



Clasificación de Texturas por la Transformada de Cúmulos Coordinados



Para la obtención de grado de Maestro en
Óptomecatrónica

Asesor: Dr. Francisco J. Cuevas de la Rosa

Alumno: Axel Herrera Cabrera

Versión Final

Agosto de 2019
León, Guanajuato, México

Dedicados a mis padres.

Resumen

Uno de los temas más abordados en aprendizaje automático está relacionado con el reconocimiento de patrones. Avances recientes en reconocimiento de patrones han aumentado la necesidad de mejorar los algoritmos para obtener características especiales de las texturas para poder realizar una mejor clasificación, la mayoría de estudios sobre análisis de texturas sólo se han llevado a cabo en un pequeño número de áreas por lo que aun sigue siendo limitado.

En el presente trabajo se propone un método de análisis de texturas el cual es la Transformada de Cúmulos Coordinados o Representación de Cúmulos Coordinados (CCR por sus siglas en ingles) este método calcula la estadística de una imagen binarizada que queda en un vector de ocurrencias, el cual es la información de la textura.

Un desafío de esta técnica es reconocer texturas invariantes, específicamente a rotación por lo que una de las mejoras que se tiene es ocupando la transformada Discreta de Fourier en una dimensión y obteniendo su espectro se mantendrá dichas características, y realizando una serie de experimentos se demostrará la eficacia del mencionado método y el de la mejora.

Agradecimientos

Agradezco a todos aquellos que permitieron llegar a la meta de finalizar la maestría.

En primer lugar a mis padres por que ellos me impulsaron a seguir preparándome y me dieron mucho apoyo de todas las maneras posibles.

A mi novia Mary, que en estos 2 años en que nuestra relación ha sido a distancia se ha convertido un pilar fundamental para seguir adelante con el posgrado.

A mi hermana Vania, que ella ha demostrado tener el temple y la fortaleza que ha sido inspiración para mi en mi vida cotidiana y profesional.

A mis compañeros de cubo, José, Jorge, Abril y Milvia que en los meses que estuve escribiendo la tesis hicieron muy ameno estar en el cubo y de igual manera por sus observaciones y sus sugerencias para mejorar los escritos.

Al CONACYT y al CIO por el apoyo que brindaron para que pudiera llegar a finalizar el proyecto de maestría.

Al Dr. Francisco J. Cuevas de la Rosa que fue el asesor que permitió que el proyecto fuera avanzando y logrando llegar en tiempo y forma.

Índice general

1. Introducción	12
1.1. Objetivo de la tesis	12
1.2. Planteamiento del problema	12
1.3. Antecedentes	13
1.4. Procesamiento digital de imágenes	15
1.5. Textura	18
1.6. Propiedades de la textura	20
1.7. Análisis de texturas	23
1.7.1. Métodos geométricos	24
1.7.2. Métodos basados en modelos	25
1.7.3. Métodos basados en señales	26
1.7.4. Método estadístico	26
1.7.5. Clasificación de texturas	27
1.8. Análisis del trabajo	29
1.9. Acerca del software	30
2. Transformada de Cúmulos Coordinados	32
2.1. Transformada de Cúmulos Coordinados	33
2.2. Propiedades de la CCR	36

<i>ÍNDICE GENERAL</i>	5
2.3. Algoritmo de la CCR	39
2.4. Método de binarización	43
2.5. Método de clasificación	44
2.5.1. Distancia de Hamming	46
3. Métodos invariantes	48
3.1. Introducción a la invarianza de una imagen	48
3.2. Categorías de Invarianza	50
3.3. Momentos invariantes a rotación, traslación y escala	52
3.3.1. Invarianza a traslación	53
3.3.2. Invarianza de escala uniforme	53
3.3.3. Invarianza a rotación	54
3.3.3.1. Patrones Binario Locales Uniformes	55
3.3.3.2. Transformada Discreta de Fourier	58
3.3.3.3. Propiedades de la transformada de Fourier	59
3.4. Volviendo Invariante la CCR	61
4. Experimentos	63
4.1. Experimento 1: Clasificación de texturas con ruido ocupando la CCR	63
4.2. Experimento 2: Comparativo de clasificación de texturas con la CCR y LBP	68
4.3. Experimento 3: Clasificación de texturas invariante a rotación por la CCR/EF1D	71
4.4. Experimento 4: Método comparativo CCR/EF1D y LBPUHF para clasificar texturas invariantes a rotación	74
4.5. Experimento 5: Método comparativo CCR/EF1D y LBPUHF para clasificar texturas sin procesamiento invariantes a rotación	80

<i>ÍNDICE GENERAL</i>	6
5. Conclusiones	84
5.1. Trabajo a futuro	88
A. Reconocimientos	89

Índice de figuras

1.1. Ejemplo de una textura	14
1.2. Principio de procesamiento de imágenes	17
1.3. Corteza de árbol en la que se observa porque su rugosidad no permite un mejor análisis.	20
1.4. Regularidad de una textura	22
1.5. Ejemplo de la aspereza en la piedra de granito.	22
1.6. Se puede apreciar que gracias a los bordes nos permite determinar hacia donde se tomó la foto para determinar la dirección que tiene la textura	23
2.1. Representación de la CCR con su ventana de escaneo	33
2.2. Patrones binarios de acuerdo a la ventana de 3x3 su número en binario y su decodificación en decimal para el histograma de fre- cuencias de la CCR.	35
2.3. Cálculo del histograma de la imagen (D9) de escala de grises junto con la binarización de la imagen y su respectivo histograma de la CCR	35
2.4. Imágenes binarizadas junto con sus respectivos histogramas de frecuencias de la CCR	39

<i>ÍNDICE DE FIGURAS</i>	8
2.5. Binarización por el método de Otsu	44
2.6. Pasos para la clasificación de patrones	45
2.7. Funciones generales de discriminación	46
3.1. Dos clases definidas idealmente en el cual se muestra la variedad de posiciones y direcciones que tienen sus miembros	50
3.2. Ventana del LBPU revisando el pixel central y calculando los pesos de los pixeles vecinos	56
3.3. Los 36 patrones binarios invariantes de rotación únicos que pue- den ocurrir en el conjunto adyacente simétrico circular de LBP. Círculos en blanco y rojo corresponden a valores de bit de 0 y 1 en la salida de 8 bits del operador. La primera fila contiene los nueve patrones uniformes y los números dentro de ellos corresponden a sus únicos códigos LBP.	57
3.4. Espectro de Fourier	59
3.5. Columna a) Imagen original e imagen rotada a 135° Columna b) Histograma de la CCR normalizado de la imagen rotada y no rotada se puede observar como existe un cambio en el histogra- ma de frecuencias al rotar la imagen Columna c) Histograma del Espectro de Fourier de una dimensión (EF1D) ahí se puede ob- servar que no hay una variación visible o es mínima que no afecta la detección del patrón que es.	62
4.1. Conjunto de texturas generadas computacionalmente	64
4.2. Eficiencia del método	66
4.3. Patrones de texturas con 30% de ruido	67
4.4. Gráfica de eficiencia de la CCR y LBP	70
4.5. Base de datos de Texturas tomadas del álbum de Brodatz	71

<i>ÍNDICE DE FIGURAS</i>	9
4.6. Conjunto de texturas rotadas a 5°, 10°, 15°, 45°, 135°, 225° y 315° para mejorar el método invariante a rotación	72
4.7. Eficiencia del experimento	73
4.8. Base de datos de texturas del álbum de Brodatz	74
4.9. Eficiencia de la CCR/EF1D y LBPUHF	78
4.10. Imagen Brodatz D38 H.E. como se observa tiene pixeles negros que al rotar afectan la detección adecuada	79
4.11. Base de datos de las diferentes texturas	80
4.12. Eficiencia de ambos métodos de las tablas 4.9 y 4.10	83

Índice de tablas

4.1. Valores aleatorio y su representación en binario	64
4.2. Resultados del experimento 2 tomando como nomenclatura: a: Patrón 22, b: Patrón 91, c: Patrón 387, d: Patrón 5767, e: Pa- trón 1304402, f: Patrón 3707135, g: Patrón 6039510, h: Patrón 1763554, i: Patrón 26625306 e j: Patrón 28625692.	69
4.3. Respuesta del método de la CCR/EF1D, donde: 1(D9), 2(D11), 3(D24), 4(D112), 5(D12 H.E.), 6(D16), 7(D19 H.E.), 8(D65), 9(D84 H.E.) Y 10(D29)	72
4.5. Eficiencia de cada ángulo	73
4.6. Resultados del método CCR/EF1D, donde: 1(D9), 2(D11), 3(D24), 4(D112), 5(D12 H.E.), 6(D16), 7(D19 H.E.), 8(D65), 9(D84 H.E.), 10(D27), 11(D38 H.E.), 12(D92 H.E.), 13(HH), 14(G), 15(D101), 16(D4), 17(D12), 18(D74), 19(D98) y 20(D110).	75
4.7. Resultados del método LBPUHF, donde: 1(D9), 2(D11), 3(D24), 4(D112), 5(D12 H.E.), 6(D16), 7(D19 H.E.), 8(D65), 9(D84 H.E.), 10(D27), 11(D38 H.E.), 12(D92 H.E.), 13(HH), 14(G), 15(D101), 16(D4), 17(D12), 18(D74), 19(D98) y 20(D110).	76
4.8. Tabla de eficiencia del método CCR/EF1D	77
4.9. Tabla de eficiencia del método LBPUHF	77

4.10. Resultados del método CCR/EF1D, donde: 1(frijol), 2(parrilla), 3(Reja), 4(Toalla), 5(Gorro), 6(Almohada), 7(Piso), 8(Pared), 9(Piso exterior) y 10(Pared exterior)	81
4.11. Resultados del método LBPUGH, donde: 1(frijol), 2(parrilla), 3(Reja), 4(Toalla), 5(Gorro), 6(Almohada), 7(Piso), 8(Pared), 9(Piso exterior) y 10(Pared exterior)	81
4.12. Tabla de eficiencias del método CCR/EF1D	82
4.13. Tabla de eficiencia del método LBPUGH	82

Capítulo 1

Introducción

1.1. Objetivo de la tesis

El presente trabajo de investigación tiene como objetivo principal el diseñar y construir un algoritmo para clasificar y reconocer texturas del tipo natural o artificial (hecha computacionalmente). Para ello, se plantean los objetivos específicos:

- Realizar la revisión, estudio y análisis de la Transformada de Cúmulos Coordinados.
- Creación de una base de datos de imágenes de patrones periódicos creados computacionalmente y de texturas reales por el álbum de Brodatz.
- Realizar una mejora para la detección de texturas invariante a rotación.

1.2. Planteamiento del problema

La clasificación y reconocimiento de objetos siempre ha sido una actividad cotidiana del ser humano mediante el uso de los sentidos con los que cuenta (olfato, oído, tacto, gusto y vista). Por ejemplo, con la utilización de la vis-

ta el ser humano puede reconocer objetos sin la necesidad de tocarlos. En el sector industrial el ser humano comete diferentes tipos de errores en áreas de calidad debido a que el ojo humano no tiene la capacidad para trabajar de manera ininterrumpida, realizando una revisión de los objetos que se están fabricando en una línea de producción, Scharcanski [1] explica esto basándose en que la experiencia empírica que tiene los operadores es subjetiva y puede tener problemáticas ya que la visión humana es limitada en términos de habilidad y distinción, por lo que los sistemas automáticos completan el trabajo que no puede realizar eficientemente el ser humano. En el área de texturas, se tiene que realizar un análisis más minucioso mismo que el ser humano con el sentido de la vista no realiza de manera eficiente en los diferentes procesos de clasificación. Sánchez et al. [2] menciona que en la industria, los sistemas automáticos dan beneficios que son de tomar en cuenta, ya que en la industria de defectos en fibra de densidad media con papel melamina (MDFMP por sus siglas en inglés) en sus líneas de producción los volúmenes son grandes entonces, cualquier pequeña mejora en la calidad de los ahorros son considerables, por lo que un algoritmo con sistema de visión artificial es una solución adecuada para este tópico.

En este trabajo se presentará una aplicación gráfica ocupando Widgets de QT combinada con el lenguaje C++ y las librerías de OpenCV.

1.3. Antecedentes

Desde la antigüedad, el ser humano sabe que existen objetos que, al tacto tiene una consistencia y una sensación muy diferente, puede ser muy suave, dura, áspera, etc. a esto con el paso del tiempo se denominó textura. De acuerdo con la RAE, textura deriva del latín *texture*, *textus* (tejido) -ura (actividad o resultado), entonces esto significa tejido, entrelazado, de igual forma proviene de la raíz indoeuropea *teks- con el mismo significado el de fabricar, ensamblar,

carpintería. Entonces, se define textura como algo entrelazado ya sea de manera natural (fibras de cortezas, hojas etc.) o de forma artificial con ayuda del ser humano (bordados, ropa, etc.) tal y como se muestra en la figura 1.1.



Figura 1.1: Ejemplo de una textura

Así como el desarrollo tecnológico que dio inicio a principios del siglo XX tanto como la invención de la informática, ha permitido el desarrollo de métodos para el análisis de imágenes de texturas. Se propusieron métodos iniciales, como el de Ojala et al. [3] llamado Patrones Binarios Locales (en inglés LBP) logrando resultados que esperaban de detección y clasificación de texturas. El operador original LBP, forma etiquetas para los pixeles de la imagen mediante el umbral del vecindario 3×3 de cada pixel con el valor central y considerando el resultado como un número binario. Ojala et al. [4] amplió el operador de LBP para utilizar vecindarios de diferentes tamaños. El uso de una vecindad circular y valores de interpolación bilineales en coordenadas de pixeles no enteros permiten cualquier radio y número de pixeles en la vecindad. Otros métodos que buscan realizar un reconocimiento eficiente de la textura son los filtros de Gaborn que sacan características específicas de la textura para poder realizar una clasificación adecuada [5], también otro método ocupado para la extracción de características

de la textura es por proyección aleatoria la cual consiste en proyectar una serie de puntos de un espacio de alta dimension a un subespacio aleatorio de baja dimensión misma que ha sido ocupada no solo en reconocimiento de texturas si no en reconocimiento facial y “Machine Learning” [6]. Así existen muchos otros métodos que buscan extraer información de la textura que sea de utilidad para la aplicación deseada.

De esta manera, el ser humano con la ayuda de los sistemas digitales logra realizar un análisis de las texturas para clasificar, estudiar o realizar una acción. Antes de continuar con el análisis de texturas es muy importante dejar claro a que se refiere el procesamiento de imágenes pues es parte del presente trabajo.

1.4. Procesamiento digital de imágenes

Desde los inicios de la era electrónica, el ser humano ha utilizado la computadora para realizar diversos procesos. Conforme fueron avanzando los descubrimientos surgió una nueva línea de investigación que es procesamiento de imágenes. El procesamiento digital de imágenes (PDI) es de gran impacto en diversos ámbitos ya sea académico, viendo diversos fenómenos, comprobando una teoría propuesta o en el industrial, en el cual bajo ciertas normas, se puede realizar una clasificación y pruebas no invasiva para determinar problemas en una línea de producción.

Un ejemplo de este método, es en el sector alimentario, tal y como lo expresa Jackman [7] en el cual se puede uno ahorrar tiempo y capital para poder realizar una inspección en la línea de producción (en este caso alimentaria) es muy importante darle la debida atención en problemas en la línea para que los alimentos no se encuentren contaminados o en mal estado, o también como Lavelin et al. [8] explica que por medio de procesamiento y segmentación de imágenes puede realizar un análisis completo de lo que se está revisando

en un microscopio, estos no son los únicos ejemplos que se tiene y se pueden enumerar otros más para poder comprender mejor la importancia del mencionado proceso. Si bien, los procesos con los que se cuentan son muy poderosos, el usuario promedio no logra sacarle todo el potencial para realizar un procesado por qué no comprenden de manera consciente los principios básicos detrás de las rutinas de procesamiento de imágenes más simples [9]. El PDI es un campo maduro, pero para diversas aplicaciones se debe abordar ciertas singularidades que el conocimiento general no nos permitiría poder realizar dicho proceso. La terminología de PDI se deriva de los procesos matemáticos que se realizan para el mejoramiento de la imagen (ya sean filtros, correcciones, análisis estadístico etc.), no se tiene estándares que obedezcan a una regla y la terminología se puede ir determinando a las necesidades [10].

En general, el PDI es un conjunto de métodos en el cual se le realizan operaciones determinadas a una imagen con el fin de extraer información que sea de utilidad para alguna tarea o bien, mejorarla, también se puede decir que es un proceso en el cual entra una imagen y sale una imagen mejorada bajo ciertas operaciones o características asociadas a la misma o un vector que contiene toda la información necesaria para el estudio en específico de la aplicación. En la actualidad, el PDI se encuentra entre las tecnologías de rápido crecimiento, al mismo tiempo forma un área de investigación central dentro de las disciplinas de ingeniería y ciencias de la computación. El PDI incluye básicamente los siguientes tres pasos (ver figura 1.2):

- Importación de la imagen a través de herramientas de adquisición de imágenes.
- Analizando y manipulando la imagen.
- El resultado puede ser imagen o informe alterado que se basa en el análisis de la imagen.

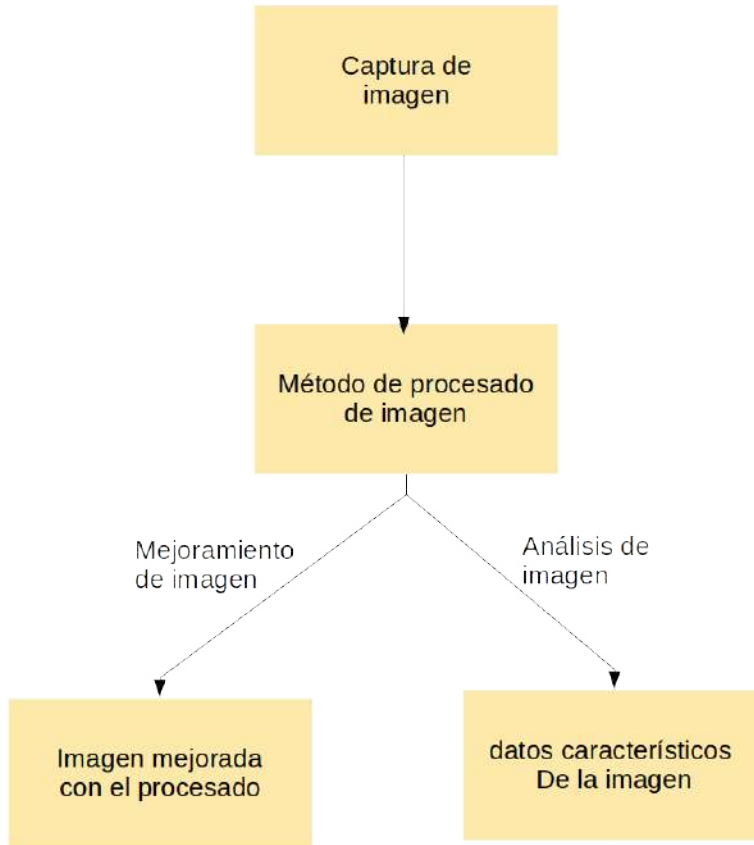


Figura 1.2: Principio de procesamiento de imágenes

Estos pasos están presente en dispositivos digitales de captura y almacenamiento de imágenes (sean cámaras digitales, smartphone, etc.) contienen un programa de procesamiento de imágenes el cual está controlado computacionalmente. El computador ingresa a la base de datos de imágenes y se hace un procesamiento y compresión a la imagen para al final guardarla en la memoria ya sea interna o externa del dispositivo [11]. Durante el PDI solo se realiza una operación a la vez,

por lo que todo el proceso de manipulación es resultado de una linealización, corrección y compresión que ocurre de acuerdo al diseño del hardware.

1.5. Textura

Las texturas juegan un papel importante en visión por computadora. En algunas imágenes, la textura puede ser la característica de definición de las regiones y en la obtención de un correcto análisis. Se puede entender lo que es una textura en el campo computacional de manera empírica, pero es muy difícil poder determinar la mejor definición de lo que es una textura, misma que se ha recopilado de los diferentes investigadores de visión por computadora, las cuales nos permiten entender mejor aun lo que es una textura.

- Es una región macroscópica estructurada de la imagen que presenta propiedades locales constantes, lentamente variables o aproximadamente periódicas [12].

- Una región de la imagen que tiene una textura constante, si una parte estática local u otra propiedad local de la función de la imagen es constante, variación lenta o aproximación periódica [13].

- La textura de la imagen considerada no es figurativa o celular, la textura es descrita por el tipo y el número de primitivas (tonalidades) que puede tener, una de las características de la textura no puede ser analizada sin un marco de referencia del objeto primitivo tonal es declarado o implícito. Si queremos analizar una textura en escala de grises, existe una escala tal que cuando se examina la superficie, no tiene textura. Entonces podemos determinar, que a medida que le aumentamos la resolución la textura se vuelve más fina y luego más gruesa [14].

- La textura se define para nuestro propósito como una característica de campo que no tiene componentes que parezcan enumerables. Las relaciones de

fase entre los componentes son por lo tanto no aparentes. Tampoco el campo debe contener un gradiente obvio. La intención de dicha definición es para dirigir la atención del observador a las propiedades globales mostradas, es decir, su "tosquedad", "irregularidad" o "finura" general. Físicamente, los patrones no enumerarles (aperiódicos) son generados por procesos estocásticos en oposición a procesos deterministas. Perceptiblemente, el conjunto de todos los patrones sin componentes enumerarles obvios incluirá muchos deterministas (e incluso periódicos) texturas" [15].

- La textura es una noción aparentemente paradójica. Por un lado, es comúnmente utilizado en el procesamiento temprano de información visual, especialmente para propósitos de clasificación práctica. Por otro, nadie ha logrado producir una definición de textura comúnmente aceptada. La resolución de esta paradoja, depende de un modelo más rico y más desarrollado para el procesamiento temprano de la información visual, un aspecto central del cual serán los sistemas de representación en muchos aspectos diferentes. Niveles de abstracción. Estos niveles probablemente incluirán intensidades reales en la parte inferior y avanzará a través de los descriptores de borde y orientación a la superficie, y quizás descriptores volumétricos. Dadas estas estructuras multinivel, parece claro que deben incluirse en la definición y cálculo de descriptores de textura [16].

- La noción de textura parece depender de tres ingredientes fundamentales: algunos "órdenes" locales se repite en una región que es grande en comparación con el tamaño del pedido, el orden consiste en la disposición no aleatoria de partes elementales, y las partes son entidades aproximadamente uniformes que tienen aproximadamente las mismas dimensiones. En todas partes dentro de la región texturizada " [17].

- La textura es una imagen de características que corresponden a ambos

valores de brillo y coordenadas del pixel [18].

Con lo anterior se concluye que, la definición de textura ha sido propuesta por un conjunto de personas que, dependiendo de la aplicación que realicen es un concepto valido, pero no necesariamente aplicado a todos los casos. En algunos casos han sido motivadas perceptualmente, mientras que en otros dependiendo de la aplicación es el concepto que se ocupará. Dentro de la visión artificial, el análisis de texturas es un proceso muy recurrente para clasificación, sin embargo, las imágenes no siempre muestran con exactitud la intensidad correspondiente a sus niveles de gris, como ejemplo, se puede tener la corteza de un árbol que, aunque tiene una textura muy regular no logra mostrar la realidad de su intensidad luminosa [17] (ver figura 1.3).



Figura 1.3: Corteza de árbol en la que se observa porque su rugosidad no permite un mejor análisis.

La textura tiene una cierta resolución y un tamaño determinado previamente y que esto es lo que permite realizar el estudio más adecuadamente.

1.6. Propiedades de la textura

En secciones 1.5 se ha planteado que las texturas tienen cierta resolución y tamaño adecuado pero dentro de esas mismas condiciones guardan diversas características, una de ellas tiene que ver con su tipo, es decir, si es visual o

táctil. Las texturas táctiles se refieren a la sensación tangible inmediata de una superficie. Las texturas visuales se refieren a la impresión visual que producen para el observador humano, que está relacionada con las variaciones espaciales locales de estímulos simples como el color, la orientación y la intensidad de una imagen. Este trabajo se centra solo en las texturas visuales, por lo que el término "textura" a partir de entonces se refiere exclusivamente a "textura visual", a menos que se mencione lo contrario. Cabe mencionar, que las texturas analizadas y trabajadas en el presente trabajo solo serán las de 2D por lo que todas las propiedades y definiciones serán acopladas a las texturas 2D visuales. Se ha mencionado que no se tiene una definición exacta, pero lo que sí se cuenta son algunas propiedades que las pueden caracterizar y podamos entender aún mejor como es su comportamiento [19] las cuales son mencionadas a continuación:

- La textura envuelve la distribución espacial de los niveles de gris.
- Las texturas visuales pueden tener percepciones diferentes por sus niveles de resolución y de escala.
- Una región puede ser percibida como textura dependiendo del número de secciones primitivas.

Xiaoying Guo [20] define 4 características primordiales para poder entenderlas estas son : regularidad, aspereza, direccionalidad y densidad.

La regularidad se define como la variación en toda la textura esto tiene que ver como esta puede tener un patrón periódico que la va constituyendo, tal y como se puede observar en la figura 1.4 (a y b):

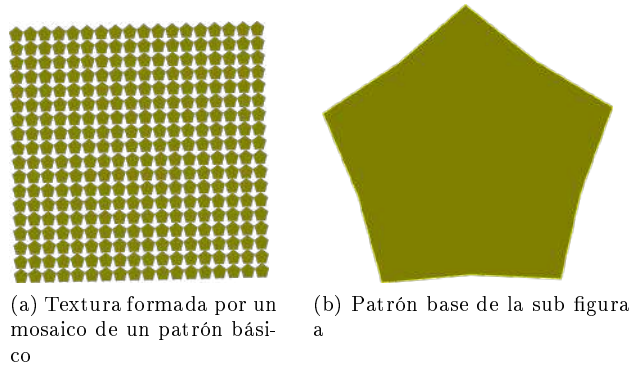


Figura 1.4: Regularidad de una textura

En estas figuras se puede observar como el patrón (también definido como textón) ayuda en la construcción de la textura permitiendo que de manera periódica aparezca en todas las filas y columnas de la imagen y permita tener esa característica.

La aspereza está definida como la variación en un área pequeña, esto nos quiere dar a entender como una textura puede tener cambios de intensidad en un pequeño espacio, aunque se mantiene periódico, dentro de la periodicidad podemos encontrar esas asperezas, tal y como se puede observar en la figura 1.5:



Figura 1.5: Ejemplo de la aspereza en la piedra de granito.

Esta imagen tomada de una pared de cemento nos muestra cómo se encuentra la textura, podemos observar esos huecos, crestas, deformaciones que fue tomando el cemento de acuerdo a como la persona que ocupó el cemento lo dejó y permitió tener una aspereza.

La direccionalidad está definida por la orientación de los bordes, porque los bordes de la imagen tiene una influencia significativa en la percepción visual, la definición nos muestra que por los bordes que tiene la imagen se puede apreciar en qué dirección se aprecia la textura tal y como se puede ver ejemplificado en la figura 1.6:



Figura 1.6: Se puede apreciar que gracias a los bordes nos permite determinar hacia donde se tomó la foto para determinar la dirección que tiene la textura

Como se puede observar, de acuerdo a la dirección que se le tomó, el ojo humano visualiza el cambio de dirección en las líneas de la textura, como bien dice la definición los bordes son los que ayudan a determinar los cambios de dirección.

Del mismo modo, la densidad es definida de la misma manera que la direccionalidad, es medido utilizando la densidad del borde.

1.7. Análisis de texturas

Las características que se revisaron permiten dar un preámbulo de cómo se

entiende y clasifica las texturas por medio de métodos visuales, en todos los ámbitos que tengan que ver con la detección y clasificación de texturas se busca llegar a entender, modelar y procesar una textura y en última instancia se busca recrear artificialmente lo que el ser humano realiza por medio de la visión con la ayuda de sistemas computacionales y/u otros medio tecnológicos.

Tal y como explica Zhou en su tesis doctoral [21] el análisis de textura busca obtener una descripción cuantitativa general, eficiente y compacta de las texturas, de modo que se puedan utilizar varias operaciones matemáticas para alterar, comparar y transformar texturas. La mayoría de los algoritmos disponibles de análisis de textura implican extraer características de textura y derivar un esquema de codificación de imagen para presentar características seleccionadas. Estos algoritmos pueden diferir en las características de textura que se extraen o en cómo se presentan en la descripción.

Tuceryan y Jain [19] mencionan que existen cuatro tipos de categorías con sus respectivos métodos para realizar un análisis detallado de texturas: geométricos, basados en modelos, procesamiento de señales y estadísticos. La primera categoría se centra en los descriptores estructurales mientras que los otros tres, son considerados como enfoques estadístico-teóricos.

1.7.1. Métodos geométricos

Estos métodos se enfocan en el análisis geométricos de la textura, esto es como están compuestos sus texturas bases o sus textones determinísticos. Para entender mejor estos métodos vamos a ir definiendo lo que es un texton. Un textón se define como un vecindario que tiene pixeles con propiedades invariantes que están arreglados sobre la imagen dada. Estos mismos pueden ser definidos por sus niveles de intensidad, forma u homogeneidad de algunas de sus propiedades locales como tamaño orientación entre otras más. Una vez que los elementos de textura se identifican en la imagen, hay dos principales enfoques

para analizar la textura. Uno calcula las propiedades estadísticas de la extracción de elementos de textura y utiliza estos como características de textura. El otro extrae la regla de colocación que describe la textura. Este último enfoque puede implicar geometría o métodos sintácticos de analizar la textura. Algunos métodos geométricos de extracción de primitivas y/o determinación estructural incluyen:

- Aproximaciones de detección de bordes.
- Extracción de regiones adaptativas.
- Morfología matemática.
- Características de la teselación Voronoi.
- Por métodos estructurales.

Este último método es el que se basa mayormente de analizar el textón, estos mismos se enfocan en dos puntos fundamentales para poder realizar dicho análisis:

1. Extracción de los elementos de textura.
2. Inferencia de la regla de colocación.

Dentro de estas categorías se encuentran el modelo de Zucker [16] y el modelo de Fu [22], este método en particular, tiene dentro de sus características el poder generar y analizar texturas. Otro método más reciente es el que proponen Xiaoming Liu [23], en el cual por medio de radiografías a las mamas se puede realizar una detección de cáncer de mama adecuada.

1.7.2. Métodos basados en modelos

El método basado en modelos se enfoca en la construcción de una imagen modelo que no sólo realiza el análisis de texturas, sino que lo hace sin sintetizarlo. Este método se enfoca en sacar las características cuantitativas de la imagen. A continuación se ejemplifican algunos de los modelos existentes:

1. Por Campos aleatorios (Como los campos aleatorios de Makarov (RFM))

y el campo aleatorio de Gibbs (RFG) [24].

2. Análisis de un solo pixel (Como el método Derin-Elliot y el método auto-binomial).

3. Fractales (Como imágenes de campo aleatorio Gauss Markov, Imágenes fractales, imágenes de cuero y superficies pintadas).

Este último método se puede ver reflejado en la aplicación que le dio Backes et al. [25] para poder realizar una clasificación adecuada de texturas por su color ya que como bien lo menciona en su artículo “Color texture analysis based on fractal descriptors” los fractales ofrecen una alternativa interesante a estos enfoques. Debido a sus irregularidades, la mayoría de las superficies naturales tienen dimensiones no enteras. Por lo tanto, parece plausible el uso de fractales.

O también con Freeman et al. [26] con sus campos aleatorios de Makarov en el cual explican que debido a otros métodos que son muy lentos su método busca realizar un proceso computacional mucho más rápido para poder obtener las características que se buscan en las texturas.

1.7.3. Métodos basados en señales

Se sabe que todo está bajo una frecuencia, y en el análisis de texturas no es la excepción, las texturas analizadas por procesamiento de señales se realiza por las propiedades que se pueden obtener. Algunas de estas técnicas de análisis de texturas se centran en el filtrado de las texturas para poder así realizar de manera adecuada dicho proceso:

1. Filtros en el dominio del espacio (Como el operador Rober o el Laplaciano).
2. Filtros en el dominio de Fourier.
3. Modelos de Gabor y de ondas [27].

1.7.4. Método estadístico

Los métodos estadísticos buscan realizar una descripción adecuada de las texturas para poder realizar un reconocimiento estadístico de las mismas efecti-

vamente [28]. Los métodos estadísticos caracterizan la textura por la distribución estadística de la intensidad de una imagen, es una de las cualidades definitorias de la textura.

La distribución espacial de los niveles de gris ocupa un vector característico el cual contiene todas las propiedades estadísticas de la textura [29]. El objetivo es encontrar un método probabilista o determinista. Estos son los más ocupados en el análisis de texturas por su uso desde los comienzos de investigación de análisis de texturas. Algunos de los métodos de este estilo son enumerados a continuación:

1. Matrices de co-ocurrencia.
2. Características de auto correlación.
3. Métodos estadísticos de primer orden.
4. Matrices de longitud de ejecución.

Existen muchos ejemplos del uso de estos métodos como el que expone De Siqueira [30] en el cual ocupa las matrices de co-ocurrencia para poder determinar los descriptores de las texturas en imágenes de escala de grises con una mejora ocupando dos diferentes método: representaciones Gaussianas espacio-escalado e imágenes piramidales.

1.7.5. Clasificación de texturas

La clasificación de texturas es un método en el cual se realiza una toma de decisiones en base sus características para determinar a qué categoría de texturas pertenece. Un dato fundamental para poder realizar la clasificación adecuadamente, es necesario conocer previamente las clases. Una vez que este conocimiento está disponible y se extraen las características de la textura, se utiliza técnicas clásicas de clasificación de patrones para hacer la clasificación.

Tal y como lo explica Sánchez Yáñez [27] en su tesis doctoral en el capítulo 1, existen cuatro diferentes problemas en el campo del análisis de texturas:

1. La clasificación de textura, tiene como objetivo el determinar una etiqueta para una región de textura uniforme de acuerdo con la clase predefinida. Es decir, la clasificación de textura consiste en decidir a qué categoría pertenece la textura visual.

2. La segmentación de texturas es la partición de imágenes en regiones que tienen propiedades homogéneas, con respecto a la textura. No es necesario saber el número de tipos de texturas que se presentan en la imagen para realizar la segmentación de la textura.

3. La síntesis de la textura, que tiene como objetivos construir un modelo de textura, con el cual se puede usar para generar aplicaciones de texturas como gráficos computacionales.

4. Las superficies de texturas, donde las imágenes 2D son consideradas como proyecciones de una escena 3D.

Como se puede observar, la clasificación de texturas tiene que ver con técnicas desde sus inicios en lo que se enfocaban era en la estadística de las texturas [31], dentro de estos mismos métodos de clasificación tiene que ver con que dada n entradas solo tendremos una salida. El proceso de clasificación de texturas consta de 2 fases importantes: el aprendizaje y el reconocimiento. En la primera fase el objetivo es construir un modelo para el contenido de textura de cada clase de textura presente en los datos de entrenamiento, que generalmente comprende imágenes con etiquetas de clase conocidas. El contenido de textura de las imágenes de entrenamiento se captura con el método de análisis de textura elegido, que proporciona un conjunto de características de textura para cada imagen. Estas características, que pueden ser números escalares o histogramas discretos o distribuciones empíricas, caracterizan las propiedades de texturas dadas de las imágenes, como la estructura espacial, el contraste, la rugosidad, la orientación, etc. En el reconocimiento, se describe el contenido de textura de

la muestra desconocida. Con el mismo método de análisis de textura.

Entonces, las características de textura de la muestra se comparan con las características de las imágenes de entrenamiento con un algoritmo de clasificación, y la muestra se asigna a la categoría con la mejor coincidencia. Opcionalmente, si la mejor coincidencia no es lo suficientemente buena de acuerdo con algunos criterios predefinidos, la muestra desconocida puede ser rechazada en su lugar. Los métodos más usados para la clasificación son:

1. Clasificadores paramétricos estadísticos, derivados de la teoría de decisiones Bayesiana.
2. Clasificador no-paramétricos con el k -ésimo vecino.
3. Redes neuronales (perceptrones multicapa) [32].

1.8. Análisis del trabajo

Al realizar un meticuloso análisis de los trabajos de investigación previos en el ámbito de la clasificación de texturas podemos inferir que lo que se intenta alcanzar es:

- a) Lograr realizar una discriminación adecuada de los patrones de acuerdo a sus características.
- b) Obtener un tiempo de respuesta adecuado dado que una base de datos grande puede suponer un tiempo máquina tardado.

El método de la Transformada de Cúmulos Coordinados es un método que basado en una ventana de tamaño $I \times J$, recorre toda la imagen binarizada pixel a pixel, dentro de la ventana se decodifica a número de binario a decimal, esto con el fin de poder obtener un histograma de frecuencias normalizado, el cual se convertirá en nuestro vector patrón para futuros procesamientos. Con esto, podemos determinar que el objetivo primordial es realizar un análisis eficiente del método y determinar qué tan viable es para realizar clasificación de texturas

tanto naturales como artificiales.

El presente trabajo está dividido en 5 capítulos que fueron organizados como a continuación: El capítulo 2 se centrara en realizar una presentación y análisis de la Transformada de Cúmulos Coordinados, de sus propiedades y la presentación del código junto con una breve explicación de cómo es el procedimiento para la binarización de las texturas. En el capítulo 3 se realiza una revisión de lo que es la invarianza en patrones, en texturas y describir como se realiza la mejora del método al buscar una invarianza a rotación por medio de la Transforma Discreta de Fourier. En el capítulo 4 hace una revisión de todos los experimentos realizados para obtener la eficiencia que tiene el método, comparar con otros métodos parecidos, como es su comportamientos y dar explicaciones a los resultados obtenidos y para finalizar en el capítulo 5 se presenta una serie de conclusiones que se obtuvieron durante todo este trabajo tanto generales como específicas para cada paso y dando un punto de vista para poder dar un panorama más amplio y dejar abierto a futuros trabajos.

1.9. Acerca del software

La interfaz y todo el código para realizar la clasificación de texturas y reconocimiento basado en la Transformada de Cúmulos Coordinados fue desarrollada en C++ ocupando un software libre llamado QT Creator y desarrollada en su totalidad en el ambiente Linux (Ubuntu 18.04) para el llamado y manipulación de las imágenes se ocupó el software OpenCV, este mismo software es multiplataforma (Linux, Mac y Windows) por lo que no es difícil que se pueda ejecutar en cualquier otro sistema operativo que tenga QT Creator instalado junto con las librerías de OpenCV. El software fue diseñado para que pudiera realizar la clasificación de texturas ocupando el método de la Transformada de Cúmulos Coordinados y en la sección de invarianza a rotación se complementó con

el espectro de Fourier. Los experimentos fueron realizados y documentados en diversos reportes. Los puntos clave del software se enlistan a continuación:

- La base de datos consistió en 3 tipos diferentes, para texturas computacionales, texturas del álbum de Brodatz y para texturas reales, las cuales fueron el punto fundamental para la clasificación.
- Las funciones de procesamiento de imágenes y de señales fueron desarrolladas (como la de binarización, el método de Otsu, el espectro de Fourier, La Transformada de Cúmulos Coordinados).
- Otros métodos desarrollados, como el cargar la imagen, manipularla y creación de gráficas fue gracias a las funciones declaradas por OpenCV y QcustomPlot respectivamente.

Capítulo 2

Transformada de Cúmulos Coordinados

Los métodos ocupados para la clasificación de texturas se propusieron, pero uno de los métodos más recientes es en el que se basa este trabajo: la Transformada de Cúmulos Coordinados. La Transformada de Cúmulos Coordinados o la Representación de Cúmulos Coordinados (CCR por sus siglas en inglés) es un método no invasivo de clasificación de texturas propuesto por Kurmyshev-Cervantes [33] y Kurmyshev-Soto [34], el cual fue utilizado para que una imagen binarizada fuera caracterizada como un histograma de ocurrencia, para convertirse en descriptor de textura y así realizar la clasificación. Para poder realizar esto se determina una ventana que recorrerá secuencialmente toda la imagen pixel a pixel, dentro de la ventana se obtendrá un número en binario, ya que como la imagen es binaria obtendremos en cada pixel un valor de 0 o 1 (negro o blanco), posteriormente se realiza una decodificación a decimal a partir del número en binario del bit menos significativo (posición derecha) al bit más significativo (posición izquierda) de acuerdo a la posición que se encuentren los bits

que valgan 1 se elevará 2 la posición que se encuentre el bit 1 y se sumará con todos los demás bits iguales a 1 esto nos dará el valor en decimal. Este valor en decimal será el índice del histograma que se construirá de acuerdo al valor decimal que la ventana detecta. Desde este punto en adelante cuando se mencione CCR se estará hablando de la Transformada de Cúmulos Coordinados.

2.1. Transformada de Cúmulos Coordinados

Para abordar de mejor manera algunas de las notaciones matemáticas representaremos la CCR de imágenes binarias en la forma algorítmica. Dada una imagen binarizada de tamaño $N \times M$, que viene representada de la siguiente manera $S^\alpha = \{s^\alpha(n, m)\}$ se caracteriza mediante el uso de una ventana rectangular de tamaño $I \times J$ tal que $I \leq N$ y $J \leq M$. Una vez obtenido dicho tamaño de la ventana, se recorre con esta toda la imagen S^α moviendo la ventana pixel a pixel y con la decodificación de binario a decimal que se realiza dentro de esta, obteniendo un valor en decimal que incrementa unitariamente el valor decodificado del histograma en resumen, en la siguiente figura podemos ver un ejemplo del comportamiento de la ventana en la imagen binarizada, un ejemplo de cómo funciona la ventana de escaneo se puede apreciar en la figura 2.1.

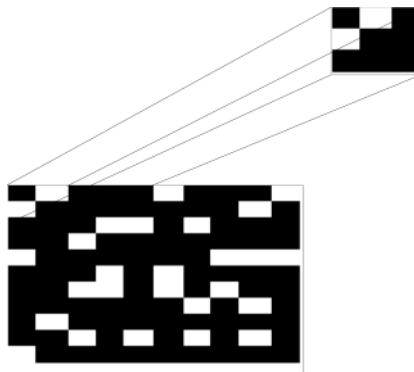


Figura 2.1: Representación de la CCR con su ventana de escaneo

El número de estados que puede tener el histograma depende, del tamaño de la ventana, en otras palabras $w = IxJ$ esto va a determinar el tamaño que puede tener el histograma de frecuencias, mismo que se puede reducir más si se toman solo los valores diferentes a cero.

Entonces, la CCR asigna a cada imagen binaria S^α un histograma de frecuencias $H_{(I,J)}^\alpha(b)$ el cual representa la frecuencia del patrón que es escaneado por la ventana, α representa el índice del conjunto de imágenes que se integren a la transformada, (I, J) representan el tamaño de la ventana de escaneo y por último la variable $b = 2^w$ que representa la codificación del número en binario de acuerdo a la ventana de escaneo.

Con respecto a lo anterior se puede determinar que el total de las frecuencias del histograma es igual al área total del histograma, esto es,

$$A = (I - N + 1)(J - M + 1). \quad (2.1)$$

Con este valor podemos normalizar el histograma $H_{(I,J)}^\alpha(b)$, el cual es considerado como una distribución en función de las frecuencias:

$$F_{(I,J)}^\alpha(b) = \frac{H_{(I,J)}^\alpha(b)}{A}. \quad (2.2)$$

Un ejemplo de cómo se realiza la transformada CCR es usando una ventana de 3x3 cuyos códigos binarios y decodificación se pueden observar en la fig. 2.2 donde el número en decimal (BCD) corresponde al índice del histograma de frecuencias. La fig. 2.3 muestra la binarización de una textura ejemplo y sus respectivos histogramas.




Patrón	Binario	Decimal
	101100001	353
	010101100	172
	010101101	173

Figura 2.2: Patrones binarios de acuerdo a la ventana de 3x3 su número en binario y su decodificación en decimal para el histograma de frecuencias de la CCR.

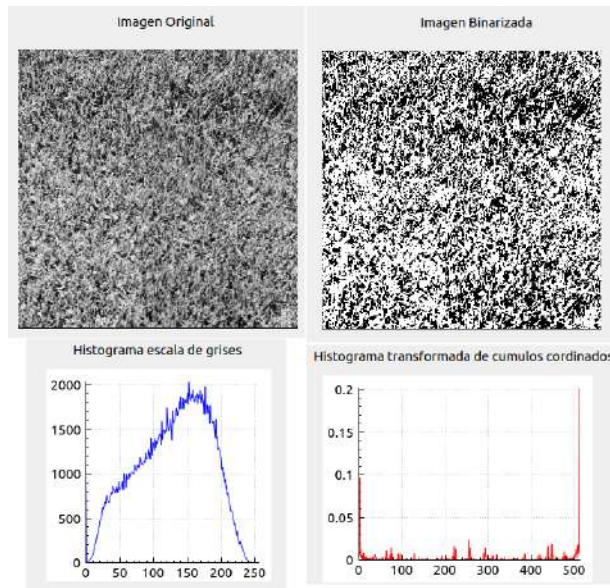


Figura 2.3: Cálculo del histograma de la imagen (D9) de escala de grises junto con la binarización de la imagen y su respectivo histograma de la CCR.

Este método a comparación de otros métodos propuestos para la clasificación de texturas citando algunos como Patrones Binario Locales (LBP) [35, 36, 37, 38, 39, 40] Filtro de Gabor [5] o la proyección aleatoria [6] es que trata de realizar más rápido el cálculo, aunque el LBP es el más similar a la CCR las características que obtiene son diferentes (histograma diferente). Puesto que el

método de LBP ha sido estudiado y revisado con anterioridad Kumyshev nos ha demostrado dicho método obtiene características de una manera diferente al de la CCR [41].

2.2. Propiedades de la CCR

Las propiedades de la CCR fueron establecidas en 2 teoremas fundamentales [42] los cuales son interpretados a continuación. El primer teorema establece sobre la estructura de la CCR específicamente en imágenes periódicas:

Teorema 1. *Si un textón (celda primitiva) de una imagen binaria traslacional S^α tiene un tamaño $\tau_1 \tau_2$ pixeles, τ_1 en uno y τ_2 en otra dirección, entonces cualquier distribución de la CCR $F_{(I,J)}^\alpha(b)$ toma no más que valores diferentes de cero. Si la ventana de escaneo de la CCR tiene el tamaño igual o mayor al tamaño del textón $I \geq \tau_1$, y $J \geq \tau_2$, entonces toma exactamente valores diferentes de cero.*

En el siguiente teorema se enfoca en la relación del histograma con el k -ésimo orden estático de una imagen binarizada. Esto es, que el histograma de frecuencias no normalizado $H_{(I,J)}^\alpha(b)$ contiene toda la información acerca de momentos de correlación de n puntos de la imagen S^α si y solo si los vectores de separación entre n pixeles caben en la ventana de escaneo, eso es una distribución función $F_{(I,J)}^\alpha(b)$ la cual ofrece suficiente información de la función de probabilidad de enésimo-punto.

Teorema 2. *Sea $S^\alpha = \{s^\alpha(n, m)\}$ una matriz imagen binaria y $F_{(I,J)}^\alpha(b)$ sea la función de distribución de la CCR. Si $\max(l_i) \leq I$ y $\max(m_i) \leq J$ ($i = 1, 2, \dots, k - 1$) y $k < IJ$, entonces ninguna función de auto correlación del k -ésimo orden*

$$\langle s^\alpha(l, m) s^\alpha(l + l_1, m + m_1) \dots s^\alpha(l + l_k, m + m_{k-1}) \rangle \equiv \lim_{L, M \rightarrow \infty} \sum_{l, m=1}^{L' \cdot M'} s^\alpha(l, m) s^\alpha(l + 1, m + m_1) \dots s^\alpha(l + l_k, m + m_{k-1}) \quad (2.3)$$

puede ser reconstruida individualmente de $F_{(I, J)}^\alpha(b)$ donde $N = LM$ es el tamaño de la imagen, $L' = L - \text{máx}\{l_i\}$, $M' = M - \text{máx}\{m_i\}$.

La clasificación de patrones con la CCR conserva toda su información estructural para poder realizar un análisis más adecuado, por otro lado, la clasificación con la CCR puede ser orientado a bits, ya que lo leído en la ventana es un número en binario (0 o 1), lo que permitirá que el cálculo sea más rápido. Cabe destacar que este método nos permite realizar una clasificación de texturas eficaz y de lo más adecuado, ya que la mayoría de los métodos estudiados se enfocan en realizar la clasificación de texturas por medio de imágenes en escala de grises, este método se basa en imágenes binarias ya que una imagen binaria guarda mejor su información estadística.

Sin embargo, uno de los problemas que se enfrenta en el escaneo de la imagen con la CCR es el tamaño de la ventana. Si es que se elige una ventana bastante grande, el histograma de la CCR será grande por lo que el computador tendrá problemas de administración de la información adecuadamente.

Una solución práctica para conectar los dos requisitos fundamentales (preservar información y capacidad de administración de la CCR) es reducir la escala de las imágenes que se van clasificando.

Este método se puede ocupar en la clasificación de texturas naturales, artificiales y computacionales, nos puede servir en determinar que tela estamos comprando esto se puede realizar con una app de celular que sea una herramienta auxiliar para el usuario esto se realiza mediante el enfoque la cámara en la tela y con una base de datos previamente cargada compara el vector ca-

racterístico obtenido con los de la base de datos y así determinar que tela se está observando, se puede ocupar en el reconocimiento de corteza de árbol en un ecosistema estudiado, el mismo método se aplica que el de las telas pero con una base de datos diferente y se puede ocupar en la detección de fallas de mangueras industriales.

En la figura 2.4 se puede mostrar un claro ejemplo de la binarización que se tiene de dos imágenes tomadas de la base de datos Brodatz [43] ambas imágenes fueron calculadas por una ventana de tamaño 3×3 , en los histogramas que se calcularon se puede observar una conexión entre la periodicidad de la textura y su representación de la CCR. La textura D12 esto está relacionado con la frecuencia de la señal (menos picos) que la D68. En consecuencia, la función de distribución D12 muestra menos picos. Por otro lado, el histograma de la CCR de la imagen D68 nos muestra que varios números son muy frecuentes encontrarlos en la imagen por lo que se encuentran más picos que son vistos como “ruido” en el histograma de la CCR. Es evidente que muchos de los valores del histograma son cero por lo que eso se podría modificar buscando discriminar los valores que sean cero y dejando todo lo que si contribuye al patrón de reconocimiento. En conclusión, entre más periódica sea la textura, menos picos se encontrarán y la representación de la CCR será mejor.

Es relevante mencionar los extremos que se tienen en el histograma de la CCR, como se puede observar en la figura 2.4 los extremos casi no son perceptibles debido a que como almacenan mucha de la información de la imagen binarizada la frecuencia en la que se encuentran estos valores es muy frecuente, esto debido a que como es una imagen binarizada (0 y 1) los extremos serán más sencillos encontrar (000000000 y 111111111 en número binario respectivamente).

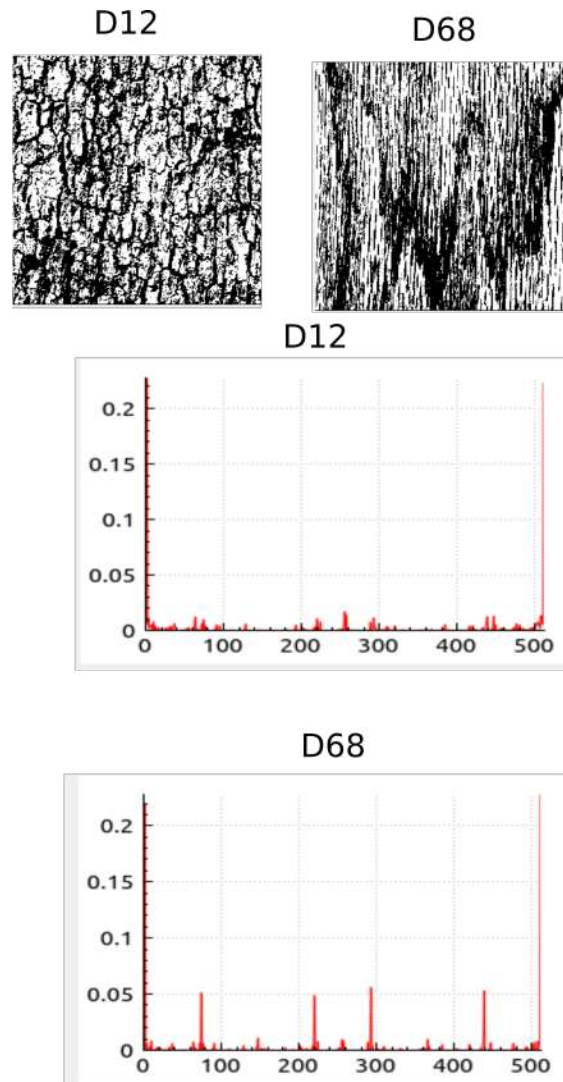


Figura 2.4: Imágenes binarizadas junto con sus respectivos histogramas de frecuencias de la CCR

2.3. Algoritmo de la CCR

Una vez entendido todo lo conceptual procedemos a explicar el algoritmo. Para poder realizar el procesamiento de la imagen binarizada debemos tomar en cuenta que en entrada tendremos una imagen y de salida tendremos un

vector cuya longitud dependerá del tamaño de la ventana entonces, pasando esto a manera computacional tenemos que la variable *Mat* será nuestra entrada mientras que un arreglo del tipo *double* será nuestra salida, para que el usuario pueda determinar el tamaño de la ventana pondremos en la función dos variables en formato *int*, ya que la ventana tiene que ser de valores enteros por que va de la mano con los pixeles que contiene la imagen, antes de presentar el código para poder dar un preámbulo de lo que se va a realizar se presenta un pseudocódigo que hará más entendible lo que se programó:

Algoritmo 2.1 Pseudocódigo de la CCR

Entrada: Función CCR

$$A = (totalfilasImagen - totalfilasVentana + 1) * (totalcolumnasImagen - totalcolumnasVentana + 1)$$
mientras $k < total\ filas\ Imagen - total\ filas\ Ventana$ y $l < total\ columnas\ Imagen - total\ columnas\ Ventana$ **hacer**
para $i = 0$ hasta $i < total\ filas\ columnas\ ventana$ **hacer**
para $j = 0$ hasta $j < total\ columnas\ ventana$ **hacer**
 $ventana[i][j] = Imagen[k + i][l + j]$

 incrementa i y j
fin para
fin para
para $i = 0$ hasta $i < total\ filas\ columnas\ ventana$ **hacer**
para $j = 0$ hasta $j < total\ columnas\ ventana$ **hacer**
 $bin[n] = ventana[i][j]$

 incrementa n
fin para
fin para
 $dec = decodificar(bin)$
 $histo[dec] + 1$

 incrementa l
si $l == total\ filas\ de\ la\ Imagen - total\ filas\ de\ la\ ventana$ **entonces**
 $l = 0$

 incremento k
fin si
fin mientras
para $i = 0$ hasta $i < longitud\ bin$ **hacer**
 $histo[i] = histo[i]/A$
fin para
devolver cierto

Entonces la función en lenguaje C++ quedará de la siguiente manera:

Algoritmo 2.2 Función de la CCR

```

void CCR(Mat ma, double *histo, int venf, int venc)
{
    int i, j, k=0, l=0, bin[9], n=0, dec, A;

    Mat Ventana=Mat::zeros(venf, venc, CV_8UC1);
    A=(ma.rows-Ventana.rows+1)*(ma.cols-Ventana.cols+1);
    for(i=0; i<pow(2, venf*venc); i++)
    {
        histo[i]=0;
    }
    while(k<ma.rows-venf && l <ma.cols-venc)
    {
        for(i=0; i<venf; i++)
        {
            for(j=0; j<venc; j++)
            {
                Ventana.at<uchar>(i, j)=ma.at<uchar>(k+i, l+j);
            }
        }
        for(i=0; i<venf; i++)
        {
            for(j=0; j<venc; j++)
            {
                bin[n]=Ventana.at<uchar>(i, j)/255;
                n++;
            }
        }
        n=0;
        dec=DecodificaBinario(9, bin);
        histo[dec]+=1;
        l++;
        if(l==ma.rows-venf)
        {
            l=0;
            k++;
        }
    }
    for(i=0; i<pow(2, venf*venc); i++)
        histo[i]=histo[i]/A;
    return;
}

```

Como se puede observar se debe realizar un ciclo anidado en el cual mientras el contador sea menor al tamaño de las filas y columnas de la imagen menos el tamaño de las filas y columnas de la ventana continuará realizando el procesamiento en la imagen, esta condición se debe mantener ya que si solo se deja que sea menor al número de filas y columnas de la imagen entonces la ventana estaría buscando pixeles inexistentes, puesto que la ventana continuaría moviéndose y eso nos afectaría en el vector característico. Para que nuestro contador no se salga de rango y proceda a seguir moviéndose por toda la imagen se vuelve a incluir la condición en las filas para que una vez que la ventana haya terminado de realizar el recorrido en la última fila menos tres entonces vuelva a reiniciar el contador de las columnas y el de las filas pueda incrementarse esto con el fin de moverse de manera vertical hacia abajo de la imagen y así recorra secuencialmente todas las filas y columnas la ventana. Dentro de esta función tenemos otra subfunción la cual nos va a permitir realizar la decodificación del número en binario, como se vio en la anterior función se tiene guardado en un arreglo de longitud n los valores de los pixeles binarizados, entonces tenemos de entrada un arreglo de tipo `int` y por consiguiente tendremos de salida un valor tipo `int` entonces la subfunción queda expresada de la siguiente manera:

Algoritmo 2.3 Función de la decodificación

```

int DecodificaBinario(int bits , int *p)
{
    int i, valor = 0;
    for (i = 0; i < bits; i++)
        valor += p[i] * pow(2, i);
    return valor;
}

```

Para que no se tenga pierda en la longitud que tiene el arreglo se le agrega la variable de bits que es la longitud y así cuando realice la suma de todos los valores que valen uno nos permita obtener adecuadamente el valor en decimal

que será guardado en el histograma de frecuencias.

2.4. Método de binarización

Como la CCR es un método que trabaja utilizando imágenes binarizadas, el método de binarización viene declarado de la siguiente forma:

$$g(x, y) = \begin{cases} 0 & \text{si } f(x, y) \leq T \\ 1 & \text{si } f(x, y) > T \end{cases}, \quad (2.4)$$

dónde T es el umbral de binarización y $g(x, y)$ es la imagen binarizada [44].

Para determinar el umbral adecuado se consideran la estadística de la imagen para permitir delimitar el objeto a estudiar se logra mediante el método de Otsu.

El método de Otsu es un método estadístico que permite utiliza el histograma de la imagen en escala de grises para poder determinar el umbral que separa los dos modos dominantes, el procedimiento es descrito a continuación:

Dada una imagen con D niveles de intensidad y asumiendo que el umbral buscado es T , las probabilidades acumuladas hasta T y desde T hasta D resultan ser:

$$w_1(t) = \sum_{z=1}^T P(z) \quad y \quad w_2(t) = \sum_{z=T+1}^D P(z). \quad (2.5)$$

Se obtiene las medias y varianzas asociadas,

$$\mu_1(t) = \sum_{z=1}^T zP(z) \quad y \quad \mu_2(t) = \sum_{z=T+1}^D zP(z). \quad (2.6)$$

$$\sigma_1^2(t) = \sum_{z=1}^T (z - \mu_1(t))^2 \frac{P(z)}{w_1(t)} \quad y \quad \sigma_2^2(t) = \sum_{z=T+1}^D (z - \mu_2(t))^2 \frac{P(z)}{w_2(t)}. \quad (2.7)$$

Al final calcula la varianza ponderada.

$$\sigma_w^2(t) = w_1(t)\sigma_1^2(t) + w_2(t)\sigma_2^2(t). \quad (2.8)$$

Con estas ecuaciones permite elegir el mejor umbral T donde la varianza sea la mínima [45], en la figura 2.5 se muestra un claro ejemplo de la binarización ocupando el método de Otsu.

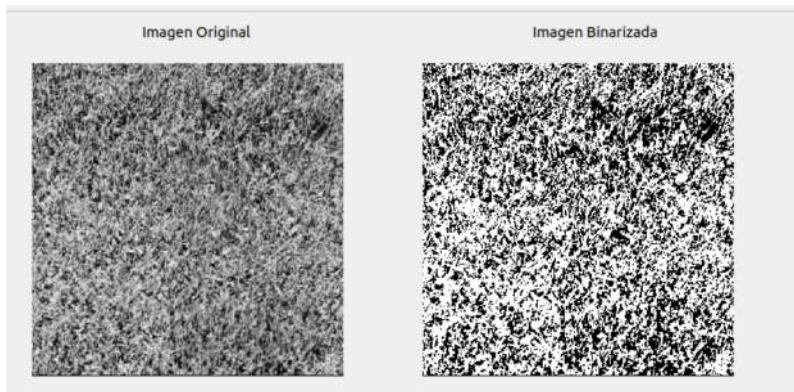


Figura 2.5: Binarización por el método de Otsu

2.5. Método de clasificación

Para entender mejor los métodos de clasificación se definirá lo que es un objeto. Un objeto es una unidad física usualmente representado como una imagen segmentada por una región en análisis de imágenes y de visión computacional. El conjunto de objetos puede ser separada en un conjunto de subsecciones que vistos desde un punto de vista de clasificación tienen ciertas características en común y son llamadas clases.

El número de clases es previamente sabido, ya que este número depende del problema que se quiera abordar. La clasificación, que es similar a la del ser humano, no determina que el objeto sea determinado a una clase en particular,

algunas de las características del objeto sirven para ese propósito.

Un ejemplo de esto es al tratar de clasificar hierro de arena, no necesariamente tenemos que ocupar sus propiedades moleculares para determinar a qué clase pertenecen pero con esta característica se podría realizar adecuadamente. Propiedades como la textura, dureza, sensaciones, etc. son algunas de estas que se ocupan.

Un patrón es aquel objeto que es sentido ocupando algunas de las características antes mencionadas, es necesario dejar en claro que la clasificación no detecta objetos pero si puede detectar patrones así que se puede decir que reconocimiento de objetos y patrones son lo mismo [28].

En la figura 2.6 se muestra la forma en que se realiza la clasificación de objetos, dentro de la teoría de clasificación, una problemática recurrente tiene que ver con el diseño del clasificador para la base de datos específica que describe objetos.



Figura 2.6: Pasos para la clasificación de patrones

Objetos estadísticos ocupan descripciones numéricas las cuales son llamadas características. El patrón (también llamado vector patrón o vector característico) es aquel que describe un objeto que es un vector elemental y el conjunto de todos los conjuntos que forman el espacio patrón (también llamado espacio característico). Si los descriptores elementales fueran apropiadamente elegidos la similitud de los objetos de cada clase resultaría en la proximidad de sus patrones en el espacio patrón [28]. Una ejemplificación de cómo funciona esta discriminación se puede observar en la fig 2.7 donde por medio de una función especial permite delimitar el espacio de cada uno de los patrones adecuadamente.

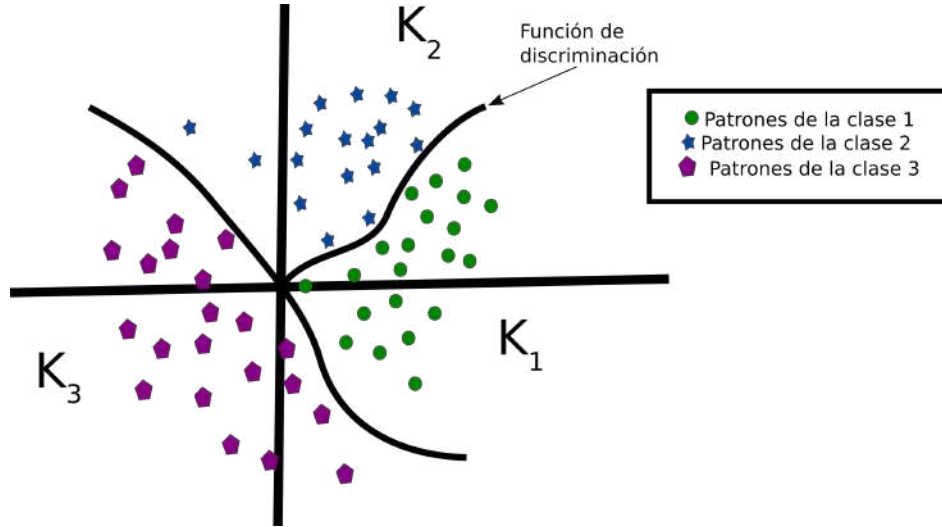


Figura 2.7: Funciones generales de discriminación

2.5.1. Distancia de Hamming

Dentro de los métodos de clasificación tenemos el aprendizaje métrico el cual consisten en mejorar los procesos incorporando información complementaria y optimizando funciones de distancia paramétrica, la distancia de Hamming es un método que es un claro ejemplo de dicha subcategoría de los métodos de clasificación [46, 47], misma que se define de la siguiente manera:

$$d_{(I,J)}(S^\alpha, S^\beta) = \sum_{b=0}^{2^v-1} \left| H_{(I,J)}^\alpha(b) - H_{(I,J)}^\beta(b) \right|. \quad (2.9)$$

Tal que el índice (I, J) indica el tamaño de la ventana de escaneo que es ocupada para la CCR y $N = IxJ$. Dado un patrón de texturas $p = (1, 2, \dots, P)$ la clase correspondiente al patrón S^P es definido como una serie de imágenes S^α tal que $p = (1, 2, \dots, P)$. Se observa claramente que las reglas de distancia asignan a la imagen S^α un patrón a la clase S^P si y solos si el histograma de la CCR de S^P es el más parecido a S^α .

En resumen, al comparar con cada uno de los patrones cargados predeter-

minadamente con la imagen con ruido seleccionada, si alguno de ellos tiene la mínima distancia es el que corresponderá con el patrón adeudado. Todo lo antes descrito será ocupado para realizar la clasificación de texturas por la CCR.

Hay que dejar muy en claro que al realizar el cálculo de la CCR si cambiamos de dirección la textura las características que se obtengan varían dependiendo de la dirección que tome la textura ya que la CCR no es un método invariante por lo que se debe realizar un proceso extra para lograr la invarianza, para eso en el siguiente capítulo se ahondará en el tema de la invarianza de texturas y se propondrá un método el cual guarde las características de la textura.

Capítulo 3

Métodos invariantes

En el capítulo 2, la clasificación de texturas con la CCR puede tener problemas con la detección de texturas que cambian de dirección, ya que al iniciar el método de análisis de la textura no obtiene el mismo número en decimal que el de la ventana al observar cuando la textura no tiene algún cambio, entonces por consiguiente el histograma de frecuencias del patrón detectado cambiará radicalmente perdiendo las propiedades que uno quisiera mantener para que, al realizar una clasificación con el método de Hamming, determine que patrón detecta. Para eso se abordará la invarianza de las texturas y por qué método se puede obtener para que al final se presente el método ocupado en este trabajo para la detección invariante de texturas.

3.1. Introducción a la invarianza de una imagen

En la vida cotidiana, se procesa una infinidad de información la cual ayuda a realizar procesos de clasificación.

Dentro de toda esta información que se procesa el 95 % es a base de imágenes, ya que estas son un medio de información y comunicación potente capaz de

representar información compleja y expresar ideas sin necesidad de palabras.

El reconocimiento de objetos y patrones que han sido deformado de varias maneras, es una meta que los investigadores desean alcanzar. Existen tres problemas mayores para poder abordar este tema: fuerza bruta, normalización de la imagen y parámetros invariantes.

En el método de la fuerza bruta lo que se busca es conseguir el espacio parametral de todas las imágenes degradadas, esto quiere decir, que en el entrenamiento de este método para obtener las clases de clasificación no solo obtendrán todas las clases representativas de la imagen, sino también su rotación, su escala y versiones deformadas.

En la normalización de la posición se selecciona un objeto y se transforma a una posición estándar que logre realizar la catalogación adecuada. Este método suele ser el más adecuado para lograr una clasificación, pero la complicación puede centrarse en las diversas transformaciones que necesite el objeto para lograr entrar en la clase adecuada; usualmente la normalización del objeto por si misma requiere la resolución de problemas difíciles inversos que usualmente son condiciones o posiciones degradadas.

El enfoque que tienen los parámetros invariantes, es de los métodos más prometedores y utilizados, básicamente es describir los objetos como un conjunto de parámetros cualitativos que son llamados invariantes, los cuales son insensibles a las diversas perturbaciones que podría tener el objeto y que mantendrían la información, lograr una discriminación potente para distinguir objetos en diferentes clases. (figura 3.1).

Para entender matemáticamente el proceso se define como I la función de invarianza en la cual se encuentran contemplada todas la características que contiene el espacio invariante en la cual es por encima del valor de degradación D tal que satisface la condición invariante $I(f) = I(D(f))$ [48].

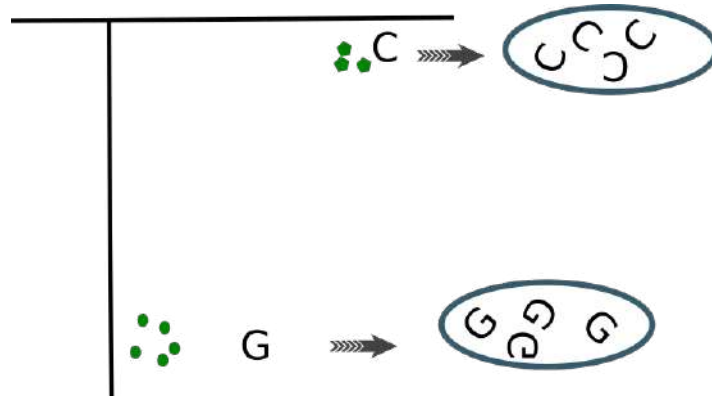


Figura 3.1: Dos clases definidas idealmente en el cual se muestra la variedad de posiciones y direcciones que tienen sus miembros

3.2. Categorías de Invarianza

La existencia de características invariantes en figuras 2D puede ser categorizada en dos puntos de vista. Los más sencillos constan de la caracterización de acuerdo al tipo de invarianza. Se reconoce rotación, traslación escala, proyectividad y geometría elástica invariante. La radiometría invariante existe de acuerdo con el contraste lineal la transformación no lineal de intensidad y la convolución. Las categorías de acuerdo a las herramientas matemáticas usadas es a continuación:

- Descriptores de contornos simples [49].
- Características de coeficientes de transformación son calculados por la transformación central de una imagen -descriptores de Fourier [50], descriptores de Hadamard [51], Coeficientes de transformación de Radón y Características basadas en onda.
- Conjunto de puntos invariante [52].
- Invariante diferencial [53].
- Momentos invariantes [54].

Otros puntos de vista tiene que ver con cómo es que se sacan las caracterís-

ticas invariantes de acuerdo a la parte del objeto a estudiar.

Invarianza Global es calculada por toda la imagen (incluido el fondo si la segmentación no ha sido aplicada). La mayoría de ellos incluye proyecciones en la imagen en funciones centrales las cuales son calculadas por integración. Comparado con las invarianzas locales, las globales son más robustas con respecto al ruido que tenga la imagen. Por otro lado, una seria desventaja es que el factor que realiza cambios locales influye todos los valores invariantes y no es localizado solamente en algunos elementos. Es por esto que la invarianza global no puede ser ocupada en detección local.

Invarianza local es en contraste, calculada centrada en una vecindad de puntos dominantes. La invarianza diferencial es representada frecuentemente en esta categoría. El limite del objeto es detectado primero para que, se calculen las características invariantes por cada limite del punto como una función de limites derivados. Como resultado, los invariantes de cualquier punto dado depende solamente del contorno de la frontera en su inmediata vecindad. Si el resto del objeto no sufre cambios, la invarianza local no es afectada. Esta propiedad hace una herramienta aparentemente perfecta para reconocimiento de objetos parcialmente obstruidos.

Invarianza semilocal mantiene las propiedades de los dos grupos sobre y evitar lo negativo de los otros. Estos métodos dividen el objeto en partes estables (la mayoría de los objetos divididos se centran en los puntos de inflexión o vértices de la frontera del objeto) y describen cada parte ocupando un tipo de invarianza global. Entonces, el objeto es caracterizado como un vector cadena de invarianzas y reconocimiento sobre oclusiones es perfeccionado por reconocimiento máximo de subcadenas [48].

3.3. Momentos invariantes a rotación, traslación y escala

Traslacional, rotación y escala (TSR por sus siglas en inglés) son las transformaciones más simples que se pueden realizar a patrones, usualmente las TSR es llamada como transformaciones similares, son cuatro transformaciones que son representadas de la siguiente forma [48]:

$$\mathbf{x}' = s\mathbf{R} \cdot \mathbf{x} + \mathbf{t}. \quad (3.1)$$

Dónde \mathbf{t} es el vector traslacional, s es coeficiente de escalamiento positivo (nótese que se se toma en cuenta la escala uniforme solo si es lo mismo de forma vertical y horizontal) y \mathbf{R} es la matriz de rotación [48].

$$R = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix}, \quad (3.2)$$

dónde α es el ángulo de rotación.

La invarianza con respecto a la TSR es necesitada en la mayoría de los sistemas ya que lo que se busca es realizar una clasificación adecuada de los objetos sin importar en que posición se encuentre el objeto o la cámara incluida la variación de la distancia entre el objeto y la cámara. Por otro lado la TSR es una aproximación adecuada a las deformaciones que pueda tener la imagen actual, si la escena es plana y (casi) perpendicular al eje óptico. Por consecuente, se ha prestado suficiente atención a estas. Mientras que las invarianzas a traslación y escala resultan ser mucho más sencillas de obtener, los mencionados métodos a rotación demuestran que no es tan elemental poderlas calcular [48].

3.3.1. Invarianza a traslación

La invarianza a traslación puede ser lograda simplemente por el objeto cambiando aparentemente tal que su centroide coincide con el origen del sistema coordinado y viceversa, cambiando la base polinomial en el centroide del objeto. En el caso de los momentos geométricos, se llaman momentos centrales geométricos [48].

$$\mu_{pq} = \iint_{-\infty}^{\infty} (x - x_c)^p (y - y_c)^q f(x, y) dx dy, \quad (3.3)$$

dónde

$$x_c = \frac{m_{10}}{m_{00}} \quad y_c = \frac{m_{01}}{m_{00}}. \quad (3.4)$$

Con las coordenadas del centroide del objeto. Por lo que la invarianza traslacional queda de lo más sencilla. Los momentos centrales pueden ser expresados en términos de momentos geométricos [48].

$$\mu_{pq} = \sum_{k=0}^p \sum_{l=0}^q \binom{p}{k} \binom{q}{l} (-1)^{k+l} x_c^k y_c^l m_{p-k, q-l}. \quad (3.5)$$

Aunque la relación es muy importante para consideraciones teóricas, normalmente se usa cuando se busca calcular los momentos centrales por medio de un algoritmo rápido de momentos geométricos [48].

3.3.2. Invarianza de escala uniforme

La invarianza a escala uniforme se obtiene mediante la normalización de cada momento. En principio, ningún momento puede ser utilizado como un factor de normalización puesto que puede dar un valor que no sea aceptado. Desde que los momentos de orden bajo son más estables al ruido y fáciles de calcular, se

normalizan la mayoría por la propiedad de potencia de μ_{00} [48]:

$$\gamma_{pq} = \frac{\gamma_{pq}}{\gamma_{00}^w}, \quad (3.6)$$

dónde

$$w = \frac{p+q}{2} + 1. \quad (3.7)$$

El momento γ_{pq} es llamado momento geométrico normalizado central [48]. Después de escalar por los factores s , momentos centrales en sus nuevas coordenadas cambian como:

$$\mu_{pq} = \iint_{-\infty}^{\infty} (x' - x'_c)^p (y' - y'_c)^q f(x', y') dx' dy'. \quad (3.8)$$

3.3.3. Invarianza a rotación

Hasta ahora, se han propuesto dos enfoques para lograr realizar una detección por medio de invarianza a traslación y escala ahora para lograr la rotación, la invarianza para la clasificación de la textura que puede dividirse ampliamente en dos categorías, es decir, métodos estadísticos y métodos basados en modelos respectivamente. En los métodos estadísticos, la textura se describe generalmente por las estadísticas de las características seleccionadas, por ejemplo, histograma invariante, elementos de textura y microestructuras. Tal y como lo menciona Zhao Davis [55] con polarogramas explotados y matrices de co-ocurrencia generalizadas para obtener características estadísticas invariantes de rotación. Duvernoy [56] propuso descriptores de Fourier para extraer la característica de textura invariante de rotación en el espectro dominio, Goyal [57] propuso un método utilizando la propiedad del textón en el histograma, Eichmann [58] presentó descriptores de textura basados en estructuras en línea

extraídas por Hough Transform.

En el modelo basados en métodos, la textura se presenta generalmente como un modelo de probabilidad o como una combinación lineal de un conjunto de funciones básicas. Kashyap [59] desarrolló un modelo circular simultáneo autorregresivo (CSAR) para Rotación de invariante clasificación de la textura. Cohen [60] caracterizan textura como campos al azar de Markov Gaussian y utiliza el máximo Probabilidad de estimar los ángulos de rotación. Chen y Kundu [61] se dirigieron rotación invariante mediante el uso de sub-bandas de descomposición de múltiples bandas y Modelo oculto de Markov (HMM). Porter [62] explotó la transformación de onda para la rotación invariante clasificación de la textura mediante el uso de la Coeficientes del filtro de onda de cuatro tomas de Daubechies.

Aunque estos métodos antes mencionados han demostrado ser invariantes a rotación, no son muy robustos a las variaciones de iluminación y así como se han mencionado algunos métodos se explicaran 2 métodos que si bien no son de los más usados son los más recurrentes en la bibliografía y estos son: Patrones binarios locales uniformes (LBPU por sus sigla en inglés) y la Transformada Discreta de Fourier.

3.3.3.1. Patrones Binario Locales Uniformes

Los patrones binario Locales (LBP por sus siglas en ingles) es uno de los métodos más recurrentes en la bibliografía dedicada a la extracción de características y clasificación de texturas [63, 64, 31, 65] ya que demuestra ser un método sencillo eficaz y de mayor impacto en diversas aplicaciones que nos permiten analizar dichas texturas , el desarrollo de dicho método.

Sea una imagen en escala de grises de tamaño $N \times M$, que viene representada de la siguiente manera se caracteriza mediante el uso de una ventana rectangular de tamaño $I \times J$. Una vez obtenido el tamaño de la ventana se recorre esta por

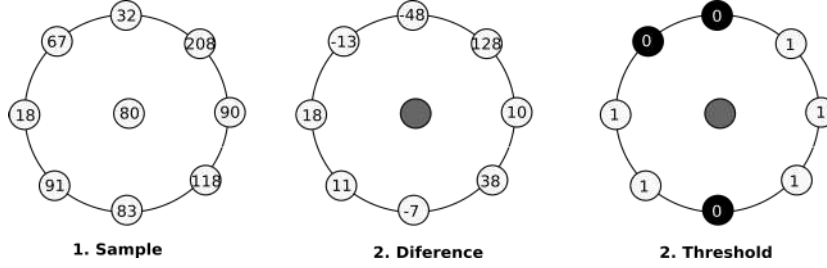


Figura 3.2: Ventana del LBPU revisando el pixel central y calculando los pesos de los pixeles vecinos

toda la imagen pixel por pixel. Entonces el método de Patrones Binario Locales determina que en la ventana toma el pixel central dado por g_c , se realiza una comparación de los de más pixeles que están dentro de la ventana que son pixeles de vecindad g_n tal que $n = 1 \dots m - 1$ que es el tamaño de la ventana y tomando la siguiente condición [63]:

$$\begin{aligned} \text{si } g_n < g_c &\Rightarrow g_n = 0 \\ g_n = 1 &\text{ en otros casos} \end{aligned} \quad (3.9)$$

Entonces, nos queda un patrón binario y se determina realizar la decodificación de dicho sistema, entonces al escribir la función que describe el funcionamiento de dicho sistema viene dada de la siguiente manera [63]:

$$LBP_{R,P} = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p, s(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases} \quad (3.10)$$

En la figura 3.2 se muestra como es que se realiza el proceso de calculo del valor en la ventana de método del LBP.

Ojala et al. [4] propuso un método eficiente, lo que se propone es que si bien una serie de patrones que se repiten en las ventanas a estudiar nos permite reducir el vector característico y realizar un análisis computacional mucho más efectivo que si solo lo hace en el método tradicional (ver figura 3.3).

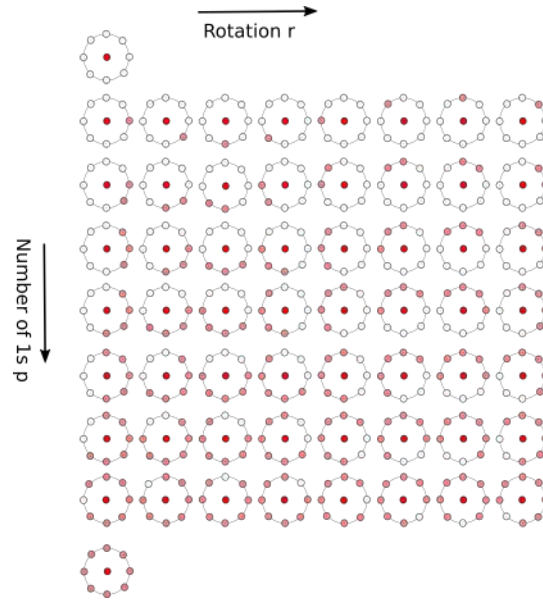


Figura 3.3: Los 36 patrones binarios invariantes de rotación únicos que pueden ocurrir en el conjunto adyacente simétrico circular de LBP. Círculos en blanco y rojo corresponden a valores de bit de 0 y 1 en la salida de 8 bits del operador. La primera fila contiene los nueve patrones uniformes y los números dentro de ellos corresponden a sus únicos códigos LBP.

El análisis se va realizando de acuerdo a la vecindad que se tiene y con los patrones base permite realizar computacionalmente más rápido el proceso. Entonces el algoritmo viene dado de la siguiente manera [4]:

$$LBP_{P,R}^{u2} = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c) & \text{si } U(LBP_{P,R}) \leq 2 \\ P + 1 & \text{otro caso} \end{cases}, \quad (3.11)$$

dónde

$$U(LBP_{P,R}) = |s(g_{P-1} - g_c) - s(g_0 - g_c)| + \sum_{p=1}^{P-1} |s(g_p - g_c) - s(g_{p-1} - g_c)|. \quad (3.12)$$

El superscritit $u2$ refleja e uso de rotación invariante uniforme en patrones

que tienen U valor de al menos 2. Por definición exactamente el patrón binario uniforme $P + 1$ puede ocurrir en un vecindario circular declarado por P pixeles. La ecuación 3.12 asigna una marca única a cada uno de ellos que corresponde con el número “1” bits en el patrón mientras el patrón “no uniforme” son agrupados debajo de las marcas diversas ($P + 1$).

3.3.3.2. Transformada Discreta de Fourier

Matsuyama et al. [66] ha propuesto que la aplicación de la transformada de Fourier se debe realizar por medio de escala de grises y sacando el Espectro de la Transformada de Fourier en dos dimensiones, otra opción como la que proponen Florindo y Bruno [67] sería realizando una binarización del mismo y aplicando el mismo método, Azencott [68] mencionan que por medio de un algoritmo geométrico de corrección en la imagen se puede realizar el algoritmo antes mencionado obteniendo muy buenos resultados en este caso solo se ocupa la transformada y espectro de Fourier. En 2003, Li-Ma et al. demostraron que si la textura es periódica, entonces se puede inferir que el espectro de Fourier es periódico [69].

Para que pueda realizarse de una manera más sencilla y realizar un cálculo más eficiente se procede a ocupar la transformada de Fourier discreta de una dimensión. Esta, por definición es: Sea una función $f(x) \in \mathbb{R}$ se calcula la Transformada de Fourier, entonces la nueva función quedaría $f(u) \in \mathbb{C}$ al que la función de conversión de dicha transformada queda descrita de la siguiente manera:

$$f(u) = \frac{\sum_{l=0}^M \sum_{k=0}^N f(k) e^{-i2\pi kl/M}}{M}. \quad (3.13)$$

Es de importancia mencionar que al obtener el espectro de Fourier, no estaría centrado, para evitar este problema se recurre a una corrección, de esta manera al reescribir la Transformada de Fourier queda de la siguiente manera [66]:

$$f(u) = \frac{\sum_{l=0}^M \sum_{k=0}^N -1^l f(k) e^{-\frac{i2\pi kl}{M}}}{M}. \quad (3.14)$$

Con dicha corrección obtendremos el espectro en el centro de la gráfica, la ecuación que describe adecuadamente el espectro es de la siguiente manera [66]:

$$f(x) = \sqrt{\mathbb{R}(f(u))^2 + \mathbb{C}(f(u))^2}. \quad (3.15)$$

La figura 3.4 nos muestra un ejemplo del espectro de Fourier.

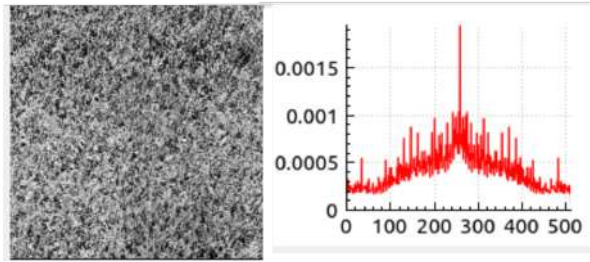


Figura 3.4: Espectro de Fourier

3.3.3.3. Propiedades de la transformada de Fourier

Tal y como las transformada de Fourier y la transformada de Laplace para señales continuas en el tiempo y con la transformada z en el dominio del tiempo discreto aperiódico secuencial, las propiedades centrales de la transformada de Fourier son fundamentalmente importantes para su éxito en el análisis de señales estas propiedades son mencionadas a continuación:

1.- Linealidad

Considere dos secuencia periódicas $\tilde{x}_1[n]$ y $\tilde{x}_2[n]$ ambas con periodo N tal que [70]:

$$\tilde{x}_1[n] \stackrel{TFD}{\Leftrightarrow} \tilde{X}_1[k]. \quad (3.16)$$

$$\tilde{x}_2[n] \stackrel{TFD}{\Leftrightarrow} \tilde{X}_2[k]. \quad (3.17)$$

Entonces [70]:

$$ax_1[n] + ax_2[n] \stackrel{TFD}{\Leftrightarrow} a\tilde{X}_1[k] + \tilde{X}_2[k]. \quad (3.18)$$

La propiedad de linealidad viene de las ecuaciones de la Transformada de Fourier y la Transformada de Fourier Inversa respectivamente.

2.- Desplazamiento de una secuencia

Si una secuencia periódica $\tilde{x}[n]$ tiene coeficientes de Fourier $\tilde{X}[k]$ entonces $\tilde{X}[n - m]$ es una versión de desplazamiento de y:

$$\tilde{x}[n - m] \stackrel{TFD}{\Leftrightarrow} e^{\frac{-i2\pi kl}{M}} \tilde{X}[K]. \quad (3.19)$$

La demostración de esta propiedad ya ha sido escrita por Oppenheim [70], cualquier desplazamiento que es mayor o igual al periodo ($m \geq N$) no puede ser distinguido en el dominio del tiempo por un desplazamiento pequeño m_1 tal que $m = m_1 + m_2N$ donde m_1 y m_2 son enteros y (otra forma de iniciar es con módulo N , o el equivalente que es cuando m es dividido por N).

3.- Dualidad

Por la fuerte similitud entre el análisis de Fourier y la síntesis de ecuaciones en el tiempo continuo, existe una dualidad entre el dominio del tiempo y el de la frecuencia. Por lo que, en el tiempo discreto de la Transformada de Fourier existe una gran variedad de funciones: señales aperiódicas en el dominio del tiempo discreto, secuencias aperiódicas, mientras su Transformada de Fourier es siempre una función periódica con una frecuencia constante.

4.- Propiedades de simetría

La transformada de Fourier como una secuencia aperiódica tiene un número

útil de propiedades simétricas.

El espectro de Fourier en 2D por sus características definidas se establece que tiene propiedades invariantes a rotación y escala [71, 72], por lo que se puede inferir que estas mismas propiedades las contiene el espectro de Fourier en 1D. Esto nos va a servir para el mejoramiento del método de la CCR la cual como se ha reiterado en diversas ocasiones.

3.4. Volviendo Invariante la CCR

Retomando lo expuesto en las secciones anteriores, el algoritmo simple procede de la siguiente manera: Para poder realizar la CCR es necesario realizar la binarización adecuada de la imagen por que como bien se explicó, solo funciona con imágenes binarizadas entonces, para encontrar el umbral adecuado se procede a calcular el umbral con el método de Otsu, una vez obtenido el umbral se realiza la binarización por umbral a la imagen y es ahí donde la CCR realiza el cálculo pertinente y obtiene el vector característico. En este punto entra el procedimiento de mejora de la CCR. El método propuesto es separado en 2 fases:

1. CCR
2. Espectro de Fourier

Cuando se carga una textura al algoritmo, el método calcula el histograma de la CCR posteriormente, ese mismo histograma se pasa al plano complejo con la transformada de Fourier, para poder obtener las propiedades invariantes de rotación, la conclusión del algoritmo es calcular el espectro de Fourier de 1 dimensión (EF1D). Para ejemplificar el producto final se observa claramente en la figura 3.5:

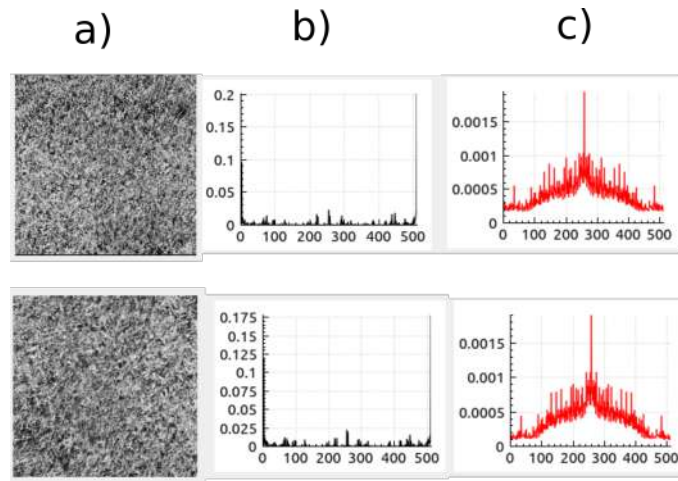


Figura 3.5: Columna a) Imagen original e imagen rotada a 135° Columna b) Histograma de la CCR normalizado de la imagen rotada y no rotada se puede observar como existe un cambio en el histograma de frecuencias al rotar la imagen Columna c) Histograma del Espectro de Fourier de una dimensión (EF1D) ahí se puede observar que no hay una variación visible o es mínima que no afecta la detección del patrón que es.

Como se puede observar en la figura 3.5, el vector característico de la textura tiene poca perturbación sin importar en qué dirección se encuentre dicha textura, estas mismas perturbaciones que aparecen en el vector por el cambio de dirección que sufre la textura son vestigios de la CCR, pero con la ayuda de EF1D logra compensar esta variación por lo que, la Transformada de Cúmulos Coordinados combinada con el espectro de Fourier de una dimensión (CCR/EF1D) es un método que promete tener detecciones adecuadas.

En este capítulo se ha realizado una revisión de lo que se trata la invarianza para poder llegar a la mejora del método, mismo que se presentará en el siguiente capítulo en el cual se explicarán experimentos tanto de la CCR sola como la mejora de invarianza a rotación con ayuda del espectro de Fourier.

Capítulo 4

Experimentos

En los capítulos 1, 2 y 3 se realizó una explicación general de varios conceptos tales como la definición de textura, la transformada de la CCR, la invarianza a rotación y la mejora del método de clasificación en este capítulo se presentan todos los experimentos realizados para poder determinar la eficiencia que tiene el método.

4.1. Experimento 1: Clasificación de texturas con ruido ocupando la CCR

Tal y como se presentó en el artículo del XVII Congreso Nacional de Ingenierías Electromecánica y de Sistemas 2018 (CNIES) [73]:

Se generó un conjunto de imágenes por computadora de la siguiente manera: Se genera un valor aleatorio entre 0 y 225 ya que el patrón base tiene una medida de 5×5 , entonces esos valores que se obtuvieron de forma aleatoria se codifican en decimal. Los valores propuestos para el experimento vienen mostrados en la tabla 4.1:

Tabla 4.1: Valores aleatorio y su representación en binario

Número decimal	Número en binario
1763554	0000110101110100011100010
26625306	1100101100100010100011010
1304402	0000100111110011101010010
3707135	0001110001001000011111111
28625692	1101101001100101100011100
22	000000000000000000010110
91	00000000000000000001011011
6039510	0010111000010011111010110
387	0000000000000000110000011
5767	0000000000010110100001111

Con estos patrones, lo que se procede hacer es configurar de manera periódica en una matriz de 100x100 para generar una imagen con textura hecha por computadora y las texturas creadas se muestran en la figura 4.1:

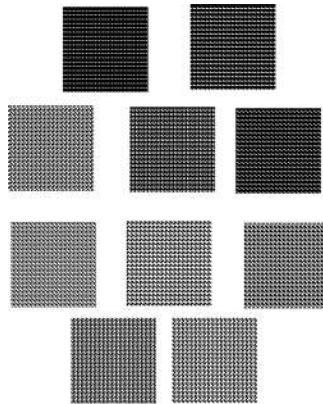


Figura 4.1: Conjunto de texturas generadas computacionalmente

Estas imágenes son la base de datos que ayuda en la detección de patrones. Ahora con una cierta cantidad de ruido se les agrega a las imágenes y se procede de la siguiente manera:

Sea un conjunto de imágenes binaria $S^P = [S_{I_m}^p]$ $p = (1, 2, \dots, P)$ se genera una serie de elementos con ruido $B = S^p(k, \alpha_p) = [s_{I_m}^p(k, \alpha_p)]$ agregándole ruido a ese patrón de texturas, en este trabajo en particular se ocupa ruido por

probabilidad. Las imágenes fueron generadas para cada k -ésimo nivel de ruido. Si Δ es la diferencia entre los dos niveles de ruido más cercanos entonces, la imagen $S^P(k, \alpha_p) = [s_{lm}^p(k, \alpha_p)]$ del k -ésimo nivel de ruido puede ser calculado de la siguiente forma:

$$s_{lm}^p(k, \alpha_p) = \begin{cases} s_{lm}^p & \text{si } \text{rand} < 1 - k\Delta \\ 1 - s_{lm}^p & \text{otro caso} \end{cases} . \quad (4.1)$$

donde α_p indica la α_p -ésima raíz del ruido agregado a $p = (1, 2, \dots, P)$ y rand es el aleatorio función generadora de números con distribución uniforme en el rango $(0,1)$. Cuando $k = 0$, la imagen generada por la ecuación del nivel de ruido coincide con la textura original del patrón, $S^P(0, \alpha_p) = S^P$. Después de que se genera el conjunto B , la CCR se usa para calcular los histogramas $H_{(I,J)}^{\alpha^2}(k, a_p; b)$ correspondientes a cada imagen.

El porcentaje de ruido puesto en cada imagen se aumenta de diez en diez empezando en 10 % y terminando en el 100 %, tomando en cuenta que la base de datos está dada por diez patrones diferentes y al tener diez niveles de ruido en total, se obtiene un conjunto de 100 imágenes de diferentes niveles de ruido para poder realizar las pruebas pertinentes.

En la CCR, la ventana que se decide trabajar es de 3×3 por lo que la longitud del histograma dado es de la siguiente manera $23 \times 3 = 512$, entonces se tiene una distribución de frecuencias que tiene como límite inferior el 0 y límite superior el 512.

Al empezar a realizar la clasificación de las texturas en el 10 % de ruido todas las imágenes fueron detectadas correctamente y determinado a que patrón corresponden pero al aumentarle el porcentaje de ruido a un 20 % se empieza a notar que no todas las imágenes pudieron ser detectadas adecuadamente teniendo a la textura con el patrón base más pequeño (22) la que no pudo acertar

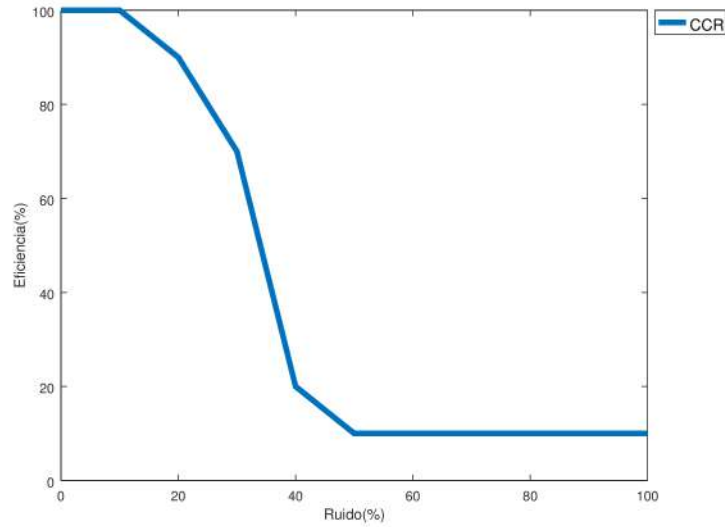


Figura 4.2: Eficiencia del método

al patrón correspondiente, a partir del 30 % de ruido, los aciertos que tienen el sistema decrecen notablemente teniendo a llegar a detectar solo una imagen adecuadamente. Para poder determinar la eficiencia que tiene el sistema, se utiliza la siguiente ecuación:

$$R = \frac{N_k}{10p} \times 100 \%. \quad (4.2)$$

Tal que $10P$ corresponde al tamaño de la población, es decir, la cantidad de texturas que se encuentran en la base de datos, N_k corresponde al total de texturas con k niveles de ruido que fueron detectadas correctamente.

En términos de eficiencia, en la figura 4.2 se muestra con mejor precisión que hasta 30 % de ruido tenemos una eficiencia de reconocimiento de patrones de un 70 % mientras que del 40 % en adelante solo tenemos una eficiencia de un 10 %.

La explicación del comportamiento del sistema a medida que aumenta el

ruido en la imagen es la siguiente; como se está generando de manera aleatoria el ruido en la imagen se afecta de manera significativa el pixel elegido, entonces cambia el valor binario del pixel, por lo que se tiende una degradación de la imagen a medida de aumentamos el ruido en la misma, luego para el tipo de ruido utilizado en este trabajo la mayoría de los pixeles tienen que cambiar su los valores y algunas texturas se reconstruyen parcialmente. El nivel de la reconstrucción depende de la regularidad de la primera textura, por lo que con los patrones más chicos, la reconstrucción de la textura es casi total mientras que los patrones más grandes su reconstrucción es mínima. Hay que tomar muy en cuenta que una textura que tiene el 100 % de ruido es una textura totalmente negra, por consiguiente los patrones de reconocimiento podrían perderse, pero puede tener cierta morfología al patrón original. En la figura 4.3 se presenta el conjunto de imágenes con el porcentaje de ruido máximo que pudo determinar ser detectadas adecuadamente.

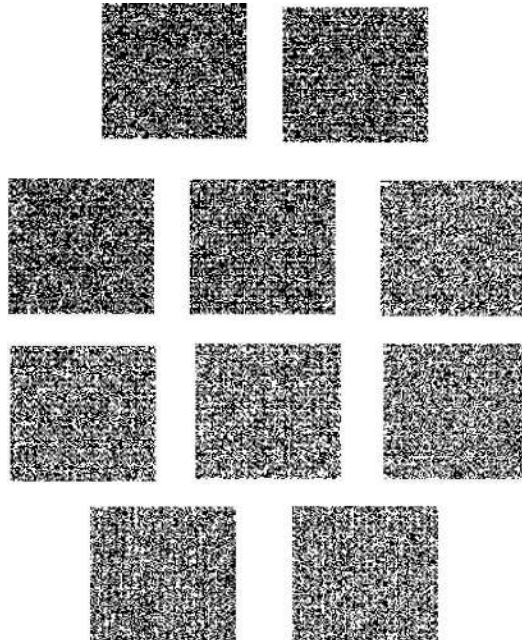


Figura 4.3: Patrones de texturas con 30 % de ruido

En este experimento se corroboró que el método nos da una eficiencia adecuada pero se necesita determinar hasta qué punto puede ser mucho mejor que los demás métodos propuestos.

4.2. Experimento 2: Comparativo de clasificación de texturas con la CCR y LBP

Para poder realizar dicho reconocimiento de texturas con diferentes métodos se decide que la base de datos será la misma ocupada en el artículo del CNIES 2018 (ver figura 4.1).

A continuación se genera la base de datos a ocupar una vez hecha, se procede a agregar ruido a la imagen para ver hasta que cierto porcentaje de ruido puede detectar el sistema, en este caso se le agregó ruido sal y pimienta.

El método propuesto, ayuda a tomar lo necesario para realizar adecuadamente la comparación para saber qué tan eficaz son entre si estos métodos que se ocuparon para realizar la comparativa con la CCR fue: Patrones Binarios Locales.

Para poder realizar el discernimiento a que patrón corresponde dicha imagen se ocupa la distancia de Hamming que previamente ha sido descrita. La forma en que fue programado el sistema fue de la siguiente manera:

Teniendo 2 bases de datos diferentes correspondientes a los diversos métodos propuestos para este estudio, se inicia la carga de una imagen que se le agrega el ruido deseado, se elige el método a probar y una vez empezada la secuencia, se realizan las operaciones pertinentes dependiendo del método y se despliega el patrón detectado. Para la transformada de cúmulo coordinados se procede a realizar de igual manera con un vector de tamaño 513 ya que como se explicó en la teoría dependiendo de la ventana se determina la longitud del vector, en este caso la ventana es de 3×3 entonces $29=512$. Por último, para los patrones

binarios locales tendremos un vector de 513, ya que la ventana será de la misma medida que la transformada de cúmulos coordinados. Los resultados obtenidos se presentan en la tabla 4.2:

Tabla 4.2: Resultados del experimento 2 tomando como nomenclatura: a: Patrón 22, b: Patrón 91, c: Patrón 387, d: Patrón 5767, e: Patrón 1304402, f: Patrón 3707135, g: Patrón 6039510, h: Patrón 1763554, i: Patrón 26625306 e j: Patrón 28625692.

	Ruido	a	b	c	d	e	f	g	h	i	j	Eficiencia
CCR	10 %	si	si	si	si	si	si	si	si	si	si	100.00 %
	20 %	no	si	si	si	si	si	si	si	si	si	90.00 %
	30 %	no	no	no	si	si	si	si	no	no	no	70.00 %
	40 %	no	no	no	no	no	si	no	no	si	no	20.00 %
	50 %	no	no	no	si	no	no	no	no	no	no	10.00 %
	60 %	no	no	no	si	no	no	no	no	no	no	10.00 %
	70 %	no	no	si	no	no	no	no	no	no	no	10.00 %
	80 %	no	no	si	no	no	no	no	no	no	no	10.00 %
	90 %	si	no	no	no	no	no	no	no	no	no	10.00 %
	100 %	si	no	no	no	no	no	no	no	no	no	10.00 %
											Promedio	34.00 %
LBP	10 %	si	si	si	no	no	no	si	no	si	si	60.00 %
	20 %	si	no	no	no	si	no	si	si	no	no	40.00 %
	30 %	si	no	no	si	no	no	si	no	si	si	50.00 %
	40 %	si	no	no	no	no	no	si	si	no	no	30.00 %
	50 %	si	no	no	no	no	no	no	si	si	no	30.00 %
	60 %	no	si	no	no	no	no	si	no	no	si	30.00 %
	70 %	si	no	no	no	no	no	si	si	no	no	30.00 %
	80 %	no	si	no	no	si	no	si	no	no	no	30.00 %
	90 %	si	no	si	si	si	no	no	no	no	no	40.00 %
	100 %	si	no	si	no	no	no	no	no	no	no	20.00 %
											Promedio	36.00 %

Y en la figura 4.4 de la eficiencia del los métodos a comparar:

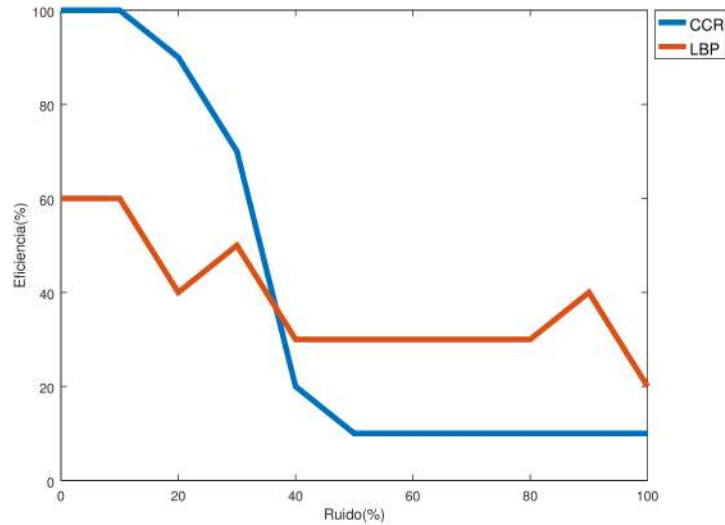


Figura 4.4: Gráfica de eficiencia de la CCR y LBP

En la figura 4.4 se obtiene la eficiencia que va teniendo cada uno de los métodos y se ocupa la ecuación 30.

Como bien se puede observar en la figura anterior, conforme se agrega ruido a los patrones es más difícil que pueda obtener el resultado esperado, sin embargo, en el método de Patrones Binarios Locales, la eficiencia que se puede observar en la gráfica no cae tan abruptamente como el de CCR pero no es tan fácil que pueda determinar que patrón es el que se está comparando.

Los resultados obtenidos, nos permiten determinar que en promedio a partir de 30 % de ruido la detección empieza a ser más difícil de realizarse, puesto que la cantidad de píxeles alterados va en aumento, se corrompe el textón base de la textura y la periodicidad de la misma deja de ser tal, para convertirse en una textura variable.

4.3. Experimento 3: Clasificación de texturas invariante a rotación por la CCR/EF1D

Tal y como se presentó en el 6° Congreso Internacional de Robótica y Computación en La Paz, BCS [74]: La base de datos que es ocupada en el experimento son diez patrones de texturas tomados del álbum de Brodatz que son mostrados en la figura 4.5:

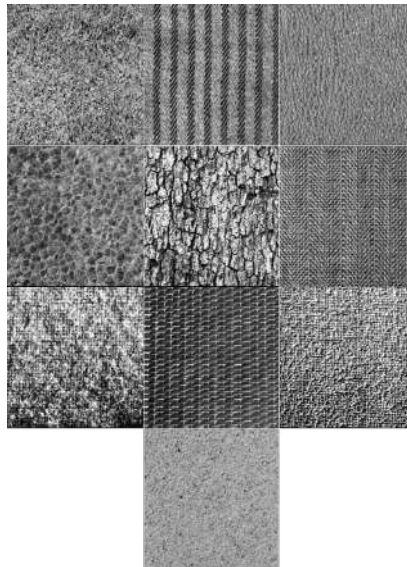


Figura 4.5: Base de datos de Texturas tomadas del álbum de Brodatz

Esta base de datos contiene diferentes texturas: pasto, tela, acrílico arrugado, burbujas sintéticas, madera, tejido de yute, lana, tejido de acero, tejido sintético y arena estas texturas se clasifican de acuerdo a la nomenclatura que Brodatz les puso (D9, D11, D24, D112, D12 H.E., D16, D19H.E., D65, D84 H.E. y D29). Para ellos se determinaron 7 direcciones pertinentes: 3 ángulos pequeño ($5^\circ, 10^\circ, 15^\circ$) y 4 ángulo grandes ($45^\circ, 135^\circ, 225^\circ, 315^\circ$) ya que se revisó que tan efectivo es el sistema por lo tanto, el total de imágenes que se ocuparan en el experimento son 70 (ver figura 4.6):

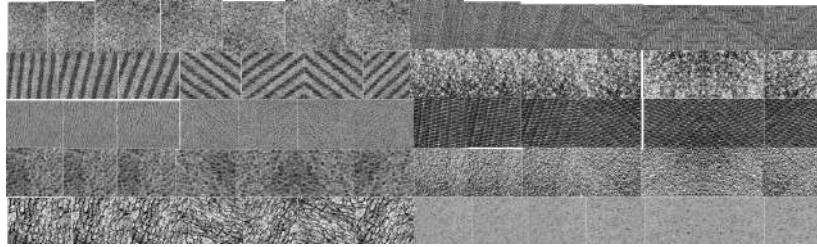


Figura 4.6: Conjunto de texturas rotadas a 5° , 10° , 15° , 45° , 135° , 225° y 315° para mejorar el método invariante a rotación

La similitud de las texturas de la base de datos a uno de los patrones de texturas cargado al algoritmo de la CCR/EF1D se calculan utilizando la distancia de Hamming (ecuación 9). La tabla 4.3 muestra un resumen de todas las texturas probadas y clasificadas correctamente:

Tabla 4.3: Respuesta del método de la CCR/EF1D, donde: 1(D9), 2(D11), 3(D24), 4(D112), 5(D12 H.E.), 6(D16), 7(D19 H.E.), 8(D65), 9(D84 H.E.) Y 10(D29)

	θ	1	2	3	4	5	6	7	8	9	10
CCR/ETF1D	5°	si	si	si	si	si	si	si	si	si	si
	10°	si	si	si	si	si	si	si	si	si	si
	15°	si	si	si	si	si	si	si	si	si	si
	45°	no	si	si	si	si	si	si	si	si	si
	135°	si	si	si	si	no	si	si	si	no	si
	225°	no	si	si	si	si	si	si	si	si	si
	315°	si	si	si	si	no	si	si	si	no	si

Para obtener la eficiencia de las texturas se ocupa la siguiente ecuación:

$$R = \frac{N_k}{10p} \times 100\%. \quad (4.3)$$

Tal que, $10P$ corresponde al tamaño de la población, es decir, la cantidad de texturas que se encuentran en la base de datos, N_k corresponde al total de texturas con k ángulos que fueron detectadas correctamente. Esta ecuación es

ocupada para cada ángulo, y el resultado es puesto en la tabla 4.5 y la figura 4.7 nos muestra la eficiencia de este método:

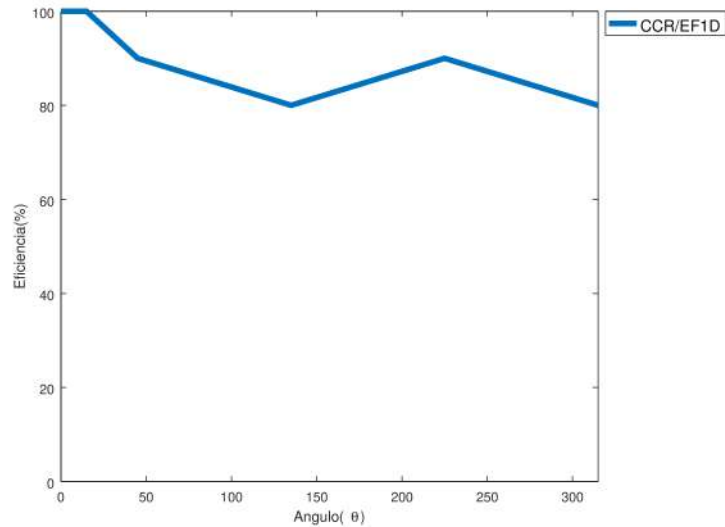


Figura 4.7: Eficiencia del experimento

Tabla 4.5: Eficiencia de cada ángulo

Ángulo (θ)	R
5°	100 %
10°	100 %
15°	100 %
45°	90 %
135°	80 %
225°	90 %
315°	80 %
Promedio	91.43 %

Los resultados de este estudio muestran que la gran mayoría de las texturas pudieron tener menos de 2 direcciones sin determinar correctamente, estos resultados son consistentes con estudios de otros métodos analizados y se sugiere que este método puede clasificarse perfectamente, sin embargo, este resultado no ha sido previamente descrito. Hay varias explicaciones posibles para estos

resultados, al tener imágenes que ya fueron procesadas previamente para realizar experimentos adecuadamente eliminan imperfecciones a la luz y por ende su detección puede llegar a resultar casi perfecta. Por lo tanto, se recomiendan estudios futuros sobre el tema actual.

4.4. Experimento 4: Método comparativo CCR/EF1D y LBPUHF para clasificar texturas invariantes a rotación

El siguiente experimento fue realizado para comparar la CCR/EF1D con otro método que también realiza la clasificación de texturas invariante, en este caso se determinó ocupar el método de Patrones Binarios Locales Uniformes con histogramas de Fourier (LBPUHF).

Para este experimento se procedió a ocupar una base de datos de 20 imágenes diferentes tomadas del álbum de Brodatz, en la figura 4.8:

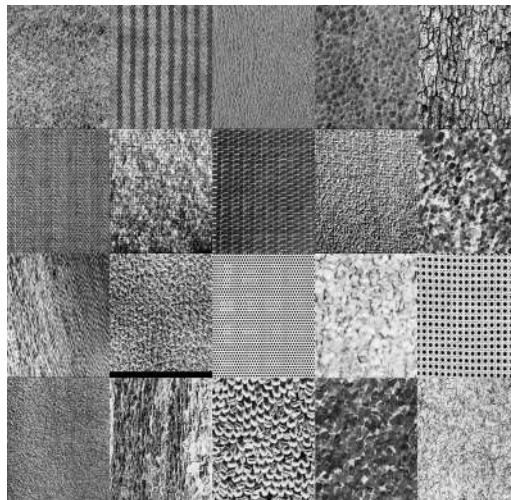


Figura 4.8: Base de datos de texturas del álbum de Brodatz

Para este experimentos se procedieron a ocupar 11 ángulos diferentes ($5^\circ, 10^\circ, 15^\circ,$

20°, 25°, 30°, 35°, 45°, 135°, 225° y 315°) lo que nos da un total de 220 imágenes probadas por cada método.

La similitud de las texturas de la base de datos a uno de los patrones de texturas cargado al algoritmo se calculan utilizando la distancia de Hamming (ecuación 9). La tabla 4.6 y 4.7 muestra todas las texturas probadas y cuales fueron clasificadas correctamente:

Tabla 4.6: Resultados del método CCR/EF1D, donde: 1(D9), 2(D11), 3(D24), 4(D112), 5(D12 H.E.), 6(D16), 7(D19 H.E.), 8(D65), 9(D84 H.E.), 10(D27), 11(D38 H.E.), 12(D92 H.E.), 13(HH), 14(G), 15(D101), 16(D4), 17(D12), 18(D74), 19(D98) y 20(D110).

CCR/ETF1D	θ	1	2	3	4	5	6	7	8	9	10	
	5°	si	si	si	si	si	si	si	si	si	si	
	10°	si	si	si	si	si	si	si	si	si	si	
	15°	si	si	si	si	si	si	si	si	si	si	
	20°	si	si	si	si	si	si	si	si	si	si	
	25°	si	si	si	si	si	si	si	si	si	si	
	30°	si	si	si	si	si	si	si	si	si	no	si
	35°	si	si	si	si	si	si	si	si	si	no	si
	45°	no	si	si	si	si	si	si	si	si	no	si
	135°	si	si	si	si	no	si	si	si	si	no	si
	225°	no	si	si	si	si	si	si	si	si	no	si
	315°	si	si	si	si	no	si	si	si	si	no	si
CCR/ETF1D	θ	11	12	13	14	15	16	17	18	19	20	
	5°	si	si	si	si	si	si	si	si	si	si	
	10°	si	si	si	si	si	si	si	si	si	si	
	15°	si	si	si	si	si	si	si	si	si	si	
	20°	no	si	si	si	si	si	si	si	si	si	
	25°	no	si	si	si	si	no	no	si	si	si	
	30°	no	si	si	si	si	no	no	si	si	si	
	35°	no	si	si	si	si	no	no	si	si	si	
	45°	no	si	si	si	si	no	no	si	si	no	
	135°	no	si	si	si	si	no	no	si	si	no	
	225°	no	si	si	si	si	no	no	si	si	no	
	315°	no	si	si	si	si	no	no	si	si	no	

Tabla 4.7: Resultados del método LBPUGHF, donde: 1(D9), 2(D11), 3(D24), 4(D112), 5(D12 H.E.), 6(D16), 7(D19 H.E.), 8(D65), 9(D84 H.E.), 10(D27), 11(D38 H.E.), 12(D92 H.E.), 13(HH), 14(G), 15(D101), 16(D4), 17(D12), 18(D74), 19(D98) y 20(D110).

LBPUGHF	θ	1	2	3	4	5	6	7	8	9	10
	5°	si	no	si	si	si	si	no	si	si	si
	10°	si	no	si	si	si	si	no	si	si	si
	15°	si	no	si	si	si	si	no	no	si	si
	20°	no	no	si	si	si	si	no	no	si	si
	25°	no	no	si	si	si	si	no	no	si	si
	30°	no	no	si	si	si	si	no	no	si	si
	35°	no	no	si	si	si	si	no	no	si	si
	45°	no	no	no	si	si	si	no	no	si	si
	135°	no	no	no	si	si	si	no	no	si	si
	225°	no	no	no	si	si	si	no	no	si	si
	315°	no	no	no	si	si	si	no	no	si	si
LBPUGHF	θ	11	12	13	14	15	16	17	18	19	20
	5°	si	no	si	si	no	si	si	no	si	si
	10°	si	no	si	si	no	si	no	no	si	no
	15°	si	no	si	no	no	si	si	no	si	si
	20°	si	no	si	no	no	si	si	no	si	si
	25°	si	no	si	no	no	si	si	no	si	si
	30°	si	no	si	no	no	si	si	no	si	no
	35°	si	no	si	no	no	si	si	no	si	si
	45°	si	no	si	no	no	si	si	no	si	si
	135°	si	no	si	no	no	si	si	no	si	si
	225°	si	no	si	no	no	si	si	no	si	si
	315°	si	no	si	no	no	si	si	no	si	si

Para obtener la eficiencia de las texturas se ocupa la ecuación 31. Esta ecuación es ocupada para cada ángulo, y el resultado de ambos métodos es puesto en la tabla 4.8 y 4.9 y la figura 4.9 nos muestra la eficiencia de estos métodos con la mencionada ecuación.

Tabla 4.8: Tabla de eficiencia del método CCR/EF1D

Ángulo (θ)	R
5°	100 %
10°	100 %
15°	100 %
20°	95 %
25°	85 %
30°	80 %
35°	80 %
45°	70 %
135°	70 %
225°	70 %
315°	70 %
Promedio	83.63 %

Tabla 4.9: Tabla de eficiencia del método LBPUIHF

Ángulo (θ)	R
5°	75 %
10°	65 %
15°	65 %
20°	65 %
25°	60 %
30°	60 %
35°	55 %
45°	55 %
135°	55 %
225°	55 %
315°	55 %
Promedio	61.36 %

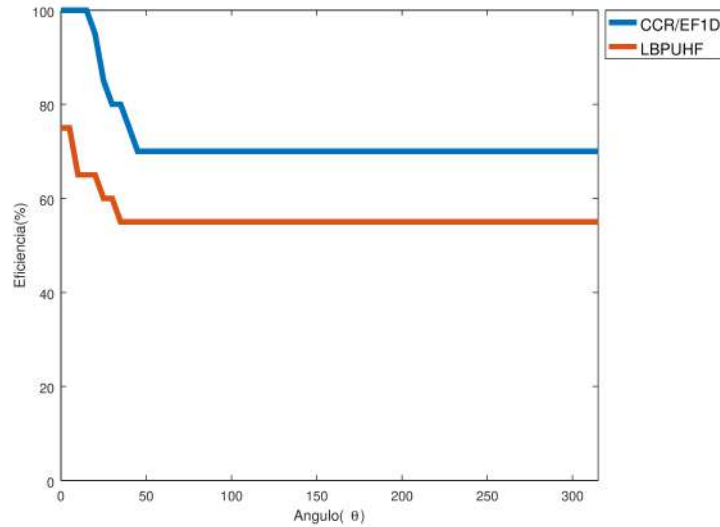
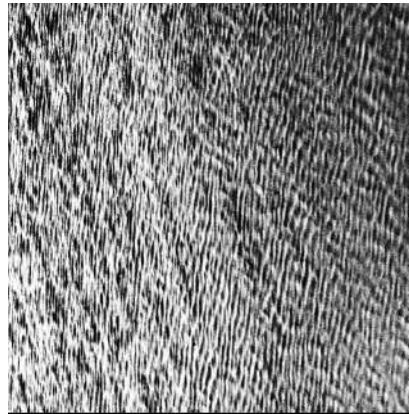


Figura 4.9: Eficiencia de la CCR/EF1D y LBPUHF

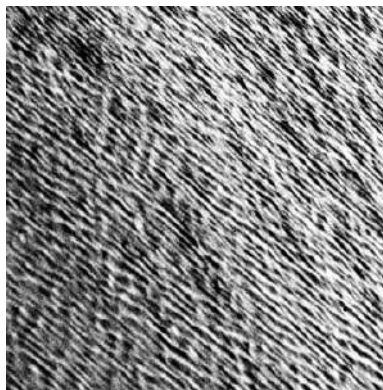
Como se puede observar el método propuesto con respecto al ya estudiado presenta una gran mejoría para poder realizar una detección de los patrones una de las posibles causas de las cuales no se hubiera podido llegar a realizar una detección adecuada es por que al rotar las imágenes los pixeles cambian de lugar nos darán un nuevo valor de intensidad en la posición y aun que, la Transformada de Fourier se encargue de mantener las propiedades, no podrá mantener todas las características de la imagen.

Un punto que se debe dejar muy claro es sobre las imágenes que se rotan pues al realizar la rotación la nueva imagen queda con espacios que se pueden volver blancos o negros dependiendo el formato de imagen que se guarde y si se realiza el procedimiento el método vera esos pixeles y afectara en el vector característico, por lo que para evitar estos excesos de pixeles se recorta la imagen tratando de mantener toda la información necesaria para que así se logre realizar la clasificación de la textura. Para lograr eso se escoge una dimensión lo bastante chica para que en cada rotación que sufra la imagen pueda

mantener la información y los píxeles se copian del centro de la imagen hasta el tamaño que se determinó y así se crean las imágenes rotadas, por lo que pueden obtener detalles de esa parte de la imagen que hacen que la detección resulte difícil tal y como se logra ver en la figura 4.10 a y b.



(a) Imagen original



(b) Imagen rotada a 45° de la figura a

Figura 4.10: Imagen Brodatz D38 H.E. como se observa tiene píxeles negros que al rotar afectan la detección adecuada

Si realizamos un análisis de LBPUHF podemos observar que tuvo dificultades en 5 patrones para poder detectar correctamente y la tabla nos presenta un resultado de eficiencia muy bajo por esa falta de detección. Comparando los dos resultados, se puede ver que el CCR / EF1D obtiene mejores resultados

que los LBPUGHF. Estos resultados son coherentes con los de otros estudios y sugieren que el método proporciona una clasificación adecuada con las texturas. Sin embargo, más investigación sobre este tema necesita ser emprendida antes de la asociación con la rotación invariante para que la clasificación se entienda más claramente.

4.5. Experimento 5: Método comparativo CCR/EF1D y LBPUGHF para clasificar texturas sin procesamiento invariantes a rotación

El siguiente experimento fue realizado para comparar con el mismo método que el experimento el cual es Patrones Binarios Locales Uniformes con histogramas de Fourier (LBPUGHF).

Para este experimento se procedió a ocupar una base de datos de 10 imágenes diferentes tomadas de objetos cotidianos mostrados en la figura 4.11:



Figura 4.11: Base de datos de las diferentes texturas

Para obtener la eficiencia de las texturas se ocupa la ecuación 31. Ésta ecuación es aplicada para cada ángulo, y el resultado de ambos métodos es puesto en la tabla 4.12 y 4.13, la figura 4.12 nos muestra la eficiencia de este método.

Tabla 4.12: Tabla de eficiencias del método CCR/EF1D

Ángulo (θ)	R
5°	90 %
10°	90 %
15°	90 %
20°	90 %
25°	80 %
30°	90 %
35°	100 %
45°	100 %
135°	90 %
225°	100 %
315°	90 %
Promedio	91.81 %

Tabla 4.13: Tabla de eficiencia del método LBPUHF

Ángulo (θ)	R
5°	100 %
10°	100 %
15°	100 %
20°	90 %
25°	90 %
30°	90 %
35°	90 %
45°	90 %
135°	90 %
225°	90 %
315°	90 %
Promedio	92.72 %

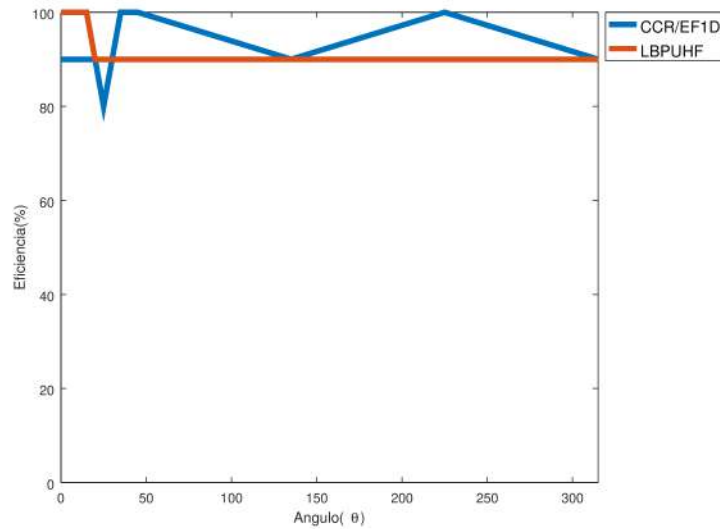


Figura 4.12: Eficiencia de ambos métodos de las tablas 4.9 y 4.10

El presente estudio fue diseñado para determinar el efecto del uso de la textura real en ambos algoritmos con el fin de probar la eficiencia. Los resultados de este estudio indican que la LBPUHF tiene tan buena eficiencia como la CCR / EF1D con diferencia para una textura en dirección diferente, los resultados actuales parecen ser consistentes con otras investigaciones que encontraron que la LBPUHF tiene una alta detección de textura correcta y las pruebas CCR / EF1D demuestran que es casi igual a la estudiada anteriormente método, sin embargo, los resultados del estudio actual no apoyan la investigación anterior porque el CCR / EF1D es un Nuevo método de rotación invariante. Hará falta acometer más estudios que tomen en cuenta estas variables.

Capítulo 5

Conclusiones

De los capítulos 2 al 4 se ha establecido cada uno de los parámetros necesarios para realizar un análisis y clasificación de las diferentes texturas por medio de la Transformada de Cúmulos Coordinados a continuación se presentan la conclusión del trabajo:

Las texturas se encuentran inmersas en nuestra vida cotidiana y es gracias a los sistemas computacionales que podemos darles un diferente enfoque y análisis para realizar diferentes procesos que pueden ser en una industria textilera, en una línea de producción, en un detector para usuario entre otros más. Durante todo el capítulo se expresó que la definición de textura no era única, si no que dependiendo de la aplicación que se le diera al trabajo era la definición a ocupar por lo que se tiene un gran conjunto de definiciones que ayudan a entender aún mejor que es una textura en el ámbito computacional.

El procesamiento de imágenes son un conjunto de métodos que nos ayudan a mejorar o sacar información a una imagen ya sea que esta se encuentre en color o en escala de grises, siempre será necesario realizar procesos matemáticos para poder entender mejor la imagen a estudiar y verificar si ciertos métodos

propuestos son buenos al implementarse en una imagen o necesitan ciertas correcciones.

Los diferentes métodos presentados en el trabajo demuestran que se tiene un gran abanico de posibilidades de representar una textura y realizar su análisis correspondiente, cabe mencionar que esos métodos no se deben ocupar para lo mismo, si no dependiendo de la aplicación que le quiera dar uno al análisis de texturas.

Los métodos estadísticos para analizar texturas como la Transformada de Cúmulos Coordinados nos abren un mundo de posibilidades para poder entender mejor las texturas ya sean hechas computacionalmente, imágenes reales y/o preprocesadas como es el caso del álbum de Brodatz, este método realizar un análisis sobre una imagen binarizada y gracias a la binarización que se realiza previamente se logra conservar las propiedades de la textura a estudiar por lo que el vector de ocurrencias logra conservar la mayor cantidad de información que le pueda ser útil para el procedimiento a seguir, cabe mencionar que la relación tamaño de la ventana vector de ocurrencias es directamente proporcional por lo que se debe considerar un tamaño pequeño de la ventana para que no tengamos un vector grande y tengamos muchos valores que sean 0 y eso será desperdicio computacional, este método no es del todo invariante a rotación por lo que si se rota la imagen a un ángulo diferente a la posición inicial difícilmente podrá conservar toda la información que corresponde al patrón y se parezca más a otro.

El método de Otsu, ocupado para la binarización de las imágenes demuestra que puede ser un procedimiento que lograr separar el objeto del fondo o mostrar con más detalle el texton que es periódico, hablando específicamente del tema abordado que es la textura. Los métodos de clasificación permiten que el sistema pueda determinar que patrón corresponde de acuerdo a un vector estadístico,

a redes neuronales, o por otro método de clasificación que nos permita realizar dicha acción.

El método de clasificación propuesto en el presente trabajo fue la distancia mínima de Hamming y demostró ser un método sencillo pero capaz de realizar la clasificación hasta cierto punto adecuada, por que dependía de que tan cercano se parecía al patrón a comparar.

La detección a diferentes tipos de invarianzas (traslación, escala y rotación) es un procedimiento que en la mayoría de las investigaciones se ha puesto como meta, puesto que conservar las propiedades de una imagen (en este caso textura) sin importar los cambios que pueda tener la misma (traslación, escala y rotación) pueda determinar que es la misma textura sin importar esos cambios.

El trabajo se centró en una sola de las invarianzas que puede presentar una imagen y fue la de rotación, existen muchos métodos que proponen realizar una invarianza a rotación como lo fue Patrones binarios Locales Uniformes sin histogramas de Fourier y con histogramas de Fourier, filtro de Gabor y con descriptores de Fourier, y se puede enumerar más métodos.

Tomando como ejemplo los algoritmos antes mencionados la mejora que se propuso para la CCR fue la transformada discreta de Fourier y con el espectro se convertiría en el nuevo vector patrón de la textura el cual conserva las propiedades de la misma y no importando en qué dirección que encuentre podrá detectar adecuadamente la textura.

Se presentaron experimentos en los cuales se buscaba realizar una revisión de la eficiencia de la Transformada de Cúmulos Coordinados. El método de reconocimiento de patrones del experimento 1 fue discutido y ejecutado con éxito las imágenes creadas de manera computacional fueron hasta cierto punto eficaces aunque las que las que tenían patrones de valores pequeños tenían cierta dificultad de ser detectados a mayor cantidad de ruido y algo sorprendente fue

que el textón con el valor más grande generó una menor cantidad de aciertos como el textón 5767 que fue el que tuvo el mayor número de aciertos en las pruebas, al final, esto posiblemente ocurre porque al generar un ruido aleatorio tiene mayor probabilidad de poder mantener la estructura adecuadamente.

La transformada de cúmulos coordinados resulta ser una herramienta de lo más adecuada para poder ir guardando las caracterizar de las imágenes de texturas binarizadas. Como trabajo a futura se realizará una revisión del dicho algoritmo para hacerlo más eficaz.

En el experimento 2 el método del LBP y CCR, nos da ciertas características de la textura que bajo ciertas condiciones pueden ser favorables para nosotros, pero para otras perjudicaría el reconocimiento de dicha textura.

En el experimento 3 ha dado cuenta de las razones del uso generalizado de la CCR/EF1D. En esta investigación, la meta era evaluar la textura en diferentes direcciones, uno de los descubrimientos más significativos que surgen de este estudio es que de acuerdo al ángulo y el tipo de textura que se presenta al algoritmo podrá o no clasificarla. Los resultados de este estudio indican que la CCR compuesta con el Espectro de Fourier resulta una herramienta adecuada para mantener las características de las imágenes de texturas binarizadas. El estudio ha ido en cierta manera a mejorar nuestra comprensión de las texturas binarizadas, este no fue diseñado específicamente para evaluar el factor relacionado con la escala invariante. Además, se recomienda la investigación y experimentación en la CCR/EF1D. Estos hallazgos sugieren varios cursos de acción para los nuevos métodos de rotación invariante.

En el experimento 4 y 5 fue diseñado para determinar el efecto del uso de la textura real en ambos algoritmos con el fin de probar la eficiencia. Los resultados de este estudio indican que la LBP/UFH tiene tan buena eficiencia como la CCR / EF1D con diferencia para una textura en dirección diferente, los resultados

actuales parecen ser consistentes con otras investigaciones que encontraron que la LBPUGHF tiene una alta detección de textura correcta y las pruebas CCR / EF1D demuestran que es casi igual a la estudiada anteriormente método, sin embargo, los resultados del estudio actual no apoyan la investigación anterior porque el CCR / EF1D es un Nuevo método de rotación invariante.

5.1. Trabajo a futuro

Como trabajo a futuro se propone abordar lo siguiente:

- Trabajo de invarianza a contraste.
- Trabajo de detección de picos dominantes.
- Análisis del vector de ocurrencia eliminando cierta información para estudiar su comportamiento en la clasificación.

Apéndice A

Reconocimientos

Reconocimiento de participación del XVII Congreso Nacional de Ingeniería Electromecánica y de Sistemas 2018 Zacatenco, CDMX.



Reconocimiento de Participación del VI Congreso Internacional de Robótica y Computación 2019 La Paz, BCS.



Bibliografía

- [1] Jacob Scharcanski. Stochastic texture analysis for monitoring stochastic processes in industry. *Pattern Recognition Letters*, 26(11):1701–1709, 2005.
- [2] Antonio Javier Sanchez Santiago, Antonio Yuste Delgado, J E. Munoz Exposito, Sebastian Galan, Rocio Prado, Juan Maqueira Marin, and S Bruque Camara. Real-time image texture analysis in quality management using grid computing: An application to the mdf manufacturing industry. *International Journal of Advanced Manufacturing Technology - INT J ADV MANUF TECHNOL*, 58, 02 2012.
- [3] Mäenpää Topi, Ojala Timo, Pietikäinen Matti, and Soriano Maricor. Robust texture classification by subsets of local binary patterns. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 3, pages 935–938. IEEE, 2000.
- [4] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (7):971–987, 2002.
- [5] Weitao Li, Kezhi Mao, Hong Zhang, and Tianyou Chai. Selection of gabor filters for improved texture feature extraction. In *2010 IEEE International*

- Conference on Image Processing*, pages 361–364. IEEE, 2010.
- [6] Li Liu and Paul Fieguth. Texture classification from random features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):574–586, 2012.
- [7] Patrick Jackman and Da-Wen Sun. Recent advances in image processing using image texture features for food quality assessment. *Trends in food science & technology*, 29(1):35–43, 2013.
- [8] Yael Paran, Irena Lavelin, Suha NaffarâAbuâAmara, Sabina WinogradâKatz, Yuvalal Liron, Benjamin Geiger, and Zvi Kam. [13] - development and application of automatic highâresolution light microscopy for cellâbased screens. In James Inglese, editor, *Measuring Biological Responses with Automated Microscopy*, volume 414 of *Methods in Enzymology*, pages 228 – 247. Academic Press, 2006.
- [9] Richard A Cardullo. Fundamentals of image processing in light microscopy. In *Methods in cell biology*, volume 72, pages 217–242. Elsevier, 2003.
- [10] A Flook. Microscopy| image analysis. *IEEE*, 2003.
- [11] Eric C Anderson and Gary Chin. Flexible digital image processing via an image processing chain with modular image processors, December 5 2000. US Patent 6,157,394.
- [12] Hideyuki Tamura, Shunji Mori, and Takashi Yamawaki. Textural features corresponding to visual perception. *IEEE Transactions on Systems, man, and cybernetics*, 8(6):460–473, 1978.
- [13] Jack Sklansky. Image segmentation and feature extraction. *IEEE Transactions on Systems, Man, and Cybernetics*, 8(4):237–247, 1978.

- [14] Robert M Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, 1979.
- [15] Whitman Richards and Andres Polit. Texture matching. *Kybernetik*, 16(3):155–162, 1974.
- [16] Steven W Zucker. Toward a model of texture. *Computer Graphics and Image Processing*, 5(2):190–202, 1976.
- [17] J. K Hawkins. Textural properties for pattern recognition. *New York*, 1969.
- [18] Kirsi K Holli, Minna Wäljas, Lara Harrison, Suvi Liimatainen, Tiina Luukkaala, Pertti Ryymin, Hannu Eskola, Seppo Soimakallio, Juha Öhman, and Prasun Dastidar. Mild traumatic brain injury: tissue texture analysis correlated to neuropsychological and dti findings. *Academic radiology*, 17(9):1096–1102, 2010.
- [19] Mihran Tuceryan. *The Handbook of Pattern Recognition and Computer Vision (2nd Edition)*. 1998.
- [20] Xiaoying Guo, Chie Muraki Asano, Akira Asano, Takio Kurita, and Liang Li. Analysis of texture characteristics associated with visual complexity perception. *Optical review*, 19(5):306–314, 2012.
- [21] Dongxiao Zhou. *Texture analysis and synthesis using a generic Markov-Gibbs image model*. PhD thesis, ResearchSpace@ Auckland, 2006.
- [22] K.S. Fu. Syntactic pattern recognition and applications. *Prentice-Hall, New Jersey*, 1982.
- [23] Xiaoming Liu and Jinshan Tang. Mass classification in mammograms using selected geometry and texture features, and a new svm-based feature selection method. *IEEE Systems Journal*, 8(3):910–920, 2014.

- [24] Ralph Versteegen, Georgy Gimelâfarb, and Patricia Riddle. Texture modeling with nested high-order markov–gibbs random fields. *Computer Vision and Image Understanding*, 143:120–134, 2016.
- [25] André Ricardo Backes, Dalcimar Casanova, and Odemir Martinez Bruno. Color texture analysis based on fractal descriptors. *Pattern Recognition*, 45(5):1984–1992, 2012.
- [26] WT Freeman and Ce Liu. Markov random fields for super-resolution and texture synthesis. *Advances in Markov Random Fields for Vision and Image Processing*, 1(155-165):3, 2011.
- [27] Raul Enrique Sanchez Yanez. *Texture clasification using the Cordinated Cluster Representation of Image*. PhD thesis, Centro de Investigaciones en Optica, 2002.
- [28] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image processing, analysis, and machine vision*. Cengage Learning, 2014.
- [29] S Selvarajah and SR Kodituwakku. Analysis and comparison of texture features for content based image retrieval. *International Journal of Latest Trends in Computing*, 2(1), 2011.
- [30] Fernando Roberti De Siqueira, William Robson Schwartz, and Helio Pedrini. Multi-scale gray level co-occurrence matrices for texture description. *Neurocomputing*, 120:336–345, 2013.
- [31] Zhenhua Guo, Lei Zhang, and David Zhang. A completed modeling of local binary pattern operator for texture classification. *IEEE Transactions on Image Processing*, 19(6):1657–1663, 2010.

- [32] Timo Ojala and Matti Pietikäinen. Texture classification. *CVonline: The Evolving, Distributed, Non-Proprietary, On-Line Compendium of Computer Vision*, 2001.
- [33] EV Kurmyshev and M Cervantes. A quasi-statistical approach to digital binary image representation. *Revista Mexicana de Física*, 42(1):104–116, 1996.
- [34] EV Kurmyshev and R Soto. Digital pattern recognition in the coordinated cluster representation. *Nordic Signal Processing Symposium, Espoo, Finland*, pages 463–466, 1996.
- [35] Baochang Zhang, Yongsheng Gao, Sanqiang Zhao, and Jianzhuang Liu. Local derivative pattern versus local binary pattern: face recognition with high-order local pattern descriptor. *IEEE transactions on image processing*, 19(2):533–544, 2010.
- [36] Loris Nanni, Alessandra Lumini, and Sheryl Brahnam. Local binary patterns variants as texture descriptors for medical image analysis. *Artificial intelligence in medicine*, 49(2):117–125, 2010.
- [37] Xueming Qian, Xian-Sheng Hua, Ping Chen, and Liangjun Ke. Plbp: An effective local binary patterns texture descriptor with pyramid representation. *Pattern Recognition*, 44(10-11):2502–2515, 2011.
- [38] Abdolhossein Fathi and Ahmad Reza Naghsh-Nilchi. Noise tolerant local binary pattern operator for efficient texture analysis. *Pattern Recognition Letters*, 33(9):1093–1100, 2012.
- [39] Gustaf Kylberg and Ida-Maria Sintorn. Evaluation of noise robustness for local binary pattern descriptors in texture classification. *EURASIP Journal on Image and Video Processing*, 2013(1):17, 2013.

- [40] Sheryl Brahmam, Lakhmi C Jain, Loris Nanni, Alessandra Lumini, et al. *Local binary patterns: new variants and applications*. Springer, 2014.
- [41] Kurmyshev EV. Is the coordinated clusters representation an analog of the local binary pattern? *Computación y Sistemas*, 14(1):53–60, 2010.
- [42] Raúl E Sánchez-Yáñez, Evguenii V Kurmyshev, and Francisco J Cuevas. A framework for texture classification using the coordinated clusters representation. *Pattern Recognition Letters*, 24(1-3):21–31, 2003.
- [43] Phil Brodatz. *Textures: a photographic album for artists and designers*. Dover Pubns, 1966.
- [44] Gonzalo Pajares, Alegre Enrique, and De la Escalera Arturo. *Conceptos y Metodos de vision por computadora*. Grupo de Vision del Comite Espanol de Automatica (CEA), 2016.
- [45] Liu Jianzhuang, Li Wenqing, and Tian Yupeng. Automatic thresholding of gray-level pictures using two-dimension otsu method. In *Circuits and Systems, 1991. Conference Proceedings, China., 1991 International Conference on*, pages 325–327. IEEE, 1991.
- [46] Mohammad Norouzi, David J Fleet, and Ruslan R Salakhutdinov. Hamming distance metric learning. In *Advances in neural information processing systems*, pages 1061–1069, 2012.
- [47] Lei Zhang, Yongdong Zhang, Jinhu Tang, Ke Lu, and Qi Tian. Binary code ranking with weighted hamming distance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1586–1593, 2013.
- [48] Jan Flusser, Barbara Zitova, and Tomas Suk. *Moments and moment invariants in pattern recognition*. John Wiley & Sons, 2009.

- [49] A Amanatiadis, VG Kaburlasos, A Gasteratos, and SE Papadakis. Evaluation of shape descriptors for shape-based image retrieval. *IET Image Processing*, 5(5):493–499, 2011.
- [50] Xiang-Yang Wang, Yong-Jian Yu, and Hong-Ying Yang. An effective image retrieval scheme using color, texture and shape features. *Computer Standards & Interfaces*, 33(1):59–68, 2011.
- [51] Michał Bereta, Paweł Karczmarek, Witold Pedrycz, and Marek Reformat. Local descriptors in application to the aging problem in face recognition. *Pattern Recognition*, 46(10):2634–2646, 2013.
- [52] Bessem Samet, Calogero Vetro, et al. Coupled fixed point, f-invariant set and fixed point of n-order. *Ann. Funct. Anal*, 1(2):46–56, 2010.
- [53] Tony Lindeberg. Scale invariant feature transform. 2012.
- [54] Michael M Leung and Allen M Peterson. Scale and rotation invariant texture classification. In *[1992] Conference Record of the Twenty-Sixth Asilomar Conference on Signals, Systems & Computers*, pages 461–465. IEEE, 1992.
- [55] Yang Zhao, De-Shuang Huang, and Wei Jia. Completed local binary count for rotation invariant texture classification. *IEEE transactions on image processing*, 21(10):4492–4497, 2012.
- [56] Jacques Duvernoy. Optical–digital processing of directional terrain textures invariant under translation, rotation, and change of scale. *Appl. Opt.*, 23(6):828–837, Mar 1984.
- [57] RK Goyal, Wee Leng Goh, Dinesh P Mital, and Kap Luk Chan. Scale and rotation invariant texture analysis based on structural property.

- In *Proceedings of IECON'95-21st Annual Conference on IEEE Industrial Electronics*, volume 2, pages 1290–1294. IEEE, 1995.
- [58] George Eichmann and BZ Dong. Coherent optical production of the hough transform. *Applied Optics*, 22(6):830–834, 1983.
- [59] Rangasami L Kashyap and Alireza Khotanzad. A model-based method for rotation invariant texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (4):472–481, 1986.
- [60] Fernand S. Cohen, Zhigang Fan, and Maqbool A Patel. Classification of rotated and scaled textured images using gaussian markov random field models. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (2):192–202, 1991.
- [61] Amlan Kundu, Yang He, and Mou-Yen Chen. Alternatives to variable duration hmm in handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1275–1280, 1998.
- [62] R Porter and N Canagarajah. Robust rotation-invariant texture classification: wavelet, gabor filter and gmrf based schemes. *IEE Proceedings-Vision, Image and Signal Processing*, 144(3):180–188, 1997.
- [63] Guoying Zhao, Timo Ahonen, Jiří Matas, and Matti Pietikainen. Rotation-invariant image and video description with local binary pattern features. *IEEE Transactions on Image Processing*, 21(4):1465–1477, 2012.
- [64] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (12):2037–2041, 2006.

- [65] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59, 1996.
- [66] Takashi Matsuyama, Shu-Ichi Miura, and Makoto Nagao. Structural analysis of natural textures by fourier transformation. *Computer vision, graphics, and image processing*, 24(3):347–362, 1983.
- [67] João Batista Florindo and Odemir Martinez Bruno. Fractal descriptors based on fourier spectrum applied to texture analysis. *Physica A: statistical Mechanics and its Applications*, 391(20):4909–4922, 2012.
- [68] Robert Azencott, Jia-Ping Wang, and Laurent Younes. Texture classification using windowed fourier filters. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (2):148–153, 1997.
- [69] Li Ma, Tieniu Tan, Yunhong Wang, and Dexin Zhang. Personal identification based on iris texture analysis. *IEEE transactions on pattern analysis and machine intelligence*, 25(12):1519–1533, 2003.
- [70] Alan V Oppenheim. *Discrete-time signal processing*. Pearson Education India, 2002.
- [71] Hamid A Jalab. Image retrieval system based on color layout descriptor and gabor filters. In *2011 IEEE Conference on Open Systems*, pages 32–36. IEEE, 2011.
- [72] Kouros Jafari-Khouzani and Hamid Soltanian-Zadeh. Radon transform orientation estimation for rotation invariant texture analysis. *IEEE transactions on pattern analysis and machine intelligence*, 27(6):1004–1008, 2005.

- [73] Herrera Axel and Cuevas Francisco J. Clasificación de texturas por el método de la transformada de cumulos coordinados. *XVII Congreso Nacional De Ingeniería Electromecánica Y De Sistemas (CNIES 2018)*, 2018.
- [74] Herrera Axel and Cuevas Francisco J. Reconocimiento invariante a rotación de texturas mediante la transformada de cumulos coordinados. *Robotica y Computación. Investigación y Desarrollo.*, 2019.