



Universidad Autónoma de Querétaro

Facultad de Ingeniería
Doctorado en Mecatrónica

**Metodología de calibración cinemática de robots manipuladores
por medio de visión artificial utilizando sensores TOF**

Tesis

Que como parte de los requisitos para obtener el grado de:

Doctor en Mecatrónica

Presenta:

Marco Antonio Garduño Ramón

Asesor:

Dr. Luis Alberto Morales Hernández

San Juan del Río, Querétaro, Octubre de 2018.



Universidad Autónoma de Querétaro
Facultad de Ingeniería
Doctorado en Mecatrónica

Metodología de calibración cinemática de robots manipuladores por medio de visión artificial utilizando sensores TOF

Tesis

Que como parte de los requisitos para obtener el grado de:

Doctor en Mecatrónica

Presenta:

Marco Antonio Garduño Ramón

Dirigido por:

Dr. Luis Alberto Morales Hernández

Dr. Luis Alberto Morales Hernández

Presidente

Dr. Roque Alfredo Osornio Rios

Secretario

Dr. Juan Primo Benítez Rangel

Vocal

Dr. Jesús Rooney Rivera Guillén

Suplente

Dr. Iván Ramón Terol Villalobos

Suplente

Dr. Manuel Toledano Ayala
Director de la facultad

Dra. Ma. Guadalupe Flavia Loarca Piña
Directora de investigación y posgrado

Centro Universitario
Querétaro, Qro.
Octubre de 2018
México

RESUMEN

En este trabajo se presenta una serie de metodologías realizadas para filtrado de mapas de profundidad de áreas de trabajo de robots manipuladores de 6 grados de libertad utilizando para ello conceptos de procesamiento de imágenes centrado en filtrado no lineal basado en la teoría de la morfología matemática. Se presentan una metodología que sólo filtra mapas de profundidad individuales, posteriormente se presenta una metodología que combina mapas de profundidad con imágenes color; ambas presentan resultados superiores a las herramientas que se encuentran comúnmente en el estado del arte. Adicionalmente se presentan un par de metodologías adicionales centradas en cuantificar daños en aspas de aerogeneradores eólicos que fue un trabajo en colaboración con la industria, así como un trabajo centrado en el área médica para medir desbalances térmicos en termogramas para detección oportuna de indicios de cáncer de mama.

(Palabras clave: mapas de profundidad, robot manipulador, morfología matemática, filtrado, reconstrucción)

SUMMARY

This thesis presents a series of methodologies for filtering depth maps for the work of 6 degrees of freedom manipulative robots to also use it for image processing focused on nonlinear filtering based on the theory of morphology math. A methodology that only filters individual depth maps is presented, then a methodology that combines depth maps with color images is presented; The results are shown in the state of the art. In addition, it is a method of focusing on the management of energy, as well as a work focused on the medical area to measure thermal and thermogenic imbalances for the early detection of signs of breast cancer.

(Keywords: depth maps, robot manipulator, mathematical morphology, filtering, reconstruction)

A Cristina (q.e.p.d) por inculcarme el amor a la escuela. A Eugenio por enseñarme el significado del trabajo. A Carlos por mostrarme el valor de la amistad.

AGRADECIMIENTOS

A mi familia por siempre ser mi principal apoyo en todo momento. Ustedes son lo más importante que tengo. No sería nada sin ustedes y sin sus enseñanzas. Gracias por todo, sin duda su ejemplo ha marcado lo que soy.

Al Dr. Luis Alberto por guiarme desde la licenciatura en mi aventura académica y por mostrarme el maravilloso potencial del procesamiento de imágenes. Su apoyo a nivel académico e incluso a nivel personal ha sido fundamental para llegar hasta donde estoy.

Al Dr. Roque Alfredo por ser un gran ejemplo que seguir en el plano académico y de la investigación, gracias por la confianza; y al Dr. Iván por encender ese chispazo referente a los temas avanzados de la morfología matemática.

A todos mis compañeros del posgrado; en especial a Juan José Saucedo, Emmanuel Reséndiz, Irving Cruz y a Gilberto Alvarado por su ayuda, pero sobre todo por su amistad. Y por supuesto, a mi amigo desde la prepa Jesús Iván Sánchez por enseñarme a programar como se debe.

También agradezco a alguien que llegó hacia el final de mi doctorado y que desde que la conozco las cosas han fluido de una forma tan genial que hasta parece magia, y porque ha estado apoyándome en este último esfuerzo. Muchas gracias Nadia.

Finalmente, agradezco al Consejo Nacional de Ciencia y Tecnología (CONACyT) por haberme ofrecido una beca de manutención durante la realización de este trabajo de investigación. La ciencia es el camino para sacar adelante a este país.

ÍNDICE GENERAL

1.	Introducción	10
1.1	Investigaciones relacionadas con calibración y estimación de posición de robots 12	
1.1.1	Sistemas de metrología convencionales	13
1.1.2	Sistemas de visión artificial.....	15
1.2	Investigaciones relacionadas con sistemas TOF	16
1.2.1	Investigaciones de carácter internacional.....	16
1.2.2	Investigaciones locales relacionadas con sistemas TOF	18
	Objetivos e hipótesis.....	20
1.2.3	Objetivos.....	20
1.2.4	Hipótesis	20
1.3	Justificación	20
1.4	Planteamiento general.....	21
2.	Fundamentación teórica	23
2.1	Mapas de profundidad	23
2.1.1	TOF.....	24
2.1.2	Luz estructurada	24
2.2	Mediana	25
2.3	Morfología matemática.....	26
2.3.1	Erosión y dilatación.....	26
2.3.2	Apertura y cerradura.....	27
2.3.3	Filtros alternados	28
2.3.4	Filtros por reconstrucción.....	29
2.3.5	Detectores morfológicos de contornos	30
2.4	Segmentación color usando histogramas bi-variables.....	31
2.5	Búsqueda de plantillas	33
2.6	Python.....	34
2.7	OpenCV	35
3.	Metodologías de filtrado de mapas de profundidad	36
3.1	Metodología de filtrado usando sólo mapas de profundidad.....	37
3.2	Metodología de filtrado combinando mapas de profundidad con imágenes color...	41

3.3	Metodologías adicionales desarrolladas durante este trabajo de tesis.....	44
3.3.1	Metodología para cuantificar fallas en aspas de aerogeneradores.....	45
3.3.2	Metodología para detección de desbalances térmicos en imágenes de mamas	46
4.	Resultados y discusiones.....	48
4.1	Filtrado de mapas de profundidad usando la cerradura por reconstrucción modificada MCbR	48
4.2	Filtrado de mapas de profundidad basado en características de color, forma y vecindad	54
4.3	Aplicaciones de las metodologías desarrolladas a la robótica.....	57
4.4	Resultados de metodologías adicionales desarrolladas durante este trabajo de tesis	62
4.4.1	Caracterización de fallas en aspas de aerogeneradores usando procesamiento de imágenes.....	63
4.4.2	Detección de desbalances térmicos en imágenes termográficas para detección temprana de cáncer de mama.....	63
4.4.2.1	Caso 1 - Saludable	65
4.4.2.2	Caso 2 - Saludable	65
4.4.2.3	Caso 3 - Angiogénesis	66
4.4.2.4	Caso 4 - Cáncer.....	68
5.	Conclusiones	70
6.	REFERENCIAS	72
7.	APÉNDICE I.....	77
8.	APÉNDICE II	78

ÍNDICE DE FIGURAS

Figura 1. Campos de aplicación de la calibración de sistemas robóticos.	12
Figura 2. Metodología propuesta.	22
Figura 3. Imagen color con su correspondiente mapa de profundidad.	23
Figura 4. Principio de funcionamiento de los dispositivos TOF.	24
Figura 5. Principio de funcionamiento de medición de profundidad basada en luz estructurada.	25
Figura 6. Acción del filtro de la media y mediana sobre una imagen con ruido sal y pimienta.	26
Figura 7. Erosión usando un elemento estructurante 3x3.	27
Figura 8. Dilatación usando un elemento estructurante 3x3.	27
Figura 9. Apertura usando un elemento estructurante 7x7.	28
Figura 10. Cerradura usando un elemento estructurante 7x7.	28
Figura 11. Ejemplo de la cerradura y apertura por reconstrucción.	30
Figura 12. Ejemplo de la cerradura y apertura por reconstrucción.	31
Figura 13. Conversión a espacio de color IHSL.	32
Figura 14. Ejemplo de segmentación color usando los histogramas bi-variables.	33
Figura 15. Ejemplo de segmentación color usando los histogramas bi-variables.	34
Figura 16. Proceso general de procesamiento de imágenes utilizado en esta tesis.	36
Figura 17. Metodología para filtrado de mapas de profundidad.	37
Figura 18. Comparativa de filtrado iterativo de un mapa de profundidad sintético usando el algoritmo original de la cerradura por reconstrucción y el algoritmo modificado propuesto.	39
Figura 19. Metodología para filtrado de mapas de profundidad color.	42
Figura 20. Proceso de clasificación de huecos.	43
Figura 21. Proceso completo de filtrado de mapas de profundidad basando en características de color, forma y vecindad.	44
Figura 22. Metodología propuesta para la clasificación de daños en aspas de turbina eólicas tipo de eje horizontal (HAWT).	45
Figura 23. Metodología propuesta para la detección de desbalances térmicos en imágenes de mamas para detección oportuna de cáncer de mama.	47
Figura 24. Proceso de filtrado de cones-disp2, a) imagen color, b) mapa de profundidad original, c) mediana 31x31, d) dilatación 8, e) cerradura 8, f) ASF W 9, g) ASF B 9, h) cerradura por reconstrucción 8 e i) método propuesto 8.	49

Figura 25. Proceso de filtrado de teddy-disp2, a) imagen color, b) mapa de profundidad original, c) mediana 29x29, d) dilatación 7, e) cerradura 7, f) ASF W 7, g) ASF B 7, h) cerradura por reconstrucción 7 e i) método propuesto 7.	50
Figura 26. Pasos de filtrado de cones-disp2.	51
Figura 27. Pasos de filtrado de teddy-disp2.	51
Figura 28. Imágenes del set Tsukuba de prueba, a) imagen original frame_1.png, b) imagen original frame_100.png, c) frame_1.png con ruido erosionado 5 y d) frame_100.png con ruido erosionado 3.	52
Figura 29. Imágenes de pruebas para probar la metodología propuesta, a) imagen color im2.png y su correspondiente, b) mapa de profundidad disp2.png, c) imagen color im6.png y su correspondiente y d) mapa de profundidad disp6.png.	54
Figura 30. Filtrado de cones-disp2, a) filtro bilateral, b) filtro anisotrópico, c) filtro de moda ponderada, d) muestreo superior bilateral, e) campos aleatorios de Markov (kernel data term), f) filtro noise-aware, g) cerradura por reconstrucción, h) cerradura por reconstrucción modificada e i) método propuesto.	55
Figura 31. Filtrado de cones-disp6, a) filtro bilateral, b) filtro anisotrópico, c) filtro de moda ponderada, d) muestreo superior bilateral, e) campos aleatorios de Markov (kernel data term), f) filtro noise-aware, g) cerradura por reconstrucción, h) cerradura por reconstrucción modificada e i) método propuesto.	56
Figura 32. Sensores TOF montados sobre el robot Yaskawa, a) sensor Mesa SR4000, b) sensor Microsoft Kinect versión 1.	58
Figura 33. Proceso de clasificación de fallas en turbinas de aerogeneradores.	63
Figura 34. Caso 1 Saludable, a) termograma original, b) y c) detección de los límites inferiores, d) polinomios de segundo grado, e) seno derecho, y f) seno derecho segmentados.	65
Figura 35. Caso 2 Saludable, a) termograma original, b) y c) detección de los límites inferiores, d) polinomios de segundo grado, e) seno derecho, y f) seno derecho segmentados.	66
Figura 36. Caso 3 Angiogénesis, a) termograma original, b) y c) detección de los límites inferiores, d) polinomios de segundo grado, e) seno derecho, y f) seno derecho segmentados.	67
Figura 37. Detalle del seno izquierdo con la forma tan característica de un caso de angiogénesis.	68
Figura 39. Detalle del seno derecho mostrando el punto más caliente.	69

ÍNDICE DE TABLAS

Tabla 3.1 Características analizadas cuantitativamente.	46
Tabla 4.1 Tiempos de procesamiento.	52
Tabla 4.2 Valores PSNR, MES y SSIM para el proceso de filtrado de frame_1.	53
Tabla 4.3 Valores PSNR, MES y SSIM para el proceso de filtrado de frame_100.	53
Tabla 4.4 Valores PSNR, MES y SSIM para el proceso de filtrado de cones-disp2.png. ...	55
Tabla 4.5 Valores PSNR, MES y SSIM para el proceso de filtrado de cones-disp6.png. ...	57
Tabla 4.6 Prueba de medición de profundidad a diferentes alturas, imágenes originales.	59
Tabla 4.7 Prueba de medición de profundidad a diferentes alturas, imágenes después de un proceso de filtrado.	61
Tabla 4.8 Validación usando la DMR.	64
Tabla 4.9 Validación usando el estudio clínico propio.	64
Tabla 4.10 Caso 1 Saludable.	65
Tabla 4.11 Caso 2 Saludable.	66
Tabla 4.12 Caso 3 Angiogénesis.	67
Tabla 4.13 Caso 4 Cáncer.	68

Capítulo I

1. Introducción

El campo de aplicación de los robots industriales es amplio como lo son ensamble, manufactura, soldadura, pintura, sujeción de precisión, entre muchas otras actividades. Estos trabajan, por lo general, en procesos con ambientes conocidos y estructurados, donde el movimiento del robot es pre-programado y ejecutado de manera continua, repitiendo las mismas rutas y movimientos, por eso se dice que los robots tienen una muy buena repetibilidad. Industrialmente hay varias maneras de medir el desempeño de un robot manipulador. Algunas de estas incluyen, ciclos de trabajo completados, funcionalidad de operaciones, operaciones pick & place por minuto, velocidad de reposicionamiento, repetibilidad, entre muchas otras. Específicamente la repetibilidad es la habilidad de un robot manipulador de regresar repetidamente a la misma posición. Los robots industriales poseen buena repetibilidad, pero carecen de precisión absoluta.

La precisión está en función de la exactitud del modelo cinemático (tipo de articulación, longitud de los eslabones, ángulos entre eslabones, la cantidad de deflexión de los eslabones o articulaciones bajo carga, entre otras). La principal razón es la diferencia entre el modelo cinemático ideal integrado en el controlador y los parámetros reales. Sin embargo, es importante hacer notar el problema de precisión absoluta presente en robots manipuladores de 6 grados de libertad, (Siciliano *et al.*, 2009). Por ejemplo, los robots se fabrican con cierto grado de tolerancia con lo que no hay dos robots del mismo modelo exactamente iguales.

Motta (2006) menciona que la calibración de robots es el proceso en el cual la precisión del posicionamiento del robot puede ser mejorada modificando el software de posicionamiento del robot, en vez de cambiar o alterar el diseño del robot o su sistema de control. Su objetivo principal es identificar los parámetros geométricos de Denavit-Hartenberg (DH). La calibración juega un papel de creciente importancia en mejorar la precisión de robots. El éxito de la programación *off-line* de un robot recae no sólo en que se logre repetibilidad, sino precisión. Los robots como dispositivos mecánicos se ven afectados por ligeros cambios o desviaciones causados por el

desgaste de las partes, desviaciones dimensionales, tolerancias y remplazo de componentes; la calibración puede minimizar la necesidad de tener que cambiar los programas de la aplicación debido a alguno de las situaciones mencionadas anteriormente.

La calibración se puede clasificar en dos tipos: calibración basada en modelos paramétricos y calibración basada en modelos no paramétricos (Siciliano, 2008). Para el primero los modelos son descritos con pocos parámetros, siendo adecuados para caracterizar la precisión del modelo en todo el rango de trabajo. Ejemplos de esto son los parámetros de DH y los parámetros inerciales del cuerpo rígido. La calibración basada en los modelos no paramétricos incluye la respuesta al impulso y diagramas de Bode para sistemas lineales, y los *kernel*s de Wiener y Volterra para sistemas no lineales (Xuan *et al.*, 2014). Este tipo de calibración se ocupa cuando las propiedades del sistema son tan complicadas que algunos cuantos parámetros agrupados no son suficientes, como ocurre en los sistemas biológicos. La calibración basada en el modelo paramétrico es apropiada para robótica donde los componentes son manufacturados y cuyas propiedades son controladas y entendidas.

La calibración cinemática basada en modelos paramétricos implica el proceso de ubicar los sistemas coordinados relativos entre los objetos de los sistemas. Por ejemplo, ubicar un robot relativo a un sistema de referencia global, ubicar un sistema de visión con referencia a un robot, localizar un objeto con referencia al marco de referencia de la herramienta de sujeción del manipulador, ubicar sistemas coordinados vecinos entre eslabones del robot manipulador, entre otros. Para llevar a cabo esto, el procedimiento involucra desarrollar un modelo cuyos parámetros representen de manera precisa el modelo del robot. A continuación, se eligen ciertas características específicas del robot para ser medidas. Este paso es seguido por un proceso de identificación para calcular los valores de esos parámetros, los cuales al entrar al modelo del robot reflejen fielmente las mediciones realizadas. La calibración de robots en conjunto con la programación off-line permiten programar un solo robot y copiar los programas de uno a otro (Elata *et al.*, 2004). Esto se traduce en minimizar los gastos de mantenimiento, dado que el robot puede volver rápido a servicio y una fácil programación significa que el software se vuelve más funcional con el menor esfuerzo.

La calibración de robots juega un papel importante en la producción de robots, así como en su implementación y operación dentro de manufactura o ensamble de sistemas. La producción,

implementación y operación de robots son todas áreas donde la calibración de robots puede llevar a una mejora significativa de la precisión y ahorro de recursos. De manera general se puede aplicar la calibración de robot mediante sistemas de visión para los aspectos mostrados en la Figura 1.

1.1 Investigaciones relacionadas con calibración y estimación de posición de robots

Elata *et al.* (2004), Motta (2006) y Xuan *et al.* (2014), mencionan cuales son los pasos generales del proceso de calibración de robots manipuladores en sendas revisiones del tema. Mencionan que en primer lugar es necesario contar con un modelo del robot cuyos parámetros lo representen de manera precisa, basándose muchas veces en la información de los diferentes elementos que los conforman la cual es proporcionada por los mismos fabricantes. A continuación, se debe realizar la identificación numérica de las características físicas del robot usando instrumentos con cierta precisión. Esta información debe ser procesada para generar un nuevo modelo que de manera más precisa represente el comportamiento del robot. Finalmente se lleva a cabo la implementación del nuevo modelo directamente sobre el software de posicionamiento. Dentro de sus trabajos igualmente mencionan los niveles de calibración (1, 2 y 3), las ventajas de cada uno de ellos y prospectivas del desarrollo del tema, en que se deja de manifiesto que dicha problemática se va adaptando al desarrollo e implementación de nuevas tecnologías de sensado y medición.

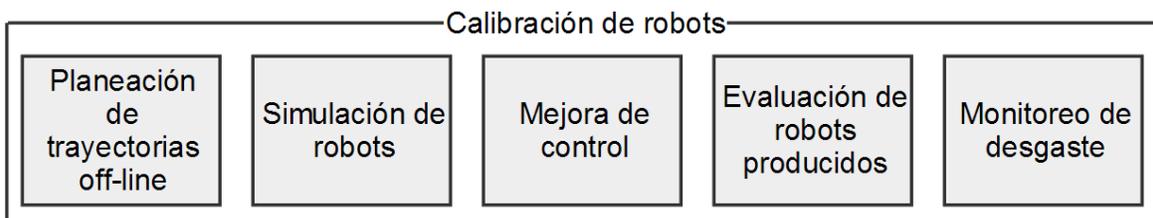


Figura 1. Campos de aplicación de la calibración de sistemas robóticos.

- Planeación off-line y para tareas de simulación de robots: La calibración de robots hace posible la implementación de sistemas integrales de cómputo fuera de línea. Con esto un gran ahorro de tiempo es posible y los costos debidos a errores pueden ser evitados cuando las tareas del robot pueden ser planeadas y simuladas fuera de línea. Para que un sistema de control ejecute de manera exitosa un programa generado en la planeación y simulación del sistema fuera de línea, las dimensiones reales del robot y de la celda

de trabajo deben ser muy similares a aquellas de los modelos usados para el diseño de programas. Si no, sus diferencias en dimensiones se traducirán en errores de posicionamiento cuando el programa se ejecute en el taller o planta de producción.

- Mejora del control y simulación del movimiento del robot: Parámetros inerciales. Control preciso, y simulación de los movimientos del robot. La fricción y rigidez estructural juegan un papel importante en esta área.
- Evaluación de la producción de robots: Una vez armados los robots es difícil medir sus parámetros geométricos, la calibración de robots ofrece una forma práctica y efectiva de medir implícitamente dichos parámetros.
- Monitoreo de desgaste de componentes del robot: La re-calibración periódica permite determinar si es necesario llevar a cabo reparaciones o si se requieren ajustes de programación.

1.1.1 Sistemas de metrología convencionales

El aspecto de calibración de robots aborda la utilización de una gran variedad de sistemas de metrología, los cuales pueden ser categorizados basándose en el número de elementos de posición que permiten medir (Hollerbech and Wampler, 1996). Para sistemas que miden un solo elemento se tienen, por ejemplo, la utilización de equipos como un *ball bar* instrumentado (Goswami *et al.*, 1993), potenciometros lineales (Driels y Swayze, 1994), y medidores de desplazamiento láser (Tang y Liu, 1993). Para dos componentes se tiene la utilización de teodolitos (Whitney *et al.*, 1986), método en el cual es necesario utilizar una distancia conocida como referencia para hacer un escalamiento. El monitoreo láser por su parte provee mediciones acertadas en tres dimensiones (3D), reflejando un rayo hacia un retro reflector montado sobre el efector final. El rayo proporciona información de longitud, mientras que el cardan usado para la dirección del láser proporciona dos mediciones de orientación (Mooring *et al.*, 1991). Dada que la parte menos precisa de esta configuración es la medición de ángulos, otra forma de abordar la medición 3D es la utilización de tres sistemas de seguimiento láser, ocupando solamente la información de distancia. Los sistemas de movimiento cámara estéreo 3D también proporcionan alta precisión en la medición de posición. Lau *et al.* (1985) presentaron un interferómetro láser orientable con reflector igualmente orientable. Con la información de guiñada y cabeceo, el interferómetro orientable obtiene las tres componentes de posición, mientras que el reflector orientable obtiene dos componentes de orientación. La posición completa puede ser estimada a partir de la posición

3D de múltiples puntos en el último eslabón con un sistema de visión estéreo. Este sistema coordinado entrega la posición además de la orientación del sistema (An *et al.*, 1988). Vincze *et al.* (1994) llevaron a cabo la medición completa de posición con un sistema de seguimiento láser de un sólo rayo. La posición fue medida usando interferometría, muy popular en esos años, aportando la medición de la orientación por formación de imágenes del patrón de difracción de los bordes del retro reflector. Conrad *et al.* (2000) propusieron una metodología de calibración de contacto para un robot manipulador utilizando un palpador como efector final. La motivación principal de este trabajo se centra en la integración de tecnologías comerciales analizando aspectos como precisión, repetibilidad y resolución del dispositivo. Bai *et al.* (2003) propusieron un sistema de rastreo utilizando un láser fijo en una posición conveniente y un blanco colocado en el efector final para ser seguido con el láser para llevar a cabo la calibración de un robot tipo PUMA. Nubiola *et al.* (2012) propusieron un método novedoso de medición para un conjunto de más de cincuenta posiciones de un robot bajo condiciones estáticas, empleando para ello un dispositivo *ball bar* telescópico junto con un dispositivo sujeto al efector final del robot con tres puntos magnéticos. Mencionan que el dispositivo de medición maneja una precisión aceptable lo cual lo hace muy competitivo contra sistemas de seguimiento láser, además de tener un costo bastante asequible. La precisión de la medición de posición oscila entre ± 0.003 mm. Klimchik *et al.* (2013) llevaron a cabo la calibración de robots seriales industriales usando mediciones parciales de posición. Dichas mediciones se realizan con la ayuda de un dispositivo de medición colocado como efector final y un seguidor láser, una de las desventajas es que necesita realizar la medición de múltiples puntos de referencia para lograr buenos resultados. Nubiola *et al.* (2014), realizaron la calibración de un robot manipulador complementando la medición física de posición realizada mediante la utilización de un *ball bar* telescópico con tecnología láser. Mediante la utilización de un dispositivo como efector final con seis puntos magnéticos llevan a cabo seis mediciones, colocando un extremo del *ball bar* en su base magnética y el otro extremo en los puntos magnéticos, en 72 posiciones de operación del robot. Complementando la información del *ball bar* con tecnología láser menciona que logra disminuir el error de posición de 0.873 mm a 0.479 mm. Nubiola *et al.* (2014) llevaron a cabo la calibración de un robot manipulador usando un sistema de seguimiento láser, reducen el error de posición media/error máximo de posición de 0.968 mm/2.158 mm hasta 0.364 mm/0.696 mm. Llama la atención la cantidad de mediciones necesarias para lograr tal desempeño, que fueron en su caso 1000 pruebas.

1.1.2 Sistemas de visión artificial

La utilización de sistemas de visión artificial juega un papel importante en la calibración de robots. Ma *et al.* (2006) presentaron un sistema de medición basado en visión artificial para estimar el error de orientación de un robot. Llevan a cabo la medición realizando la calibración por parámetros extrínsecos (R y T , donde R es una matriz de rotación y T es la posición del origen del sistema coordenado expresado en coordenadas del sistema coordenado del centro de la cámara) dos veces, antes y después del movimiento del efector final del robot. El elemento final del robot tiene colocado un tablero de calibración tipo ajedrez y además usan una cámara de imagen color estándar. Su método propuesto tiene un error menor a 0.1 grados el cual podría ser reducido usando múltiples sistemas de visión o otros sensores. Li *et al.* (2009) simularon una metodología para calibración de manipuladores seriales en línea, mediante la colocación de un sistema de visión en el efector final acompañado del procesamiento de las imágenes obtenidas. Xue *et al.* (2011) mencionan que "la planeación de movimientos o trayectorias basada en sensores (como los de tiempo de vuelo, TOF por sus siglas en inglés) permiten a los robots moverse dentro de ambientes dinámicos y desordenados sin colisionar". En su trabajo proponen una metodología que permite al robot manipulador realizar el agarre y manipulación de objetos de manera automática utilizando una cámara TOF. Hace mención de las ventajas computacionales que conlleva trabajar sólo con un sensor de tiempo de vuelo comparado con la visión estéreo. Particularmente destaca la utilización del sensor TOF para corregir la trayectoria del robot ante objetos no deseados en el área de trabajo y, en una segunda parte, la utilización de la información del sensor de profundidad para recoger helado, el cual varía su altura en cada pasada con la cuchara con la que lo recoge. Wang *et al.* (2012) propusieron un método de calibración robótica, utilizando un sistema de visión y un dispositivo de trabajo cuadriculado para obtener la posición de un manipulador mediante procesamiento de imágenes. Hu *et al.* (2012) propusieron una metodología que hace uso de un dispositivo de visión artificial y un sistema láser para la calibración de un robot manipulador en línea utilizando procesamiento de imágenes. Svaco *et al.* (2013), realizan el proceso de calibración de un robot industrial utilizando un sistema de visión estéreo. Hacen mención que una ventaja muy importante de realizar la calibración usando sensores de visión es que esta se hace sin necesidad de contacto. En su caso disminuye el error de posicionamiento del robot de 3.63 mm a 1.29 mm. Du y Zhang (2013) mencionan que la calibración de robots es un método de diagnóstico útil para mejorar la precisión y ayudar en las tareas de mantenimiento de este. Colocaron un sistema de

visión firmemente sujeto como efector final y alrededor del robot pusieron un tablero de calibración, del cual se buscan sus esquinas como referencias de posición del robot, al alcance de la vista del sensor. La calibración usando visión artificial puede ser llevada a cabo sin tocar en ningún momento al robot. Finalmente mencionan que la utilización de una cámara con mas elementos de la imagen (por ejemplo, distancias de la escena) podría mejorar la precisión. Sánchez-López *et al.* (2013) mencionan que la manipulación de objetos por robots móviles sigue siendo una tarea desafiante, así mismo que dicha tarea se descompone en tres fases, aproximación a los objetos, planificación de rutas y trayectorias para el brazo manipulador, y sujeción y ajuste fino. Precisamente propusieron un sistema de servo posicionamiento visual tridimensional para un manipulador autónomo móvil atacando la problemática para agarre y ajuste fino usando visión estéreo, la cual es combinada con segmentación de imágenes, reconstrucción 3D, y estimación de posición.

1.2 Investigaciones relacionadas con sistemas TOF

1.2.1 Investigaciones de carácter internacional

Chiabrando *et al.* (2009) llevaron a cabo la calibración y evaluación métrica del dispositivo TOF SR4000. Mencionan que es posible reducir el error sistemático presentes en las mediciones realizadas con estos dispositivos ocasionados tanto por los componentes internos como por los propios de la escena observada. Logran una excelente reducción en los errores presentes en el rango entre 1.5 m a 4 m, sin embargo, mencionan que es necesario mejorar en rangos más cortos y en los más lejanos. Pomares *et al.* (2010) mencionan la importancia que han ganado los sistemas TOF en aplicaciones que requieran obtener información 3D del área de trabajo. Sugieren la utilización de un dispositivo de este tipo para guiar un brazo robótico. Usaron la información de profundidad que proporciona el sensor TOF como retroalimentación visual. Proponen además un método de calibración basado en la actualización del tiempo de integración del dispositivo TOF que permita reducir los errores sistemáticos por factores externos (como la luz del sol e iluminación posterior) y de la misma escena (como distancias, orientaciones, y reflectividad de los objetos) a fin de eliminar fluctuaciones que afecten el control.

Smisek *et al.* (2011) analizaron experimentalmente las mediciones de profundidad del sensor Kinect en términos de su resolución y precisión. Además, realizaron una comparación

cuantitativa de las capacidades de medición 3D para tres diferentes cámaras, incluyendo un sensor Kinect, una cámara estéreo y un dispositivo TOF. Sorpresivamente sus resultados revelan que las mediciones realizadas por medio del sensor Kinect son inclusive más precisas que las de una cámara TOF y equiparables a las obtenidas por un sistema de visión estéreo. En ese mismo sentido Stoyanov *et al.* (2011) compararon el sensor Kinect con otros dos dispositivos TOF de medición 3D. Las mediciones se contrastan contra las obtenidas por medio de un sensor láser de alta precisión. Para un rango menor a 3.5 metros las mediciones con el sensor Kinect son cercanas a las hechas con el sensor láser. Los sistemas TOF tienen diferencias significativas a las mediciones realizadas con láser en rangos cortos. Los tres sistemas presentan mucha disparidad en rangos de medición largos. Finalmente concluyen que el sensor Kinect es la opción ideal en caso de necesitar trabajar en un rango de distancias corto con un precio muy competitivo. Piatti y Rinaudo (2012) realizaron la comparativa de desempeño entre dos dispositivos TOF comerciales, la cámara SR4000 y la CamCube3.0. Es de llamar la atención que la CamCube 3.0 requiere de un tiempo de calentamiento que llega a ser de hasta 40 minutos para lograr una cierta estabilidad en las mediciones que entrega, cuyas variaciones llegan a ser de hasta 12 centímetros en frío. En ambos casos la precisión de las mediciones oscila entre 3 a 4 mm. Una de las ventajas de la CamCube 3.0 es que cuenta con un sistema de supresión de iluminación de fondo haciéndola más robusta ante iluminación externa. La precisión de estos equipos los hace candidatos a ser utilizados en aplicaciones de robótica. Khoshelham *et al.* (2012) proporcionan una idea de la calidad geométrica de la información de profundidad del sensor Kinect basados en el análisis de la precisión y resolución de la información de profundidad. Mediante sus resultados experimentales muestran que el error aleatorio de la medición de profundidad aumenta cuando la distancia entre los objetos y el sensor crece, variando desde unos cuantos milímetros hasta 4 centímetros en el rango máximo de operación del dispositivo. Cho y Lee (2012) desarrollaron un sensor para interacción humano-robot, para guiar robots. El dispositivo está enfocado en la ayuda de personas de la tercera edad y débiles visuales. Básicamente se tiene un brazo robótico unido a una andadera que sirve para dar dirección a la misma, mientras que con sistema TOF se obtiene la posición del usuario, a partir de la cual el brazo robótico se ajusta para mantener una distancia segura con el usuario, mover la andadera y no estorbarle en sus actividades. Se deben resaltar las grandes ventajas que ofrecen los sistemas TOF para múltiples aplicaciones, al entregar información cuantitativa de profundidad como retroalimentación. Chen *et al.* (2012), lograron calcular parámetros cuantitativos como el índice de área y el ángulo de distribución de la hoja del maíz haciendo uso sólo de la información

de profundidad provista por el sensor Kinect. La información de profundidad forma una nube de puntos con la cual llevan a cabo un proceso de reconstrucción mediante una red de triangulación irregular. Para llevar a cabo su proceso hacen uso de Visual Studio C++ y OpenGL (Open Graphics Library). Mediante este método obtienen resultados con un error de hasta 25 por ciento comparando la medición real contra la obtenida a través del sensor. El combinar su uso con la información de color provista por el mismo sensor Kinect podría mejorar la precisión de la medición e incluso hacer más intuitivo el uso del sistema para el usuario final. Park *et al.* (2012) estimaron el modelo de incertidumbre espacial de características visuales de un sensor TOF Kinect. La intención de este trabajo es proporcionar un análisis cualitativo y cuantitativo para la utilización de dicho dispositivo como sensor de percepción 3D. Para lograr esto, derivaron la relación de la propagación de la incertidumbre entre el espacio de disparidad de la imagen y el espacio real cartesiano, con la función de mapeo entre los dos espacios. Cruz *et al.* (2012) y Han *et al.* (2013) llevaron a cabo una revisión muy completa sobre el sensor Kinect, hacen una semblanza histórica acerca de él, mencionan sus principales características, sus ventajas y desventajas, además enumeran trabajos realizados a nivel académico que hacen uso del sensor. Ambos documentos sirven de base para la realización de trabajos que usen este dispositivo. Deben resaltarse los problemas de ruido y huecos que presentan las imágenes adquiridas por el equipo, así como el desfase existente entre las imágenes *RGB* y de profundidad que entrega el sensor. Qi *et al.* (2014) desarrollaron una metodología para medir la deflexión de vigas de concreto reforzadas sujetas a cargas cíclicas a partir de la información entregada por el sensor Kinect. Implementaron una serie de técnicas de segmentación para extracción de objetos a partir de la información de profundidad del sistema TOF, de igual forma diseñaron una serie de algoritmos de reconstrucción de desplazamiento vertical en función del tiempo presente en las vigas de hormigón. En su trabajo demuestran que la amplitud y frecuencia de los desplazamientos verticales pueden ser reconstruidas con precisión sub-milimétrica y de mili-Hertz respectivamente.

1.2.2 Investigaciones locales relacionadas con sistemas TOF

Hernández *et al.* (2012) propusieron la implementación de un algoritmo para la detección de objetos en ambientes cerrados usando técnicas de segmentación color en el espacio CIE-Lab y de profundidad a imágenes adquiridas a través del sensor Kinect. Aplican además un proceso de homografía para sincronizar la imagen *RGB* y de profundidad provistas por el sensor Kinect. La

detección de objetos funciona adecuadamente, sin embargo, para aplicaciones específicas contar con información como la distancia a la que se encuentran los objetos a modo de retroalimentación podría brindar un mundo de oportunidad en robótica y control. Peralta (2012) desarrolló, una interfaz para interpretar el lenguaje natural del ser humano usando el sensor Kinect. Para realizar su trabajo utilizó el lenguaje C/C++ y OpenNI (Open Natural Interface) bajo sistema operativo Linux. A fin de verificar el correcto funcionamiento de su sistema desarrollado, implementó tres diferentes entornos, primero un ratón virtual, un sistema de menús y una interfaz para manipulación de objetos virtuales. Para corregir el problema de ruido presente en las imágenes entregadas por el sensor Kinect hizo uso de un filtro Kalman. En la UAQ, Díaz *et al.* (2012) realizaron una comparativa entre dos aplicaciones enfocadas al reconocimiento de patrones en vídeo para el control virtual de un mouse, una desarrollada bajo la aplicación de software libre *Processing* y la otra mediante C# en el SDK (Software Development Kit) de Microsoft para Kinect. Implementaron un sistema para la detección del usuario y asignan funciones del ratón a los gestos obtenidos mediante el rastreo de sus manos. Mencionaron que debido a que el SDK de Microsoft es de uso específico para el Kinect, éste puede sacar todo el provecho de dicho hardware para desarrollar una gran cantidad de aplicaciones, aunque no es una limitante total para no hacer uso de software libre. De León *et al.* (2012), llevaron a cabo una comparativa para determinar el potencial del sensor Kinect contra medios tradicionales para el control de un vehículo móvil. El control de la dirección del vehículo se realiza mediante un volante virtual que son las manos del usuario simulando a éste. Mediante el Kinect se detecta la posición de éstas y se determina la dirección y la velocidad del vehículo. Si bien acostumbrarse a esta nueva forma de control es difícil al principio, basta un rato de pruebas para lograr resultados similares a los obtenidos contra un sistema de control físico tradicional. Garduño-Ramón *et al.* (2014a) desarrollaron un sistema para cuantificar parámetros corporales que sirve como auxiliar en análisis postural mediante la función de detección de articulaciones del sensor Kinect. Su mayor aportación fue verificar la repetitividad que ofrece el dispositivo, demostró que las variaciones entre las mediciones que entrega el sensor llegan a ser menores a 1 cm, con lo cual se dice que el dispositivo posee buena repetitividad. Continuando sus investigaciones con el sensor Kinect. Garduño-Ramón *et al.* (2014b) propusieron aplicar filtros morfológicos, y compararlos con filtros estadísticos, a la información de profundidad que entrega el sensor Kinect de manera numérica. En base a esto logran recuperar una buena cantidad de información de profundidad de la imagen la cual se pierde debido al ruido presente en los datos, principalmente ocasionado por el tipo de tecnología que ocupan estos dispositivos.

Objetivos e hipótesis

1.2.3 Objetivos

Desarrollar una serie de herramientas metodológicas basadas en procesamiento de imágenes utilizando filtros no lineales derivados de la teoría de morfología matemática para realizar reconstrucción de áreas de trabajo de robots manipuladores de 6 grados de libertad para realizar calibración de posición.

- Hacer una comparativa de desempeño de sistemas TOF existentes en el mercado mediante la extracción de parámetros dimensionales en ambientes controlados para analizar sus ventajas y desventajas.
- Desarrollar sistemas de reconstrucción tridimensional de objetos, de ambientes generales y de áreas de trabajo de robots manipuladores mediante el uso de sistemas TOF y herramientas de procesamiento de imágenes como segmentación morfológica en software para verificar su precisión.
- Desarrollar metodologías de filtrado no lineal que permitan reconstruir escenas generales y que puedan ser aplicados a áreas de trabajo de robots manipuladores.
- Utilizar sistemas de medición de profundidad basados en tecnología TOF y de luz estructurada para verificar la robustez de los métodos de reconstrucción de escenas propuestos ante ambas tecnologías.

1.2.4 Hipótesis

Las herramientas de procesamiento de imágenes en combinación con herramientas avanzadas de morfología matemática tienen la capacidad de reconstruir con una precisión elevada, comparada contra otras herramientas del estado del arte, áreas de trabajo de robots manipuladores de 6 grados de libertad caracterizadas usando sensores TOF para su posterior calibración.

1.3 Justificación

La calibración de robots manipuladores nivel 2 permite obtener un modelo más cercano a los parámetros reales del robot de modo que, sin alterar su configuración de hardware o la regla de control, se puedan realizar correcciones que mejoren su desempeño en la línea de producción. El sistema de visión estéreo es el más utilizado en el desarrollo de metodologías para realizar calibración nivel 2 a nivel científico y académico combinado con técnicas de procesamiento de imágenes con buenos resultados. Una tecnología que recientemente ha tenido grandes avances son los sistemas de visión TOF los cuales también puede cuantificar información de la escena, pero para ello utilizan una serie de emisores y sensores de luz infrarroja, con la gran ventaja de requerir sólo un punto de vista de la escena que significa una reducción en tiempos de procesamiento. A pesar de que en la literatura se enumeran las grandes ventajas de los sistemas TOF, cuyo desempeño se equipara en ciertos rangos de operación a los de sistemas láser, esta tecnología no ha sido aplicada directamente al desarrollo de metodologías que permitan integrarla junto con herramientas de procesamiento de imágenes para realizar calibración nivel 2 de robots manipuladores esto, sin duda, abre nuevas áreas de oportunidad y desarrollo. En base a esto y después de una extensa revisión de la literatura se puede decir que la aportación científica de este trabajo será el desarrollo de una metodología que integre tecnologías de visión TOF en conjunto con herramientas de procesamiento de imágenes que permita estimar la posición del efector final de un robot manipulador de 6 grados de libertad y llevar a cabo su calibración nivel 2, y que además permita un ahorro de tiempo en procesamiento dado las bondades que ofrece dicho sistema de visión. En el aspecto social y económico, la propuesta de este trabajo permitirá el desarrollo de una nueva serie de metodologías, así como la implementación de nuevas tecnologías funcionales, orientadas a resolver un conjunto de problemáticas muy específicas en la industria al mejorar la calidad de sus procesos operativos que involucren sistemas robóticos manipuladores, minimizando desperdicio de componentes, errores en producción e inclusive accidentes.

1.4 Planteamiento general

Con la descripción de problema, la hipótesis, así como el objetivo general y los particulares es posible proponer una serie de etapas metodológicas que permitan llevar a cabo la realización de este trabajo de investigación (Figura 2).

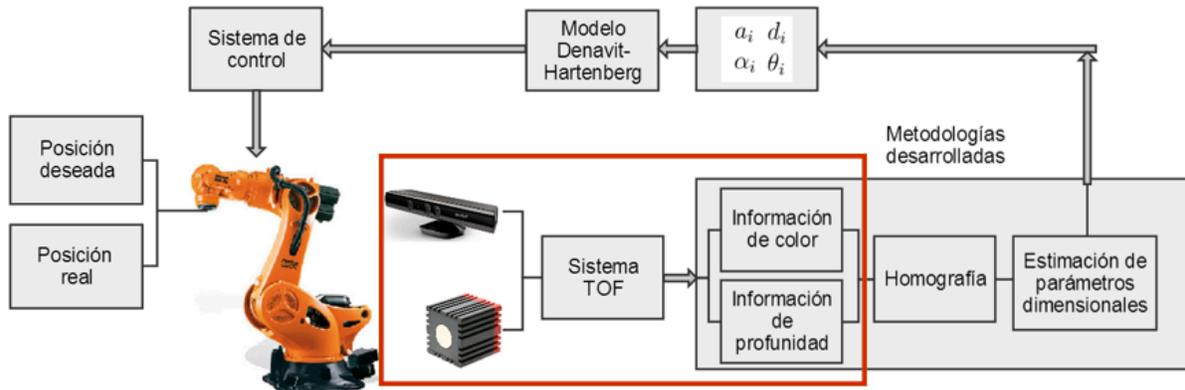


Figura 2. Metodología propuesta.

De manera general se tiene un robot manipulador el cual debe llegar a una posición deseada, sin embargo, como se observó en los antecedentes, esta posición es muchas veces diferente a la real que se pretende alcanzar. A partir de la información de profundidad de un sistema TOF, ya sea acoplado al robot como efector final o incluso viendo desde otra posición el área de trabajo del robot y al robot mismo, se busca en primer lugar estimar su error de posición y posteriormente sus parámetros geométricos, llevando a cabo antes un pre-procesamiento de la información de profundidad y, en su caso, de color que proporcionan los sistemas de tiempo de vuelo que existen en el mercado. La etapa de pre-procesamiento incluye filtrado por morfología y filtros estadísticos. Con la información procesada el paso siguiente es estimar los parámetros $a_i, d_i, \alpha_i, \theta_i$ necesarios para el modelo de Denavit-Hartenberg, los cuales entran a un algoritmo de calibración el cual a su vez retroalimenta al modelo del robot contenido en el controlador, de modo que se logre una mejor precisión en el posicionamiento del manipulador. De igual forma con dicha información es posible desarrollar simuladores más cercanos al modelo real del robot que sirvan para planificación y ajustes de trayectorias.

Capítulo II

2. Fundamentación teórica

Con el objetivo de desarrollar e implementar una metodología de filtrado de mapas de profundidad, se realizó una revisión de los diferentes aspectos teóricos y consideraciones que se deben de tener. Por lo tanto, en este capítulo se presentan aspectos teóricos relacionados con la implementación de la metodología de filtrado propuesta y que se desarrolla en esta tesis, con especial atención en teoría, técnicas y algoritmos cuya principal tarea se enfoca en el filtrado de imágenes usando morfología matemática.

2.1 Mapas de profundidad

Un mapa de profundidad (Figura 3) es una imagen bidimensional $m \times n$ generalmente codificada en 8 bits en canal sencillo, es decir en escala de grises. Cada píxel es una función $f(x, y)$ que contiene, en el rango de 0 a 255, un valor de gris que se puede relacionar con la distancia a la que se encuentra en unidades dimensionales. No existe una convención establecida, pero de manera general, un píxel negro en un mapa de profundidad significa un valor de profundidad desconocido; a partir de ahí tonos más claros indican regiones más cercanas, mientras que tonos más oscuros indican una mayor lejanía a la fuente de la imagen. Debido a su forma de adquisición se pueden clasificar dos tipos de mapas de profundidad, aquellos adquiridos mediante el principio de TOF y aquellos adquiridos mediante luz estructurada.



Figura 3. Imagen color con su correspondiente mapa de profundidad.

2.1.1 TOF

Los sistemas basados en el principio de TOF usan un hardware especial que consiste en un emisor de luz infrarroja que emite la luz de forma casi radial, así como un sensor de luz infrarroja que mide la luz infrarroja reflejada cuando ésta incide en los objetos. El emisor genera la luz a una frecuencia de fase conocida, y al chocar con los objetos y reflejarse de regreso se genera un desfase natural, este desfase es medido internamente por el sensor TOF y a partir de cálculos internos se hace una estimación de la profundidad a la que está ubicado el píxel correspondiente (Figura 4). Ejemplos de dispositivos que funcionan con este principio son la cámara MESA SR4000 y el sensor Kinect v2.

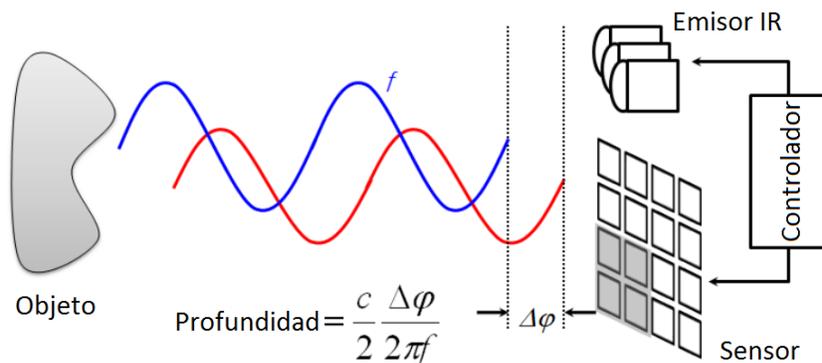


Figura 4. Principio de funcionamiento de los dispositivos TOF.

2.1.2 Luz estructurada

Por otro lado, existen mapas de profundidad generados a partir del principio de luz estructurada. Este principio consiste en proyectar un patrón de luz infrarroja, casi siempre una cuadrícula, la cual el fabricante calibra a diferentes distancias. Este patrón al incidir sobre una superficie o cierta escena invariablemente se verá deformada (Figura 5). A partir de esta deformación y el proceso de calibración llevado a cabo por el fabricante es lo que permite realizar una inferencia de la distancia a la que se encuentran los píxeles. Ejemplos de sensores que usan este tipo de tecnología son el Kinect v1.

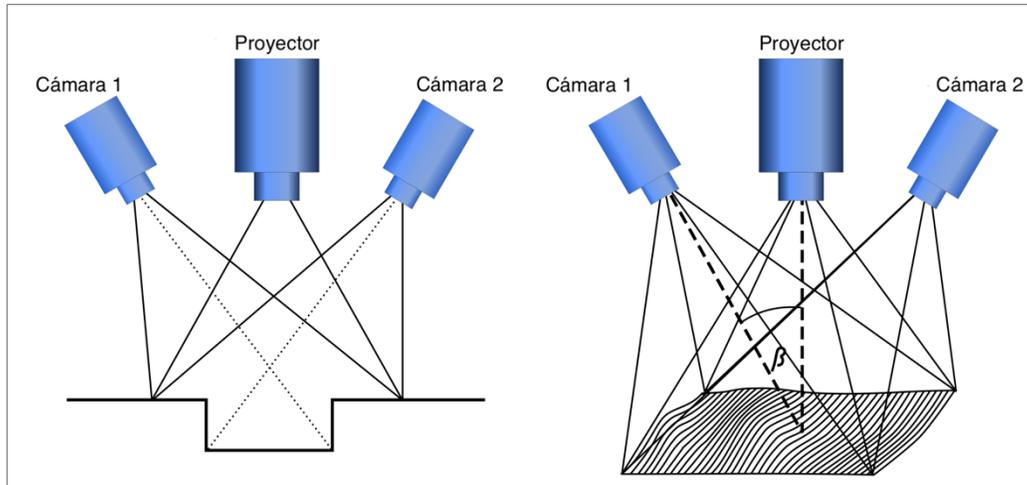


Figura 5. Principio de funcionamiento de medición de profundidad basada en luz estructurada.

2.2 Mediana

El filtro estadístico de mediana tiene un principio de funcionamiento sencillo que consiste en tomar los píxeles vecinos de un píxel de interés central y junto con él mismo estimar el valor de intensidad promedio, sustituyendo dicho valor en el lugar del píxel central. Esta vecindad de píxeles, generalmente con forma de ventanas cuadradas de tamaño $n \times n$ con n siendo un número impar para tener un píxel central se le denomina *kernel*. En la práctica se utilizan ventanas de tamaño 3×3 y 5×5 ya que tamaños mayores suelen deformar los objetos o generar o borrar regiones originales de la imagen, Figura 6. El filtro de mediana es un estándar para la eliminación de huecos pequeños o ruido del tipo sal y pimienta. En los softwares comerciales generalmente viene implementado por default y computacionalmente es fácil de implementar utilizando listas y funciones estándar que se encuentran en todos los lenguajes de programación modernos.

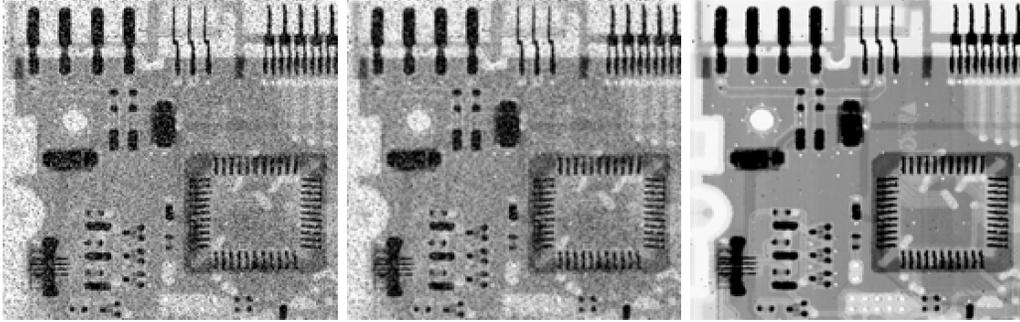


Figura 6. Acción del filtro de la media y mediana sobre una imagen con ruido sal y pimienta.

2.3 Morfología matemática

Desarrollada en Francia en los 60's, la morfología matemática es una poderosa técnica de procesamiento de imágenes basada en la teoría de conjuntos con sus funciones básicas: unión, intersección y complemento. El matemático Jean Serra planteó que los diferentes objetos en una cierta imagen pueden ser tratadas como regiones independientes e interactuar entre ellas mediante la teoría planteada por John Venn. A partir de sus estudios se formularon las bases para lo que hoy se conoce como el procesamiento de imágenes basado en morfológica matemática, de la cual a continuación se enumeran sus operaciones básicas, así como la formación de herramientas de filtrado elementales y complejas.

2.3.1 Erosión y dilatación

La erosión ($\varepsilon_{\lambda B}$) y la dilatación ($\delta_{\lambda B}$) son las funciones fundamentales de la morfología matemática. La primera se basa en el concepto de la intersección, mientras que la segunda en la operación unión. Su formulación se muestra en las Ecuaciones (2.1) y (2.2).

$$\begin{aligned} \varepsilon_{\lambda B}(f)(x) &= \bigwedge \{f(y) : y \in \lambda \check{B}\} \\ &= \min \{f(y) : y \in \lambda \check{B}\} \end{aligned} \tag{2.1}$$

$$\begin{aligned} \delta_{\lambda B}(f)(x) &= \bigvee \{f(y) : y \in \lambda \check{B}\} \\ &= \max \{f(y) : y \in \lambda \check{B}\} \end{aligned} \tag{2.2}$$

Donde λB denota un elemento estructurante de tamaño λ . B , representa el elemento estructurante básico de forma cuadrada 3×3 . Por su parte, \wedge y \vee denotan los operadores *infimum* y *supremum* respectivamente. Finalmente, \check{B} es el conjunto transpuesto $\check{B} = \{-x, x \in B\}$; y \max y \min son el mínimo y máximo valor respectivamente. La erosión tiende a desgastar los objetos de una imagen, su efecto puede ser visto en la Figura 7. Por su parte la dilatación crece los objetos de la imagen como se muestra en la Figura 8.

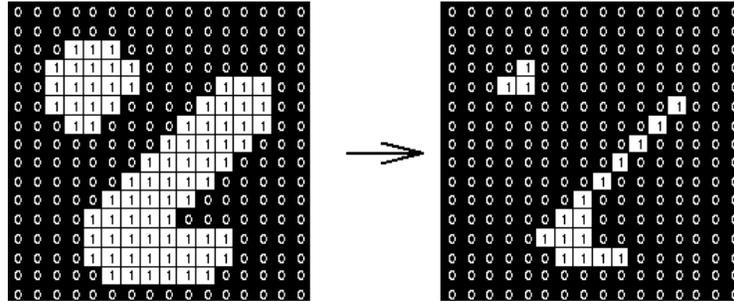


Figura 7. Erosión usando un elemento estructurante 3×3 .

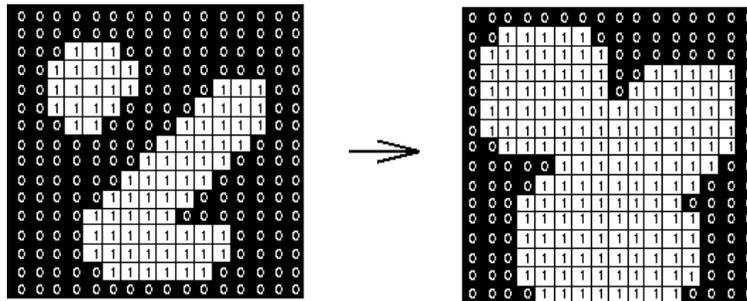


Figura 8. Dilatación usando un elemento estructurante 3×3 .

2.3.2 Apertura y cerradura

Un filtro morfológico es una operación que es no creciente e idempotente. La primera propiedad establece que el operador (ϕ) preserva el orden estructural, esto es $\forall f, g$ tal que $f \leq g \Rightarrow \phi(f) \leq \phi(g)$. Por su parte la propiedad de idempotencia significa que aplicando el operador una segunda vez no tendrá efecto sobre la imagen. Otras propiedades de los filtros morfológicos

son la extensividad y la anti extensividad. Estas propiedades significan que en algunos operadores morfológicos los objetos crecen, mientras que en otros se reducen. Los dos filtros morfológicos elementales son la apertura y la cerradura morfológica. El primero es extensivo, mientras que el segundo es anti extensivo. Su formulación se puede ver en la Ecuación (2.3).

$$\begin{aligned} \gamma_{\lambda B}(f) &= \delta_{\lambda B}(\varepsilon_{\lambda B}(f)) \\ \varphi_{\lambda B}(f) &= \varepsilon_{\lambda B}(\delta_{\lambda B}(f)) \end{aligned} \tag{2.3}$$

La operación del filtro de apertura puede ser vista en la Figura 9, mientras que el filtrado con cerradura puede ser visto en la Figura 10.



Figura 9. Apertura usando un elemento estructurante 7x7.



Figura 10. Cerradura usando un elemento estructurante 7x7.

2.3.3 Filtros alternados

Los filtros alternados secuenciales (*ASF*) también son operadores morfológicos increcientes e idempotentes. Básicamente ellos se forman a partir de la composición alternada de aperturas y cerraduras. La Ecuación (2.4) muestra su formulación.

$$\begin{aligned}\Psi_n(f) &= \varphi_{\lambda_n} \gamma_{\lambda_n} \dots \varphi_{\lambda_2} \gamma_{\lambda_2} \varphi_{\lambda_1} \gamma_{\lambda_1} \\ \Psi_n^*(f) &= \gamma_{\lambda_n} \varphi_{\lambda_n} \dots \gamma_{\lambda_2} \varphi_{\lambda_2} \gamma_{\lambda_1} \varphi_{\lambda_1}\end{aligned}\tag{2.4}$$

El filtro *ASF* para blancos $\Psi_n(f)$ y el *ASF* para regiones oscuras $\Psi_n^*(f)$ se denotan con su respectiva nomenclatura. El tamaño es decreciente en cada iteración de modo que se cumple $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$.

2.3.4 Filtros por reconstrucción

Las transformaciones por reconstrucción permiten eliminación de regiones no deseadas evitando una alta modificación de los objetos restantes. Las reconstrucciones morfológicas por reconstrucción están construidas a partir de la dilatación y la erosión geodésicas. La dilatación geodésica de tamaño 1 de un conjunto g (llamado marcador), dentro del conjunto f esta definida como $\delta_f^1(g) = \delta(g) \wedge f$. De una forma similar, la erosión geodésica de tamaño 1 esta definida como $\varepsilon_f^1(g) = \varepsilon(g) \vee f$. Iterando la dilatación y la erosión geodésica hasta la estabilidad se construyen la reconstrucción R y la reconstrucción dual R^* respectivamente. Las definiciones matemáticas de ambas reconstrucciones se muestran en la Ecuación (2.5).

$$\begin{aligned}R(f, g) &= \lim_{n \rightarrow \infty} \delta_f^n(g) = \delta_f^1 \dots \delta_f^1 \delta_f^1(g) \\ R^*(f, g) &= \lim_{n \rightarrow \infty} \varepsilon_f^n(g) = \varepsilon_f^1 \dots \varepsilon_f^1 \varepsilon_f^1(g)\end{aligned}\tag{2.5}$$

El marcador g establece los objetos a ser preservados. Cuando $g = \varepsilon_{\lambda_B}(f)$ o $g = \delta_{\lambda_B}(f)$ se obtiene la apertura o la cerradura por reconstrucción como se muestra en la Ecuación (2.6).

$$\begin{aligned}\tilde{\gamma}_{\lambda_B} &= R(f, \varepsilon_{\lambda_B}(f)) \\ \tilde{\varphi}_{\lambda_B} &= R^*(f, \delta_{\lambda_B}(f))\end{aligned}\tag{2.6}$$

Su operación puede ser vista en la Figura 11. Ahí se especifica el marcador y la forma en que las operaciones geodésicas trabajan para lograr el proceso de filtrado.

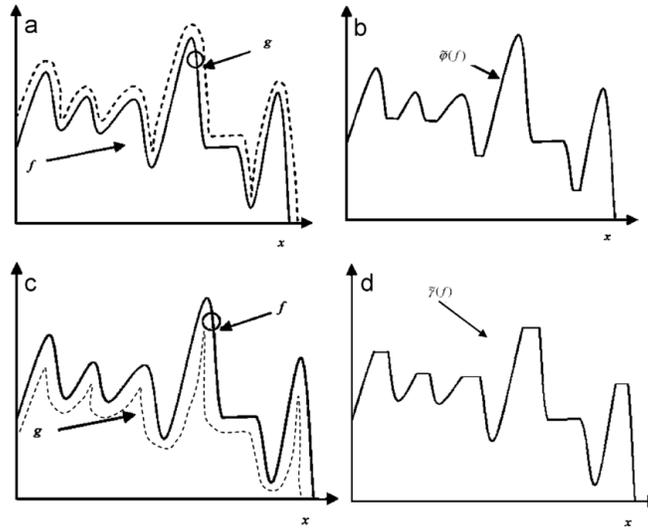


Figura 11. Ejemplo de la cerradura y apertura por reconstrucción.

2.3.5 Detectores morfológicos de contornos

En la morfología matemática hay tres operadores básicos para la detección de contornos. Dichos operadores toman ventaja de las propiedades de extensividad y de antiextensividad explicadas anteriormente. El gradiente interno se define como se muestra en la Ecuación (2.7).

$$IG = f - \varepsilon_{\lambda B}(f) \quad (2.7)$$

Donde f es la imagen y $\varepsilon_{\lambda B}$ es la erosión. Por su parte el gradiente externo se formula como se muestra en la Ecuación (2.8).

$$EG = \delta_{\lambda B}(f) - f \quad (2.8)$$

Donde f es la imagen y $\delta_{\lambda B}(f)$ es la dilatación. Finalmente, el gradiente morfológico se define como la diferencia de la dilatación y de la erosión de la imagen, su formulación se muestra en la Ecuación (2.9).

$$MG = \delta_{\lambda B}(f) - \varepsilon_{\lambda B}(f) \quad (2.9)$$

La Figura 12 muestra la acción de los tres operadores de gradiente.

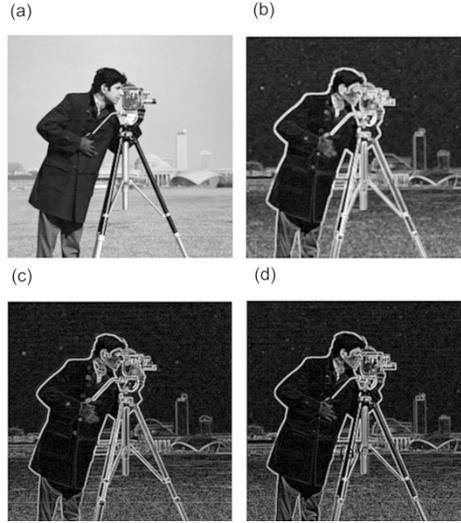


Figura 12. Ejemplo de la cerradura y apertura por reconstrucción.

2.4 Segmentación color usando histogramas bi-variables

El espacio de color *IHSL* propuesto por Angulo *et al.* (2005) proporciona una interesante herramienta para la realización de segmentación de imágenes color basada en morfología matemática. Las ecuaciones para transformar una imagen color *RGB* a el espacio *IHSL* se muestran en las Ecuaciones (2.10), (2.11) y (2.12). Donde r , g y b denotan los valores *RGB* de el píxel actual a ser transformado a *IHSL*, \min y \max son las funciones máximo y mínimo respectivamente.

$$H_{IHSL} = \begin{cases} \frac{g - b}{\max(r, g, b) - \min(r, g, b)}, & \text{if } r = \max(r, g, b) \\ \frac{g - b}{\max(r, g, b) - \min(r, g, b)} + 2, & \text{if } g = \max(r, g, b) \\ \frac{g - b}{\max(r, g, b) - \min(r, g, b)} + 4, & \text{if } b = \max(r, g, b) \end{cases} \quad (2.10)$$

$$L_{IHSL} = 0.213r + 0.715g + 0.072b \quad (2.11)$$

$$S_{IHSL} = \max(r, g, b) - \min(r, g, b) \quad (2.12)$$

En la Figura 13 se muestra la conversión de una imagen *RGB* al espacio *IHSL*.

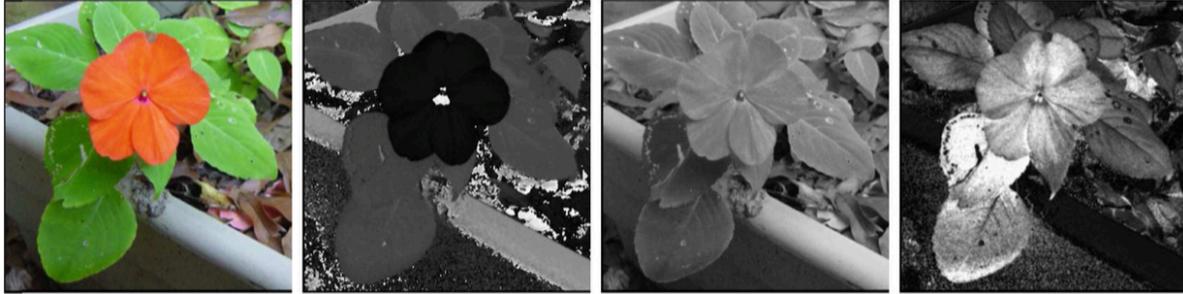


Figura 13. Conversión a espacio de color *IHSL*.

Con la imagen transformada a el espacio *IHSL*, dos histograma bi-variables pueden ser definidos, el primero en función del tono y la saturación (*HS*), con una representación en coordenadas polares (*HS*), y el segundo en función de la luminancia y la saturación (*LS*) con una representación en coordenadas rectangulares (*LS*). Una representación logarítmica para expresar los histogramas en una imagen tamaño 256x256 en 8 bits se calcula usando las Ecuaciones (2.13) y (2.14).

$$f_{LS}(X) = 255 * \frac{\log(hist_{LS}(x, y))}{\max(\log(hist_{LS}(x, y)))} \quad (2.13)$$

$$f_{HS}(X) = 255 * \frac{\log(hist_{HS}^{\rho}(x, y))}{\max(\log(hist_{HS}^{\rho}(x, y)))} \quad (2.14)$$

Donde $X = (x, y)$, $x = f_L \in [0, 255]$, $y = f_S \in [0, 255]$ y $hist_{LS}$ es el número de ocurrencias del par (x, y) para la primera ecuación, mientras que, para la segunda, $X = (x, y)$, $x = f_H \in [0, 360]$, $y = f_S \in [0, 255]$ y $hist_{HS}^{\rho}$ y es el numero de ocurrencias del par (x, y) . A partir de estos dos histogramas es posible evaluar las características de luminancia, saturación y tono para la segmentación de los histogramas. Dado que dichos histogramas son un compendio de la información color es posible, de hecho, reducir la cantidad de información aplicando técnicas de

filtrado a los histogramas bi-variables y agruparlos en regiones, para finalmente aplicar, el proceso inverso para regresar, esta vez a un par de imágenes *RGB*, una con la información cromática y otra con la información acromática modificada en ambos casos. El proceso de filtrado puede ser realizado aplicando filtros morfológicos. El proceso completo de segmentación color se puede seguir en la Figura 14.

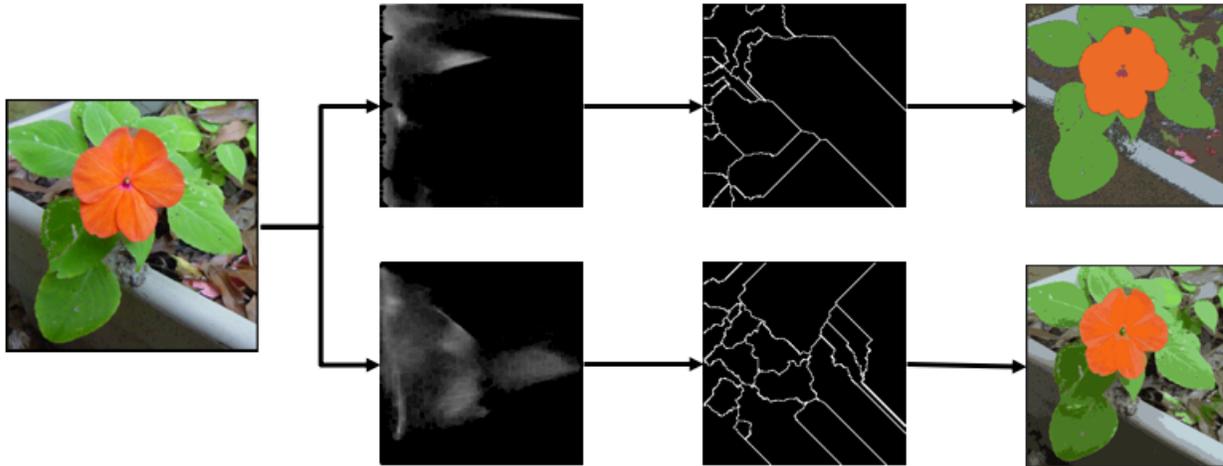


Figura 14. Ejemplo de segmentación color usando los histogramas bi-variables.

2.5 Búsqueda de plantillas

El concepto de comparar imágenes y determinar una medida que cuantifique dicha tarea es interesante y todo un reto. La búsqueda de plantillas consiste en comparar un patrón R (plantilla) con una imagen grande I (imagen de búsqueda). La plantilla R es desplazada sobre la imagen de búsqueda I y en cada posición (u, v) la diferencia entre ella y la sub-imagen es medida buscando por la mayor similitud. La distancia de desplazamiento se denota por (r, s) y se usa para indicar el desplazamiento de una plantilla R sobre la imagen de búsqueda I . A partir de estas ecuaciones es posible encontrar la posición con el mayor parecido entre la referencia R y la imagen de búsqueda I , esto minimizando la diferencia de cuadrados d_E , resultando en $d_E^2(r, s) = \sum_{(i,j) \in R} (I(r + i, s + j) - R(i, j))^2$. Desarrollando la ecuación anterior, y despreciando algunos términos, nos permite obtener un término similar a la correlación lineal cruzada (\odot) entre I y R , la cual es mostrada en la Ecuación (2.15).

$$(I \otimes R)(r, s) = \sum_{(i,j) \in R} |I(r+i, s+j) * R(i, j)| \quad (2.15)$$

El hecho de rechazar algunos términos del cuadrado de d_E , ocasiona algunos problemas en el resultado de la correlación cruzada, dado que varía con la intensidad encontrada en la imagen I . La correlación cruzada normalizada compensa esto tomando en cuenta la energía en la imagen de referencia y la sub-imagen actual. La formulación de la correlación cruzada normalizada (C_N) se muestra en la Ecuación (2.16).

$$C_N(r, s) = \frac{\sum_{(i,j) \in R} I(r+i, s+j) * R(i, j)}{|\sum_{(i,j) \in R} I^2(r+i, s+j) * R^2(i, j)|^{1/2}} \quad (2.16)$$

El resultado de C_N varía en el rango $[0,1]$, donde 1 indica la mejor similitud y 0 denota nulo parecido. Un ejemplo conceptual del proceso de búsqueda de plantillas puede ser visto en la Figura 15.

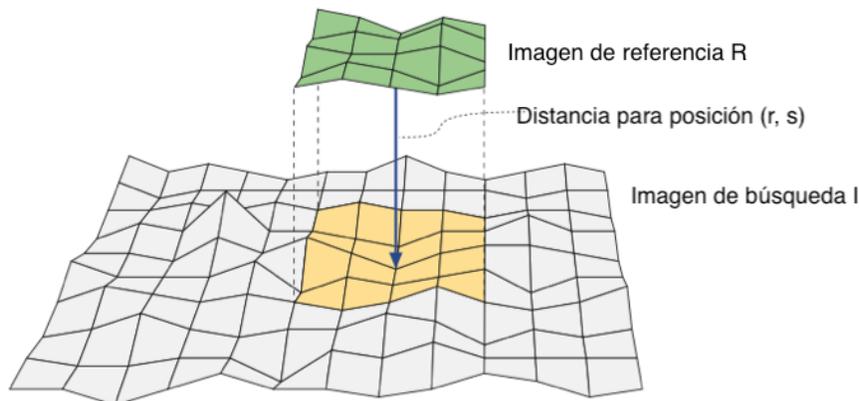


Figura 15. Ejemplo de segmentación color usando los histogramas bi-variables.

2.6 Python

Python es un lenguaje de súper alto nivel desarrollado por Guido Von Rossum en los años 80's. Es un lenguaje interpretado, lo que significa que no requiere un proceso de compilación para la generación de un código objeto, sino más bien de un intérprete que lee el código fuente y

ejecuta línea por línea el programa de forma secuencial. Al ser un lenguaje de súper alto nivel, como Ruby, su sintaxis es muy parecida al lenguaje humano, además, a diferencia de C/C++, su sintaxis muy simple permitiendo escribir una aplicación con pocas líneas de código y con la misma funcionalidad que la de cualquier lenguaje compilado, además de que los tipos de las variables son implícitos. Por el contrario, al ser un lenguaje interpretado los tiempos de ejecución dependen del hardware y de la estructura de programación utilizada. A nivel científico, Python es un lenguaje utilizado en la NASA y en los grandes centros de investigación como el CERN y el ESRF. A nivel industrial, Google, Apple y Facebook lo utilizan en su día a día gracias a la gran cantidad de *frameworks* desarrollados para él. Entre los más populares están: Django, para aplicaciones web, SciPy y Numpy, para matemáticas y álgebra científica, wxPython y Tinker, para desarrollo de GUI's, y OpenCV y PIL (*Python Imaging Library*) para el procesamiento de imágenes.

2.7 OpenCV

OpenCV (por sus siglas en inglés *Open Computer Vision*) es una librería desarrollada por Intel para el procesamiento digital de imágenes de forma matricial bajo el concepto de *Open Source*. Esta implementada en C/C++ y fue adaptada para los lenguajes de programación Java y Python. Incluso existe una adaptación para desarrollos en Android, así como una versión hermana llamada EmguCV desarrollada para C#. Cuenta con un conjunto de herramientas de lectura y escritura de imágenes, manejando múltiples formatos, así como con una GUI (*Graphical User Interface*) para visualización y desarrollo de interfaces simples. OpenCV cuenta además con filtros lineales y no lineales implementados para el procesamiento y análisis de imágenes, incluyendo morfología matemática. En la industria esta librería es un estándar para el procesamiento de imágenes con la gran ventaja de que está en constante desarrollo y hoy en día incluye algoritmos de procesamiento basados en *machine learning*. Además, la implementación de nuevos algoritmos y metodologías es muy sencillo usando su cualquier lenguaje de programación.

Capítulo III

3. Metodologías de filtrado de mapas de profundidad

En este capítulo se presentan las metodologías desarrolladas durante este trabajo de investigación. Se presentan de manera general, una metodología basada de filtrado que trabaja directamente sobre los mapas de profundidad sin requerir de información adicional. Se hace el planteamiento de la metodología, así como los algoritmos y pasos propuestos. A continuación, como segunda metodología propuesta se presenta una herramienta de filtrado que combina mapas de profundidad con una imagen color emparejada, de igual forma se hace la presentación de la metodología y se muestran los algoritmos y pasos utilizados. Adicionalmente se presentan dos metodologías extras, la primera está enfocada a cuantificar fallas en aspas de aerogeneradores utilizando procesamiento de imágenes; la segunda se centra en la aplicación de un sistema para detectar desbalances térmicos en imágenes termográficas de mammas originalmente propuesto por Sofía Giovanna Vega Mancilla. Las metodologías presentadas fueron programadas en Python usando librerías libres disponibles en la red. Cuando así se requirió dichas metodologías se portaron a C++, o Java para aumentar su velocidad y mejorar su desempeño o para utilizarlos en dispositivos móviles. De manera general todos los algoritmos usados fueron programados desde cero, sin embargo, para ciertas excepciones se utilizó la implementación que se encontraba en OpenCV. Dentro del proceso robótico las metodologías propuestas se ubican en la región marcada en la Figura 16.

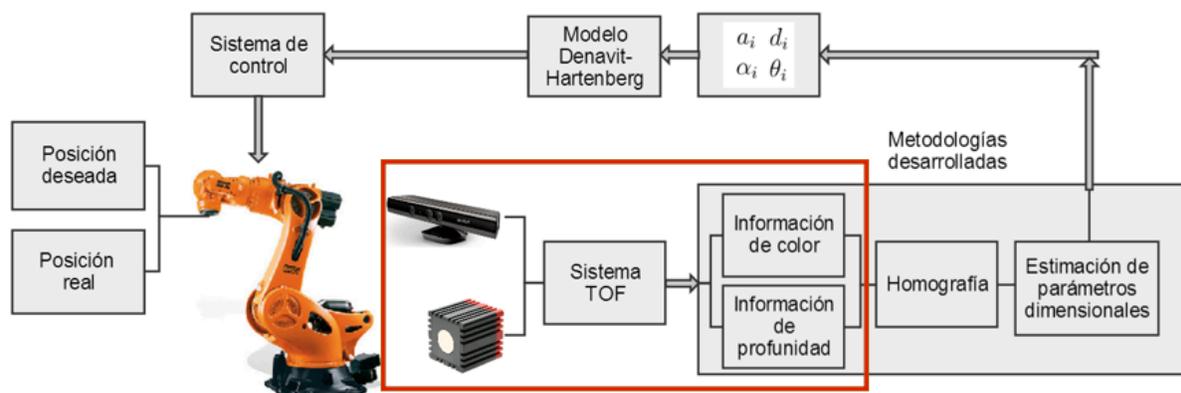


Figura 16. Proceso general de procesamiento de imágenes utilizado en esta tesis.

3.1 Metodología de filtrado usando sólo mapas de profundidad

Los sensores que permiten obtener un mapa de profundidad generalmente solo cuentan con un sensor y un emisor IR. Eso reduce la cantidad de información a procesar y limita el problema del filtrado del mapa. Teniendo esto en cuenta y las capacidades de las reconstrucciones morfológicas se propuso la metodología mostrada en la Figura 17. Dicha metodología presenta una novedosa modificación al algoritmo original de cerradura por reconstrucción.

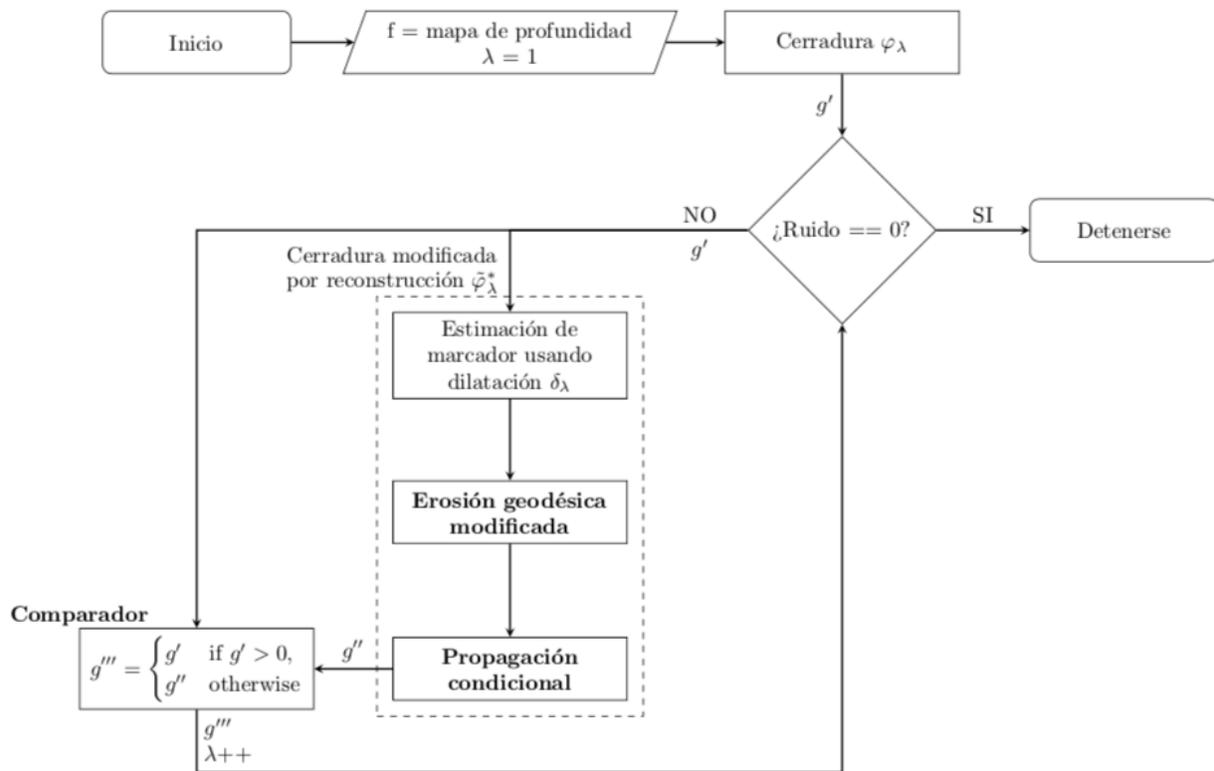


Figura 17. Metodología para filtrado de mapas de profundidad.

El proceso inicia con un mapa de profundidad sencillo f , en escala de grises y de un canal simple. La metodología consiste en un proceso iterativo que se repite en función del tamaño de los huecos encontrados. Inicialmente se propone un tamaño de $\lambda = 1$, y como paso inicial se aplica una cerradura de tamaño φ_1 , este pre-procesamiento busca eliminar aquellos huecos muy pequeños que pudiera aparecer en el mapa de profundidad resultando en el mapa de profundidad de salida g' . A continuación, se hace una evaluación del mapa de profundidad contando el total de pixeles negros remanentes en el mapa de profundidad g' . Esta cuenta de pixeles negros es la variable de

control para el proceso de filtrado iterativo. Si la cuenta de pixeles negros es cero, entonces el proceso termina dado que no haya nada más que hacer. Por el contrario, si la cuenta de pixeles es diferente de cero, el proceso iterativo comienza. El mapa de profundidad filtrado entra a la modificación de la cerradura por reconstrucción propuesta la cual consta de tres pasos, el primero es una dilatación del mapa de profundidad tamaño δ_1 , esta dilatación permite estimar el marcador a usar en la reconstrucción modificada $\tilde{\varphi}_1^*$, el segundo paso es llevar a cabo la erosión geodésica modificada ε_f^{1*} , y el tercer paso consiste en aplicar una propagación condicional κ . Respecto a la erosión geodésica modificada está ligada a la propagación condicional, la acción de ambas es mostrada en la Ecuación (3.1).

$$\varepsilon_f^{1*}(g)(x) = \kappa\{f(x), \varepsilon(g)(x)\} \quad (3.1)$$

La propagación condicional κ busca que el resultado de la erosión geodésica sea el valor máximo entre los mínimos de los vecinos del marcador $(g)(x)$ y el píxel actual de la imagen $f(x)$, pero que cumpla con ser diferente de 0 o 255. Si dicha condición no se cumple, el valor propagado debe ser cero. Esto puede ser visto claramente en la Ecuación (3.2).

$$\begin{aligned} &\varepsilon_f^{1*}(g)(x) \\ &= \begin{cases} \max\{f(x), (g)(x)\} & \text{if } 0 < \max\{f(x), (g)(x)\} < 255 \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (3.2)$$

La erosión geodésica original permite la recuperación solo de los huecos marcados, de tal modo que si la dilatación usada como marcador para llenarlos es demasiado grande se perderán pequeños detalles. Debido a esto, la Ecuación (3.3) mostrada a continuación es propuesta para preservar sin cambio todos los detalles originales que están presentes en el mapa original diferentes de cero, de tal forma que solo la información recuperada es agregada en las subsecuentes iteraciones.

$$g''' = \begin{cases} g' & \text{if } g' > 0 \\ g'' & \text{otherwise} \end{cases} \quad (3.3)$$

Donde g' es el mapa de profundidad resultante del ultimo paso de filtrado, g'' es el mapa de profundidad g' después de la aplicación de la cerradura por reconstrucción modificada y g''' es el mapa de profundidad filtrado resultante. Después de aplicar este proceso, una cuenta del ruido que queda en el mapa de profundidad es llevada a cabo sobre el mapa de profundidad filtrado g''' , y el tamaño del filtro λ es incrementado en uno. Si la cuenta de ruido sigue siendo diferente de cero se hace una dilatación para calcular un nuevo marcador y el método es aplicado de nuevo. El proceso se repite hasta que la cuenta de ruido sea igual a cero, lo cual significa que el ruido y huecos han sido completamente eliminados. Para ejemplificar esto en la Figura 18 se utiliza un mapa de profundidad sintético, con una cuenta inicial de ruido de 56 píxeles. Dicho mapa es procesado de forma paralela usando el algoritmo original de la cerradura por reconstrucción y el algoritmo propuesto. Se observa el efecto de propagación desde dentro del hueco y como la condición de propagación logra mejorar el resultado obtenido.

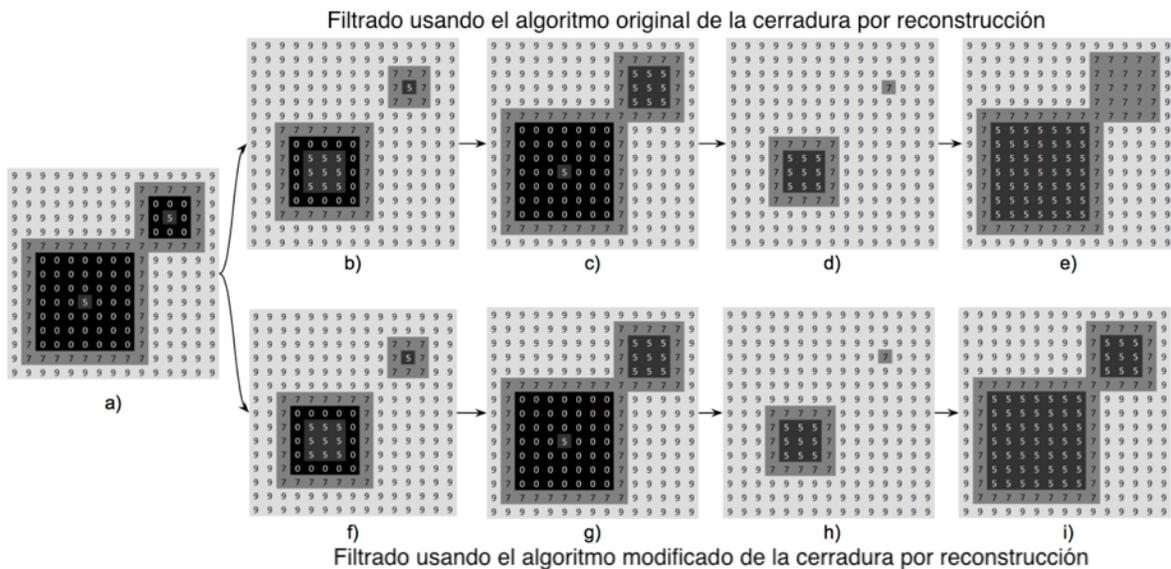


Figura 18. Comparativa de filtrado iterativo de un mapa de profundidad sintético usando el algoritmo original de la cerradura por reconstrucción y el algoritmo modificado propuesto.

La modificación propuesta el algoritmo original de la cerradura por reconstrucción se muestra en el Algoritmo (1) a manera de pseudocódigo. La función propuesta recibe como argumentos la imagen original I y el marcador J . Dicho muestra la inclusión de la función de propagación condicional (κ) aplicada a los argumentos originales.

```

1: función CERRADURA POR RECONSTRUCCIÓN
MODIFICADA  $\tilde{\varphi} * (I, J)$ 
2:    $aux = J$ 
3:    $bandera = Verdad$ 
4:   mientras  $bandera$  hacer
5:     para cada pixel  $p$  en  $I$  en orden normal hacer
6:        $aux = \min(J(q), q \in N_G^+ \cup \{p\})$ 
7:        $J(p) = \kappa(aux, I(P))$ 
8:     para cada pixel  $p$  en  $I$  en orden inverso hacer
9:        $aux = \min(J(q), q \in N_G^- \cup \{p\})$ 
10:       $J(p) = \kappa(aux, I(P))$ 
11:       $diff = J - aux$ 
12:      si  $\max(diff) == 0$  entonces
13:         $bandera = Falso$ 
14:      regresa  $J$ 

```

Algoritmo 1 Cerradura por reconstrucción modificada propuesta $\tilde{\varphi}^*$.

Uno de los principales sustentos del método propuesto es la necesidad de propagar los valores de gris cercanos a los huecos en una forma no abusiva, y sólo propagar valores diferentes de 0 y 255, de tal modo que se tiene una función que busque dichos para lograr esto, el pseudocódigo de la función de propagación κ se muestra en Algoritmo (2).

```

1: función PROPAGACIÓN CONDICIONAL  $\kappa(lista)$ 
2:    $propagado = 0$ 
3:   para cada elemento  $x$  en  $lista$  hacer
4:     si  $x > propagado$  y  $propagado < 255$  entonces
5:        $propagado = x$ 
6:   regresa  $propagado$ 

```

Algoritmo 2 Función condicional de propagación propuesta κ .

Finalmente, la función de comparación que evita la modificación de la información de profundidad original y que solo preserve la profundidad estimada de los huecos de interés es presentada en forma de pseudocódigo en el Algoritmo (3).

```

1: función COMPARADOR ( $g', g''$ )
2:   para cada pixel  $p$  en  $g'$  hacer

```

```

3:      si  $g'(p) > 0$  entonces
4:           $g'''(p) = g'(p)$ 
5:      además si
6:           $g'''(p) = g''(p)$ 
7:      regresa  $g'''$ 

```

Algoritmo 3 Función de comparador propuesto.

3.2 Metodología de filtrado combinando mapas de profundidad con imágenes color

De una forma similar existen en el mercado sensores que hacen estimaciones de profundidad que cuentan con un emisor receptor infrarrojo y con una cámara *RGB* emparejada para obtener no solo información de distancias sino de color de la escena observada. En dado caso se tiene una mayor cantidad de datos que procesar lo que abre las posibilidades a generar metodologías más completas. A partir de esto se propone utilizar como base la metodología presentada en el punto anterior y presentar una metodología basada en características de forma, color y vecindad para filtrado de mapas de profundidad. Dicha metodología se muestra en la Figura 19. Al inicio de la metodología se tienen dos imágenes emparejadas de una cierta escena, la primera corresponde a una imagen *RGB* codificada en 24 bits, 8 bits por canal, mientras que la segunda corresponde a un mapa de profundidad almacenada en 8 bits en escala de grises. Inicialmente, al mapa de profundidad se le aplica una cerradura φ_1 y entonces el ruido es removido utilizando la metodología presentada en el punto anterior. A continuación se propone un algoritmo de evaluación de ruido que busca determinar si las regiones de huecos fueron rellenadas utilizando la información al interior del hueco, si es que la hubiese o si fueron cerradas usando la información de los alrededores del hueco, en el segundo caso, se obtienen una serie de regiones $R(u, v)$ que posteriormente serán utilizadas en el proceso de búsqueda de plantillas.

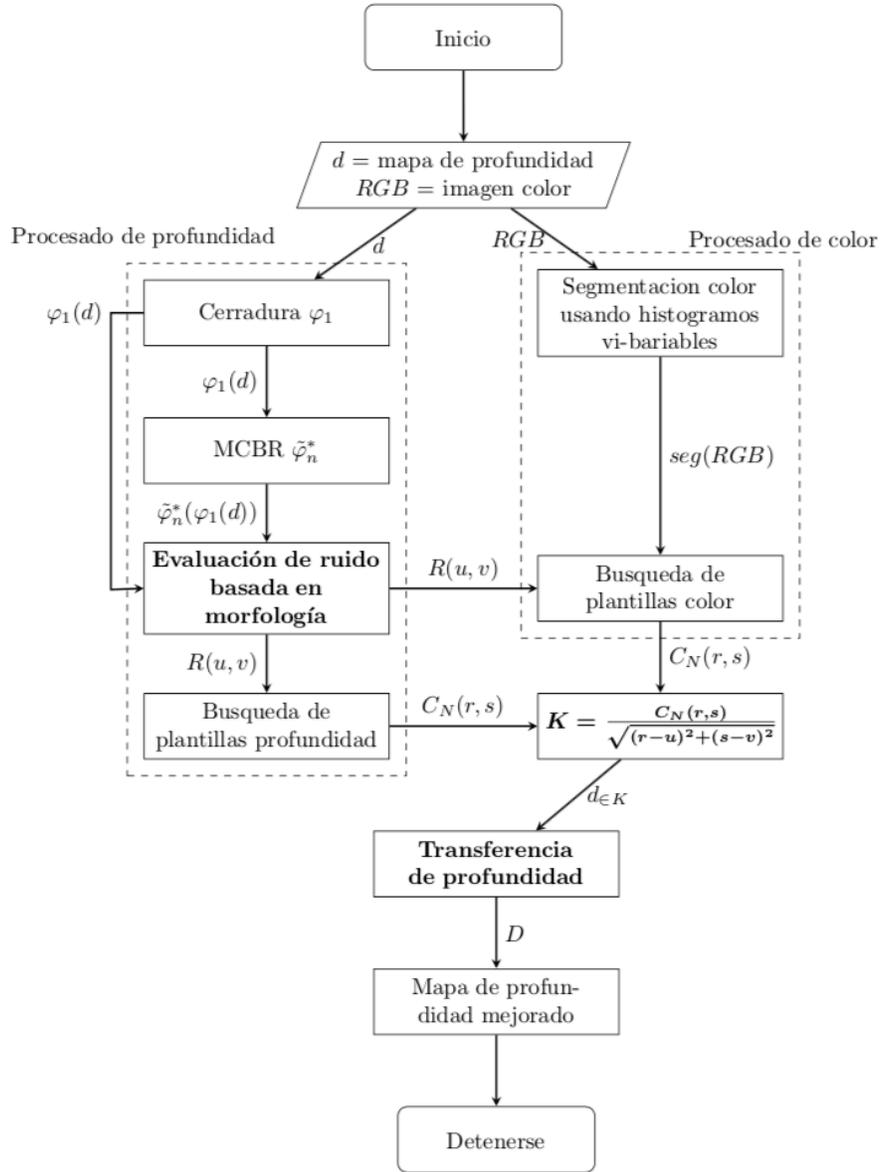


Figura 19. Metodología para filtrado de mapas de profundidad color.

El proceso de evaluación de ruido basado en morfología es mostrado en la Figura 20. Básicamente, la cerradura φ_1 y la cerradura φ_n^* son las dos entradas para dicho proceso, cada hueco encontrado es etiquetado, y para cada hueco se calcula el gradiente interno y el gradiente externo para obtener sus coordenadas (i, j) de los n número de píxeles que forman parte de esas líneas de gradiente. Entonces, se hace una estimación de la media de los niveles gris de ambas líneas de gradiente, pero considerando los píxeles correspondientes al mapa de profundidad filtrado φ_n^* . Finalmente, la diferencia de valores de media es calculada de tal forma que, si dicha diferencia es mayor que 1, implica que el hueco fue llenado con información diferente a la que rodea al hueco.

Una diferencia igual o menor que uno, por lo tanto, indica que dicho hueco necesita un tratamiento adicional para lograr un llenado más adecuado.

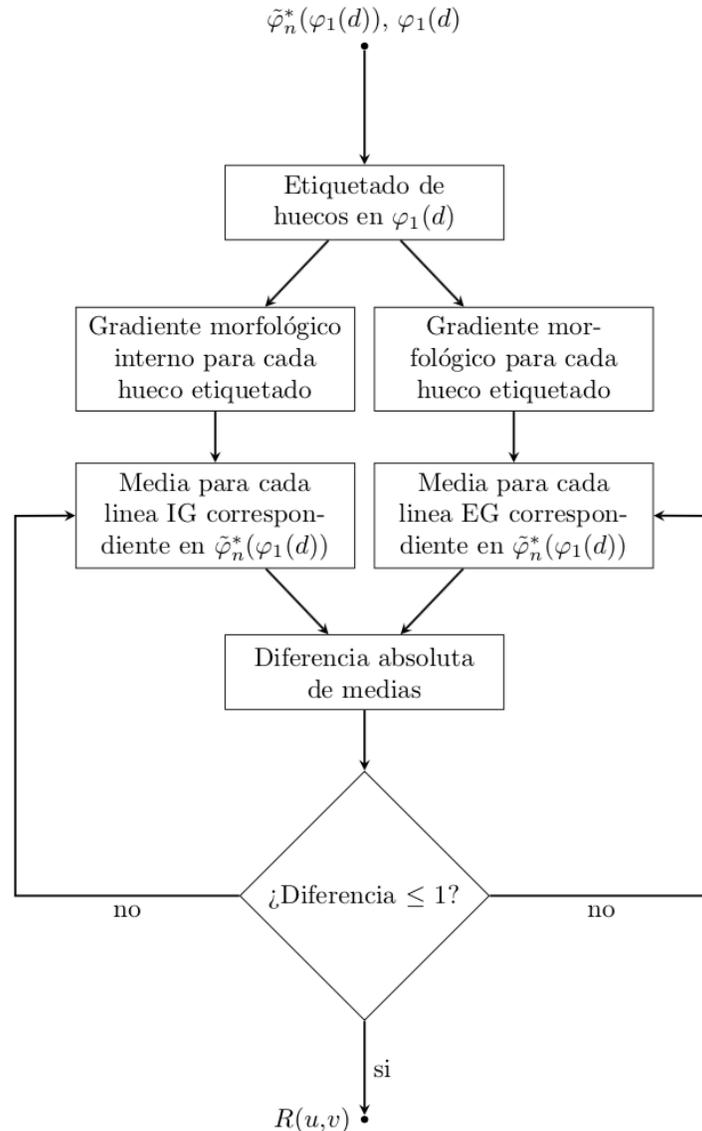


Figura 20. Proceso de clasificación de huecos.

A continuación, se propuso un criterio K para determinar las regiones más parecidas en profundidad y color, dicho criterio toma en cuenta las regiones con la correlación cruzada normalizada más alta tanto en profundidad en el mapa de profundidad como en la imagen RGB, pero también tomando en cuenta la información de vecindad a través de la distancia Euclidiana entre el punto central de dichas regiones con respecto al centroide del hueco original a rellenar. Esto puede ser visto en la Ecuación (3.4).

$$K = \frac{C_N(r, s)}{\sqrt{(r - u)^2 + (s - v)^2}} \quad (3.4)$$

Usando dicho criterio, la región que logre la mejor correlación, tanto en color como en forma y que se ubique más cerca de una región específica con profundidad incorrecta puede ser clasificada como el mejor candidato para rellenar la región con información de profundidad faltante y para proveer un filtrado más adecuado. Finalmente, si la región que logra el mejor match fue clasificada correctamente usando el método de *MCbR*, se realiza una transferencia de profundidad, considerando el promedio de las regiones correspondiente con un nivel de cero de la plantilla del hueco a ser rellenado con la correspondiente región con la mejor correlación (en color o en profundidad) a partir de la imagen filtrada con el método *MCbR*. El proceso completo propuesto puede ser visto gráficamente en la Figura 21.

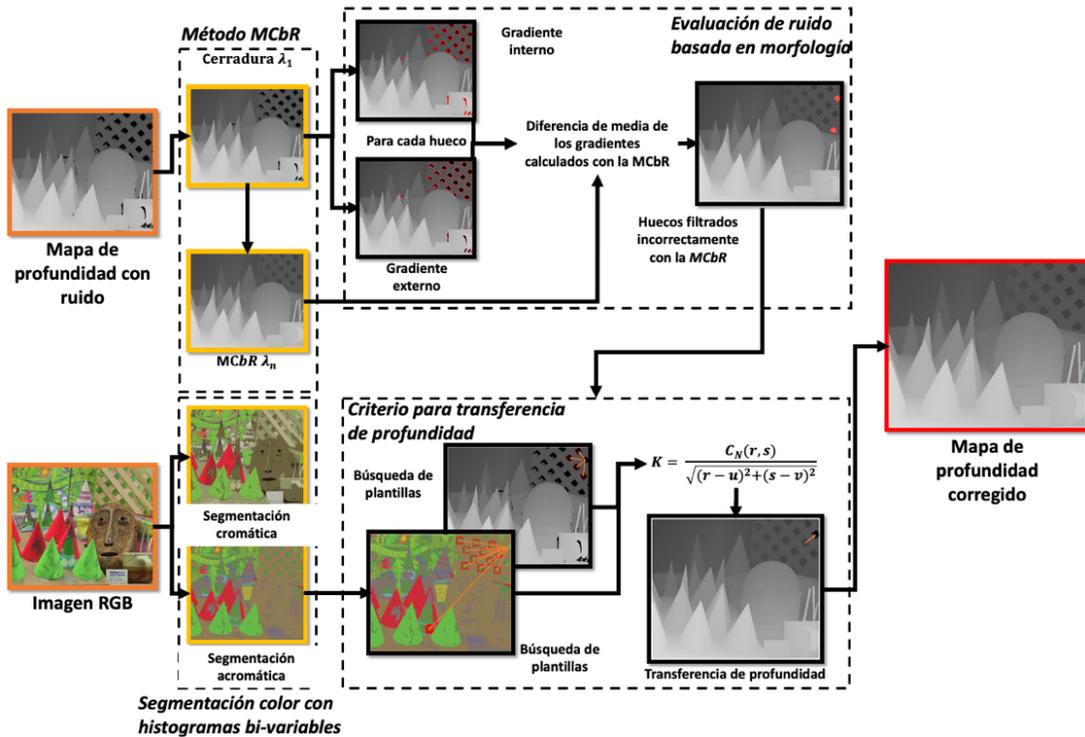


Figura 21. Proceso completo de filtrado de mapas de profundidad basando en características de color, forma y vecindad.

3.3 Metodologías adicionales desarrolladas durante este trabajo de tesis

En esta sección se muestran una serie de metodologías adicionales desarrolladas durante este trabajo de doctorado que no se relacionan directamente con el tema de filtrado de mapas de profundidad pero que están en el campo de acción del procesamiento de imágenes.

3.3.1 Metodología para cuantificar fallas en aspas de aerogeneradores

De forma adicional se trabajó en la parte de procesamiento de imágenes relacionado con proyectos en conjunto con la industria. Para eso se planteó la siguiente metodología de trabajo mostrada en la Figura 22.

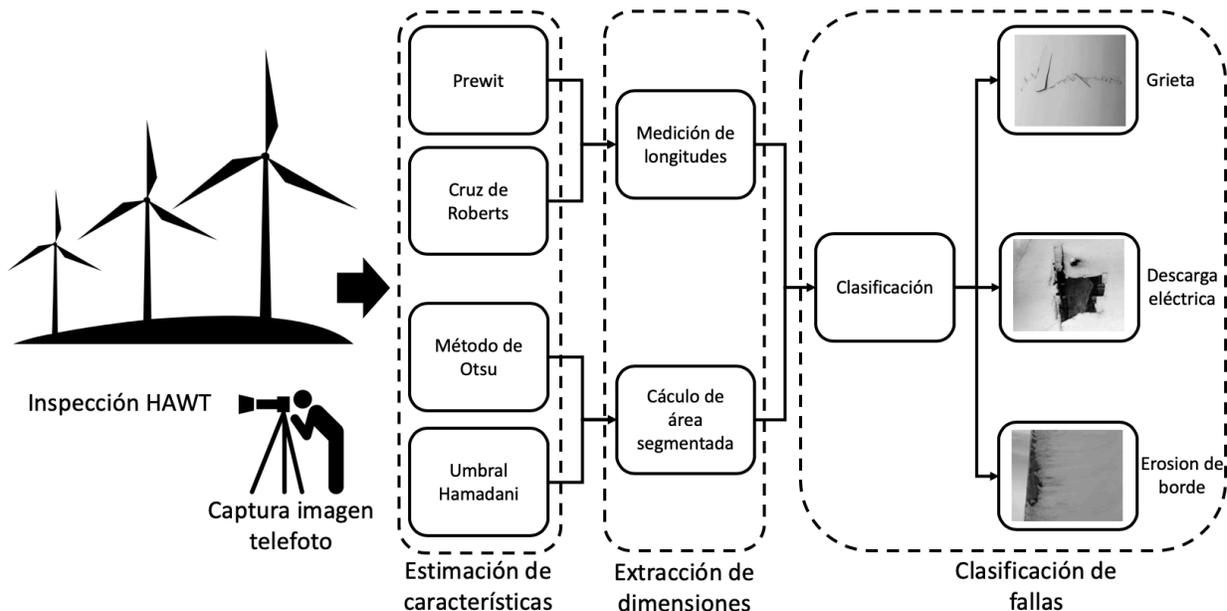


Figura 22. Metodología propuesta para la clasificación de daños en aspas de turbina eólica tipo de eje horizontal (HAWT).

Dicha metodología inicia con la propuesta de una inspección de aspas de aerogeneradores tipo eje horizontal a nivel de tierra. Esta inspección está basada en el uso de cámara réflex montada sobre un trípode. La cámara réflex tiene las siguientes especificaciones: Nikon D5300, sensor CMOS, 24.4 megapíxeles de imagen, conexión Wi-Fi integrada y GPS; mientras que el lente telefoto montado es un SIGMA 70-200 mm f/2.8. La adquisición de las imágenes es llevada a cabo para cada una de las aspas siguiendo sus numeraciones (1, 2 y 3) como este impreso en ellas. Después de tomar las imágenes, se transfieren vía inalámbrica a un PC que contiene el sistema de

análisis automático instalado. Dado que cada imagen esta geo referenciada, dicho parámetro permite al usuario organizar cada set de imágenes por turbina eólica. A continuación, una selección manual de. Un área de interés es llevada a cabo de forma manual por el usuario para centrarse directamente en los problemas específicos detectados en cada imagen. Es posible seleccionar múltiples regiones de interés dado el caso de que se detecten varios problemas. Finalmente, el análisis inicia: la imagen recortada se procesa primero para realizar una estimación de características, dichas características son remarcadas usando algoritmos de umbralizado automáticos y detectores de contornos; cada imagen es procesada con cuatro algoritmos para asegurarse que, si cierto algoritmo pierde detalles, otro pueda proveer soporte al proceso. Una vez que las características han sido mejoradas, se miden en un paso subsecuente, se mide longitud y área en píxeles. A partir de ahí el proceso de clasificación da inicio para determinar el tipo de falla registrada, grieta, descarga eléctrica o erosión del borde de ataque del aspa. El usuario recibe como salida una imagen procesada con la falla segmentada y el resultado de la clasificación. La clasificación de tipo de falla se basa en las características mostradas en la Tabla 3.1.

Tabla 3.1 Características analizadas cuantitativamente.

Tipo de falla	Evaluación cuantitativa
Grieta	- Líneas delgadas - Áreas segmentadas pequeñas - Líneas de segmentación largas
Descarga eléctrica	- Un objeto grande - Altura y ancho proporcionales
Erosión del borde de ataque	- Un objeto grande - Más largo que ancho

3.3.2 Metodología para detección de desbalances térmicos en imágenes de mamas

Aquí se presenta la metodología reportada que fue inicialmente propuesta por Sofia Giovanna Vega Mancilla. Primero se presenta el protocolo de captura de termogramas propuesto, a continuación, se muestra la etapa de pre-procesamiento para las imágenes cuyo propósito es facilitar los pasos subsecuentes y eliminar regiones innecesarias. Entonces se muestra el proceso de segmentación automático donde cada mama es separada del resto del cuerpo de la paciente. En este punto se realiza un análisis estadístico para evaluar las mamas segmentadas y buscar una

diferencia en la simetría de la temperatura. Finalmente, en la etapa de evaluación se detalla cómo es que se establece si una paciente está en riesgo de salud o se encuentra normal de acuerdo con los estándares del estado del arte, buscando las regiones de mayor temperatura de acuerdo con las mediciones realizadas por el sensor infrarrojo para establecer la sospecha de la presencia de algún problema. El flujo de trabajo de la metodología puede ser visto en la Figura 23.

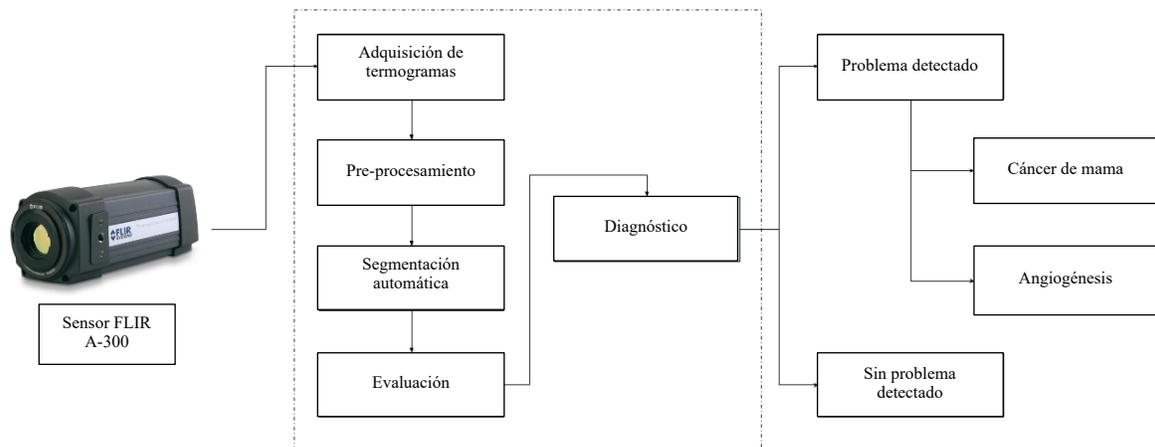


Figura 23. Metodología propuesta para la detección de desbalances térmicos en imágenes de mamas para detección oportuna de cáncer de mama.

Capítulo IV

4. Resultados y discusiones

En este capítulo se detallan los resultados obtenidos a partir de las metodologías presentadas en el capítulo anterior. Dichas metodologías fueron evaluadas de forma cualitativa y cuantitativa según fuera posible. Para las metodologías de filtrado de mapas de profundidad se utilizaron imágenes de bases de datos como los son Middlebury y Tsukuba. Estas metodologías fueron evaluadas de forma cualitativa en base a la evaluación visual de los resultados provistos; por su parte la evaluación cuantitativa fue realizada utilizando descriptores estadísticos que se encuentran comúnmente en el estado del arte como lo son el PSNR (*Peak Signal Noise to Ratio*), MSE (*Mean Squared Error*) y SSIM (*Structural Similarity*). Para el caso de la metodología de caracterización de fallas en aerogeneradores se realizó una evaluación cuantitativa usando imágenes de referencia tomadas en parques eólicos de Oaxaca. Se contrastó el resultado de la clasificación contra el tipo de falla real. Finalmente, para los termogramas se realizó una comparación de los resultados encontrados utilizando la *Mastology Database* de un grupo de investigación de Brasil con previo diagnóstico para evaluar la sensibilidad del método propuesto.

4.1 Filtrado de mapas de profundidad usando la cerradura por reconstrucción modificada MCbR

Para evaluar la metodología propuesta se utilizó el ser de imágenes de Middlebury y Taukuba. Para el caso del primer set se utilizaron las imágenes de Cones y Teddy, en dichas pruebas se evaluó el comportamiento del algoritmo propuesto, la *MCbR* contra los algoritmos del estado del arte relacionados con la morfología matemática y contra la mediana estadística. Cualitativamente el método propuesto se desempeña de forma correcta evitando una alta modificación de bordes y objetos en los mapas de profundidad. Para el caso de la imagen Cones, Figura 24, hay regiones de interés que merecen ser resaltadas como lo son los lados rectos de los conos y el enrejado de la parte posterior. Además, del lado derecho hay una taza con lápices que es una región compleja dado la geometría de los objetos. Todos los filtros logran eliminar los huecos, pero a un precio muy alto distorsionando por completo la imagen lo cual desde un punto

de vista de la robótica es inaceptable al introducir incertidumbre incensaría. La metodología propuesta logra superar visualmente al algoritmo original de la cerradura por reconstrucción, logrando recuperar detalles que de otra forma se pierden dada la naturaleza del método primario.

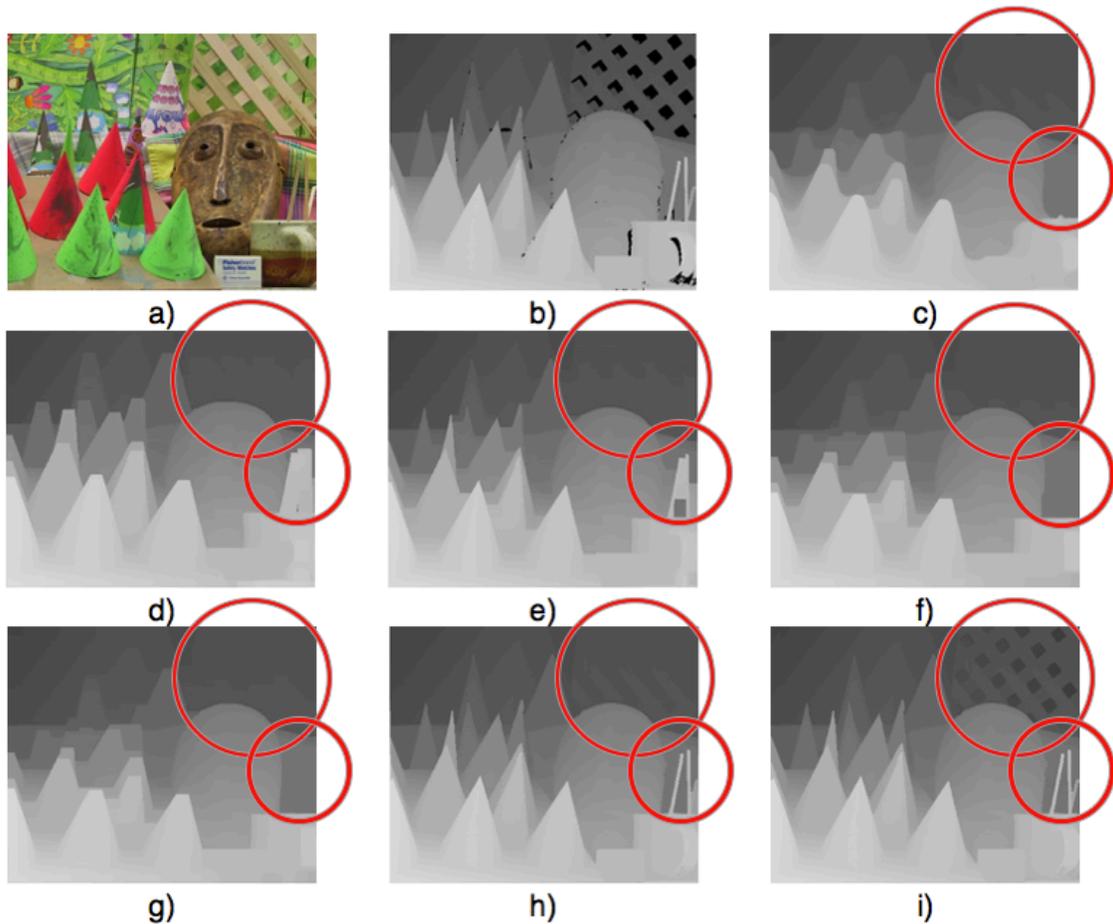


Figura 24. Proceso de filtrado de conos-disp2, a) imagen color, b) mapa de profundidad original, c) mediana 31x31, d) dilatación 8, e) cerradura 8, f) ASF W 9, g) ASF B 9, h) cerradura por reconstrucción 8 e i) método propuesto 8.

En el caso de la imagen Teddy, se presenta una situación muy peculiar. Si bien es una imagen compleja, hay una zona que llama mucho la atención. En la parte inferior del muñeco de peluche hay un hueco que permite visualizar hasta la pared del fondo. Esto es claramente visible en la imagen a color. En los mapas de profundidad se observa como dicha región es rellenada al nivel del brazo del muñeco lo cual puede ser una solución adecuada, sin embargo, el método propuesto logra detectar pequeñas componentes de profundidad que corresponden precisamente a la pared del fondo. Una vista a detalle permite verificar esta situación con lo que el recto de propagación de información de profundidad se cumple de forma muy eficiente. Esto es

especialmente útil si se requiere detectar detalles finos de la escena observada. De forma similar, los demás algoritmos logran rellenar todos los huecos, pero con una alta modificación de las regiones, Figura 25.

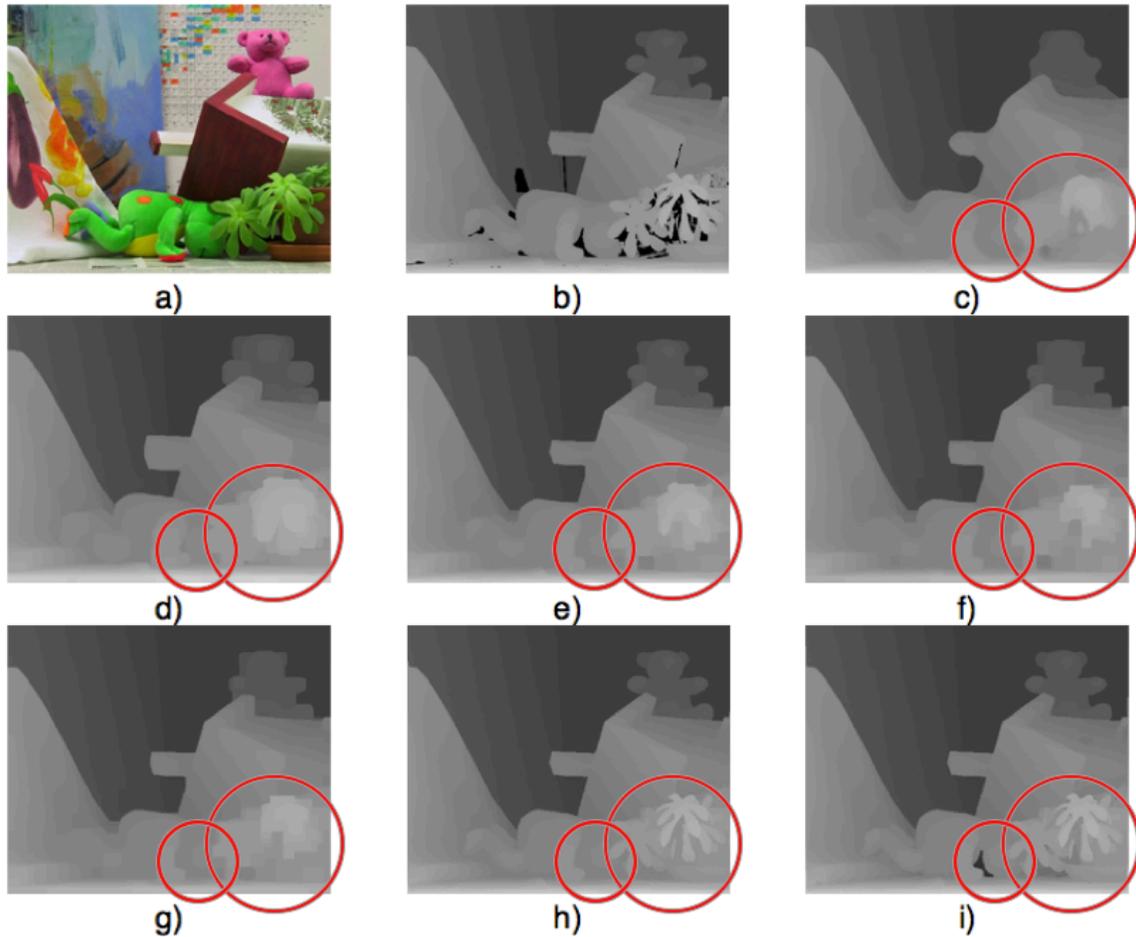


Figura 25. Proceso de filtrado de teddy-disp2, a) imagen color, b) mapa de profundidad original, c) mediana 29x29, d) dilatación 7, e) cerradura 7, f) ASF W 7, g) ASF B 7, h) cerradura por reconstrucción 7 e i) método propuesto 7.

Un análisis del comportamiento del filtro se observa que básicamente requiere de una iteración adicional en todos los casos para lograr llenar por completo los huecos de los mapas de profundidad, Figura 26 y 27. Este paso adicional, de hecho, corresponde a la cerradura tamaño 1 aplicada al inicio de la metodología. Con eso en mente se tiene un algoritmo que computacionalmente no requiere de mayores pasos y que presenta una alta preservación de bordes.

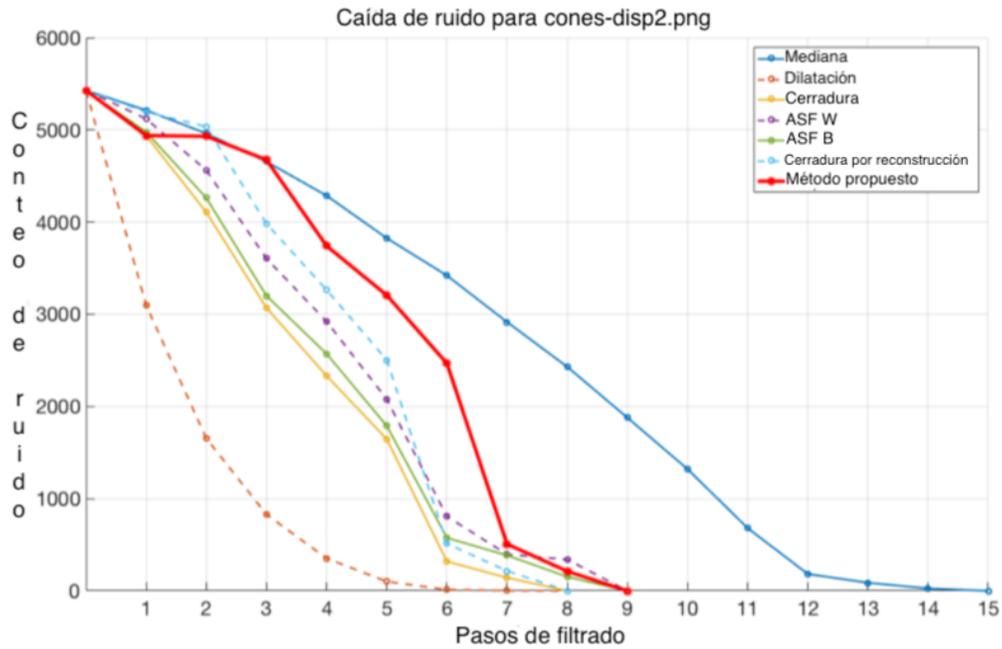


Figura 26. Pasos de filtrado de cones-disp2.

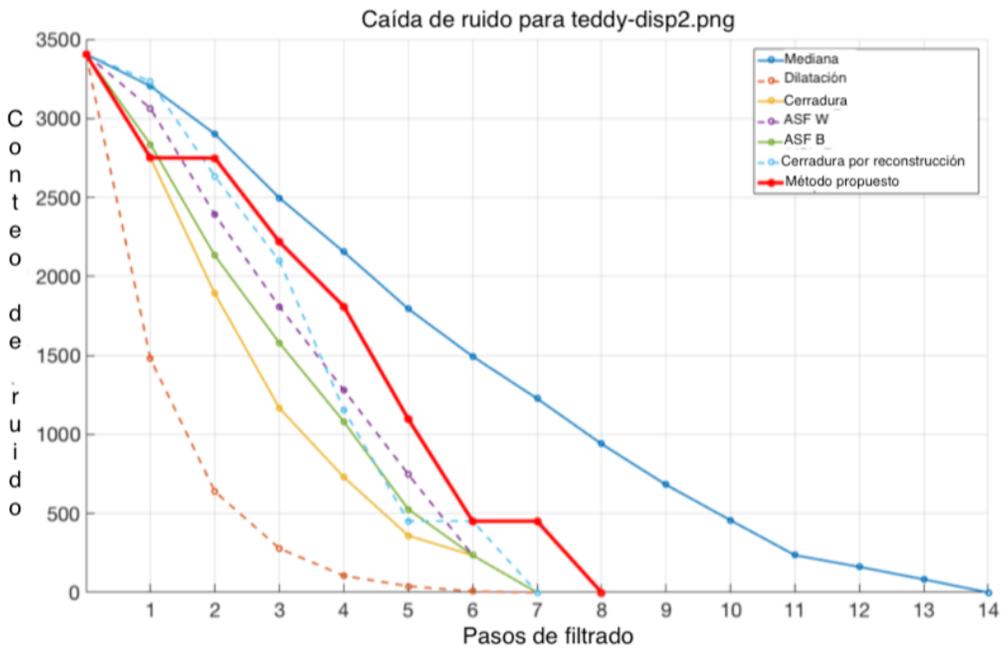


Figura 27. Pasos de filtrado de teddy-disp2.

Adicionalmente se midieron los tiempos de procesamiento, estos resultados se pueden observar en la Tabla (4.1).

Tabla 4.1 Tiempos de procesamiento.

Filtro	cones-disp2.png	teddy-disp2.png
Mediana	0.60	0.70
Dilatación	0.48	0.56
Cerradura	1.25	1.13
ASF W	2.42	1.77
ASF B	2.44	1.89
Cerradura por reconstrucción	7.67	9.06
Método propuesto	7.11	5.80

Para el segundo caso se tomaron imágenes del ser de Tsukuba y se les indujo ruido aleatorio mediante Python y OpenCV, Figura 28. Esta vez el proceso de filtrado se evaluó contra otras herramientas de filtrado generales encontradas en el estado del arte. Se midió el desempeño de la metodología propuesta usando los indicadores PSNR, SSIM, y MSE.

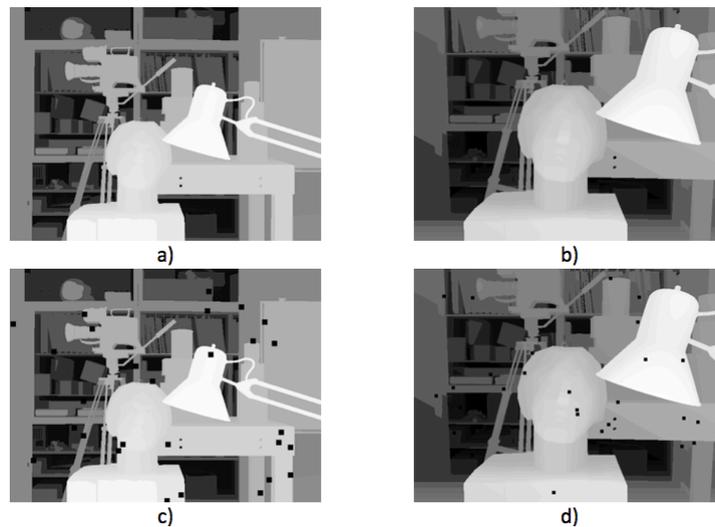


Figura 28. Imágenes del set Tsukuba de prueba, a) imagen original frame_1.png, b) imagen original frame_100.png, c) frame_1.png con ruido erosionado 5 y d) frame_100.png con ruido erosionado 3.

Tomando como referencia base las imágenes sin ruido del set Tsukuba se observa que la metodología propuesta logra resultados muy buenos medidos en términos de los evaluados es estadísticos mencionados. El algoritmo propuesto logra superar a todos y cada uno de los algoritmos comparados y se sitúa como una clara opción para filtrado de mapas de profundidad. Los resultados de filtrado para frame_1.png se muestran en la Tabla 4.2.

Tabla 4.2 Valores PSNR, MES y SSIM para el proceso de filtrado de frame_1.

Filtro	PSNR	MSE	SSIM
Filtro bilateral	26.09	159.66	0.88
Filtro noise-aware	25.77	216.61	0.86
Filtro de moda ponderada	25.17	197.32	0.87
Difusión anisotrópica	25.99	163.41	0.87
Joint WMF	22.87	335.49	0.84
Cerradura por reconstrucción	24.64	223.38	0.92
Método propuesto	31.01	51.51	0.96

Por su parte los resultados de filtrado para frame_100.png se muestran en la Tabla 4.3.

Tabla 4.3 Valores PSNR, MES y SSIM para el proceso de filtrado de frame_100.

Filtro	PSNR	MSE	SSIM
Filtro bilateral	29.28	76.68	0.91
Filtro noise-aware	28.13	99.87	0.90
Filtro de moda ponderada	28.32	95.62	0.90
Difusión anisotrópica	28.85	84.70	0.91
Joint WMF	27.35	119.62	0.89

Cerradura por reconstrucción	31.09	50.47	0.97
Método propuesto	36.79	13.61	0.98

4.2 Filtrado de mapas de profundidad basado en características de color, forma y vecindad

La metodología propuesta fue probada usando el set de Middlebury con la imagen Cones desde dos diferentes puntos de vista o perspectivas comparando los resultados de filtrado contra el mismo algoritmo *MCbR* en que está basada y contra los filtros más comunes de mapas de profundidad encontrados en el estado del arte. Las imágenes fueron rellenadas manualmente para establecer una comparativa natural contra el resultado del filtro. De igual forma se probó la metodología utilizando imágenes del set sin ruido como referencia e introduciéndoles ruido sintético mediante OpenCV con formas cuadradas y circulares en posiciones aleatorias. En ambos casos se realizó la medición de los parámetros PSNR, MSE y SSIM.

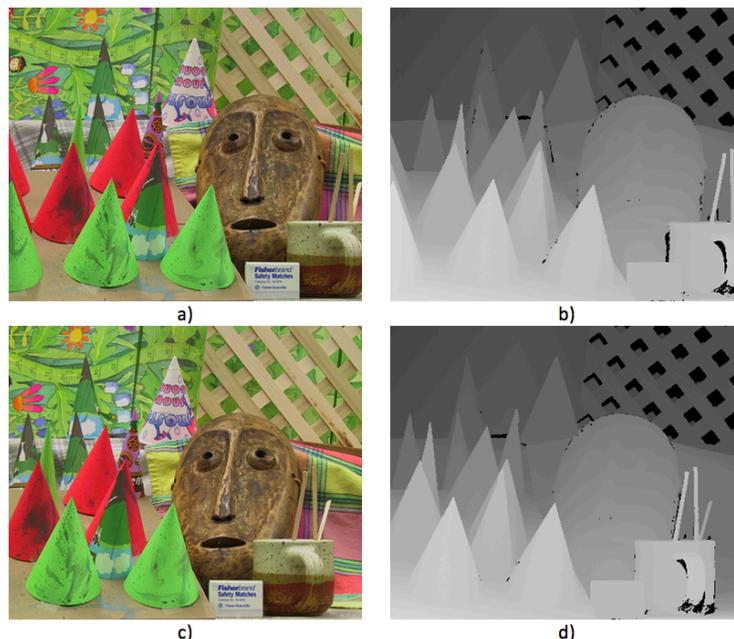


Figura 29. Imágenes de pruebas para probar la metodología propuesta, a) imagen color im2.png y su correspondiente, b) mapa de profundidad disp2.png, c) imagen color im6.png y su correspondiente y d) mapa de profundidad disp6.png.

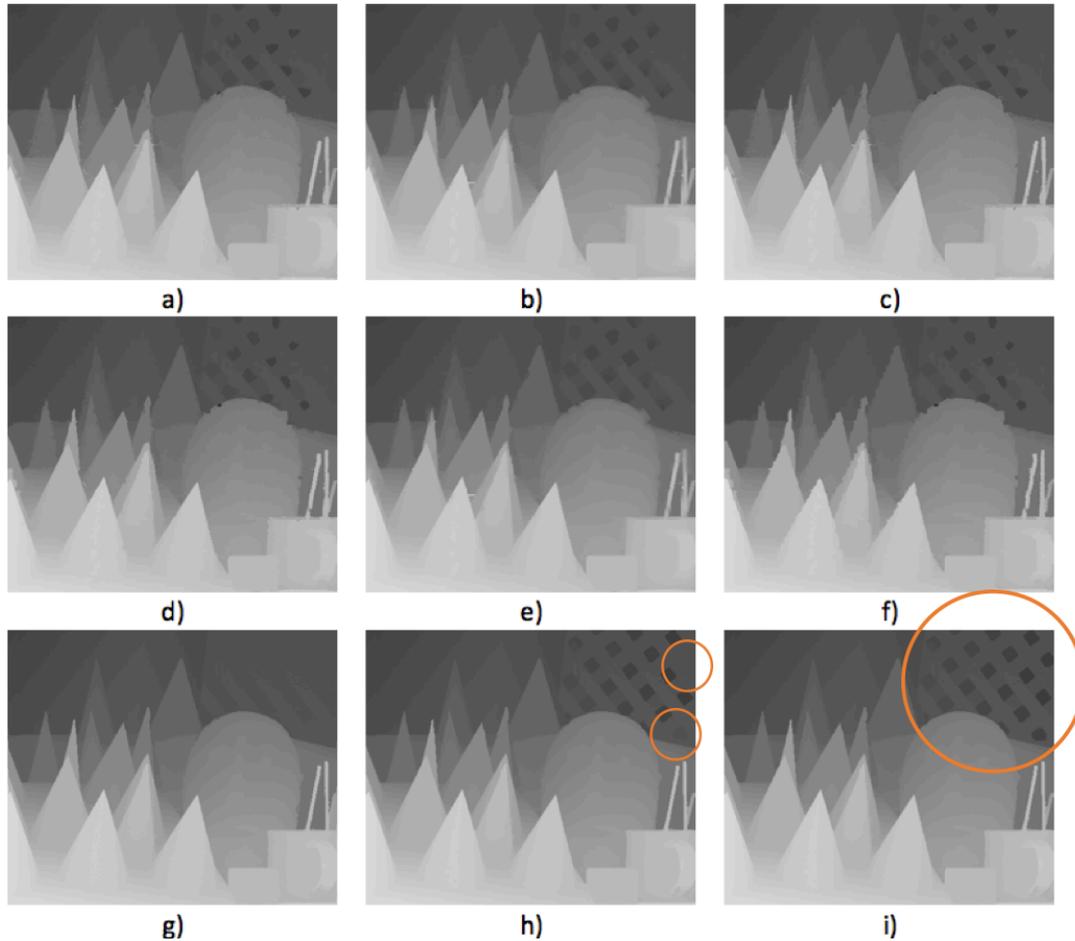


Figura 30. Filtrado de cones-disp2, a) filtro bilateral, b) filtro anisotrópico, c) filtro de moda ponderada, d) muestreo superior bilateral, e) campos aleatorios de Markov (kernel data term), f) filtro noise-aware, g) cerradura por reconstrucción, h) cerradura por reconstrucción modificada e i) método propuesto.

El análisis cuantitativo del filtrado de cones-disp2.png es mostrado en la Tabla 4.4. En el que el método propuesto se desempeña mejor que los demás filtros usados en la comparación.

Tabla 4.4 Valores PSNR, MES y SSIM para el proceso de filtrado de cones-disp2.png.

Filtro	PSNR	SSIM	MSE
Difusión anisotrópica	34.8675	0.9591	21.1995
Filtro bilateral	34.7865	0.9518	21.5988
Muestro superior bilateral	34.4389	0.9513	23.3984
Filtro de capas bilateral	25.3492	0.8901	189.7390

MRF (kernel data term)	34.7652	0.9584	21.7049
MRF (tensor)	31.6071	0.9155	44.9128
Filtro noise-aware	33.1645	0.9397	31.3786
Filtro de moda ponderada	34.3600	0.9480	23.8274
Cerradura por reconstrucción	37.3571	0.9821	11.9500
<i>MCbR</i>	46.4138	0.9976	1.4849
Método propuesto	52.6652	0.9987	0.3520

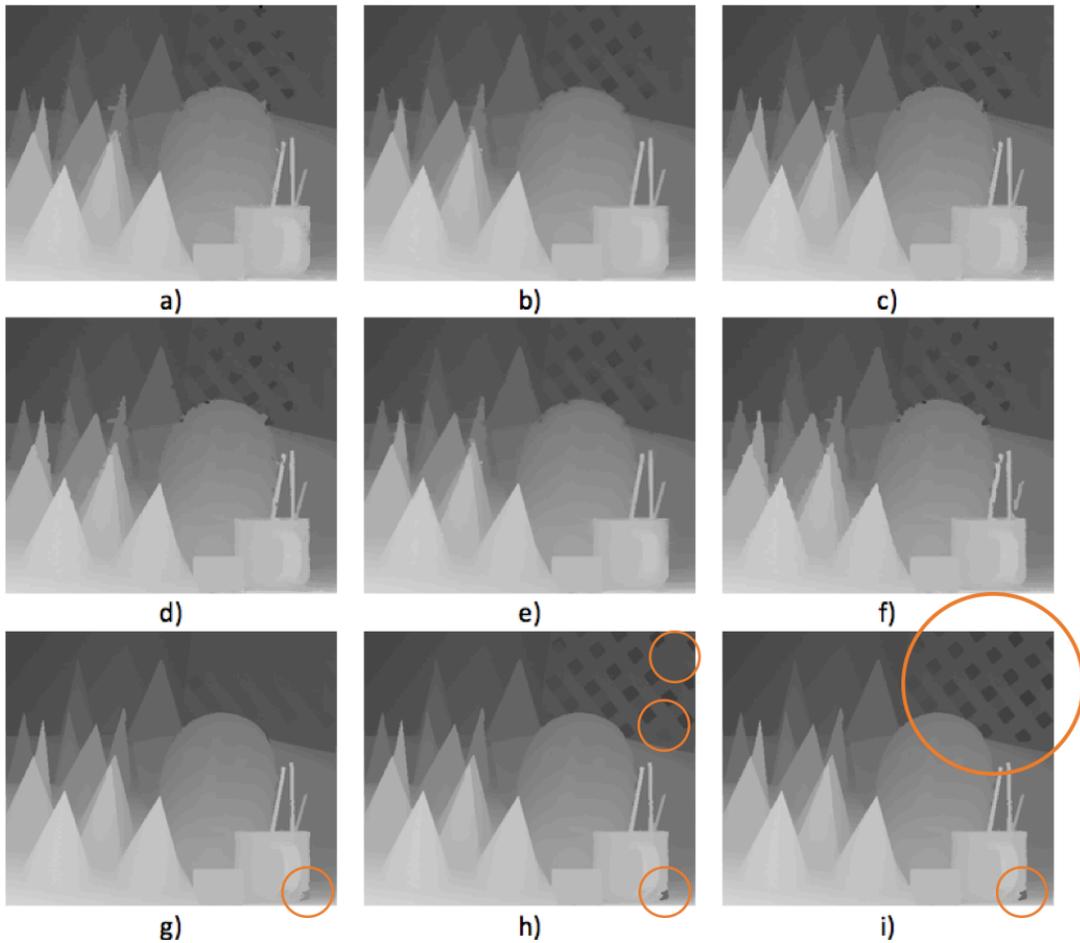


Figura 31. Filtrado de conos-disp6, a) filtro bilateral, b) filtro anisotrópico, c) filtro de moda ponderada, d) muestreo superior bilateral, e) campos aleatorios de Markov (kernel data term), f) filtro noise-aware, g) cerradura por reconstrucción, h) cerradura por reconstrucción modificada e i) método propuesto.

El análisis cuantitativo del filtrado de cones-disp6.png es mostrado en la Tabla 4.5. En el que el método propuesto se desempeña mejor que los demás filtros usados en la comparación.

Tabla 4.5 Valores PSNR, MES y SSIM para el proceso de filtrado de cones-disp6.png.

Filtro	PSNR	SSIM	MSE
Difusión anisotrópica	36.1637	0.9689	15.7292
Filtro bilateral	35.4634	0.9558	18.4817
Muestro superior bilateral	35.3598	0.9558	18.9279
Filtro de capas bilateral	26.7205	0.9254	138.3657
MRF (kernel data term)	36.0277	0.9083	16.2295
MRF (tensor)	32.5415	0.9289	36.2188
Filtro noise-aware	34.8717	0.9524	21.1793
Filtro de moda ponderada	35.0014	0.9482	20.5563
Cerradura por reconstrucción	38.6494	0.9908	8.8741
<i>MCbR</i>	41.2286	0.9988	4.9002
Método propuesto	41.2288	0.9988	4.9901

En ambos ejemplos mostrados la metodología propuesta obtiene los mejores parámetros evaluados cuantitativamente. En ambos casos también la *MCbR* obtiene los segundos mejores indicadores lo cual se puede interpretar como una clara mejora al algoritmo propuesto originalmente.

4.3 Aplicaciones de las metodologías desarrolladas a la robótica

Para validar la propuesta presentada en la introducción de este trabajo y en los antecedentes se propusieron una serie de pruebas aplicadas al robot manipulador marca Yaskawa que se encuentra en la Facultad de Ingeniería campus San Juan del Río relacionado con sus áreas de trabajo aplicando las metodologías desarrolladas durante este trabajo de investigación enfocadas al filtrado y reconstrucción morfológicos comparando con las demás herramientas del estado del arte. Se montaron un sensor TOF modelo Mesa SR4000 y un sensor de profundidad basado en luz estructurada Kinect versión 1 utilizando un dispositivo diseñado y fabricado en la Universidad Autónoma de Querétaro por parte del alumno de licenciatura Rogelio Cedeño Moreno y del alumno de doctorado Gilberto Alvarado Robles. El diseño se muestra en los anexos de este documento de tesis. Ambos sensores fueron montados en el robot y se realizó una prueba de medición vertical en el eje Z. El montaje de ambos dispositivos se puede observar en la Figura 32.

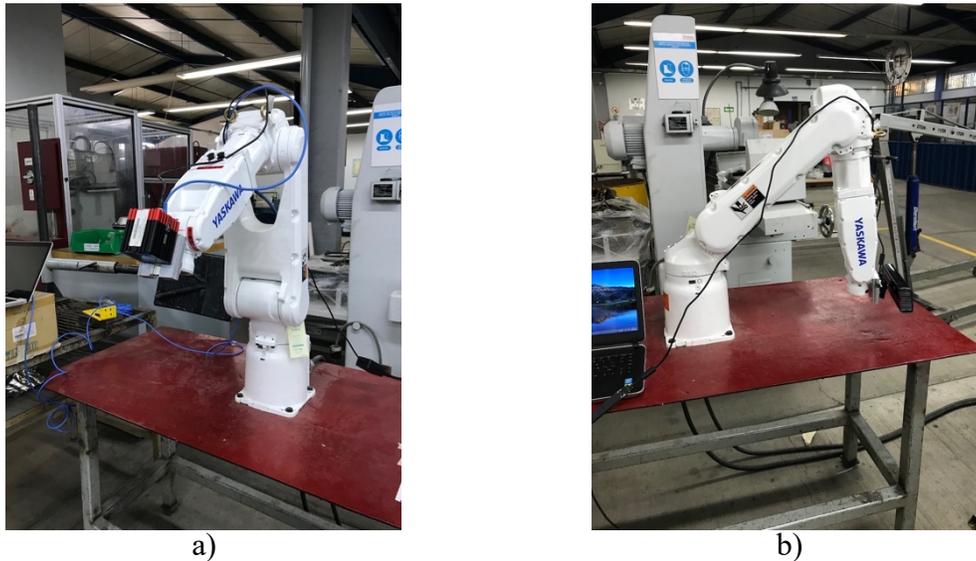
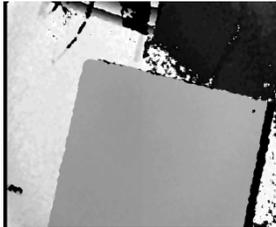
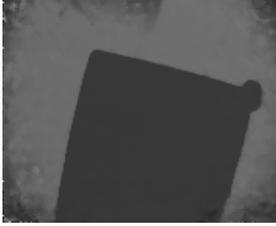
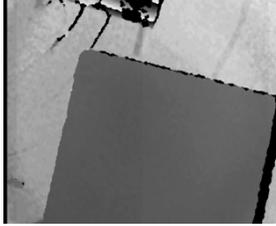
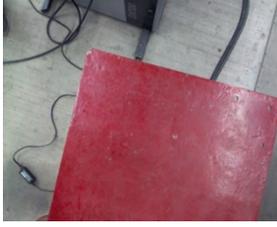
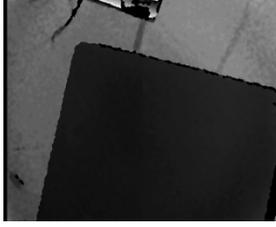
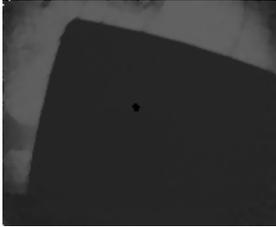
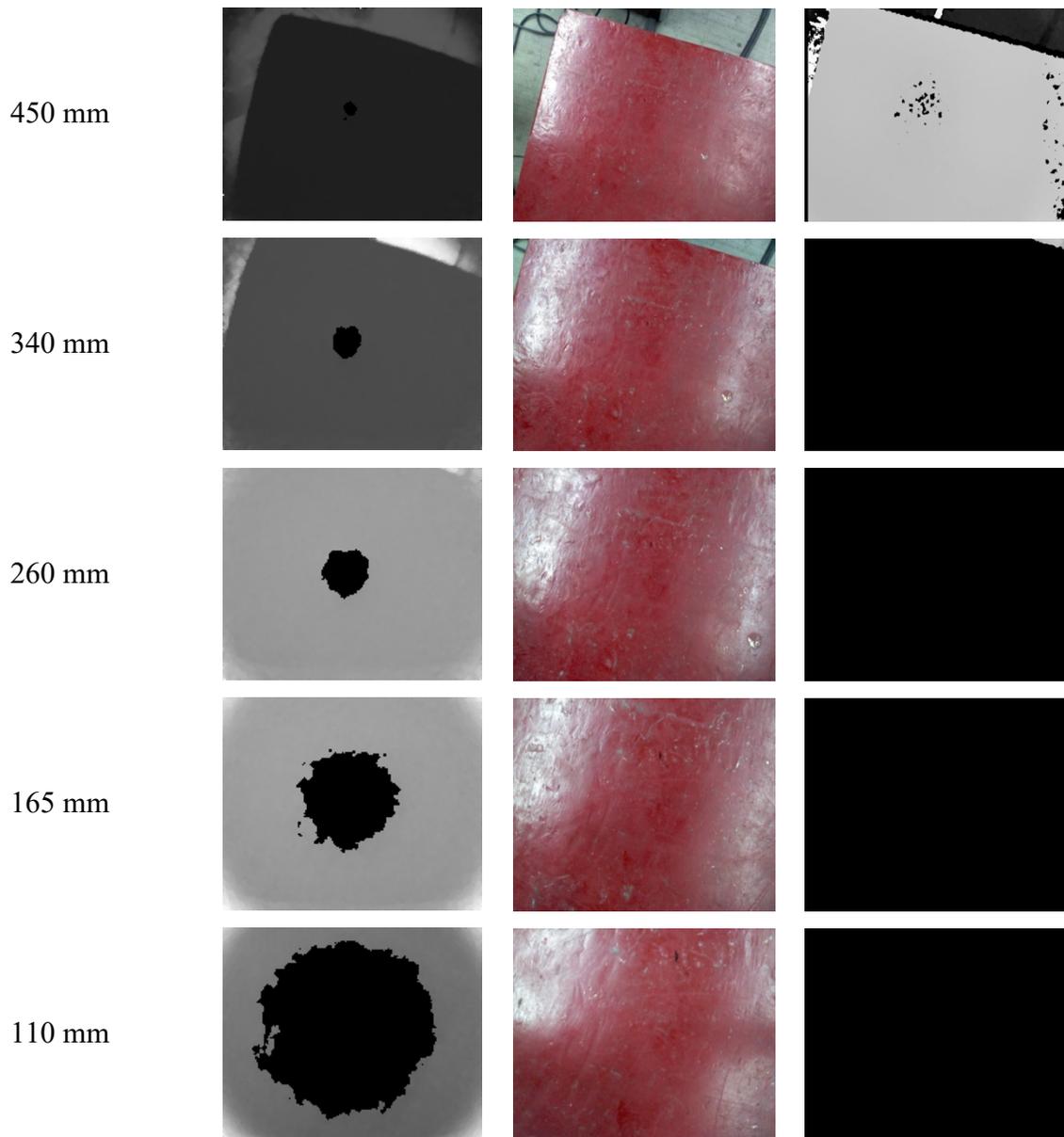


Figura 32. Sensores TOF montados sobre el robot Yaskawa, a) sensor Mesa SR4000, b) sensor Microsoft Kinect versión 1.

Se programó en el robot una trayectoria perpendicular desde una altura de 930 mm hasta una altura de 110 mm, durante la cual se realizaron mediciones en posiciones aleatorias de altura utilizando los sensores de profundidad. Para la cámara MESA SR se utilizó como herramienta de filtrado la cerradura por reconstrucción modificada, mientras que para las imágenes adquiridas usando el sensor Kinect se filtraron usando la metodología soportada por color. Para la realización de estas pruebas se contó con el apoyo del estudiante de la maestría, Ervin Galán Uribe.

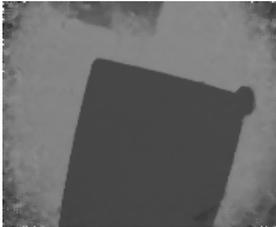
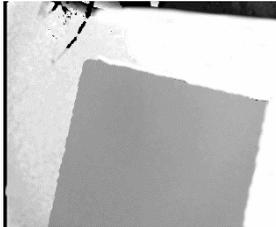
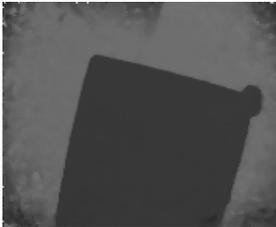
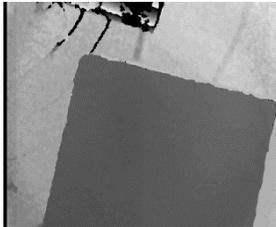
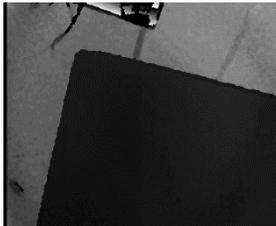
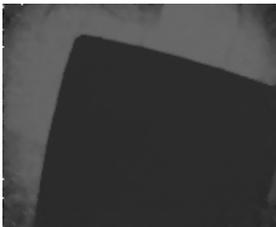
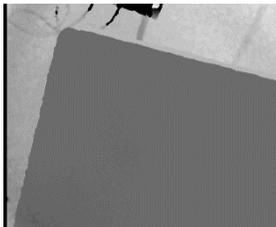
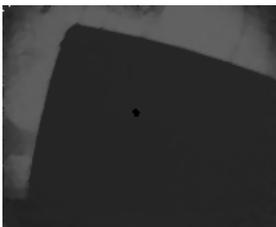
Tabla 4.6 Prueba de medición de profundidad a diferentes alturas, imágenes originales.

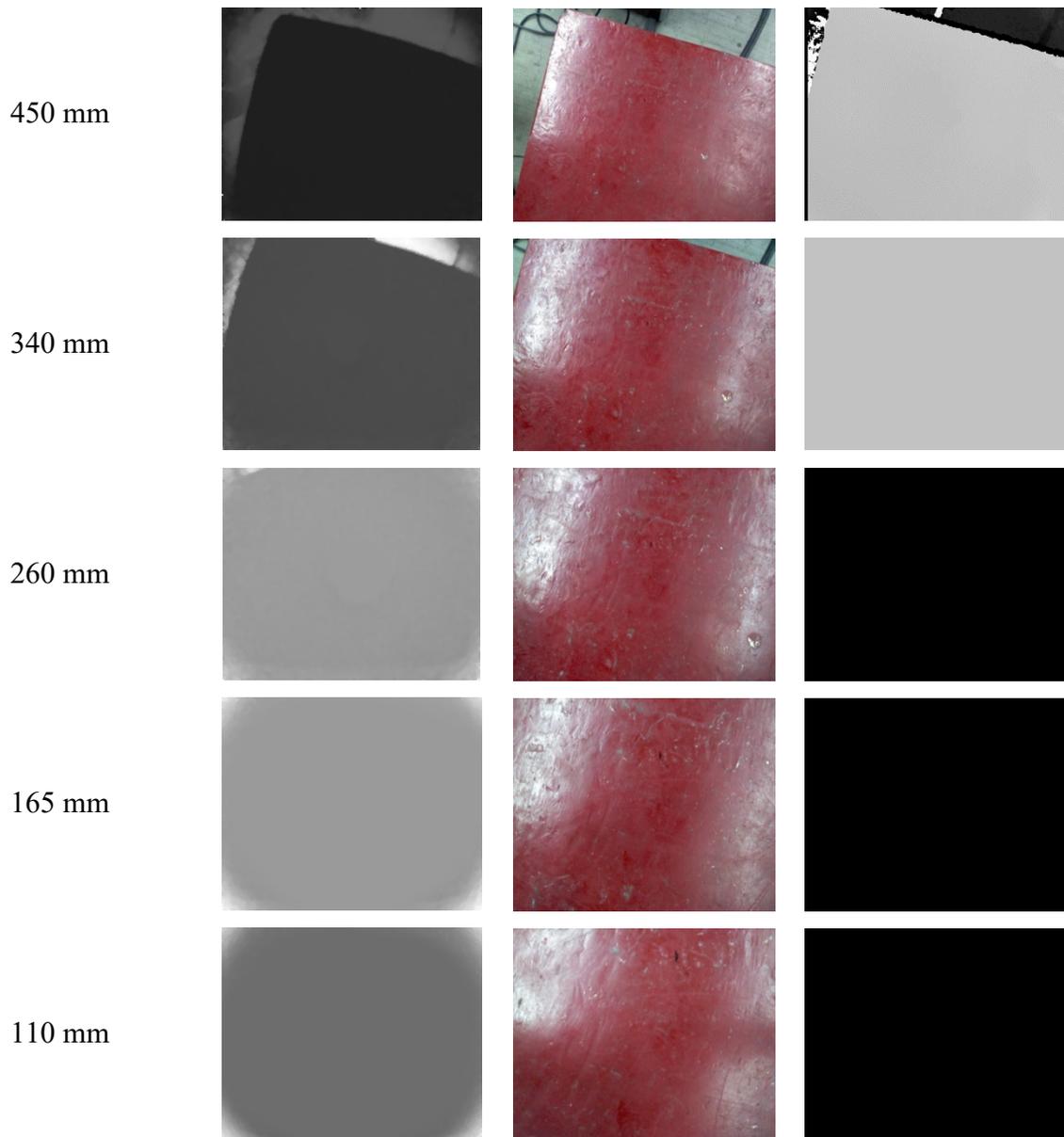
Altura	MESA SR4000	Kinect color	Kinect profundidad
930 mm			
860 mm			
800 mm			
740 mm			
610 mm			
520 mm			



Como se puede apreciar en la Tabla 4.6, las imágenes presentan diferentes niveles de ruido denotado con píxeles en negro lo cual dificulta realizar mediciones de profundidad en los píxeles disponibles dentro de la resolución total de la imagen haciendo necesario una etapa de filtrado. Estos ruidos pueden deberse a reflejos, o debido a las características propias del material observado. Para las imágenes tomadas sólo con el sensor Mesa SR4000 se filtraron usando la metodología *MCbR*, por su parte para las imágenes tomadas con el Kinect se filtraron usando la metodología apoyada por imagen color.

Tabla 4.7 Prueba de medición de profundidad a diferentes alturas, imágenes después de un proceso de filtrado.

Altura	MESA SR4000	Kinect color	Kinect profundidad
930 mm			
860 mm			
800 mm			
740 mm			
610 mm			
520 mm			



La reconstrucción de las imágenes sólo se puede evaluar de forma cualitativa dado que no existe una referencia para comparación. Salvo las imágenes del Kinect tomadas a una altura de 260 mm, 165 mm y 110 mm en las cuales debido al rango de operación del sensor fue imposible detectar información, en todas las demás es posible realizar mediciones de profundidad. Por su parte para las imágenes tomadas con la Mesa SR4000 se observa un patrón radial y se pierde detalle por completo de la base del robot observada.

4.4 Resultados de metodologías adicionales desarrolladas durante este trabajo de tesis

En esta sección se muestran los resultados de las metodologías adicionales desarrolladas durante este trabajo de tesis.

4.4.1 Caracterización de fallas en aspas de aerogeneradores usando procesamiento de imágenes

Para validar la metodología propuesta se utilizaron imágenes obtenidas a partir de una visita de campo a campos aerogeneradores de una empresa ubicada en La Ventosa, Oaxaca, por cuestiones de secrecía industrial es imposible mostrar imágenes de dicho lugar, sin embargo, se puede mostrar las estadísticas obtenida del trabajo de clasificador usando una validación apoyada por una persona que evaluaba la clasificación. Básicamente se tuvo un 90% de éxito en el proceso de clasificación de fallas tipos grietas. Los problemas detectados se debieron a ruido y a un filtrado del tipo de falla de forma excesivo como en el caso de grietas que fueran apenas visibles en las imágenes. Para las fallas de erosión en el borde de ataque del aspa y de descarga eléctrica se tuvo un 80% de eficiencia en la clasificación debido a altas similitudes entre ambas. Los principales errores se pueden asociar a una incorrecta selección de la región de interés y errores de perspectiva, Figura 33.

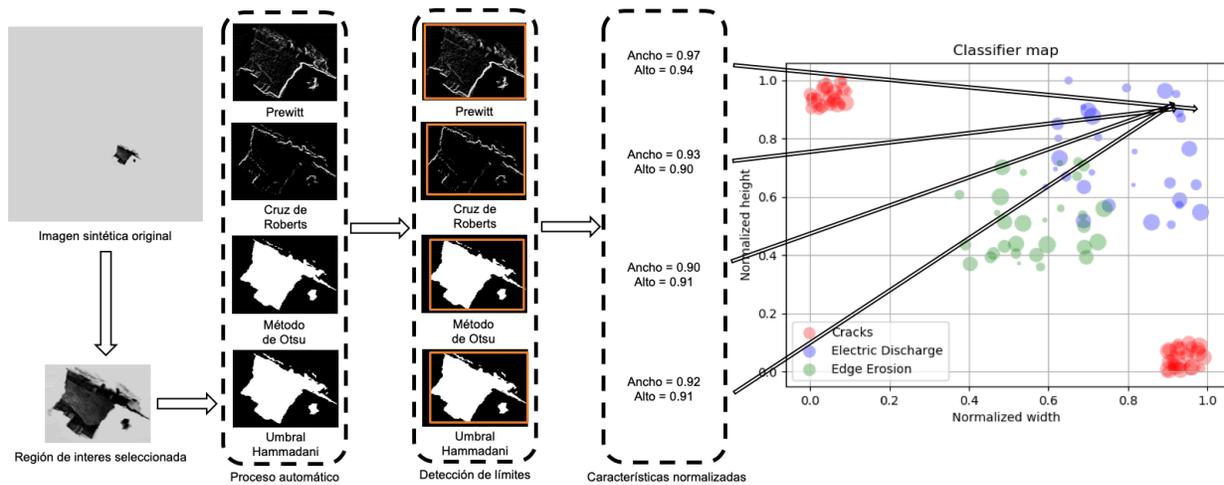


Figura 33. Proceso de clasificación de fallas en turbinas de aerogeneradores.

4.4.2 Detección de desbalances térmicos en imágenes termográficas para detección temprana de cáncer de mama

Para validar la metodología propuesta se utilizaron imágenes de la Database for Mastology Research (DMR). Dicha base de datos contiene termogramas clasificados como saludable o enfermo, con diagnósticos previos obtenidos mediante mastografía o estudios de biopsia. La base de datos incluye imágenes tomadas usando protocolos estáticos y dinámicos. Específicamente, para el protocolo estático se tomaron imágenes a 1 m de distancia con un tiempo de estabilización térmico de entre 10 y 15 minutos. Dicho tiempo corresponde a una diferencia del 60% con el tiempo propuesto en la sección anterior. Del set de imágenes clasificado como saludable, 376 termogramas aleatorios fueron seleccionados, mientras todos los 42 clasificados como con condición enfermo fueron tomados. Nuestro sistema logra un 80.95% de coincidencia para los casos saludables y un 78.26% de coincidencia para los casos enfermos, comparados con los diagnósticos provistos por la base de datos, sin embargo, es importante mencionar que para el 44.3% de las imágenes analizadas nuestro sistema fue incapaz de proveer un resultado debido a segmentación incorrecta. Los resultados de dichos análisis pueden ser vistos en la Tabla 4.8.

Tabla 4.8 Validación usando la DMR.

Casos (Total)	Sanos	Enfermos	Desconocido
Sanos (37)	17	4	16
Enfermos (42)	5	18	19
Total (79)	22	22	35

Para las pruebas realizadas de forma local con la base de datos de ENSAIN de la facultad de Enfermería de la UAQ tomadas en conjunto con Grupo Reto de San Juan del Río, se obtuvieron los siguientes resultados mostrados en la Tabla 4.9.

Tabla 4.9 Validación usando el estudio clínico propio.

Casos (Total)	Sanos	Enfermos	Desconocido
Sanos (414)	364	43	7
Enfermos (40)	5	33	2
Total (454)	369	76	9

4.4.2.1 Caso 1 - Saludable

El proceso de segmentación de las imágenes para el primer caso sano puede ser visto en la Figura 34.

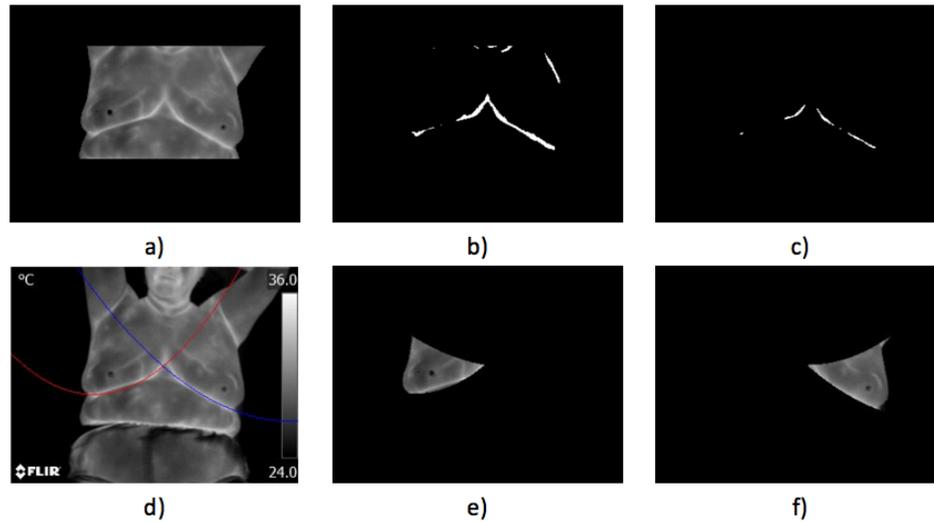


Figura 34. Caso 1 Saludable, a) termograma original, b) y c) detección de los límites inferiores, d) polinomios de segundo grado, e) seno derecho, y f) seno derecho segmentados.

Para el primer caso de ejemplo se midieron los siguientes valores estadísticos mostrados en la Tabla 4.10.

Tabla 4.10 Caso 1 Saludable.

Temperatura	Izquierdo	Derecho	Diferencia
Promedio	26.43°C	25.7°C	0.73°C
Max	35.56°C	36.5°C	0.94°C
Min	24.04°C	24.04°C	0°C
SD	3.34°C	3.61°C	0.27°C

4.4.2.2 Caso 2 - Saludable

El proceso de segmentación de las imágenes para el segundo caso sano puede ser visto en la Figura 35.

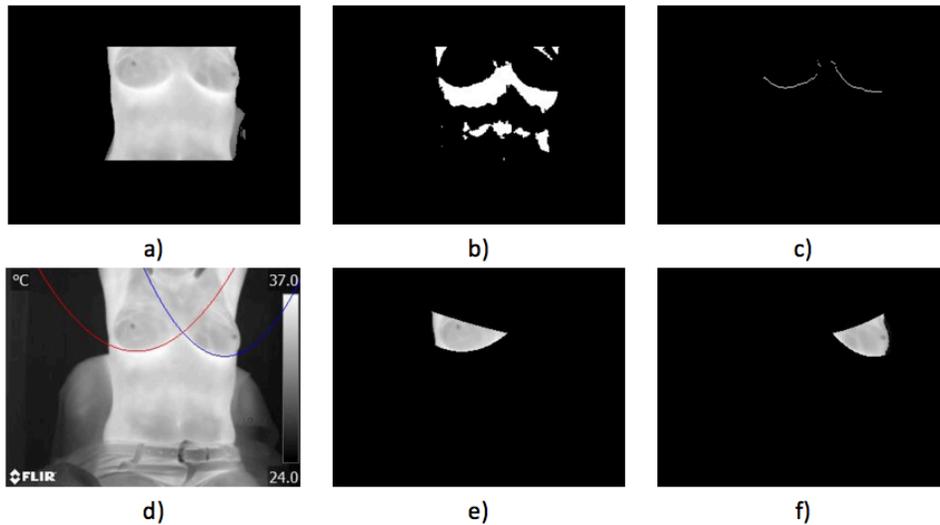


Figura 35. Caso 2 Saludable, a) termograma original, b) y c) detección de los límites inferiores, d) polinomios de segundo grado, e) seno derecho, y f) seno derecho segmentados.

Para el segundo caso de ejemplo se observaron los siguientes valores estadísticos mostrados en la Tabla 4.11.

Tabla 4.11 Caso 2 Saludable.

Temperatura	Izquierdo	Derecho	Diferencia
Promedio	29.07°C	29.65°C	0.58°C
Max	36.45°C	34.29°C	2.16°C
Min	24.04°C	24.04°C	0°C
SD	3.6°C	2.98°C	0.62°C

4.4.2.3 Caso 3 - Angiogénesis

El resultado de la segmentación automática de las imágenes para el tercer caso que denota signos de angiogénesis se puede observar en la Figura 36.

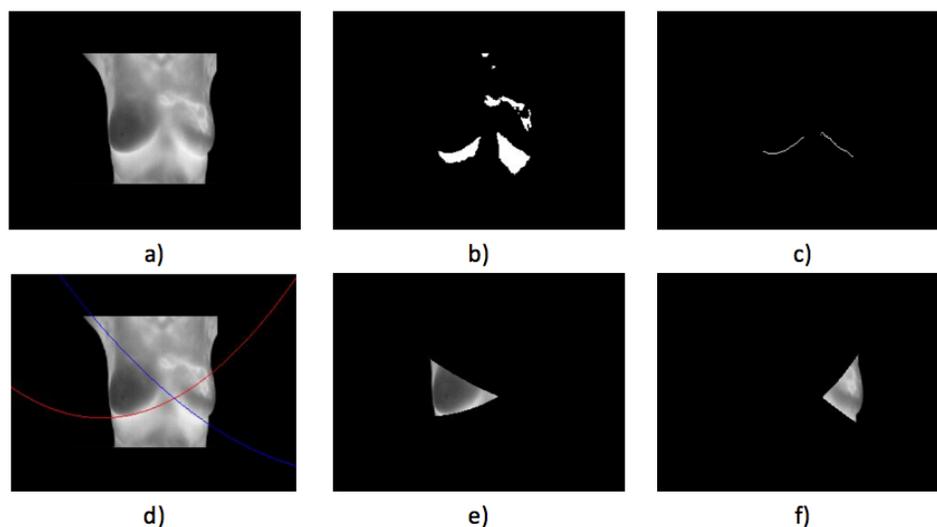


Figura 36. Caso 3 Angiogénesis, a) termograma original, b) y c) detección de los límites inferiores, d) polinomios de segundo grado, e) seno derecho, y f) seno derecho segmentados.

Para el tercer caso de ejemplo con signos de angiogénesis se midieron los siguientes valores estadísticos mostrados en la Tabla 4.12.

Tabla 4.12 Caso 3 Angiogénesis.

Temperatura	Izquierdo	Derecho	Diferencia
Promedio	30.1°C	28.29°C	1.81°C
Max	36.45°C	33.6°C	2.85°C
Min	24.04°C	24.04°C	0°C
SD	3.6°C	2.78°C	0.82°C

A partir de la diferencia significativa de 1.81°C en el promedio de temperaturas entre ambos senos. Se hizo una inspección adicional identificando la forma mostrada en la Figura 37.

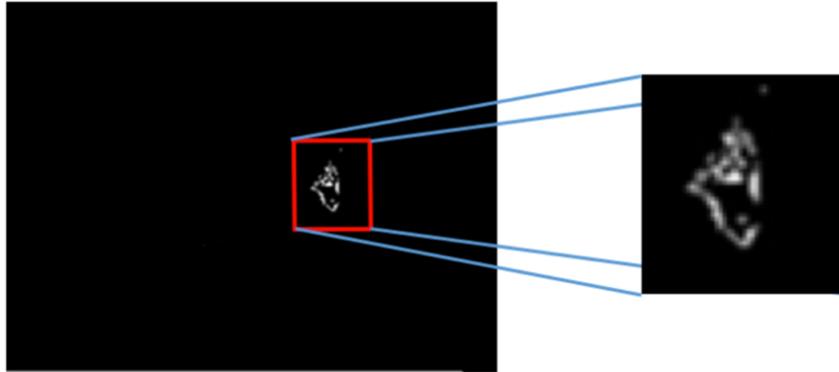


Figura 37. Detalle del seno izquierdo con la forma tan característica de un caso de angiogénesis.

4.4.2.4 Caso 4 - Cáncer

El proceso de segmentación de las imágenes para el cuarto caso clasificado como sospechoso de cáncer puede ser visto en la Figura 38.

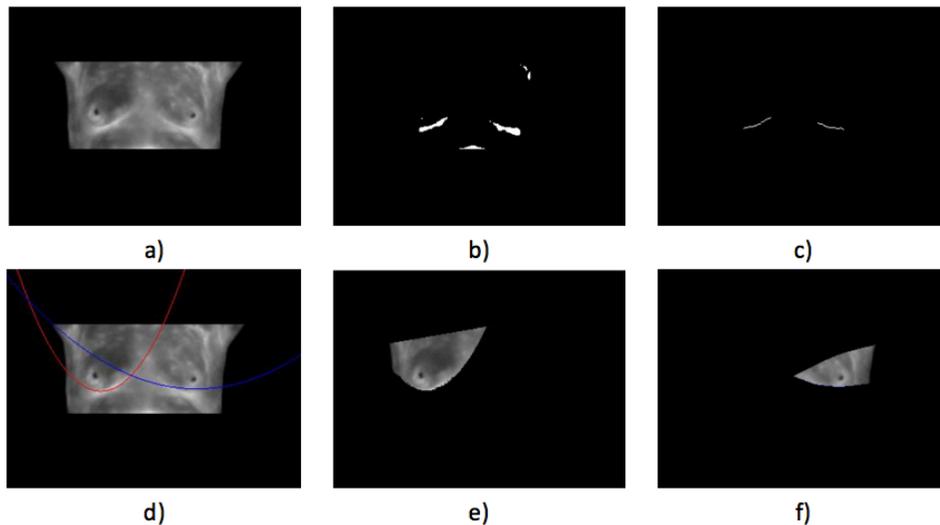


Figura 38. Caso 4 Caso sospechoso de cáncer de mama, a) termograma original, b) y c) detección de los límites inferiores, d) polinomios de segundo grado, e) seno derecho, y f) seno derecho segmentados.

Para el último caso de ejemplo se midieron los siguientes valores estadísticos mostrados en la Tabla 4.13.

Tabla 4.13 Caso 4 Cáncer.

Temperatura	Izquierdo	Derecho	Diferencia
Promedio	26.91°C	28.84°C	1.92°C
Max	27.82°C	33.35°C	8.53°C
Min	24.04°C	24.04°C	0°C
SD	1.11°C	3.57°C	2.46°C

A partir de la diferencia de temperaturas medidas de 1.92°C. Se realizó una segmentación usando el algoritmo Watershed. Este algoritmo permite aislar regiones con temperaturas semejantes. A partir de esta segmentación y determinando que en el seno derecho se mide la temperatura más alta. Se detecta el punto más caliente medido que puede servir como indicativo para ubicar la localización de un posible tumor cancerígeno. Este punto puede ser visto en la Figura 39.



Figura 39. Detalle del seno derecho mostrando el punto más caliente.

Capítulo V

5. Conclusiones

En este trabajo se realizaron una serie de metodologías de filtrado de mapas de profundidad de sistemas TOF y de luz estructurada con el fin de que sirvan para su implementación en la calibración de sistemas robóticos de 6 grados de libertad y sus áreas de trabajo, así como para su aplicación en otras áreas que así lo requieran.

Los sistemas de imágenes que permiten obtener mapas de profundidad están en sus etapas tempranas de desarrollo, si bien es cierto que bajo ciertas condiciones pueden lograrse medición con un alto nivel de precisión esta se ve afectada por la naturaleza propia de los sensores. El ruido afecta en gran medida sus mediciones y genera una cierta incertidumbre que debe ser minimizada antes de poderse aplicarse en contraposición a los sistemas metrológicos tradicionales, de hecho, se recomienda una combinación de sistemas de visión con electrónica instrumentada para lograr buenos resultados. Respecto al tema del ruido, las metodologías propuestas se comportaron de manera adecuada superando a filtros reportados en el estado del arte midiendo factores estadísticos como lo son el PSNR, MSE, y SSIM, además dichas metodologías fueron validadas usando imágenes de bases de datos estándar que se encuentran en la red, este paso es fundamental antes de probar las metodologías presentadas en ambientes reales.

Otra característica importante de las metodologías propuestas es que estas ofrecen mejores resultados que aquellas herramientas en que están basados; el algoritmo propuesto, la cerradura por reconstrucción modificada (*MCbR*) se coloca por encima de la cerradura por reconstrucción original permitiendo preservar áreas reconstruidas a grandes tamaños de elemento estructural usados gracias a su característica iterativa. Si bien su implementación en un lenguaje interpretado como Python resulto lenta, al traducirla a C++ se obtuvieron tiempos muy aceptables que caen dentro del rango de los demás filtros comparados. Finalmente, el algoritmo presentado y dadas las propiedades de la morfología matemática permite generar de forma inmediata su dual, la apertura por reconstrucción modificada (*MCbR*) con muchas posibles aplicaciones por delante.

La combinación color más profundidad ofrece aún más posibilidades de filtrado, donde el único inconveniente es la misma calibración entre cámaras, sin embargo, una vez superado ese problema es posible intuir información perdida en el mapa de profundidad a partir de características de forma, color y vecindad. Doble información, en contraposición, se traduce en el doble o más de tiempo de procesado lo cual debe ser tomado en cuenta dados los requerimientos y capacidad computacionales. Se propone para trabajos futuros utilizar secuencias de mapas de profundidad y secuencias de pares profundidad más color para realizar la reconstrucción de los mapas de profundidad con las necesidades de optimización de algoritmos y de uso de hardware potente que conlleva.

Adicionalmente se realizaron propuestas novedosas en vinculación con la industria que permiten la caracterización de fallas en fuselajes de aviones y en aerogeneradores con la generación de metodologías de procesado, filtrado y medición, así como con la combinación de hardware comercial. Finalmente, en el campo médico se dio continuidad al trabajo de Sofía Giovanna Vega Mancilla relacionado con el cáncer de mama. Para ambas propuestas se concluye que el área de procesamiento de imágenes tiene un gran potencial en la Universidad Autónoma de Querétaro y que se ha generado un gran conocimiento que permite la propuesta de experimentos de validación en laboratorio, áreas interiores y exteriores. Condiciones ambientales, de iluminación y de requerimientos de hardware ya son tomados en cuenta para la generación de ciencia de alto nivel.

6. REFERENCIAS

- An, C. H., Atkeson, C. H., and Hollerbach, J. M. 1998. Model Based Control of a Robot Manipulator (MIT Press, Cambridge).
- Angulo, J., Serra, J. Segmentación de imágenes en color utilizando histogramas bi-variables en espacios color polares luminancia/saturación/matiz, *Computación y Sistemas* 8 (4) (2005) 303–316.
- Anonymous. Manipulating industrial robots | Performance criteria and related test methods, ISO 9283; 1998.
- Ceder, V. L. 2010. The Quick Python Book. Second Edition. Manning Publications Co, Sound View Court 3B, Greenwich, CT 06830.
- Chen, Y., Zhang, W., Yan, K. Li, X. and Zhou, G. 2012. Extracting corn geometric structural parameters using Kinect. 2012 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 6673-6676. July 2012.
- Chiabrando, F., Chiabrando, R., Piatti, D., and Rinaudo, F. 2009. Sensors for 3D Imaging: Metric Evaluation and Calibration of a CCD/CMOS Time-of-Flight Camera. *Sensors* 2009, 9, 10080-10096; doi:10.3390/s91210080.
- Cho, K.-B., and Lee, B.-H. 2012. Intelligent Lead: A Novel HRI Sensor for Guide Robots. *Sensors* 2012, 12, 8301-8318; doi:10.3390/s120608301.
- Conrad, K. L., Shiakolas, P. S., and Yih, T. C. 2000. Robotic Calibration Issues: Accuracy, repeatability and calibration. Proceedings of the 8th Mediterranean Conference on Control & Automation (MED 2000), Rio, Patras, GREECE, 17-19.
- Cruz, L., Lucio, D., and Velho, L. 2012. Kinect and RGBD Images: Challenges and Applications. 25th SIBGRAPI Conference on Graphics, Patterns and Image Tutoriais, pp. 36-49, Aug. 2012.
- De León-Cuevas, A., Tovar-Arriaga, S., Vargas-Soto, J. E. y Pérez-Arreguín, J. I. 2012. Telecontrol de un robot móvil por medio de un sensor de proximidad RGB-D. Octavo Congreso Internacional de Ingeniera. Universidad Autónoma de Querétaro. pp. 258-268. ISBN 978-607-513-021-7.
- Díaz-Guerrero, R. E., Hernández-Barrón, L. A., Pedraza-Ortega, J. C., Gorrostieta-Hurtado, E. y Vargas-Soto, E. 2012. Análisis comparativo entre aplicaciones de reconocimiento de patrones mediante el uso del hardware Kinect TM. Octavo Congreso Internacional de Ingeniera. Universidad Autónoma de Querétaro. pp. 8-23. ISBN 978-607-513-021-7.
- Dolson, J., Baek, J., Plagemann, C., and Thrun, S. 2010. Upsampling range data in dynamic environments. 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp 1141-1148.
- Driels, M. R., and Swayze, W. E. 1994. Automated partial pose measurement system for manipulator calibration experiments, *IEEE Trans. Robot. Autom.* 10, 430-440.
- Du, G., and Zhang, P. 2013. Online robot calibration based on vision measurement. *Robotics and Computer-Integrated Manufacturing* 29 (2013) 484-492.

- Du, H., Oggier, T., Lustenberger, F., and Charbon, E. 2005. A virtual keyboard based on true-3d optical ranging. In Proc. British Machine Vision Conference 2005. pp. 220-229.
- Elatta, A. Y., Gen, L. P., Zhi, F. L., Daoyuan, Y. and Fei, L. 2004. An Overview of Robot Calibration. *Information Technology Journal* 3 (1): 74-78. ISSN 1682-6027.
- Flores-Guerra, J. 2008. Homography. Based Multiple Camera Detection of Camouflaged Targets. Department of Electrical & Computer Engineering, ECE. Thesis. University of Texas at El Paso.
- Foix, S., Alenya, G., and Torras, C. 2011. Lock-in time-of-flight (ToF) cameras: A survey. *IEEE Sensors Journal*, 11(9):1917-1926.
- Freedman, B., Shpunt, A., Machline, M., and Arieli, Y. 2008. Depth mapping using projected patterns. Patent. Publication number US20080240502 A1.
- Garduño-Ramón, M. A., Morales-Hernández, A. G., Morales-Hernández, L. A., Cruz-Albarrán, I. A. y Osornio-Rios, R. A. 2014a. Determinacion de parametros dimensionales en imágenes de líneas gravitacionales de referencia postural por medio de un sensor Kinect. XI Congreso Internacional sobre Innovación y Desarrollo Tecnológico, CIINDET 2014. Cuernavaca, Morelos.
- Garduño-Ramón, M. A., Morales-Hernández, L. A. y Osornio-Rios, R. A. 2014b. Morphological filters applied to Kinect depth images for noise removal as pre-processing stage. X Congreso Internacional de Ingeniera, CONIIN 2014. Universidad Autónoma de Querétaro. pp. 41-45. ISBN 978-607-513-XXX-X.
- Gonzalez, R. C., Woods, R. E. and Eddins, S. L. 2004. *Digital Image Processing*. Second Edition. Pearson Prentice Hall, Upper Saddle River, New Jersey 07458.
- Gonzalez, R. C., Woods, R. E. and Eddins, S. L. 2004b. *Digital Image Processing Using Matlab*. First Edition. Pearson Prentice Hall, Upper Saddle River, New Jersey 07458.
- Goswami, A., Quaid, A., and Peshkin, M. 1993. Identifying robot parameters using partial pose information, *IEEE Contr. Syst.*13, 6{14.
- Han, J., Shao, L., Xu, D., and Shotton, J. 2013. Enhanced Computer Vision with Microsoft Kinect Sensor: A Review. *IEEE Transactions on Cybernetics*. Vol. 43, No. 5, October, 2013.
- Hansard, M., Lee, S., Choi, O. and Horaud, R. 2012. *Time-of-Flight Cameras: Principles, Methods and Applications*. Unknown Edition. Springer, Upper Saddle River, New Jersey 07458.
- Hartley, R. and Zisserman, A. 2004. *Multiple View Geometry in Computer Vision*. Second Edition. Cambridge University Press, The Edinburg Building, Cambridge CB2 2RU, UK.
- Henry, P., Krainin, M., Herbst, E., Ren, X., and Fox, D. 2010. Rgbd mapping: Using depth cameras for dense 3d modeling of indoor environments. *RGB-D: Advanced Reasoning with Depth Cameras Workshop in conjunction with RSS*.
- Hernández-López, J. J., Quintanilla-Olvera, A. L., López-Ramírez, J. L., Rangel-Butanda, F. J., Ibarra-Manzano, M. A. and Almanza-Ojeda, D. L. 2012. Detecting objects using color and depth segmentation with Kinect sensor. *The 2012 Iberoamerican Conference on Electronics Engineering and Computer Science*. Elsevier Ltd.
- Hollerbach, J. M., and Wampler, C.W. 1996. The calibration index and taxonomy of kinematic calibration methods, *Int. J. Robot. Res.*15, 573-591.

- Khoshelham K., and Elberink, S. 2012. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, vol. 12, no. 2, pp. 1437-1454.
- Klimchik, A., Wu, Y., Caro, S., Furet, B., and Pashkevich, A. 2013. Advanced robot calibration using partial pose measurements. 18th International Conference on Methods and Models in Automation and Robotics (MMAR). pp. 264 - 269.
- Lau, K., Hocken, R., and Haynes, L. 1985. Robot performance measurements using automatic laser tracking techniques, *Robot. Comput.-Integr. Manuf.*2, 227-236.
- Ma, G., Wang, F., Qu, Z., and Feng, B. 2006. A feasible vision-based measurement method for robot orientation error. 1st International Symposium on Systems and Control in Aerospace and Astronautics, ISSCAA. pp. 1214 - 1217. IEEE.
- Marmellis, P. Z., and Marmellis, V. Z. 1978. *Analysis of Physiological Systems: The White-Noise Approach*. Plenum Press, New York.
- May, S., Werner, B., Surmann, H., and Pervolz, K. 2006. 3d time-of-flight cameras for mobile robotics. 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp.790-795.
- Mooring, B. W., Roth, Z.S., and Driels, M.R. 1991. *Fundamentals of Manipulator Calibration* (Wiley Interscience, New York).
- Motta, J. M. 2006. *Robot Calibration: Modeling Measurement and Applications. Industrial Robotics: Programming, Simulation and Applications*. ISBN 3-86611-286-6, pp. 702.
- Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohli, P., Shotton, J., Hodges, S., and Fitzgibbon, A. 2011. Kinectfusion: Real-time dense surface mapping and tracking. 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR). pp. 1-8.
- Nof, S. Y. 1999. *Handbook of Industrial Robotics. Second Edition*. John Wiley & Sons, Inc. 605 Third Avenue, New York, NY 10158-0012. ISBN 0-471-17783-0.
- Nubiola, A., Slamani, M., and Bonev, I. A. 2012. A new method for measuring a large set of poses with a single telescoping ballbar. *Precision Engineering* 37 (2013) 451-460.
- Nubiola, A. and Bonev, I. A. 2013. Absolute calibration of an ABB IRB 1600 robot using a laser tracker. *Robotics and Computer-Integrated Manufacturing* 29 (2013) 236-245.
- Nubiola, A. and Bonev, I. A. 2014. Absolute robot calibration with a single telescoping ballbar. *Precision Engineering* 38 (2014) 472-480.
- Pajares-Martinsanz G. y de la Cruz-García J. M., 2004. *Visión por Computador. Imágenes Digitales y Aplicaciones, Segunda Edición*. Editorial Alfaomega. México.
- Park, J., Kim, H., Tai, Y.-W., Brown, M.-S., and Kweon, I. S. 2011. High quality depth map upsampling for 3D-TOF cameras. *IEEE International Conference on Computer Vision (ICCV)*. pp. 1623-1630.
- Park, J.-H., Shin, Y.-D., Bae, J.-H., and Baeg M.-H. 2012. Spatial Uncertainty Model for Visual Features Using a KinectTM Sensor. *Sensors* 2012, 12, 8640-8662; doi:10.3390/s120708640.
- Peralta-Benhumea, S. H. 2012. *Interfaz de lenguaje natural usando Kinect*. Tesis de maestra. Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional. Unidad Zacatenco. Departamento de Computación. México, Distrito Federal.

- Piatti, D., and Rinaudo, F. 2012. SR-4000 and CamCube3.0 Time of Flight (ToF) Cameras: Tests and Comparison. *Remote Sens.* 2012, 4, 1069-1089; doi:10.3390/rs4041069.
- Plan Nacional de Desarrollo 2013 - 2018. Gobierno de la Republica.
- Pomares, J., Gil, P. and Torres, F. 2010. Visual Control of Robots Using Range Images. *Sensors* 2010, 10, 7303-7322; doi:10.3390/s100807303.
- Qi, X., Lechti, D., El-Badry, M., Chow, J., and Ang, K. 2014. Vertical Dynamic Detection Measurement in Concrete Beams with the Microsoft Kinect. *Sensors* 2014, 14, 3293-3307; doi:10.3390/s140203293.
- Russ J. C. 2007. *The Image Processing Handbook*. Fifth Edition. Taylor & Francis. USA.
- Ryden, F., Chizeck, H., Kosari, S. N., King, H., and Hannaford, B. 2010. Using kinect and a haptic interface for implementation of real-time virtual xtures. *Robotics, Science and Systems (RSS) Workshop on RGB-D Cameras*.
- Sánchez-López, J. R., Marin-Hernández, A., Palacios-Hernández, E. R., Rios-Figuera, H. V., and Marin-Urias, L. F. 2013. A Real-time 3D Pose Based Visual Servoing Implementation for an Autonomous Mobile Robot Manipulator. *Procedia Technology* 7 (2013) 416-423.
- Schamm, T., Strand, M., Gump, T., Kohlhaas, R., Zollner, J., and Dillmann, R. 2009. Vision and tof-based driving assistance for a personal transporter. *ICAR 2009, International Conference on Advanced Robotics, 2009*. pp 1-6.
- Shotton, J., Fitzgibbon, A., Cook, M., and Blake, A. 2011. Real-time human pose recognition in parts from single depth images. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 1297-1304.
- Siciliano, B., and Khatib, O. 2008. *Springer Handbook of Robotics*. Springer. Springer-Verlag Berlin Heidelberg. ISBN: 978-3-540-23957-4.
- Siciliano, B., Sciavicco, L., Villani, L. and Oriolo, G. 2009. *Robotics: Modeling, Planning and Control*. Springer, London.
- Smisek, J., Jancosek, M., and Pajdla, T. 2011. 3D with kinect. *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. pp. 1154-1160.
- Soutschek, S., Penne, J., Hornegger, J., and Kornhuber, J. 3-d gesture-based scene navigation in medical imaging applications using time-of-ight cameras. 2008. *CVPRW '08. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2008*. pp. 1-6.
- Spong, M. W., Hutchinson, S., and Vidyasagar, M. 1989. First Edition. John Wiley & Sons, Inc.
- Stoyanov, T., Louloudi, A., Andreasson, H. and Lilienthal, A. 2011. Comparative evaluation of range sensor accuracy in indoor environments. In *Proc. Eur. Conf. Mobile Robots*, pp. 19-24.
- Svaco, M., Sekoranja, B., Suligoj, F. and Jerbic, B. 2013. Calibration of an Industrial Robot using a Stereo Vision System. *24th DAAAM International Symposium on Intelligent Manufacturing and Automation. Procedia Engineering* 69 (2014) 459-463.
- Tang, G.-R., and Liu, L.-S. 1993. Robot calibration using a single laser displacement meter, *Mechatronics* 3, 503-516.
- Vincze, M., Prenninger, J. P., Gander, H. 1994. A laser tracking system to measure position and orientation of robot end effectors under motion, *Int. J. Robot. Res.*13, 305-314.

- von Engelhardt J., 2002. *The Language of Graphics: A Framework for the Analysis of Syntax and Meaning in Maps, Charts and Diagrams*. Universiteit van Amsterdam. Institute for Logic, Language and Computation.
- Webb, J. and Ashley, J. 2012. *Beginning Kinect Programming with the Microsoft Kinect SDK*. First Edition. Apress. Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. ISBN-13: 978-1-4302-4104-1.
- Whitney, D. E., Lozinski, C. A., and Rourke, J.M. 1986. Industrial robot forward calibration method and results, *ASME J. Dyn. Syst. Meas. Contr.*108, 1-8.
- Xue, Z., Ruhl, S. W., Hermann, A., Kerscher, T. and Dillmann, R. 2011. Autonomous grasp and manipulation planning using a ToF camera. *Robotics and Autonomous Systems* 60 (2012). pp. 387-395.
- Yuan, F., Swadzba, A., Philippsen, R., Engin, O., Hanheide, M., and Wachsmuth, S. 2009. Laser-based navigation enhanced with 3d time-of-flight data. *ICRA '09, IEEE International Conference on Robotics and Automation*, 2009. pp 2844-2850.

7. APÉNDICE I

Artículos publicados en revista Indizada JCR (Journal Citation Report)

-
- 2018 *Garduño-Ramón, M. A., Terol-Villalobos, I. R., Osornio-Rios, R. A., & Morales-Hernandez, L. A. (2019). Methodology for filtering of depth maps based on the MCbR algorithm supported by color, shape and neighboring features. Signal Processing: Image Communication, 70, 220-232.*
Factor de impacto: 2.073.
- 2018 *Garduño-Ramón, M. A., Reséndiz-Ochoa, E., Morales-Hernández, L. A., & Osornio-Rios, R. A. (2019). Novel method for wind turbines blades damage classification using image processing. Journal of scientific and Industrial Research (JSIR), #(#), #.*
Factor de impacto: 0.534.
- 2017 *Garduño-Ramón, M. A., Vega-Mancilla, S. G., Morales-Hernández, L. A., & Osornio-Rios, R. A. (2017). Supportive noninvasive tool for the diagnosis of breast cancer using a thermographic camera as sensor. Sensors, 17(3), 497.*
Factor de impacto: 2.475.
- 2017 *Garduño-Ramón, M. A., Terol-Villalobos, I. R., Osornio-Rios, R. A., & Morales-Hernandez, L. A. (2017). A new method for inpainting of depth maps from time-of-flight sensors based on a modified closing by reconstruction algorithm. Journal of Visual Communication and Image Representation, 47, 36-47.*
Factor de impacto: 1.836.

Artículos de congreso

-
- 2017 *Alvarado-Robles, G., Terol-Villalobos, I. R., Garduño-Ramon, M. A., & Morales-Hernandez, L. A. (2017, September). Segmentation of Green Areas Using Bivariate Histograms Based in Hue-Saturation Type Color Spaces. In International Conference on Image Analysis and Processing (pp. 277-287). Springer, Cham*
- 2014 *Garduño-Ramón, M. A., Sánchez-Gómez, J. I., Morales Hernández, L. A., Benítez-Rangel, J. P., & Osornio-Rios, R. A. (2015). Methodology for automatic detection of trees and shrubs in aerial pictures from UAS. Congreso Internacional de Ingeniería 11-15 de Mayo de 2015. Querétaro México.*

8. APÉNDICE II

Código de la Cerradura por Reconstrucción Modificada (MCbR)

```
//  
// main.cpp  
// image_processing  
//  
// Created by Marco Garduño on 03/11/17.  
// Copyright © 2017 Marco Garduño. All rights reserved.  
//  
  
#include <iostream>  
#include <opencv2/highgui/highgui.hpp>  
#include "functions.h"  
#include "mcbR.h"  
  
int main(){  
    cv::Mat src;  
  
    std::cout << "MCbR" << "\n";  
    src = cv::imread("images/cones-disp2.png", 1);  
    src = rgb2gray(src);  
  
    if(!src.data){  
        std::cout << "Error...\n";  
    }  
  
    cv::Mat filtered = mcbR(src);  
  
    cv::namedWindow("Original", CV_WINDOW_AUTOSIZE);  
    cv::imshow("Original", src);  
    cv::namedWindow("Filtered", CV_WINDOW_AUTOSIZE);  
    cv::imshow("Filtered", filtered);  
  
    cv::imwrite("filteredMCbR.png", filtered);  
  
    cv::waitKey();  
  
    return 0;  
}
```

Código: main.cpp

```

//
// functions.cpp
// image_processing
//
// Created by Marco Garduño on 24/07/17.
// Copyright © 2017 Marco Garduño. All rights reserved.
//

#include <iostream>
#include <vector>
#include <opencv2/highgui/highgui.hpp>
#include "functions.h"

cv::Mat rgb2gray(cv::Mat image){
    cv::Mat img = cv::Mat::zeros(image.rows, image.cols, CV_8UC1);
    uchar gray;
    uchar* pixel;

    for (int j = 0; j < img.rows; ++j){
        cv::Vec3b* pixel_rgb = image.ptr<cv::Vec3b>(j);
        pixel = img.ptr<uchar>(j);
        for (int i = 0; i < img.cols; ++i){
            gray = 0.299*pixel_rgb[i][2] + 0.587*pixel_rgb[i][1] +
                0.114*pixel_rgb[i][0];
            pixel[i] = gray;
        }
    }

    return img;
}

cv::Mat negative_gray(cv::Mat image){
    cv::Mat img = image.clone();
    uchar* pixel;

    for (int j = 0; j < img.rows; ++j){
        pixel = img.ptr<uchar>(j);
        for (int i = 0; i < img.cols; ++i){
            pixel[i] = 255 - pixel[i];
        }
    }
}

```

```

    return img;
}

cv::Mat difference(cv::Mat image1, cv::Mat image2){
    cv::Mat img;

    std::vector<std::vector<int>> matrix1 = mat2vector(image1);
    std::vector<std::vector<int>> matrix2 = mat2vector(image2);

    for(int j = 0; j < image1.rows; j++){
        for(int i = 0; i < image1.cols; i++){
            matrix1[j][i] = matrix1[j][i] - matrix2[j][i];
        }
    }

    img = vector2mat(matrix1);
    return img;
}

std::vector<std::vector<int>> mat2vector(cv::Mat image){
    std::vector<std::vector<int>> myVector;
    myVector.resize(image.rows, std::vector<int>(image.cols, 0));
    uchar* pixel;

    for (int j = 0; j < image.rows; ++j){
        pixel = image.ptr<uchar>(j);
        for (int i = 0; i < image.cols; ++i){
            myVector[j][i] = pixel[i];
        }
    }

    return myVector;
}

cv::Mat vector2mat(std::vector<std::vector<int>> vector){
    cv::Mat image = cv::Mat::zeros(vector.size(), vector[0].size(), CV_8UC1);
    uchar* pixel;
    for (int j = 0; j < vector.size(); ++j){
        pixel = image.ptr<uchar>(j);
        for (int i = 0; i < vector[0].size(); ++i){
            pixel[i] = vector[j][i];
        }
    }
}

```

```
    return image;
}
```

Código: functions.cpp

```
//
// mabr.cpp
// image_processing
//
// Created by Marco Garduño on 31/10/17.
// Copyright © 2017 Marco Garduño. All rights reserved.
//

#include <iostream>
#include <vector>
#include <opencv2/highgui/highgui.hpp>
#include "mabr.h"
#include "morphology.h"
#include "functions.h"

int min_no_zero(std::vector<int> list){
    int min = 255;

    if(list[0] < 255 && list[0] > 0){
        min = list[0];
    }
    if(list[1] < 255 && list[1] > 0){
        min = list[1];
    }

    return min;
}

int noise_counter(cv::Mat image){
    std::vector<std::vector<int>> matrix_aux = mat2vector(image);
    int counter = 0;

    for(int j = 0; j < image.rows; j++){
        for(int i = 0; i < image.cols; i++){
            if(matrix_aux[j][i] == 0){
                counter++;
            }
        }
    }
}
```

```

    return counter;
}

cv::Mat comparator(cv::Mat f, cv::Mat g){
    std::vector<std::vector<int>> matrix_aux = mat2vector(f);
    std::vector<std::vector<int>> matrix_f = mat2vector(f);
    std::vector<std::vector<int>> matrix_g = mat2vector(g);

    for(int j = 0; j < f.rows; j++){
        for(int i = 0; i < f.cols; i++){
            if(matrix_f[j][i] > 0){
                matrix_aux[j][i] = matrix_f[j][i];
            }
            else{
                matrix_aux[j][i] = matrix_g[j][i];
            }
        }
    }

    cv::Mat mat_aux = vector2mat(matrix_aux);

    return mat_aux;
}

cv::Mat geodesic_dilationM(cv::Mat I, cv::Mat J){
    std::vector<std::vector<int>> matrix_I = mat2vector(I);
    std::vector<std::vector<int>> matrix_J = mat2vector(J);
    std::vector<std::vector<int>> matrix_aux = matrix_I;
    std::vector<int> list_1, list_2;

    cv::Mat mat_aux;
    cv::Mat dif;

    bool flag = true;
    double maxVal;

    while(flag){
        matrix_aux = matrix_J;

        for(int j = 1; j < I.rows; j++){
            for(int i = 1; i < I.cols-1; i++){

```

```

        list_1 = { matrix_J[j-1][i-1], matrix_J[j-1][i], matrix_J[j-
1][i+1],
                matrix_J[j][i-1], matrix_J[j][i]};
        auto max_ptr_1 = std::max_element(std::begin(list_1),
std::end(list_1));
        matrix_J[j][i] = min_no_zero(*max_ptr_1, matrix_I[j][i]);
    }
}

for(int j = I.rows-2; j >= 0 ; j--){
    for(int i = I.cols-2; i >= 1; i--){
        list_2 = {
matrix_J[j][i],
matrix_J[j][i+1],
matrix_J[j+1][i-1], matrix_J[j+1][i],
matrix_J[j+1][i+1]};
        auto max_ptr_2 = std::max_element(std::begin(list_2),
std::end(list_2));
        matrix_J[j][i] = min_no_zero(*max_ptr_2, matrix_I[j][i]);
    }
}

J = vector2mat(matrix_J);
mat_aux = vector2mat(matrix_aux);
dif = difference(mat_aux, J);

double min, max;
cv::minMaxLoc(dif, &min, &max);

if(max == 0){
    flag = false;
}
}

return J;
}

cv::Mat geodesic_erosionM(cv::Mat I, cv::Mat J){
    I = negative_gray(I);
    J = negative_gray(J);

    J = geodesic_dilationM(I, J);

    I = negative_gray(I);

```

```

    J = negative_gray(J);

    return J;
}

cv::Mat closing_by_reconstructionM(cv::Mat image, int lambda){
    cv::Mat mat_aux = image.clone();
    cv::Mat Y = image.clone();
    cv::Mat dilated = image.clone();
    cv::Mat J;

    Y = dilation(image, lambda);
    J = geodesic_erosionM(mat_aux, Y);

    return J;
}

cv::Mat mcbn(cv::Mat image){
    cv::Mat mat_aux = image.clone();
    cv::Mat filtered = image.clone();
    int noise, size;

    size = 1;

    noise = noise_counter(image);
    std::cout << noise << std::endl;

    mat_aux = closing(image, 1);

    noise = noise_counter(mat_aux);
    std::cout << noise << std::endl;

    while(noise != 0){
        mat_aux = closing_by_reconstructionM(mat_aux, size);
        filtered = comparator(mat_aux, filtered);

        noise = noise_counter(mat_aux);
        std::cout << noise << std::endl;

        size += 1;
    }

    return mat_aux;
}

```

```
}
```

Código: mcbr.cpp

```
//
```

```
// morphology.cpp
// image_processing
//
// Created by Marco Garduño on 24/07/17.
// Copyright © 2017 Marco Garduño. All rights reserved.
//

#include <iostream>
#include <vector>
#include <opencv2/highgui/highgui.hpp>
#include "morphology.h"
#include "functions.h"
#include <queue>

cv::Mat dilation(cv::Mat image, int lambda){
    cv::Mat img = image.clone();
    std::vector<std::vector<int>> matrix = mat2vector(img);

    for(int k = 0; k < lambda; k++){
        // B1
        for(int j = 0; j < image.rows; j++){
            for(int i = 0; i < image.cols-1; i++){
                if(matrix[j][i] < matrix[j][i+1]){
                    matrix[j][i] = matrix[j][i+1];
                }
            }
        }
    }

    // B2
    for(int j = 0; j < image.rows-1; j++){
        for(int i = 0; i < image.cols; i++){
            if(matrix[j][i] < matrix[j+1][i]){
                matrix[j][i] = matrix[j+1][i];
            }
        }
    }

    // B3
    for(int j = 0; j < image.rows; j++){
```

```

    for(int i = image.cols - 1; i > 0; i--){
        if(matrix[j][i] < matrix[j][i-1]){
            matrix[j][i] = matrix[j][i-1];
        }
    }
}

// B4
for(int j = image.rows - 1; j > 0; j--){
    for(int i = 0; i < image.cols; i++){
        if(matrix[j][i] < matrix[j-1][i]){
            matrix[j][i] = matrix[j-1][i];
        }
    }
}

}

img = vector2mat(matrix);

return img;
}

cv::Mat erosion(cv::Mat image, int lambda){
    cv::Mat img = image.clone();

    img = negative_gray(img);
    img = dilation(img, lambda);
    img = negative_gray(img);

    return img;
}

cv::Mat closing(cv::Mat image, int lambda){
    cv::Mat img = image.clone();

    img = dilation(img, lambda);
    img = erosion(img, lambda);

    return img;
}

```

Código: morphology.cpp

```
//
// functions.h
// image_processing
//
// Created by Marco Garduño on 24/07/17.
// Copyright © 2017 Marco Garduño. All rights reserved.
//

#ifdef FUNCTIONS_H_INCLUDED
#define FUNCTIONS_H_INCLUDED

cv::Mat rgb2gray(cv::Mat image);
cv::Mat negative_gray(cv::Mat image);
cv::Mat difference(cv::Mat image1, cv::Mat image2);

std::vector<std::vector<int>> mat2vector(cv::Mat image);
cv::Mat vector2mat(std::vector<std::vector<int>> vector);

#endif /* functions_h */
```

Código: functions.h

```
//
// mabr.h
// image_processing
//
// Created by Marco Garduño on 31/10/17.
// Copyright © 2017 Marco Garduño. All rights reserved.
//

#ifdef MCBR_H_INCLUDED
#define MCBR_H_INCLUDED

int min_no_zero(std::vector<int> list);
int noise_counter(cv::Mat image);
cv::Mat comparator(cv::Mat f, cv::Mat g);
cv::Mat geodesic_dilationM(cv::Mat I, cv::Mat J);
cv::Mat geodesic_erosionM(cv::Mat I, cv::Mat J);
cv::Mat closing_by_reconstructionM(cv::Mat image, int lambda);
cv::Mat mabr(cv::Mat image);

#endif /* mabr_h */
```

Código: mabr.h

```
//  
// morphology.h  
// image_processing  
//  
// Created by Marco Garduño on 24/07/17.  
// Copyright © 2017 Marco Garduño. All rights reserved.  
//  
  
#ifndef MORPHOLOGY_H_INCLUDED  
#define MORPHOLOGY_H_INCLUDED  
  
cv::Mat dilation(cv::Mat image, int lambda);  
cv::Mat erosion(cv::Mat image, int lambda);  
cv::Mat closing(cv::Mat image, int lambda);  
  
#endif /* morphology_h */
```

Código: morphology.h