



Universidad Autónoma de San Luis Potosí

Facultad de Ingeniería

Centro de Investigación y Estudios de Posgrado

*“Desarrollo e Integración de un Cabezal
con Visión Artificial con un Robot”*

QUE PARA OBTENER EL GRADO DE
MAESTRO EN INGENIERÍA MECÁNICA

Presenta:

Ing. Víctor Hugo Juárez Matus

Asesor de Tesis:

Dr. Dirk Frederik De Lange

Septiembre 2013, San Luis Potosí, S.L.P.



17 de enero de 2013

AL ING. VÍCTOR HUGO JUÁREZ MATUS
P R E S E N T E. –

En atención a su solicitud de Temario, presentada por el **Dr. Dirk Frederik De Lange**, Asesor de la Tesis que desarrollará Usted, con el objeto de obtener el Grado de *Maestría en Ingeniería Mecánica*. Me es grato comunicarle que en la Sesión de Consejo Técnico Consultivo celebrada el día 17 de enero del presente año, fue aprobado el Temario propuesto:

TEMARIO:

**“DESARROLLO E INTEGRACIÓN DE UN CABEZAL CON VISIÓN
ARTIFICIAL CON UN ROBOT”**

Introducción.

- I. Estado de arte.
- II. Desarrollo del método de procesamiento de imágenes por optimización.
- III. Evaluación y pruebas del procesamiento de imágenes.
- IV. Integración física de cámara y manipulador.

Conclusiones.

Bibliografía.

Apéndice.

“MODOS ET CUNCTARUM RERUM MENSURAS AUDEBO”

ATENTAMENTE

M. I. JORGE ALBERTO PÉREZ GONZÁLEZ
DIRECTOR

UNIVERSIDAD AUTÓNOMA
DE SAN LUIS POTOSÍ
FACULTAD DE INGENIERÍA
DIRECCIÓN

**bcb



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

“90 Años de Autonomía, UASLP Primera Universidad Autónoma en México”

Índice general

Introducción	1
1. Estado del arte	5
1.1. Propuesta de un método alternativo	7
1.2. Aplicación	8
1.3. Métodos de optimización	15
1.4. Método del simple descenso	17
2. Desarrollo del método de procesamiento de imágenes por optimización	21
2.1. Generador de imágenes	22
2.1.1. Generador de puntos	23
2.2. Modelo de cámara (Pinhole)	32
2.3. Ensamblaje de puntos en una imagen	35
2.3.1. Método 1: <i>Puntos Gaussianos</i>	36
2.3.2. Método 2: <i>4Pixel</i>	39
2.4. Comparación de imágenes	44
2.5. Calibración de cámara	45
2.6. Calibración de láseres	50
3. Evaluación y pruebas del procesamiento de imágenes	53
3.1. Pruebas a partir de imágenes virtuales	53
3.1.1. Procesamiento de imágenes	59
3.2. Pruebas de calibración a partir de imágenes virtuales	69
3.2.1. Calibración de cámara	69
3.2.2. Calibración del cabezal	72
3.2.3. Procesamiento de una secuencia de imágenes	76
4. Integración física de cámara y manipulador	81
4.1. Desarrollo del cabezal	81
4.1.1. Cámara	82
4.1.2. Lente de resolución macroscópica	84

4.1.3. Diodos láser	84
4.1.4. Cabezal o interfaz	86
4.2. Manipulador serial	87
4.3. Acoplamiento/interfaz/comunicación	87
4.4. Integración del sistema	91
4.5. Pruebas con imágenes de cámara	94
4.5.1. Calibración	94
4.5.2. Procesamiento de una secuencia de imágenes	101
4.6. Evaluación del método para imágenes con ruido	111
Conclusiones	115
Bibliografía	117
A. Comunicación en matlab	119
Comunicación en matlab	119
A.1. Comunicación localhost	119
A.2. Comunicación TCP/IP	119

Introducción

En los procesos de automatización por visión artificial existe una gama amplia de técnicas de análisis aplicadas a la imagen, en las cuales se aplica un procesamiento sobre la imagen para poder determinar información que ayude al reconocimiento de formas, patrones y extracción de características, que ayudan al usuario, a hacer una interpretación sobre la imagen para una tarea específica. En este proceso se deben establecer las condiciones requeridas, considerando distancias, posiciones, orientaciones, así como también, las condiciones de luz del medio ambiente en la que se desarrollan las pruebas. Las técnicas de análisis sobre la imagen dependen en gran parte de las condiciones de luz que se dan sobre la imagen, si éstas son malas, entonces no se obtienen resultados buenos.

En esta tesis se pretende aplicar una nueva metodología que permita un análisis sobre la imagen para poder obtener información de interés, usualmente un número de variables desconocidas como distancias, ángulos, coordenadas de objetos entre otras. Lo deseable es que dicha metodología sea rápida, robusta y que sobre todo, que sea menos sensible a las condiciones de luz del entorno. En lugar de procesar la imagen, la metodología propuesta “genera una imagen virtual” a partir de información predeterminada y un supuesto de la información a determinar. Una vez que se tenga la imagen, ésta se lleva a comparar con la imagen real obtenida de la cámara. De la comparación entre las imágenes se obtiene un valor de calidad. Para mejorar la correspondencia entre la imagen real y la imagen virtual se busca optimizar esta calidad por medio de un algoritmo de optimización, en la cual se modifica el valor de las variables desconocidas para generar otra imagen para ser comparada nuevamente, hasta obtener una calidad óptima. Cuando la calidad es óptima y de valor alto, entonces se asume que la información de interés ha convergido al valor buscado en la imagen real.

En este trabajo se enfoque al desarrollo del método aplicado a la necesidad de un posicionamiento preciso de un cabezal, manipulado por un robot tipo brazo, con respeto a una pieza de trabajo. El interés surge en particular por la precisión de posicionamiento requerido por ejemplo en el proceso de soldadura por láser.

Objetivo general

El objetivo general de esta tesis es el desarrollo de un sistema de visión artificial que haga uso de la proyección de luz estructurada sobre una superficie e incluir la metodología que se plantea desarrollar, con lo cual se puede determinar la orientación y posición del sistema de visión respecto a la superficie de proyección. Se busca llevar la metodología a la prueba para evaluar su desempeño en la práctica.

Al integrar el sistema desarrollado a un robot soldador de tipo industrial, entonces éste podría identificar el tipo y la posición de la junta, para seguirla y soldarla.

Objetivos particulares

Dentro del desarrollo del trabajo se puede establecer algunos objetivos particulares como:

1. Generar imágenes artificiales a través de un método rápido y eficiente con una calidad aceptable de representación de la imagen real correspondiente.
2. Optimizar el código, optimizando el algoritmo en términos de rapidez. El aspecto de rapidez es importante para que se pueda aplicar el método para corrección de movimientos en tiempo real en sistemas dinámicos.
3. Realizar un caso de estudio en particular, que incluye el análisis y generación de imágenes en una superficie plana, determinando la posición de un cabezal con el sistema de visión con respecto a esta superficie.
4. Integración del cabezal con un sistema manipulador tipo robot y realizar una implementación del método que se desarrolla.
5. Realizar la evaluación del método. Se busca determinar si el método tiene ventajas prácticas en términos de su rapidez, robustez y simplicidad, de acuerdo a una serie de pruebas a las que será sometido el método.

Hipótesis del trabajo

“Existe una metodología que puede llegar a determinar información sobre la imagen de forma más precisa sin realizar un procesamiento sobre la imagen, en particular para aplicación sobre geometrías irregulares, pero conocidas. A su vez, la metodología a desarrollar podría ser menos susceptible a las condiciones de luz o visibilidad parcial de objetos, haciendo el método más robusto que el procesamiento convencional de imágenes.”

Estructura de la tesis

En el primer capítulo de esta tesis se describen algunos trabajos que hacen uso de luz estructurada, siendo casos variados en los que se usan formas distintas para aplicar la luz proyectada sobre la pieza de trabajo. También, se describe en más detalle la metodología propuesta para la visión artificial y la aplicación por la cual se desarrolla el sistema. Enseguida se propone y describe el método de optimización que se emplea para el desarrollo del trabajo. En el segundo capítulo, se detalla el desarrollo de cada una de las funciones que intervienen en el método que genera imágenes y la propuesta para los procesos de calibración de cámara y cabezal. En el tercer capítulo se realiza la evaluación del método a través de una serie de pruebas virtuales: buscando la información deseada en imágenes virtuales generadas por el propio método desarrollado. Dado que se conoce el valor de las variables con la que fueron generadas las imágenes virtuales, se tiene un valor de referencia exacta, para poder determinar la precisión con la cual se detecta las variables a determinar. Además, se determina la rapidez del método para generar imágenes bajo diferentes condiciones, se obtienen tiempos y precisión para determinar información sobre una imagen y una secuencia de imágenes. De igual forma también se realiza la validación de los procesos de calibración establecidos en el segundo capítulo. En el cuarto capítulo, se realiza la implementación física del cabezal y su integración con un brazo manipulador del tipo industrial, para realizar pruebas reales validando el método usando imágenes obtenidas de la cámara.

Capítulo 1

Estado del arte

El uso de luz estructurada para la detección de superficies y piezas es muy utilizada en visión artificial. En cada una de las técnicas se utiliza una o más cámaras que observan la escena de trabajo, lo que lleva a emplear técnicas de procesamiento de imágenes para obtener información a partir de la cual se puede reconstruir los objetos presentes en la imagen.

El procesamiento de imágenes está dividido en tres segmentos básicos de programación: bajo, medio y alto nivel. Cada nivel de programación tiene ciertas tareas que entran en estos niveles, para ello se usa la pirámide mostrada en la figura 1.1.

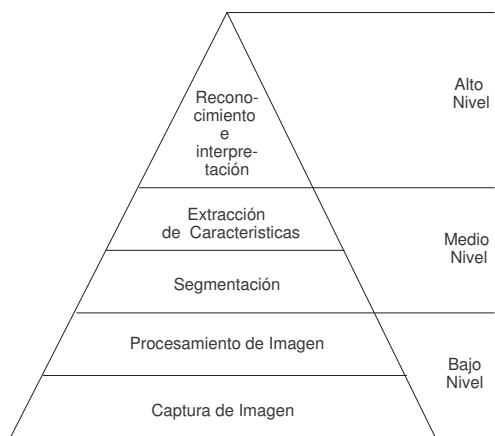


Figura 1.1: Niveles de programación en el procesamiento de imagen.

Para las operaciones de bajo nivel se hace la captura y procesamiento de imagen, involucra el diseño de las propiedades de la captura de la imagen, el tipo de cámara a utilizar, distancia del objeto, por mencionar algunos. También, contempla la reducción del entorno que no es de interés en la imagen, como puede ser el fondo. Para las tareas del segundo

nivel: Segmentación y extracción de características, es donde se trata de obtener la información que interesa de la imagen, es decir, reconocer y extraer cada uno de los objetos que se logran identificar, así como también, seleccionar y extraer las características apropiadas para identificar los objetos deseados. Finalmente las de alto nivel: Reconocimiento e interpretación de la información extraída de los procesos anteriores sobre la imagen. Para fines prácticos de referencia para este trabajo de tesis, estará referido como una metodología convencional y tradicional.

Algunos de los ejemplos de luz estructurada que es usada para la detección de superficies y juntas de soldadura es la utilización de proyección de luz en forma circular sobre una superficie y la cámara que observa la escena (Xu et al., 2006). Dependiendo del tipo de junta es la deformación del láser y después aplican segmentación, filtros y se obtiene reconstrucción de la imagen.

Otra de las alternativas propuestas para detectar la orientación y posicionamiento de una pieza respecto a un brazo manipulador de soldadura es el empleo de cuatro láseres de línea, posicionados en forma de pirámide rectangular los cuales son proyectados sobre la pieza desde la parte superior; después se realizan la proyección de los láseres sobre el costado de la pieza para detectar más información y poder trabajar sobre la pieza (Jae Byung et al., 2009).

Otra técnica de luz estructurada es en la que se proyecta un par de líneas paralelas separadas una distancia conocida con un ángulo de proyección y una cámara que observa para detectar un cordón de soldadura (Gao et al., 2008), para realizar el procesamiento de la imagen y obtener información de la misma por medio de la detección de bordes ortogonales de Zernike¹. Los movimientos deben de ser siempre en una dirección y deben de cuidar que el cordón de soldadura pase por ambos láseres (entrada y salida).

También sucede que se dan técnicas donde se sitúa un láser de línea a una distancia de la superficie, y se tiene un par de cámaras realizando una visión estereoscópica las cuales se encuentran por un costado del láser (He-xi et al., 2009). Este trabajo se caracteriza por que realiza tres controles sobre el sistema, control de un brazo, visión estereoscópica sobre la superficie y un control sobre la antorcha de soldadura por medio de un algoritmo de optimización llamado Colonia de Hormigas.

También se reporta el uso de una unidad Pan-Tilt la cual es un mecanismo de 2 grados de libertad, que tiene una velocidad de $300^\circ/s$ y una resolución de $0,012857^\circ$, con un láser de línea en su extremo para realizar un barrido o hacer un tipo escáner de una superficie y un par de cámaras que permite observar el movimiento del láser (visión estereoscópica por medio de Manipulación de Espacio de Cámara por Modelo de Cámara Lineal (LCM-CSM)) (García Cedillo, 2011). Después de aplicar técnicas de procesamiento de imágenes y obtener

¹Los polinomios de Zernike, junto con la la técnica de ortogonalización de Gram-Schmidt, han sido los métodos utilizados para determinar las aberraciones de un sistema óptico. Se hace uso de los polinomios de Zernke debido a su relación con las aberraciones de: esfericidad, coma, astigmatismo, distorsión y curvatura del campo. Debido a sus propiedad de de invariancia rotacional, ortogonalidad y normalización, los polinomios de Zernike han sido utilizados para describir las aberraciones de una lente rígida.

información del objeto para realizar una re-construcción del objeto escaneado (contorno y orientación) para después manipular un robot serial industrial y tomar el objeto.

Finalmente, otros trabajos en donde se emplea luz estructurada para la detección de juntas para soldadura láser usando luz proyectada en forma de triángulo; esto realiza por medio de láser cruzados entre sí, teniendo un ángulo de proyección, (Iakovou, 2009) y (Entzinger, 2005). Los láseres proyectores de línea se encuentran montados alrededor de la cámara y, por medio de triangularización, obtienen la posición y orientación del manipulador respecto a la superficie.

El uso de proyección de luz estructurada puede ser variado, en cada implementación se puede obtener beneficios que, junto con un análisis, se puede resolver la problemática que es planteada en cada caso.

1.1. Propuesta de un método alternativo

En este trabajo se propone un método diferente y que pueda ser alternativo para que permita dar mejores soluciones o resultados a los que se obtienen con el procesamiento de imágenes. Una vez que se desarrolle el método, entonces deberá ser sometida a una evaluación para realizar un análisis de los resultados obtenidos.

En contrario al método convencional discutido anteriormente, en la cual se procesa la imagen para extraer datos de ella, en el método propuesto se usa la imagen solo para compararla iterativamente con una imagen creada virtualmente. Este se realiza en un proceso de optimización con un diagrama de flujo que se muestra en la figura 1.2.

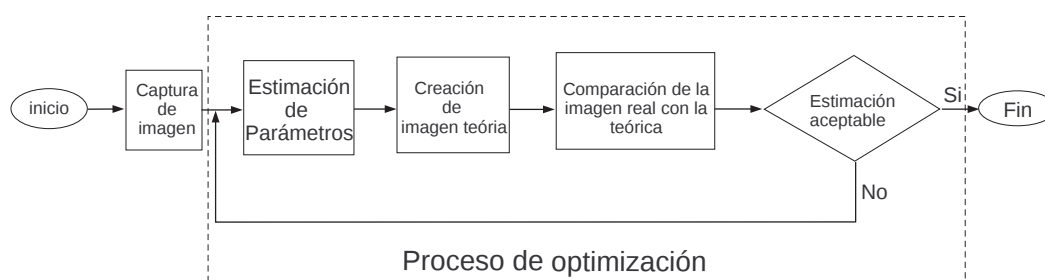


Figura 1.2: Metodología de trabajo.

De izquierda a derecha en el diagrama 1.2, se comienza con una captura de imagen a través de la cámara que estará contenida en el cabezal de visión que se desarrolla. Dicha imagen es enriquecida en información por medio de láseres de tipo línea. Después en una primera aproximación se proporciona información *estimada* que es usada en el siguiente bloque de trabajo, para la generación de una imagen teórica. El siguiente paso consiste en realizar una comparación entre imagen real capturada y la imagen teórica generada con base en su primera estimación de información. Si de la comparación se obtiene una estimación

no aceptable, (criterio de correspondencia entre imágenes), entonces se vuelve a crear una nueva imagen teórica modificando la información estimada utilizando los resultados de la evaluación anterior y esto se realiza usando un algoritmo de optimización y se hace nuevamente la comparación sin perder la imagen real. Si el criterio de correspondencia es *aceptable* entonces termina el proceso de iteraciones. Como resultado del proceso se conoce un conjunto de variables que crea o reproduce una imagen teórica que es semejante o exacta con la imagen real. Entonces esta tarea se realizará para cada nueva imagen que se pueda adquirir durante un ciclo de trabajo.

1.2. Aplicación

En un procesos de soldadura, donde es necesario unir dos piezas de metal a través de la fundición de las mismas, es de suma importancia tener un control del avance y suministro de energía que se aplica al proceso de soldadura. Cuando el proceso es por medio de electrodos o antorcha y de manera artesanal, se confía en la habilidad de quien hace el proceso de soldadura, no considerando la exactitud con que debe de realizarse la tarea. En un proceso que puede ser automatizado, por medio de un robot de tipo industrial, la precisión con la cual se hace la soldadura puede ser muy elevado y considerando error de menos de un milímetro, por las capacidades técnicas y de operación de los mismos. Cuando se integra al robot un sistema de soldadura por láser se puede lograr procesos de soldadura muy precisos. El sistema se puede hacer automático a través de una enseñanza y repetición que se puede programar al robot, y ejecutar las tareas de forma cíclica. De anexarse un modulo de visión, entonces se convierte en un proceso que se puede automatizar y que posiblemente pueda ser más independiente del proceso de enseñanza y repetición. Si el robot puede reconocer su posición y orientación de la superficie de trabajo y, que a su vez, pueda reconocer, detectar y seguir una ranura de soldadura y que posteriormente la pueda soldar, daría una alta autonomía y precisión al proceso de soldadura por láser. Considerando que sería para algunos casos en específico, en los cuales se pueda lograr un estudio y en esa dirección se plantea este trabajo que se desarrolla.

Considerando lo anterior, se desglosa un estudio que ayudaría en gran medida a conocer la ubicación del sistema de soldadura. Para medir la posición de una herramienta con respecto a un objeto o una pieza de trabajo, se desarrolla un cabezal que proyecta luz estructurada a la pieza y una cámara que observa la proyección resultante. Esto se muestra en la figura 1.3.

En la proyección de la luz estructurada *a)* sobre la pieza u objeto de trabajo se usan tres láseres de tipo línea. Además, una cámara observa la superficie sobre la cual se proyectan los láser *b)*. Estos tres láser se encuentran distribuidos alrededor de la cámara a una distancia al eje focal de cámara. Si en la superficie sobre la cual se proyectan las líneas *(c)* se coloca un plano de referencia, denominado como $(x - y)$, entonces la normal a dicha superficie bien se puede emplear para referenciar un ángulo de proyección de los láseres.

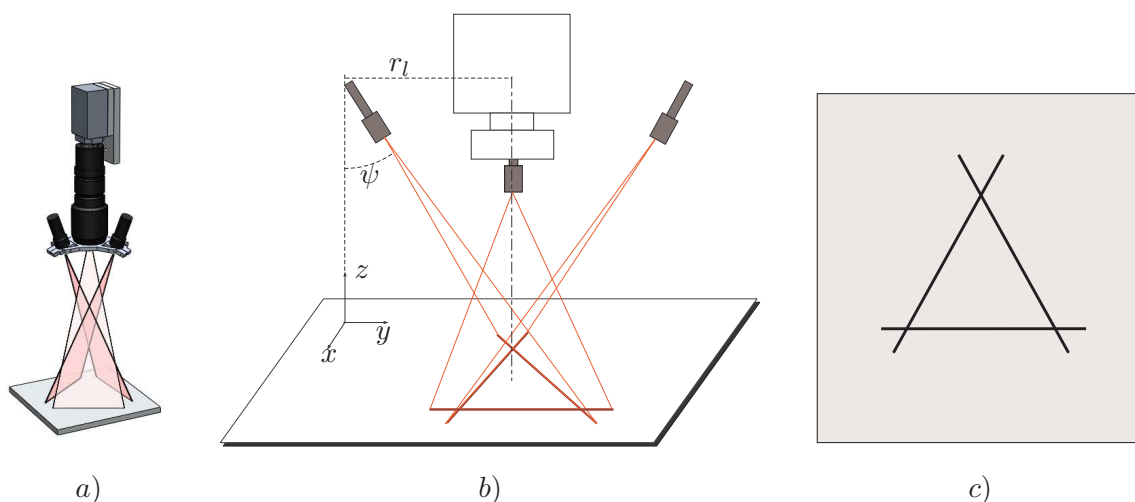


Figura 1.3: Configuración de espacio de trabajo, a) diseño virtual de la problemática que se plantea, b) primer modelo matemático de análisis para el estudio del problema, c) proyección de los haces de luz sobre la superficie que estaría observando la cámara.

Al montar esta configuración de trabajo, se puede obtener la imagen que se muestra en la figura 1.4.

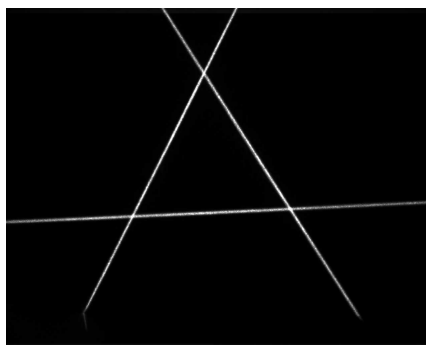


Figura 1.4: Imagen resultado de cámara-láser-superficie.

Para la generación de imágenes debe de ser considerada las variaciones de altura de la superficie sobre la cual se proyectan los láser, como se puede observar en la figura 1.5, la característica de imagen es que el sensor de cámara es paralelo a la superficie de proyección de los haces de luz.

Ahora bien, si existiera un ángulo de inclinación de la superficie respecto al sensor de cámara, las situaciones que se podrían presentar son mostrados en la figura 1.6.

Las imágenes que se muestran en la figura 1.7 son el resultado de la proyección de los

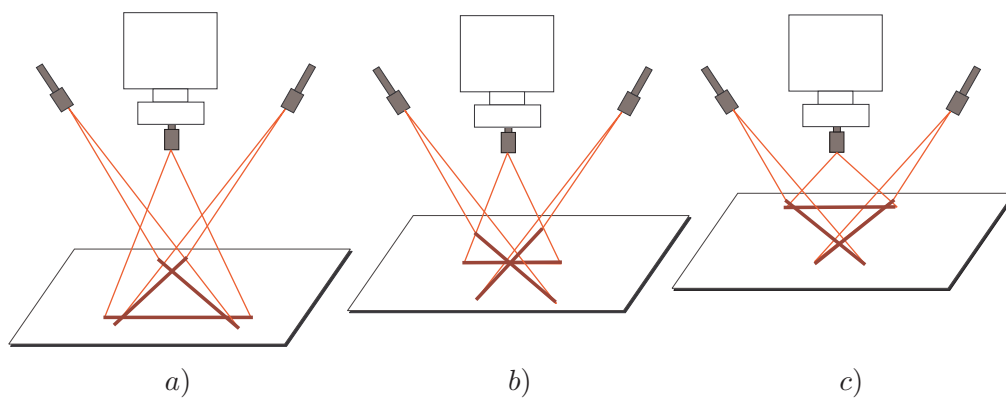


Figura 1.5: Proyección de los láseres de línea sobre una superficie plana. *a)* la distancia entre el plano y la cámara es grande, *b)* una distancia media entre plano de proyección y cámara, *c)* una distancia reducida entre el superficie de proyección y la cámara.

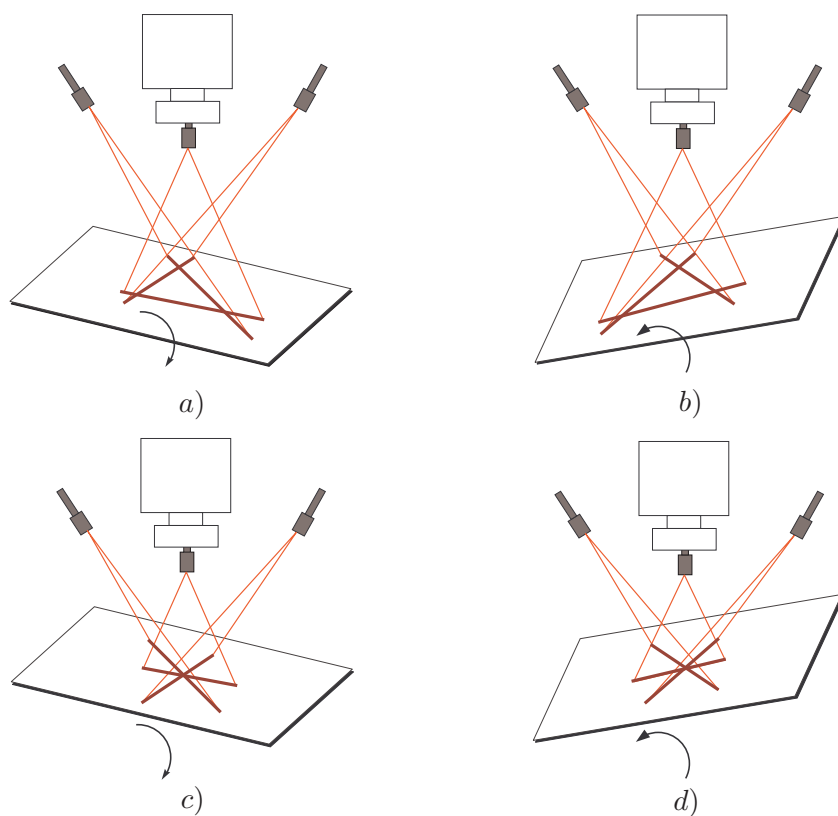


Figura 1.6: Proyección de la luz estructurada sobre una superficie con declive. *a)* y *c)* superficie con un declive en sentido horario con dos variaciones de alturas, *b)* y *d)* superficie con un declive en sentido anti-horario, en diferentes alturas.

láser de tipo línea sobre una superficie como se plantean en este capítulo. Se coloca una superficie a una distancia del cabezal a), después se acerca la superficie de proyección al cabezal, lo que hace que cambien las dimensiones del triángulo que se proyecta por los láseres b) (debido al ángulo de proyección que éstos tienen), en algún momento se pasa por el punto de convergencia de los haces de luz c), y después se invierte el triángulo de proyección d), finalmente el plano está más cerca del cabezal e).

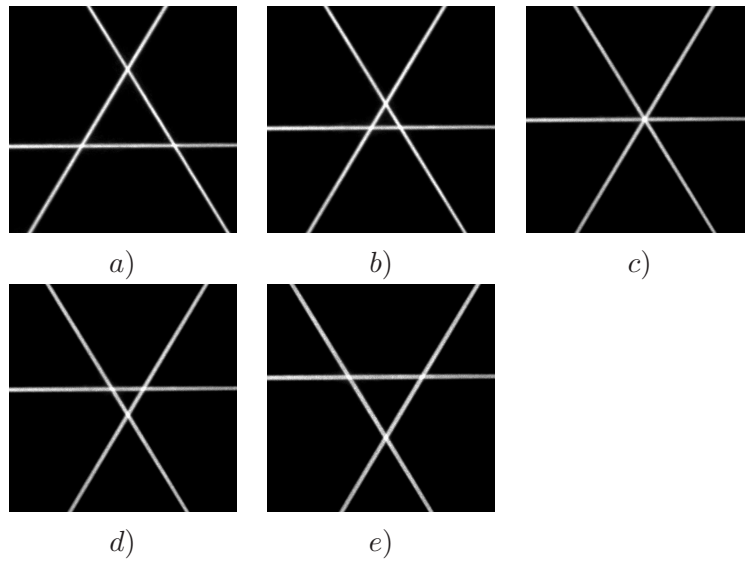


Figura 1.7: a) distancia entre cámara y plano muy grande, b) se disminuye la distancia entre la cámara y plano y reduce las dimensiones del triángulo proyectante, c) se muestra un punto de convergencia entre los láseres que se suita al reducir la distancia entre plano y cámara, d) a cabo de seguir reduciendo la distancia se realiza una inversión del triángulo proyectante por los haces de línea, e) el plano se encuentra lo suficientemente cerca de cámara para la inversión del triángulo y a un tamaño considerable.

Cuando las superficie tiene un ángulo de inclinación respecto al sensor de cámara, se pueden presentar las deformaciones de los triángulos como se observan en las imágenes mostradas en la figura 1.8.

De forma resumida, *la proyección de los láser sobre una superficie plana se convierte en el enfoque de interés para la generación de imágenes a través de la metodología que se desarrolla.*

La forma en que se enriquece la imagen con la proyección de luz estructurada con el uso de geometría y algún par de consideraciones se puede obtener solución a este planteamiento. El objetivo se centra en poder determinar la posición y orientación de la superficie respecto al cabezal de visión que se desarrolla. Así mismo, este cabezal va a ser integrado a un manipulador para el cual se puede determinar información suficiente que permita realizar manipulaciones básicas de movimientos.

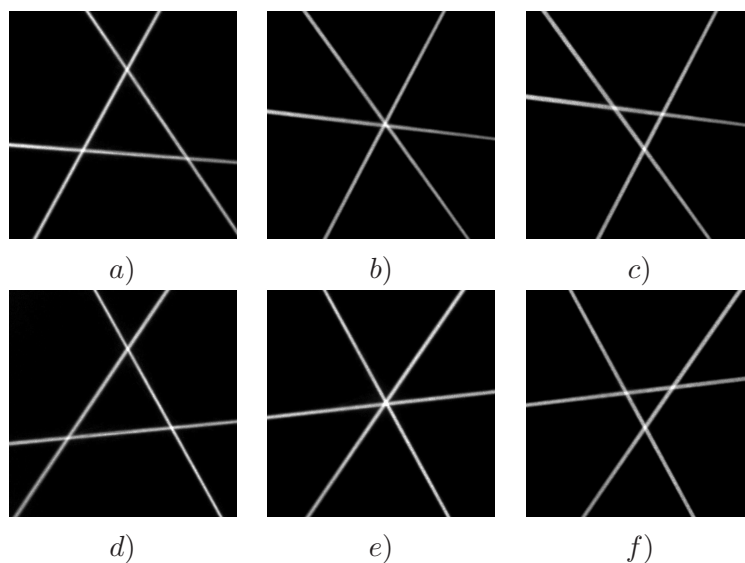


Figura 1.8: En *a)*, *b)* y *c)* es cuando la superficie tiene un declive en sentido horario. Para *d)*, *e)* y *f)*, cuando la superficie tiene una pendiente en sentido anti-horario.

Una vez integrados el cabezal de visión con el robot serial, se puede emplear esto para la detección de ranuras de soldadura.

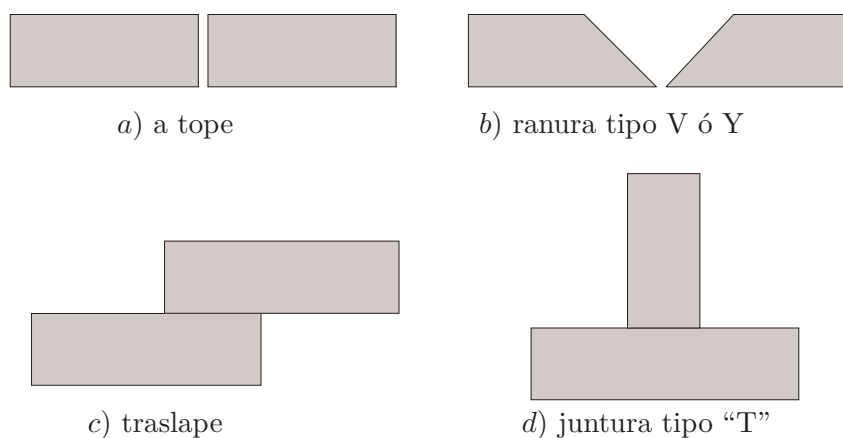


Figura 1.9: Juntas de soldadura.

En la figura 1.9 son mostradas las configuraciones de soldadura empleadas más frecuentemente. Se fija particular interés en *a)* ya que es el más difícil de detectar porque no deja mucha "huella" en la imagen, mientras que los otros dejan una marca más notoria para reconocer. En otras palabras, es el más crítico y más deseable para poder manejar para soldadura con láser. Con la luz estructurada proyectada sobre la superficie se espera poder identificar y seguir la ranura de soldadura y con el robot manipulador poder realizar los

movimiento necesarios para hacer la tarea.

Las imágenes que se logran obtener para las configuraciones presentadas anteriormente y las proyecciones de la luz estructurada, se pueden visualizar en las fotografías mostradas en la figura 1.10.

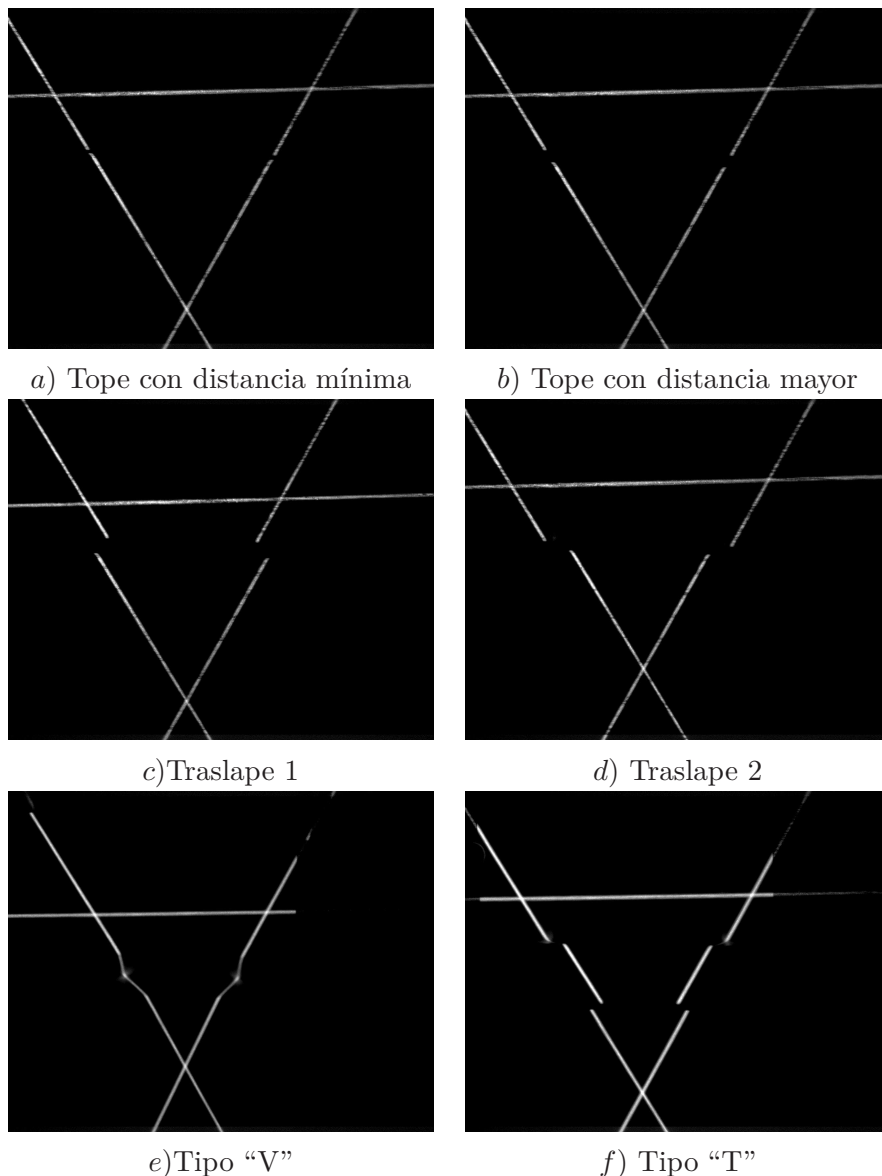


Figura 1.10: Configuraciones de soldadura con luz estructurada proyectada.

Para las imágenes de *a)* y *b)* se muestra la configuración a “tope”, solo que la distancia es mínima en la primera y no se alcanza a preciar de forma clara, mientras que en la segunda es más visible la interrupción de línea. En *c)* y *d)* se pone en evidencia el tipo “traslape”; en la imagen de *c)* la parte superior de la discontinuidad de línea es más cercana

a la cámara que la parte inferior, y caso contrario sucede para la imagen de d). La imagen de configuración en tipo “V” le corresponde a e) y finalmente la tipo “T” es para la imagen de f).

En la figura mostrada en 1.11 se puede visualizar el montaje del sistema del cabezal de visión y el robot serial industrial.

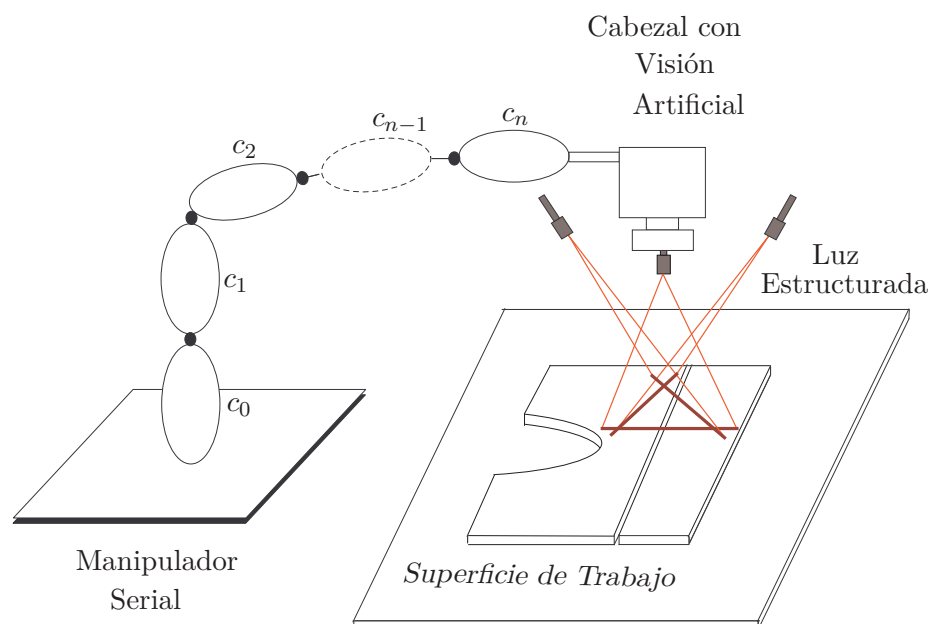


Figura 1.11: Detección de superficie en posición y orientación mediante cabezal de visión.

Un manipulador serial industrial convencional es una cadena cinemática abierta formada por un conjunto de eslabones o elementos de la cadena interrelacionados mediante articulaciones o pares cinemáticos (Baturone, 2001).

Al integrar todo el sistema sobre un robot industrial de tipo soldador, es posible que se pueda detectar algunas de las juntas de soldadura con una gran precisión, como se está planteando que sea, para poder detectar, seguir y soldarla. Si el planteamiento propuesto y la metodología es buena y rápida, entonces, sería posible llegar a realizar una corrección en tiempo real del movimiento que se programa al robot manipulador.

Una vez que se ha realizado el análisis de la proyección de la luz estructurada sobre una superficie plana, se puede *generar un código* que permita realizar una imagen de forma pre-determinada con las características que se observan al momento de proyectar la haces de luz sobre el plano. Dicha imagen es la que se podría generar con la información estimada y subsecuentemente se llevaría a comparar con la imagen real.

Lo que se espera del método es que sea más robusto que el método tradicional, ya que solo con partes de una línea se podría definir los parámetros, podría ser más preciso para superficies curvas, o para tomar en cuenta efectos de deformación radial de la lente,

porque el generador puede generar líneas curvas y compararlos, mientras que el método convencional trabaja principalmente con líneas rectas.

En algunos trabajos citados anteriormente reportan errores de detección del objeto a nivel de varios milímetros como sucede en (García Cedillo, 2011), (Xu et al., 2006), (Jae Byung et al., 2009) mientras que se espera poder mejorar el error.

1.3. Métodos de optimización

Parte esencial del método propuesto es el proceso de optimización. Dentro de la literatura, se puede encontrar una diversidad de métodos de optimización, entre algunos que figuran son los que utilizan derivadas y otros que no. A base de iteraciones y evaluación de una función objetivo es que llegan a obtener un máximo o mínimo de la función. Algunas características que tienen los métodos que no usan derivadas es que llegan a ser muy flexibles ya que no se limitan a funciones diferenciables, que dado en algunos casos, la función objetivo llega a ser compleja y obtener su derivada en ocasiones se convierte en tarea muy difícil. Algunos otros métodos de optimización son del tipo estocásticos, un proceso estocástico puede interpretarse como una sucesión de variables aleatorias cuyas características pueden variar a lo largo del tiempo y lo hacen en función de otra variable. Los métodos son lentos en comparativa contra los métodos para funciones analíticas diferenciable. Dentro de las mejores opciones, pueden llegar a ser métodos de gran utilidad para la optimización y generación de modelos.

Dentro de la categoría de métodos sin derivadas, se pueden mencionar algunos:

Simulado recocido (Simulated annealing o SA) Para ciertos problemas, puede ser más eficiente que la *enumeración exhaustiva*²; siempre que el objetivo sea simplemente para encontrar una solución aceptablemente en un tiempo determinado, en lugar de la mejor solución posible.

Búsqueda de descenso del simplex (Downhill simplex search) Es un arreglo de $n + 1$ puntos en un espacio de n -dimensiones. El simplex (un triángulo en espacio 2D, un tetraedro en espacio 3D) se adapta al entorno de la función objetivo y (como una ameba) eventualmente encuentra un mínimo local cercano. El simplex tiene que contener un volumen de n -dimensiones, y para generar el simplex se comienza con un punto inicial P_0 y los otros se generan.

Búsqueda aleatoria (Random search) La búsqueda aleatoria explora el espacio paramétrico de una manera aparentemente aleatoria para encontrar el punto (x) que minimiza o maximiza la función $f(x)$.

²Consiste en enumerar todas las soluciones posibles, a partir de los valores tomados para las variables enteras y realizar todas las combinaciones posibles hasta encontrar una combinación que proporcione el valor óptimo de la función objetivo y que cumpla con todas las restricciones del problema. Una de las objeciones principales que presenta éste método es el número de variables, ya que se presentan demasiadas combinaciones antes de encontrar la solución óptima.

Programación genética Busca la generación de programas a través de métodos evolutivos, esto incluye sistemas que constituyen o contienen referencias explícitas a programas. Los programas evolucionados pueden ser en *C*, *LISP* u otros lenguajes como *netlists* (SPICE). Los programas pueden ser representados en variados tipos de estructuras incluyendo árboles, listas (como en *LISP*) u otros. Los programas a evolucionar no necesariamente tienen que ser código secuenciales, pueden ser redes neuronales u otros.

Optimización de enjambre de partícula (Particle swarm optimization o PSO)

El método PSO de Kennedy y Eberhardt fue originalmente implementado para simular el comportamiento social de una bandada de pájaros, un efecto de esta simulación es que los individuos generalmente llamados partículas estaban haciendo optimización. En esta búsqueda las partículas exploran el espacio paramétrico para encontrar el punto (x) que minimiza o maximiza la función de costo $f(x)$, $f : R^n \rightarrow R$.

Subiendo la colina estocástica (Stochastic hill climbing) Es un método que hace una búsqueda en un espacio discreto S con el fin de encontrar un estado que tenga un *fitness* lo mas alto (o bajo) posible. El algoritmo lo hace al hacer sucesivas mejoras a un estado $\sigma \in S$. La forma de los estados en S depende de como se codificaron las soluciones al problema a resolver.

En la gama amplia de métodos de optimización, se puede tener ventajas de un método sobre el otro que dependa del tipo y clasificación del método. El método de “Downhill simplex search” es de los que no necesitan la derivada de la función objetivo para optimizar, y a su vez, escapa al concepto de ser estocástico y trabaja con sistemas continuos. Los algoritmos estocásticos tienen la posibilidad de trabajar con sistemas que pueden llegar a ser discretos o discontinuos, cuando tienen que llegar a converger a un punto máximo o mínimo, llegan a tomar muchas iteraciones para generar una nube de puntos y determinar si es o no un punto óptimo. Aunque tienen la posibilidad de caer en mínimos locales, la nube de puntos puede llegar a concentrarse alrededor de los mínimos, pero termina identificando que no es el más óptimo y sigue iterando hasta encontrar el mejor valor, cabe mencionar que de iteración a iteración no converge el punto. Lo anterior, llega a consumir demasiado tiempo que, para el término de rapidez, ya no es factible para el proyecto. Los métodos que usan derivada convergen rápidamente en comparación de los que no los usan, por que usan el gradiente de la función y saben en que dirección ha de desplazarse para llegar al óptimo. Cuando la función es altamente lineal y es un método que usa derivadas, entonces obtener el mínimo de la función es rápido por que puede usar pasos grandes en incrementos utilizando las evaluaciones anteriores, mientras que para los que no usan derivadas se debe de hacer una evaluación en todas las vértices y desplazarse ortogonal a ellos, lo que conlleva a realizar $2n$ (izquierda o derecha) evaluaciones por cada vértice antes de moverse. Para el caso en que la función es no lineal, los métodos que usan derivadas pierden mucha rapidez por que ya no es tan predecible el valor próximo para converger y es cuando los métodos sin derivadas obtienen mayor beneficio por la forma en que se desplazan a los óptimos de

las funciones. La anterior situación hace que los algoritmos sin derivadas sean robustos y no evitan que estos caigan en mínimos locales; una forma de evitar que los métodos caigan en mínimos locales es que se proporcione información, de tal manera, que sea cercana a un mínimo global cuando sea posible esto.

1.4. Método del simple descenso

El algoritmo “The Downhill Simplex Method” propuesto por Nelder-Mead (J. A. Nelder, 1965), es un método numérico multidimensional por que puede manejar tantas variables se deseen y usa una tendencia de modelo de máximos y mínimos sin emplear el cálculo de derivadas. Recibe una función objetivo³ (FO) de N variables independientes a optimizar y a partir de un punto inicial propuesto por P_0 y N puntos adicionales según la ecuación 1.1. El método realiza la evaluación de la función para los puntos propuestos y determina que punto es peor para reemplazarlo por uno nuevo, de acuerdo a la figura 1.12.

Para la evaluación de esta función es necesario generar N puntos propuestos, y toma la forma de:

$$P_i = P_0 + \lambda \varrho_i, \quad (1.1)$$

donde P_0 es el punto inicial de partida, λ son las constantes que caracterizan a la escala de magnitud para cada dirección del vector que típicamente pueden ser designados con valor unitario, siendo éstas magnitudes de las variables las que se modifican en las primeras iteraciones del método y ϱ son los vectores base (basis) ortonormal de nuestro espacio de N dimensiones.

Dentro del algoritmo se pueden encontrar algunas ventajas y también algunas limitaciones:

Ventajas:

- Solo realiza la evaluación de funciones objetivo.
- Siempre termina por converger, a comparación de otros métodos numéricos o aleatorios.
- El programa es robusto y ya esta programado en matlab (también en C por medio de Numerical Recipes).
- Las evaluaciones del FO deben de costar poco esfuerzo computacional.
- No es necesario obtener la derivada de la función, ya que algunas veces se torna complicado hacerlo, esto permite que el método sea poderoso.

Desventajas:

- El método converge relativamente lento.

³Por convenio suele hablarse de minimización y por ello la función a minimizar recibe muchas veces el nombre de función objetivo o de coste.

- Si la evaluación de la función es costosa, no es eficiente el método.
- Puede caer en mínimos locales y es necesario reiniciar el método en caso que se detecta que el punto encontrado no es el mínimo global.

Se puede realizar una descripción del “Simplex” de forma geométrica de la siguiente manera. La dimensión se refiere al número de variables desconocidas o, dicho de otra forma, al número de argumentos de la función objetivo que deben de ser optimizados. El tamaño del simplex es el número de vértices de la figura que se forma para realizar la optimización y esta dado por λ .

Dimensión 1 El simplex tiene 2 vértices, en esta caso se interpreta como una porción de una recta en el espacio. Los valores propuestos iniciales pueden cambiar de posición en cualquier sentido, es decir, la porción de la recta puede aumentar en tamaño o puede reducirlo, así mismo también pueden cambiar el sentido en el que crece o disminuye.

Dimensión 2 El simplex es de 3 vértices, esto genera un triángulo en un plano. La evaluación de cada vértices pueden hacer que las dimensiones del triángulo puedan variar de tal forma que pueda incrementar o reducir el área que encierra, como también la forma que toma éste.

Dimensión 3 El simplex es de 4 vértices, y forma un tetraedro en el espacio 3D. Se forma un tetraedro y no necesariamente tiene que ser una forma regular. Los vértices podrían expandir, contraer, reflejarse en sentido contrario, expansión o contracción con reflexión.

Más dimensiones El simplex es de $N + 1$ puntos dentro de un espacio de N dimensiones. En este caso ya no se puede imaginar de manera geométrica la forma del simplex, pero el método sigue funcionando según los mismos principios.

En la figura 1.12 se muestra la descripción del simplex en su forma inicial para el caso de 3 dimensiones. a) el punto P_4 en el original es el peor punto. Se busca que el nuevo P_4 tendría un mejor valor y es por ello que se muestra una reflexión del vértice. b) si el punto P_4 es el mejor, entonces se asume que la dirección de búsqueda es la correcta entonces se trata de ampliar la distancia. c) si P_4 no fue mucho mejor, se puede decidir de reducir más bien el tamaño del paso. En caso que ninguna de las opciones anteriores mejora el valor en el punto P_4 , el simplex se contrae alrededor del mejor punto que en este caso es P_3 .

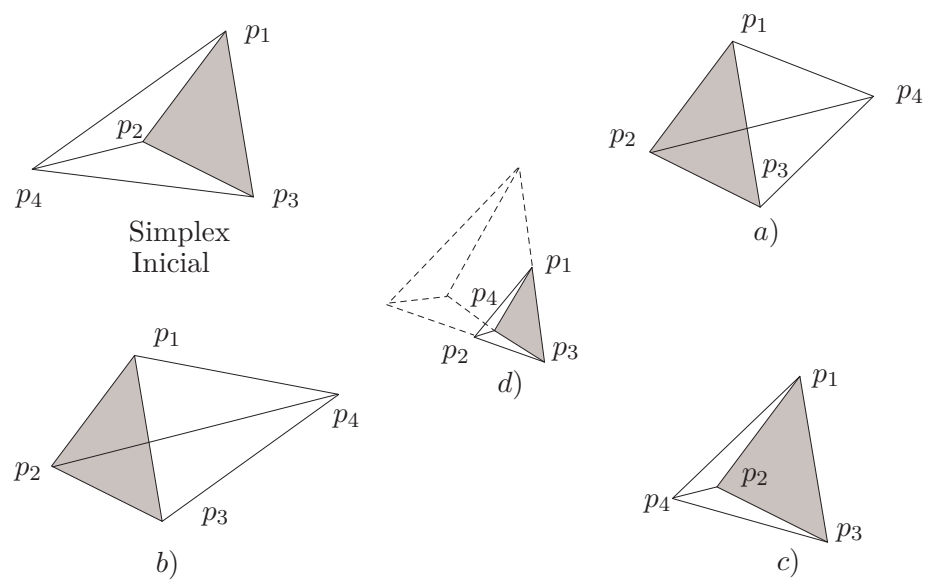


Figura 1.12: Descripción gráfica del Simplex (J. A. Nelder, 1965).

Capítulo 2

Desarrollo del método de procesamiento de imágenes por optimización

En el capítulo anterior, en la sección 1.1 pag. 7, se describía a groso modo la metodología que se emplea en este trabajo. Ahora, se muestra en forma más detallada la metodología que ayuda a entender el proceso a realizar, usando el diagrama en la figura 2.1.

Para iniciar la metodología se debe proporcionar información en el proceso de optimización. Datos del caso de estudio se refiere a proporcionar información que establece si es un tablero de ajedrez para calibración o láseres para determinar posición y orientación del robot respecto a la superficie. La información estimada, en el diagrama, es usada para generar una imagen virtual con una primera aproximación de posición y orientación de robot y posteriormente llevarla a la comparación con la imagen real de cámara. De la comparación realizada, se estiman resultados que pueden ser satisfactorios o no, para el caso en que no son buenos los resultados se genera una nueva imagen con nuevos parámetros propuestos por el optimizador y unos criterios establecidos, para posteriormente volver a realizar la comparación entre imágenes. La comparación se realiza de forma cíclica hasta que las variables convergen a un valor y que cumplen con los criterios establecidos, una vez terminado el proceso de comparación se asume que la información en las variables de interés ha convergido al valor buscado en la imagen real.

Sin perder el objetivo del proceso de optimización, en las secciones posteriores, se desarrollan y describen cada uno de los bloques que intervienen dentro de la metodología que se implementa.

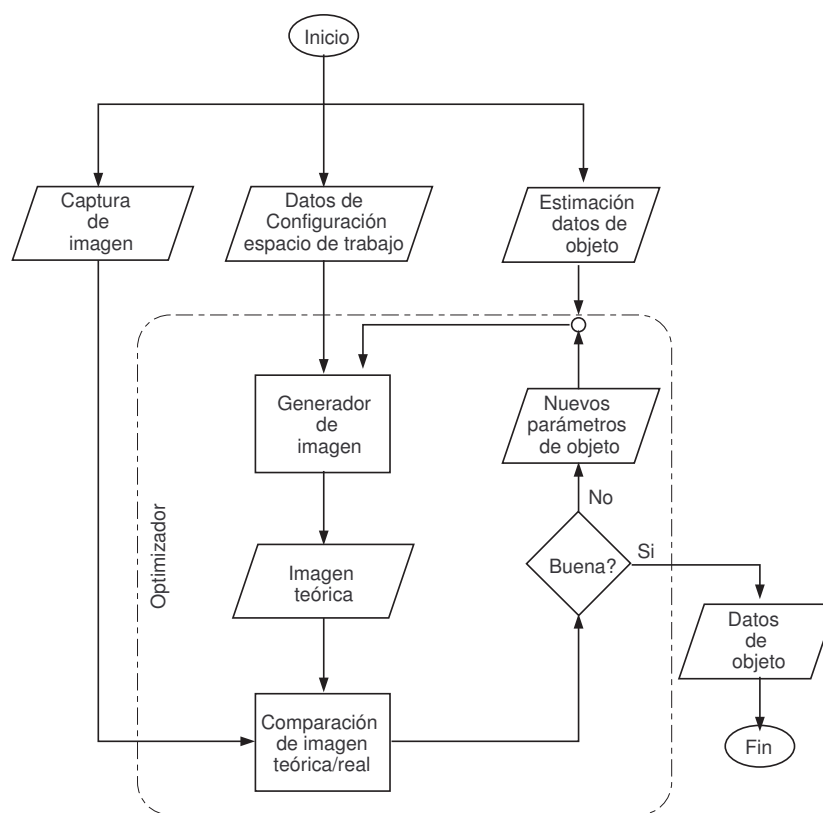


Figura 2.1: Generación de imágenes.

2.1. Generador de imágenes

Se desarrolló una función que permita albergar una serie de bloques con diferentes funciones, que en conjunto permita generar una imagen teórica.

El concepto de imagen teórica estará orientada a un resultado de imagen que se derivará de una *suposición de información estimada del espacio de trabajo* y que será proporcionada al generador de imagen.

La función del “Generador de Imagen” está compuesto por tres bloques de trabajo y se pueden identificar en el diagrama de la figura 2.2.

El primer bloque que constituye la función es “generador de puntos”, el cual permite generar una listado de puntos, cada punto en esta lista contiene tres coordenadas (x, y, z) que describen un punto en el espacio, más aparte se asigna una intensidad luminosa a cada punto. El segundo bloque contempla un “modelo de cámara (Pinhole)” que permite hacer una transformación de los puntos en el mundo real 3D a un espacio de imagen 2D, es decir, hacer pasar la lista de puntos generados con el bloque anterior sin tomar en cuenta la intensidad luminosa de cada punto; de esta forma se obtiene un listado de puntos en coordenadas en dos dimensiones (u, v) con su intensidad luminosa. Finalmente,

un bloque denominado “ensamblaje de puntos” es utilizado para juntar la lista de puntos y conjuntarlos en una imagen, para después aplicar uno de los dos métodos de filtrado propuestos sobre la imagen y generar la imagen final teórica.

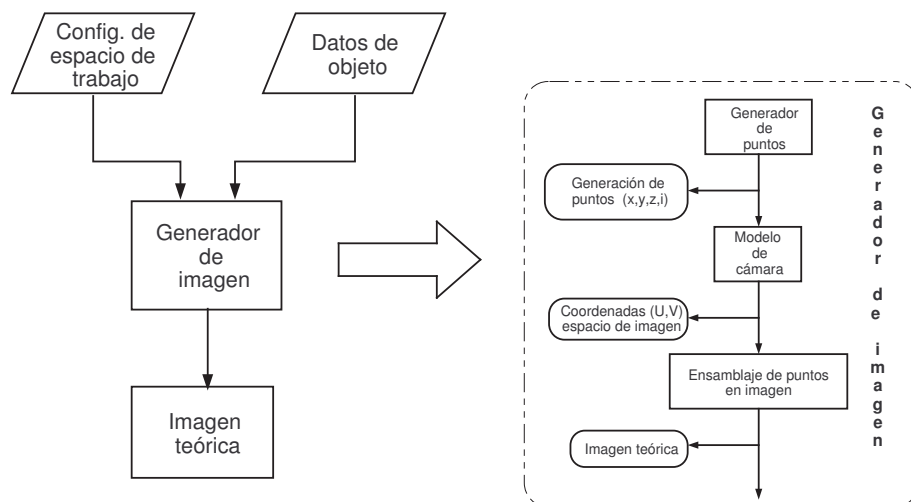


Figura 2.2: Generador de imagen.

En la imagen de la figura 2.2 se puede observar en diagrama de bloque del *generador de imagen*, éste bloque tiene dos entradas de información, *Configuración de espacio de trabajo* y *Datos de objeto*. La primera tiene información invariante del sistema durante el procesamiento de imágenes y contiene información de posición de los láser, ángulo proyección de los mismos, parámetros de cámara y otros más, mientras que la segunda entrada contiene información sobre el objeto, es decir, posición del plano de trabajo, orientación por mencionar algunos.

Paulatinamente se estarán abordando cada uno de los bloques que integran la función y serán desarrollados cada uno de ellos, es decir, se mostrará la forma del funcionamiento.

2.1.1. Generador de puntos

Este bloque de trabajo permite albergar dos funciones por lo pronto, pero en un futuro se podría considerar programar otro caso alternativo y con facilidad se puede integrar en este bloque.

Como se había descrito en la sección anterior, en esta función se buscó crear un listado de puntos que lograra contener cuatro datos que los describieran en su totalidad. Tres datos destinados a ser una coordenada en el espacio denotados por (x, y, z) , y un cuarto dato i para asignarle una intensidad luminosa que, dependiendo del caso a seleccionar, se asigna intensidad de luz blanca u oscuro.

Las dos funciones desarrolladas son:

- Intersección de láser de línea con una superficie plana de proyección.
- Tablero de ajedrez.

El primer caso contempla un listado de puntos para generar una imagen que contiene la proyección de los tres láser de línea sobre una superficie. El segundo caso consiste en generar un listado de puntos que en conjunto forman un tablero de ajedrez, el cual es usado para un proceso de calibración de cámara.

Cada tema será tratado de forma más abundante en las siguientes secciones.

2.1.1.1. Intersección de láser de línea con una superficie

Como se ha revisado en el planteamiento del problema, se da énfasis a la proyección de luz estructurada (láser de tipo línea) sobre una superficie para poder obtener información para un fin en particular. Saliendo un poco de contexto sobre el planteamiento general del problema, se toma particular atención en la línea que se genera con el láser, ya que es la médula principal de desarrollo del trabajo y los temas que se derivan de esto son tratados a continuación.

Para poder dar inicio con el trabajo de estudio es necesario considerar algunas definiciones simples. A partir de ellas se podrán realizar planteamientos que ayudarán a enfocar el trabajo que se realiza. Y para ello se recurre al concepto de línea, que puede ser encontrada en cualquier libro de texto.

Línea recta una línea esta conformada por una sucesión continua de puntos finitos, de la cual solo puede tener dos sentido y una orientación.

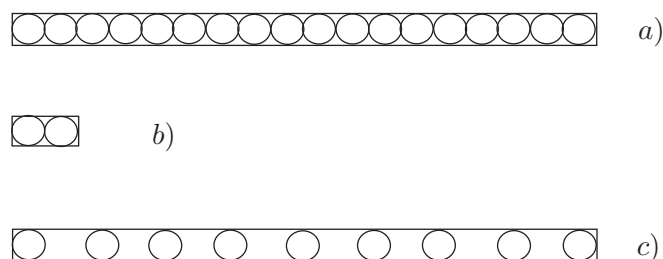


Figura 2.3: Descripción de la línea recta.

De la figura 2.3 se visualiza una línea que puede ser representada por una serie de puntos desde un inicio hasta un final, como lo muestra *a)*. En *b)* una sección de una recta también puede ser representada por un par de puntos y se sigue cumpliendo la definición. En *c)* se representa una serie de puntos que son distribuidos dentro de un segmento de recta y como tal, puede ser considerado como una línea recta la alineación de los mismos.

Inicialmente se cuenta con diodo emisor de luz láser, figura 2.4 a), después en su extremo se coloca un módulo óptico que convierte el rayo en un abanico de luz que, al ser proyectada sobre una superficie, forma una línea de láser b).

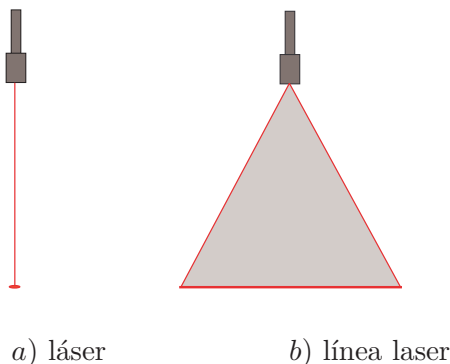


Figura 2.4: Haz de luz láser.

A continuación se hacen un par de consideraciones sobre el láser para darle un punto de vista desde la geometría analítica:

1. El abanico del láser es considerado como un plano.
2. Cada abanico es considerado como una sucesión finita de rayos de luz con separación equidistante emanando desde un mismo origen.

En la figura 2.5 se puede observar el abanico del láser, en a) aparece un par de vectores unitarios, $\vec{d}_l$ que emanan desde un punto hacia un punto de proyección sobre la línea. A este vector se le denominará *dirección de láser*. El segundo vector es $\vec{d}_{hl}$, nombrado como *abanico del láser* y es ortogonal al primero. En b) aparece un escalar σ que, al ir creciendo, permite hacer una suma del producto de los vectores perpendiculares y sus parámetros para describir un plano. De esta manera se busca representar todos los vectores que emanan desde un origen y se proyectan sobre la superficie para generar un láser de línea ¹.

Cuando se realiza una proyección de los planos de luz sobre una superficie cual quiera (curva o plano), se vuelve en una resolución de problema de geometría analítica, donde al intersectar un plano con la superficie en el espacio, se origina una curva en el espacio.

Ahora se recurre a definiciones formales de geometría analítica, que es aplicable para un rayo que pertenece al abanico del láser:

Definición de línea: Se definen dos vectores cualesquiera $\vec{d}_{la}$ y $\vec{r}_l$ como vectores en el espacio Euclidiano tridimensional (E^3) con ($\vec{d}_{la} \neq 0$). Se le denomina línea y es

¹Las operaciones matemáticas que se realizan entre los dos vectores mencionados ($\vec{d}_l$, $\vec{d}_{hl}$), junto con el escalar σ , dan como resultado un vector denotado como $\vec{d}_{la}$ al cual hace referencia posteriormente en la definición de línea.

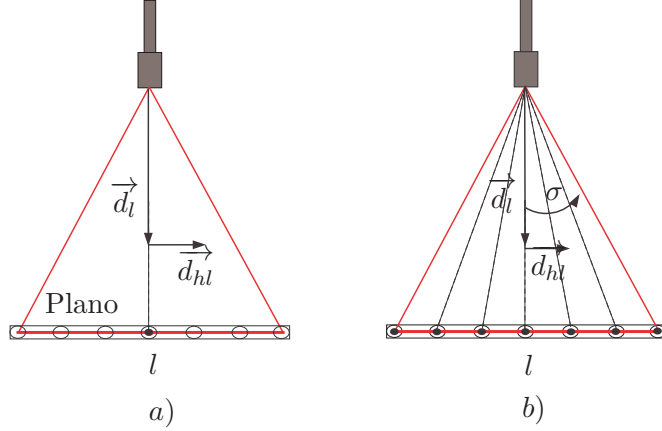


Figura 2.5: Consideración para el láser. a) descripción del vector para un punto que se encuentra en la línea, b) consideración para todos los puntos que representan la línea del laser.

paralela a $\vec{d}_{la}$, al conjunto de los $\vec{Z}_{lin}$ de E^3 que puedan representarse mediante la expresión:

$$\vec{Z}_{lin} = i_l \vec{d}_{la} + \vec{r}_l \quad (2.1)$$

Que dicho esto de otra forma, la ecuación (2.1) es llamada la ecuación paramétrica de la recta. Si el valor de i_l hace que $\vec{Z}_{lin}$ engranda la recta sobre la recta real ($-\infty < i_l < \infty$) (Lipschutz, 1970).

Para el plano sobre el cual se hace la proyección de los haces de luz se hace una definición:

Definición de plano: Se denomina *plano* que pasa por k y es paralelo a dos vectores independientes $\vec{d}_{p1}$ y $\vec{d}_{p2}$ al conjunto de los $\vec{Z}_{pla}$ de E^3 que se pueden representar mediante la expresión:

$$\vec{Z}_{pla} = i_p \vec{d}_{p1} + j_p \vec{d}_{p2} + \vec{r}_p \quad (2.2)$$

También conocida como ecuación paramétrica del plano, y $\vec{Z}_{pla}$ se engranda al plano si los parámetros i_p, j_p varían de forma independiente ($-\infty < i_p, j_p < \infty$) en el conjunto de los números reales (Lipschutz, 1970).

Analizando la figura 2.6 se puede observar la distribución del láser en el cabezal montado sobre la cámara, en la lateral izquierda de la figura, visto desde la parte superior del esquema. También, se muestra el diseño o esquema de análisis, que es usado para la deducción de ecuaciones, que resuelven el problema planteado de intersección de láser de línea con un plano.

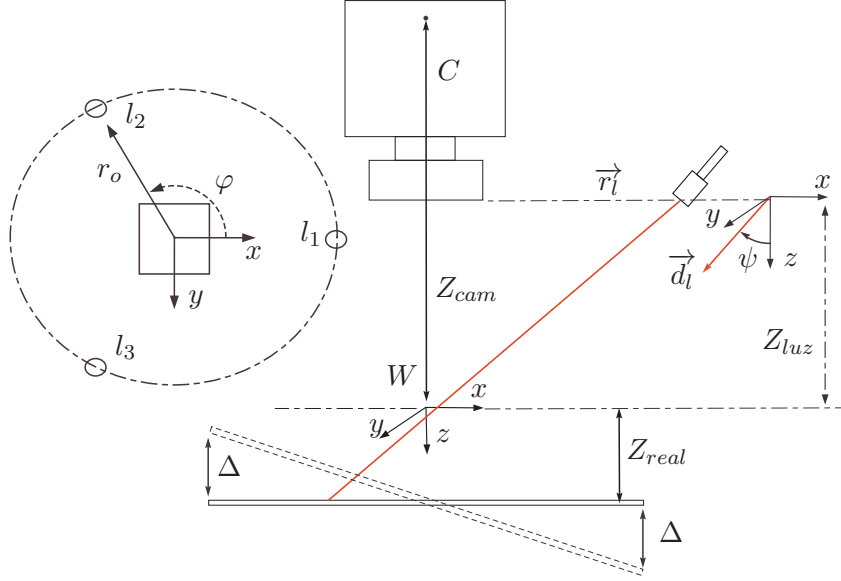


Figura 2.6: Diagrama de análisis del planteamiento del problema.

En la figura 2.6 se establece un sistema de referencia absoluto en W con (x, y, z) . A partir del origen, sobre el eje focal de cámara, se tiene una distancia a la cámara Z_{cam} y cuyo valor puede ser arbitrario sin problema alguno. De hecho, el punto C no necesariamente se encuentra cerca a la cámara, porque la lente puede mover la “pupila” muy lejos de la cámara física. A partir del eje focal de cámara, se puede medir la distancia y posición del láser, este valor se determina con $\vec{r}_l$ a través de:

$$\vec{r}_l = \begin{bmatrix} r_{lx} \\ r_{ly} \\ r_{lz} \end{bmatrix} = \begin{bmatrix} r_0 \cos(\varphi) \\ r_0 \sin(\varphi) \\ -Z_{luz} \end{bmatrix} \quad (2.3)$$

Es conveniente mencionar que las componentes en x e y permiten describir la distribución de los láser alrededor del eje focal de cámara. Del sistema de referencia absoluto W , se puede medir la distancia a la cual se encuentran los láseres, con $-Z_{luz}$.

En la imagen de la figura 2.5 se describieron los vectores unitarios $\vec{d}_l$, $\vec{d}_{hl}$ y el escalar σ . Ahora, para definir la dirección del haz de luz $\vec{d}_l$, se puede describir por medio de la siguiente ecuación (2.4):

$$\vec{d}_l = \begin{bmatrix} 0 \\ 0 \\ \cos(\psi) \end{bmatrix} - \sin(\psi) \begin{bmatrix} r_{lx} \\ r_{ly} \\ 0 \end{bmatrix} \frac{1}{r_0} \quad (2.4)$$

donde el primer término es proyección sobre el eje z con respecto al ángulo ψ que se forma y solo hay componentes en z . El segundo término es el complemento de la proyección sobre

el eje que a su vez es normalizado por $\frac{1}{r_0}$ y por la dirección del vector respecto al marco de referencia es negativo.

Ahora bien, como $\vec{d}_{hl}$ es perpendicular a $\vec{d}_l$, se puede escribir como en la ecuación (2.5):

$$\vec{d}_{hl} = \frac{1}{\sin(\psi)} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \vec{d}_l \quad (2.5)$$

con la única componente existente en eje z solo aparece el vector en esa posición y es normalizado a través de la proyección de $\sin(\psi)$.

La descripción del plano con estos vectores solo se realiza para abanico, es decir, desde el haz central hacia un extremo. Entonces se debe proponer una región para σ ($-0,5 \leq \sigma \leq 0,5$), si es necesario una línea corta entonces se debería de reducir el intervalo de la variable o viceversa en dado caso que se deseara incrementar la longitud de la misma.

Con esta información es posible describir la ecuación de línea como se establece en la definición de la misma, y para ello se hace uso de la ecuación (2.1):

$$\vec{Z}_{lin} = i_l \vec{d}_{la} + \vec{r}_l$$

$$\vec{Z}_{lin} = i_l [\underbrace{\vec{d}_l + \sigma [\vec{d}_{hl}]}] + \vec{r}_l \quad (2.6)$$

$$\vec{Z}_{lin} = i_l \underbrace{\begin{bmatrix} d_{lx} \\ d_{ly} \\ d_{lz} \end{bmatrix} + \sigma \begin{bmatrix} d_{hlx} \\ d_{hly} \\ d_{hlz} \end{bmatrix}} + \begin{bmatrix} r_{lx} \\ r_{ly} \\ r_{lz} \end{bmatrix} \quad (2.7)$$

Como el resultado que se encuentra asociado bajo las llaves es un vector, entonces se puede escribir la ecuación de manera simplificada, como se estableció en la definición de línea.

En forma paramétrica se tiene:

$$\vec{Z}_{lin} = i_l \begin{bmatrix} d_{lax} \\ d_{lay} \\ d_{laz} \end{bmatrix} + \begin{bmatrix} r_{lx} \\ r_{ly} \\ r_{lz} \end{bmatrix} \quad (2.8)$$

De esta forma se logra realizar la descripción matemática de la línea como un plano, haciendo uso de los vectores que se definieron en la consideración del plano.

Ahora es necesario establecer algunas condiciones para la cámara y la superficie de proyección de los láser, donde existe un relación de paralelismo entre ellos. Esto debido a que el eje focal de la cámara siempre será normal al plano de la superficie y si bien se llega a detectar un cambio de posición u orientación de plano, entonces la cámara solo apreciará un cambio de profundidad en la imagen y variaciones en los láseres que se muestran en las imágenes; dicho de otra manera, solo existe una variación sobre el eje focal de cámara o eje z , así como se evidencia en la imágenes de la figura 1.5, 1.7 y 2.6.

De la definición del plano se toma la ecuación (2.2):

$$\overrightarrow{Z_{pla}} = i_p \overrightarrow{d_{p1}} + j_p \overrightarrow{d_{p2}} + \overrightarrow{k}$$

Como el plano será normal al sensor de cámara, las componentes tanto en x e y toman los valores como en la ecuación (2.9) ya que garantizan que estos dos elementos sean paralelos y solo en el eje en z se registran variaciones o incrementos. Con base en este criterio de apreciación, se puede establecer la ecuación de plano definida de la siguiente manera:

$$\overrightarrow{Z_{pla}} = i_p \begin{bmatrix} 1 \\ 0 \\ \Delta z_1 \end{bmatrix} + j_p \begin{bmatrix} 0 \\ 1 \\ \Delta z_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ z_{p0} \end{bmatrix} \quad (2.9)$$

$$\overrightarrow{Z_{pla}} = i_p \begin{bmatrix} 1 \\ 0 \\ z_{p1} - z_{p0} \end{bmatrix} + j_p \begin{bmatrix} 0 \\ 1 \\ z_{p2} - z_{p0} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ z_{p0} \end{bmatrix} \quad (2.10)$$

z_{p0} esta definido por Z_{real} , z_{p1} y z_{p2} es la posición del plano tomando como referencia el origen. Δz es el incremento de cambio de posición o pendiente del plano.

De esta manera es como queda descrita la ecuación de línea generada por el láser y la superficie en la cual son proyectados los haces de luz.

Para encontrar la intersección entre el plano que se genera de la línea y la superficie sobre la cual se proyectan, se debe de cumplir que:

$$\overrightarrow{Z_{pla}} = \overrightarrow{Z_{lin}} \quad (2.11)$$

$$i_p = i_l d_{lax} + r_{lx} \quad (2.12)$$

$$j_p = i_l d_{lay} + r_{ly} \quad (2.13)$$

$$i_p \Delta z_1 + j_p \Delta z_2 + z_{p0} = i_l d_{laz} + r_{lz} \quad (2.14)$$

$$(2.15)$$

Usando la ecuación (2.11) e igualando los términos paramétricos del plano con la línea recta se obtiene:

$$i_p = i_l d_{lax} + r_{lx} \quad (2.16)$$

$$j_p = i_l d_{lay} + r_{ly} \quad (2.17)$$

$$i_p \Delta z_1 + j_p \Delta z_2 + z_{p0} = i_l d_{laz} + r_{lz} \quad (2.18)$$

Sustituyendo (2.16) y (2.17) en (2.18):

$$(i_l d_{lax} + r_{lx}) \Delta z_1 + (i_l d_{lay} + r_{ly}) \Delta z_2 + z_{p0} = i_l d_{laz} + r_{lz} \quad (2.19)$$

Y obteniendo la solución para i_l de (2.19):

$$i_l = \frac{r_{lz} - r_{lx}\Delta z_1 - r_{ly}\Delta z_2 - z_{p0}}{d_{lax}\Delta z_1 + d_{lay}\Delta z_2 - d_{laz}} \quad (2.20)$$

donde i_l es el escalar que permite relacionar la posición de cada punto que se encuentra dentro del láser de línea y el plano o superficie donde se proyecta el láser (intersección). Ya que se tiene el valor de i_l , se puede sustituir dicho valor en la ecuación (2.8) y de esta manera, se conocerían los valores en las tres coordenadas de (x, y, z) del punto de intersección entre la línea y el plano, por defecto se le asigna una intensidad luminosa de 1, es decir, por cada coordenada se genera un punto de luz blanca.

Es importante recordar que la intersección encontrada es única y exclusivamente para un punto en la línea y para un solo láser de línea. Si la longitud de la línea proyectada por el laser se decide representar por n puntos y si son tres laser los que se usan, se debe de determinar la intersección según este método en $3n$ ocasiones.

Es así como se obtiene la lista de puntos que son empleados para generar la imagen de la intersección de láser de línea con la superficie de proyección.

Para visualizar de manera gráfica la resolución del sistema de ecuaciones, se muestra la imagen en la figura 2.7:

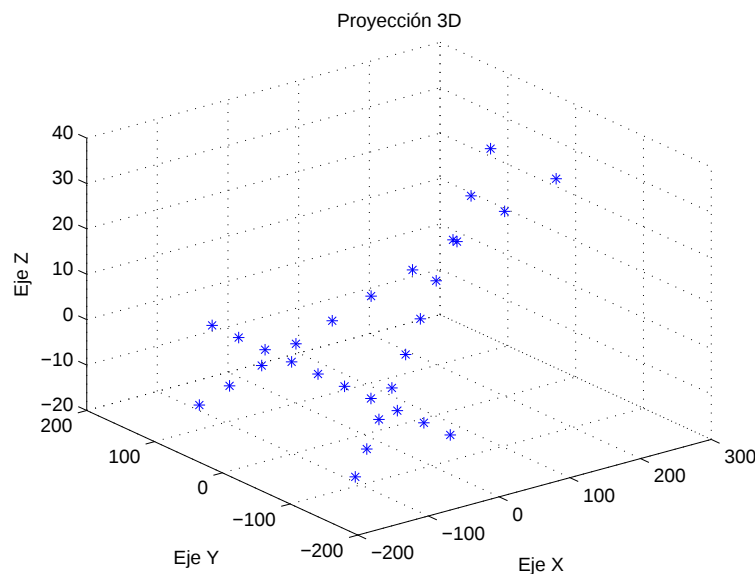


Figura 2.7: Intersección de los tres láser sobre una superficie con una inclinación.

Finalmente, en la gráfica mostrada en la figura 2.7 se puede visualizar perfectamente dicha lista de puntos que se extraen de la solución matemática; también se puede apreciar una pendiente que se le da a la superficie de proyección y se puede observar el tipo de imagen que se genera como si los láser se estuvieran proyectando.

Si se usara mayor número de puntos para representar la línea del láser, se lograría observar una línea más uniforme, pero como resultado demostrativo de la formulación y solución del problema se deja en este nivel de resolución.

2.1.1.2. Generador de tablero de ajedrez

Como característica del generador de imágenes se tiene la versatilidad de poder seleccionar entre varios casos contemplados. Uno de ellos es un tablero de ajedrez y se integra esta opción para un proceso de calibración de cámara que, en temas posteriores, será abordado.

Para poder generar un tablero de ajedrez, se usan dos variables. La primera variable es denominada como *Número Campos* (Nc), que determina el número de cuadros que debe de contener el tablero (oscuros y blancos) y debe de ser cuadrado, por tanto, el tablero es de dimensión de $(Nc \times Nc)$. Cada campo será definido con una asociación de puntos de la misma dimensión y se denotan como *Número de Puntos por Campo* (Npc). La visualización del tablero quedaría como lo muestra el esquema siguiente mostrado en la figura 2.8.

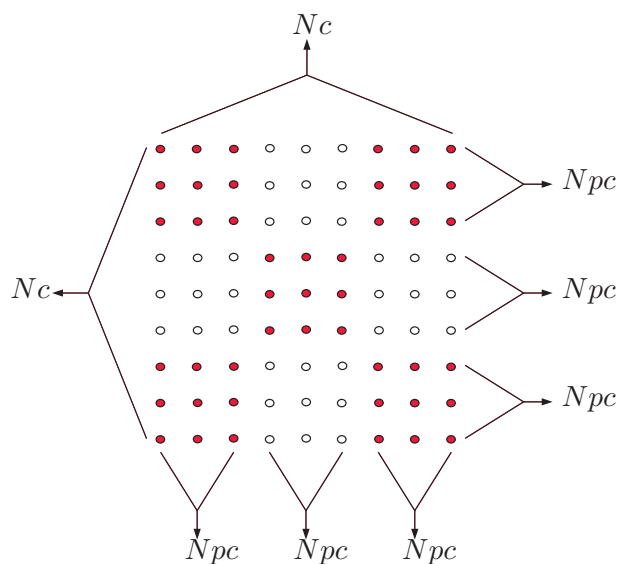


Figura 2.8: Diseño del tablero de ajedrez.

Al realizar un arreglo matricial de puntos de $Nc \times Npc$ se puede obtener la dimensión del tablero con todos los puntos. Una vez que son agrupados los Npc se les asigna una intensidad luminosa de 1 para luz blanca y de -1 para oscuro. En la figura 2.9 se puede observar la generación del tablero de ajedrez.

Se genera el tablero de ajedrez por medio del arreglo matricial de puntos con coordenadas (x, y) como es mostrado en la figura 2.9 a). Seguido de esto se le realiza un recorrido

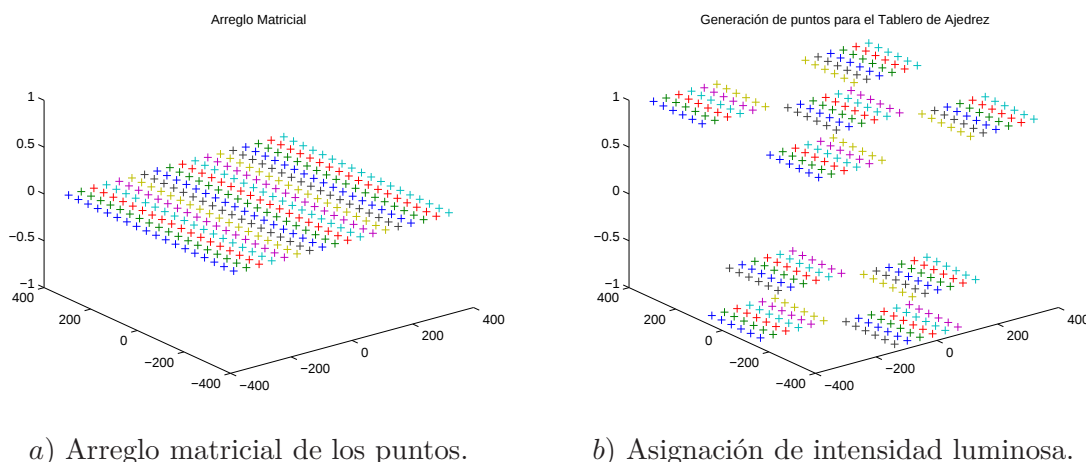


Figura 2.9: Tablero de ajedrez en su forma primaria.

de cada elemento de la lista y, según la condición establecida para Npc , se le asigna una intensidad luminosa de 1 para una región blanca o -1 para una región oscura hasta terminar con cada elemento del arreglo. Finalmente, una vez que han sido discretizados todos los elementos de la lista de puntos, se le asigna una posición sobre el eje z .

El resultado que se sigue obteniendo de esta lista de puntos son datos que contienen (x, y, z, i) ; los tres primeros valores corresponden a una posición y el último que corresponde a una intensidad luminosa.

2.2. Modelo de cámara (Pinhole)

Ahora que se han generado la lista de puntos para el caso de “intersección de láser de línea con una superficie ” y el “tablero de ajedrez” donde cada punto contiene (x, y, z, i) es necesario hacer pasar la información a través de un modelo de cámara de “pinhole”, el cual es usado para proyectar un punto x, y, z del espacio $3D$ a un espacio (u, v) de imagen $2D$.

Este modelo considera que hay rayos que provienen de un objeto y que deben de pasar a través de un orificio muy fino para que finalmente sean proyectados sobre el sensor de imagen de forma invertida. Como esta consideración no es uniforme para las lentes que intervienen en los modelos de cámara (modelo no lineal), deben de integrarse variables que ayuden a modelar en lo mejor posible el sistema.

Se puede mostrar el modelo a través de la figura 2.10, donde se proyecta un haz de luz de un objeto que se hace pasar por el orificio o centro de cámara hasta proyectarla sobre un plano o sensor de imagen; se puede observar la inversión de la imagen que se da después de pasar por el orificio de cámara.

De aquí puede obtener que el factor de escala de una imagen a otra está dada por las

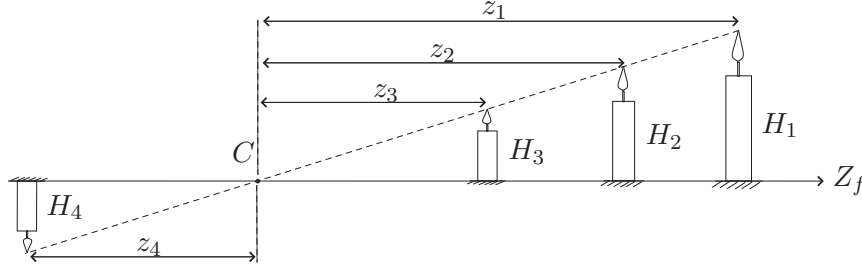


Figura 2.10: Proyección de imágenes por modelo de cámara de Pinhole.

siguientes relaciones:

$$\frac{H_1}{z_1} = \frac{H_2}{z_2} = \frac{H_3}{z_3} = \frac{H_4}{z_4}$$

El factor de escalamiento de las imágenes también se puede observar por medio de triángulos semejantes y la relación que existe entre H_3 y H_1 solo se debe al factor de escalamiento.

La representación del modelo de cámara de pinhole se puede ver en la figura 2.10, donde C es el orificio por donde pasan todos los rayos de luz del objeto y, mientras más fino es la imagen se vuelve más nítida y definida. Al reducir en demasía el orificio sucede también que la imagen tiende a hacerse más oscura.

Usualmente suele colocarse el centro de proyección de cámara sobre el eje focal de cámara o eje axial y podría considerarse como constante pero es a priori desconocido. El plano de imagen que se genera suele ser perpendicular al eje focal de cámara y se encuentra a una distancia igual a la longitud focal del objeto y, para evitar la inversión de imagen, podría colocarse antes del centro de proyección. Se denomina como *centro óptico* al punto donde coincidencia del eje focal y el plano de imagen.

A partir del modelo que se presenta en la figura 2.11 se pueden obtener las ecuaciones que describen el modelo de cámara.

Donde C es el centro óptico de cámara, x_c y y_c corresponden a las coordenadas de la cámara, $(u, v)_I$ es el sistema coordenado en la imagen que usualmente se encuentra situado en uno de los extremos del sensor de cámara. $(u, v)_{I'}$ se usa para cambiar la referencia al centro de la imagen; $(x, y, z)_w$ se refiere a las coordenadas de referencia mundial; z_i es la distancia a la imagen; Z_{cam} que es la distancia a la cual se encuentra la cámara al objeto de proyección; P_{xyz} es el punto en espacio mientras que p es el punto donde se proyecta en la imagen.

$$(x, y, z)_C = (x, y, z)_w + (0, 0, Z_{cam}) \quad (2.21)$$

Con las ecuaciones del modelo de cámara se puede obtener la relación de transformación del mundo real al espacio de imagen y, por tanto, se puede transformar el listado de puntos que se generó en la función anterior del espacio $3D$ al espacio de imagen de cámara en $2D$. Ya que se tienen las tres coordenadas (x, y, z) omitiendo para esta función la intensidad luminosa, el listado de puntos en una imagen quedaría como en la imagen de la figura 2.13.

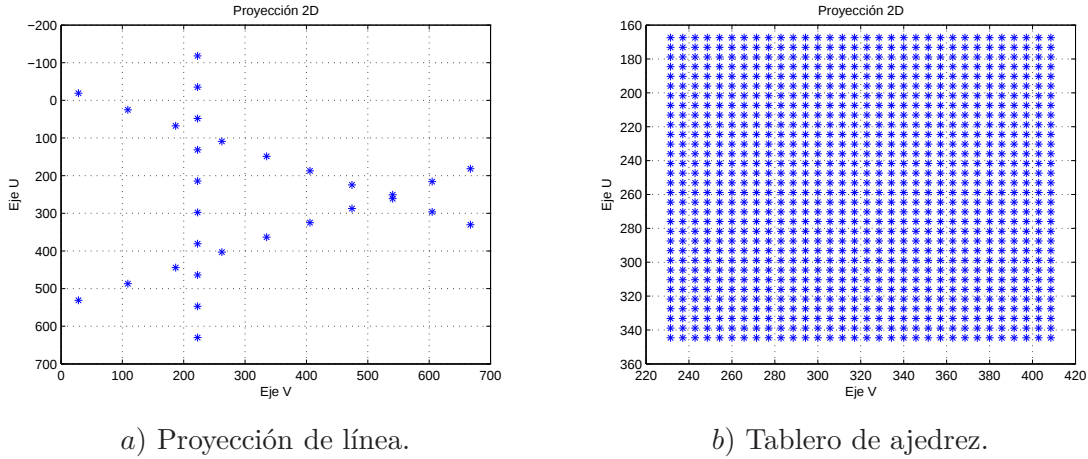


Figura 2.13: Puntos proyectados en espacio de cámara 2D.

En la figura 2.13 se muestra la proyección de los puntos tanto en el caso de proyección de línea como el tablero de ajedrez.

2.3. Ensamblaje de puntos en una imagen

En este nuevo bloque, se desarrolla una función que permita conjuntar la lista de puntos, los que se generaron en el bloque anterior, en un imagen. Para esto se propone el uso de dos métodos:

- Puntos Gaussianos
- Método de 4Pixel

El método de “Puntos gaussianos ” utiliza una función de distribución de energía gaussiana y el método de “4Pixel” concentra la energía en cuatro pixeles por medio de una matriz y después, se aplica un filtro de tipo cónico ó gaussiano para distribuir la energía de los cuatro pixeles y ambos métodos serán evaluados en tiempo. Se busca que sean versátiles y ventajas de uno sobre el otro y establecer condiciones de uso.

En la figura 2.14, se muestra la función del “Generador de Imágenes” y sus estructura general. Contempla las dos opciones para generar imágenes y los métodos ensamble de puntos, los cuales se revisarán a continuación.

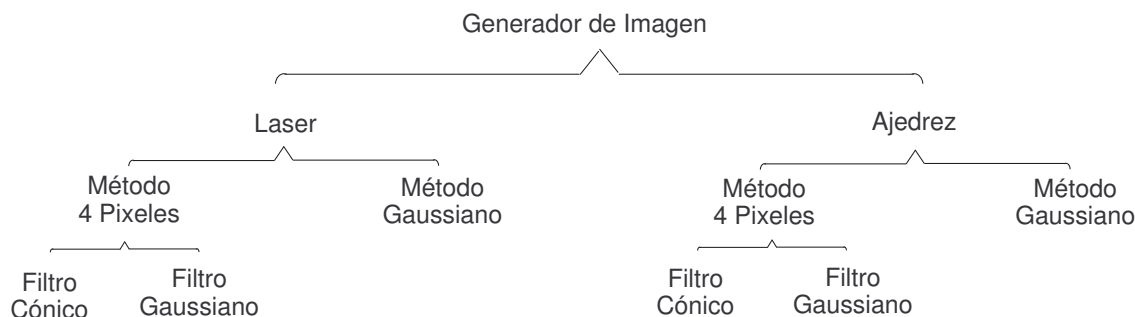


Figura 2.14: Árbol del generador de imágenes.

2.3.1. Método 1: *Puntos Gaussianos*

En estadística y probabilidad se llama distribución normal, distribución de Gauss o distribución gaussiana a un modelo matemático que trata de describir como se comporta una variable dada que es continua y dicha función es mostrada en la figura 2.18.

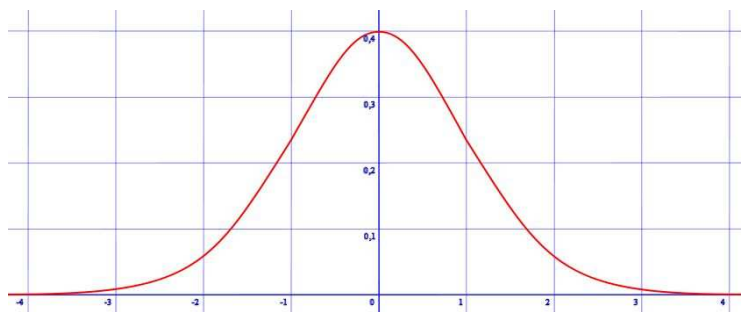


Figura 2.15: Campana de Gauss o distribución normal.

Para cada punto que se genera, es necesario aplicar la función gaussiana para que se pueda conformar una imagen independientemente si es una línea de un láser o un tablero ajedrez (ver figura 2.17). En la figura 2.16 se toma como caso representativo la forma en que se ajusta el método a un láser de línea.

En la figura 2.16 se puede observar cómo se ajustaría la curva de campana sobre la línea del láser. En esta imagen el gradiente de luminosidad del láser pasa de un 0 a un 1 muy abruptamente mientras que la función de Gauss es muy suavizado el gradiente luminoso. El ajuste de la función permitirá en gran medida obtener buenos resultados para la detección y representación de línea que se generará con el método.

Para este planteamiento, la función gaussiana es considerada como una distribución de energía alrededor de un punto de interés y puede ser descrita a través de las ecuaciones

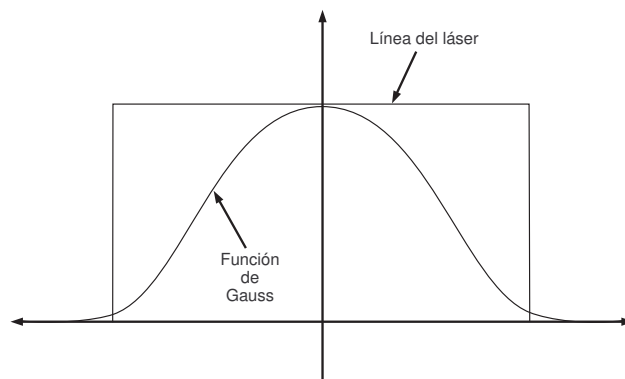


Figura 2.16: Función de Gauss.

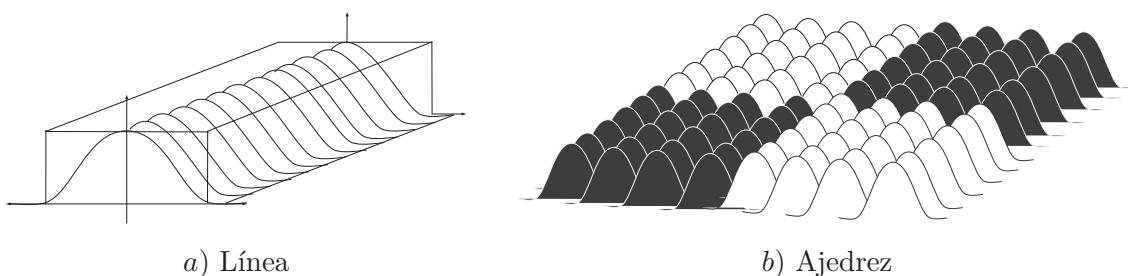


Figura 2.17: Representación de imágenes por medio de la función de Gauss.

que se plantean a continuación.

Primero, es necesario generar dos vectores de longitud n y m los cuales describen la dimensión de la imagen que se debe de generar.

$$x = [1, 2, 3 \dots n] \quad (2.26)$$

$$y = [1, 2, 3 \dots m] \quad (2.27)$$

Después se genera un par de arreglos con X e Y .

$$X = x' * [1, 1, 1 \dots m] \quad (2.28)$$

$$Y = [1, 1, 1 \dots n]' * y \quad (2.29)$$

Para calcular el punto de interés sobre la imagen, se usan los arreglos X e Y y se les resta la coordenada del punto donde será situado el punto.

$$r = \sqrt{(X - x_c)^2 + (Y - y_c)^2} \quad (2.30)$$

donde x_c e y_c son las coordenadas del origen para establecer la cresta de la función gaussiana. También hay que mencionar por este cálculo afecta a toda la imagen que se genera con los arreglos de X e Y , es decir, si la cresta de la función se localiza en el centro de imagen

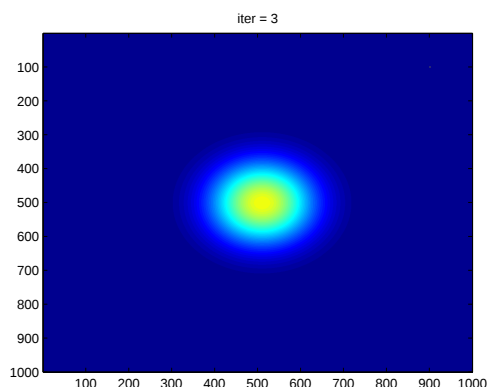
entonces la distribución de energía afectaría hasta los contornos de la imagen, aunque dicha contribución de energía en estos pixeles es casi cero, debe de ser calcula por método.

Finalmente la ecuación (2.31) permite realizar la descripción de la función gaussiana.

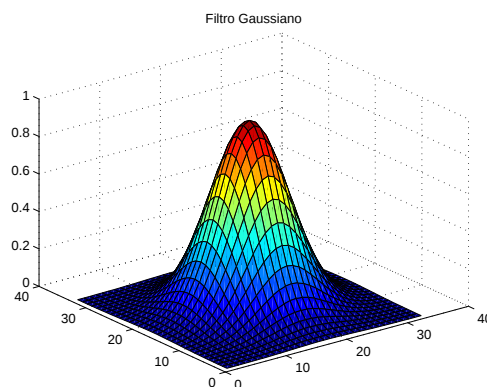
$$G = \exp \left(-f \left(\frac{r}{R} \right)^E \right) \quad (2.31)$$

con el factor f se tiene el control de intensidad gaussiana para r dentro de la imagen y E permite suavizar la función, es decir, abrir o cerrar la campana y si $E = 2$ es suave y conforme aumenta se puede obtener muy abrupto el cambio de luz.

En la imagen de la figura 2.18 *a*) se muestra la función gaussiana en forma distribución de energía para un punto mientras que en *b*) se visualiza la función como una curva de campana.



a) Imagen de un punto gaussiano.



b) Campana de Gauss.

Figura 2.18: Distribución de energía Gaussiana en una imagen.

La función que se describió anteriormente es aplicado para cada punto que se quiere conformar en la imagen final teórica, es decir, se hace una suma de distribución gaussiana para cada punto existente en la imagen. Entonces de la lista de coordenadas que se obtiene del modelo de cámara (u, v) son usados para hacerlos pasar a través de este modelo para generar la imagen.

Finalmente, el resultado de la imagen generada es mostrada en la imagen de la figura 2.19, donde se puede ver en *a*) los láser con 10 puntos por línea(10 pts por línea y son 3 láseres, entonces 30 veces se aplicó el método). En *b*) el tablero de ajedrez con 7 campos.

La desventaja que podría encontrarse en este método es que el un punto de Gauss muestra una contribución en todos los pixeles de la imagen, es decir, si fuese necesario colocar 30 puntos sobre la imagen, había 30 veces la contribución de la función sobre los pixeles de la imagen. Esto se refleja en un cálculo computacional muy elevado.



a) Intersección de láser y superficie.

b) Tablero de ajedrez.

Figura 2.19: Ensamble de puntos en una imagen.

2.3.2. Método 2: *4Pixel*

Se propone un método alternativo y diferente por la desventaja mencionada en el apartado anterior. La intención de éste es que puede llegar a ser más rápido y preciso que el anterior en cuestión de poder ubicar la posición exacta del punto a nivel de resolución de sub-píxel en la imagen, más aparte, que solo se afectan 4 píxeles por cada punto que se desea colocar en la imagen, en lugar de modificar toda la imagen.

La metodología consta de dos pasos sencillos, la primer etapa considera en usar cuatro píxeles de la imagen (los que encierran la posición exacta del punto en el píxel correcto) y concentrar toda la energía luminosa en ellos mediante una matriz; el segundo paso es aplicar una máscara como filtro con una operación de convolución para poder distribuir la energía concentrada de los cuatro píxeles alrededor del punto de interés.

Para la primera etapa que concentra la intensidad luminosa en cuatro píxeles, se hace uso de una matriz como la que se presenta a continuación:

$$A(i, j) = \begin{bmatrix} a_{i,j} & a_{i,j+1} \\ a_{i+1,j} & a_{i+1,j+1} \end{bmatrix} \quad (2.32)$$

donde $A(i, j)$ es la coordenada exacta del punto de interés en la imagen, la cual está dado por u y v de la función de modelo de cámara. Ésta contiene una parte entera y decimal, en la cual el entero es usado para ubicar el píxel en la imagen ($a_{i,j}$), así mismo, se usa un píxel posterior para encerrar la posición correcta ($a_{i,j+1}$) y esto es aplicable tanto para el renglón como en la columna de la imagen ($a_{i+1,j+1}$).

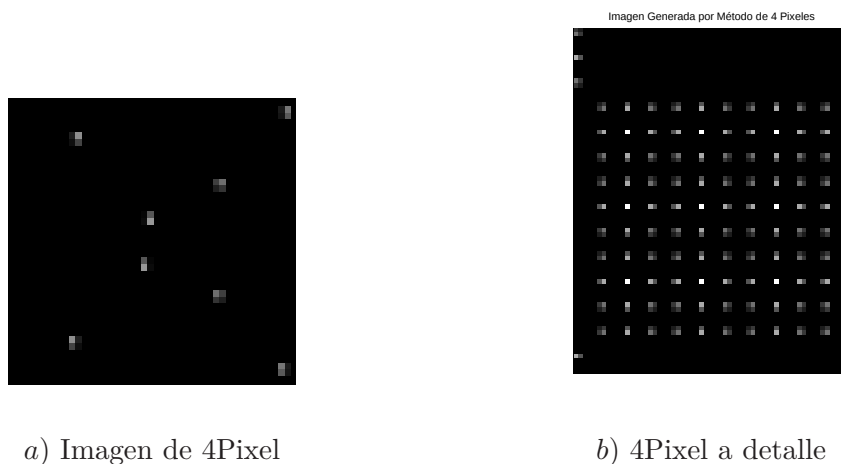
De un 100 % de energía que se tiene, ésta se reparte en los cuatro píxeles de la matriz por medio de la parte decimal usando un criterio de brazo de palanca, es decir, si cierto porcentaje de energía se encuentra concentrado en un píxel el otro complemento se encuentra en el próximo píxel. La parte decimal establece la exactitud a nivel sub-píxel lo que da la correcta posición del punto sobre la imagen.

La matriz con la que se asigna la intensidad luminosa es descrita a continuación en la ecuación (2.33):

$$A(u, v) = \begin{bmatrix} (1 - d_u) \\ (d_u) \end{bmatrix} \begin{bmatrix} (1 - d_v) & d_v \end{bmatrix} = \begin{bmatrix} (1 - d_u)(1 - d_v) & (1 - d_u)d_v \\ d_u(1 - d_v) & d_u d_v \end{bmatrix} \quad (2.33)$$

donde u y v es el pixel de interés (parte entera), d_u y d_v es la parte decimal o fraccionaria. El resultado de esta matriz es justamente el 100 % de energía repartida en los cuatro pixeles.

En la imagen mostrada en la figura 2.20, se puede observar el efecto provocado por la matriz de 4 pixeles para un punto en una imagen. En *a)* son mostrados unos puntos de la línea recta a detalle, mientras que en *b)* se muestra a detalle un campo del tablero de ajedrez.



a) Imagen de 4Pixel

b) 4Pixel a detalle

Figura 2.20: Imagen generada a través del método 4Pixel.

En cada imagen mostrada es posible ver cómo es la concentración de energía en los pixeles, en algunos se puede apreciar un pixel totalmente blanco y en otros se puede observar cuatro pixeles pintados con diferentes intensidades luminosas, las cuales se pueden interpretar de la siguiente manera:

Si la energía es más intensa en un pixel, el punto exacto coincide con la posición de un pixel. Si la energía se distribuye sobre los cuatro pixeles, entonces el punto exacto se encuentra repartido entre los pixeles de la imagen, pero de tal forma que el promedio con peso ponderado, según la intensidad luminosa, entrega la posición exacta del punto con una precisión o resolución a nivel sub-pixel.

Como paso subsecuente a este proceso donde se concentra la energía en cuatro pixeles, es necesario expandir o distribuir la energía de los cuatro pixeles alrededor del punto señalado. Para ello se aplica un filtro de difusión, lo cual se puede pensar como una operación de convolución entre la imagen y una máscara o distribución del filtro. Una convolución es un

operador matemático que transforma dos funciones f y g en una tercera función que en cierto sentido representa la magnitud en la que se superponen f y una versión trasladada e invertida de g . Una convolución es un tipo muy general de promedio móvil.

La operación de convolución está denotado de la siguiente manera:

$$h(t) = (f \star g)(t) = \int_0^t f(t - \tau)g(\tau)d\tau \quad (2.34)$$

donde $f(t)$ representa la imagen con cuatro pixeles y $g(t)$ es la máscara para aplicar a la imagen.

Se tiene la posibilidad de utilizar dos funciones como filtro:

- Máscara Gaussiana.
- Máscara Cónica.

La máscara Gaussiana ya se tiene descrita y aplicada anteriormente, solo es cuestión de redimensionar la función con algunos parámetros y es aplicable directamente, mientras que el filtro cónico se puede describir a continuación.

Primero es necesario generar dos vectores de longitud n y m los cuales describen la dimensión de la máscara, que para este caso es cuadrada.

$$x = [1, 2, 3 \dots n] \quad (2.35)$$

$$y = [1, 2, 3 \dots m] \quad (2.36)$$

Después se genera un par de arreglos con X e Y .

$$X = x' * [1, 1, 1 \dots m] \quad (2.37)$$

$$Y = [1, 1, 1 \dots n]' * y \quad (2.38)$$

Para calcular el radio de distribución de energía alrededor del pixel de interés, se usa la ecuación:

$$r = \sqrt{\left(\frac{X - (n + 1)}{2}\right)^2 + \left(\frac{Y - (m + 1)}{2}\right)^2} \quad (2.39)$$

Se establece el factor:

$$F_{actor} = \frac{n - 1}{2} \quad (2.40)$$

Y finalmente queda la máscara de la siguiente forma:

$$M_{ask} = 1 - \frac{r}{F_{actor}} \quad (2.41)$$

Así es como se describe la máscara cónica que puede ser usada para difuminar la energía alrededor del punto de interés que se genera por medio del método de 4Pixel.

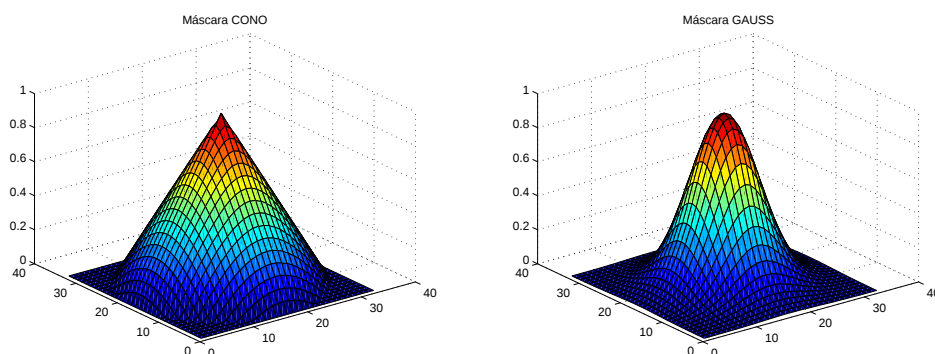


Figura 2.21: Máscaras para la convolución.

Ahora en la imagen de la figura 2.21 se puede visualizar los dos filtros gráficos empleados para la convolución y la generación de la imagen teórica final.

De las gráficas de los filtros se puede observar las diferencias entre máscaras, en el cónico se puede ver que la pendiente o gradiente de la función es lineal mientras que en la gaussiana es de tipo antilogaritmo, que corresponde a la función que los describe, además es posible ver que la gráfica cónica el centro es un punta mientras que en la función de gauss es plana, lo que permite señalar con mayor precisión el punto correcto de la ubicación en la imagen.

Una vez que se tiene el diseño de los dos filtros se realiza una prueba para un punto en la imagen y se aplica la operación de convolución para cada caso y el efecto que produce en la imagen es mostrado en la imagen de la figura 2.22.

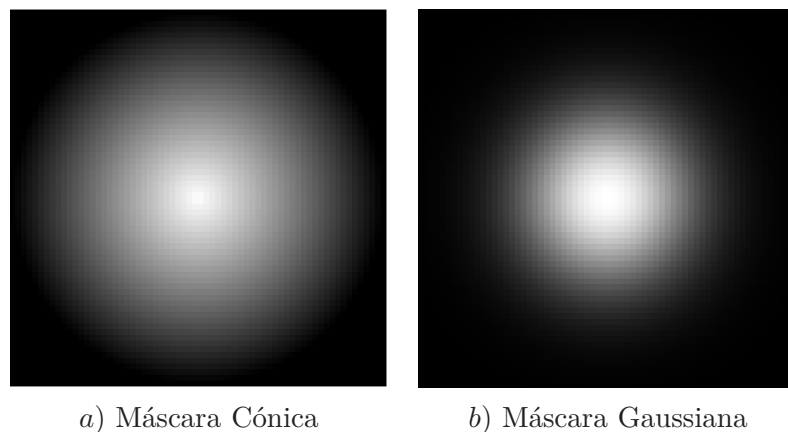
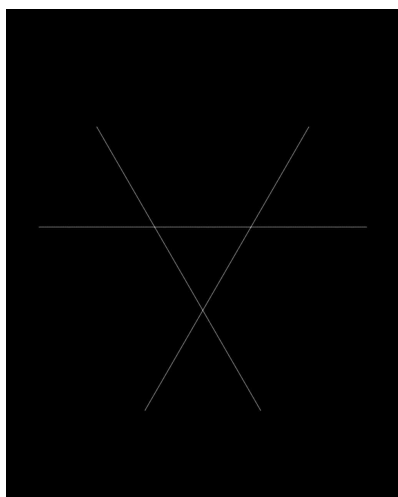


Figura 2.22: Máscaras para la convolución a detalle en la imagen.

Como se puede observar en las imágenes de la figura 2.22 los efectos son notorios en los puntos que se generaron, entonces en una sucesión de puntos donde la separación es mínima se podría observar una intensidad luminosa más marcada en el centro de línea con

el filtro cónico que con el gaussiano. Después en la sección de resultados podría analizarse de forma detenida para ver si esto representa alguna ventaja.

Finalmente, se muestran las imágenes generadas en las figuras 2.23 y 2.24 de las líneas del láser y un tablero de ajedrez, se incluye el caso para el filtro gaussiano y el cónico respectivamente.

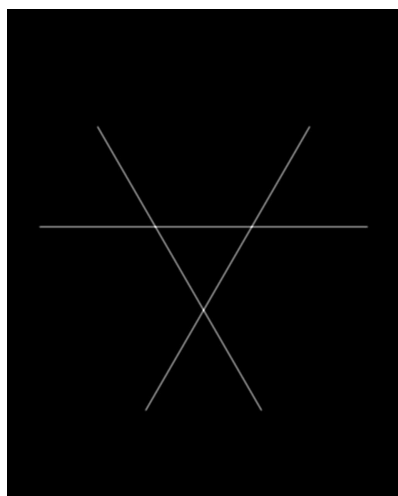


a) Láser.

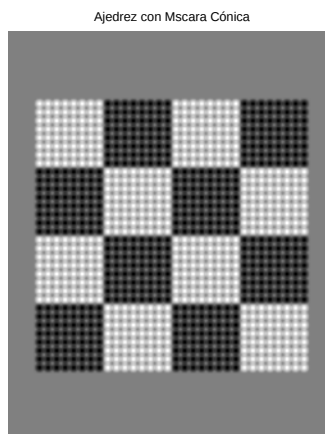


b) Tablero de ajedrez.

Figura 2.23: Imagen generada después del filtro de convolución Método 4Pixel y máscara de Gauss.



a) Láser.



b) Tablero de ajedrez.

Figura 2.24: Imagen generada después del filtro de convolución Método 4Pixel y máscara cónica.

2.4. Comparación de imágenes

Una vez que se ha generado la imagen virtual, ésta se puede llevar a un proceso de comparación con la imagen real. Ver diagrama de la página 22, sección 2.1. Para explicar la comparación entre imágenes, se recurre al uso de vectores y el *producto punto* o *interno*.

Cuando se definen dos vectores cualesquiera $\vec{u}$ y $\vec{v}$, el producto interno entre ellos se define mediante:

$$\vec{u} \bullet \vec{v} = |\vec{u}| |\vec{v}| \cos(\alpha) \quad (2.42)$$

O bien:

$$\vec{u} \bullet \vec{v} = u_1 v_1 + u_2 v_2 + u_3 v_3 + \dots + u_n v_n \quad (2.43)$$

donde el producto será máximo cuando el $\cos(\alpha) = 1$. El producto interno es un valor escalar que indica que los vectores se encuentran paralelos, en la misma longitud y dirección. El producto interno podría ser neutro cuando el $\cos(\alpha) = 0$, es decir, que no hay correspondencia entre los vectores.

Para que se pueda realizar la comparación entre los vectores y el producto interno, es necesario realizar una normalización de ambos. Esto es, para suponer que cualquier valor incluido dentro del vector sea siempre positivo y que tienen la misma magnitud o longitud de vector. La norma del vector se calcula con la ecuación (2.44).

$$||\vec{u}|| = \frac{\vec{u}}{\sqrt{\vec{u} \bullet \vec{u}}} \quad (2.44)$$

Las imágenes son matrices en dos dimensiones o vectores demasiado largos (desde el punto de vista de programación) el producto *interno* se puede realizar mediante la siguiente forma:

Sea $\mathbf{A}$ y $\mathbf{B}$ de dimensiones $(m \times n)$, entonces se puede definir el producto interno como la expresión:

$$\mathbf{A} \bullet \mathbf{B} = \text{tr}(\mathbf{A}' \cdot \mathbf{B}) \quad (2.45)$$

donde la tr se refiere a la traza de una matriz, que es equivalente a decir que es la suma de la diagonal principal de la matriz.

Para ello debe de cumplir con algunas propiedades del producto punto: Sea $\mathbf{A}$, $\mathbf{B}$ y $\mathbf{C}$ matrices de $m \times n$ y sea k un escalar, entonces el producto punto cumple con las siguientes propiedades.

1. Simetría: $\mathbf{A} \bullet \mathbf{B} = \mathbf{B} \bullet \mathbf{A}$
2. No negativa: $\mathbf{A} \bullet \mathbf{A} \geq 0$, habiendo igualdad ssi $\mathbf{A} = \mathbf{0}$
3. Proporcionalidad: $(k\mathbf{A}) \bullet \mathbf{B} = k(\mathbf{A} \bullet \mathbf{B})$
4. Distributividad $(\mathbf{A} + \mathbf{B}) \bullet \mathbf{C} = \mathbf{A} \bullet \mathbf{C} + \mathbf{B} \bullet \mathbf{C}$
5. $\mathbf{0} \bullet \mathbf{A} = 0$

La norma del vector establece una longitud o magnitud de un vector por medio del producto interno consigo mismo, así como se denotaba en los vectores.

$$\mathbf{A} \bullet \mathbf{A} = \text{tr}(\mathbf{A}'\mathbf{A}) \quad (2.46)$$

$$= \sum_{i=1}^m \sum_{j=1}^n a_{ij}^2 \geq 0 \quad (2.47)$$

Entonces la norma de una matriz $m \times n$ queda definida como:

$$\|A\|_2 = \|A\| = \sqrt{\mathbf{A} \bullet \mathbf{A}} \quad (2.48)$$

Para normalizar la imagen, se usa la expresión de la ecuación (2.49).

$$\mathbf{I}_n = \frac{\mathbf{I}}{\|\mathbf{I}\|} = \frac{\mathbf{I}}{\sqrt{\mathbf{I} \bullet \mathbf{I}}} \quad (2.49)$$

De la comparación de las imágenes (imagen artificial I_a e imagen real de cámara I_r), por medio del producto interno, se obtiene una calidad (ξ), es decir, un valor de correspondencia entre las imágenes.

$$\xi = I_a \bullet I_r \quad (2.50)$$

Para establecer la función objetivo FO, que se describió en la sección 1.4 y pag. 17, se usa la relación de $1 - \xi$, que es equivalente a decir el error (ε) entre la comparación de las dos imágenes y se convierte en una función a minimizar. Cuando ambas imágenes tienen la misma dirección o sentido, entonces el error tiende a un cero; para el caso que las imágenes no tienen correspondencia, entonces la relación del error tiende a 1, un máximo.

De esta forma se genera la imagen teórica con la metodología propuesta, y en capítulos posteriores se evaluara el funcionamiento del método en.

2.5. Calibración de cámara

El proceso de calibración de cámara se refiere a la determinación de parámetros que la describen. Se pueden mencionar las características internas, geométricas y ópticas de la cámara las cuales son denominadas como *parámetros intrínsecos*. Mientras que la posición y la orientación de la cámara en un marco de referencia 3D en el espacio, se pueden aludir como *parámetros extrínsecos*. Siendo cada uno de ellos independientes entre sí.

La determinación de las características antes mencionadas se logra mediante la utilización de técnicas muy elaboradas y otras no tanto, así como también usar una o un par de cámaras (stereovisión), haciendo uso de observaciones de patrones en los cuales se conoce información de ellos en todo momento y así también con solo tomar una serie de imágenes que logren proporcionar información suficiente para obtener dichos parámetros (Zhang., 2010), (Trucco et Verri, 1998), (Tsai., 1987) por mencionar algunos. De igual

manera, hay varios artículos que se dedican a realizar una comparación entre cada metodología para determinar cuál es mejor y analizan ventajas de una con respecto a otra (Hoyos Gutiérrez et al., 2010), (Sánchez Martín N. et J., 2004), (Romero, 2002).

Los parámetros intrínsecos se agrupan en una matriz de calibración que se debe de proponer para el modelo de cámara, y dichos parámetros son modificables debido a las condiciones reales a las cuales está expuesto, y éstos son:

- Desplazamiento del punto principal de imagen.
- Escalado de imagen.
- Distorsión óptica.
- Distancia focal.

Del modelo de cámara de pinhole se realizaba la transformación de un plano $\mathbb{R}^3$ a un modelo de $\mathbb{R}^2$, por medio de $(x, y, z)^T \mapsto (fx/z, fy/z)^T$. Se utiliza un sistema homogéneo para representar esta transformación como:

$$x = \begin{bmatrix} fx \\ fy \\ z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (2.51)$$

Lo cual sería equivalente a decir:

$$x = \mathbf{P}\mathbf{X}$$

también se puede representar de manera vectorial como $\text{diag}(f, f, 1)[\mathbf{I}|\mathbf{0}]$.

Ahora bien, si se considera el origen del plano de la imagen con respecto al centro de la cámara, se usa $(x, y, z)^T \mapsto (fx/z + u_0, fy/z + v_0)^T$ y representado en coordenadas homogéneas:

$$x = \begin{bmatrix} fx + zu_0 \\ fy + zv_0 \\ z \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (2.52)$$

Si se llama:

$$\mathbf{M}_c = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.53)$$

Entonces se puede representar de forma más compacta con: $x = \mathbf{M}_c[\mathbf{I}|\mathbf{0}]$ y $\mathbf{M}_c$ es una matriz de calibración de la cámara.

Cuando las cámaras no tienen sus pixeles del mismo tamaño(cuadrados), en la matriz de $\mathbf{M}_c$ cambia el modelo por un factor de escala m_x y m_y .

$$\mathbf{M}_c = \begin{bmatrix} \alpha_x & 0 & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.54)$$

donde $\alpha_x = fm_x$, $\alpha_y = fm_y$ representan la distancia focal de la imagen en términos de la dimensión del pixel. Además, (u_0, v_0) es el punto principal en términos de la dimensión del pixel.

Finalmente, aparece un último parámetro en $\mathbf{M}_c$.

$$\mathbf{M}_c = \begin{bmatrix} \alpha_x & s & u_0 \\ & \alpha_y & v_0 \\ & & 1 \end{bmatrix} \quad (2.55)$$

s (skew o encuadre) el cual es un factor con el que se puede modelar el caso en que los ejes de la cámara no sean perfectamente ortogonales, aunque en la mayoría de los casos éste va a ser igual a cero.

Finalmente el sistema queda de la siguiente forma:

$$\begin{bmatrix} U \\ V \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & 0 & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} \quad (2.56)$$

Si en el proceso de calibración se considera las deformaciones radiales o tangenciales (Weng et al., 1992), por efectos ópticos debido a la lente, se debería de tener en cuenta la inclusión de un sistema de ecuaciones de la siguiente forma:

$$\begin{bmatrix} U_d \\ V_d \end{bmatrix} = \left[1 + k_{c1}r^2 + k_{c2}r^4 + k_{c5}r^6 \right] \begin{bmatrix} U \\ V \end{bmatrix} \quad (2.57)$$

donde $r^2 = U^2 + V^2$, siendo los términos de ésta ecuación no lineales. Mientras que k_{c1} , k_{c2} y k_{c5} son parámetros de distorsión radial.

El efecto de distorsión en la imagen puede verse minimizada o eliminada utilizando lentes telecéntricas. Las lentes de éste tipo son usadas comúnmente para realizar mediciones muy precisas y eliminan dicho efecto de distorsión por que su óptica permite que los haces de luz incidan de manera perpendicular(colimada) al sensor de cámara usando el principio de proyección paralela. El resultado es que la magnificación del objeto siempre será igual, independientemente de la distancia del objeto, esto se debe a que el centro de perspectiva siempre se encuentra en el infinito.

Por el momento este tipo de errores sobre la imagen no se considera como lo más importante, en comparación con errores como esfericidad, coma, astigmatismo y errores de separación de colores. Por tanto, se logra concentrar la atención en la matriz de calibración de cámara.

El proceso de calibración estará enfocado en dos fases: El primero es referido a calibrar los parámetros intrínsecos de cámara, mientras que el segundo a calibración del espacio de trabajo. Este último se refiere a las descripción de la posición y orientación de láser de línea alrededor de cámara, así como el ángulo con el que se proyectan al plano de trabajo.

La metodología para la calibración será idéntica a la que se empleó en la *generación de imágenes*, salvo que ahora tendrá algunas modificaciones en las variables de búsqueda e información que se le proporciona para comenzar a trabajar.

Para la generación de imágenes se proponía información como parámetros de cámara los cuales eran considerados como correctos, de la misma forma sucedía para la información que se encuentra relacionada con los láseres y la información del plano en la que se proyectan los láser de línea, dando como resultado una imagen real virtual generada. A través de un algoritmo de búsqueda por optimización se integra la función de *generador de imágenes* proporcionando la información necesaria para su funcionamiento, salvo que ahora se da por desconocida la información del plano (posición y orientación), la cual se convierten en parámetros de búsqueda por el optimizador.

En la primera fase de calibración de cámara se vuelve a tomar el procedimiento del algoritmo de optimización, solo que ahora el enfoque de búsqueda son los parámetros de cámara que están denotados por la matriz de calibración de la ecuación (2.56). Basta con una imagen para determinar el parámetro de magnificación y una segunda imagen para la distancia focal. Para ésta última fase ya que se tiene la información sobre la imagen, parámetros de cámara calibrados se hace, de igual forma, el uso del algoritmo de optimización para buscar ahora los parámetros sobre los láseres y en la cual se toman dos imágenes en las que se logrará determinar la orientación, posición y ángulo con el que se proyectan hacia la superficie.

Lo anterior se logra resumir en el diagrama mostrado en la figura 2.25

Para este proceso de calibración solo se ha considerado la adquisición de cuatro imágenes y dos están destinadas para el tablero de ajedrez y las otras para los láser de líneas. Si fuera necesario realizar la adquisición de más imágenes éstas se lograrían anexar al esquema mostrado en la figura 2.25.

Primero se debe de generar una imagen real virtual de un tablero de ajedrez, el cual se logra por medio de la función del “generador de imagen” y en la estimación de parámetros se proporciona toda la información necesaria para crear la imagen (parámetros de cámara, posición del plano, números de campos con los que cuenta el ajedrez). Con esta situación se supone que toda la información es conocida y correcta. Como el requisito por proceso de calibración es de dos imágenes, entonces una segunda imagen se genera con la misma información que la anterior, pero ahora cambiando la posición del plano del ajedrez. Estas imágenes son las que se consideran como captura de imagen el diagrama mostrado para la calibración (ver figura 2.25). Ahora a lo que se procede es que al algoritmo de optimización se le proporciona toda la información para que inicie, con excepción de que ahora la información de búsqueda son los parámetros de cámara. Al final del proceso de optimización se deben de encontrar los parámetros de cámara que se usaron para generar la imagen por medio de la función. De forma similar sucede el proceso de los láseres, donde se genera la imagen con información conocida y después se esconden los parámetros que se relacionan con estos para por medio del algoritmo de optimización se realice la búsqueda.

La información a determinar en el primer bloque de trabajo en la calibración está enfocada a u_0 y v_0 , el cual tiene que ver con el eje óptico de la imagen. En el algoritmo de optimización se proporciona tal cual esta variable, ya que no depende de ningún factor y

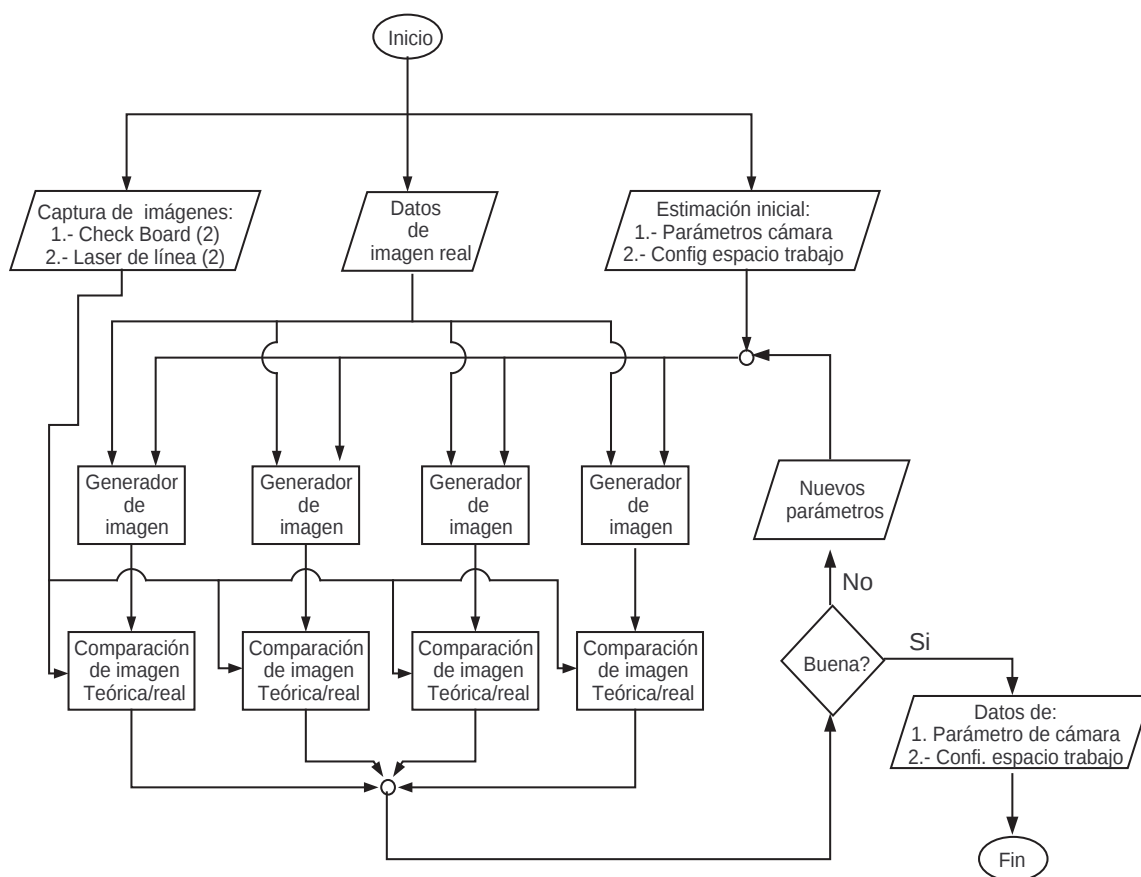


Figura 2.25: Metodología para calibración de cámara.

típicamente su valor se encuentra a la mitad de la dimensión tanto del eje U y V en espacio de imagen.

La magnificación es un parámetro que está relacionado entre el tamaño real del objeto y el tamaño del objeto en la imagen. Dicho de manera más coloquial, la magnificación consiste en comparar el tamaño de un objeto con el del sensor de la cámara y sacar una proporción. Mientras que el efecto de perspectiva hace que la imagen pueda aumentar de tamaño o disminuir si la distancia al objeto se incrementa o se aleja.

Los valores de magnificación y perspectiva son términos que se tratarán de trabajar por separado, debido a que son parámetros que afectan casi de la misma manera. Algunas veces no se logran distinguir en las variaciones que sufren éstos y es por ello que se tratarán

de desacoplar para este proceso que se propone.

$$\begin{pmatrix} u_0 \\ v_0 \\ k_x \\ k_y \\ z_{cam} \end{pmatrix} \rightarrow \begin{pmatrix} u_0 \\ v_0 \\ m \\ \beta \\ \frac{1}{z_{cam}} \end{pmatrix} \quad (2.58)$$

De lo que se desprende la variable m y β como un factor de magnificación. Con la condición que $m \gg 1$ como condición necesaria para este problema en particular y $\beta \approx 1$.

Entonces, para el efecto de perspectiva se define el z_{cam} con su inverso, para pasar de un modelo de cámara pinhole a un modelo ortográfico que permite hacer variaciones en el infinito y no se ve afectado la magnificación, que dicho de otra forma es la distancia focal.

Si la distancia focal se representa por:

$$f = \frac{1}{z_{cam}} \quad (2.59)$$

Los parámetros de k_x pueden ser descritos con:

$$k_x = mf = \frac{m}{z_{cam}} \quad (2.60)$$

y de forma similar para k_y :

$$k_y = mf\beta = \frac{m\beta}{z_{cam}} \quad (2.61)$$

2.6. Calibración de láseres

La calibración de los láseres, como se describió en la sección anterior, es el uso de la misma metodología del proceso de calibración de cámara, pero ahora cambian unos parámetros de búsqueda.

Para ello se recurre al diagrama general del cual se hizo el análisis del sistema (ver figura 2.6) . En ello se lograrán identificar las variables que están relacionadas con los láseres.

La variable r_o tiene que ver con la distancia a la cual se encuentra el láser respecto al eje focal de cámara, el ángulo φ es la distribución que tiene los láser alrededor de la cámara y tiene un ángulo $\approx 120^\circ$ entre ellos. El otro parámetro que habrá que identificarse es ψ que es el ángulo con el cual se proyecta la línea sobre el plano de proyección.

Los tres variables están descritas para cada láser y, como se tienen tres de estos dispositivos incluidos en el cabezal de visión artificial, entonces son *nueve parámetros de búsqueda por cada imagen*, entonces se esperaría que el tiempo de búsqueda se incrementara. Habrá que mencionar que los láseres tienen un ángulo ψ de proyección al plano, pero que a su vez estos tienen un ángulo sobre su propio eje de proyección (alabeo), dicho ángulo se logra compensar haciendo una redistribución de los láseres alrededor de la cámara, es decir, que posiblemente al menos un láser no esté en 120° de separación respecto a otro.

Entonces las variables que se ponen en búsqueda por el algoritmo de optimización son:

$$\begin{bmatrix} r_0 \\ \varphi \\ \psi \end{bmatrix} \quad (2.62)$$

y cada parámetro está dado para tres láseres; r_0 es un dato que se busca con información muy pequeña en sus variaciones, por lo que lleva a que ésta distancia es establecida en un valor.

Capítulo 3

Evaluación y pruebas del procesamiento de imágenes

En esta sección, se podrán encontrar las evaluaciones que se realizaron a los métodos desarrollados en secciones anteriores. El objetivo de las pruebas está enfocado en determinar costos computacionales que éstas pudieran generar. También están enfocados en evaluar la precisión con la cual se puede extraer información de las imágenes que se procesarán, siendo estas, imágenes virtuales que ya han sido generadas con información conocida.

3.1. Pruebas a partir de imágenes virtuales

En la figura 3.1, se puede observar el árbol de pruebas que se ha propuesto para obtener la información deseada.

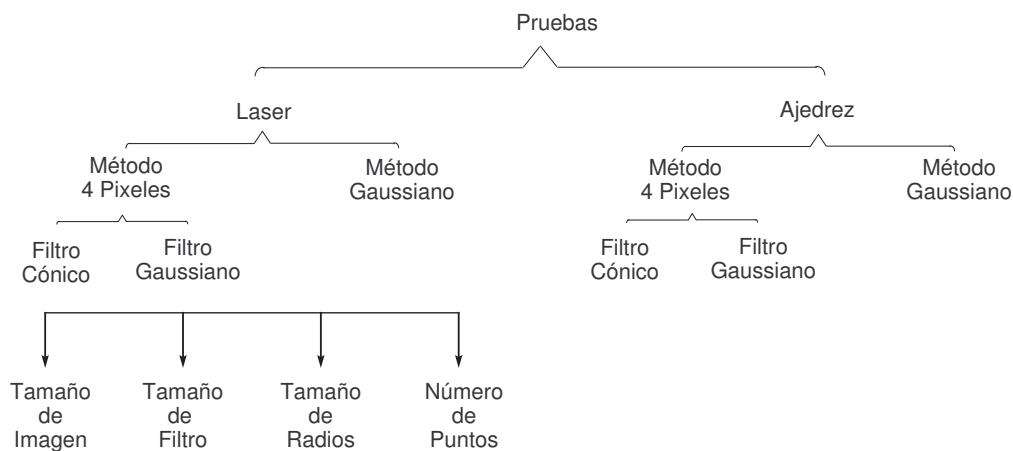


Figura 3.1: Árbol de pruebas.

Se realiza la evaluación de las funciones programas en conjunto, para determinar el tiempo que puede llevar acabo en realizar una generación de una imagen virtual. La idea esencial es, plantear una serie de combinaciones de métodos de ensamble, filtros, dimensiones de filtro, tamaño de imagen a generar, etc. Se busca determinar la combinación que logre mejores resultados en cuestión de rapidez, para que posteriormente, se pueda evaluar la eficiencia y exactitud para determinar información sobre la imagen, a través de la metodología que se propuso.

De los dos clases de imágenes que se pueden obtener del método, solo se realizan las evaluaciones para las imágenes de láser ya que es el tema principal del trabajo, mientras que el tablero de ajedrez solo es utilizado para calibración de cámara. Entonces, tomando la rama del láser y los métodos de ensamble de puntos, se diseñan las pruebas para el caso en que existen dos tamaños de imágenes (cambio de resolución). De la misma forma, para cuando existe cambio en el tamaño de los filtros, que concierne para el caso del “ensamble de 4Pixel”. Para el método de Gauss se consideran los cambios de radios del filtro; también se toma en cuenta el número de puntos que son ensamblados en la imagen.

Las opciones que son usadas para validar las pruebas, son mostradas en el cuadro 3.1.

Prueba	Opción 1	Opción 2
Tamaño de imagen	1024×1280	512×640
Tamaño de filtro ($m \times n$)	10×10	20×20
Tamaño de radios (r)	5	10
Número de Puntos	10	100

Cuadro 3.1: Especificación de opciones en pruebas de complemento.

Se generan 24 posibles combinaciones en pruebas y cada una será realizada en 10, 100 y 1000, iteraciones con la finalidad de poder obtener resultados sustanciales. Otra finalidad que se persigue es la de obtener una ecuación que permita estimar el tiempo bajo condiciones que no son establecidas en las pruebas.

Las evaluaciones se definen de manera específica en el cuadro 3.2. Las pruebas de la 1 a la 8 son dedicadas al método gaussiano, mientras las pruebas de la 9 a la 16 son para el método “4Pixel” con un ensamblaje de puntos con filtro gaussiano. Finalmente, de la prueba 17 a la 24 es para la segunda opción del método de “4Pixel” con un ensamblaje de puntos con filtro cónico.

Como es importante determinar el tiempo que toma realizar las evaluaciones, se debe de tomar en cuenta la plataforma experimental sobre la cual se trabaja (sistema operativo y características de máquina). Y para ello, en el cuadro 3.3 se incluyen las características de las máquinas utilizadas para desarrollar las evaluaciones.

A juzgar por las características de máquina, lo que se esperaría es que la de memoria

Método Gaussiano							
Prueba 1	Prueba 2	Prueba 3	Prueba 4	Prueba 5	Prueba 6	Prueba 7	Prueba 8
1024×1280	1024×1280	1024×1280	1024×1280	512×640	512×640	512×640	512×640
Gaussiano	Gaussiano	Gaussiano	Gaussiano	Gaussiano	Gaussiano	Gaussiano	Gaussiano
10 Pts	100 Pts	10 Pts	100 Pts	10 Pts	100 Pts	10 Pts	100 Pts
R=5	R=5	R=10	R=10	R=5	R=5	R=10	R=10
E=2	E=2	E=2	E=2	E=2	E=2	E=2	E=2
Método 4 Píxeles con Filtro Gaussiano							
Prueba 9	Prueba 10	Prueba 11	Prueba 12	Prueba 13	Prueba 14	Prueba 15	Prueba 16
1024×1280	1024×1280	1024×1280	1024×1280	512×640	512×640	512×640	512×640
4Pixel	4Pixel	4Pixel	4Pixel	4Pixel	4Pixel	4Pixel	4Pixel
Gaussiano	Gaussiano	Gaussiano	Gaussiano	Gaussiano	Gaussiano	Gaussiano	Gaussiano
10 Pts	100 Pts	10 Pts	100 Pts	10 Pts	100 Pts	10 Pts	100 Pts
n=10	n=10	n=20	n=20	n=10	n=10	n=20	n=20
m=10	m=10	m=20	m=20	m=10	m=10	m=20	m=20
r=5	r=5	r=10	r=10	r=5	r=5	r=10	r=10
Método 4 Píxeles con Filtro Cónico							
Prueba 17	Prueba 18	Prueba 19	Prueba 20	Prueba 21	Prueba 22	Prueba 23	Prueba 24
1024×1280	1024×1280	1024×1280	1024×1280	512×640	512×640	512×640	512×640
4Pixel	4Pixel	4Pixel	4Pixel	4Pixel	4Pixel	4Pixel	4Pixel
Cónico	Cónico	Cónico	Cónico	Cónico	Cónico	Cónico	Cónico
10 Pts	100 Pts	10 Pts	100 Pts	10 Pts	100 Pts	10 Pts	100 Pts
n=10	n=10	n=20	n=20	n=10	n=10	n=20	n=20
m=10	m=10	m=20	m=20	m=10	m=10	m=20	m=20
r=5	r=5	r=10	r=10	r=5	r=5	r=10	r=10

Cuadro 3.2: Pruebas diseñadas para la generación de imágenes. n y m es el tamaño de la máscara del filtro, r es el radio del filtro dentro de la máscara.

Marca-Modelo	DELL VOSTRO 460	Sun Microsystem
Procesador	Intel@CORE TM i5 2400 CPU @3.10 GHz	Dual-Core AMD Opteron TM Procesador 2214 @2.20 GHz
RAM	4.00 GB	2.00 GB
Sistema Operativo	Windows 7	Windows XP

Cuadro 3.3: Características de máquinas.

RAM de 4,0 Gb tenga mejores resultados que la otra, ya que comprende el doble de memoria RAM y frecuencia de trabajo del procesador mayor.

Los resultados de los tiempos registrados a través del software se pueden observar en la figura 3.2, del total de pruebas solo se escogieron 3 para mostrar de manera gráfica y lograr evidenciar las evoluciones de tiempo, dependiendo del método a emplear en la generación de imágenes y equipo de trabajo.

En las máquinas el reloj es accionado por un cristal de cuarzo. Las moléculas en el cristal de cuarzo vibran millones de veces por segundo, a una velocidad que nunca cambia.

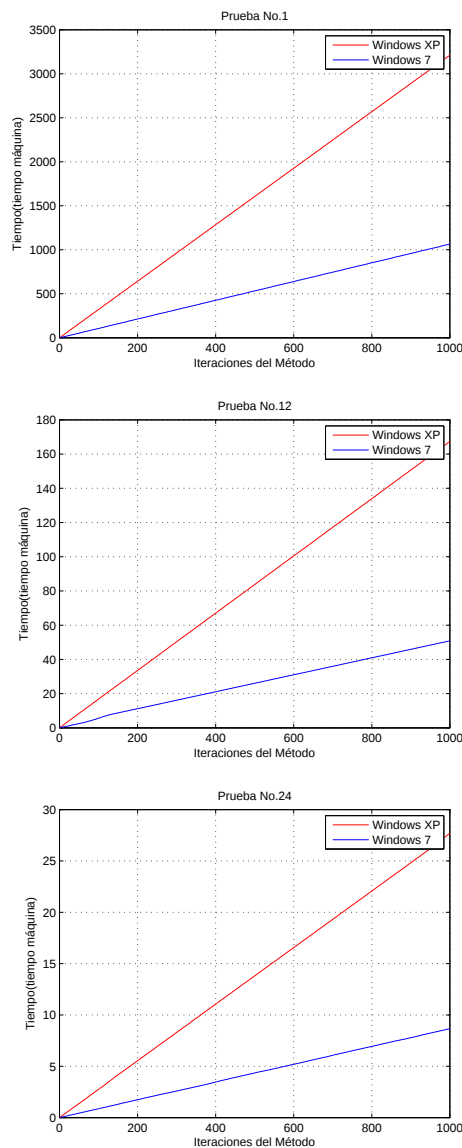


Figura 3.2: Evaluaciones para los método de generación de imágenes.

La computadora usa las vibraciones en el reloj del sistema para tomar el tiempo de sus operaciones de procedimiento y por tanto, ese tiempo es el que está registrado en la tabla 3.4. En la columna de tiempos para Windows 7 se denota con t_{w7} y para Windows XP se usa la nomenclatura t_{XP} respectivamente.

Para obtener el promedio entre cada iteración de las evaluaciones, se utiliza la ecuación (3.1):

$$\overline{X} = \frac{\sum_{n=2}^{1000} (x_n - x_{n-1})}{n - 1} \quad (3.1)$$

donde x_{n-1} es el elemento anterior y x_n es el actual.

Una vez que se tiene el promedio de la muestra, se procede a determinar las variaciones entre ellas, a partir de la ecuaciones siguientes, lo que hace la varianza es establecer la dispersión de la variable aleatoria por medio de (3.2) ó (3.3):

$$S^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{X})^2 \quad (3.2)$$

ó también se puede obtener mediante la ecuación :

$$S^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{X})^2 \quad (3.3)$$

A las ecuaciones (3.2) y (3.3) se les denomina varianza muestral. El primero traslada directamente la varianza de la muestra al de la población y el segundo es un estimador insesgado de la varianza de la población, y para este último la relación de $(n-1)$ en muestras muy grandes es indiferente.

Y la desviación standard (σ) que es también una medida de la dispersion de los datos respecto de su valor medio, se convierte en la raíz cuadrada de la varianza y es calculado por la ecuación (3.4):

$$S = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{X})^2} \quad (3.4)$$

Dichos resultados de las operaciones antes mencionadas están registrados en la tabla 3.4.

Prueba	t_{W7}	$\bar{X}_{W7}$	S_{W7}	σ_{W7}	t_{XP}	$\bar{X}_{XP}$	S_{XP}	σ_{XP}
1	1064.85943	1.06488854	0.01467204	0.12112821	3210.65719	3.21066727	0.00010345	0.01017088
2	10461.3942	10.4616858	0.3977646	0.63068582	31857.9106	31.858286	0.08752314	0.29584311
3	1142.36878	1.1424188	0.0143205	0.11966828	3683.986	3.68401294	0.00452371	0.06725856
4	11248.1719	11.2483981	0.37886388	0.61551919	39389.6431	39.3926693	12.0067953	3.46508229
5	249.38135	0.24938878	0.00224977	0.04743172	843.18858	0.84319589	0.00040108	0.02002697
6	2575.73131	2.57402629	0.17044347	0.412848	8631.95764	8.6320407	0.02750533	0.16584732
7	288.79406	0.28880605	0.00288151	0.05367971	1002.58256	1.00259593	0.00068591	0.0261898
8	2972.60453	2.97273688	0.19613276	0.44286878	10312.913	10.3129925	0.02754685	0.16597245
9	35.2822	0.03528095	8.79E-08	0.00029648	120.78373	0.12078295	8.44E-07	0.00091844
10	37.27285	0.03727355	7.87E-08	0.00028055	124.66168	0.12466557	9.63E-07	0.00098149
11	47.3664	0.04736741	1.36E-06	0.0011669	163.88794	0.16389236	4.12E-06	0.00202926
12	50.84257	0.05084336	0.00010212	0.01010559	167.35267	0.16735716	4.53E-06	0.0021294
13	9.31311	0.00931179	2.82E-08	0.00016798	32.14089	0.03214066	6.29E-07	0.00079286
14	10.95417	0.0109532	2.98E-08	0.00017258	34.97764	0.03497789	4.97E-08	0.00022291
15	13.91154	0.0139133	7.72E-05	0.00878862	40.56042	0.04056161	2.66E-06	0.0016304
16	14.64599	0.01464671	6.02E-07	0.00077602	43.77071	0.04377057	2.56E-06	0.00159904
17	21.1971	0.02119261	2.33E-07	0.00048297	83.66472	0.08366795	3.18E-06	0.00178284
18	23.18183	0.023182	1.33E-07	0.00036461	86.74473	0.08674802	3.23E-06	0.00179598
19	23.69099	0.02369344	3.85E-06	0.00196122	86.54932	0.08655329	5.17E-05	0.00718839
20	25.63183	0.02563346	2.88E-06	0.00169569	90.20901	0.09021086	3.17E-05	0.00563095
21	5.50537	0.00550389	4.17E-08	0.00020432	22.50275	0.0225025	9.61E-08	0.00031008
22	7.40198	0.00740213	3.00E-08	0.00017314	26.15727	0.02615794	2.14E-06	0.00146354
23	6.87079	0.00687228	4.03E-07	0.00063473	23.71817	0.02371968	3.95E-06	0.00198758
24	8.65703	0.00865716	3.70E-07	0.00060842	27.70415	0.02770585	6.22E-06	0.00249347

Cuadro 3.4: Resultados de las evaluaciones realizadas.

Se pueden utilizar los tiempos registrados de las dos plataformas para obtener la proporcionalidad entre ellos dependiendo del sistema operativo, los cuales se pueden observar en la tabla 3.5.

Prueba	t_{W7}	t_{XP}	Proporción	Prueba	t_{W7}	t_{XP}	Proporción
1	1064.85943	3210.65719	3.01509955	13	9.31311	32.14089	3.45114468
2	10461.3942	31857.9106	3.04528345	14	10.95417	34.97764	3.19308902
3	1142.36878	3683.986	3.22486579	15	13.91154	40.56042	2.91559525
4	11248.1719	39389.6431	3.50187064	16	14.64599	43.77071	2.98857981
5	249.38135	843.18858	3.38112124	17	21.1971	83.66472	3.94698897
6	2575.73131	8631.95764	3.35126479	18	23.18183	86.74473	3.74192762
7	288.79406	1002.58256	3.47161766	19	23.69099	86.54932	3.6532589
8	2972.60453	10312.913	3.46931887	20	25.63183	90.20901	3.51941356
9	35.2822	120.78373	3.42336164	21	5.50537	22.50275	4.08741828
10	37.27285	124.66168	3.34457065	22	7.40198	26.15727	3.53382068
11	47.3664	163.88794	3.46000414	23	6.87079	23.71817	3.45202953
12	50.84257	167.35267	3.29158557	24	8.65703	27.70415	3.20019106

Cuadro 3.5: Relación de proporcionalidad de tiempos en la generación de imágenes.

De los registros de tiempos en la tabla 3.5, se puede notar que las primeras 8 pruebas dedicadas al método gaussiano generan tiempos muy elevados. Las evaluaciones realizadas para el método de ensamble “4Pixel” registran mejores tiempos y por tanto puede ser considerado como un método viable. Después habrá de evaluarse cuál de los dos ensambles utilizados en el método puede llegar a ser más preciso. Las proporciones en velocidad dependiendo, del sistema operativo, se dan en una relación de 1 a 3 en promedio de forma escueta, y siendo más precisos es 3,4026 la relación de proporcionalidad promedio.

Como se trata de determinar tiempos en la generación de imágenes a través de los diferentes ensambles de puntos y, de igual manera, deducir una ecuación que permita predecir el tiempo que tardaría en realizar una corrida de este tipo, para ello se plantea una ecuación de la siguiente forma:

$$t = aN + t_0 \quad (3.5)$$

donde t es el tiempo que tomaría en realizar el método una evaluación, a una constante de tiempo que se da entre cada iteración y está depende de las condiciones de las pruebas, N es el número de iteraciones a realizar en la evaluación, y finalmente t_0 sería una constante de inicialización del proceso.

Por las condiciones de a bien se puede establecer el valor como el promedio en tiempo ($\bar{X}$) de las pruebas. El valor de t_0 se obtiene realizando la resta entre el primer valor de cada prueba y el valor del tiempo promedio ($\bar{X}$) de la prueba, si el residuo de la operación es menor a su varianza (S), entonces el tiempo de inicialización del proceso puede ser despreciable.

Para lo anterior se genera el cuadro 3.6, en la cual solo se incluyen los tiempos a partir de el método de “4Pixel”, es decir, de la prueba 9 a la 24.

	Prueba 9	Prueba 10	Prueba 11	Prueba 12	Prueba 13	Prueba 14	Prueba 15	Prueba 16
t	0.03653	0.03657	0.04636	0.05005	0.01063	0.01192	0.01215	0.01393
X	0.03528095	0.03727355	0.04736741	0.05084336	0.00931179	0.0109532	0.0139133	0.01464671
$t-X$	0.00124905	0.00070355	0.00100741	0.00079336	0.00131821	0.0009668	0.0017633	0.00071671
S	8.79E-08	7.87E-08	1.36E-06	0.00010212	2.82E-08	2.98E-08	7.72E-05	6.02E-07
	Prueba 17	Prueba 18	Prueba 19	Prueba 20	Prueba 21	Prueba 22	Prueba 23	Prueba 24
t	0.02568	0.02301	0.02124	0.024	0.00698	0.00725	0.00538	0.00853
X	0.02119261	0.023182	0.02369344	0.02563346	0.00550389	0.00740213	0.00687228	0.00865716
$t-X$	0.00448739	0.000172	0.00245344	0.00163346	0.00147611	0.00015213	0.00149228	0.00012716
S	2.33E-07	1.33E-07	3.85E-06	2.88E-06	4.17E-08	3.00E-08	4.03E-07	3.70E-07

Cuadro 3.6: Offset de tiempo para iniciar las evaluaciones en W7.

Para las evaluaciones de XP se tiene la tabla 3.7:

	Prueba 9	Prueba 10	Prueba 11	Prueba 12	Prueba 13	Prueba 14	Prueba 15	Prueba 16
t	0.12156	0.12078	0.15947	0.16287	0.03237	0.03473	0.03937	0.04391
X	0.12078295	0.12466557	0.16389236	0.16735716	0.03214066	0.03497789	0.04056161	0.04377057
$t-X$	0.00077705	0.00388557	0.00442236	0.00448716	0.00022934	0.00024789	0.00119161	0.00013943
S	8.44E-07	9.63E-07	4.12E-06	4.53E-06	6.29E-07	4.97E-08	2.66E-06	2.56E-06
	Prueba 17	Prueba 18	Prueba 19	Prueba 20	Prueba 21	Prueba 22	Prueba 23	Prueba 24
t	0.08044	0.08346	0.08258	0.08836	0.02275	0.02549	0.02221	0.02601
X	0.08366795	0.08674802	0.08655329	0.09021086	0.0225025	0.02615794	0.02371968	0.02770585
$t-X$	0.00322795	0.00328802	0.00397329	0.00185086	0.0002475	0.00066794	0.00150968	0.00169585
S	3.18E-06	3.23E-06	5.17E-05	3.17E-05	9.61E-08	2.14E-06	3.95E-06	6.22E-06

Cuadro 3.7: Offset de tiempo para iniciar las evaluaciones en XP.

Con estos resultados se logran determinar los valores de la ecuación que permiten estimar un tiempo para la realización de la prueba con N iteraciones, pero solo es el tiempo que tardaría el método en generar la imagen pero sin visualizar. El mostrar una imagen en pantalla genera un costo computacional propio del software que puede influir en agregar un costo en tiempo; sin embargo, el proceso de visualización puede ser omitido debido a que el objetivo esencialmente del proceso es “determinar” los valores correctos con los que se genera la imagen teórica para compararla con la imagen real y el proceso de visualización es un apoyo visual de la comparación de imágenes. También es necesario enfatizar que el interés no es guardar las imágenes teóricas sino parámetros de la imagen.

3.1.1. Procesamiento de imágenes

Con las pruebas realizadas anteriormente se lograron determinar los tiempos de procesamiento con N imágenes o iteraciones, donde las condiciones de las pruebas consideraban cambiar tamaños de imagen, parámetros en los filtros y ensamble de puntos de la imagen. Ahora se pretende determinar la calidad y exactitud para detectar los parámetros con los que se describe una imagen, y para ello se emplean dos criterios.

- El primer criterio es dar una tolerancia exigente de calidad dejando libre el número de iteraciones que permita realizar comparación de la imagen teórica generada con una

imagen real, recordando que se plantean resoluciones en la descripción de parámetros de la imagen que se genera a nivel de micrómetros.

- El segundo criterio es ver el número de iteraciones permisibles para llegar a encontrar un resultado de comparación con los parámetros de imagen permisible, es decir, sin llegar a usar una tolerancia muy elevada (acotar la tolerancia).

Un par de hipótesis son las que pueden surgir, para el primer caso, se espera que llegue a tomar un elevado número de iteraciones para encontrar los mejores resultados, es decir, una exactitud elevada contra tiempos muy elevados y de forma análoga se espera que suceda para el segundo caso, resultados con calidad media (milímetros) y un tiempo muy corto en ejecución de tareas.

Las pruebas se plantean de la siguiente forma: Al generador de imágenes se le dan parámetros de una cierta configuración de láser, posición del plano, orientación del mismo, parámetros de cámara conocidos, método de ensamble para generar una imagen. La imagen generada será considerada como real, pero de forma virtual. El segundo proceso será tomar esa imagen real virtual y proporcionarla al algoritmo optimizador quien se encargará de detectar los parámetros de posición y orientación de la superficie en la imagen. Al algoritmo de optimización se le asignan valores para iniciar la búsqueda y a partir de ahí, se realizan los registros de tiempo, iteraciones y tolerancias que se planteaban anteriormente.

Subsecuentemente se generará un cierto número de imágenes con variaciones mínimas de datos en parámetros de posición y orientación, para generar un cronología de imágenes, lo cual simulará una adquisición de imágenes en un proceso real. El segundo paso consiste en proporcionar las imágenes al optimizador para que éste se encargue de encontrar esos parámetros en cada imagen y obtener criterios como número de iteraciones, tolerancia y calidad de reconocimiento de las imágenes. La detección de la primera imagen es la que seguramente tomará mayor número de iteraciones debido a los valores iniciales de búsqueda y en las imágenes subsecuentes que contienen variaciones mínimas se esperan tiempos menores con respecto a la primera imagen.

3.1.1.1. Prueba para una sola imagen.

Los valores de los parámetros con los cuales se genera la imagen real virtual se identificaron en la tabla 3.8, Δz_1 y Δz_2 determinan la pendiente y orientación del plano mientras que z_{P0} es la distancia a la cual se encuentra el plano respecto a la cámara. Los datos contenidos en la columna de “valores” corresponden a la información con la que se generó la imagen, en la columna de “estimación” se muestran los valores iniciales que se propusieron para realizar la búsqueda, y la de “incrementos” establece los pasos de búsquedas sobre la imagen usando el algoritmo de optimización.

La primer prueba consiste en realizar variaciones en las tolerancias, ver cuadro 3.9, y observar como se estabilizan los parámetros de búsqueda en la imagen Δz_1 , Δz_2 y z_{P0} de

Parámetros	Valores	Estimación	Incrementos
Δz_1	0°	.1 (5.72°)	0.08 (4.58°)
Δz_2	0°	.4 (22.91°)	0.08 (4.58°)
z_{p0}	0 mm	100 mm	30 mm

Cuadro 3.8: Características de construcción de imagen virtual real.

acuerdo a los incrementos iniciales que se proponen. **Nota:**El tiempo indicado en la tabla es considerando una previsualización de la imagen.

Tolerancia	Tiempo	Iteración	Δz_1	Δz_2	z_{p0}	Minimización
1×10^{-3}	13,376796	165	$70,0 \times 10^{-6}$	400×10^{-6}	-0.05006	$395,866 \times 10^{-6}$
1×10^{-4}	15,136056	176	100×10^{-6}	-10×10^{-6}	-0.03137	$394,145 \times 10^{-6}$
1×10^{-5}	16,813559	194	110×10^{-6}	0	-0.03704	$393,85 \times 10^{-6}$
1×10^{-6}	17,163609	200	110×10^{-6}	0	-0.03675	$393,849 \times 10^{-6}$
1×10^{-7}	18,985639	217	110×10^{-6}	0	-0.03686	$393,849 \times 10^{-6}$
1×10^{-8}	19,726282	227	110×10^{-6}	0	-0.03687	$393,849 \times 10^{-6}$
1×10^{-9}	21,24104	239	110×10^{-6}	0	-0.03687	$393,849 \times 10^{-6}$

Cuadro 3.9: Registro para las diferentes tolerancias en comparación de imágenes.

En las imágenes mostradas en en la figura 3.3 se logra ver en *a*) un triángulo pequeño en el centro de la imagen y ésta es la imagen real virtual generada. Seguido de ello un triángulo con dimensiones mayores que es la imagen generada con los parámetros propuestos para el inicio de búsqueda en su primera iteración. También se logran identificar puntos luminosos entre los cruces de líneas de láser, esto sucede por una contribución de energía en los puntos de coincidencia. En la imagen *b*) se muestran las dos imágenes después de la búsqueda, es decir, la imagen de búsqueda superpuesta sobre la imagen real virtual.

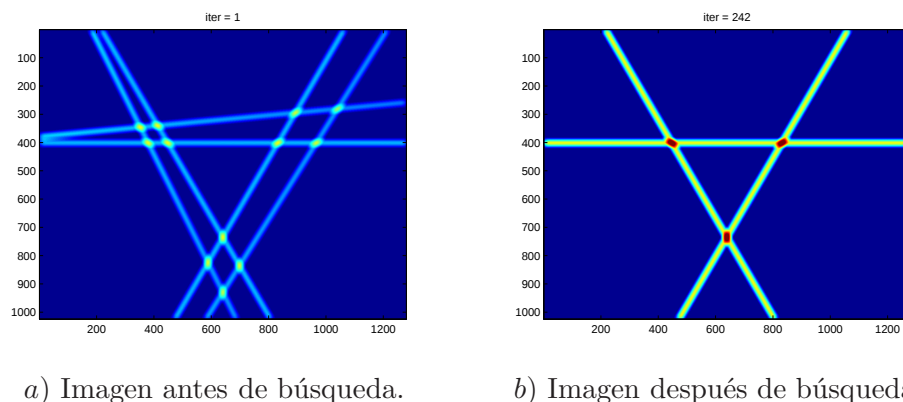


Figura 3.3: Imagen de búsqueda.

El comportamiento de los parámetros de Δz_1 , Δz_2 y z_{p0} se logran ver en las gráficas

mostradas en 3.4; la primera columna representa la tolerancia más baja y la segunda columna de imágenes es para la tolerancia más alta.

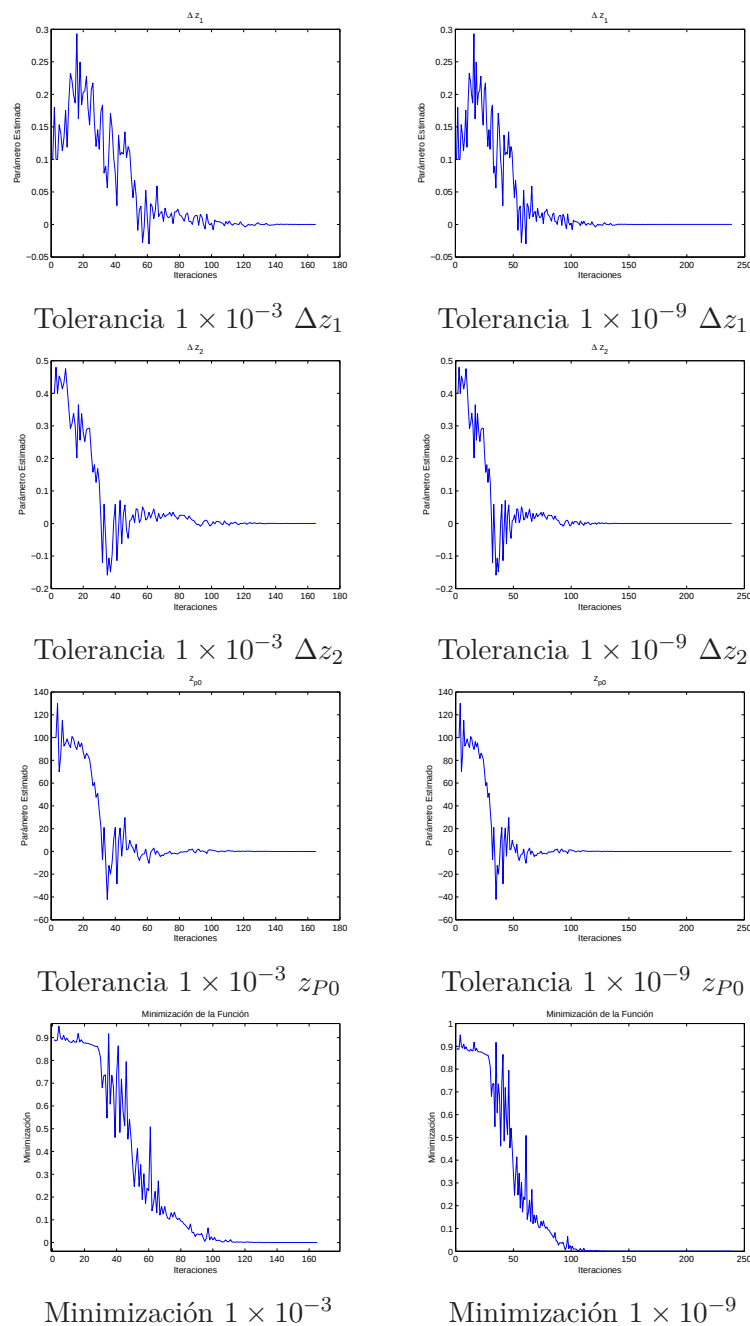


Figura 3.4: Detección de parámetros con cambios de tolerancias.

Las gráficas de la figura 3.4 muestran las evoluciones de los parámetros de búsqueda y, a pesar de ser diferentes niveles de tolerancia, se repite el mismo gráfico en ambos. Al

parecer, después de 140 iteraciones los valores que se obtienen de los parámetros ya son estables, por tanto se tomaría el valor que se encuentra próximo a éste y corresponde a la tolerancia de $1,00 \times 10^{-4}$. En esta opción los resultados de los parámetros de búsqueda muestran una alta precisión.

Para comprobar la validez de la tolerancia utilizada, puede generarse otra imagen real virtual con otras especificaciones y obtener la gráfica de la evolución de parámetros que la describen, ver imagen de la figura 3.5. Una vez más, el triángulo interior es la imagen real teórica y el triángulo exterior es el de búsqueda con el algoritmo de optimización.

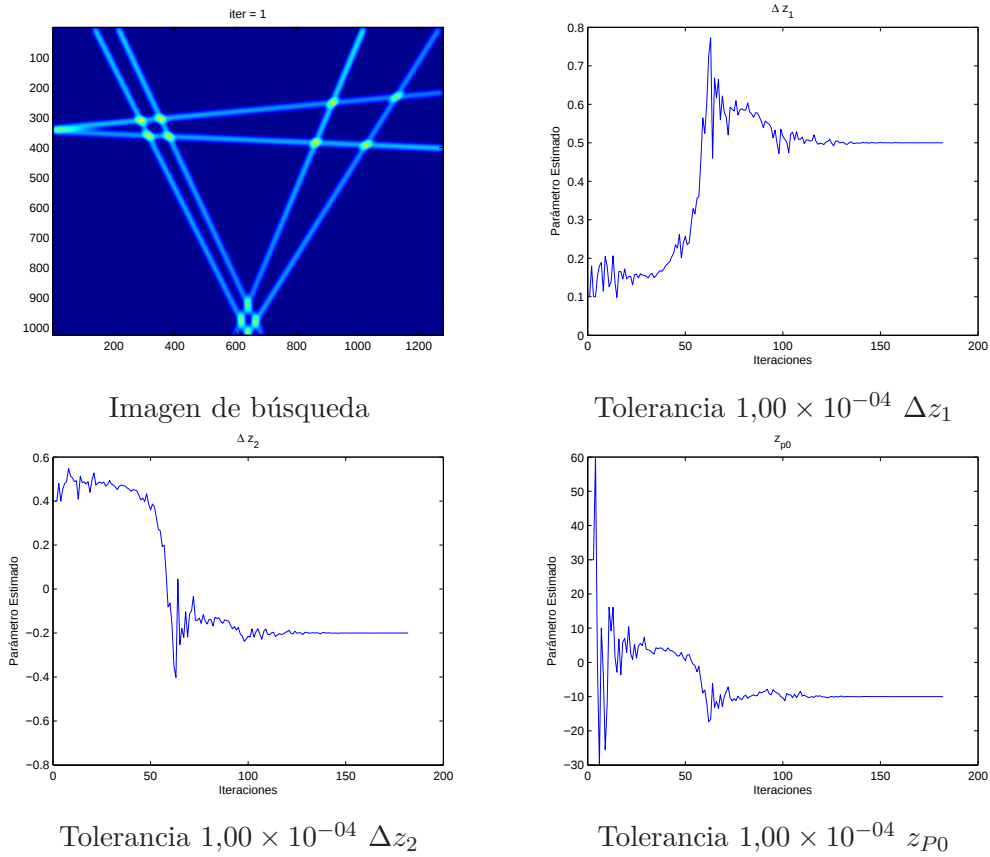


Figura 3.5: Segunda detección de parámetros con tolerancia $1,00 \times 10^{-04}$.

Para tratar de reducir el número de iteraciones se pueden proponer incrementos y tolerancias grandes para hacer una primera búsqueda burda y detectar parámetros rápidos; subsecuentemente se realiza un segundo proceso de búsqueda reduciendo los incrementos y subiendo la tolerancia para lograr hacer converger los parámetros más rápidamente.

En las imágenes de la figura 3.6 muestran varias búsquedas realizando variaciones en los parámetros de inicio, cambiando los incrementos y tolerancias en cuatro ocasiones.

De estas pruebas se logra concluir que en la exploración sobre las imágenes, los valores

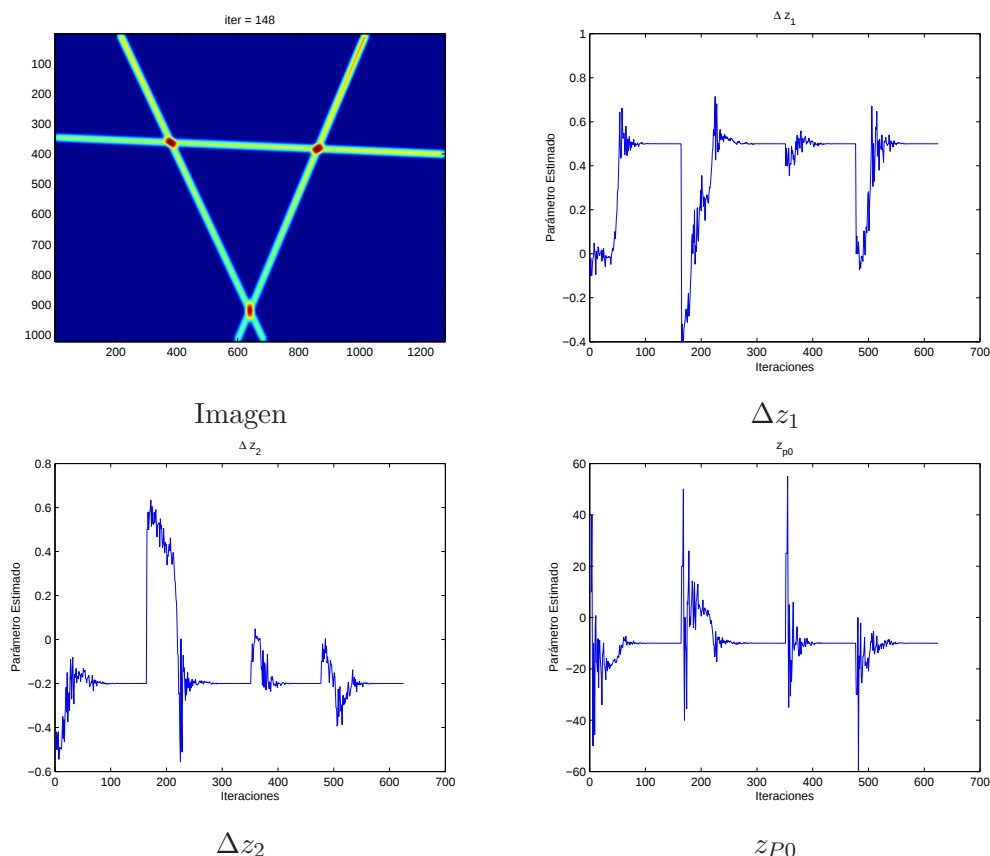


Figura 3.6: Múltiples búsquedas, iniciando con parámetros diferentes.

con los que se inician la búsqueda de parámetros ayudan a reducir el número de iteraciones. También se logra ver que, independientemente de la información con la que se realiza la búsqueda, mientras sea una aproximación, el método logrará converger a una solución; si dicha aproximación es muy errónea no se llega a converger en parámetros correctos. En algunas ocasiones la información parecería ser acorde, pero en en ciertas situaciones, el método queda estancado o cae en mínimos locales, lo que afecta la determinación de información correcta.

En las imágenes mostradas en la figura 3.7 se muestra una imagen que es usada en el proceso de búsqueda *a)*, en *b)* a *e)* se muestran los parámetros registrados por el algoritmo. De la gráfica se logra ver que no existen parámetros estables, además en la gráfica de minimización de la función *e)* queda muy lejano del cero. Finalmente en *f)* se ve la imagen final con la que termina el proceso.

Con los valores que se obtienen de una búsqueda que cae en un mínimo local o bien, que no converge en valores correctos, usualmente termina en pocas iteraciones y con valores muy alejados a los que se podrían esperar. Por ende, sería necesario reiniciar el proceso de búsqueda sobre la imagen con nueva información y esperar resultados.

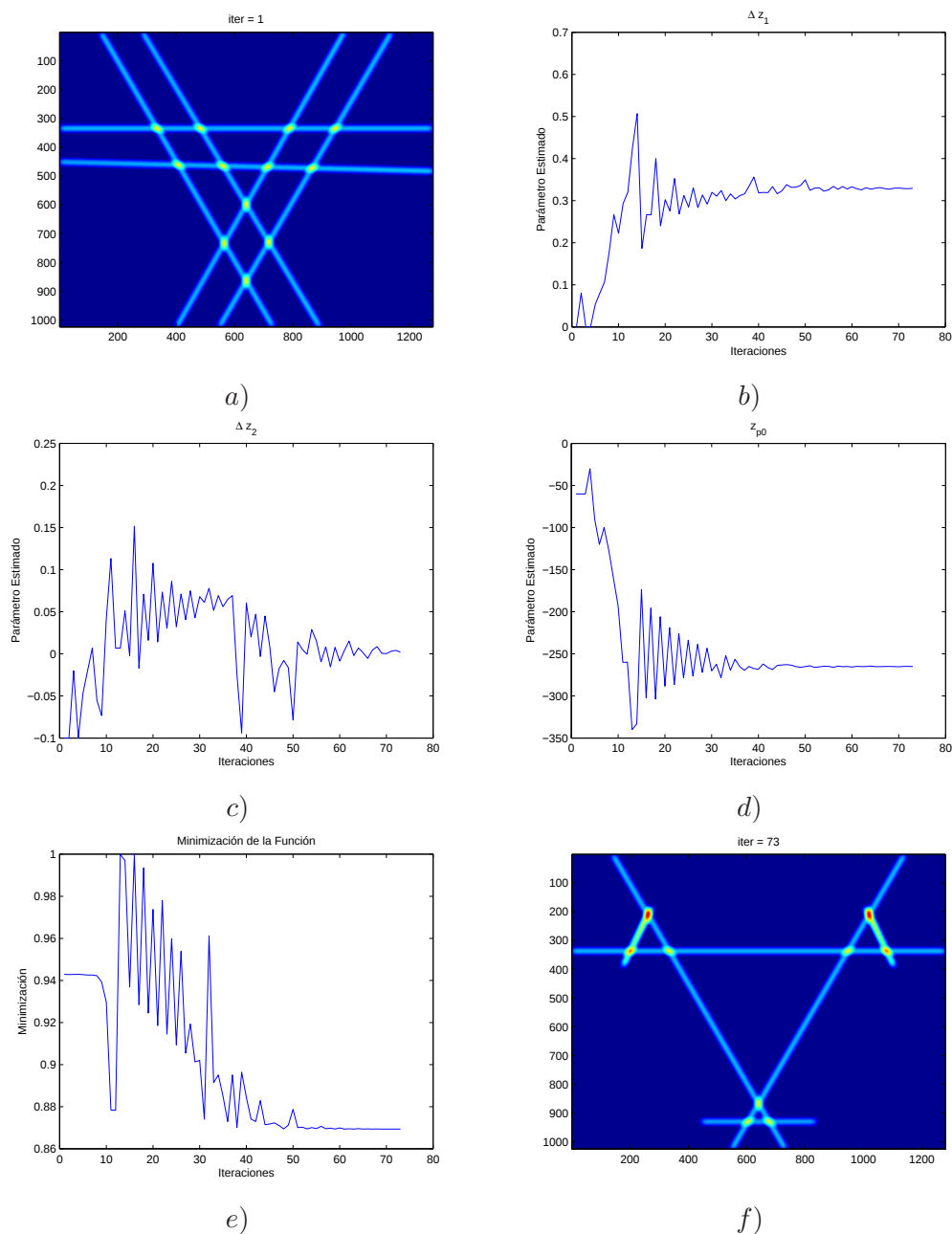


Figura 3.7: Imagen con mínimos locales. a) Imagen de búsqueda, b) registro del parámetro de Δz_1 , c) registro del parámetro de Δz_2 , d) registro del parámetro de z_{p0} , e) historial de la minimización de la función, f) final de búsqueda de información sobre la imagen.

Finalmente, se muestra una búsqueda errónea de información sobre la imagen, ésta se presenta en las imágenes de la figura 3.8.

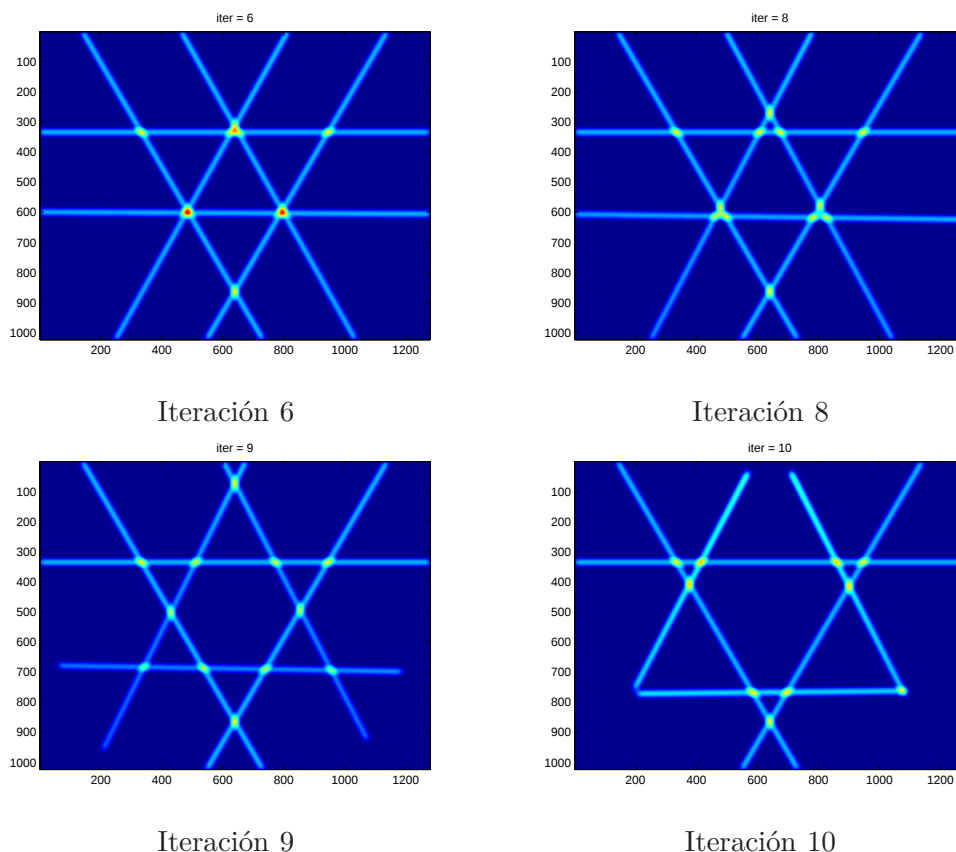


Figura 3.8: Evolución de la búsqueda errónea.

3.1.1.2. Prueba de Múltiples Imágenes

Como se había planteado anteriormente, se diseña una prueba en donde se brindan varias imágenes al método para que, en cada una de ellas, se detecte el parámetro correcto. Las imágenes serán generadas con datos conocidos de tal forma que de una imagen a otra solo varíe un dato, y que como historia de todas ellas lleguen a describir un barrido de una superficie como si fuera un caso real de estudio. Se espera que la primer imagen de búsqueda sea como los casos anteriores y, después de tener parámetros cercanos de una imagen a la otra, la detección de cada información sobre la imagen se llegue a realizar de forma rápida. La intención de la prueba es enfocada en determinar el tiempo que logre procesar una serie de imágenes y determinar la rapidez del método.

A continuación en la tabla 3.9 se muestra el historial de imágenes que serán proporcionadas al método de optimización, en las cuales se podrán identificar las variaciones que sufren de una imagen a otra.

En el cuadro 3.10 se muestran los valores de los parámetros con los cuales han sido generadas las imágenes de la tabla 3.9, cada imagen está denotada por un inciso que

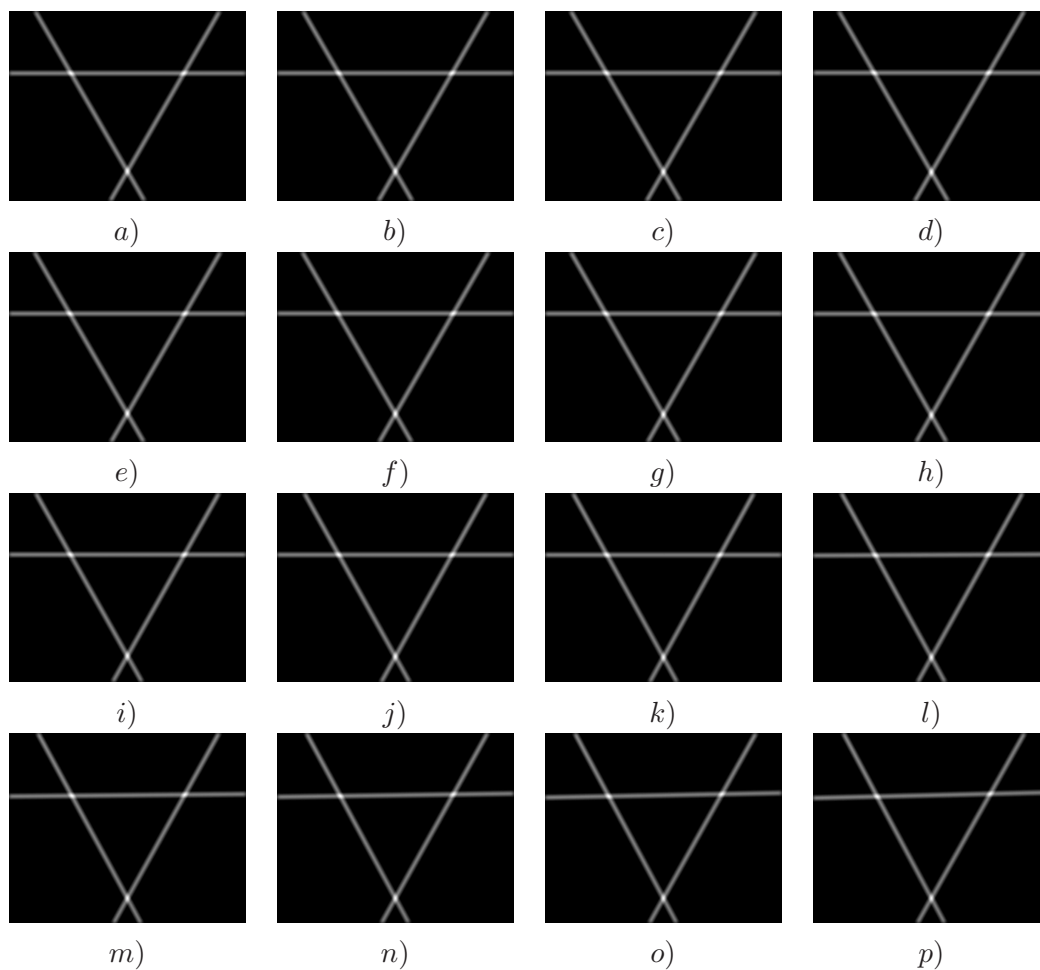


Figura 3.9: Secuencias de imágenes para el optimizador. Los incisos de cada imagen corresponden a los valores del cuadro 3.10

permite ubicarlas respectivamente.

Imagen	a	b	c	d	e	f	g	h
z_{p0}	0	0.4	0.8	1.2	1.6	2	2	2
Δz_1	0	0	0	0	0	0	0.02	0.04
Δz_2	0	0	0	0	0	0	0	0

Imagen	i	j	k	l	m	n	o	p
z_{p0}	2	2	2	2	2	2	2	2
Δz_1	0.06	0.08	0.1	0.1	0.1	0.1	0.1	0.1
Δz_2	0	0	0	0.02	0.04	0.06	0.08	0.1

Cuadro 3.10: Parámetros que describen cada imagen de prueba.

Las variaciones que se observan para el z_{p0} están dadas en milímetros, mientras que

para Δz_1 y Δz_2 están dadas en incrementos de $1,14^\circ = 0,02$ radianes, que sería equivalente decir de 0° a $5,7295^\circ$. Como las variaciones son pequeñas, no se percibe de forma visual los cambios entre las imágenes de la figura 3.9.

La búsqueda de los parámetros en las imágenes se realiza a través de las dos metodologías empleadas anteriormente; una búsqueda sobre la imagen con parámetros que no cambian, como lo es la tolerancia y la otra que consiste en realizar una búsqueda escueta con parámetros burdos sobre las imágenes y después una segunda detección de parámetros de forma más fina.

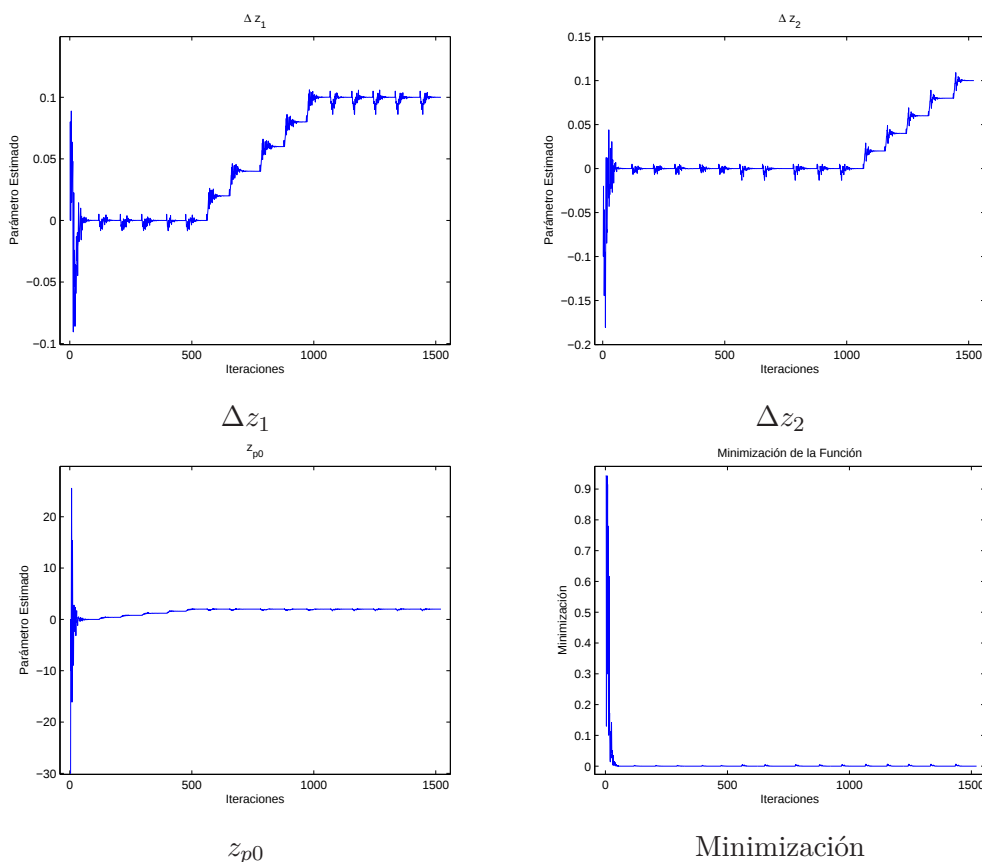


Figura 3.10: Detección de parámetros de múltiples imágenes.

Como se logra apreciar en las gráficas mostradas en la figura 3.10, al inicio de las iteraciones se logran observar brinco demasiados grandes hasta detectar información correcta sobre la imagen para el parámetro z_{p0} , en la que son cuatro imágenes con variaciones y después se queda constante. Después continua Δz_1 con cuatro variaciones y seguido de ellas permanece constante y finalmente Δz_2 con la misma secuencia. Como el proceso de minimización en las primeras iteraciones refleja incrementos muy altos y después que ya encuentra la información correcta queda estable, y en cada cambio de imagen se registra mínimas variaciones.

3.2. Pruebas de calibración a partir de imágenes virtuales

Al momento se ha evaluado la rapidez del método para generar imágenes, así como el algoritmo de optimización que se está utilizando para realizar la búsqueda de información sobre una y múltiples imágenes. Ahora se emplea el mismo algoritmo de optimización para hacer la calibración del sistema, así como también la calibración de cámara. Como en la sección 2.1.1.1, en la pag. 27, se describía que el origen del sistema era arbitrario, ahora en esta sección se describirá como es la definición del mismo.

En la adquisición de las imágenes que serán usadas para calibrar tanto la cámara como al sistema, se habla de la necesidad de dos imágenes, donde el sistema de visión es llevado a una posición cualquiera sobre el plano de interés. A partir de la fijación de este punto en particular (sin conocer dicha coordenada) se realiza una variación en altura y, a partir de ello, una distancia positiva así como negativa haciendo conocida esta diferencia (Δh). En cada movimiento realizado se hace la adquisición de una imagen y, a partir de ello, se establece el origen del sistema del cual se emplea para calibrar. Esto es mostrado en la figura 3.11.

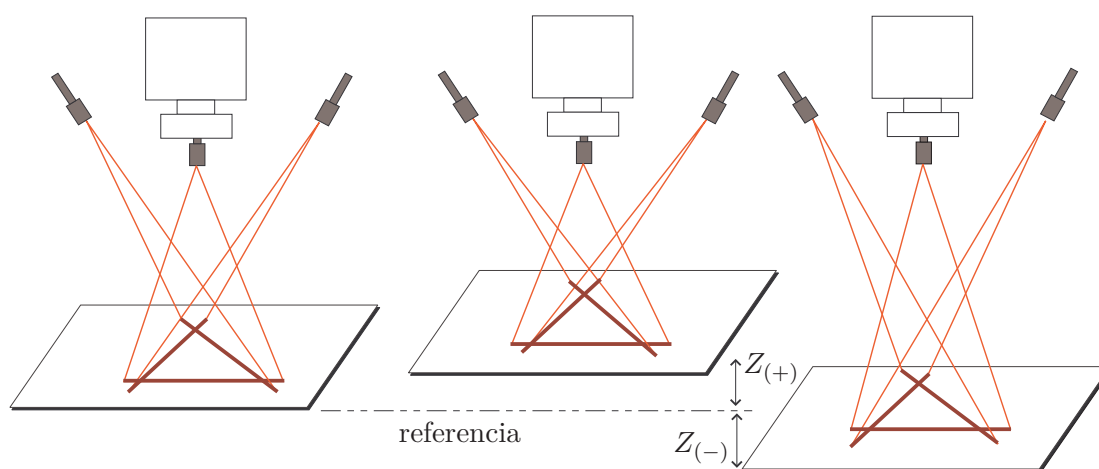


Figura 3.11: Método de calibración.

Esta metodología es aplicable tanto en la calibración de cámara como del cabezal de visión artificial.

3.2.1. Calibración de cámara

Para la calibración de cámara se usa el desarrollo descrito en la sección 2.5, pag. 45, haciendo uso de la matriz de calibración y creando una variable que permita desacoplar los efectos de magnificación y perspectiva. Los resultados que ha arrojado el algoritmo de optimización al momento son muy aceptables ya que dependiendo de la tolerancia con la

que se llega a buscar se genera un error que es muy bajo e inclusive llegan a coincidir los valores con los reales. Este proceso solo se ejecuta una vez para determinar los parámetros intrínsecos y no se vuelven a tocar en procesos posteriores, es por ello que sin importar el tiempo que éste proceso dure se realiza para una sola ocasión. Para que esto suceda así, entonces se deberían poner criterios demasiados altos para dar por concluido el proceso de calibración.

Como se esta realizando pruebas virtuales al momento para comprobar la validez del método, se da pie a generar imágenes virtuales una vez más que serán usadas para proporcionarlas al método como imágenes reales y después se busca información sobre ellas.

Se empleará la imagen del ajedrez para realizar la calibración de cámara y para ello se genera con 6 campos¹ y cada campo contiene 100×100 puntos, que es equivalente a utilizar 36000 puntos para generar el ajedrez con una buena resolución en sus campos. Y sus variaciones en información solo incluye dos cambios de altura. Si hay un número par de campos, entonces el centro de la imagen se encuentra donde coinciden cuatro campos y, si es impar, entonces el centro de la imagen coincide en el centro del campo.

Las imágenes que fueron creadas a partir del algoritmo generador para el ajedrez son mostradas en las figuras 3.12 y 3.13:

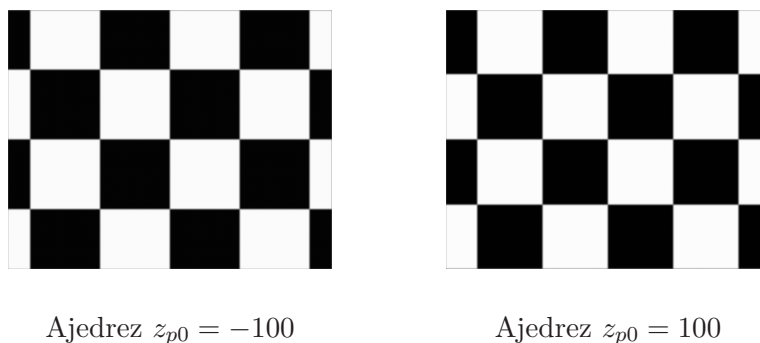


Figura 3.12: Imágenes para calibrar la cámara (par).

A continuación se muestran las imágenes en la tabla 3.14, cuando se realiza el proceso de calibración.

Como se logra ver en las imágenes de la figura 3.14, el calor azul cielo y marrón establece la coincidencia en cuadros del mismo color, entre más área de coincidencia ésta queda bien definido. Para el color amarillo, se muestra un contraste que sería equívoco en la superposición de un campo al que no corresponde. En medio proceso, se logra observar que el ajedrez está superpuesto en las dimensiones casi exactas a la imagen real que se utiliza.

¹El uso de números de campos así como números de puntos por campos puede variar indistintamente a como se desee, abría repercusiones en la digitalización de la imagen si se usan pocos puntos por campo.

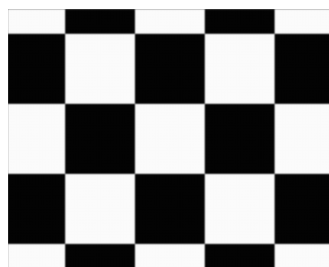
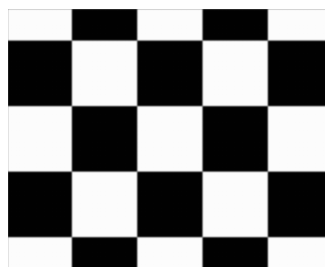
Ajedrez $z_{p0} = -100$ Ajedrez $z_{p0} = 100$

Figura 3.13: Imágenes para calibrar la cámara (impar).

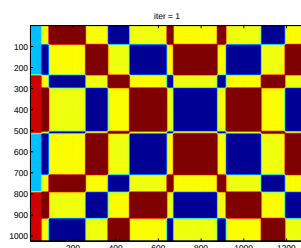
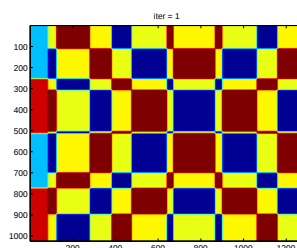
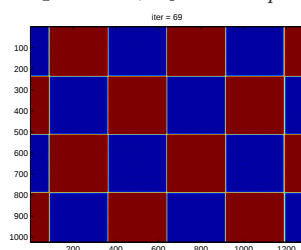
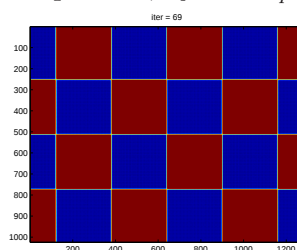
Inicio de proceso, ajedrez $z_{p0} = -100$ Inicio de proceso, ajedrez $z_{p0} = 100$ Medio proceso, ajedrez $z_{p0} = -100$ Medio proceso, ajedrez $z_{p0} = 100$

Figura 3.14: Imágenes en el proceso de calibración.

La información de los parámetros de cámara, con la cual fue creada la imagen para el proceso de calibración, está contenida en la tabla 3.11 en la columna de *propuesto*, mientras que del algoritmo de optimización se obtuvieron los valores en la columna de *determinado*.

Como habrá de notarse, los valores con los que fue creada la imagen y los valores de los parámetros determinados por el algoritmo de optimización son muy similares, es decir, con errores muy aceptables.

Parámetro	Propuesto	Determinado
u_0	640	639,9991
v_0	512	512,0751
z_{cam}	30000	$3,0171 \times 10^3$
k_u	$2,6667 \times 10^4$	$2,6823 \times 10^4$
k_v	$2,6667 \times 10^4$	$2,6849 \times 10^4$

Cuadro 3.11: Parámetros de cámara.

3.2.2. Calibración del cabezal

La situación para el cabezal que contiene los láseres de línea es diferente, debido a que el ángulo con el que se proyectan los láser puede variar de un momento a otro por cuestiones físicas que logren suceder a éste, como puede ser un golpe que logre desajustar el ángulo e inclusive la posición del cabezal sobre la cámara. Así como una agente externo que de alguna manera llegue a realizar modificaciones al cabezal y por seguridad misma de pruebas. Es por ello que se debe de buscar alternativas que haga dicho proceso de forma rápida y eficiente.

Para ello se utilizará dos alternativas en la calibración de los láseres y en las dos propuestas se deben de realizar varias búsquedas de información.

1. Utilizar una imagen para buscar la orientación de los láseres sobre la superficie proyectante usando líneas de búsqueda muy gruesas. Seguido de éste proceso se emplearían dos imágenes para realizar la búsqueda del ángulo con el que se proyecta el láser y su radio, dicha información se proponen con parámetros grandes. Una tercera búsqueda de información consiste en hacer uso de la orientación, ángulo de proyección y radio de los láseres en forma fina².

Método de Calibración 1					
	φ	ψ	r_l	#Imágenes	Filtros
Búsqueda 1	X_g	0	0	1	Ancho
Búsqueda 2	X_m	X_m	X_m	2	Medio
Búsqueda 2	X_f	X_f	X_f	2	Fino

Cuadro 3.12: Parámetros de búsqueda, primer método .

2. Emplear dos imágenes para búsqueda de la orientación, ángulo de proyección y radio de los láseres, dando mayor peso de búsqueda a la orientación, mientras que, el ángulo de proyección es medio y el radio en forma fina. En esta etapa se usan líneas muy

² El subíndice f , m y g se refieren a valores *finos*, *medios* y *grandes* respectivamente.

gruesas para realizar menos iteraciones y abarcar más área. En la segunda búsqueda se proponen parámetros finos y se reduce el ancho de las líneas pensando en que es muy probable la determinación de la información.

Método de Calibración 2					
	φ	ψ	r_l	$\#Imágenes$	Filtros
Búsqueda 1	X_g	X_m	X_f	2	Ancho
Búsqueda 2	X_f	X_f	X_f	2	Fino

Cuadro 3.13: Parámetros de búsqueda, segundo método.

Finalmente, como un bonus extra al proceso de calibración, se propone un tercer método que no tienen nada en especial, solo que no esta tan definido como los anteriores. Es decir, se proponen valores para iniciar la búsqueda sin cuidar detalles de incrementos finos o muy grandes, sino que se proponen valores medios y con algo de sentido común y para el proceso hace uso de dos imágenes. Esta última propuesta solo sirve para tener más condiciones para evaluar la metodología de calibración y de ellos se podrá tener otro criterio para comparar los resultados.

De los métodos de calibración propuestos se desprenden varias condiciones posibles:

1. Que los tres métodos arrojen valores muy cercanos o parecidos.
2. Cualquier combinación, entre dos métodos arrojen valores semejantes y el otro sea diferente.
3. Ningún método tenga valores semejantes.

Las variaciones anteriores se dan por que cada método modifica un parámetro distinto de búsqueda sobre las imágenes.

3.2.2.1. Primer método de búsqueda sobre la imagen

Como se describía en la sección anterior, primero se usa una sola imagen para buscar la orientación de los láseres, y en la figura 3.16 se muestra algunas imágenes de la evolución del parámetro en los láseres.

Subsecuentemente de éste proceso, se utilizan dos imágenes para determinar el ángulo de incidencia de los láseres sobre el plano proyectante, así como su radio.

Al final de cada proceso se logra observar los láseres finos usados para llegar a los valores final de calibración, que serán usados en procesos posteriores para el procesamiento de una secuencia de imágenes.

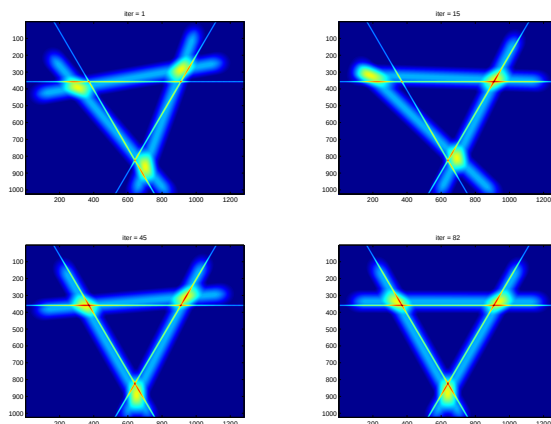


Figura 3.15: Primer propuesta para la calibración de láseres en la primera etapa de búsqueda con una sola imagen.

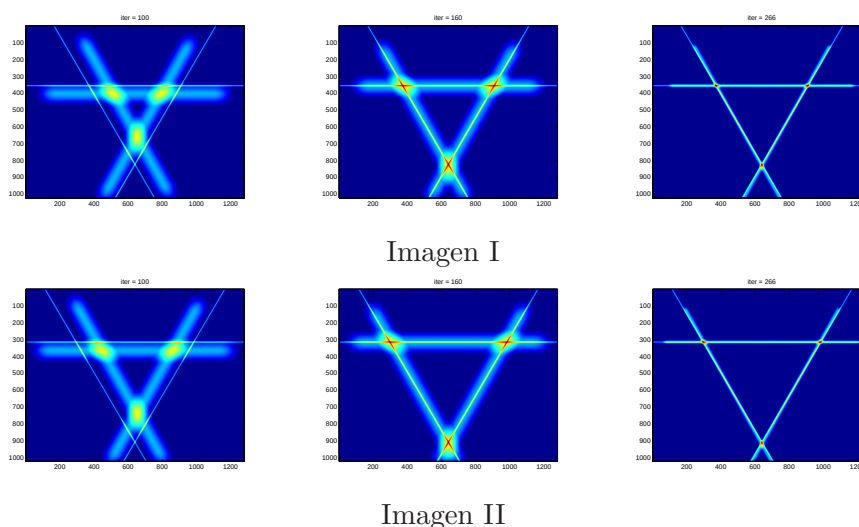


Figura 3.16: Primer propuesta para la calibración de láseres en su segunda etapa de búsqueda usando doble imagen.

3.2.2.2. Segundo método de búsqueda sobre la imagen

Se propone realizar dos etapas de búsquedas sobre la imagen, la primera usando máscaras del filtros muy grandes, que permita detectar el láser de línea de forma rápida con incrementos de búsqueda elevados y una tolerancia grande, esto se muestra en las imágenes de la figura 3.17 a) y b) para la primer imagen, y e) y f) para la segunda. Después que se haya ubicado la posición y orientación del láser sobre la imagen, se podría realizar una búsqueda más fina que permita detectar la información de interés más detallada.

En la segunda etapa, se reduce el tamaño del filtro para las líneas de búsqueda, figura

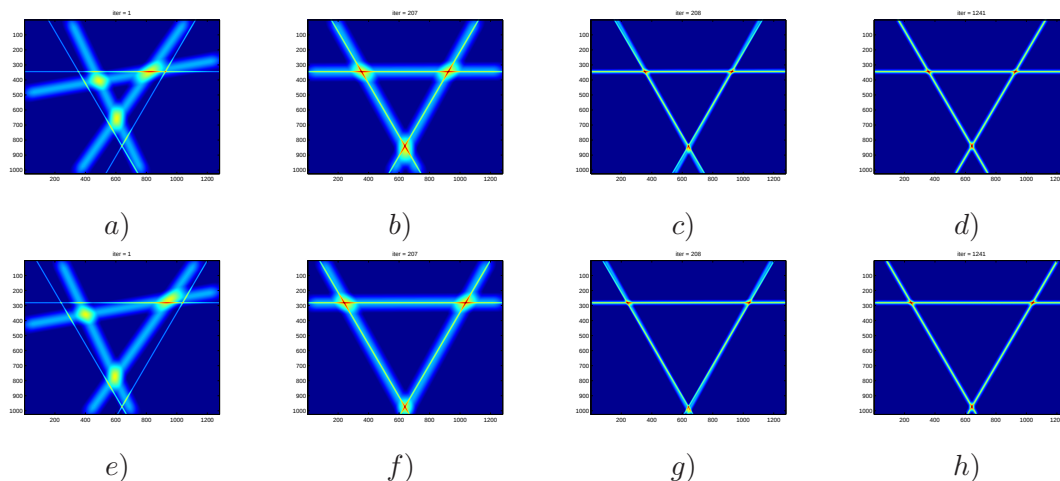


Figura 3.17: Segunda propuesta para la calibración de láseres en sus etapa de búsqueda usando doble imagen. *a)–d)* corresponde a la primer imagen, *e)–h)* corresponde a la segunda imagen.

3.17 *c)* y *d)* para la primera imagen y *g)* y *h)* para la segunda. Ésta etapa usa una tolerancia elevada e incrementos muy pequeños.

Finalmente, se ha logrado realizar la calibración de los láseres contenidos sobre el cabezal de visión, esta información se guarda en un archivo como base de datos para ser llamado en el procesamiento de imágenes.

Los valores que se propusieron para generar la imagen están incluidos en la tabla 3.14, siendo tres parámetros de cada láser:

Parámetro	láser 1	láser 2	láser 3
Valores Propuestos			
φ	0°	120°	240°
ψ	20°	20°	20°
r_o	55 mm	55mm	55mm
Valores Determinados			
φ	$0,0018^\circ$	$119,9991^\circ$	$239,9999^\circ$
ψ	$20,0009^\circ$	$19,9741^\circ$	$20,0002^\circ$
r_o	$55,0609\text{mm}$	$54,9118\text{mm}$	$54,9258\text{mm}$

Cuadro 3.14: Parámetros de láseres.

Como se puede observar en los valores de la tabla 3.14, los parámetros con los que se generaron la imagen, se logran recuperar con una gran exactitud por medio del algoritmo de optimización y el procesamiento de las imágenes.

Lo que habrá que tomar en cuenta es que, como son imágenes virtuales, la información es conocida y exacta, y por ende, se tiene la certeza de la misma. En el parámetro de r_0 estimado desde el eje focal de la cámara al láser. A su vez, el eje focal de cámara coincide con el origen que se establece para el tablero de ajedrez.

En la parte experimental, quien habrá de determinar el eje óptico del sistema es tablero de ajedrez, entonces es necesario que se logre situar lo más próximo en el centro de la imagen de cámara. Si esto no se logra realizar, el parámetro de r_0 para los tres láseres será muy próximo, lo que conlleva a que esta distancia sea mayor para un láser y menor para los demás. Éste parámetro también afecta al ángulo de proyección φ de los láseres que, aunque éstos tengan una medición física, pueden no corresponder al valor determinado por la calibración pero que sin embargo siguen siendo correctos los valores.

Con ésta etapa se da por concluida la calibración tanto del sistema de visión como del cabezal de visión artificial. Quedando toda la información contenida en un archivo que después será proporcionado a un programa general que realiza la búsqueda de información en una secuencia de imágenes adquiridas de cámara. Para determinar la velocidad del procesamiento del flujo de imágenes, se realiza una secuencia de imágenes virtuales y se procesan como si fuera el caso real de estudio y se obtienen gráficas de evolución de los parámetros de búsqueda. Para ello, cada información sobre las imágenes virtuales son conocidas en todo momento y se verificará con el algoritmo de optimización si dichos valores corresponden, y la exactitud en la determinación de información nuevamente dependen de la tolerancia con la cual se realiza la búsqueda. Este procesamiento de imágenes se muestra en un siguiente apartado.

3.2.3. Procesamiento de una secuencia de imágenes

Esta sección está dedicada a mostrar resultados de la evolución de parámetros de búsqueda sobre la secuencia de imágenes que se emplea. Los parámetros de interés son z_{p0} que da la altura o profundidad en la cual se encuentra el plano de proyección, Δz_1 y Δz_2 que describen el alabeo del plano de proyección (inclinación respecto a la cámara) y la función FO (descrita en la sección 1.4, pag. 17) que se busca minimizar, que es la calidad con la que se reproduce la imagen con respecto a la imagen original de búsqueda (porcentaje de error).

Si se realiza un acercamiento a las gráficas mostradas en la figura 3.18 se logra observar el comportamiento de la evolución del parámetro.

A continuación, se muestra la información que se utilizó para formar una secuencia de imágenes virtuales, que posteriormente, el método de optimización debería de entregar resultados muy cercanos a los que se muestran en la tabla 3.15.

La información que determinó el algoritmo de optimización de las imágenes que se le proporcionaron es mostrado en la tabla 3.16:

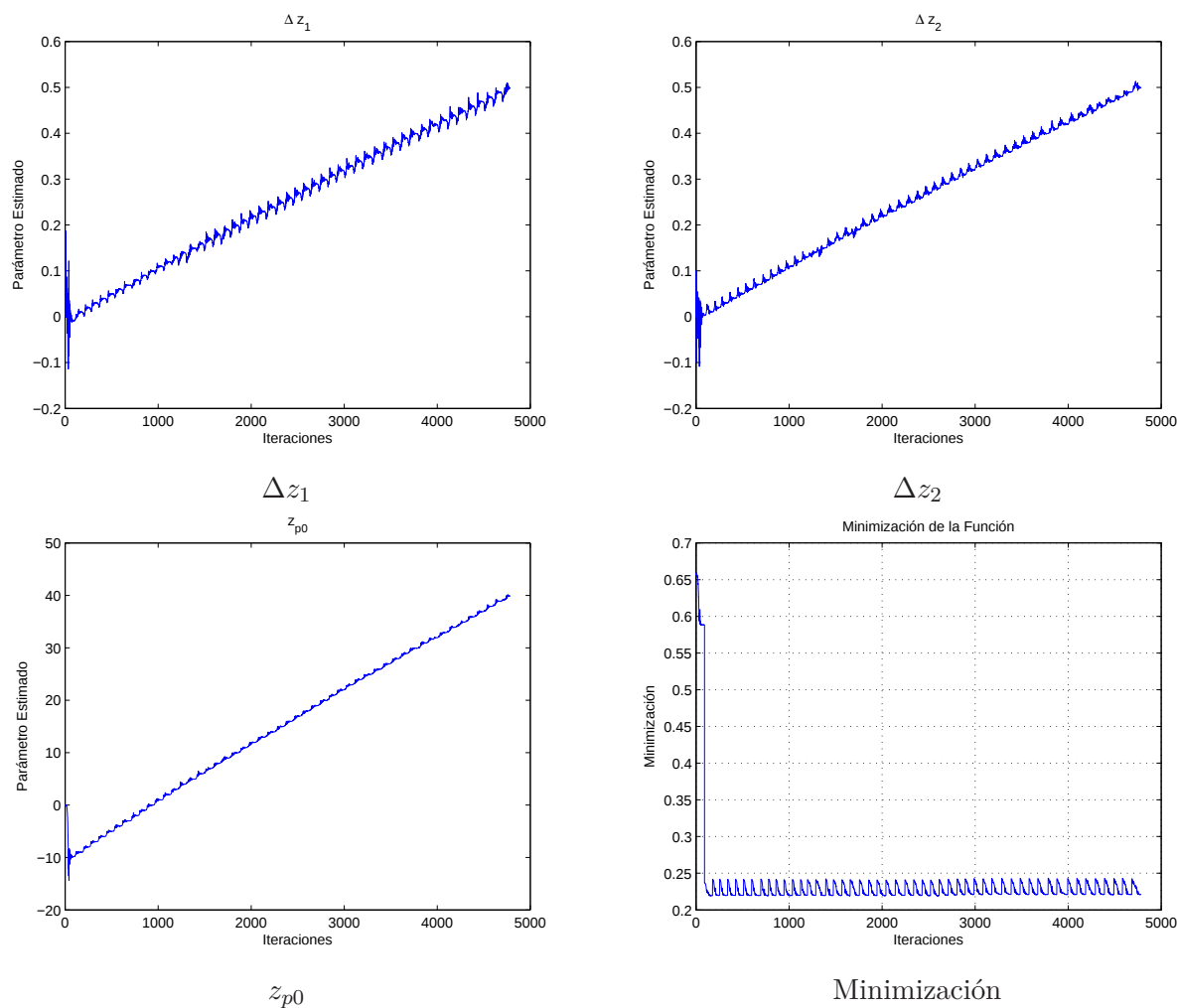


Figura 3.18: Detección de parámetros en una secuencia de imágenes virtuales.

Como se puede observar en las tablas mostradas en los cuadros 3.15 y 3.16, los valores propuestos y los valores determinados son altamente equiparables, con ello se logra establecer una fiabilidad muy elevada en los resultados que se logran obtener, por medio del método que se propone, para encontrar la posición del cabezal respecto a la superficie de proyección. Conforme al método se le exija mayor precisión, entonces el método se hace más lento y si es menor precisión, entonces cabe la posibilidad que la información que se determina de la imagen no se muy precisa.

En la tabla de datos mostrados en el cuadro 3.17, se calcula el porcentaje de error que se obtiene entre la información que se utiliza para generar las imágenes virtuales y los datos que se llega a obtener a través del algoritmo de optimización en la secuencia de imágenes.

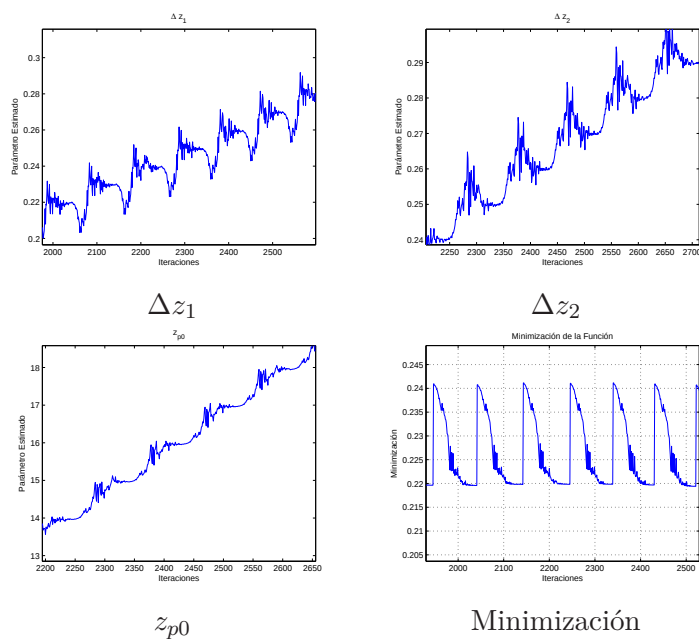


Figura 3.19: Detección de parámetros en una secuencia de imágenes virtuales a detalle.

z_{P0}	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4
Δz_1	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1	0.11	0.12	0.13	0.14
Δz_2	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1	0.11	0.12	0.13	0.14
z_{P0}	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Δz_1	0.15	0.16	0.17	0.18	0.19	0.2	0.21	0.22	0.23	0.24	0.25	0.26	0.27	0.28	0.29
Δz_2	0.15	0.16	0.17	0.18	0.19	0.2	0.21	0.22	0.23	0.24	0.25	0.26	0.27	0.28	0.29
z_{P0}	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
Δz_1	0.3	0.31	0.32	0.33	0.34	0.35	0.36	0.37	0.38	0.39	0.4	0.41	0.42	0.43	0.44
Δz_2	0.3	0.31	0.32	0.33	0.34	0.35	0.36	0.37	0.38	0.39	0.4	0.41	0.42	0.43	0.44
z_{P0}	35	36	37	38	39	40									
Δz_1	0.45	0.46	0.47	0.48	0.49	0.5									
Δz_2	0.45	0.46	0.47	0.48	0.49	0.5									

Cuadro 3.15: Valores de parámetros con los cuales fueron creadas las imágenes virtuales para el procesamiento de la secuencia.

z_{P0}	-9.81462	-8.99796	-7.99401	-6.98935	-5.99899	-4.99759	-4.00111	-3.01455	-2.00946	-1.00506
Δz_1	-0.00909	0.00987	0.01996	0.02984	0.0398	0.04983	0.05966	0.06873	0.07983	0.08986
Δz_2	0.00301	0.01048	0.02026	0.03021	0.04022	0.05032	0.06003	0.07012	0.07994	0.08992
z_{P0}	-0.00557	0.98678	1.99143	2.99089	3.98303	4.95495	5.97809	6.97669	7.9681	8.97412
Δz_1	0.09962	0.10964	0.11957	0.12996	0.13964	0.14896	0.15943	0.16937	0.17949	0.18949
Δz_2	0.10016	0.11009	0.1202	0.13004	0.13994	0.15023	0.15996	0.17007	0.18028	0.19009
z_{P0}	9.97774	10.97138	11.97228	12.95886	13.96379	14.96031	15.96214	16.96541	17.95034	18.95763
Δz_1	0.19936	0.20954	0.21968	0.22975	0.23953	0.24921	0.25933	0.26961	0.27966	0.28936
Δz_2	0.20002	0.20993	0.22016	0.23045	0.24007	0.2498	0.25972	0.26971	0.27993	0.28971
z_{P0}	19.95046	20.96076	21.94503	22.94598	23.93416	24.93292	25.93029	26.92407	27.92553	28.92269
Δz_1	0.29917	0.30923	0.31929	0.32928	0.33895	0.34877	0.35893	0.36889	0.37906	0.3888
Δz_2	0.29991	0.30967	0.31957	0.32969	0.33975	0.34962	0.35985	0.36974	0.37961	0.38947
z_{P0}	29.91439	30.92223	31.91893	32.91483	33.91473	34.91224	35.9058	36.90987	37.90328	38.90202
Δz_1	0.39882	0.40854	0.41901	0.42888	0.43898	0.44893	0.45883	0.46869	0.4785	0.48922
Δz_2	0.39927	0.40939	0.41953	0.42933	0.43954	0.44945	0.45951	0.46945	0.47933	0.4899
z_{P0}	39.90082									
Δz_1	0.49881									
Δz_2	0.49913									

Cuadro 3.16: Determinación de parámetros de las imágenes virtuales en el procesamiento de la secuencia.

z_{P0}	1.8538	0.022667	0.074875	0.152143	0.016833	0.0482	0.02775	0.485	0.473	0.506
Δz_1	0	1.3	0.2	0.533333	0.5	0.34	0.566667	1.814286	0.2125	0.155556
Δz_2	0	4.8	1.3	0.7	0.55	0.64	0.05	0.171429	0.075	0.088889
z_{P0}	0	1.322	0.4285	0.303667	0.42425	0.901	0.365167	0.333	0.39875	0.287556
Δz_1	0.38	0.327273	0.358333	0.030769	0.257143	0.693333	0.35625	0.370588	0.283333	0.268421
Δz_2	0.16	0.081818	0.166667	0.030769	0.042857	0.153333	0.025	0.041176	0.155556	0.047368
z_{P0}	0.2226	0.260182	0.231	0.316462	0.258643	0.2646	0.236625	0.203471	0.275889	0.223
Δz_1	0.32	0.219048	0.145455	0.108696	0.195833	0.316	0.257692	0.144444	0.121429	0.22069
Δz_2	0.01	0.033333	0.072727	0.195652	0.029167	0.08	0.107692	0.107407	0.025	0.1
z_{P0}	0.2477	0.186857	0.249864	0.23487	0.274333	0.26832	0.268115	0.281222	0.265964	0.266586
Δz_1	0.276667	0.248387	0.221875	0.218182	0.308824	0.351429	0.297222	0.3	0.247368	0.307692
Δz_2	0.03	0.106452	0.134375	0.093939	0.073529	0.108571	0.041667	0.07027	0.102632	0.135897
z_{P0}	0.285367	0.250871	0.253344	0.258091	0.250794	0.250743	0.261667	0.243595	0.254526	0.251231
Δz_1	0.295	0.356098	0.235714	0.260465	0.231818	0.237778	0.254348	0.278723	0.3125	0.159184
Δz_2	0.1825	0.14878	0.111905	0.155814	0.104545	0.122222	0.106522	0.117021	0.139583	0.020408
z_{P0}	0.24795									
Δz_1	0.238									
Δz_2	0.174									

Cuadro 3.17: Cálculo del porcentaje de error entre los valores reales y determinados por algoritmo de optimización.

Capítulo 4

Integración física de cámara y manipulador

En el desarrollo de este trabajo se contempla una cámara, unos diodos emisores de luz láser y un manipulador serial del tipo industrial. Tanto la cámara como los láser son integrados en uno mismo en una estructura de aluminio, que será montado sobre la lente de la cámara y a todo este conjunto se denomina como *cabezal de visión*. Dicho interfaz será integrado con el manipulador serial para que pueda ser manipulado a través de este, para poder realizar la validación de las pruebas.

El trabajo contempla un diseño de una estructura de aluminio para montar los láser junto con la cámara y una base que permitirá integrar el sistema con el robot serial. Además, es considerado una interfase a través del software entre el algoritmo de cálculo (generación y comparación de imagen) y el software de control para el manipulador. Cabe hacer mención que la interfase de software será entre dos lenguajes de programación distintos y es por ello que también toma gran relevancia en todo esto.

Los elementos que fueron necesarios para desarrollar el trabajo son incluidos en la lista siguiente:

- Cámara Pixelink.
- Lente de resolución macroscópica.
- Diodos láser.
- Cabezal
- Manipulador serial MOTOMAN HP3 de YASKAWA Company.

4.1. Desarrollo del cabezal

El cabezal se podría definir como un dispositivo, que generalmente es móvil, y es usado para colocar en él una herramienta que elabora o manipula una tarea determinada. Dentro

de los cabezales se podría tener el proceso de manufacturar pero también se podría controlar algún proceso determinado. Este proyecto que se desarrolla esta enfocado a realizar un control en tiempo real de los movimientos de un manipulador serial.

La integración de los elementos, que han sido enlistados en la sección anterior, permiten generar un cabezal de visión, el cual es montado en un manipulador serial, y para poderlo fijar sobre el manipulador se diseña una interface que permita su fácil montaje y desmontaje. Para generar la pieza se recurre a un programa de diseño CAD que permite primeramente visualizar parte del ensamblaje de las piezas de forma virtual y después, para realizar el diseño de la interfaz mecánica entre cabezal y robot.

Tomando en cuenta primero el cabezal, se puede referir a la imagen mostrada en la figura 4.1, donde se logra ver diversas vistas del cabezal ya ensamblado de manera virtual.

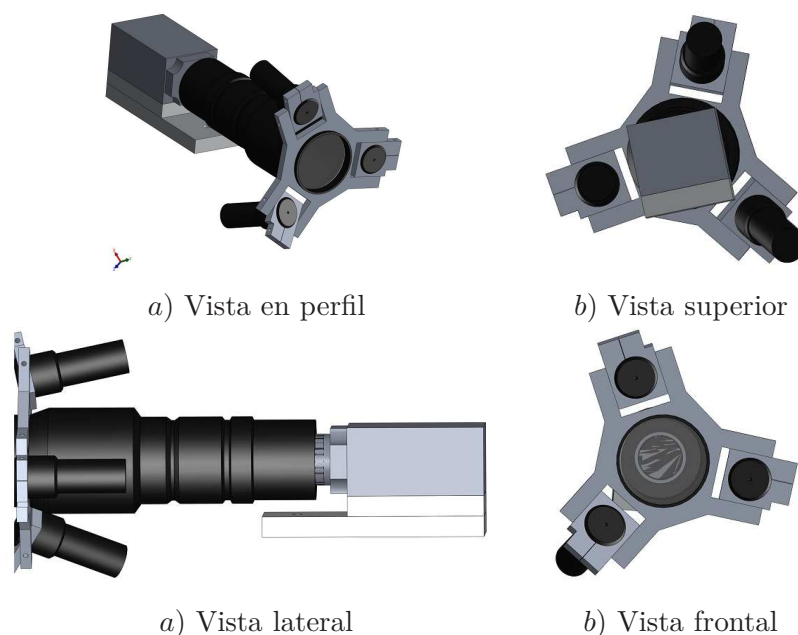


Figura 4.1: Diseño virtual del cabezal ensamblado.

En los siguientes temas se describirá y mostrará por separado las piezas que integran el cabezal de visión.

4.1.1.1. Cámara

Dentro de la fabricación y construcción de cámaras en el mercado es posibles catalogarlas por medio de su tecnología y estriban en dos clases.

- Las cámaras del tipo CMOS(Complementary metal-oxide-semiconductor).
- Las cámaras CCD (charge-coupled device).

Dicha tecnología permite ofrecer características que para el consumidor pueden llegar ser importantes y de igual forma la aplicación. El principio de trabajo es el mismo para ambas, donde cada pixel del sensor de cámara es exitado a través de la energía luminosa que incide en él. El haz de luz exita a un fotosito ¹ y éste a su vez, produce una descarga eléctrica. Los CMOS integran electrónica en cada uno de los fotositos para poder amplificar la señal eléctrica y mandarla a la computadora, mientras que las CCD colectan la señal de un arreglo y después aplican electrónica para amplificar la señal a la salida del arreglo. Entonces, en los CMOS se puede tener un control en cada uno de los pixeles del sensor sobre la luminosidad, correcto de contraste y convertidor análogo-digital (www.teledynedalsa.com, 2012).

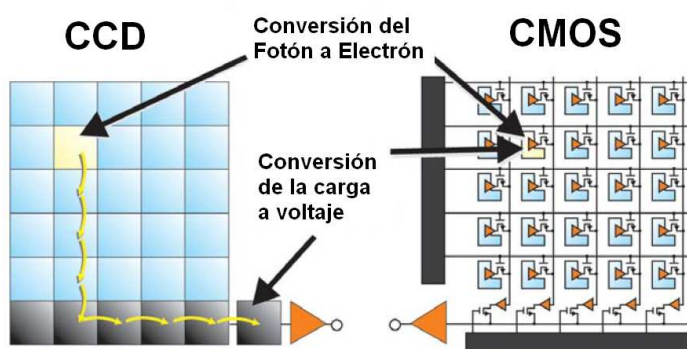


Figura 4.2: Construcción de sensores en cámaras CCD y CMOS.

La cámara que se emplea en este proyecto es de tipo CMOS por las características que llega a ofrecer, y es de tipo monocromática, es decir, de un solo canal en escala de grises.

El modelo de cámara es *PL-B741* de la marca *Pixelink*, donde la curva a la sensibilidad de luz es mostrada en la figura 4.3.

La línea continua es la que representa la curva de sensibilidad para este modelo, y en la gráfica de la figura 4.3 *b*) se puede apreciar que en la longitud de onda (λ) de $650nm$ es donde muestra mejor respuesta la cámara y para niveles abajo o superiores la eficiencia de cámara puede bajar.

En la imagen mostrada en la figura 4.4 se pueden ubicar el espectro de luz visible para el ojo humano, también se puede identificar los rangos de longitudes de onda que se encuentran por debajo y por encima del espectro visible para el ojo humano.

Algunas de las características de cámara que se podrían resaltar se encuentran en la tabla 4.1.

¹El fotosito es un receptáculo situado dentro del sensor digital que puede contener uno o más fotosensores. Además de sensor de imagen, el fotosito suele tener a los lados varias micro-lentes que ayudan a captar mejor la luz en sus distintos ángulos de incidencia.(www.fotonostra.com, 2013)

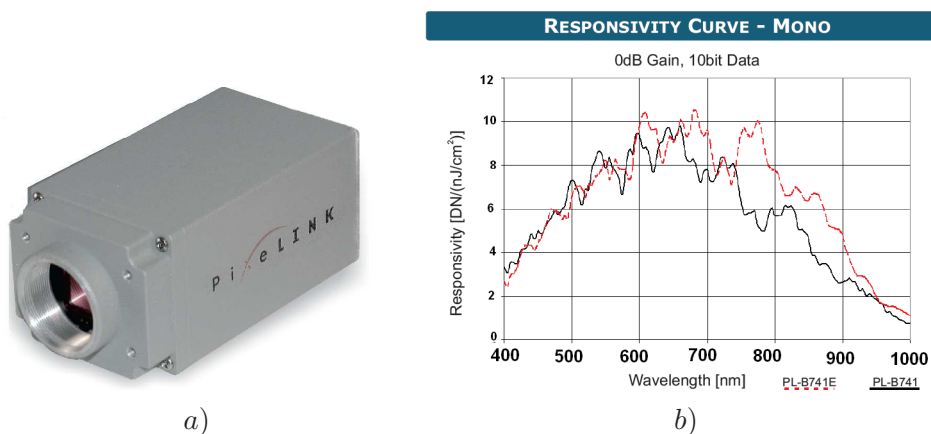


Figura 4.3: Cámaras Pixelink CMOS.

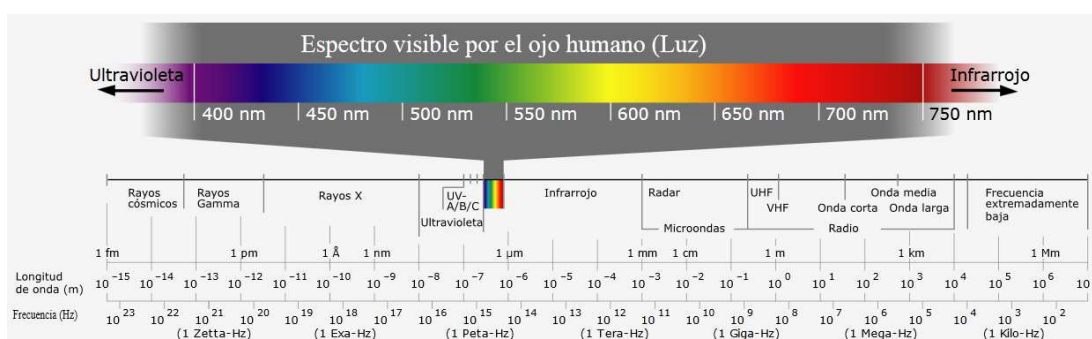


Figura 4.4: Espectro de luz (Wikimedia, 2012).

4.1.2. Lente de resolución macroscópica

La lente de resolución macroscópica *ZOOM 7000 NAVITAR TV ZOOM* de la línea NAVITAR es ensamblada con la cámara. Su potencia de acercamiento permite obtener buena imágenes y buena calidad.

En la imagen de la figura 4.5 es mostrada la imagen real de lente junto con su diseño CAD, que se realizó para el diseño del cabezal que será visto en un tema subsecuente.

Las características a resaltar de este artículo son mostrados en la tabla 4.2:

4.1.3. Diodos láser

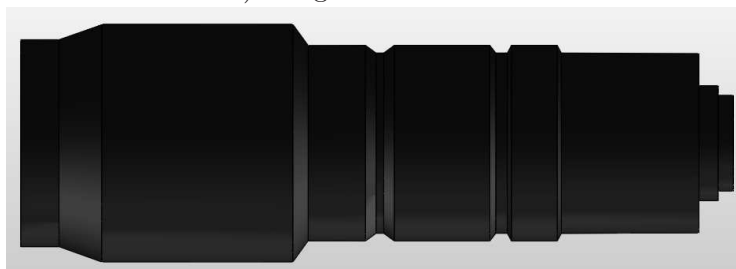
Los diodos emisores de luz láser contienen un módulo que permite enfocar el haz de luz (LDM), de tal forma que, de un punto de proyección lo llevan a crear una línea corta de espesor grueso hasta una línea suficientemente larga y delgada.

Datos Técnicos de la Cámara	
Sensor	Cypress IBIS 5B
Tamaño de píxel	$(6,7 \times 6,7) \mu m$
Resolución	$1280(H) \times 1024(V)$
Velocidad de obturador	27 Fps
Área activa	$(8,57 \times 6,86 - 11,01 \text{ diagonal}) mm$
Interface	IEEE 1394 (2) Puerto FireWire
Programación	Lenguaje C

Cuadro 4.1: Datos referidos de la hoja de datos.



a) Imagen real de lente



b) Imagen CAD de la lente.

Figura 4.5: Lente macroscópico.

Y la ecuación con la cual se puede determinar la longitud de línea es:

$$x = 2y \tan \frac{\vartheta}{2} \quad (4.1)$$

donde ϑ el ángulo del abanico de luz que forma la línea a distancia y desde el lente proyector de la fuente láser.

Aunque para los fines que se persiguen en éste trabajo no es considerado esta parte y solo se menciona como información del producto.

Datos Técnicos de la Lente NAVITAR	
Distancia focal	18 – 108 (6X)(cerrado-abierto)
Longitud Focal Posterior	24,28 mm
Diámetro del Filtro	$\Phi 52$ $P = 0,52$
Peso(gramos)	595
Posición de Pupila	238,6 (imagen de plano frontal)

Cuadro 4.2: Algunos datos importantes de la lente.

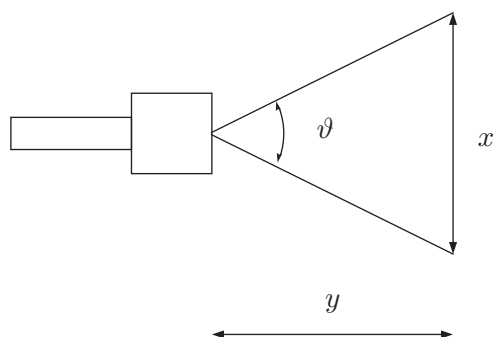


Figura 4.6: Láser de línea.

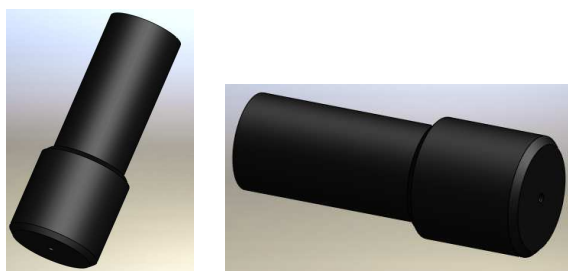


Figura 4.7: Diseño CAD de los láser y LDM.

4.1.4. Cabezal o interfaz

Como se desea montar tres láser alrededor de la lente de cámara, es suficiente un placa de aluminio, que por material de construcción, es ligero y resistente. Además, es necesario cuidar el peso en el manipulador en el actuador final para su manipulación, ya que si es muy pesado, puede generar inercia y puede tener una repercusión grande en los movimientos realizados deseables.

El diseño CAD de la pieza es mostrado en las imágenes de la figura 4.8, en varias

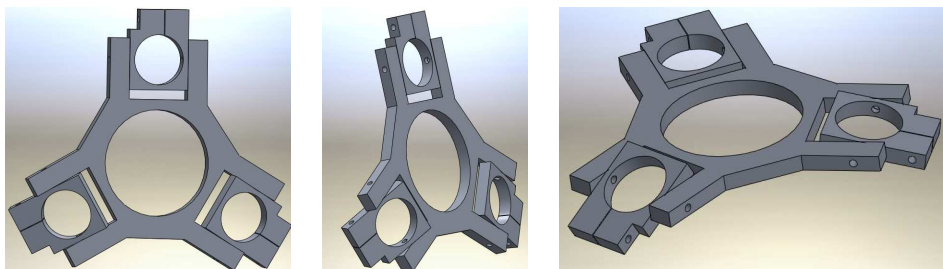


Figura 4.8: Diseño CAD del cabezal.

perspectivas. El cabezal contempla tres espacios para albergar láser con su LDM respectivamente, y están distribuidos en (120°) alrededor del cabezal. El espesor de la placa es de 10mm y es fabricada en material de aluminio.

Datos Técnicos del Cabezal	
Material	Aluminio
Espesor	10 mm
Distribución de láser	120°
Peso	0.2 Kg

Cuadro 4.3: Datos de cabezal.

4.2. Manipulador serial

El manipulador serial que fue empleado en el trabajo es de YASKAWA Company modelo MOTOMAN HP3 con 6 grados de libertad, tienen alcances en 1163 mm en el eje vertical y 701 mm en horizontal. Tiene una capacidad de carga de 3 kg y una repetibilidad de $\pm 0,03\text{ mm}$. Cuenta con un controlador de la serie *NXC100* y un control de mando “Teach-PendantT” y su ambiente de trabajo es Windows CE®. Cuenta con 2 puertos de comunicación que es el tipo “Ethernet” con (100 Mbps) y “RS232” tipo *C*.

4.3. Acoplamiento/interfaz/comunicación

La interacción entre software y hardware siempre se realiza a través de librerías desarrolladas por personal especializado en la creación software. Algunos productos son fabricados de tal forma que, su integración con software sea simple y genérico, mientras que la contra parte del hardware es para que sea uso exclusivo de algún lenguaje, o bien, que funcionen a través de llaves (hardware) que contienen código que permite manipular el componente



Figura 4.9: Robot serial MOTOMAN HP3 de YASKAWA Company.

por medio de algún lenguaje especializado. Otro de los casos comunes es, el de tener hardware que funciona en algunos sistemas operativos (Windows o Linux), o algunas versiones de una plataforma, como puede ser (*Windows XP*, *Windows Vista*, *Windows Seven*), esto comúnmente generan muchos problemas para realizar interfaces cuando se cuenta con hardware especializado y, de algún u otro modo, debe de buscarse alternativa que resuelva el problema.

El código de programación que es utilizado en este trabajo es Matlab ® (Laboratorio de Matrices). Matlab es un lenguaje de alto nivel y un entorno al cálculo numérico, visualización y programación. Se puede utilizar para una gama de aplicaciones, incluyendo el procesamiento de señales y comunicaciones, procesamiento de imágenes y vídeo, sistemas de control, prueba, medición y más. Este tipo de programas contiene controladores genéricos que permite la comunicación con hardware de cualquier tipo.

Se emplean los controladores genéricos que contiene Matlab para detectar la cámara Pixelink y, por medio de las funciones generales que ofrece el software, se puede acceder a todas las funciones de cámara. Habrá que resaltar que la cámara tiene sus controladores de software que solo son funcionales en plataforma de Windows XP. La razón por la cual se realiza una programación de código en este programa es por que, siendo un lenguaje de alto nivel, trae muchas funciones integradas en comando que permite una reducción en líneas de código.

Entonces, el planteamiento de la comunicación entre cámara y maquina está como lo refiere en la figura 4.10.

La arquitectura en un robot industrial es cerrada, por lo que, es necesario una llave de

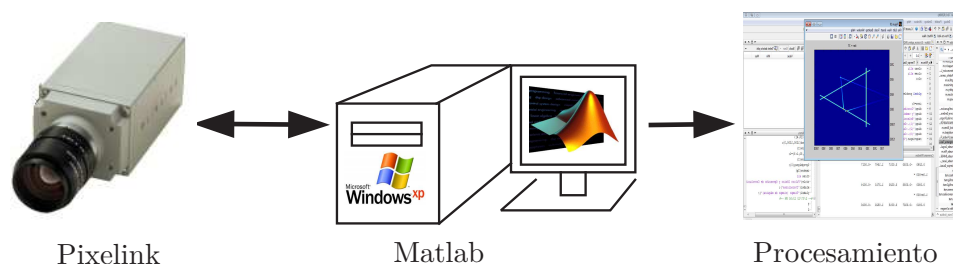


Figura 4.10: Conexión de cámara y computadora.

acceso a las funciones que se encuentran disponibles en lenguaje de programación en “C”. Entonces la interacción entre el manipulador y lenguaje C se visualiza como lo muestra la imagen de la figura 4.11.

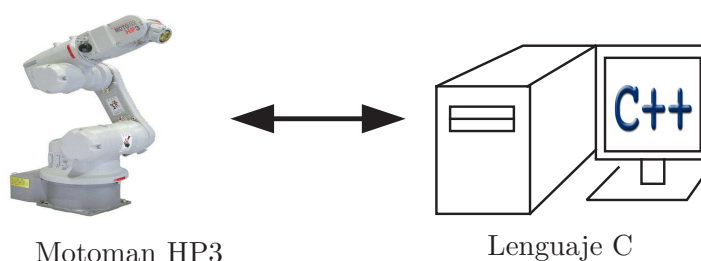


Figura 4.11: Conexión de manipulador y computadora.

La integración del equipo y software, en conjunto entre la parte que realiza la generación de imágenes y el control del manipulador serial, se hace a través de comunicación ethernet. La plataforma de Matlab permite realizar comunicación por medio de su puerto de comunicación ethernet, mientras que, la plataforma de lenguaje *C* también lo permite por medio de los llamados *Sockets*. La información estructurada que sale de una plataforma se puede recibir en la otra estación de trabajo y se puede entablar comunicación entre ambos. Los Sockets designa un concepto abstracto, por el cual, dos programas (posiblemente situados en computadoras distintas) pueden intercambiar cualquier flujo de datos, generalmente de manera fiable y ordenada. Un socket queda definido por un par de direcciones IP local y remota, un protocolo de transporte y un par de números de puerto local y remoto.

La familia de protocolos de Internet, es un conjunto de protocolos de red en los que se basa Internet y que permiten la transmisión de datos entre computadoras. En ocasiones se le denomina conjunto de protocolos TCP/IP, en referencia a los dos protocolos más importantes que la componen: Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP), que fueron dos de los primeros en definirse, y que son los más utilizados de la

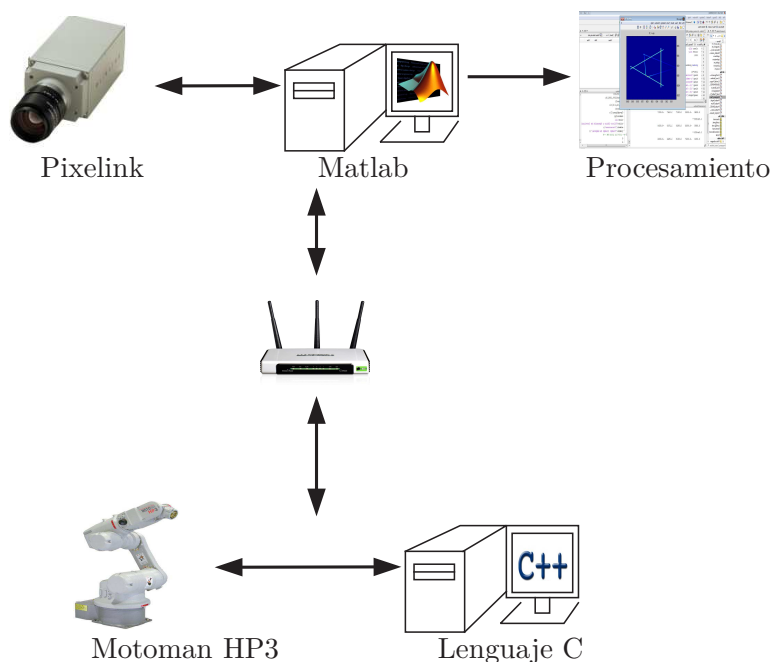


Figura 4.12: Integración de la Infraestructura.

familia. Existen tantos protocolos en este conjunto que llegan a ser más de 100 diferentes, entre ellos se encuentra:

- HTTP (HyperText Transfer Protocol), que es el que se utiliza para acceder a las páginas web.
- ARP (Address Resolution Protocol) para la resolución de direcciones.
- FTP (File Transfer Protocol) para transferencia de archivos.
- SMTP (Simple Mail Transfer Protocol).
- POP (Post Office Protocol) para correo electrónico.
- TELNET para acceder a equipos remotos.

Para poder entablar comunicación entre los dos equipos de trabajo se realizó una prueba de comunicación para un equipo. Se utiliza (para el caso de Matlab) dos sesiones en una misma maquina y se realiza una comunicación de una sesión a otra mediante un *localhost*. Localhost es un nombre reservado que tienen todas las computadoras, router o dispositivo independientemente de que disponga o no de una tarjeta de red ethernet. El nombre localhost es traducido como la dirección IP de loopback *127.0.0.1* en *IPv4* (Protocolo de Internet versión 4, 4.294.967.296 (2^{32}) direcciones de red diferentes), o como la dirección *::1* en *IPv6* (Protocolo de Internet versión 6, 670 mil billones de direcciones).

Se pueden plantear dos tipos de pruebas para la transmisión de datos, por medio de comunicación TCP/IP :

- Comunicación *localhost* (de sesión a sesión en una misma computadora)
- Comunicación o transferencia de datos de una pc a otra pc.

Los códigos pueden ser revisados y explicados en apartados para cada prueba en la sección de apéndices en A.1.

4.4. Integración del sistema

La integración del cabezal de visión con el manipulador se realiza por medio de una pieza elaborada en aluminio que debe de ser montada en el efector final del manipulador serial y éste mismo es mostrado en la imagen de la figura 4.13 *a)* y *b)*.

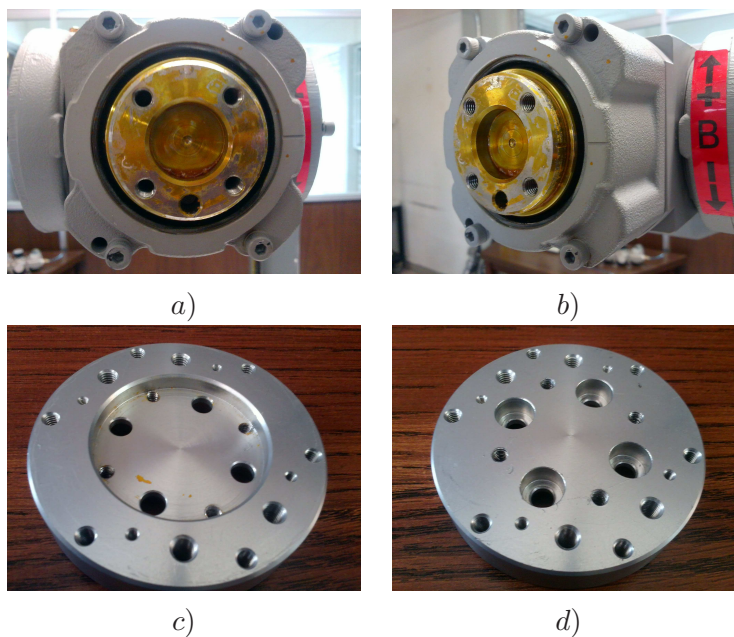


Figura 4.13: Placa de aluminio con la que cuenta el manipulador.

Las imágenes de *c)* y *d)* muestran el interface que comúnmente se utiliza en éste tipos de robots, y se puede observar que cuenta con un gran variedad de orificios a diferentes distancias, así como diámetros que permiten montar y desmontar objetos.

En este trabajo de tesis se hizo el diseño de una pieza en aluminio que permitiera poner y quitar el cabezal de visión de forma rápida y fácil. Esto mismo ayudaría a que otros dispositivos lograran hacer una sujeción pronta con el manipulador.

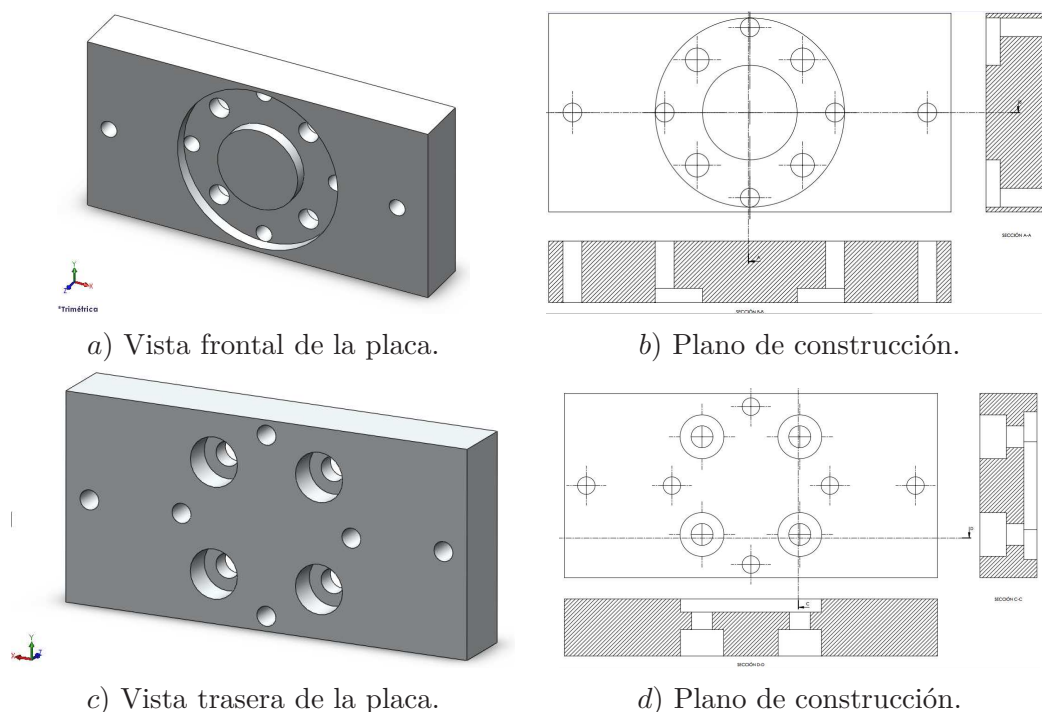


Figura 4.14: Diseño de la placa que irá montada en el robot serial de forma permanente.

Se hizo uso de un software de diseño CAD para su elaboración y previsualización. La pieza que sería usada para reemplazar por la que ya se tiene, es mostrada en la figura 4.14.

En la imagen de la figura 4.14 a) y c) se muestran las vista frontal y trasera de la pieza diseñada, mientras que en b) y d) se muestran los planos de construcción. Para la imagen en a) se observa la muesca sobre la pieza que permite embonar justamente con la última articulación del robot serial. En c) se aprecia la caja para los tornillos, que permiten sujetar esta pieza con el manipulador. Los orificios que se encuentran a una distancia un poco mayor son usados para poder sujetar alguna pieza de trabajo directamente al manipulador, mientras que, los dos orificios más lejanos (horizontales) son empleados para hacer la sujeción de una segunda pieza que complementa la interface entre robot y cabezal de visión.

La pieza diseñada, fabricada y montada sobre el manipulador se puede ver en la figura 4.15.

La segunda pieza que se propuso para la integración del sistema, es la que se muestra en la figura 4.16, se visualiza en una vista frontal a), así como la vista trasera b), junto con sus planos de construcción. Ésta pieza debe de ser fija en el cabezal, para éste caso.

En las imágenes mostradas en la figura 4.17, se puede observar como esta placa se encuentra sujeta en el cabezal de visión.

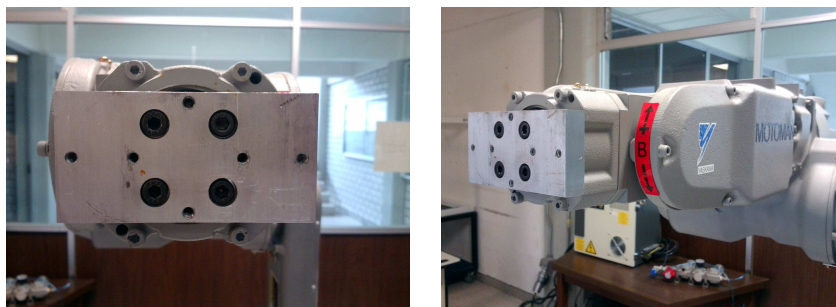


Figura 4.15: Placa construida y montada sobre el robot.

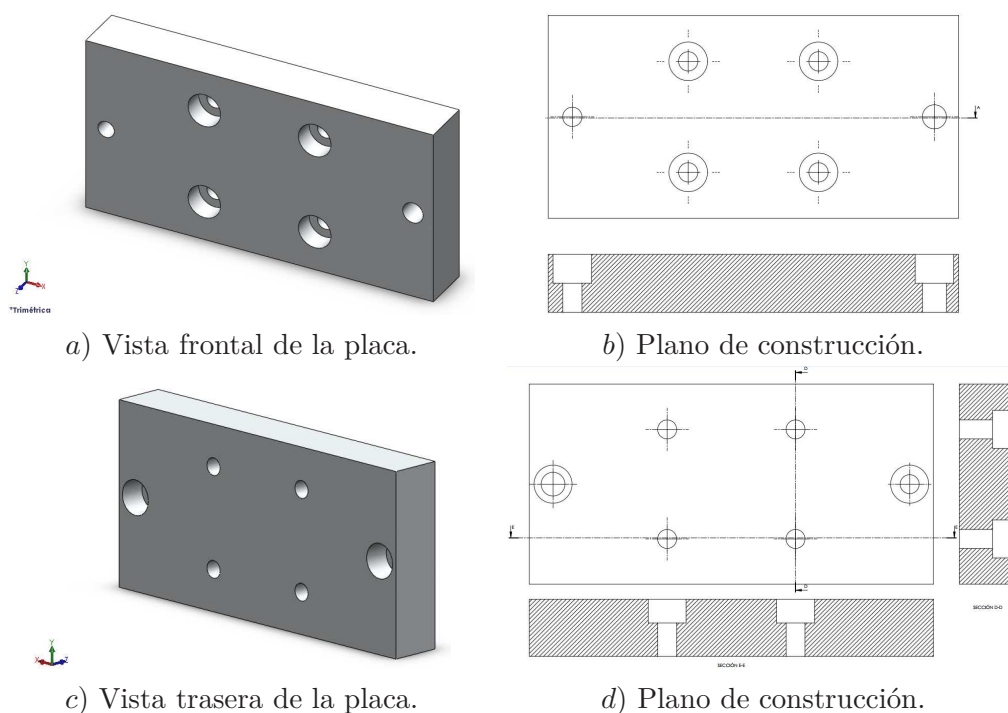


Figura 4.16: Segunda placa en diseño que será ensamblada con la cámara.

La integración completa del cabezal de visión con el manipulador quedaría así como se muestra en las imágenes de la figura 4.18.

En la imagen de la figura 4.18 en *a)* se muestra las dos placas antes de la integración, mientras que *b)*, *c)* y *d)* se tiene vistas del ensamble del cabezal de visión y el manipulador. Los orificios que se encuentran ubicados en las laterales de la placa son usados para unirlos y hacer fijo el sistema, mediante un par de tornillos se realiza la sujeción y de forma rápida. Se propone la fabricación de la placa que irá sujeta a la herramienta en varias unidades para que esto permita hacer los cambios de forma rápida y fácil.

Finalmente en el control del encendido de los láseres se usa una de las funciones que

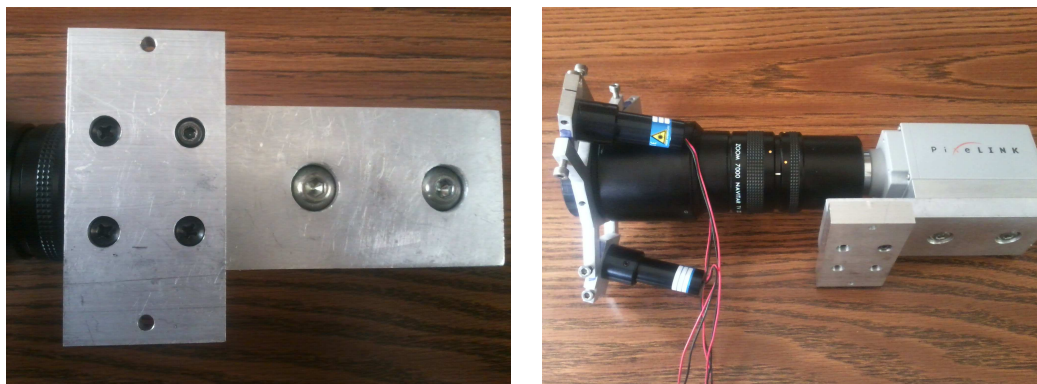


Figura 4.17: Placa en ensamble con cámara en dos vistas.

contiene el manipulador, ésta permite realizar la activación (on/off) de una señal. Los láseres dependerían de esta señal y se logra realizar por medio de programación, y el circuito que sería utilizado se muestra en diseño virtual en la imagen 4.19.

La tarjeta impresa final es mostrada en la imagen de la figura 4.20:

4.5. Pruebas con imágenes de cámara

Las pruebas que se realizan en esta sección son con imágenes reales adquiridas por cámara ya integrada en el manipulador. Para la calibración que se realiza de cámara y del cabezal se hace la adquisición de dos imágenes.

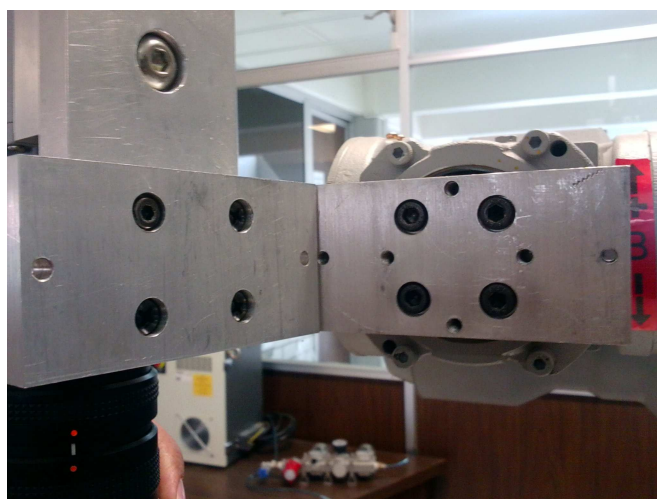
La toma de fotografía se hace como se muestra en las imágenes presentadas en la figura 4.21 y en *a*) se realiza la toma de la imagen del tablero de ajedrez, mientras que en *b*) se hace la toma de los láseres sobre la superficie para la calibración que se realizará seguido de éste paso.

4.5.1. Calibración

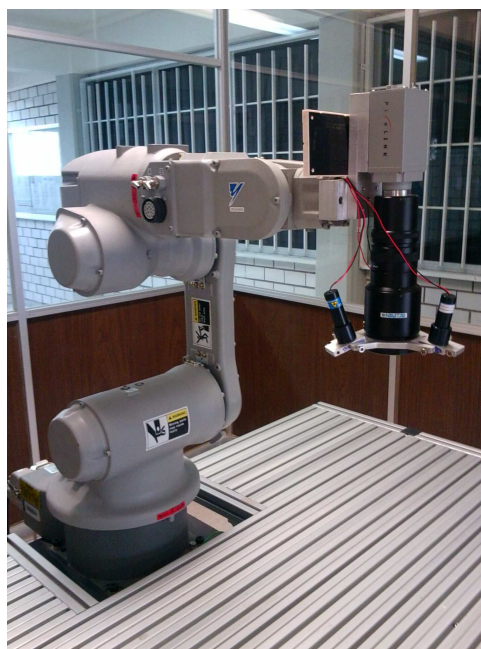
Habría que recordar que en la calibración se hace la toma de dos fotografías en alturas diferentes, las cuales realizarán con el cabezal de visión y una vez que son adquiridas éstas se proporcionan al algoritmo de optimización y se realiza el proceso de calibración. Entonces se muestra en la figura 4.22 un par de imágenes que son empleadas para éste proceso.

En la imagen de la figura 4.22 *a*) se realiza la toma de la imagen del tablero de ajedrez a una altura, mientras que en *b*) se hace la toma de la segunda imagen para realizar la calibración.

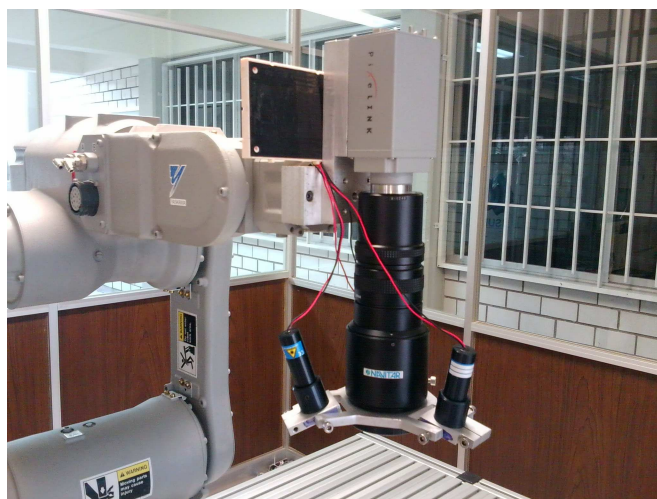
Para el proceso de calibración se realiza una doble búsqueda sobre las imágenes, en la primer etapa se usan pocos puntos por campo para que el cálculo computacional sea bajo y



a)



b)



c)



d)

Figura 4.18: Cabecal de visión artificial montado en robot serial.

se logre llegar a un primer acercamiento a la descripción de la imagen. En la segunda etapa se usan mayor número de puntos para representar el campo de ajedrez con mejor calidad, y con ello permite hacer la comparación con mayor fiabilidad, pero que lógicamente aumenta

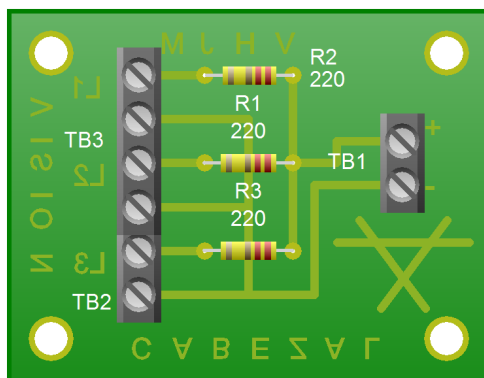


Figura 4.19: Diseño de la tarjeta en PCB

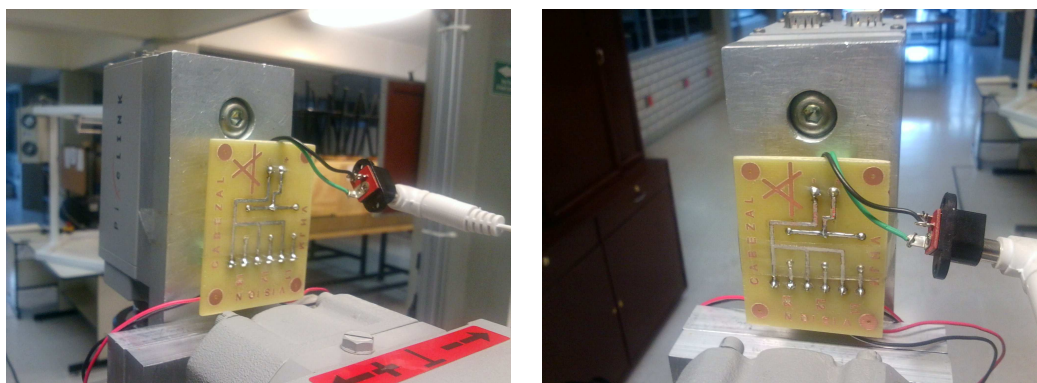
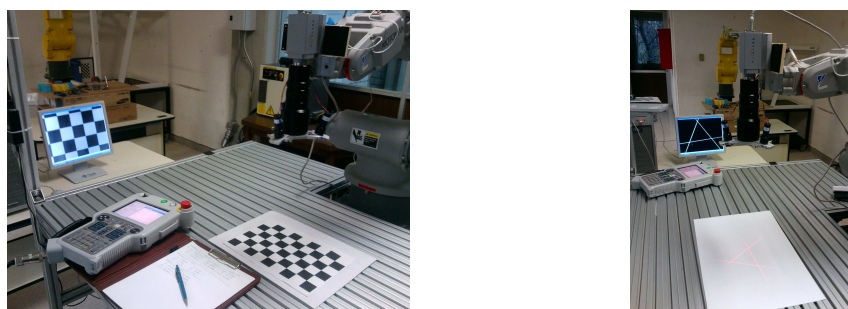


Figura 4.20: Tarjeta construida y ensamblada con la cámara.



a) Adquisición de imagen del ajedrez b) Adquisición de imagen de láser de línea

Figura 4.21: Proceso de adquisición de imágenes.

el costo computación del proceso. Éstas condiciones se logran combinar con la tolerancia de búsqueda de los parámetros de cámara y puede demorar el proceso para la determinación de la información.

A continuación se muestra la imagen del ajedrez con menor número de puntos al ser

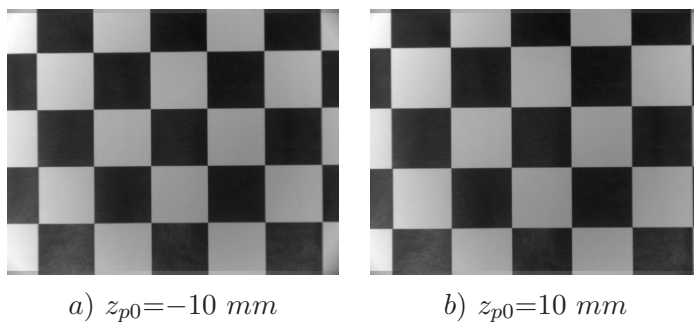


Figura 4.22: Imágenes reales para calibrar.

comparada con la imagen real.

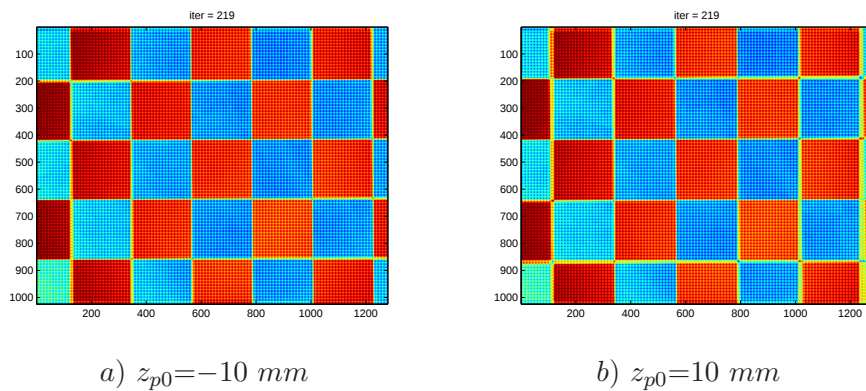


Figura 4.23: Calibración de cámara con imágenes reales, primera etapa

Ahora se muestran la imagen en la segunda etapa, usando mayor número de puntos por campo.

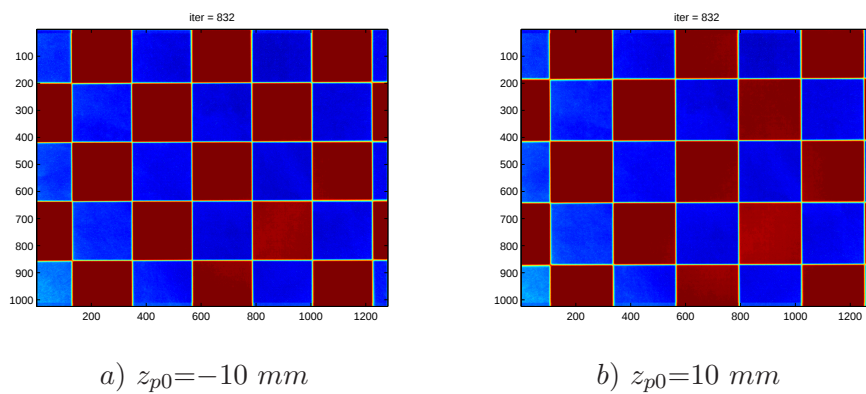


Figura 4.24: Calibración de cámara con imágenes reales, segunda etapa.

Los valores que determinó el algoritmo para éste caso en particular es mostrado en la tabla 4.4:

Parámetro	Valor
$u0$	393.0140
$v0$	392.2521
Z_{cam}	490.7680
K_u	3702.1000
K_v	3704.5000

Cuadro 4.4: Parámetros de cámara calibrado para éste caso en específico.

En la calibración de los láseres se usaron un par de imágenes como sucedió con la calibración de cámara y las imágenes del ajedrez, en la figura 4.25 son mostradas las imágenes reales adquiridas por la cámara. Como se logra observar en *a*) los láseres se muestran bien definidos y en *b*) no lo son tanto.

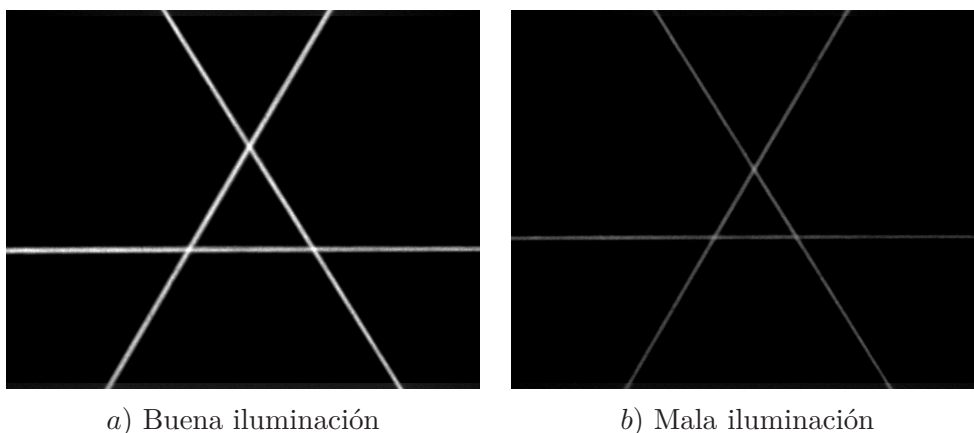


Figura 4.25: Imágenes reales de los láseres de línea y mala uliminación.

Las irregularidades en la intensidad luminosa de los láseres son modificadas por un auto ajuste de sensibilidad o amplificación que tiene la cámara. Cuando la cámara detecta demasiada luminosidad por un cierto tiempo ésta tiende a atenuar la luminiscencia sobre la imagen y es por ello que habrá casos esporádicos sobre las imágenes. Para evitar estos efectos tendría que darse un tiempo para que la imagen se estabilice y después adquirir la fotografía para fines prácticos, pero como es de hacerse notar que el algoritmo generado para este trabajo contempla ser lo suficientemente robusto que permitiría absorber condiciones de éste tipo. Para ello se realizará la calibración del los láseres con imágenes en los cuales se encuentren sometidos a condiciones en las cuales carezca de mayor intensidad los haz de línea. Ver sección 4.6 en pag. (111).

Por el momento se utilizará imágenes reales donde se encuentren en condiciones normales los haz de luz y se definan de manera correcta para mostrar el proceso de calibración de los láseres.

Las imágenes reales de los láseres que son usados para la calibración del sistema se muestran en la figura 4.26:

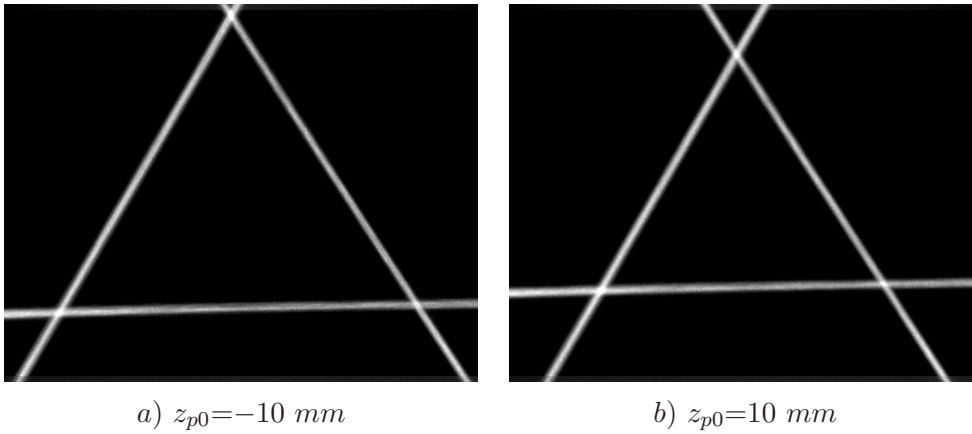


Figura 4.26: Calibración de los láseres con imágenes reales y buena iluminación.

En el proceso de calibración se suscita como en las imágenes virtuales, y se aplican las dos metodologías de búsqueda que se propusieron en el proceso virtual. El primero, consiste en usar una imagen para ubicar la orientación de los láseres y después dos imágenes para encontrar el ángulo de proyección y los radios en los cuales se encuentran los haz de luz. El segundo método usa dos imágenes en tres etapas de búsqueda (uno con valores muy grandes, medios y finos) respectivamente.

Como los parámetros que se buscan de las imágenes se señalan en la figura mostrada en 4.27, donde los variables denotados en primas (') y en línea seccionada son los que realmente se calcularían esperando que sean lo más próximo a los denotados con las variables sin apóstrofe y línea continua, que serían los ideales o semejantes a los que se usan en la calibración virtual.

Recordando que r_o se refiere a la distancia que hay desde el eje focal de cámara al láser, φ es el ángulo de separación que hay de un láser a otro y ψ es el ángulo con el que se proyecta el haz de línea a la superficie.

Las imágenes que se pudieran mostrar en ésta sección son equivalentes a las que usan en los métodos de búsqueda sobre las imágenes, por lo que no aportan nada nuevo a ésta parte.

Pero habrá que mencionar que en las propuestas de calibración de los láseres se describían tres métodos en los cuales se les daban diferentes pesos a las variables de búsqueda y por ello se podría tener tres calibraciones.

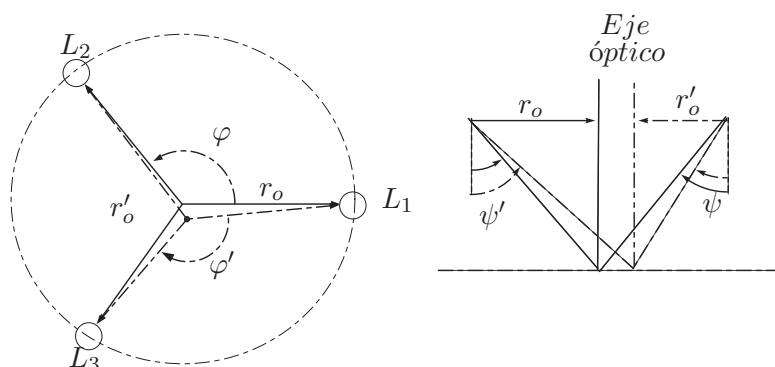


Figura 4.27: Determinación de los parámetros de calibración de los láseres.

Calibración 1			
	L_1	L_2	L_3
φ	-58.5541°	59.1732°	180.3460°
ψ	13.1676°	12.2767°	12.8080°
r_l	55.0013	55.0010	55.0008

Calibración 2			
	L_1	L_2	L_3
φ	-58.5544°	59.1733°	180.3472°
ψ	13.1673°	12.2764°	12.8080°
r_l	55.0000	54.9998	54.9999

Calibración 3			
	L_1	L_2	L_3
φ	-58.5551°	59.1744°	180.3438°
ψ	13.7015°	12.8947°	13.4067°
r_l	57.9231	58.5043	58.2202

Cuadro 4.5: Calibraciones de los tres métodos que se propusieron.

Finalmente se tiene tres calibraciones que fueron planteadas cada una en forma diferente, por así decirlo, en metodología. En las pruebas subsecuentes se podrá poner en evidencia la efectividad de los valores que se obtuvieron de cada método y poder evaluar la exactitud de que puede arrojar cada calibración en un proceso.

4.5.2. Procesamiento de una secuencia de imágenes

Una vez que se ha calibrado el cabezal de visión artificial, se da pie a realizar una sucesión de movimientos con el robot serial para registrar una secuencia de imágenes por medio del sistema de visión. Una vez que se halla realizado la adquisición de la secuencia, de cada imagen se determinará la información sobre ella y se hará un registro de la misma para compararla con la secuencia programada y obtener una relación entre el movimiento realizado por el manipulador y el desplazamiento detectado por el cabezal de visión.

Se proponen tres pruebas de movimiento que serán usadas para validar el sistema de visión que se ha implementado.

1. Realizar un movimiento sobre el eje z_{p0} , es decir, a partir del punto de referencia que se establece como un cero, desplazar el manipulador de manera ascendente y descendente con movimientos en milímetro hasta llegar nuevamente al punto de referencia. Después hacer el movimiento de manera descendente y ascendente hasta llegar nuevamente al punto de referencia.
2. Realizar un movimiento en la pendiente de Δz_1 , con variación de medio grado en sus movimientos y de manera similar al anterior.
3. Realizar un movimiento en la pendiente Δz_2 , con variación de medio grado en sus movimientos de tal forma que se repita como el movimiento anterior.

4.5.2.1. Desplazamiento en Z_{p0}

Las pruebas que se presentan a continuación están enfocadas a realizar variaciones en las alturas con el robot industrial. Fijando el cabezal de visión en el último eslabón de manipulador de tal manera en que se tiene el sensor de cámara y la superficie en la cual se proyectan los láseres se encuentran de forma paralela. O dicho de otro modo, el eje focal de cámara se encuentra perpendicular a la superficie en la que se proyectan los láser de línea.

Por cada desplazamiento que realizó el manipulador se fue adquiriendo una imagen por medio del cabezal de visión. Una vez que se completó la rutina se continuó con el procesamiento de las imágenes por medio del algoritmo para determinar la información sobre las mismas y finalmente se realizaron gráficas de los movimientos. Se busca comparar el movimiento que en teoría debía de realizar el robot y el movimiento que se registró en las imágenes debido al desplazamiento del robot y la deformación de los láseres sobre la superficie.

Primero se analizan las calibraciones obtenidas de los métodos propuestos, para ello se programa un movimiento como los descritos anteriormente comenzando en una posición 0 hasta una posición de -21 mm y después hasta una posición de 21 mm y regresar nuevamente a su posición 0, siendo los incrementos de 3 mm por cada desplazamiento.

En la gráfica mostrada en la figura 4.28 se puede distinguir tres líneas por cada calibración. Una línea en rojo que en la leyenda se establece como el movimiento programado

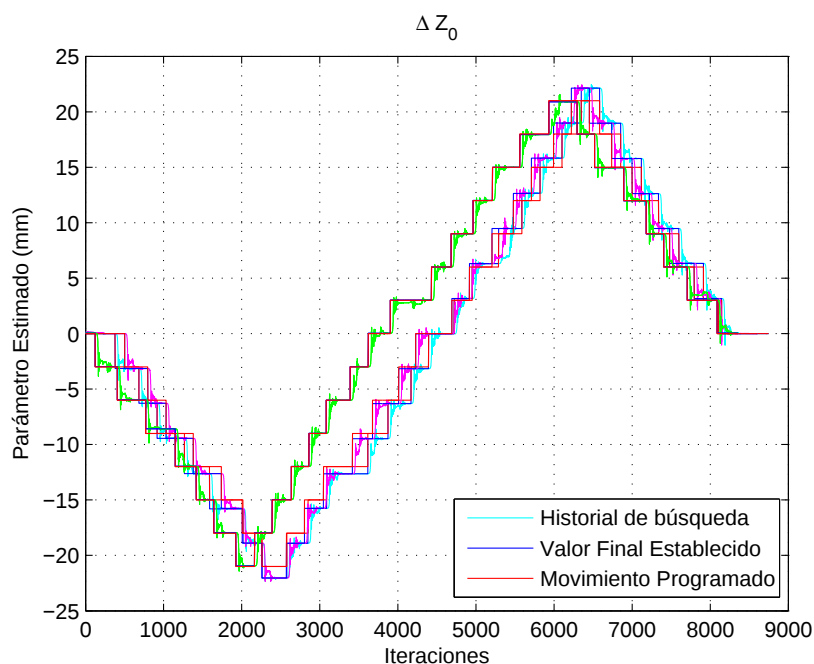


Figura 4.28: Secuencia de imágenes y las tres calibraciones.

al manipulador, las líneas verde, magenta(violeta) y celeste representan el historial de búsqueda sobre las imágenes y el color azul marino representa el valor en el cual se estableció la búsqueda en cada imagen. Para el primer método que se estableció le corresponde el color *celeste*, en un segundo método que se propuso se usa el color *magenta* y para un tercer método le corresponde el color *verde*.

También algo que se puede visualizar es que en los límites tanto superior como inferior se puede apreciar un error en los movimientos programados y los detectados en las imágenes.

Para poder revisar a detalle éste error se realiza un acercamiento de la cresta positiva en la figura 4.29, donde es posible identificar los errores con mayor visibilidad.

Como era evidente, tanto en la primera y segunda calibración los valores que se muestran en la tabla 4.5 son muy semejantes y en las gráficas mostradas en la figura 4.29 se logra ver que el valor en cual se fijan ambas calibraciones respecto al valor programado para el manipulador son iguales, mientras que la tercera calibración que se propuso ofrece valores muy cercanos a los programados.

Como se ha identificado que la calibración número tres ofrece mejor resultados, se hace una gráfica única en la figura 4.30 de la secuencia programada y se buscan más detalles en la misma.

Ésta gráfica permite observar un detalle sobre el historial de la evolución en la determinación de parámetros. La discrepancia mostrada será usada en la determinación del error.

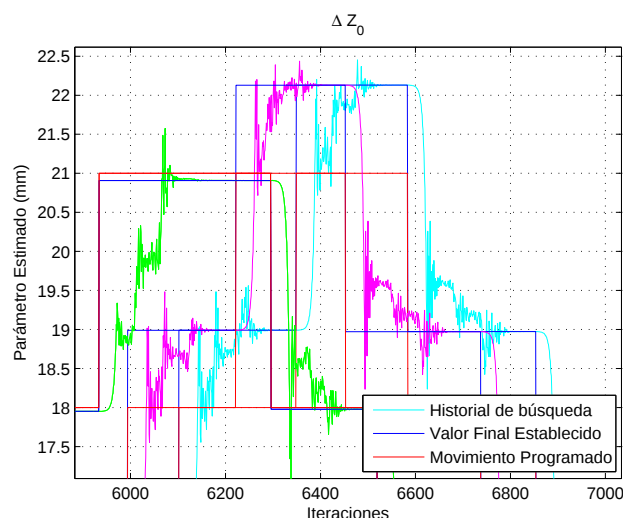


Figura 4.29: Secuencia de imágenes a detalle.

Aunque en la evolución mostrada en la gráfica se puede apreciar una buena aproximación a los valores reales o programados y el error puede ser o no tomado en cuenta ya que solo se presenta para una sola imagen. Esto podría ser difícil de explicar si solo se presenta en una imagen en la cual tiene todas las condiciones buenas para ser procesada. Puede ser a un error del mismo método en el peor de los casos.

Para calcular el error relativo se utiliza:

$$\varrho_{relativo} = \frac{\varrho_{absoluto}}{\varrho_{exacto}} \times 100\% \quad (4.2)$$

Método	Error (mm)	Porcentaje
Calibración 1	1.13	5.380 %
Calibración 2	1.13	5.380 %
Calibración 3	0.409	4.544 %

Cuadro 4.6: Porcentaje de error por calibración de imágenes.

Como se logra apreciar en los valores de la tabla 4.6, en los dos proceso de calibración en los cuales se les da diferentes pesos de búsqueda a los parámetros pueden ofrecer un mayor error a comparación a los que se les da un poco de libertad de búsqueda.

Para el desarrollo de las pruebas previstas en ésta sección se escoge la calibración que ofrece mejores resultados. Esto permitirá que se logren alcanzar parte de los objetivos que se plantearon al inicio de éste trabajo y si no, por lo menos llegar a un exactitud muy elevada.

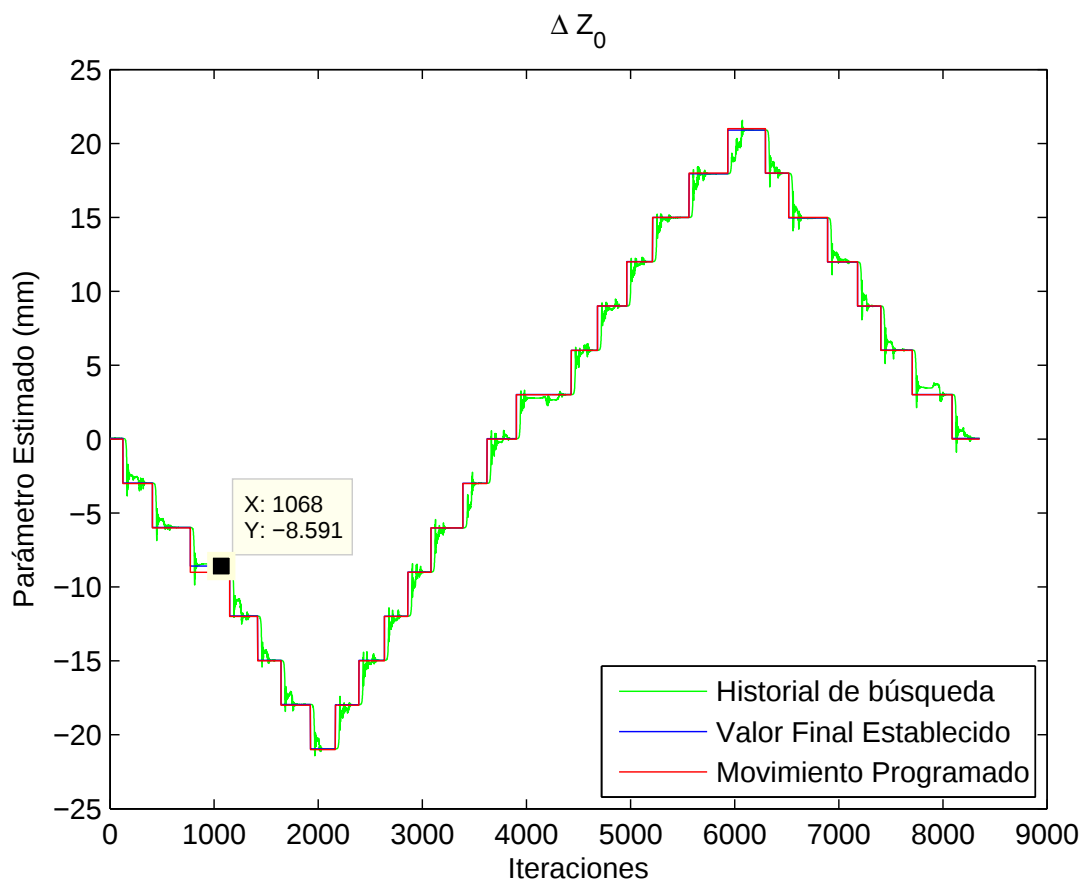


Figura 4.30: Calibración usada para el procesamiento de las imágenes.

Para revisar la susceptibilidad del algoritmo ante pequeñas variaciones de desplazamientos se diseña una prueba de movimiento, como las que se describían al principio de éste apartado, con variaciones en milímetros y con la misma trayectoria. El límite inferior a partir de la referencia es de -20 mm hasta un límite superior de 20 mm . En esta rutina se espera una variación en los cambios de altura del plano z_{p0} y que el alabeo del plano o la pendiente en Δz_1 y Δz_2 se mantenga constante o dicho de otra forma $\Delta z_1 = 0$ y $\Delta z_2 = 0$.

La gráfica de la figura 4.31 muestra el registro del movimiento del manipulador en la cambios de altura para z_{p0} .

En 4.32 se muestra el desplazamiento que se registró en la pendiente Δz_1 a causa del movimiento en z_{p0} .

En la figura 4.33 se muestra el desplazamiento que se registró en la pendiente Δz_2 a causa del movimiento en z_{p0} .

Realizando un acercamiento a las gráficas mostradas en las figuras 4.31, 4.32 y 4.33, se

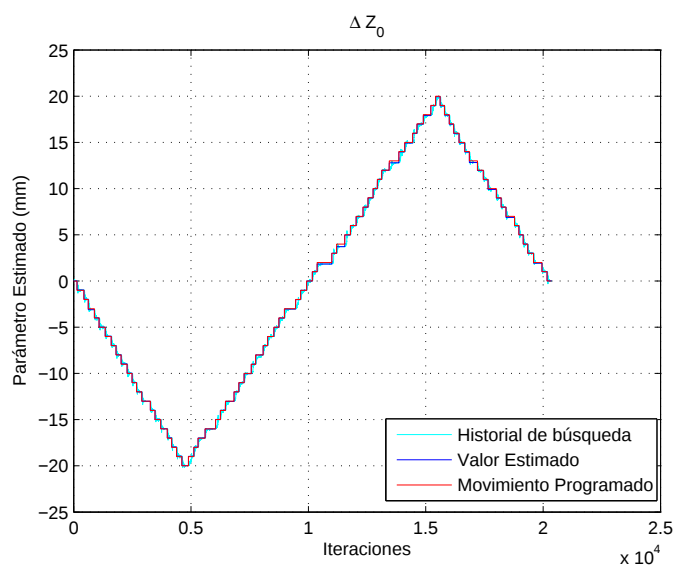


Figura 4.31: Registro del movimiento del robot en cambio de altura z_0 .

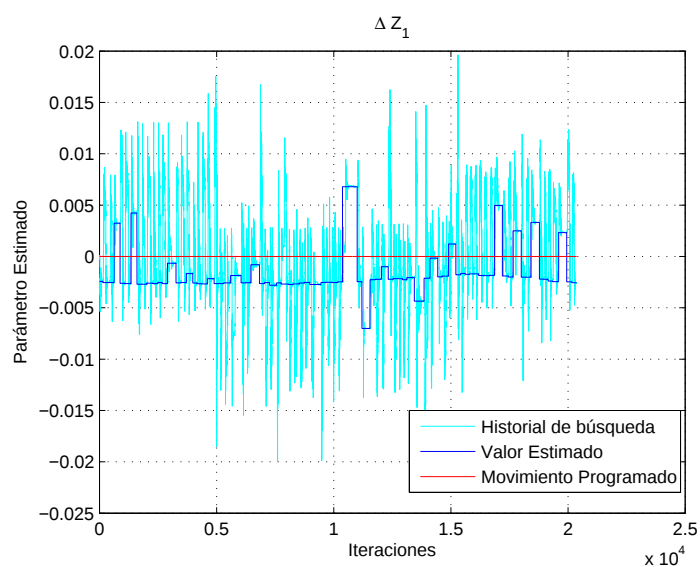


Figura 4.32: Registro del movimiento del robot en cambio de pendiente (Δz_1).

podría tomar algún punto cualquiera en la gráfica con el propósito de ver el movimiento detectado en las imágenes por el algoritmo de optimización y el programado para el robot.

En las tablas 4.7 y 4.8 se muestra el cálculo del error absoluto entre el valor calculado y el valor programado en el movimiento del robot. En la tabla 4.7 se presenta el error para el movimiento descendente que se ejecuta en la trayectoria.

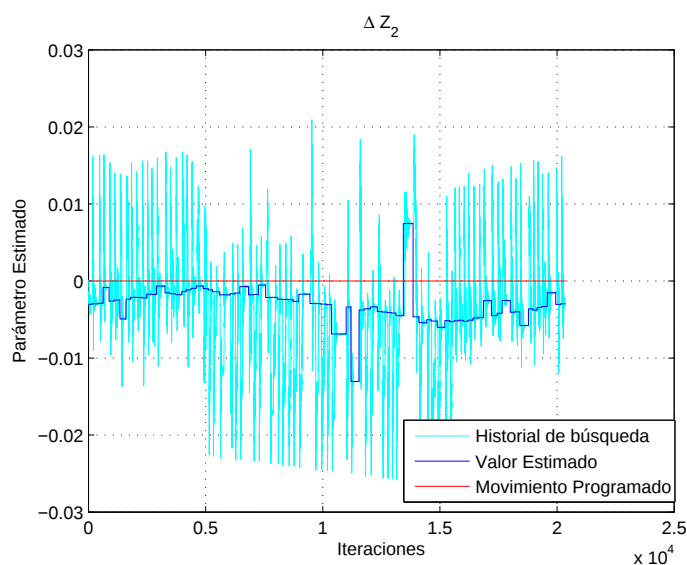


Figura 4.33: Registro del movimiento del robot en cambio de pendiente (Δz_2).

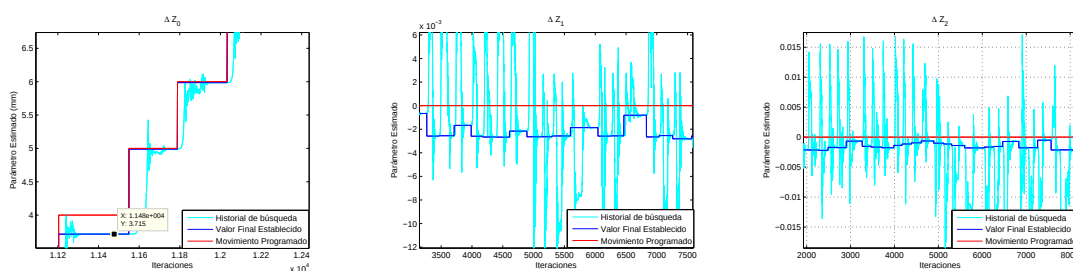


Figura 4.34: Acercamiento a las gráficas.

En la tabla 4.8 se calcula el error en el movimiento ascendente que realiza el robot manipulador respecto a la referencia.

En la imagen 45 se logra obtener el error más significativo de toda la rutina generada con el manipulador. El valor que corresponde en ésta posición es de 0,2846, mientras que el error mínimo se da en la imagen 44 y su error es de $8,7000e - 004$.

Pero lo que es importante es ver que en las pendientes Δz_1 y Δz_2 hay un valor distinto al 0 que debería de haber por sistema. Los registros en los movimientos para Δz_1 están dados a niveles de micrómetros y por tanto se pueden considerar como valor de 0. Mientras que para Δz_2 se alcanza apreciar un movimiento ondulatorio muy pequeño, el cual se debe a que el punto de trabajo del robot de la herramienta (TCP) se encuentra fuera del punto de trabajo.

Ima.	1	2	3	4	5	6	7	8
V. Real	0	-1	-2	-3	-4	-5	-6	-7
V. Calculado	0.01664	-0.97648	-1.98252	-3.09228	-3.99234	-4.9962	-5.97315	-6.97834
Error	0.01664	0.02352	0.01748	-0.09228	0.00766	0.0038	0.02685	0.02166
Ima.	9	10	11	12	13	14	15	16
V. Real	-8	-9	-10	-11	-12	-13	-14	-15
V. Calculado	-7.98353	-8.97604	-9.97385	-10.98715	-11.9861	-13.02129	-13.98678	-14.97888
Error	0.01647	0.02396	0.02615	0.01285	0.0139	-0.02129	0.01322	0.02112
Ima.	17	18	19	20	21	22	23	24
V. Real	-16	-17	-18	-19	-20	-19	-18	-17
V. Calculado	-15.98403	-16.96745	-17.96086	-18.95318	-19.97188	-18.96803	-17.97855	-16.98484
Error	0.01597	0.03255	0.03914	0.04682	0.02812	0.03197	0.02145	0.01516
Ima.	25	26	27	28	29	30	31	32
V. Real	-16	-15	-14	-13	-12	-11	-10	-9
V. Calculado	-15.99545	-14.98388	-13.99557	-13.03176	-11.99529	-11.00524	-10.02321	-9.00347
Error	0.00455	0.01612	0.00443	-0.03176	0.00471	-0.00524	-0.02321	-0.00347
Ima.	33	34	35	36	37	38	39	40
V. Real	-8	-7	-6	-5	-4	-3	-2	-1
V. Calculado	-8.00565	-6.99263	-5.99713	-5.01472	-4.01515	-3.03659	-2.00553	-1.0033
Error	-0.00565	0.00737	0.00287	-0.01472	-0.01515	-0.03659	-0.00553	-0.0033

Cuadro 4.7: Tabla de error absoluto(*valor calculado* – *valor programado*). Movimiento descendente.

4.5.2.2. Desplazamiento en ΔZ_1

La segunda prueba se estableció para variaciones de $0,5^\circ$ en ΔZ_1 de forma ascendente y descendente. El parámetro de Z_0 no debería de mostrar variaciones en altura y la pendiente ΔZ_2 debería quedar en forma estática o una referencia en 0.

La primer gráfica que se muestra en la figura 4.35 es para el desplazamiento en z_{p0} .

Como se logra observar en la figura 4.35, hay un ligero movimiento a partir del punto de referencia, el cual está dado en un rango de $\pm 0,4$ mm aproximadamente. Éste desplazamiento sigue siendo a causa de que el TCP del robot sigue moviéndose en el momento en que se realizan las rutinas. A su vez puede dar pie a que los valores que se determinan para las pendientes sean distintos a los que teóricamente deberían de ser.

El movimiento registrado en la pendiente Δz_1 se puede observar en la figura 4.36.

Como se logra observar en la gráfica de la figura 4.36, el movimiento que se realiza en la pendiente Δz_1 , valores reportados en radianes, se dibuja el descenso y ascenso que se planteó. Aunque los valores no se logran comparar a simple vista en la gráfica, el máximo valor estimado está en $0,2$ radianes que es aproximadamente 12° y el valor mínimo en el descenso es de $-0,21$ radianes equivalente a $12,5^\circ$, sacando la equivalencia de $\pm 15^\circ$ es igual a $0,2617$ radianes. Esto quiere decir que hay una perdida de $2,5^\circ$ en los movimientos

Ima.	41	42	43	44	45	46	47	48
V. Real	0	1	2	3	4	5	6	7
V. Calculado	0.00534	1.00688	1.8409	2.99913	3.7154	4.9899	5.98594	6.95571
Error	0.00534	0.00688	-0.1591	-0.00087	-0.2846	-0.0101	-0.01406	-0.04429
Ima.	49	50	51	52	53	54	55	56
V. Real	8	9	10	11	12	13	14	15
V. Calculado	7.98071	8.98543	9.98871	10.99462	11.97776	12.79255	13.9733	14.93486
Error	-0.01929	-0.01457	-0.01129	-0.00538	-0.02224	-0.20745	-0.0267	-0.06514
Ima.	57	58	59	60	61	62	63	64
V. Real	16	17	18	19	20	19	18	17
V. Calculado	15.96782	16.97981	17.90308	18.96947	19.95132	18.95624	17.95371	16.95964
Error	-0.03218	-0.02019	-0.09692	-0.03053	-0.04868	-0.04376	-0.04629	-0.04036
Ima.	65	66	67	68	69	70	71	72
V. Real	16	15	14	13	12	11	10	9
V. Calculado	15.94897	14.94931	13.95207	12.81962	11.95248	10.96615	9.90368	8.97487
Error	-0.05103	-0.05069	-0.04793	-0.18038	-0.04752	-0.03385	-0.09632	-0.02513
Ima.	73	74	75	76	77	78	79	80
V. Real	8	7	6	5	4	3	2	1
V. Calculado	7.98079	6.88057	5.98903	4.99122	3.99614	3.00181	1.93165	1.02167
Error	-0.01921	-0.11943	-0.01097	-0.00878	-0.00386	0.00181	-0.06835	0.02167

Cuadro 4.8: Tabla de error absoluto(*valor calculado – valor programado*). Movimiento ascendente.

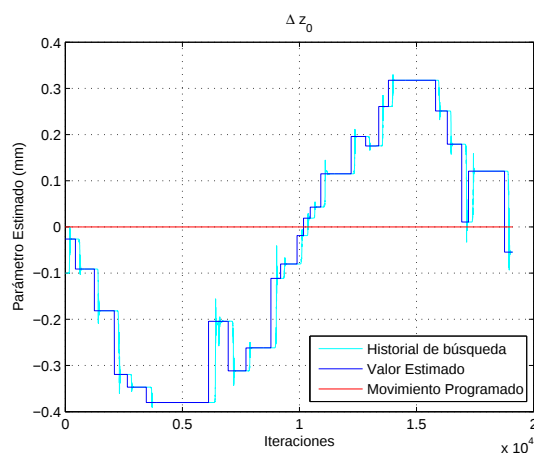


Figura 4.35: Movimiento del robot en z_{p0} en un desplazamiento para la pendiente Δz_1 .

detectados por el algoritmo.

El movimiento registrado en la pendiente Δz_2 se puede observar en la figura 4.37:

En ésta gráfica 4.37 se logra observar algunas protuberancias que se van detectando conforme se realiza el procesamiento de las imágenes, y como se logra apreciar en algún

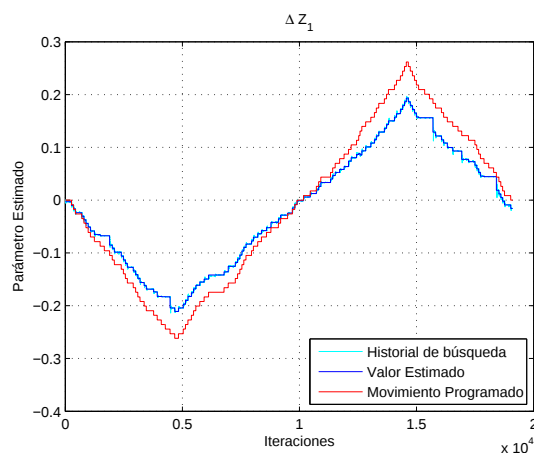


Figura 4.36: Registro de movimiento en Δz_1 en desplazamiento para la pendiente Δz_1 .

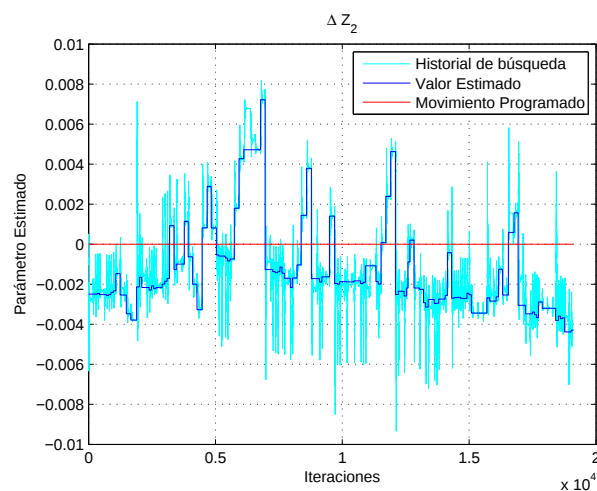


Figura 4.37: Registro del movimiento del robot Δz_2 en desplazamiento para la pendiente Δz_2 .

caso se detecta un pico que alcanza a llegar hasta 0,008 radianes, lo que equivaldría a $0,5^\circ$. Si se observa nuevamente los movimientos en los cambios de altura z_{p0} y los de Δz_2 podría ser una buena causa por la cual no se logre detectar con mejoría el valor en la secuencia para Δz_1 .

4.5.2.3. Desplazamiento en Δz_2

El último movimiento que se propuso fue para la pendiente en Δz_2 , que es igual que Δz_1 , que se espera a que quede constante z_{p0} al igual que Δz_1 y quien hace la variación es la otra pendiente.

La gráfica mostrada en la figura 4.38 es un tanto parecida a la que se revisó en Δz_1 ,

así que no se realiza comentario alguno.

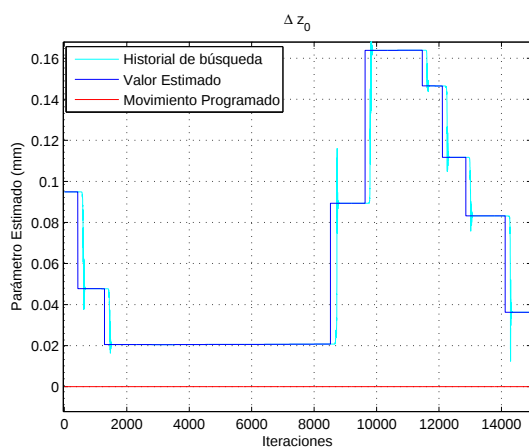


Figura 4.38: Movimiento del robot en z_{p0} en desplazamiento para la pendiente Δz_2 .

De igual forma el tipo de gráfica que se obtuvo en la figura 4.39 es en gran forma parecida al movimiento anterior.

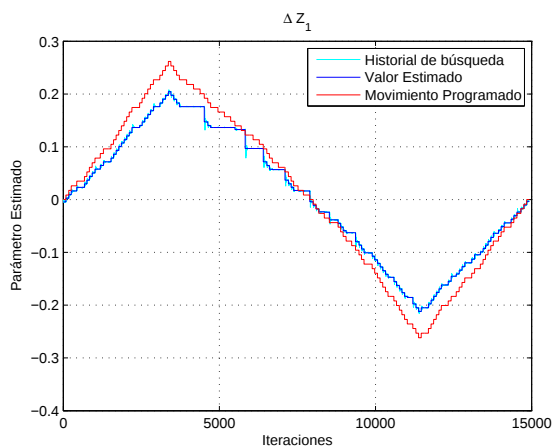


Figura 4.39: Movimiento del robot en Δz_1 en desplazamiento para la pendiente Δz_2 .

En la gráfica mostrada en la figura 4.40 se logra observar un desplazamiento mayor que en el anterior gráfico.

Para el valor de referencia que se tiene en la estimación del parámetro anda en un promedio de $-0,15$ radianes equivalente a $0,9^\circ$. Más a parte se logra apreciar un tipo de onda oscilatoria, la cual se debe a que el manipulador serial para realizar un movimiento en la orientación de Δz_1 es diferente al movimiento que debe de realizar en Δz_2 . Las

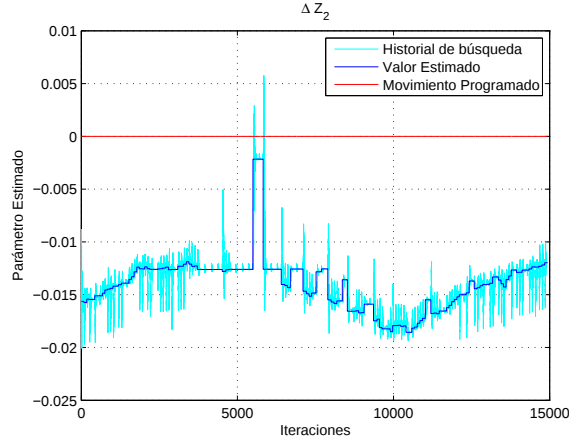


Figura 4.40: Movimiento del robot en Δz_2 en el desplazamiento para pendiente Δz_2 .

articulaciones de orientación del robot para Δz_1 toma las que tiene por default (180,-90,0) y mientras que para hacer un movimiento orientado en Δz_2 éste debe de cambiar el sentido de la orientación (-90,-90,-90).

4.6. Evaluación del método para imágenes con ruido

Este apartado esta dedicado a mostrar como el método en la visión artificial puede llegar a ser robusto. Para ello, se usan algunas imágenes que no muestran buenas condiciones de la luz del láser de línea. Las líneas son proyectados sobre el tablero de ajedrez con la finalidad de que, al ser tomada la imagen, la línea tengan partes donde se pueda marcar de buena forma (cuando inciden en cuadros blancos del tablero) y en otras en las cuales no se logre definir bien la línea (cuadros negros).

Las imágenes que son consideradas con ruido, son mostradas en las imágenes de la figura 4.41.

Las imágenes que son mostradas en la figura 4.42, son imágenes al azar de una secuencia que está siendo analizadas.

Como se logra observar en las imágenes, los láser se acoplan de buena forma sobre las secciones de línea que se generaron al proyectar los láseres sobre el tablero de ajedrez, básicamente basta con una porción pequeña de las líneas para que se logre la superposición de la láseres virtuales generados.

Por último en (4.43) se muestran las gráficas del proceso de búsqueda que se realizó de las imágenes con ruido.

Como se puede observar, en cada imagen se logra determinar la posición con una muy buena precisión, mientras que los cambios en las pendientes Δz_1 y Δz_2 muestran una

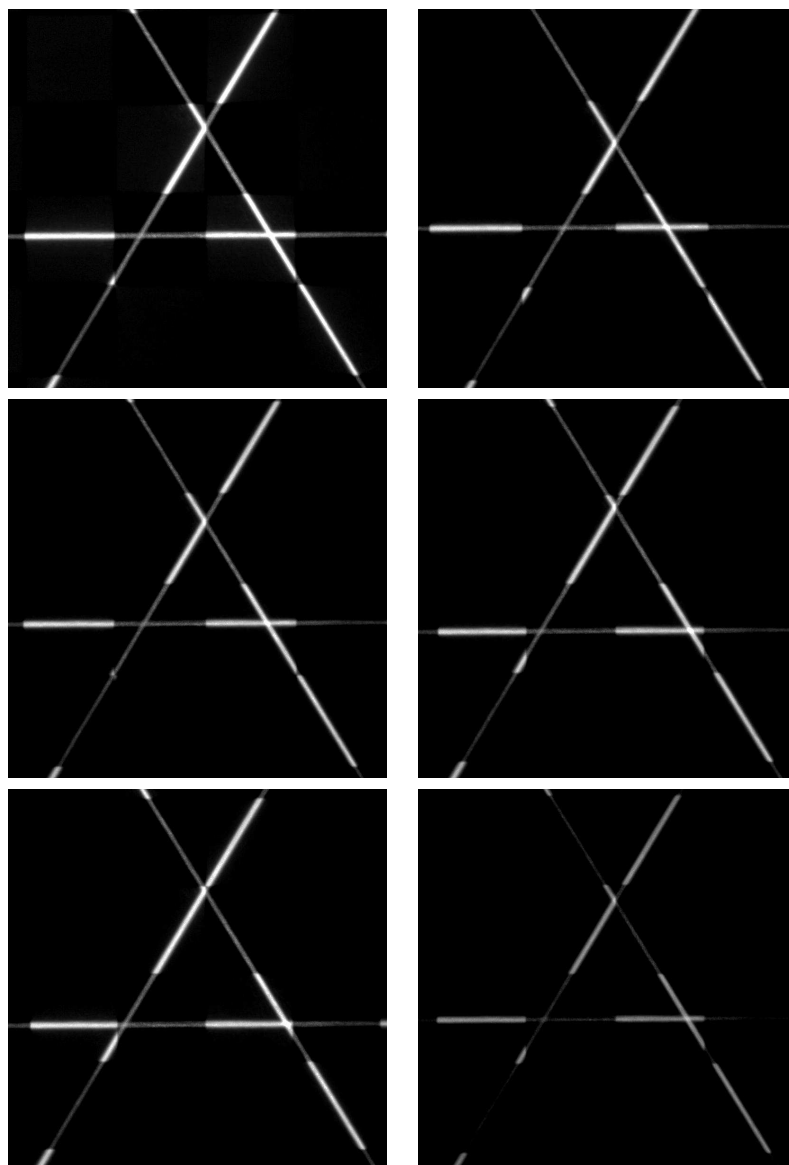


Figura 4.41: Secuencia de imágenes con ruido.

buena estabilidad, inclusive con mayor estabilidad que en la validación del método con una secuencia de fotos reales. Estos resultados se pueden deber a que se hicieron algunas modificaciones en los filtros que se aplicaron en la búsqueda, así como también en las tolerancias.

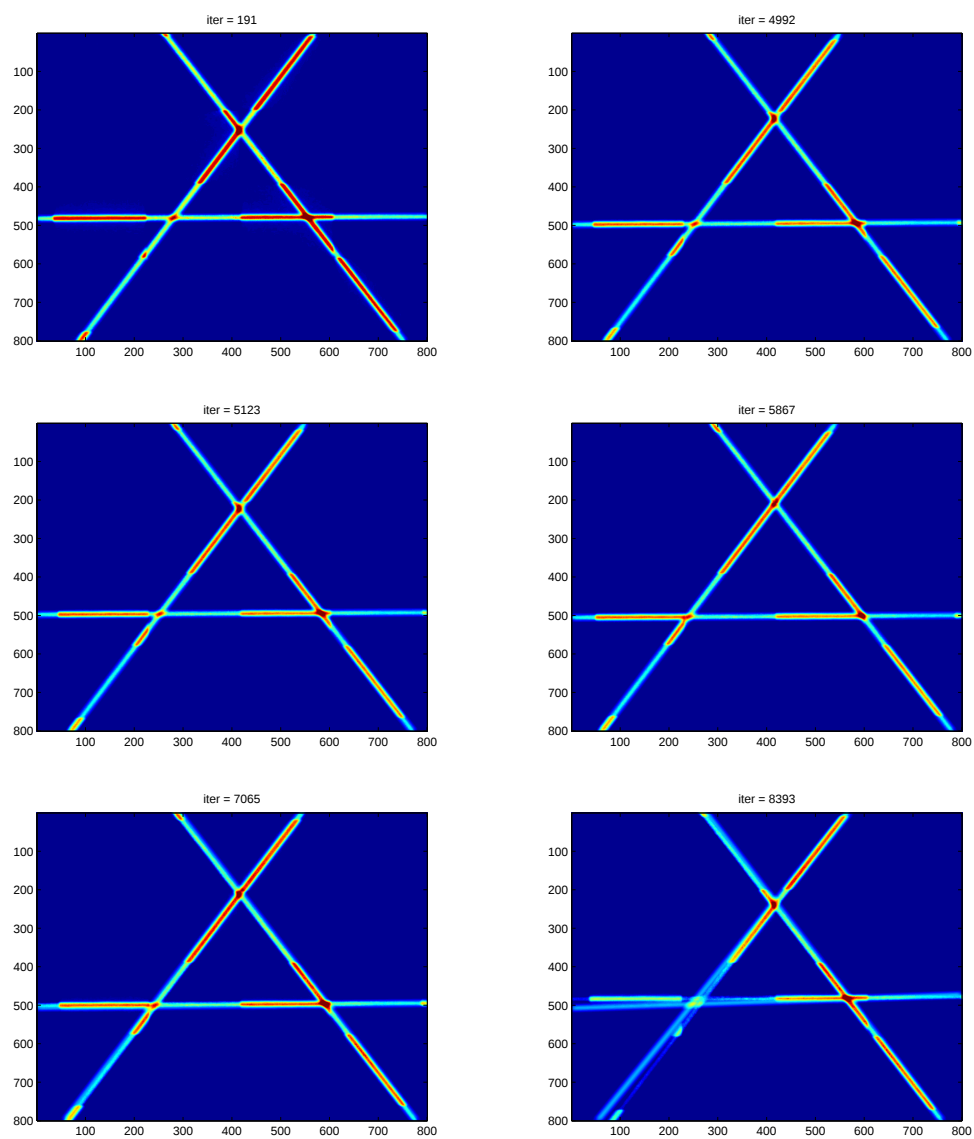


Figura 4.42: Imágenes al azar durante un análisis de imágenes con ruido.

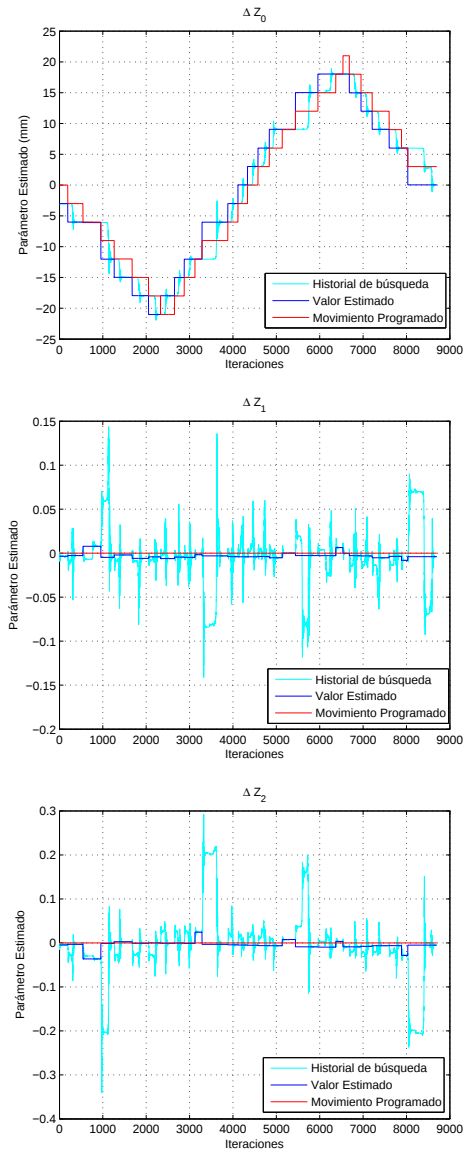


Figura 4.43: Determinación de los parámetros sobre las imágenes con ruido.

Conclusiones

En la generación de imágenes artificiales se implementaron dos métodos distintos para ensamblar una imagen. Método Gaussiano y método de “4Pixel”. La elaboración de las pruebas mostraron que el método con mayor velocidad fue el segundo.

El método de “4Pixel” tenía contemplados dos filtros que eran aplicados a los puntos para generar la imagen. Entre ellos está una máscara gaussiana y una cónica. La precisión con la que se puede ubicar el centro de una línea era mucho mayor con la máscara cónica, debido a su gradiente de luminosidad que es constante y una cresta puntal, mientras que la máscara Gaussiana su pendiente es muy suave y una cresta muy plana.

Generar imágenes con dimensión grandes tienen un costo computacional mayor que una de menor dimensión.

El número de puntos a ensamblar en una imagen y el tamaño de los filtros a usar se convierten en factores a considerar cuando se habla de 1000 iteraciones. Cuando el procesamiento tarda mucho tiempo se considera que ya no es crítico los tamaños. Cuando el tiempo es menor es notable considerar las dimensiones de los filtros. Ejemplo, prueba 1 y 21 de la tabla 3.4, pag. 57.

La pruebas con imágenes virtuales demostraron que se puede obtener con una resolución de información de $1 \times 10^{-6} \mu m$ de precisión sobre la imagen de estudio. Lo anterior conlleva a que, a mayor precisión del método, entonces mayor tiempo de procesamiento. A menor precisión entonces usa menos tiempo para determinar información sobre la imagen.

En una secuencia de imágenes, toma mayor tiempo en detectar la información sobre la primer imagen que en imágenes subsecuentes. Ya que parte desde la información estimada hasta encontrar la información correcta, mientras que para las siguientes imágenes ya les antecede información muy cercana a la correcta.

Las condiciones de luz sobre las imágenes no afectan en el método para detectar información sobre la imagen. La condición sobre la imagen es que debería de tener definiciones parciales o tenues de las líneas para que ésta pueda ser procesada. Mientras que en la metodología convencional, si no se tienen condiciones de luz adecuadas sobre la imagen, pueden haber perdida de información y no obtener buenos resultados.

El desarrollo de pruebas también demostró que la metodología para buscar y detectar información sobre la imagen puede ser altamente configurable. Y considerando que las condiciones de luz del ambiente que no afectan, le dan la capacidad al método de ser

robusto.

Trabajo futuro

Para mejorar el trabajo que se ha desarrollado al momento, se puede sugerir nuevas tareas que ayudarían a consolidar el trabajo, para ello se describen a continuación.

1. Hacer una comparación en términos de rapidez del método y la precisión con la que se determina información, entre el método desarrollado en este trabajo de tesis y la metodología de procesamiento de imágenes conocido.
2. Establecer una comunicación ethernet bi-direccional entre código de programación del análisis de imágenes (Matlab) y el lenguaje de control del robot manipulador (Lenguaje C).
3. Ampliar el caso de estudio de una superficie plana (en el plano simple) a un doble plano, para dar pie a realizar un análisis sobre la imagen en juntas de soldadura de tipo traslape.
4. Realizar un estudio e implementar un código de programación para tipos de superficies alternativas, como puede ser superficies curvas (reconocimiento de tubos) y superficies irregulares.
5. Implementar pruebas para el análisis y detección de información sobre la imagen en tiempo real, con la finalidad de realizar una corrección del movimiento del robot.
6. Diseñar una estrategia de control en tiempo real del manipulador. Con la información que se obtiene del análisis de las imágenes por el método de optimización se puede realizar el seguimiento de una ranura de soldadura.

Bibliografía

- BATURONE, A. O. (2001). Robótica Manipuladores y robots móviles . Barcelona (España).
- ENTZINGER, J. (2005). A flexible seam detection technique for robotic laser welding . <http://essay.utwente.nl/58056/>.
- GAO, S., ZHAO, M., ZHANG, L., et ZOU, Y. (2008). Dual-beam structured light vision system for 3D coordinates measurement . Dans *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on*, pages 3687 –3691.
- GARCÍA CEDILLO, G. F. (2011). Un Algoritmo Basado en Visión para la Detección y Manipulación de Piezas en una Celda de Manufactura . Stage de DEA, Universidad Autónoma de San Luis Potosí.
- HE-XI, L., YONG-HUA, S., GUO-RONG, W., et XIAO-XI, Z. (2009). Automatic Teaching of Welding Robot for 3-Dimensional Seam Based on Ant Colony Optimization Algorithm . Dans *Intelligent Computation Technology and Automation, 2009. ICICTA '09. Second International Conference on*, volume 3, pages 398 –402.
- HOYOS GUTIÉRREZ, J. G., CARDONA ARISTIZÁBAL, J. E., CAPACHO VALBUENA, L. M., et OROZCO, L. F. (2010). Técnicas de Calibración de Cámaras para Visión Estéreo y Reconstrucción . *XV SIMPOSIO DE TRATAMIENTO DE SEÑALES, IMÁGENES Y VISIÓN ARTIFICIAL, STSIVA 2010*, pages 76–81.
- IAKOVOU, D. (2009). *Sensor development and integration for robotized laser welding* . Thèse de Doctorat, University of Twente, Enschede. <http://doc.utwente.nl/60601/>.
- J. A. NELDER, R. M. (1965). A Simplex Method for Function Minimization . *The Computer Journal*, 7.
- JAE BYUNG, P., SEUNG HUN, L., et IL JAE, L. (2009). Precise 3D Lug Pose Detection Sensor for Automatic Robot Welding Using a Structured-Light Vision System. . *Sensors* (14248220), 9(9):7550 – 7565.
- LIPSCHUTZ, M. M. (1970). *Teoría y Problemas de Geometría Diferencial*, McGraw-Hill de México, S.A. de C.V. McGraw-Hill, 1971. Capítulo 2.

- ROMERO, A. E. (2002). Calibración de parámetros para cámaras CCD:Una guía Práctica . Dans *Revista del Centro de Investigación, Universidad LA SALLE*, volume 5, pages 41–50. <http://redalyc.uaemex.mx/pdf/342/34251905.pdf>.
- SÁNCHEZ MARTÍN N., Arias Pérez B., G. A. D. et J., G. L. (2004). Análisis aplicado de métodos de calibración de cámaras para usos fotogramétricos. . *VIII Congreso Nacional de Topografía y Cartografía, España.*, pages 1–12.
- TRUCCO, E. et VERRI, A. (1998). *Introductory to techniques for 3-D computer vision*. Prentice Hall.
- TSAI., R. (1987). A Versatile Camera Calibration Technique for Hogh-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses . *IEEE Journal of Robotics and Automation*, 4(Ra3):323–344.
- WENG, J., COHEN, P., et HERNIOU, M. (1992). Camera calibration with distortion models and accuracy evaluation . *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(10):965 –980.
- WIKIMEDIA (2012). Espectro Electromagnético . Internet. http://upload.wikimedia.org/wikipedia/commons/b/b6/Electromagnetic_spectrum-es.svg.
- WWW.FOTONOSTRA.COM (2013). Fotosito . Internet. <http://www.fotonostra.com/glosario/fotosito.htm>.
- WWW.TELEDYNEDALSA.COM (2012). Image Sensor Architectures for Digital Cinematography . Internet.
- XU, P. Q., TANG, X. H., LU, F. Q., et YAO, S. (2006). Welded seam 3D calculation and seam location for welding robot system. . *Science Technology of Welding Joining*, 11(3):352 – 357.
- ZHANG., A. (2010). A Flexible New Technique for Camera Calibration . <http://research.microsoft.com/~zhang>.

Apéndice A

Comunicación en matlab

A.1. Comunicación localhost

Para realizar la comunicación vía localhost de matlab, se deben de tener dos sesiones distintas de matlab en una misma computadora. En la primera sesión se puede tener escribir en un archivo (.m) el siguiente texto:

A.2. Comunicación TCP/IP

```
%% Archivo Servidor
data = membrane(1);
s = whos('data')

tcpipServer =tcpip('0.0.0.0',55000,'NetworkRole','Server');
set(tcpipServer,'OutputBufferSize',s.bytes);
fopen(tcpipServer);
fwrite(tcpipServer,data(:),'double');
fclose(tcpipServer);
```

Este archivo permite sugerir a esta sesión de Matlab como un *Servidor* y es quien va a proporcionar información principal a la otra sesión que será denominada como *Cliente*.

```
data = membrane(1);
s = whos('data')
```

Toma la función *membrana* predeterminado de Matlab y lo asigna a *data*, después *who* despliega información de la membrana que es un variable del tipo estructura.

```
tcpipServer =tcpip('0.0.0.0',55000,'NetworkRole','Server');
set(tcpipServer,'OutputBufferSize',s.bytes);
```

La función *tcpip* habilita el canal de datos y con la dirección *0.0.0.0* permite que cualquier equipo pueda interactuar con el *Servidor*. En caso contrario de que existiera una dirección IP el sistema otorga permiso para que solo esa dirección interactúe con el servidor. 55000 es el número del canal de comunicación, *NetworkRole* activa el soporte para sockets de servidor, utilizando dos valores, cliente o servidor. *set* establece el tamaño del buffer de salida de datos según el tamaño de la membrana.

```
fopen(tcpipServer);
fwrite(tcpipServer,data(:),'double');
fclose(tcpipServer);
```

fopen habilita el handle, mientras que *fwrite* escribe la transferencia de datos de un tipo *double* y finalmente *fclose* cierra el canal.

La segunda sesión de Matlab puede escribirse el siguiente código en un archivo (.m) para fungir como cliente:

```
tcpipClient = tcpip('148.224.55.53',55000,'NetworkRole','Client')
set(tcpipClient,'InputBufferSize',7688);
set(tcpipClient,'Timeout',30);

fopen(tcpipClient);
rawData = fread(tcpipClient,961,'double');
fclose(tcpipClient);
reshapedData = reshape(rawData,31,31);
surf(reshapedData);
```

La descripción es la siguiente:

```
tcpipClient = tcpip('148.224.55.53',55000,'NetworkRole','Client')
set(tcpipClient,'InputBufferSize',7688);
set(tcpipClient,'Timeout',30);
```

El *tcpip* ahora asigna una dirección de IP para la máquina con la cual se debe de establecer la conexión. El canal sigue siendo el mismo pero ahora el rol de la sesión se establece como *Cliente*. En el *InputBufferSize* se establece el tamaño del buffer de entrada (debe de ser conocido) y se establece un tiempo de espera *Timeout* para comunicación, si se sobrepasa entonces se cierra el canal.

```
fopen(tcpipClient);
rawData = fread(tcpipClient,961,'double');
fclose(tcpipClient);
```

En comparación a la descripción de servidor mientras que era escritura, en esta opción se realiza la lectura del archivo que recibe con *fread* y especifica el número de elementos de lectura y la clasificación de los datos.

```
reshapedData = reshape(rawData,31,31);  
surf(reshapedData);
```

En estas líneas solo es manipulación de datos y mostrar el tipo de dato enviado a través de la comunicación *TCPIP*.

