

**IMPLEMENTACIÓN DE UN SISTEMA AUTOMÁTICO DE IDENTIFICACIÓN Y  
CARACTERIZACIÓN DE FALLAS PARA UNA PLATAFORMA ROBÓTICA DE  
EXPLORACIÓN Y REPARACIÓN DE TUBERÍAS HIDROSANITARIAS MEDIANTE  
VISIÓN POR COMPUTADOR.**

FERNANDO NICOLAS VARGAS PLAZAS

**TRABAJO DE GRADO DE PROFUNDIZACIÓN DE MAESTRÍA PARA OPTAR POR EL  
TÍTULO DE MAGISTER EN INGENIERÍA ELECTRÓNICA**

**DIRECTOR**

ING. MANUEL RICARDO PÉREZ CERQUERA Ph.D

PONTIFICIA UNIVERSIDAD JAVERIANA  
FACULTAD DE INGENIERIA  
DEPARTAMENTO DE ELECTRÓNICA  
BOGOTÁ DC, 2014

**PONTIFICIA UNIVERSIDAD JAVERIANA  
FACULTAD DE INGENIERIA  
MAESTRÍA EN INGENIERIA ELECTRONICA**

RECTOR MAGNÍFICO: PADRE JORGE HUMBERTO PELÁEZ PIEDRAHITA S.J

DECANO ACADÉMICO: ING. JORGE LUIS SANCHEZ TELLEZ M.Sc

DECANO DEL MEDIO UNIVERSITARIO: PADRE ANTONIO JOSÉ SARMIENTO NOVA S.J

DIRECTOR DE MAESTRÍA: ING. CESAR LEONARDO NIÑO BARRERA Ph.D

DIRECTOR DE PROYECTO: ING. MANUEL RICARDO PÉREZ CERQUERA Ph.D

## **ARTÍCULO 23 DE LA RESOLUCIÓN N° 13 DE JUNIO DE 1946**

“La universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado.

Sólo velará porque no se duplique nada contrario al dogma y la moral católica y porque los trabajos no contengan ataques o polémicas puramente personales. Antes bien, que se vea en ellos el anhelo de buscar la verdad y la justicia”.

# ÍNDICE GENERAL

<b>1. INTRODUCCIÓN.....</b>	<b>8</b>
1.1 OBJETIVO GENERAL Y OBJETIVOS ESPECÍFICOS.....	9
1.1.1 OBJETIVO GENERAL.....	9
1.1.2 OBJETIVOS ESPECIFICOS.....	9
<b>2. MARCO TEÓRICO.....</b>	<b>10</b>
2.1 LUZ ESTRUCTURADA.....	10
2.1.1 CALIBRACIÓN.....	11
2.2 PROCESAMIENTO DIGITAL DE IMÁGENES.....	11
2.2.1 REPRESENTACIÓN DE IMÁGENES Y MODELAMIENTO.....	11
2.2.2 MEJORAMIENTO DE IMÁGENES.....	12
2.2.3 ANÁLISIS DE IMÁGENES.....	12
2.2.4 RECONSTRUCCIÓN DE IMÁGENES POR PROYECCIONES.....	12
2.3 RECONOCIMIENTO DE PATRONES.....	12
2.4 OPENCV (OPEN SOURCE COMPUTER VISION LIBRARY).....	13
<b>3. ESPECIFICACIONES.....</b>	<b>14</b>
3.1 ENTRADAS Y SALIDAS DEL SISTEMA.....	14
<b>4. DESARROLLOS.....</b>	<b>15</b>
4.1 DESCRIPCIÓN GENERAL Y DIAGRAMA EN BLOQUES.....	15
4.2 FALLAS ESTRUCTURALES.....	17
4.3 ADQUISICIÓN DE VIDEOS.....	19
4.3.1 ESPECIFICACIONES ELEMENTOS DE ADQUISICIÓN.....	19
4.3.2 MONTAJE DE PROTOCOLO DE ADQUSICIÓN.....	19
4.4 MODELAMIENTO Y CALIBRACIÓN DE CÁMARA.....	21
4.4.1 MODELO DE ZHANG.....	21
4.4.2 MÉTODO DE CALIBRACIÓN DE ZHANG.....	23
4.4.3 ALGORITMO DE CALIBRACIÓN DE CÁMARA.....	26
4.5 MODELAMIENTO Y CALIBRACIÓN DEL PROYECTOR.....	28
4.5.1 MODELO DE PROYECTOR DE ZHANG.....	28
4.5.2 CALIBRACIÓN DEL PROYECTOR.....	29
4.6 INTERSECCIÓN DE PATRONES.....	30
4.6.1 CÓMPUTO DE MAPA DE DISTORSIÓN.....	31
4.6.2 ELIMINACIÓN DE DISTORSIÓN.....	31
4.6.3 GENERACIÓN DE MAPA FINAL.....	33
4.6.3.1 ALGORITMO DE KONOLIGE.....	33
4.7 ETAPA DE FILTRADO.....	35
4.7.1 FILTRO <i>SHARPENER</i> .....	35
4.7.2 FILTRO MEDIANA.....	36

4.8	ANÁLISIS DE TEXTURAS.....	37
4.8.1	MATRICES DE CO-OCURRENCIA.....	37
4.8.2	EXTRACCIÓN DE DESCRIPTORES DE TEXTURA.....	41
4.8.1.1	ENERGIA.....	41
4.8.1.2	ENTROPÍA.....	41
4.8.1.3	CONTRASTE.....	41
4.8.1.4	CORRELACIÓN.....	41
4.8.1.5	VARIANZA.....	42
4.8.1.6	MOMENTO DE DIFERENCIA INVERSA (HOMOGENEIDAD).....	42
4.8.1.7	SUMA DE PROMEDIO.....	42
4.8.1.8	SUMA DE VARIANZA.....	42
4.8.1.9	DISIMILARIDAD.....	43
4.8.1.10	DIFERENCIA DE VARIANZAS.....	43
4.8.1.11	SUMA DE ENTROPÍAS.....	43
4.8.1.12	MEDIDA DE INFORMACIÓN DE CORRELACIÓN 1.....	43
4.8.1.13	MEDIDA DE INFORMACIÓN DE CORRELACIÓN 2.....	43
4.8.3	EXTRACCIÓN DE DESCRIPTORES GEOMÉTRICOS.....	43
4.9	CLASIFICACIÓN.....	44
4.9.1	MAQUINA DE SOPORTE VECTORIAL (SVM).....	45
4.9.1.1	KERNEL LINEAL.....	45
4.9.1.2	KERNEL NO LINEALES.....	46
4.9.2	ENTRENAMIENTO CLASIFICADOR.....	47
4.9.2.1	ENTRENAMIENTO MULTICLASE.....	47
4.9.2.2	ELECCIÓN DE PARÁMETROS DE ENTRENAMIENTO.....	48
4.10	TEST DE CLASIFICADOR E INTERFAZ FINAL.....	50
<b>5.</b>	<b>ANÁLISIS Y RESULTADOS.....</b>	<b>52</b>
5.1	RESULTADOS Y ANÁLISIS DE CALIBRACIÓN.....	52
5.2	EJEMPLO DE EVALUACIÓN DE VECTOR DE CARACTERÍSTICAS.....	56
5.3	EVALUACIÓN DE DESEMPEÑO DE CLASIFICADORES.....	60
5.3.1	EVALUACIÓN DE RENDIMIENTO ETAPA DE ENTRENAMIENTO.....	61
5.3.2	EVALUACIÓN DE RENDIMIENTO ETAPA DE TEST.....	66
5.3.3	ANÁLISIS DE ERRORES ENCONTRADOS.....	72
5.3.3.1	PÉRDIDA DE CONTRASTE EN PATRONES PROYECTADOS.....	73
5.3.3.2	MOVIMIENTOS BRUSCOS DE CÁMARA.....	74
5.3.3.3	FALTA DE PATRÓN EN VIDEO ADQUIRIDO.....	74
5.3.3.4	FALLAS POCO VISIBLES EN <i>FRAMES</i> .....	75
5.3.3.5	ERRORES DE CLASIFICACIÓN.....	75
5.4	ESTUDIO DE COSTOS.....	76
5.5	TRABAJOS E INVESTIGACIONES A SEGUIR.....	76
<b>6.</b>	<b>CONCLUSIONES.....</b>	<b>77</b>
	<b>BIBLIOGRAFÍA.....</b>	<b>79</b>
	<b>ANEXOS.....</b>	<b>82</b>

## ÍNDICE DE FIGURAS

Figura 1. Ejemplo de distribución de elementos en detección por luz estructurada.....	10
Figura 2. Etapas de un proceso de reconocimiento de patrones.....	12
Figura 3. Logo OpenCV.....	13
Figura 4. Diagrama en bloques de sistema de detección.....	15
Figura 5. Tubería en PVC para pruebas.....	17
Figura 6. Región sin falla.....	18
Figura 7. Rotura.....	18
Figura 8. Fisura.....	18
Figura 9. Grieta.....	18
Figura 10. Montaje de adquisición.....	19
Figura 11. a) Patrón positivo b) Patrón negativo.....	20
Figura 12. Efecto de distorsión radial.....	22
Figura 13. Pasos de modelamiento de cámara de Zhang.....	23
Figura 14. Toma de imágenes de tablero de ajedrez en diferentes orientaciones para calibración de cámara.....	26
Figura 15. Diagrama de flujo de calibración de cámara.....	26
Figura 16. Imágenes de calibración de cámara.....	27
Figura 17. Esquinas detectadas sobre tablero de ajedrez.....	27
Figura 18. Imágenes de calibración de proyector.....	30
Figura 19. a) Problema de correspondencia en triangulación de visión estéreo. b) Correspondencias en triangulación de luz estructurada.....	31
Figura 20. Diagrama de flujo de superposición de patrones por luz estructurada.....	31
Figura 21. a) Imagen de calibración con distorsión de cámara. b) Imagen de calibración sin distorsion de cámara.....	32
Figura 22. a) Imagen al interior de tubería con distorsión de todo el sistema. b) Imagen al interior de tubería sin distorsion de todo el sistema.....	32
Figura 23. a) Imagen con patrón positivo. b) Imagen con patrón negativo.....	34
Figura 24. Mapa generado.....	35
Figura 25. Ejemplo convolución entre máscara de mejoramiento de contraste e imagen.....	36
Figura 26. Ejemplo filtro mediana.....	36
Figura 27. a) Mapa sin filtrar. b) Mapa filtrado.....	37
Figura 28. a) Imagen con cuatro niveles de grises. b) Matriz de co-ocurrencia calculada.....	38
Figura 29. Matriz de co-ocurrencia simétrica.....	38
Figura 30. Ejemplo de matriz de co-ocurrencia final.....	39
Figura 31. a) Diagrama de flujo matriz de co-ocurrencia a 0°. b) Diagrama de flujo matriz de co-ocurrencia a 45°. c) Diagrama de flujo matriz de co-ocurrencia a 90°. d) Diagrama de flujo matriz de co-ocurrencia a 135°.....	40
Figura 32. Diagrama de flujo de extracción de característica geométrica.....	44
Figura 33. a) Contornos de muestra no falla b) Contornos muestra rotura c) Contornos muestra fisura d) Contornos muestra grieta.....	44
Figura 34. Hiperplano de separación óptimo de clasificador SVM.....	46
Figura 35. Arbol binario de discriminación de clases por SVM.....	47
Figura 36. Diagrama de flujo entrenamiento SVM.....	48
Figura 37. Cambio de tasa de error de entrenamiento con parámetros C y $\gamma$ .....	49
Figura 38. Tasas de error de entrenamiento de distintas combinaciones (C, $\gamma$ ) con el método de búsqueda en grilla.....	49
Figura 39. Diagrama de flujo de interfaz final.....	50
Figura 40. Grieta visualizada en <i>frame</i> .....	51
Figura 41. Comparación errores de reproyección para enfoque manual y automático.....	53

Figura 42. a) Error de reproyección para cada punto en enfoque manual b) Error de reproyección para cada punto en enfoque automático.....	54
Figura 43. Errores de reproyección en imágenes de calibración del proyector.....	55
Figura 44. Error de reproyección para cada punto.....	56
Figura 45. a) Muestras de grietas b) Muestras de roturas c) Muestras de fisuras d) Muestras de no fallas.....	56
Figura 46. Siete Características de textura de muestras de ejemplo sin etapa de filtrado.....	57
Figura 47. Seis Características de textura de muestras de ejemplo sin etapa de filtrado.....	58
Figura 48. Siete Características de textura de muestras de ejemplo con etapa de filtrado.....	59
Figura 49. Seis Características de textura de muestras de ejemplo con etapa de filtrado.....	60
Figura 50. Variación de verdaderos positivos al cambiar el parámetro C.....	63
Figura 51. Variación de verdaderos positivos al cambiar el parámetro C.....	63
Figura 52. Curva ROC de resultados de entrenamiento de clasificadores.....	64
Figura 53. Curva ROC con mejor clasificador.....	65
Figura 54. Curva ROC con clasificadores según tipo de falla.....	65
Figura 55. Ejemplos de <i>frames</i> de videos de prueba.....	67
Figura 56. Ejemplos de detecciones de fallas.....	68
Figura 57. Curva ROC para videos de prueba.....	69
Figura 58. Curva ROC de videos de prueba para grietas.....	70
Figura 59. Curva ROC de videos de prueba para roturas.....	71
Figura 60. Curva ROC de videos de prueba para fisuras.....	72
Figura 61. Fuentes de error en la clasificación.....	73
Figura 62. Par de muestras para superposición donde una resulta afectada por la pérdida de contraste en patrón proyectado.....	73
Figura 63. a) Grieta detectada correctamente. b) Grieta no detectada por problema de pérdida de contraste.....	73
Figura 64. a) Grieta detectada como fisura b) Grieta no detectada y falso positivo detectado.....	74
Figura 65. Falso positivo detectado en sección con falta de patrón.....	75
Figura 66. Grieta no detectada en la parte inferior del <i>frame</i> .....	75

## ÍNDICE DE TABLAS

Tabla 1. Entradas y salidas del sistema.....	14
Tabla 2. Especificaciones cámara.....	19
Tabla 3. Especificaciones video beam.....	19
Tabla 4. Ejemplo combinaciones de niveles de grises para matriz de co-ocurrencia.....	38
Tabla 5. Parámetros de entrenamiento SVM.....	49
Tabla 6. Error de reproyección para cada imagen de calibración con enfoque manual y automático.....	53
Tabla 7. Error de reproyección para cada imagen de calibración del proyector.....	55
Tabla 8. Vector de características para muestras sin etapa de filtrado.....	57
Tabla 9. Vector de características para muestras con etapa de filtrado.....	58
Tabla 10. Número de muestras para cada clase.....	61
Tabla 11. Resultados de entrenamiento para clasificadores SVM.....	62
Tabla 12. Mediciones de rendimiento en entrenamiento para clasificadores SVM.....	62
Tabla 13. Resultados de entrenamiento para tipos de falla.....	65
Tabla 14. Videos de prueba final.....	66
Tabla 15. Mediciones de rendimiento de clasificación de videos de prueba fallas vs no fallas.....	68
Tabla 16. Relación de clases de fallas para cada video.....	69
Tabla 17. Mediciones de rendimiento de clasificación de videos con grietas.....	70
Tabla 18. Mediciones de rendimiento de clasificación de videos con roturas.....	71
Tabla 19. Mediciones de rendimiento de clasificación de videos con fisuras.....	71
Tabla 20. Mediciones de rendimiento de clasificación de videos de prueba fallas vs no fallas.....	74
Tabla 21. Costos reales de los recursos utilizados para la realización del Trabajo de Grado.....	76

## ÍNDICE DE SIGLAS Y ABREVIATURAS

PVC: Policloruro de vinilo  
AVI: Audio Video Interleave  
JPG: Joint Photographic Experts Group  
SVM: Support Vector Machine  
ROI: Region of Interest  
CMOS: Complementary Metal Oxide Semiconductor  
DLP: Digital Light Processing  
SAD: Sum of Absolute Differences  
CCD: Charge-Coupled Device  
DOV: Difference of Variances  
SOA: Sum of Average  
SOV: Sum of Variance  
SOE: Sum of Entropy  
IDM: Inverse Difference Moment  
IMC: Information Measure of Correlation  
RBF: Radial Basis Function  
KKT: Karush-Kuhn-Tucker  
VP: Verdaderos Positivos  
VN: Verdaderos Negativos  
FP: Falsos Positivos  
FN: Falsos Negativos  
VPR: Razón de Verdaderos Positivos  
FPR: Razón De Falsos Positivos  
ACC: Accuracy  
SPC: Specificity  
PPV: Valor Predictivo Positivo  
NPV: Valor Predictivo Negativo  
FDR: Razón de Falsos Descubrimientos  
[fps]: *Frames* por segundo  
[m]: Metro  
[s]: Segundo



# CAPÍTULO 1

## INTRODUCCIÓN

Las exploraciones y reparaciones en tuberías hidrosanitarias mediante sistemas de inspección móvil operadas de forma remota son uno de los retos a desarrollar en los últimos años en éste sector, con el fin de optimizar el uso de un recurso vital como lo es el agua. Esto implica que el sistema debe estar en la capacidad de visualizar las posibles fallas que pueda presentar la tubería que se está inspeccionando y lograr repararlas de forma óptima, asegurando el buen funcionamiento de cada uno de estos elementos, [1], [2], [3].

Tradicionalmente, la detección y clasificación de fallas o defectos en las tuberías son realizadas de forma manual por un operario que mediante su conocimiento de experto en ésta área, realiza una interpretación visual en los videos que genera el robot móvil, con el fin de determinar en qué sección del tubo se encuentra una falla que genera filtraciones de agua o problemas en el transporte de la misma. Este método genera varios problemas y desventajas, como lo son: la dependencia de los conocimientos del operario, la ineficiencia del mismo para evaluar videos con una tasa de *frames* muy alta, lo que a la vez genera cansancio en la persona que finalmente conlleva a una detección más lenta de las fallas y en ocasiones una escogencia de forma subjetiva, causando errores en la detección [4].

Con el fin de mejorar la efectividad de la inspección de tuberías hidrosanitarias, en los últimos años se han propuesto sistemas automáticos y semi-automáticos de detección de fallas mediante técnicas de procesamiento de imágenes y reconocimiento de patrones [4], [6], [13], lo que produce ventajas en el procedimiento realizado, tales como la coherencia con las fallas detectadas en un intervalo de tiempo y mayor eficacia en la identificación de los defectos en la tubería.

El trabajo de grado a realizar cambia la perspectiva tradicional de detección de fallas y defectos en tuberías hidrosanitarias y se enfoca en realizar un sistema automático de identificación y caracterización de fallas utilizando técnicas de procesamiento de imágenes y reconocimiento de patrones.

El sistema consta de una base de datos conformada por *frames*<sup>1</sup> de videos adquiridos con ayuda del método de luz estructurada, el cual permite generar un mapa de forma tal que los defectos a clasificar tengan mayor relevancia en los *frames* adquiridos. Posterior a ello, estos *frames* pasan por unas etapas de preprocesamiento con la finalidad de resaltar características importantes de las fallas y eliminar componentes no deseados de la adquisición.

Por último, se realiza la extracción de las características necesarias y relevantes para clasificar las muestras y determinar si alguna sección de la tubería posee un defecto estructural a través de un algoritmo de clasificación por aprendizaje supervisado [8], generando como resultado final una interfaz que visualice estos defectos, indicando el tipo de falla presente en cada sección por medio del software OPENCV.

El documento presenta en el capítulo 2 una recopilación del marco teórico de los temas más relevantes del proyecto. En el capítulo 3 se presentan las especificaciones, los requerimientos básicos y las operaciones que deben ser realizadas por el sistema de detección para la clasificación correcta de las fallas y en el capítulo 4 se realiza la descripción general del sistema propuesto, así como los desarrollos realizados de cada una de las partes que componen el diseño del sistema implementado en C++ usando la biblioteca OPENCV. Finalmente, en el capítulo 5 se presentan los resultados y análisis obtenidos, teniendo en cuenta el proceso de adquisición de las imágenes con el método de luz estructurada, el proceso de programación e implementación de los algoritmos utilizados, así como las pruebas de rendimiento del proceso de clasificación.

### 1.1 OBJETIVO GENERAL Y OBJETIVOS ESPECÍFICOS

#### 1.1.1 OBJETIVO GENERAL

Proponer un sistema de detección de fallas en videos de tuberías hidrosanitarias mediante técnicas de procesamiento de imágenes y reconocimiento de patrones.

#### 1.1.2 OBJETIVOS ESPECIFICOS

- Construir una base de datos a partir de videos al interior de tuberías hidrosanitarias teniendo en cuenta aspectos de iluminación y resolución.
- Identificar y determinar las características relevantes con las cuales se realiza una clasificación exitosa de fallas encontradas en tuberías.
- Implementar un algoritmo de reconocimiento de patrones a partir de técnicas existentes que permita identificar fallas estructurales en una tubería hidrosanitaria.
- Realizar una caracterización y clasificación de las fallas detectadas en el algoritmo por medio de características geométricas de las mismas.
- Evaluar y comparar el rendimiento de las técnicas de clasificación que se adapten a las condiciones ambientales y técnicas de la inspección móvil, en el algoritmo desarrollado.

---

<sup>1</sup> Frame es la notación en inglés utilizada para hacer referencia a un fotograma o cuadro particular dentro de una secuencia de imágenes que forma un video determinado [7].

# CAPÍTULO 2

## MARCO TEÓRICO

### 2.1. LUZ ESTRUCTURADA:

Un proceso de adquisición de imágenes por luz estructurada consiste en proyectar un haz o configuración de luz sobre un determinado objeto junto con una cámara, de tal forma que el haz proyecta un patrón de líneas o puntos mientras la cámara adquiere las deformaciones de este [5].

Para un proceso de adquisición de imágenes por luz estructurada es indispensable realizar un procedimiento de calibración, debido a las posibles distorsiones que se pueden presentar por cuestiones de perspectiva u óptica y por otro lado, se deben conocer los parámetros extrínsecos (traslación, rotación) e intrínsecos (distancia focal, punto principal, etc.) con los cuales se puede realizar la reconstrucción del entorno de la imagen y hacer un correcto análisis de los patrones generados. Para realizar esta calibración, se tiene en cuenta la distancia al objetivo y la distancia entre el haz de luz y la cámara como se puede apreciar en la figura 1:

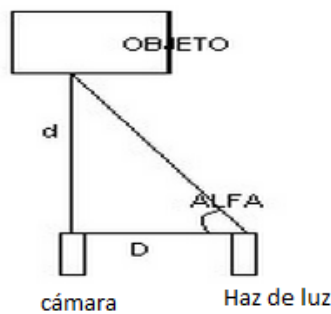


Figura 1. Ejemplo de distribución de elementos en detección por luz estructurada

### 2.1.1 CALIBRACIÓN:

El objetivo de la calibración de una cámara consiste en realizar la medida de distancias en un mundo real, que en este caso sería la escena en estudio, desde sus proyecciones en una imagen.

Algunas de las aplicaciones comunes en las cuales es necesario realizar un proceso de calibración previo al procesamiento de la imagen son los siguientes:

- **RECONSTRUCCIÓN DENSA:** Obtener la posición de los puntos 3-D de una escena por medio de las proyecciones 2-D obtenidas en la imagen.
- **INSPECCIÓN VISUAL:** Comparar las imágenes con un modelo con el fin de determinar fallas o aspectos inusuales de un objeto determinado.
- **LOCALIZACIÓN DE LA CÁMARA:** La posición y orientación de la cámara se pueden calcular mediante la localización de algunos puntos de referencia conocidos en la escena [6].

El proceso de calibración de una cámara consta de dos fases, las cuales se describen a continuación.

- 1) Modelamiento matemático de la cámara de acuerdo a su comportamiento físico y óptico, usando un conjunto de parámetros.
- 2) Uso de métodos iterativos para estimar los valores de esos parámetros, los cuales se dividen en parámetros extrínsecos e intrínsecos. Los parámetros extrínsecos miden la posición y orientación de la cámara respecto a un sistema de coordenadas (Rotación y traslación). Por otro lado, los parámetros intrínsecos modelan la geometría interna de la cámara y las características ópticas del sensor (Distancia focal, punto principal, tamaño de pixel, asimetría, etc.) [6].

### 2.2 PROCESAMIENTO DIGITAL DE IMÁGENES:

El análisis y procesamiento digital de imágenes consiste en procesar una señal de dos dimensiones por medio de un computador, lo cual implica que se deben analizar imágenes como arreglos de números reales o complejos representados por un número finito de bits. Partiendo de este contexto, existen varias aplicaciones y problemas que pueden ser considerados bajo las siguientes clases de procesamiento [7]:

#### 2.2.1 REPRESENTACIÓN DE IMÁGENES Y MODELAMIENTO:

Este tipo de procesamiento implica la caracterización de cada pixel de la imagen procesada. Una imagen puede representar luminancia de objetos en una escena determinada, perfiles de temperatura en una región o características internas de un ser humano. Una consideración importante en la representación de la imagen son los criterios para medir la calidad de la misma y el rendimiento de la técnica de procesamiento de usar para realizar el modelamiento, tales como contraste, color, frecuencia espacial, etc. A partir de ello, se consideran tres tipos de modelamiento y representación, que contienen distintas técnicas para lograrlo:

- **MODELOS DE PERCEPCIÓN:** Percepción visual de contraste, frecuencia espacial y color, percepción temporal, percepción de escena.

- **MODELOS LOCALES:** Muestreo y reconstrucción, modelos determinísticos, expansión en series y transformadas unitarias, modelos estadísticos.
- **MODELOS GLOBALES:** Modelos de inteligencia artificial, modelos secuenciales y clustering.

### 2.2.2 MEJORAMIENTO DE IMÁGENES:

El logro del proceso de mejoramiento de una imagen es acentuar características importantes de la misma para un posterior análisis, clasificación o visualización para una determinada aplicación. Las técnicas de mejoramiento de imágenes no incrementan el contenido de información que posee una imagen, solo enfatiza ciertas características relevantes. Algunos ejemplos de técnicas y algoritmos de mejoramiento de imágenes incluyen mejoramiento de contraste, detección de bordes, filtrado de ruido, transformada de Fourier, etc [7].

### 2.2.3 ANÁLISIS DE IMÁGENES:

Este proceso se basa en la toma de medidas cuantitativas de una imagen para producir una descripción de ella. Las técnicas de análisis de imágenes requieren la extracción de ciertas características de la imagen para lograr identificar objetos deseados al interior de la escena, como puede presentarse en un proceso de clasificación con ayuda de técnicas de reconocimiento de patrones. Estas técnicas incluyen algoritmos de segmentación, separación de datos y de extracción de características [7].

### 2.2.4 RECONSTRUCCIÓN DE IMÁGENES POR PROYECCIONES:

La reconstrucción de imágenes por medio de proyecciones es un caso especial de restauración de imágenes, donde un objeto en dos dimensiones es reconstruido a partir de varias proyecciones en una sola dimensión. Cada proyección se obtiene de proyectar rayos paralelos a través del objeto, como se puede observar en aplicaciones como rayos X, imágenes de radar, imágenes médicas, etc [7].

## 2.3 RECONOCIMIENTO DE PATRONES:

El reconocimiento de patrones es la ciencia encargada de la descripción y clasificación de un conjunto de datos, basado en las categorías previamente establecidas para cada patrón. El margen de aplicaciones de esta ciencia es muy amplio, siendo las más importantes la visión artificial y el aprendizaje de máquina [8]. El esquema de un sistema o aplicación que usa reconocimiento de patrones consta de las siguientes etapas principales:

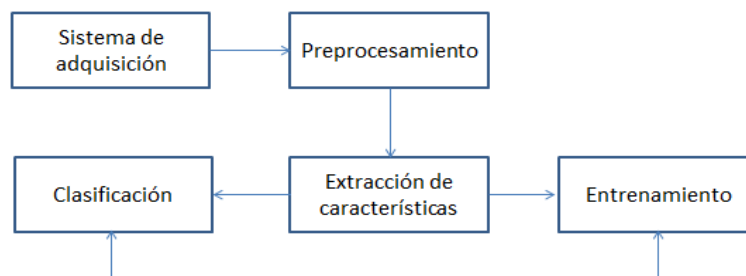


Figura 2. Etapas de un proceso de reconocimiento de patrones

- **SISTEMA DE ADQUISICION:** Se compone de los sensores necesarios para medir las características o datos deseados para ser clasificados y proporciona una representación factible de los elementos del universo a ser discriminados [9].
- **PREPROCESAMIENTO:** Los datos que han sido adquiridos por medio de algún sensor necesitan pasar por un proceso en el cual se resalten características de ellos o eliminar componentes no deseados que pueden afectar las etapas siguientes del sistema, como puede ser ruido en una señal de audio o imagen.
- **EXTRACCIÓN DE CARACTERÍSTICAS:** El propósito de esta etapa consiste en reducir la información de los datos a ser clasificados, midiendo y escogiendo ciertas características o propiedades, con las cuales se pueda realizar un análisis discriminatorio de las clases que han sido previamente definidas.
- **ENTRENAMIENTO:** El sistema de clasificación necesita primero tomar unos datos o muestras de entrenamiento con las cuales los algoritmos implementados puedan reconocer y aprender las diferentes clases que han sido determinadas para discriminar.
- **CLASIFICACIÓN:** Es la etapa en la cual se toman las decisiones en el sistema. Su objetivo es asignar los patrones de clase desconocida a la categoría correspondiente, teniendo en cuenta que la etapa de entrenamiento arroja un error en la discriminación al igual que esta etapa [9].

#### 2.4 OPENCV (OPEN SOURCE COMPUTER VISION LIBRARY):

OpenCV es una biblioteca de software de código abierto orientada para desarrollo de aplicaciones y algoritmos de visión artificial y aprendizaje de máquina (machine learning). Esta librería cuenta con más de 2.500 algoritmos desarrollados y optimizados, entre los que se encuentran algoritmos de reconocimiento de rostros, clasificación de acciones de personas en videos e imágenes, movimientos determinados, extracción de modelos 3D a partir de cámaras estéreo.

OpenCV es multiplataforma, con versiones para sistemas operativos de Linux, MAC OS X y Windows. Uno de sus objetivos consiste en entregar un entorno fácil de desarrollar y entender, teniendo en cuenta aspectos de eficiencia y baja complejidad computacional, lográndolo por medio de programación en lenguajes como C y C++ y optimizándolos por medio de las características de varios núcleos que poseen los procesadores en la actualidad [10].



Figura 3. Logo OpenCV.

# CAPÍTULO 3

## ESPECIFICACIONES

El sistema automático de detección de fallas implementado en OPENCV debe cumplir con los siguientes requerimientos:

- Poder hacer identificación automática usando videos adquiridos en el interior de tuberías de PVC por medio del método de luz estructurada, teniendo en cuenta un ambiente controlado en términos de iluminación y resolución de la escena.
- Realizar el método de superposición de patrones a partir del método de luz estructurada, teniendo en cuenta patrones proyectados y calibración de los elementos a utilizar.
- Lograr una extracción de características relevantes de la imagen, con la finalidad de realizar una clasificación exitosa a partir de ellas, a partir de técnicas de preprocesamiento de los *frames* previamente adquiridos.
- Desarrollar una implementación de un algoritmo de clasificación por aprendizaje supervisado en OPENCV, el cual a partir de un proceso de entrenamiento con la base de datos adquirida, pueda determinar las secciones de la tubería donde se encuentra una falla y el tipo de defecto al cual pertenece.
- Obtener una visualización gráfica de las fallas estructurales detectadas en el video, junto con la respectiva descripción del tipo de defecto que ha sido encontrado por el sistema,

### 3.1 ENTRADAS Y SALIDAS DEL SISTEMA:

A continuación se enumeran las entradas y salidas que debe poseer el sistema de adquisición a implementar:

Tabla 1. Entradas y salidas del sistema

<b>ENTRADA</b>	<b>SALIDA</b>
Secuencia de video	Falla detectada
Base de datos para entrenamiento	Caracterización de defecto

# CAPÍTULO 4

## DESARROLLOS

### 4.1 DESCRIPCIÓN GENERAL Y DIAGRAMA EN BLOQUES:

El siguiente diagrama permite observar cada uno de los bloques que describen de forma detallada el procesamiento que se realiza sobre un video que ingresa al sistema de detección y la forma como se determina si una sección de la tubería corresponde a una falla o no:

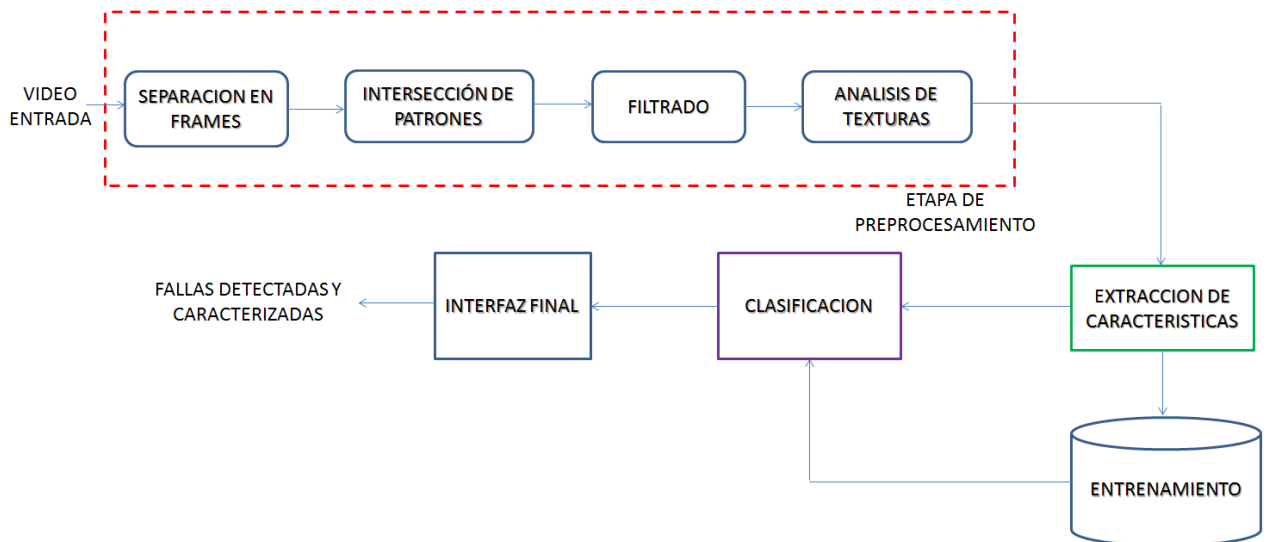


Figura 4. Diagrama en bloques de sistema de detección.

El proceso comienza con la entrada del video de prueba tomado con la cámara SONY SNC-EP580 cuya extensión es AVI al software OPENCV, donde se realiza una separación de éste en *frames* en formato JPG, con el fin de hacer el procesamiento respectivo de cada imagen de forma separada e ir almacenándolos para realizar la clasificación final.



El video que ingresa al sistema, es adquirido mediante el método de luz estructurada, que como ya fue descrito en la sección de marco teórico, consiste en proyectar un haz de luz con un patrón específico sobre el objeto en estudio, en este caso la tubería, con la finalidad de realizar una intersección de los patrones proyectados y obtener detalles que puedan ser relevantes para el proceso de clasificación y detección de fallas.

Los patrones proyectados sobre las secciones de la tubería son franjas horizontales negras y blancas con un tamaño determinado para realizar el método de intersección de patrones invertidos [17].

Para lograr la intersección de patrones a partir del método de luz estructurada, se realiza previo a la ejecución del identificador, un proceso de calibración de la cámara y del proyector utilizando el método de Zhang [18], con el cual se pueden obtener los parámetros intrínsecos de esta así como los coeficientes de distorsión radial de la lente, los cuales serán usados para generar un mapa de distorsión y con ello remover este efecto indeseado en los *frames* adquiridos. Finalmente, se utiliza el método de intersección de patrones por medio del algoritmo de block matching de Konolige [21], utilizando cada *frame* sin distorsión con su respectivo patrón proyectado y contrastándolo con otro *frame* pero con el patrón inverso al anterior, intersectándolos entre sí y formando un mapa entre los dos *frames*, obteniendo información importante de profundidad o de texturas que serán utilizadas como características para realizar la discriminación de las fallas.

Posteriormente, se realiza una etapa de filtrado de ruido, que tendrá como objetivo eliminar componentes no deseados en la imagen que se generen por las condiciones dentro de la tubería y de la cámara como tal, que pueda derivarse en algunos tipos de ruido como lo son ruido sal y pimienta, gaussiano, térmico, etc. Para esto, se utilizan técnicas como filtros mediana para eliminación de ruido impulsivo y filtros *sharpen* contraste para mejorar el detalle de los posibles defectos que se puedan encontrar.

Luego se procede a realizar una etapa de análisis de texturas sobre el mapa generado. Esta información de texturas en las imágenes resulta relevante para esta aplicación, puesto que una falla como una grieta o fisura tienen cambios en su textura respecto a las secciones del tubo que no poseen falla. Además, en el caso que exista una mancha o suciedad de la tubería, su textura será completamente diferente a la de una falla a clasificar y por tanto no será tenida en cuenta. En esta etapa se realizará el cálculo de las 13 medidas de textura propuestas por Haralick [11], por medio del método de matriz de co-ocurrencia, que realiza un análisis estadístico sobre los niveles de grises de la imagen, para que en la siguiente etapa se puedan determinar cuáles son las mejores para conformar el vector de características para la clasificación.

La siguiente etapa a desarrollar en el trabajo de grado consiste en la extracción de características relevantes obtenidas en las etapas de preprocesamiento, con el fin de obtener los datos relevantes de cada uno de ellos y formar por tanto un vector de características.

Para este caso, las características que se obtuvieron de la etapa de preprocesamiento corresponden a las medidas de textura ya nombradas, realizando la extracción de información a partir de la matriz de co-ocurrencia calculada para cada mapa que ingresa al algoritmo y formando así un vector con aquellas características de textura y color que permitan diferenciar claramente los tipos de fallas definidas y las secciones que no poseen defectos.

La etapa de entrenamiento se realizará con la base de datos creada bajo las condiciones establecidas en la adquisición, con el fin de ingresar ésta información al clasificador y lograr un aprendizaje de cuáles son las secciones del video que corresponden a fallas y cuáles no, además de qué tipo de falla se trata. Este aprendizaje es de tipo supervisado, puesto que todas las muestras de entrenamiento son etiquetadas y resulta oportuno éste método, debido a que las clases a discriminar ya son conocidas.

La etapa de clasificación es fundamental en el sistema de detección de fallas, puesto que con ella, se puede saber efectivamente a que clase corresponde cada una de las regiones de la imagen, teniendo en cuenta un

error de test y uno de entrenamiento que depende del número de muestras a entrenar, número de iteraciones, etc. y por tanto del clasificador implementado. Además, se tiene como criterio para la escogencia de estas técnicas, la complejidad y costo computacional que implica implementar cada una de ellas, puesto que en el caso en el que se implemente en un futuro este sistema a la plataforma robótica, este punto será fundamental para el buen funcionamiento del mismo.

Por esto, se evaluó la técnica SVM (Máquina de soporte vectorial), la cual posee un alto rendimiento computacional frente a técnicas como boosting o redes neuronales [12]. Además, con unas buenas muestras de entrenamiento, el error de test de este clasificador es bajo y es usualmente es implementado en éste tipo de aplicaciones [13]. El rendimiento de este clasificador se evalúa realizando una comparación de los resultados obtenidos ante diferentes funciones kernel que pueden entregar comportamientos distintos debido a la naturaleza de las muestras que posea la base de datos, mediante las medidas de rendimiento de un clasificador, como lo son sensibilidad (Verdaderos positivos), especificidad (Verdaderos negativos), falsos positivos y falsos negativos.

Es pertinente aclarar, que se realiza una implementación por jerarquías de clasificadores, donde con un árbol de decisión binario se pueda primero determinar si la región evaluada es una falla o no, y si lo es, que tipo de falla es. La forma como el árbol determinará el proceso de decisión dependerá del clasificador SVM implementado en cada nivel.

Por último, se implementa una interfaz final que permite visualizar y diferenciar la región de interés (ROI) en cada *frame* respecto al resto de la imagen, que serán las fallas detectadas dentro de la tubería, además clasificando el tipo de defecto en el tubo.

#### 4.2 FALLAS ESTRUCTURALES:

Como ya se ha mencionado, el objetivo principal del proyecto consiste en detectar fallas estructurales al interior de tuberías hidrosanitarias y clasificarlas según sus características de textura. Para la adquisición de la base de datos, se utilizó un tubo de PVC de 12 pulgadas de diámetro, lo cual está dentro del diámetro de tuberías en PVC que se usan en los acueductos principales, cuyo diámetro oscila entre 8 y 16 pulgadas.



Figura 5. Tubería en PVC para pruebas.

A partir de estas pruebas, se determinaron tres clases de falla y una de no falla para ser discriminadas por el sistema de detección automático, según sus características geométricas.

**SIN FALLA:** La región de la imagen no pertenece a una falla.



Figura 6. Región sin falla

**ROTURA:** Partes de una tubería que se destruyen por acción de una carga externa [14].



Figura 7. Rotura

**FISURA:** Separación de una tubería que no supera el 30% del espesor del tubo [14].



Figura 8. Fisura

**GRIETA:** Separación abierta de una tubería supera el 30% de espesor del tubo [14].

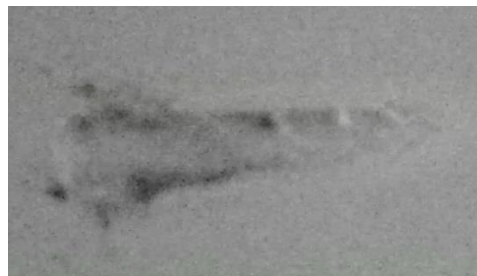


Figura 9. Grieta

### 4.3 ADQUISICIÓN DE VIDEOS:

El proyecto tiene como primer paso la adquisición de los videos al interior de tuberías hidrosanitarias teniendo en cuenta unas condiciones controladas en términos de iluminación, distancia, material del tubo y resolución, mediante el método de luz estructurada. En la sección 4.3.1 y 4.3.2 se detallan los elementos utilizados para la misma y el procedimiento realizado para lograr la adquisición de los videos de entrenamiento y prueba.

#### 4.3.1 ESPECIFICACIONES ELEMENTOS DE ADQUISICIÓN:

Como ya se ha mencionado, los videos tomados para el entrenamiento y posterior clasificación de las fallas estructurales al interior de la tubería fueron adquiridos mediante el método de luz estructurada, el cual consta de una cámara y un haz de luz que proyecta un patrón específico en toda la escena con el fin de capturar deformaciones que se producen cuando este pasa a través de un objeto determinado.

La adquisición de los videos se realizó por medio de la cámara SONY SNC-EP580, que fue elegida por el grupo de investigación del departamento de Ingeniería Electrónica de la Pontificia Universidad Javeriana CEPIT (*Control, Power Electronics and Management of Technological Innovation*) para la plataforma robótica por sus siguientes características respecto a calidad de imagen:

Tabla 2. Especificaciones cámara.

<b>Calidad</b>	1080p Full HD
<b>Máxima resolución</b>	3 Megapíxeles
<b>Sensor</b>	CMOS de bajo ruido [15]
<b>Resolución efectiva</b>	1920 x 1080 píxeles
<b>Formato videos</b>	AVI

Por otro lado, la proyección de los patrones se realizó por medio del video beam EPSON 2400MP, con el objetivo de proyectar varios patrones previamente configurados por medio de un computador destinado para tal fin y lograr mayor resolución que otros métodos, como es el caso de proyección de patrones mediante rejillas, en el cual los patrones no poseen una buena resolución y la luz queda desenfocada a lo largo de la escena. Las especificaciones del video beam utilizado son las siguientes:

Tabla 3. Especificaciones video beam.

<b>Brillo</b>	3000 lúmenes
<b>Tecnología</b>	DLP
<b>Resolución</b>	1024 x 768 píxeles [16]

#### 4.3.2 MONTAJE DE PROTOCOLO DE ADQUISICIÓN:

De acuerdo a los elementos ya descritos, el montaje realizado para la toma de videos al interior de la tubería con el método de luz estructurada fue el siguiente:

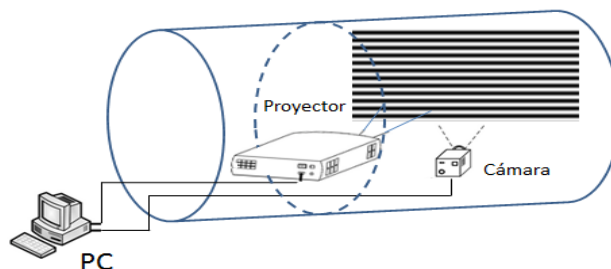


Figura 10. Montaje de adquisición

Este protocolo de adquisición tiene como característica que todos los videos adquiridos tienen condiciones controladas de iluminación por medio del haz que proyecta el video beam, el cual es la única fuente de luz que impacta sobre la tubería, condiciones controladas respecto al material de la tubería, distancias y posiciones controladas de la cámara, que está posicionada de tal manera que las tomas se realizan en las paredes internas del tubo, con el objetivo de no generar fenómenos como distorsión óptica y cumplir con la distancia propuesta entre la lente de la cámara y las paredes del tubo que contendrán las fallas a identificar, la cual es de 12 cm. Por otro lado, la lente proyector está posicionada a una distancia de la pared del tubo de 40 cm, con la finalidad de mantener una distancia mínima en la cual el patrón proyectado posea la resolución esperada con la cual pueda ser capturado de forma correcta por la cámara.

Los patrones escogidos para la proyección de los mismos para cada uno de los videos adquiridos corresponden a un patrón binario de franjas horizontales negras y blancas y su patrón negativo como se muestra en la figura 11, con el fin de implementar el método de intersección entre patrones inversos [17] que será detallado en la sección 4.5.3.

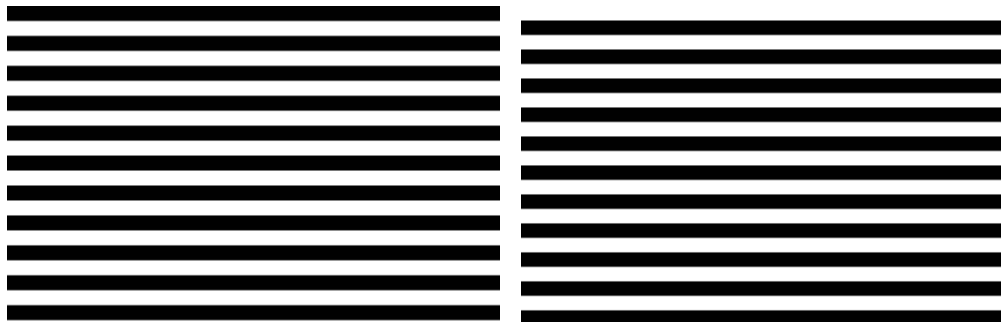


Figura 11. a) Patrón positivo b) Patrón negativo

Este tipo de patrón fue utilizado en lugar de otros como códigos n-arios o arreglos de puntos, puesto que el primero tiene como desventaja que se necesitan proyectar varios patrones sobre la misma escena para lograr reconstruirla, lo cual no es muy útil en ambientes en los cuales existe movimiento y se están tomando videos a una tasa de *frames* alta. Por otro lado, la proyección de un arreglo de puntos en lugar de franjas, podría causar que no todos los puntos puedan ser visibles por la cámara debido a sombras u oclusiones en la imagen, lo cual dificulta el proceso de intersección de patrones [17].

La generación de cada patrón se realiza por medio del software Matlab, donde previamente han sido creados, configurándolos de tal forma que un patrón sea proyectado por medio del video beam durante 500 ms y el otro patrón durante los 500 ms siguientes, de tal manera que en la adquisición se obtenga la misma cantidad de *frames* positivos como de *frames* negativos, que dada la frecuencia de captura de la cámara utilizada (10fps), se obtienen cinco *frames* con un patrón positivo y cinco *frames* con un patrón negativo cada segundo, con la finalidad de realizar la intersección de patrones.

Es pertinente aclarar, que la cámara está conectada por medio de un cable de red Ethernet con un computador en el que se ejecuta la orden de adquirir la secuencia de video por medio del software RealShot Manager Advanced que posee la cámara SONY [15].

Por último, la distancia entre el proyector y la cámara es de 30 cm, de tal modo que pueda ser implementado el método de triangulación para luz estructurada que será explicado en la sección 4.5.3.

#### 4.4 MODELAMIENTO Y CALIBRACIÓN DE CÁMARA:

Posterior a la adquisición de los videos de entrenamiento y de prueba al interior de las tuberías, es necesario realizar un proceso de calibración sobre el sistema de luz estructurada. Dado que un sistema de luz estructurada trabaja de forma similar a un sistema de visión estéreo, pero reemplazando una de las cámaras por un proyector como ya se nombró anteriormente, es necesario realizar un proceso de calibración de la cámara y del proyector con el objetivo de encontrar correspondencias entre puntos del plano de la imagen tomado por la cámara y la escena real por medio del modelamiento de la misma y corregir efectos no deseados como distorsión radial o tangencial de la lente [12].

A continuación se describirá el procesamiento requerido para modelar la cámara y posterior calibración de la misma teniendo en cuenta la técnica utilizada en el software OPENCV para la realización de tal fin, los cuales son necesarios después de realizar el proceso de adquisición por luz estructurada.

##### 4.4.1 MODELO DE ZHANG:

El modelo de una cámara es una aproximación de la geometría interna a lo largo de la posición y orientación de la misma. El método de Zhang además de realizar esta aproximación, modela la distorsión radial de la lente, obteniendo así un modelo más cercano a la realidad y que puede entregar mejores resultados en el momento de realizar un proceso de calibración [18]. Los pasos en el modelamiento se describen a continuación:

- 1) Se realiza una transformación del sistema de coordenadas de la escena a un sistema de coordenadas para la cámara:

$$\begin{bmatrix} {}^cX_w \\ {}^cY_w \\ {}^cZ_w \end{bmatrix} = {}^cR_w \begin{bmatrix} {}^wX_w \\ {}^wY_w \\ {}^wZ_w \end{bmatrix} + {}^cT_w \quad (1)$$

Dónde:

$({}^wX_w, {}^wY_w, {}^wZ_w)$ : Punto 3-D de una escena expresado con respecto al sistema de coordenadas de la escena.

$({}^cX_w, {}^cY_w, {}^cZ_w)$ : Punto 3-D de una escena expresado con respecto al sistema de coordenadas de la cámara.

${}^cR_w$ : Matriz de rotación.

${}^cT_w$ : Matriz de traslación.

- 2) Se realiza una proyección de los puntos 3-D la escena a través del punto focal de la cámara hacia el plano de la imagen, obteniendo un punto 2-D:

$${}^cX_u = f \frac{{}^cX_w}{Z_w}, \quad {}^cY_u = f \frac{{}^cY_w}{Z_w} \quad (2)$$

Dónde:

$({}^cX_u, {}^cY_u)$ : Punto que expresa la proyección lineal de un punto 3-D en el plano de la imagen.

$f$ : Distancia entre el plano de la imagen y centro óptico de la cámara.

3) Se modela la distorsión de la lente, obteniendo un nuevo punto 2-D:

$${}^cX_u = {}^cX_d + \delta_x, \quad {}^cY_u = {}^cY_d + \delta_y \quad (3)$$

Dónde:

( ${}^cX_d, {}^cY_d$ ): Punto 2-D de la imagen incluyendo distorsión de la lente.

$\delta_x$ : Distorsión en el eje X.

$\delta_y$ : Distorsión en el eje Y.

En el caso del modelo de Zhang, la distorsión de la lente se modela como una distorsión radial, que es causada por la curvatura de la lente [15]:

$$\delta_x = \delta_{xr}, \quad \delta_y = \delta_{yr} \quad (4)$$

$$\delta_{xr} = k_1 {}^cX_u ({}^cX_u^2 + {}^cY_u^2), \quad \delta_{yr} = k_1 {}^cY_u ({}^cX_u^2 + {}^cY_u^2) \quad (5)$$

Donde  $k_1$  es el primer término de la serie de Taylor que describe la distorsión radial.

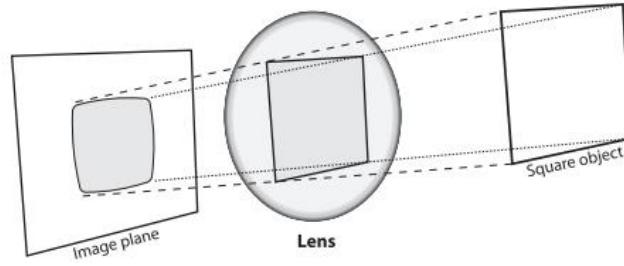


Figura 12. Efecto de distorsión radial [12]

4) Se realiza el cambio del sistema de coordenadas de la cámara al sistema de coordenadas del computador en el cual se realiza el procesamiento, es decir, se obtiene una imagen en píxeles [18]:

$${}^iX_d = -s_x d'_x X_d + u_0, \quad {}^iY_d = -d_y Y_d + v_0 \quad (6)$$

Dónde:

( ${}^iX_d, {}^iY_d$ ): Punto 2-D relacionado a una imagen en píxeles.

( $u_0, v_0$ ): Punto principal en píxeles.

$s_x$ : Factor de escala de la imagen.

$$d'_x = \frac{d_x N_{cx}}{N_{fx}}$$

$d_x$ : Distancia entre elementos del sensor de la cámara en la dirección x.

$d_y$ : Distancia entre elementos del sensor de la cámara en la dirección y.

$N_{cx}$ : Número de elementos del sensor en la dirección x.

$N_{fx}$ : Número de píxeles en cada fila de la imagen [18].





Se obtiene la matriz simétrica B:

$$B = A^{-T}A^{-1} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} \quad (10)$$

$$B = \begin{bmatrix} \frac{1}{\alpha^2} & \frac{-\gamma}{\alpha^2\beta} & \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} \\ \frac{-\gamma}{\alpha^2\beta} & \frac{\gamma}{\alpha^2\beta} + \frac{1}{\beta^2} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} \\ \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} & \frac{(v_0\gamma - u_0\beta)^2}{\alpha^2\beta^2} + \frac{v_0^2}{\beta^2} + 1 \end{bmatrix} \quad (11)$$

De acuerdo con la proyección realizada anteriormente entre el plano de la imagen al sistema de coordenadas del computador donde se procesa la imagen, se define la siguiente transformación lineal entre los dos planos [18]:

$$X' = H * X \quad (12)$$

Siendo,

H: Matriz de transformación.

X': Plano proyectado.

X: Plano original de la imagen.

A partir de las matrices B y H, se definen los siguientes vectores:

$$b = [B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}]^T \quad (13)$$

$$h_i = [h_{i1}, h_{i2}, h_{i3}]^T \quad (14)$$

Por tanto, se define la siguiente relación entre estos vectores:

$$h_i^T B h_j = v_{ij} b \quad (15)$$

Donde,

$$v_{ij} = [h_{i1}h_{j1}, h_{i1}h_{j2} + h_{i2}h_{j1}, h_{i2}h_{j2}, h_{i3}h_{j1} + h_{i2}h_{j3}, h_{i3}h_{j3}]^T \quad (16)$$

Esta ecuación posee las siguientes restricciones:

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{bmatrix} b = 0 \quad (17)$$

Si en el procedimiento de calibración se capturan  $n$  imágenes:

$$Vb = 0 \quad (18)$$

Donde  $V$  es una matriz  $2n \times 6$ . Si  $n \geq 3$ , se obtiene una solución general para  $b$  con un factor de escala. Si  $n = 2$ , se puede imponer una restricción tal como  $\gamma = 0$ .

La solución a (18), se obtiene a través del eigenvector  $V^T V$  asociado al menor eigenvalor.

Una vez estimado el vector  $b$ , se pueden calcular todos los parámetros intrínsecos de la cámara, donde la matriz  $B$  es estimada por medio de un factor de escala:

$$B = \lambda A^{-T} A \quad (19)$$

Siendo  $\lambda$  un factor de escala arbitrario. Por tanto, los parámetros extrínsecos se calculan de la siguiente manera:

$$v_0 = \frac{B_{12}B_{13} - B_{11}B_{23}}{B_{11}B_{22} - B_{12}^2} \quad (20)$$

$$\lambda = B_{33} - \frac{[B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})]}{B_{11}} \quad (21)$$

$$\alpha = \sqrt{\lambda/B_{11}} \quad (22)$$

$$\beta = \sqrt{\lambda B_{11}/(B_{11}B_{22} - B_{12}^2)} \quad (23)$$

$$\gamma = -\frac{B_{12}\alpha^2\beta}{\lambda} \quad (24)$$

$$u_0 = \frac{\gamma v_0}{\alpha} - \frac{B_{13}\alpha^2}{\lambda} \quad (25)$$

Una vez  $A$  es conocida, se pueden calcular los parámetros extrínsecos para cada una de las imágenes de calibración de la siguiente forma:

$$r_1 = \lambda A^{-1}, \quad r_2 = \lambda A^{-1}, \quad r_3 = r_1 \times r_2, \quad t = \lambda A^{-1} h_3 \quad (26)$$

Donde,

$$\lambda = \frac{1}{\|A^{-1}h_1\|} = \frac{1}{\|A^{-1}h_2\|} \quad (27)$$

Ahora, teniendo en cuenta la distorsión radial que genera la lente de la cámara, se pueden refinar los parámetros obtenidos con el procedimiento anterior mediante el método de inferencia de maximum-likelihood. Este estimado se obtiene minimizando la siguiente función:

$$\sum_{i=1}^n \sum_{j=1}^m \|m_{ij} - \hat{m}(A, R_i, t_i, M_j)\|^2 \quad (28)$$

Donde  $\hat{m}(A, R_i, t_i, M_j)$  es la proyección del punto  $M_j$  en la imagen  $i$ .

Minimizando esta ecuación por medio del algoritmo Levenberg-Marquardt [18] y teniendo en cuentas las consideraciones ya descritas para  $n$  imágenes, se pueden calcular de nuevo los parámetros extrínsecos e intrínsecos de la cámara.

**4.4.3 ALGORITMO DE CALIBRACIÓN DE CÁMARA:**

La implementación del algoritmo de calibración se basa en la detección de patrones regulares sobre un objeto que contenga este tipo de caracterización, como es el caso de un tablero de ajedrez o una grilla que contenga círculos. Para este caso, se utilizó un tablero de ajedrez de 6x8, puesto que en la práctica usualmente se utilizan tableros asimétricos con la finalidad de tener varias orientaciones únicas al tomar varias imágenes y con ello tener información suficiente para resolver la ubicación del objeto respecto a la cámara y obtener de una manera más precisa los parámetros intrínsecos y de distorsión de la misma. Por otro lado, el tablero utilizado tiene recuadros blancos y negros, lo cual asegura no tener resultados sesgados entre una medida y otra [12].

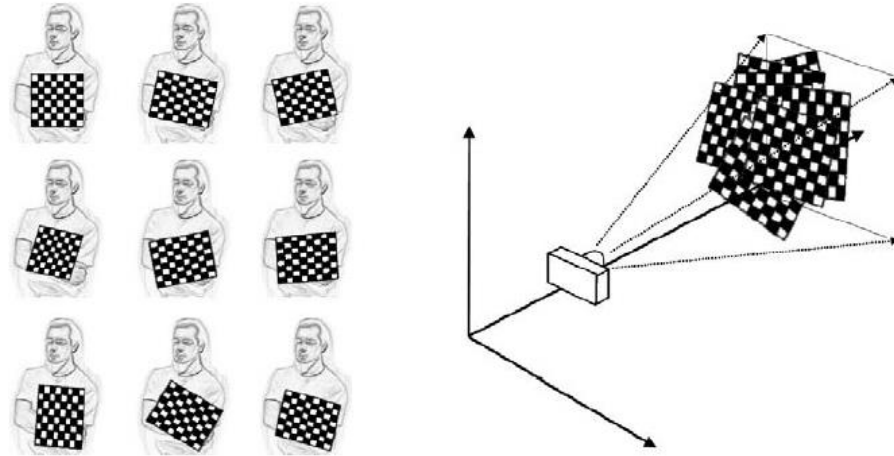


Figura 14. Toma de imágenes de tablero de ajedrez en diferentes orientaciones para calibración de cámara [12].

A continuación se puede detallar el diagrama de flujo del algoritmo de calibración de la cámara:

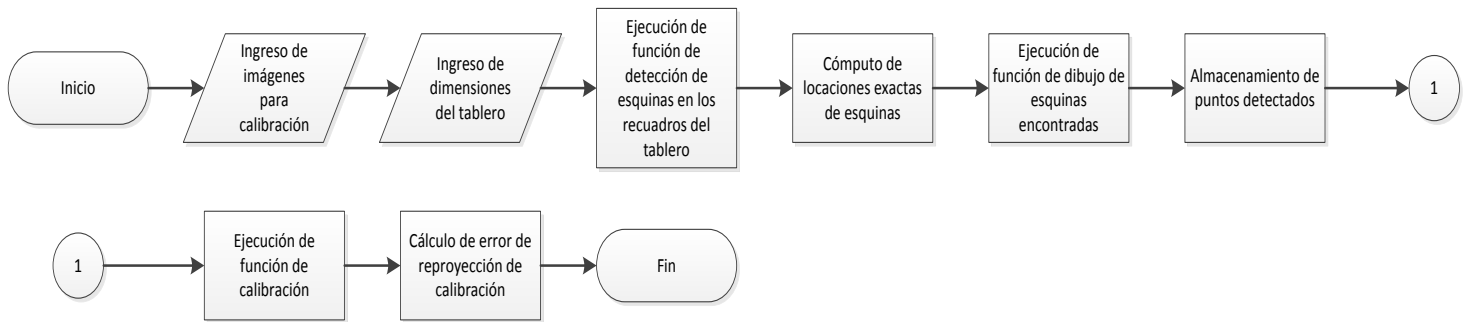


Figura 15. Diagrama de flujo de calibración de cámara

El algoritmo comienza con el ingreso de las imágenes de calibración tomadas con la cámara, en este caso veinte imágenes, con la finalidad de tener varias orientaciones del tablero y obtener una mayor precisión en los parámetros a ser hallados en la calibración:



Figura 16. Imágenes de calibración de cámara

Además, se define el tamaño del tablero de ajedrez que se utiliza, que en este caso es 6x8 por las ventajas ya nombradas. Posteriormente, se realiza una conversión a escala de grises de las imágenes ya introducidas al algoritmo, puesto que las funciones a utilizar en los siguientes pasos solo permiten como entrada imágenes de 8 bits en esta escala.

El siguiente paso consiste en detectar los patrones regulares del objeto de calibración, que en este caso son los recuadros del tablero. Para ello, se utiliza una función que posee OPENCV denominada *cvFindChessboardCorners()*, que tiene como finalidad detectar las esquinas de cada uno de los recuadros del tablero de ajedrez según el número de cuadros que se han calculado (ancho del tablero x largo del tablero). Por otro lado, se utilizan en esta función parámetros específicos que mejoran la detección de las esquinas, como lo es el uso de un umbral adaptativo basado en el brillo de la imagen para realizar una eualización de histograma con el fin de hacer un estiramiento de contraste en toda la imagen.

Posterior a ello, se ejecuta una función que computa de forma más exacta las locaciones de las esquinas de los recuadros del tablero. Esto se debe a que la función anterior entrega unas esquinas aproximadas a las reales, puesto que las imágenes contienen distorsión radial debido a la adquisición realizada por la cámara, por lo que es necesario realizar un ajuste adicional a esa búsqueda ya hecha.

Para comprobar si el algoritmo ha encontrado de forma correcta las esquinas en los recuadros del tablero de ajedrez, se utiliza una función implementada por OPENCV denominada *cvDrawChessboardCorners()*, que dibuja las esquinas halladas como se puede apreciar a continuación:

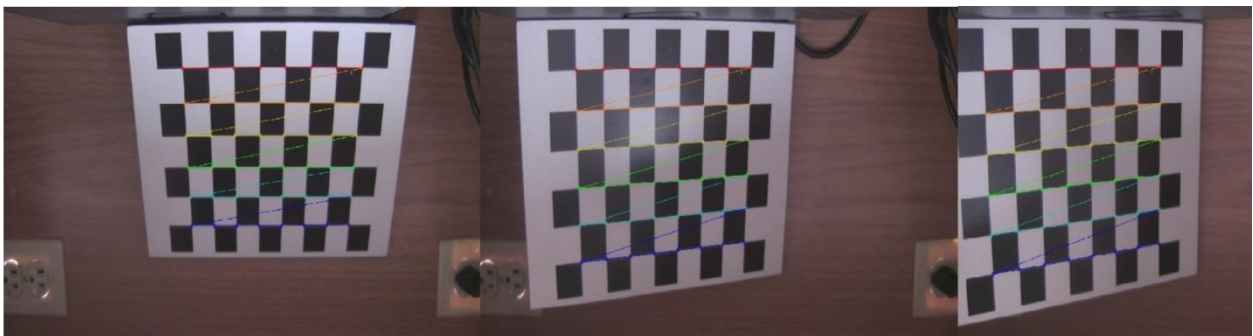


Figura 17. Esquinas detectadas sobre tablero de ajedrez

Finalmente, se almacena la información de los puntos detectados en matrices que corresponden a los puntos 2-D de la imagen denominados *image points* y junto con los puntos 3-D de la misma que se obtienen al definir la ubicación espacial de cada esquina del tablero denominados *object-points*, se ingresan a la función *cvCalibrateCamera2()* creada por OPENCV, que utiliza el método de Zhang ya explicado, a partir de los puntos hallados sobre las 20 imágenes tomadas para tal fin, que entregan 2NK restricciones linealmente independientes para la solución de las ecuaciones ya descritas en este método (Siendo N el número de esquinas y K el número de imágenes).

Esta función retorna los parámetros intrínsecos, extrínsecos y coeficientes de distorsión que serán utilizados para la intersección de patrones a partir del método de luz estructurada. Para verificar que la calibración de la cámara ha sido realizada de forma exitosa, se ejecuta una función que calcula el error de reproyección sobre las esquinas calculadas, la cual proyecta los puntos 3D sobre las imágenes de calibración según los parámetros obtenidos. El error se calcula a partir de la diferencia entre los puntos proyectados y los puntos obtenidos en la calibración [12].

A continuación se pueden detallar los parámetros intrínsecos y coeficientes de distorsión obtenidos al calibrar la cámara y que son fundamentales en los procedimientos siguientes a realizar:

$$A = \begin{bmatrix} 1954.5 & 0 & 941.0199 \\ 0 & 1937 & 532.0867 \\ 0 & 0 & 1 \end{bmatrix}$$

$$kc = [-0.2171 \quad 0.1963 \quad -0.000054 \quad 0.015]$$

Donde los dos primeros parámetros del vector de distorsión *kc* pertenecen a los coeficientes de distorsión radial ( $k_1$  y  $k_2$ ) y los dos últimos pertenecen a los coeficientes de distorsión tangencial ( $p_1$  y  $p_2$ ) [12].

#### 4.5 MODELAMIENTO Y CALIBRACIÓN DEL PROYECTOR:

Con el objetivo de calibrar el sistema de luz estructurada en su totalidad, es necesario calibrar el proyector con el cual se generan los patrones para la posterior intersección de patrones, puesto que la lente del mismo genera fenómenos de distorsión sobre los patrones que se proyectan en la escena. Por tal motivo, es necesario conocer los parámetros intrínsecos y de distorsión del proyector para eliminar estos efectos sobre los *frames* adquiridos.

A continuación se describirá el procesamiento requerido para modelar el proyector y posterior calibración del mismo.

##### 4.5.1 MODELO DE PROYECTOR DE ZHANG:

Un proyector puede ser considerado como el sistema inverso a una cámara, debido a que este proyecta imágenes en lugar de capturarlas. Según Lathuilière [19], se ha confirmado que un proyector puede ser modelado a partir de la misma forma que una cámara, bajo ciertas condiciones y diferencias. Por lo tanto, se puede partir del método de Zhang para modelar el proyector y poder utilizar las funciones que posee OPENCV para calibración de cámaras.

Según el modelo de Zhang, un proyector es considerado como una cámara con una imagen en una sola dimensión (1-D), actuando de forma inversa a la cámara. Por tanto, el modelo difiere en que el sistema de coordenadas final es 1-D. En un principio, las ecuaciones (7) y (8) del modelo de la cámara pueden ser extrapoladas al modelo del proyector:

$$-( {}^I X_{pd} - u_{p0} ) s_{px}^{-1} d_{px}'^{-1} + \delta_{pxr} = f_p \frac{r_{p1} {}^c X_{pw} + r_{p2} {}^c Y_{pw} + r_{p3} Z_{pw} + T_{px}}{r_{p7} {}^c X_{pw} + r_{p8} {}^c Y_{pw} + r_{p9} Z_{pw} + T_{pz}} \quad (29)$$

$$-( {}^I Y_{pd} - v_{p0} ) d_{px}^{-1} + \delta_{pyr} = f_p \frac{r_{p4} {}^c X_{pw} + r_{p5} {}^c Y_{pw} + r_{p6} Z_{pw} + T_{py}}{r_{p7} {}^c X_{pw} + r_{p8} {}^c Y_{pw} + r_{p9} Z_{pw} + T_{pz}} \quad (30)$$

Donde el subíndice  $p$  indica la diferencia entre los parámetros de la cámara ya enunciados y los parámetros del proyector.

Dado que el sistema de coordenadas final es 1-D, las anteriores ecuaciones pueden ser simplificadas de la siguiente manera:

$${}^I X_{pd} = s_{px} d'_{px} f_p \frac{r_{p1} {}^c X_{pw} + r_{p2} {}^c Y_{pw} + T_{px}}{r_{p7} {}^c X_{pw} + r_{p8} {}^c Y_{pw} + r_{p9} Z_{pw} + T_{pz}} + \delta_{pxr} \quad (31)$$

$${}^I X_{pd} = \frac{(\lambda_1 r_{p1} + \delta_{pxr} r_{p7}) {}^c X_{pw} + (\lambda_1 r_{p2} {}^c Y_{pw} + \delta_{pxr} r_{p8}) {}^c Y_{pw}}{r_{p7} {}^c X_{pw} + r_{p8} {}^c Y_{pw} + r_{p9} Z_{pw} + T_{pz}} + \frac{(\lambda_1 r_{p3} + \delta_{pxr} r_{p9}) Z_{pw} + (\lambda_1 T_{px} {}^c Y_{pw} + \delta_{pxr} T_{pz})}{r_{p7} {}^c X_{pw} + r_{p8} {}^c Y_{pw} + r_{p9} Z_{pw} + T_{pz}} \quad (32)$$

Siendo,

$$\lambda_1 = s_{px} d'_{px} f_p$$

#### 4.5.2 CALIBRACIÓN DEL PROYECTOR:

La calibración del proyector se basa en el algoritmo propuesto por Zhang para cámaras, el cual ya fue descrito en la sección 4.4.2. Dado que el proyector se modela como el inverso a una cámara, el proceso de calibración ahora consiste en proyectar un patrón regular por medio de este y detectar los puntos tal como se hizo con el tablero de ajedrez para la calibración de la cámara. La diferencia en este proceso de calibración respecto al de la cámara, se basa en que las coordenadas físicas de los  $K$  puntos que se detectan del patrón regular proyectado no se conocen, por tanto se necesita de la cámara ya calibrada para conocerlos.

Para realizar este procedimiento, nuevamente se utiliza para este caso un tablero de ajedrez de tamaño  $6 \times 8$ , el cual es proyectado con el video beam hacia una superficie regular, tomando imágenes de este en varias orientaciones colocando la cámara calibrada muy cercana al proyector, de tal forma que se pueda asociar el plano físico de la escena que se toma con la cámara y el plano de proyección del video beam, tal como se realiza en [19] y con esto realizar la suposición que el punto donde el rayo de luz de la escena que intersecta el plano de la imagen es igual al punto donde el rayo de luz del proyector intersecta con el plano físico.

El algoritmo de calibración del proyector comienza con el ingreso de las imágenes de calibración tomadas con la cámara, en este caso dieciséis imágenes, mostradas en la figura 18:

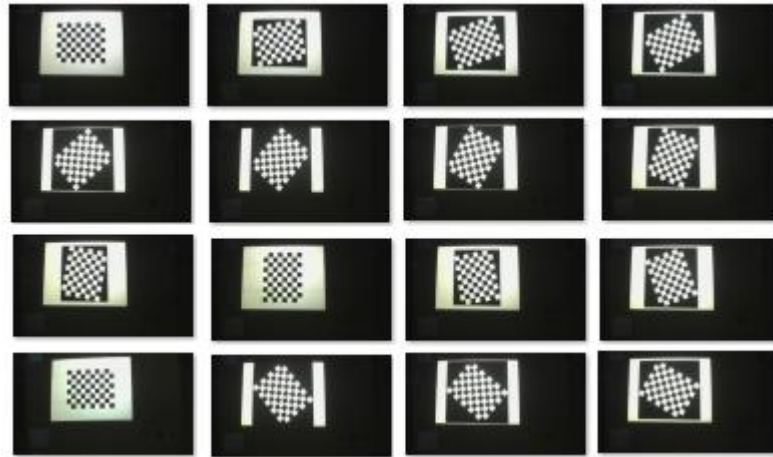


Figura 18. Imágenes de calibración de proyector

La diferencia de este algoritmo respecto al algoritmo de calibración de cámara, radica en que las imágenes de calibración tomadas por la cámara deben pasar primero por un proceso de eliminación de distorsión, de acuerdo a los parámetros intrínsecos y de distorsión de la cámara, como se explicará con más detalle en la sección 4.6.2. Esto se realiza con el objetivo de eliminar la distorsión que introduce la cámara sobre las imágenes y por tanto hallar solo el modelo de calibración del proyector, por eso es indispensable realizar el proceso previo de calibración de cámara [19]. Posterior a ello, se puede realizar el mismo procedimiento del diagrama de flujo mostrado en la figura 15.

A continuación se pueden detallar los parámetros intrínsecos y coeficientes de distorsión obtenidos al calibrar el proyector:

$$A = \begin{bmatrix} 2435 & 0 & 2595.8 \\ 0 & 4031.2 & 850.4051 \\ 0 & 0 & 1 \end{bmatrix}$$

$$kc = [0.0326 \quad 0.000277 \quad -0.0133 \quad -0.0755]$$

Es pertinente aclarar que estos parámetros obtenidos son una aproximación a los parámetros reales del proyector, debido a la suposición ya descrita y que este resultado final dependerá del resultado de calibración de la cámara.

#### 4.6 INTERSECCIÓN DE PATRONES:

Una etapa fundamental en el desarrollo del sistema de identificación, es la intersección de patrones a partir del método de luz estructurada con el cual se realizó la adquisición de los videos para la base de datos y las pruebas finales. Una de las ventajas que posee el método de luz estructurada frente a métodos de reconstrucción de escena como visión estéreo es evitar el problema de encontrar correspondencias entre puntos entre las dos imágenes tomadas para ajustar ángulos y distancias entre cámaras (rectificación), si se escogen los patrones adecuados para tal fin, como los escogidos ya previamente [20]. Por tal motivo, puede ser implementado el método de triangulación usado en visión estéreo sin utilizar un método complicado de encontrar correspondencias, sino con el objetivo de detectar los patrones que se proyectan a lo largo de la escena.

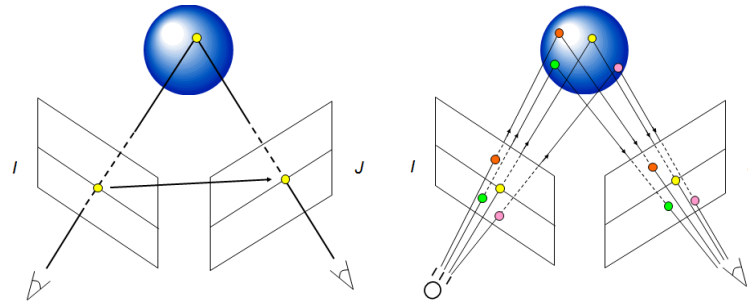


Figura 19. a) Problema de correspondencia en triangulación de visión estereó. b) Correspondencias en triangulación de luz estructurada.

Para implementar este proceso, se creó una función que realiza la intersección de los patrones proyectados, la cual recibe una imagen con uno de los patrones proyectados y otra imagen con el patrón inverso de la imagen anterior, con la finalidad de encontrar disparidades entre las dos imágenes y con ello formar un mapa que permita diferenciar de una manera eficiente secciones de la tubería que poseen falla frente a las que no poseen, resaltando características de textura de la misma.. A continuación se presenta el diagrama de flujo de la función creada con la respectiva explicación de cada una de las etapas:



Figura 20. Diagrama de flujo de intersección de patrones por luz estructurada

#### 4.6.1 CÁMPUTO DE MAPA DE DISTORSIÓN:

La primera tarea que se debe realizar en un proceso de reconstrucción de escena es la eliminación de los efectos de distorsión que introduce la cámara y el proyector con los cuales se realiza la adquisición. Para ello, es necesario utilizar los resultados arrojados por la calibración realizada previamente del sistema para eliminar estos efectos no deseados. Para este caso, el objetivo consiste en eliminar los efectos de distorsión radial y distorsión tangencial, puesto que el método de Zhang con el cual se realizó la calibración, arroja los coeficientes de distorsión de estos dos efectos.

El método utilizado, calcula un mapa de distorsión por medio de la función `cvInitUndistortMap()`, el cual relaciona cada punto de la imagen original al lugar en el cual ese punto es mapeado por esa función, de acuerdo a los parámetros intrínsecos y coeficientes de distorsión obtenidos en la calibración, tanto para la cámara como para el proyector, los cuales se ingresan como argumentos de esta función. El resultado es representado por medio de dos mapas de distorsión para el caso de la cámara y otros dos para el caso del proyector, uno que entrega los puntos en el eje x de la imagen que van a ser mapeados y el otro los puntos en el eje y.

#### 4.6.2 ELIMINACIÓN DE DISTORSIÓN:

El siguiente paso consiste en crear el nuevo par de imágenes a los cuales ha sido eliminada la distorsión radial y la distorsión tangencial de la lente de la cámara y del proyector. Para ello, se utiliza la función `cvRemap()`, la cual toma los mapas de distorsión calculados anteriormente, y realiza un mapeo entre los píxeles de las imágenes originales con distorsión y los interpola hacia una nueva imagen según los nuevos puntos calculados por los mapas de distorsión, que dará como resultado la imagen sin distorsión. Esto se realiza primero eliminando la distorsión causada por la cámara y posteriormente se realiza el mismo procedimiento para eliminar la distorsión causada por el proyector.



La corrección de distorsión radial se realiza bajo las siguientes ecuaciones:

$$x_{\text{corregido}} = x(1 + k_1r^2 + k_2r^4) \quad (33)$$

$$y_{\text{corregido}} = y(1 + k_1r^2 + k_2r^4) \quad (34)$$

Siendo,

$(x_{\text{corregido}}, y_{\text{corregido}})$ : Punto de la imagen sin distorsión.

$(x, y)$ : Punto de la imagen con distorsión.

$k_1, k_2$ : Coeficientes de distorsión radial de la cámara o del proyector.

$$r = \sqrt{(x - u_0)^2 + (y - v_0)^2}$$

$(u_0, v_0)$ : Punto principal de la cámara o del proyector.

Por otro lado, la corrección de distorsión tangencial se realiza bajo las siguientes ecuaciones:

$$x_{\text{corregido}} = x + [2p_1y + p_2(r^2 + 2x^2)] \quad (35)$$

$$y_{\text{corregido}} = y + [p_1(r^2 + 2y^2) + 2p_2x] \quad (36)$$

Siendo,

$p_1, p_2$ : Coeficientes de distorsión tangencial de la cámara o del proyector [12].

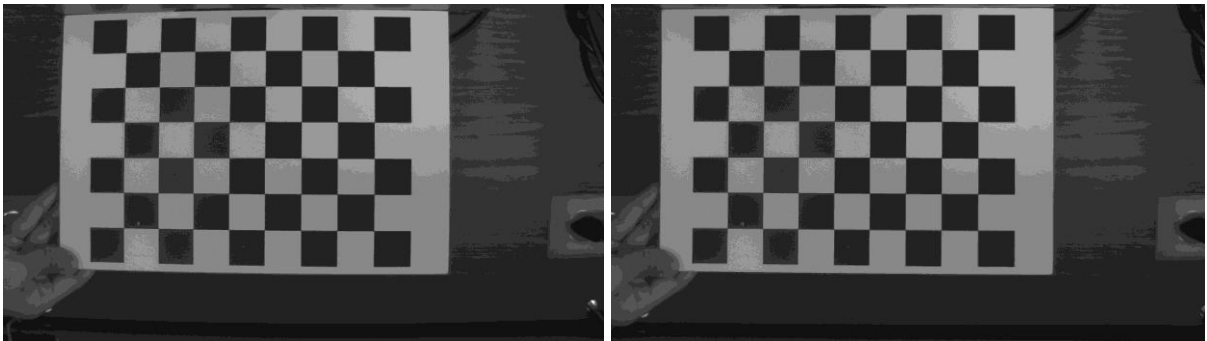


Figura 21. a) Imagen de calibración con distorsión de cámara. b) Imagen de calibración sin distorsión de cámara.

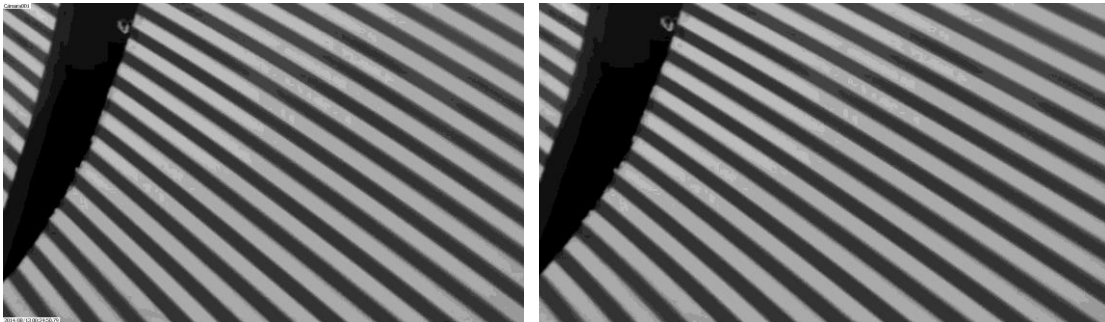


Figura 22. a) Imagen al interior de tubería con distorsión de todo el sistema. b) Imagen al interior de tubería sin distorsión de todo el sistema.

### 4.6.3 GENERACIÓN DE MAPA FINAL:

La principal característica que poseen métodos como visión estereo y luz estructurada es realizar una reconstrucción de la escena por medio de técnicas de reproyección de puntos 3D como mapas de profundidad, a través de los patrones proyectados en la escena. En el caso de un sistema de visión stereo, el principal reto en la reconstrucción consiste en encontrar correspondencias entre los puntos que son vistos por cada una de las dos cámaras que componen el sistema, teniendo en cuenta un proceso previo de rectificación, que se encarga de ajustar los ángulos y distancias entre las cámaras. Por otro lado, un sistema de visión estero tiene problemas para encontrar correspondencias entre objetos con texturas muy suaves y posee restricciones geométricas que dependen de la triangulación realizada por las dos cámaras [21].

En el caso de luz estructurada, los problemas de correspondencias entre puntos, rectificación y restricciones geométricas desaparecen, puesto que el objetivo en este caso para la reconstrucción consiste en la búsqueda del patrón proyectado sobre la imagen, es decir, hacer una decodificación del mismo para lograr detectar las deformaciones que se presentan en los patrones cuando estos pasan por un objeto determinado [20]. Además, con el algoritmo utilizado para la intersección de los patrones proyectados, las características de textura son mas relevantes y se mejoran los efectos no deseados que presenta el método de vision stereo.

#### 4.6.3.1 ALGORITMO DE KONOLIGE:

El reto a desarrollar en luz estructurada consiste en la detección de los patrones que se proyectan en la imagen y de esta forma, encontrar deformaciones que pueden servir para encontrar diferencias relevantes en profundidad y textura, que son de gran importancia para la detección final de las fallas en las tuberías hidrosanitarias.

Dos formas existentes para la detección de estos patrones consiste en la detección de los bordes de estas líneas proyectadas o la detección de forma geométrica de las líneas. Sin embargo las dos poseen problemas a la hora de ser implementadas. La primera, tiene como dificultad que el umbral a ser definido para detectar los bordes de estos patrones depende del contraste entre las líneas negras y blancas proyectadas, por lo que existe una alta complejidad en determinarlo y por tanto las líneas no siempre son detectadas de una forma favorable. Por otro lado, la detección de líneas de forma geométrica presenta como dificultad que las líneas al ser proyectadas pueden presentar diferentes tamaños, por lo que la reconstrucción no se realizaría correctamente [20].

Por esta razón, la opción escogida para decodificar los patrones sobre la imagen, consiste en proyectar líneas con un patrón positivo y su patrón inverso (negativo), de la forma ya explicada anteriormente, de tal manera que al tomar los videos con éste método, se obtengan la misma cantidad de *frames* con un patrón y con otro, para poder contrastarlos y de esta manera reconstruir características de textura por medio de las disparidades que presentan los dos *frames*, puesto que cada uno tendrá información de la escena que no posee el otro, lo cual se denomina como algoritmo de Konolige [21]. Sin embargo, la principal desventaja del método empleado se presenta cuando hay cambios bruscos en el video, puesto que el *frame* con el patrón negativo no puede contener exactamente la misma información del *frame* con el patrón positivo, por lo que en el proceso de adquisición se tuvo como característica tratar de realizar una sincronización entre el movimiento de la cámara y el cambio del patrón.

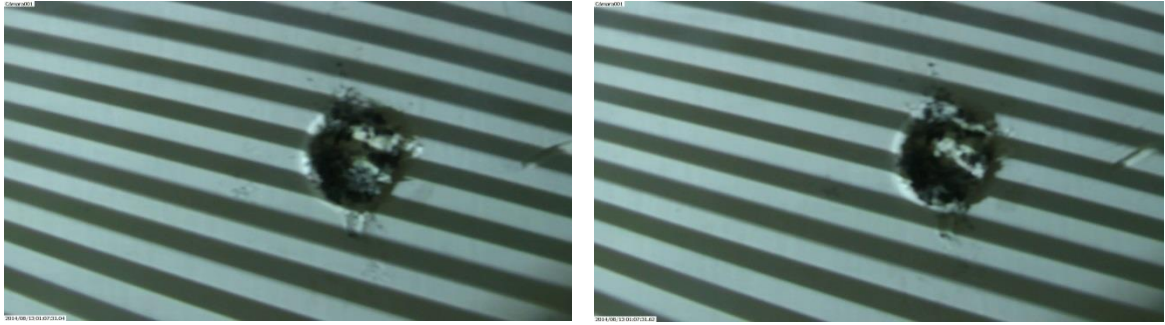


Figura 23. a) Imagen con patrón positivo. b) Imagen con patrón negativo

Este método tiene como principio realizar una intersección entre las dos imágenes que poseen un patrón inverso al otro, por medio de un algoritmo de *block-matching* que utiliza un bloque de  $n \times n$  píxeles en una imagen y se compara con la otra imagen por medio de un barrido horizontal. Las correspondencias encontradas se denominan suma de diferencias absolutas *SAD*. Este algoritmo encuentra correspondencias de puntos tanto de alta textura como de baja textura, según los parámetros que se definen previamente a ejecutar la función que implementa OPENCV de block matching entre dos imágenes, formando así un mapa reconstruido mediante las diferencias en las coordenadas del eje x de los planos de las dos imágenes [21]. El algoritmo posee tres etapas, teniendo en cuenta que las imágenes han pasado por un proceso previo de eliminación de distorsión:

- **PREFILTRADO:** En esta etapa, las imágenes de entrada son normalizadas para reducir posibles diferencias en iluminación entre ellas y mejorar características de textura que serán de gran importancia en el proceso de clasificación [12]. Esto se realiza corriendo una ventana de un tamaño en específico a través de la imagen, el cual fue escogido 5 X 5, con el fin de evitar eliminación de detalles de importancia ante ventanas de mayor tamaño.
- **BUSQUEDA DE CORRESPONDENCIAS:** En este paso se ejecuta el proceso de utilizar una ventana *SAD*, con el objetivo de encontrar disparidades entre puntos a lo largo del eje x, formando un mapa reconstruido con estas diferencias. Se define un parámetro denominado disparidad mínima (*minDisparity*), el cual es el mínimo número de diferencias que se deben encontrar en la búsqueda de correspondencias, que en este caso es cero. Además, se define el número de disparidades en píxeles que el algoritmo va a detectar para la intersección, el cual se definió como 64 con el fin de reducir complejidad computacional si se llegaran a utilizar valores mucho mayores a este [12].
- **POSFILTRADO:** Esta etapa es utilizada para evitar falsas correspondencias que se pudieron haber realizado en la etapa anterior, por medio de un parámetro denominado *uniquenessRatio*, tomando su valor por defecto (12) para filtrar estas falsas disparidades. Por otro lado, para asegurar que el mapa generado tenga detalles de textura suficientes, se usa un umbral denominado *textureThreshold*, actúa como límite en la ventana *SAD*, que no considera correspondencias con un valor inferior a este umbral, que en este caso se ha tomado un valor de dos, puesto que ante valores superiores de umbral, las fallas no son diferenciables de forma correcta frente a secciones que no poseen defectos [12].

A continuación se puede observar el mapa reconstruido de un *frame* que contiene una falla clasificada como fisura y otra como grieta, donde se pueden observar como los patrones proyectados han sido deformados cuando pasan por estos defectos y como son claramente diferenciables a comparación de las demás secciones:



Figura 24. Mapa reconstruido.

#### 4.7 ETAPA DE FILTRADO:

Esta etapa del sistema automático de detección tiene como finalidad realizar filtrado de ruido, es decir, eliminar componentes no deseados en la imagen que se generen por las condiciones dentro de la tubería y de la cámara como tal, que pueda derivarse en algunos tipos de ruido como lo son ruido sal y pimienta, gaussiano, térmico, etc. Aunque el sensor CMOS que posee la cámara a utilizar tiene una tasa de ruido mucho menor que la de sensor CCD que utilizan las cámaras convencionales [22] y se realizan etapas de filtrado en el mapa generado, se asegurará que el ruido restante sea eliminado en una mayor proporción. Para esto, se utilizarán las siguientes técnicas que cumplen con distintas funcionalidades.

##### 4.7.1 FILTRO *SHARPENER*:

El primer paso antes de eliminar posible ruido restante sobre los *frames* capturados, consiste en resaltar las características de textura que poseen las fallas estructurales y que de esta forma el algoritmo de clasificación tenga mayor facilidad en el momento de realizar el entrenamiento y posterior discriminación de las clases definidas. Para eso, se utiliza un filtro *sharpenner* que tiene como función enfatizar texturas y detalles de los objetos en la escena, tal como lo haría un filtro pasa altos, el cual ofrece mejoras en la extracción de bordes en el dominio espacial y refuerza los contrastes que tiene la imagen [23].

Como todo filtro espacial, se realiza una convolución lineal 2D de una ventana de tamaño  $N \times N$  con la imagen de entrada de la siguiente manera:

$$G(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b K(s, t) I(x + s, y + t) \quad (37)$$

Siendo,

$I(x, y)$ : Imagen de entrada.

$K(s, t)$ : Máscara de filtrado.

$G(x, y)$ : Imagen de salida [22].

Para este problema en particular, la máscara utilizada para realizar el mejoramiento de contraste es la siguiente:

$$K = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad (38)$$

Esta máscara tiene como característica que preserva los bordes en la imagen, realizando un procedimiento conocido como *weighted average*, con el cual se obtendrá el siguiente resultado sobre la imagen, siendo  $I(i, j)$  el pixel del centro de la sección de la imagen que se convoluciona con la máscara [23]:

$$G(i, j) = 5 * I(i, j) - [I(i - 1, j) + I(i + 1, j) + I(i, j - 1) + I(i, j + 1)] \quad (39)$$

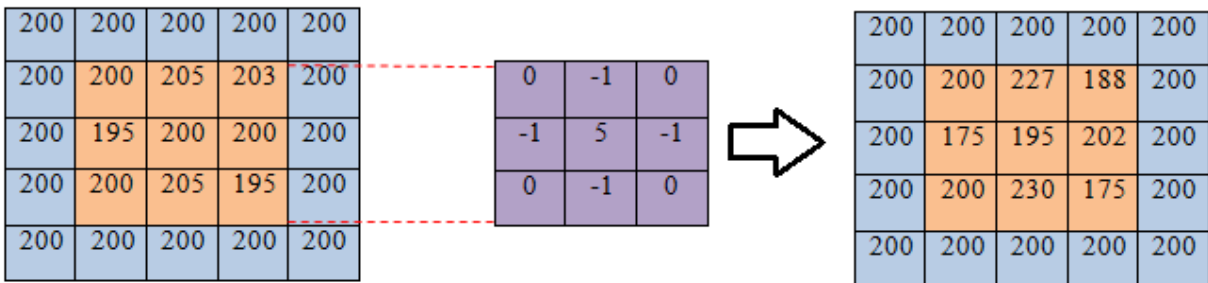


Figura 25. Ejemplo convolución entre máscara de mejoramiento de contraste e imagen.

#### 4.7.2 FILTRO MEDIANA:

Con el fin de evitar suavizados excesivos sobre bordes y detalles que pueden afectar las características de textura de las fallas estructurales que pueden ser causados por filtros pasa bajos, se utiliza un filtro mediana con el fin de eliminar ruido impulsivo en las imágenes proveniente de la adquisición de los videos y del proceso de superposición de patrones, preservando los bordes de las mismas. El filtro mediana reemplaza el pixel de entrada por la mediana de los pixeles contenidos en una ventana alrededor del pixel:

$$v(m, n) = \text{mediana}\{y(m - k, n - l), (k, l) \in W\} \quad (40)$$

Donde  $y(m, n)$  y  $v(m, n)$  son la imagen de entrada y la de salida respectivamente,  $W$  es una ventana previamente escogida, que en este caso es de 3 x 3, al igual que la máscara escogida para el mejoramiento de contraste. El algoritmo para realizar el filtro mediana requiere organizar los valores de los pixeles en la ventana  $W$  en orden ascendente o descendente de tal manera que se pueda escoger el valor del medio. El filtro tiene las siguientes propiedades:

1. Es un filtro no lineal.
2. Es usado para remover líneas o pixeles aislados como ruido binario que puede generar problemas para el procesamiento de la imagen [7].

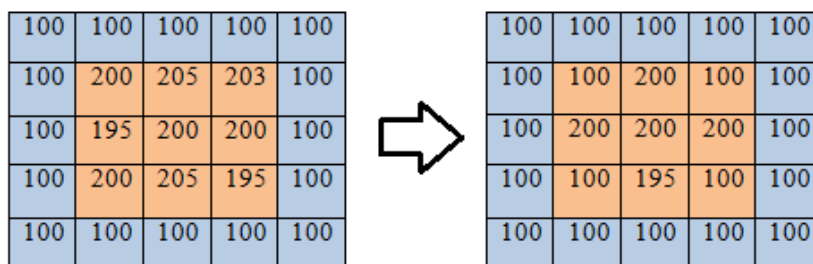


Figura 26. Ejemplo filtro mediana.

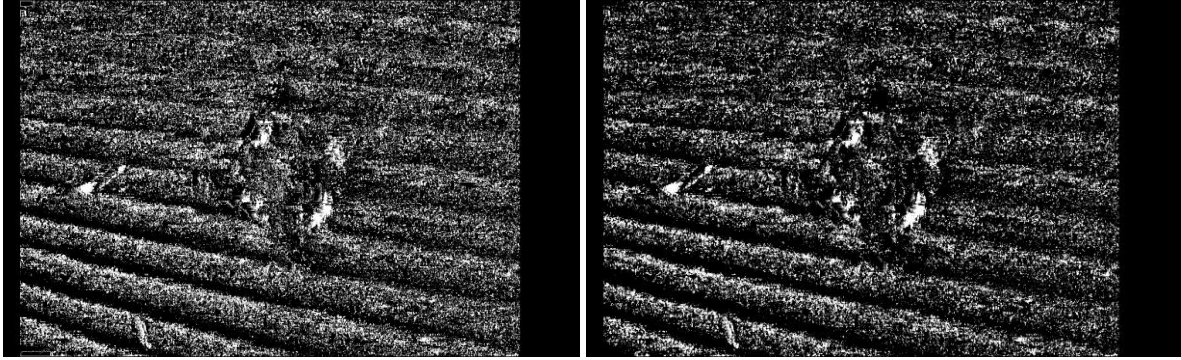


Figura 27. a) Mapa sin filtrar. b) Mapa filtrado.

#### 4.8 ANÁLISIS DE TEXTURAS:

La siguiente etapa en el desarrollo del sistema de identificación consiste en el análisis y extracción de características relevantes sobre las fallas estructurales con la finalidad de lograr una clasificación correcta de estas. Para ello, primero se procede a realizar un análisis de texturas sobre el mapa generado anteriormente. Esta información de texturas en las imágenes resulta de gran importancia para esta aplicación, puesto que una falla como una grieta o fisura tienen cambios en su textura respecto a las secciones del tubo que no poseen falla. Además, en el caso que exista una mancha o suciedad de la tubería, su textura será completamente diferente a la de una falla a clasificar y por tanto no será tomada en cuenta.

La textura de una imagen se define como una variación espacial de las intensidades de los píxeles. El análisis de texturas se estudia generalmente bajo dos enfoques, un análisis estadístico y un análisis estructural. El primero modela las texturas como campos aleatorios, donde las características locales se calculan de manera paralela para cada punto de la imagen. Por otro lado, el análisis estructural considera que la textura se compone de la repetición de patrones elementales espacialmente localizados.

Para este caso, se escogió el análisis estadístico como solución al problema, con el objetivo de calcular y extraer las características propuestas por Haralick [11], que formarán el vector de características con el cual se realizará la clasificación final, utilizando el método de matrices de co-ocurrencia, el cual fue implementado como parte de una función creada que se aplica a los mapas generados, tal como será detallado a continuación.

##### 4.8.1 MATRICES DE CO-OCURRENCIA:

Una de las formas de caracterizar texturas por análisis estadístico se basa en medidas estadísticas de primer orden, que se extraen por medio de histogramas que proporcionan información como intensidad media de niveles de grises, dispersión y contraste en la imagen. Cuando estas características no son suficientes y se desea extraer mayor información de textura, se utilizan medidas estadísticas de segundo orden, que se describen por medio de matrices de co-ocurrencia [24].

Una matriz de co-ocurrencia se basa en el análisis de la ocurrencia de píxeles en la distribución de los niveles de grises en la imagen, incluyendo información espacial en forma de posición relativa entre estos niveles de intensidad dentro de la textura, por lo que son considerados histogramas bidimensionales. Esta relación espacial entre los niveles de grises, es calculada en una dirección determinada entre los píxeles vecinos dentro de una ventana móvil en la imagen, la cual puede ser de  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  o  $135^\circ$  [24].

Usualmente, software de procesamiento de imágenes como Matlab calculan la matriz de co-ocurrencia con una dirección de 0° por defecto. En el caso de OPENCV que no posee funciones de cálculo de estas matrices, se creó una función que calcula una matriz de co-ocurrencia para cada una de las cuatro direcciones ya especificadas, de las cuales se extraen los descriptores requeridos en cada matriz y son promediados para obtener un solo vector de características, puesto que en casos en los que los *frames* o los objetos estén rotados respecto a los anteriores *frames*, no hay seguridad sobre cuál de las direcciones del cálculo de la matriz debe ser tomada, por lo que se recomienda promediar los cuatro resultados y obtener solo uno [11].

La relación espacial entre los niveles de grises se da entre dos pixeles, denominados pixel de referencia y pixel vecino. En el caso de una dirección de 0°, el pixel vecino es el que se encuentra a la derecha del pixel de referencia y se denota como (1,0). Sucesivamente, cada pixel de la imagen se va convirtiendo en el pixel de referencia, empezando desde el pixel que se ubica en la posición (0,0) hasta el pixel que se ubica en la posición (N-1, M-1), donde N y M son las dimensiones de la imagen.

La ocurrencia de pixeles en la distribución de niveles de grises se determina a partir de la relación espacial ya definida y de acuerdo a la orientación. Para el caso de una imagen que posee cuatro niveles de grises, las posibles combinaciones que determinan la ocurrencia se presentan a continuación:

Tabla 4. Ejemplo combinaciones de niveles de grises para matriz de co-ocurrencia.

Pixel de referencia	Pixel vecino			
	0	1	2	3
0	(0,0)	(0,1)	(0,2)	(0,3)
1	(1,0)	(1,1)	(1,2)	(1,3)
2	(2,0)	(2,1)	(2,2)	(2,3)
3	(3,0)	(3,1)	(3,2)	(3,3)

Esto quiere decir que la celda (0,0) de la matriz de co-ocurrencia debe contener el número de veces que ocurre la combinación (0,0) en la imagen según la orientación escogida. La matriz de co-ocurrencia será cuadrada con un tamaño de  $(K + 1) \times (K + 1)$ , donde K es el valor máximo en escala de grises que posee la imagen.

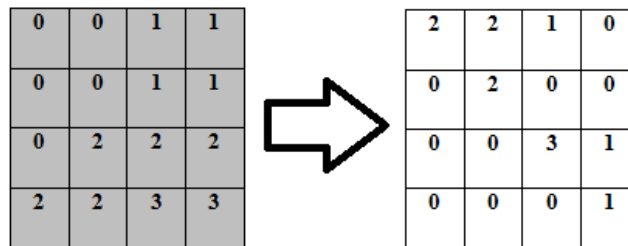


Figura 28. a) Imagen con cuatro niveles de grises. b) Matriz de co-ocurrencia calculada

La matriz de co-ocurrencia debe ser simétrica respecto a la diagonal, por tanto la matriz actual calculada que no es simétrica es sumada con su traspuesta para obtener esta matriz simétrica:

4	2	0	0
2	4	0	0
0	0	6	1
0	0	1	2

Figura 29. Matriz de co-ocurrencia simétrica.

Finalmente, la matriz es expresada como una probabilidad bajo la siguiente relación:

$$P_{ij} = \frac{e_{i,j}}{\sum_{i,j=0}^{N-1} e_{i,j}} \quad (41)$$

Siendo:

$P_{ij}$ : Probabilidad de ocurrencia del nivel de gris en la celda (i,j)

$i$ : Fila de la matriz

$j$ : Columna de la matriz

$e_{i,j}$ : Valor de la celda i, j de la matriz de co-ocurrencia [24].

Esto significa que la probabilidad es calculada como el valor de cada celda de la matriz dividido por el número total de combinaciones encontradas en la imagen según la orientación definida, como se puede apreciar en el ejemplo ya descrito, donde ocurren 12 combinaciones en la matriz no simétrica, pero para cumplir con la simetría este número de combinaciones se duplica (24), arrojando el siguiente resultado:

<b>0.166</b>	<b>0.083</b>	<b>0.042</b>	<b>0</b>
<b>0.083</b>	<b>0.166</b>	<b>0</b>	<b>0</b>
<b>0.042</b>	<b>0</b>	<b>0.250</b>	<b>0.042</b>
<b>0</b>	<b>0</b>	<b>0.042</b>	<b>0.083</b>

Figura 30. Ejemplo de matriz de co-ocurrencia final.

Con el fin de obtener esta matriz, se implementó un algoritmo que será descrito en el siguiente diagrama de flujo:



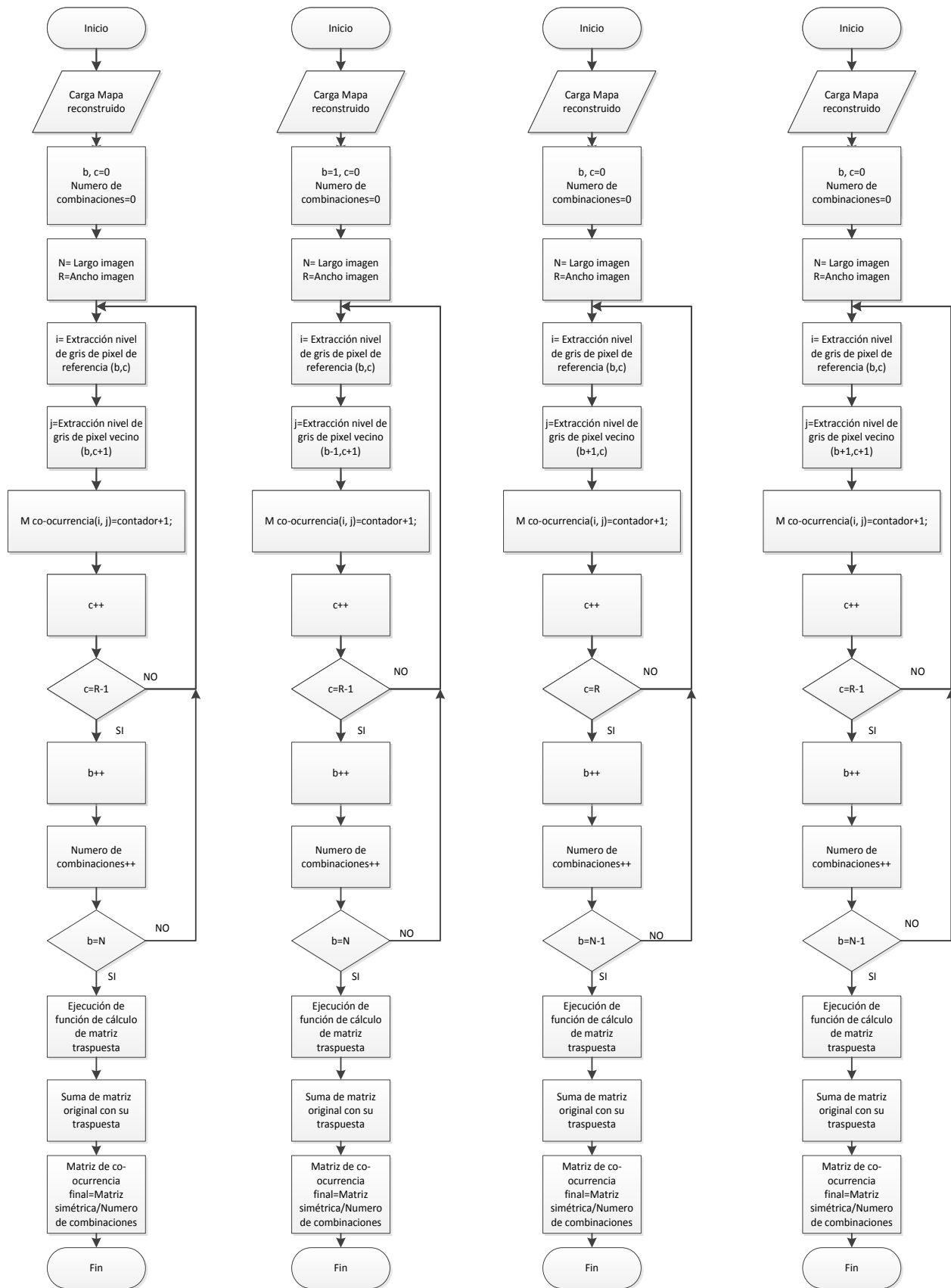


Figura 31. a) Diagrama de flujo matriz de co-ocurrencia a  $0^\circ$ . b) Diagrama de flujo matriz de co-ocurrencia a  $45^\circ$ . c) Diagrama de flujo matriz de co-ocurrencia a  $90^\circ$ . d) Diagrama de flujo matriz de co-ocurrencia a  $135^\circ$ .

## 4.8.2 EXTRACCIÓN DE DESCRIPTORES DE TEXTURA:

Una vez ha sido construida la matriz de co-ocurrencia para cada una de las orientaciones definidas ( $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$ ), se pueden extraer 13 medidas de textura a partir de ellas, obteniendo como resultado final el promedio de las cuatro mediciones por las razones ya mencionadas. Estos descriptores igualmente son relacionados en la función de análisis de textura creada y se implementan de acuerdo a las ecuaciones que rigen a cada uno de ellos y como se detalla en el siguiente apartado.

### 4.8.2.1 ENERGÍA:

Esta medida es usada para medir la uniformidad de la distribución de los niveles de grises en la imagen y el nivel de grosor de la textura. Entrega valores altos cuando la matriz de co-ocurrencia tiene pocas entradas de gran magnitud y es alta cuando las entradas son similares en su valor [11].

$$Energía = \sum_{i=1}^N \sum_{j=1}^N p^2(i, j) \quad (42)$$

Siendo  $p(i, j)$  el valor de la matriz de co-ocurrencia en la posición  $(i, j)$ .

### 4.8.2.2 ENTROPÍA:

La entropía mide la aleatoriedad de la distribución de los niveles de intensidad en la imagen. Si el valor de la entropía es alto, entonces la distribución se encuentra entre mayores niveles de intensidad en la imagen. Es alta cuando los elementos de la matriz de co-ocurrencia tienen valores relativamente iguales [11].

$$Entropía = - \sum_{j=1}^N \sum_{i=1}^N p(i, j) \log[p(i, j)] \quad (43)$$

### 4.8.2.3 CONTRASTE:

El contraste de una textura proporciona información acerca de las variaciones bruscas de color en la imagen. Es la medida opuesta a la homogeneidad [24].

$$Contraste = \sum_{n=0}^{N-1} n^2 \left\{ \sum_{i=1}^N \sum_{j=1}^N p(i, j) \right\} \quad (44)$$

Donde  $n = i - j$ .

### 4.8.2.4 CORRELACIÓN:

La correlación en una textura mide la probabilidad de relación entre las diferentes intensidades presentes en el entorno definido en la matriz de co-ocurrencia. Por definición, pixeles cercanos deberían estar más correlacionados que pixeles lejanos [24].

$$Correlación = \sum_{i=1}^N \sum_{j=1}^N p(i, j) \left[ \frac{(i - \mu_i)(j - \mu_j)}{\sqrt{(\sigma_i^2)(\sigma_j^2)}} \right] \quad (45)$$

Siendo:

$$\mu_i = \sum_{i=1}^N \sum_{j=1}^N i * p(i, j) \quad (46)$$

$$\mu_j = \sum_{i=1}^N \sum_{j=1}^N j * p(i, j) \quad (47)$$

$$\sigma_i = \sqrt{\sum_{i=1}^N \sum_{j=1}^N (i - \mu_i)^2 p(i, j)} \quad (48)$$

$$\sigma_j = \sqrt{\sum_{i=1}^N \sum_{j=1}^N (j - \mu_j)^2 p(i, j)} \quad (49)$$

#### 4.8.2.5 VARIANZA:

La varianza describe la variación de intensidad en la textura alrededor de la media de la intensidad de la imagen que está siendo procesada [25].

$$Varianza = \sum_{i=1}^N \sum_{j=1}^N (i - m)^2 p(i, j) \quad (50)$$

Donde  $m$  es la media de  $p(i, j)$ .

#### 4.8.2.6 MOMENTO DE DIFERENCIA INVERSA (HOMOGENEIDAD):

La homogeneidad entrega información sobre la regularidad de la textura en toda la imagen. Este valor es alto si los valores en la matriz de co-ocurrencia están alrededor o en la diagonal de la misma [25].

$$IDM = \sum_{i=1}^N \sum_{j=1}^N \frac{p(i, j)}{[1 + (i - j)^2]} \quad (51)$$

#### 4.8.2.7 SUMA DE PROMEDIO:

Esta medida refleja el brillo, profundidad de sombra de la imagen y es una medida de promedio de niveles de gris [25].

$$SOA = \sum_{k=2}^{2N} k \left\{ \sum_{i=1}^N \sum_{j=1}^N p(i, j) \right\} \quad (52)$$

#### 4.8.2.8 SUMA DE VARIANZA:

Esta medida muestra valores altos cuando la frecuencia de ocurrencia de niveles de grises está igualmente concentrada en las celdas inferiores y superiores de la matriz de co-ocurrencia [26].

$$SOV = \sum_{k=2}^{2N} (k - SOA) \left\{ \sum_{i=1}^N \sum_{j=1}^N p(i, j) \right\} \quad (53)$$

$$k = |i + j|$$

#### 4.8.2.9 DISIMILARIDAD:

Es una medida similar al contraste. Es alta si la región tiene alto contraste [28].

$$Disimilaridad = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} p(i, j) |i - j| \quad (54)$$

A partir de las medidas de textura ya descritas, surgen otras cuatro medidas que se pueden obtener a partir de la matriz de co-ocurrencia [11]:

#### 4.8.2.10 DIFERENCIA DE VARIANZAS:

$$DOV = \sum_{k=2}^{2N} (k - SOA) \left\{ \sum_{i=1}^N \sum_{j=1}^N p(i, j) \right\} \quad (55)$$

$$k = |i - j|$$

#### 4.8.2.11 SUMA DE ENTROPÍAS:

$$SOE = - \sum_{k=1}^{2N} P_{x+y}(k) \log\{p_{x+y}(k)\} \quad (56)$$

$$p_{x+y}(k) = \sum_{i=1}^N \sum_{j=1}^N p^2(i, j) \quad (57)$$

$$k = |i + j|$$

#### 4.8.2.12 MEDIDA DE INFORMACIÓN DE CORRELACIÓN 1:

$$IMC 1 = \frac{- \sum_i \sum_j p(i, j) \log(p(i, j)) + \sum_i \sum_j p(i, j) \log(\sum_{i=1}^N p(i, j) * \sum_{j=1}^N p(i, j))}{\max(- \sum_{i=1}^N p(i, j) \log[p(i, j)], - \sum_{j=1}^N p(i, j) \log[p(i, j)])} \quad (58)$$

#### 4.8.2.13 MEDIDA DE INFORMACIÓN DE CORRELACIÓN 2:

$$IMC 2 = \sqrt{1 - e^{-2(- \sum_i \sum_j (\sum_{i=1}^N p(i, j) * \sum_{j=1}^N p(i, j)) \log(\sum_{i=1}^N p(i, j) * \sum_{j=1}^N p(i, j)) + \sum_i \sum_j p(i, j) \log(p(i, j)))}} \quad (59)$$

En la sección de análisis y resultados se relacionarán tablas de datos que comprueban que estas características de textura efectivamente son descriptores que diferencian las fallas estudiadas al interior de las tuberías.

### 4.8.3 EXTRACCIÓN DE DESCRIPTORES GEOMÉTRICOS:

Con el fin de complementar el vector de características de textura y diferenciar de una mejor manera cada tipo de falla, se extrajo una característica geométrica sobre cada muestra original por medio de la detección de los contornos de cada uno de ellos. Puesto que el patrón de luz estructurada se deforma una vez existe una falla, el contorno de la muestra cambiará por lo que algunas medidas geométricas resultan pertinentes para este problema. A continuación se presenta el diagrama de flujo de la implementación realizada:

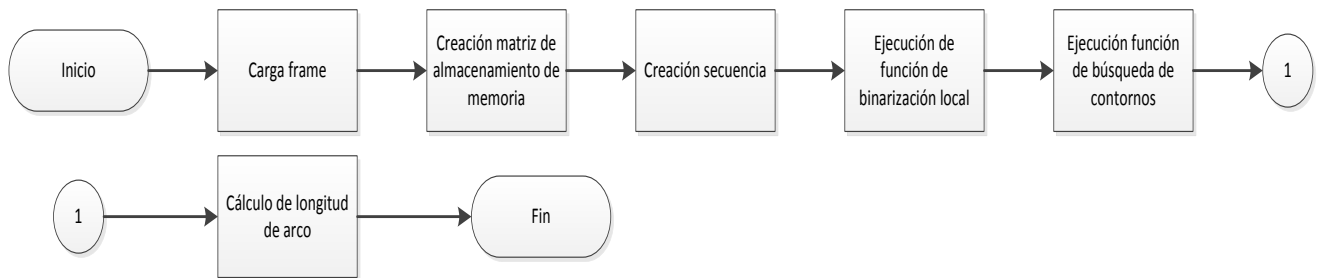


Figura 32. Diagrama de flujo de extracción de característica geométrica

El algoritmo comienza con la carga de cada *frame* al cual se le van a extraer las características de textura y se crea una matriz de almacenamiento de memoria por medio de la función `cvCreateMemStorage()`, que será utilizada posteriormente cuando se ejecute la función de búsqueda de contornos, puesto que esta función necesita que se le indique un lugar donde exista espacio de memoria libre para almacenar los contornos que se han hallado y se crea un objeto denominado secuencia por medio del objeto `CvSeq*`, donde se guardará una lista con los contornos calculados.

Posteriormente se ejecuta una función que realiza una binarización local sobre el *frame* original a partir de un umbral que es definido previamente, en este caso 127. Por medio de la imagen binarizada, se utiliza la función `cvFindContours()`, la cual calcula los contornos de esta y los almacena en la secuencia creada.

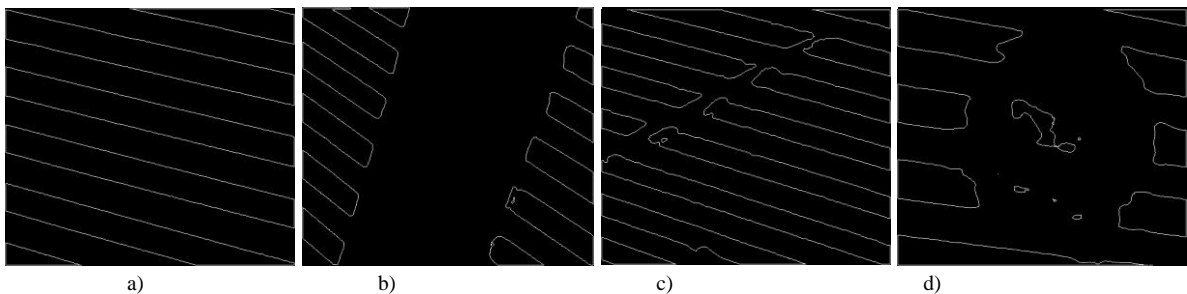


Figura 33. a) Contornos de muestra no falla b) Contornos muestra rotura c) Contornos muestra fisura d) Contornos muestra grieta

Por último se ejecuta la función que calcula la característica geométrica deseada. Este descriptor se denomina longitud de arco, que se calcula por medio de la función `cvArcLength()` [12], la cual calcula la longitud de la curva del contorno que se ha detectado. Por tanto, en el caso que exista una grieta, por ejemplo, la longitud de esta curva detectada será diferente que la longitud del patrón que ha sido proyectado sin deformarse.

#### 4.9 CLASIFICACIÓN:

La etapa de clasificación es fundamental en el sistema de detección de fallas, puesto que con ella, se puede saber efectivamente a que clase corresponde cada una de las regiones de la imagen, teniendo en cuenta primero una etapa de entrenamiento y después una fase de test que corroborará el buen funcionamiento del sistema de detección. Por eso, se evaluó la técnica SVM (Máquina de soporte vectorial), la cual posee un alto rendimiento computacional respecto a la fase de entrenamiento frente a técnicas como boosting o redes neuronales [12]. Además, con unas buenas muestras de entrenamiento, el error de test de este clasificador es bajo y es usualmente es implementado en éste tipo de aplicaciones [13] que utilizan como descriptores de separación de clases, las características de textura de la imagen. Con el fin de realizar una comparación y evaluación de resultados se usaron cuatro tipos de funciones de separación de muestras

para la SVM, denominadas kernel y así determinar cuál de ellos presenta un mayor rendimiento ante los tipos de fallas ya definidos.

A continuación se explicará en detalle la técnica de clasificación y los algoritmos implementados tanto para la fase de entrenamiento como para la fase de test.

#### 4.9.1 MAQUINA DE SOPORTE VECTORIAL (SVM):

Una máquina de soporte vectorial o SVM es un clasificador que se caracteriza por representar datos en una dimensión alta, típicamente mucho más alta que el espacio de características original, lo cual implica que los datos sean mejor separados de forma lineal que en el espacio del vector de características. Este clasificador tiende a tener rendimientos altos con muestras limitadas, a excepción de clasificadores como Boosting o árboles de decisión que necesitan una gran cantidad de muestras para disminuir el error de entrenamiento y de test.

Asumiendo que cada muestra  $x_k$  ha sido transformada a  $y_k = \varphi(x_k)$ , siendo  $\varphi(\cdot)$  una función kernel, para cada una de las  $k$  muestras se escoge una etiqueta  $z_k = \pm 1$ , de acuerdo a si la muestra está en una clase u otra (En el caso de una clasificación binaria) [8].

A continuación se detallarán las funciones kernel más comunes utilizadas en una SVM.

##### 4.9.1.1 KERNEL LINEAL:

Es utilizado cuando el conjunto de datos es linealmente separable [8]. Definiendo una función discriminante lineal en el espacio  $y$ :

$$g(y) = a^T y \quad (60)$$

Donde el vector de pesos  $w$  y el vector de transformación han sido aumentados. Mediante un hiperplano de separación óptimo se asegura que:

$$z_k g(y_k) \geq 1, \quad k = 1, \dots, n \quad (61)$$

El logro de entrenar una SVM es encontrar un hiperplano de separación con el margen más grande que sea la mejor generalización del clasificador. La distancia desde un hiperplano a  $y$  se define como  $\frac{|g(y)|}{\|a\|}$  y asumiendo que existe un margen  $b$  positivo:

$$\frac{z_k g(y_k)}{\|a\|} \geq b \quad (62)$$

Con ésta ecuación se pretende encontrar el vector de pesos  $a$  que maximice  $b$ .

Posterior a realizar la transformación, se generan unos vectores paralelos de soporte que dan un margen para que la separación de las clases sea más confiable y datos que pertenecen a una clase, no se clasifiquen en otra por su cercanía a esta. Estos vectores de soporte están cerrados al hiperplano [8].

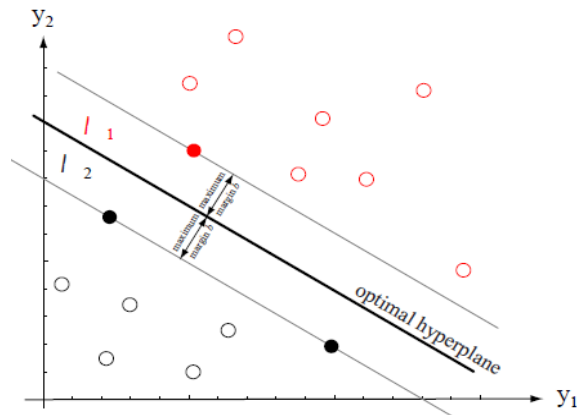


Figura 34. Hiperplano de separación óptimo de clasificador SVM [8].

#### 4.9.1.2 KERNEL NO LINEALES:

Una SVM puede ser extendida desde su concepción lineal hacia el uso de funciones kernel no lineales [8], puesto que el uso de una función discriminante lineal no asegura buenos resultados cuando los datos no son linealmente separables. Estas funciones son un mapeo del espacio lineal un espacio de alta dimensionalidad por medio de una función no lineal y posteriormente construye un hiperplano de separación óptimo [8]. Esta relación entre los dos espacios se define como:

$$(z_1 \cdot z_2) = K(x_1, x_2) \quad (63)$$

El hiperplano óptimo de separación se define como:

$$\sum_{i=1}^N y_i a_i^0 K(x_i, x) + b_0 \quad (64)$$

Estas funciones deben ser positivas y las que usualmente son utilizadas son las siguientes:

- **POLINOMIAL:**

$$K(x, y) = (x \cdot y + c_n)^p \quad (65)$$

$p$ : Grado funcional polinomial.

- **FUNCION DE BASE RADIAL:**

$$K(x, y) = e^{-\gamma \|x-y\|^2} \quad (66)$$

Siendo,

$$\gamma = \frac{1}{2\sigma^2}$$

$\sigma$ : Desviación estándar del kernel gaussiano RBF.

- **SIGMOIDAL:**

$$K(x_i, x_j) = \text{Tanh}(\gamma x \cdot y + c_n), \quad c_n \geq \gamma \quad (67)$$

**4.9.2 ENTRENAMIENTO CLASIFICADOR:**

La primera etapa en el proceso de implementación de la máquina de soporte vectorial, consiste en la ejecución de un algoritmo de entrenamiento, de tal manera que el clasificador sea capaz de reconocer las clases definidas (tipos de fallas y no fallas) a partir del vector de características que posee cada una de las muestras y que fue definido en la etapa de análisis y extracción de texturas.

El aprendizaje del clasificador SVM es de tipo supervisado, lo cual implica que en la etapa de entrenamiento, las muestras destinadas para tal fin son etiquetadas según la clase a la cual pertenecen y según la experiencia de la persona que realiza este proceso.

La idea en el entrenamiento de una SVM radica en minimizar la magnitud del vector de pesos mediante el método de multiplicadores de Lagrange. Esto se realiza minimizando la siguiente función:

$$L(a, \alpha) = \frac{1}{2} \|a\|^2 - \sum_{k=1}^n \alpha_k [z_k a^t y_k - 1] \quad (68)$$

Se minimiza respecto al vector de pesos  $a$  y maximizándola respecto a los multiplicadores  $\alpha_k \geq 0$  [8].

**4.9.2.1 ENTRENAMIENTO MULTICLASE:**

En este problema en particular, se necesita realizar una clasificación multiclase, puesto que fueron definidas cuatro clases diferentes ya descritas (Grietas, fisuras, roturas y sin falla). Para ello, se reduce este problema multiclase a un problema de clasificación binaria por medio de una estructura de jerarquía de clasificadores, mediante la implementación de un árbol binario que discrimina las muestras de entrenamiento y de test de la siguiente manera:

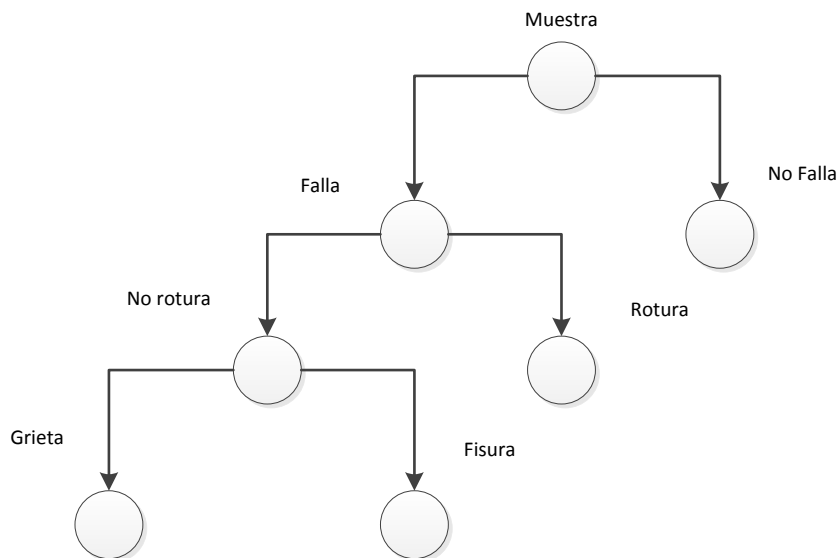


Figura 35. Arbol binario de discriminación de clases por SVM

Esta implementación de clasificación binaria se realizó discriminando una clase contra el resto en el entrenamiento (One vs all) [5], es decir, para el caso de muestras de entrenamiento etiquetadas como grietas, estas eran etiquetadas con  $z_k = 1$ , mientras que el resto de muestras (no fallas, fisuras y roturas) eran etiquetadas de la forma  $z_k = -1$ . Por tanto se obtienen cuatro resultados de entrenamiento que serán agrupados en forma de árbol binario para la etapa de test.

A continuación se presenta el diagrama de flujo de la implementación del entrenamiento de la SVM para el caso de muestras etiquetadas como fallas y muestras etiquetadas como no fallas:



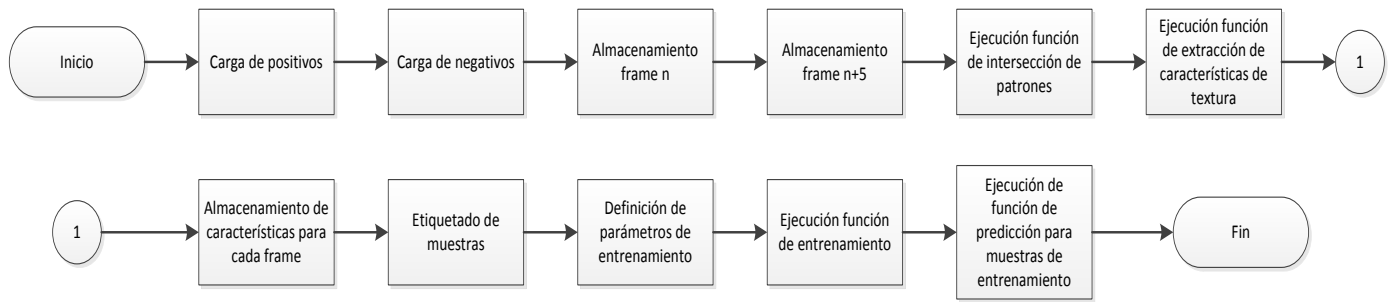


Figura 36. Diagrama de flujo entrenamiento SVM

El algoritmo comienza con la carga de las muestras de entrenamiento que pertenecen a fallas y no fallas y que han sido adquiridas por medio del método de luz estructurada. Posterior a ello, se almacena cada *frame* junto con el *frame* correspondiente con patrón negativo, tal como se indicó en la sección de adquisición de videos, con el objetivo de ingresar cada par de *frames* a la función de intersección de patrones y filtrado. Posteriormente se extrae y se asigna el vector de características de textura y el descriptor geométrico para cada mapa generado, etiquetándolos según la clase a la cual pertenecen, si es falla tiene una etiqueta de  $z_k = 1$  y si no es falla se etiqueta con  $z_k = -1$ .

El siguiente paso consiste en generar los parámetros del clasificador SVM que serán usados para el entrenamiento. Los primeros a definir son los criterios de iteración que tendrá el algoritmo una vez sea ejecutado. Para ello se definió un total de 30.000 iteraciones en el entrenamiento, con la finalidad de lograr convergencia en el clasificador y un factor de tolerancia de  $1 \times 10^{-8}$ , con el cual se asegura que las condiciones Karush-Kuhn-Tucker (KKT) [8] con las que se resuelve el problema de optimización de una SVM, puedan ser cumplidas y por tanto el resultado del entrenamiento sea exitoso.

Lo siguiente a definir es el kernel de separación a utilizar y los parámetros que conciernen a cada uno de ellos. Dado que se utilizaron los cuatro kernel disponibles (lineal, RBF, polinomial, sigmoideal) con el objetivo de realizar comparaciones de rendimiento, se definieron parámetros para cada uno de ellos, tal como será explicado en la siguiente sección.

El último paso consiste en ejecutar la función de entrenamiento del clasificador con las muestras etiquetadas para cada caso y finalmente se valida el resultado del entrenamiento clasificando estas muestras con los resultados generados y con ello conocer el error de entrenamiento que se da ante esas condiciones.

#### 4.9.2.2 ELECCIÓN DE PARÁMETROS DE ENTRENAMIENTO:

Los últimos parámetros que se escogen antes de comenzar el proceso de entrenamiento son los factores de escala que posee el problema de optimización de la SVM y aquellos que pertenecen a cada uno de los kernel con los cuales se va a realizar el aprendizaje de máquina. En la tabla 5 pueden apreciar cada uno de ellos:

Tabla 5. Parámetros de entrenamiento SVM

Parámetro	Descripción	Kernel
Degree	Grado p de kernel polinomial	Polinomial
Gamma	$\gamma = \frac{1}{2\sigma^2}$ . Factor de escala	Polinomial, RBF, sigmoidal
C	Parámetro de optimización de SVM	N/A
$\nu$	Parámetro de optimización de SVM	N/A
$\epsilon$	Parámetro de optimización de SVM	N/A
Coef0	Parámetro de proyección de kernel	Polinomial, sigmoidal [10]

Las pruebas que se realizaron con cada uno de los kernel, implicaron la variación de estos parámetros con el fin de encontrar un rendimiento alto para lograr la detección eficaz de las fallas al interior de las tuberías, como es el caso del grado del kernel polinomial, que se incrementó con el objetivo de observar que curva realizaba una mejor separación de los datos. Sin embargo, dos parámetros que son fundamentales para el buen funcionamiento de la etapa de entrenamiento son C y  $\gamma$ , puesto que con una mala elección de ellos, el clasificador puede presentar *overfitting* (sobreentrenamiento) [29], lo cual implica que el clasificador no será capaz de discriminar una muestra que no haya sido incluida en el entrenamiento.

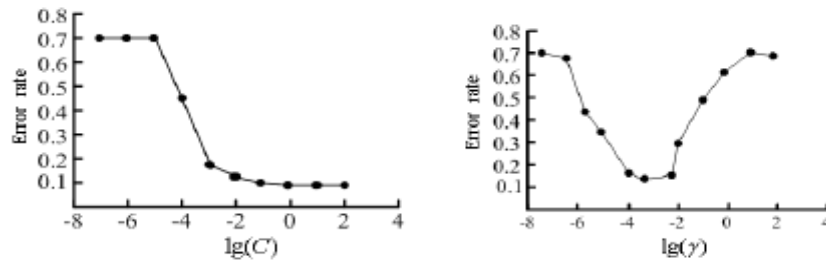


Figura 37. Cambio de tasa de error de entrenamiento con parámetros C y  $\gamma$ . [29]

Existen varias técnicas que estiman estos parámetros de forma conjunta, puesto que el clasificador requiere encontrar un balance entre estos dos factores para obtener una baja tasa de error en el entrenamiento. En este caso se utilizó el método de búsqueda en grilla (*Grid Search Method*), en el cual se toman  $m$  valores de C y  $n$  valores de  $\gamma$ , formando  $m \times n$  combinaciones de estos dos parámetros, con los cuales se entrenan de forma separada cada SVM, obteniendo un estudio de rendimiento de cada una de esas combinaciones [29].

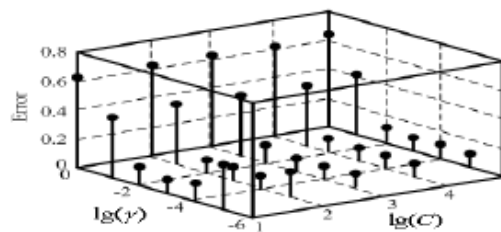


Figura 38. Tasas de error de entrenamiento de distintas combinaciones (C,  $\gamma$ ) con el método de búsqueda en grilla. [29]

Aunque este método tiene una carga computacional alta, tiene como ventaja que cada entrenamiento es independiente y no se necesita hacer un entrenamiento previo con un kernel lineal, como ocurre con otros métodos que requieren de este entrenamiento para encontrar un valor óptimo a  $C$  [29].

#### 4.10 TEST DE CLASIFICADOR E INTERFAZ FINAL:

Las dos últimas etapas del sistema de identificación consisten en la clasificación de los videos que han sido destinados para el test final y la posterior implementación de una interfaz final que permita visualizar las fallas detectadas en las tuberías. A continuación se presenta el diagrama de flujo del programa realizado para estas dos tareas:

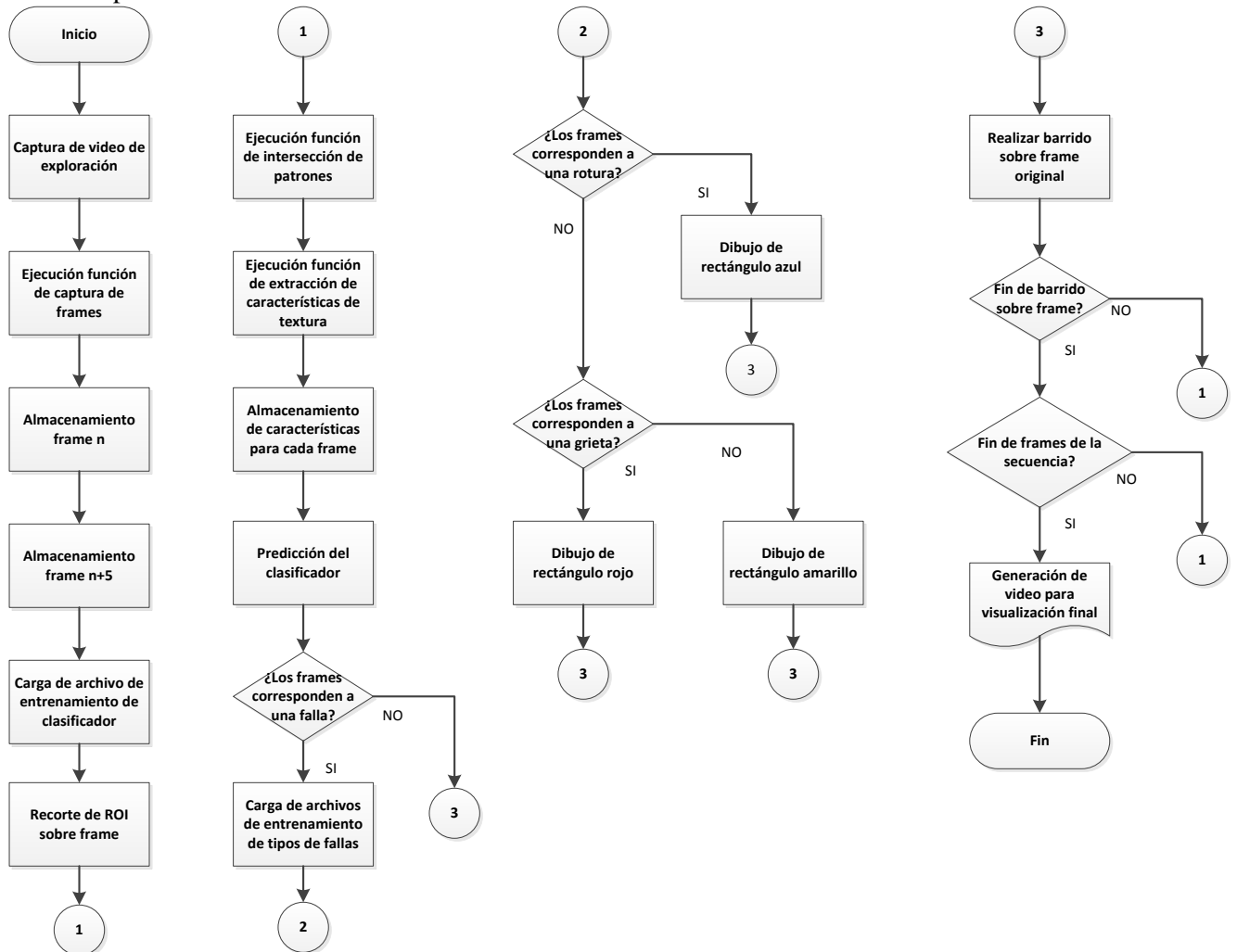


Figura 39. Diagrama de flujo de interfaz final

El algoritmo comienza con la carga del video de exploración en el cual se va a realizar el análisis y clasificación de las fallas estructurales, ejecutando la función de captura de *frames* que posee OPENCV para ello. Posterior a ello, se almacena el *frame* actual y el *frame* que se encuentra 5 posiciones más adelante en el video, el cual corresponde a la imagen con el patrón inverso proyectado al *frame* actual.

El siguiente paso consiste en la carga del archivo de entrenamiento del clasificador SVM correspondiente a la primera discriminación entre fallas y no fallas y que ha arrojado el mejor comportamiento según las pruebas realizadas con cada uno de los kernel, que serán descritas en la sección de análisis y resultados.

El análisis de cada *frame* consiste en realizar un barrido sobre este por medio de un recuadro de tamaño específico y según como ha sido realizado el entrenamiento (503 x 446 píxeles), por lo tanto se extrae la ROI (región de interés) en la parte superior izquierda de la imagen y se realiza el preprocesamiento ya indicado, que consiste en ejecutar la función de intersección de patrones a partir de los dos *frames* con patrones inversos al otro y ejecutando la función de extracción de descriptores de textura creada para asignar el respectivo vector de características a cada ROI. Con estas características almacenadas, se realiza el test del clasificador a esa muestra respecto al archivo de entrenamiento que ha sido cargado previamente, determinando si la ROI corresponde a una falla o no. Esto se relaciona respecto a la etiqueta que asigna el clasificador a la muestra, es decir, si se asigna una etiqueta de  $z_k = 1$ , esto indica que la muestra ha sido clasificada como una falla.

En el caso en el cual la muestra haya sido detectada como una falla, se cargan los archivos de entrenamiento de cada tipo de falla y se implementa la jerarquía de clasificadores ya enunciada. A partir de esto, se determina primero si la falla es una rotura o no, con el test del clasificador destinado para ello. Si la falla tampoco pertenece a una rotura se determina con los otros dos clasificadores si pertenece a una grieta o una fisura. Para poder visualizar las fallas detectadas, se ejecuta una función que dibuja un recuadro de un color específico en el *frame* original sobre las coordenadas en las cuales se ha extraído la ROI y se coloca una etiqueta indicando el tipo de falla. Posterior a clasificar esta muestra, se recorre la imagen de forma horizontal extrayendo cada ROI y se discrimina de acuerdo a los clasificadores ya definidos. Este proceso se realiza para cada uno de los *frames* que han sido extraídos del video, generando como salida estos mismos *frames* con la visualización de las fallas detectadas.

Por último, se genera un video por medio de la función `cvCreateVideoWriter()` en formato `.avi`, que toma todos los *frames* analizados y los almacena de tal forma que se cree una secuencia con una tasa de 10 fps, que es la misma de los videos capturados originalmente.

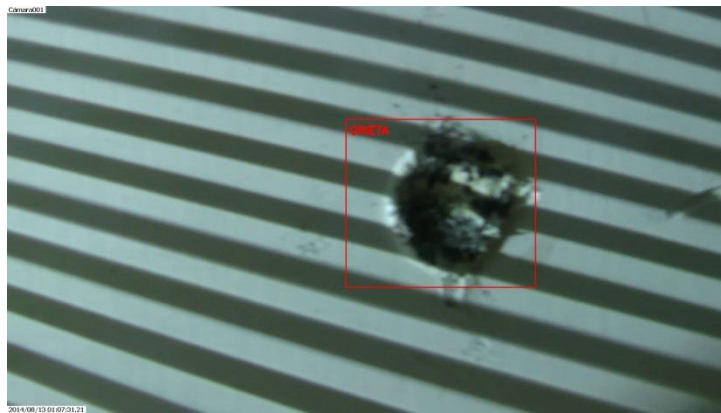


Figura 40. Grieta visualizada en *frame*

# CAPÍTULO 5

## ANÁLISIS Y RESULTADOS

### 5.1 RESULTADOS Y ANÁLISIS DE CALIBRACIÓN:

El proceso de calibración de cámara y proyector fue evaluado por medio del error de reproyección que arroja el algoritmo implementado, que como ya fue explicado, calcula el error como la suma total de distancias al cuadrado entre los puntos proyectados 3-D de las imágenes de calibración y los puntos encontrados en la calibración. El criterio que indica si la calibración ha sido realizada de forma correcta es el número de imágenes del patrón, que debe ser entre 10 y 20, con un error de reproyección entre 0.1 y 1 píxeles [12].

Otro de los puntos cruciales del cual depende un buen resultado la calibración, es el tipo de enfoque en la cámara, que se selecciona en todo el proceso de calibración o adquisición. La cámara SONY utilizada, permite seleccionar un modo de enfoque automático y manual [15]. El primero tiene como característica que la cámara enfoque el objetivo de forma automática, pero presenta como desventaja para esta aplicación, que éste enfoque automático afecte el parámetro intrínseco de distancia focal al tomar varias imágenes para la calibración [30], variándolo de forma instantánea cada vez que la cámara vaya a tomar una imagen, además podrían ocurrir variaciones en el diafragma de la misma, modificando el modelo de distorsión con el cual se pretende hacer una corrección de distorsión de cada *frame*.

Estos fenómenos en el enfoque automático implicarían un aumento del error de reproyección a calcular y por tanto la calidad de la calibración realizada. Por otro lado, estos problemas no se presentan cuando se utiliza un enfoque manual para capturar todas las imágenes de calibración. A continuación se presentan los resultados obtenidos respecto al error de reproyección en la calibración de la cámara, comparando el rendimiento de los dos enfoques.

Tabla 6. Error de reproyección para cada imagen de calibración con enfoque manual y automático

IMAGEN	ERROR DE REPROYECCIÓN ENFOQUE MANUAL (Píxeles)	ERROR DE REPROYECCIÓN ENFOQUE AUTOMÁTICO (Píxeles)
1	0.481	0.508
2	0.426	0.557
3	0.47	0.516
4	0.469	0.514
5	0.468	0.491
6	0.403	0.487
7	0.458	0.486
8	0.416	0.535
9	0.475	0.537
10	0.424	0.511
11	0.41	0.515
12	0.453	0.507
13	0.402	0.537
14	0.508	0.554
15	0.502	0.639
16	0.446	0.658
17	0.481	0.665
18	0.433	0.572
19	0.431	0.713
20	0.43	0.568
<b>ERROR DE REPROYECCIÓN PROMEDIO (Píxeles)</b>	0.4493	0.5535

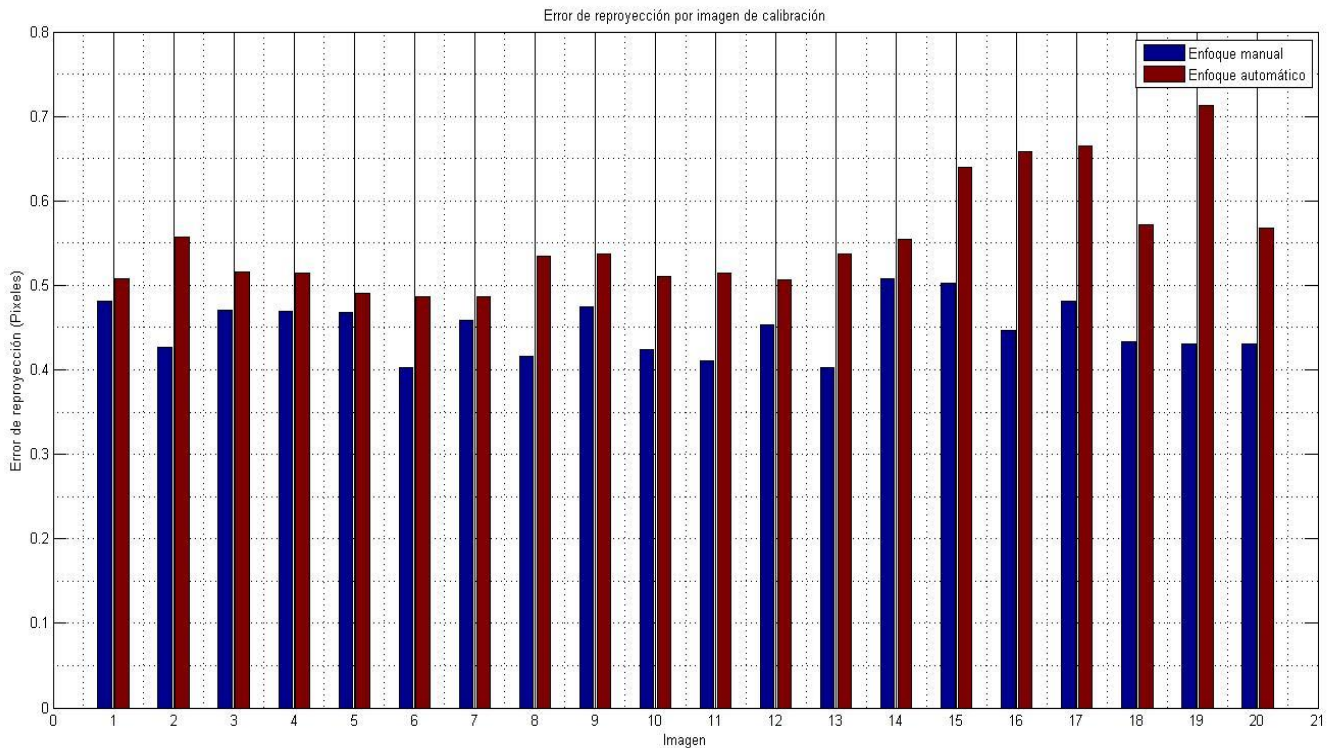


Figura 41. Comparación errores de reproyección para enfoque manual y automático.

Los datos mostrados en la tabla 6 y en la figura 41 denotan los resultados obtenidos al hallar el error de reproyección del proceso de calibración realizado con 20 imágenes del patrón de calibración (tablero de ajedrez) que se muestran en la figura 16, teniendo en cuenta el tipo de enfoque utilizado por la cámara. Como se puede observar en la figura 41, el error de reproyección en ambos casos se encuentra en el rango  $[0.1, 1]$  píxeles, lo cual indica que el proceso de calibración ha tenido un resultado exitoso con los dos enfoques.

Sin embargo, cuando se usa el enfoque manual, el error de reproyección para cada imagen es menor que con enfoque automático tanto para cada imagen de calibración como para el promedio entre todas las imágenes, puesto que como se había enunciado antes, éste método de enfoque no mantiene inmóvil el foco ni el diafragma de la cámara, generando variaciones en parámetros intrínsecos como lo es la distancia focal y en los parámetros de distorsión radial y tangencial de la misma [29]. Por tanto, para la adquisición de la base de datos y de los videos de prueba, se utilizó el enfoque manual de la cámara que asegura mejores resultados en la intersección de patrones cuando se requieren los resultados de la calibración.

A continuación se puede observar el error de reproyección para cada punto detectado en el patrón de calibración (Esquinas del tablero de ajedrez) respecto al eje X y al eje Y de la imagen para los dos enfoques por medio de un diagrama de dispersión:

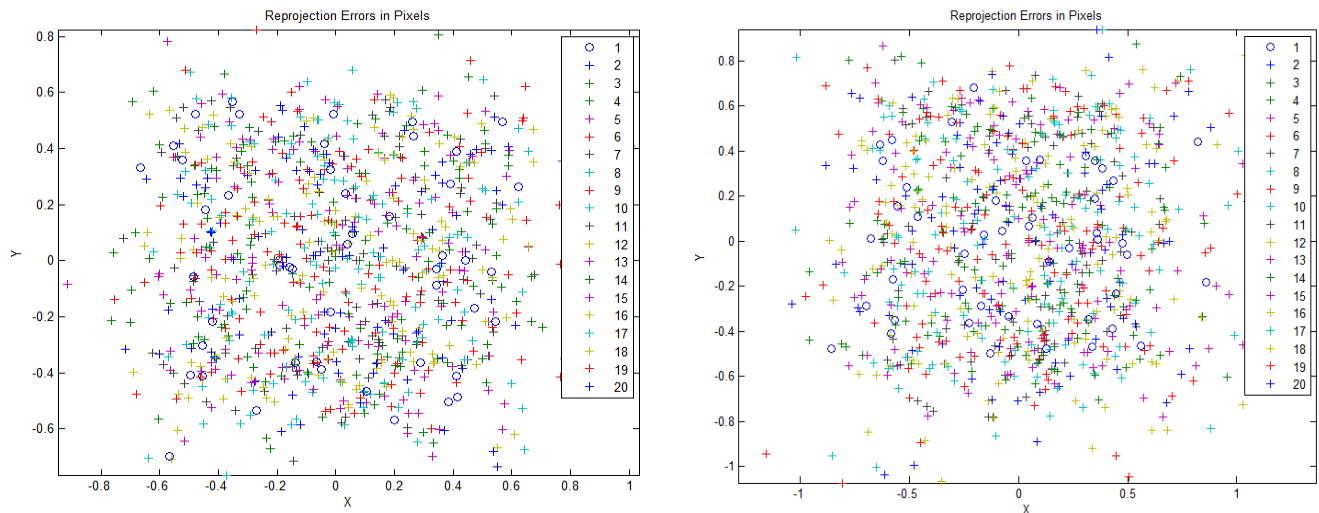


Figura 42. a) Error de reproyección para cada punto en enfoque manual b) Error de reproyección para cada punto en enfoque automático.

Nuevamente se puede observar como el enfoque manual de la cámara presenta un menor error de reproyección que el enfoque automático de la misma, puesto que la mayoría de los puntos para el enfoque manual se encuentran más cercanos hacia cero que en el enfoque automático y no posee puntos que posean un error de reproyección superior a 1 píxel respecto al eje x, como si ocurre con el enfoque automático.

Respecto a la calibración del proyector, los resultados respecto al error de reproyección son los siguientes:

Tabla 7. Error de reproyección para cada imagen de calibración del proyector

IMAGEN	ERROR DE REPROYECCIÓN (Píxeles)
1	0.235
2	0.214
3	0.242
4	0.277
5	0.251
6	0.256
7	0.226
8	0.239
9	0.314
10	0.237
11	0.246
12	0.211
13	0.266
14	0.219
15	0.202
16	0.306
<b>ERROR DE REPROYECCIÓN PROMEDIO (Píxeles)</b>	<b>0.25</b>

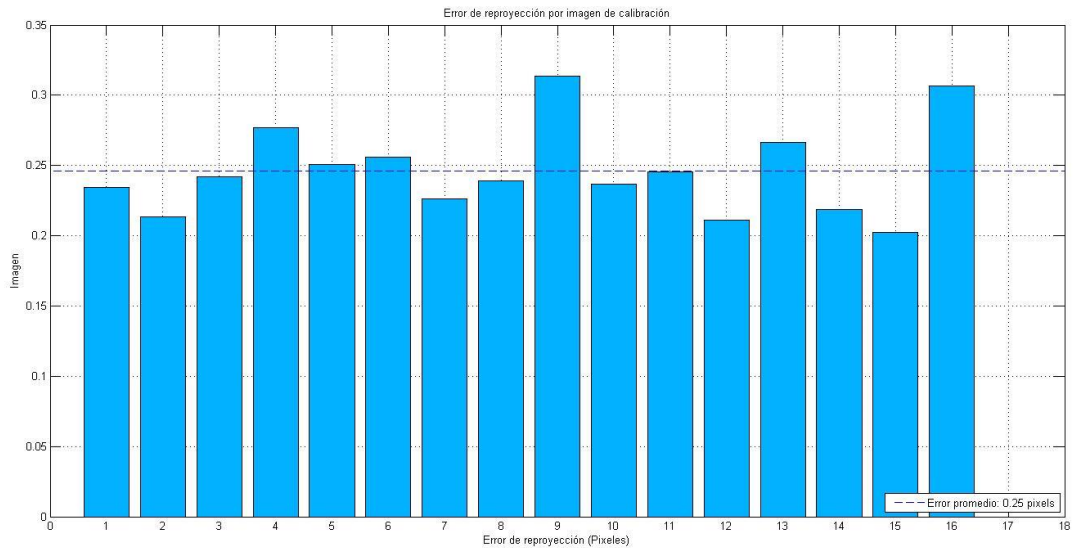


Figura 43. Errores de reproyección en imágenes de calibración del proyector

Los datos mostrados en la tabla 7 y en la figura 43 denotan los resultados obtenidos al hallar el error de reproyección del proceso de calibración del proyector realizado con 16 imágenes del patrón de calibración (tablero de ajedrez) que se muestran en la figura 18, el error de reproyección promedio en este caso también se encuentra en el rango  $[0.1, 1]$  píxeles, lo cual indica que el proceso de calibración ha tenido un resultado exitoso.



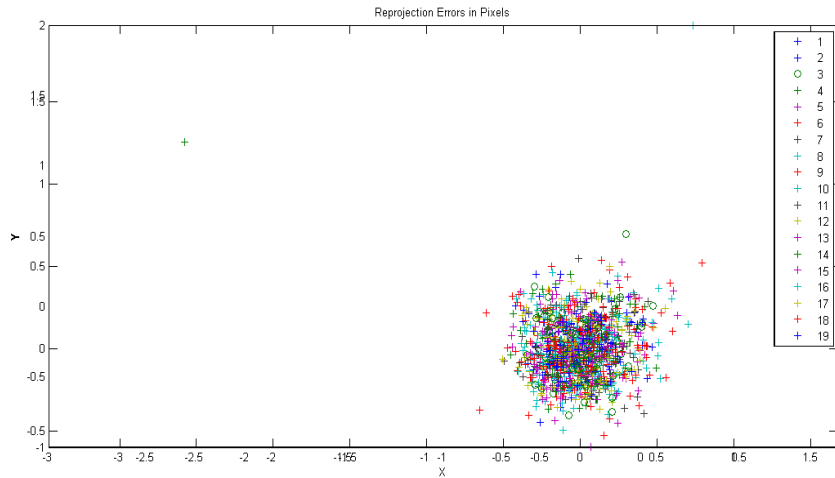


Figura 44. Error de reproyección para cada punto

## 5.2 EJEMPLO DE EVALUACIÓN DE VECTOR DE CARACTERÍSTICAS:

En esta sección se presentan resultados al extraer el vector de características de muestras de ejemplo que pasan por el proceso de intersección de patrones y filtrado, comparando aquellas que son negativas (no falla) con las que son positivas (fallas) y con cada tipo de falla definida. Esto resulta útil al lector para visualizar como las características escogidas para discriminar las fallas efectivamente son un punto de partida para el clasificador, observando aquellas que generan mejores resultados respecto al proceso de diferenciación de clases.

Para el ejemplo se utilizaron 4 par de *frames* con el fin de lograr la intersección de patrones inversos, siendo cada par una clase diferente, extrayendo las características de textura del mapa generado y la características geométrica de la primera muestra de cada falla, como se puede observar a continuación:

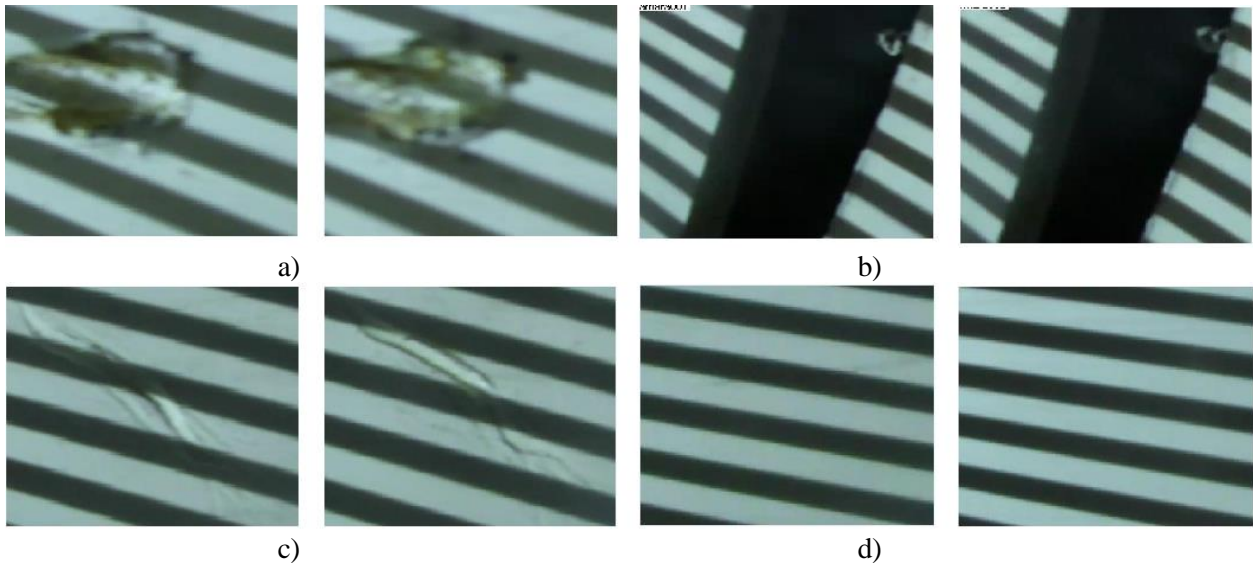


Figura 45. a) Muestras de grietas b) Muestras de roturas c) Muestras de fisuras d) Muestras de no fallas

En la siguiente tabla se relacionan los resultados obtenidos del vector de características para cada una de las clases definidas con las muestras de ejemplo cuando no se utiliza la etapa de filtrado:

Tabla 8. Vector de características para muestras sin etapa de filtrado

CARACTERÍSTICA	SIN FALLA	GRIETA	ROTURA	FISURA
1. ENERGIA	0,110388	0,126804	0,149648	0,110405
2. ENTROPIA	3,75642	3,55826	3,44354	3,71196
3. CONTRASTE	37,9368	0,43619	0,488949	11,2782
4. HOMOGENEIDAD	0,825603	0,856511	0,860661	0,843611
5. CORRELACION	0,358682	0,389215	0,867677	0,849463
6. VARIANZA	5,68693	1,40167	1,35939	3,0261
7. SUMA DE PROMEDIO	249,976	249,28	248,815	250,973
8. SUMA DE VARIANZA	64,3569	7,3751	0,39143	25,3509
9. DISIMILARIDAD	0,665847	0,318421	0,31228	0,504646
10. DIFERENCIA DE VARIANZA	37,4934	0,32227	0,39143	7,69693
11. SUMA DE ENTROPÍAS	-0.0024	-0.08	-0.02	0.008
12. MEDIDA DE CORRELACIÓN 1	0,440133	0,45233	0,491387	0,481898
13. MEDIDA DE CORRELACIÓN 2	0,938055	0,950724	0,945457	0,951451
14. LONGITUD DE ARCO	317	84	6	511

En la figura 46, se presenta la gráfica que relaciona las características de energía, homogeneidad, correlación, disimilaridad, suma de entropías, medida de correlación 1 y medida de correlación 2 de las muestras de ejemplo, siendo el eje x la posición de cada descriptor según el número asignado en la tabla 8 y el eje y el valor correspondiente a cada uno. En esta figura se puede apreciar que las diferencias de cada una de las siete características entre las cuatro clases definidas son del orden de décimas para las características 4 y 9 (Homogeneidad y disimilaridad), y del orden de centésimas para las otras cinco características y no se aprecian diferencias notables entre ellas, por lo cual en la etapa de clasificación estas últimas podrían no aportar información relevante de discriminación en el caso en el cual no haya etapa de filtrado.

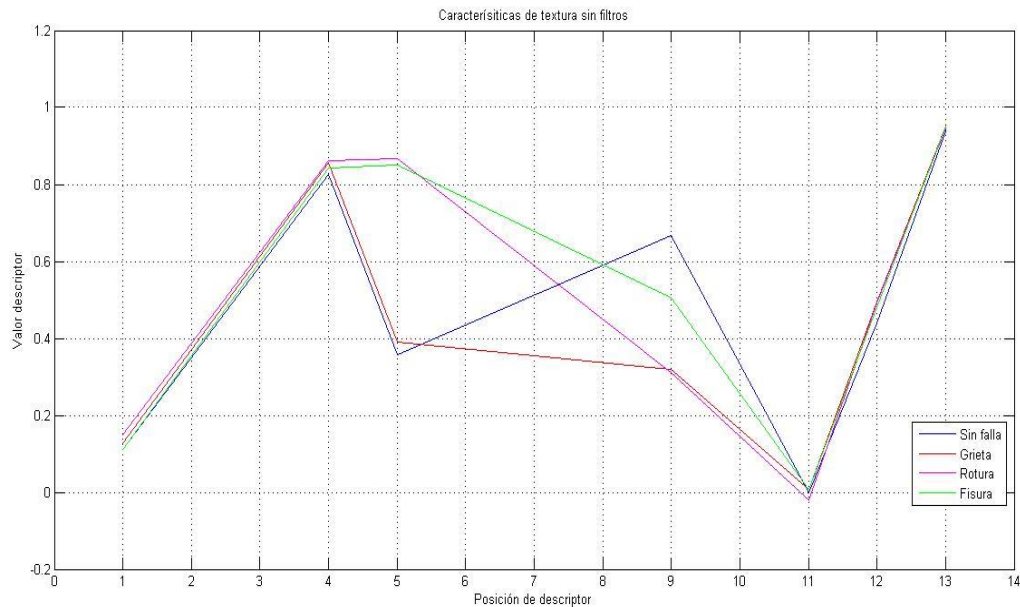


Figura 46. Siete Características de textura de muestras de ejemplo sin etapa de filtrado

En la figura 47, se presenta la gráfica que relaciona las características de entropía, contraste, varianza, suma de promedio, suma de varianza y diferencia de varianza de las muestras de ejemplo, siendo el eje x la posición de cada descriptor según el número asignado en la tabla 8 y el eje y el valor correspondiente a cada uno. En esta figura se puede apreciar que las diferencias entre las cuatro clases definidas son del orden de unidades para las características 3, 8 y 10 (Contraste, suma de varianza y diferencia de varianza), lo cual le puede entregar información relevante al clasificador a la hora de discriminar cada tipo de falla por medio de estos descriptores. Por otro lado no se aprecian diferencias notables para las características 2,6 y 7 (Entropía, varianza y suma de promedio), por lo cual en la etapa de clasificación estas últimas podrían no aportar información relevante de discriminación en el caso en el que nuevamente no exista etapa de filtrado.

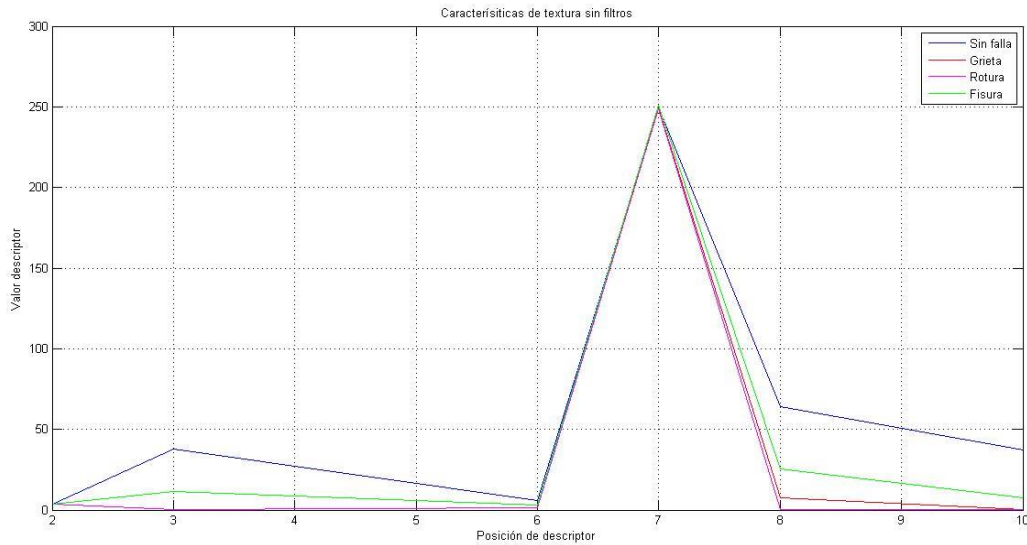


Figura 47. Seis Características de textura de muestras de ejemplo sin etapa de filtrado

En la siguiente tabla se relacionan los resultados obtenidos del vector de características para cada una de las clases definidas con las muestras de ejemplo cuando se utiliza la etapa de filtrado:

Tabla 9. Vector de características para muestras con etapa de filtrado

CARACTERÍSTICA	SIN FALLA	GRIETA	ROTURA	FISURA
1. ENERGIA	0,0220697	0,0342569	0,0497658	0,0257885
2. ENTROPIA	6,83018	6,37371	6,18368	6,55768
3. CONTRASTE	27,8465	10,9594	10,7076	21,7013
4. HOMOGENEIDAD	0,486028	0,554124	0,582137	0,527345
5. CORRELACION	0,346767	0,876917	0,40554	0,384141
6. VARIANZA	4,7815	2,39534	3,00113	8,68259
7. SUMA DE PROMEDIO	249,719	249,355	248,862	249,864
8. SUMA DE VARIANZA	92,6337	24,929	25,3196	280,314
9. DISIMILARIDAD	2,23728	1,79397	1,70683	1,89243
10. DIFERENCIA DE VARIANZA	22,8411	7,74106	7,79435	21,4467
11. SUMA DE ENTROPIAS	0.00360	0.016	-0.02	-0.064
12. MEDIDA DE CORRELACIÓN 1	0,147987	0,185327	0,207853	0,16635
13. MEDIDA DE CORRELACIÓN 2	0,815049	0,853214	0,872774	0,834103
14. LONGITUD DE ARCO	317	84	6	511

En la figura 48, se presenta la gráfica que relaciona las características de energía, homogeneidad, correlación, disimilaridad, suma de entropías, medida de correlación 1 y medida de correlación 2 de las muestras de ejemplo luego de aplicar la etapa de filtrado, siendo el eje x la posición de cada descriptor según el número asignado en la tabla 9 y el eje y el valor correspondiente a cada uno. En esta figura se aprecia como las diferencias entre las características son más notorias en orden de magnitud respecto a cuándo no existía etapa de filtrado, especialmente para la característica de disimilaridad, puesto que el principal objetivo que posee el filtro *sharpenner* consiste en resaltar los detalles del mapa generado y por tanto diferenciar de una mejor manera las características de textura entre cada tipo de falla y no falla.

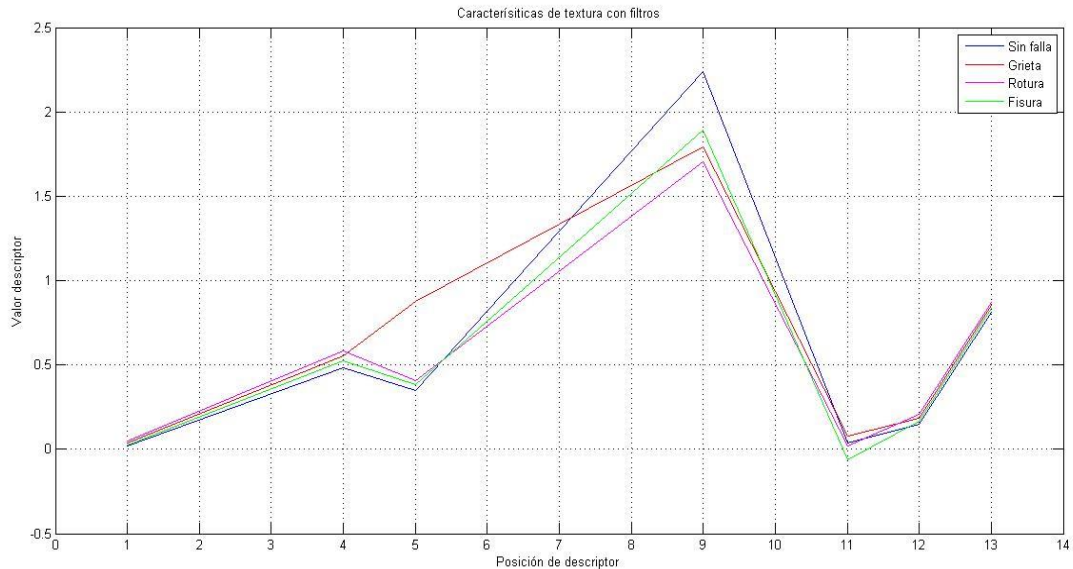


Figura 48. Siete Características de textura de muestras de ejemplo con etapa de filtrado

En la figura 49, se presenta la gráfica que relaciona las características de entropía, contraste, varianza, suma de promedio, suma de varianza y diferencia de varianza de las muestras de ejemplo luego de aplicar la etapa de filtrado, siendo el eje x la posición de cada descriptor según el número asignado en la tabla 9 y el eje y el valor correspondiente a cada uno. En esta figura se aprecia como ciertas características mejoran después de aplicar el filtro *sharpenner* sobre el mapa generado, como lo son las características 3, 8 y 10 (Contraste, suma de varianza y diferencia de varianza), como era de esperarse tras aplicar esta etapa, aunque las otras tres características se mantienen de forma muy similar a como estaban antes de aplicar el filtro, lo cual significa que las primeras serán las que mayor información aportan al proceso de clasificación

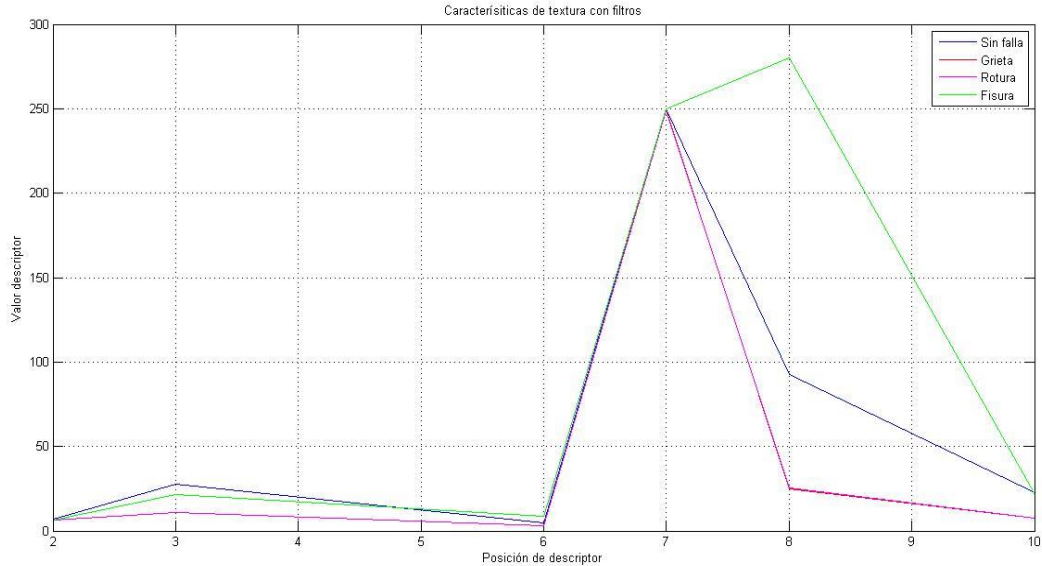


Figura 49. Seis Características de textura de muestras de ejemplo con etapa de filtrado

### 5.3 EVALUACIÓN DE DESEMPEÑO DE CLASIFICADORES:

Para evaluar el desempeño de los clasificadores implementados, se realizaron pruebas de rendimiento tradicionales para un clasificador las cuales son: Sensibilidad, especificidad, razón de falsos positivos, exactitud, valor predictivo negativo, valor predictivo positivo y razón de falsos descubrimientos, los cuales se describen a continuación [9]:

**VERDADEROS POSITIVOS (VP):** Éxitos en la clasificación (Fallas detectadas).

**VERDADEROS NEGATIVOS (VN):** Rechazos correctos (Región sin falla).

**FALSOS POSITIVOS (FP):** Falsas alarmas (Clasificación de fallas inexistentes).

**FALSOS NEGATIVOS (FN):** Clasificación no exitosa (Fallas no detectadas).

- 1) **SENSIBILIDAD O RAZON DE VERDADEROS POSITIVOS (TPR):** Porcentaje de verdaderos positivos clasificados del total de muestras etiquetadas como positivas:

$$VPR = \frac{VP}{VP + FN} \quad (69)$$

- 2) **RATIO O RAZON DE FALSOS POSITIVOS (FPR):** Porcentaje de falsas alarmas:

$$FPR = \frac{FP}{FP + VN} \quad (70)$$

- 3) **EXACTITUD (ACC):** Porcentaje de predicciones que han sido correctas.

$$ACC = \frac{VP + VN}{VP + FN + FP + VN} \quad (71)$$

- 4) **ESPECIFICIDAD O RAZÓN DE VERDADEROS NEGATIVOS (SPC):** Porcentaje de muestras que han sido etiquetadas como negativos y que fueron clasificadas como negativos.

$$SPC = \frac{VN}{FP + VN} \quad (72)$$

- 5) **VALOR PREDICTIVO POSITIVO (PPV):** Porcentaje de muestras que han sido etiquetadas como positivos del total de muestras clasificadas como positivos.

$$PPV = \frac{VP}{FP + VP} \quad (73)$$

- 6) **VALOR PREDICTIVO NEGATIVO (NPV):** Porcentaje de muestras que han sido etiquetadas como negativos del total de muestras clasificadas como negativos.

$$NPV = \frac{VN}{FN + VN} \quad (74)$$

- 7) **RATIO O RAZÓN DE FALSOS DESCUBRIMIENTOS (FDR):** Porcentaje de falsos positivos del total de muestras clasificadas como positivos.

$$FDR = \frac{FP}{FP + VP} \quad (75)$$

### 5.3.1 EVALUACIÓN DE RENDIMIENTO ETAPA DE ENTRENAMIENTO:

Como ya fue mencionado, la etapa de entrenamiento se divide en cuatro partes con el fin de lograr una jerarquía de clasificadores ante el problema de discriminación multiclase, en el cual primero se realiza un entrenamiento para determinar si una muestra pertenece o no a una falla y posteriormente se realizan tres entrenamientos más para diferenciar los tipos de fallas definidos. Para ello se utilizaron un total de 4390 muestras extraídas de 70 videos capturados en las tuberías ya descritas, repartidas de la siguiente manera:

Tabla 10. Número de muestras para cada clase

CLASE	NÚMERO DE MUESTRAS
FALLAS	3210
ROTURAS	810
GRIETAS	1200
FISURAS	1200
NO FALLAS	1180

A partir de estas muestras, se realizó el entrenamiento de distintos clasificadores SVM, que se diferenciaron por el kernel utilizado (Lineal, RBF, polinomial, sigmoïdal) y por los parámetros escogidos de acuerdo al método *Grid Search* explicado en la sección 4.9.2.2. A continuación se presentan los resultados arrojados para cada uno de los clasificadores para un entrenamiento entre fallas y no fallas:

Tabla 11. Resultados de entrenamiento para clasificadores SVM

KERNEL	$\gamma$	C	Coef0	$\rho$	VP	FN	VN	FP
Lineal 1	NA	1	NA	NA	2920	290	740	440
Lineal 2	NA	150	NA	NA	2866	344	766	414
RBF 1	0.0001	10	NA	NA	3122	88	920	260
RBF 2	0.0001	80	NA	NA	3146	64	1044	136
RBF 3	0.0001	100	NA	NA	3154	56	1050	130
RBF 4	0.0001	150	NA	NA	3156	54	1062	90
RBF 5	0.00001	150	NA	NA	3100	110	796	384
RBF 6	0.0001	500	NA	NA	3176	34	1104	76
RBF 7	0.0001	1000	NA	NA	3182	28	1126	54
RBF8	0.0001	1200	NA	NA	3190	20	1132	48
RBF9	0.0001	1500	NA	NA	3192	18	1136	44
RBF10	0.0001	3000	NA	NA	3196	14	1140	40
Sigmoidal 1	$1 \times 10^{-8}$	100	0	NA	2328	882	250	930
Sigmoidal 2	$1 \times 10^{-8}$	1000	0	NA	2302	908	262	918
Sigmoidal 3	$1 \times 10^{-8}$	10	0	NA	2608	602	148	1032
Sigmoidal 4	$1 \times 10^{-9}$	1000	0	NA	2264	946	234	946
Sigmoidal 5	$1 \times 10^{-8}$	10	0.1	NA	2670	540	130	1050
Sigmoidal 6	$1 \times 10^{-8}$	1500	0	NA	2304	906	262	918
Sigmoidal 7	$1 \times 10^{-8}$	3000	0	NA	2302	908	262	918
Sigmoidal 8	$1 \times 10^{-9}$	3000	0	NA	2264	946	234	946
Sigmoidal 9	$1 \times 10^{-8}$	1500	0.01	NA	2302	908	262	918
Polinomial 1	0.0001	1000	0	2	2526	684	494	686
Polinomial 2	0.0001	1000	0	3	1993	1217	1126	54

Tabla 12. Mediciones de rendimiento en entrenamiento para clasificadores SVM

KERNEL	TPR	FPR	ACC	SPC	PPV	NPV	FDR
Lineal 1	90.97%	37.29%	83.37%	62.71%	86.9%	71.84%	13.1%
Lineal 2	89.28%	35.08%	82.73%	64.92%	87.38%	69.01%	12.62%
RBF 1	97.26%	22.03%	92.07%	77.97%	92.31%	91.27%	7.69%
RBF 2	98.01%	11.53%	95.44%	88.47%	95.86%	94.22%	4.14%
RBF 3	98.26%	11.02%	95.76%	88.98%	96.04%	94.94%	3.95%
RBF 4	98.32%	10%	96.08%	90%	96.4%	95.16%	3.6%
RBF 5	96.57%	32.54%	88.75%	67.46%	88.98%	87.86%	11.02%
RBF 6	98.94%	6.44%	97.49%	93.56%	97.66%	97.01%	2.34%
RBF 7	99.13%	4.58%	98.13%	95.42%	98.33%	97.57%	1.67%
RBF8	99.38%	4.07%	98.45%	95.93%	98.52%	98.26%	1.48%
RBF9	99.44%	3.73%	98.59%	96.27%	98.64%	98.44%	1.36%
RBF10	99.56%	3.39%	98.77%	96.61%	98.76%	98.79%	1.24%
Sigmoidal 1	72.52%	78.81%	58.72%	21.19%	71.45%	22.08%	28.55%
Sigmoidal 2	71.71%	77.8%	58.41%	22.2%	71.49%	22.39%	28.51%
Sigmoidal 3	81.25%	87.46%	62.78%	12.54%	71.65%	19.73%	28.35%
Sigmoidal 4	70.53%	80.17%	56.9%	19.83%	70.53%	19.83%	29.47%
Sigmoidal 5	83.18%	88.98%	63.78%	11.02%	71.77%	19.4%	28.23%
Sigmoidal 6	71.78%	77.8%	58.45%	22.2%	71.51%	22.43%	28.49%
Sigmoidal 7	71.71%	77.8%	58.41%	22.2%	71.49%	22.39%	28.51%
Sigmoidal 8	70.53%	80.17%	56.9%	19.83%	70.53%	19.83%	29.47%
Sigmoidal 9	71.71%	77.8%	58.41%	22.2%	71.49%	22.39%	28.51%
Polinomial 1	78.69%	58.14%	68.79%	41.86%	78.64%	41.94%	21.36%
Polinomial 2	62.09%	4.58%	71.05%	95.42%	97.36%	48.06%	2.64%

En las tabla anteriores es posible observar como los clasificadores que tienen como kernel una función de base radial (RBF), presentan un mayor rendimiento respecto a tasas de verdaderos positivos y verdaderos negativos respecto a los otros clasificadores, teniendo en cuenta que la variación de los parámetros  $\gamma$  y C son determinantes para el buen funcionamiento del mismo, puesto que en otras ocasiones, una mala escogencia de ellos arrojó como resultado un overfitting en el clasificador, con un 100% de verdaderos positivos y un 0% de verdaderos negativos, como fue el caso de  $\gamma = 10$  y C=1000. A continuación se presentan las gráficas que visualizan de una mejor manera como el cambio del parámetro C varía el rendimiento del clasificador para un kernel de base radial (RBF):

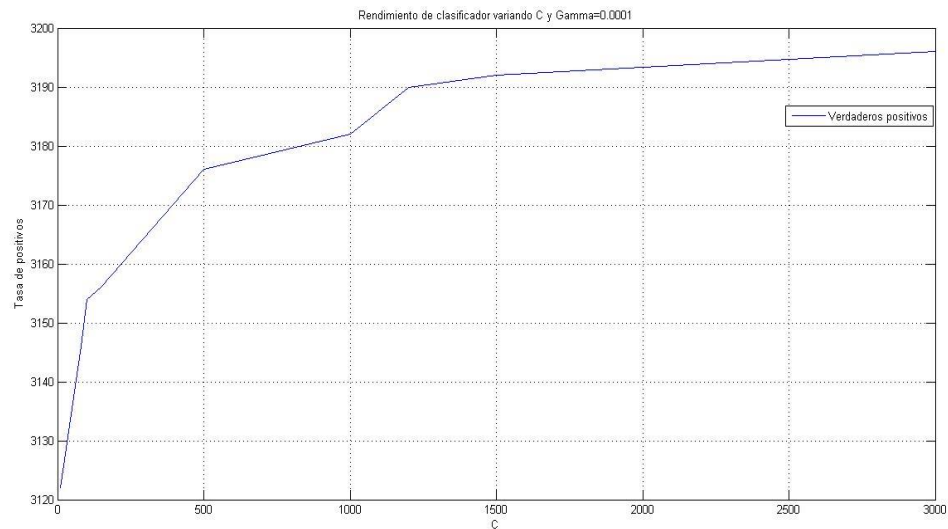


Figura 50. Variación de verdaderos positivos al cambiar el parámetro C

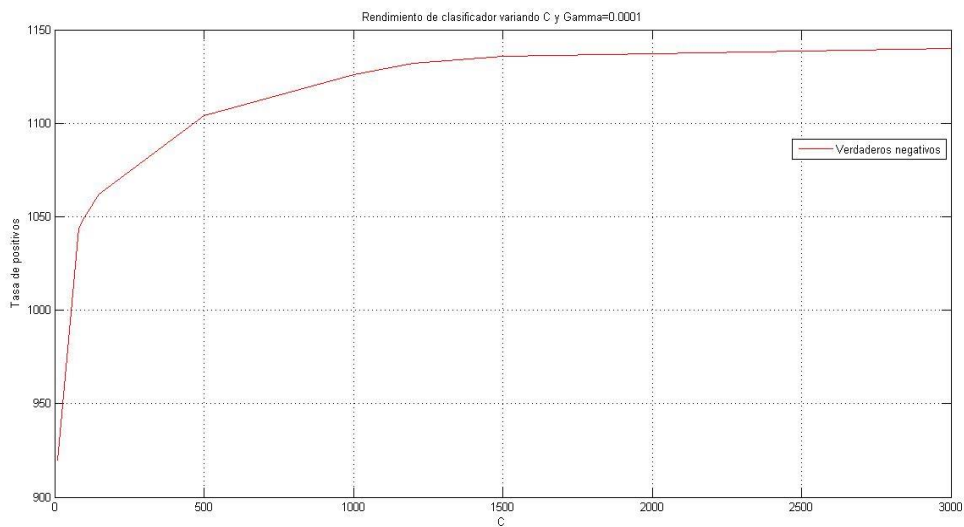


Figura 51. Variación de verdaderos negativos al cambiar el parámetro C

Por otro lado, los clasificadores que no arrojan buenos resultados fueron aquellos con un kernel sigmoideal, que pese a tener rendimientos hasta del 80% de verdaderos positivos, presentaron tasas muy bajas de verdaderos negativos, lo cual implica que en la aplicación final, se detectaría la mayor parte de un *frame* adquirido en una tubería como una falla.

Con el objetivo de escoger el mejor clasificador para realizar los siguientes entrenamientos y generar el test final, además de tener una mejor visualización de todos los resultados de la etapa de entrenamiento, se utilizó la curva ROC (*Receiver Operating Characteristic*), la cual representa gráficamente las medidas de sensibilidad contra la tasa de falsos positivos (1-especificidad) de clasificadores para determinar de forma óptima cual posee el mejor rendimiento [31], como se puede observar a continuación:



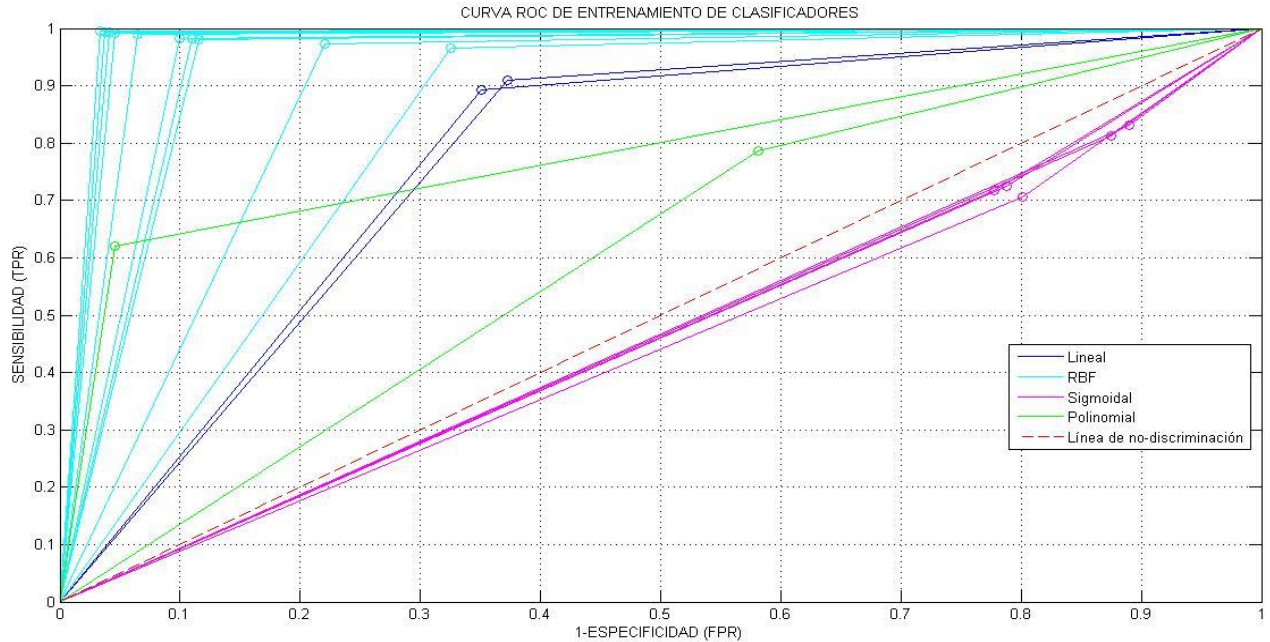


Figura 52. Curva ROC de resultados de entrenamiento de clasificadores.

Analizando una curva ROC, la mejor clasificación posible se daría en el caso en el cual un clasificador, que es representado como un punto en esta gráfica, estuviera situado en la esquina superior izquierda, lo que significaría ningún falso negativo y ningún falso positivo, que sería equivalente a poseer un área bajo la curva (AUC) igual a uno. Por otro lado, la línea punteada, denominada línea de no-discriminación indica que un clasificador situado allí generaría una predicción netamente aleatoria. Por tanto, los clasificadores que se encuentren por encima de esta línea son considerados buenos resultados y los que se encuentren por debajo son considerados clasificadores con resultados pobres [31].

Respecto a la figura 52, se pueden observar cada uno de los clasificadores entrenados y su respectiva ubicación de acuerdo a los resultados de sensibilidad y especificidad. Los clasificadores que presentan un área bajo la curva más grande son aquellos que poseen un kernel RBF, seguidos por los que tienen un kernel lineal y un kernel polinomial, los cuales se encuentran por encima de la línea de no discriminación, indicando que son buenos resultados. Por el contrario, los clasificadores de kernel sigmoidal se encuentran por debajo de esta línea, indicando que son generados pobres resultados para este problema en particular, debido a que el índice de verdaderos negativos es muy alto.

El clasificador que genera el rendimiento más alto es el indicado en la tabla 12 con el kernel RBF 10, con una sensibilidad de 99.56% y un FPR de 3.39% como se puede observar con más detalle en la figura 53. A partir de la elección de este clasificador, se realizaron los siguientes entrenamientos de tipos de fallas y las pruebas finales del sistema de detección.

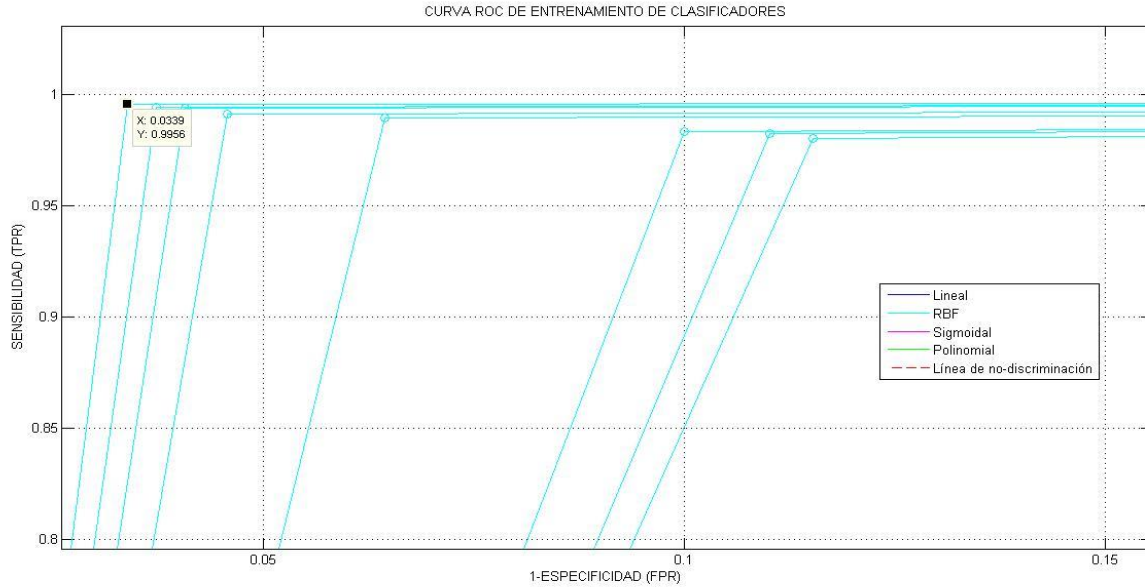


Figura 53. Curva ROC con mejor clasificador.

De acuerdo con el clasificador SVM con kernel RBF escogido, se realizaron los siguientes entrenamientos, que corresponden a los tipos de fallas determinados (Roturas, grietas y fisuras). Como ya se enunció en la sección 4.9.2.1, el entrenamiento se realiza de la forma *One vs All*, es decir, se etiqueta un tipo de falla como  $z_k = 1$  y las otras fallas y no fallas como  $z_k = -1$ , realizando así tres entrenamientos adicionales, obteniendo los siguientes resultados:

Tabla 13. Resultados de entrenamiento para tipos de falla

ENTRENAMIENTO	TPR	FPR	ACC	SPC	PPV	NPV	FDR
ROTURAS VS RESTO	95.43%	1.56%	97.88%	98.44%	93.24%	98.96%	6.76%
GRIETAS VS RESTO	97.83%	1.35%	98.43%	98.65%	96.47%	99.18%	3.53%
FISURAS VS RESTO	90.17%	0.47%	96.97%	99.53%	98.63%	96.42%	1.37%

A continuación se presenta la curva ROC para cada uno de estos tres entrenamientos:

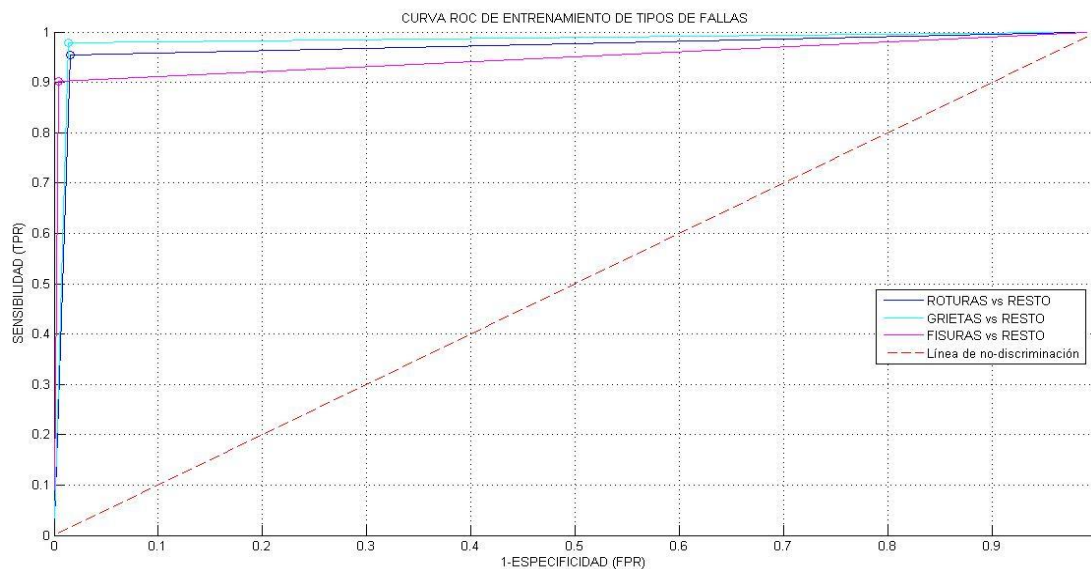


Figura 54. Curva ROC con clasificadores según tipo de falla

En la figura anterior se puede observar como los tres clasificadores se ubican en la parte superior izquierda de la gráfica, indicando que el rendimiento de ellos en la etapa de entrenamiento es muy alto y que la elección de este clasificador es pertinente no solo para discriminar fallas y no fallas, sino para determinar cada tipo de falla según el vector de características propuesto. En el siguiente apartado se mostrarán los resultados del clasificador realizados respecto a videos de prueba, analizando su rendimiento final.

### 5.3.2 EVALUACIÓN DE RENDIMIENTO ETAPA DE TEST:

Para validar el correcto funcionamiento del clasificador SVM implementado, es necesario realizar una etapa de prueba sobre videos adquiridos al interior de las tuberías con el método de luz estructurada que no fueron tenidos en cuenta para la etapa de entrenamiento con dimensiones de 1920 x 1080 pixeles, los cuales se relacionan a continuación:

Tabla 14. Videos de prueba final

NUMERO	NOMBRE	No. DE FRAMES
1	video 1.avi	120
2	video 2.avi	120
5	video 5.avi	120
7	video 7.avi	110
8	video 8.avi	120
9	video 9.avi	80
10	video 10.avi	120
11	video 11.avi	150
13	video 13.avi	180
14	video 14.avi	170
15	video 15.avi	150
16	video 16.avi	120
17	video 17.avi	140
18	video 18.avi	110
19	video 19.avi	150
20	video 20.avi	110
24	video 24.avi	150
25	video 25.avi	100
26	video 26.avi	160
27	video 27.avi	180

A continuación se pueden observar algunos *frames* de los diferentes videos de prueba adquiridos para el test final:

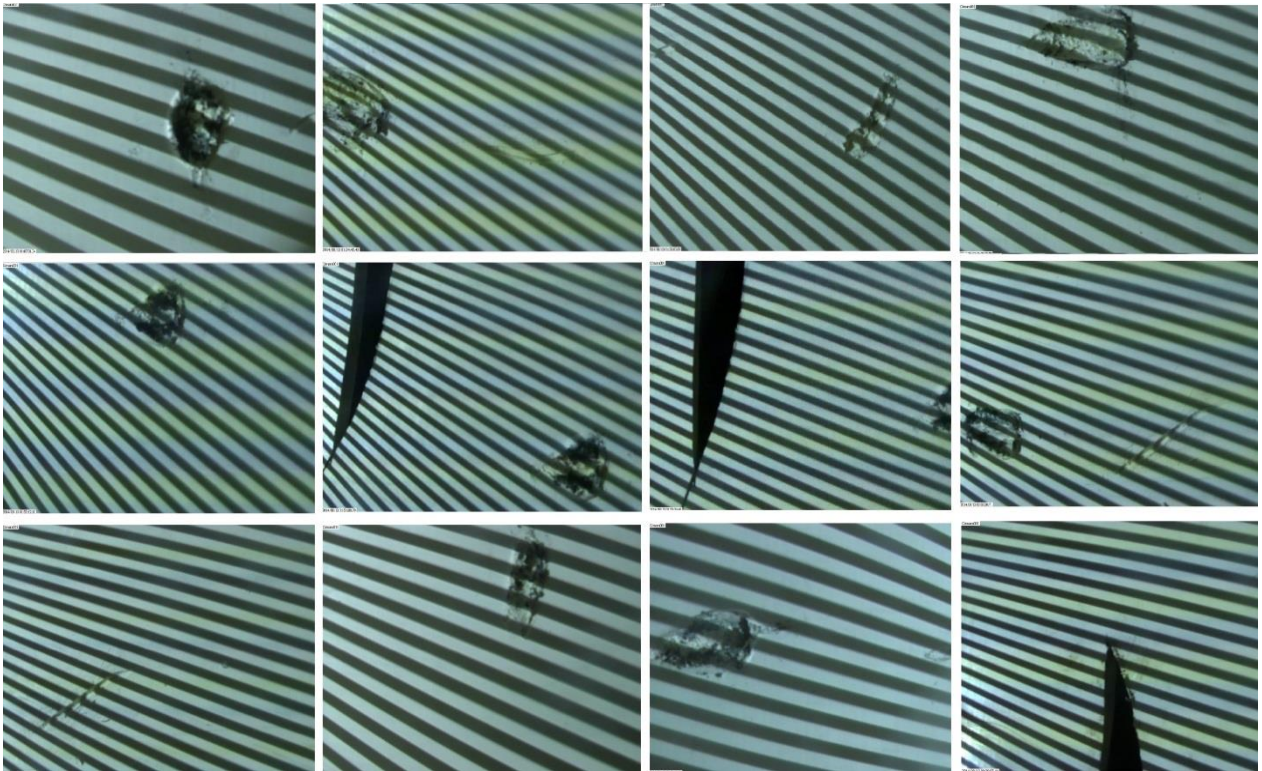


Figura 55. Ejemplos de *frames* de videos de prueba.

Los 20 videos de prueba fueron seleccionados luego de tomar los videos de los cuales se extrajeron las muestras para el entrenamiento, teniendo en cuenta además que existieran los tres tipos de fallas definidos en estas secuencias con el objetivo de evaluar el rendimiento de toda la jerarquía de clasificadores implementada.

Cada uno de estos videos es ingresado a la interfaz final implementada y explicada en la sección 4.10, donde se realiza una búsqueda exhaustiva de verdaderos positivos a lo largo de cada *frame* de la secuencia ingresada, teniendo en cuenta el tipo de falla para la visualización de cada una de ellas. Al igual que la etapa de entrenamiento, se evaluó la cantidad de verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos en cada *frame* con el fin de arrojar un consolidado final de las medidas de rendimiento del clasificador para cada uno de ellos. Para esto, se tuvo en cuenta que cuando se realiza el barrido sobre cada *frame*, se evalúan 42 cuadros en cada uno de ellos, obteniendo así una cantidad de 42xK muestras sobre cada video, siendo K el número de *frames* del mismo.

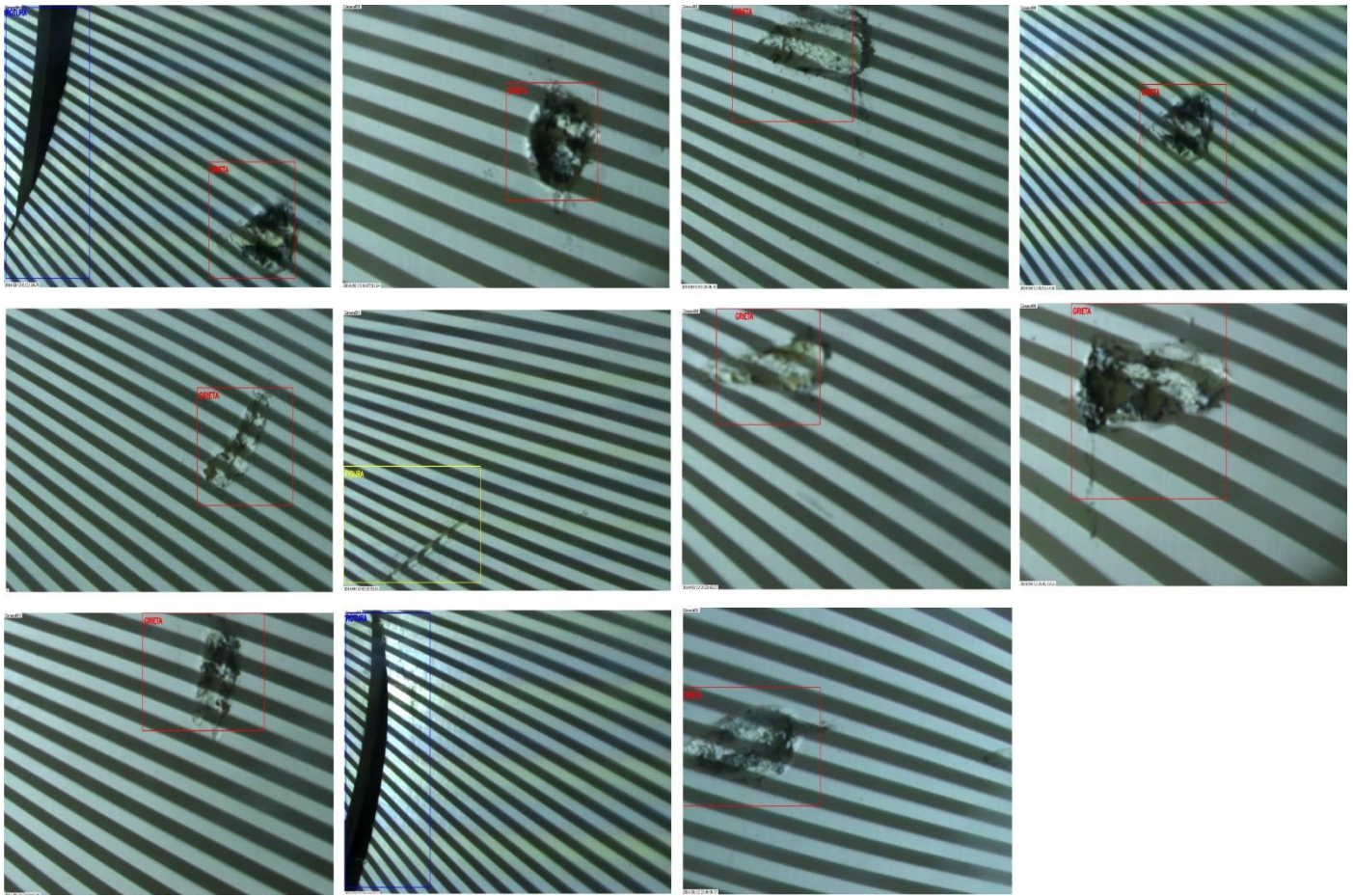


Figura 56. Ejemplos de detecciones de fallas.

A continuación se relacionan los resultados obtenidos para cada video respecto a la primera etapa del clasificador, fallas vs no fallas:

Tabla 15. Mediciones de rendimiento de clasificación de videos de prueba fallas vs no fallas.

VIDEO	TPR	FPR	ACC	SPC	PPV	NPV	FDR
Video 1.avi	91,94%	0,58%	99,10%	99,42%	87,69%	99,64%	12,31%
Video 2.avi	61,86%	0,87%	96,20%	99,13%	85,88%	96,82%	14,12%
Video 5.avi	76,79%	0,20%	99,54%	99,80%	81,13%	99,74%	18,87%
Video 7.avi	72,86%	0,51%	99,09%	99,49%	68,92%	99,58%	31,08%
Video 8.avi	83,75%	0,65%	99,11%	99,35%	67,68%	99,74%	32,32%
Video 9.avi	92,31%	0,19%	99,40%	99,81%	96,55%	99,56%	3,45%
Video 10.avi	83,33%	0,15%	99,38%	99,85%	94,34%	99,51%	5,66%
Video 11.avi	76,36%	0,16%	99,63%	99,84%	80,77%	99,79%	19,23%
Video 13.avi	57,78%	0,48%	97,78%	99,52%	83,87%	98,19%	16,13%
Video 14.avi	84,75%	0,21%	99,49%	99,79%	89,29%	99,69%	10,71%
Video 15.avi	95,00%	0,11%	99,81%	99,89%	93,14%	99,92%	6,86%
Video 16.avi	84,17%	0,77%	98,87%	99,23%	72,66%	99,61%	27,34%
Video 17.avi	90,16%	0,70%	99,20%	99,30%	57,29%	99,90%	42,71%
Video 18.avi	84,06%	2,16%	97,12%	97,84%	68,24%	99,11%	31,76%
Video 19.avi	96,43%	0,89%	99,06%	99,11%	66,26%	99,93%	33,74%
Video 20.avi	78,05%	0,62%	99,00%	99,38%	69,57%	99,60%	30,43%
Video 24.avi	92,93%	0,46%	99,11%	99,54%	93,38%	99,51%	6,62%
Video 25.avi	88,89%	0,35%	99,02%	99,65%	93,91%	99,32%	6,09%
Video 26.avi	96,67%	0,33%	99,60%	99,67%	87,88%	99,92%	12,12%
Video 27.avi	84,93%	2,33%	96,81%	97,67%	72,51%	98,89%	27,49%

A continuación se presenta la curva ROC para cada uno de los videos de prueba seleccionados, tomando como curva principal el promedio de los resultados obtenidos, con el objetivo de mirar el comportamiento final del clasificador SVM. Se obtiene como resultado una sensibilidad promedio del 83.65% y una tasa de falsos positivos (1-especificidad) del 0.62%, lo cual significa que el comportamiento del clasificador presenta muy bajos falsos positivos respecto a la totalidad de cuadros que se analizan en cada uno de los videos y la tasa de falsos negativos es más alta, es decir, existen varias fallas que no son detectadas por problemas que se detallarán más adelante.

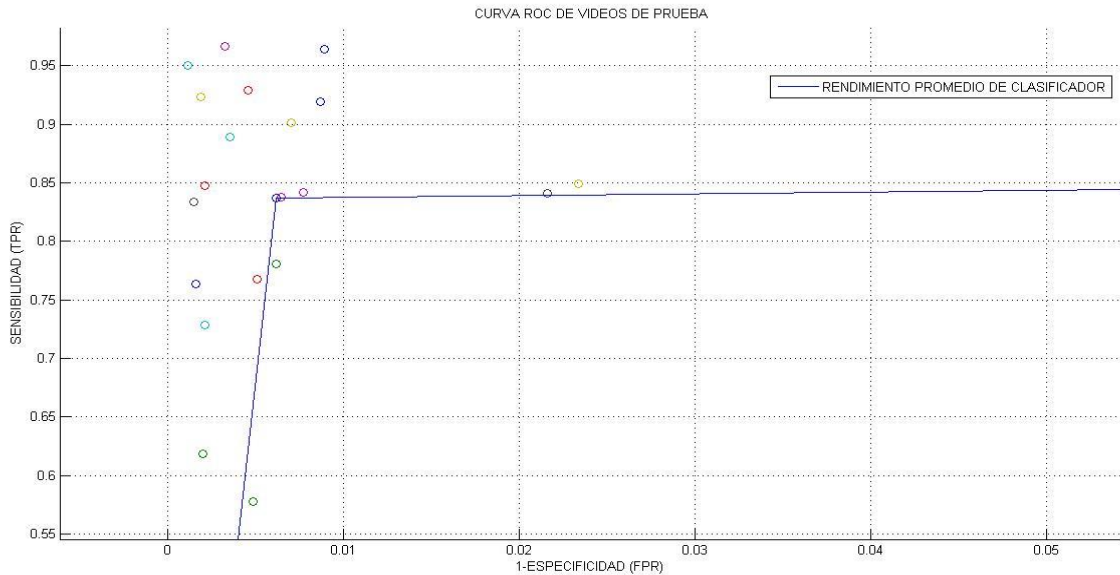


Figura 57. Curva ROC para videos de prueba

Con el fin de medir el comportamiento del clasificador respecto a cada tipo de falla, se relaciona la siguiente tabla donde se define que falla existe en cada uno de los videos de prueba y a partir de esto se relacionan las medidas de rendimiento para cada uno de ellos:

Tabla 16. Relación de clases de fallas para cada video

NOMBRE	GRIETAS	ROTURAS	FISURAS
video 1.avi	✓		
video 2.avi	✓		✓
video 5.avi	✓		
video 7.avi	✓		
video 8.avi	✓		
video 9.avi	✓	✓	
video 10.avi		✓	
video 11.avi	✓		
video 13.avi			✓
video 14.avi	✓		
video 15.avi	✓		
video 16.avi	✓		
video 17.avi	✓		
video 18.avi	✓		
video 19.avi	✓		
video 20.avi	✓		
video 24.avi		✓	
video 25.avi		✓	
video 26.avi		✓	
video 27.avi			✓

A excepción de los videos 2 y 9 que poseen dos tipos de falla, todos los videos presentan un tipo de falla, por lo que las mediciones respecto a positivos se mantiene de la misma manera. Para evaluar falsos negativos, se incluyen en los videos de una determinada falla, aquellos que aparecieron en otros videos pero que pertenecen a la falla en cuestión, como se podrá observar en las siguientes tablas y gráficas de curvas ROC.

Tabla 17. Mediciones de rendimiento de clasificación de videos con grietas

VIDEO	TPR	FPR	ACC	SPC	PPV	NPV	FDR
Video 1.avi	91,94%	0,58%	99,17%	99,49%	89,06%	99,44%	10,93%
Video 2.avi	71,67%	0,87%	97,99%	99,13%	78,18%	98,77%	21,82%
Video 5.avi	76,79%	0,20%	99,54%	99,80%	81,13%	99,54%	18,87%
Video 7.avi	72,86%	0,51%	99,09%	99,49%	68,92%	99,58%	31,08%
Video 8.avi	83,75%	0,17%	99,52%	99,83%	94,44%	99,58%	5,56%
Video 9.avi	60%	0,19%	99,5%	99,81%	71,43%	99,69%	28,57%
Video 11.avi	76,36%	0,16%	99,73%	99,84%	91,3%	99,79%	86,96%
Video 14.avi	84,75%	0,21%	99,49%	99,79%	89,29%	99,69%	10,71%
Video 15.avi	95,00%	0,11%	99,81%	99,89%	93,14%	99,52%	6,86%
Video 16.avi	84,17%	0,61%	99,03%	99,39%	77,09%	99,61%	22,9%
Video 17.avi	90,16%	0,67%	99,23%	99,33%	58,51%	99,60%	41,49%
Video 18.avi	84,06%	2,16%	97,12%	97,84%	68,24%	98,71%	31,76%
Video 19.avi	96,43%	0,81%	99,14%	99,19%	6,84%	99,83%	31,65%
Video 20.avi	78,05%	0,51%	99,11%	99,49%	73,56%	99,60%	24,37%

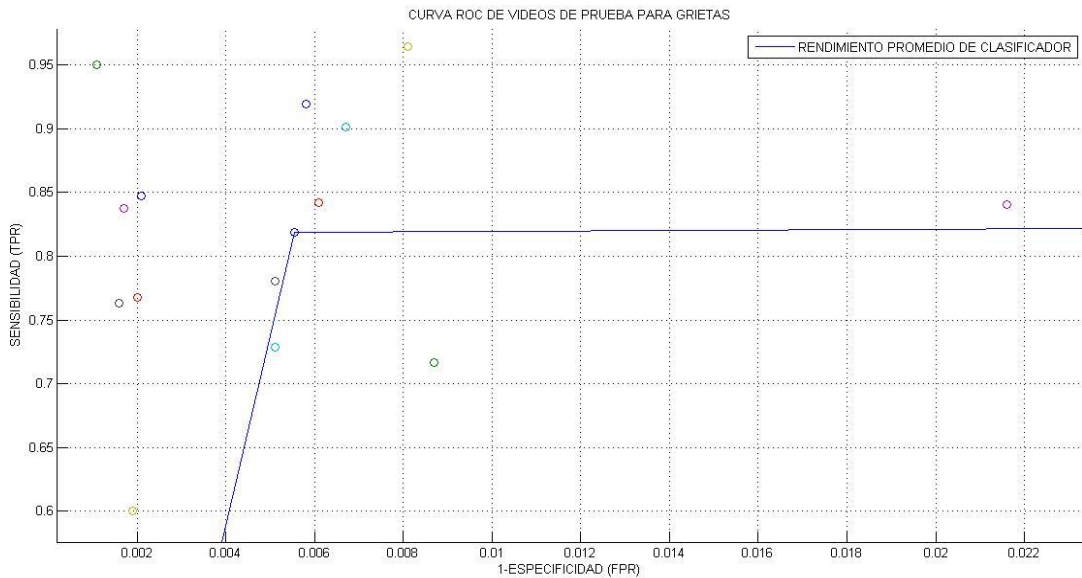


Figura 58. Curva ROC de videos de prueba para grietas

Analizando la tabla 17 y la figura 58, se puede observar que el rendimiento promedio del clasificador que discrimina grietas contra el resto de las clases es del 81.86% de sensibilidad y del 0.55% de tasa de falsos positivos, lo cual indica que este clasificador es robusto frente a posibles secciones de la tubería que son detectadas como fallas y posee una tasa más alta de verdaderos negativos, tal como ocurre en el video 9 y el video 11, que son los que presentan la tasa de verdaderos positivos más bajas (60% y 76.36% respectivamente) para esta clase, que se debe a problemas de contraste cuando existe cambio de patrón, que varias de estas grietas son poco visibles en los últimos *frames* en los que aparecen y *frames* en los cuales hay cambios bruscos debido al movimiento de la cámara, reduciendo la efectividad del clasificador para este caso. En la siguiente sección se hará referencia con más detalle a estos problemas presentados.

A continuación se presentan los resultados obtenidos para la clase rotura:

Tabla 18. Mediciones de rendimiento de clasificación de videos con roturas

VIDEO	TPR	FPR	ACC	SPC	PPV	NPV	FDR
Video 9.avi	97,45%	0,16%	99,82%	99,84%	98,71%	99,87%	1,29%
Video 10.avi	83,33%	0,15%	99,38%	99,85%	94,34%	99,51%	5,66%
Video 24.avi	92,93%	0,44%	99,13%	99,56%	93,61%	99,51%	6,39%
Video 25.avi	88,89%	0,33%	99,05%	99,67%	94,23%	99,32%	5,68%
Video 26.avi	96,67%	0,36%	99,56%	99,63%	86,57%	99,92%	13,43%

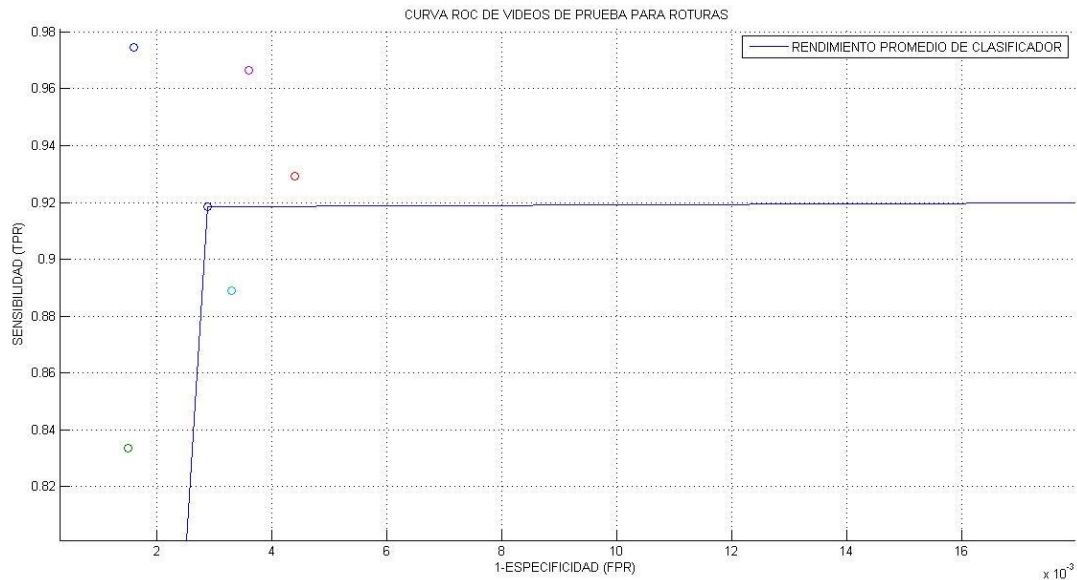


Figura 59. Curva ROC de videos de prueba para roturas

Al analizar los resultados obtenidos para el clasificador SVM que discrimina roturas de las otras tres clases definidas, es posible notar que este es el que presenta el mejor rendimiento sobre los videos de prueba, obteniendo un promedio de 91.85% de sensibilidad y un promedio de 0.29% de razón de falsos positivos, siendo las principales razones de falsos negativos en los videos 9, 24, 25 y 26 los cambios de contraste cuando ocurre un cambio de patrón proyectado por el video beam y a cambios bruscos en la escena debido a movimientos realizados por la cámara. Por otro lado, el video 10 que es el que presenta la menor tasa de verdaderos positivos (83.33%), tiene como principal fuente de error que al principio del video solo es posible observar una pequeña parte de la rotura, la cual no es inmediatamente identificada por el clasificador en estos primeros *frames*, sino cuando ya aparece en una mayor proporción en el video.

A continuación se presentan los resultados arrojados para la clase fisura:

Tabla 19. Mediciones de rendimiento de clasificación de videos con fisuras

VIDEO	TPR	FPR	ACC	SPC	PPV	NPV	FDR
Video 2.avi	55,56%	2,1%	96,06%	97,9%	54,69%	97,97%	45,31%
Video 13.avi	57,78%	1,3%	96,99%	98,7%	65,82%	98,17%	34,17%
Video 27.avi	96,67%	0,32%	96,81%	96,82%	62,78%	99,74%	31,22%



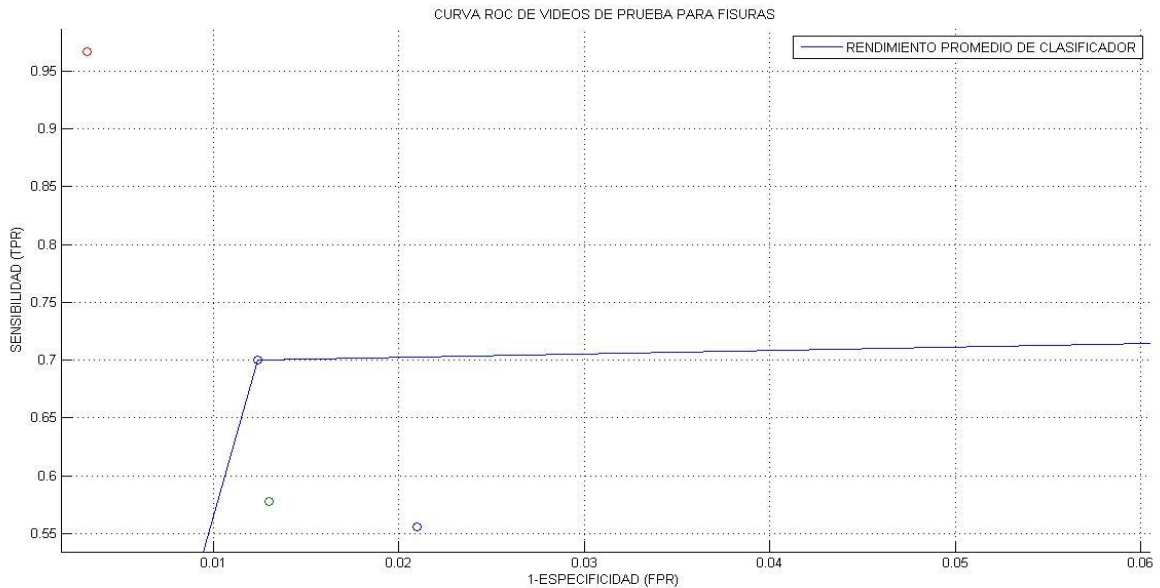


Figura 60. Curva ROC de videos de prueba para fisuras

Analizando los resultados obtenidos para la clasificación de fisuras contra el resto de clases definidas para este proyecto, se puede observar que este clasificador es el que menor rendimiento posee frente a los videos de prueba analizados, aunque se debe resaltar que esta clase es la que menos aparece en estos videos. El rendimiento promedio obtenido para este clasificador es del 70% de sensibilidad y del 1.24% de tasa de falsos negativos, siendo al igual que los anteriores clasificadores, muy robusto frente a falsos positivos pero con menores tasas de verdaderos positivos, principalmente en los videos 2 y 13, donde se presentan los menores porcentajes de estos en todos los videos analizados (55.56% y 57.78%), debido principalmente a errores en la clasificación, es decir, que el vector de características extraído para estas muestras difiere bastante respecto a las muestras con las cuales se entrenó el clasificador y son clasificadas como no fallas. Además, se debe tener en cuenta que este tipo de falla es la que presenta menor variación en los parámetros de textura respecto a las muestras que no son falla, por lo que la detección de ellas es más difícil para el clasificador.

Por otro lado, las dos fisuras que se presentan en el video 27 son en su mayoría bien clasificadas, siendo la principal fuente de error los cambios bruscos de movimiento que detecta la cámara una vez se ha movido esta.

### 5.3.3 ANÁLISIS DE ERRORES ENCONTRADOS:

Observando los resultados arrojados al evaluar cada uno de los videos de prueba por medio del clasificador SVM con kernel RBF, se encuentran ciertos errores en la discriminación que se presentan de acuerdo a ciertas características que se presentan a continuación:

- Pérdida de contraste en patrones proyectados.
- Movimientos bruscos de cámara.
- Falta de patrón en video adquirido.
- Fallas poco visibles en *frames*.
- Error aleatorio.

Estos errores poseen la siguiente distribución de porcentaje respecto al total de errores encontrados en las pruebas finales:

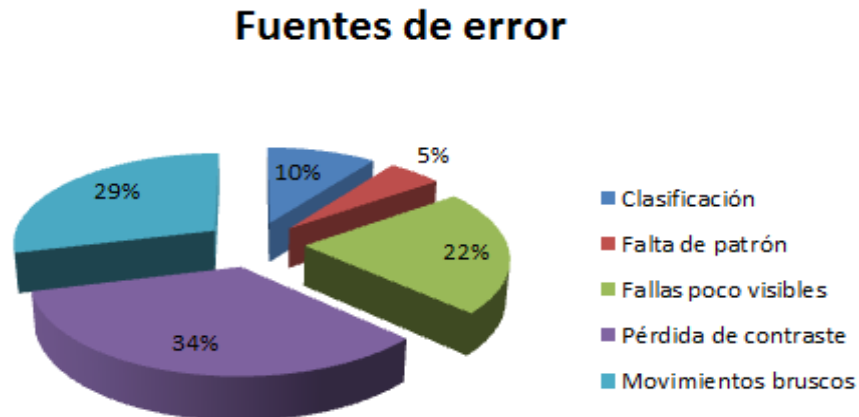


Figura 61. Fuentes de error en la clasificación

#### 5.3.3.1 PÉRDIDA DE CONTRASTE EN PATRONES PROYECTADOS:

El error más frecuente en la clasificación de los videos de prueba ocurre cuando existe el cambio de patrón positivo a negativo y el video beam en muchas ocasiones no realiza el cambio de color de forma instantánea, por lo que el contraste del patrón se pierde y por tanto en el momento de la intersección de patrones, el borde a detectar entre las líneas intersectadas no se obtendrá de forma correcta debido a que no existe un contraste alto entre los colores intersectados, como ocurre cuando existe la intersección entre líneas blanco-negro:

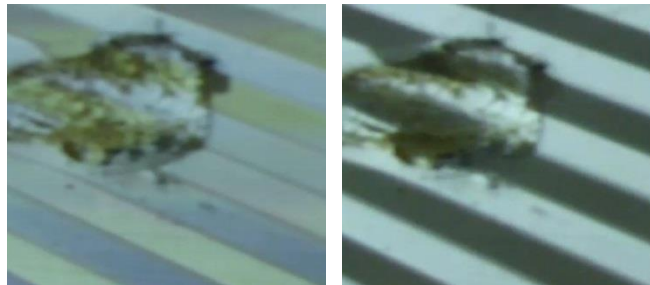


Figura 62. Par de muestras para intersección de patrones donde una resulta afectada por la pérdida de contraste en patrón proyectado.

La figura 63 presenta un ejemplo en el cual una grieta es detectada en un *frame*, mientras que en el *frame* inmediatamente siguiente no es detectada debido a este problema:

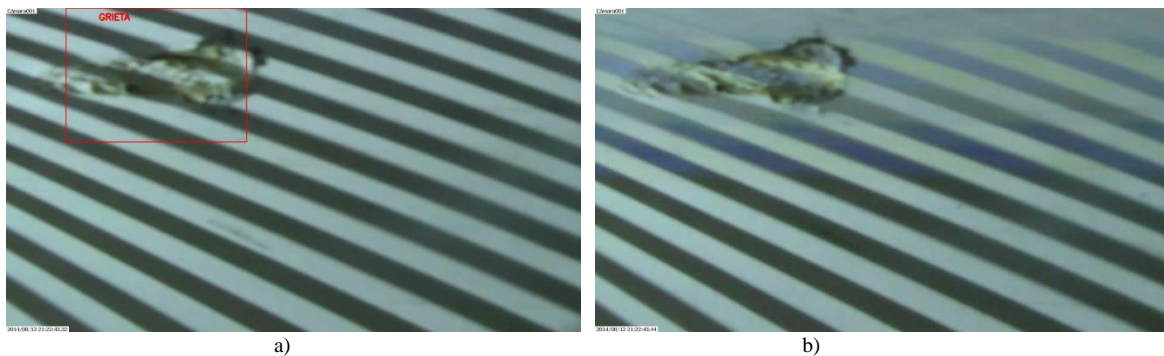


Figura 63. a) Grieta detectada correctamente. b) Grieta no detectada por problema de pérdida de contraste

### 5.3.3.2 MOVIMIENTOS BRUSCOS DE CÁMARA:

Otro de los errores frecuentes que ocurren en la detección de fallas, se presenta cuando la cámara emula el movimiento del robot al interior de la tubería y existen cambios bruscos que se alcanzan a detectar en los *frames* adquiridos debido a la velocidad que posee la cámara para adquirir *frames* (10 fps), distorsionando por tanto la escena y el mapa generado, dejando como resultado que secciones de la tubería no defectuosas sean caracterizadas como fallas (falsos positivos) o detectando una falla de una determinada clase como otra clase, el cual es el error más recurrente con este problema, debido a que los descriptores de textura pueden ser seriamente afectados por este efecto no deseado.



Figura 64. a) Grieta detectada como fisura b) Grieta no detectada y falso positivo detectado

Por otro lado, estas muestras no son detectadas forma óptima por el sistema, puesto que en la base de datos que se creó para el entrenamiento del clasificador, no se incluyeron muestras que tuvieran este problema, por lo que el sistema tiene más inconvenientes para evitar este efecto no deseado. A continuación se presenta una tabla comparativa de descriptores de textura entre una muestra de una grieta tomada de un *frame* que no tiene este problema con una que si lo posee, donde se puede observar descriptores como contraste, varianza y suma de varianza se ven seriamente afectados por este efecto de movimiento:

Tabla 20. Mediciones de rendimiento de clasificación de videos de prueba fallas vs no fallas.

CARACTERÍSTICA	MUESTRA SIN PROBLEMA	MUESTRA CON PROBLEMA
ENERGIA	0.0058	0.0015
ENTROPIA	9.5434	9.8352
CONTRASTE	12.3996	4.2747
HOMOGENEIDAD	0.525	0.4812
CORRELACION	0.9976	0.9984
VARIANZA	50.6878	36.9178
SUMA DE PROMEDIO	246.539	256.455
SUMA DE VARIANZA	10264.6	5447.42
DISIMILARIDAD	2.064	1.5094
DIFERENCIA DE VARIANZA	8.1363	1.9963
SUMA DE ENTROPIAS	0.0032	0.0012
MEDIDA CORRELACION 1	0.5624	0.5967
MEDIDA CORRELACION 2	0.9998	0.9999

### 5.3.3.3 FALTA DE PATRÓN EN VIDEO ADQUIRIDO:

Uno de los problemas que se presentó en el proceso de adquisición de los videos por medio del método de luz estructurada fue lograr la misma ubicación del video beam respecto a la cámara y la escena en cada secuencia adquirida, puesto que el tamaño del mismo dificultó la proyección de los patrones con una buena resolución al interior de la tubería. Esto generó que en ocasiones el patrón proyectado no estuviera a lo largo de la escena al final de un video, generando problemas en la intersección de patrones, puesto que no existía información suficiente en estas secciones para lograr la intersección de patrones.

Este problema da como resultado que en ocasiones existan errores en la clasificación, que se derivan en falsos positivos detectados en estas secciones, como se puede apreciar en la siguiente figura:



Figura 65. Falso positivo detectado en sección con falta de patrón

#### 5.3.3.4 FALLAS POCO VISIBLES EN *FRAMES*:

Otro problema de detección se presenta cuando la falla que está siendo capturada está desapareciendo de la escena, dejando como resultado su no detección por parte del clasificador. Esto se debe principalmente a que las características de textura, las cuales se calculan a partir del análisis de distribución de grises de toda la imagen, pueden variar radicalmente respecto de una falla que se presenta en su totalidad en el *frame*.

El efecto que genera esto, se nota principalmente sobre la clase grieta, sobre la cual ocurre más este problema. Además, el clasificador no fue entrenado con este tipo de muestras, por lo que puede presentar falsos negativos en estas secciones:

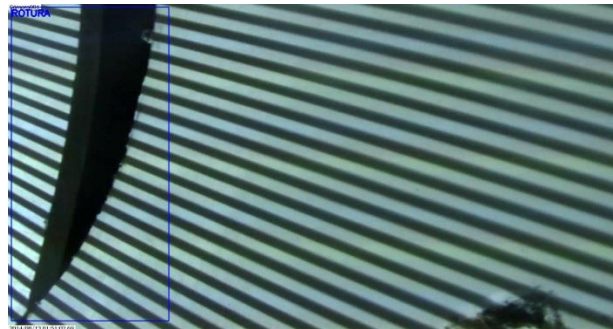


Figura 66. Grieta no detectada en la parte inferior del *frame*.

#### 5.3.3.5 ERRORES DE CLASIFICACIÓN:

La última causa de error encontrada en el sistema de clasificación, es cuando se presenta un error aleatorio en la discriminación de clases, donde algunas muestras son ubicadas en una región del espacio n-dimensional de clasificación utilizado que no corresponde a las de su clase, generando así falsos positivos o falsos negativos según el caso.

Se debe tener en cuenta que todo clasificador posee en su entrenamiento y test un error en su resultado final, dejando una tasa de falsos positivos y falsos negativos que puede ser baja cuando el mismo es entrenado con las características, parámetros y número de muestras adecuadas. Sin embargo, el clasificador que presenta más de este tipo de error es el que discrimina fisuras de las otras tres clases, principalmente porque en los videos 2 y 13, las fisuras que se presentan allí no son del todo analizadas, debido principalmente a que sus características extraídas pueden ser muchas veces comparables con las características de la clase no falla.

**5.4 ESTUDIO DE COSTOS:**

Este trabajo de grado fue sustentado con los recursos del autor y del departamento de Ingeniería Electrónica de la Pontificia Universidad Javeriana. A continuación se muestra una tabla donde se discriminan los costos reales y el valor aproximado a cobrar por el proyecto en caso de venderlo.

Durante la realización de este trabajo de grado se trabajó un promedio de 30 horas semanales durante 40 semanas lo que equivale a un total de 1200 horas.

Tabla 21. Costos reales de los recursos utilizados para la realización del Trabajo de Grado.

Recurso	Unidad	Costo unidad	Total Unidades	Costo Total
Investigador	Hora	20.000	1200	24'000.000
Director	Hora	60.000	120	7'200.000
Uso de recursos del Laboratorio, Computador de la Facultad, video beam Internet, etc.	Créditos-Semestre	375.400	6	2'252.400
Impresión papelería	Unidad	200	400	80.000
Tubos PVC de prueba	Metro	30.000	2	60.000
Computador para programación y ejecución	Unidad	1'600.000	1	1'600.000
Costo total proyecto				35'192.400

**5.5. TRABAJOS E INVESTIGACIONES A SEGUIR:**

La implementación del sistema automático de fallas al interior de tuberías hidrosanitarias tiene como finalidad ser implementada en un futuro a la plataforma robótica de exploración que actualmente está desarrollando el departamento de electrónica de la Pontificia Universidad Javeriana para la empresa Hidráulica & Urbanismo. Para esto, uno de los siguientes trabajos a seguir consiste en integrar el software desarrollado a la tarjeta de procesamiento que va a tener el robot, diseñando un sistema embebido que permita manejar el código implementado en OPENCV para una aplicación final, disminuyendo los tiempos de procesamiento para ejecutarlo en tiempo real.

Para esta aplicación final, se debe tener en cuenta una integración mecánica entre el proyector y la cámara, con el fin de moverse a lo largo de la tubería de forma sincronizada y evitando problemas como el de la falta de patrón en algunas secciones y asegurar que la intersección de los patrones por medio de luz estructurada sea óptima en cada movimiento del robot. Además, debe incluirse un picoprojector, que permita proyectar los patrones a cortas distancias, que su manejo al interior de las tuberías no sea dispendioso y que posea un cambio de contraste más rápido que el video beam utilizado para este proyecto, evitando así problemas como el de pérdida de contraste cuando existe cambio de patrón.

Por último, otras investigaciones futuras podrían enfocarse en la implementación como tal del sistema, es decir, investigar métodos de reconstrucción de escena por luz estructurada más óptimos y compararlos con los actuales, además de aumentar la base de datos de los tipos de falla con el objetivo de mejorar los resultados de prueba de los clasificadores que se implementen.

# CAPÍTULO 6

## CONCLUSIONES

La motivación de realizar un sistema de identificación automático de fallas para tuberías hidrosanitarias, surgió de la idea de integrar esta solución en un futuro a una plataforma de exploración robótica que se está desarrollando actualmente en el departamento de electrónica de la Pontificia Universidad Javeriana y con esto, disminuir y evitar los problemas que surgen cuando un operario de forma manual intenta detectar en todo instante las fallas en estas tuberías, generando posibles adversidades respecto a pérdida de agua o disminución de tasa de flujo de la misma cuando existen defectos al interior de ellas.

El desarrollo de este sistema permitió como primer logro la adquisición de una base de datos de 70 videos al interior de tuberías de prueba que poseen algún tipo de defecto a través del método de luz estructurada, el cual por medio de una cámara y un proyector, se detallan características relevantes de textura de las fallas que se pueden encontrar, en el momento en el cual se deforma el patrón proyectado sobre estas, entregando así al clasificador implementado características más relevantes para realizar la clasificación. Por otro lado, este método de luz estructurada evita uno de los problemas principales de visión estéreo, el cual es la costosa búsqueda de correspondencias entre las imágenes tomadas por cada una de las cámaras. Además, para la implementación futura del sistema en la plataforma robótica, resulta más conveniente la integración de un picoprojector junto a la cámara que otra cámara adicional para generar visión estéreo, debido a las dimensiones y forma del robot que se desea construir.

Para la implementación del método de intersección de patrones por luz estructurada es pertinente realizar un proceso de calibración de cámara y proyector puesto que con ella es posible eliminar los efectos de distorsión radial y tangencial que introducen estos elementos, mejorando así los mapas generados que se obtienen intersectando patrones inversos. Esto se realiza a partir de los parámetros intrínsecos y de distorsión que entrega el algoritmo de calibración, teniendo en cuenta que este proceso mejora cuando se realiza la adquisición con un enfoque manual que con uno automático para el caso de la cámara, entregando un error de reproyección promedio de 0.4493 píxeles. El error de reproyección obtenido para el proyector fue de 0.25 píxeles.

La utilización de una máquina de soporte vectorial (SVM) con un kernel de base radial (RBF) resulta conveniente para la detección del tipo de fallas definido a través del vector de 14 características construido, por medio de trece características de textura que se calculan a través del promedio de cuatro matrices de co-ocurrencia y de una característica geométrica (Longitud de arco), obteniendo una sensibilidad del 99.56% en el entrenamiento y una sensibilidad máxima del 96.67% en uno de los videos de prueba. Por otro lado, la implementación de una jerarquía de clasificadores para discriminar cada una de las fallas por medio del método de entrenamiento *one vs all*, dejó como resultado una exactitud máxima del 98.43%.

Al evaluar y comparar los clasificadores SVM entrenados, se determinó que aquellos que poseían un kernel sigmooidal no eran convenientes para esta aplicación, entregando resultados muy bajos, al igual que los que poseían kernel lineal y polinomial que no alcanzan a tener rendimientos tan altos como el kernel RBF, que de acuerdo a buena selección de los parámetros  $C$  y  $\gamma$  alcanzan los resultados ya descritos. Por medio del método *Grid Search* fue posible determinar que el mejor clasificador SVM con un kernel RBF tiene un parámetro de  $C$  de 3000 y  $\gamma$  de 0.0001.

Las principales fuentes de error del sistema de detección automático surgen cuando existen cambios de contraste erróneos por parte del proyector utilizado en el momento en el cual hay cambio de patrón en la secuencia de video, generando en su mayoría falsos negativos en el sistema. Otras fuentes de error existentes ocurren cuando hay movimientos bruscos en la cámara simulando estos movimientos del robot, cuando el patrón no se proyecta sobre la totalidad de la escena, fallas poco visibles y errores aleatorios surgidos del error que presenta un clasificador de aprendizaje de máquina.

Los resultados obtenidos en este proyecto, pueden mejorarse al implementar primero un proyector que evite los problemas de contraste y de movimiento al interior de una tubería, además de incrementar la tasa de *frames* por segundo en la cámara escogida, de tal manera que los cambios bruscos en el video puedan tomarse de una mejor manera. Por tanto, el clasificador implementado podría tener un mayor rendimiento y podría ser probado en ambientes más amplios y con menores restricciones respecto al montaje de adquisición.

Como ya se dijo anteriormente, este estudio tiene como finalidad servir de base ante una futura implementación del sistema automático de fallas en la plataforma robótica de exploración de tuberías hidrosanitarias que actualmente se está desarrollando en la Pontificia Universidad Javeriana, de tal manera que sea posible diseñar un sistema embebido que sea capaz de detectar las fallas definidas en tiempo real y que sirva de apoyo para el proceso de reparación que lo lleva a cabo la empresa Hidráulica & Urbanismo.

# BIBLIOGRAFÍA

- [1] X. J.Wang, M. F. Lambert, A. R. Simpson, and J. P.Vitkovsky, “*Detection in pipelines and pipe networks: A review*”, in 6th Hydraulics Civil Eng.: State of Hydraulics Conf., Barton, Australia, 2001, pp. 391–400.
- [2] M. Lahlou, “*Leak detection and water loss control*”, Tap Mag., vol. 1, no. 1, May 2001, Tech. Brief.
- [3] “*Diseño y construcción de una plataforma robótica de exploración y reparación de tuberías hidrosanitarias, operada remotamente*”, Informe técnico interno, Pontificia Universidad Javeriana, 2013.
- [4] M. Yang and T. Su, “*Automated diagnosis of sewer pipe defects based on machine learning approaches*”. In: Department of Civil Engineering, National Chung Hsing University, ScienceDirect, Taiwan, pp. 1327-1337, 2007.
- [5] K. Fu, R. González. C, Lee. “*Robótica, Control, Detección Visual e Inteligencia*”. In: Madrid, Mc Graw Hill, pp. 276-281, 1988.
- [6] J. Salvi, X. Armangué and J. Batlle, “*A comparative review of camera calibrating methods with accuracy evaluation*”. In: Institute of Informatics and Applications, University of Girona, Elsevier Science, Girona, Spain, pp. 1617-1635, 2002.
- [7] A. Jain. “*Fundamentals of Digital Image Processing*”. Universidad de California, Prentice Hall International, pp. 244-291, 347-355, 407-410, 414-417, 1991.
- [8] R. Duda. P, Hart. D, Stork. “*Pattern Classification*”. Nueva York, Wiley-Interscience Publication, pp. 20-33, 114-124, 174-193.
- [9] UNAM, [En línea]. [Consultado 18 de Marzo de 2013]. Disponible en Internet: <http://redyseguridad.fi-p.unam.mx/proyectos/biometria/basesteoricas/reconocimiento.html>
- [10] OPENCV. [En línea]. 2014. [citado 23 de abril de 2014], Formato HTML, disponible en internet: <http://opencv.org/>
- [11] R. Haralick, K. Shanmugan and I. Dinstein, “*Textural Features for Image Classification*”. In: School of Electrical Engineering and the Center for Research, University of Kansas, IEEE Transactions on Systems Man and Cybernetics, Vol. SMC-3, No.6, USA, pp. 610-621, 1973.
- [12] G. Bradski and A. Kaehler . “*Learning OpenCV*”. Sebastopol, O’Reilly Media, pp. 162, 222-234, 370-396, 405-415, 459-517, 2008.



- [13] M. Yang and T. Su, “*Segment ideal morphologies of sewer pipe defects on CCTV images for automated diagnosis*”. In: Department of Civil Engineering, National Chung Hsing University, ScienceDirect, Taiwan, pp. 3562-3573, 2006.
- [14] C. Duarte. “*Estado del arte de las tecnologías de inspección de sistemas de alcantarillado*”. In: Facultad de ingeniería, Universidad de los Andes, Bogotá D.C, pp. 7-9, 2009.
- [15] Sony Electronics Inc. “*Network Camera*”. In: Sony Corporation, Park Ridge, USA, pp 1-2, 2012.
- [16] DELL, Inc. “*Proyector Dell 2400MP*”. In: Dell Corporation, Park Ridge, USA, pp 1, 2006.
- [17] J. Salvi, J. Pagès and J. Battle, “*Pattern codification strategies in structured light systems*”. In: Informatics and Applications Institute, Girona University, Elsevier, Spain, pp. 827-849, 2003.
- [18] Z. Zhang, “*A Flexible New Technique for Camera Calibration*”. In: Microsoft Research, Redmond, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No.11, USA, pp. 1330-1334, 2000.
- [19] I. Din, H. Anwar, I Syed, H. Zafar and L. Hasan, “*Projector Calibration for Pattern Projection Systems*”. In: Department of Electronics Engineering, Incheon National University, Incheon, South Korea, pp. 80-86, 2014.
- [20] G. Gerig, “*Structured Lighting*”. In: School of Computer Science, Carnegie Mellon University, Pittsburgh, USA pp. 1-51, 2012.
- [21] K. Konolige, “*Projected Texture Stereo*”. In: Willow Garage, Menlo Park, USA, pp. 1-8, 2009.
- [22] J. Pareja, J. Rodríguez. “*Plataforma Robótica de exploración y reparación de sistemas hidrosanitarios*”. In: Pontificia Universidad Javeriana, Bogotá D.C, 2012.
- [23] Y. Wang, “*Image Filtering: Noise Removal, Sharpening*”. In: School of Computer Science, Carnegie Mellon University, Pittsburgh, USA pp. 1-51, 2012.
- [24] Universidad de Sevilla. “*Métodos Estadísticos Basados en la Transformada Wavelet*”. In: Escuela Superior de Ingenieros, Sevilla, pp.1-13.
- [25] J. Botero, A. Restrepo. “*Análisis de textura en panes usando la matriz de co-ocurrencia*”. In: Instituto Tecnológico Metropolitano, Revista Politécnica, No. 10, Colombia, pp. 74-80, 2010.
- [26] B. Park, Y. Chen, “*Co-occurrence Matrix Texture of Multi-Spectral Images on Poultry Carcasses*”. In: Instrumentation and Sensing Laboratory, Beltsville Agricultural Research Center, Idea Library, Beltsville, USA, pp. 127-139, 2001.
- [27] H. Lin, J. Wang and S. Liu, “*The Classification Study of Image Texture based on the Rough Set Theory*”. In: College of mathematical information, Zhejiang Normal University, IEEE International Conference on Granular Computing, Jinhua, China, pp. 720-723, 2010.
- [28] F. Malik, B. Baharudin. “*Analysis of distance metrics in content-based image retrieval using statistical quantized histogram texture features in the DCT domain*”. In: Computer and Information Sciences Department, University Technology of Petronas, Journal of King Saud University, Malaysia, pp. 207-218, 2013.

[29] H. Shunjie, C. Qubo and H. Meng, “*Parameter selection in SVM with RBF kernel function*”. In: College of Electric and Electronic Engineering, Changchun University of Technology, Changchun, China, pp. 1-4.

[30] AULACLIC. [En línea]. 2014. [citado 10 de Agosto 2014], Formato HTML, disponible en internet: <http://aulaclinic.es/>

[31] T. Fawcett, “*An introduction to ROC analysis*”. In: Institute for the Study of Learning and Expertise, Science Direct Pattern Recognition Letters, Palo Alto, USA, pp. 861-874, 2006.

# ANEXOS

En los DVD's adjuntos al documento se podrán encontrar los anexos del proyecto, los cuales incluyen la base de datos recolectada, los videos de prueba finales así como el código fuente de los algoritmos implementados en OPENCV.