



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

PROGRAMA DE MAESTRÍA Y DOCTORADO EN
INGENIERÍA

FACULTAD DE INGENIERÍA

SISTEMA ROBÓTICO PARA AJUSTE DE
CONCENTRADORES SOLARES

T E S I S

QUE PARA OBTENER EL GRADO DE:

MAESTRO EN INGENIERÍA

MECÁNICA - MECATRÓNICA

P R E S E N T A :

JORGE LUIS RODRÍGUEZ MENDOZA



TUTOR

DR. ERNST KUSSUL

2011

JURADO ASIGNADO

Presidente: M.C José Luis Perez Silva

Secretario: Dr. Francisco Cuenca Jimenez

Vocal: Dr. Ernst Kussul

1° Suplente: Dra. Tetyana Baydyk

2° Suplente: M.I Alejandro Vega de la

Lugar donde se realizó la tesis:

CCADET, UNAM

**TUTOR DE TESIS:
DR. ERNST KUSSUL**

FIRMA

AGRADECIMIENTOS

Por la Gracia de Dios y el apoyo de aquellos que han cobijado con tanto entusiasmo cariño y respeto mi carrera, se ha podido hacer realidad este proyecto. En él deposito mi entero agradecimiento a ustedes quienes a lo largo del camino me brindaron los medios para llevar a cabo los sueños, los consejos para dirigirlos y el impulso para lograrlos.

Naturalmente, mi eterna gratitud a mi familia. Imposible resulta poner en papel el inmenso cariño que por ellos siento. Para ellos y por ellos Todo.

El más humilde y sincero reconocimiento para mis asesores, revisores y compañeros de trabajo, por el incondicional apoyo brindado y sus invaluable contribuciones. Lo que aquí se materializa no hubiese sido posible sin su ayuda. Para ustedes mi admiración y respeto.

De forma muy especial quiero extender mi gratitud a la Dra. Tetyana Baydyk, al M.C. Oscar Rendón y a la M.I. Anabel Avilés, pilares fundamentales en la realización de este proyecto, y catalizadores de mi formación académica y crecimiento personal durante este gran ciclo.

Agradezco también al Consejo Nacional de Ciencia y Tecnología por los recursos otorgados por medio de la beca para estudios de posgrado, así como a las autoridades del Centro de Ciencias Aplicadas y Desarrollo Tecnológico y de la Facultad de Ingeniería de la Universidad Nacional Autónoma de México por las facilidades prestadas para la realización de esta investigación. Así como a quienes de una forma u otra contribuyeron en este esfuerzo.

A todos ustedes...

Muchas Gracias

La situación energética actual y futura de nuestro planeta nos obliga a invertir recursos en el desarrollo de fuentes alternativas y sustentables de energía, capaces de satisfacer la elevada demanda de una creciente sociedad.

El potencial del sol como fuente de energía nos ofrece una alternativa de altísimas prestaciones, con la ventaja de ser una fuente ambientalmente amigable. Por otro lado, desarrollos tecnológicos recientes han demostrado que es factible almacenar la energía proveniente del sol para utilizarla en los periodos de no incidencia, eliminando uno de los principales motivos que frenan la adopción de este tipo de tecnología. De manera similar, se han producido adelantos que permiten no solo generar electricidad, sino también combustibles, como el hidrógeno, a partir de la concentración de la energía solar.

El grupo de investigadores del CCADET, UNAM, desarrolla actualmente un concentrador solar de superficie parabólica con espejos triangulares planos. En el proceso de fabricación del concentrador, la superficie parabólica que conforman los espejos es ajustada mediante un sistema de tuercas que determinan la posición de cada uno de ellos, optimizando los resultados de la concentración.

La producción en serie, la precisión, el aumento de la calidad global de los productos, y la disminución del costo de producción, son características de los procesos automatizados, y son, a la vez, rasgos indispensables en el desarrollo de una infraestructura funcional de abastecimiento de energía basado en fuentes renovables.

Este trabajo presenta el diseño y desarrollo de una herramienta automática de producción basada en visión artificial, para posicionar el conjunto de tuercas que determina la superficie parabólica de concentradores solares.

Este trabajo fue realizado en el Centro de Ciencias Aplicadas y Desarrollo Tecnológico de la Universidad Nacional Autónoma de México, bajo la dirección del Dr. Ernst Kussul, con el apoyo de los proyectos CONACYT 50231, ICyTDF 330/2009, PAPIIT IN110510-3 y PAPIIT IN119610.

ÍNDICE GENERAL

Introducción	1
Objetivos	10
1. Sistema robótico para ajuste de superficie parabólica de un concentrador solar	12
1.1. Planteamiento del problema	12
1.2. Requerimientos del sistema	14
1.3. Solución propuesta	15
1.4. Subsistemas	20
2. Subsistema mecánico	22
2.1. Descripción de funciones	22
2.2. Diseño y especificación de componentes	23
2.3. Diseño del engranaje	25
2.4. Diseño del mecanismo de ajuste	29
2.5. Optimización masa - rigidez de la pluma	38
3. Subsistema electrónico	41
3.1. Definición de funciones	41
3.2. Requerimientos técnicos	42
3.3. Diseño electrónico	42

3.3.1.	Selección del μ Controlador	42
3.3.2.	Diagrama de conexiones PIC18F4553	47
3.3.3.	Control de motores a pasos	48
3.3.4.	Alimentación de los motores	50
3.3.5.	Conexión de sensores	51
3.4.	Diseño de interfaz (API) con el subsistema de visión computacional	53
3.5.	Software y circuito de control	55
4.	Subsistema de visión computacional. Hardware	60
4.1.	Definición de funciones	60
4.2.	Requerimientos de hardware	61
4.3.	Diseño del contenedor estereoscópico	62
4.4.	Optimización masa - frecuencia del contenedor	66
5.	Fundamentos de visión computacional	70
5.1.	Imagen	71
5.2.	Subimágen centrada en un pixel	73
5.2.1.	Operador de ciclo (\odot)	74
5.3.	Histogramas	74
5.3.1.	Histograma de brillo	75
5.3.2.	Cálculo del histograma de brillo en subimágenes	82
5.3.3.	Histograma de contraste	84
5.3.4.	Cálculo del histograma de contraste en subimágenes	84
5.3.5.	Histograma de microcontornos	86
5.3.6.	Cálculo del histograma de microcontornos en subimágenes	87
5.4.	Fundamentos de visión estereoscópica	90
5.4.1.	Modelo de cámara pinhole	90
5.4.2.	Marcos de referencia en visión computacional	90
5.4.3.	Aberraciones del lente de la cámara	92
5.4.4.	Matriz de parámetros intrínsecos	93
5.4.5.	Matriz fundamental	96
5.4.6.	Triangulación	98

6. Subsistema de visión computacional. Desarrollo de software	101
6.1. Requerimientos de software	101
6.2. Detección e identificación de marcas de color	102
6.2.1. Eliminación de fondo y realce de pixeles de color saturado	102
6.2.2. Umbralización de la imagen en canal S	103
6.2.3. Eliminación de ruido en imagen binaria	104
6.2.4. Extracción de regiones	105
6.3. Generación de mapas de disparidad. Percepción de la tercera dimensión	108
6.3.1. Algoritmo de comparación de nivel de brillo entre pixeles	108
6.3.2. Resultados. Aplicación del algoritmo de comparación de nivel de brillo	113
6.3.3. Algoritmo de comparación de histogramas entre pixeles	115
6.3.4. Resultados. Aplicación del algoritmo de comparación de histogramas	116
6.4. Estimación de medidas del mundo real (triangulación)	119
7. Conclusiones y trabajo futuro	122
7.1. Conclusiones	122
7.2. Trabajo futuro	125
Bibliografía	126
A. Modelos de Color	133
A.1. Saturación (S)	137
A.2. Intensidad (I)	142
A.3. Tono (H)	143

ÍNDICE DE FIGURAS

1.	Transformación en el uso de las fuentes de energía [101]	2
2.	Promedio de radiación solar anual 1981-2000 [101]	4
3.	Concentrador solar de canal parabólico [33]	5
4.	Concentrador solar tipo Fresnel [33]	6
5.	Concentrador solar de campo de heliostatos [33]	6
6.	Concentrador solar de plato parabólico [33]	7
7.	Concentrador parabólico compuesto	7
1.1.	Estructura de concentrador solar de espejos triangulares	13
1.2.	Ajuste de superficie con 1) regla parabólica sobre 2) concentrador solar	13
1.3.	Ajuste de superficie por medio de 1) regla parabólica y 2) manipulador	14
1.4.	Sistema de ajuste automático	15
1.5.	Componentes del sistema de ajuste automático	16
1.6.	Marca tipo para detección de tuercas	16
1.7.	Arreglo estereoscópico de cámaras	17
1.8.	Disposición de marcas y tuercas de ajuste	17
1.9.	Circuitos montados sobre PCB para manejo de actuadores	18
1.10.	Placa móvil	18
1.11.	Actuador para controlar el giro alrededor del vástago de soporte	19
1.12.	Actuador para avance lineal del mecanismo de ajuste y del subsistema de visión	19

1.13. Mecanismo de ajuste	20
1.14. Motor con dado para ajuste de tuercas	20
1.15. Diagrama de componentes e interconexiones del sistema	21
2.1. Vástago de soporte	23
2.2. Pluma	23
2.3. Placa móvil	24
2.4. Base del mecanismo de ajuste	24
2.5. Bases para rodamientos del tornillo sin fin	25
2.6. Esquema de engranaje para transmisión de potencia del actuador q_1	26
2.7. Engranajes para transmisión de potencia del actuador q_1	29
2.8. Configuración geométrica del mecanismo de ajuste	30
2.9. Bases locales, e inercial, ángulos de rotación, y dimensiones de los elementos del mecanismo	31
2.10. Vectores de posición en base inercial	32
2.11. Mecanismos obtenidos en el proceso de diseño	35
2.12. Torque requerido para el movimiento del mecanismo	37
2.13. Mecanismo de ajuste	38
2.14. Magnitud de la deformación de la pluma	39
2.15. Comportamiento mecánico de la pluma para cada iteración	40
3.1. μ Controlador en encapsulado PDIP [61]	45
3.2. Extracto de la guía de selección de productos de Microchip [65]	46
3.3. Diagrama de terminales del PIC18F4553[63]	46
3.4. Diagrama de conexiones básicas para PIC18F4553	48
3.5. Esquema de 1) motor a pasos unipolar y 2) dispositivo de control y potencia [72]	48
3.6. Diagrama de terminales del controlador de motores a pasos L297 [81]	49
3.7. Driver UNL2803A [80]	51
3.8. Conexiones del μ Controlador para manejo de actuadores	51
3.9. Conexión del MOC70 al μ Controlador	52
3.10. Estructura de conector USB macho tipo A	53

3.11. Esquemático del subsistema electrónico	56
3.12. Subsistema electrónico. Módulo principal	57
3.13. PCB para MOC70	57
4.1. Cámara PS3 EYE	62
4.2. PCB de la cámara PS3 EYE	62
4.3. Modelo virtual de la cámara PS3 EYE: 1) micrófonos, 2) protector de micrófonos	63
4.4. Cámaras enlazadas mediante protector de micrófonos	63
4.5. Contenedor rectangular para cámaras	64
4.6. Geometría en planta del contenedor estereoscópico	64
4.7. Componente superior e inferior del contenedor: 1) ceja, 2) anclajes, 3) conducto para cables	65
4.8. Diseño del componente superior del contenedor estereoscópico	65
4.9. Frecuencias de vibración del contenedor estereoscópico para cada iteración	68
4.10. Modos de vibración principales del contenedor estereoscópico	68
4.11. Componentes del contenedor estereoscópico	69
5.1. Esquema general de una imagen digital	72
5.2. Subimagen centrada en un pixel	73
5.3. Subimagen con regiones fuera de la imagen original	76
5.4. Distribución de pixeles en ventanas ($m \times n$) con regiones fuera de la imagen original	77
5.5. Distribución de pixeles en ventanas ($m \times n$) y centro $\neq (0, 0)$	79
5.6. Límites de las áreas de restricción identificadas	80
5.7. Etiquetas de identificación para las áreas de restricción	81
5.8. Criterios para localizar una subimagen en áreas de restricción	82
5.9. Grupo de pixeles para cálculo de orientación de microcontornos	86
5.10. Modelo de cámara pinhole	90
5.11. Marcos de referencia en visión computacional	91
5.12. Disposición del plano retinal con respecto al centro óptico de la cámara	93
5.13. Geometría epipolar del arreglo estereoscópico	96
5.14. Relación entre marcos de referencia de un sistema estereoscópico	97

5.15. Triangulación de un punto en el espacio 3D	99
6.1. Diagrama de flujo del algoritmo de detección e identificación de marcas de color . .	102
6.2. Imagen con marcas de color	103
6.3. Canal S de la Fig. 6.2	103
6.4. Imágenes binarizadas	104
6.5. Imagen binaria con ruido	104
6.6. Erosión morfológica	105
6.7. Resultado de aplicar la erosión a Fig. 6.5	105
6.8. Etiquetado de píxeles	106
6.9. Reetiquetado de regiones	107
6.10. Vecinos izquierdo y derecho del pixel centroide	107
6.11. Resultado de extracción e identificación de regiones	108
6.12. Comparación de nivel de brillo entre píxeles	109
6.13. Disparidad entre píxeles correspondientes de la imagen derecha e izquierda	109
6.14. Diagrama de flujo del algoritmo de comparación de nivel de brillo	110
6.15. Algoritmo para el cálculo de la matriz de disparidad usando comparación de brillo entre píxeles	112
6.16. Par estereoscópico original (Habitación)	113
6.17. Mapas de disparidad para distintos valores del umbral t_k sobre el par “Habitación” .	114
6.18. Par estereoscópico original (Cubo)	115
6.19. Mapas de disparidad para distintos valores del umbral t_k del par “Cubo”	115
6.20. Mapas de disparidad para distintas ventanas de análisis sobre el par “Habitación”, con umbral $t_h = 8$	117
6.21. Mapas de disparidad para distintas ventanas de análisis sobre el par “Habitación”, con umbral $t_h = 16$	118
6.22. Mapas de disparidad para distintas ventanas de análisis sobre el par “Habitación”, con umbral $t_h = 32$	119
6.23. Geometría del arreglo estereoscópico canónico	120
A.1. Modelo de color RGB	134
A.2. Círculo cromático	135

A.3. Rotación del cubo RGB	135
A.4. Modelo de color HSI	136
A.5. Proyección de color RGB sobre el eje I	138
A.6. Transformación de imagen de RGB a S	141
A.7. Transformación de imagen de RGB a I	143
A.8. Discretización de círculo cromático	144
A.9. Determinación de H para punto RGB	144
A.10. Comportamiento del valor de H al variar RGB	145
A.11. Transformación de imagen de RGB a H	147

ÍNDICE DE TABLAS

1.	Características de los distintos tipos de concentradores solares [68]	8
2.1.	Valor de los criterios de selección del mecanismo de ajuste	35
3.1.	Parámetros y valores para selección de μ Controlador	45
3.2.	Secuencia de conmutación de un motor a pasos unipolar con avance de pasos completos [3]	49
3.3.	Terminales del L297 que serán utilizadas para control de motores a pasos	50
3.4.	Comandos de la API para comunicación del subsistema de visión computacional con el subsistema electrónico	55
3.5.	Comandos de la API para comunicación del subsistema electrónico con el subsistema de visión computacional	55
5.1.	Datos obtenidos de los experimentos presentados en la Fig. 5.4	78
5.2.	Valores de V_r y Sg para el cálculo de orientación de microcontornos	87
A.1.	Comparación de algoritmos para cálculo de S	140
A.2.	Valores de Rel	146
A.3.	Comparación de algoritmos para cálculo de H	147

La energía, en cualquiera de sus manifestaciones, es la fuerza fundamental del universo, responsable de la vida y de la ocurrencia de los procesos naturales a cualquier nivel. Esta característica le confiere la condición de ser el recurso natural más deseado y explotado por todos aquellos seres para los que resulta vital, y en particular, para el ser humano.

El consumo mundial de energía se ha incrementado dramáticamente desde que la revolución industrial introdujo métodos de producción mecanizados [58]. Solo en 2008, Estados Unidos consumió un total de 99.4 mil billones de BTUs¹, convirtiéndose en el consumidor de energía número uno a nivel mundial [66], con una población que representa alrededor del 4.5 % de la población de la tierra [95]. De dicha energía, aproximadamente el 40 % está relacionada con el uso y producción de electricidad [58]. Se estima que para mediados del siglo XXI las necesidades globales de energía se duplicarán, y para finales del siglo serán del triple del volumen actual [27].

Detrás de los Estados Unidos se encuentra China, que se establece como el segundo consumidor de energía, y como el primer productor de gases de efecto invernadero (SO_2 y CO_2) a nivel mundial [12]. En ambos países, los combustibles fósiles representan la principal fuente de recursos energéticos, con alrededor del 89 % de las aportaciones a la producción de energía.

El impacto sobre la salud, el modo de vida, y el clima del planeta, producto de la utilización de este tipo de carburantes, es cada vez más importante, trastocando incluso la forma en que se usa y se consume la energía [78].

El consumo per cápita de energía, combinado con el rápido crecimiento de la población mundial, arrojan una innegable conclusión, nos enfrentamos a una inevitable crisis energética [86] si no logramos modificar nuestro hábitos de consumo y las fuentes de donde extraemos la energía. El apetito energético de nuestro planeta llegará a ser insostenible [58], los crecientes precios de la energía, la disminución de la disponibilidad de las misma, y una visión ecoamigable están cambiando rápidamente el panorama energético global [27].

¹Unidad Térmica Británica (*British Thermal Unit*)

Sin embargo, paulatinamente los países alrededor del mundo han comenzado a dirigir sus esfuerzos y recursos al desarrollo de energías renovables [12], tal como lo demuestran múltiples estudios realizados [101, 21, 31, 78] para cuantificar el potencial de estas recién adoptadas fuentes.

El concepto de energía renovable está ligado a aquellas energías que se obtienen de fuentes naturales “virtualmente inagotables”, bien sea por la inmensa cantidad de energía que contienen, o porque son capaces de regenerarse por medios naturales. Sin embargo, esta idea debe ceñirse también a un uso prudente que nos acerque a la sustentabilidad [58], esto es, ajustar nuestros niveles de consumo a los niveles de producción, y que estos no sobrepasen límites después de los cuales provoquemos nuevamente daños irreparables. Es decir, las energías renovables son la respuesta a muchos de nuestros problemas futuros, sin embargo, no deben verse como la solución total, no al menos si no cambiamos nuestra tecnología y hábitos de consumo, pues de otra forma, es difícil satisfacer nuestra sed energética.

Las energías renovables también reciben el nombre de energías alternativas, pues este tipo de fuentes pueden suplir a las fuentes energéticas actuales, ya sea por su menor efecto contaminante o, fundamentalmente, por su posibilidad de renovación. Los expertos creen que este tipo de fuentes de energía dominarán el suministro energético mundial a largo plazo por una sencilla razón, no hay una alternativa diferente [20], y así lo demuestran las proyecciones realizadas por el Consejo Mundial de la Energía sobre las fuentes de energía que serán usadas para satisfacer el consumo a mediados y finales del siglo XXI (Fig. 1).

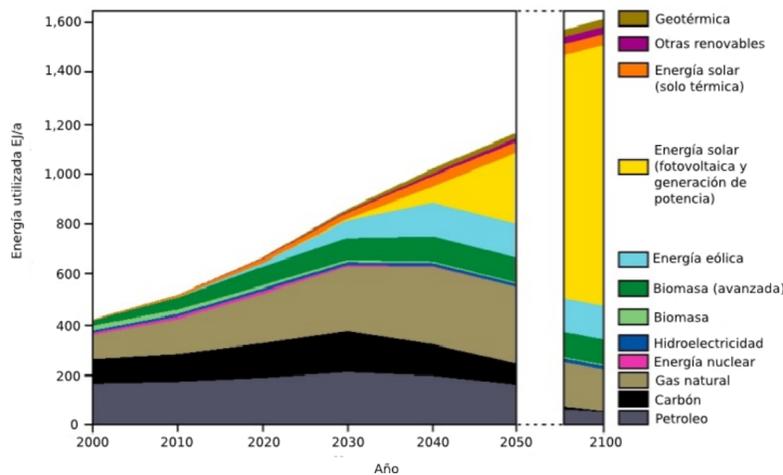


Fig. 1: Transformación en el uso de las fuentes de energía [101]

Existen múltiples fuentes de energía renovables, sin embargo, las más reconocidas son: energía eólica, hidráulica, solar, mareomotriz, geotérmica, undimotriz (olas), biocombustibles, biomasa y energía azul (confluencia de masas de agua dulce a masas de agua salada) [86]. Las energías alternativas presentan múltiples ventajas, son sostenibles, omnipresentes, ambientalmente amigables (no contaminantes y libres de carbono), seguras, una vez instaladas resultan muy económicas, la diversidad de opciones que ofrecen es muy amplia, y al contrario de las fuentes convencionales no requiere del transporte de

materias primas para su aprovechamiento [27, 58, 20, 78]. Sin embargo, existen también desventajas, las cuales han frenado su expansión masiva, estas son: la variabilidad en el suministro, un costo inicial alto, la relativamente baja densidad de producción, la necesidad de grandes áreas para su instalación, así como su estatus de tecnología en desarrollo [27, 78, 86], situaciones que poco a poco comienzan a ser atacadas de forma efectiva con las investigaciones y desarrollos más recientes.

Las fuentes alternativas de energía no son nuevas en absoluto, han estado al lado del ser humano desde los comienzos de su historia, bien sea en forma de viento, radiación solar, madera, agua o energía geotérmica, sin embargo, solo una pequeña fracción de su potencial técnico y económico ha sido explotado [20]. Las primeras civilizaciones se dieron cuenta del potencial de la energía almacenada en el agua, el sol y el viento, y la utilizaron para impulsar la navegación, el comercio y la difusión de la información [78], desafortunadamente, la utilización de estas fuentes, con excepción del agua, decreció con el paso del tiempo.

La energía solar es la fuente de suministro energético más antigua y más importante para la tierra, toda la vida del planeta depende de esta fuente, y es el punto de inicio para los procesos químicos y biológicos, es incluso la raíz de casi todos los tipos de fuentes fósiles y renovables [33, 43, 78]. Al mismo tiempo, es la fuente de energía más amigable desde el punto de vista ambiental, y puede ser utilizada en muy diversas formas, e.g., para la generación de electricidad mediante motores térmicos, calentamiento de agua para uso doméstico e industrial, producción de hidrógeno, desalinización de agua salobre, pasteurización de leche, entre otra amplia gama de oportunidades de aplicación [9, 11, 28, 85]. De esta forma es posible no solo producir energía eléctrica limpia, sino también, disminuir el nivel de consumo de energía proveniente de fuentes más agresivas con el medio.

Además, su potencial es enorme, es la fuente permanente de energía más abundante, puede ser utilizada de forma directa como radiación solar, o de forma indirecta como viento, biomasa, corrientes oceánicas, entre otras. Se estima que el sol seguirá brillando durante cinco mil millones de años más. El sol emite energía a una tasa de $3.8 \times 10^{23} \text{ kW}$; la tierra intercepta aproximadamente $1.8 \times 10^{14} \text{ kW}$, de los cuales alrededor del 60 % alcanza la superficie terrestre. Si solo el 0.1 % de esta energía pudiera ser aprovechada, con una eficiencia del 10 %, se obtendría cuatro veces la capacidad total de generación mundial de energía. El sol ofrece más energía en cuatro horas que la que la raza humana consume en un año. Dicho de otra manera, la radiación solar total anual que llega a la tierra es 7,500 veces mayor que el consumo energético anual del planeta tierra producido con fuentes de energía primaria ($\approx 450 \text{ EJ}$) [101, 33].

La Fig. 2 muestra los valores medios de radiación solar anual por metro cuadrado entre 1981 y 2000.

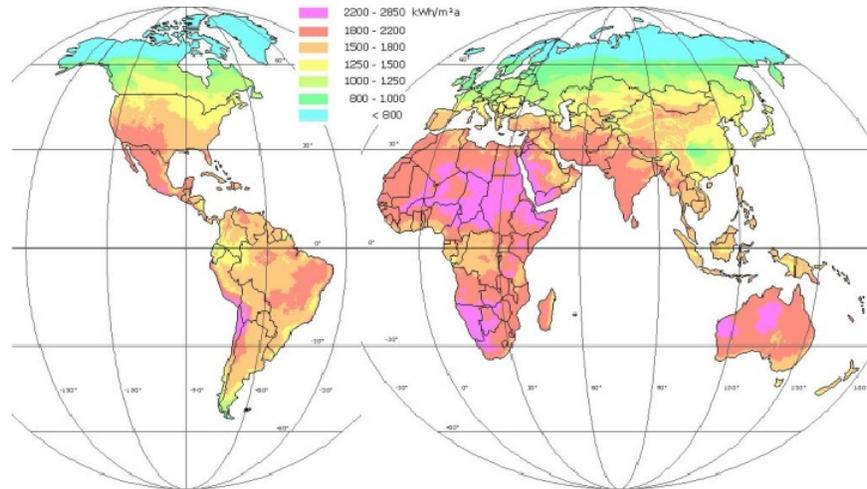


Fig. 2: Promedio de radiación solar anual 1981-2000 [101]

Para el aprovechamiento de la energía solar existen dos tecnologías dominantes, la basada en la transformación de la radiación solar en electricidad por medio del efecto fotovoltaico, y la de los colectores solares. Si bien las celdas fotovoltaicas son en algún sentido colectores solares, la literatura especializada [33, 27, 43] marca una diferencia, pues en el caso de las celdas se aprovechan los fotones incidentes, y no propiamente la acumulación del calor generado por la energía radiante como tal. Los colectores solares son un tipo especial de intercambiadores de calor, que transforman la energía radiante del sol en energía térmica. El componente más importante de cualquier sistema de energía solar es el colector [43].

Dado el intervalo de temperaturas de trabajo de los procesos termosolares, las aplicaciones más importantes son [27]:

- $T < 100\text{ }^{\circ}\text{C}$: calentamiento de agua para uso doméstico y de albercas, calefacción de edificios, destiladores y desecadores
- $100\text{ }^{\circ}\text{C} \leq T < 150\text{ }^{\circ}\text{C}$: acondicionamiento de temperatura, enfriamiento, calentamiento de agua, aceite o aire para usos industriales
- $200\text{ }^{\circ}\text{C} \leq T < 2,000\text{ }^{\circ}\text{C}$: generación de electricidad y energía mecánica
- $2,000\text{ }^{\circ}\text{C} \leq T < 5,000\text{ }^{\circ}\text{C}$: hornos para tratamientos térmicos

Existen básicamente dos tipos de colectores solares, los estacionarios (no concentradores) y los de concentración solar. Los estacionarios tienen la misma área para interceptar y absorber la radiación, son los también llamados colectores planos. Por otro lado, los colectores de concentración generalmente cuentan con una superficie de reflexión cóncava que intercepta y concentra la radiación solar en una área pequeña denominada colector [43]. Dependiendo de la geometría que los define, existen cinco tipos principales de concentradores solares:

Concentrador de canal parabólico. Se conforma por un arreglo de espejos en forma cilíndrico parabólica. En el eje focal se coloca un colector tubular que recorre el largo del cuerpo del concentrador (Fig. 3). Los haces de luz que llegan al concentrador se reflejan sobre la superficie del tubo, que los absorbe calentando el líquido que se hace circular por este. Se caracterizan por poseer mecanismos de seguimiento solar de un solo eje, generar temperaturas de trabajo superiores a los 400°C [18], y un factor de concentración de entre 30 y 100 soles [68].



Fig. 3: Concentrador solar de canal parabólico [33]

Concentradores tipo Fresnel. Los reflectores lineales tipo Fresnel actúan como un canal parabólico fragmentado (Fig. 4), son fabricados con espejos planos ensamblados de forma que asemejan la superficie del canal. Los espejos planos que se utilizan resultan más económicos que los espejos cóncavos, además, al discretizar la superficie reflejante, se reduce el monto de material necesario para su construcción [27]. A mayor número de espejos, se consigue una mejor aproximación óptica de la superficie que se intenta recrear, sin embargo, a causa de las discontinuidades se produce una pérdida de eficiencia con respecto a los concentradores de canal parabólico. El colector está separado de los reflectores y es estacionario, lo que reduce los costos en el sistema de seguimiento. Sin embargo, la complejidad del sistema de control aumenta pues se debe manejar un mayor número de interfaces con actuadores, hecho que ha ocasionado que esta tecnología no se haya difundido a gran escala [68]. El intervalo de temperatura de trabajo de este tipo de concentradores inicia en los 150°C , sin embargo, desarrollos recientes han alcanzado los 300°C [24], mientras que su factor de concentración se mantiene alrededor de 30.



Fig. 4: Concentrador solar tipo Fresnel [33]

Concentrador de campo de helióstatos. También conocido como torre solar, consiste en un arreglo de espejos tipo helióstato, capaces de girar alrededor de dos ejes para seguir el movimiento del astro, y reflejar los rayos que sobre ellos inciden a un único colector, ubicado en la cima de la torre de concentración. La torre se localiza en el punto central del campo de espejos (Fig. 5). Los espejos reflectores pueden ser planos o ligeramente cóncavos. En operación, se deben llevar a cabo ajustes constantes de la orientación de cada uno de los espejos para mantener la temperatura en el colector, en el cual se utiliza generalmente vapor de agua o sales fundidas. Su temperatura de trabajo es superior a los 500°C , alcanzando valores máximos de hasta $1,750^{\circ}\text{C}$ [42]. El intervalo típico de los valores del factor de concentración de este tipo de desarrollos ronda entre 300 y 1,500 soles [19, 42]. Sus aplicaciones son solo a nivel industrial, debido a la gran cantidad de superficie que requieren para su construcción y operación.



Fig. 5: Concentrador solar de campo de helióstatos [33]

Concentrador de plato parabólico. Se compone de un espejo cóncavo con el perfil revolucionado de una parábola que concentra la luz solar en un solo punto (Fig. 6), denominado colector o receptor. El espejo puede estar formado por un solo cuerpo, o bien, por un conjunto de espejos planos. El tipo más común de colector consiste en un motor tipo Stirling [44, 68] acoplado

en el foco de la parábola, sin embargo, también se encuentran aquellos que integran tuberías, microturbinas, y otro tipo de motores. Técnicamente, los concentradores de plato parabólico tienen el mayor potencial de aprovechamiento; un concentrador de plato parabólico ostenta el récord mundial de eficiencia (31.25 %) en la conversión de energía solar a eléctrica [91]. Con un mecanismo de seguimiento de dos ejes, los concentradores de plato parabólico pueden alcanzar eficiencias ópticas de más del 94 %, así como temperaturas superiores a los 1,000°C, y factores de concentración entre 500 y 2,000 [68].



Fig. 6: Concentrador solar de plato parabólico [33]

Concentrador parabólico compuesto. Se componen de dos reflectores parabólicos truncados (Fig. 7). A diferencia de los concentradores de plato y canal parabólico, no requieren que los rayos solares sean paralelos al eje del concentrador [27], lo que hace posible captar radiación solar con ángulos de aproximación más amplios [68]. Las dos parábolas que lo conforman son simétricas con respecto al eje del concentrador. La principal ventaja de este tipo de concentradores es que pueden alcanzar siempre algún grado de concentración sin necesidades de mecanismos de seguimiento, sin embargo, el factor de concentración se puede ver severamente castigado. Existen desarrollos alternativos que integran una estructura de espejos Fresnel en este tipo de concentradores, con los que es posible alcanzar temperaturas por encima de los 200°C [68], sin embargo, esto sacrifica en algunos grados el ángulo de aceptación con respecto a la estructura original. Su factor de concentración está en el intervalo de 3 a 20 soles.

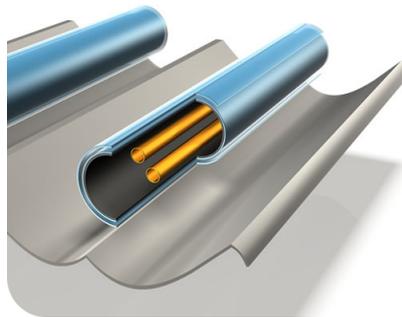


Fig. 7: Concentrador parabólico compuesto

La Tabla 1 presenta un resumen de las características de cada uno de los principales tipos de concentradores solares. Es posible apreciar que los concentradores de plato parabólico y los de campos de helióstatos presentan la mayor eficiencia, y el intervalo de temperaturas que pueden alcanzar es muy amplio, posibilitándolos para una amplia gama de tareas.

Tabla 1: Características de los distintos tipos de concentradores solares [68]

Característica	Canal Parabólico		Campo de helióstatos			Tipo Fresnel		Plato Parabólico	Parabólico compuesto		
	Aceite Sintético	Vapor	Colector salino	Vapor de agua	Volumétrico	Compacto lineal	Lineal	Espejo	Lineal	Con espejos Fresnel	
Eficiencia											
Eficiencia de conversión ideal	33%	Mayor	45%		Mayor	25%	Menor	65%	Menor	22%	
Eficiencia del colector	63%		72%			36%		66%	36%		
Temperatura de estancamiento (°C)	600	Mayor	1,750		ND	≈ 300	ND	1,200	ND		
Eficiencia óptica	80%		Varía		73%	67%	Menor	94%	ND	60% - 65%	
Factor de concentración	30 - 100		300 - 1,500		Menor	≈ 30		500 - 2,000	3	10 - 20	
Eficiencia de captura	91%		Varía		ND	76%	Menor	100%	ND	40% - 50%	
Ángulo medio de aceptación	0.5°		ND			0.75°		0.4°	20	3	
Compatibilidad de trabajo con fluidos											
Tolerancia a la presión (bar)	40 - 100		≈ 100		10 - 20	69		20	ND		
Tolerancia a la temperatura (°C)	100 - 140	Mayor	150 - 800		≈ 1,000	100 - 300		500 - 1500	< 100	< 200	
Disponibilidad	Uso de tecnología estándar		Medio - Bajo	Medio	Medio - Bajo	Medio	Medio - Bajo	Alto	Muy Bajo	Alto	Medio - Alto
Asequibilidad											
Costo de instalación (\$/kW)	3,972	2,300	> 4,000			ND	ND	12,578	ND		
Costo de mantenimiento y operación (\$/kWh)	0.012 - 0.02	Menor	0.034			Menor	Mucho menor	0.21	ND		
Uso de recursos											
Uso de suelo (m ² /MWh/año)	3.2	Menor	4.6			1.8	Mayor	4.15	ND		
Tolerancia al desnivel	< 1°		Flexible			< 1°		Flexible	Nivelado		
Escalabilidad											
Eficiencia a diferentes escalas	Mejor		Peor			Mejor		Mejor	Mejor		
Rango de operación eléctrico (MW)			0.05 - 100					0.025 - 100	ND		

Como desventajas, el campo de helióstatos ocupa grandes áreas, descartándolo así como una opción para uso residencial o de inversión individual, mientras que los concentradores de plato parabólico de espejo cóncavo presentan un alto costo inicial, lo que refrena su adopción en masa. Sin embargo, existen desarrollos actuales [45, 46, 47, 48, 49, 50, 51, 52] que han conseguido excelentes resultados en el cometido de crear concentradores parabólicos de bajo costo.

Como se ha mencionado antes, la gama de aplicaciones para la tecnología de concentración solar es muy amplia, desde el calentamiento de agua, hasta la producción de electricidad [44] o combustibles como el hidrógeno, pasando por la desalinización de agua; todo depende de los implementos colocados en el colector. El aspecto clave no es solo cómo aprovechar la energía térmica captada, sino también, aumentar la eficiencia con que ésta se obtiene, proponiendo nuevos diseños de concentradores. Algunas de las desventajas que relentizan la adopción masiva de este tipo de tecnologías han sido abordadas y solucionadas con éxito, tal es el caso de la capacidad de almacenar la energía mediante sales fundidas [39], para no suspender el servicio durante los periodos de no insolación, resolviendo así una de las principales desventajas de este tipo de sistemas.

El grupo de investigadores del CCADET, UNAM, desarrolla actualmente un concentrador solar de

superficie parabólica con espejos triangulares planos [45, 47, 49, 51, 52]. La superficie parabólica del concentrador es ajustada mediante un sistema de tuercas que determinan la posición de cada espejo, optimizando los resultados de la concentración.

La producción en serie, la precisión, el aumento de la calidad global de los productos, y la disminución del costo de producción, son características de los procesos automatizados, y son a la vez, rasgos indispensables en el desarrollo de una infraestructura funcional de abastecimiento de energía basado en fuentes renovables.

El diseño y desarrollo de una herramienta robótica de producción basada en visión artificial, para posicionar el conjunto de tuercas que determina la superficie parabólica de concentradores solares, se presenta en este trabajo.

General

Diseñar un sistema automático para el ajuste de la superficie parabólica reflejante de concentradores solares de espejos triangulares, basado en visión estereoscópica.

Particulares

1. Diseñar un subsistema de soporte y actuación mecánico capaz de acceder de forma automática (una vez colocado sobre el concentrador) a cada una de la tuercas que determinan la posición de los espejos que forman la superficie parabólica del concentrador.
2. Especificar dimensionalmente cada uno de los componentes necesarios para el funcionamiento del subsistema mecánico, así como sus relaciones de ensamble.
3. Diseñar los circuitos electrónicos, de señales y potencia, necesarios para servir de interfaz entre el subsistema de visión computacional y el subsistema mecánico, accionando de manera controlada los actuadores del sistema.
4. Diseñar un conjunto de instrucciones (API²) para la transmisión de las necesidades de posicionamiento y acción del sistema global, desde el subsistema de visión computacional a los circuitos electrónicos, para el manejo de actuadores.
5. Diseñar un contenedor rígido de soporte y ajuste para albergar al hardware del subsistema de visión computacional. El contenedor debe facilitar el montaje del arreglo de cámaras evitando el contacto directo con estas.

²Interfaz de programación (*Application Programming Interface*)

OBJETIVOS

6. Diseñar e implementar un algoritmo que calcule el mapa de disparidad entre imágenes estereoscópicas, y determine las coordenadas métricas tridimensionales de los puntos en la escena.
7. Desarrollar una aplicación de visión computacional capaz de detectar e identificar las tuercas de ajuste sobre la estructura del concentrador.
8. Generar una estructura modular en el desarrollo del software de visión computacional, dotando al sistema con la capacidad de crecimiento y escalabilidad, basado en la comunicación intermódulos.

CAPÍTULO 1

SISTEMA ROBÓTICO PARA AJUSTE DE SUPERFICIE PARABÓLICA DE UN CONCENTRADOR SOLAR

El verdadero progreso es el que pone la tecnología al alcance de todos

[HENRY FORD]

1.1. Planteamiento del problema

La situación energética actual y futura de nuestro planeta [56, 67, 88] nos obliga a invertir recursos en el desarrollo de fuentes alternativas y sustentables de energía. El potencial del sol como recurso energético [31, 88] nos ofrece una alternativa de muy altas prestaciones.

Actualmente, el grupo de investigadores del CCADET desarrolla un concentrador solar parabólico de espejos triangulares planos [45, 46, 47, 48, 49, 50, 51, 52], montados sobre una estructura de micronodos (Fig. 1.1), que reduce dramáticamente el peso del dispositivo y que busca, en conjunto con la utilización de espejos planos, reducir el costo de fabricación, como un esfuerzo para hacer proliferar este tipo de tecnologías, bien sea para la producción de electricidad a escala industrial, o para el suministro eléctrico a nivel doméstico.

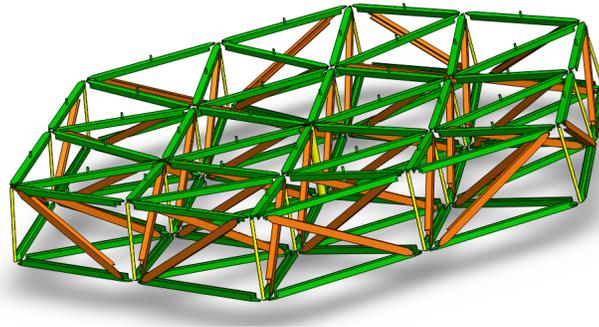


Fig. 1.1: Estructura de concentrador solar de espejos triangulares

La estructura del concentrador ha sido diseñada de forma modular, lo que permite agregar o disminuir módulos para obtener la superficie de concentración necesaria para cada aplicación. Los espejos son ensamblados sobre la estructura del concentrador de forma que se logre una superficie parabólica para la concentración. La superficie se ajusta fijando cada uno de los espejos mediante un sistema de tuercas que determinan su posición con respecto a la estructura principal del concentrador.

Para llevar a cabo el ajuste de posición de los espejos fue propuesta una regla de perfil parabólico (Fig. 1.2), que se coloca sobre la estructura del concentrador. La regla determina la posición que debe adoptar cada una de las tuercas de ajuste para formar la superficie parabólica. El proceso es completamente manual, volviéndose una tarea lenta y laboriosa, haciéndola inviable frente a los objetivos que se persiguen.

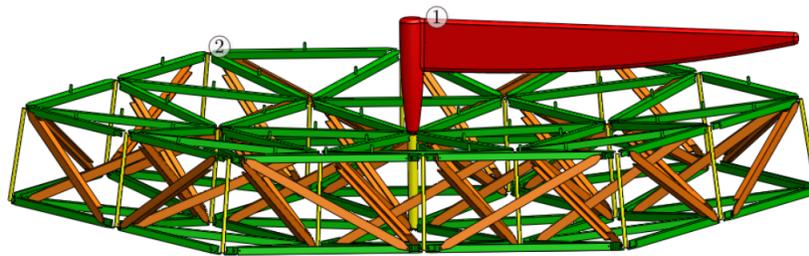


Fig. 1.2: Ajuste de superficie con 1) regla parabólica sobre 2) concentrador solar

La gran cantidad de tuercas que deben ser ajustadas para formar la superficie parabólica hacen necesario desarrollar métodos de ensamble automático que reduzcan el tiempo, simplifiquen la tarea y garanticen el grado de precisión necesaria. Además, la automatización de este proceso contribuye en mantener un bajo costo en el producto final, y optimiza los resultados de la concentración, pues de la precisión con que la superficie sea ajustada depende la eficiencia conseguida en ésta. Así mismo, al modificar de forma precisa la curvatura de la superficie reflejante, es posible modificar la posición del colector o la zona de concentración.

Con el fin de solucionar dicha problemática fue propuesto un segundo método de ajuste, esta vez

automatizado. La propuesta contemplaba nuevamente el uso de una regla parabólica, pero el ajuste se hacía por medio de un manipulador (Fig. 1.3), llevando la tuerca hasta la posición de contacto con la regla. El uso de un manipulador proporciona ventajas que inciden directamente en el tiempo de ajuste de la superficie, sin embargo, también genera desventajas que se relacionan con problemas de acceso para alcanzar todos los componentes, así como colisiones entre los dispositivos involucrados, lo cual complica la operación y el diseño del manipulador.

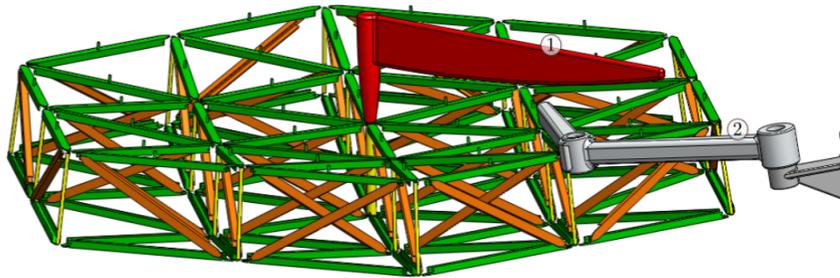


Fig. 1.3: Ajuste de superficie por medio de 1) regla parabólica y 2) manipulador

De esta forma queda clara la necesidad de automatizar de forma eficiente el proceso de ajuste de superficie del concentrador solar durante su fabricación, superando las desventajas que las propuestas anteriores presentan, dotando al diseño de la posibilidad de producirse en serie, así como de un mayor control sobre el ajuste de la superficie de concentración, potenciando de ésta forma el desarrollo de una infraestructura funcional y económica de abastecimiento de energía basado en fuentes renovables.

1.2. Requerimientos del sistema

Dada la naturaleza del problema a resolver, y la configuración geométrica del concentrador sobre el que se trabaja, el sistema a diseñar debe contar con ciertas características fundamentales que le brinden la capacidad de llevar a cabo la tarea planeada. Estas características se presentan como los requerimientos del sistema.

La tarea del sistema robótico a diseñar es ajustar de forma automática la posición de cada una de las tuercas que soportan los espejos que conforman la superficie reflejante del concentrador. Es por ello que se requiere que el sistema, una vez colocado en su posición de trabajo sobre el concentrador, sea capaz de:

- Localizar el conjunto de tuercas que deben ser ajustadas, e identificar cada una de ellas dentro de la configuración global, para poder planear el desarrollo de la tarea.
- Determinar la posición tridimensional de cada tuerca en la secuencia de ajuste, con respecto a un marco de referencia conveniente.

- Posicionarse sobre cada una de las tuercas de ajuste de superficie del concentrador, y acceder a ellas para llevar a cabo el ajuste de posición necesario.
- Ajustar de forma precisa cada una de las tuercas.
- Comprobar, una vez hecho el ajuste, que la nueva posición de la tuerca sea la adecuada.
- Realizar el trabajo anteriormente descrito de forma automática.

1.3. Solución propuesta

El diseño es el despliegue racional de la creatividad, el ingenio y la intuición, aplicadas a idear una solución para resolver un problema, o para satisfacer una necesidad. Un factor fundamental para un buen diseño es tener clara la tarea que se espera lleve a cabo el sistema a diseñar. Una especificación concreta de lo que dicho sistema debe realizar, y sobre todo, de lo que no se espera que realice, permiten dedicar todos los recursos al desarrollo de una mejor solución, evitando distraerse de la causa principal de la solución requerida.

Con base en los requerimientos descritos en la sección anterior, y aprovechando la estructura del concentrador sobre el que se trabaja, se ha diseñado un sistema robótico para el ajuste de la superficie parabólica del concentrador solar que utiliza, como medio de adquisición de información, procesos de visión computacional (Fig. 1.4).



Fig. 1.4: Sistema de ajuste automático

La solución planteada se compone de dos macrocomponentes principales: un vástago de soporte, y una canastilla de actuación, denominada pluma (Fig. 1.5), que alberga los elementos que hacen posible el proceso de ajuste automático. El vástago de soporte se encarga de elevar la pluma sobre el cuerpo del

concentrador, para que a esta le sea posible acceder al conjunto de tuercas que conforman la superficie parabólica.

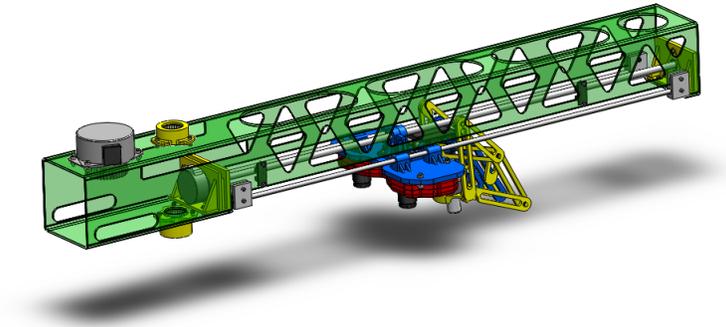


Fig. 1.5: Componentes del sistema de ajuste automático

Para facilitar la detección de las tuercas, se han diseñado marcas especiales de colores (Fig. 1.6) que se colocan sobre cada una de ellas. Un arreglo estereoscópico de cámaras (Fig. 1.7), que conforma el hardware del subsistema de visión computacional, filtra las imágenes recibidas mediante la implementación de un algoritmo de búsqueda que detecta las marcas adheridas a las tuercas de ajuste y, al compararlas con la configuración global conocida (Fig. 1.8), determina la posición actual del sistema de ajuste.

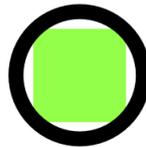


Fig. 1.6: Marca tipo para detección de tuercas



Fig. 1.7: Arreglo estereoscópico de cámaras

Del mismo modo, se determina la posición objetivo a la que el sistema debe desplazarse para llevar a cabo el ajuste de las tuercas. Para realizar dicha tarea, se sigue una secuencia de ajuste predefinida, que depende del punto inicial de trabajo.

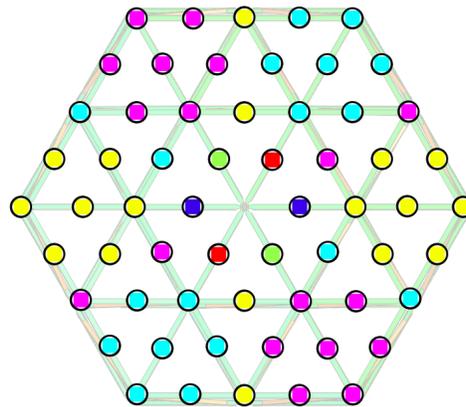


Fig. 1.8: Disposición de marcas y tuercas de ajuste

Las ordenes generadas por el subsistema de visión computacional para llevar a cabo el desplazamiento se transmiten al subsistema electrónico de manejo de actuadores (Fig. 1.9), mediante un conjunto de instrucciones desarrolladas para tal fin. El subsistema electrónico se encuentra montado en la pluma.

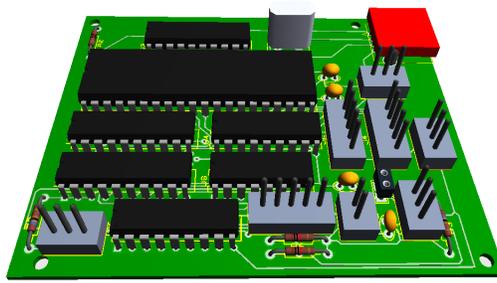


Fig. 1.9: Circuitos montados sobre PCB para manejo de actuadores

Una vez alcanzada la posición deseada sobre la tuerca por ajustar, el subsistema de visión computacional genera un mapa de profundidad de la zona captada en el campo visual, y determina la posición espacial, con respecto a un marco de referencia conveniente, de la tuerca en cuestión. A partir de la información reunida, se ordena al mecanismo de ajuste descender y manipular la tuerca para llevarla a la posición deseada. La posición final de todas las tuercas de ajuste de superficie es un dato de entrada, y dependen de la curvatura que se desea obtener en la superficie reflejante.

A la parte inferior de la pluma se encuentra adosada una placa móvil (Fig. 1.10) donde se alojan el soporte físico del subsistema de visión y el mecanismo de ajuste.

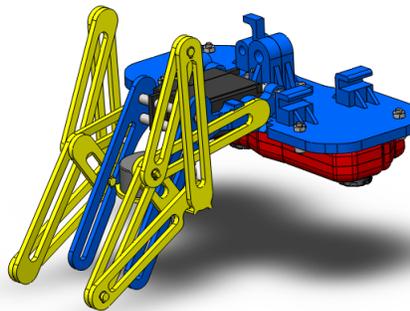


Fig. 1.10: Placa móvil

El sistema global cuenta con cuatro grados de libertad (GDL) actuados, distribuidos entre la pluma y el mecanismo de ajuste. La pluma cuenta con dos GDL (q_1 y q_2) mediante los cuales se lleva a cabo el posicionamiento en coordenadas polares del subsistema de visión computacional y del mecanismo de ajuste.

Un motor acoplado a un engranaje controla el giro (q_1) alrededor del vástago de soporte (Fig. 1.11). Un segundo motor, unido a un tornillo sin fin (Fig. 1.12), controla el avance lineal (q_2) de la placa de soporte del subsistema de visión computacional y del mecanismo de ajuste.

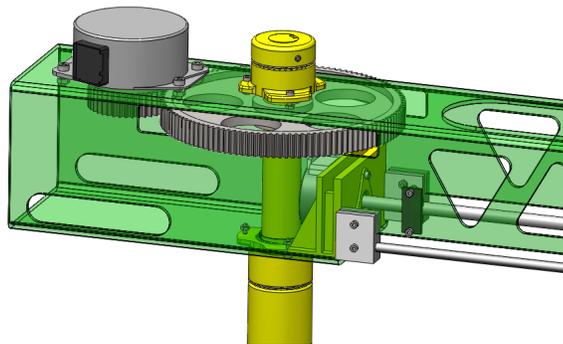


Fig. 1.11: Actuador para controlar el giro alrededor del vástago de soporte

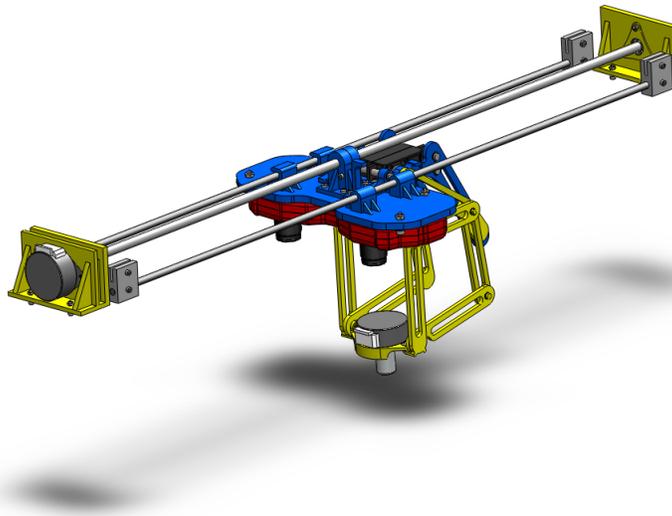


Fig. 1.12: Actuador para avance lineal del mecanismo de ajuste y del subsistema de visión

Los dos GDL restantes (q_3 y q_4) se encuentran en el mecanismo de ajuste, el cuál ha sido diseñado como un mecanismo de barras plegable (Fig. 1.13) de 1 GDL, que realiza su movimiento controlado por el giro (q_3) de un tercer motor. El mecanismo es capaz de desplegarse desde la pluma y llevar su extremo actuante hasta la tuerca que debe ser reposicionada.



Fig. 1.13: Mecanismo de ajuste

En el extremo del mecanismo descrito se encuentra el cuarto actuador del sistema, al que se acopla un dado para llevar a cabo el ajuste de las tuercas (Fig. 1.14). El mecanismo se acopla a la tuerca que se desea ajustar y, mediante el giro (q_4) de dicho motor, se desplaza la tuerca hasta alcanzar la elevación deseada. El sistema corrige el error en la posición final de la tuerca mediante retroalimentación visual.

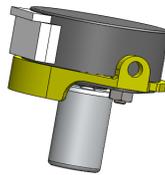


Fig. 1.14: Motor con dado para ajuste de tuercas

1.4. Subsistemas

En el desarrollo de una solución para la automatización del proceso de ajuste de superficie parabólica se ha adoptado un enfoque modular. Esta orientación nos permite contemplar el sistema como un conjunto de subsistemas interconectados, cuyo trabajo simultáneo garantiza el cumplimiento de la meta establecida.

La definición de las relaciones de interconexión entre componentes necesarias para asegurar la operación conjunta esclarece los protocolos de comunicación que deben ser diseñados. Por otro lado, la propiedad de independencia impuesta a los módulos define claramente la línea divisoria entre estos, y permite enfocar el esfuerzo de diseño de forma individual a la especialización de las funciones respectivas, y a la optimización de cada uno de ellos.

La Fig. 1.15 muestra el diagrama de los componentes que constituyen el sistema, su división modular, y las conexiones entre estos.

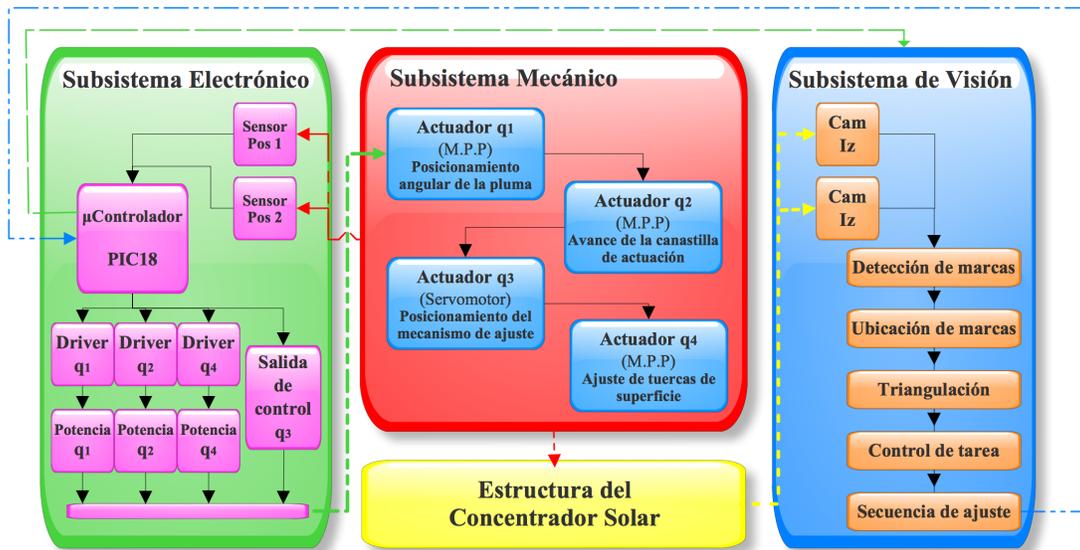


Fig. 1.15: Diagrama de componentes e interconexiones del sistema

En los capítulos siguientes se presenta la definición de las funciones que cada uno de los módulos debe llevar a cabo, así como el diseño particular de los componentes que los constituyen. Los módulos o subsistemas definidos son:

Subsistema Mecánico. Conformar la estructura física que otorga soporte y locomoción al sistema.

Forman parte de este subsistema los actuadores y sus elementos de acoplamiento, la plancha de soporte, el cuerpo de la pluma, el vástago de soporte, y el mecanismo de ajuste. El módulo mecánico alberga también los elementos materiales que conforman el resto de los subsistemas, con excepción de la unidad de procesamiento del subsistema de visión computacional.

Subsistema Electrónico. Constituido por los elementos electrónicos y el conjunto de instrucciones almacenadas en ellos (software), que se encarga de manejar los actuadores e interpretar la información de los sensores, así como por la tarjeta de circuito impreso (PCB) que los aloja e interconecta. Este subsistema se divide en una etapa de control o de señales, y una etapa de potencia.

Subsistema de Visión computacional. Este subsistema se conforma por elementos de hardware y software. El hardware está constituido por los sensores de visión en arreglo estereoscópico (que dotan al sistema de la capacidad de percepción tridimensional), el contenedor de las cámaras o soporte físico, y la unidad de procesamiento (PC). En esta última se ejecutan las implementaciones de los algoritmos (software) de visión artificial, planeación de tareas, y envío de ordenes para su realización.

CAPÍTULO 2

SUBSISTEMA MECÁNICO

*Soy de un mundo que ha podido desprender del suelo minerales,
para construir el sueño de ser aves y volar*

[JESÚS ECHEVERRÍA]

2.1. Descripción de funciones

Una vez ideada una solución, es necesario especificar las funciones concretas que llevará a cabo cada uno de los módulos que la integran, haciendo posible efectuar el diseño geométrico, dimensional, y funcional que dé sustento al proyecto, en vías de llevarlo a la realidad.

Las funciones del subsistema mecánico son:

1. Elevar y soportar la canastilla de actuación en una posición conveniente sobre la estructura del concentrador, de forma que se garantice la viabilidad del ajuste automático.
2. Brindar la estructura mecánica, tanto los elementos como el acoplamiento entre estos, necesaria para posicionar el mecanismo de ajuste y el subsistema de visión computacional sobre cada una de las tuercas que deben ser ajustadas.
3. Proveer de un mecanismo de ajuste de tuercas, así como los medios para su operación.
4. Reducir las vibraciones producidas por los actuadores mediante la integración de elementos de amortiguamiento.
5. Servir como contenedor y protector de los elementos materiales del resto de módulos que integran el sistema completo, con excepción de la unidad de procesamiento.

2.2. Diseño y especificación de componentes

La estructura general del sistema, así como la interacción y función de sus distintos componentes, se ha expuesto en la Sec. 1.3. En ésta sección se presentan las especificaciones de dimensión y forma de los componentes principales del subsistema mecánico.

El material de todos los componentes, con excepción de aquellos prefabricados, es aluminio aleación 6063-T5. El vástago de soporte (Fig. 2.1) tiene una longitud total de 429.5 mm , con diámetro de $\phi_1 = 19.05\text{ mm}$ en los 126 mm superiores, enseguida de los cuales se crea una zona de transición suave a lo largo de 180 mm que lleva el diámetro hasta los $\phi_2 = 12.7\text{ mm}$. El tramo final de 125.5 mm , con diámetro $\phi_3 = 10\text{ mm}$, se aloja dentro del tubo central de la estructura del concentrador, brindando así el soporte necesario a la pluma.



Fig. 2.1: Vástago de soporte

El cuerpo principal de la pluma de actuación (Fig. 2.2) se conforma por un perfil tubular cuadrangular de 76.2 mm por lado con esquinas redondeadas, y una longitud total de 730 mm . Para el libre tránsito de la placa móvil que contiene el mecanismo de ajuste y el hardware del subsistema de visión computacional, se ha retirado el material de la base del perfil, así mismo se realizan cortes en los costados para alojar los ejes de 4.76 mm de diámetro que guían el desplazamiento de la placa móvil. Los ejes se acoplan al cuerpo de la pluma mediante dos piezas prismáticas de aluminio, sujetas por tornillos M3 de cabeza hueca hexagonal, que son el estándar utilizado en el desarrollo. En los límites de las guías de la pluma se han colocado dos sensores tipo optointerruptor, que determinan el inicio y final de carrera de la placa móvil.

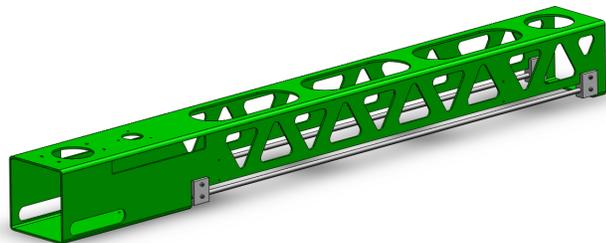


Fig. 2.2: Pluma

La pluma se une al vástago de soporte mediante un juego de rodamientos acoplados a la parte superior e inferior de la pluma.

La placa móvil (Fig. 2.3) provee soporte físico para los elementos de sensado y actuación que se encargan de realizar la tarea directamente. Cuenta con la geometría y los barrenos necesarios para el acople del arreglo estereoscópico de cámaras, y para fijar la base del mecanismo de ajuste (Fig. 2.4). El diseño del mecanismo se presenta en la Sec. 2.4. En los costados de la parte superior de la placa se encuentran los elementos de sujeción a las guías de la pluma. Para facilitar la operación y disminuir la precisión requerida en el maquinado, un par de dichos elementos cuentan con un barreno circular coincidente con el de las guías, mientras que el par contrario tiene una canal rectangular que evita el giro de la placa móvil. Al centro de la parte superior de la placa se ha colocado una pieza que captura una tuerca de 6.35 mm de diámetro, mediante la cual se logra el acoplamiento mecánico de la placa con el tornillo sin fin que controla su avance. La geometría de la placa no corresponde a una figura regular, pero queda contenida dentro de un rectángulo de $70 \times 168\text{ mm}$, y ha sido diseñada de aluminio de 5 mm de espesor.

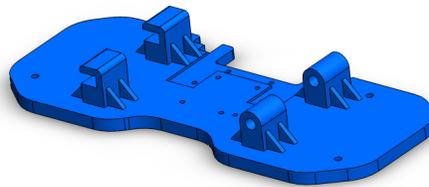


Fig. 2.3: Placa móvil

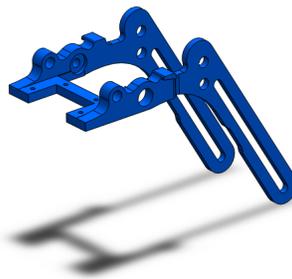


Fig. 2.4: Base del mecanismo de ajuste

En lo que respecta a los cuatro actuadores con que cuenta el sistema, tres de ellos son motores paso a paso unipolares, el primero de los cuales, q_1 , cuenta con ángulo de paso de 8° y torque máximo de $1.9\text{ kg} \cdot \text{cm}$. El acoplamiento de éste con el resto del sistema se lleva a cabo mediante el engranaje de dos engranes rectos, con 40 y 120 dientes respectivamente, y módulo $m = 1$. El diseño del engranaje

se muestra en la Sec. 2.3. Los dos motores a pasos correspondientes a los actuadores q_2 y q_4 se caracterizan por un ángulo de paso de 15° y torque máximo de $1 \text{ kg} \cdot \text{cm}$. El actuador q_2 se acopla a una tornillo sin fin de 6.35 mm de diámetro, el cual descansa sobre rodamientos contenidos en bases diseñadas para tal fin (Fig. 2.5), colocadas en la pluma al inicio y final de carrera de la placa móvil. Al actuador q_4 se acopla un dado para ajustar cada una de las turcas del concentrador.

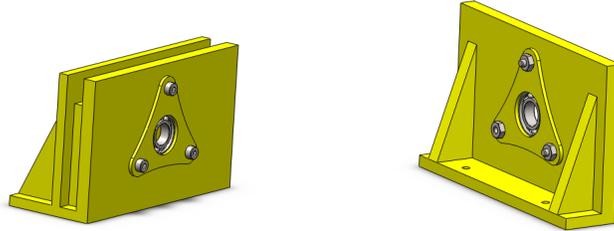


Fig. 2.5: Bases para rodamientos del tornillo sin fin

El actuador restante (q_3), tercero en la cadena cinemática, es un servomotor VIGOR VS-2A con ángulo de salida en el intervalo $[0^\circ - 170^\circ]$, voltaje de operación de 4.5 V , y torque de salida de $4.5 \text{ kg} \cdot \text{cm}$ [98], que se acopla a la manivela del mecanismo para controlar su posición.

2.3. Diseño del engranaje

Para transmitir el movimiento del actuador q_1 de forma que se logre el giro de la pluma alrededor del vástago, es necesario diseñar los elementos de transmisión de potencia. Un sistema de dos engranes es capaz de transmitir el movimiento de un motor evitando deslizamiento entre los elementos, además, dependiendo de la relación dimensional entre estos, es posible multiplicar el torque de salida.

La relación entre el torque de entrada (T_E) y el torque de salida (T_S) está determinada por:

$$T_E \cdot \omega_E = T_S \cdot \omega_S, \quad (2.1)$$

donde ω_E es la velocidad angular del engrane de entrada, y ω_S es la velocidad angular del engrane de salida. Despejando T_S de la Ec. 2.1,

$$T_S = \frac{\omega_E}{\omega_S} \cdot T_E, \quad (2.2)$$

es posible observar que si $\omega_E > \omega_S$, entonces $\omega_E/\omega_S > 1$, es decir, el torque de entrada se magnifica en la salida por la proporción ω_E/ω_S . Para evitar causar sobrecargas al actuador encargado del movimiento de la pluma, es deseable disponer de tres veces el torque nominal del motor. De esta forma, para el diseño se fija:

$$\omega_E : \omega_S = 3 : 1, \quad (2.3)$$

y la distancia entre ejes (motor - vástago de soporte) es de:

$$c = 80 \text{ mm.} \quad (2.4)$$

Se debe diseñar entonces el diámetro de los círculos de paso, el número de dientes, y el paso diametral (P) (si se trabaja en unidades del sistema inglés) o el módulo (m) (para unidades métricas), de los engranes de entrada y salida. El módulo y el paso diametral están relacionados como:

$$m = \frac{1}{P}, \quad (2.5)$$

además, recordando que la relación entre el paso diametral (P), el número de dientes (N), y el diámetro (d) de un engrane están relacionados por:

$$P = \frac{N}{d}, \quad (2.6)$$

y que la relación de velocidad entre engranes es:

$$\frac{\omega_S}{\omega_E} = \frac{\# \text{ de engranes impulsores}}{\# \text{ de engranes impulsados}}, \quad (2.7)$$

entonces, con base en el engranaje mostrado en la Fig. 2.6, es posible llevar a cabo el diseño así:

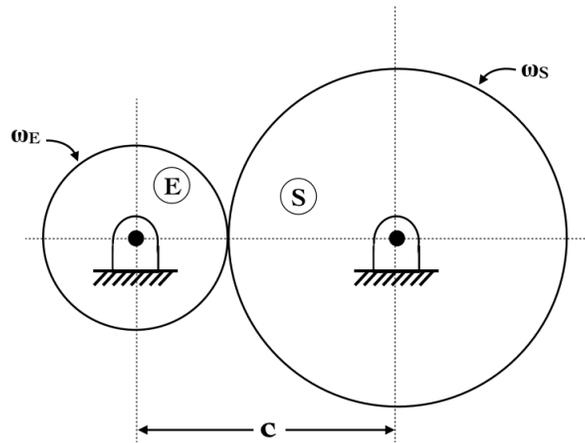


Fig. 2.6: Esquema de engranaje para transmisión de potencia del actuador q_1

$$\frac{\omega_S}{\omega_E} = \frac{N_E}{N_S} = \frac{P \cdot d_E}{P \cdot d_S} = \frac{d_E}{d_S}, \quad (2.8)$$

sustituyendo 2.3 en 2.8:

$$\frac{d_E}{d_S} = \frac{1}{3}. \quad (2.9)$$

Por otro lado, la distancia entre ejes se puede escribir como una relación de radios (r),

$$c = r_E + r_S = \frac{d_E + d_S}{2} = 80, \quad (2.10)$$

$$d_E + d_S = 160. \quad (2.11)$$

Las Ec. 2.9 y 2.11 forman un sistema de ecuaciones, cuya solución determina el diámetro del círculo base de los engranes de entrada y salida, esto es:

$$d_E = 40 \text{ mm}, \quad (2.12)$$

$$d_S = 120 \text{ mm}. \quad (2.13)$$

Utilizando la Ec. 2.6 se obtienen los números de dientes y el módulo. Dado que, para que dos engranes se acoplen satisfactoriamente, deben tener un mismo módulo, entonces:

$$d_E = \frac{N_E}{P} = N_E \cdot m \longrightarrow 40 = N_E \cdot m, \quad (2.14)$$

$$d_S = \frac{N_S}{P} = N_S \cdot m \longrightarrow 120 = N_S \cdot m. \quad (2.15)$$

De esta forma, los parámetros que caracterizan el engranaje diseñado son:

$$N_E = 40, \quad (2.16) \quad N_S = 120, \quad (2.17)$$

$$m = 1, \quad (2.18) \quad P = 1, \quad (2.19)$$

Solamente resta validar el diseño comprobando que no exista interferencia entre los dientes de los engranes escogidos, y que la razón de contacto sea adecuada.

La razón de contacto especifica el número de dientes que permanecen en contacto en todo momento. Una razón de contacto igual a uno significa que un par de dientes está en contacto siempre. Los engranes son usualmente diseñados con razones de contacto $r.c. = 1.2$ [100]. La razón de contacto se calcula como:

$$r.c. = \frac{P}{\pi \cdot \cos(\phi)} \left(\sqrt{(r_S + a_S)^2 - (r_S \cos(\phi))^2} - r_S \sin(\phi) + \sqrt{(r_E + a_E)^2 - (r_E \cos(\phi))^2} - r_E \sin(\phi) \right), \quad (2.20)$$

donde ϕ es el ángulo de contacto entre dientes ($\phi = 20^\circ$ para este caso), r_E y r_S son los radios de los

engranes E y S respectivamente, mientras que a_E y a_S son los adendum¹, cuyo valor para engranes rectos estandar es de $a_i = \frac{1}{P}$ [100]. Con los valores arrojados por el diseño se calcula la razón de contacto, y se obtiene:

$$r.c. = 1.7925 > 1.2. \quad (2.21)$$

Por lo tanto, la razón de contacto es buena.

Por último, es necesario revisar la no interferencia entre dientes, es decir, que el perfil del engrane impulsor no se incruste en la raíz del engrane impulsado, y viceversa. Para determinar si existe interferencia es necesario comparar el radio del círculo de adendum (r_{ai}) del engrane i , con el radio máximo que el círculo de adendum (r_{aiMax}) de dicho engrane puede adoptar, con base en la geometría del sistema. El engranamiento estará libre de interferencia si $r_{ai} < r_{aiMax}$ para cada uno de los engranes que lo componen. El cálculo de los radios está dado como:

$$r_{ai} = r_i + a_i, \quad (2.22)$$

$$r_{aiMax} = \sqrt{(r_i \cos(\phi))^2 + (c \cdot \sin(\phi))^2}. \quad (2.23)$$

Así, sustituyendo en las Ec. 2.22 y 2.23 las magnitudes correspondientes a los engranes E y S , ($i = \{E, S\}$), los valores encontrados son:

$$r_{aE} = 21 < 33.1944 = r_{aEMax}, \quad (2.24)$$

$$r_{aS} = 61 < 62.6701 = r_{aSMax}. \quad (2.25)$$

Por lo tanto, es posible afirmar que no existe interferencia, quedando así el diseño completo. La Fig. 2.7 muestra el arreglo de los engranes diseñados, acoplados a los elementos correspondientes sobre el sistema de ajuste. El diseño utiliza engranes estándar ANSI Métrico.

¹Distancia que existe entre el círculo de adendum (que circunscribe al engrane) y el círculo de paso.

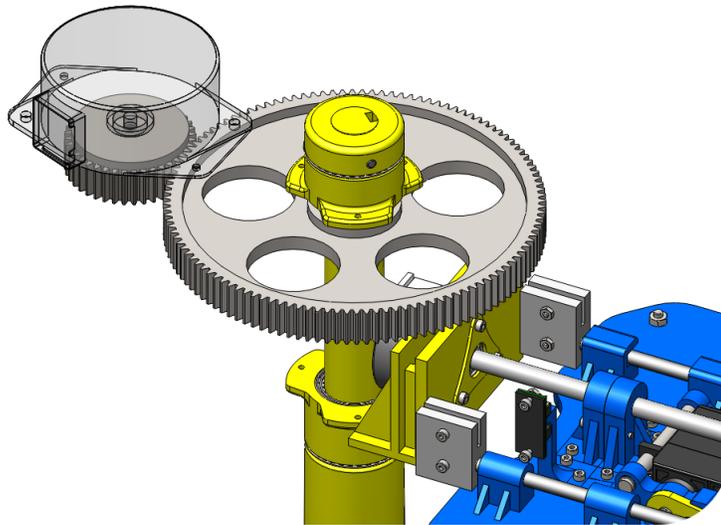


Fig. 2.7: Engranajes para transmisión de potencia del actuador q_1

2.4. Diseño del mecanismo de ajuste

La labor principal del sistema a diseñar es automatizar el ajuste de las tuercas de soporte de la superficie del concentrador solar anteriormente descrito. Si bien para llevarla a cabo se requiere la interacción de varios subsistemas enfocados en tareas diversas, todos ellos se desarrollan alrededor del supuesto de contar con un mecanismo capaz de acceder físicamente a las tuercas y hacerlas girar hasta su posición de ajuste.

Para tal fin, se ha diseñado un mecanismo de barras capaz de desplegarse desde la pluma y alcanzar cada una de las tuercas de ajuste. Una vez conseguida la tarea, el mecanismo se pliega nuevamente dejando libre el campo visual de las cámaras que dirigen la acción del sistema completo. Teniendo en cuenta la operación general del sistema, los requisitos impuestos al mecanismos de ajuste son:

- Debe ser ligero y compacto.
- Capaz de desplegarse y retraerse desde su posición en la pluma un total de 14 cm , que corresponde a la distancia vertical máxima entre la cara inferior del dado de ajuste de la pluma y las tuercas del concentrador.
- Requerir para su accionamiento de un solo actuador, con el fin de ahorrar espacio, peso, y recursos.
- Acoplarse a la placa móvil montada en la pluma para aprovechar su movimiento.
- Contar con una configuración geométrica que evite colisiones con el resto de los componentes del sistema.

La solución ideada está basada en los mecanismos de tijera que unen sus barras no solo por los extremos, sino en puntos intermedios de su longitud, consiguiendo así el efecto de acordeón que les confiere la capacidad de desplegarse. La Fig. 2.8 muestra el esquema básico que se ha ideado como punto de partida. Se trata de un mecanismo plano de cinco barras, con un apoyo fijo en tierra, al cual se une una de sus barras, denominada R_1 . La barra R_2 se une con R_1 en un punto intermedio entre estas, y a su vez, en uno de sus extremos, con la corredera R_5 que desliza sobre una superficie plana. Las barras R_3 y R_4 se unen entre si en uno de los extremos, y a su vez con los extremos de las barras R_2 y R_1 respectivamente.

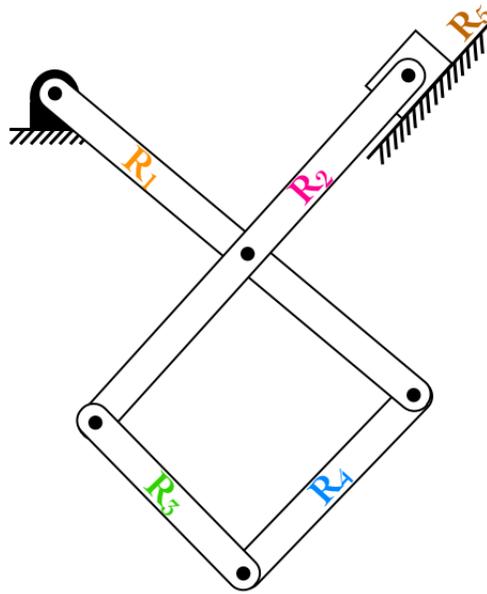


Fig. 2.8: Configuración geométrica del mecanismo de ajuste

El mecanismo está formado por un total de $L = 6$ cuerpos (cuatro barras, una corredera, y la tierra), y $J_1 = 7$ juntas completas (seis pernos y una corredera), es decir, que restringen dos grados de libertad. Así, calculando los grados de libertad (GDL) del sistema:

$$GDL = 3(L - 1) - 2J_1 - J_2^0 = 15 - 14 = 1, \quad (2.26)$$

se obtiene que $GDL = 1$. Por facilidad de implementación y manejo del mecanismo, se elige como manivela a la barra R_1 , controlando su posición mediante el giro θ_1 , alrededor del perno a tierra.

Una vez establecida la geometría base del mecanismo, es necesario dimensionar cada uno de sus elementos para cumplir la tarea asignada. En particular, para este diseño se utilizará un método basado en la formación de ecuaciones vectoriales de lazo entre los elementos del mecanismo. Dicho método requiere definir una base inercial en tierra, que se utiliza como marco de referencia global, y una serie de bases locales adosadas a cada uno de los elementos que conforman el mecanismo, y que representan el origen de los vectores que determinan la posición y orientación de cada una de las

barras en el espacio. La Fig. 2.9 muestra las bases locales correspondientes a cada una de las barras, y los ángulos medidos desde la horizontal con que dichos marcos rotan.

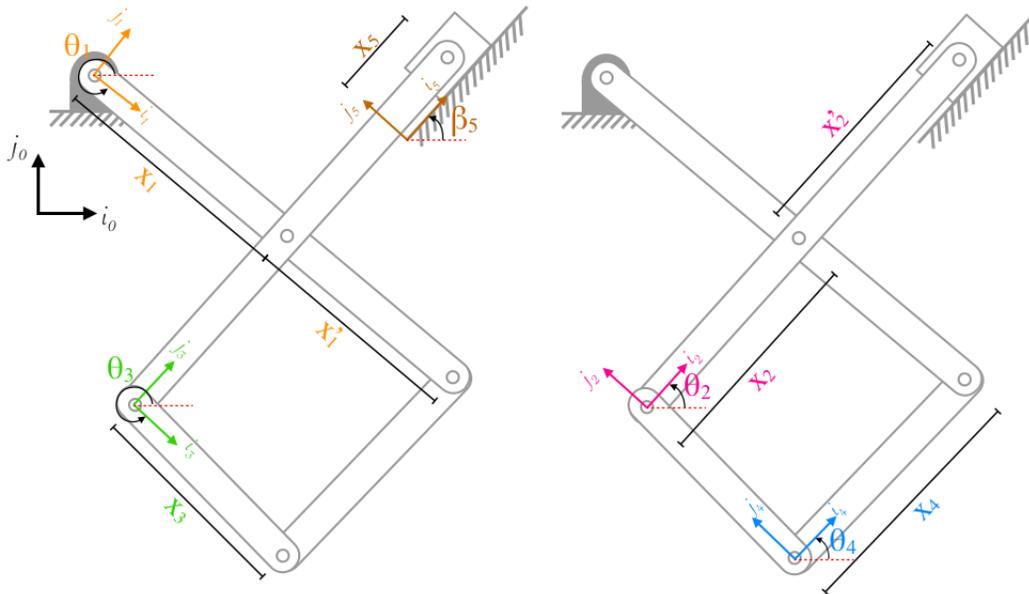


Fig. 2.9: Bases locales, e inercial, ángulos de rotación, y dimensiones de los elementos del mecanismo

Los vectores locales se construyen con la dimensión longitudinal de la barra que representan, en términos de la base local correspondiente. Así, los vectores locales de las barras son:

$$r_1 = \{x_1, 0\}^T, \quad (2.27)$$

$$r'_1 = \{x'_1, 0\}^T, \quad (2.28)$$

$$r_2 = \{x_2, 0\}^T, \quad (2.29)$$

$$r'_2 = \{x'_2, 0\}^T, \quad (2.30)$$

$$r_3 = \{x_3, 0\}^T, \quad (2.31)$$

$$r_4 = \{x_4, 0\}^T, \quad (2.32)$$

$$r_5 = \{x_5, 0\}^T, \quad (2.33)$$

Las ecuaciones de lazo se construyen mediante igualación de sumas vectoriales. Para que dos vectores puedan ser sumados es necesario que ambos se expresen bajo el mismo marco de referencia. Una matriz de rotación, o de cambio de base, es la expresión de un operador que transforma vectores de un determinado marco de referencia a su equivalente en otro marco dado. En particular, para el plano, la

matriz de rotación con respecto al ángulo θ es:

$$R[\theta] = \begin{bmatrix} \cos \theta & -\text{sen} \theta \\ \text{sen} \theta & \cos \theta \end{bmatrix}. \quad (2.34)$$

La Fig. 2.10 presenta los vectores en base inercial que determinan la posición de cada una de las barras del mecanismo, y con los cuales se forman las ecuaciones de lazo. Éstos se obtienen al aplicar la matriz de rotación, con respecto al ángulo correspondiente, a cada uno de los vectores en base local.

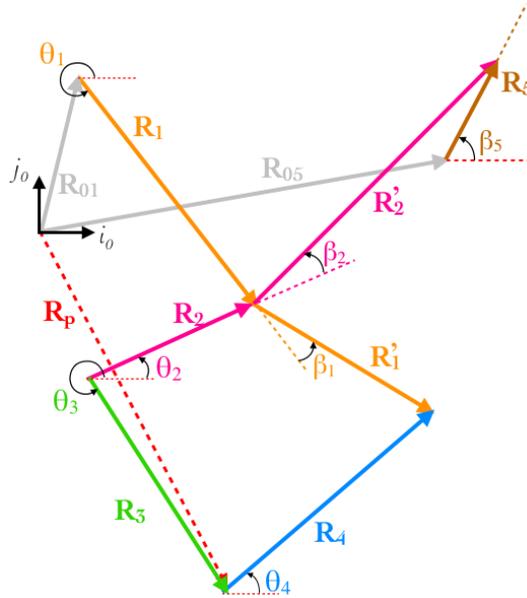


Fig. 2.10: Vectores de posición en base inercial

Adicionalmente se definen tres vectores, R_{01} localiza el apoyo fijo con respecto al origen del marco de referencia, R_{05} posiciona un punto de referencia para medir el desplazamiento de la corredera, y R_p que describe la trayectoria que el mecanismo diseñado debe recorrer. Así:

$$R_{01} = \{x_{01}, y_{01}\}^T, \quad (2.35) \quad R_{05} = \{x_{05}, y_{05}\}^T, \quad (2.36)$$

$$R_p = \{x_p, y_p\}^T, \quad (2.37) \quad R_1 = R[\theta_1] r_1, \quad (2.38)$$

$$R'_1 = R[\theta_1 + \beta_1] r'_1, \quad (2.39) \quad R_2 = R[\theta_2] r_2, \quad (2.40)$$

$$R'_2 = R[\theta_2 + \beta_2] r'_1, \quad (2.41) \quad R_3 = R[\theta_3] r_3, \quad (2.42)$$

$$R_4 = R[\theta_4] r_4, \quad (2.43) \quad R_5 = R[\beta_5] r_5. \quad (2.44)$$

El ángulo constante β_5 define la orientación de la superficie sobre la que se desliza la corredera. Dos ángulos constantes más, β_1 y β_2 , se han introducido en las Ec. 2.39 y 2.41 respectivamente, y pueden ser apreciados en la Fig. 2.10. Su función consiste en brindar a las barras R_1 y R_2 un dobléz que facilite la tarea que estas deben desarrollar, otorgando mayor flexibilidad al proceso de dimensionamiento del mecanismo. Por otro lado, el vector R_p puede ser expresado como una función que, al recorrerse entre los límites de su dominio, marque el camino deseado a desarrollar por el extremo del mecanismo, sin embargo, también puede ser definido como un conjunto de n puntos en \mathbb{R}^2 , por los cuales el mecanismo debe pasar. Este último enfoque es el adoptado en este trabajo.

Las ecuaciones de lazo que se forman para el diseño deben recorrer, en su conjunto, todos los elementos del mecanismo para garantizar que la interacción de la configuración completa es tomada en cuenta en el proceso. Las ecuaciones de suma vectorial utilizadas son:

$$R_2 + R'_1 + -R_4 - R_3 = 0, \quad (2.45)$$

$$R_{01} + R_1 + R'_2 - R_5 - R_{05} = 0, \quad (2.46)$$

$$R_{01} + R_1 + R'_1 - R_4 - R_p = 0, \quad (2.47)$$

Las Ec. 2.45 y 2.46 describen la configuración del mecanismo, mientras que la información de la trayectoria deseada se proporciona mediante la Ec. 2.47.

El proceso de dimensionamiento consiste en encontrar las dimensiones de las barras del mecanismo que hagan posible que el extremo de este pase por los puntos indicados por el vector R_p . El objetivo principal es hacer descender el motor de ajuste, por lo tanto la variable más importante del vector $R_p = \{x_p, y_p\}^T$ es y_p , pues de la especificación de esta depende la magnitud del descenso, mientras que x_p puede dejarse como incógnita, ampliando así el espacio de solución de las dimensiones del mecanismo. Dado que el mecanismo debe desplegarse un total de 14 cm, se establecen $n = 3$ puntos de control por donde el extremo del mecanismo debe pasar. R_p se escribe entonces como un arreglo

de n vectores, de modo que:

$$R_p [1] = \{x_p [1], -2\}^T, \quad (2.48)$$

$$R_p [2] = \{x_p [2], -7\}^T, \quad (2.49)$$

$$R_p [3] = \{x_p [3], -16\}^T. \quad (2.50)$$

Una vez definido R_p , se deben de encontrar valores para las constantes de dimensión $x_1, x'_1, x_2, x'_2, x_3, x_4, x_{01}, y_{01}, x_{05}, y_{05}, \beta_1, \beta_2, \beta_5$, y para las variables $\theta_1, \theta_2, \theta_3, \theta_4, x_5$ del mecanismo, que satisfagan las Ec. 2.45, 2.46 y 2.47 en cada una de las n posiciones especificadas en R_p , es decir, se generan 3 ecuaciones de lazo vectoriales (o 6 ecuaciones escalares) para cada una de las n posiciones de R_p , así como un vector de n elementos para cada una de las variables del mecanismo.

Debido a que se tiene más incógnitas ($13 + 6n$) que ecuaciones ($6n$), es necesario aplicar un método de optimización numérica para resolver el sistema. El primer paso consiste en elegir una función objetivo, en este caso se toma la suma de los errores cuadrados, esto es, se eleva al cuadrado cada una de las componentes x e y de las ecuaciones de lazo igualadas a cero y evaluadas en cada punto de la iteración, en busca de llevar a cero el valor de dicha función. Para la búsqueda de la solución se utiliza un método quasiNewton [69], con un punto inicial de búsqueda aleatorio. Se sugieren los valores del perno en tierra en los intervalos $x_{01} \in [-1, 0]$, $y_{01} \in [0, 3]$, y se fija el valor de la manivela en la posición inicial plegada, $\theta_1 = 350^\circ$.

La Fig. 2.11 muestra cuatro diferentes diseños obtenidos mediante el procedimiento anteriormente descrito. Para determinar el mecanismo a utilizar es importante definir los criterios de selección y la relevancia de estos. Los criterios utilizados son:

1. La suma de la longitud de las barras que componen el mecanismo, medida directamente relacionada con la masa del sistema.
2. Compacidad horizontal. Expresa la dimensión horizontal de extremo a extremo del mecanismo plegado, se relaciona con espacio que este ocupa, y determina en gran medida la posibilidad de colisiones.
3. Coordenada x_p del extremo del mecanismo (donde se acopla el motor de ajuste) cuando se llega al final de la carrera ($y_p = -16$). Este es el criterio más importante pues determina la capacidad del mecanismo y de la placa móvil de la pluma para alcanzar la tuerca de ajuste más cercana al vástago de soporte. Valores menores o cercanos a cero son preferibles.

La Tabla 2.1 muestra los valores de los criterios de selección para cada uno de los diseños del mecanismo de ajuste mostrados en la Fig. 2.11.

Tabla 2.1: Valor de los criterios de selección del mecanismo de ajuste

Concepto	Suma de longitud de barras (cm)	Compacidad horizontal (cm)	Distancia al cero (cm)
1	45.65	14.90	3.96
2	41.70	14.17	0.08
3	50.23	15.30	0.86
4	36.57	10.49	-0.36

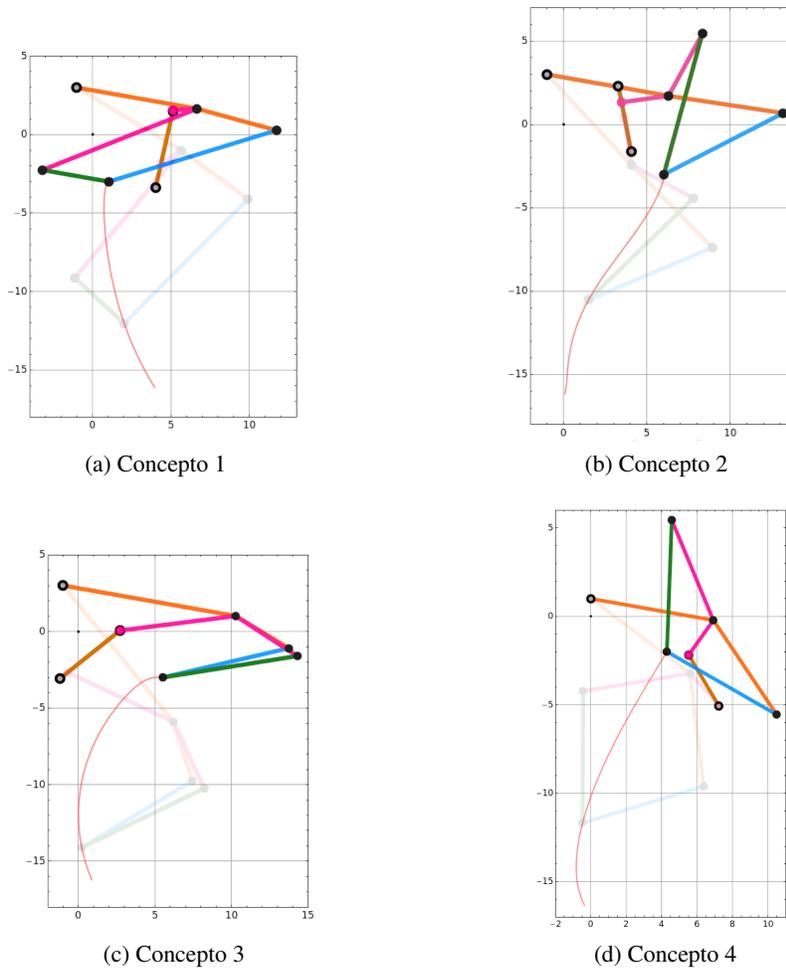


Fig. 2.11: Mecanismos obtenidos en el proceso de diseño

Con base en los valores de la Tabla 2.1, el concepto seleccionado es el número 4. Las dimensiones finales del mecanismo son²:

²Las dimensiones son medidas c.a.c, y los magnitudes negativas en los componentes de los vectores indican un cambio de sentido con respecto a los vectores de la Fig. 2.10

$$x_{01} = 0 \text{ cm}, \quad (2.51)$$

$$y_{01} = 1 \text{ cm}, \quad (2.52)$$

$$x_{05} = 5.44 \text{ cm}, \quad (2.53)$$

$$y_{05} = -2.04 \text{ cm}, \quad (2.54)$$

$$x_1 = 7.025 \text{ cm}, \quad (2.55)$$

$$x'_1 = 6.423 \text{ cm}, \quad (2.56)$$

$$x_2 = -6.131 \text{ cm}, \quad (2.57)$$

$$x'_2 = 2.408 \text{ cm}, \quad (2.58)$$

$$x_3 = 7.449 \text{ cm}, \quad (2.59)$$

$$x_4 = 7.145 \text{ cm}, \quad (2.60)$$

$$\beta_1 = -0.80596 \text{ rad}, \quad (2.61)$$

$$\beta_2 = 2.13213 \text{ rad}, \quad (2.62)$$

$$\beta_5 = 5.24649 \text{ rad} \quad (2.63)$$

Hasta ahora, para el proceso de diseño, el ángulo de la manivela ha sido llamado θ_1 , sin embargo, desde el punto de vista del sistema completo, este corresponde al tercer GDL, denominado q_3 . El intervalo de movimiento de la manivela es de $[311^\circ - 350^\circ]$; el inicio del intervalo corresponde al estado completamente desplegado, mientras que al alcanzar el límite superior del intervalo el mecanismo estará completamente plegado.

El material elegido para las barras del mecanismo es aluminio aleación 6063-T5; las dimensiones de las barras son: la longitud especificada por el diseño, 14 mm de ancho, y 3 mm de espesor. Para disminuir el desgaste en las justas y aumentar así la vida útil del mecanismo, se utilizan bujes de bronce y ejes de acero. El mecanismo ensamblado con dichas barras tiene una masa total (de los componentes móviles) de 97.38 g. Con el fin de aligerar las barras para disminuir los efectos dinámicos provocados por la masa sobre el torque requerido, se realizó un análisis estático lineal de esfuerzos y deformación para conocer el comportamiento mecánico del mecanismo plegado y desplegado. La carga a la que es sometido corresponde al peso propio de las barras y el motor de ajuste. El estudio arroja un factor de seguridad mínimo, bajo el criterio de Von Mises [14], igual a $F.S. = 60.81$, es decir, la resistencia de los elementos está muy por encima de las solicitaciones impuestas.

Con base en el resultado anterior, se decide eliminar material de las barras. Una vez aligeradas, se lleva a cabo nuevamente el estudio del comportamiento mecánico, obteniendo un factor de seguridad mínimo de $F.S. = 55.73$, por lo tanto, el mecanismo es apto para realizar su tarea sin sufrir efectos mecánicos de consideración. El peso final de las barras es de 67.84 g; se consigue así un ahorro del 30.33 % en la masa que determina la magnitud de los efectos dinámicos.

Finalmente, se lleva a cabo la determinación del torque requerido en la manivela, para llevar al me-

canismo de la posición completamente plegada hasta alcanzar su extensión máxima, y de regreso. El análisis se realiza mediante el software de simulación SolidWorks Simulation [14], y se alimenta con la fricción entre las piezas tomando en cuenta el material elegido, las restricciones de los pasadores, y una rapidez de desplazamiento basada en la hoja de especificación del servomotor propuesto para realizar la tarea. El ciclo de trabajo para el análisis consiste en llevar la posición angular q_3 de la manivela de 350° a 311° en 0.26 s , esperar 0.5 s al llegar al extremo inferior, y llevar nuevamente la manivela a 350° en el mismo periodo de tiempo que tomó el descenso³. La Fig. 2.12 muestra el torque requerido bajo el ciclo de trabajo propuesto para el movimiento del mecanismo a la máxima velocidad de operación del servomotor.

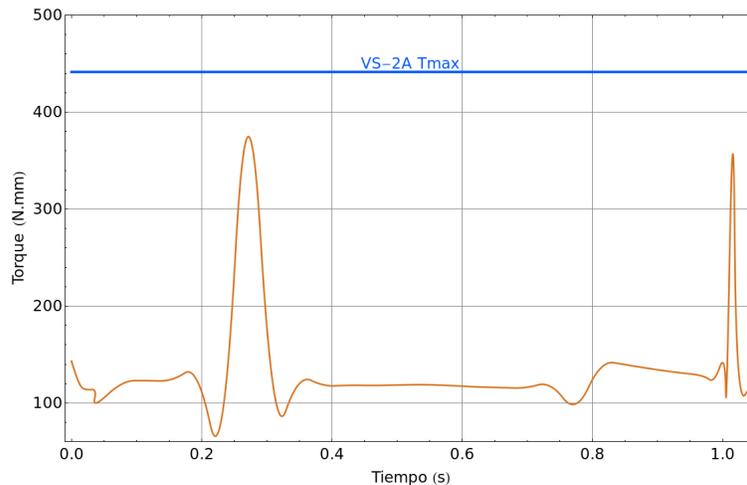


Fig. 2.12: Torque requerido para el movimiento del mecanismo

En la gráfica de la Fig. 2.12 es posible observar dos picos en el torque requerido, el primero corresponde al instante en que el mecanismo debe frenarse para alcanzar la posición de máximo despliegue, mientras que el segundo está relacionado con el frenado que se produce para llegar a la posición completamente plegada durante el ascenso. El servomotor propuesto, Vigor VS-2A, es capaz de entregar un torque de $441.45\text{ N} \cdot \text{mm}$, mientras que el máximo requerido es de $374.62\text{ N} \cdot \text{mm}$. Por lo tanto se valida la elección del mismo, señalando que el controlador interno con que este cuenta es capaz de regular la entrega de torque que se requiere.

La Fig. 2.13 muestra el ensamble del mecanismo de ajuste en dos diferentes posiciones. A el se encuentra adosado el motor que hace girar cada una de las tuercas del concentrador. La base del mecanismo, donde se encuentra la corredera, se acopla a la placa móvil de la canastilla de actuación, tal como se ha descrito en la Sec. 2.2.

³La hoja de datos del Vigor VS-2A [98] especifica un velocidad de operación sin carga, $0.2\text{ s}/60^\circ$. Para fines prácticos se considera una velocidad de operación en estado de carga del doble, es decir, $0.4\text{ s}/60^\circ$.

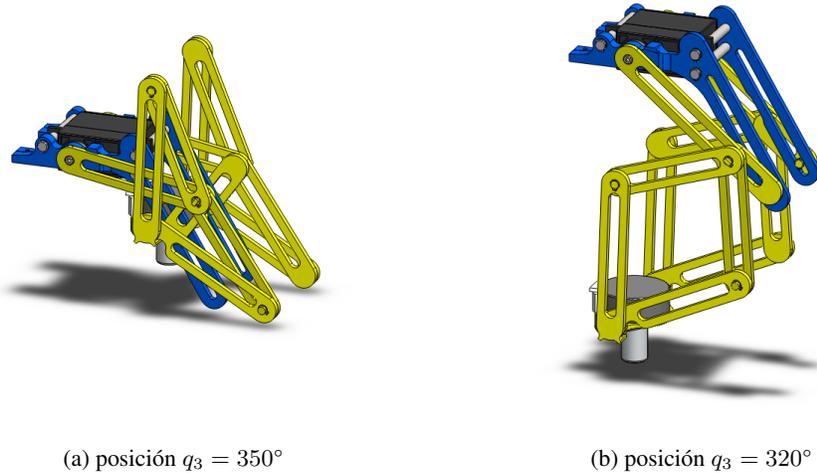


Fig. 2.13: Mecanismo de ajuste

2.5. Optimización masa - rigidez de la pluma

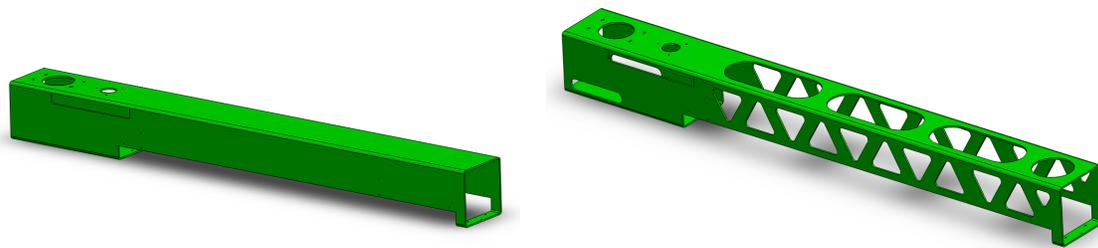
Uno de los factores más importantes de la estructura mecánica del sistema es la masa, pues la magnitud de esta propiedad es determinante en el desarrollo de la tarea, así como en el buen funcionamiento de los actuadores involucrados, y en la velocidad máxima con que pueda llevarse a cabo el ajuste bajo los parámetros establecidos y los elementos de actuación seleccionados.

La pluma, uno de los dos macrocomponentes principales, incide de forma considerable en la masa final del sistema. Teniendo en cuenta las funciones que desarrolla, y la geometría que para ello requiere, es posible proponer la eliminación de material en aquellas zonas en que no resulta indispensable, o incluso de utilidad. Sin embargo, retirar material de forma indiscriminada podría inducir problemas como la deformación excesiva del cuerpo de la pluma. Queda entonces claro que el material debe ser retirado bajo criterios que tengan en cuenta el efecto de la disminución de masa y la geometría resultante, sobre algún parámetro relacionado con el bienestar de la pieza y su funcionamiento.

La variable seleccionada para evaluar el efecto de la disminución de masa, a lo largo del proceso de aligeramiento de la pluma, es la magnitud del desplazamiento vertical en el punto de la pluma más alejado del vástago de soporte. Dicho desplazamiento será medido en la condición estática extrema, es decir, con la placa móvil colocada al final de su carrera, y el mecanismo de ajuste plegado, para aumentar así el momento que éste ocasiona.

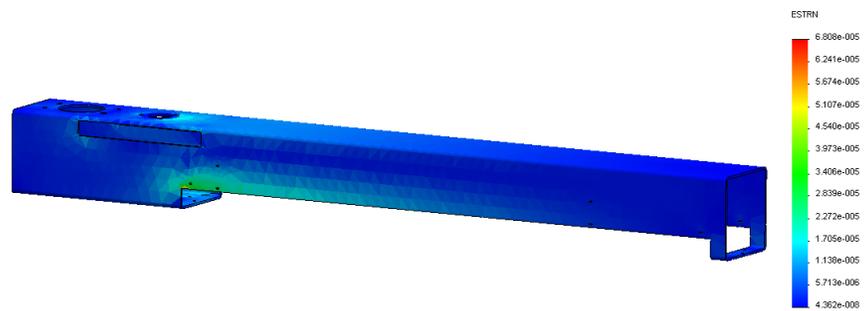
El experimento consiste en llevar a cabo un análisis estático de deformación de la pluma, afectada por la acción de la gravedad sobre su propia masa, así como por el peso de todos los elementos acoplados a esta, y definiendo las restricciones adecuadas en el punto de unión con el vástago de soporte. La Fig. 2.14 muestra la magnitud de la deformación a lo largo del cuerpo de la pluma para la geometría

original y para el modelo optimizado, de donde es posible apreciar el comportamiento mecánico de ambas configuraciones.

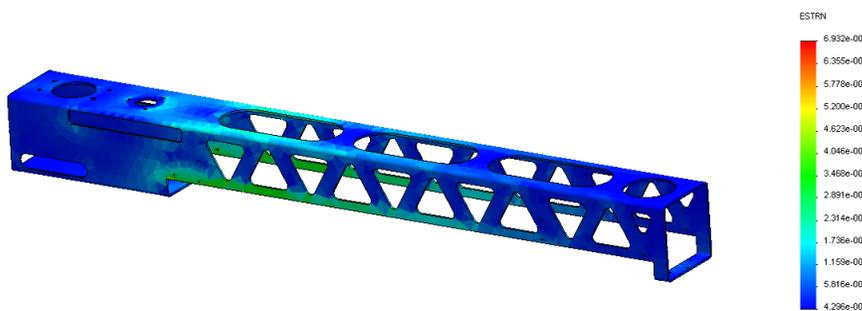


(a) geometría original

(b) geometría final



(c) deformación modelo original



(d) deformación modelo final

Fig. 2.14: Magnitud de la deformación de la pluma

El desplazamiento del modelo original en el punto seleccionado es de $d_e = 0.1008 \text{ mm}$. El criterio utilizado para finalizar el proceso de sustracción de material es alcanzar un valor de desplazamiento máximo (d_{max}) de una décima mayor a d_e , es decir, $d_{max} = d_e + 0.1 \text{ mm}$, o que no exista más material por retirar.

La Fig. 2.15 muestra los resultados del análisis de deformación en cada una de las iteraciones. En el eje horizontal se presenta el número de iteración, la masa de la pluma, y el porcentaje de disminución de masa con respecto a la iteración 0, mientras que el eje vertical muestra el valor de la deformación en

el punto extremo, la deformación máxima para cada iteración, y el desplazamiento del punto extremo.

Los valores del esfuerzo y la deformación de la pluma están dentro de los límites de seguridad para todas las iteraciones. Al iniciar el proceso se tiene un factor de seguridad $F.S. = 13.36$, y se finaliza con $F.S. = 15.06$. El aumento en el $F.S.$ se debe a la disminución de masa, y a la redistribución de esfuerzos producto de los cambios en la geometría.

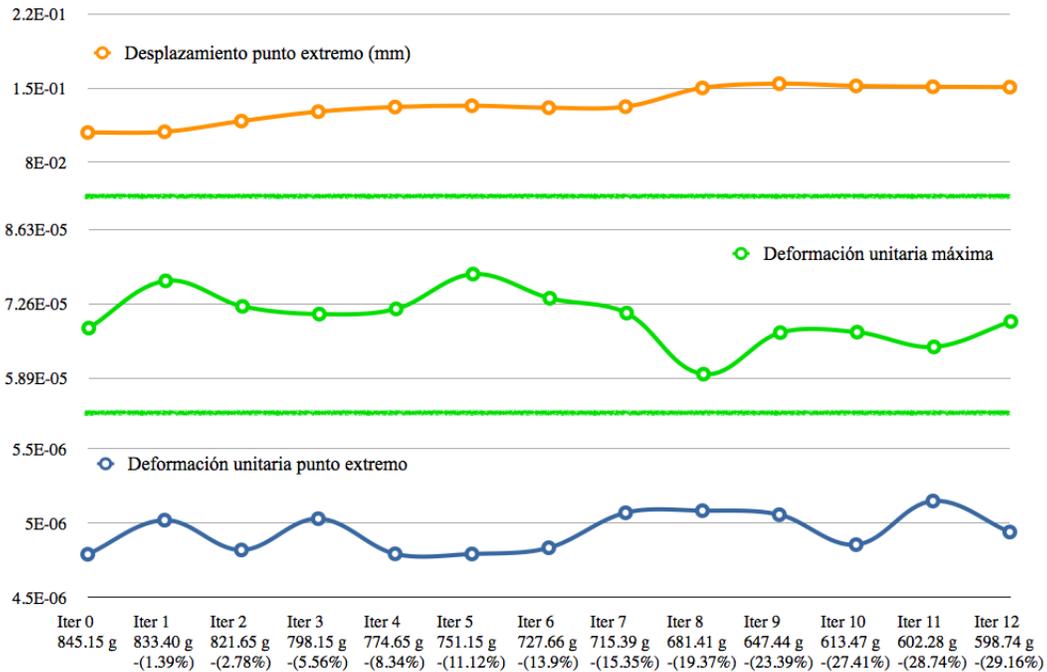


Fig. 2.15: Comportamiento mecánico de la pluma para cada iteración

La masa del modelo original es de 845.15 g , y el proceso concluye tras 12 iteraciones, con masa final de 598.74 g , es decir, una disminución del 29.16% de la masa de la pluma, consiguiendo así el objetivo fijado para el proceso de optimización masa - rigidez. El desplazamiento en la iteración 12 es $d_{ef} = 0.1508\text{ mm}$, es decir, cinco centésimas de milímetro más que el original (d_e), conservándose por debajo del límite de servicio (d_{max}) establecido.

¿Está vivo ... vivo!

[DR. FRANKENSTEIN]

3.1. Definición de funciones

El subsistema electrónico está compuesto por el hardware y el software necesario para el manejo de los actuadores, así como de un sistema codificado de instrucciones utilizadas para la comunicación entre éste y el subsistema de visión computacional.

Las funciones del subsistema electrónico son:

1. Servir de interfaz entre el módulo mecánico y el de visión computacional, recibiendo e interpretando las órdenes para el posicionamiento y acción del mecanismo de ajuste.
2. Dotar al μ Controlador¹ de la electrónica complementaria necesaria para su correcto funcionamiento, y para la transmisión de instrucciones desde éste a la etapa de potencia.
3. Proporcionar una etapa de potencia capaz de aumentar la amplitud de las señales a los niveles necesarios para accionar los actuadores del sistema en la forma requerida, es decir, que lleve a cabo el acoplamiento de estos últimos con la etapa de señales manejada por el μ Controlador.
4. Servir de canal para hacer llegar la energía necesaria para el funcionamiento del subsistema mecánico.

¹Microcontrolador.

3.2. Requerimientos técnicos

El subsistema electrónico es el intérprete entre las instrucciones emitidas por el subsistema de visión computacional y los comandos que el conjunto de actuadores del sistema debe recibir.

Así mismo, éste subsistema se encarga de comunicar al de visión computacional el estado de los sensores con que cuenta el mecánico. Para cumplir con las funciones descritas en la sección anterior, el subsistema electrónico requiere de:

- Un soporte físico (PCB²) que proporcione los canales de transmisión (pistas) de señales y potencia necesarios para la interconexión y el correcto funcionamiento de los componentes electrónicos que conforman el subsistema.
- μ Controlador capaz de emitir y captar las señales de control necesarias para el manejo de los actuadores y el estado de los sensores respectivamente, en un lapso de tiempo adecuado para el correcto funcionamiento del sistema.
- Conjunto de instrucciones necesarias (software) para la recepción y ejecución de comandos de control (vía emisión de señales), así como un μ Controlador con memoria ROM³ suficiente para almacenar el programa generado, la memoria RAM⁴ necesaria para su ejecución, y velocidad de reloj adecuada para la tarea a desarrollar.
- μ Controlador con el número de entradas y salidas necesarias para el manejo de tres motores a pasos unipolares, un servomotor, y dos sensores infrarrojos tipo optointerruptor.
- μ Controlador con capacidad de comunicación USB⁵.
- Etapa de potencia para suministrar a los actuadores de la corriente y el voltaje necesarios para su funcionamiento, conforme a las señales de control emitidas por el μ Controlador.

3.3. Diseño electrónico

3.3.1. Selección del μ Controlador

El elemento encargado de procesar la información necesaria para manipular los actuadores del sistema es el μ Controlador, es el componente fundamental del subsistema electrónico, y quien comanda todas las acciones que en este subsistema recaen. El μ Controlador a utilizar debe ser elegido cuidadosamente para garantizar el cumplimiento que los requerimientos impuestos por el subsistema.

²Tarjeta de circuito impreso (*Printed Circuit Board*).

³Memoria de sólo lectura (*Read-only memory*).

⁴Memoria de acceso aleatorio (*Random-access memory*).

⁵Bus serie universal.

En la actualidad, existe una gran variedad de productos y marcas en el sector de los microcontroladores [1, 30, 61, 79, 92], sin embargo, por la gran disponibilidad de información técnica, la relativa facilidad de programación, su bajo costo, la existencia de un compilador para lenguaje ANSI C⁶, y su gran presencia en el mercado, se ha elegido trabajar con los productos Microchip.

El proceso de selección de un μ Controlador requiere la especificación de ciertos parámetros objetivos acerca de lo que se busca, los tomados en cuenta en el presente trabajo son:

- Línea (familia) de producto.
- Arquitectura / procesador.
- Pines de entrada / salida.
- Canales de modulación de ancho de pulso (PWM⁷).
- Puertos de comunicación.
- Velocidad de operación.
- Tipo de encapsulado.

Microchip comercializa en el sector de μ Controladores principalmente cuatro líneas de producto, μ Controladores PIC de 8, 16 y 32 bits, y dsPIC (procesadores de señales digitales) [61], cada una de ellas con características propias que las destacan para distintos trabajos.

Por la naturaleza de la aplicación desarrollada es posible cubrir los requerimientos con un μ Controlador de la línea de 8 bits, especialmente enfocados en el manejo de periféricos y la captura básica de señales digitales y analógicas.

En la actualidad, Microchip maneja cuatro gamas de PICs de 8 bits, PIC10F, PIC12F, PIC16F y PIC18F [65]. Con la gama alta (PIC18) se introducen mejoras que reducen las limitaciones de la gama baja y media [32]. Los PIC18 cuentan con una arquitectura RISC⁸ con 16 bits de bus de programa y 8 bits de bus de datos. La memoria de programa aumenta con respecto a las gamas inferiores hasta un total de 128 KB, la memoria RAM puede llegar hasta los 4 KB, y la memoria EEPROM⁹ hasta 1 KB. La frecuencia máxima de trabajo es de 48 MHz, incluye periféricos de comunicación avanzados (CAN¹⁰ y USB), y ha sido desarrollado un compilador de lenguaje C específico para esta gama. Por las ventajas que esta arquitectura presenta, se decide trabajar con el núcleo PIC18.

⁶Estándar publicado por el Instituto Nacional Estadounidense de Estándares (ANSI), para el lenguaje de programación C.

⁷Modulación de ancho de pulso (*Pulse Width Modulation*)

⁸Arquitectura computacional con instrucciones de tamaño fijo, presentadas en un reducido número de formatos. Sus siglas corresponden a *Reduced Instruction Set Computer*.

⁹ROM programable y borrrable eléctricamente (*Electrically-Erasable Programmable Read-Only Memory*).

¹⁰Protocolo de comunicaciones basado en una topología bus para la transmisión de mensajes en entornos distribuidos (*Controller Area Network*).

Tal como se ha expuesto en la Sec. 2.2, los actuadores que controlan la locomoción del sistema, con excepción del mecanismo de ajuste, son tres motores a pasos unipolares. Éste tipo de motores se comunican con el exterior mediante seis cables en su formato más general; el hecho de contar con una derivación central en sus bobinas hace posible unir dos de los seis cables, y conectar estos a la línea de voltaje (V+) [3], dejando así solamente cuatro cables para la comunicación. Se requiere entonces para el control de cada motor a pasos de cuatro salidas digitales. Si bien es cierto que dichas salidas pueden ser multiplexadas para controlar los tres motores, se prefiere cuantificar las salidas de control por separado, con el fin de seleccionar un μ Controlador capacitado para manejar el hardware de forma natural. Al tratarse de tres motores a pasos, se requieren doce salidas digitales para este fin.

Con el fin de simplificar el desarrollo del software de control almacenado en el μ Controlador, es posible optar por un manejador de motores a pasos, es decir, un circuito integrado (CI) comercial que se encarga de recibir órdenes del μ Controlador y ejecutarlas sobre el motor a pasos. En general este tipo de CI requieren de una señal de reloj para fijar la frecuencia de trabajo, es por ello que es conveniente tener un canal de PWM disponible para este fin.

El sistema de ajuste cuenta con dos sensores de tipo optointerruptor, los cuales requieren de dos entradas digitales para que el μ Controlador pueda leer su estado.

Por otro lado, el actuador que gobierna al mecanismo de ajuste es un motor de CD servocontrolado. El comando de control que le ordena la posición a la que el motor debe desplazarse se consigue mediante modulación de ancho de pulso. Este hecho hace necesario que el μ Controlador cuente con un canal de PWM para este propósito.

En cuanto al puerto y protocolo de comunicación entre el subsistema de visión computacional y el subsistema electrónico se ha elegido el USB, por ser un protocolo estandarizado, bien documentado, ampliamente difundido, y con una tasa de transferencia que satisface las necesidades del presente desarrollo. Por tanto, es indispensable que el μ Controlador cuente con un puerto USB.

La utilización del protocolo USB fija la velocidad requerida del oscilador del μ Controlador para su correcto funcionamiento, de esta forma, la velocidad de operación del μ Controlador a seleccionar debe ser mayor o igual a 48 MHz [32].

Finalmente es necesario establecer el tipo de encapsulado que se desea emplear. Si bien es cierto que los desarrollos que involucran elementos electrónicos de montaje superficial presentan amplias ventajas, a nivel experimental y de desarrollo de prototipos. las técnicas tradicionales ofrecen mayor flexibilidad, y requieren equipo menos especializado. Es por ello que se ha optado por un encapsulado del tipo PDIP o SPDIP (Fig. 3.1), por sobre encapsulados de TQFP o QFN, ambos de montaje superficial.

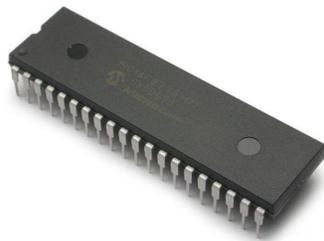


Fig. 3.1: μ Controlador en encapsulado PDIP [61]

La Tabla 3.1 presenta el conjunto de parámetros y sus valores correspondientes, utilizados en la selección del μ Controlador.

Tabla 3.1: Parámetros y valores para selección de μ Controlador

Parámetro	Valor
Línea de producto	8 bits
Arquitectura / procesador	PIC18
Pins I/O	≥ 14
Canales PWM	≥ 2
Puerto de comunicación	USB
Velocidad de operación	48 MHz
Tipo de encapsulado	PDIP

Con la información condensada en la Tabla 3.1 es posible realizar la selección del μ Controlador que cumpla de mejor forma con los requerimientos. La selección se lleva a cabo mediante la utilización de la guía de selección de productos [65] editada por Microchip.

La Fig. 3.2 muestra un extracto de la guía de selección de productos, donde es posible observar cómo está organizada la información, y cómo se puede buscar la paridad entre los μ Controladores disponibles y las necesidades propias.

8-bit PIC® Microcontrollers																																			
Product	Released (R) Not Released (NR)	Pins		Core	Memory				Operating Speed			LCD Segments	mTouch™ Channels	Analog Sensing & Measurement			Digital				Communication			Monitors		5 to Pricing*									
		Total	IO		Program	Self-Read	Self-Write	Data RAM (B)	Data EE (B)	Voltage Range	Maximum Speed			Internal Oscillator	Change Time Measurement Unit	8-bit ADC	10-bit ADC	12-bit ADC	Comparators	CCP	ECCP	8-bit Timer	16-bit Timer	AUSART	EUSART		IC™	SPI	Ethernet (MAC/PHY)	F5-USB	ECAN	BOR/PBOR	PLVD	SR-Latch	Timer 1 Gate
PIC18F46J11	R	40	34	PIC18	64 KB 32 Kw	✓	✓	3000	-	2V-3.6V	48 MHz	8 MHz, 31 kHz	0	13	✓	-	13	-	0	1	-	2	2	3	-	2	2	2	-	-	-	-	-	-	\$2.37
PIC18F4450	R	40	34	PIC18	16 KB 8 Kw	✓	✓	768	-	2V-5.5V	48 MHz	31 kHz	0	13	-	-	13	-	0	1	-	1	2	3	-	1	-	-	-	-	-	-	-	-	\$2.39
PIC18F46K80	NR	40/44	35	PIC18	64 KB 32 Kw	✓	✓	3648	1024	1.8V-5.5V	64 MHz	8 MHz, 31 kHz	0	15	✓	-	-	15	2	4	1	2	3	-	2	1	1	-	-	-	1	-	-	-	\$2.45
PIC18F46J13	R	44	34	PIC18	64 KB 32 Kw	✓	✓	3808	-	2V-3.6V	48 MHz	8 MHz, 31 kHz	0	13	✓	-	-	13	3	7	3	4	4	-	2	2	2	-	-	-	-	-	-	-	\$2.52
PIC18F46J50	R	40	34	PIC18	64 KB 32 Kw	✓	✓	3800	-	2V-3.6V	48 MHz	8 MHz, 31 kHz	0	13	✓	-	-	13	-	2	-	2	2	3	-	2	2	2	-	-	1	-	-	-	\$2.58
PIC18F46J53	R	44	33	PIC18	64 KB 32 Kw	✓	✓	3808	-	2V-3.6V	48 MHz	8 MHz, 31 kHz	0	13	✓	-	-	13	3	7	3	4	4	-	2	2	2	-	-	1	-	-	-	-	\$2.73
PIC18F47J13	R	44	34	PIC18	128 KB 64 Kw	✓	✓	3808	-	2V-3.6V	48 MHz	8 MHz, 31 kHz	0	13	✓	-	-	13	3	7	3	4	4	-	2	2	2	-	-	-	-	-	-	-	\$2.76
PIC18F47J53	R	44	33	PIC18	128 KB 64 Kw	✓	✓	3808	-	2V-3.6V	48 MHz	8 MHz, 31 kHz	0	13	✓	-	-	13	3	7	3	4	4	-	2	2	2	-	-	1	-	-	-	-	\$2.97
PIC18F4550	R	40	35	PIC18	32 KB 16 Kw	✓	✓	2048	256	2V-5.5V	48 MHz	8 MHz, 31 kHz	0	13	-	-	13	-	2	1	1	1	3	-	1	1	1	-	-	1	-	-	-	-	\$3.65
PIC18F4523	R	40	36	PIC18	32 KB 16 Kw	✓	✓	1536	256	2V-5.5V	40 MHz	8 MHz, 31 kHz	0	13	-	-	13	2	1	1	1	1	3	-	1	1	1	-	-	-	-	-	-	-	\$3.67
PIC18F4553	R	40	35	PIC18	32 KB 16 Kw	✓	✓	2048	256	2V-5.5V	48 MHz	8 MHz, 31 kHz	0	13	-	-	13	2	1	1	1	1	3	-	1	1	1	-	-	1	-	-	-	-	\$4.33

Fig. 3.2: Extracto de la guía de selección de productos de Microchip [65]

El proceso de selección consiste en filtrar aquellos dispositivos que cumplan con los requerimientos mínimos expresados. De los 107 dispositivos de 8 bits con núcleo PIC18 solamente 10 cumplen con los requerimientos de la Tabla 3.1. De los dispositivos restantes son eliminados aquellos que por su bajo número de entradas y salidas digitales ≈ 20 o por su limitada memoria de programa pueden limitar el desarrollo futuro del sistema diseñado, siendo entonces viables solamente tres distintos dispositivos. Para seleccionar uno de ellos se siguen las recomendaciones de implementación del fabricante, que informan sobre los usos específicos para los que se recomienda cada dispositivo, siendo de especial interés para este proyecto la interconexión USB con la PC y el control de sistemas embebidos.

El μ Controlador seleccionado es el PIC18F4553 en encapsulado PDIP de 40 terminales. Su diagrama se muestra en la Fig. 3.3. La lista de características detallada puede ser consultada en la hoja de especificaciones del μ Controlador [63].

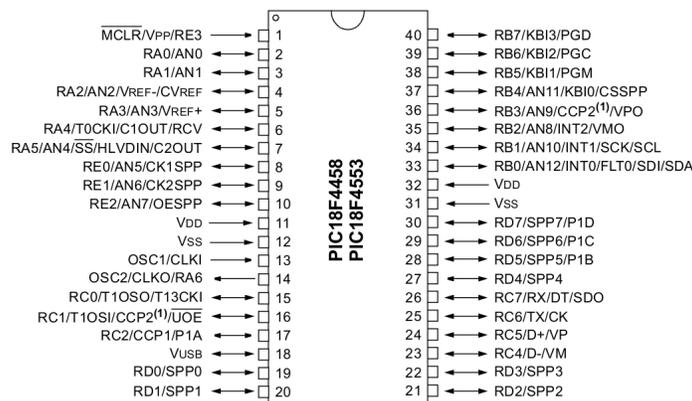


Fig. 3.3: Diagrama de terminales del PIC18F4553[63]

3.3.2. Diagrama de conexiones PIC18F4553

Antes de idear la forma en que los actuadores y sensores del sistema estarán conectados al μ Controlador, es necesario dotar a éste con las conexiones básicas para su correcto funcionamiento. Se trata de proveer al PIC de energía, proporcionarle de una señal de reloj adecuada, y generar el mecanismo para tener control sobre el reset del sistema.

El μ Controlador acepta valores de voltaje en la alimentación en el intervalo $[2V - 5.5V]$; normalmente se alimenta con $5V$ aplicados a los pines V_{DD} y V_{SS} que son la alimentación y la tierra del chip respectivamente. El circuito de alimentación del μ Controlador debe tratarse como el de cualquier otro dispositivo digital, debiendo conectarse un capacitor de desacoplo de alrededor de $C_1 = 100nF$ lo más cerca posible de los pines de alimentación, tal como se muestra en la Fig. 3.4. El manual de especificaciones de programación [64] del PIC18F4553 establece que todos los pines de alimentación deben estar conectados, bien sea a la alimentación o a tierra.

El μ Controlador requiere de un circuito que le indique la velocidad de trabajo, es el llamado oscilador. Éste genera una onda cuadrada de alta frecuencia que se utiliza como señal para sincronizar todas las operaciones del sistema. Generalmente todos los componentes del oscilador se encuentran integrados en el μ Controlador, y tan solo requiere de unos pocos componentes externos. En el PIC18F4553 los pines OSC1/CLKI y OSC2/CLKO/RA6 son las líneas utilizadas para la conexión de los elementos externos del oscilador, estos son dos capacitores y un cristal de cuarzo. Para este proyecto se utilizará un cristal de cuarzo de $48MHz$ y dos capacitores $C_2 = C_3 = 15pF$ en la disposición que se observa en la Fig. 3.4. El valor de los capacitores se obtuvo de las especificaciones técnicas [62, 63] de que provee el fabricante.

El denominado reset de un μ Controlador provoca la reinicialización del funcionamiento del mismo, y es un método útil para llevar al sistema a un punto de estados conocidos ante la ocurrencia de un error. Para tener control sobre el reset del sistema se puede colocar un pulsador de forma tal que cuando este sea presionado proporcione al pin MCLR un estado lógico "0". El reset permanecerá mientras se tenga presionado el pulsador y no comenzará la secuencia de arranque hasta que este sea liberado, es decir, se proporcione al pin MCLR un estado lógico "1". El fabricante recomienda conectar en serie con el pulsador una resistencia de 100Ω , adicional a la resistencia de corto circuito que se recomienda sea de $10k\Omega$ (Fig. 3.4).

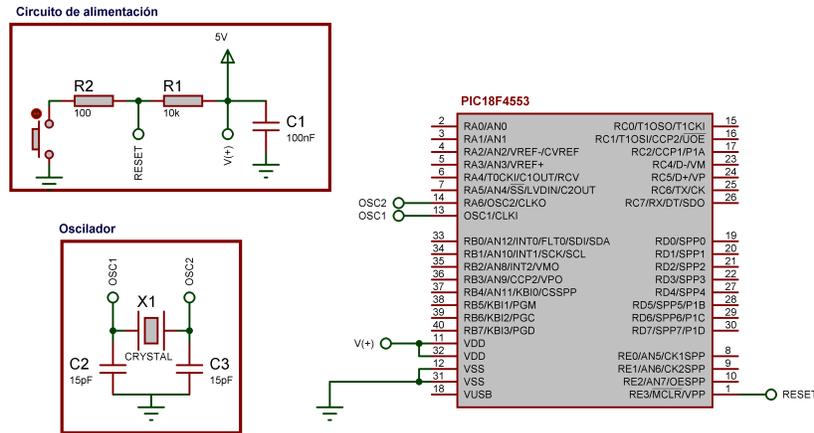


Fig. 3.4: Diagrama de conexiones básicas para PIC18F4553

El consumo de corriente para el funcionamiento del μ Controlador depende de la tensión de alimentación, de la frecuencia de trabajo, y de las cargas que soporten sus salidas. Como se detalla más adelante en el texto, para el manejo de los actuadores del sistema es necesario hacer uso de una etapa de potencia que reciba señales del μ Controlador.

3.3.3. Control de motores a pasos

Como se ha expuesto en las secciones anteriores, desde el μ Controlador es posible enviar la secuencia de señales de control necesaria para manejar los motores a pasos. Dichas señales controlan los interruptores electrónicos que energizan las distintas bobinas que componen al motor. La activación secuencial de dichas bobinas origina fuerzas de atracción y repulsión magnética sobre los polos que se encuentran en el rotor del motor a pasos, y de esta forma se provoca el avance del mismo. Los motores a pasos que se utilizan para actuar el sistema de ajuste que se diseña son de tipo unipolar (Fig. 3.5), la Tabla 3.2 muestra la secuencia de señales que debe ser enviada al interruptor de potencia conectado a cada una de sus bobinas.

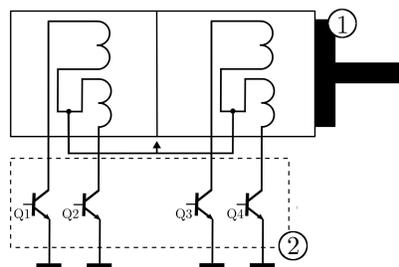


Fig. 3.5: Esquema de 1) motor a pasos unipolar y 2) dispositivo de control y potencia [72]

Tabla 3.2: Secuencia de conmutación de un motor a pasos unipolar con avance de pasos completos [3]

Paso	Bobina 1	Bobina 2	Bobina 3	Bobina 4
1	1	0	1	0
2	1	0	0	1
3	0	1	0	1
4	0	1	1	0

Sin embargo, con el fin de facilitar el desarrollo del software para el μ Controlador, disminuir el espacio de memoria que este utilizará, y aminorar la carga de trabajo del mismo, es posible emplear los denominados controladores de motores a pasos. Se trata de un circuito integrado (CI) que cuenta con la estructura de elementos electrónicos necesaria para emitir la secuencia de control de los motores a pasos al recibir una serie de instrucciones codificadas desde el microcontrolador. El L297 es un popular controlador en encapsulado DIP de 20 pines, capaz de controlar un motor a pasos unipolar en modo de paso completo o de medio paso. Para su funcionamiento requiere solamente de una señal de reloj que determine la frecuencia de envío de comandos de control, la dirección de giro deseada, y el modo de funcionamiento (paso completo o medio paso) [81]. La Fig. 3.6 muestra el diagrama de terminales del L297.

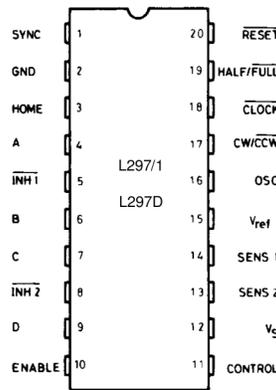


Fig. 3.6: Diagrama de terminales del controlador de motores a pasos L297 [81]

El CI cuenta con un pin de alimentación (V_s) que debe conectarse a una tensión de 5 V, así mismo el pin *GND* debe conectarse a tierra, y *RESET* debe enlazarse al pulsador de reset del μ Controlador. El resto de pines que serán conectados y sus funciones se muestran en la Tabla 3.3.

Tabla 3.3: Terminales del L297 que serán utilizadas para control de motores a pasos

Pin	Función
\overline{CLOCK}	Determina la frecuencia de envío de señales a los motores
$HALF/\overline{FULL}$	Modo de medio paso (“1” lógico) o paso completo (“0” lógico)
CW/\overline{CCW}	Sentido de giro horario (“1” lógico) o antihorario (“0” lógico)
$ENABLE$	Determina si el motor está en funcionamiento o no

Utilizando el L297 es necesario mandar desde el microcontrolador solamente tres señales, una que indica la dirección de giro y se conecta al pin CW/\overline{CCW} , otra que determina el modo de avance, bien sea paso completo o medio paso, pin $HALF/\overline{FULL}$, y una tercera señal que habilita o deshabilita el motor. Mediante estas es posible controlar los parámetros de que depende el movimiento del sistema con una programación más sencilla. Adicionalmente una señal de reloj debe ser enviada al L297 mediante el pin \overline{CLOCK} , esta señal determina la frecuencia con que serán enviadas las señales de control a los motores; en este caso se utiliza una salida PWM para poder controlar la frecuencia desde el μ Controlador. El diagrama de conexión de los L297 con los motores y el μ Controlador se muestra en la Fig. 3.8.

3.3.4. Alimentación de los motores

Como se ha descrito, el sistema se compone de cuatro actuadores, tres motores a pasos para posicionar el sistema de ajuste y girar las tuercas, y un servomotor para posicionar el mecanismo de ajuste. Los cuatro motores deben ser alimentados de energía y controlados por el μ Controlador. En el caso de los motores a pasos, para controlarlos se genera una secuencia de señales con ayuda de los L297, sin embargo, su salida es una señal, es decir, no es capaz de suministrar la potencia necesaria para energizar las bobinas del motor. Por lo tanto se hace necesario implementar una etapa de potencia, que se encargará de recibir la señal de control y accionar el motor tal como esta lo indique, con la capacidad de cumplir con los requerimientos energéticos del motor.

Tal como lo muestra la Fig. 3.5, un motor a pasos unipolar se acciona mediante el uso de cuatro transistores, que completan el circuito para energizar cada una de las bobinas. Este efecto puede lograrse mediante un arreglo tipo Darlington de transistores, que se encargarán, una vez reciban la señal, de otorgarle al motor la energía necesaria. El ULN2803A es un circuito integrado compuesto por un conjunto de ocho arreglos de transistores Darlington, cada uno de ellos es capaz de entregar una carga pico de 600 mA (500 mA en continuo). Las salidas pueden ser conectadas en paralelo para obtener una mayor corriente de salida [80]. El UNL2803A está preparado para recibir una señal de 5 V TTL o CMOS para activarse. Las Fig. 3.7a y 3.7b muestran el diagrama de terminales del UNL2803A y la disposición de los elementos internos para cada arreglo Darlington respectivamente.

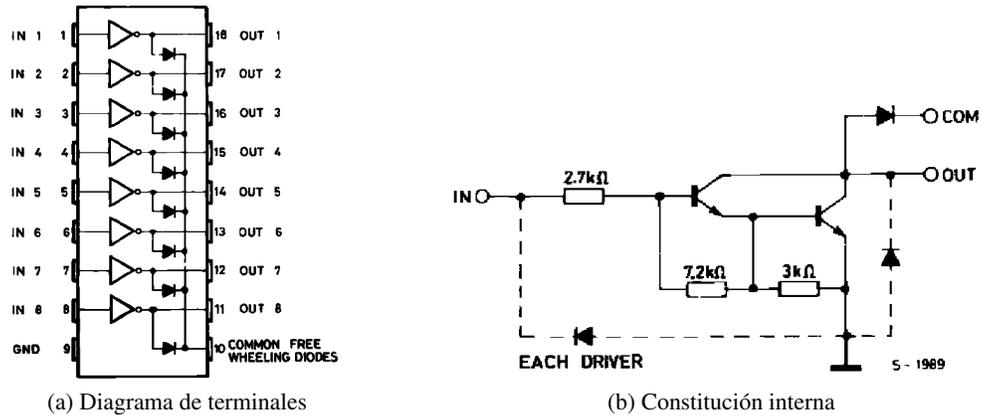


Fig. 3.7: Driver UNL2803A [80]

Mediante la utilización de este CI es posible conectar de forma segura las señales emitidas por los CI de control con la alimentación de los motores. Los diagramas de conexión se muestran en la Fig. 3.8, en donde es posible apreciar la interrelación de los distintos componentes necesarios para el manejo de los actuadores.

Por otra parte, el servomotor se alimenta directamente de la línea de tensión de 5 V que llega desde la fuente, así mismo, su terminal puesta debe conectarse a tierra. El servomotor cuenta en su interior con un CI que se encarga de controlar la posición que el motor adopta, basado en las instrucciones que recibe del μ Controlador. Dicha instrucción se envía mediante una señal de PWM, y al tratarse de una señal es posible conectar directamente la salida del μ Controlador a la entrada del servomotor.

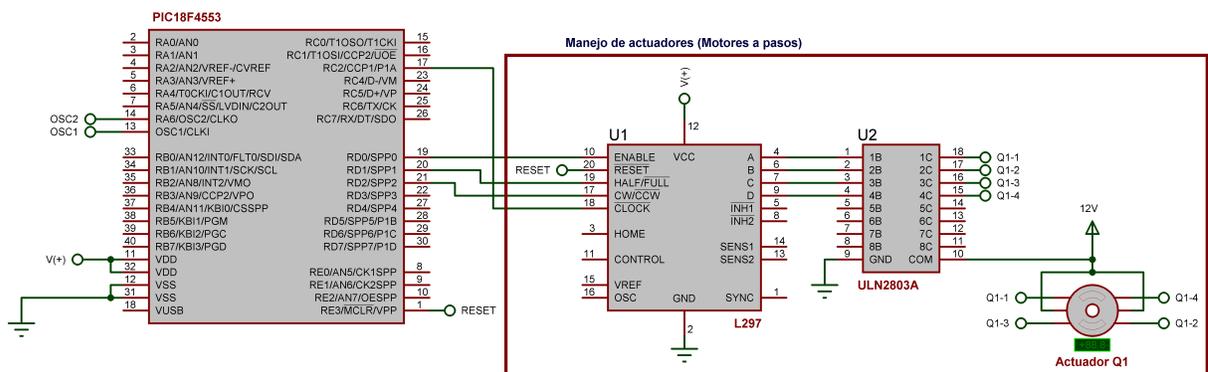


Fig. 3.8: Conexiones del μ Controlador para manejo de actuadores

3.3.5. Conexión de sensores

Para la detección de la posición límite de la placa móvil sobre la que se encuentra el mecanismo de ajuste, y el hardware del subsistema de visión computacional, se han colocado dos optointerruptores dentro de la pluma de actuación, en los extremos de la carrera de la placa. Se trata de dos interruptores

de ranura ópticos, este tipo de sensores tienen el emisor y el detector insertados en el paquete que sirve para mantener el alineamiento óptico. El emisor es un diodo LED infrarrojo, mientras que el detector es un fototransistor. Cuando se encuentran libres de interferencia, es decir, no hay objetos en la ranura, la luz emitida por el diodo LED del emisor alcanza la base del fototransistor y lo satura, mientras que cuando un objeto se interpone entre estos, es decir, se coloca en la ranura, el fototransistor no recibe más la excitación, y pasa a corte. Los estados lógicos obtenidos de esta interacción dependen de la configuración que se adopte al colocar la resistencia de corto circuito, puede hacerse bajo un esquema *pull-up* o *pull-down*. El MOC70 es un optointerruptor de gran disponibilidad comercial capaz de cumplir con las funciones requeridas. La configuración elegida se muestra en la Fig. 3.9, está planeada para entregar un “1” lógico al microcontrolador cuando exista una interferencia en la ranura, y un “0” lógico cuando no exista interferencia alguna.

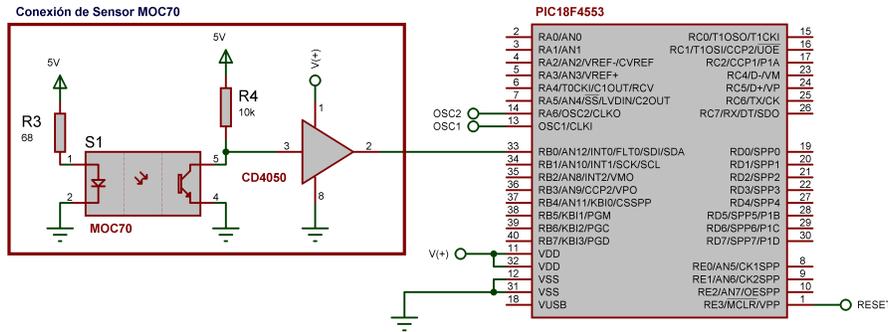


Fig. 3.9: Conexión del MOC70 al μ Controlador

De acuerdo con los datos proporcionados por el fabricante [22], el voltaje aplicado a las terminales del emisor debe ser $V_{LED} \geq 1.8 V$, y debe circular por él una corriente máxima de $i_{max} 50 mA$. Tomando en cuenta estos datos, se calcula la resistencia R_3 (Fig. 3.9) como:

$$R_3 = \frac{V_{cc} - V_{LED}}{i_{max}} = \frac{5 V - 1.8 V}{50 mA} = 64 \Omega, \quad (3.1)$$

se elige al valor comercial de resistencia más cercano,

$$R_3 = 68 \Omega. \quad (3.2)$$

Si bien el circuito diseñado para acoplar el optointerruptor es capaz de entregar señal en niveles TTL, se decide conectar un buffer no inversor (CD4050) para acoplar la impedancia entre la salida del sensor y la entrada del μ Controlador, garantizando así que la señal proveniente del optointerruptor sea capaz de accionar los pines de entrada.

3.4. Diseño de interfaz (API) con el subsistema de visión computacional

El *Bus Serie Universal* (USB) fue creado en los años noventa con la idea, entre otras, de mejorar las técnicas de “plug & play” que permiten a los periféricos conectarse y desconectarse sin la necesidad de reiniciar el dispositivo maestro con el que se comunica. Además, se aumentó la velocidad de transferencia de datos con respecto a los protocolos de comunicación comerciales de la época, alcanzando una velocidad de hasta 12 *Mbps* en su versión 2.0 de alta velocidad, y se le dotó de transmisión de energía para los dispositivos conectados.

Físicamente, los datos del USB se transmiten por medio de un par trenzado D+, D- (Fig. 3.10), además de la alimentación y la tierra. El USB es un protocolo de comunicación basado en el paso de testigo (token). El controlador USB distribuye testigos por el bus. El dispositivo cuya dirección coincide con la que porta el testigo responde aceptando o enviando datos al controlador. Cada vez que se conecta un dispositivo, el controlador USB especifica el ancho de banda, el tipo de transferencia, y el tamaño del paquete de datos, entre otras características, para el nuevo canal establecido. Esto se logra gracias a la especificación del dispositivo conectado [32].

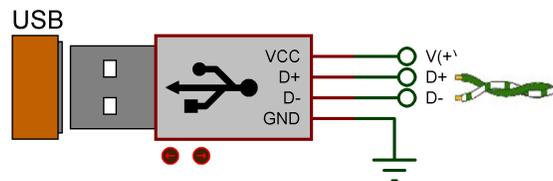


Fig. 3.10: Estructura de conector USB macho tipo A

Microchip ha introducido en algunos miembros de la familia PIC18 un motor de interfase serial USB con capacidades de alta y baja velocidad, que permite la comunicación entre cualquier dispositivo maestro y el μ Controlador [62]. En particular, para el desarrollo del sistema de ajuste resulta provechoso poder establecer comunicación entre el subsistema de visión computacional y el subsistema electrónico mediante un protocolo nativo en ambos subsistemas, además de las ventajas añadidas que el USB proporciona.

Una API es un conjunto de instrucciones mediante las cuales el usuario de una aplicación puede comunicarse con ella, para hacer uso de las funciones y capacidades de ésta en el momento que así lo requiera. Las ventajas de utilizar una API son principalmente tres: por un lado se reduce el volumen de las comunicaciones mediante el envío de una instrucción concisa y pequeña; es posible indicar a la aplicación la tarea que debe de ejecutar sin la necesidad de que el usuario transmita paso por paso el procedimiento a seguir, es decir, la codificación de la información acelera y simplifica la tarea, y al mismo tiempo se evitan ambigüedades. En segundo lugar, no es necesario que el usuario conozca a detalle los procesos internos necesarios para llevar a cabo la tarea deseada, solo debe conocer los comandos existentes y agruparlos de modo que estos completen el objetivo deseado, es decir, facilita

la planificación de las tareas. Finalmente, hace posible el mejoramiento y actualización del código de la aplicación principal, sin que esto repercuta en los usuarios o en el modo de manejo de la aplicación, pues las instrucciones de la API no necesitan ser modificadas cada vez que se realicen cambios, mejoras, o adaptaciones en el sistema.

El objetivo es diseñar un conjunto de instrucciones para que el subsistema de visión computacional se comunique de forma eficiente con el subsistema electrónico, para que éste comande los actuadores en la forma que el subsistema de visión computacional lo requiera. Dichas instrucciones deben ser transmitidas mediante el protocolo USB, por lo que han sido codificadas como un entero de ocho bits, es decir, cada vez que el subsistema de visión computacional necesite enviar una instrucción lo hará enviando un solo byte, que el subsistema electrónico interpreta (decodifica), y ejecuta la acción en este indicada.

El sistema de comunicación es bilateral pues se requiere que el subsistema electrónico informe acerca del estado de los sensores. La Tabla 3.4 presenta la codificación de ocho bits, el equivalente decimal, y el significado de cada una de las instrucciones que se envía desde la PC al subsistema electrónico. La Tabla 3.5 presenta los datos equivalente a la comunicación en sentido contrario.

Recordemos que el actuador q_1 es una motor a pasos que se encarga de hacer girar la pluma ala rededor del vástago de soporte, para posicionar angularmente al mecanismo de ajuste. q_2 , también un motor a pasos, se encarga por medio de un tornillo sin fin, de posicionar radialmente la placa móvil que contiene los componentes de visión computacional y del mecanismo de ajuste. El actuador q_3 es un servomotor encargado de controlar la posición vertical del mecanismo de ajuste. Finalmente q_4 , un motor a pasos montado en el extremo del mecanismo de ajuste, se encarga de hacer girar cada una de las tuercas que deben ser posicionadas.

El movimiento del actuador q_3 es un caso especial pues al ser un servomotor difiere del resto de los actuadores. Su control se basa en el envío de una señal de PWM al CI contenido dentro de éste, así se establece la posición angular que el sistema requiere que adopte. Debe recordarse que el intervalo del ángulo de salida del servomotor es de $[0^\circ - 170^\circ]$, mientras que $q_3 \in [0^\circ - 360^\circ]$, por lo tanto, una alternativa es enviar solamente el valor angular q_3 que se desea que el servomotor adopte, y dejar que el μ Controlador lleve a cabo las operaciones necesarias para hacer la conversión de valores. La segunda opción consiste en mandar directamente el valor de posición en el intervalo $[0^\circ - 170^\circ]$. En cualquier caso, es necesario enviar un valor numérico, en particular para el segundo caso, dicho valor estará en el intervalo $[0 - 180]$. Para evitar que un comando sea confundido con una posición angular del actuador q_3 , es necesario que se envíe primero un comando que especifica el inicio del envío de datos de posición. Al finalizar este proceso se envía una instrucción que establece el término de dicho estado, regresando al envío de comandos convencionales.

El código binario, y por consiguiente el valor decimal asociado, ha sido asignado a cada uno de los comandos dependiendo de la frecuencia relativa de uso de cada uno de ellos. De esta forma se disminuye el número de operaciones necesarias para reconocer cada uno de los comandos enviados.

Tabla 3.4: Comandos de la API para comunicación del subsistema de visión computacional con el subsistema electrónico

Binario	Decimal	Función
00000000	0	Todos los motores apagados
00000001	1	Giro de los actuadores q_1 y q_2 en sentido horario, modo paso completo
00000010	2	Giro de los actuadores q_1 y q_2 en sentido antihorario, modo paso completo
00000011	3	Giro del actuador q_1 en sentido horario, modo medio paso
00000100	4	Giro del actuador q_1 en sentido antihorario, modo medio paso
00000101	5	Giro del actuador q_2 en sentido horario, modo medio paso
00000110	6	Giro del actuador q_2 en sentido antihorario, modo medio paso
00000111	7	Giro del actuador q_1 en sentido horario, modo paso completo
00001000	8	Giro del actuador q_1 en sentido antihorario, modo paso completo
00001001	9	Giro del actuador q_2 en sentido horario, modo paso completo
00001010	10	Giro del actuador q_2 en sentido antihorario, modo paso completo
00001011	11	Giro de los actuadores q_1 y q_2 en sentido horario, modo medio paso
00001100	12	Giro de los actuadores q_1 y q_2 en sentido antihorario, modo medio paso
00001101	13	Giro del actuador q_4 en sentido horario, modo paso completo
00001110	14	Giro del actuador q_4 en sentido antihorario, modo paso completo
00001111	15	Giro del actuador q_4 en sentido horario, modo medio paso
00010000	16	Giro del actuador q_4 en sentido antihorario, modo medio paso
00010001	17	Inicio del periodo para comunicación de posición deseada del actuador q_3
10110101	181	Finalización del periodo para comunicación de posición deseada del actuador q_3
<i>xxxxxxxx</i>	[0 – 180]	Posición angular deseada en el actuador q_3

Tabla 3.5: Comandos de la API para comunicación del subsistema electrónico con el subsistema de visión computacional

Binario	Decimal	Función
00000001	1	Sensor de inicio activado. Placa móvil completamente retraída
00000010	2	Sensor de fin activado. Placa móvil completamente desplegada
10000000	128	Falla general del sistema, se requiere “reset”
11111111	255	Sistema en orden y preparado
00010010	18	Confirmación de inicio de periodo para posicionamiento del actuador q_3
00010011	19	Confirmación de finalización del periodo para posicionamiento del actuador q_3

3.5. Software y circuito de control

Finalmente, una vez seleccionados los componentes electrónicos que integrarán el subsistema en cuestión, diseñados los enlaces entre estos, establecido el protocolo de comunicación, y las pautas de fun-

cionamiento, es posible concluir con el desarrollo de este subsistema. La Fig. 3.11 muestra el esquemático del subsistema electrónico completo, que integra los diversos CI empleados, el μ Controlador, la interfaz física del sistema de comunicación con la PC, así como las terminales por las que se hará llegar la energía al subsistema y aquellas que interconectan la PCB con los motores y sensores.

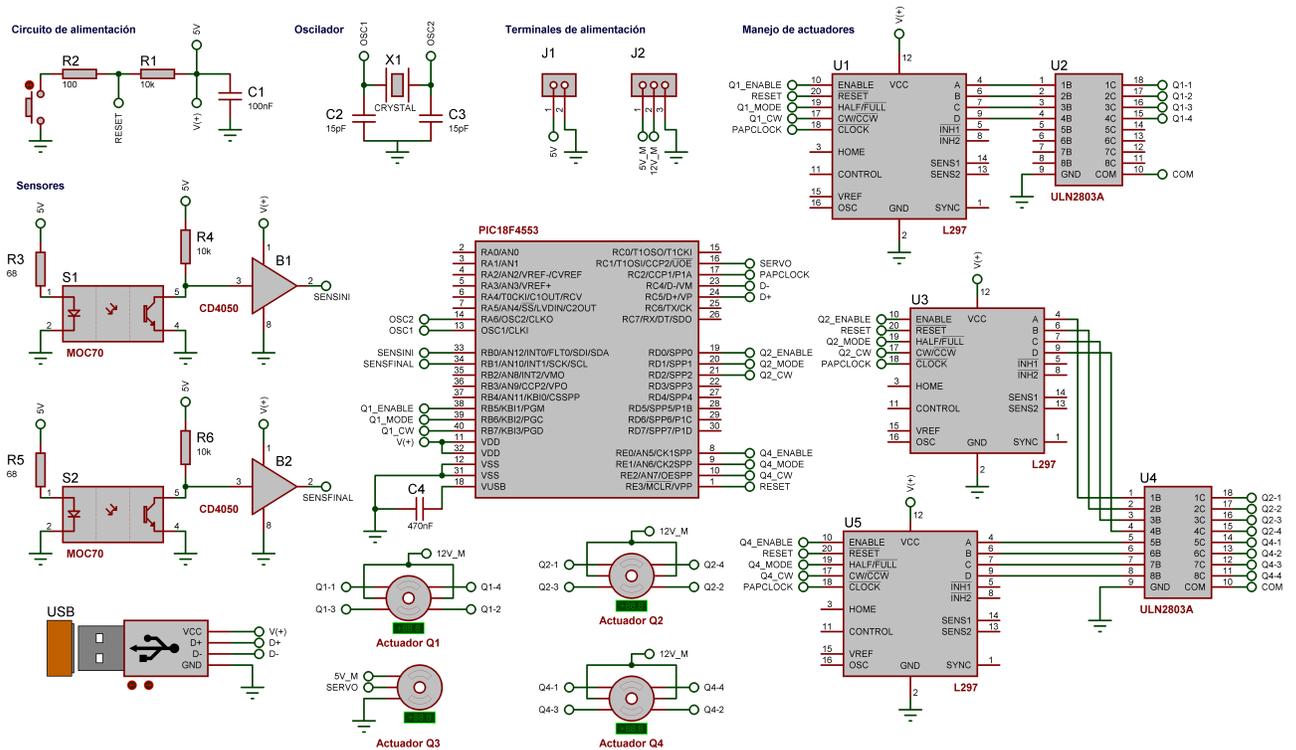


Fig. 3.11: Esquemático del subsistema electrónico

La interconexión de todos los componentes requiere de un sustento físico. Con este fin ha sido diseñada una tarjeta de circuito impreso (Fig. 3.12) que alberga los elementos y conectores, ésta representa el módulo principal del subsistema electrónico y se alimenta directamente de la fuente de poder del sistema. Adicionalmente se desarrolló una PCB para albergar a los sensores MOC70, así como sus terminales de conexión con la electrónica adicional que garantiza su funcionamiento (Fig. 3.13).

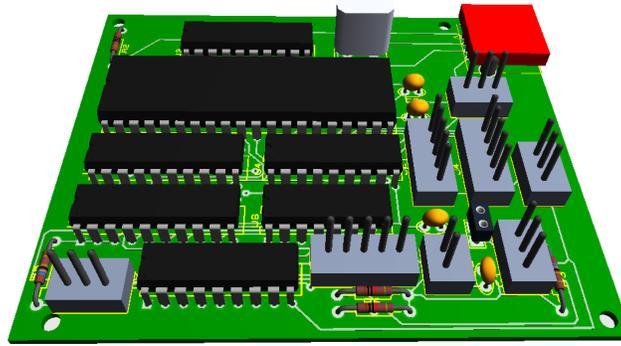


Fig. 3.12: Subsistema electrónico. Módulo principal

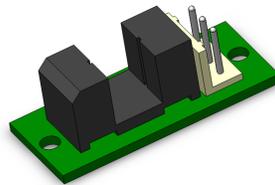


Fig. 3.13: PCB para MOC70

Los μ Controladores PIC18 poseen múltiples fuentes de interrupción. Esta característica es de especial interés cuando el sistema diseñado requiere de una repuesta inmediata por parte del μ Controlador ante la ocurrencia de un determinado evento. En particular, para este desarrollo serán utilizadas tres fuentes de interrupción, estas son: interrupción por recepción de datos en USB, interrupción por cambio de bit en la terminal RB0/INT0, y finalmente interrupción por cambio de bit en la terminal RB1/INT1.

A continuación se presenta una descripción del software desarrollado para ejecutarse en el μ Controlador, el cual ha sido diseñado con base en la ocurrencia de las interrupciones. Las interrupciones por cambio de bit en las terminales a donde se conectan los sensores son declaradas con alta prioridad, mientras que el resto son de baja prioridad, de esta forma se atenderá a la activación de los sensores por sobre todas las otras instrucciones, evitando así daños en el sistema.

1. Inicialización del sistema. Se declaran los bits RB0 y RB1 como entradas digitales para la lectura de los optointerruptores, los bits RB5-RB7, RD0-RD2 y RE0-RE2 como salidas digitales para el control de los motores a pasos. Se inicializa el módulo de PWM en los canales CCP1 y CCP2 para la emisión de señal de reloj a los L297 y para el control del servomotor, respectivamente. Se inicializa el módulo de comunicación USB. Se retrae el servomotor a la posición inicial (0°) y se hace girar actuador q_2 para llevar a la placa móvil hasta accionar el sensor del extremo de la pluma. Acto seguido se gira el actuador en sentido contrario hasta que se accione el sensor más próximo al vástago de soporte. Una vez se lleva a cabo este proceso se envía a la PC, vía USB, el comando 11111111 “sistema en orden y preparado”. En caso de que la placa no

active alguno de los sensores dentro del tiempo esperado, se envía el comando 10000000 “falla general del sistema, se requiere reset”.

2. Si no existe ninguna llamada a función por parte de las interrupciones, o algún proceso derivado de ellas ejecutándose, el sistema pasa a modo de espera.
3. Si se produce un cambio de bit en los puertos RB0 o RB1, es decir, la activación o desactivación de alguno de los sensores, se actualizan las variables globales booleanas SENSOR_INI o SENSOR_FIN dependiendo del caso de activación. Tomarán el valor 1 si el sensor se ha activado, o el valor 0 si no hay interferencia en estos. Si el estado resultante corresponde a una activación, se detienen todos los motores, poniendo en “0” lógico su bit ENABLE (RB5, RD0, RE0) y se mantiene el servomotor en la última posición enviada. En caso de que se trate de una desactivación, es decir, la placa sale del rango del sensor, solamente se actualiza el valor de éste y termina la labor de la interrupción. En los casos de activación, al finalizar los procesos descritos, se envía a la PC un informe de la situación, mediante los comandos 00000001 “Sensor de inicio activado. Placa móvil completamente retraída”, o 00000010 “Sensor de fin activado. Placa móvil completamente desplegada”.

Si se detecta un activación simultánea de los sensores, indicaría que un error ha tenido lugar, todos los actuadores se desactivan y se envía el comando 10000000 “falla general del sistema, se requiere reset” a la PC.

4. Si se activa la interrupción por recepción de comunicación USB, el comando recibido (1 byte) es el parámetro con que se alimenta un proceso de identificación de comandos, desde el cual se hace la llamada a la rutina de control correspondiente, para satisfacer las necesidades del subsistema de visión computacional. Antes de comenzar a buscar la función de llamada que corresponde a cada comando, es necesario comprobar el estado de la variable global PWM_ON, en ésta se indica si el comando debe ser interpretado de manera directa, o si se trata de un posición deseada para el actuador q_3 .
5. Si el comando recibido es 00010001 “inicio del periodo para comunicación de posición deseada del actuador q_3 ”, se fija la variable PWM_ON = 1, y se comunica a la PC la aceptación de la instrucción mediante el comando 00010010 “confirmación de inicio de periodo para posicionamiento del actuador q_3 ”. De esta forma, los comandos que sean enviados desde la PC se interpretarán a partir de este momento como posiciones deseadas para el actuador q_3 . En el mismo sentido, si se recibe el comando de finalización de envío de posiciones, se fija PWM_ON = 0 y se confirma a la PC del cambio de estado.

- a) En el caso de que el comando recibido corresponda a instrucciones para el movimiento de los motores a pasos, se llevan a 0 lógicos los bits ENABLE de los motores que no participen en la ejecución del comando.

Posteriormente, si está involucrado el actuador q_2 , debe comprobarse el valor de las variables SENSOR_INI y SENSOR_FIN, dependiendo del sentido de giro solicitado. Si el

sentido de avance no es compatible con el estado de los sensores, se envía un mensaje del estado de los sensores a la PC mediante los comandos antes descritos. Si, por otro lado, q_1 es el actuador que se debe activar, se retrae el mecanismo de ajuste hasta una altura segura para evitar colisiones, y se concede un tiempo muerto para que el servomotor alcance la posición deseada.

Finalmente, se lleva a 1 lógico el bit ENABLE de los motores correspondientes, se establece el estado correspondiente de los bits de sentido de giro y modo de avance, y se mantiene esta configuración hasta que un nuevo comando se reciba, o salte la interrupción de cambio de bit en la señal de alguno de los sensores.

CAPÍTULO 4

SUBSISTEMA DE VISIÓN COMPUTACIONAL. HARDWARE

*La imaginación es más importante que el conocimiento.
El conocimiento es limitado, mientras que la imaginación no*
[ALBERT EINSTEIN]

4.1. Definición de funciones

El subsistema de visión computacional es un conjugado de hardware y software encargado de recopilar información del medio, planear el desarrollo de las tareas asignadas, emitir órdenes para llevarlas a cabo, y cerciorarse del cumplimiento de las mismas. Para llevar a cabo el trabajo descrito cuenta con sensores de visión que adquieren la información del mundo real, y que actúan a la vez como mecanismo de retroalimentación.

Las funciones específicas del subsistema de visión computacional son:

1. Detectar e identificar las tuercas de ajuste alojadas sobre la estructura del concentrador, basándose para ello en las marcas de color colocadas sobre ellas.
2. Determinar la posición inicial del sistema dentro de la configuración global de tuercas, con base en imágenes captadas al inicio de ciclo de ajuste.
3. Emitir las órdenes necesarias para llevar al sistema a la posición inicial de trabajo.
4. Planear la progresión de los ajustes para completar la tarea, a partir de la posición inicial de trabajo.
5. Determinar las coordenadas tridimensionales de la tuerca que se desea ajustar en cada momento.

6. Transmitir la secuencia de órdenes de movimiento necesarias para alcanzar la siguiente tuerca por ajustar.
7. Ajustar la elevación de las tuercas y corroborar que la posición alcanzada sea la deseada.
8. Dotar de un contenedor rígido para albergar el arreglo estereoscópico de cámaras, capaz de atenuar su vibración, y que garantice una configuración geométrica constante, es decir, que mantenga las cámaras con su posición relativa constante durante la operación del sistema.

4.2. Requerimientos de hardware

El módulo de visión computacional, tal como su nombre lo sugiere, debe contar con la capacidad de extraer imágenes del mundo real que puedan ser procesadas para obtener una interpretación del medio. Si bien la integración a éste de un arreglo de sensores fotoeléctricos es fundamental, existen algunos requerimientos especiales que obedecen no solo a la propia naturaleza de la tarea asignada, sino, al objetivo de desarrollar un sistema de visión computacional autocontenido, de fácil montaje, robusto, y con una estructura de software modular que le permita añadir funciones que se comuniquen con el resto del desarrollo de forma sencilla.

El módulo de visión computacional se conforma de hardware y software. Los requerimientos y el desarrollo del software se presentan en el siguiente capítulo. En ésta sección se describen los requerimientos que imperan en el diseño del hardware del subsistema de visión computacional.

- Contenedor rígido que albergue y proteja los componentes del subsistema de visión computacional.
- El peso del contenedor debe ser del mismo orden de magnitud que el peso del par estereoscópico de cámaras, para no incrementar de forma considerable el peso total del subsistema.
- El contenedor debe brindar el soporte para fijar firmemente el par de cámaras con el fin de evitar movimientos relativos entre estas, garantizando así una configuración geométrica constante.
- La frecuencia natural de vibración del contenedor debe distar por un margen razonable de la frecuencia natural de los actuadores del sistema, con el fin de evitar que el arreglo de cámaras entre en resonancia.
- Cámaras con resolución apropiada para llevar a cabo la tarea de ajuste, que cuenten con una tasa de cuadros por segundo que minimice el número de tomas borrosas y se mantenga bajo un límite procesable, con interfaz natural al equipo de procesamiento (PC), y que brinden la capacidad de extraer la información de ellas mediante software propio, sin intermediarios.
- Puntos de anclaje en la estructura del contenedor para ajustarlo firmemente al sistema global, de forma sencilla y sin tener contacto directo con el sistema de cámaras.

4.3. Diseño del contenedor estereoscópico

Para llevar a cabo la tarea asignada al subsistema de visión computacional se ha debido seleccionar un modelo de cámara apto para ello. La cámara seleccionada es la PS3 EYE de Sony (Fig. 4.1), cuenta con una tasa de muestreo de 60 FPS^1 a una resolución de $640 \times 480\text{ pixeles}$, y de 120 FPS con resolución de $320 \times 240\text{ pixeles}$ [84].



Fig. 4.1: Cámara PS3 EYE

La lente proporciona dos configuraciones de enfoque con ajuste manual. La primera genera un campo de visión de 75° , y la segunda un campo de visión de 56° . La PS3 EYE es capaz de enviar la señal de video sin compresión, cuenta con interfaz de comunicación USB, capacidad de obtener la información captada mediante varios lenguajes de programación, y una relación costo beneficio que la vuelve el candidato idóneo para labores de visión computacional como la desarrollada en este trabajo.

La PS3 EYE es distribuida por Sony con un valor aproximado de 50 USD, y ha sido especialmente desarrollada para usarse como interfaz visual para la consola PS3, es decir, desde su concepción ha sido enfocada para aplicaciones de visión computacional.

La cámara está protegida por una carcasa de plástico, tal como se aprecia en la Fig. 4.1. Dentro de esta se encuentra la PCB que alberga el arreglo de sensores fotoeléctricos, la lente, y la electrónica adicional de la cámara (Fig. 4.2).

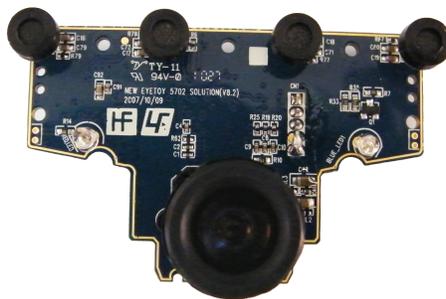


Fig. 4.2: PCB de la cámara PS3 EYE

Para formar el arreglo estereoscópico con que se trabajará se utilizan dos cámaras PS3 EYE. Dado

¹Cuadros por segundo (*frames per second*)

que se busca un sistema estereoscópico ligero, es conveniente retirar la carcasa protectora dejando desnuda la PCB, eliminando así peso de la cámara. La Fig. 4.3 muestra un modelo tridimensional creado con las dimensiones reales de la cámara, utilizado para el diseño del contenedor de las mismas. Como se observa en la Fig. 4.3, el hardware de la cámara cuenta con un arreglo de cuatro micrófonos, protegidos mediante una pieza plástica diseñada para tal fin.

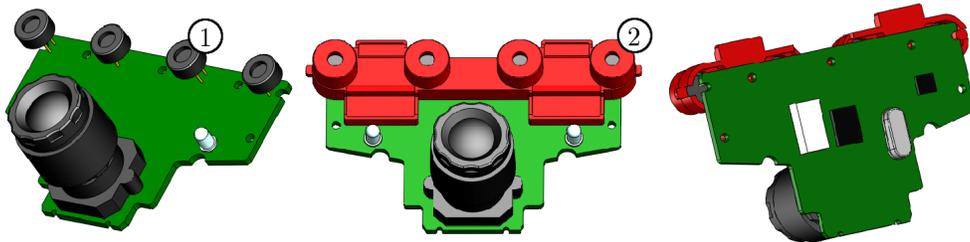


Fig. 4.3: Modelo virtual de la cámara PS3 EYE: 1) micrófonos, 2) protector de micrófonos

Dicho protector puede ser utilizado para fijar la posición relativa entre las cámaras, enlazando el par de micrófonos de cada una de la cámaras que colinda con la cámara opuesta, tal como se muestra en la Fig. 4.4. Una vez se tienen las dimensiones del arreglo estereoscópico, es posible comenzar el diseño del contenedor que lo albergará.

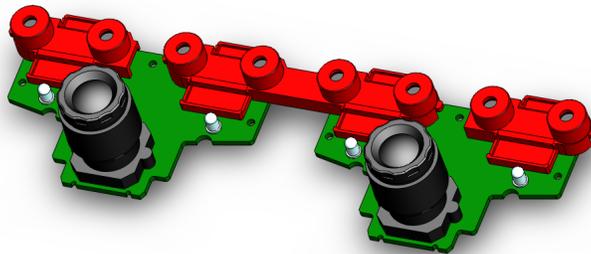


Fig. 4.4: Cámaras enlazadas mediante protector de micrófonos

La aproximación obvia es pensar en un contenedor rectangular que aloje las cámaras, tal como lo muestra la Fig. 4.5. Sin embargo, por la propia configuración geométrica del sistema, un rectángulo generaría grandes áreas vacías (zonas grises en Fig. 4.5), que representan alrededor del 35 % del área total del contenedor rectangular, lo que se traduce en peso innecesario para el sistema, contraviniendo el objetivo de diseñar un contenedor ligero.

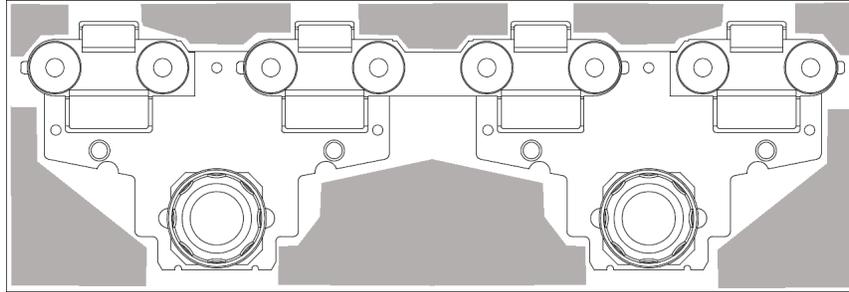


Fig. 4.5: Contenedor rectangular para cámaras

Con base en las afirmaciones anteriores, se modificó la geometría del contenedor para disminuir el número de espacios no utilizados, conservando solo algunas áreas libres para no perder la capacidad de instalar, en desarrollos futuros, hardware que agregue nuevas funciones. El resultado del rediseño se muestra en la Fig. 4.6, donde se aprecia la geometría en planta del contenedor estereoscópico.

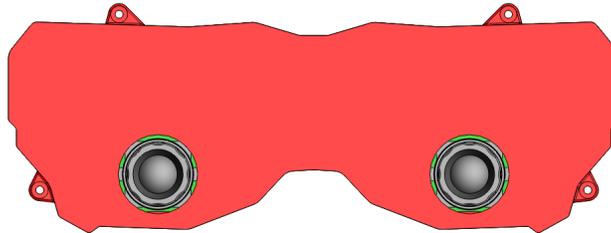


Fig. 4.6: Geometría en planta del contenedor estereoscópico

El siguiente paso es crear el volumen necesario para que las cámaras puedan insertarse en el contenedor. Una vez que se ha establecido dicha dimensión, el contenedor se divide para formar la parte superior e inferior de la máscara. Naturalmente, el contenedor es hueco, pero el espesor de las paredes que los conforman es importante, pues de ser grande agregaría peso innecesario, y en el caso contrario lo volvería frágil. Se decide utilizar 2 mm de espesor, pues se trata de una dimensión que cumple con los requerimientos especificados, y además, es factible desde el punto de vista de la manufactura, considerando que el material del contenedor es aluminio, un metal económicamente accesible, cuya razón de peso resistencia es ideal para este desarrollo. Al dividir la máscara es necesario crear una ceja que sirva como guía al ensamble y fije la posición de una pieza con respecto a su opuesta, tal como se aprecia en la Fig. 4.7. En ambas partes del contenedor se crean redondeos para disminuir el material utilizado, con esto se logra aligerar el componente superior en un 10.4% , y el inferior en 6.2% , también se eliminan ángulos que podrían dificultar el maquinado. Adicionalmente, se agregan cuatro anclas al componente inferior que albergan los tornillos encargados de fijar el contenedor estereoscópico a la estructura portante. Los tornillos utilizados son ANSI métrico M3 de cabeza hueca hexagonal. Finalmente es necesario crear un conducto de salida para los cables que comunican las cámaras con el sistema de procesamiento. Así mismo, se requiere un componente adicional que selle el conducto y mantenga en posición el cableado (Fig. 4.7). La tapa del conducto para cables se fija al

componente inferior mediante dos tornillos de 1.3 mm de diámetro.

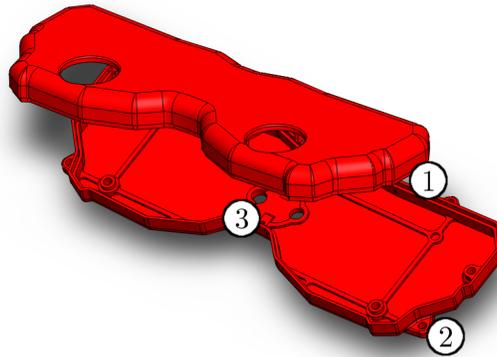


Fig. 4.7: Componente superior e inferior del contenedor: 1) ceja, 2) anclajes, 3) conducto para cables

Una vez que la geometría externa del contenedor ha quedado completamente definida, es tiempo de diseñar el interior; el objetivo es fijar firmemente las cámaras al contenedor. Para tal fin, se dispone de tres barrenos en cada una de las PCBs de las cámaras, a través de los cuales se insertarán los tornillos de 1.3 mm de diámetro que las anclarán a la estructura de la máscara. Se definen los puntos de anclaje y la geometría interna de la máscara para que las PCBs sean albergadas, así como los conductos para que los cables que parten desde estas puedan llegar al exterior. El diseño final de la parte interna del componente superior del contenedor se muestra en la Fig. 4.8.

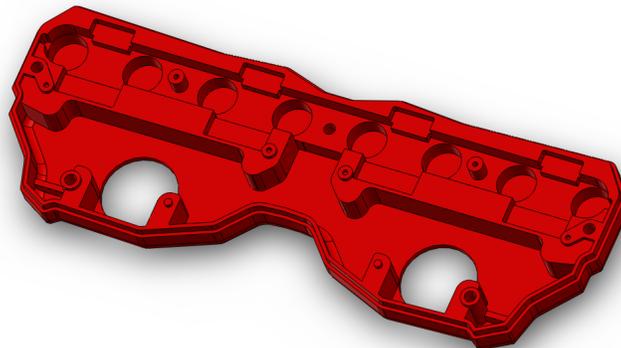


Fig. 4.8: Diseño del componente superior del contenedor estereoscópico

El diseño desarrollado incluye los barrenos necesarios para unir el contenedor superior e inferior de la máscara con tornillos ANSI métrico M3 de cabeza hueca hexagonal, de 12 mm de longitud. La masa del contenedor diseñado, sin incluir los tornillos, es de 155.4 g . El material elegido para el contenedor es la aleación de aluminio 6063 T-5.

4.4. Optimización masa - frecuencia del contenedor

Uno de los aspectos más importantes a considerar en el diseño del contenedor estereoscópico es su masa, pues si se presenta en exceso podría dificultar el movimiento del sistema de ajuste o de cualquier otro sistema con el que trabaje. La alternativa lógica es reducir la masa adelgazando secciones de la máscara, o creando orificios en las zonas en donde no sea necesario un determinado volumen de material, desde el punto de vista del comportamiento mecánico. Sin embargo, la extracción deliberada de material puede acarrear problemas de importante consideración por las funciones con las que cumple el contenedor. En particular, retirar material de la máscara de forma aleatoria puede disminuir su rigidez, lo que conlleva una disminución en el valor de su frecuencia natural de vibración, haciéndola propensa a entrar en resonancia a causa de la vibración de los actuadores, o del propio movimiento del sistema. Es éste el efecto que se debe tener en cuenta de forma prioritaria, pues si bien el contenedor no recibe cargas estáticas o dinámicas externas de consideración, está afectado por la fuerza de inercia producto de su masa acelerada, por la vibración de los actuadores, y por la vibración del propio sistema, todas ellas, fuentes de carga dinámica. Por tanto, un análisis estático de resistencia no arrojaría información útil en comparación con un análisis de la frecuencia natural de vibración.

La función del contenedor estereoscópico es definir una configuración geométrica constante de las cámaras, y protegerlas. Las vibraciones excesivas contravienen ambos objetivos, pues por un lado provocan que la información visual captada sea inutilizable, y por otro, pueden dañar físicamente las cámaras debido a las deformaciones generadas.

Cuando un cuerpo sólido es alterado de su posición de reposo tiende a vibrar a ciertas frecuencias, las cuales se denominan frecuencias naturales o resonantes [14]. Para cada frecuencia natural el sólido se deforma en una determinada forma, la cual recibe el nombre de forma modal, y está relacionada con la dirección de las acciones que se ejercen sobre el sólido, así mismo con la frecuencia con que estas se producen.

Cuando un sólido es excitado adecuadamente por una carga dinámica, con una frecuencia que coincide con una de sus frecuencias naturales, este sufre grandes deformaciones y esfuerzos; fenómeno denominado resonancia. Un modelo real tiene un cantidad infinita de frecuencias naturales, sin embargo, un modelo de elementos finitos tiene una cantidad finita de frecuencias, que es igual a la cantidad de grados de libertad considerados en el modelo. Para evaluaciones prácticas solo es necesario conocer los principales modos de vibración.

La vibración que puede propagar un actuador por el resto del sistema se debe principalmente al giro de su rotor. Los motores utilizados en robótica tienen velocidades del orden de los miles de revoluciones por minuto (RPM). La frecuencia natural correspondiente a cada uno de los modos de vibración de un sólido se expresa comúnmente en Hertz (Hz), por lo tanto, es conveniente convertir las RPM a Hz, de esta forma es posible comparar las capacidades del contenedor con las solicitaciones a las que se le someterá. Una revolución equivale a un ciclo, por lo tanto, basta con dividir la velocidad en RPM entre 60 s, de esta forma obtendremos revoluciones por segundo o Hertz. Así, la frecuencia de vibración que

transmite un actuador se encuentra en los cientos de Hz. Por lo tanto, el diseño del contenedor debe tener la rigidez necesaria para que la frecuencia de vibración de cada uno de los modos principales esté sobre un límite seguro, por encima de los cientos de Hz.

Las frecuencias naturales y sus formas modales correspondientes dependen de la geometría, las propiedades del material, y las condiciones de soporte. Haciendo uso del simulador de fenómenos mecánicos basado en análisis de elemento finito, SolidWorks Simulation [14], se llevó a cabo el análisis de frecuencia. Las restricciones impuestas al ensamble son:

- Geometría fija en los barrenos de las anclas del componente inferior del contenedor, imitando la función de los tornillos, es decir, restringiendo el giro y la traslación.
- Geometría fija en la base de las anclas del componente inferior del contenedor. La restricción se aplica únicamente al área de la base del ancla circundante al tornillo, restringiendo el movimiento por la acción del tornillo y por el contacto con una superficie rígida. El resto de la base del contenedor no se restringe de forma explícita.

De manera semejante es necesario imponer condiciones de contacto entre los componentes del ensamble, estas son:

- Unión rígida entre los distintos componentes en contacto, garantizando la unión firme y la no penetración del material.
- Unión de pasador con restricción al giro y la traslación en cada uno de los barrenos de unión del componente inferior y superior, así como en los barrenos entre el componente inferior y la tapa del conducto para cables.

El proceso de optimización consiste en disminuir el peso del contenedor retirando material de aquellas zonas en que no se requiere, cuidando en cada iteración que la frecuencia natural de vibración del ensamble no caiga por debajo del límite seguro. El proceso finaliza cuando no sea posible retirar más material por haber alcanzado la configuración de soportes indispensable, porque la frecuencia natural decrezca a niveles indeseados, o por una geometría que dificulte la manufactura. Debido a que la frecuencia de vibración de los actuadores está definida en función de su orden de magnitud, para este caso se considera el límite de seguridad como una frecuencia de vibración de al menos un orden de magnitud superior a la de los actuadores, es decir, el contenedor debe tener frecuencias en el orden de los miles de Hz.

Los resultados del análisis de frecuencia en cada una de las iteraciones se muestran en la Fig. 4.9. El eje horizontal muestra el número de iteración, la masa del contenedor, y el porcentaje de disminución de la masa con respecto al contenedor original (iteración 0), mientras que el eje vertical muestra la frecuencia natural de vibración de cada uno de los cinco modos principales. La Fig. 4.10 muestra un esquema del contenedor deformado por el efecto de la resonancia, para cada uno de los cinco principales modos de vibración.

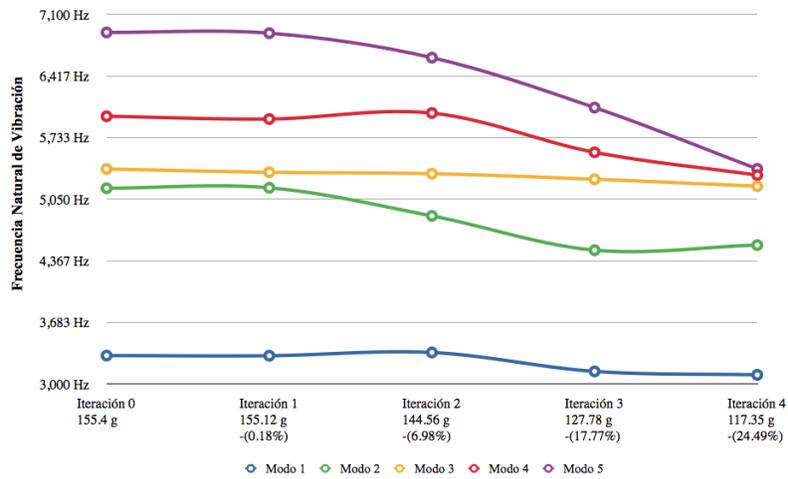


Fig. 4.9: Frecuencias de vibración del contenedor estereoscópico para cada iteración

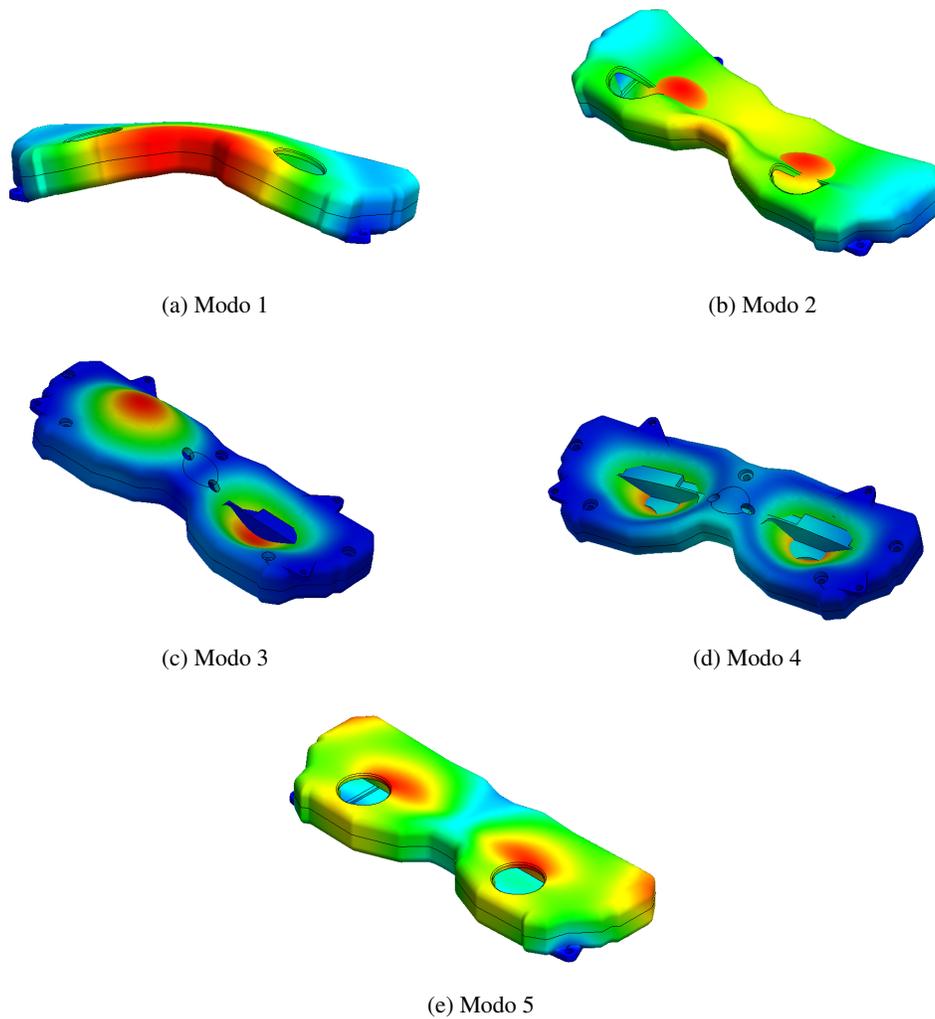


Fig. 4.10: Modos de vibración principales del contenedor estereoscópico

El proceso de disminución de masa concluye tras cuatro iteraciones; se consigue la configuración deseada en los soportes internos del componente superior del contenedor. Se logra una disminución en la masa del contenedor del 24.49 % con respecto al diseño original, pasando así de 155.4 *g* a 117.35 *g*. Se consigue un ahorro de 38.05 *g*, y el valor mínimo de frecuencia de entre los cinco modos principales es de 3, 102.2 *Hz*, dentro del límite de seguridad.

El diseño final de los componentes del contenedor se muestra en la Fig. 4.11.

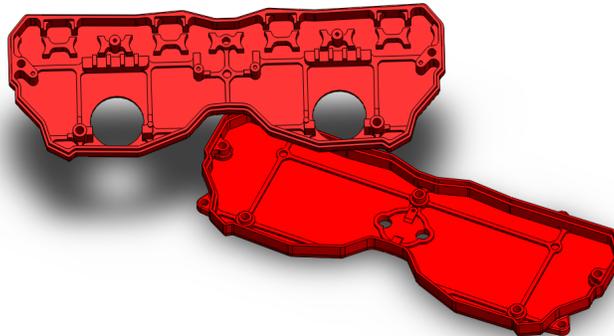


Fig. 4.11: Componentes del contenedor estereoscópico

CAPÍTULO 5

FUNDAMENTOS DE VISIÓN COMPUTACIONAL

*Los que se enamoran de la práctica sin la teoría son como los pilotos sin timón ni brújula,
que nunca podrán saber a adónde van*
[LEONARDO DA VINCI]

El sentido de la vista permite a los seres vivos percibir y entender, a través de los rayos de luz que alcanzan sus ojos, el medio ambiente en que se desenvuelven [83]. La visión es sin lugar a dudas el sentido que prevalece por sobre el resto de los sentidos de que el ser humano dispone [15]. Es por medio de éste canal que se recibe la mayor parte de la información necesaria para el desarrollo y la supervivencia de los organismos vivos [36], además, facilita la interacción con el medio tendiendo un puente entre la percepción y el control del movimiento [97]. Se estima que el monto de información recibida por medio del sistema visual humano es al menos dos órdenes de magnitud mayor que el recibido por cualquiera de los sentidos restantes [15].

La visión artificial, o visión computacional, es una rama de la ciencia enfocada en la extracción de información acerca del medio físico a partir del análisis y la interpretación de las imágenes de este. En otras palabras, busca duplicar el efecto de la visión humana mediante la percepción y el procesamiento digital [83, 96].

El objetivo de la visión artificial es ambicioso y complejo [83], pues desea imitar la capacidad de un sistema con cientos de millones de años de evolución, además de enfrentarse a fenómenos como:

- La pérdida de información al pasar del mundo tridimensional a una representación en dos dimensiones.
- El ruido electrónico de los sensores de captura, que distorsiona la percepción del mundo físico.
- La enorme cantidad de información recibida en una imagen o en un flujo de video, y la necesidad de ofrecer análisis en tiempos relativamente cortos.

- El gran número de factores que intervienen en la medición de la intensidad luminosa.
- La necesidad de generar una interpretación global del medio visto a través de una ventana local, con base en la experiencia y los conocimientos previos.

Sin embargo, las ventajas que ofrece el dotar a los sistemas artificiales con la capacidad de ver motiva el desarrollo de la visión computacional. Los sistemas que cuentan con esta capacidad son inherentemente adaptables y versátiles, están preparados para lidiar con la incertidumbre del ambiente [97]. La visión computacional añade seguridad, confiabilidad, consistencia, retroalimentación, y la posibilidad de la automatización flexible.

Gracias al surgimiento de equipos de cómputo más potentes, nuevos desarrollos en el área, y las ventajas que provee la visión computacional, los campos de aplicación crecen rápidamente. Actualmente es posible encontrar sistemas de visión en aplicaciones de inspección y vigilancia [36, 97], navegación autónoma [37, 60, 89], metrología [29], medicina [7], meteorología [16], geotecnia [73], mapeo y reconstrucción 3D [57, 59, 77, 87, 89, 90, 93], control de acceso [25, 82], agricultura [6, 55, 77, 103], acuicultura [99], reconocimiento de patrones [2, 7, 25, 53, 104], análisis estructural [8], ensamblaje automático [76], entre otras.

El creciente número de aplicaciones basadas en visión computacional marca una clara tendencia hacia la adopción masiva de este tipo de tecnología. El futuro estará, en gran medida, marcado por los desarrollos en visión computacional.

A continuación se presenta la nomenclatura relacionada con los conceptos manejados en el presente trabajo, y que será utilizada en el desarrollo e implementación de los algoritmos diseñados.

5.1. Imagen

Una imagen digital es una representación discreta del espacio, cada uno de sus elementos es denominado pixel¹, y almacena el valor de la intensidad lumínica captada por la celda correspondiente del sensor de imagen [34].

Para el caso de imágenes en escala de grises, el conjunto de valores que describe a la imagen se guarda en forma de arreglo bidimensional, pues para cada pixel existe un único dato que representa la intensidad lumínica percibida. En el caso de las imágenes a color, se perciben para cada punto del espacio tres intensidades luminosas, correspondientes a las tres principales longitudes de onda captadas por el ojo humano, es decir, rojo, verde y azul. Este tipo de imágenes son denominadas multibanda o multi-canal, pues representan la imagen como un conjunto de tres arreglos bidimensionales que contienen, cada uno, la información de un canal distinto. La imagen resultante es la combinación del total de canales. La Fig. 5.1 muestra el esquema general de una imagen digital, donde:

¹Picture element.

- w es el ancho de la imagen en píxeles, y denota las columnas de la imagen.
- h denota la altura de la imagen en píxeles, representa el número de filas de la imagen. Se dice que una imagen tiene dimensiones $(w \times h)$.
- (i, j) par coordenado que describe completamente la posición de un pixel en la imagen.
- $i \in [0, h - 1]$ es la coordenada correspondiente a la fila en que se encuentra un pixel en la posición (i, j) .
- $j \in [0, w - 1]$ representa la columna en que se encuentra un pixel con coordenadas (i, j) .

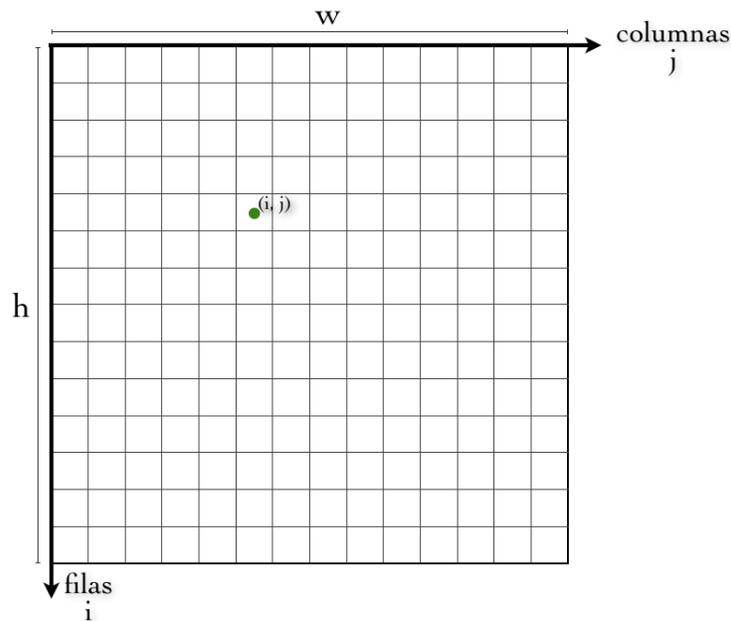


Fig. 5.1: Esquema general de una imagen digital

Para hacer referencia a un determinado pixel en la imagen es suficiente con especificar sus coordenadas (i, j) . De esta manera, el pixel que se encuentra en la intersección de la fila i con la columna j de la imagen recibe el nombre de pixel (i, j) .

La información que cada pixel almacena corresponde a la intensidad luminosa captada por el sensor en su posición, dicha magnitud se denomina brillo del pixel, y se representa como $f(i, j) = f_{ij}$. Donde f_{ij} es una función discreta que toma un par coordenado (i, j) del espacio de la imagen y devuelve un valor real relacionado con el brillo de dicho punto.

Cuando se trabaja con imágenes multicanal, puede resultar conveniente utilizar una notación que especifique no solo el brillo del pixel con respecto a su ubicación dentro de la imagen, sino también el canal al que dicho pixel pertenece. Así, la notación utilizada para indicar el brillo f del pixel (i, j) en el canal k es ${}_k f_{ij}$.

Los valores que un pixel puede adoptar dependen del número de bits utilizados para su representación. La escala más común es la de ocho bits, es decir, se es capaz de representar 256 valores enteros, que van de 0 al 255, donde 0 corresponde al tono negro, o la ausencia de color, y 255 representa el blanco. El número de valores que un pixel puede adoptar se denomina rango de tonos de la imagen (L), y los posibles valores de f son $f \in [0, L - 1]$.

5.2. Subimágen centrada en un pixel

En ocasiones es conveniente trabajar con subimágenes, o ventanas de la imagen original, que contienen solamente la información de una región de interés de la imagen. La Fig. 5.2 muestra el esquema de una subimágen centrada en un pixel, donde:

- m es el ancho de la subimágen.
- n es el alto de la subimágen.
- (i, j) es la coordenada con respecto a la imagen original, del pixel central de la ventana.
- a y b son los radios de vecindad de la imagen para las direcciones i y j respectivamente.

La vecindad de un pixel (i, j) está definida como los $(m \times n - 1)$ pixeles que rodean o envuelven al pixel (i, j) . Una función de la ventana es obtener la información que contienen los vecinos de un pixel, y que es de interés para caracterizarle.

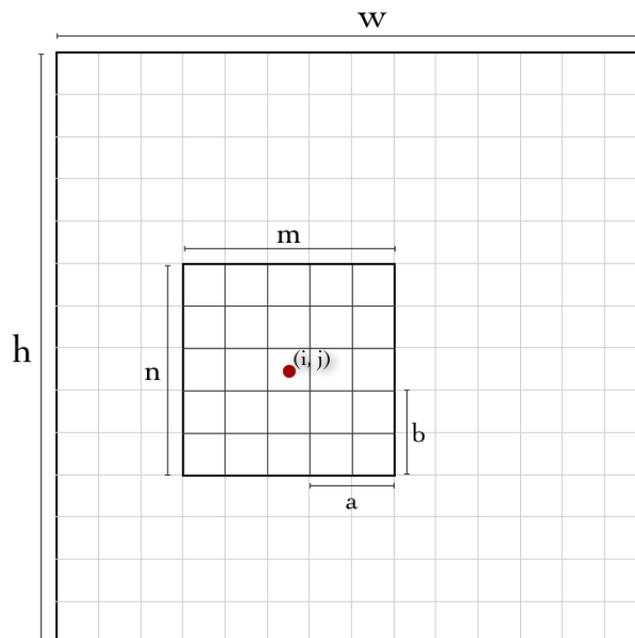


Fig. 5.2: Subimágen centrada en un pixel

Las dimensiones de la subimágen se definen mediante los radios de vecindad como:

$$m = 2a + 1, \tag{5.1}$$

$$n = 2b + 1 \tag{5.2}$$

$S_{m \times n}(i, j)$ denota a la subimágen de dimensiones $(m \times n)$ centrada en el pixel (i, j) de la imagen original.

5.2.1. Operador de ciclo (\odot)

Para escribir de forma clara y compacta las expresiones matemáticas utilizadas a lo largo del texto que describen procesos cíclicos, pero que no involucran operaciones de suma o multiplicación en cada una de sus repeticiones, como es el caso del operador sumatoria (\sum_i) o el operador piatoria (\prod_i), se define el operador ciclo:

$$\bigcirc_{r=k}^n (\cdot), \tag{5.3}$$

que representa un ciclo finito de $(n - k) + 1$ repeticiones, en las que ejecuta el conjunto de instrucciones denotado por (\cdot) . El operador definido en la Ec. 5.3 representa el equivalente simbólico a la sentencia:

```

for ( r = k; r <= n; r++)
{
    .
    .
}
    
```

utilizada en multitud de lenguajes de programación.

5.3. Histogramas

El histograma es una herramienta estadística que se enfoca en contabilizar el número de elementos que pertenecen a un determinado grupo, dado un conjunto de datos que representan cierto fenómeno. Es un descriptor del estado del conjunto al que se aplica, y un paso fundamental en muchas de las tareas de clasificación y reconocimiento en el campo de la visión computacional.

Existen tantos tipos de histogramas como variables medibles en un fenómeno. En particular, en visión computacional uno de los más utilizados es el histograma de brillo de pixeles. A continuación se presentan las ideas, técnicas, y procedimientos utilizados para el cálculo de los distintos histogramas utilizados en el presente trabajo.

5.3.1. Histograma de brillo

El histograma de brillo de una imagen es una función discreta $h(r_k) = n_k$ que representa el número de pixeles n_k en una imagen, en función de los niveles de intensidad de cada pixel, donde r_k es la intensidad de k -ésimo tono de gris [34, 71].

Este histograma expresa el número de pixeles que tienen un determinado valor de intensidad, dentro del rango L , o bien, en un cierto intervalo de valores de L . El histograma ofrece información estadística para la clasificación de las imágenes.

El histograma puede ser calculado contando pixel a pixel y clasificando el valor que cada uno de ellos toma. Para el cálculo del histograma de una imagen completa se tiene que este es igual a:

$$\sum_{r=0}^{h-1} \sum_{c=0}^{w-1} Hist[b(f_{rc})] + +, \quad (5.4)$$

donde $Hist[k]$ es un arreglo que almacena el histograma de la imagen, f_{rc} el brillo del pixel localizado en la fila r y columna c de la matriz que representa a la imagen, y $b(\cdot)$ es una función que toma un elemento del rango L de la imagen (enteros positivos), y devuelve el identificador del rango de L a que pertenece su argumento:

$$b : \mathbb{Z} \rightarrow \mathbb{Z}, \quad (5.5)$$

cuya regla de correspondencia queda definida como:

$$b(x) = x/binW, \quad (5.6)$$

donde $binW$ es el ancho del intervalo de valores de L . Como se puede observar, el método consiste en recorrer punto a punto la imagen aumentando el numero del contador correspondiente a cada intervalo cada vez que se detecta la ocurrencia de este, tomando como dato de entrada el nivel de intensidad del pixel bajo análisis.

Sin embargo, al trabajar con ventanas de la imagen original, o subimágenes, existen algunas consideraciones especiales que deben ser tomadas en cuenta. Principalmente, se debe definir el protocolo de conteo cuando el origen de la ventana se aproxima a los extremos de la imagen y algunas regiones de la ventana quedan fuera del dominio espacial de la imagen original, tal como se muestra en la Fig. 5.3.

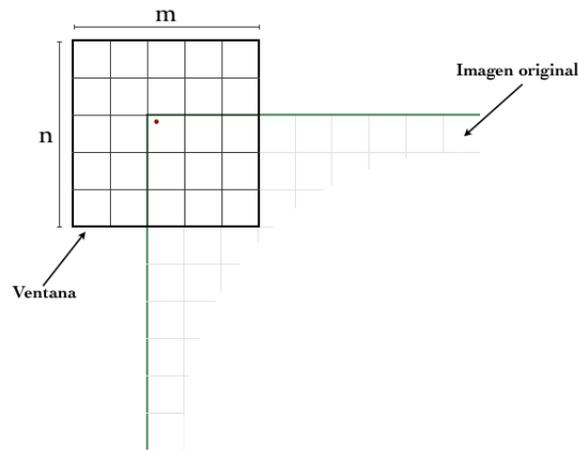


Fig. 5.3: Subimágen con regiones fuera de la imagen original

Los histogramas de ventana son utilizados, como se describe en la Sec. 6.3, sobre algoritmos de correspondencia entre píxeles de imágenes estereoscópicas para obtener un descriptor del píxel central de la ventana, con el objeto de conseguir una característica comparable con otros píxeles, y así, encontrar su par entre las imágenes estereoscópicas. Por tanto, es importante generar histogramas que brinden una buena descripción de los píxeles. Existen diferentes posturas y opiniones sobre cómo se debe generar el histograma en esta situación. Para los fines de este trabajo se ha decidido que una forma de aproximar la realidad es duplicar los píxeles que se encuentran en la frontera de la imagen original para llenar los huecos de la ventana, pues la probabilidad de que exista un cambio abrupto en el tono de gris de la imagen es menor que la de encontrar en el píxel vecino un tono cercano al del píxel conocido.

La Fig. 5.4 muestra tres ejemplos de ventanas ($m \times n$) centradas en el extremo superior izquierdo ($i = 0, j = 0$) de una imagen ($w \times h$). Se observa que los píxeles de la fila y columna de la frontera han sido duplicados para llenar los espacios vacíos de la ventana con el objeto de mejorar la calidad del histograma que se obtendrá, el cual estará relacionado con la descripción estadística del píxel central de la imagen.

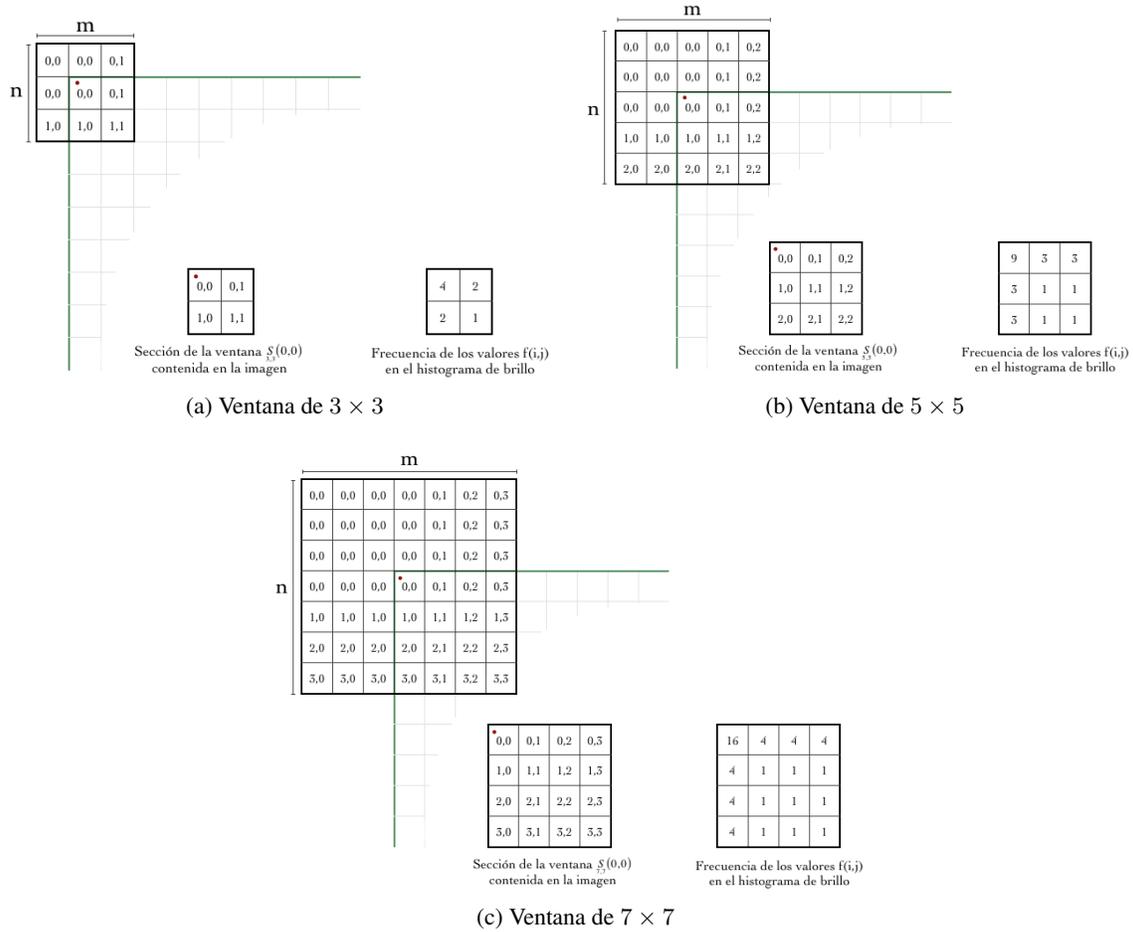


Fig. 5.4: Distribución de píxeles en ventanas ($m \times n$) con regiones fuera de la imagen original

En cada una de las imágenes de la Fig. 5.4 se puede apreciar la sección de la ventana $S_{m \times n}(0,0)$ que se encuentra efectivamente dentro de la imagen original. A un costado de ésta se muestra un arreglo con la frecuencia de ocurrencia del brillo de cada uno de los píxeles de la ventana, considerando que se han completado los píxeles externos. Este número de ocurrencia será utilizado para calcular el histograma de brillo sin necesidad de modificar la imagen original.

El objetivo es obtener una función que, tomando como entrada los parámetros de dimensión de la ventana (a, b), la coordenada del píxel central de la imagen (i, j), y las dimensiones de la imagen original (w, h), arroje el número de veces que el brillo de un determinado píxel debe ser tomado en cuenta, para el histograma de la ventana que se extiende más allá de los límites espaciales de la imagen.

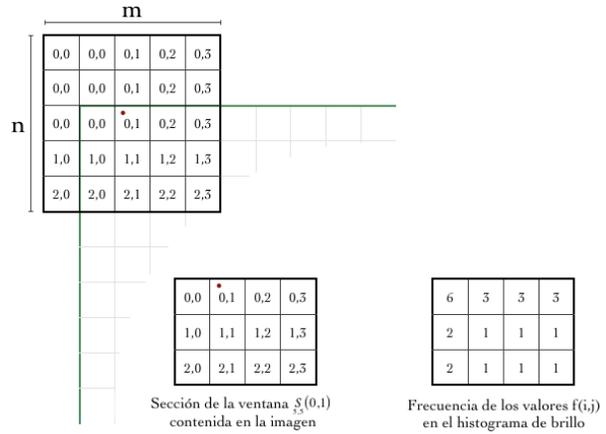
Por ejemplo, al analizar los datos arrojados por los experimentos ilustrados en la Fig. 5.4, caso de ventana simétrica con centro en la esquina superior izquierda (Tabla 5.1), es posible afirmar que la Ec. 5.7 describe la frecuencia de ocurrencia de cada uno de los píxeles efectivamente contenidos en la ventana.

$$\sum_{r=0}^b \sum_{c=0}^a Hist_{ij}[b(f_{rc})] + = \begin{cases} (a+1)^2 & r=0 \& c=0 \\ a+1 & r=0 \mid c=0 \\ 1 & r \neq 0 \& c \neq 0 \end{cases} . \quad (5.7)$$

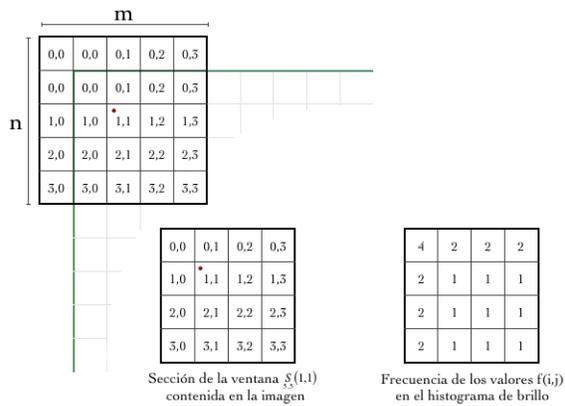
Tabla 5.1: Datos obtenidos de los experimentos presentados en la Fig. 5.4

Figura	a	b	i	j	m	n
5.4a	1	1	0	0	3	3
5.4b	2	2	0	0	5	5
5.4c	3	3	0	0	7	7

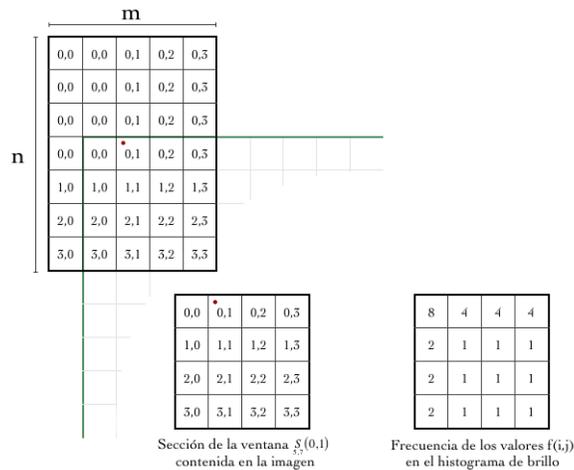
Sin embargo, se encontrarán también casos en los que la ventana que define a la subimágen excede los límites espaciales de la imagen, y no tiene su centro en la esquina superior izquierda de esta. Por lo que se vuelve necesario desarrollar expresiones genéricas, que contemplen la posición (i, j) del centro de la ventana, las dimensiones de esta, y la existencia de una posibles asimetría, es decir, $a \neq b$. La Fig. 5.5 presenta casos en los que los límites de la subimágen están fuera de la imagen original, y su centro en una coordenada $(i, j) \neq (0, 0)$.



(a) Ventana 5×5 , centro en $(0, 1)$



(b) Ventana 5×5 , centro en $(1, 1)$



(c) Ventana 5×7 , centro en $(0, 1)$

Fig. 5.5: Distribución de píxeles en ventanas $(m \times n)$ y centro $\neq (0, 0)$

Para los casos observados, se puede deducir una expresión que indique el número de veces que el valor de cada uno de los píxeles de la subimagen, que están dentro de la imagen original, debe ser conta-

bilizado para obtener el histograma de brillo con el criterio expuesto anteriormente. Dicha expresión es:

$$\sum_{r=0}^{i+b} \sum_{c=0}^{j+a} Hist_{ij}[b(f_{rc})] + = \begin{cases} (b+1-i)(a+1-j) & r=0 \& c=0 \\ (b+1-i) & r=0 \\ (a+1-j) & c=0 \\ 1 & r \neq 0 \& c \neq 0 \end{cases} \quad (5.8)$$

Si se observan los límites superiores de los ciclos de las Ec. 5.4 y 5.7, es evidente que estos varían. La variación es producto de la localización de la ventana que define a la subimágen, pues mientras que en la Ec. 5.4 se trata de la imagen completa, la Ec. 5.7 denota el histograma de una subimágen cuyos límites están fuera de la imagen original, y la exceden por la esquina superior izquierda. El experimento se puede repetir para cada una de las posiciones que puede adoptar una subimágen centrada en un pixel (i, j) , y que excede los límites de la imagen original. Queda claro que, dependiendo de la localización de la subimágen sobre la imagen original, el número de los pixeles que deben ser visitados para calcular el histograma de la subimágen varía. Por tal motivo, se han identificado ocho zonas de la imagen en las que se presenta el fenómeno hasta ahora descrito (Fig. 5.6). Cada una de estas áreas define un zona de exploración de pixeles para el cálculo del histograma.

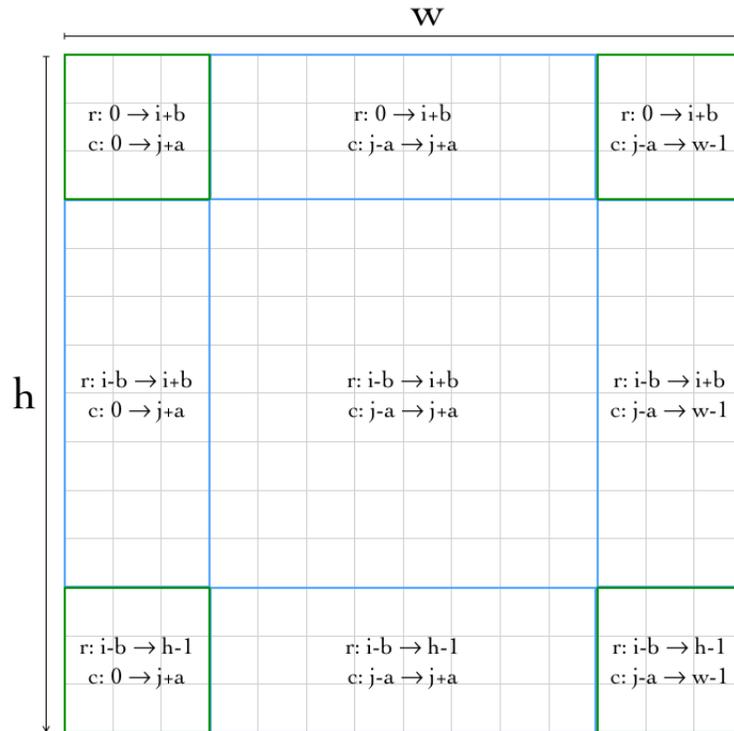


Fig. 5.6: Límites de las áreas de restricción identificadas

Cuando la subimágen sobrepasa los límites espaciales de la imagen original, en cualquiera de los casos posibles, el fenómeno queda definido por la combinación de cuatro situaciones fundamentales:

1. La subimágen sobresale del borde izquierdo de la imagen.
2. La subimágen sobrepasa el límite superior de la imagen.
3. La subimágen sobrepasa el límite derecho de la imagen original.
4. La imagen supera el límite inferior de la imagen.

La combinación de estas situaciones define el área en que se localiza una determinada subimágen de dimensiones (m, n) y centro en (i, j) . Para facilitar el manejo de las distintas áreas de restricción identificadas se les ha asignado una etiqueta de identificación a cada una, las cuales se presentan en la Fig. 5.7, donde:

ESI: esquina superior izquierda

BI: banda izquierda

ESD: esquina superior derecha

BD: banda derecha

EII: esquina inferior izquierda

BS: banda superior

EID: esquina inferior derecha

BInf: banda inferior

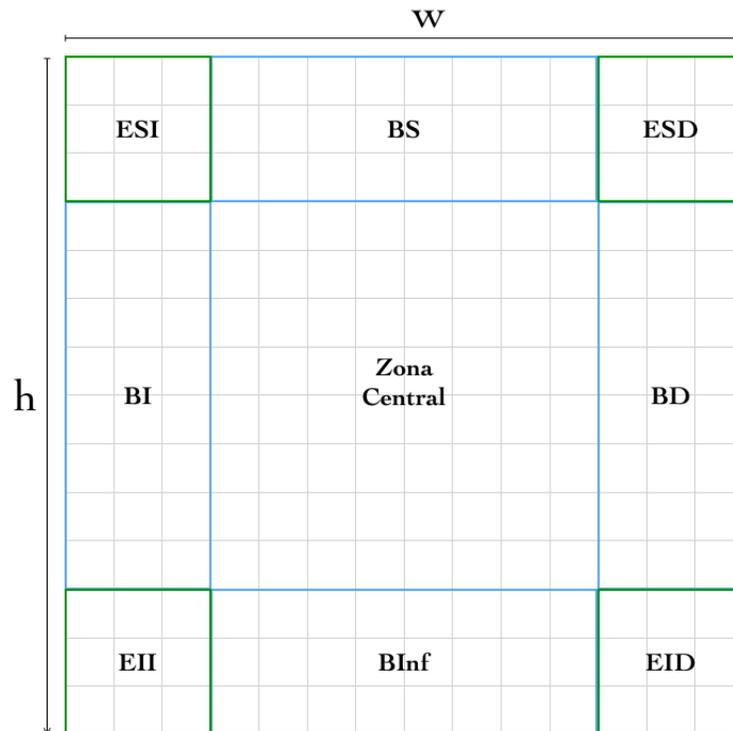


Fig. 5.7: Etiquetas de identificación para las áreas de restricción

La Fig. 5.8 describe el criterio para localizar una subimágen en la zona correspondiente sobre la imagen original, basándose en la coordenada (i, j) del pixel central de la subimágen, y en sus dimensiones.

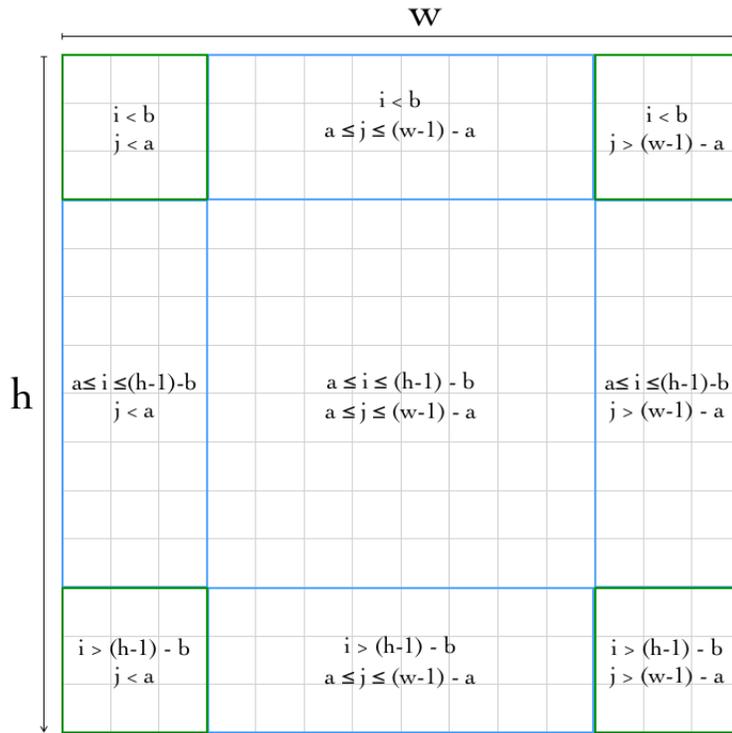


Fig. 5.8: Criterios para localizar una subimágen en áreas de restricción

Una vez que se ha localizado el área de la imagen original en que se encuentra la subimágen, es posible aplicar para el cálculo de su histograma de brillo una expresión del tipo de la Ec. 5.8, específica para cada caso.

5.3.2. Cálculo del histograma de brillo en subimágenes

Conociendo la coordenada del centro de la subimágen (i, j) , los parámetros dimensionales (a, b) de la misma y el área de la imagen en que se encuentra la subimágen, el histograma de brillo se calcula con las siguientes expresiones:

ESI: subimágen localizada en la esquina superior izquierda

$$\sum_{r=0}^{i+b} \sum_{c=0}^{j+a} Hist_{ij} [b(f_{rc})] + = \begin{cases} (b+1-i)(a+1-j) & r=0 \& c=0 \\ (b+1-i) & r=0 \\ (a+1-j) & c=0 \\ 1 & r \neq 0 \& c \neq 0 \end{cases}, \quad (5.9)$$

ESD: subimágen en esquina superior derecha

$$\bigcirc_{r=0c=j-a}^{i+b} \bigcirc_{w-1} Hist_{ij} [b(f_{rc})] + = \begin{cases} (b+1-i)(a+1+j-(w-1)) & r=0 \& c=w-1 \\ (b+1-i) & r=0 \\ (a+1+j-(w-1)) & c=w-1 \\ 1 & r \neq 0 \& c \neq w-1 \end{cases}, \quad (5.10)$$

EII: subimágen en esquina inferior izquierda

$$\bigcirc_{r=i-bc=0}^{h-1} \bigcirc_{j+a} Hist_{ij} [b(f_{rc})] + = \begin{cases} (b+1+i-(h-1))(a+1-j) & r=h-1 \& c=0 \\ (b+1+i-(h-1)) & r=h-1 \\ (a+1-j) & c=0 \\ 1 & r \neq h-1 \& c \neq 0 \end{cases}, \quad (5.11)$$

EID: subimágen localizada en la esquina inferior derecha

$$\bigcirc_{r=i-bc=j-a}^{h-1} \bigcirc_{w-1} Hist_{ij} [b(f_{rc})] + = \begin{cases} (b+1+i-(h-1))(a+1+j-(w-1)) & r=h-1 \& c=w-1 \\ (b+1+i-(h-1)) & r=h-1 \\ (a+1+j-(w-1)) & c=w-1 \\ 1 & r \neq h-1 \& c \neq w-1 \end{cases}, \quad (5.12)$$

BI: subimágen en la banda izquierda

$$\bigcirc_{r=i-bc=0}^{i+b} \bigcirc_{j+a} Hist_{ij} [b(f_{rc})] + = \begin{cases} (a+1-j) & c=0 \\ 1 & c \neq 0 \end{cases}, \quad (5.13)$$

BD: subimágen en la banda derecha

$$\bigcirc_{r=i-bc=j-a}^{i+b} \bigcirc_{w-1} Hist_{ij} [b(f_{rc})] + = \begin{cases} (a+1+j-(w-1)) & c=w-1 \\ 1 & c \neq w-1 \end{cases}, \quad (5.14)$$

BS: subimágen en la banda superior

$$\bigcirc_{r=0c=j-a}^{i+b} \bigcirc_{j+a} Hist_{ij} [b(f_{rc})] + = \begin{cases} (b+1-i) & r=0 \\ 1 & r \neq 0 \end{cases}, \quad (5.15)$$

BInf: subimágen en la banda inferior

$$\bigcirc_{r=i-bc=j-a}^{h-1} \bigcirc_{j+a} Hist_{ij} [b(f_{rc})] + = \begin{cases} (b+1+i-(h-1)) & r=h-1 \\ 1 & r \neq h-1 \end{cases}, \quad (5.16)$$

Zona Central: subimágen en la zona central

$$\bigcirc_{r=i-bc=j-a}^{i+b} \bigcirc_{j+a} Hist_{ij} [b(f_{rc})] + +. \quad (5.17)$$

5.3.3. Histograma de contraste

De manera general, el contraste se refiere a la diferencia de iluminación entre dos pixeles [10]. La definición puede ser ampliada para referirse a la diferencia relativa en intensidad entre un pixel y sus pixeles vecinos. Naturalmente, entre cada par de pixeles formado por el pixel central y uno de sus vecinos existirá una medida de contraste.

El histograma de contraste de una imagen se calcula contabilizando la ocurrencia del valor absoluto de la diferencia de brillo entre un pixel y su vecino derecho, es decir, así:

$$Contraste = |f_{rc} - f_{rc+}|, \quad (5.18)$$

donde f_{rc} es el brillo del pixel ubicado en $(i, j) = (r, c)$, y f_{rc+} es el brillo del pixel ubicado en la intersección de la fila r y la columna $c + 1$ de la imagen, es decir, $(i, j) = (r, c + 1)$. La expresión utilizada para el cálculo del histograma de contraste de la imagen completa es:

$$\sum_{r=0}^{h-1} \sum_{c=0}^{w-2} Hist [b (|f_{rc} - f_{rc+}|)] + +, \quad (5.19)$$

5.3.4. Cálculo del histograma de contraste en subimágenes

Bajo el mismo principio en que fueron determinadas las expresiones para el cálculo del histograma de brillo en subimágenes, y tomando la Ec. 5.18, las expresiones para el cálculo del histograma de contraste en subimágenes son:

ESI: subimagen localizada en la esquina superior izquierda

$$\sum_{r=0}^{i+b} \sum_{c=0}^{j+a-1} Hist_{ij} [b (|f_{rc} - f_{rc+}|)] + = \begin{cases} (b + 1 - i) & r = 0 \\ 1 & r \neq 0 \end{cases} \quad (5.20)$$

$$+ Hist_{ij} [0] + = (2b + 1) (a - j),$$

ESD: subimagen en esquina superior derecha

$$\sum_{r=0}^{i+b} \sum_{c=j-a}^{w-2} Hist_{ij} [b (|f_{rc} - f_{rc+}|)] + = \begin{cases} (b + 1 - i) & r = 0 \\ 1 & r \neq 0 \end{cases} \quad (5.21)$$

$$+ Hist_{ij} [0] + = (2b + 1) (a + j - (w - 1)),$$

EII: subimagen en esquina inferior izquierda

$$\begin{aligned} \sum_{r=i-b}^{h-1} \sum_{c=0}^{j+a-1} Hist_{ij} [b(|f_{rc} - f_{rc+}|)] + = & \begin{cases} (b+1+i-(h-1)) & r = h-1 \\ 1 & r \neq h-1 \end{cases} \quad (5.22) \\ + Hist_{ij} [0] + = & (2b+1)(a-j), \end{aligned}$$

EID: subimágen localizada en la esquina inferior derecha

$$\begin{aligned} \sum_{r=i-bc=j-a}^{h-1} \sum_{c=0}^{w-2} Hist_{ij} [b(|f_{rc} - f_{rc+}|)] + = & \begin{cases} (b+1+i-(h-1)) & r = h-1 \\ 1 & r \neq h-1 \end{cases} \quad (5.23) \\ + Hist_{ij} [0] + = & (2b+1)(a+j-(w-1)), \end{aligned}$$

BI: subimágen en la banda izquierda

$$\begin{aligned} \sum_{r=i-b}^{i+b} \sum_{c=0}^{j+a-1} Hist_{ij} [b(|f_{rc} - f_{rc+}|)] + + & \quad (5.24) \\ + Hist_{ij} [0] + = & (2b+1)(a-j), \end{aligned}$$

BD: subimágen en la banda derecha

$$\begin{aligned} \sum_{r=i-bc=j-a}^{i+b} \sum_{c=0}^{w-2} Hist_{ij} [b(|f_{rc} - f_{rc+}|)] + + & \quad (5.25) \\ + Hist_{ij} [0] + = & (2b+1)(a+j-(w-1)), \end{aligned}$$

BS: subimágen en la banda superior

$$\sum_{r=0}^{i+b} \sum_{c=j-a}^{j+a-1} Hist_{ij} [b(|f_{rc} - f_{rc+}|)] + = \begin{cases} (b+1-i) & r = 0 \\ 1 & r \neq 0 \end{cases}, \quad (5.26)$$

BInf: subimágen en la banda inferior

$$\sum_{r=i-bc=j-a}^{h-1} \sum_{c=0}^{j+a-1} Hist_{ij} [b(|f_{rc} - f_{rc+}|)] + = \begin{cases} (b+1+i-(h-1)) & r = h-1 \\ 1 & r \neq h-1 \end{cases}, \quad (5.27)$$

Zona Central: subimágen en la zona central

$$\sum_{r=i-bc=j-a}^{i+b} \sum_{c=0}^{j+a-1} Hist_{ij} [b(|f_{rc} - f_{rc+}|)] + +. \quad (5.28)$$

5.3.5. Histograma de microcontornos

La textura es una característica intuitiva relacionada con la sensación visual que provoca un objeto en la escena, está presente en casi cualquier parte de la naturaleza y es una poderosa herramienta en procesos de discriminación y reconocimiento. La textura puede ser descrita en términos espaciales, de frecuencia y periodicidad, de dirección, o de grado de complejidad. En particular, el cálculo de la orientación de microcontornos nos acerca de cierto modo a conocer la orientación de la textura partiendo de micropatrones. El histograma de microcontornos de una región de imagen es una herramienta de paridad que brinda un análisis más profundo de la imagen.

El cálculo de orientación de microcontornos se lleva a cabo en grupos de cuatro píxeles, tal como se observa en la Fig. 5.9. El grupo se identifica mediante la posición (i, j) del píxel de la esquina superior izquierda, y a esta posición se asocia la orientación del microcontorno encontrado.

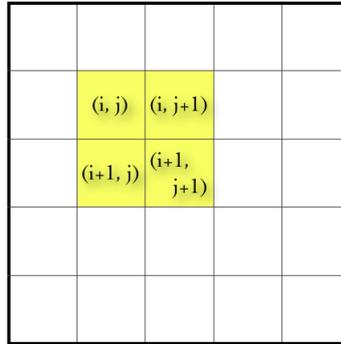


Fig. 5.9: Grupo de píxeles para cálculo de orientación de microcontornos

Para llevar a cabo el cálculo de la orientación del microcontorno de un determinado grupo, es necesario determinar en primer lugar si existe dicho microcontorno en el área de análisis. Así, sea f_{ij} el brillo del píxel ubicado en (i, j) dentro de la imagen, y $C_1 \in \mathbb{N}$ el umbral que define la existencia del microcontorno, se definen las sumas cruzadas entre los píxeles del grupo como:

$$y_1 = f_{ij} + f_{i+1, j+1}, \quad (5.29)$$

$$y_2 = f_{i, j+1} + f_{i+1, j}, \quad (5.30)$$

de esta forma, si $|y_1 - y_2| > C_1$ no existe contorno en el área, así mismo, si $|y_1 - y_2| \leq C_1$ el contorno existe y su orientación debe ser calculada.

La orientación de un microcontorno en radianes se define de manera general como:

$$\theta_{ij} = V_r + Sg \cdot y, \quad (5.31)$$

donde θ_{ij} es la orientación del microcontorno correspondiente al conjunto de píxeles (i, j) , V_r es el valor de referencia a partir del cual se calcula la orientación, $Sg \in \{-1, +1\}$ determina la relación de

suma entre V_r e y , mientras que y se define como:

$$y = \frac{\Delta_2}{\Delta_1} = \frac{\min(|d_1|, |d_2|)}{\max(|d_1|, |d_2|)}. \quad (5.32)$$

Para determinar el valor de las variables V_r y Sg es necesario definir:

$$d_1 = f_{ij} - f_{i+1,j+1}, \quad (5.33)$$

$$d_2 = f_{i,j+1} - f_{i+1,j}, \quad (5.34)$$

$$diff = |d_1| - |d_2|. \quad (5.35)$$

Con el signo resultante en cada una de las variables definidas en las Ec. 5.33, 5.34 y 5.35 se ingresa a la Tabla 5.2, de donde se obtienen los valores correspondientes para el cálculo de la orientación deseada, utilizando la Ec. 5.31.

Tabla 5.2: Valores de V_r y Sg para el cálculo de orientación de microcontornos

V_r	$\frac{\pi}{4}$		$\frac{3\pi}{4}$		$\frac{5\pi}{4}$		$\frac{7\pi}{4}$	
Sg	-1	+1	-1	+1	-1	+1	-1	+1
d_1	+	+	+	-	-	-	-	+
d_2	-	+	+	+	+	-	-	-
$diff$	+	+	-	-	+	+	-	-

De forma general, el histograma de microcontornos de una imagen es:

$$\sum_{r=0}^{h-2w-2} \sum_{c=0}^{w-1} Hist[b(\theta_{rc})] + +. \quad (5.36)$$

5.3.6. Cálculo del histograma de microcontornos en subimágenes

Tomando la filosofía utilizada para el cálculo del histograma de brillo en subimágenes, se llevó a cabo la determinación de las expresiones para los histogramas de microcontornos en subimágenes. Sin embargo, dadas las condiciones para el cálculo de los microcontornos, es necesario definir dos nuevas variables que simplifican el cálculo de los microcontornos en conjuntos de pixeles que exceden los límites de la imagen, de esta forma:

$$\theta_{rc}^r = \frac{\pi}{4} \cdot (3 + 2 \cdot Sign[f_{rc} - f_{r,c+1}]), \quad (5.37)$$

$$\theta_{rc}^c = \frac{\pi}{4} \cdot (3 + 2 \cdot Sign[f_{rc} - f_{r+1,c}]), \quad (5.38)$$

donde la función signo ($Sign[\cdot]$) se define como:

$$Sign[\alpha] = \begin{cases} 1 & \alpha \geq 0 \\ -1 & \alpha < 0 \end{cases} . \quad (5.39)$$

Así, con las Ec. 5.31, 5.37, y 5.38, las expresiones son:

ESI: subimágen localizada en la esquina superior izquierda

$$\begin{aligned} & \sum_{r=0}^{i+b-1} \sum_{c=0}^{j+a-1} Hist_{ij}[b(\theta_{rc})] \quad ++ \\ & + \sum_{c=0}^{j+a-1} Hist_{ij}[b(\theta_{0c}^r)] \quad ++ = (b-i) \\ & + \sum_{r=0}^{i+b-1} Hist_{ij}[b(\theta_{r0}^c)] \quad ++ = (a-j), \end{aligned} \quad (5.40)$$

ESD: subimágen en esquina superior derecha

$$\begin{aligned} & \sum_{r=0}^{i+b-1} \sum_{c=j-a}^{w-2} Hist_{ij}[b(\theta_{rc})] \quad ++ \\ & + \sum_{c=j-a}^{w-2} Hist_{ij}[b(\theta_{0c}^r)] \quad ++ = (b-i) \\ & + \sum_{r=0}^{i+b-1} Hist_{ij}[b(\theta_{r,w-1}^c)] \quad ++ = (a+j-(w-1)), \end{aligned} \quad (5.41)$$

EII: subimágen en esquina inferior izquierda

$$\begin{aligned} & \sum_{r=i-b}^{h-2} \sum_{c=0}^{j+a-1} Hist_{ij}[b(\theta_{rc})] \quad ++ \\ & + \sum_{c=0}^{j+a-1} Hist_{ij}[b(\theta_{h-1,c}^r)] \quad ++ = (b+i-(h-1)) \\ & + \sum_{r=i-b}^{h-2} Hist_{ij}[b(\theta_{r0}^c)] \quad ++ = (a-j), \end{aligned} \quad (5.42)$$

EID: subimágen localizada en la esquina inferior derecha

$$\begin{aligned}
 & \begin{array}{c} \overset{h-2}{\circlearrowleft} \quad \overset{w-2}{\circlearrowleft} \\ r=i-bc=j-a \end{array} Hist_{ij} [b(\theta_{rc})] \quad ++ \\
 & + \begin{array}{c} \overset{w-2}{\circlearrowleft} \\ c=j-a \end{array} Hist_{ij} [b(\theta_{h-1,c}^r)] \quad += (b+i-(h-1)) \\
 & + \begin{array}{c} \overset{h-2}{\circlearrowleft} \\ r=i-b \end{array} Hist_{ij} [b(\theta_{r,w-1}^c)] \quad += (a+j-(w-1)),
 \end{aligned} \tag{5.43}$$

BI: subimágen en la banda izquierda

$$\begin{aligned}
 & \begin{array}{c} \overset{i+b-1j+a-1}{\circlearrowleft} \quad \overset{w-2}{\circlearrowleft} \\ r=i-b \quad c=0 \end{array} Hist_{ij} [b(\theta_{rc})] \quad ++ \\
 & + \begin{array}{c} \overset{i+b-1}{\circlearrowleft} \\ r=i-b \end{array} Hist_{ij} [b(\theta_{r0}^c)] \quad += (a-j),
 \end{aligned} \tag{5.44}$$

BD: subimágen en la banda derecha

$$\begin{aligned}
 & \begin{array}{c} \overset{i+b-1}{\circlearrowleft} \quad \overset{w-2}{\circlearrowleft} \\ r=i-bc=j-a \end{array} Hist_{ij} [b(\theta_{rc})] \quad ++ \\
 & + \begin{array}{c} \overset{i+b-1}{\circlearrowleft} \\ r=i-b \end{array} Hist_{ij} [b(\theta_{r,w-1}^c)] \quad += (a+j-(w-1)),
 \end{aligned} \tag{5.45}$$

BS: subimágen en la banda superior

$$\begin{aligned}
 & \begin{array}{c} \overset{i+b-1j+a-1}{\circlearrowleft} \quad \overset{w-2}{\circlearrowleft} \\ r=0 \quad c=j-a \end{array} Hist_{ij} [b(\theta_{rc})] \quad ++ \\
 & + \begin{array}{c} \overset{j+a-1}{\circlearrowleft} \\ c=j-a \end{array} Hist_{ij} [b(\theta_{0c}^r)] \quad += (b-i),
 \end{aligned} \tag{5.46}$$

BInf: subimágen en la banda inferior

$$\begin{aligned}
 & \begin{array}{c} \overset{h-2}{\circlearrowleft} \quad \overset{j+a-1}{\circlearrowleft} \\ r=i-bc=j-a \end{array} Hist_{ij} [b(\theta_{rc})] \quad ++ \\
 & + \begin{array}{c} \overset{j+a-1}{\circlearrowleft} \\ c=j-a \end{array} Hist_{ij} [b(\theta_{h-1,c}^r)] \quad += (b+i-(h-1)),
 \end{aligned} \tag{5.47}$$

Zona Central: subimágen en la zona central

$$\begin{array}{c} \overset{i+b-1j+a-1}{\circlearrowleft} \quad \overset{w-2}{\circlearrowleft} \\ r=i-bc=j-a \end{array} Hist_{ij} [b(\theta_{rc})] \quad ++. \tag{5.48}$$

5.4. Fundamentos de visión estereoscópica

5.4.1. Modelo de cámara pinhole

Una cámara es una máquina proyectiva que capta escenas del mundo tridimensional y las mapea a un plano. Al modelo de la cámara que permite transformar un punto tridimensional en un punto bidimensional sobre un plano llamado imagen, se le conoce como el modelo de la cámara pinhole (Fig. 5.10). Una cámara pinhole posee un centro de proyección (C), el cual se ubica entre el plano de la imagen y la escena 3D. Este modelo se basa en que todo haz de luz reflejado por los objetos en la escena cruza a través del centro de proyección hasta llegar al plano del sensor (equivalente al plano de la imagen), por lo que existe una correspondencia de cada punto en la imagen 2D con un punto 3D de la escena. El origen del sistema de coordenadas euclidianas de la cámara se fija en el punto C .

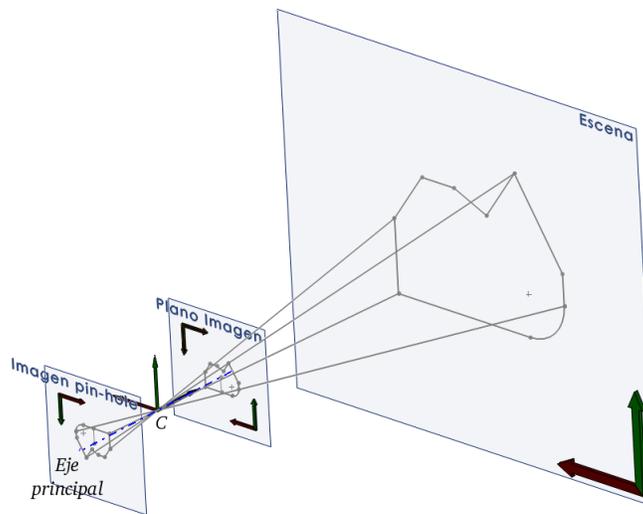


Fig. 5.10: Modelo de cámara pinhole

Al centro de proyección también se le conoce como el centro de la cámara o centro óptico, a la línea que pasa por el centro de la cámara y es perpendicular al plano de la imagen se le denomina eje principal de la cámara, y al punto donde el eje principal se une con el plano de la imagen se le conoce como punto principal (c).

5.4.2. Marcos de referencia en visión computacional

Cuando se desea que un sistema de visión computacional sea capaz de determinar la posición métrica de objetos en el espacio, es necesario establecer una serie de marcos de referencia que permitan expresar las coordenadas de los objetos captados en las diferentes representaciones utilizadas a lo largo del proceso de captura, detección, y medición. La Fig. 5.11 muestra los diferentes marcos de referencia a los que se hace mención en el presente trabajo.

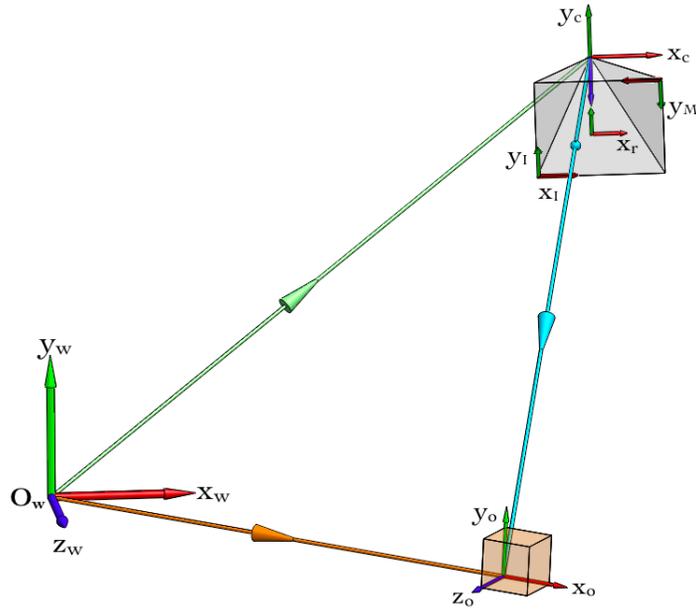


Fig. 5.11: Marcos de referencia en visión computacional

donde:

$\{x_w, y_w, z_w, O_w\}$ marco de referencia inercial o del mundo, con centro en el origen global (O_w).

$\{x_C, y_C, z_C, C\}$ marco de la cámara, con centro en el punto C . Sus vectores se encuentran en unidades métricas al igual que el marco del mundo.

$\{x_r, y_r, c\}$ marco retinal 2D, con centro en el punto c . Es el marco de transición entre unidades métricas y píxeles.

$\{x_I, y_I, o_I\}$ marco de la imagen, con centro en el punto o_I . Preserva la dirección del eje z hacia afuera de la imagen, en dirección a la escena captada, al ser coincidente con la orientación del marco de la cámara.

$\{x_M, y_M, o_M\}$ marco de referencia de la representación matricial de la imagen, se ubica en la esquina superior izquierda de la misma. Su orientación se corresponde con la forma en que se accede a la información que representa la escena captada.

$\{x_o, y_o, z_o, o_o\}$ marco de referencia del objeto. Corresponde al marco de referencia que se coloca en cada uno de los objetos captados, y por medio del cual es posible determinar su orientación.

El vector V_C^X , con origen en el marco de la cámara, localiza al punto \mathbf{X} . En un punto de su trayectoria intersecciona el plano retinal en un punto \mathbf{x} , que puede ser medido desde el marco retinal o desde el marco de la imagen. Dado que la imagen se representa en la computadora como una matriz de valores enteros, el marco de la representación matricial tiene las direcciones x e y positivas, en sentido contrario

a las direcciones del marco de referencia de la imagen. La transformación entre coordenadas de la representación matricial y del marco de la imagen es:

$$\mathbf{x}_M = R_M^I \mathbf{x}_I, \quad (5.49)$$

$$\begin{pmatrix} x_M \\ y_M \\ 1 \end{pmatrix} = \begin{bmatrix} -1 & 0 & W_{img} \\ 0 & -1 & H_{img} \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x_I \\ y_I \\ 1 \end{pmatrix}, \quad (5.50)$$

donde W_{img} es el ancho de la imagen en pixeles, y H_{img} es la altura.

Como se verá en la Sec. 5.4.4, es posible definir una transformación entre coordenadas de la imagen y coordenadas retinales. Por lo tanto, la tarea consiste en transformar las coordenadas del punto $\mathbf{x} \in \mathbb{R}^2$, visto desde el marco retinal en pixeles, a sus coordenadas equivalentes $\mathbf{X} \in \mathbb{R}^3$ en el marco de la cámara, en unidades métricas. Esta conversión no es trivial, pues involucra un gran número de procesos:

1. **Eliminar las aberraciones producidas por el lente.**
2. **Cálculo de parámetros intrínsecos de la cámara.** Como se ha visto en el modelo pinhole, la proyección del mundo real en el sensor depende de las dimensiones y la disposición de los elementos de la cámara (dimensión del sensor, y la distancia y alineación entre este y el lente). Estas propiedades se denominan parámetros intrínsecos, y son necesarias para reconstruir puntos 3D con base en información 2D.
3. **Triangulación (cálculo de la profundidad).** La triangulación es el proceso mediante el cual se obtienen las coordenadas métricas tridimensionales de un punto del mundo proyectado en el plano de la imagen.

Para conocer dichas componentes es necesario observar el objeto desde al menos dos puntos de vista, o en este caso, cámaras. Naturalmente es necesario identificar el objeto en cada una de las imágenes, es decir, definir la correspondencia entre los pixeles de las distintas imágenes.

La búsqueda de correspondencias es un problema bidimensional, pues dos pixeles correspondientes pueden encontrarse en filas y columnas distintas en cada una de las imágenes. Si el problema puede ser reducido a una sola dimensión, el ahorro de tiempo durante la búsqueda es considerable. Esto se logra restringiendo la región de búsqueda a una línea, por medio de cálculo e implementación de la matriz fundamental.

5.4.3. Aberraciones del lente de la cámara

Para capturar una escena completa del mundo real sobre la reducida área del sensor es necesario concentrar los haces de luz reflejada por los elementos de la escena, esto se logra utilizando un lente. Sin embargo, al pasar por ella, los haces sufren ligeras desviaciones de su dirección original.

Si se desea conocer las coordenadas tridimensionales de cada punto proyectado en el sensor, el primer paso es determinar la dirección real de cada uno de los haces incidentes que han generado dicho punto. Esto se logra eliminando las aberraciones tangenciales y radiales que el lente produce, las cuales son modeladas mediante transformaciones polinomiales. Los coeficientes de dicho modelo se obtienen mediante un proceso de optimización, con base en la información obtenida al capturar imágenes de un patrón conocido.

La eliminación de las aberraciones radiales [5] se lleva a cabo aplicando las transformaciones:

$$x_{radcorr} = x (1 + k_1 r^2 + k_2 r^4 + k_3 r^6), \quad (5.51)$$

$$y_{radcorr} = y (1 + k_1 r^2 + k_2 r^4 + k_3 r^6), \quad (5.52)$$

y las tangenciales son corregidas con:

$$x_{corr} = x_{radcorr} + (2p_1 xy + p_2 (r^2 + 2x^2)), \quad (5.53)$$

$$y_{corr} = y_{radcorr} + (p_1 (r^2 + 2y^2) + 2p_2 xy), \quad (5.54)$$

donde (x_{corr}, y_{corr}) son las coordenadas corregidas del pixel bajo análisis, (x, y) son las coordenadas originales del pixel en el marco retinal, y $r^2 = x^2 + y^2$.

5.4.4. Matriz de parámetros intrínsecos

Considerando el modelo pinhole, y colocando el plano retinal frente al centro de proyección de la cámara para coincidir con la forma en que la cámara entrega la información, tal como se observa en la Fig. 5.12, por triángulos semejantes, el punto $\mathbf{X} = (X_c, Y_c, Z_c) \in \mathbb{R}^3$ se proyecta en el plano retinal como $\mathbf{x}_r = (x_r, y_r) \in \mathbb{R}^2$ bajo las siguientes relaciones:

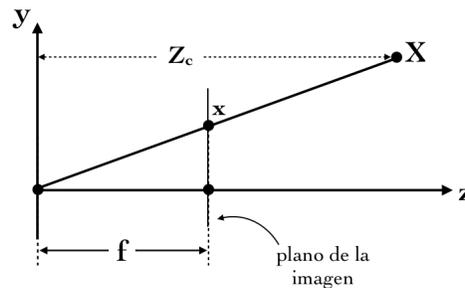


Fig. 5.12: Disposición del plano retinal con respecto al centro óptico de la cámara

$$\frac{y_r}{f} = \frac{Y_c}{Z_c}, \quad (5.55) \qquad \frac{x_r}{f} = \frac{X_c}{Z_c}, \quad (5.56)$$

$$y_r = f \cdot \frac{Y_c}{Z_c}, \quad (5.57) \qquad x_r = f \cdot \frac{X_c}{Z_c}, \quad (5.58)$$

donde f es la distancia focal de la cámara. Si se representa el vector del punto proyectado \mathbf{x}_r en coordenadas homogéneas [38], y se multiplica éste por Z_c , utilizando las propiedades de esta representación, se tiene que:

$$\begin{pmatrix} x_r \\ y_r \\ 1 \end{pmatrix} = \begin{pmatrix} f \cdot \frac{X_c}{Z_c} \\ f \cdot \frac{Y_c}{Z_c} \\ 1 \end{pmatrix} = \begin{pmatrix} f \cdot X_c \\ f \cdot Y_c \\ Z_c \end{pmatrix}, \quad (5.59)$$

cuya representación matricial es:

$$\begin{pmatrix} x_r \\ y_r \\ 1 \end{pmatrix} = \begin{pmatrix} f \cdot X_c \\ f \cdot Y_c \\ Z_c \end{pmatrix} = \mathbf{P}_r \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix}, \quad (5.60)$$

$$\mathbf{P}_r = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad (5.61)$$

donde \mathbf{P}_r se denomina matriz de proyección, y se encarga de mapear los puntos del espacio 3D al plano del sensor. Sin embargo, dado que los parámetros intrínsecos se obtienen por medio de calibración con base en la información del ambiente recogida por la cámara, y ésta se encuentra en coordenadas de la imagen, es conveniente expresar la proyección en este sistema de referencia.

El marco de referencia de la imagen difiere del marco retinal solamente por una traslación, pues su orientación es la misma. Así, la transformación entre sistemas de coordenadas está dada por:

$$x_I = x_r + p_x, \quad (5.62)$$

$$y_I = y_r + p_y, \quad (5.63)$$

donde p_x y p_y son las componentes del vector que separa al marco de referencia retinal del marco de la imagen, medidas desde este último, y (x_I, y_I) son las coordenadas del pixel referido por (x_r, y_r) , medidas desde el marco de la imagen, en unidades métricas.

Así, el mapeo de un punto del espacio 3D (marco de la cámara) a un punto en el plano de la imagen es,

$$y_I = f \cdot \frac{Y_c}{Z_c} + p_y, \quad (5.64)$$

$$x_I = f \cdot \frac{X_c}{Z_c} + p_x. \quad (5.65)$$

Representadas de forma matricial en coordenadas homogneas,

$$\begin{pmatrix} x_I \\ y_I \\ 1 \end{pmatrix} = \begin{pmatrix} Z_c \cdot x_I \\ Z_c \cdot y_I \\ Z_c \end{pmatrix} = \begin{pmatrix} f \cdot X_c + Z_c \cdot p_x \\ f \cdot Y_c + Z_c \cdot p_y \\ Z_c \end{pmatrix} = \mathbf{P_I} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix}, \quad (5.66)$$

$$\mathbf{P_I} = [K|0], \quad (5.67)$$

$$K = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (5.68)$$

donde K se denomina matriz de parámetros intrínsecos o matriz de calibración de cámara [13, 23, 38], y sus parámetros se encuentran en unidades métricas.

Sin embargo, las coordenadas de la imagen están en pixeles, además los sensores reales pueden tener pixeles no cuadrados, por lo tanto se debe introducir un factor que tome en cuenta la conversión de unidades métricas a pixeles y la condición de desigualdad entre las dimensiones del pixel. Así, sean m_x y m_y el número de pixeles por mm en la dirección x e y del sensor, y aplicando a K (Ec. 5.68) los factores de proporcionalidad, se obtiene la matriz de parámetros intrínsecos en pixeles \tilde{K} como:

$$\tilde{K} = \begin{bmatrix} m_x & 0 & 0 \\ 0 & m_y & 0 \\ 0 & 0 & 1 \end{bmatrix} K = \begin{bmatrix} \alpha_x & 0 & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (5.69)$$

donde:

$$\alpha_x = m_x \cdot f, \quad (5.70)$$

$$\alpha_y = m_y \cdot f, \quad (5.71)$$

$$x_0 = m_x \cdot p_x, \quad (5.72)$$

$$y_0 = m_y \cdot p_y. \quad (5.73)$$

5.4.5. Matriz fundamental

Para poder triangular las coordenadas de un punto en el espacio, es necesario observarlo desde al menos dos puntos de vista distintos. Así, un punto $\mathbf{X} = (X_c, Y_c, Z_c)$ en el espacio tridimensional se proyecta en dos distintas cámaras como,

$$\mathbf{x} = \mathbf{P}\mathbf{X}, \quad (5.74)$$

$$\mathbf{x}' = \mathbf{P}'\mathbf{X}, \quad (5.75)$$

donde \mathbf{P} y \mathbf{P}' son las matrices de proyección de las cámaras, y los píxeles \mathbf{x} y \mathbf{x}' representan la proyección del punto \mathbf{X} sobre el sensor de cada una de las cámaras. Además, \mathbf{x} y \mathbf{x}' son correspondientes pues son mapeados desde el mismo punto en el espacio 3D.

Para facilitar la búsqueda de correspondencias es conveniente deducir una relación que, dado el píxel \mathbf{x} de la primera imagen, determine la región en la segunda imagen en donde el píxel correspondiente \mathbf{x}' puede ser encontrado. Dicha relación está dada por la matriz fundamental \mathbf{F} , que encapsula la geometría interna y externa de las cámaras en un arreglo matricial de 3×3 . Si un punto en el espacio 3D se mapea como \mathbf{x} en la primera imagen, y como \mathbf{x}' en la segunda, estos puntos correspondientes satisfacen la restricción [23, 38]

$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0, \quad (5.76)$$

denominada restricción epipolar (Fig. 5.13).

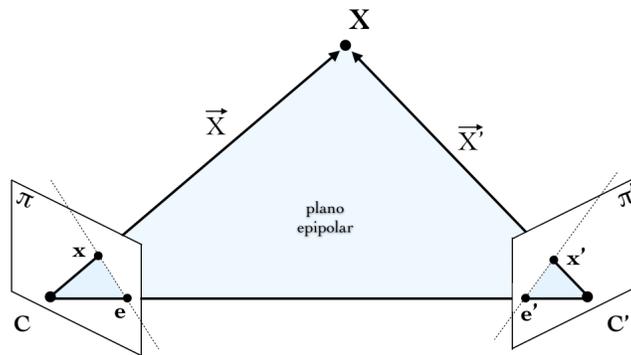


Fig. 5.13: Geometría epipolar del arreglo estereoscópico

Existe una gran variedad de aproximaciones para el cálculo de la matriz fundamental. En particular, si se dispone previamente de la matriz de parámetros intrínsecos y se conoce la disposición geométrica de las cámaras, se puede calcular vía la matriz esencial (\mathbf{E}).

Al arreglo geométrico de un sistema multicámara se le denomina parámetros extrínsecos del sistema. Los cuales toman en cuenta los marcos de referencia de las distintas cámaras, y proporcionan las

herramientas para representar las coordenadas de todas ellas en un marco único, perteneciente a la cámara principal. Estos parámetros se componen de la matriz de rotación (R), y un vector de traslación (T), que orientan y localizan a cada uno de los marcos de cámara distintos al de la cámara principal.

La Fig. 5.14 muestra un sistema de dos cámaras en donde es posible apreciar el vector de traslación,

$$T = \overrightarrow{C'C} = C' - C = \begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix}, \quad (5.77)$$

y la matriz de rotación $R_C^{C'}$, que transforma las coordenadas medidas en el marco C' a coordenadas del marco C .

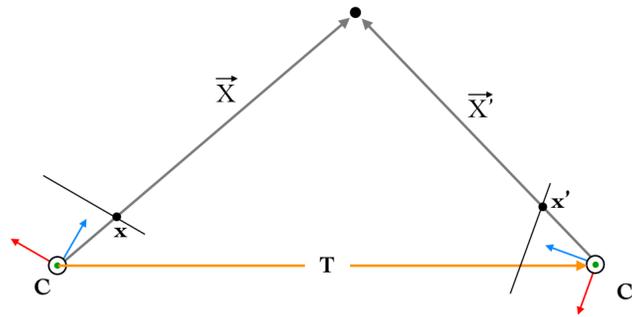


Fig. 5.14: Relación entre marcos de referencia de un sistema estereoscópico

La construcción de la matriz esencial se basa en el plano epipolar, el cual está descrito como:

$$(\vec{X} - T)^T (T \times \vec{X}) = 0, \quad (5.78)$$

$$(R_C^{C'} \vec{X}')^T (T \times \vec{X}) = 0. \quad (5.79)$$

El producto escalar se puede escribir como la multiplicación de una matriz antisimétrica (S) por un vector, y se tiene entonces $T \times \vec{X} = [T] \vec{X} = S \vec{X}$, donde

$$S = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}. \quad (5.80)$$

Sustituyendo en Ec. 5.79,

$$(R_C^{C'} \vec{X}')^T S \vec{X} = \vec{X}'^T R_C^{C'} S \vec{X} = 0, \quad (5.81)$$

$$\vec{X}'^T \mathbf{E} \vec{X} = 0, \quad (5.82)$$

$$\mathbf{E} \equiv R_C^{C'} S. \quad (5.83)$$

La matriz \mathbf{E} relaciona el punto \mathbf{X} observado desde dos puntos de vista, y medido en coordenadas métricas. Para explotar la utilidad de la restricción epipolar (Ec. 5.76) debe ser expresada en coordenadas de imagen (o en coordenadas de matriz según convenga). Introduciendo la relación entre el punto proyectado y el rayo incidente:

$$\vec{X} = \tilde{K}^{-1} \mathbf{x}, \quad (5.84) \quad \vec{X}' = \tilde{K}'^{-1} \mathbf{x}', \quad (5.85)$$

y sustituyendo en la Ec. 5.82,

$$\left(\tilde{K}'^{-1} \mathbf{x}' \right)^T \mathbf{E} \tilde{K}^{-1} \mathbf{x} = 0, \quad (5.86)$$

$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0, \quad (5.87)$$

$$\mathbf{F} \equiv \tilde{K}'^{-T} \mathbf{E} \tilde{K}^{-1} \quad (5.88)$$

Una vez calculada la matriz fundamental (\mathbf{F}), es posible utilizarla en el proceso de rectificación de imágenes [13], que guarda una estrecha relación con el proceso de búsqueda de correspondencias.

5.4.6. Triangulación

La labor del proceso de triangulación es recuperar el valor de las coordenadas de un punto \mathbf{X} en el espacio 3D a partir de dos o más de sus proyecciones ($\mathbf{x}, \mathbf{x}', \dots$), captadas desde distintos puntos de vista.

Si no existiera ruido en las imágenes a las que pertenecen \mathbf{x} y \mathbf{x}' , los rayos que estos forman con su centro de proyección correspondiente se intersectarían en el punto único \mathbf{X} . Sin embargo, debido al ruido presente, esto no ocurre. En situaciones en donde el error de medición y la búsqueda de correspondencias están controlados, siempre es posible encontrar la perpendicular a los dos rayos, que es la distancia mínima que los separa [13, 23, 38]. En estos casos tomar el punto medio de dicha línea ofrece una buena aproximación a \mathbf{X} . La Fig. 5.15 muestra el esquema de triangulación descrito.

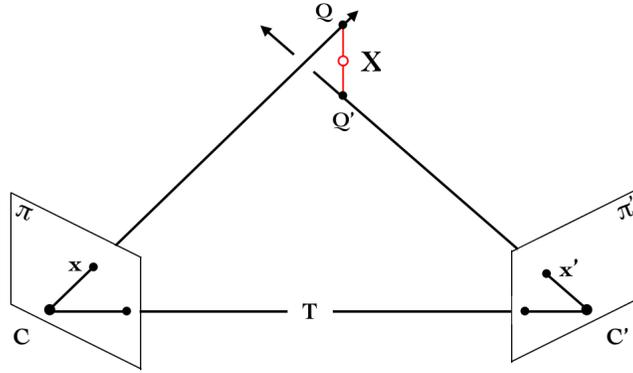


Fig. 5.15: Triangulación de un punto en el espacio 3D

Sea Q y Q' los puntos sobre los rayos $\overrightarrow{C\mathbf{x}}$ y $\overrightarrow{C'\mathbf{x}'}$ respectivamente, que determinan el punto de intersección de la perpendicular común, el punto \mathbf{X} que se busca se encuentra en la mitad de dicho trayecto.

Tomando la ecuación de lazo vectorial

$$\overrightarrow{CQ} - \lambda (\overrightarrow{C\mathbf{x}} \times \overrightarrow{C'\mathbf{x}'}) - \overrightarrow{C'Q'} - T = 0. \quad (5.89)$$

Para que se pueda llevar a cabo la suma de vectores, es necesario que todos ellos se expresen en un mismo sistema de coordenadas, en este caso, se utiliza el marco de la cámara, con origen en C . Así, utilizando la Ec. 5.74 y 5.75,

$$\mu \mathbf{P}^{-1} \mathbf{x} - \lambda \left((\mathbf{P}^{-1} \mathbf{x}) \times \left(R_C^{C'} (\mathbf{P}')^{-1} \mathbf{x}' \right) \right) - \mu' \cdot R_C^{C'} (\mathbf{P}')^{-1} \mathbf{x}' - T = 0. \quad (5.90)$$

Note que μ, μ', λ son variables que se introducen para ajustar la posición de los puntos de corte Q y Q' , así como la longitud del segmento perpendicular común, obtenido por el producto vectorial de los dos rayos. La Ec. 5.90 representa un sistema de tres ecuaciones simultáneas con tres incógnitas (μ, μ', λ). El sistema debe ser resuelto para encontrar sus valores, mediante los cuales es posible calcular las coordenadas del punto \mathbf{X} como:

$$\mathbf{X} = \overrightarrow{CQ} - \frac{\lambda}{2} (\overrightarrow{C\mathbf{x}} \times \overrightarrow{C'\mathbf{x}'}), \quad (5.91)$$

$$\mathbf{X} = \mu \mathbf{P}^{-1} \mathbf{x} - \frac{\lambda}{2} \left((\mathbf{P}^{-1} \mathbf{x}) \times \left(R_C^{C'} (\mathbf{P}')^{-1} \mathbf{x}' \right) \right). \quad (5.92)$$

Se debe tomar en cuenta que a pesar de la aparente complejidad de la Ec. 5.92 es posible, operando simbólicamente, obtener una expresión simplificada de fácil implementación computacional. Para ello es necesario calcular los parámetros de cada una de las cámaras, así como su configuración geométrica.

Además, es importante observar que los pixeles correspondientes \mathbf{x} y \mathbf{x}' se encuentran en el marco de la imagen. Para poder llevar a cabo una implementación directa, tal como se leen los datos de la imagen, es necesario llevar a cabo la transformación a coordenadas de la representación matricial (Ec. 5.50).

Otra alternativa para triangular un punto, en aquellos casos en que no se cuenta con los elementos para acotar de forma certera el error, es llevar a cabo un proceso de optimización numérica. Recordando las Ec. 5.74 y 5.75, es posible definir la función objetivo como:

$$f_{obj} = ((\mathbf{x} \cdot \mathbf{X}_{[3]} - \mathbf{P}\mathbf{X}) \bullet (\mathbf{x} \cdot \mathbf{X}_{[3]} - \mathbf{P}\mathbf{X})) + ((\mathbf{x}' \cdot \mathbf{X}'_{[3]} - \mathbf{P}'\mathbf{X}') \bullet (\mathbf{x}' \cdot \mathbf{X}'_{[3]} - \mathbf{P}'\mathbf{X}')), \quad (5.93)$$

donde \mathbf{X} corresponde al punto espacial que se desea triangular, visto desde el marco de la cámara principal, \mathbf{X}' es el mismo punto pero observado desde el marco de la cámara secundaria, $\mathbf{X}_{[i]}$ es la componente i del vector \mathbf{X} y $\mathbf{X}'_{[i]}$ es la componente i del vector \mathbf{X}' . Para poder llevar a cabo la optimización \mathbf{X} y \mathbf{X}' se deben expresar en el mismo marco de referencia, de esta forma, haciendo uso de los parámetros extrínsecos que relacionan las cámaras, se tiene que:

$$\mathbf{X}' = R_C^{C'-1} (\mathbf{X} - T). \quad (5.94)$$

Así, la función objetivo se reescribe como:

$$f_{obj} = ((\mathbf{x} \cdot \mathbf{X}_{[3]} - \mathbf{P}\mathbf{X}) \bullet (\mathbf{x} \cdot \mathbf{X}_{[3]} - \mathbf{P}\mathbf{X})) + \left((\mathbf{x}' \cdot \mathbf{X}'_{[3]} - \mathbf{P}'R_C^{C'-1} (\mathbf{X} - T)) \bullet (\mathbf{x}' \cdot \mathbf{X}'_{[3]} - \mathbf{P}'R_C^{C'-1} (\mathbf{X} - T)) \right) \quad (5.95)$$

donde las variables a optimizar son $X_c, Y_c, Z_c \in \mathbb{R}$.

CAPÍTULO 6

SUBSISTEMA DE VISIÓN COMPUTACIONAL. DESARROLLO DE SOFTWARE

Cualquier tecnología suficientemente avanzada es indistinguible de la magia
[ARTHUR CLARKE]

6.1. Requerimientos de software

En este capítulo se presenta el desarrollo del software que complementa el hardware descrito en la Sec. 4.3, para el desarrollo del módulo de visión computacional. El desarrollo de éste se enfoca en satisfacer los requerimientos generados al establecer su función dentro del sistema, estos son:

- Desarrollado bajo una estructura modular de funciones, con capacidad de intercomunicación y compartición de datos, que facilite la incorporación de nuevas capacidades en forma de módulos.
- Implementar el desarrollo en un lenguaje orientado a objetos que permita el manejo directo de la memoria, cuente con estructuras de datos abstractos, mecanismos de paralelismo, y bibliotecas para el manejo de información visual, álgebra lineal, métodos numéricos, y manejo de periféricos.
- Ser independiente del hardware, es decir, no debe existir una relación de dependencia explícita, para evitar la obsolescencia entre el código desarrollado y el hardware en que se ejecuten las instrucciones que éste genera. El código debe ser compilable sobre la mayoría de las plataformas de hardware comerciales.
- Compilado para ejecutarse sobre un sistema UNIX / Linux, para aprovechar las capacidades de configuración y estabilidad que estos sistemas ofrecen.

6.2. Detección e identificación de marcas de color

Como se ha expuesto, a cada una de las tuercas de ajuste le ha sido colocada una marca de color para facilitar su identificación por medio del subsistema de visión computacional. El objetivo es lograr que el dicho subsistema extraiga, con base en la imagen captada por la cámara, la posición de cada una de las marcas dentro de su campo visual, ubicando su centroide e identificando el color de la misma. Con la información obtenida es posible generar un mapa de la configuración global del ambiente observado, que es un paso fundamental en el desarrollo de este trabajo.

Para llevar a cabo la detección e identificación de las marcas se ha diseñado el algoritmo que se ilustra en la Fig. 6.1, y que se detalla a continuación.

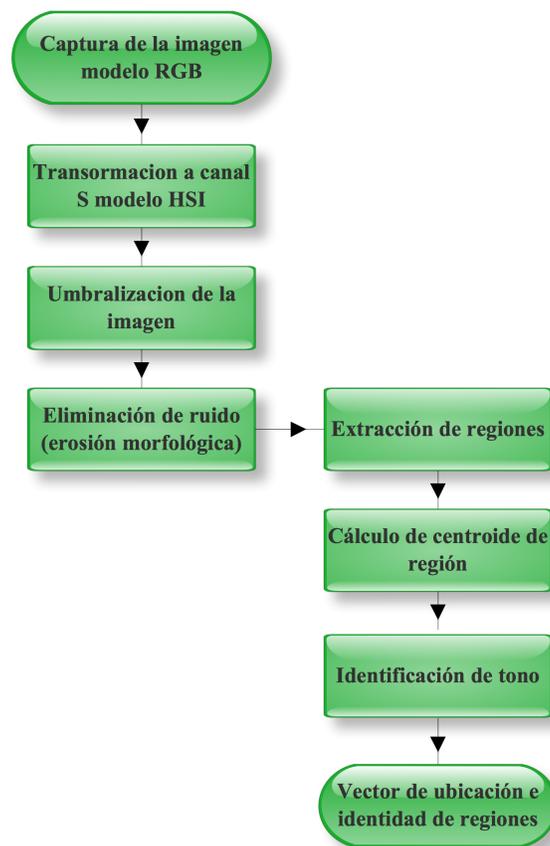


Fig. 6.1: Diagrama de flujo del algoritmo de detección e identificación de marcas de color

6.2.1. Eliminación de fondo y realce de pixeles de color saturado

Una vez la imagen ha sido capturada por la cámara (Fig. 6.2), es necesario llevar a cabo un filtrado para eliminar el fondo y realzar solo las áreas de la imagen que corresponden a las marcas de identificación.

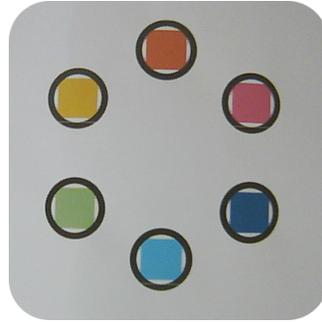
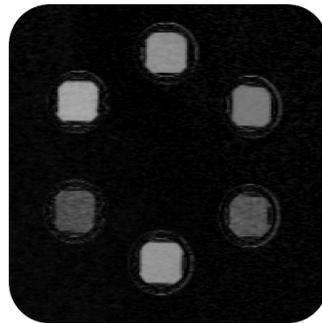


Fig. 6.2: Imagen con marcas de color

Tomando como base el hecho de que dichas marcas han sido diseñadas utilizando colores saturados, es decir, con baja concentración de luz blanca, el primer paso consiste en llevar a cabo una transformación del modelo de color de la imagen, teniendo como entrada el cuadro original en RGB y como salida una imagen correspondiente al canal S del modelo HSI. Con esta transformación se logra que aquellos colores con baja saturación en la imagen original tengan valores pequeños en la imagen resultante, mientras que los colores puros, o de alta saturación, adopten valores mayores. De esta manera, mediante una sola transformación es posible discriminar entre las zonas que corresponden a las marcas de identificación y aquellas que no los son, las cuales se consideran para fines prácticos como fondo.

La Fig. 6.3 muestra el resultado de aplicar la transformación a canal S de una imagen RGB (Fig. 6.2), donde se aprecia la capacidad de la transformación para resaltar los colores saturados, ocultando zonas de fondo blanco o superficies metálicas claras.

Fig. 6.3: Canal S de la Fig. 6.2

6.2.2. Umbralización de la imagen en canal S

El proceso de umbralización consiste en catalogar los píxeles de una imagen dentro de dos o más categorías, tomando como referencia un criterio determinado y asignando a cada píxel el valor de la categoría a la que corresponde. En particular, la umbralización simple consiste en binarizar la imagen, es decir, asignar a cada píxel un valor $f_{ij} \in \{0, 1\}$, dependiendo de si el valor para dicho píxel bajo el criterio adoptado es menor o mayor a un determinado umbral T_h .

Una vez que se tiene la imagen en niveles de saturación, es necesario aislar las áreas correspondientes a las marcas de identificación, con tal fin, se binariza la imagen. El criterio adoptado para llevar a cabo la segmentación es comparar el valor de cada pixel con el umbral T_h . El valor de T_h se determina experimentalmente. La Fig. 6.4 muestra una imagen binarizada para diferentes valores del umbral T_h .

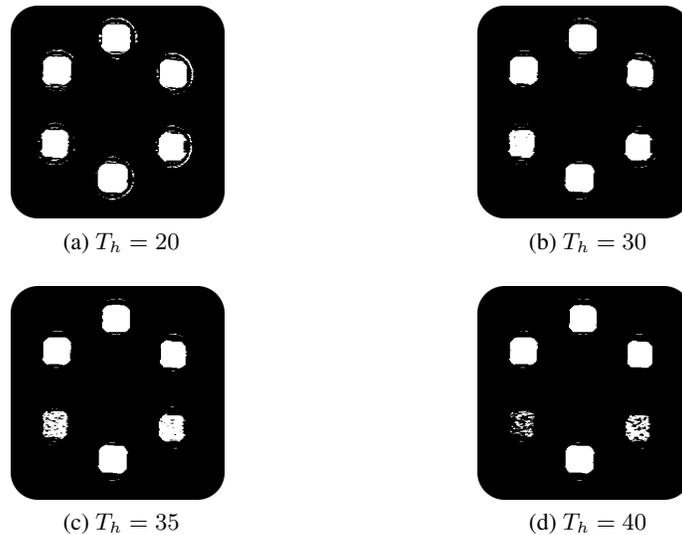


Fig. 6.4: Imágenes binarizadas

Una vez realizadas las pruebas, y aplicando diferentes niveles de iluminación, el valor del umbral elegido es $T_h = 30$. Sin embargo, este valor puede ser ajustado para trabajar en condiciones inusuales, dotando al sistema de un cierto grado de adaptabilidad.

6.2.3. Eliminación de ruido en imagen binaria

Comúnmente, la imagen binaria obtenida en el proceso anterior contiene regiones ruidosas, es decir, secciones de la imagen que adoptan el valor de 1 aún cuando no corresponden a marcas de identificación. Estas secciones son fáciles de identificar pues generalmente se encuentran como puntos aislados, o como sucesiones de puntos tipo borde (Fig. 6.5).

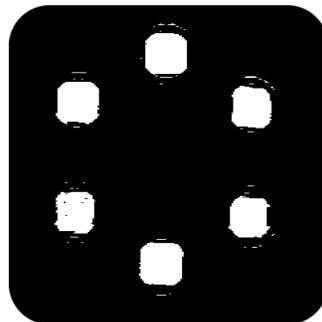


Fig. 6.5: Imagen binaria con ruido

La morfología matemática es una herramienta comúnmente utilizada en procesos de visión computacional para suavizar los bordes de una región, separar determinadas regiones que el proceso de segmentación presenta unidas, unir regiones, y eliminar puntos aislados [71]. Las operaciones morfológicas fundamentales son la dilatación y la erosión. En particular, la erosión consiste en barrer una imagen A con una máscara (kernel) B , haciendo coincidir el centro de la máscara con cada uno de los píxeles A_{ij} de la imagen. Si al colocar el centro de la máscara B en el píxel A_{ij} , los valores de la imagen bajo la máscara coinciden con los de B de forma exacta, el píxel (i, j) de la imagen resultante será igual a 1, en caso contrario, el valor será 0 (Fig. 6.6).

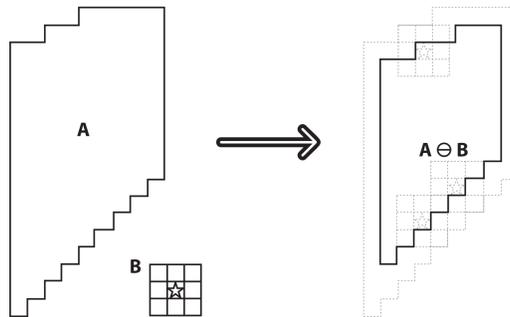


Fig. 6.6: Erosión morfológica

Para eliminar el ruido en la imagen binarizada de las marcas de identificación se realiza un ciclo de erosión en la imagen completa, los resultados se muestran en la Fig. 6.7.

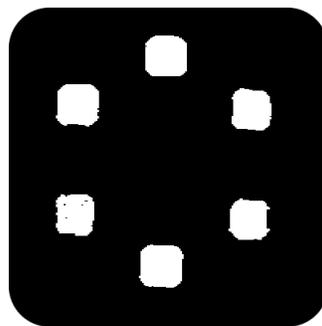


Fig. 6.7: Resultado de aplicar la erosión a Fig. 6.5

6.2.4. Extracción de regiones

Para la identificación de las marcas resultantes del proceso de erosión, es necesario etiquetar las regiones existentes y determinar qué píxeles corresponden a cada una de dichas regiones. El proceso de detección y ubicación de marcas que se lleva a cabo en cada una de las imágenes que captura la cámara está basado en el algoritmo de componentes conexas para grafos [71], y adaptado particularmente a la tarea que se desarrolla. Dicho algoritmo se describe a continuación.

Una vez que se tiene la imagen binarizada del paso anterior, ésta se recorre píxel a píxel en busca

de aquellos pixeles con valor de 1. Cada vez que uno de ellos se encuentra se le asigna una etiqueta numérica, dependiendo de alguno de los siguientes criterios:

1. Si ninguno de los 8-vecinos del pixel está etiquetado, se crea una nueva etiqueta y se le asigna al pixel en cuestión.
2. En caso de que uno o más de los 8-vecinos del pixel tenga etiqueta, el valor que se asignará al pixel bajo análisis será igual a la etiqueta de menor valor de sus vecinos.

Las etiquetas se almacenan en un arreglo independiente con las mismas dimensiones de la imagen original. Cada vez que a un pixel le es asignada su etiqueta, las coordenadas del pixel se almacenan en el contenedor correspondiente a la región identificada por dicha etiqueta.

La Fig. 6.8 muestra un ejemplo del proceso de etiquetado para los dos casos descritos.

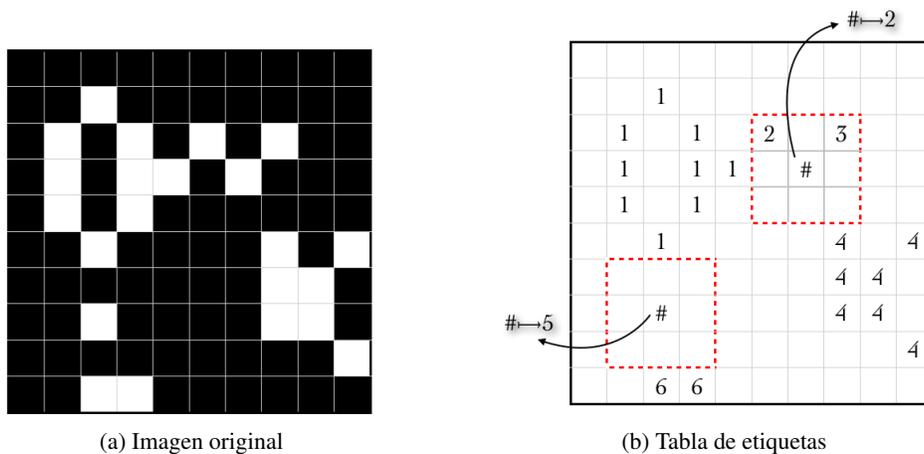


Fig. 6.8: Etiquetado de pixeles

Concluida la asignación de etiquetas, se ejecuta un proceso de reetiquetado, donde se buscan etiquetas para las que sus 8-vecinos estén etiquetados con valores menores al propio. Se crea una lista de correspondencias donde se almacena el valor original de la etiqueta y el menor de los valores que la circundan. Con esto se logra corregir la verdadera identidad (etiqueta real) de la región conexa detectada, y los valores de las coordenadas almacenados en el contenedor de la etiqueta original se pasan al contenedor de la etiqueta real. El concepto de reetiquetado se ilustra en la Fig. 6.9.

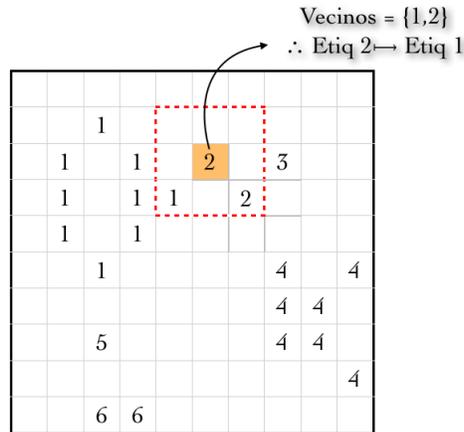


Fig. 6.9: Reetiquetado de regiones

Una vez se han unificado los píxeles de las regiones conexas bajo una misma etiqueta, se calcula el centroide de la región k como:

$$\{X_{c_k}, Y_{c_k}\} = \frac{1}{N} \cdot \left\{ \sum_{h=1}^N x_h, \sum_{h=1}^N y_h \right\}, \quad (6.1)$$

donde X_{c_k} es la columna en la cual se ubica el centroide de la región k , Y_{c_k} es la fila en que se encuentra el centroide, N es el número de píxeles que conforman la región, x_j e y_j son los valores de la columna y la fila respectivamente, de cada uno de los píxeles de la región.

Finalmente, para cada región se calcula el tono de la etiqueta que la identifica. El tono se obtiene transformando al canal H el valor RGB promedio entre el pixel del centroide y sus vecinos izquierdo y derecho sobre su misma fila (Fig. 6.10).

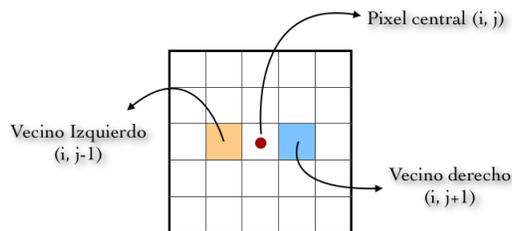


Fig. 6.10: Vecinos izquierdo y derecho del pixel centroide

La Fig. 6.11 muestra el resultado gráfico de extracción e identificación de regiones, así como un listado de las regiones encontradas y sus propiedades (coordenadas del centroide y tono característico).

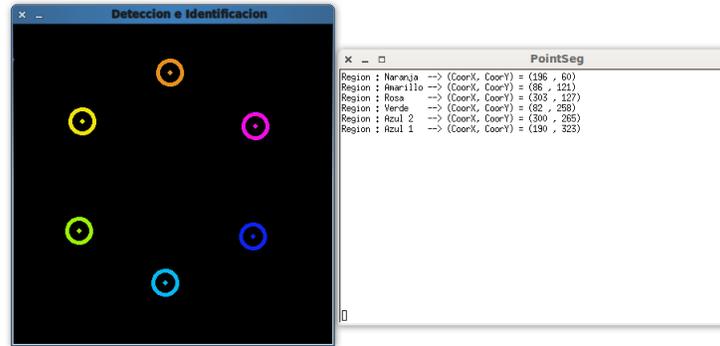


Fig. 6.11: Resultado de extracción e identificación de regiones

6.3. Generación de mapas de disparidad. Percepción de la tercera dimensión

La visión estereoscópica es una técnica muy útil para la obtención de información acerca del medio observado. Los algoritmos de reconocimiento utilizados en visión y en visión estereoscópica pueden ser usados como fuente de información sensorial y sistema de control para robots, en una amplia gama de tareas [2, 4, 35, 53, 74].

Un punto clave para la visión estereoscópica es determinar qué pixel o región de la imagen tomada por una cámara está relacionada con la imagen tomada por la otra cámara. Para establecer la correspondencia entre las imágenes de entrada es necesario estimar qué puntos característicos de ambas imágenes son proyectados por el mismo punto 3D del espacio [102]. Por medio de la correspondencia encontrada es posible determinar la distancia al punto que pertenece a ambas imágenes. La correspondencia entre puntos de imágenes estereoscópicas es la primera tarea de la visión estereoscópica, la calidad de la correspondencia determina el nivel de precisión que se puede alcanzar en la medición de distancias.

Una gran cantidad de algoritmos han sido propuestos, pero sigue siendo difícil hacer una correspondencia muy precisa [54]. La precisión en las mediciones sobre el ambiente observado depende en gran manera de la precisión de la correspondencia que las origina. Una extensa investigación ha demostrado [17, 40] que la correspondencia de características entre imágenes es en general un problema complicado.

A continuación se presentan los algoritmos desarrollados para determinar la correspondencia entre pixeles de un par de imágenes estereoscópicas.

6.3.1. Algoritmo de comparación de nivel de brillo entre pixeles

El algoritmo de búsqueda de correspondencia entre puntos de imágenes estereoscópicas por comparación de nivel de brillo consiste en: para cada pixel j en la fila i de la imagen derecha (Img_d), la fila i

de la imagen izquierda (Img_i) es escaneada en busca de un pixel con el nivel de brillo más cercano al pixel de la imagen derecha en cuestión (Fig. 6.12).

$$\text{Min } |Img_d(i, j) - Img_i(i, k)|, \quad k = j \dots w - 1 \quad (6.2)$$

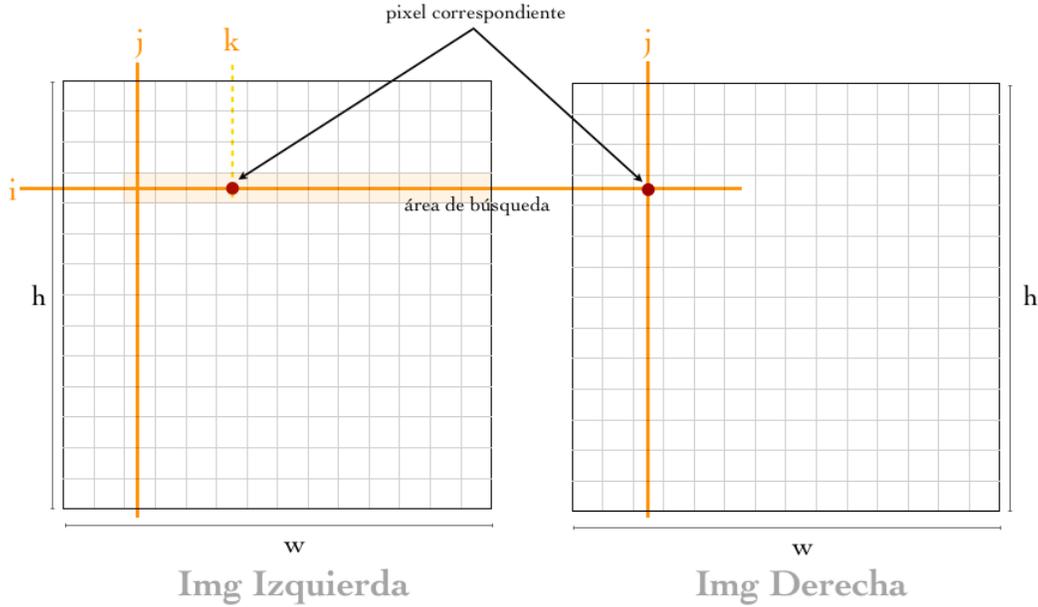


Fig. 6.12: Comparación de nivel de brillo entre pixeles

La disparidad (d_{ij}) se define como la diferencia entre la coordenada k de un pixel en la imagen izquierda y la coordenada j del pixel correspondiente en la imagen derecha, $d_{ij} = k - j$, donde el subíndice i representa la fila en la que se encuentra el pixel de la imagen derecha. Los valores de disparidad obtenidos al aplicar el algoritmo de comparación de nivel de brillo se almacenan en un arreglo matricial denominado matriz de disparidad (Fig. 6.13), donde cada valor d_{ij} representa la distancia entre el pixel (i, j) de la imagen derecha y su correspondiente en la imagen izquierda. Si no se encuentra una correspondencia entre pixeles, el valor de d_{ij} es cero.

$$\begin{matrix} d_{0,0} & d_{0,1} & d_{0,2} & \cdots & d_{0,h-1} \\ d_{1,0} & d_{1,1} & 0 & \cdots & d_{1,h-1} \\ 0 & d_{2,1} & 0 & \cdots & d_{2,h-1} \\ \vdots & & & & \vdots \\ d_{w-1,0} & d_{w-1,1} & d_{w-1,2} & \cdots & d_{w-1,h-1} \end{matrix}$$

Fig. 6.13: Disparidad entre pixeles correspondientes de la imagen derecha e izquierda

Con el fin de obtener resultados visualmente interpretables, la matriz de disparidad se expresa como una imagen monocromática, denominada mapa de disparidad, en el que cada nivel de disparidad se percibe como un tono de gris distinto.

Para la implementación del algoritmo diseñado se utiliza el lenguaje C++ y la librería de visión OpenCV [70]. La secuencia de ejecución del algoritmo se presenta en la Fig. 6.14 y cada uno de sus bloques funcionales se describe a continuación.

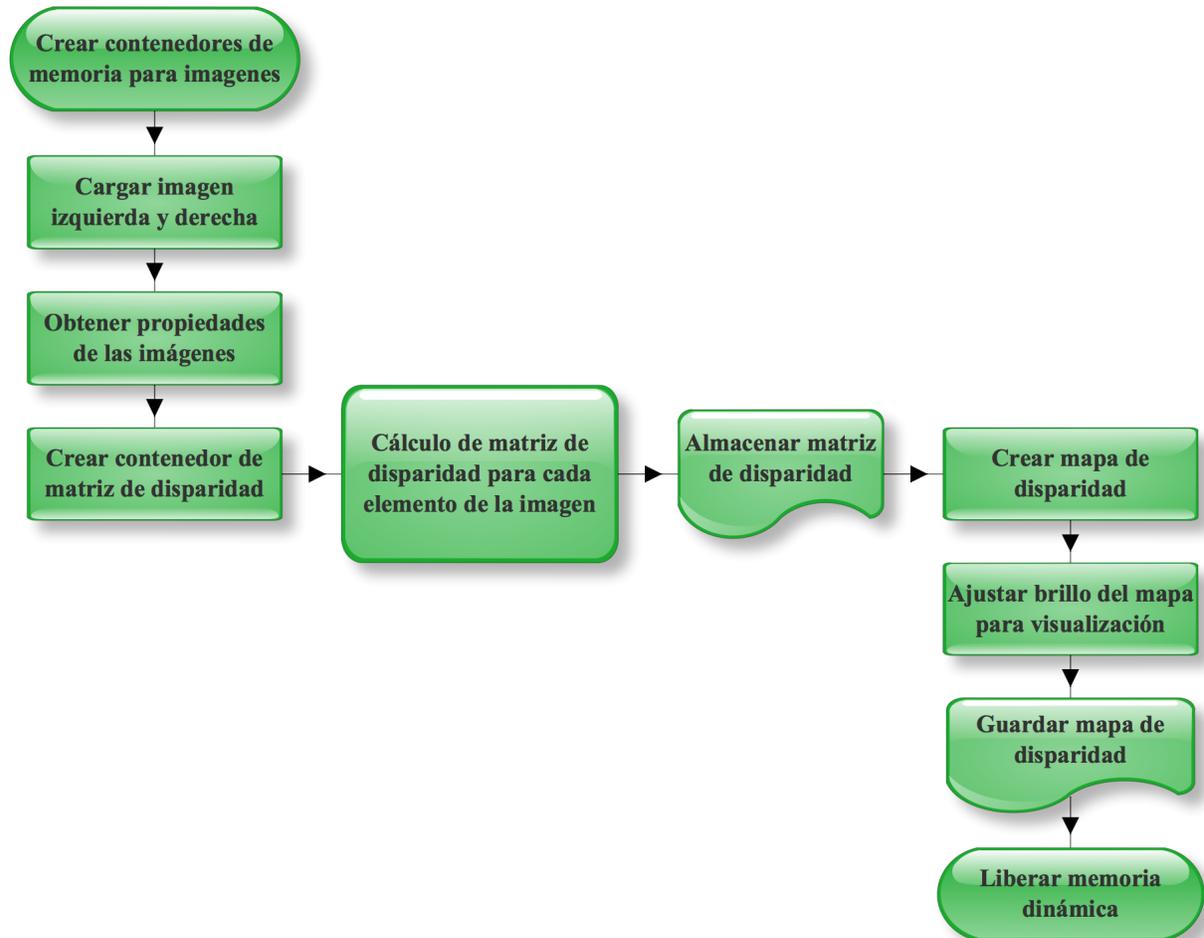


Fig. 6.14: Diagrama de flujo del algoritmo de comparación de nivel de brillo

1. **Crear contenedores de imagen.** El primer paso para la búsqueda de correspondencia entre píxeles de la imagen es reservar la memoria dinámica necesaria para almacenar las imágenes que serán tratadas en el proceso. En este caso deben ser creados tres contenedores, uno para la imagen derecha, otro para la imagen izquierda y finalmente un tercero que almacenará la matriz de disparidad que se obtiene como resultado. Si bien se trata de una matriz de distancias, la base de la estructura de un contenedor de imagen es una matriz bidimensional, por lo tanto el almacenamiento es posible sin que sea necesario ningún tipo de modificación en la información.
2. **Cargar imágenes en memoria.** Para acceder a la información de cada píxel de las imágenes es necesario cargarlas en memoria desde origen. Cada una de las imágenes es enlazada con el contenedor que le corresponde. Para este algoritmo se trabaja con imágenes en escala de grises con profundidad de 8 bits. Si la imagen original no se encuentra en este formato, debe ser

adaptada al momento de cargarla.

- 3. Obtener propiedades de la imagen y crear matriz de disparidad.** La matriz de disparidad debe contener la información de los píxeles correspondientes a cada uno de los puntos de la imagen, es por esto que la matriz debe ser de igual tamaño que el par de imágenes estereoscópicas de donde se obtiene. Haciendo uso de la información de tamaño y número de canales contenida en la variable que representa la imagen derecha se crea la matriz de disparidad (en formato de imagen) que contendrá el resultado del algoritmo descrito. Cada uno de los valores de disparidad se almacena como un entero de 8 bits sin signo, es decir, un valor entre 0 y 255, puesto que el cálculo de la disparidad $d_{ij} = k - j$ arrojará siempre valores positivos, ya que por las propiedades del par estereoscópico (Fig. 6.12) el pixel de la imagen derecha (i, j) siempre se encuentra en una columna j menor al valor de la columna k en que está el pixel correspondiente en la imagen izquierda (i, k) .
- 4. Cálculo de la matriz de disparidad.** El punto medular del algoritmo de comparación de nivel de brillo es la forma en que determina la matriz de disparidad. La operación fundamental es el cálculo del valor absoluto de la diferencia de brillo entre dos píxeles. El valor obtenido es comparado con el umbral t_k .

$$|Img_d(i, j) - Img_i(i, k)| \leq t_k. \quad (6.3)$$

Si la Ec. 6.3 se cumple, los píxeles (i, j) de la imagen derecha e (i, k) de la imagen izquierda son correspondientes. En la posición (i, j) de la matriz de disparidad se almacena el valor correspondiente a $d_{ij} = k - j$ y el algoritmo pasa a analizar el siguiente pixel de la fila i . Si no existe un pixel en la fila i de la imagen izquierda que cumpla con la Ec. 6.3 para un pixel (i, j) dado de la imagen derecha, entonces el valor de d_{ij} de la matriz de disparidad será igual a cero. El proceso se repite para cada uno de los píxeles de la imagen.

Para reducir el tiempo de ejecución del algoritmo, el escaneo se efectúa seleccionando el pixel (i, j) de la imagen derecha y buscando su par en la imagen izquierda, comenzando en el pixel (i, k) donde $k = j$, evitando calcular la diferencia de brillo, el valor absoluto y la comparación con el intervalo t_k para todos los píxeles previos a k , con base en el hecho de que el pixel (i, j) de la imagen derecha no encontrará en la imagen izquierda su par correspondiente en una columna con valor menor a j , debido al fenómeno de la intersección de campo visual que da lugar a la visión estereoscópica.

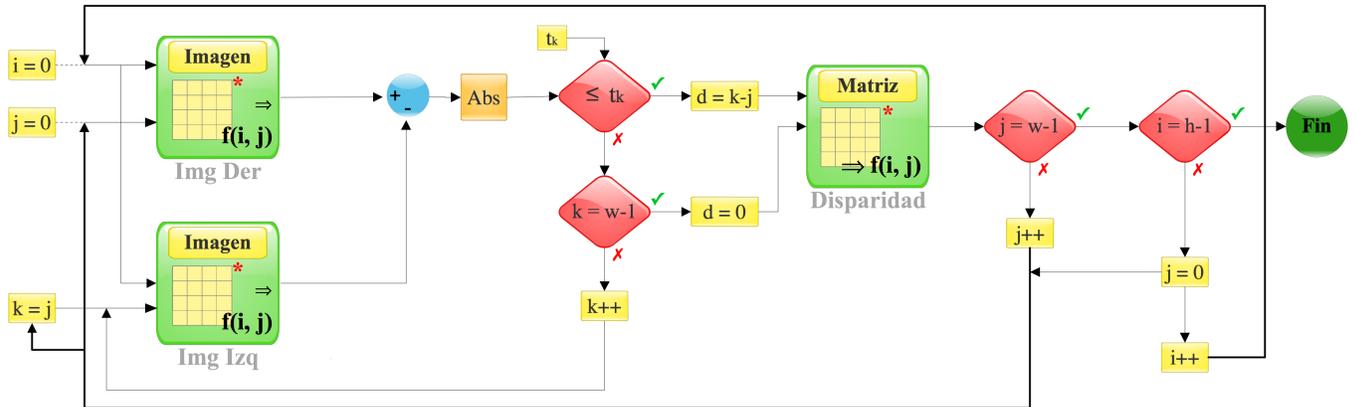


Fig. 6.15: Algoritmo para el cálculo de la matriz de disparidad usando comparación de brillo entre pixeles

5. **Almacenar matriz de disparidad.** Previo a la visualización de los resultados obtenidos es necesario almacenar la matriz de disparidad calculada, en este punto los valores no han sufrido ninguna alteración y es conveniente para el cálculo de distancias almacenarlos de esta forma. La matriz se almacena en disco como si se tratase de una imagen, sin embargo, vía código dicha imagen se observa como una matriz de valores discretos, a la que se puede acceder simplemente cargándola en memoria.
6. **Ajuste de brillo de mapa de disparidad.** Dado que la distancia entre coordenadas de dos pixeles correspondientes no es, en general, un valor muy grande, la imagen generada a partir de la matriz de disparidad y denominada mapa de disparidad es oscura, evitando que los detalles se perciban correctamente. Para mejorar la calidad de la imagen es necesario aumentar la intensidad en los tonos de gris, volviendo más clara la imagen que representa los resultados. El proceso consiste en buscar el máximo valor almacenado en la matriz de disparidad. Una vez localizado, definir un factor de aumento, como el valor máximo de brillo que se desea en el mapa de disparidad menos el valor máximo encontrado en la matriz. A continuación, a cada uno de los pixeles de la matriz de disparidad se le suma dicho factor, aumentando así la calidad visual de la imagen y generando el mapa de disparidad con que se evaluará la calidad del algoritmo diseñado.
7. **Guardar mapa de disparidad.** Para la interpretación de los resultados se requiere tener acceso al mapa de disparidad calculado por el algoritmo para distintos niveles del umbral t_k , en los diferentes pares de imágenes estereoscópicas analizados. Con tal motivo se almacena en disco la imagen generada. Así mismo el mapa de disparidad, junto con el par de imágenes estereoscópicas analizado, se muestra en pantalla.
8. **Liberación de memoria dinámica.** Finalmente, concluido el proceso es necesario descargar las imágenes y liberar la memoria ocupada por las mismas.

6.3.2. Resultados. Aplicación del algoritmo de comparación de nivel de brillo

Para llevar a cabo las pruebas del algoritmo de comparación de nivel de brillo entre píxeles se seleccionaron un conjunto de pares estereoscópicos de la base de datos CMU VASC Image Database [94]. Fueron utilizadas imágenes en escala de grises rectificadas, a las cuales se les aplicó el algoritmo diseñado.

La Fig. 6.16 presenta una par estereoscópico original, denominado “Habitación”, utilizado como entrada para la aplicación desarrollada.

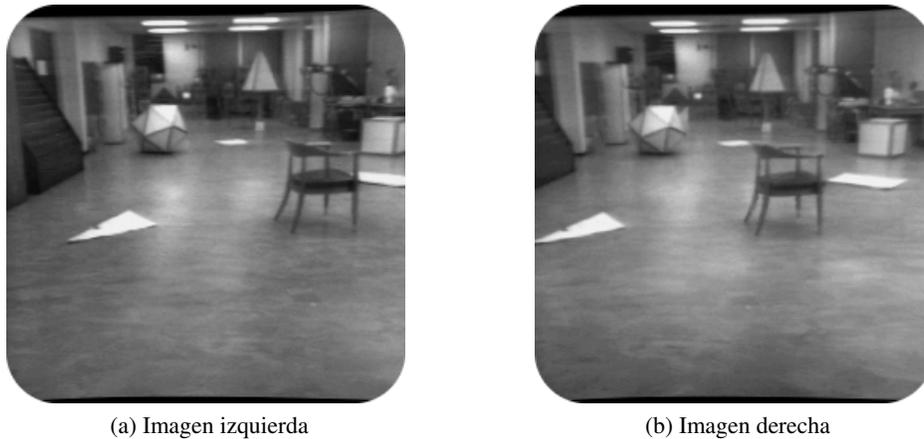


Fig. 6.16: Par estereoscópico original (Habitación)

Los resultados obtenidos (Fig. 6.17) al aplicar el algoritmo de correspondencia al par de la Fig. 6.16 muestran que el algoritmo de comparación de nivel de brillo entre píxeles tiene habilidad para encontrar la correspondencia entre píxeles de un par estereoscópico. La matriz de disparidad obtenida tiene un alto nivel de ruido para valores bajos del umbral t_k (por ejemplo $t_k = 1$, $t_k = 5$), sin embargo, es posible apreciar objetos bien definidos, como la silla que aparece en el primer plano. Incrementando el valor de t_k es posible obtener matrices de disparidad con menor nivel de ruido (por ejemplo $t_k = 9$, $t_k = 13$). Para estos casos, la disparidad en planos equidistantes es uniforme, y las transiciones entre niveles de disparidad son suaves en aquellos objetos donde la profundidad es apreciable.

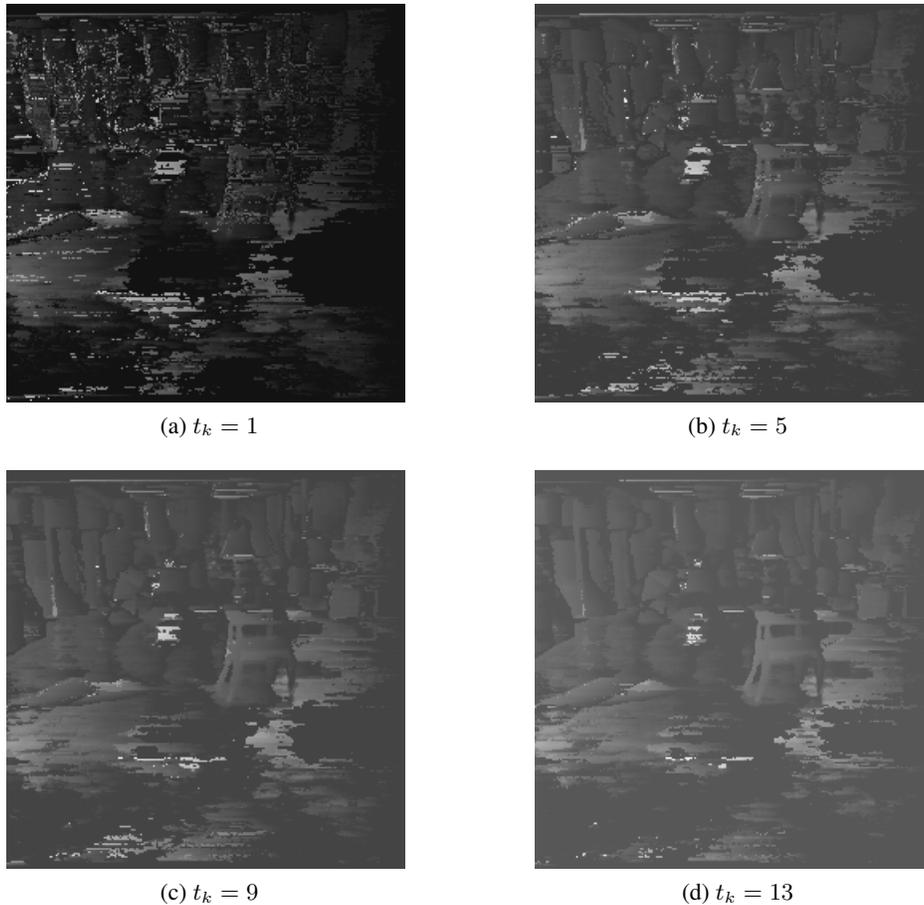


Fig. 6.17: Mapas de disparidad para distintos valores del umbral t_k sobre el par “Habitación”

Si bien los resultados obtenidos al aplicar el algoritmo de búsqueda de correspondencia por comparación de brillo de píxeles hacen posible formar un mapa de disparidad en el que se pueden apreciar ciertos objetos, el nivel de ruido es alto, teniendo en cuenta la aplicación en que se desea emplear el cálculo de disparidad obtenido. Así mismo, la frecuencia con que varían los niveles de disparidad entre píxeles próximos, principalmente entre píxeles de fondo o píxeles en áreas afectadas por múltiples fuentes de iluminación, hace que los resultados sean difíciles de interpretar, introduciendo errores significativos en la determinación de la posición tridimensional de los objetos en escena. Motivo por el cual el algoritmo de correspondencia basado en la comparación del nivel de brillo no cumple satisfactoriamente con las necesidades expuestas, dejando clara la necesidad de desarrollar un algoritmo que haga un análisis del área circundante al píxel, y no solo de este.

Los resultados obtenidos al aplicar el algoritmo al par estereoscópico “Cubo” (Fig. 6.18) se presentan en la Fig. 6.19. En este caso la imagen corresponde a un par de cuerpos bien definidos, ubicados muy cerca del lente de la cámara. La calidad del mapa de disparidad aumenta, al igual que en el par presentado previamente, a medida que se aumenta el valor del intervalo t_k .

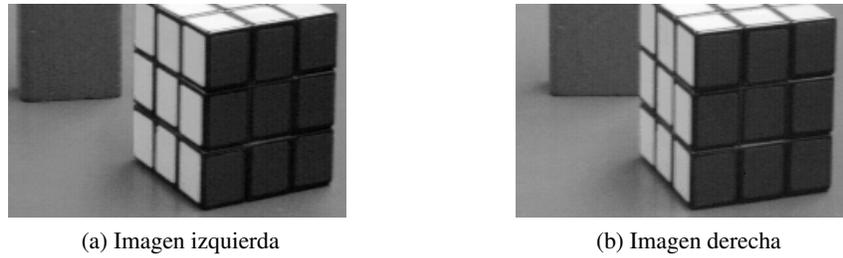


Fig. 6.18: Par estereoscópico original (Cubo)

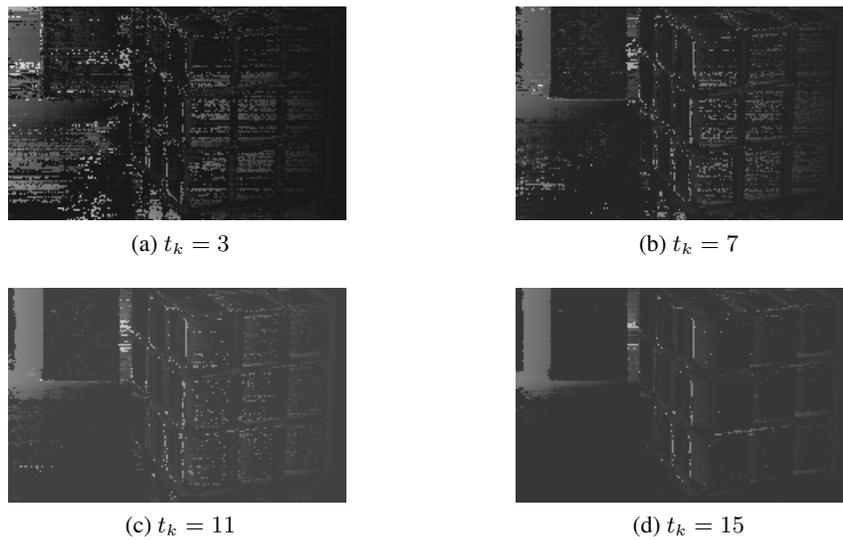


Fig. 6.19: Mapas de disparidad para distintos valores del umbral t_k del par "Cubo"

6.3.3. Algoritmo de comparación de histogramas entre pixeles

Con base en los resultados obtenidos al realizar el cálculo de la disparidad con el algoritmo de comparación de brillo entre pixeles, se propone un nuevo algoritmo que compare no solo el estado del pixel aislado, sino una medida del estado de sus vecinos. Es decir, la correspondencia de los pixeles de un par estereoscópico se establece al comparar la similitud entre las áreas circundantes a estos. La medida que determina las propiedades del área circundante es el histograma.

La forma en que se debe recorrer la imagen, y el calculo de la disparidad ($d_{ij} = k - j$) en este nuevo algoritmo es exactamente igual que la utilizada en el algoritmo de comparación de brillo. El cambio radica en la información en que se basa la selección de la correspondencia, mientras que en el algoritmo anterior se basa únicamente en encontrar una diferencia de brillo entre pixeles, que esté por debajo de un determinado umbral, la búsqueda de correspondencias por histogramas busca la máxima similitud entre vecindades de pixeles. Cuando una correspondencia exacta es encontrada, no existe diferencia alguna entre sus vecindades, es decir, la similitud es total.

De esta forma es posible definir la similitud como el valor real que se obtiene al calcular la norma euclidiana ($\|\cdot\|$) de la diferencia entre el vector de propiedades de vecindad de dos pixeles distintos, cada uno de ellos en una de las imágenes. Cuando la similitud sea total (correspondencia exacta) el valor obtenido será 0, y mientras mayor sea la diferencia entre las vecindades comparadas, mayor será el valor obtenido.

Dado que las imágenes cuentan con ruido, no es posible encontrar correspondencias exactas para cada uno de los pixeles en ellas, por tanto el valor obtenido con la norma euclidiana es comparado con el umbral t_h .

$$\|\mathbf{u}_d(i, j) - \mathbf{u}_i(i, k)\| \leq t_h, \quad (6.4)$$

donde $\mathbf{u}_d(i, j)$ y $\mathbf{u}_i(i, k)$ son vectores que contienen las propiedades de la vecindad del pixel (i, j) de la imagen derecha e (i, k) de la imagen izquierda, respectivamente. Si la Ec. 6.4 se cumple, los pixeles (i, j) de la imagen derecha e (i, k) de la imagen izquierda son correspondientes. Si no existe un pixel en la fila i de la imagen izquierda que cumpla con la Ec. 6.4 para un pixel (i, j) dado de la imagen derecha, entonces el valor de d_{ij} de la matriz de disparidad será igual a cero.

Dicha similitud se define en términos de los vectores (\mathbf{u}) de propiedades de la vecindad, que gracias a su construcción, se pueden definir en términos del histograma de brillo, de contraste, o de microcontornos, o incluso de una combinación de estos.

El vector \mathbf{u} correspondiente a un pixel determinado se ensambla ordenando las lecturas del histograma h_1 para cada uno de sus intervalos de conteo, en el orden ascendente del intervalo. Si se decide incluir la información de un segundo histograma h_2 , sus lecturas serán ordenadas en el mismo sentido, y enlistadas tras las lecturas de h_1 , donde $h_1, h_2 \in \{\text{brillo, contraste, microcontornos}\}$. De esta forma se puede extender la aplicación a n histogramas u otro tipo de propiedades a comparar.

Con respecto a la secuencia de ejecución del algoritmo de comparación de brillo (Fig. 6.14), es necesario considerar, previo al cálculo de la matriz de disparidad, el cálculo de los histogramas que serán tomados en cuenta, para cada uno de los pixeles en la zona de interés.

6.3.4. Resultados. Aplicación del algoritmo de comparación de histogramas

Las pruebas se llevaron a cabo utilizando el par estereoscópico denominado “Habitación” (Fig. 6.16). Las Fig. 6.20, 6.21, y 6.22 muestran los resultados obtenidos al aplicar el algoritmo de búsqueda de correspondencias por comparación de histogramas de brillo, para umbrales $t_h = 8$, $t_h = 16$, $t_h = 32$ respectivamente.

Así mismo, para cada uno de los valores de T_h se presenta el mapa de disparidad, obtenido utilizando ventanas de análisis $S_{m \times n}$, con $(m, n) = \{(3, 3), (5, 5), (7, 7)\}$. Como se describió, este algoritmo lleva a cabo una búsqueda más exhaustiva, y con una ventana de búsqueda de $(3, 3)$. Sus resultados se asemejan en gran medida a los conseguidos con el algoritmo de comparación de brillo, sin embargo, la imagen comienza a degenerarse con ventanas mayores o iguales a $(7, 7)$, hecho que guarda relación

con la cantidad de ruido de la imagen, pues, con base en los resultados obtenidos, el algoritmo de comparación de histogramas resulta muy sensible al ruido para ventanas suficientemente grandes.

Sin embargo, los mejores resultados se obtienen con ventanas de $(5, 5)$, encontrando un equilibrio entre la obtención de zonas de disparidad continua y transiciones suaves (como se espera que ocurra), y la tolerancia al ruido. Se debe apreciar que los resultados obtenidos corresponden solamente a la comparación del histograma de brillo. Una opción para compensar la sensibilidad al ruido es tomar en cuenta el histograma de contraste, e incluso el de microcontornos.

Como era de esperar, por la construcción del algoritmo, se obtiene mejores resultados que los obtenidos con la comparación de brillo, pero el proceso computacional es considerablemente más costoso.

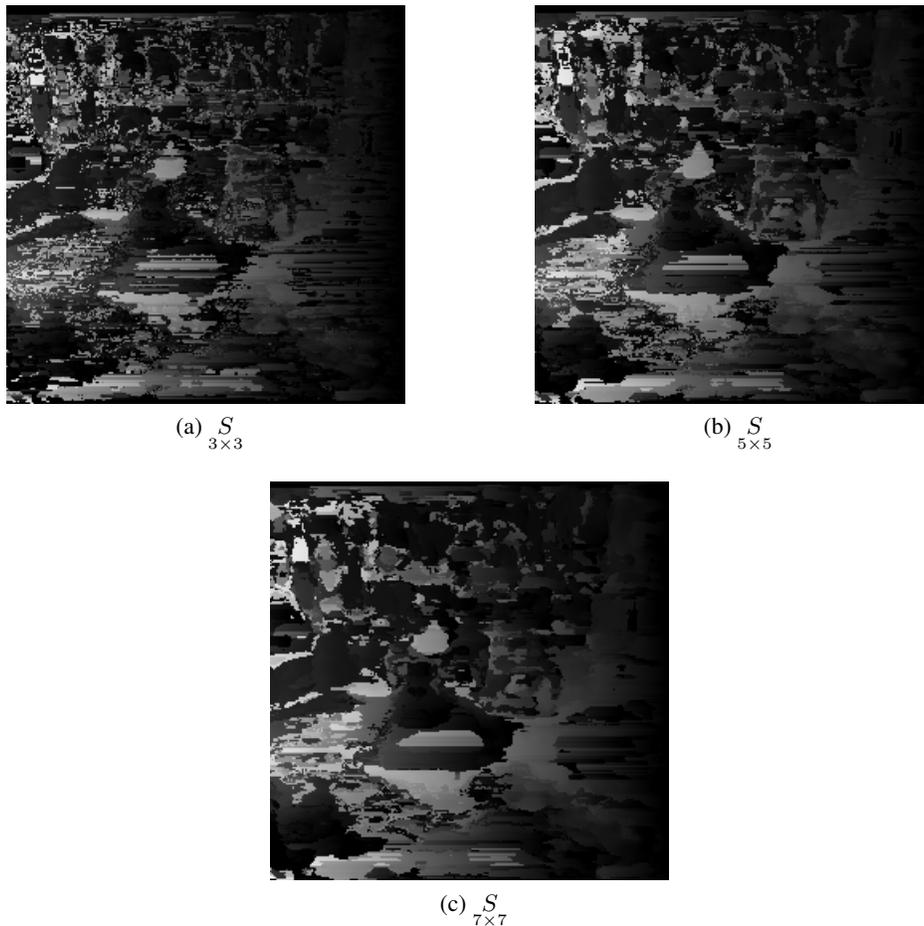


Fig. 6.20: Mapas de disparidad para distintas ventanas de análisis sobre el par “Habitación”, con umbral $t_h = 8$



(a) $S_{3 \times 3}$



(b) $S_{5 \times 5}$



(c) $S_{7 \times 7}$

Fig. 6.21: Mapas de disparidad para distintas ventanas de análisis sobre el par “Habitación”, con umbral $t_h = 16$

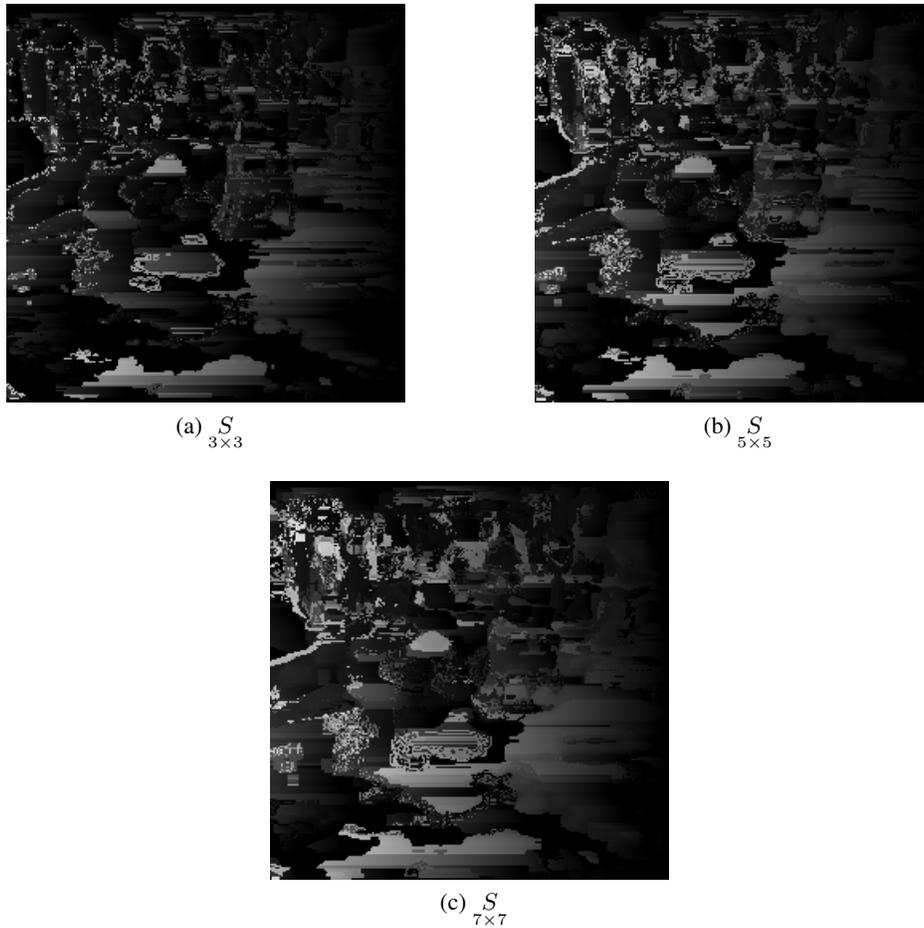


Fig. 6.22: Mapas de disparidad para distintas ventanas de análisis sobre el par “Habitación”, con umbral $t_h = 32$

6.4. Estimación de medidas del mundo real (triangulación)

El objetivo, una vez determinada la correspondencia entre píxeles, es obtener las coordenadas del punto espacial que fue proyectado en los píxeles en cuestión. Tomando en cuenta el proceso de triangulación presentado en la Sec. 5.4, particularizando para el arreglo estereoscópico canónico¹, y trabajando con imágenes rectificadas, se tiene una configuración como la mostrada en la Fig. 6.23.

¹Dos cámaras posicionadas de forma tal que sus ejes focales sean paralelos.

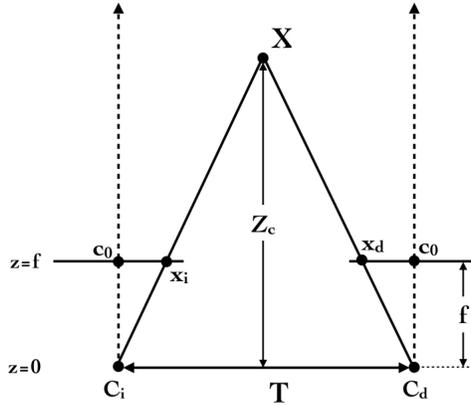


Fig. 6.23: Geometría del arreglo estereoscópico canónico

Por triángulos semejantes, de la Fig. 6.23 se tiene que:

$$\frac{T}{Z_c} = \frac{L}{H}. \quad (6.5)$$

Midiendo los píxeles en correspondencia $\mathbf{x}_i = (x_i, y_i)$ y $\mathbf{x}_d = (x_d, y_d)$ de la imagen izquierda y derecha respectivamente, desde el marco de la representación matricial, se tiene:

$$\frac{T}{Z_c} = \frac{T - (c_0 - x_d) - (x_i - c_0)}{Z_c - f}, \quad (6.6)$$

donde f es la distancia focal de las cámaras, y c_0 es la coordenada x de la intersección del eje focal con la imagen. Despejando Z_c , y tomando en cuenta la definición de disparidad, $d = x_i - x_d$,

$$Z_c = f \cdot \frac{T}{d}, \quad (6.7)$$

se obtiene la distancia Z_c perpendicular al plano de la imagen del punto espacial bajo estudio. Sin embargo, existe un conflicto dimensional pues f está en unidades métricas mientras que la disparidad (d) está en píxeles. Por tanto es necesario introducir un factor (m_x) de conversión de unidades métricas a píxeles, de la misma forma en que se presentó en la Ec. 5.70 $\alpha_x = m_x \cdot f$. Finalmente, la distancia que separa el sistema de cámaras del punto $\mathbf{X}_c = (X_c, Y_c, Z_c)$ es:

$$Z_c = \alpha_x \cdot \frac{T}{d}. \quad (6.8)$$

Con base en el modelo pinhole, en la información de la profundidad, y en los parámetros intrínsecos de la cámara, es posible calcular las coordenadas restantes del punto \mathbf{X}_c y concretar la triangulación:

$$Y_c = Z_c \cdot \frac{\tilde{y}_i}{\alpha_y}, \quad (6.9)$$

$$X_c = Z_c \cdot \frac{\tilde{x}_i}{\alpha_x}, \quad (6.10)$$

donde $\alpha_y = m_y \cdot f$, tal como se definió en la Ec. 5.71, y $(\tilde{x}_i, \tilde{y}_i)$ están expresados en el marco de referencia retinal.

La formulación para la triangulación presentada en esta sección es una simplificación para valores conocidos de la presentada en la Sec. 5.4, imponiendo algunas restricciones, que son:

- Distancia focal f igual para ambas cámaras.
- Misma orientación de los marcos de referencia de ambas cámaras.
- El vector que relaciona los centros de proyección de las cámaras solo tiene componente x , el resto son iguales a cero.
- Los centros de ambos sensores son iguales.

Estos datos pueden ser confirmados mediante el proceso de calibración.

Saber y saberlo demostrar, es saber dos veces
[BALTASAR GRACIÁN]

7.1. Conclusiones

El presente trabajo ha sido desarrollado atacando particularmente cada uno de los bloques fundamentales que componen el sistema robótico. En cada uno de los subsistemas se han trabajado las ideas necesarias para el desarrollo, generando los resultados que dan forma al proyecto en su totalidad. El diseño final ha logrado crear una amalgama de tecnologías nuevas y clásicas que representan una respuesta mecatrónica para contribuir en la solución de un problema de carácter medioambiental, que incide no solo en la mejora de la calidad de vida del ser humano, sino también a mejorar la salud del medio ambiente.

Una herramienta automática capaz de afrontar la tarea de ajuste de la superficie del concentrador solar, superando las limitaciones e inconvenientes que presentaban las anteriores metodologías de ajuste propuestas, fue diseñado, y sus componentes fundamentales desarrollados.

Se ideó una estructura mecánica que aprovecha la geometría del concentrador sobre el que trabajará. En particular, el diseño de las correderas de la placa móvil está orientado a disminuir la precisión necesaria en el maquinado de las piezas, utilizando una corredera común en una de las guías, y una corredera que solamente restringe el giro alrededor de la guía, pero con la capacidad de adaptarse a pequeñas irregularidades en el trayecto.

El engranaje que acopla el vástago de soporte con el primer actuador incrementa en la salida el torque entregado por este, esto se traduce en la posibilidad de utilizar un motor de menor capacidad que el que debería ser usado en ausencia del engranaje.

Aplicando los conceptos de la teoría de análisis y diseño de mecanismos, se planteó el sistema de ecuaciones que describen el funcionamiento y la geometría base de un mecanismo plegable para realizar la tarea de posicionamiento del motor de ajuste de tuercas. Por medio de un método de optimización numérica las ecuaciones fueron resueltas. Un mecanismo fue elegido y sobre el se llevó a cabo mejoras. El diseño final del mecanismo se aligeró un 30 % retirando material de sus barras, lo que disminuye el torque requerido para su movimiento, y ofrece la posibilidad de operarlo a mayor velocidad. Al haber sido diseñado con un solo GDL, se facilita el control del mismo. La geometría final de las barras permite trabajar sin interferencias y mantener el motor de ajuste en la posición deseada, desplegándose desde su posición inicial un total de 14 *cm* de forma segura, y hasta 16 *cm* para llegar a su límite de operación.

En el caso de la pluma, una vez es lograda la configuración correcta para todos los elementos que en ella se encuentran, se llevó a cabo un proceso de optimización de masa, reduciéndola en un 29.16 % sin afectar su comportamiento mecánico de forma considerable, y abriendo la posibilidad de operar el sistema a mayor velocidad, pues la disminución de masa causa directamente un decremento de los efectos dinámicos.

Se diseñó una API de comunicación entre el subsistema electrónico y el subsistema de visión computacional. Se cuenta así con el conjunto completo de instrucciones para llevar a cabo la manipulación de las funciones que el subsistema en cuestión puede desarrollar, además de poder incluir nuevas funciones sin necesidad de hacer modificaciones físicas en los dispositivos, evitando así imponer limitaciones al desarrollo futuro de dispositivo.

Con respecto al hardware del subsistema de visión, distintos modelos de cámara con interfaz USB fueron probados, y se eligió aquel con mejores prestaciones y relación costo beneficio adecuada. La cámara elegida es la PS3 EYE de Sony. Un par de cámaras se extrajeron de su carcasa protectora y se modificó la disposición de algunos de sus componentes para lograr el arreglo estereoscópico requerido. Bajo el criterio de frecuencia natural de vibración se diseñó un contenedor rígido para albergar el sistema estereoscópico construido, que facilita el montaje y manipulación de las cámaras sin tener contacto directo con estas, preservando así la calibración efectuada.

El diseño del contenedor estereoscópico permite llevarlo y adaptarlo de forma sencilla a un gran número de aplicaciones futuras. Se logra así comenzar con el desarrollo de una plataforma (hardware y software) especializada de visión estereoscópica.

La tarea del subsistema de visión computacional dentro del sistema de ajuste numerosos procesos, para cada uno de los cuales fue necesario desarrollar el código particular, haciendo uso de bibliotecas de manejo de imágenes, captura de video, desarrollos de interfaces gráficas, y de funciones matemáticas para la ejecución de las tareas. Los programas compilados son totalmente operativos y se ejecutan sobre una plataforma GNU/Linux. Su funcionamiento se basa en la interrelación de los módulos descritos en el contenido de este trabajo, los cuales son:

1. Identificación de marcas de color en la imagen, binarizando y segmentando del canal *S* del modelo HSI de la imagen captada.

2. Biblioteca de transformación de modelos de color. Se desarrollaron nuevos algoritmos para acelerar la transformación entre los modelos de color requeridos por el sistema, logrando una disminución media del 83 % del tiempo con respecto a las formulaciones presentadas en la literatura, así mismo, se obtienen como salida imágenes de composición más uniforme.
3. Determinación de posición tridimensional de las marcas por medio de optimización numérica.
4. Generación de mapas de disparidad. Se propusieron dos aproximaciones para la búsqueda de correspondencias entre imágenes estereoscópicas. El código de ambos algoritmos fue escrito pensando en la potencial paralelización futura de los procesos.
5. Cálculo de histogramas. Para la implementación de correspondencia basado en histogramas fue necesario deducir expresiones para el cálculo de histogramas en subimágenes, que redujeran el número de operaciones necesarias para su determinación bajo el criterio de repetitividad de bordes.
6. Calibración de cámara. Se desarrollaron las aplicaciones necesarias para automatizar el proceso de obtención, mediante calibración, de los parámetros intrínsecos de las cámaras, así como los parámetros extrínsecos del sistema estereoscópico.

El total del software desarrollado se condensó en bibliotecas que facilitan su uso particular y genérico, suponiendo el inicio del desarrollo de una librería de funciones para la implementación de aplicaciones en visión multicámara. Con respecto a la operación del sistema de visión, es conveniente remarcar que si bien el modelo de color HSI utilizado en la detección de marcas pretende ser invariante a los cambios de iluminación, estos le afectan en cierta medida, y aunque el software busca ganar robustez en este sentido, su operación se facilita y beneficia de un ambiente de trabajo bien iluminado.

Es importante hacer notar que cada uno de los módulos desarrollados puede ser utilizado de manera independiente para tareas distintas a la planteada sin requerir grandes modificaciones, es decir, no solo se consigue un sistema modular, sino, módulos con un campo de aplicación más amplio que el del propio sistema, reafirmando la esencia mecatrónica del desarrollo.

La tendencia actual en el campo tecnológico define un futuro con gran influencia de procesos automatizados y agentes robóticos en los ambientes y tareas más cotidianos. La capacidad del sentido de la vista para captar información de calidad en gran volumen marca una clara pauta a la adopción de su equivalente artificial en las tecnologías venideras. La fusión de la robótica con la visión y la inteligencia artificial, potenciarán el crecimiento de los campos de aplicación, los desarrollos, y la velocidad con que estos puedan ser adoptados. El presente proyecto supone una incursión en dicho campo, sentando la base de conocimiento para el desarrollo de proyectos futuros en esta línea.

Un aspecto importante en el entendimiento del desarrollo de una solución, en la que interactúan múltiples fuentes de conocimiento y experiencia, es la adquisición de una base teórica sólida. Es solo vía el entendimiento de la causa de los hechos y los modelos que los representan que se vuelve posible

operarlos y manipularlos con el fin de obtener el mejor resultado. En otras palabras, cualquier implementación sólida que busque consolidarse en su campo de aplicación debe estar cimentada en la teoría que le proporciona validez y coherencia.

7.2. Trabajo futuro

Con base en el desarrollo del sistema alcanzado con el cumplimiento de los objetivos propuestos, es posible plantear el trabajo futuro. En primer lugar, se encuentra la construcción del subsistema mecánico completo, es decir, el maquinado de las piezas correspondientes a la pluma y al mecanismo de ajuste, así como el ensamble de las mismas, y el acoplamiento de los actuadores. Posteriormente se deben realizar las pruebas de ajuste de superficie y la calibración sobre la estructura del concentrador de que ya se dispone.

Para el caso del subsistema electrónico, se debe materializar la PCB que alberga los controladores y el μ Controlador, y llevar a cabo su montaje, pasando así de la etapa de prototipo cableado a la de estado implementable.

De forma complementaria al desarrollo de software realizado, es necesario generar los manuales de referencia para el uso de las bibliotecas de calibración, color, y mapas de disparidad, contribuyendo así en el desarrollo de un paquete especializado en visión estereoscópica. En este mismo sentido, es importante buscar la paralelización de los algoritmos diseñados, aumentando la eficiencia del proceso y haciendo posible la implementación de algoritmos más avanzados, sin estar limitados a un solo flujo de trabajo.

A largo plazo se propone el balanceo del mecanismo de ajuste y la modificación física de las cámaras que conforman el par estereoscópico, con el fin de obtener un sistema de visión más ligero y compacto. Con este mismo enfoque, se debe buscar la forma de llevar el subsistema electrónico al montaje superficial, de forma económica y práctica.

- [1] Atmel Corporation. Official website. <http://www2.atmel.com>.
- [2] T. Baidyk and E. Kussul. *Redes neuronales, visión computacional y micromecánica*. ITACA-UNAM, México, 2009.
- [3] W. Bolton. *Mecatrónica*. Alfaomega, México, 3rd edition, 2006.
- [4] F. Bonin-Font, A. Ortiz, and G. Oliver. Visual navigation for mobile robots: A survey. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 53(3):263–296, 2008.
- [5] G. Bradski and A. Kaebler. *Learning OpenCV. Computer vision with the OpenCV Library*. Oreilly, USA, 2008.
- [6] T. Brosnan and D. Sun. Inspection and grading of agricultural and food products by computer vision systems. a review. *Computers and Electronics in Agriculture*, 36(2006):193–213, 2006.
- [7] A. Chalechale, A. Bahri, and M. Vatanchian. Vision-based bone image recognition using geometric properties. *Iranian Journal of Science and Technology*, 34(6):597–604, 2010.
- [8] C. Chang and Y. Ji. Flexible videogrammetric technique for three-dimensional structural vibration measurement. *Journal of Engineering Mechanics*, 133(6):656–664, 2007.
- [9] B. Chaouchi, A. Zrelli, and S. Gabsi. Desalination of brackish water by means of a parabolic solar concentrator. *Desalination*, 217(2007):118–216, 2007.
- [10] H. Cheng, M. Xue, and X. Shi. Contrast enhancement based on a novel homogeneity measurement. *Pattern Recognition*, 36(2003):2687–2697, 2003.
- [11] B. Coelho, A.C. Oliveira, and A. Mendes. Concentrated solar power for renewable energy and hydrogen production from water. a review. *Energy and Environmental Science*, 3(2010):1398–1405, 2010.

- [12] Committee on Energy Futures and Air Pollution in Urban China and the United States, National academy of engineering academy of engineering, National research council research council, Chinese academy of engineering academy of engineering, and Chinese academy of sciences academy of sciences. *Energy futures and urban air pollution. Challenges for China and the United States*. National academies press, USA, 2008.
- [13] B. Cyganek and J. Siebert. *An introduction to 3D computer vision, Techniques and algorithms*. John Wiley & Sons, UK, 2009.
- [14] Dassault Systemes. *SolidWorks Simulation Help*, 2010.
- [15] E. Davies. *Machine vision. Theory, algorithms, practicalities*. Morgan Kauffmann, USA, 3rd edition, 2005.
- [16] G. Davis, D. Griggs, and G. Sullivan. Automatic estimation of cloud amount using computer vision. *Journal of Atmospheric & Oceanic Technology*, 9(1):81–85, 1992.
- [17] U. Dhond and J.K. Aggarwal. Structure from stereo - a review. *IEEE Transactions on Systems, Man and Cybernetics*, 19(6):1489–1510, 1989.
- [18] J.A. Duffie and W.A. Beckman. *Solar engineering of thermal processes*. John Wiley & Sons, USA, 3rd edition, 2006.
- [19] Energy Information Administration. *Official Energy Statistics from the U.S Government, Solar thermal*, 2008.
- [20] European renewable energy council (EREC). *Renewable energy in Europe. Markets, trends and technologies*. Earthscan, UK, 2010.
- [21] D. Fadare. Modelling of solar energy potential in nigeria using an artificial neural network model. *Applied Energy*, 86(9):1410–1422, 2009.
- [22] Fairchild semiconductor. *MOC70 phototransistor optical interrupter switch Data Sheet*, 2001.
- [23] O. Faugeras and Q. Luong. *The Geometry of Multiple Images*. MIT Press, USA, 2001.
- [24] D. Feuermann and J. Gordon. Analysis of a two-stage linear fresnel reflector solar concentrator. *Solar Energy Engineering*, 113(4), 272-279 1991.
- [25] N. Finch, P. Murray, M. Dunn, and J. Billingsley. Using machine vision classification to control access of animals to water. *Australian Journal of Experimental Agriculture*, 46(2006):837–839, 2006.
- [26] D. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, USA, 2002.
- [27] R. Foster, M. Ghassemi, and A. Cota. *Solar energy. Renewable energy and the environment*. CRC Press, USA, 2010.

- [28] J. Franco, L. Saravia, V. Javi, R. Caso, and C. Fernandez. Pasteurization of goat milk using a low cost solar concentrator. *Solar Energy*, 82(2008):1088–1094, 2008.
- [29] C. Fraser. Innovations in automation for vision metrology systems. *Photogrammetric Record*, 15(90):901–911, 1997.
- [30] Freescale Semiconductor. Official website. <http://www.freescale.com>.
- [31] V. Fthenakis, J. Mason, and K. Zweibel. The technical, geographical, and economic feasibility for solar energy to supply the energy needs of the us. *Energy Policy*, 37(2):387–399, 2009.
- [32] E. García Breijo. *Compilador C CCS y simulador PROTEUS para Microcontroladores PIC*. Alfaomega, México, 2008.
- [33] German Solar Energy Society (DGS). *Planning & installing solar thermal systems: A Guide for Installers, Architects and Engineers*. Earthscan, UK, 2nd edition, 2010.
- [34] R. Gonzalez and R. Woods. *Digital Image Processing*. Prentice Hall, USA, 2nd edition, 2001.
- [35] A. Gopalakrishnan, S. Greene, and A. Sekmen. Vision-based mobile robot learning and navigation. In *14th IEEE Workshop on Robot and Human Interactive Communication*, pages 48–53. IEEE, August 13 - August 15 2005. Nashville, TN, United states.
- [36] M. Graves and B. Batchelor, editors. *Machine Vision for the Inspection of Natural Products*. Springer-Verlag, UK, 2004.
- [37] S. Han, Q. Zhang, B. Ni, and J. Reid. A guidance directrix approach to vision-based vehicle guidance systems. *Computers and Electronics in Agriculture*, 43(2004):179–195, 2004.
- [38] R. Hartley and A. Zisserman. *Multiple View Geometry in computer vision*. Cambridge University Press, UK, 2nd edition, 2003.
- [39] H. Hasuike, Y. Yoshizawa, A. Suzuki, and Y. Tamaura. Study on design of molten salt solar receivers for beam-down solar concentrator. *Solar Energy*, 80(2006):1255–1262, 2006.
- [40] J. Heikkilä. Geometric camera calibration using circular control points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1066–1077, 2000.
- [41] I. Howard and B. Rogers. *Binocular Vision and Stereopsis*. Oxford University Press, USA, 1995.
- [42] S. Kalogirou. Solar thermal collectors and applications. *Progress in Energy and Combustion Science*, 3(30):231–295, 2004.
- [43] S. Kalogirou. *Solar energy engineering. Processes and systems*. Academic press, USA, 2009.

- [44] E. Kussul and T. Baidyk. Thermal motor for solar power plants. In *3er Congreso Internacional de Ciencias, Tecnología, Artes y Humanidades*, pages 684–688, Jun 3 - 6 2009. Portoroz (Portotose), Slovenia.
- [45] E. Kussul, T. Baidyk, F. Lara-Rosano, J. M. Saniger, and N. Bruce. Desarrollo del concentrador solar con micro espejos. In *XXII Congreso de Instrumentación SOMI*, pages 1–5, Sep 30 - Oct 4 2007. Portoroz (Portotose), Slovenia.
- [46] E. Kussul, T. Baidyk, F. Lara-Rosano, J. M. Saniger, and N. Bruce. Support frame for micro facet solar concentrator. In *The 2-nd IASME/WSEAS International Conference on Energy and Environment (EE07)*, pages 300–304, May 15 - 17 2007. Portoroz (Portotose), Slovenia.
- [47] E. Kussul, T. Baidyk, F. Lara-Rosano, J. M. Saniger, and N. Bruce. Micro-facet solar concentrator. *International Journal of Sustainable Energy*, 27(2):61–71, 2008.
- [48] E. Kussul, T. Baidyk, F. Lara-Rosano, J. M. Saniger, and N. Bruce. Prototipo del concentrador solar con microespejos. In *XXIII Congreso de Instrumentación SOMI*, pages 1–5, Oct 1 - 3 2008. Portoroz (Portotose), Slovenia.
- [49] E. Kussul, T. Baidyk, O. Makeyev, F. Lara-Rosano, J. M. Saniger, and N. Bruce. Development of micro mirror solar concentrator. In *2-nd IASME/WSEAS International Conference on Energy and Environment (EE07)*, pages 294–299, May 15 - 17 2007. Portoroz (Portotose), Slovenia.
- [50] E. Kussul, T. Baidyk, O. Makeyev, F. Lara-Rosano, J. M. Saniger, and N. Bruce. Development of micro mirror solar concentrator. *WSEAS Transactions on Power Systems*, 2(8):188–194, 2007. Portoroz (Portotose), Slovenia.
- [51] E. Kussul, T. Baidyk, O. Makeyev, F. Lara-Rosano, J. M. Saniger, and N. Bruce. Flat facet parabolic solar concentrator. In *2nd WSEAS/IASME International Conference on Renewable Energy Sources (RES08)*, pages 46–51, Oct 26 - 28 2008. Portoroz (Portotose), Slovenia.
- [52] E. Kussul, T. Baidyk, O. Makeyev, F. Lara-Rosano, J. M. Saniger, and N. Bruce. Flat facet parabolic solar concentrator with support cell for one and more mirrors. *WSEAS Transactions on Power Systems*, 3(8):577–586, 2008. Portoroz (Portotose), Slovenia.
- [53] E. Kussul, T. Baidyk, and D. Wunsch. *Neural Networks and Micro Mechanics*. Springer-Verlag, USA, 2010.
- [54] J. Lee, D. Kim, and J. Choi. A study on matching for stereo image using disparity space image. *HC12004*, 2:542–550, 2004.
- [55] V. Leemans and M. Destain. A computer-vision based precision seed drill guidance assistance. *Computers and Electronics in Agriculture*, 59(2007):1–12, 2007.
- [56] N. Lenssen. Sustainable energy for tomorrow’s world the case for an optimistic view of the future. *Energy Policy*, 24(9):769–781, 1996.

- [57] T. Lin, Y. Hsiung, G. Hong, H. Chang, and F. Lu. Development of a virtual reality gis using stereo vision. *Computers and Electronics in Agriculture*, 63(2008):38–48, 2008.
- [58] A. Maczulak. *Renewable energy. Sources and methods*. Facts on file, USA, 2010.
- [59] H. Mayer. Object extraction in photogrammetric computer vision. *Journal of Photogrammetry & Remote Sensing*, 63(2008):213–222, 2008.
- [60] S. Mehta, T. Burks, and W. Dixon. Vision-based localization of a wheeled mobile robot for greenhouse applications: A daisy-chaining approach. *Computers and Electronics in Agriculture*, 63(2008):28–37, 2008.
- [61] Microchip Technology Incorporated. Official website. <http://www.microchip.com>.
- [62] Microchip Technology Incorporated. *PIC18F2455/2550/4455/4550 Data Sheet*, 2009.
- [63] Microchip Technology Incorporated. *PIC18F2458/2553/4458/4553 Data Sheet*, 2009.
- [64] Microchip Technology Incorporated. *Flash Microcontroller Programming Specification*, 2010.
- [65] Microchip Technology Incorporated. *Focus Product Selector Guide*, fourth quarter 2010 edition, 2010.
- [66] National academy of sciences, National academy of engineering academy of engineering, and National research council research council. *Real Prospects for Energy Efficiency in the United States*. National Academies Press, USA, 2010.
- [67] P Naukkarinen. Solar air conditioning and its role in alleviating the energy crisis of the mediterranean hotels. *International Journal of Sustainable Energy*, 28(1):93–100, 2009.
- [68] J. D. Nixon, P. K. Dey, and P. A. Davies. Which is the best solar thermal collection technology for electricity generation in north-west india? evaluation of options using the analytical hierarchy proces. *Energy*, 35(2010):5230–5240, 2010.
- [69] J. Nocedal and S. Wrioth. *Numerical Optimization*. Springer, USA, 1999.
- [70] OpenCV. Open source computer vision library. <http://opencv.willowgarage.com/wiki>.
- [71] G. Pajares-Martinsanz and J. de la Cruz-Gracia. *Visión por Computador. Imágenes digitales y aplicaciones*. Alfaomega Ra-Ma, México, 2nd edition, 2008.
- [72] E. Palacios, F. Remiro, and L. J. López. *Microcontrolador PIC16F84. Desarrollo de proyectos*. Alfaomega Ra-Ma, México, 3rd edition, 2009.
- [73] M. Potsch, G. Pischinger, R. Bewick, and A. Gaich. Geotechnical data collection and analysis in jointed rock: Application of computer vision technology and metric 3d images. *Felsbau*, 25(5):66–73, 2007.

- [74] J. Ramirez, P. Gómez-Gil, and F. Larios. A robot-vision system for autonomous vehicle navigation with fuzzy-logic control using lab-view. In *Electronics, Robotics and Automotive Mechanics Conference, CERMA 2007 - Proceedings*, pages 295–300. IEEE, September 25 - September 28 2007. Cuernavaca, Morelos, Mexico.
- [75] Alvy Ray Smith. Dr. alvy ray smith website. <http://www.alvyray.com>.
- [76] J. Rodriguez-Mendoza, T. Baydyk, and E. Kussul. Stereoscopic image recognition with neural networks. In *1st International congress on instrumentation and applied sciences*, pages 1–12, Oct 26 - 29 2010.
- [77] F. Rovira-Más, Q. Zhang, and J. Reid. Stereo vision three-dimensional terrain maps for precision agriculture. *Computers and Electronics in Agriculture*, 60(2008):133–143, 2008.
- [78] Z. Sen. *Solar energy fundamentals and modeling techniques. Atmosphere, environment, climate change and renewable energy*. Springer, USA, 2008.
- [79] SGS-Thomson Microelectronis. Official website. <http://www.st.com>.
- [80] SGS-Thomson Microelectronis. *UNL2803 eight Darlington arrays Data Sheet*, 1997.
- [81] SGS-Thomson Microelectronis. *L297 Stepper motor controller Data Sheet*, 2001.
- [82] F. Smeraldi and J. Bigun. Retinal vision applied to facial features detection and face authentication. *Pattern Recognition Letters*, 23(2002):463–475, 2002.
- [83] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. Thomson, USA, 2008.
- [84] Sony Corporation. Ps3 eye camera website. <http://us.playstation.com/ps3/accessories/playstation-eye-camera-ps3.html>.
- [85] B. Sorensen. *Renewable energy. Conversion, transmission and storage*. Academic press, USA, 2007.
- [86] B. Sorensen, P. Breeze, T. Storvick, and S. Yang. *Renewable energy focus handbook*. Academic press, USA, 2009.
- [87] A. Sowmya and J. Trinder. Modelling and representation issues in automated feature extraction from aerial and satellite images. *Journal of Photogrammetry & Remote Sensing*, 55(2000):34–47, 2000.
- [88] A Sozen, E Arcakliolu, M Ozalp, and E Kanit. Solar-energy potential in turkey. *Applied Energy*, 80(4):367–381, 2005.
- [89] K. Stefanik, J. Gassaway, K. Kochersberger, and A. Abbott. Uav-based stereo vision for rapid aerial terrain mapping. *GIScience & Remote Sensing*, 48(1):24–49, 2011.

- [90] R. Sumantra, C. Santanu, and B. Subhashis. Active recognition through next view planning: a survey. *Pattern Recognition*, 37(2004):429–446, 2004.
- [91] S. Taggart. Csp: dish projects inch forward. *Renewable Energy Focus*, 4(9):52–54, 2008.
- [92] Texas Instruments. Official website. <http://www.ti.com>.
- [93] Y. Tsai, Z. Hu, and Z. Wang. Vision-based roadway geometry computation. *Journal of Transportation Engineering*, 136(3):223–233, 2010.
- [94] Carnegie Mellon University. Vision and autonomous systems center image database. <http://www.ius.cs.cmu.edu/idb>.
- [95] U.S. Census Bureau. *International Data Base*, 2011. <http://www.census.gov/idb/ranks.html>.
- [96] J. Veléz-Serrano, A. Moreno-Dáz, A. Sánchez-Calle, and J. Esteban-Sánchez-Marín. *Visión por computador*. Ediciones UPC, España, 2003.
- [97] D. Vernon. *Machine Vision. Automated visual inspection and robot vision*. Prentice Hall, UK, 1991.
- [98] Vigor Precision Ltd. *VS-2A Servo Data Sheet*, 2009.
- [99] D. White, C. Svellingen, and N. Strachan. Automated measurement of species and length of fish by computer vision. *Fisheries Research*, 80(2006):203–210, 2006.
- [100] C. Wilson and J. Sadler. *Kinematics and Dynamics of Machinery*. Prentice Hall, USA, 3rd edition, 2002.
- [101] World Energy Council. *2010 Survey of energy resources*, 2010.
- [102] D. Yuan and R. Chung. Correspondence-free stereo vision for the case of arbitrarily-positioned cameras. In *2003 IEEE International Conference on Robotics and Automation*, pages 1688–1693. IEEE, September 14 - September 19 2003. Taipei, Taiwan.
- [103] C. Yud-Ren, C. Kuanglin, and S. Moon. Machine vision technology for agricultural applications. *Computers and Electronics in Agriculture*, 36(2002):173–191, 2002.
- [104] Q. Zhou, D. Parrott, M. Gillen, D. Chelberg, and L. Welch. Agent-based computer vision in a dynamic, real-time environment. *Pattern Recognition*, 37(2004):691–705, 2004.

El color es la manifestación visual de las diversas longitudes de onda de la luz comprendidas en la franja visible del espectro electromagnético. Se trata de un descriptor poderoso que a menudo simplifica la identificación de objetos en el medio [34], y representa una de las principales formas de discernir la información visual para el humano [83]. El número de colores que el humano puede detectar supera en dos órdenes de magnitud los diferentes tonos de gris que es capaz de diferenciar [34], lo que implica una expansión en la gama de identificadores disponibles para procesos de reconocimiento de objetos, tanto en escenas naturales como artificiales. Es por ello que el color juega un papel crucial en los procesos de visión computacional que involucran reconocimiento de escenas.

Al igual que el ojo humano, que cuenta con células especializadas para la detección de color [41, 96], las cámaras cuentan con un arreglo de sensores fotosensibles especializados en captar cada uno de los colores primarios [71], rojo, verde y azul, pues mediante la combinación en distintas proporciones de cada uno de ellos, es posible conseguir cualquier color del espectro visible.

De manera general, los dispositivos de captura de imágenes en color trabajan bajo dicha estructura, como consecuencia de la forma natural en que el humano capta el color. Para especificar la relación entre los colores del espectro visible y los colores primarios en este tipo de hardware, se ha desarrollado una representación matemática denominada modelo de color RGB¹.

El espacio de color RGB se representa como un cubo unitario en el espacio cartesiano tridimensional (Fig. A.1), en donde cada uno de sus ejes representa uno de los colores primarios. Este espacio está normalizado, es decir, el valor sobre cada uno de sus ejes está en el intervalo cerrado $[0, 1]$. Cada uno de los colores del espectro se representa mediante una triada (R, G, B) , que determina la contribución necesaria de cada uno de los colores primarios para generarlo [26]. En el modelo RGB, los colores primarios o puros se representan por medio de los vectores $(1, 0, 0)$ para el color rojo, $(0, 1, 0)$ para el verde, y $(0, 0, 1)$ para el azul. El color negro está en el origen $(0, 0, 0)$, mientras que el blanco se

¹*Red, Green, Blue*

forma por la adición de los tres colores puros en proporción máxima (1, 1, 1). La escala de grises se representa por medio de la línea que une a los puntos negro y blanco.

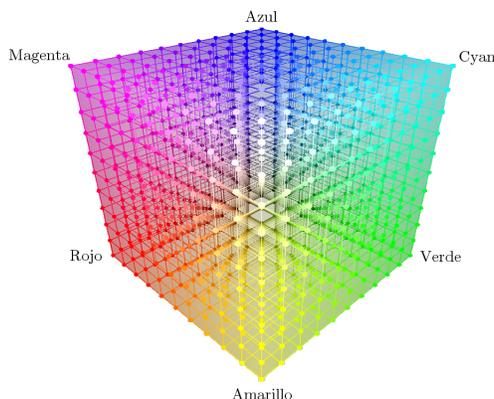


Fig. A.1: Modelo de color RGB

Aunque el modelo RGB resulta conveniente desde el punto de vista de la adquisición del color del medio, puede resultar complicado llevar a cabo la interpretación del color real bajo esta forma de representación [36], pues no es natural para el humano describir un color en términos de cada una de las componentes RGB que lo conforman, sino en términos del tono, la saturación y el brillo que lo caracterizan.

Un gran número de modelos (CMY, YIQ, CMYK, HSI, HSV, CIE XYZ, CIE Luv, CIE LAB, por ejemplo) han sido especialmente desarrollados para diferentes aplicaciones [26, 71], de los cuales, el modelo HSI² es el más cercano a la manera en que el humano describe e interpreta el color, convirtiéndose en una herramienta ideal para el desarrollo de algoritmos de procesamiento de imagen basados en la descripción de color [34].

El modelo HSI representa cada color mediante una triada de números (H, S, I) que corresponden al tono, la saturación y la intensidad del mismo. El tono o matiz es el atributo asociado con la longitud de onda dominante de entre las longitudes de onda que componen un color, se puede entender como “el color puro” percibido por el observador. La saturación mide la proporción entre la longitud de onda dominante y el resto de longitudes de onda [96], es el grado de pureza relativa del color, es decir, la cantidad de luz blanca mezclada con el tono. Mientras que la intensidad incorpora la noción acromática de la luz, el paso del blanco al negro, y es uno de los factores clave en la descripción de la sensación del color.

Para los efectos de reconocimiento de patrones es conveniente considerar que el color tiene una topología circular, en el sentido de que el tono cambia del rojo al naranja, del naranja al amarillo, posteriormente al verde, seguido del cian, el azul, el púrpura, y por último al rojo nuevamente (Fig.

²Tono, saturación e intensidad (*Hue, Saturation, Intensity*)

A.2), dejando clara la necesidad de un modelo de color no lineal, pues el tono es una variable cíclica [26].

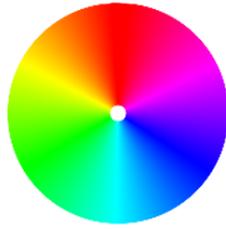


Fig. A.2: Círculo cromático

El modelo HSI logra dicha representación del color tomando el cubo unitario de RGB y fijando al origen el punto correspondiente al color negro. Posteriormente, lleva a la posición vertical el vector que representa la escala de grises, es decir, el vector que va de negro $(0, 0, 0)$ a blanco $(1, 1, 1)$, tal como se observa en la Fig. A.3. Este modelo se simplifica como dos conos unidos por su cara plana (Fig. A.4), en donde el color negro se encuentra en el vértice del cono inferior, localizado en el origen del sistema de coordenadas. Por otro lado, el color blanco se encuentra en el vértice opuesto. La línea que se forma al unir ambos vértices recibe el nombre de eje I, a lo largo de este se manifiesta la intensidad luminosa de cada combinación del espacio RGB.

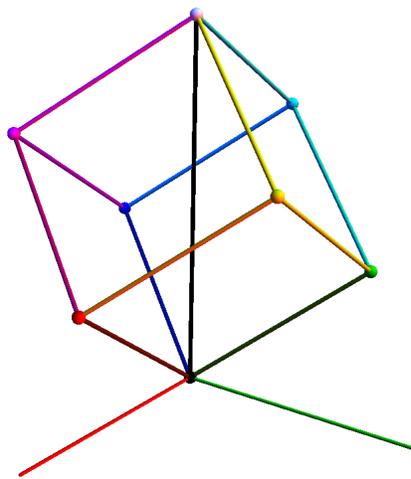


Fig. A.3: Rotación del cubo RGB

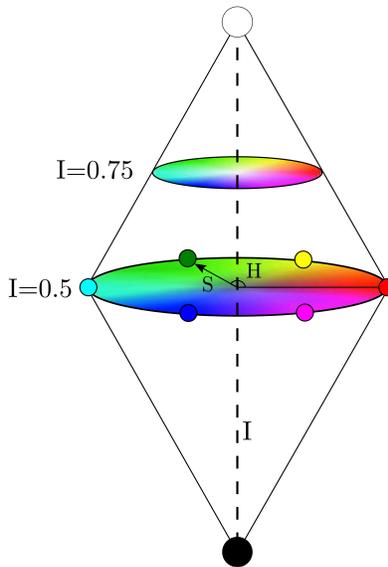


Fig. A.4: Modelo de color HSI

El intervalo de valores del tono (H) es $[0^\circ, 360^\circ]$, y denota la posición angular del tono dentro del círculo cromático, e.g., el tono rojo está localizado en 0° . El valor de la intensidad (I) se encuentra en el intervalo $[0, 1]$, donde 0 representa el color negro y 1 el color blanco. Los valores intermedios del intervalo corresponden a los tonos de gris asociados a las distintas combinaciones RGB, es decir, la cantidad de luz presente en cada combinación.

Para la saturación (S) el intervalo es $[0, 1]$, donde 0 representa la máxima dilución del tono con luz blanca, mientras que 1 indica un tono puro. Gráficamente S indica qué tanto se aleja la combinación RGB del eje I de forma perpendicular a este. Como se ha dicho, los colores blanco y negro se encuentran en los vértices de los conos, por lo tanto, un valor $S = 0$ no representa el blanco, sino el tono de gris que se puede alcanzar en el centro del círculo cromático, para el valor correspondiente sobre el eje I de cada una de las combinaciones RGB.

Para llevar a cabo la conversión del espacio RGB al espacio HSI la literatura [34, 71] propone las siguientes ecuaciones:

$$H = \begin{cases} \theta & B \leq G \\ 360 - \theta & B > G \end{cases}, \quad (\text{A.1})$$

$$\theta = \arccos \left[\frac{\frac{1}{2} [(R - G) + (R - B)]}{\sqrt{(R - G)^2 + (R - B)(G - B)}} \right], \quad (\text{A.2})$$

$$S = 1 - \frac{3}{R + G + B} \cdot \text{mín}(R, G, B), \quad (\text{A.3})$$

$$I = \frac{1}{3} (R + G + B), \quad (\text{A.4})$$

en donde (R, G, B) se encuentran en el intervalo $[0, 1]$. Cuando se trabaja con imágenes de 8 bits , los valores que adoptan los pixeles para cada uno de los canales RGB se encuentran en el intervalo $[0, 255]$. Para llevar dichos valores al intervalo $[0, 1]$ es necesario normalizarlos, dividiendo cada uno por 255, lo que a su vez implica un mayor número de operaciones para realizar la conversión.

En adelante se considera que las variables mayúsculas (R, G, B) se encuentran en el intervalo $[0, 1]$, mientras que las variables minúsculas (r, g, b) en el intervalo $[0, 255]$. Ambas representaciones mantienen equivalencia mediante:

$$(R, G, B) = \left(\frac{r}{255}, \frac{g}{255}, \frac{b}{255} \right). \quad (\text{A.5})$$

Para que los valores de HSI calculados puedan ser representados en una imagen digital, estos deben ajustarse al rango de valores que la imagen puede tomar, e.g., de 0 a 255 para imágenes de 8 bits de resolución.

Cuando la visión computacional es utilizada como una fuente de información en tiempo real, el tiempo que consume cada uno de los algoritmos en el procesamiento de la imagen juega un papel crucial. Motivo por el cual es importante tratar de reducir el tiempo de ejecución de las tareas, realizando mejoras en los algoritmos de cálculo.

Sin embargo, no solo el tiempo es importante, sino también la calidad de la imagen resultante, pues un proceso que arroje como salida, por ejemplo, una imagen con un nivel de ruido considerable, por rápida que sea esta, dificultará, o incluso, impedirá la labor de las tareas subsecuentes, dejando atrás las ventajas ganadas con el ahorro de tiempo.

La transformación de modelo de color en una imagen no es un proceso computacionalmente económico. Teniendo en cuenta los factores de tiempo y calidad, se ha decidido rediseñar los algoritmos de transformación del modelo RGB a HSI. Tomando como base la construcción teórica del modelo, la nueva formulación para cada uno de los canales es:

A.1. Saturación (S)

La base del desarrollo del modelo HSI es la rotación del cubo RGB dejando al eje I en posición vertical. Una vez logrado esto, para determinar S , es necesario obtener la proyección perpendicular al eje I del vector de la combinación RGB a convertir, tal como se observa en la Fig. A.5.

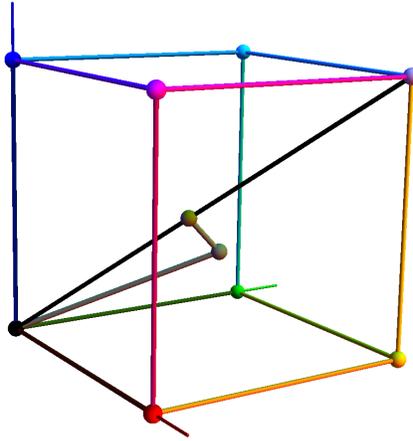


Fig. A.5: Proyección de color RGB sobre el eje I

Sin embargo, de forma algebraica no es necesario llevar a cabo la rotación para conocer el valor de la proyección deseada. Sea $C = \{r, g, b\}$ donde $r, g, b \in [0, 255]$, y sea el eje I, $I_n = \{1, 1, 1\}$, para poder proyectar el vector C sobre I_n es necesario conocer el ángulo que los separa. Así, mediante el producto punto:

$$\theta = \arccos \left[\frac{C \cdot I_n}{\|C\| \|I_n\|} \right]. \quad (\text{A.6})$$

Conocido θ , se calcula la proyección de C perpendicular a I_n como:

$$\begin{aligned} \|C\| \sin(\theta) &= \sqrt{b^2 + g^2 + r^2} \cdot \sqrt{1 - \frac{\frac{1}{3}(b+g+r)^2}{b^2 + g^2 + r^2}} \\ &= \sqrt{\frac{2}{3}} \cdot \sqrt{b^2 + g^2 + r^2} \cdot \sqrt{\frac{b^2 + g^2 + r^2 - gr - b(g+r)}{b^2 + g^2 + r^2}} \\ &= \sqrt{\frac{2}{3}} \cdot \sqrt{b^2 + g^2 + r^2 - gr - b(g+r)}, \end{aligned} \quad (\text{A.7})$$

Como se ha mencionado, S está en el intervalo $[0, 1]$ de forma natural, pues se trata de una proporción. Dicha proporción se obtiene dividiendo $\|C\| \sin(\theta)$ entre la proyección de una triada (r, g, b) , ubicada en el extremo del círculo cromático del modelo HSI, con intensidad $I = 0.5$, es decir, el máximo valor posible de $\|C\| \sin(\theta)$. Sea entonces $C_{lim} = \{255, 0, 0\}$ una triada RGB en el extremo, su proyección perpendicular al eje I es:

$$\theta_{lim} = \arccos \left[\frac{C_{lim} \cdot I_n}{\|C_{lim}\| \|I_n\|} \right], \quad (\text{A.8})$$

$$\|C_{lim}\| \sin(\theta_{lim}) = 255 \sqrt{\frac{2}{3}}. \quad (\text{A.9})$$

Calculando S en el intervalo $[0, 1]$ como el cociente entre Ec. A.7 y Ec. A.9:

$$S_{[0,1]} = \frac{\|C\| \sin(\theta)}{\|C_{lim}\| \sin(\theta_{lim})} = \frac{1}{255} \cdot \sqrt{b^2 + g^2 + r^2 - gr - b(g+r)}. \quad (\text{A.10})$$

Finalmente se lleva $S_{[0,1]}$ al rango dinámico de la imagen, multiplicándola por 255:

$$S = 255 \cdot S_{[0,1]} = \sqrt{b^2 + g^2 + r^2 - gr - b(g+r)}. \quad (\text{A.11})$$

Una vez que se obtiene la ecuación para calcular el valor de S , el objetivo es reducir el tiempo de cálculo, el cual está directamente relacionado con la cantidad de operaciones requeridas y la complejidad de las mismas. Tomando el radicando de la Ec. A.11 se observa que sus variables están relacionadas por dos potencias, cinco multiplicaciones y cinco sumas algebraicas. Reordenando el radicando se tiene que:

$$S = \sqrt{(r-g)(r-g) + (r-b)(g-b)}, \quad (\text{A.12})$$

con lo que las operaciones en el radicando disminuyen a dos multiplicaciones y cinco sumas algebraicas. Se logra entonces disminuir el número de operaciones necesarias para calcular S . La raíz cuadrada es una operación computacionalmente muy costosa, por lo que es conveniente buscar una alternativa para evitarla a modo de simplificar el cálculo de S .

Como solución se propone utilizar un arreglo en memoria que use como índice de entrada el valor arrojado por el radicando,

$$r_s = (r-g)(r-g) + (r-b)(g-b), \quad (\text{A.13})$$

y ofrezca como respuesta el valor de la raíz cuadrada, evitando así su cálculo directo. De manera adicional, ofrece como ventaja la posibilidad de modificar el tipo de redondeo que se desea aplicar al resultado de la raíz cuadrada, pues la salida que se almacena en la imagen debe ser un entero. El valor de S (Ec. A.12) ha sido ajustado al intervalo $[0, 255]$, por lo tanto el radicando r_s (Ec. A.13) estará en el intervalo $[0, 255^2]$, el cual determina el número de elementos contenidos en el arreglo de esta forma:

$$S = \text{SqrtArr}[(r-g)(r-g) + (r-b)(g-b)]. \quad (\text{A.14})$$

La Tabla A.1 muestra el tiempo de ejecución sobre cuatro diferentes procesadores de cada una de las tres formulaciones para el cálculo de S . La prueba consiste en calcular el valor de S para cada punto de RGB hasta barrer completamente el espacio de RGB con una resolución de 8 bits (256 valores por eje), es decir, se realizan $256^3 = 16,777,216$ conversiones por ciclo. Los tiempos que se presentan en la Tabla A.1 corresponden al promedio calculado con el resultado de cien ciclos de conversión. Se puede observar que la conversión basada en la Ec. A.14 ahorra, con respecto al cálculo convencional de S (Ec. A.3), alrededor de un 70 % de tiempo, aumentando dramáticamente la eficiencia del proceso. Se consigue así el objetivo de reducir la carga computacional que implica la transformación.

Tabla A.1: Comparación de algoritmos para cálculo de S

Formulaciones para calcular S	AMD PhenomII X4 810 2.6 GHz	Intel Core 2 Duo T5550 1.83 GHz	AMD PhenomII X4 945 3.0 GHz	Intel Xeon 2 Core 3.2 GHz
Convencional Ec. A.3	380.28 <i>ms</i>	381.67 <i>ms</i>	319.25 <i>ms</i>	366.56 <i>ms</i>
Con raíz cuadrada Ec. A.12	687.85 <i>ms</i>	843.79 <i>ms</i>	599.85 <i>ms</i>	786.52 <i>ms</i>
Con arreglo en memoria Ec. A.14	111.16 <i>ms</i>	117.47 <i>ms</i>	97.40 <i>ms</i>	100.33 <i>ms</i>
% de ahorro entre Ec. A.3 y Ec. A.14	70.77 %	69.22 %	69.49 %	72.63 %

El segundo punto importante que debe ser atacado es la calidad del resultado. El ideal es obtener una imagen con áreas uniformes en el nivel de saturación y cambios bien definidos entre estas, tal como el humano lo percibe. Sin embargo, existen desviaciones producto del ruido electrónico de los sensores, las variaciones de iluminación en el ambiente, y por el redondeo en el cálculo de S .

La Fig. A.6 muestra el resultado de la transformación al canal S operando la imagen de cuatro maneras distintas. La primera transformación (Fig. A.6b) se llevó a cabo con el método convencional (Ec. A.3). La Fig. A.6c muestra el resultado de aplicar el arreglo para el cálculo de la raíz cuadrada (Ec. A.14) utilizando redondeo hacia arriba, es decir, el valor de S será el entero más pequeño, mayor o igual al resultado de la raíz. En las dos imágenes restantes (Fig. A.6d y Fig. A.6e) se emplea nuevamente el arreglo, pero esta vez el redondeo es hacia abajo, y por el método común respectivamente. El redondeo hacia abajo arroja como salida el mayor entero, menor o igual al resultado de la raíz, mientras que el redondeo común ofrece el entero más cercano al valor de la raíz cuadrada.



(a) Imagen original



(b) Formulación convencional



(c) Arreglo con redondeo hacia arriba



(d) Arreglo con redondeo hacia abajo



(e) Arreglo con redondeo común

Fig. A.6: Transformación de imagen de RGB a S

Como se observa en la Fig. A.6, la imagen producto de aplicar la transformación convencional a S es sensiblemente más ruidosa en comparación con el resto de las imágenes. En las imágenes generadas utilizando el arreglo en memoria se aprecia una transición suave del nivel de saturación entre zonas

contiguas de un mismo cuerpo, y cambios bien definidos en el límite de los objetos. Éste último método presenta una menor sensibilidad a las sombras provocadas por la luz ambiental, cumpliendo así con los objetivos del modelo HSI. Además, no existe una diferencia apreciable entre las imágenes del canal S al utilizar distintos criterios para el redondeo. Con base en estos resultados y con el fin de evitar un sesgo estadístico, se decide trabajar con el método común de redondeo.

A.2. Intensidad (I)

El caso del canal I se puede manejar de manera semejante al del canal S . Sin embargo, esta vez el valor RGB por transformar será proyectado sobre el eje I . Así, sea $C = \{r, g, b\}$ donde $r, g, b \in [0, 255]$, el eje I , $I_n = \{1, 1, 1\}$, y tomando θ de la Ec. A.6, se calcula la proyección de C sobre I_n como:

$$\|C\| \cos(\theta) = \frac{b + g + r}{\sqrt{3}}. \quad (\text{A.15})$$

Se calcula ahora la proyección sobre el eje I del punto extremo RGB, que corresponde a $I = 1$. Sea entonces $C_{lim} = \{255, 255, 255\}$ una triada RGB en el extremo, su proyección sobre el eje I es, tomando la Ec. A.8:

$$\|C_{lim}\| \cos(\theta_{lim}) = 255\sqrt{3}. \quad (\text{A.16})$$

Calculando I en el intervalo $[0, 1]$ como el cociente entre Ec. A.15, y Ec. A.16 y multiplicando por 255 para ajustarla al intervalo de valores que puede adoptar la imagen:

$$I = \frac{\|C\| \cos(\theta)}{\|C_{lim}\| \cos(\theta_{lim})} = 255 \cdot \frac{\frac{r+g+b}{\sqrt{3}}}{255\sqrt{3}}, \quad (\text{A.17})$$

$$I = \frac{b + g + r}{3}. \quad (\text{A.18})$$

Esta ecuación es idéntica a la propuesta de manera convencional (Ec. A.4), por lo tanto no es necesario llevar a cabo comparaciones. La Fig. A.7 muestra la transformación de una imagen RGB al canal I del modelo HSI.

Fig. A.7: Transformación de imagen de RGB a I

A.3. Tono (H)

El canal H del modelo HSI proporciona información sobre el tono de los colores captados en las imágenes. La diversidad de tonos existentes se puede apreciar al recorrer el perímetro del círculo cromático. Si bien el humano es capaz de discernir miles de tonos de colores, no le es posible realizarlo de forma simultánea; en este caso, su capacidad desciende hasta las decenas de tonos [34]. Adicionalmente, de forma intuitiva el humano identifica los objetos de su entorno con solo algunos tonos básicos de color que facilitan la comunicación y el análisis del medio.

La conversión de modelo RGB a HSI juega un papel crucial en los algoritmos de detección e identificación de marcas que han sido diseñados, y que se presentan en el siguiente capítulo. Por lo tanto, disminuir el tiempo de cálculo requerido para la conversión de RGB a H es un factor importante en aplicaciones donde el tiempo debe ser tomado en cuenta. Con este fin, y en busca de lograr zonas de tono uniforme que faciliten la identificación de objetos, se decide discretizar el círculo cromático en doce zonas con respecto al continuo (Fig. A.2), que corresponden a los tonos primarios y secundarios, así como algunos tonos adyacentes e ellos (Fig. A.8).

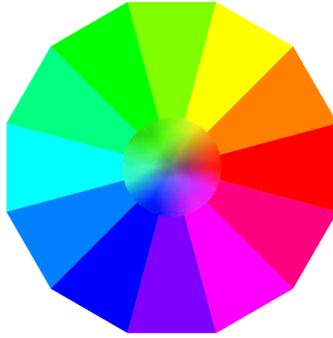


Fig. A.8: Discretización de círculo cromático

El valor de H se determina midiendo el ángulo entre la proyección del vector RGB sobre el plano cuya normal es el eje I , y el eje inercial R_I (Fig. A.9).

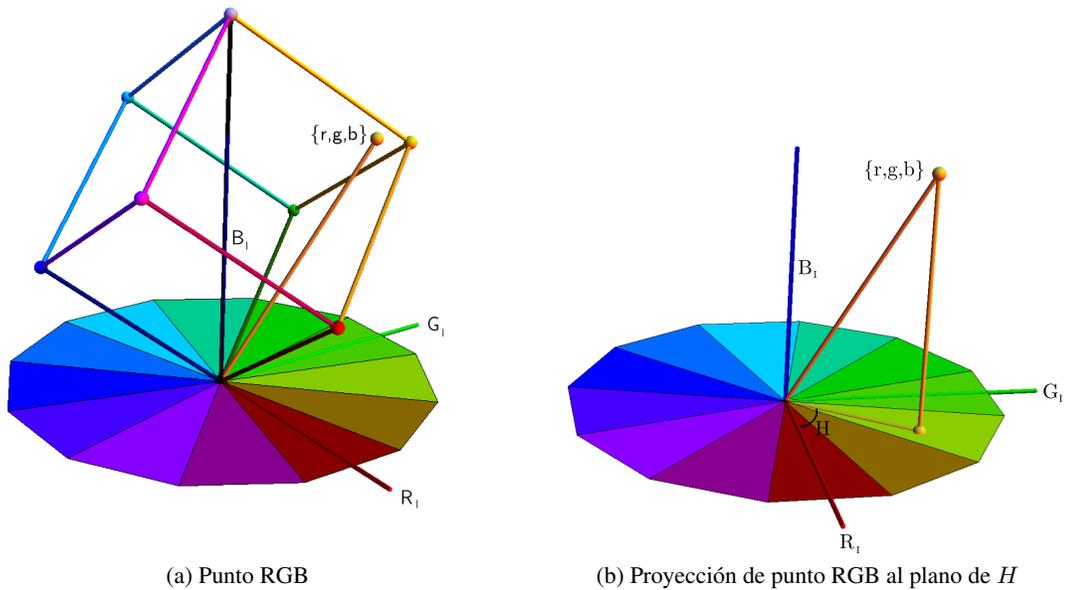
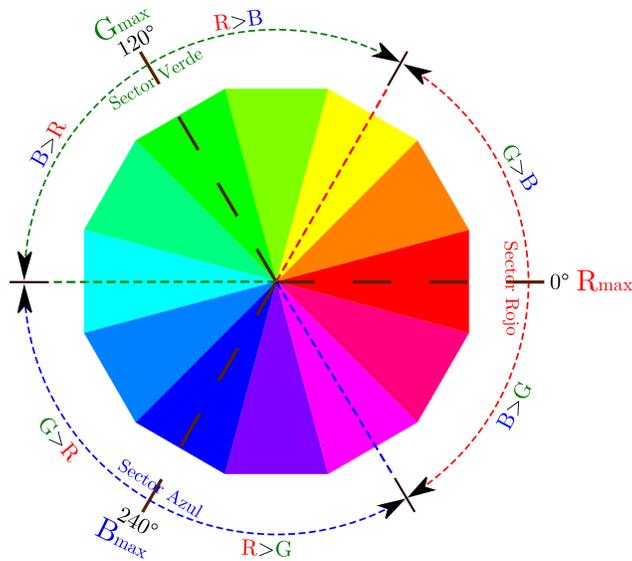


Fig. A.9: Determinación de H para punto RGB

Al recorrer el espacio de color RGB es posible observar que el valor de H sigue un cierto patrón, es decir, dependiendo de qué componente de la triada (r, g, b) sea la dominante, el valor de H estará en cierto sector del círculo cromático. Existen tres sectores, el rojo, el verde y el azul, nombrados en consecuencia de la componente dominante, tal como se muestra en la Fig. A.10. Las bisectrices de los sectores representan los tonos de los colores primarios, formando un ángulo de 120° entre ellos. Tomando en cuenta la relación de desigualdad entre los valores de las componentes no dominantes, el valor de H en cada sector se desplaza hacia un lado u otro de la bisectriz correspondiente.


 Fig. A.10: Comportamiento del valor de H al variar RGB

Por ejemplo, para el caso donde la componente r es el máximo de los tres valores RGB, el valor de H estará contenido en la sección roja. Si $g > b$, el valor de H se desplaza en sentido positivo (giro antihorario), alejándose de la bisectriz. El desplazamiento será en sentido contrario si $b > g$.

Calcular el valor de H con la formulación convencional (Ec. A.2 y Ec. A.1) resulta computacionalmente costoso, pues es necesario calcular una raíz cuadrada y una función trigonométrica inversa, ambas, operaciones que consumen niveles importantes de tiempo y recursos. Es necesario entonces buscar alternativas para la conversión de imágenes RGB al canal H del modelo HSI. En 1978 el Dr. Alvy Ray Smith desarrolló un modelo de color denominado HSV [75]. Una de las aportaciones de este trabajo fue encontrar una relación sencilla entre la componente dominante de la triada RGB y las dos componentes restantes. Dicha relación se expresa como un factor que describe el cambio del valor de H en un cierto sector del círculo cromático. Por ejemplo, si se considera el caso en que r es máximo, es decir, el sector rojo, la relación descrita es:

$$F_{rmax} = \frac{g - b}{r - \min\{r, g, b\}}, \quad (\text{A.19})$$

De igual forma se definen relaciones entre los componentes RGB para el resto de los sectores. El rango de cada uno de estos factores es $[-1, 1]$, donde -1 representa que el valor de H se encuentra en el límite inferior del sector, mientras que 1 lo sitúa en el límite superior del mismo. Para llevar a cabo la transformación al canal H de un punto RGB, se debe reconocer en cual de los tres sectores se sitúa, y posteriormente su posición angular dentro de este.

La posición angular dentro del sector se determina de manera general como:

$$\angle_S = 60 \cdot \frac{Rel}{Dif} + 60, \quad (\text{A.20})$$

donde $\angle_S \in [0^\circ - 120^\circ]$ es la posición angular correspondiente a la triada RGB dentro del sector, Rel es la diferencia entre las componentes RGB no dominantes, y su valor se obtiene con base en la Tabla A.2, Dif es la diferencia entre el valor máximo (dominante) y el mínimo, y el cociente $\frac{Rel}{Dif}$ corresponde al factor definido por Ray Smith para cada uno de los sectores.

Tabla A.2: Valores de Rel

Sector	Rel
Rojo	$g - b$
Verde	$b - r$
Azul	$r - g$

La Ec. A.20 resulta fácil de implementar utilizando un arreglo que contenga el resultado del coeficiente $\frac{60}{Dif}$, y un índice de acceso igual al valor de Dif . Una vez calculada la posición angular (\angle_S) del punto RGB dentro del sector, solo resta determinar a que tono corresponde. Esto se logra mediante la implementación de un arreglo de memoria que relaciona la posición angular dentro del sector, y por ende, dentro del círculo cromático discretizado, con el tono correspondiente expresado dentro del intervalo de valores que puede adoptar la imagen. De esta forma el valor de H es:

$$H = Arreglo_i \left[60 \cdot \frac{Rel}{Dif} + 60 \right] \in [0, 255], \quad i \in \{rojo, verde, azul\}. \quad (A.21)$$

La Tabla A.3 muestra una comparación entre el tiempo de ejecución requerido para el cálculo de H utilizando la formulación convencional (Ec. A.2 y Ec. A.1), y la propuesta en este trabajo (Ec. A.21). La prueba consiste en calcular el valor de H para cada punto de RGB, hasta barrer completamente el espacio de RGB con una resolución de 8 bits. Los tiempos que se presentan en la tabla corresponden al promedio calculado con el resultado de cien ciclos de conversión. Se puede observar que la conversión basada en la Ec. A.21 ahorra, con respecto al cálculo convencional de H (Ec. A.1), más del 90 % de tiempo, aumentando la eficiencia del proceso.

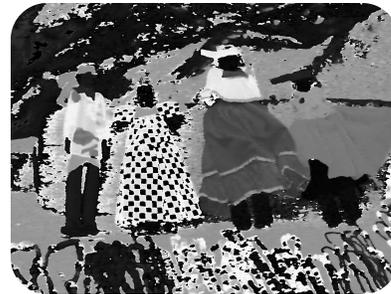
Tabla A.3: Comparación de algoritmos para cálculo de H

Formulaciones para calcular H	AMD PhenomII X4 810 2.6 GHz	Intel Core 2 Duo T5550 1.83 GHz	AMD PhenomII X4 945 3.0 GHz	Intel Xeon 2 Core 3.2 GHz
Convencional Ec. A.1	3,077.01 <i>ms</i>	4,358.54 <i>ms</i>	2,740.69 <i>ms</i>	3,830.14 <i>ms</i>
Con arreglo en memoria Ec. A.21	155.34 <i>ms</i>	143.86 <i>ms</i>	136.11 <i>ms</i>	118.66 <i>ms</i>
% de ahorro entre Ec. A.1 y Ec. A.21	94.95 %	96.70 %	95.03 %	96.90 %

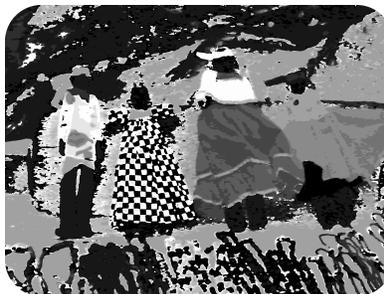
La Fig. A.11 muestra el resultado de aplicar la transformación al canal H de una imagen en RGB, utilizando la formulación convencional (Fig. A.11b), y la desarrollada en este trabajo (Fig. A.11c). Es posible apreciar que la conversión realizada con la nueva formulación arroja un resultado con áreas bien diferenciadas, y con menor discontinuidad que para el caso convencional. Queda claro que la conversión para el canal H que se propone no solo presenta ventajas en la disminución del tiempo de procesamiento, sino también en lo que a la calidad de la información arrojada se refiere.



(a) imagen original



(b) formulación convencional



(c) formulación nueva

Fig. A.11: Transformación de imagen de RGB a H