatrix} t_{x1} \\ t_{y1} \\ t_{z1} \end{pmatrix} = -\mathbf{R}_1 \mathbf{C}_1 = \begin{pmatrix} -x_1 \cos(\phi_1) - z_1 \sin(\phi_1) \\ 0 \\ x_1 \sin(\phi_1) - z_1 \cos(\phi_1) \end{pmatrix}, \quad (3.29)$$

$$\mathbf{t}_2 = \begin{pmatrix} t_{x2} \\ t_{y2} \\ t_{z2} \end{pmatrix} = -\mathbf{R}_2 \mathbf{C}_2 = \begin{pmatrix} -x_2 \cos(\phi_2) - z_2 \sin(\phi_2) \\ 0 \\ x_2 \sin(\phi_2) - z_2 \cos(\phi_2) \end{pmatrix}, \quad (3.30)$$

$$\mathbf{t}_3 = \begin{pmatrix} t_{x3} \\ t_{y3} \\ t_{z3} \end{pmatrix} = -\mathbf{R}_3 \mathbf{C}_3 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \quad (3.31)$$

Con lo anterior es posible representar la expresión de la ecuación 3.24 en forma matricial:

$$\mathbf{A} \begin{pmatrix} \mathbf{X} \\ -\lambda_1 \\ -\lambda_3 \\ -\lambda_3 \end{pmatrix} = 0, \quad (3.32)$$

donde:

$$\mathbf{A} = \begin{pmatrix} \mathbf{P}_1 & \mathbf{u} & 0 & 0 \\ \mathbf{P}_2 & 0 & \mathbf{v} & 0 \\ \mathbf{P}_3 & 0 & 0 & \mathbf{w} \end{pmatrix}, \quad (3.33)$$

$$\mathbf{A} = \begin{pmatrix} \cos(\phi_1) & 0 & \sin(\phi_1) & t_{x1} & u^1 & 0 & 0 \\ 0 & 1 & 0 & t_{y1} & u^2 & 0 & 0 \\ -\sin(\phi_1) & 0 & \cos(\phi_1) & t_{z1} & u^3 & 0 & 0 \\ \cos(\phi_2) & 0 & \sin(\phi_2) & t_{x2} & 0 & v^1 & 0 \\ 0 & 1 & 0 & t_{y2} & 0 & v^2 & 0 \\ -\sin(\phi_2) & 0 & \cos(\phi_2) & t_{z2} & 0 & v^3 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & w^1 \\ 0 & 1 & 0 & 0 & 0 & 0 & w^2 \\ 0 & 0 & 1 & 0 & 0 & 0 & w^3 \end{pmatrix}. \quad (3.34)$$

La relación establecida por la ecuación previa debe prevalecer para cada uno de los puntos de la escena y por lo tanto la matriz  $\mathbf{A}$  es de rango 6. De lo anterior resulta evidente que cada menor de  $7 \times 7$  de la matriz tiene determinante igual a cero, lo que establece las relaciones trilineales del tensor trifocal. Dependiendo de las filas seleccionadas se pueden formar diferentes determinantes. A partir de estos determinantes el tensor trifocal en notación tensorial se define como en la ecuación 3.17:

$$\mathbf{T}_i^{jk} = \mathbf{a}_i^j \mathbf{b}_4^k - \mathbf{a}_4^j \mathbf{b}_i^k.$$

De la ecuación 3.17 es posible obtener cada uno de los elementos del tensor trifocal, partiendo de que cada uno de los elementos de las matrices de cámara se denotan como  $P_1 = [a_i^j]$  y  $P_2 = [b_i^j]$ , donde los sub-índices corresponden a las columnas y los super-índices a las filas. Considerando el caso en el que el movimiento se restringe a un plano, de los 27 elementos del tensor únicamente 12 tienen valores no nulos, y se obtienen como se muestran a continuación. En lo siguiente, la notación empleada en este caso será

$\mathbf{T}_{ijk}$  en lugar de  $\mathbf{T}_i^{jk}$ , para diferenciar entre los elementos obtenidos a partir de información métrica y los obtenidos a partir de información visual.

Tenemos:

$$\begin{aligned}
T_{111} &= -t_{x1} \cos(\phi_2) + t_{x2} \cos(\phi_1) \\
T_{113} &= t_{x1} \sin(\phi_2) + t_{z2} \cos(\phi_1) \\
T_{131} &= -t_{z1} \cos(\phi_2) - t_{x2} \sin(\phi_1) \\
T_{133} &= t_{z1} \sin(\phi_2) - t_{z2} \sin(\phi_1) \\
T_{212} &= -t_{x1} \\
T_{221} &= t_{x2} \\
T_{223} &= t_{z2} \\
T_{232} &= -t_{z1} \\
T_{311} &= -t_{x1} \sin(\phi_2) + t_{x2} \sin(\phi_1) \\
T_{313} &= -t_{x1} \cos(\phi_2) + t_{z2} \sin(\phi_1) \\
T_{331} &= -t_{z1} \sin(\phi_2) + t_{x2} \cos(\phi_1) \\
T_{333} &= -t_{z1} \cos(\phi_2) + t_{z2} \cos(\phi_1),
\end{aligned} \tag{3.35}$$

donde  $t_{xi} = -x_i \cos(\phi_i) - z_i \sin(\phi_i)$  y  $t_{zi} = x_i \sin(\phi_i) - z_i \cos(\phi_i)$  para  $i = 1, 2$ . El tensor trifocal que se estima tiene un factor de escala desconocido, por lo cual en el caso de un control visual para robots móviles basado en el tensor trifocal, donde el factor de escala está cambiando constantemente conforme el robot se desplaza, es necesario normalizar cada uno de los elementos del tensor de la siguiente manera:

$$\mathbf{T}_{ijk} = \mathbf{T}_{ijk}^e / \mathbf{T}_N, \tag{3.36}$$

donde  $\mathbf{T}_{ijk}^e$  representa lo elementos del tensor estimados con las coordenadas de los puntos emparejados en las imágenes,  $\mathbf{T}_{ijk}$  los elementos normalizados del Tensor, y  $\mathbf{T}_N$  un factor de normalización. Observando la ecuación 3.35, se puede notar que los elementos  $\mathbf{T}_{313}$  y  $\mathbf{T}_{323}$  son constantes y diferentes de cero, siempre y cuando  $\mathbf{C}_1$  sea distinto de  $\mathbf{C}_3$ , lo cual siempre ocurrirá en los casos de interés, puesto que se desea que la posición inicial y final del robot sean distintas; por lo tanto estos elementos son buenos candidatos para el factor de normalización.

La forma general del tensor trifocal métrico cuando el movimiento se restringe a un plano es:

$$\mathbf{T}_1 = \begin{bmatrix} T_{111} & 0 & T_{113} \\ 0 & 0 & 0 \\ T_{131} & 0 & T_{133} \end{bmatrix} \quad \mathbf{T}_2 = \begin{bmatrix} 0 & T_{212} & 0 \\ T_{221} & 0 & T_{223} \\ 0 & T_{232} & 0 \end{bmatrix} \quad \mathbf{T}_3 = \begin{bmatrix} T_{311} & 0 & T_{313} \\ 0 & 0 & 0 \\ T_{331} & 0 & T_{333} \end{bmatrix}. \tag{3.37}$$

### 3.4. Descriptores y detectores

Con la finalidad de poder extraer información de una imagen, es necesario detectar partes importantes en ella, a las cuales se les denomina características visuales. En esta tesis se ha elegido el utilizar puntos de interés, porque son características visuales que abundan en las escenas.

Para la detección de puntos de interés en las imágenes, existen algoritmos denominados detectores. Una vez que se dispone de un conjunto puntos de interés en las imágenes el problema ahora es como poder relacionar puntos correspondientes en diferentes imágenes de una misma escena; para ello existen los descriptores, los cuales son algoritmos que buscan caracterizar por medio de un vector la información necesaria para definir y diferenciar a un punto de interés, asignando un número de identidad único a cada punto.

A continuación se presenta una breve descripción de los descriptores y detectores más comunes.

**SURF (Speeded Up Robust Features):** Es un algoritmo que permite detectar puntos de manera robusta ante las variaciones de escala ocasionada por las diferencias de profundidad de un mismo objeto con respecto a diferentes posiciones de cámara, y ante las rotaciones [4]. El algoritmo se realiza en cuatro etapas: 1. Cálculo del determinante del Hessiano: En esta etapa, se realiza un cálculo de una cantidad que permitirá detectar puntos de interés que sean invariantes a los cambios de rotación y de escala. SURF utiliza el valor del determinante de la matriz Hessiana para el cálculo de la posición y la escala de los puntos de interés. 2. Localización de puntos clave: La localización de los puntos de interés se obtiene mediante la eliminación de los puntos que no cumplan con una condición de máximo en un vecindario del punto. 3. Asignación de orientación: Se asigna una orientación a cada punto de interés de acuerdo a la respuesta obtenida por los puntos de interés, a filtros de Haar. De esta forma se obtiene invarianza a la rotación. 4. Descriptor del punto de interés: En esta etapa, los puntos de interés son representados mediante vectores descriptores como una medida de las respuestas a filtros de Haar en las direcciones horizontal y vertical, respectivamente, relativas a la orientación del punto de interés.

**SIFT (Scale Invariant Feature Transform):** es un algoritmo que detecta puntos distintivos de imágenes y calcula el descriptor de los mismos [24]. Los puntos extraídos son invariantes a la escala, rotación, y parcialmente invariantes a cambios de iluminación. Las características SIFT se forman calculando el gradiente en cada píxel en una ventana de  $16 \times 16$  alrededor del punto de interés, y utilizando el nivel apropiado de la pirámide gaussiana (escala) en el cual fue detectado el punto. El plano escalado se divide en cuadrantes de  $4 \times 4$  para los cuales se forma un histograma de las orientaciones del gradiente. El vector del descriptor se forma con las entradas de estos histogramas. Una de las principales diferencias entre SURF y SIFT es la velocidad de computo. Como el descriptor SURF se basa en las variaciones de intensidad es menor el tiempo de computo. Sin embargo, el descriptor SIFT es considerado de manera general más eficiente para encontrar correspondencias correctas entre puntos de interés de diferentes vistas de una misma escena (matching).

**Harris:** es un detector de puntos de interés que permite extraer las esquinas presentes en una imagen, las cuales en general deben abundar en una escena con objetos hechos por el hombre (paredes, puertas, mesas, etc.); además las esquinas son características visuales bidimensionales que se pueden localizar con precisión ya que se presentan en la unión de dos bordes. Éste es uno de los detectores de puntos que se utiliza con mayor frecuencia, debido a su elevada invarianza ante la escala, la rotación, los cambios de iluminación y el ruido en la imagen [12]. Se basa en la siguiente matriz  $\mathbf{C}(x, y)$ , que se calcula sobre una sub-ventana para cada punto de interés en la posición  $(x, y)$ .

$$\mathbf{C}(x, y) = \begin{pmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{pmatrix}, \quad (3.38)$$

donde  $I_x$  e  $I_y$  son los gradientes de la imagen en dirección horizontal y vertical respectivamente. Siendo  $\lambda_1$  y  $\lambda_2$  los valores propios de la matriz  $\mathbf{C}(x, y)$ , la función de auto correlación  $R$  tendrá un pico si ambos valores son altos. Esto significa que un desplazamiento en cualquier dirección producirán un incremento significativo, indicando que se trata de una esquina.

$$R = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2). \quad (3.39)$$

Existen otros detectores y descriptores que no fueron utilizados en esta tesis como ORB [32], FAST [31], BRIEF [7], FREAK, etc. En la práctica es posible utilizar cualquier combinación de detectores y descriptores.



# Capítulo 4

## Metodología

Este capítulo está dedicado a la explicación y análisis del procedimiento seguido para la implementación de los esquemas de control visual que se presentan en esta tesis. Se abordan los temas de la extracción y selección de las características visuales a partir de las imágenes, calibración de cámara, la estimación del tensor trifocal y la implementación de los controladores.

Ambos esquemas de control, los cuales se presentan en los capítulos 5 y 6 se basan en las restricciones geométricas inducidas por el tensor trifocal, de manera que el procedimiento para poder implementarlos es bastante similar diferenciándose básicamente en las leyes de control que los rigen. Por lo tanto la metodología se presenta de manera general, y en los próximos capítulos se muestran de manera independiente cada uno de los esquemas de control, y los resultados obtenidos al implementarlos.

### 4.1. Extracción de las características visuales

En el capítulo 3 se explicó que existen diferentes métodos para estimar los elementos del tensor trifocal, ya sea a partir de características visuales extraídas de un conjunto de imágenes, o de las posiciones relativas de cámara. La razón por la que en éste caso es necesario trabajar con el método basado en características visuales, es debido a que durante la navegación del robot no es posible extraer información métrica correspondiente a la posición relativa del sistema (robot-cámara), utilizando únicamente una cámara monocular como sensor.

La parte concerniente a la extracción de información visual de las imágenes se realiza con ayuda de OpenCV [29], el cual es una librería con varias funciones enfocadas principalmente a la visión por computadora en tiempo real.

El primer paso para la implementación del control consiste en la extracción de características visuales a partir de las imágenes, para ello se utilizaron diferentes descriptores como son SIFT, SURF, y el detector de Harris, ya que nos permiten seleccionar puntos. SIFT se utiliza por proporcionar puntos de interés invariantes a la rotación y escala, y parcialmente invariantes a cambios de iluminación. Los puntos SURF por ser invariantes a la escala y a la rotación. Y los puntos de Harris debido a su elevada invarianza ante la escala, rotación, cambios de iluminación, y ruido en la imagen.

Después de que se dispone de un conjunto de puntos de interés en cada una de las imágenes, es necesario encontrar aquellos que están presentes en las tres vistas; Para ello se realiza un emparejamiento (matching) entre la primera y la segunda imagen, y entre la segunda y la tercera, dejando solamente el conjunto de puntos que comparte la segunda vista con las otras dos.

Para garantizar que los puntos emparejados realmente corresponden a la misma característica visual en cada una de las imágenes, es conveniente utilizar desde esta etapa un algoritmo que mejore la robustez del

proceso de emparejamiento como es RANSAC (Random Sample Consensus). En esta etapa el algoritmo RANSAC se utiliza para emparejar los puntos entre cada par de imágenes, utilizando las restricciones geométricas inducidas por la geometría epipolar.

Con las características visuales proporcionadas por los diferentes descriptores y detectores utilizados, se realizaron pruebas con la finalidad de seleccionar el tipo de puntos de interés que permita la estimación de un tensor trifocal estable, y que los puntos se puedan rastrear a lo largo de la trayectoria a seguir por el robot.

## 4.2. Calibración de cámara.

Para estimar el tensor trifocal se forma un sistema de ecuaciones con 27 incógnitas. En dicho sistema de ecuaciones los coeficientes de algunas de las variables dependen directamente de las coordenadas de los puntos involucrados en los cálculos, y el resto de los coeficientes se fijan a un valor unitario como se puede apreciar a partir de la ecuación 4.10. De modo que al trabajar con una imagen con dimensiones considerablemente grandes en comparación con la unidad (por ejemplo de  $640 \times 480$  píxeles), se forma un sistema de ecuaciones mal condicionado al tener coeficientes muy grandes y muy pequeños. Al solucionar el sistema de ecuaciones mal condicionados se obtendría un TT estimado que no es confiable para el cálculo de las velocidades del robot.

Para evitar formar un sistema mal condicionado al estimar el TT se tienen dos opciones, la primera es una normalización, lo cual consiste en trasladar los puntos de la imagen de tal forma que el centroide esté en el origen, y escalarlos de manera que la distancia media al origen sea  $\sqrt{2}$  [20]. La segunda opción consiste en calibrar la cámara para obtener los parámetros intrínsecos y obtener la inversa de la matriz de cámara, con la finalidad de pasar los puntos del plano de la imagen (con coordenadas en píxeles), al plano normalizado (con coordenadas en unidades métricas).

La razón por la cual se prefiere utilizar el método basado en la calibración de cámara, es porque permite obtener los parámetros de distorsión radial y tangencial, ya que en caso de que éstos no sean muy pequeños se tienen que considerar para obtener las coordenadas de los puntos en ausencia de distorsión, y así poder estimar un TT estable. De lo contrario, al utilizar el método de la normalización, si la cámara tiene valores grandes de distorsión afectarán considerablemente en el cálculo del TT, ya que se trata de parámetros no lineales.

La calibración de cámara se realizó con ayuda de OpenCV y un tablero clásico de ajedrez en blanco y negro. Para la obtención de los parámetros de cámara se debe formar un sistema de ecuaciones, donde en teoría basta con 2 fotos para poder darle solución, sin embargo en la práctica las imágenes capturadas presentan ruido, por lo tanto para obtener buenos resultados son necesarias mínimo 10 fotografías. El procedimiento básicamente consiste en tomar fotografías del tablero en diferentes posiciones y orientaciones, en donde cada una de las imágenes representa una nueva ecuación. La Figura 4.1 ejemplifica esta idea donde se muestra la posición y orientación de diferentes planos de imagen con respecto a la ubicación de la cámara.

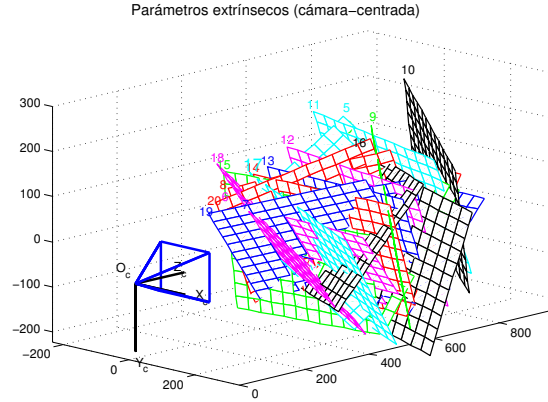


Figura 4.1: Imágenes tomadas a diferentes distancias y orientaciones con respecto a un marco de referencia asociado al centro óptico de la cámara, para formar un sistema de ecuaciones con la finalidad de obtener los parámetros intrínsecos de la cámara.

Como resultado de la calibración de cámara, se obtiene la matriz de cámara y los coeficientes de distorsión, tal como se muestra en las siguientes 2 expresiones:

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (4.1)$$

$$\text{Coeficientes}_{distorsion} = [k_1, k_2, p_1, p_2, k_3], \quad (4.2)$$

donde  $f_x$  y  $f_y$  son las longitudes focales de cámara;  $c_x$  y  $c_y$  representan la ubicación del punto principal en la imagen (en coordenadas en píxeles);  $k_1$ ,  $k_2$  y  $k_3$  son los coeficientes de distorsión radial;  $p_1$  y  $p_2$  los coeficientes de distorsión tangencial.

La vista de una escena en ausencia de distorsión radial y tangencial, se puede formar a partir de una proyección de puntos 3D, utilizando la siguiente transformación proyectiva:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} [\mathbf{R}|\mathbf{t}] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad (4.3)$$

donde  $(u, v)$  son las coordenadas en píxeles de los puntos proyectados.  $[\mathbf{R}|\mathbf{t}]$  de dimensiones  $3 \times 4$  es la matriz de rotación-traslación, que traslada las coordenadas de puntos 3D a un sistema de coordenadas asociado a la cámara.  $(X, Y, Z)$  son las coordenadas de un punto en el espacio 3D con respecto al sistema de coordenadas mundo.

La transformación proyectiva de la ecuación 4.3 es equivalente a:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{R} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \mathbf{t}. \quad (4.4)$$

A partir de la ecuación anterior se tiene:

$$\begin{aligned} u &= f_x * x' + c_x, \\ v &= f_y * y' + c_y, \end{aligned} \quad (4.5)$$

donde  $x' = x/z$  y  $y' = y/z$ . Sin embargo, las lentes reales suelen tener distorsión. Considerando los efectos de la distorsión radial y tangencial, el modelo anterior se extiende como:

$$\begin{aligned} u &= f_x * x'' + c_x, \\ v &= f_y * y'' + c_y, \end{aligned} \quad (4.6)$$

donde:

$$\begin{aligned} r^2 &= x'^2 + y'^2, \\ x'' &= x' \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + 2p_1 x' y' + p_2 (r^2 + 2x'^2), \\ y'' &= y' \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + 2p_2 x' y' + p_1 (r^2 + 2y'^2). \end{aligned} \quad (4.7)$$

Para obtener las imágenes sin distorsión radial se utiliza la función “undistor” de Opencv. Cabe mencionar que Opencv no considera los coeficientes de distorsión de orden superior ( $k_4, k_5, k_6$ ).

### 4.3. Estimación del tensor trifocal

Con los puntos emparejados de forma robusta entre pares de imágenes (usando RANSAC con las restricciones geométricas inducidas por la geometría epipolar), y con las coordenadas en el plano normalizado, se procede a realizar una primer estimación de los elementos del TT. Las ecuaciones para el cálculo del TT al estar trabajando con puntos en las imágenes, se obtienen de la siguiente relación para cada tripleta de puntos:

$$\mathbf{x}^i \mathbf{x}'^j \mathbf{x}''^k \epsilon_{jqs} \epsilon_{krt} \mathbf{T}_i^{qr} = 0_{st}. \quad (4.8)$$

De la expresión anterior se forman un total de 9 ecuaciones al considerar todas las combinaciones posibles para  $s$  y  $t$ , pero solamente 4 de ellas son linealmente independientes, las cuales se pueden obtener eligiendo 2 valores para cada  $s$  y  $t$ , por ejemplo 1 y 2.

La relación trilineal entre puntos dada por la ecuación 4.8 para valores dados de  $s$  y  $t$  se puede expandir de la siguiente manera:

$$\mathbf{x}^k (\mathbf{x}'^i \mathbf{x}''^m \mathbf{T}_k^{jl} - \mathbf{x}'^j \mathbf{x}''^m \mathbf{T}_k^{il} - \mathbf{x}'^i \mathbf{x}''^l \mathbf{T}_k^{jm} - \mathbf{x}'^j \mathbf{x}''^l \mathbf{T}_k^{im}) = 0^{ijlm}. \quad (4.9)$$

Cuando  $i, j \neq s$  y  $l, m \neq t$ . Una forma de obtener las 4 ecuaciones linealmente independientes es fijar  $j = m = 3$ , y dejar que  $i$  y  $l$  tomen valores libremente entre 1 y 3.  $\mathbf{x}^3$ ,  $\mathbf{x}'^3$  y  $\mathbf{x}''^3$  se deben fijar a un valor de 1 para obtener una relación entre las coordenadas de las imágenes observadas. Con lo anterior la ecuación 4.9 se convierte en:

$$\mathbf{x}^k (\mathbf{x}'^i \mathbf{x}''^l \mathbf{T}_k^{33} - \mathbf{x}''^l \mathbf{T}_k^{i3} - \mathbf{x}'^i \mathbf{T}_k^{3l} + \mathbf{T}_k^{il}) = 0. \quad (4.10)$$

Las posibles opciones para  $i, l = 1, 2$ , dan 4 ecuaciones linealmente independientes en término de las coordenadas de los puntos en las imágenes. El tensor trifocal como se mencionó anteriormente está compuesto por 27 elementos, por lo cual se requiere de un mínimo de 7 tripletas de puntos correspondientes para formar una matriz de ecuaciones  $\mathbf{A}\mathbf{t} = 0$  con al menos 27 filas, y así poder estimar cada uno de los elementos del TT.

Para resolver  $\mathbf{A}\mathbf{t} = 0$ , se utiliza una solución por mínimos cuadrados para un sistema homogéneo de ecuaciones lineales. Considerando que la matriz  $\mathbf{A}$  debe tener al menos tantas filas como columnas, el problema consiste en encontrar un vector  $\mathbf{t}$  que minimice  $|\mathbf{A}\mathbf{t}|$  sujeto a  $|\mathbf{t}| = 1$ .

Una posible solución para encontrar  $\mathbf{t}$  es por medio de una descomposición en valores singulares (SVD), donde la matriz  $\mathbf{A}$  se expresa como  $\mathbf{A} = \mathbf{UDV}^T$ . De lo anterior  $\mathbf{t}$  se obtiene como la última columna de  $V$ .

Una vez formado y resuelto el sistema de ecuaciones lineales se obtiene el tensor trifocal, sin embargo los cálculos se realizan para este caso en particular utilizando únicamente 7 tripletas de puntos, bajo un esquema en la cual se pueden tener muchos puntos de interés. El objetivo en ésta etapa consiste en obtener el tensor trifocal que mejor defina el conjunto características visuales disponibles, utilizando únicamente 7 tripletas como se menciono anteriormente, para ello se vuelve a implementar RANSAC.

Para implementar RANSAC utilizando las restricciones geométricas del tensor trifocal se seleccionan 7 tripletas de puntos al azar y se estima el tensor. Conociendo las coordenadas de un punto correspondiente en las primeras dos imágenes, y utilizando los valores calculados para el TT es posible estimar la ubicación de ese mismo punto en la tercera vista tal como se describe en la sección 4.3.1. Como las coordenadas del punto correspondiente en la tercera imagen se conocen previamente, es posible evaluar si el punto transferido se encuentra dentro de un margen de error aceptable considerándose como un valor típico (inlier) del modelo establecido, o si cae fuera del margen de error considerándose como un valor atípico (outlier).

La transferencia de características visuales se lleva a cabo para cada uno de los puntos de interés detectados en las imágenes, y se cuenta el número de valores atípicos para el modelo evaluado. Después se seleccionan otras 7 tripletas de puntos al azar para formar un nuevo modelo del sistema, y se repite el procedimiento  $N$  veces (donde  $N$  se define adaptativamente como se describe en la sección 4.3.2) para encontrar el tensor trifocal que cuente con el menor número de valores atípicos. En caso de que exista más de un tensor con el mismo número de outliers se escoge el que esté mejor condicionado.

Por mejor condicionado se entiende aquel sistema de ecuaciones que al resolverlo utilizando SVD, arroje un vector  $U$  (de valores singulares) con el último término mucho menor en comparación con el resto, lo cual depende directamente de la escena descrita por los puntos involucrados en los cálculos del TT. El hecho de que más de uno de los elementos del vector  $U$  sean prácticamente nulos significa que el sistema puede tener más de una solución, siendo posible que la solución que se considere buena no defina la escena descrita por las tres vistas, y por consiguiente los elementos del tensor trifocal estimados sean erróneos.

Otro criterio para descartar algún TT cuando se tenga el mismo número de inliers, es el valor RMS de la distancia geométrica entre puntos reales y transferidos. Sin embargo, en la práctica se observó que el tensor trifocal con menos puntos atípicos no precisamente es el que mejor describe el arreglo geométrico entre las posiciones de las cámaras. Por otro lado, cuando el sistema esta mejor condicionado implica que las características visuales son extraídas de planos con diferentes profundidades, lo cual es necesario para calcular un tensor trifocal geoméricamente valido.

Los valores atípicos para el tensor trifocal que mejor describen al conjunto de puntos se descartan, y se vuelve a calcular un nuevo tensor utilizando todas las tripletas de puntos persistentes en las imágenes después de haber aplicado RANSAC, formando un sistema de ecuaciones de  $n \times 4$ , donde  $n$  es el número de tripletas formadas.

Al momento de estimar el TT, las tres imágenes involucradas en los cálculos deben ser distintas, de lo contrario se presenta un caso degenerado donde el resultado obtenido no describe el arreglo geométrico entre las cámaras. Otro Aspecto importante que se debe considerar, es que los puntos de interés extraídos de las imágenes, deben corresponder a puntos que se encuentren a diferentes profundidades en la espacio 3D; de lo contrario no se podrá definir en forma correcta la escena, y por consiguiente no se podrá calcular un TT geoméricamente válido.

### 4.3.1. Transferencia de puntos utilizando el TT

Una manera de transferir puntos a la tercera imagen es utilizando la siguiente expresión  $\mathbf{x}''^K = \mathbf{x}^i \mathbf{l}'_j \mathbf{T}_i^{jk}$ , Donde  $\mathbf{l}'$  representa una línea que pasa por el punto  $\mathbf{x}'$  en la segunda vista. Sin embargo, para el caso cuando  $\mathbf{l}'$  es la línea epipolar correspondiente al punto  $\mathbf{x}$  se tiene que  $\mathbf{x}^i \mathbf{l}'_j \mathbf{T}_i^{jk} = 0$ , y el punto  $\mathbf{x}''$  queda indefinido. Por lo tanto la elección de la línea que pase por el punto en la segunda imagen es un factor muy importante.

Una manera para evitar seleccionar únicamente la línea epipolar es utilizar dos o 3 líneas que pasen a través de  $\mathbf{x}'$ , denotadas por  $\mathbf{l}_{jp}' = \mathbf{x}'^r \epsilon_{rjp}$  para  $p = 1, 2, 3$ . A partir de cada línea se calcula la posición de  $\mathbf{x}''$ , dejando como resultado correcto el valor que tiene la mayor norma. Otro método para calcular  $\mathbf{x}''$  consiste en obtener la solución por mínimos cuadrados para el sistema lineal de ecuaciones  $\mathbf{x}^i \mathbf{x}'^j \mathbf{x}''^k \epsilon_{jqs} \epsilon_{krt} \mathbf{T}_i^{qr} = 0_{st}$ ; sin embargo puede resultar un procedimiento tedioso.

El método que aquí se utiliza para lidiar con la elección de la línea que pasa a través del punto de interés en la segunda imagen es el siguiente: partiendo de que se aplicó RANSAC con la matriz fundamental para obtener un buen emparejamiento entre cada par de imágenes, se usa la siguiente ecuación para la transferencia de puntos

$$\mathbf{x}''^K = \hat{\mathbf{x}}^i \mathbf{l}'_j \mathbf{T}_i^{jk}, \quad (4.11)$$

donde  $\hat{\mathbf{x}}$  y  $\hat{\mathbf{x}}'$  representan los puntos exactamente emparejados en las dos primeras vistas, y una buena opción para  $\mathbf{l}'$  es la línea perpendicular a  $\mathbf{l}_e' \mathbf{F}_{21} \hat{\mathbf{x}}$ .

Si  $\mathbf{l}'_e = (\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3)^T$  y  $\hat{\mathbf{x}}' = (\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, 1)^T$ , entonces:

$$\mathbf{l}' = (\mathbf{l}_2, -\mathbf{l}_1, -\hat{\mathbf{x}}_1 \mathbf{l}_2 + \hat{\mathbf{x}}_2 \mathbf{l}_1)^T. \quad (4.12)$$

Un aspecto importante que se debe tomar en cuenta al transferir información visual utilizando la ecuación 4.11, es que solo se pueden pasar puntos a la imagen que se definió como la tercera entrada para los cálculos del tensor trifocal; debido a que como se explico en la sección 3.3.1 existen diferentes tensores que se pueden calcular a partir del mismo conjunto de datos, y los cuales no guardan una relación directa entre sí.

Para que los elementos del tensor trifocal que se estiman de información visual al aplicar la ecuación 4.10, correspondan a los que se calculan a partir de información métrica utilizando las expresiones de la ecuación 3.35, es necesario definir como primera entrada ( $\mathbf{x}$ ) la imagen que corresponde a la posición objetivo (origen del sistema de coordenadas absoluto), en este caso la tercera imagen, y como entradas 2 y 3 ( $\mathbf{x}'$  y  $\mathbf{x}''$ ), las imágenes 1 y 2 en lugar de introducirlas en orden ascendente como se podría pensar.

De lo anterior resulta evidente que al realizar la transferencia de puntos durante RANSAC introduciendo las imágenes en orden 3, 2 y 1, no existe ningún inconveniente debido a que para evaluar el modelo del sistema solo hay que considerar que se transfieren puntos a la segunda; Sin embargo al momento de tener que transferir puntos a la tercera imagen como ocurre en el caso de la imagen virtual del esquema de control del capítulo 6, el orden de las entradas para estimar el TT tendría que ser 1, 2 y 3, con la finalidad de poder transferir información visual de las primeras dos imágenes a una tercera.

### 4.3.2. Número de muestreos necesarios en RANSAC

El número de muestreos ( $N$ ) que se deben realizar al implementar RANSAC debe ser menor al número de valores atípicos del conjunto de datos, e incrementa con el número mínimo de elementos necesarios para cada muestreo.  $N$  se puede calcular a partir de la siguiente expresión:

$$N = \frac{\log(1 - p)}{\log(1 - (1 - \epsilon)^s)}. \quad (4.13)$$

Donde  $p$  representa la probabilidad de que al menos una de las muestras aleatorias este libre de valores atípicos, el cual usualmente se selecciona como  $p = 0.99$ . El tamaño del conjunto de datos se representa como  $s$ . Y  $\epsilon$  es la probabilidad de que cualquier punto del conjunto de datos sea un valor atípico.

En la mayoría de los casos se desconoce de entrada el número de valores atípicos del sistema. El algoritmo para determinar  $N$ , se puede iniciar considerando que en el peor de los casos la mitad de los datos son outliers ( $\epsilon = 0.5$ ), y refinar dicho valor conforme se ejecuta el algoritmo.

El procedimiento para obtener adaptativamente  $N$  es el siguiente, y se repite hasta que el valor encontrado sea menor al número de elementos del muestreo.

1. Se escoge una muestra de datos para definir un modelo del sistema y se cuenta el número de valores atípicos encontrados.
2. Se determina la probabilidad de que algún elemento del conjunto de datos sea un valor atípico.
3. Se calcula un valor de  $N$  a partir de la ecuación 4.13.
4. Se incrementa el tamaño del muestro.

Como el RANSAC que se implementa para el tensor trifocal utiliza puntos que previamente fueron filtrados con un RANSAC a base de las restricciones inducidas por la geometría epipolar, se parte de que la mayoría de los puntos son inliers del sistema y se fija desde el inicio un valor pequeño de  $\epsilon$ , para estimar  $N$  sin tanto costo computacional.

## 4.4. Rastreo de puntos

Como la tarea del controlador consiste en conducir el robot móvil desde una posición de partida hasta una posición final, la imagen actual vista por el robot cambia constantemente y es necesario estimar la ubicación de los puntos de interés en la imagen capturada por el robot a cada paso.

Una manera de determinar la posición de los puntos es volver a aplicar matching a cada iteración tal como se hizo con las primeras 3 imágenes. Sin embargo, este procedimiento suele ser muy costoso computacionalmente, lo cual no es conveniente ya que el cálculo de los elementos del TT y por consiguiente de las velocidades del robot ( $\nu$  y  $\omega$ ) tardaría bastante tiempo. Esto puede repercutir considerablemente en el desempeño del controlador, ocasionado en algunos caso que el robot se desvíe del objetivo, o lo pierda de vista, por lo que el robot no cumplirá con la tarea especificada.

Una mejor opción para obtener las coordenadas de los puntos en la imagen actual a cada iteración, consiste en calcular el flujo óptico de las características visuales de interés utilizando el método iterativo de Lucas-Kanade. Este método además de ser menos costoso computacionalmente que el emparejamiento entre imágenes, permite obtener una evolución más suavizada de los elementos del TT.

Inicialmente se trabajó con una secuencia de imágenes previamente capturadas para garantizar que el rastreo de puntos se efectuara con éxito. Después de realizar varias pruebas se determinó experimentalmente que es mejor utilizar la posición de los puntos en la imagen previa para el rastreo de puntos a cada paso, en lugar de siempre usar la ubicación de las características visuales en la imagen correspondiente a la posición inicial. Esto con la finalidad de evitar la deriva de puntos en la imagen, lo cual impide calcular un tensor trifocal estable.

Como los puntos a rastrear son los que comparten las primeras tres imágenes, en ocasiones algunos de ellos se salen del campo de vista durante la navegación del robot móvil. Sin embargo, el método iterativo de Lucas-Kanade los deja en los bordes de la imagen. Esto provocando que el TT calculado a partir de este punto sea inconsistente, ya que se utilizan coordenadas falsas en su estimación. Para solucionar el problema mencionado es necesario descartar estos puntos de los cálculos del tensor trifocal, eliminando en cada paso aquellos puntos que se encuentran muy próximos a los bordes de la imagen.

## 4.5. Implementación

Un vez estimado el TT y resuelto el rastreo de puntos, el paso restante consiste en introducir las leyes de control para cada uno de los esquemas, dentro del cuerpo del programa y proceder a implementarlos.

Para analizar el desempeño de los esquemas de control se realizan pruebas utilizando un simulador que incluye la dinámica del robot (Webots), pruebas con imágenes reales capturadas desde una cámara montada sobre robot Pioneer P3-DX, y simulaciones en MATLAB.

El robot Pioneer P3-DX es un robot móvil diferencial, protegido con una estructura de aluminio de 16 mm de espesor, cuenta con ruedas de 19 cm de radio y con 8 sensores ultrasónicos en la parte frontal. La velocidad de traslación máxima alcanzada por el robot es de 1.6 m/s, y la velocidad de rotación máxima alcanzada es de 300°/s. La carga máxima de operación es 17 Kg. El tiempo máximo de operación es de 10 horas continuas. El robot dispone de 32 entradas digitales, 8 salidas digitales, 7 entradas analógicas, 3 puertos seriales, y entrada ethernet.

El simulador sobre el cual se realizan las pruebas (Webots) es un entorno de desarrollo utilizado para modelar, programar y simular robots móviles. En el simulador webots el comportamiento del robot se puede probar en ambientes físicamente realistas, y además es posible utilizar el mismo código que se usaría en el caso de un robot real, permitiendo evaluar el desempeño del algoritmo de control.

Con la finalidad de poder trabajar con un robot móvil real, primero fue necesario familiarizarse con la forma de programación de los robots tipo Pioneer. Una manera de poder programar esta clase de robots es utilizando las librerías de ARIA (Advanced Robot Interface for Applications) en C++ [3], las cuales permiten controlar de manera dinámica las velocidades, posición y orientación del robot, ya sea a través de comandos simples o a través de su infraestructura de acciones de alto nivel. Antes de pasar a realizar pruebas directamente con el robot real, se utilizó otro software de simulación para robots móviles llamado "MobileSim", el cual permite experimentar con la forma de programación de Aria. En este simulador es posible programar algunas rutinas y probar que realmente el comportamiento del robot sea el esperado, con la finalidad de evitar incidentes al momento de pasar el código a un robot real. MobileSim únicamente se utilizó para familiarizarse con la programación de los robots móviles.

En la Figura 4.2, se muestra el robot real con el que se realizaron pruebas, y el robot con el que se trabajó en las simulaciones.



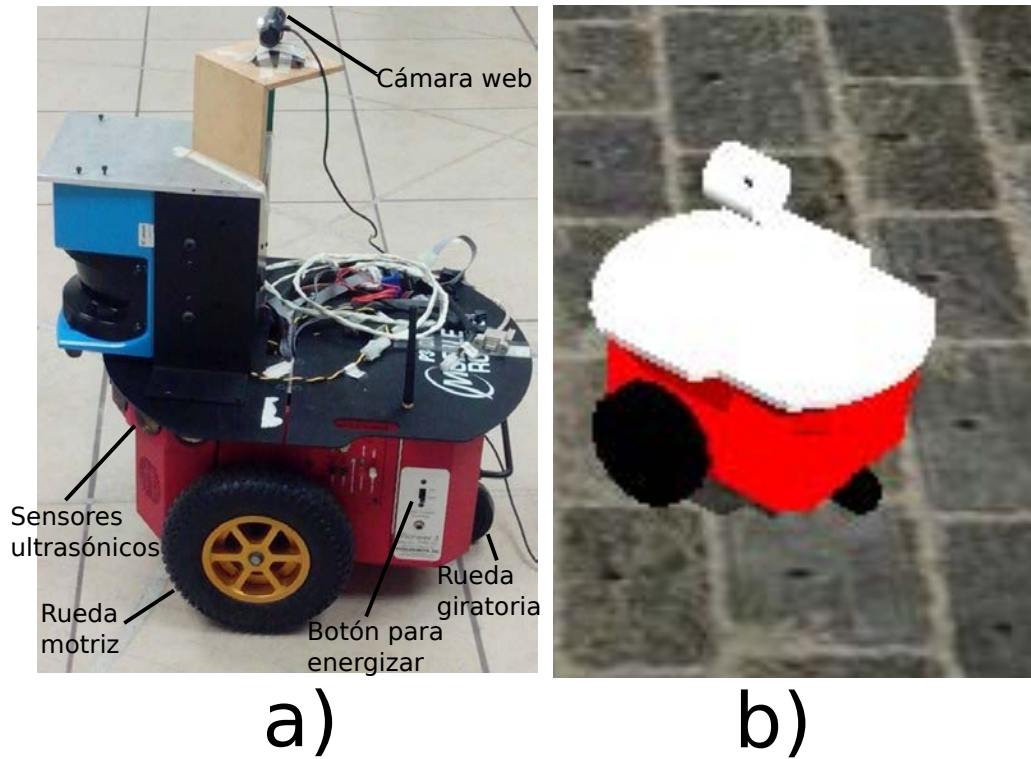


Figura 4.2: Robots móviles. a) Robot real. Robot simulado en Webots.

Como el disco duro del robot móvil utilizado es pequeño, y el procesamiento de las imágenes es costoso computacionalmente, se ejecuta el programa del controlador desde una computadora portátil montada sobre el robot, y se establece una comunicación TCP a través de la cual se pasan los comandos necesarios para el desplazamiento del robot. Para poder establecer la comunicación entre el robot y la computadora se utiliza ArNetworking, el cual es un protocolo de red que permite establecer una arquitectura cliente-servidor; donde el cliente (en este caso la computadora portátil) envía peticiones a un servidor para emitir un comando, y el servidor es un programa que se ejecuta desde el ordenador del robot.

## Capítulo 5

# Controlador con primitivas de movimiento simples

En este capítulo se presentan un esquema de control visual para robots móviles basado en imagen, en el cual se utilizan las restricciones geométricas inducidas por el tensor trifocal.

Debido a que el TT requiere de tres imágenes para poder calcularse, y solamente se utiliza una cámara, la imagen correspondiente a la ubicación final deben de ser previamente capturada. El robot inicia moviéndose una distancia corta en lazo abierto, pero antes se almacena la imagen correspondiente a la ubicación inicial. Una vez que se dispone de las imágenes correspondientes a las ubicación inicial y final del robot, y que éste avanza una distancia corta de modo que la imagen actual es distinta de la inicial, es posible estimar un TT el robot móvil comienza a desplazarse en lazo cerrado.

Es importante mencionar que durante la navegación del robot móvil jamás se debe perder del campo de vista el objetivo, de lo contrario resultaría imposible estimar un tensor trifocal estable.

El esquema de control que se presenta aquí consta de 3 etapas para lograr que el robot móvil se desplace desde una ubicación inicial hasta su destino, lo cual se representa de manera esquemática en la Figura 5.1. La primer etapa consta de un movimiento de rotación pura, de tal manera que el eje óptico de la cámara montada sobre el robot quede alineado en dirección al objetivo. En la segunda etapa el robot avanza en línea recta hasta alcanzar la posición destino. En la última etapa el robot vuelve a realizar un movimiento de rotación para corregir la orientación.

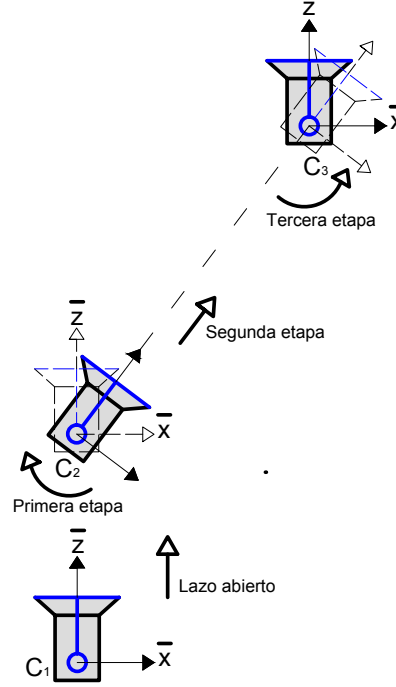


Figura 5.1: Etapas del control visual basado en el tensor trifocal, utilizando primitivas de movimiento simples.

Como se menciono anteriormente la primer etapa consta de un movimiento simple de rotación pura, donde se orienta el robot en dirección al objetivo de tal manera que se logre la condición  $\phi_2 = \phi_t$ , momento en el cual  $x_2 = -z_2 \tan(\phi_t)$ , y  $t_{x2} = 0$ . Por lo tanto al final de ésta etapa los valores de los elementos del tensor trifocal métrico son los siguiente:

$$\mathbf{T}_1 = \begin{bmatrix} -t_{x1} \cos(\phi_2) & 0 & t_{x1} \sin(\phi_2) + t_{z2} \cos(\phi_1) \\ 0 & 0 & 0 \\ -t_{z1} \cos(\phi_2) & 0 & t_{z1} \sin(\phi_2) - t_{z2} \sin(\phi_1) \end{bmatrix},$$

$$\mathbf{T}_2 = \begin{bmatrix} 0 & -t_{x1} & 0 \\ 0 & 0 & t_{z2} \\ 0 & -t_{z1} & 0 \end{bmatrix},$$

$$\mathbf{T}_3 = \begin{bmatrix} -t_{x1} \sin(\phi_2) & 0 & -t_{x1} \cos(\phi_2) + t_{z2} \sin(\phi_1) \\ 0 & 0 & 0 \\ -t_{z1} \sin(\phi_2) & 0 & -t_{z1} \cos(\phi_2) + t_{z2} \cos(\phi_1) \end{bmatrix}.$$
(5.1)

En la segunda etapa el robot se mueve en línea recta para corregir el error longitudinal, momento en el cual  $\phi_2 = \phi_t$ ,  $t_{x2} = 0$  y  $t_{z2} = 0$ . Por consiguiente los elementos del TT métrico toman los siguientes valores:

$$\mathbf{T}_1 = \begin{bmatrix} -t_{x1} \cos(\phi_2) & 0 & t_{x1} \sin(\phi_2) \\ 0 & 0 & 0 \\ -t_{z1} \cos(\phi_2) & 0 & t_{z1} \sin(\phi_2) \end{bmatrix},$$

$$\mathbf{T}_2 = \begin{bmatrix} 0 & -t_{x1} & 0 \\ 0 & 0 & 0 \\ 0 & -t_{z1} & 0 \end{bmatrix},$$

$$\mathbf{T}_3 = \begin{bmatrix} -t_{x1} \sin(\phi_2) & 0 & -t_{x1} \cos(\phi_2) \\ 0 & 0 & 0 \\ -t_{z1} \sin(\phi_2) & 0 & -t_{z1} \cos(\phi_2) \end{bmatrix}.$$
(5.2)

Al final de la última etapa el robot se orienta para llegar a su destino. Cuando dicho error es cero se tiene  $\phi_2 = 0$ ,  $t_{x2} = 0$  y  $t_{z2} = 0$ . El TT métrico al final de la trayectoria seguida por el robot presenta los siguientes valores:

$$\begin{aligned}\mathbf{T}_1 &= \begin{bmatrix} -t_{x1} & 0 & 0 \\ 0 & 0 & 0 \\ -t_{z1} & 0 & 0 \end{bmatrix}, \\ \mathbf{T}_2 &= \begin{bmatrix} 0 & -t_{x1} & 0 \\ 0 & 0 & 0 \\ 0 & -t_{z1} & 0 \end{bmatrix}, \\ \mathbf{T}_3 &= \begin{bmatrix} 0 & 0 & -t_{x1} \\ 0 & 0 & 0 \\ 0 & 0 & -t_{z1} \end{bmatrix}.\end{aligned}\tag{5.3}$$

## 5.1. Definición del control para cada etapa

### 5.1.1. Control directo con elementos del TT

Para establecer la ley de control que permita alcanzar los objetivos deseados al final de cada etapa, es posible utilizar ciertos elementos del tensor trifocal como salidas del sistema. Así pues las velocidades del robot (de rotación y traslación) necesarias para que el robot logre realizar la tarea de posicionamiento deseada, serán definidas como controladores proporcionales en función de algunos elementos del tensor.

Las salidas del sistema se seleccionaron al observar el comportamiento de los elementos del tensor a lo largo de cada una de las trayectorias seguidas por el robot. Por ejemplo, los valores iniciales y finales del TT en la primer etapa están determinados por las ecuaciones 3.35, y 5.1 respectivamente, donde se puede observar que existen algunos términos, o combinación de ellos que parten de un valor y tienden a cero conforme el robot se alinea en dirección al objetivo, es decir existen valores que dependen directamente de la orientación del robot como el elemento  $T_{221}$ , el cual es un buen candidato para salida del sistema en la primera etapa. De manera similar se lleva a cabo el análisis en las etapas posteriores, y se aprecia que existen elementos que dependen directamente de la distancia longitudinal del robot al origen del sistema de referencia, los cuales decrecen su valor conforme el robot se desplaza en línea recta hacia el objetivo, como es el caso del elemento  $T_{223}$ . Por último se aprecia que el elemento  $T_{311}$  es un buen candidato para corregir la orientación del robot en la tercera etapa.

A continuación se presentan las leyes de control para cada una de las etapas.

Primer etapa:

$$\begin{pmatrix} \nu_1 \\ \omega_1 \end{pmatrix} = \begin{pmatrix} 0 \\ k_{\omega 1} T_{221} \end{pmatrix},\tag{5.4}$$

Segunda etapa:

$$\begin{pmatrix} \nu_2 \\ \omega_2 \end{pmatrix} = \begin{pmatrix} k_{\nu 2} T_{223} \\ k_{\omega 2} T_{221} \end{pmatrix},\tag{5.5}$$

Tercera etapa:

$$\begin{pmatrix} \nu_3 \\ \omega_3 \end{pmatrix} = \begin{pmatrix} 0 \\ k_{\omega 3} T_{311} \end{pmatrix},\tag{5.6}$$

donde  $k_{\omega 1}$ ,  $k_{\omega 2}$ ,  $k_{\omega 3}$  y  $k_{\nu 2}$  son las ganancias del control. De manera similar a como se describió anteriormente se pueden seleccionar otras salidas del sistema para cada etapa. Por ejemplo, la velocidad de rotación en la primera etapa también se puede obtener de la siguiente forma  $\omega_1 = k_{\omega} (T_{311}/T_{111}) - (T_{331}/T_{131})$ . La velocidad de traslación en la segunda etapa se pueden obtener mediante  $\nu_2 = k_{\nu 2} (T_{113}/T_{313}) - (T_{133}/T_{333})$ ,

$\nu_2 = k_{\nu 2} (T_{111}/T_{313})$  o  $\nu_2 = k_{\nu 2} (T_{131}/T_{333})$ . Y la velocidad de rotación en la última etapa como  $\omega_3 = k_{\omega} (T_{113})$ ,  $\omega_3 = k_{\omega} (T_{133})$  o  $\omega_3 = k_{\omega} (T_{331})$ .

### 5.1.2. Control para seguimiento de trayectoria

Con las leyes de control presentadas en la sección anterior, en las ecuaciones 5.4, 5.5, y 5.6, se logra llevar al robot móvil al objetivo deseado al final de cada etapa. Sin embargo, como las velocidades (de rotación y traslación) dependen directamente de los valores de algunos elementos del tensor trifocal, estas parten de un determinado valor y comienzan a decrecer conforme se aproximan al objetivo, lo que ocasiona que el robot se mueva a velocidades muy bajas al final de cada etapa. Para evitar una convergencia muy lenta y poder controlar el tiempo en el que se realiza la tarea de posicionamiento, se optó por definir una trayectoria de referencia para los elementos del tensor involucrados en las leyes de control. De esta manera, el control por seguimiento de una referencia permite fijar un tiempo de duración de la tarea evitando velocidades muy bajas al final de cada etapa, y velocidades grandes al inicio. Este esquema donde el movimiento del robot es suave al inicio de cada etapa también facilita el rastreo de puntos de interés en las imágenes, ya que el esquema de retroalimentación directa de los elementos del tensor puede exigir una velocidad de rotación tan alta que el algoritmo de rastreo de puntos llegue a fallar.

Al introducir trayectorias de referencia para las salidas del sistema se puede lograr la evolución deseada de los elementos del TT, de tal manera que ahora las leyes de control dependen directamente del error en cada etapa, el cual se puede definir de manera general como:

$$e = \xi - \xi_d, \quad (5.7)$$

donde  $\xi$  es la salida del sistema, la cual está en función de algunos elementos del tensor en particular según la etapa de control en la que se esté trabajando, y  $\xi_d$  es la trayectoria de referencia la cual para éste caso se definió de la siguiente forma:

$$\xi_d = \begin{cases} \frac{\xi_{inicial}}{2} (1 + \cos(\frac{\pi t}{\tau})), & 0 \leq t \leq \tau \\ \xi_d = 0, & t > \tau. \end{cases} \quad (5.8)$$

En la Figura 5.2, se muestra la forma de la trayectoria de referencia que se definió en la ecuación anterior.

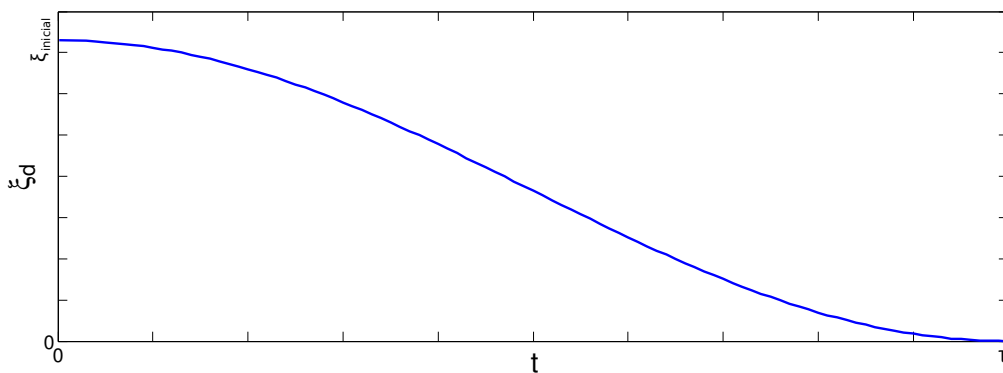


Figura 5.2: Trayectoria de referencia utilizada en las etapas de control.

En la ecuación anterior  $\tau$  representa un tiempo horizonte, en el cual el robot debe alcanzar el objetivo deseado al término de cada etapa, con lo cual además de controlar la evolución de las salidas del sistema, y con ello las velocidades para el desplazamiento del robot, es posible definir el tiempo en el que deseamos

se ejecute cada sub-objetivo. Por otro lado  $t$  representa el tiempo transcurrido para cada etapa de manera independiente.

Las leyes de control después de introducir las funciones de referencia son:

Primera etapa:

$$\begin{pmatrix} \nu_1 \\ \omega_1 \end{pmatrix} = \begin{pmatrix} 0 \\ k_{\omega 1} e_1 \end{pmatrix}, \quad (5.9)$$

Segunda etapa:

$$\begin{pmatrix} \nu_2 \\ \omega_2 \end{pmatrix} = \begin{pmatrix} k_{\nu 2} (e_2) \\ k_{\omega 2} T_{221} \end{pmatrix}, \quad (5.10)$$

Tercera etapa:

$$\begin{pmatrix} \nu_3 \\ \omega_3 \end{pmatrix} = \begin{pmatrix} 0 \\ k_{\omega 3} e_3 \end{pmatrix}. \quad (5.11)$$

## 5.2. Resultados

### 5.2.1. Evaluación general del tensor trifocal

En esta sección se muestran los resultados obtenidos de simulaciones y experimentos que se realizaron para el esquema de control visual que se presentó en las secciones previas, y además se muestra de manera general una evaluación realizada al algoritmo implementado para el cálculo del TT.

Las primeras pruebas que se realizaron fueron para comprobar que el tensor trifocal se calculara de manera correcta. Para esto se utilizaron datos de imágenes sintéticas generadas en Matlab utilizando una nube de puntos con la finalidad de estimar los elementos del TT a partir de información visual, y poder contrastarlos con los valores obtenidos de información métrica teniendo disponibles las posiciones de cámaras.

La razón por la cual se realizaron pruebas con imágenes generadas en Matlab fue para comprobar que el algoritmo implementado en C++ para estimar el tensor trifocal funcionara correctamente. En este caso se tiene una escena con condiciones ideales, donde el movimiento del cámara se puede restringir a un plano de manera precisa, y donde se puede conocer con exactitud la posición y orientación de la cámara. Esto permite comparar los resultados del tensor métrico y visual para la misma configuración; de lo contrario, con imágenes reales siempre se presentarán errores en las mediciones de la posición y orientación de la cámara, ocasionando que los tensores comparados no correspondan exactamente al mismo arreglo geométrico.

De los datos que se muestran en la Tabla 5.1, se puede observar que el algoritmo implementado para calcular el tensor trifocal a partir de las coordenadas de puntos de interés en las imágenes funciona correctamente, ya que los valores de cada uno de los elementos son bastantes cercanos a los obtenidos con información métrica. Es importante mencionar que estas pruebas se realizaron con imágenes sintéticas sin ruido, lo que contribuye en gran medida a obtener buenos resultados. En este caso, y en los próximos resultados que se presenten de simulaciones realizadas en Matlab, las ubicaciones de las cámaras se definen como  $\mathbf{C}_1 = (x_1, y_1, \theta_1)$ ,  $\mathbf{C}_2 = (x_2, y_2, \theta_2)$  y  $\mathbf{C}_3 = (x_3, y_3, \theta_3)$ ; debido a que las pruebas se realizan considerando un movimiento restringido a un plano.

Tensor trifocal		
Elementos	Métrico	Visual
$T_{111}$	-0.1994	-0.199441
$T_{112}$	0.0	-6.01542e-08
$T_{113}$	-0.8603	-0.860303
$T_{121}$	0.0	2.72193e-07
$T_{122}$	0.0	-5.32391e-09
$T_{123}$	0.0	-3.93512e-08
$T_{131}$	0.9522	0.952193
$T_{132}$	0.0	-1.61329e-06
$T_{133}$	0.1108	0.110762
$T_{211}$	0.0	-6.78681e-07
$T_{212}$	0.080	0.0800233
$T_{213}$	0.0	-3.66266e-06
$T_{221}$	-0.2793	-0.279264
$T_{222}$	0.0	3.82209e-07
$T_{223}$	-0.8806	-0.880598
$T_{231}$	0.0	-1.01543e-06
$T_{232}$	1.0	1.0
$T_{233}$	0.0	-1.27019e-05
$T_{311}$	0.012	0.0119814
$T_{312}$	0.0	-1.17915e-08
$T_{313}$	0.1663	0.16634
$T_{321}$	0.0	1.07438e-07
$T_{322}$	0.0	-2.78409e-10
$T_{323}$	0.0	7.85427e-07
$T_{331}$	-0.4765	-0.476539
$T_{332}$	0.0	-1.1835e-09
$T_{333}$	0.1039	0.103863

Tabla 5.1: Elementos del tensor trifocal calculados a partir de información métrica e información visual, donde las ubicaciones de cámara están determinadas por  $C_1 = [-2, -11, 0]$ ,  $C_2 = [-5, -9, 0]$  y  $C_3 = [0, 0, 0]$ .

### 5.2.2. Pruebas con un simulador dinámico

En este apartado se tratan de manera particular los resultados obtenidos para el esquema de control visual que se presento en la sección usando el simulador Webots. Las primeras se realizaron con la finalidad de evaluar el cálculo del tensor trifocal a lo largo de una trayectoria donde los puntos de interés se tienen que rastrear adecuadamente en cada una de las imágenes, y se tienen que descartar aquellos que por alguna razón se lleguen a salir del plano de la imagen para poder estimar un tensor trifocal estable.

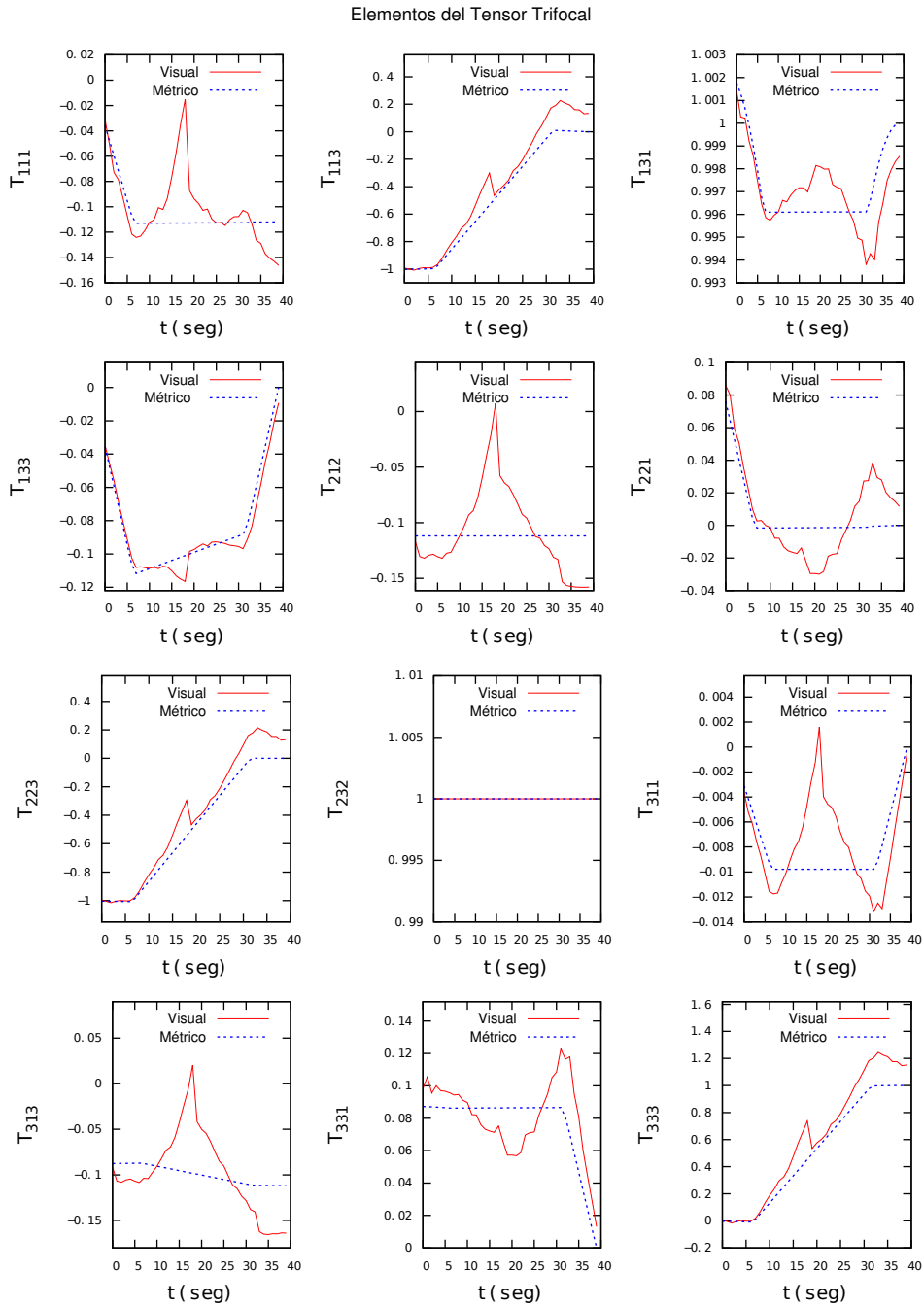


Figura 5.3: Evolución de los elementos no nulos del tensor trifocal para el caso de un movimiento plano. La línea roja continua y la línea azul punteada corresponden a los elementos estimados a partir información visual e información métrica respectivamente.

El simulador proporciona información adicional a la extraída de las características visuales de las imágenes procesadas, como la localización (posición y orientación) del robot, permitiendo así calcular el TT métrico y compararlo con el TT visual durante el recorrido del robot. La Figura 5.3 muestra la evolución de ambos tensores a lo largo de una trayectoria donde el robot móvil inicia con una rotación in situ, después avanza en línea recta y concluye el desplazamiento con otra rotación in situ. Durante estas



pruebas se utilizó el descriptor y el detector SIFT. La gráficas que se presentan permiten apreciar que el TT visual se comporta de la manera esperada (de forma muy similar al TT métrico), asegurando así que los puntos de interés se rastrean adecuadamente en la secuencia de imágenes capturadas, utilizando el KLT.

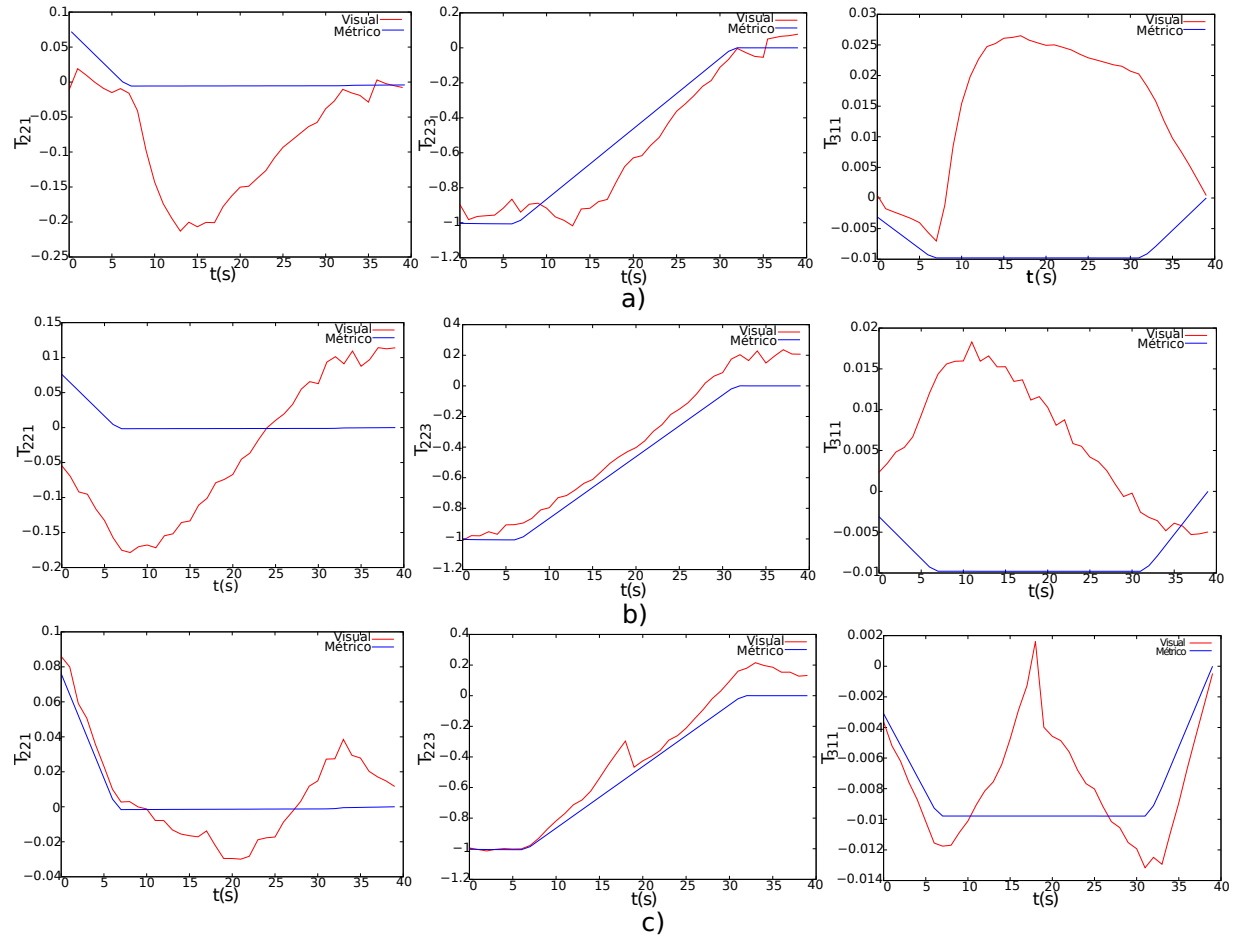


Figura 5.4: Análisis del comportamiento de los elementos del tensor trifocal visual (línea roja continua) que se seleccionaron como salidas del sistema (robot-cámara) a lo largo de una trayectoria en contraste con el TT métrico (línea azul punteada). a) Puntos de Harris. b) Puntos SURF. c) Puntos SIFT.

Una vez que se comprobó que el tensor trifocal visual se estima correctamente durante el desplazamiento del robot, se realizaron pruebas con diferentes descriptores y detectores para determinar con cual tipo de puntos se obtiene un mejor desempeño del TT. La Figura 5.4 muestra el comportamiento de algunos elementos del TT para la misma trayectoria pero utilizando puntos extraídos con diferentes Descriptores y Detectores, donde se puede observar que los mejores resultados se obtienen empleado puntos SIFT.

Utilizando el detector y descriptor SIFT se obtuvo una gran cantidad de puntos correspondientes (255 puntos) en las tres imágenes, lo cual ayuda a definir bien la escena. Sin embargo, hay que considerar que aumentando el número de puntos también incrementa el tiempo de computo para las velocidades del robot, lo cual en ciertos casos podría repercutir en el esquema de control.

Al emplear el detector y descriptor SURF se obtuvieron buenos resultados, pero el número de puntos correspondientes en las tres imágenes es muy pequeño en comparación con los obtenidos con SIFT (40

putos), por lo cual es necesario tener un panorama con mucha textura donde sea posible extraer suficientes puntos para definir bien la escena. Para el caso donde se usaron puntos de Harris, a los cuales se asociaron descriptores SIFT para la búsqueda de correspondencias, los resultados fueron satisfactorios, pero igual que al utilizar SURF el número de correspondencias (63 puntos) fue pequeño, lo cual por una parte podría ser perjudicial ya que se requiere una escena muy texturizada para obtener suficientes puntos para calcular un tensor trifocal estable. Sin embargo, también se observó que utilizando puntos SURF o de Harris la cantidad de correspondencias necesarias para obtener resultados aceptables es menor a la que se requiere en el caso de puntos SIFT, lo cual puede ayudar en el desempeño del control al disminuir el tiempo de cómputo.

Para implementar el control visual en este caso se optó por utilizar puntos SIFT, ya que con ellos se obtuvieron los mejores resultados para el tensor trifocal y el tiempo de cómputo no aumenta mucho en comparación con los otros dos casos analizados. A pesar de que se tomó la decisión de emplear SIFT es importante tener en cuenta las observaciones anteriores para realizar la mejor elección dependiendo de la escena observada, o del tiempo óptimo para calcular las velocidades ( $\nu$  y  $\omega$ ) del robot.

Para evaluar el controlador se realizaron pruebas utilizando las leyes de control que están en función de los elementos del tensor trifocal, y las que están en función de un error al definir trayectorias de referencia para los elementos del Tensor (seguimiento de referencia).

#### 5.2.2.1. Control directo con elementos del TT

Con la finalidad de observar el desempeño del controlador se realizaron diferentes pruebas, donde el robot móvil debe avanzar o retroceder después de la primera rotación en sitio, con la finalidad de alcanzar el objetivo a partir de su ubicación inicial.

Para las pruebas realizadas donde el robot debe avanzar para alcanzar su objetivo se utiliza el arreglo mostrado en la Figura 5.5. El tamaño de las imágenes empleadas en todas las pruebas para la extracción de las características visuales fue de  $640 \times 480$  píxeles. Las unidades para las velocidades de rotación y traslación del robot móvil fueron en ( $\text{rad}/\text{seg}$ ) y ( $\text{m}/\text{seg}$ ) respectivamente.

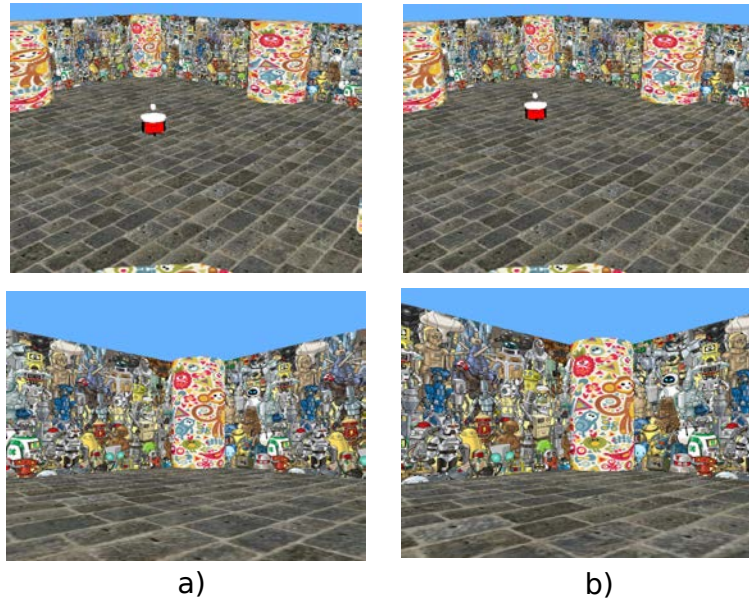


Figura 5.5: Configuración del experimento donde el robot móvil debe avanzar para alcanzar su objetivo. a) Ubicación inicial del robot (superior) e imagen observada en dicho momento (inferior). b) Ubicación destino del robot (superior) e imagen observada.

En la primer prueba realizada donde el robot debe avanzar hasta alcanzar su objetivo (posición deseada que corresponde a la imagen 5.5 b) donde se asocia el origen del sistema de coordenadas) se utilizaron las siguientes ganancias de control  $k_{\omega_1} = k_{\omega_2} = 0.42$ ,  $k_{\nu_1} = 0.064$  y  $k_{\omega_3} = 1.95$ .

La Figura 5.6 muestra la evoluciones de los elementos del tensor involucrados en el esquema de control, cada uno de estos valores tienden a converger a cero al término de la etapa en la cual intervienen como salidas del sistema. Al finalizar la prueba el robot prácticamente llega a la ubicación deseada. Como se puede apreciar el elemento  $T_{223}$  converge de manera casi exacta a cero lo cual indica que en teoría el error longitudinal se corrigió completamente. El elemento  $T_{311}$  alcanza un valor casi nulo corrigiéndose en gran medida la orientación del robot; el error resulta ser más pequeño de lo esperado probablemente debido a que el elemento  $T_{221}$  tampoco converge exactamente a cero, razón por la cual al finalizar la segunda etapa el error longitudinal se corrige pero se introduce un pequeño error lateral, permitiendo corregir la orientación durante la tercera etapa sin que se presenté el caso degenerado del tensor trifocal. La Figura 5.7 muestra la imagen observada por el robot al término de la trayectoria y la imagen objetivo para que se puedan contrastar, las cuales son muy similares.

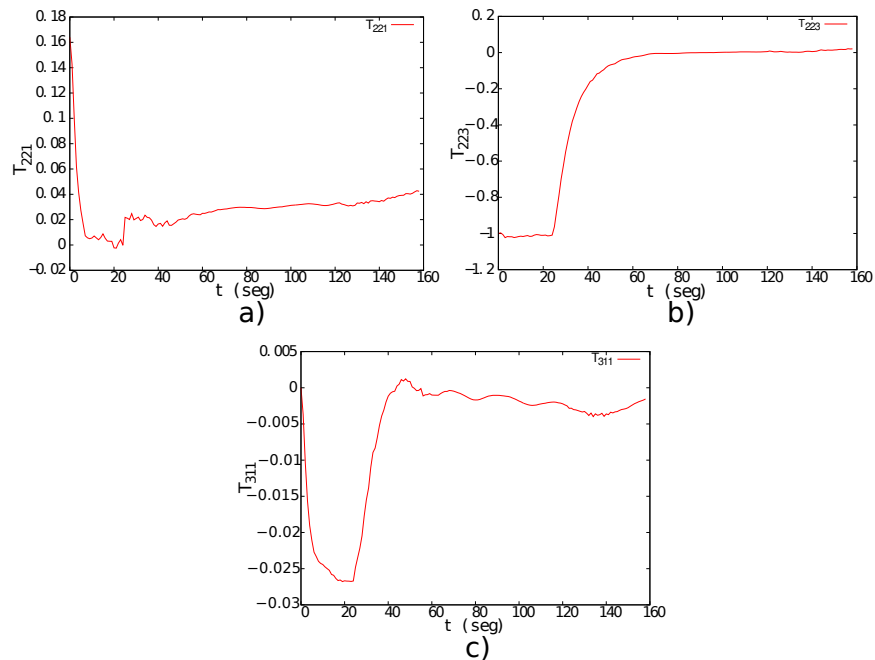


Figura 5.6: Evolución de las salidas del sistema. a) Elemento utilizado en la ley de control durante la primera etapa. b) Elemento empleado en la segunda etapa de control. c) Elemento utilizado en la tercera etapa de control.



Figura 5.7: Resultados visuales. a) Imagen observada por el robot al termino de su trayectoria. b) Imagen correspondiente a la ubicación objetivo.

El error longitudinal obtenido fue 0.072 metros y el error en la orientación de 0.015 radianes. Para poder realizar una mejor evaluación del desempeño del sistema se calculo el valor RMS de la distancia euclidiana, de la distancia en  $x$ , y de la distancia en  $y$ , entre las coordenadas de los puntos de la imagen objetivo y la imagen capturada durante el recorrido del robot. El valor RMS de la distancia en la coordenada  $y$  alcanza un valor pequeño de aproximadamente 2 píxeles, indicando que el error longitudinal se corrigió prácticamente por completo. Los valores RMS para la distancia euclidiana y la distancia en  $x$  son un poco mayores debido a que la orientación no se corrige totalmente y por la desviación lateral que se introduce,

sin embargo el resultado obtenido es bueno pues como se puede apreciar de la Figura 5.8 los valores RMS al final de la trayectoria son pequeños en general. La evolución de la ubicación del robot en el plano  $x-z$ , las velocidades de rotación y traslación, así como el movimiento de algunos puntos de interés en el plano de la imagen se muestran en las Figuras 5.9, 5.10 y 5.11.

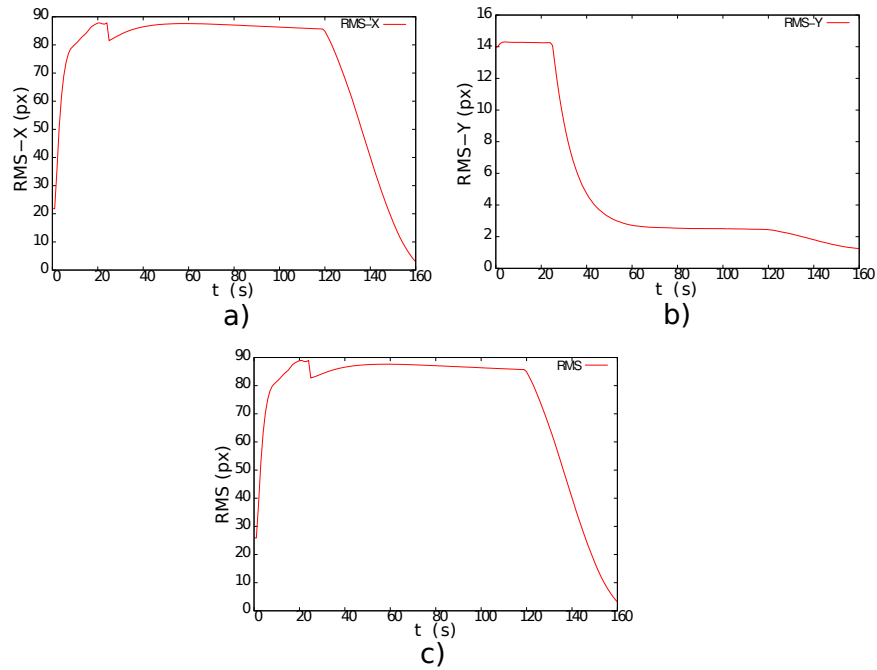


Figura 5.8: Evolución de los valores RMS a lo largo de la trayectoria del robot. a) Valores RMS de la distancia en  $x$  entre puntos en el plano imagen. b) Valores RMS de la distancia en  $y$  entre puntos. c) Valores RMS de la distancia euclidiana entre puntos en la imagen.

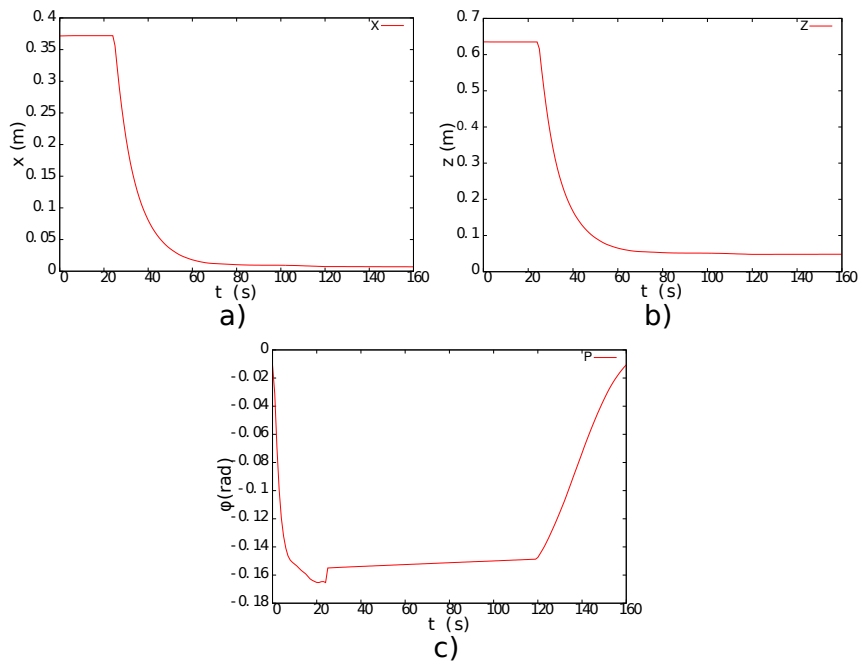


Figura 5.9: Evolución de la ubicación del robot. a) Posición lateral. b) Posición longitudinal. c) Orientación.

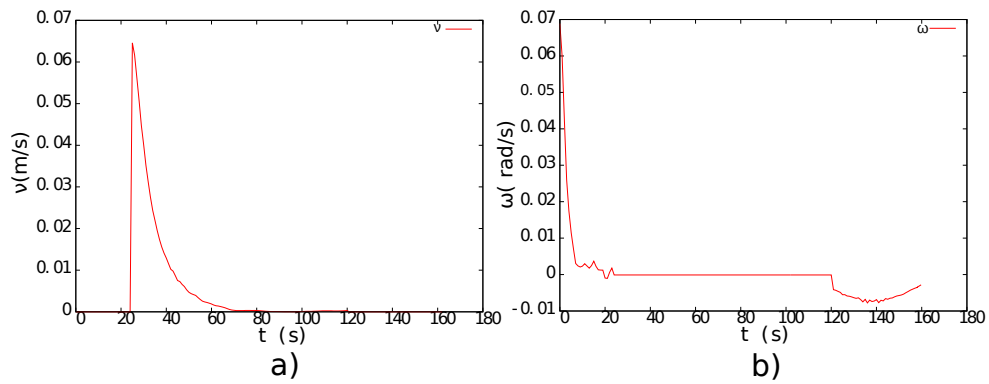


Figura 5.10: Velocidades del robot. a) Velocidad de traslación. b) Velocidad de rotación.

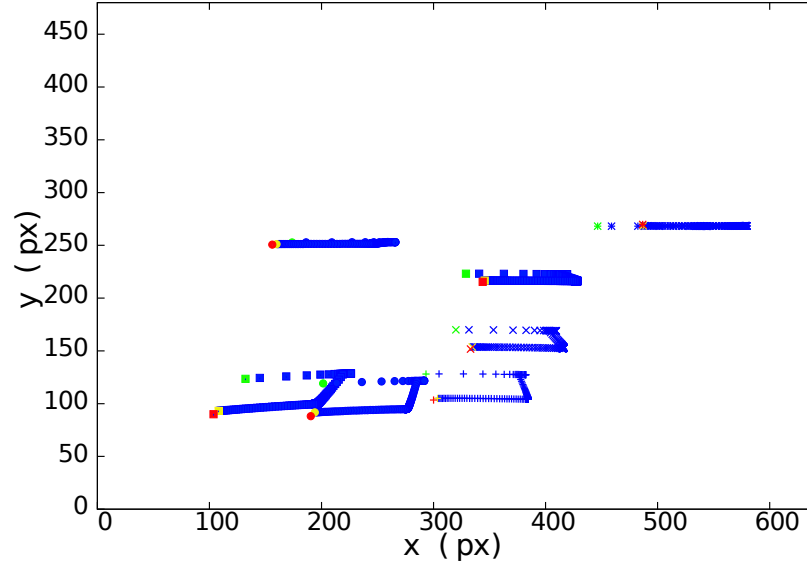


Figura 5.11: Movimiento de los puntos en la imagen. Los puntos verdes indican la posición inicial; En azul se representa la trayectoria de los puntos en la imagen. Los puntos amarillos representan la posición alcanzada al final de la trayectoria del robot móvil. Y los puntos rojos corresponden a la posición objetivo.

El experimento se repitió 20 veces con la finalidad de obtener datos estadísticos que permitan tener una mejor evaluación de los resultados obtenidos. La desviación estándar y la media aritmética del error longitudinal fueron 0.0008312 y 0.0708 metros respectivamente. La media y la desviación estándar para el error en la orientación fueron 0.01518 y 0.00035627 radianes respectivamente.

A continuación se presentan los resultados de una prueba donde el robot debe desplazarse hacia atrás para alcanzar su objetivo, el arreglo experimental se muestra en la Figura 5.12. Para evaluar el controlador se utilizaron las siguientes ganancias de control  $k_{\omega 1} = k_{\omega 2} = 0.45$ ,  $k_{\nu 1} = 0.0215$  y  $k_{\omega 3} = 3.2$ .

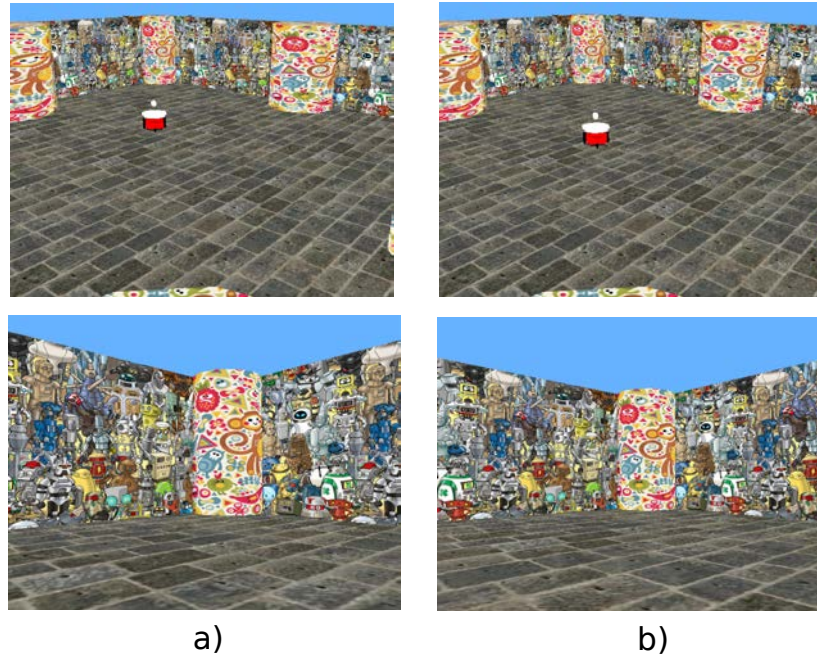


Figura 5.12: Configuración del experimento donde el robot móvil debe retroceder para alcanzar su objetivo. a) Ubicación inicial del robot (superior) e imagen observada en dicho momento (inferior). b) Ubicación destino del robot (superior) e imagen observada.

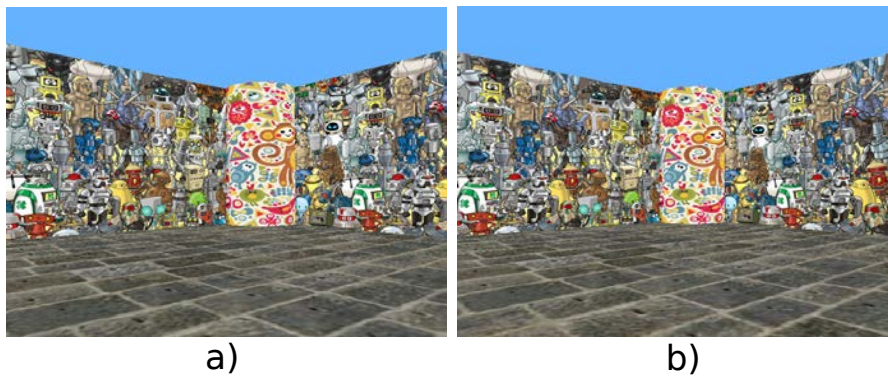


Figura 5.13: Resultados visuales. a) Imagen observada por el robot al termino de su trayectoria. b) Imagen correspondiente a la ubicación objetivo.



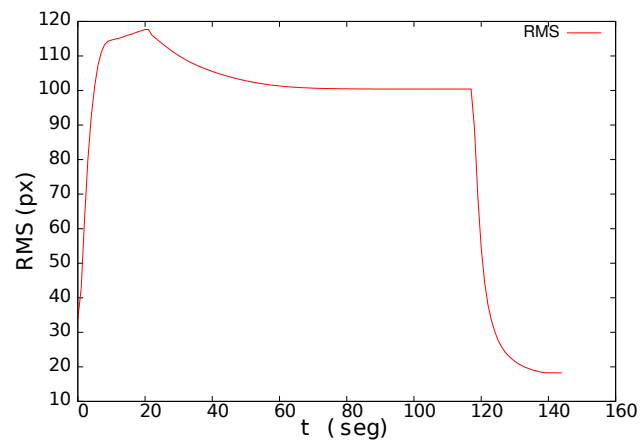


Figura 5.14: Evolución del valor RMS de la distancia euclidiana entre puntos en el plano de la imagen.

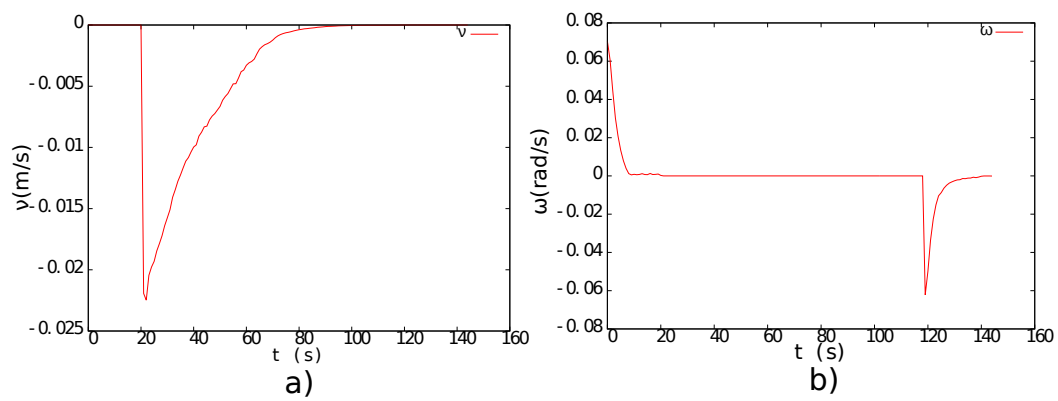


Figura 5.15: Velocidades del robot. a) Velocidad de traslación. b) Velocidad de rotación.

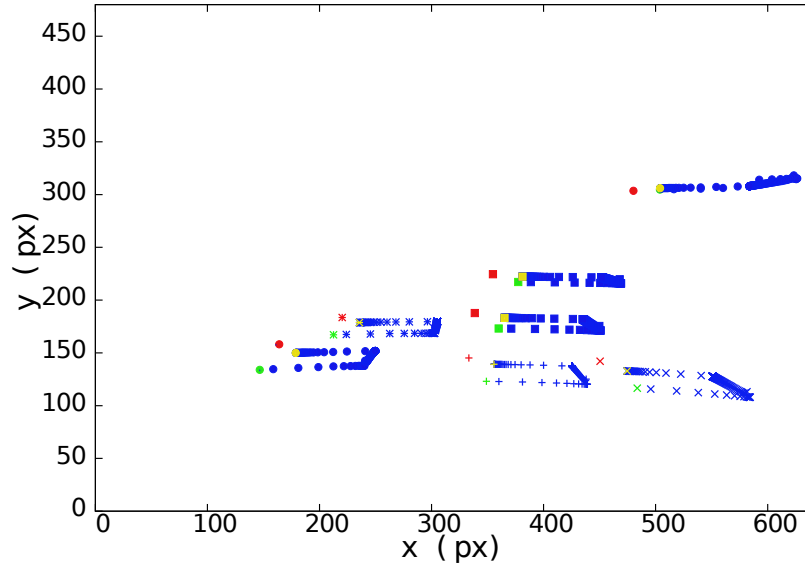


Figura 5.16: Movimiento de los puntos en la imagen. Los puntos verdes indican la posición inicial. En azul se representa la trayectoria de los puntos en la imagen. Los puntos amarillos representan la posición alcanzada al final de la trayectoria del robot móvil. Y los puntos rojos corresponden a la posición objetivo.

Observando la Figura 5.13 aparentemente los resultados obtenidos son bastante buenos, ya que la imagen observada por el robot móvil al termino de la trayectoria es muy similar a la imagen objetivo, sin embargo el valor RMS de la distancia euclidiana entre los puntos de interés al final de la trayectoria del robot no es tan pequeño como se desearía. La causa principal por la cual se llegan a obtener este tipo de resultados es debido a que el tensor trifocal estimado a partir de información visual, siempre difiere del tensor métrico aunque sea en lo más mínimo como se puede apreciar de la Figura 5.3; por ejemplo el elemento  $T_{223}$  converge a cero al corregirse el error longitudinal para el caso del tensor métrico, sin embargo con la información visual el tensor alcanza un valor aproximado de 0.2 cuando se corrige el error longitudinal. En ocasiones esta diferencia puede repercutir más de lo esperado, ya que cuando el elemento alcanza un valor de cero realmente no se ha corregido completamente el error longitudinal, sin embargo se asume que si, debido a que los elementos del tensor son las salidas del sistema de control. Lo anterior puede ocurrir para cualquiera de los elementos utilizados en las diferentes etapas, lo cual también ocasiona que el error de la orientación se corrija más de lo supuesto, debido a que jamás se presenta el caso degenerado del tensor trifocal como se esperaría teóricamente cuando la imagen actual es prácticamente igual a la imagen objetivo.

El error longitudinal obtenido fue de 0.098 metros, y el error en la orientación 0.025 radianes. La desviación estándar y la media aritmética del error longitudinal para un conjunto de 20 pruebas, fueron 0.001489 y 0.093 metros respectivamente. La media y la desviación estándar para el error en la orientación fueron 0.0263 y 0.00044 radianes respectivamente.

#### 5.2.2.2. Control para seguimiento de trayectoria.

Con el esquema de control para seguimiento de trayectoria, también se realizaron pruebas donde el robot móvil debe avanzar o retroceder con la finalidad de llegar a su objetivo. Las configuraciones del experimento para cada caso son las mismas que se utilizaron en las pruebas donde las leyes de control están en función de los elementos del tensor trifocal, las cuales se pueden apreciar en las imágenes 5.5 y 5.12.

Para evaluar el controlador cuando el robot debe avanzar hasta su objetivo se realizó una prueba donde se utilizaron las siguientes ganancias para el control  $K_{\omega_1} = K_{\omega_2} = 0.55$ ,  $K_{\nu_1} = 0.10$  y  $K_{\omega_3} = 3.7$

Al término de la trayectoria del robot móvil la imagen observada es bastante parecida a la imagen objetivo como se puede apreciar en la Figura 5.17. Las salidas del sistema en cada una de las etapas alcanzan valores prácticamente nulos, y la posición y orientación del robot también convergen a cero como se puede apreciar en las Figuras 5.18 y 5.19. La gráficas de los valores RMS mostrada en la Figura 5.20, permiten comprobar que efectivamente se obtienen resultados satisfactorios a pesar de que no se alcanza exactamente el objetivo. No es posible llegar exactamente al objetivo, en primer lugar por que siempre se esperaría un pequeño error en la orientación del robot con la finalidad de evitar el caso degenerado del TT, y además por las diferencias que presenta el tensor trifocal visual con respecto al métrico. Una de las principales diferencias de esta prueba con respecto a las anteriores son las velocidades del robot, las cuales como se puede observar de la Figura 5.21, tienen el comportamiento senoidal de las trayectorias de referencia que se emplearon, y de la evolución de las funciones de error que se muestran en la Figura 5.22, lo que mejora el desempeño del control evitando cambios bruscos de velocidad al término de cada etapa de control.



Figura 5.17: Resultados visuales. a) Imagen observada por el robot al termino de su trayectoria. b) Imagen correspondiente a la ubicación objetivo.

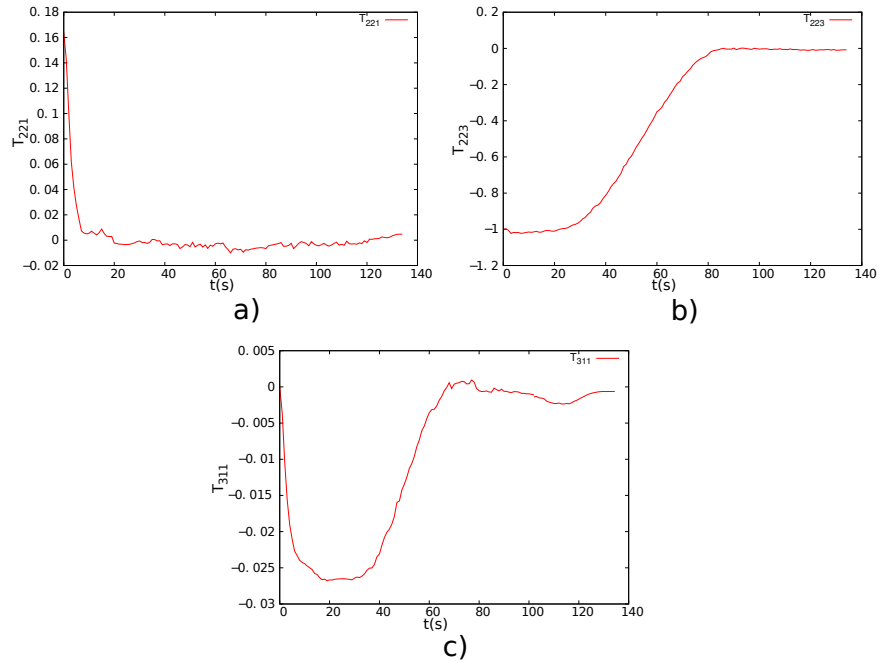


Figura 5.18: Evolución de las salidas del sistema. a) Elemento utilizado en la ley de control durante la primera etapa. b) Elemento empleado en la segunda etapa de control. c) Elemento utilizado en la tercera etapa de control.

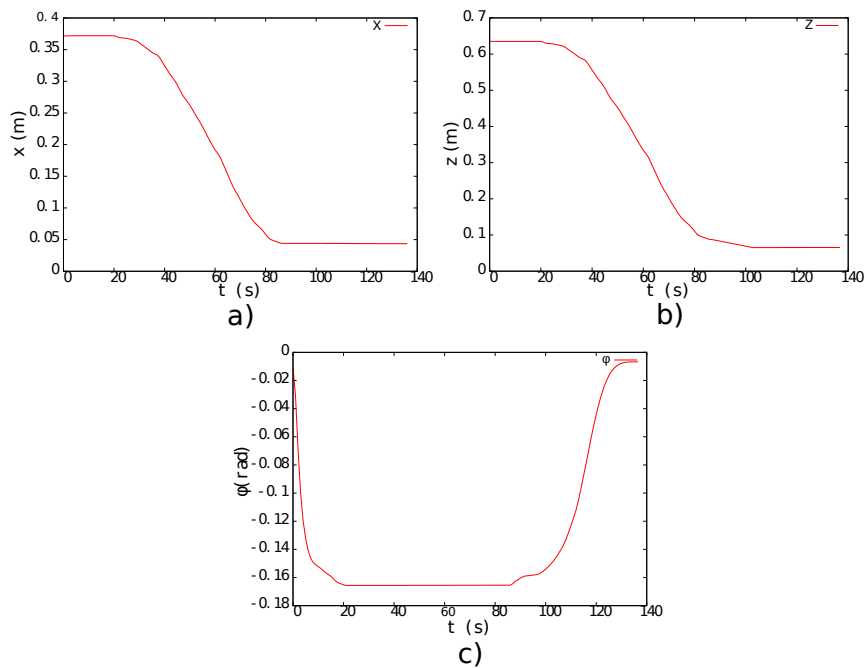


Figura 5.19: Evolución de la ubicación del robot. a) Posición lateral. b) Posición longitudinal. c) Orientación.

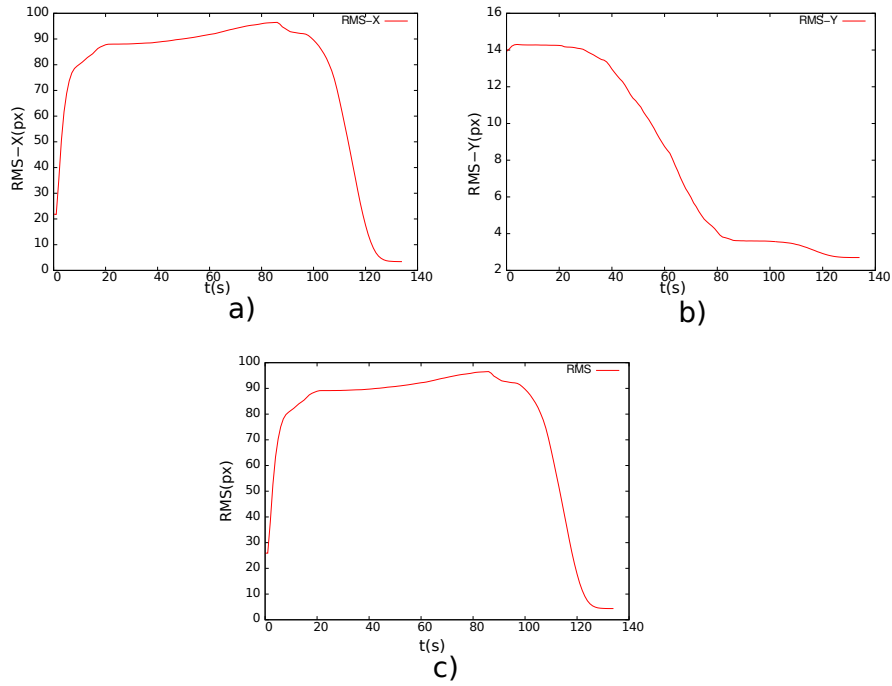


Figura 5.20: Evolución de los valores RMS a lo largo de la trayectoria del robot. a) Valores RMS de la distancia en  $x$  entre puntos en el plano imagen. b) Valores RMS de la distancia en  $y$ . c) Valores RMS de la distancia euclidiana entre puntos.

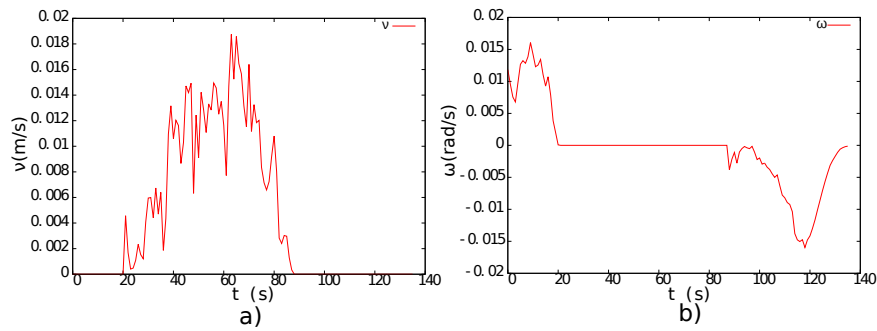


Figura 5.21: Velocidades del robot. a) Velocidad de traslación. b) Velocidad de rotación.

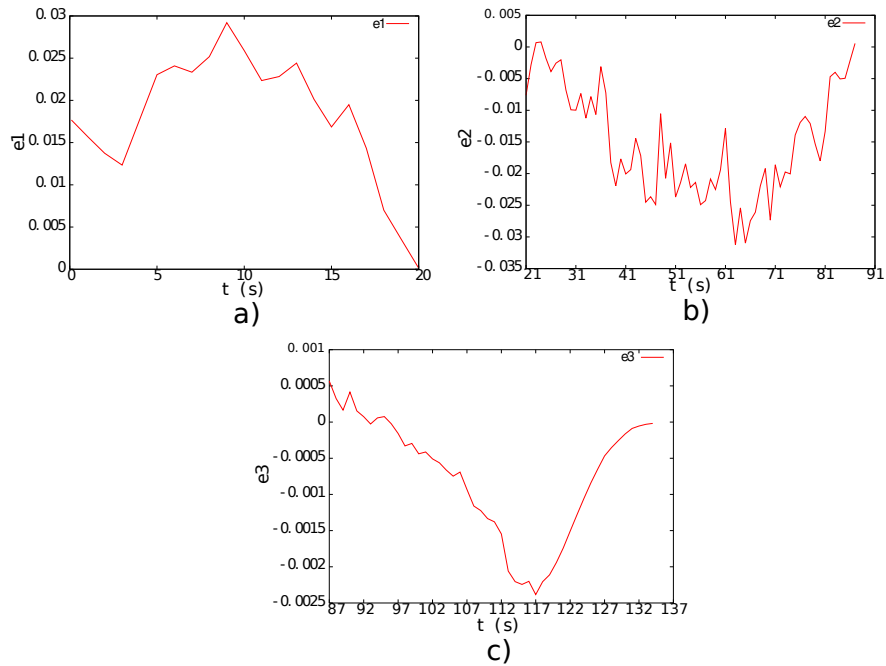


Figura 5.22: Evolución de las funciones de error del sistema. a) función de error de la primera etapa del control. b) función de error de la segunda etapa del control. c) Función de error de la tercera etapa del control.

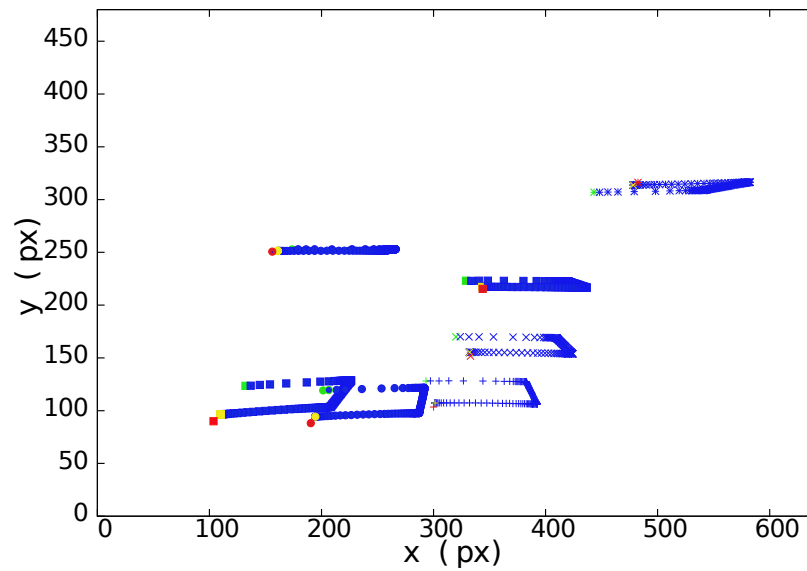


Figura 5.23: Movimiento de los puntos en la imagen. Los puntos verdes indican la posición inicial. En azul se representa la trayectoria de los puntos en la imagen. Los puntos amarillos representan la posición alcanzada al final de la trayectoria del robot móvil. Y los puntos rojos corresponden a la posición objetivo.

El error longitudinal obtenido fue de 0.074 metros, y el error en la orientación 0.01641 radianes. La desviación estándar y la media aritmética del error longitudinal para un conjunto de 20 pruebas, fueron 0.001063 y 0.0722 metros respectivamente. La media y la desviación estándar para el error en la orientación fueron 0.01641 y 0.00021 radianes respectivamente.

En la última prueba realizada donde el robot móvil debe retroceder hasta su objetivo se utilizaron las siguientes ganancias de control  $k_{\omega_1} = k_{\omega_2} = 0.15$ ,  $k_{\nu_1} = 0.0195$  y  $k_{\omega_3} = 3.2$ . En general los resultados fueron satisfactorios, la imagen observada por el robot al término de la trayectoria es similar a la imagen objetivo, y el valor RMS de la distancia euclidiana entre los puntos de interés de la imagen al final y la imagen objetivo alcanza un valor pequeño. Los resultados obtenidos se pueden apreciar en las Figuras 5.24, 5.25, 5.26 y 5.27.

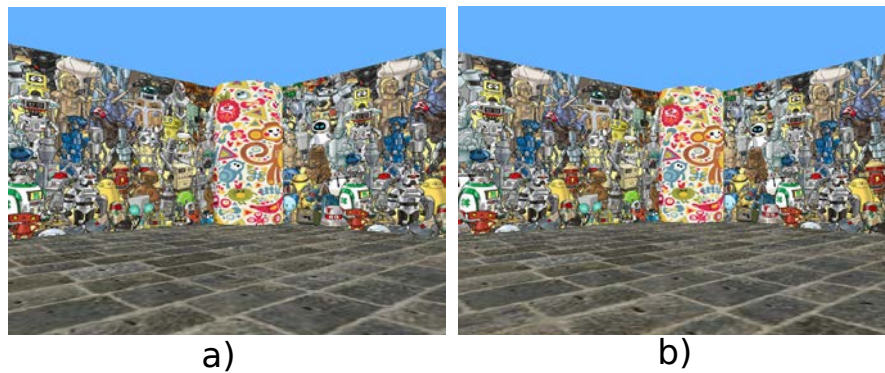


Figura 5.24: Resultados visuales. a) Imagen observada por el robot al término de su trayectoria. b) Imagen correspondiente a la ubicación objetivo.

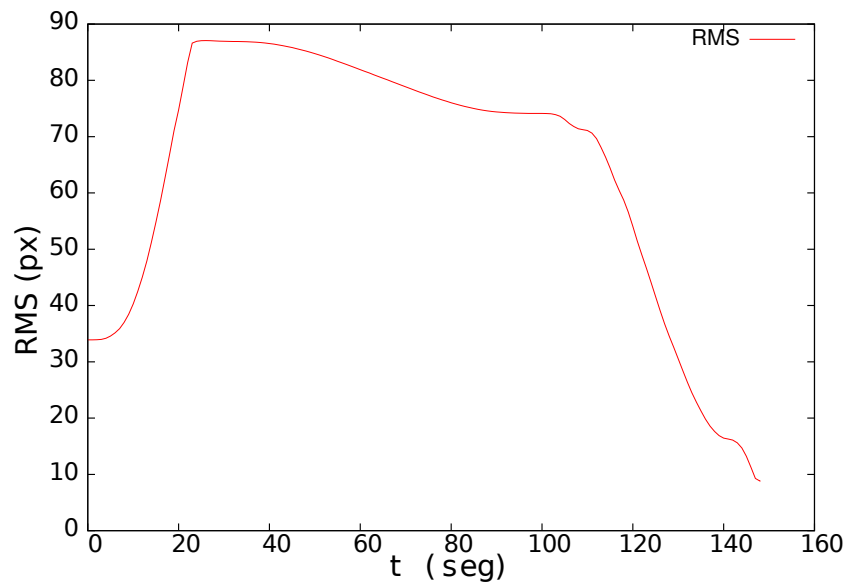


Figura 5.25: Evolución del valor RMS de la distancia euclidiana entre puntos en el plano de la imagen.

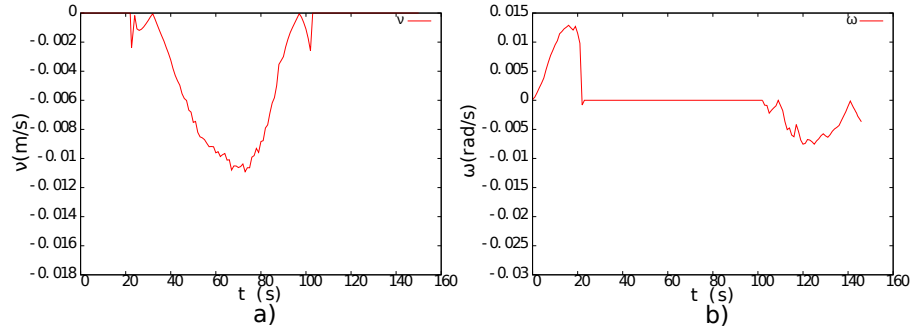


Figura 5.26: Velocidades del robot. a) Velocidad de traslación; b) Velocidad de rotación.

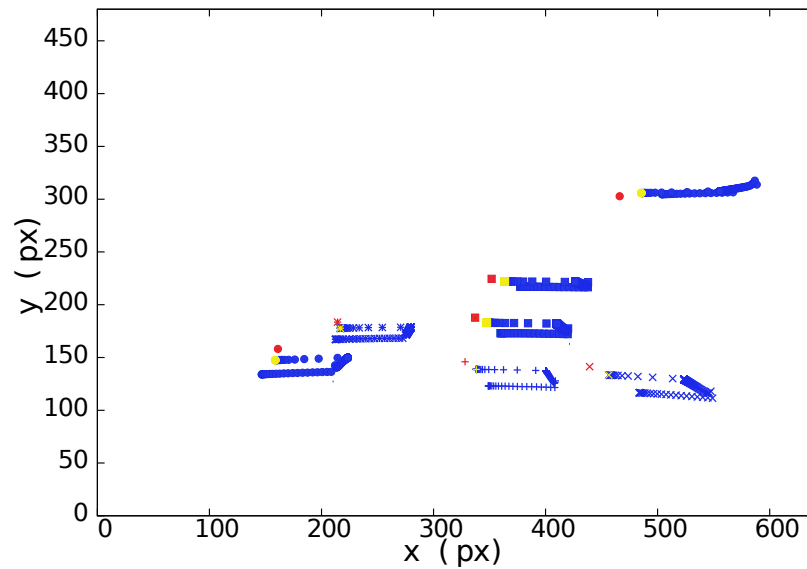


Figura 5.27: Movimiento de los puntos en la imagen. Los puntos verdes indican la posición inicial. En azul se representa la trayectoria de los puntos en la imagen. Los puntos amarillos representan la posición alcanzada al final de la trayectoria del robot móvil. Y los puntos rojos corresponden a la posición objetivo.

El error longitudinal obtenido de la prueba fue de 0.106 metros, y el error en la orientación 0.0226 radianes. La desviación estándar y la media aritmética del error longitudinal para un conjunto de 20 pruebas, fueron 0.001904 y 0.10243 metros respectivamente. La media y la desviación estándar para el error en la orientación fueron 0.02284 y 0.000518 radianes respectivamente.

### 5.2.3. Pruebas con imágenes reales

Las pruebas se realizaron utilizando una cámara web (hp), cuyos parámetros intrínsecos son los siguientes:

$$\mathbf{K} = \begin{bmatrix} 850.70 & 0 & 319.5 \\ 0 & 844.46 & 239.5 \\ 0 & 0 & 1 \end{bmatrix},$$



y sus coeficientes de distorsión son:

$$\mathbf{Coeficientes}_{distorsion} = [ 0.231, -1.8035, 0.0, 0.0, 3.464 ] .$$

Con la finalidad analizar el desempeño del algoritmo implementado para estimar el TT al trabajar con imágenes reales, se realizaron pruebas con una secuencia de imágenes previamente capturada. La secuencia consta de 43 imágenes; se utilizó la primera, la cuarta y la última de ellas para extraer los puntos de interés que se rastrearon con KLT en el resto de las imágenes. La secuencia corresponde a una trayectoria donde el robot inicia con rotación en sitio, y posteriormente continua con un desplazamiento en línea recta. La Figura 5.28, muestra la imagen observada al inicio y al final de la trayectoria sobre la cual se capturaron las imágenes.



Figura 5.28: Experimento con imágenes reales. a) Imagen correspondiente al punto de partida en la trayectoria evaluada. b) Imagen final.

Se realizaron prueba para evaluar el algoritmo con puntos seleccionados usando SIFT, SURF y Harris. Las Figuras 5.29, 5.30 y 5.31, muestran la evolución de dos elementos del TT para cada caso. Los elementos presentados son  $T_{221}$  y  $T_{222}$ ; el primero de ellos se ve afectado de manera directa por la rotación de la cámara con respecto a su eje vertical, y el segundo por el desplazamiento longitudinal.

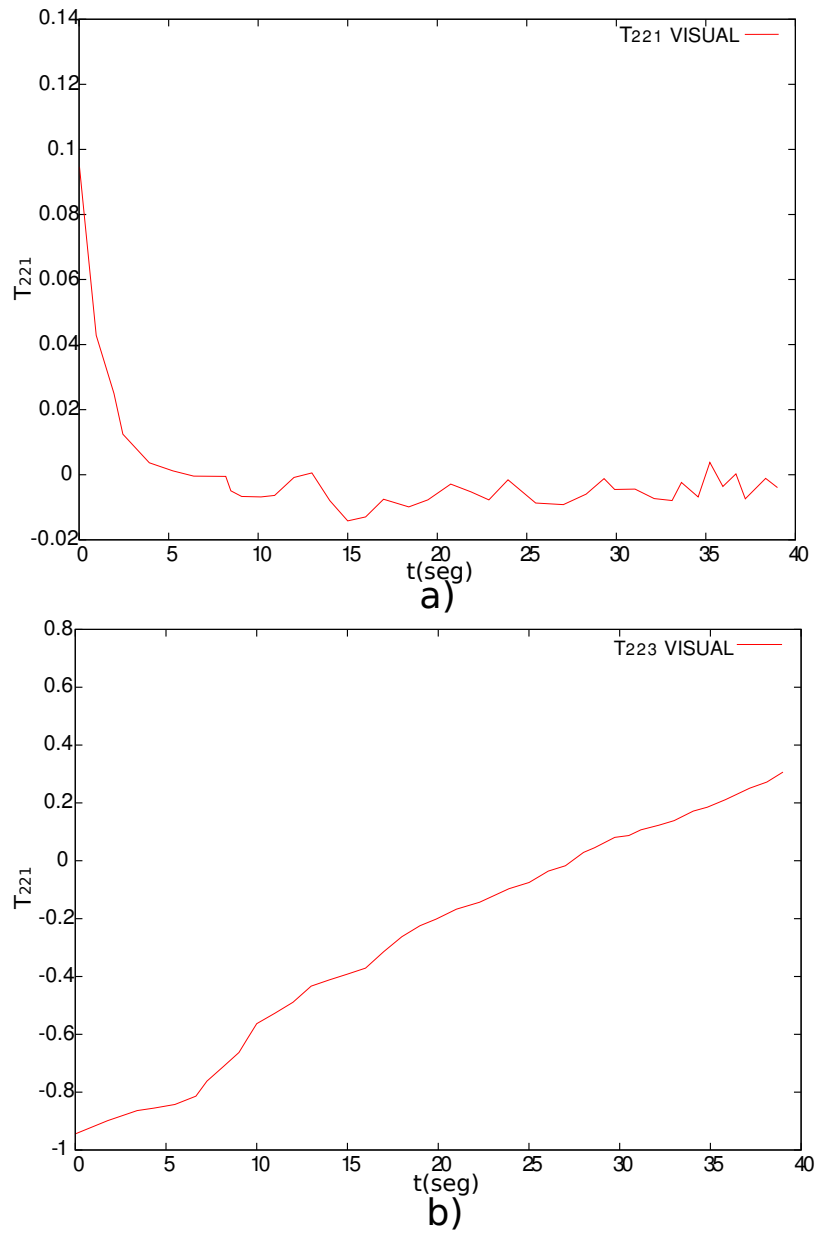


Figura 5.29: Elementos del TT estimados al emplear puntos seleccionados con SIFT. a) Elemento  $T_{221}$ . b) Elemento  $T_{223}$ .

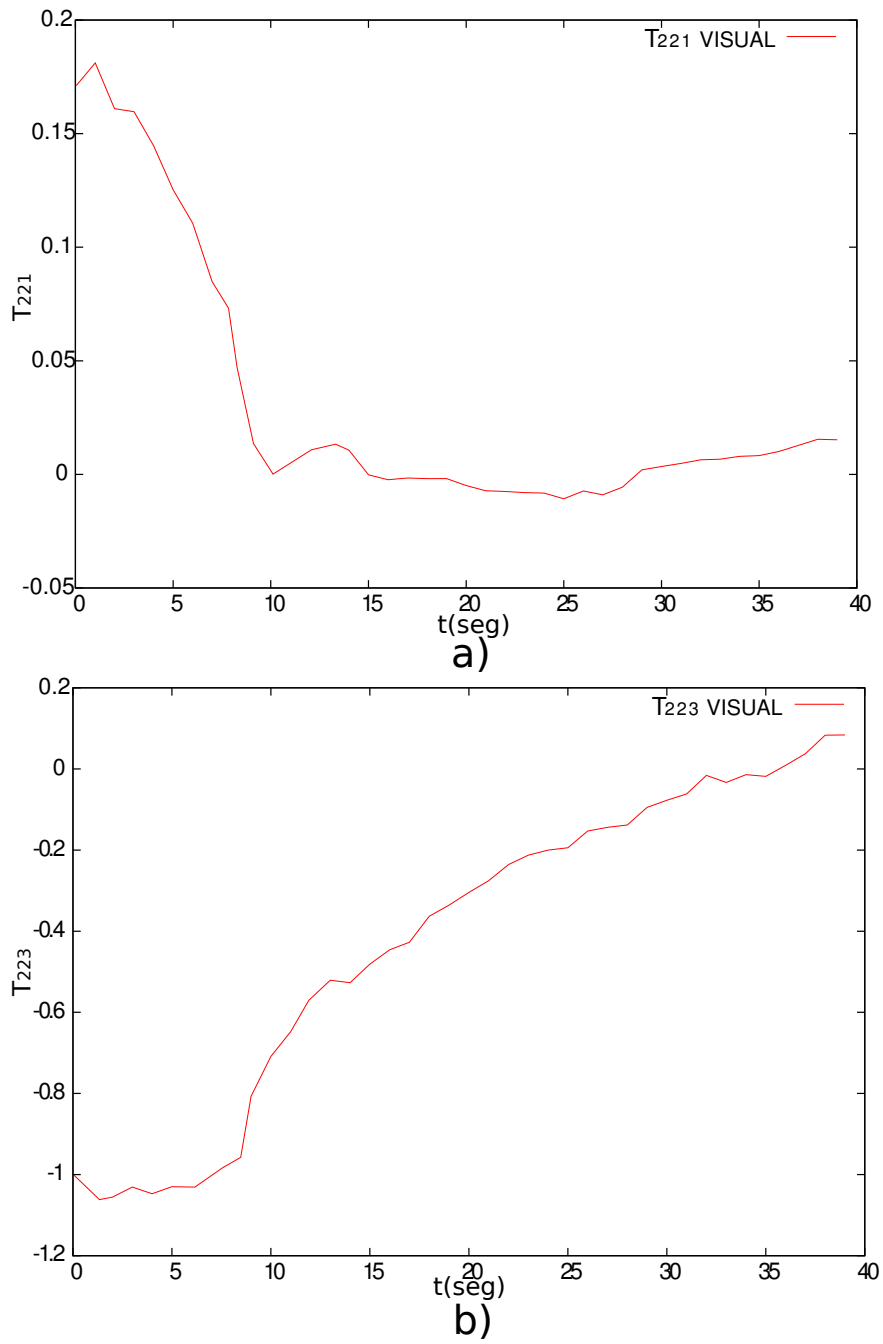


Figura 5.30: Elementos del TT estimados al emplear puntos seleccionados con SURF. a) Elemento  $T_{221}$ . b) Elemento  $T_{223}$ .

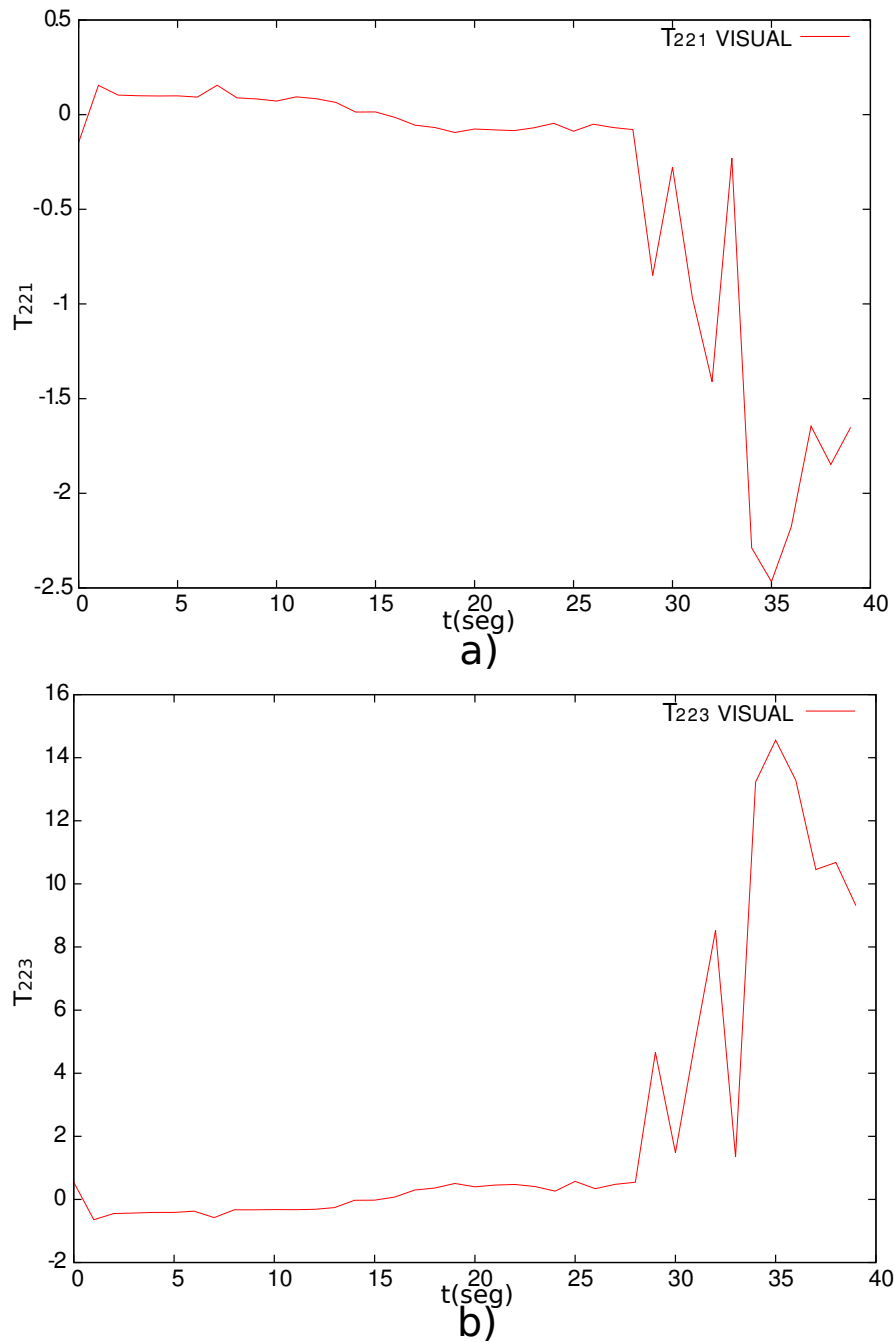


Figura 5.31: Elementos del TT estimados al emplear puntos seleccionados con Harris. a) Elemento  $T_{221}$ . b) Elemento  $T_{223}$ .

Los resultados obtenidos al usar puntos seleccionados con SIFT y SURF son congruentes. En las gráficas 5.29 y 5.30, se puede observar que el elemento  $T_{221}$  converge a cero en el momento que termina la rotación, tal y como se esperaba ya que en dicho momento el sistema (robot-cámara) se alinea con respecto a la posición objetivo (definida por la última imagen de la secuencia). De manera similar el elemento  $T_{223}$  para estos casos converge a cero al momento en que se corrige el error longitudinal.

Los resultados obtenidos al trabajar con Harris no fueron satisfactorios. Es importante mencionar,

que el número de puntos de interés seleccionados para este caso fue muy pequeño (9 puntos). Lo anterior repercute en la estimación de un TT estable en la secuencia de imágenes. El número de puntos disponibles resultó apenas mayor que el mínimo requerido (7 puntos), lo cual no ayudó a definir bien el arreglo geométrico entre cámaras. Para los casos en los que se trabajó con SIFT se seleccionó un conjunto de 43 puntos correspondiente, y al trabajar con SURF se seleccionaron 15 puntos.

Analizando los resultados obtenidos, podemos darnos cuenta de la importancia de trabajar con un número suficiente de puntos correspondientes para estimar un TT estable. Un caso que se puede presentar al disponer de un número pequeño de correspondencia, es que inicialmente se pueda estimar un TT geoméricamente válido; sin embargo durante el tracking, es posible que algunos de los puntos no se puedan rastrear y se tengan que descartar, disminuyendo el número de correspondencia, de manera que no se pueda estimar un TT estable, o que no se disponga ni siquiera del mínimo puntos necesarios para estimarlo.

### 5.3. Conclusiones

En este capítulo se presentó un esquema de control visual basado en imagen utilizando las restricciones geométricas inducidas por el TT, con la finalidad de lograr la navegación autónoma de un robot móvil. Se evaluó el desempeño del controlador utilizando una ley de control en función de los elementos del TT, y una ley de control en función de un error al definir una trayectoria de referencia.

Con los resultados obtenidos de las simulaciones en Webots, podemos darnos cuenta que el esquema que se presentó, permite conducir un robot móvil hasta una ubicación objetivo de manera satisfactoria; ya en efecto se presenta un error geométrico pero sin llegar a ser crucial. El error geométrico, principalmente se presenta por las diferencias entre el TT métrico y el TT visual estimado a partir de información extraña del conjunto de imágenes.

En las simulaciones con Webots se trabajó con imágenes obtenidas a partir de una escena con iluminación uniforme, y por consiguiente libre de sombras. El hecho de no contar con sombras en el espacio 3D, y por lo tanto imágenes sin zonas oscurecidas, ayuda a obtener resultados satisfactorios con los tres tipos de puntos evaluados (SIFT, SURF y Harris), ya que no se presentan puntos difíciles de rastrear.

Al trabajar con imágenes reales, donde es común que se presenten zonas más oscuras que otras debido a las variaciones de iluminación en la escena 3D, los resultados se ven afectados. Estas zonas oscurecidas pueden ser detectadas como puntos de interés, dependiendo del tipo de detector empleado. Sin embargo, al observar la escena desde otro punto de vista cuando el robot se desplaza o rota, es posible que esas variaciones de iluminación no se presenten del mismo modo. Por consiguiente, los cambios de iluminación ocasionan que en las imágenes tomadas desde diferentes ubicaciones, no se presente exactamente el mismo conjunto de puntos de interés, impidiendo realizar un buen rastreo de puntos con KLT. En general se obtienen buenos resultados con puntos SIFT y Harris, pero no con SURF, ya que el tipo de puntos seleccionados con SURF son más difíciles de rastrear.

La cantidad de información visual obtenida a partir de las imágenes, es otro factor que influye directamente en el desempeño del controlador. En los casos con poca textura, el número de puntos emparejados entre las tres imágenes pueden ser insuficientes para estimar el tensor trifocal. Aunque el algoritmo que se implementó para estimar el TT, solo requiere de 7 tripletas de puntos correspondientes en las tres imágenes, en general los resultados obtenidos mejoran al involucrar más puntos en los cálculos. Por otro lado, cuando sólo se dispone del mínimo de puntos para estimar el TT, es posible que durante la navegación del robot algunos de esos puntos se salgan del campo de vista, y resulte imposible estimar un tensor válido, impidiendo cumplir con la tarea de posicionamiento del robot móvil.

En general los resultados obtenidos son buenos. El robot se desplaza de manera autónoma hasta una localización (posición y orientación) muy próxima al objetivo. Sin embargo, el esquema de control que aquí se presenta utiliza una ley de control incapaz de corregir por completo el error de orientación al

final de la trayectoria del robot, dejándolo bastante cercano de la orientación deseada para evitar el caso degenerado del TT. En el siguiente capítulo se presenta un esquema de control visual, donde se introduce el uso de un objetivo virtual, con la finalidad de poder cumplir con la tarea de posicionamiento utilizando una sola ley de control, sin riesgo de que se presente un caso degenerado para el TT.

## Capítulo 6

# Control visual con control super-twisting

El esquema de control que se presenta en éste capítulo esta basado en las restricciones geométricas inducidas por el tensor trifocal, pero con una importante variante con respecto a otros esquemas, en donde se introduce el uso de un objetivo virtual. Para que el TT estimado a partir de características visuales extraídas de un conjunto de imágenes sea geoméricamente valido, las 3 imágenes incluidas en los cálculos deben ser distintas, de lo contrario se presenta un caso degenerado donde es posible obtener alguna solución numérica para el sistema de ecuaciones formado, pero que no defina el arreglo geométrico entre las posiciones de cámara.

El caso degenerado del que se hablo anteriormente se presenta comúnmente cuando el robot llega a su destino, ya que en ese momento la imagen objetivo (referencia), y la imagen actual vista por el robot serán iguales, volviendo imposible controlar el robot móvil con la información proporcionada por el tensor trifocal. En algunos casos para solucionar el problema mencionado se conmuta de una a otra ley de control cuando se presenta algún caso degenerado.

En el capítulo 5 se presentó un esquema de control visual donde el robot móvil no puede llegar exactamente al objetivo, ya que en teoría se puede corregir el error longitudinal al término de la segunda etapa, pero no se corrige completamente la orientación al final de la tercera etapa pues el robot se detiene cuando esta muy cercano de la orientación deseada, lo cual evita que se presente el caso degenerado del tensor trifocal. Con la finalidad de poder llevar al robot hasta la posición deseada sin la necesidad de utilizar dos leyes de control se opta por utilizar un objetivo virtual, de tal manera que cuando el robot llegue a su destino la imagen objetivo y la imagen actual sean distintas.

### 6.1. Modelo cinemático de un robot móvil

El vector de estados de un robot móvil se puede definir como  $\mathbf{X} = (x, z, \phi)^T$ , donde  $x$  y  $z$  son las coordenadas de la posición del robot en el plano, y  $\phi$  su orientación tal como se puede apreciar en la Figura 6.1.

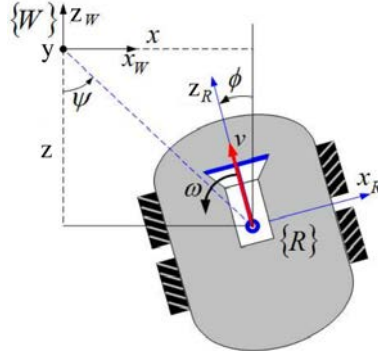


Figura 6.1: Definición del esquema del robot móvil.

En el caso en que una cámara central se monta sobre un robot móvil, de tal manera que el origen de los sistemas de referencia del robot y de la cámara coinciden, el modelo cinemático del sistema en el espacio de estados se puede expresar como:

$$\begin{bmatrix} \dot{x} \\ \dot{z} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -\sin(\phi) & 0 \\ \cos(\phi) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \nu \\ \omega \end{bmatrix} \quad (6.1)$$

Donde  $\nu$  y  $\omega$  representan las velocidades de traslación y de rotación respectivamente.

## 6.2. Planteamiento de la imagen virtual

El objetivo virtual es una imagen que contiene exactamente la misma información visual (puntos de interés) que la imagen real, pero como si la cámara se desplazara verticalmente, de tal manera que las coordenadas de los puntos se ven afectados por dicho desplazamiento. En la figura 6.2 se presenta un esquema de la imagen virtual, la cual se obtiene al desplazar  $t_y$  la imagen real.



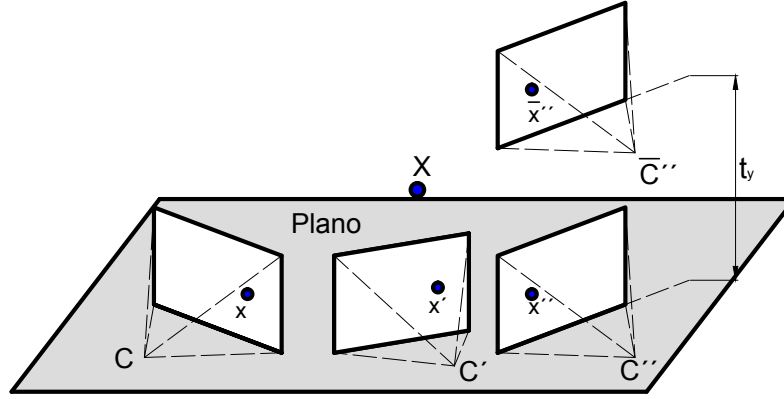


Figura 6.2: Esquema de la imagen virtual. Las 3 posiciones de cámara inicial se encuentran sobre un mismo plano.

Donde  $\bar{C}''$  representa la nueva posición de cámara.

Para generar la imagen virtual es necesario transferir información visual de la primera y segunda vista, a una tercera imagen. Utilizando los valores del tensor trifocal estimados, y las coordenadas de un punto en particular en las dos primeras imágenes, es posible realizar la transferencia de puntos utilizando la ecuación 4.11.

Al considerar un caso en el que las posiciones de cámara no se restringen a un plano, como ocurre al introducir la imagen virtual, existen otros elementos del tensor trifocal además de los presentados en la ecuación 3.35 que son no nulos, los cuales se pueden obtener a partir de la siguiente ecuación:

$$\begin{aligned}
 T_{112} &= t_{y2} \cos(\phi_1) \\
 T_{121} &= -t_{y1} \cos(\phi_2) \\
 T_{123} &= t_{y1} \sin(\phi_2) \\
 T_{132} &= -t_{y2} \sin(\phi_1) \\
 T_{222} &= t_{y2} - t_{y1} \\
 T_{312} &= t_{y2} \sin(\phi_1) \\
 T_{321} &= -t_{y1} \sin(\phi_2) \\
 T_{323} &= -t_{y1} \cos(\phi_2) \\
 T_{332} &= t_{y2} \cos(\phi_1),
 \end{aligned} \tag{6.2}$$

donde los valores de  $\phi_1$  y  $\phi_2$ , se pueden calcular de los elementos del TT estimados a partir de las imágenes reales con centros ópticos en un mismo plano, de la siguiente manera:

$$\phi_1 = \arcsin\left(\frac{T_{232}T_{313} - T_{212}T_{131}}{T_{232}T_{223} + T_{212}T_{221}}\right), \tag{6.3}$$

$$\phi_2 = \arcsin\left(\frac{T_{223}T_{131} + T_{221}T_{313}}{T_{232}T_{223} + T_{212}T_{221}}\right). \tag{6.4}$$

Por lo tanto, para generar la imagen virtual es necesario estimar los elementos del tensor utilizando la configuración inicial (esquema plano), y establecer un desplazamiento vertical para la nueva posición de cámara. Lo anterior con la finalidad de obtener los valores de los nuevos elementos a partir de la ecuación

6.2, de modo que se contemple el desplazamiento vertical para la generación de la imagen virtual, al transferir los puntos de interés compartidos por las dos primeras vistas.

### 6.3. Esquema con una sola ley de control usando el TT

Como se menciono anteriormente, algunos esquemas de control visual para poder conducir un robot móvil hasta su destino, de tal manera que se logre la posición y la orientación deseada utilizan más de una ley de control, conmutando entre ellas cuando se presentan singularidades en el control o degeneraciones en las mediciones visuales [16]. Para utilizar una sola ley de control en el esquema que se presenta, se introduce el uso de un objetivo virtual; y el control super-twisting [22] para garantizar un buen rastreo de las trayectorias de referencia.

Después de analizar la información proporcionada por el tensor trifocal utilizando la imagen virtual, se optó por escoger las siguientes salidas para el sistema:

$$\xi_1 = T_{223}, \quad \xi_2 = \frac{T_{123}}{T_{121}}. \quad (6.5)$$

Las salidas que se escogieron se pueden relacionar con el estado del sistema de la siguiente manera:

$$\xi_1 = \alpha (x_2 \sin(\phi_2) - z_2 \cos(\phi_2)), \quad \xi_2 = -\tan(\phi_2), \quad (6.6)$$

donde  $\alpha$  es un factor de escala desconocido. Las salidas del sistema se vuelven cero en el momento que el robot llega a su destino, sin embargo de la ecuación 6.6 se puede apreciar que para que ambas salidas se vuelvan cero basta con que  $\phi_2 = 0$  y  $z_2 = 0$ , siendo  $x_2$  un grado de libertad en la solución, de modo que el error longitudinal y de orientación siempre se corrigen, pero el error lateral no precisamente será cero. Como la salida  $\xi_1$  depende directamente del error longitudinal, y la salida  $\xi_2$  de la orientación, es posible corregir la desviación lateral haciendo rastreo de una trayectoria de referencia para la salida  $\xi_2$ . Introduciendo un par de trayectorias de referencia se pueden definir las siguientes funciones de error:

$$e_1 = \xi_1 - \xi_1^d, \quad e_2 = \xi_2 - \xi_2^d, \quad (6.7)$$

donde  $\xi_1^d$  y  $\xi_2^d$  son trayectorias de referencia con valor nulo al final. Como se puede notar a partir de la ecuación 6.5, las salidas del sistema están en función de los elementos del tensor trifocal, y no se relacionan directamente con las entradas del sistema ( $\nu$  y  $\omega$ ), por lo tanto no es evidente la manera de definir una ley de control para lograr la evolución deseada de las salidas del sistema. Para establecer una relación directa se realiza una linealización entrada-salida, convirtiendo el problema de control no lineal en un problema de tracking, donde es necesario definir las trayectorias a rastrear.

Para la linealización entrada-salida se requieren las derivadas respecto al tiempo de las funciones de error, y por consiguiente las derivadas respecto al tiempo de las salidas del sistema, las cuales se obtienen de la siguiente manera:

$$\begin{aligned} \dot{\xi}_1 &= \alpha (\dot{x}_2 \sin(\phi_2) + x_2 \dot{\phi}_2 \cos(\phi_2) - \dot{z}_2 \cos(\phi_2) + z_2 \dot{\phi}_2 \sin(\phi_2)), \\ \dot{\xi}_2 &= \frac{t_{y1} \dot{\phi}_2 \cos(\phi_2) T_{121} - T_{123} t_{y1} \dot{\phi}_2 \sin(\phi_2)}{T_{121}^2}. \end{aligned} \quad (6.8)$$

Empleando el modelo cinemático del robot móvil que se presenta en la ecuación 6.1, las derivadas de las salidas del sistema con respecto al tiempo se pueden expresar como:

$$\begin{aligned} \dot{\xi}_1 &= \alpha (-\nu \sin^2(\phi_2) - \nu \cos^2(\phi_2) + (x_2 \cos(\phi_2) + z_2 \sin(\phi_2)) \omega), \\ \dot{\xi}_2 &= \frac{t_{y1} \cos(\phi_2) T_{121} - T_{123} t_{y1} \sin(\phi_2)}{T_{121}^2} \omega. \end{aligned} \quad (6.9)$$

Siendo  $\alpha(x_2 \cos(\phi_2) + z_2 \sin(\phi_2)) = -T_{221}$ ,  $t_{y1} \cos(\phi_2) = -T_{121}$  y  $t_{y1} \sin(\phi_2) = -T_{123}$ , el sistema de error se puede definir como:

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \end{bmatrix} = \begin{bmatrix} -\alpha & -T_{221} \\ 0 & -\frac{T_{121}^2 + T_{123}^2}{T_{121}} \end{bmatrix} \begin{bmatrix} \nu \\ \omega \end{bmatrix} + \begin{bmatrix} \rho 1(\xi_1^d, \rho) \\ \rho 2(\xi_2^d, \rho) \end{bmatrix}. \quad (6.10)$$

En la ecuación previa  $\rho$  representa un termino de perturbación que depende de las derivadas temporales de las trayectorias de referencia, y  $\sigma$  es un parámetro de ruido en la imagen. Con la finalidad de obtener las velocidades ( $\nu$  y  $\omega$ ) necesarias para rastrear las trayectorias de referencias es necesario invertir el sistema de error, obteniendo así:

$$\begin{bmatrix} \nu \\ \omega \end{bmatrix} = \begin{bmatrix} -\frac{1}{\alpha} & \frac{T_{221}T_{121}^2}{\alpha(T_{121}^2 + T_{123}^2)} \\ 0 & -\frac{T_{121}^2}{T_{121}^2 + T_{123}^2} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \quad (6.11)$$

donde  $v_1$  y  $v_2$  son entradas auxiliares de control que definen la convergencia del error. Utilizando el algoritmo de control super-twisting el cual es un tipo específico de control por modos deslizantes, se definen las siguientes entradas auxiliares:

$$\begin{aligned} v_1 &= -k_1|e_1|^{\frac{1}{2}} \text{sign}(e_1) + \mu_1, \\ v_2 &= -k_3|e_2|^{\frac{1}{2}} \text{sign}(e_2) + \mu_2. \end{aligned} \quad (6.12)$$

Siendo:

$$\begin{aligned} \dot{\mu}_1 &= -k_2 \text{sign}(e_1), \\ \dot{\mu}_2 &= -k_4 \text{sign}(e_1), \end{aligned} \quad (6.13)$$

donde  $k_1$ ,  $k_2$ ,  $k_3$  y  $k_4$  son ganancias del control.

Las trayectorias de referencia debe conducir suavemente las salidas del sistema desde su valor inicial hasta cero. La trayectoria de referencia para la primera salida se define como:

$$\begin{aligned} \xi_1^d &= \frac{\xi(0)}{2} \left(1 + \cos\left(\frac{\pi t}{\tau}\right)\right), & 0 \leq t \leq \tau \\ \xi_1^d &= 0, & t > \tau. \end{aligned} \quad (6.14)$$

Dependiendo del valor inicial del ángulo  $\psi$ , el cual se representa esquemáticamente en la Figura 6.1, y el cual se obtiene a partir de la siguiente ecuación:

$$\psi = \arctan\left(\frac{T_{223} \sin(\phi_2) - T_{221} \cos(\phi_2)}{T_{223} \cos(\phi_2) + T_{221} \sin(\phi_2)}\right), \quad (6.15)$$

la trayectoria de referencia para la salida  $\xi_2$  se puede definir en 2 formas diferentes.

Si  $|\phi_2(0)| > |\psi_2(0)|$ :

$$\begin{aligned} \xi_2^d &= \xi_2(0) \frac{\psi(t)}{\psi(0)}, & 0 \leq t \leq \tau \\ \xi_2^d &= 0, & t > \tau \end{aligned} \quad (6.16)$$

Si  $|\phi_2(0)| < |\psi_2(0)|$ :

$$\begin{aligned} \xi_2^d &= \frac{\xi_2(0) + \xi_2^*(\tau_1)}{2} + \frac{\xi_2(0) - \xi_2^*(\tau_1)}{2} \cos\left(\frac{\pi t}{\tau_1}\right), & 0 \leq t \leq \tau_1 \\ \xi_2^d &= \xi_2(0) \frac{\psi(t)}{\psi(\tau_1)}, & \tau_1 \leq t \leq \tau \\ \xi_2^d &= 0, & t > \tau \end{aligned} \quad (6.17)$$

## 6.4. Resultados

Una de las partes más importantes dentro de éste esquema de control es la imagen virtual ya que se define como señal de referencia del sistema, y sin ella la ley de control establecida no se puede aplicar. La imagen virtual se representa como un conjunto de puntos de interés apropiadamente ubicados en el plano imagen de acuerdo a la nueva posición de cámara, ya que lo que interesa conocer son sus coordenadas. Cabe resaltar que no es necesario obtener una imagen completamente nueva donde se aprecie el desplazamiento de todos los píxeles. Con la finalidad de obtener la imagen virtual se realizaron diferentes pruebas tanto con imágenes capturadas utilizando una cámara real, como con imágenes generadas en Webots, algunos de los resultados obtenidos se presenta en las Figuras 6.3 y 6.4.



a)

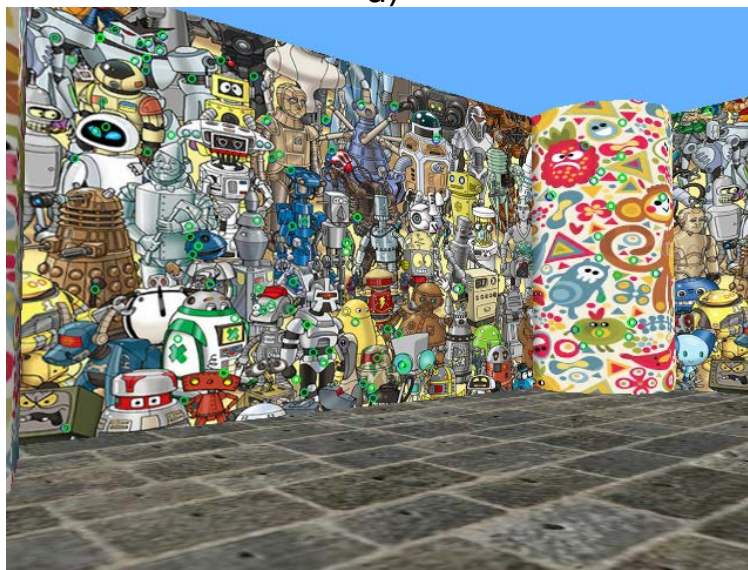


b)

Figura 6.3: Imágenes generadas transfiriendo puntos con el tensor trifocal, utilizando imágenes reales. Los puntos azules y verdes representan la imagen original y virtual respectivamente. a) Con desplazamiento vertical  $t_y = 0.2$ . b) Sin desplazamiento vertical.



a)



b)

Figura 6.4: Imágenes generadas transfiriendo puntos con el tensor trifocal utilizando imágenes de Webots. Los puntos azules y verdes representan la imagen original y virtual respectivamente. a) Con desplazamiento vertical  $t_y = 0.2$ . b) Sin desplazamiento vertical.

La imagen virtual no se pudo generar con éxito debido que a los puntos no se transfirieron apropiadamente. Las primeras pruebas aparentemente resultaron correctas como en el caso de la Figura 6.3 a), donde los puntos en general sólo presentan un desplazamiento vertical en el plano de la imagen con respecto a la posición original, sin embargo realizando otras pruebas se presentaron problemas como en el caso de la Figura 6.4 a), donde la transferencia resulto bastante ajena a lo esperado volviendo imposible estimar un tensor trifocal geoméricamente válido.

En un principio se pensó que el problema podría estar en el algoritmo implementado para transferir las características visuales, por lo cual se realizaron más pruebas pero sin considerar el desplazamiento vertical de la cámara. En este caso se obtuvieron resultados prometedores ya que visualmente los puntos transferidos se encontraban en la posición deseada (empalmados con los puntos originales), como se aprecia en las Figuras 6.3 b) y 6.4 b); El problema se volvió a presentar al comparar los elementos del tensor trifocal estimados utilizando la imagen real y los estimados con la imagen generada, los cuales deberían ser muy semejantes ya que se supone que las imágenes son iguales, sin embargo todos los elementos presentaban variaciones considerables aun cuando la diferencia entre las coordenadas de los puntos originales y los transferidos era mínima.

Para tratar de dar una explicación se realizaron pruebas con imágenes sintéticas generadas en Matlab introduciendo ruido Gaussiano, donde se manipuló el valor de la desviación estándar ( $\sigma$ ) con la finalidad de analizar el comportamiento del tensor trifocal ante la presencia de diferentes niveles de ruido en la imagen. Los resultados de algunas de las pruebas utilizando imágenes con y sin ruido se presentan en la Tabla 6.1, y a demás se compara el TT estimado utilizando las 3 imágenes reales, con el tensor calculado empleando la imagen virtual para cada uno de los casos.

Tensor trifocal con ruido en la imagen						
Elemento	$\sigma=0.0$		$\sigma=0.001$		$\sigma=0.03$	
	TT	$\overline{TT}$	TT	$\overline{TT}$	TT	$\overline{TT}$
$T_{111}$	-1.56e-07	-5.539e-07	0.0005598	0.092679	-7.0063	-0.022315
$T_{112}$	1.772e-08	-0.060546	-0.0002356	-0.043899	1.86183	-0.103751
$T_{113}$	-0.969695	-0.969755	-0.965101	0.763478	-24.8206	-1.91034
$T_{121}$	-7.393e-08	-0.060544	0.00010145	-0.046582	1.98914	0.109975
$T_{122}$	4.960e-09	1.411e-08	5.170e-06	0.008104	-0.011731	0.002680
$T_{123}$	1.1512e-07	1.6410e-05	-0.000225	0.087283	-1.36687	-0.18803
$T_{131}$	1	1.00006	0.999588	0.89882	6.00321	0.690494
$T_{132}$	7.0761e-09	-1.391e-05	0.000223	0.087627	-2.04149	0.266724
$T_{133}$	7.3538e-07	8.756e-06	-0.015763	-13.3333	192.837	2.1725
$T_{211}$	7.9950e-09	-1.867e-08	-3.508e-05	0.028467	-0.332292	-0.002965
$T_{212}$	4.7270e-08	-3.202e-05	-0.00032	0.009470	-3.7616	0.019249
$T_{213}$	2.0305e-07	-7.566e-05	-0.000661	0.060604	1.76732	-0.009432
$T_{221}$	-1.475e-07	3.142e-05	0.000893	0.005590	-3.36185	-0.041784
$T_{222}$	-4.541e-08	2.288e-07	-0.000117	-0.047259	4.0957	0.012945
$T_{223}$	-0.969695	-0.969696	0.965145	0.871841	-27.829	-15.041
$T_{231}$	-1.769e-07	7.752e-05	-0.000158	0.073754	2.98907	0.011463
$T_{232}$	1	1	1	1	1	1
$T_{233}$	-1.077e-07	6.342e-06	-0.017643	-17.0551	182.137	1.19022
$T_{311}$	5.5300e-10	-3.074e-09	-7.563e-06	-0.000615	-0.001949	-0.000931
$T_{312}$	-2.994e-10	-1.609e-07	2.4829e-06	-4.385e-05	-0.007442	0.004833
$T_{313}$	1.7679e-07	-7.909e-05	-0.001099	0.087679	-10.7932	0.054818
$T_{321}$	1.1621e-09	1.647e-07	8.3296e-07	8.3866e-05	-0.005807	-0.010298
$T_{322}$	-3.661e-10	-3.081e-09	-6.473e-07	0.000722	-0.004763	0.003330
$T_{323}$	-1.105e-07	0.060407	0.000261	-0.101558	6.37072	0.093702
$T_{331}$	-4.689e-07	7.867e-05	0.002941	0.077756	-9.6529	-0.116498
$T_{332}$	-1.214e-08	0.060409	-0.000738	-0.095986	6.04304	-0.072837
$T_{333}$	0.030309	0.030306	0.046632	4.92567	-65.7932	-0.071421

Tabla 6.1: Pruebas realizadas para estimar el tensor trifocal con diferentes niveles de ruido en la imagen.  $TT$  y  $\overline{TT}$  representan el tensor calculado con las 3 imágenes originales y el estimado introduciendo la imagen virtual respectivamente.

De los datos de la Tabla 6.1 se puede observar que el tensor trifocal es afectado incluso con niveles muy pequeños de ruido, de tal manera que los elementos del TT presentan alteraciones. Cuando la imagen esta libre de ruido el tensor trifocal calculado con las 3 imágenes de entrada, y el calculado al introducir la imagen virtual difieren únicamente en los elementos que deben cambiar. En este caso particular donde las tres cámaras se ubican de forma colineal, los elementos que deben volverse no nulos son...  $T_{112}$ ,  $T_{121}$ ,  $T_{323}$  y  $T_{332}$ , ya que el resto de los elementos que de acuerdo a la ecuación 6.2, también deben presentar un cambio al considerar un desplazamiento vertical ( $t_y$ ) en la posición de la cámara, se vuelven cero debido a que para el experimento  $\phi_1 = 0$  y  $\phi_2 = 0$ . Para casos con poco ruido en la imagen ( $\sigma = 0.001$ ) los resultados aún son buenos aunque presentan un pequeño error, sin embargo al incrementar el ruido en la imagen el tensor trifocal se ve fuertemente afectado incluso sin utilizar la imagen virtual.

Con lo anterior resulta evidente que el ruido en la imagen afecta directamente en el calculo del tensor trifocal. Al realizar el emparejamiento entre puntos siempre se presentara un pequeño error en las coordenadas, el cual puede llegar a ser tan pequeño que permite estimar apropiadamente el tensor utilizando las imágenes de entrada, pero al generar la imagen virtual se introduce más ruido ya que el tensor estimado inicialmente (para una configuración plana) presenta pequeñas variaciones con respecto al TT métrico que define de manera precisa la geometría de la escena entre cámaras, impidiendo estimar un tensor trifocal válido.

En el caso en que se introduce un desplazamiento vertical logrando un resultado visualmente bueno (puntos reales y virtuales empalmados), pero con valores del tensor completamente diferentes a lo esperado, probablemente se debe a que utilizando el proceso de mínimos cuadrados para solucionar el sistema de ecuaciones lineales, se obtiene un resultado que matemáticamente permite transferir apropiadamente las características visuales en la imagen a pesar de que no describa el arreglo geométrico entre cámaras. En la literatura se presentan algunos métodos no lineales que incluyen una etapa de optimización con algoritmos como Levenberg-Marquardt, para garantizar la obtención de un TT geoméricamente, valido a partir de puntos correctamente emparejados con RANSAC. Un ejemplo es el algoritmo Gold Standard detallado en [19]. Un algoritmo de esta naturaleza resultaría computacionalmente costoso para introducirlo en un lazo de control visual, por lo que explorar esta alternativa es un trabajo futuro que habría que optimizar en rendimiento.

La evaluación de este esquema de control se realizará únicamente con simulaciones en Matlab donde se puede trabajar con imágenes libres de ruido.

Las imágenes sintéticas utilizadas durante la simulación se generaron utilizando el modelo convencional de una cámara pinhole. Las ganancias del controlador se fijaron en valores de  $k_1 = 0.5$ ,  $k_2 = 0.002$ ,  $k_3 = 0.01$  y  $k_4 = 0.001$ .

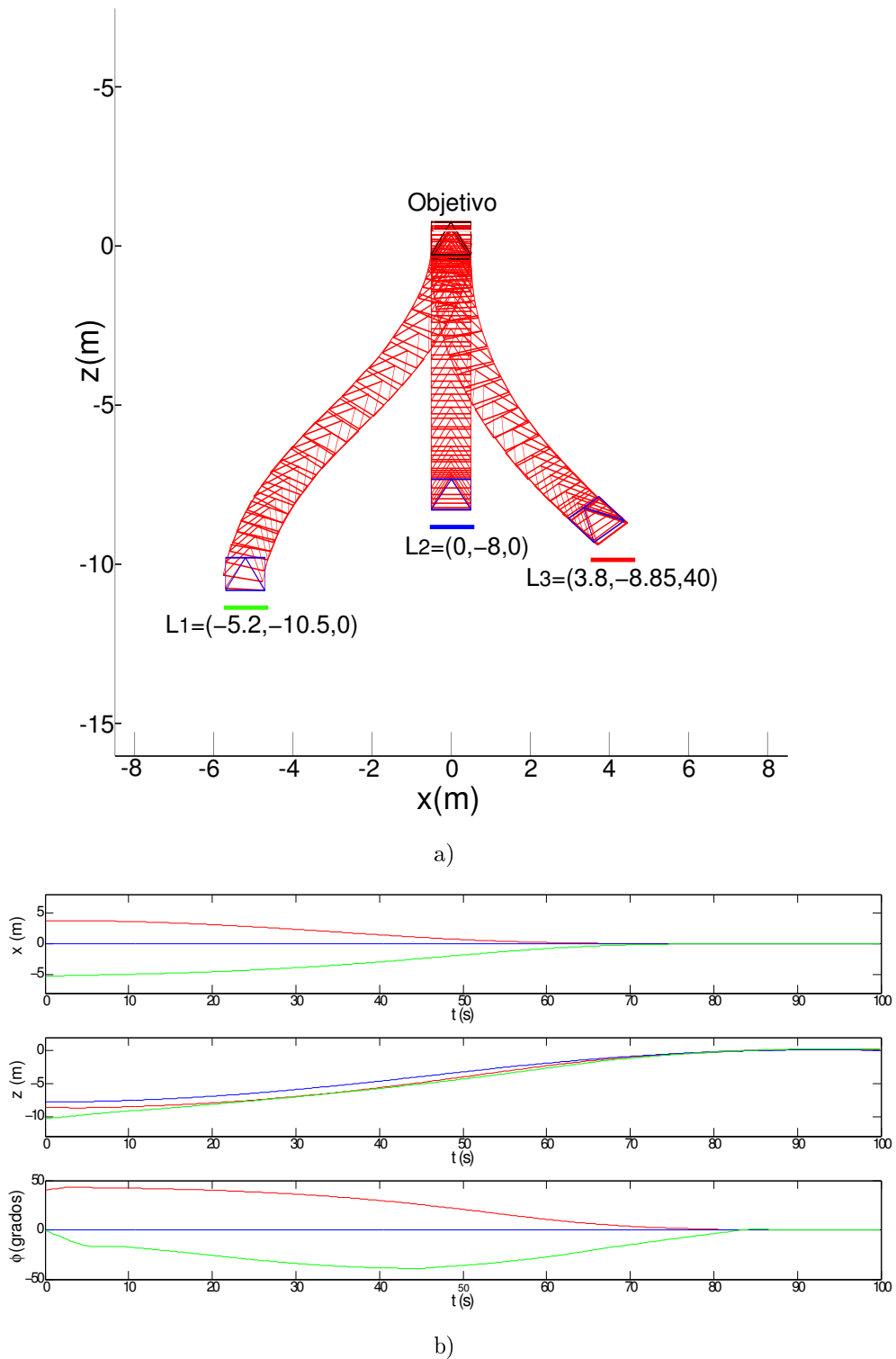
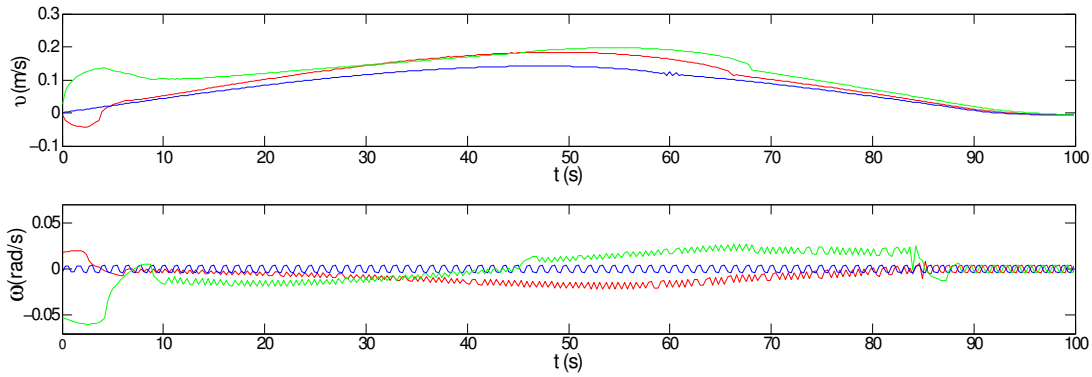


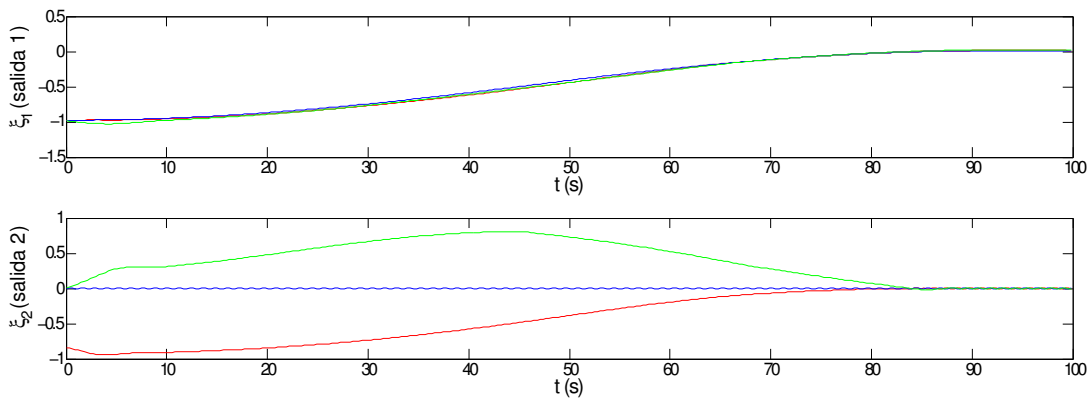
Figura 6.5: Resultados de simulaciones en Matlab. a) Trayectorias seguida por el robot móvil desde diferentes ubicaciones en el plano  $x$ - $z$ . b) Evolución de la posición y orientación del robot.



La evolución de la trayectoria del robot desde tres diferentes ubicaciones en el plano hasta su objetivo se puede apreciar en la Figura 6.5. En la simulación se presentan ejemplos de los tres posibles casos para definir la trayectoria de referencia, el primero de ellos ( $\mathbf{L}_1$ ) cuando  $|\phi_2(0)| < |\psi_2(0)|$  y la trayectoria de referencia  $\xi_2^d$  está definida por la ecuación 6.17. El segundo ( $\mathbf{L}_2$ ) es un caso especial cuando  $\phi_2(0) = \psi_2(0)$  para el cual durante todo el recorrido del robot  $\xi_2^d = 0$ . Por último en el tercer caso ( $\mathbf{L}_3$ ) cuando  $|\phi_2(0)| > |\psi_2(0)|$  la trayectoria de referencia  $\xi_2^d$  está definida por la ecuación 6.16.



a)



b)

Figura 6.6: Resultados de simulaciones en Matlab. a) Velocidades calculadas. b) Evolución de las salidas del sistema (mediciones visuales).

En la Figura 6.5 b) se muestra la evolución de la posición y orientación del robot móvil en el plano  $x$ - $z$ , de donde se puede observar que para los tres casos presentados el robots llega de manera precisa al objetivo siguiendo una trayectoria suave. Las velocidades calculadas para el desplazamiento del robot y la evolución de las salidas del sistema se muestran en la Figura 6.6.

## 6.5. Conclusiones

En este capítulo se presentó un esquema de control visual basado en imagen con un nuevo enfoque, donde se logra cumplir con la tarea de posicionamiento, utilizando las restricciones geométricas inducidas por el TT.

En el esquema de control, se introdujo el uso de un objetivo virtual generado a partir de la transferencia de puntos, utilizando el TT estimado previamente para el caso de un escenario plano, y considerando un desplazamiento vertical de la cámara. El introducir la imagen virtual evitó la presencia de singularidades en el esquema de control, permitiendo cumplir con la tarea de posicionamiento utilizando una sola ley de control, sin la necesidad de conmutar entre dos controladores diferentes, como ocurre en otros esquemas que también se basan en el TT, o en las restricciones inducidas por la geometría epipolar.

El desempeño de este esquema de control se evaluó mediante simulaciones realizadas en Matlab. Los resultados obtenidos de las simulaciones fueron satisfactorios, ya que en efecto se logra cumplir con la tarea de posicionamiento del robot móvil.

Por otro lado al trabajar con imágenes capturadas a partir de una cámara, se observó que la generación de la imagen virtual es un poco delicada, ya que el TT es bastante sensible al ruido. Como se menciona en las secciones anteriores, la transferencia de los puntos introducen ruido en la imagen, impidiendo calcular un TT estable. En trabajos futuros, se buscará introducir algún método no lineal como el algoritmo Gold Standard o la aproximación de Sampson, que permita refinar el cálculo del TT con la finalidad de poder generar la imagen virtual, e implementar el esquema de control en un robot móvil real.

# Bibliografía

- [1] H.H. Abdelkader, Y. Mezouar, and N. Andreff. Image-based Control of Mobile Robot with Central Catadioptric Cameras . *IEEE International Conference on Robotics and Automation*, pages 3522–3527, April 2005.
- [2] Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. FREAK: Fast Retina Keypoin.
- [3] ARIA. ARIA. <http://robots.mobilerobots.com/wiki/ARIA>, 2008-2014.
- [4] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-Up Robust Features. *Computer Vision and Image Understanding*, pages 346–358, 2008.
- [5] H. M. Becerra, G. Lopéz. Nicolás, and C.Sagiüés. A sliding mode control law for mobile robots based on epipolar visual servoing from three views. volume 27, pages 175–183, 2011.
- [6] S. Benhimane, E. Malis, P. Rives, and J. R. Azinheira. Vision-based control for car platooning using homography decomposition. *IEEE International Conference on Robotics and Automation*, pages 2173–2178, Abril 2005.
- [7] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF: Binary Robust Independent Elementary Features .
- [8] F. Chaumette and S. Hutchinson. Visual servo control, Part I: Basic Approaches. *IEEE Robotics and Aut. Mag*, 13(14):82–90, 2006.
- [9] F. Chaumette and S. Hutchinson. Visual servo control, Part II: Advanced Approaches . *IEEE Robotics and Aut. Mag*, 14(1):109–118, 2007.
- [10] F. Chaumette, P. Rives, and B. Espiau. Positioning of a robot with respect to an object, tracking it and estimating its velocity by visual servoing. *IEEE Intemational Conference on Robotics and Automation*, pages 2248–2253, 1991.
- [11] Rerngwut Choomuang and Nitin Afzulpurkar. Hybrid Kalman Filter/Fuzzy Logic based Position Control of Autonomous Mobile Robot. *International Journal of Advanced Robotic Systems*, 24(3):197–208, 2005.
- [12] Chris. Harris and Mike. Stephens. A combined corner and edge detector. 1998.
- [13] P. I. Corke and S. A. Hutchinson. A new partitioned approach to image-based visual servo control. *IEEE Transactions on Robotics and Automation*, 17(4):507–515, Agosto 2001.
- [14] G. N. DeSouza and A. C. Kak. Vision for mobile robot navigation: A survey . volume 24, pages 237–267, 2002.
- [15] Y. Fang, W. E. Dixon, D. M. DawsonDixon, and P. Chawda. Homography-based visual servo regulation of mobile robots . *IEEE Transactions on Systems*, pages 1041–150, 2005.
- [16] H. M. Becerra , G. López-Nicolás , and C. Sagues . Omnidirectional visual control of mobile robots based on the 1D trifocal tensor. *Robotics and Autonomous Systems 58 (6)*, pages 796–808, 2010.

- [17] Richard Hartley and Andrew Zisserman. Computation of the Trifocal Tensor . In *Multiple View Geometry in Computer Vision*, chapter 15-16, pages 365–406. Second edition, 2003.
- [18] Richard Hartley and Andrew Zisserman. Epipolar Geometry and the Fundamental Matrix. In *Multiple View Geometry in Computer Vision*, chapter 9, pages 239–259. Second edition, 2003.
- [19] Richard Hartley and Andrew Zisserman. The Gold Standard method for the trifocal tensor. In *Multiple View Geometry in Computer Vision*, chapter 16, pages 396–398. Second edition, 2003.
- [20] Richard Hartley and Andrew Zisserman. Transformation invariance and normalization . In *Multiple View Geometry in computer vision*, chapter 4, pages 104–110. Second edition, 2003.
- [21] Robert Laganière. *OpenCV 2 Computer Vision Application Programming Cookbook*. 1 edition, Mayo 2011.
- [22] A. Levant and L. Fridman. Higher order sliding modes. in: *W. Perruqueti, J. Barbot (Eds.), Sliding Mode Control in Engineering, Marcel Dekker*, pages 53–101, 2002.
- [23] G. López-Nicolás, J. Guerrero, and C. Sagüés. Visual control through the trifocal tensor for nonholonomic robots. *Robotics and Autonomous Systems*, 58(2):216–226, 2010.
- [24] D. G. Lowe. *International Journal of Computer Vision*, 60(2):91, 2004.
- [25] A. De Luca, G. Oriolo, and P. Robuffo. n-line estimation of feature depth for image-based visual servoing schemes. pages 2823–2828, 2007.
- [26] K. Lutvica, N. Kadic, and G. Dzampo. Remote position control of mobile robot based on visual feedback and ZigBee communication . *IEEE International Conference on Robotics and Automation*, pages 169–172, Sept 2011.
- [27] E. Malis, F. Chaumette, and S. Boudet. 2 1/2 D visual servoing . *IEEE Transactions on Robotics and Automation*, pages 234–246, Abril 1999.
- [28] G. L. Mariottini, G. Oriolo, and D. Prattichizzo. Image-based visual servoing for nonholonomic mobile robots using epipolar geometry . *IEEE Transactions on Robotics* 23 (1), pages 87–100, 2007.
- [29] Opencv. OpenCV installation for Ubuntu. <http://www.raben.com/book/export/html/3>.
- [30] E. Rivlin, R. Basri, and I. Shimshoni. Image-based robot navigation under the perspective model . In *IEEE International Conference on Robotics and Automation*, 4:2578–2583, Mayo 1999.
- [31] Edward Rosten and Tom Drummond. Machine learning for high speed corner detection. *9th European Conference on Computer Vision*, 1:430–443, 2006.
- [32] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R. Bradski. An efficient alternative to SIFT or SURF. *IEEE International Conference on Computer Vision*, pages 2564–2571, 2011.
- [33] F. Schramrn, G. More, A. Micaell, and A. Lottin. Extended-2D visual servoing. In *IEEE International Conference on Robotics and Automation*, pages 267–273, 2004.
- [34] A. Usai and P. D. Giamberardino. Visual feedback for nonholonomic mobile robots: Homography based approach. *G. Rigatos (Ed.), Intelligent Industrial Systems, IGI Global*, pages 152–181, 2010.