



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS
DE LA COMPUTACIÓN

ANÁLISIS DE TEXTURA EN IMÁGENES DIGITALES POR MEDIO DE OPERADORES INVARIANTES

Tesis para obtener el título de:

MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

Presenta

SERGIO MANUEL DORANTES LÓPEZ

Asesor

DR. IVO HUMBERTO PINEDA TORRES

Co-asesor

M.C. JOSÉ DE JESÚS LAVALLE MARTÍNEZ

Puebla, Pue
Mayo, 2010



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS
DE LA COMPUTACIÓN

ANÁLISIS DE TEXTURA EN IMÁGENES DIGITALES POR MEDIO DE OPERADORES INVARIANTES

Tesis para obtener el título de:

MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

Presenta

SERGIO MANUEL DORANTES LÓPEZ

Asesor

DR. IVO HUMBERTO PINEDA TORRES

Co-asesor

M.C. JOSÉ DE JESÚS LAVALLE MARTÍNEZ

Puebla, Pue
Mayo, 2010

AGRADECIMIENTOS

Siempre es difícil resumir los agradecimientos en una hoja, sobre todo cuando se involucra a tantas personas en el logro de un objetivo como es la publicación de una tesis.

Quiero agradecer la paciencia de todos y cada uno de mis maestros, a los que se esforzaron por ayudarme y a los que no dejaron que renunciara en los momentos más complicados. Nombrarlos a todos sería difícil y se corre el riesgo de dejar alguno fuera, es por ello que en general doy gracias a cada uno de ellos.

Si quiero tomarme el tiempo de agradecer al Dr. Ivo Pineda por las oportunidades que me dio durante todo mi proceso de formación, especialmente durante la elaboración de esta tesis y la consecución de mi objetivo primordial. Le debo cada uno de los días que estuve involucrado en el proyecto.

Agradezco al Dr. Manuel Martín por la asesoría, comentarios y sugerencias durante el desarrollo de mi trabajo, además de ser un ejemplo constante de dedicación y conocimiento pleno. Bajo su tutela aprendí a no renunciar y no darme por vencido.

También agradezco al personal de la Facultad de Ciencias de la Computación, al personal de la biblioteca y en general de la Universidad por ofrecerme apoyo diario e incondicional, a todas las personas que estuvieron cerca de mí, a todos ellos, gracias.

Quiero agradecer a las personas que compartieron mi vida en Puebla, a los que viví como amigos y a los que sentí como hermanos, gracias a Jaime Lara por la paciencia mostrada durante todo este tiempo, por la ayuda en la escuela y por el apoyo fuera de ella, compartir un techo conmigo y aún así sobrevivir a la experiencia, gracias.

Pero sobre todo gracias a ti, por compartir parte de tu vida conmigo, por ser mi guía en el sinuoso camino, por darme la oportunidad de vivir y de sentir aún cuando me encontraba lejos de casa, por no dejarme gobernar por la nostalgia, por presionarme por hacer lo que sentía y por terminar mi tesis, y gracias por aceptarme como soy, lo que es más, gracias por quererme de la manera en que lo hiciste.

A todos y cada uno de ustedes, gracias... totales.

Resumen

En los últimos años la visión por computadora ha experimentado un crecimiento considerable debido a los avances en el poder de cómputo de los equipos actuales, los estudios que en otros años solo eran considerados teóricos ahora pueden ser llevados a la práctica gracias a los medios con los que se cuenta. Este trabajo toma la técnica de detección de texturas basado en los filtros de Gabor, propuesto por Denis Gabor en 1946 como parte de la teoría de la comunicación. Dicha teoría fue extendida más tarde por John Daugman en 1980 quien demostró que los filtros de Gabor en dos dimensiones son óptimos mostrando la resolución máxima posible de información tanto en el dominio espacial como en el dominio de la frecuencia y que sirve como fundamento para su utilización dentro de la visión por computadora, específicamente en la detección de texturas.

El problema específico dentro de la detección de texturas es su segmentación y su clasificación. Tareas en las cuales han existido distintos estudios buscando una solución que no requiera una gran cantidad de recursos de cómputo y sin sacrificar la calidad de la segmentación.

Lo que proponemos en esta tesis es la utilización de dos técnicas separadas y complementarias, por un lado extraer las características de textura de un conjunto de imágenes y posteriormente realizar la clasificación por medio de las Máquinas de Vector de Soporte.

El capítulo uno sienta las bases de lo que es una textura y de los diferentes enfoques que han existido para su manejo, se define de manera específica lo que se considera una textura y sus propiedades, así también se mencionan investigaciones relacionadas con el tema tanto del uso de los filtros de Gabor como el uso de las Máquinas de Vector de Soporte. Y por último se menciona de manera general el tema de dominio en frecuencia el cual es básico para el entendimiento de los filtros de Gabor.

En el capítulo dos se menciona el trabajo realizado por Dennis Gabor en el área de estudio de señales y la manera en que eventualmente fue creciendo y desarrollando para la detección de texturas. Desde los primeros estudios realizados en señales unidimensionales hasta la generalización hecha por Daugman para señales bidimensionales. También cubrimos aspectos fundamentales de las Máquinas de Vector de Soporte, la fortaleza intrínseca que mantiene como clasificador, y la manera en que modifica el espacio de datos para volverlo clasificable.

El capítulo tres establece las bases del proceso de clasificación de imágenes, el método base para modelar el sistema de clasificación es la extracción de características. Se mencionan los métodos de clasificación más utilizados en publicaciones, y se realiza una descripción general del método de clasificación propuesto. Una parte importante del capítulo muestra el proceso de preparación de los experimentos, comenzando con el pre-procesamiento de las imágenes de

entrada para que sean adaptadas de acuerdo a las necesidades de los filtros, y de manera más representativa, el procedimiento para determinar los valores de los parámetros de los filtros. Por último se describe la manera en que los datos filtrados son distribuidos para servir como entrada al clasificador SVM.

En el capítulo cuatro se ahonda un poco más en la definición de las Máquinas de Vector de Soporte, los distintos *kernels* que pueden manejarse y las características y parámetros de cada uno de ellos. Se preparan los experimentos para poder seleccionar el *kernel* adecuado de acuerdo al conjunto de datos que buscamos separar y en base a los resultados obtenidos se definen los valores de parámetros para los filtros de Gabor.

El capítulo cinco, último de esta tesis, sirve para la presentación de resultados de clasificación obtenidos, se muestran algunas gráficas comparativas de desempeño del clasificador así como el porcentaje de error de clasificación obtenidos.

ÍNDICE

Índice de Figuras	1
Índice de Tablas	2
Introducción.....	3
Capítulo I. Textura y Dominio en Frecuencia	
1.1 Textura	7
1.2 Trabajo Relacionado.....	11
1.3 Dominio en Frecuencia	13
Capítulo II. Filtros de Gabor	
2.1 Filtros de Gabor	17
2.1.1 Formulaciones de Filtros de Gabor	21
2.2 Diseño de Filtros versus Bancos de Filtros	22
2.3 Sistema Visual Humano	24
2.4 Máquinas de Vector de Soporte	25
2.4.1 El Margen Blando	28
2.4.2 La Función <i>Kernel</i>	29
Capítulo III. Implementación del Método	
3.1 Clasificación.....	35
3.1.1 Extracción de Características	35
3.1.2 Clasificación de Imágenes	37
3.2 Evaluación de la Clasificación	38
3.3 Método Propuesto	39
3.4 Preparación de Experimentos	40
3.4.1 Imágenes de Entrada	41
3.4.2 Pre-procesamiento y Filtrado de Imágenes	42
3.4.2.1 Etapa de Pre-procesamiento	42
3.4.2.2 Utilización de Filtros y Filtros de Gabor	43
3.4.2.3 Familia de Filtros de Gabor y Porque Dominio en Frecuencia	47
3.4.2.4 Parámetros de los Filtros y Convolución con las Imágenes.....	50
3.4.2.5 Conformar la Matriz de Características.....	51
Capítulo IV. Experimentos con el Clasificador SVM	
4.1 Selección de Datos de Entrenamiento y Prueba.....	55
4.2 Experimentación SVM	56

4.2.1 <i>Kernel</i> /Lineal	56
4.2.2 <i>Kernel</i> /Polinomial.....	56
4.2.3 <i>Kernel</i> /de Función Radial.....	57
4.3 Selección del <i>Kernel</i>	58
4.4 Valores de Sigma para el Clasificador SVM y Frecuencia Central	64
Capítulo V. Resultados	
5.1 Frecuencia Central	65
5.2 Errores de Clasificación.....	67
5.2.1 Set S1 Cañones Rocosos	67
5.2.2 Set S2 Playa Tropical	68
5.2.3 Set S3 Línea Costera.....	68
5.2.4 Set S4 Ciudad	69
5.2.5 Set S5 Formaciones Rocosas	70
5.2.6 Set S6 Montañas Nevadas	71
5.3 Tiempo de Procesamiento.....	72
Conclusiones	73
Referencias	74

ÍNDICE DE FIGURAS

- Figura 1.1 Cuatro imágenes de textura, su clasificación y segmentación.
- Figura 2.1 Representación de un filtro de Gabor con los cuatro parámetros que lo definen.
- Figura 2.2 Parámetros de un grupo de filtros de Gabor con 9 orientaciones.
- Figura 2.3. Conjunto de hiperplano de separación óptima en R^2 .
- Figura 2.4 Ejemplo de hiperplano de separación en donde existen datos no separables.
- Figura 2.5 Elemento considerado *outlier* dentro del conjunto de datos.
- Figura 2.6 *Outlier* aceptado después de trazar el hiperplano de separación.
- Figura 2.7 Transformación del plano al aplicar el *kernel*.
- Figura 2.8 Datos de muestra que no son separables linealmente.
- Figura 2.9 Hiperplano de separación después de la transformación.
- Figura 2.10 Generación de múltiples hiperplanos con *kernel* de alta dimensión.
- Figura 3.1 Proceso de detección de bordes.
- Figura 3.2 Identificación de cuatro etiquetas de los datos pertenecientes a dos conjuntos.
- Figura 3.3 Representación de la vecindad de un pixel.
- Figura 3.4 Máscara de un filtro con *kernel* de 3x3.
- Figura 3.5 Representación de una familia de filtros de Gabor en donde las orillas están en contacto unas con otras.
- Figura 3.6 Representación de una familia de filtros de Gabor con superposición de orillas.
- Figura 5.1 Comparativo de porcentaje de errores obtenidos en fase de experimentación del conjunto Cañones Rocosos.
- Figura 5.2 Comparativo de porcentaje de errores obtenidos en fase de experimentación del conjunto Playa Tropical.
- Figura 5.3 Comparativo de porcentaje de errores obtenidos en fase de experimentación del conjunto Línea Costera.
- Figura 5.4 Comparativo de porcentaje de errores obtenidos en fase de experimentación del conjunto Ciudad.
- Figura 5.5 Comparativo de porcentaje de errores obtenidos en fase de experimentación del conjunto Formaciones Rocosas.

Figura 5.6 Comparativo de porcentaje de errores obtenidos en fase de experimentación del conjunto Montañas Nevadas.

Figura 5.7 Comparativo de velocidades de procesamiento al ejecutar tres optimizadores implementados en SPIDER

Índice de Tablas

Tabla 3.1 Lista de parámetros de filtro de Gabor.

Tabla 3.2 Valores de parámetros tomados por algunos autores de acuerdo a distintas publicaciones.

Tabla 4.1 Listado de *Kernel* soportados por *SPIDER* con su respectivo parámetro.

Tabla 5.1 Conjuntos de imágenes con su respectiva frecuencia central.

INTRODUCCIÓN

El propósito de este trabajo de tesis es diseñar e implementar un método de detección de texturas basado en la extracción de características por medio de filtros de Gabor y el uso de Máquinas de Vector de Soporte con el objeto de encontrar la máxima separación entre texturas diferentes y poder definir de manera correcta las características de cada región que sirva como base para la recuperación y consulta de imágenes. Se desea explorar las ventajas que nos puede ofrecer el análisis de la detección de texturas utilizando como base los filtros de Gabor, apoyados por la funcionalidad de clasificación que ofrecen las Máquinas de Vector de Soporte. Los filtros de Gabor serán utilizados para extraer las características de textura de cada una de las imágenes analizadas, la máquina de vector de soporte será la encargada de definir el área máxima de separación entre dos texturas. Este esquema dual ofrece una alternativa de implementación contra los modelos de reconocimiento de texturas que actualmente existen.

El reconocimiento de texturas ha sido un tema muy explorado en el área de visión por computadora, lo cual se debe en gran medida a que la mayoría de las superficies presentes en las imágenes contienen algún tipo de textura. Los estudios concernientes a la presencia de texturas se han hecho en una amplia gama de imágenes, desde imágenes con áreas naturales hasta imágenes médicas, pasando por la manufactura de productos industriales. Además se ha tenido un crecimiento considerable debido a los avances en el poder de cómputo de los equipos actuales, los estudios que en otros tiempos sólo eran considerados teóricos ahora pueden ser llevados a la práctica gracias a los medios con los que se cuenta.

El estudio presentado en la tesis se enfoca en problemas específicos de la detección de texturas, así como en texturas particulares, pero los resultados son suficientemente representativos como para extenderlos a otros dominios de aplicación. El aporte de este trabajo está fundamentado en la utilización de una herramienta que nace de la Teoría de Aprendizaje Estadístico como son las Máquinas de Vector de Soporte, con ellas se busca lograr una máxima separación entre texturas tomando como datos de entrada el vector de características proporcionado por los filtros de Gabor al ser aplicados en una imagen. Los resultados obtenidos de este trabajo pueden servir como base para otras disertaciones y publicaciones en el área. Esta investigación puede tener aplicaciones prácticas en áreas como desarrollo urbano, estudios de impacto ambiental, estudios de impacto ecológico y, por supuesto, en visión por computadora.

De manera general el trabajo está dividido en tres etapas, la primera consiste en la obtención de los vectores de características de imágenes que contengan la textura a estudiar después de haber aplicado los filtros de Gabor, los filtros serán aplicados de acuerdo al tipo de textura que se desea examinar. En segundo lugar, se toman los datos obtenidos y se le proporcionan a la Máquina de

Vector de Soporte como entrenamiento. Por último, una vez que se haya hecho el entrenamiento se proporcionará a la Máquina de Vector de Soporte una imagen completa que contenga alguna de las texturas, ya sea conocida o desconocida, para las cuales fue entrenada y se comprobará su funcionalidad.

Motivación.

La capacidad natural que tiene el ser humano de distinguir entre diversas texturas es una acción muy difícil de reproducir por un sistema computacional debido a la cantidad de propiedades existentes en la imagen que deben tomarse en cuenta, como son: la variedad de los patrones presentes, la frecuencia de aparición, la dirección de dichos patrones, las condiciones de iluminación, etc. Tradicionalmente se ha buscado que los algoritmos de procesamiento de imágenes sean capaces de emular la manera en que el sistema visual humano lleva a cabo la identificación de texturas.

Dentro del área de visión por computadora y el procesamiento digital de imágenes es de gran importancia el estudio de las texturas. Las tareas que deben efectuar para el reconocimiento de texturas emulan la manera en que el sistema visual humano reacciona. Lo que es más, la complejidad que manifiestan las texturas requiere del uso de una herramienta de estudio que cumpla cabalmente con las necesidades inherentes al tema.

El análisis de texturas se ha enfocado básicamente en dos actividades: en primer lugar segmentación de texturas, y en segundo lugar la clasificación de las texturas, siendo la primera tarea objeto de mayor escrutinio por parte de los investigadores por medio de la publicación de diversos artículos. A pesar de que los primeros estudios en análisis de texturas se llevaron a cabo desde la década de 1970, es hasta principio de los 90 que su desarrollo se acrecienta gracias a las publicaciones hechas por investigadores como Bovik, Weldon, Farrokhnia, Dunn, Jain y otros, con propuestas de investigación que aprovechan las propiedades descubiertas que relacionan de manera intrínseca las funciones modeladas en espacio y en frecuencia con el funcionamiento de las células de la corteza visual.

Dentro de los métodos de reconocimiento de texturas existen dos vertientes muy bien identificadas, por un lado tenemos los algoritmos de segmentación supervisada y por otro lado la segmentación no supervisada. La diferencia radica en que en la segmentación supervisada se asume el conocimiento previo de las características presentes en la textura, es decir, es necesario que el algoritmo sea diseñado para el análisis de texturas muy bien definidas y reconocidas. Mientras que la segmentación no supervisada no requiere de conocer previamente el tipo de textura a la que será enfrentada, aunque si es necesario conocer el número de texturas que tendrá que reconocer [32]. Si la tarea de segmentación requiere identificar una textura previamente establecida, se puede utilizar el primer enfoque mencionado, cuando se va a analizar una gran cantidad de imágenes de las cuales no se tiene pleno conocimiento de las texturas que contiene es recomendable utilizar el segundo enfoque.

La selección de algún método sobre otro radica en el tipo de funcionalidad que se desee obtener, en [33] se muestran resultados de la implementación de algoritmos supervisados y no supervisados, e incluso se incluye un tercero, semi-supervisado; los resultados presentados permiten concluir que el método supervisado ofrece una mayor precisión en la segmentación de las texturas, de acuerdo a las imágenes que ahí se analizaron. Los métodos no supervisados son utilizados en grupos muy grandes de imágenes, o con una gran cantidad de texturas presentes, ya que al no ser necesario conocer el tipo de textura que se tenga le otorga la ventaja de procesar de manera rápida una mayor cantidad de texturas, aunque esta propiedad le resta precisión. Por el contrario los métodos supervisados requieren que se tenga información *a priori* del tipo de texturas presentes en la imagen, aunque esto le resta rapidez pueden ser, en la práctica, más precisos.

Debido a que el objeto de nuestro estudio es encontrar la máxima separación entre texturas optaremos por seleccionar el método supervisado el cual nos pudiera ofrecer características de precisión más efectivas.

Una vez definido el método de segmentación seleccionado hay que identificar las herramientas necesarias para llevar a cabo dicha segmentación. Debemos tomar en cuenta que una particularidad de las texturas es que pueden ser modeladas como patrones dominados por una banda estrecha de frecuencias y orientaciones espaciales [16].

CAPITULO I. TEXTURA Y DOMINIO EN FRECUENCIA

1.1 Textura.

En muchos sistemas de visión por computadora y de procesamiento de imágenes se toma por entendido que los objetos presentes en las imágenes tienen una superficie uniforme, lo cual no puede aceptarse de forma generalizada, ya que la mayoría de los objetos de la vida real no presentan regiones de intensidad uniformes. Tomemos como ejemplo una superficie de madera, en ella existen algunos patrones repetitivos y que tienen cierta dirección. Estos patrones pueden deberse a la naturaleza física de la superficie, como su rugosidad, como en el caso de la madera, o pueden deberse a diferencias de reflectancia como puede ser el cambio de colores en una misma superficie. Esas pequeñas variaciones en la intensidad de una región dentro de una imagen es lo que los seres humanos identificamos como textura. Como se menciona en [11] las texturas son utilizadas por el ojo humano para separar distintos objetos dentro de escenarios visuales, así como para el análisis de superficies.

Aunque somos capaces de reconocer fácilmente una textura, es muy difícil encontrar una definición estándar de la misma. Esta dificultad está demostrada por la cantidad de definiciones que se han hecho en el medio de la visión por computadora. En [14] se distingue a las texturas como una medida eficiente de estimar las diferencias estructurales, de orientación, rugosidad, suavidad, o de regularidad que existen entre diversas regiones de una imagen. Se presenta una recopilación de definiciones que han aparecido en alguna de la literatura de visión por computadora, presentada en [34]:

- “podemos considerar una textura como lo que constituye una región macroscópica. Su estructura es simplemente atribuida a los patrones repetitivos en los cuales los elementos o primitivos son organizados de acuerdo a una regla de distribución”
- “una región en una imagen tiene una textura constante si un conjunto de estadísticas locales u otras propiedades locales de la función de la imagen son constantes, variando lentamente, o aproximadamente periódicas.”
- “la textura de la imagen que consideramos es no-figurativa y celular... Una textura de imagen está descrita por el número y tipo de sus primitivos (tonales) y la organización espacial o diseño de sus primitivos (tonales)... Una característica fundamental de textura: no puede ser analizada sin un marco de referencia de primitivos tonales ya sea implícitos o previamente declarados. Para cualquier superficie suave de tonos grises, existe una escala tal que cuando la superficie es examinada, no tiene textura. Por lo tanto conforme la resolución aumente, adquiere una textura fina y después de esto una textura gruesa.”

- “la textura es definida para nuestro propósito como un atributo de un campo que no contiene componentes que sean numerables. Las relaciones de fase entre los componentes son por lo tanto no aparentes. Tampoco en el caso en que el campo contiene un gradiente obvio. El intento de esta definición requiere atención directa del observador de las propiedades globales de la visualización - por ejemplo su <<grosor>>, o <<fineza>>. Físicamente, los patrones (no-periódicos) no numerables son generados por procesos estocásticos opuestos a los determinísticos. De cualquier forma, de manera perceptiva, el conjunto de todos los patrones sin componentes obvios numerables incluirá muchas texturas determinísticas (y hasta periódicas)”.

- “textura es aparentemente una noción paradójica. Por un lado, se usa comúnmente en procesamientos tempranos de información visual, especialmente para propósitos prácticos de clasificación. Por otro lado, nadie ha triunfado en producir una definición de textura que sea aceptada de manera general. La resolución de esta paradoja, creemos, dependerá de un modelo más rico y desarrollado para el procesamiento temprano de información visual, un aspecto central que será representativo en diferentes niveles de abstracción.”

- “La noción de textura parece depender de tres ingredientes: (i) algún ‘orden’ local se repite sobre una región la cual es larga en comparación con el tamaño del orden, (ii) el orden consiste en un arreglo no-aleatorio de partes elementales, y (iii) esas partes son entidades más o menos uniformes teniendo aproximadamente las mismas dimensiones en todos lados dentro de la región texturizada”

Tanto en [13] como en [34] se menciona que las “definiciones” de textura que se han formulado hasta el momento muestran la característica de que son dependientes del objetivo para el cual se está utilizando la textura. Algunos autores toman como motivación la sola percepción de lo que creen adecuado, y otros se dejan llevar completamente por el tipo de aplicación en la cual será usada la definición.

La unidad de medida de las texturas es el texel (*Texture Element*, elemento de textura, [12]), y la repetición de los texels determina el tipo de textura y decide el tipo de enfoque en análisis de texturas.

En [34] se indica que la textura, definida como una función de variación espacial de las intensidades de los píxeles (valores de grises), es útil dentro de una variedad de aplicaciones y ha sido sujeto de estudios por varios investigadores. Además divide el problema del reconocimiento de texturas en tres partes fundamentalmente: 1) clasificación de texturas, 2) segmentación de texturas y 3) síntesis de texturas. Una aplicación directa de la textura es el reconocimiento de regiones dentro de una imagen utilizando las propiedades de la textura. Por

ejemplo, en la figura 1.1(a) se pueden identificar cuatro tipos de textura como son Madera, Muro de Ladrillos, Lámina de Acero y Arbusto.

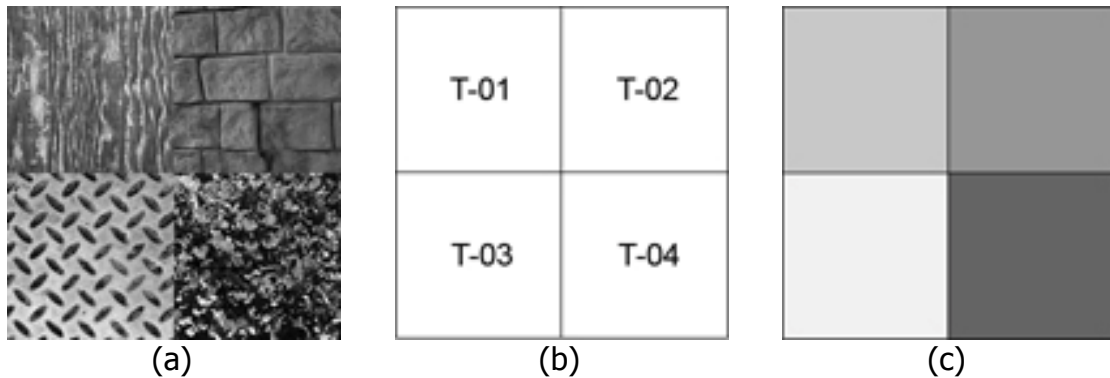


Figura 1.1. (a) Imagen con cuatro texturas presentes: Madera, Muro de Ladrillos, Lámina de Acero y Arbusto (en orden de las manecillas del reloj). (b) La clasificación de la imagen (a) con cada región identificada con una etiqueta. (c) La segmentación de la imagen (a) separando las texturas e identificando los límites que la dividen.

La textura es el elemento visual más importante para poder identificar este tipo de regiones homogéneas. Al hecho de identificar las regiones de una imagen de acuerdo a las texturas que contiene se le llama **clasificación** de texturas. El propósito de la clasificación de texturas es producir un mapa de clasificaciones de la imagen de entrada en donde cada región texturizada uniforme es identificada con la clase de textura a la que pertenece como se muestra en la figura 1.1(b). También es posible identificar las fronteras entre texturas aún si no es posible clasificarlas. Este es el segundo problema que el análisis de texturas trata de resolver, **segmentación** de texturas. La meta de la segmentación de texturas es obtener el mapa de límites entre texturas, como se muestra en la figura 1.1(c). Por último la síntesis de texturas es usada en aplicaciones de compresión de imágenes. También es importante en graficación por computadora en donde el objetivo es modelar superficies de objetos para que luzcan lo mas reales posibles.

Algunas de las aplicaciones en las que podemos utilizar el análisis de texturas incluyen inspección automática de la calidad de productos industriales, estimación de la orientación de un objeto, análisis de formas, procesamiento de documentos, imágenes satelitales, observación remota o tele-observación, procesamiento de imágenes médicas, y últimamente la recuperación de imágenes y video basado en contenido [13]. La idea central es disminuir la participación humana en las tareas antes mencionadas buscando eliminar el porcentaje de falla debido a su intervención.

La inspección de productos industriales se basa en la detección de defectos de los productos terminados, este enfoque ha sido utilizado primordialmente en la industria textil y en la industria metalúrgica. Es así como en [35] se presenta un artículo que investiga la detección de ondulaciones en los rieles de las vías ferroviarias por medio de imágenes tomadas de los rieles instalados. También en el

área de aplicaciones médicas el análisis de texturas ha jugado un rol importante. El papel primordial que desempeña el análisis de texturas es encontrar comportamientos anormales de los órganos, como puede ser la diferenciación entre tejido sano y tejido dañado como es el cáncer, o la presencia de algún elemento en el organismo que anteriormente no estaba presente como los tumores. Por otro lado el procesamiento de documentos tiene aplicaciones desde el reconocimiento de direcciones postales hasta el análisis e interpretación de mapas, también se utiliza para separar elementos de interés presentes en los periódicos y documentos impresos. Por último la observación remota se ha utilizado mucho para la clasificación de usos de suelo en donde se tienen regiones con distintos tipos de terrenos, como ríos, regiones urbanas, selva, etc., para llevar a cabo esta tarea pueden ser utilizadas imágenes aéreas o satelitales.

En [34] se enlistan algunos de los atributos que según el mismo autor conciernen a la textura y se aceptan como verdaderas:

- La textura es una propiedad que involucra áreas, la textura de un punto no existe y por lo tanto no está definida. Por lo tanto, la textura es una propiedad contextual y su definición debe involucrar niveles de grises en una especie de vecindario espacial. El tamaño de este vecindario depende ya sea del tipo de textura o del tamaño de los primitivos que definen la textura.
- La textura implica la distribución espacial de niveles de grises. Además, histogramas de dos dimensiones o matrices de co-ocurrencia son herramientas convenientes de análisis de texturas.
- La textura en una imagen puede ser percibida a diferentes escalas o niveles de resolución. Si tomamos como ejemplo la textura representada por un muro de ladrillos. A una resolución ordinaria la textura puede percibirse como un grupo de ladrillos conformando el muro, aquí los detalles interiores de cada ladrillo se pierden. Sin embargo a una resolución más grande, en donde solo puede observarse algunos ladrillos, la textura que puede percibirse son los detalles interiores del ladrillo.
- Se dice que una región tiene textura cuando el número de objetos primitivos en esa región es grande. Si se tienen pocos objetos primitivos, entonces se dice que tenemos un grupo de objetos numerables en lugar de una textura. En otras palabras una textura es percibida cuando no existen en la imagen "formas" individuales significativas.

Las texturas tienen un número de cualidades perceptibles que juegan un papel importante en su descripción. En [36] se identifica un grupo de propiedades que juegan un papel importante en la descripción de una textura: uniformidad,

densidad, grosor, rugosidad, regularidad, linealidad, direccionalidad, dirección, frecuencia y fase.

El estudio de las texturas se ha abordado desde distintas perspectivas, como son (i) métodos estadísticos, por medio de matrices de co-ocurrencia, características de auto-correlación, (ii) métodos geométricos, por medio del mosaico de características de Voronoi y métodos estructurales, (iii) métodos basados en modelos, como son los modelos de campo aleatorio, fractales, (iv) métodos de procesamiento de señales, por medio de los filtros de dominio espacial, filtrado en el dominio de Fourier, modelos de filtros y *wavelets* de Gabor. En este sentido no existe una manera de asegurar la superioridad de un método sobre otro, pero se ha visto que los métodos en los que se utiliza el enfoque de procesamiento de señales por medio de filtros regularmente producen resultados más precisos, probados sobre imágenes de varios tipos (aéreas, satelitales, paisajes) [11].

Uno de los métodos que han sido ampliamente desarrollados para el problema del análisis de texturas es la utilización de los filtros de Gabor, su utilización se ha convertido en una constante. Esto se debe a la funcionalidad que ofrecen estos filtros de detectar cambios en las texturas presentes en una imagen de modo similar a como lo hace la vista humana.

1.2 Trabajo Relacionado

En diversos artículos podemos encontrar métodos referentes al tema, abordados con distintas técnicas y obteniendo resultados diversos. Todos ellos guardan la característica de utilizar texturas artificiales para llevar a cabo sus experimentos.

En [9] se busca la clasificación de imágenes aéreas y satelitales de ciudades, áreas de cultivo y bosques, se centra en la caracterización de texturas presentes en dichas imágenes por medio de técnicas relacionadas con los filtros de Gabor, el objetivo final se refiere a la capacidad de obtener información temática de acuerdo a la textura contenida en imágenes pancromáticas. El método propuesto inicia con la definición de un Banco de Filtros, dicho banco se aplica a cada imagen de entrada y se obtiene la magnitud de las imágenes filtradas, se aplica un filtro pasa-bajo Gaussiano, para reducir la varianza por ende el error de clasificación. Las imágenes resultantes de este proceso de filtrado sirven como entrada a un proceso de clasificación supervisada utilizando la Estimación de Probabilidad Máxima. Las conclusiones del artículo mencionan que el uso de un banco completo de filtros de Gabor demuestra ser el método más efectivo. En términos generales los resultados de clasificación oscilan entre el 60-80% de precisión.

En [3] se propone un método de clasificación de imágenes usando el histograma de color de la imagen como su representación, y como método de clasificación las Máquinas de Vector de Soporte, las conclusiones refieren la posibilidad de disminuir el error de clasificación de imágenes tan bajo como 11%

para la clasificación de 14 categorías de la base de imágenes de Corel, y 16% para un conjunto mas general de objetos. Menciona que el buen desempeño obtenido en los resultados se debe primordialmente a la habilidad superior de generalización que tienen las Máquinas de Vector de Soporte en espacios de dimensionalidad superior.

En [4] se describe un método de clasificación de texturas realzando el potencial que tienen las Máquinas de Vector de Soporte en el área de clasificación de texturas. En lugar de utilizar un esquema de extracción de características externo para las imágenes, las SVM reciben los valores de niveles de grises de cada pixel de la imagen de entrada sin tratar. La base para dicha decisión es aprovechar la habilidad de las SVM de aprender en espacios de alta dimensionalidad, como lo son los espacios de patrones de niveles de grises, además puede incorporar un esquema de extracción de características dentro de su misma arquitectura, este método es exhaustivo computacionalmente debido al número de regiones que pueden definir una imagen.

Otro método desarrollado y presentado en [5] utiliza una modificación de las SVM conocida como GSVM (*Generalized Support Vector Machines*, Máquinas de Vector de Soporte Generalizadas). La base de su estudio se encuentra en la Recuperación de Imágenes Basada en Contenido (*Content Based Image Retrieval*, por sus siglas en inglés), principalmente en sistemas basados en regiones. La utilización de SVM convencional en sistemas de CBIR basados en regiones no puede hacerse de manera directa ya que viola los requerimientos que SVM plantea en el *kernel*. Las GSVM tienen la particularidad de utilizar un *kernel* arbitrario, y proponen utilizarlo en sistemas de CBIR que utilizan medidas de similitud arbitrarias. El objetivo que buscan es clasificar imágenes de acuerdo a regiones contenidas en las mismas. Se menciona que los resultados experimentales en imágenes de contenido general muestran la eficacia del método propuesto, sin embargo no presentan porcentajes de efectividad de dicho método.

En [6] la Máquina de Vector de Soporte se combina con las características de textura generadas por la *Discrete Wavelet Frame Transform* (DWFT, por sus siglas en inglés), para llevar a cabo la clasificación de texturas. La selección del parámetro del *kernel* se lleva a cabo por medio de un sistema de votación efectuado entre varias SVM's. Los experimentos se realizan comparando texturas de álbum Brodatz, y al compararlo con sistemas de clasificación clásicos como la Distancia de Bayes, Distancia de Mahalanobis y *Linear Vector Quantization* ofrece mejores resultados de clasificación, llegando hasta el 96% de precisión en la clasificación, por encima de los demás métodos.

En [7] proponen la clasificación de texturas combinando el uso de una variación de la Transformada *Wavelet* como extractor de características y las SVM. La variación, conocida como la Transformada *Wavelet* de Estructura Piramidal, la cual utilizan para descomponer las imágenes de entrada, con lo cual conforman un vector, y SVM como método de clasificación. La manera en que se implementa SVM tiene que ver con su naturaleza binaria, ya que es necesario que el

clasificador sea multiclase, por lo que diseñaron una Red Neuronal llamada *Feedforward Neural Network* (FNN) en donde se pusieron en cascada varias SVM. Los resultados que presentan demuestran que a una resolución de 32 X 32 pixeles, se tiene una efectividad de clasificación de hasta 93%, y con una resolución de 64 X 64 pixeles, una efectividad de 98%.

De los diversos métodos descritos podemos ver que comparten la utilización de algún método externo para llevar a cabo la extracción de las características más importantes de las texturas, ya sea de manera individual o presentes en alguna imagen completa. Y que los resultados obtenidos en este primer paso, sirven de entrada para algún tipo de clasificador. De acuerdo a las propiedades que tienen las SVM, se han convertido es un clasificador muy poderoso y utilizado por diversos investigadores.

El método que proponemos en esta tesis consiste en la utilización de los Filtros de Gabor como método de extracción de características de una imagen, y posteriormente utilizar una SVM para conseguir la clasificación de los datos proporcionados por el filtro aplicado a cada imagen.

1.3 Dominio en Frecuencia.

Para entender la naturaleza de los filtros y la manera en que éstos actúan sobre las imágenes, es necesario conocer un par de conceptos que han ayudado en el procesamiento digital de imágenes, éstos son la transformada de Fourier y el concepto de dominio en frecuencia.

La contribución del francés Joseph Fourier al procesamiento de señales plantea que cualquier función que se repite de manera periódica puede ser expresada como la suma de senos y/o cosenos de distintas frecuencias, cada una multiplicada por un coeficiente distinto (ahora llamamos a esto una Serie de Fourier) [37]. No importa lo complicada que puede ser la función, mientras que sea periódica y cumpla con algunas condiciones matemáticas, puede ser representada por dicha suma.

Incluso las funciones que no son periódicas (pero cuya área bajo la curva es finita) pueden ser expresada como la integral de senos y/o cosenos multiplicados por una función ponderada. En este caso se le conoce a la formulación como la Transformada de Fourier, y su utilidad es aún mayor que la Serie de Fourier para aplicaciones prácticas. Ambas representaciones comparten la característica de que una función, ya sea expresada en Serie de Fourier o en Transformada de Fourier, puede ser reconstruida completamente a través de un proceso inverso, sin pérdida de información. Esta es una de sus características más importantes ya que nos permite trabajar con una señal en el "dominio de Fourier" y luego regresar al dominio original de la función sin pérdida de información.

La transformada de Fourier de una función discreta de una variable $f(x)$, con $x = 0, 1, 2, \dots, M - 1$ esta dada por la ecuación 1.1.

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) e^{-j2\pi ux/M} \quad \text{para } u=0,1,2,\dots,M-1 \quad (1.1)$$

De manera similar, dada $F(u)$, se puede obtener la función original usando el inverso de la Transformada de Fourier Discreta, de la forma:

$$f(x) = \sum_{u=0}^{M-1} F(u) e^{j2\pi ux/M} \quad \text{para } x=0,1,2,\dots,M-1 \quad (1.2)$$

Para poder calcular el valor de $F(u)$ en la ecuación 1.1 se comienza por sustituir el valor $u = 0$ en el término exponencial y después sumar todos los valores que puedan tenerse de x . Después se sustituye el valor $u = 1$ en el exponencial y se repite la suma para todos los valores de x , éste proceso se repite para todos los M valores de u con el fin de obtener la transformada de Fourier. Al igual que la señal de entrada $f(x)$, la transformada es una cantidad discreta, y tiene el mismo número de componentes que $f(x)$.

El concepto de dominio en frecuencia se desprende directamente de la fórmula de Euler.

$$e^{j\theta} = \cos \theta + j \sin \theta \quad (1.3)$$

Sustituyendo la expresión 1.3 en la ecuación 1.1, tenemos:

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) [\cos 2\pi ux/M - j \sin 2\pi ux/M] \quad (1.4)$$

para $u=0,1,2,\dots,M-1$

Por consiguiente, vemos que cada término de la Transformada de Fourier (esto es, el valor de $F(u)$ para cada valor de u) está compuesto de la suma de todos los valores de la función $f(x)$. Los valores de $f(x)$, a su vez, están multiplicados por senos y cosenos de varias frecuencias. El dominio (los valores de u) sobre los cuales se encuentran definidos los valores de $F(u)$ se le llama **dominio en frecuencia**, por que u determina la frecuencia de los componentes de la transformada.

Como se observa en las ecuaciones 1.1 y 1.2, los componentes de la Transformada de Fourier son números complejos. En el análisis de números complejos algunas veces es conveniente expresar $F(u)$ en coordenadas polares de la forma:

$$F(u) = |F(u)| e^{-j\phi(u)} \quad (1.5)$$

En donde:

$$|F(u)| = [R^2(u) + I^2(u)]^{1/2} \quad (1.6)$$

Es llamada la magnitud o espectro de la Transformada de Fourier. Los términos $R(u)$ e $I(u)$ se refieren a la parte real y a la parte imaginaria de $F(u)$, respectivamente. Además,

$$\phi(u) = \tan^{-1} \left[\frac{I(u)}{R(u)} \right] \quad (1.7)$$

es llamada el ángulo de fase o ángulo del espectro de la transformada.

Dentro del análisis de imágenes digitales es muy utilizado el concepto de Transformada de Fourier por la facilidad que implica trabajar en un dominio distinto al original sin pérdida alguna de información. Desafortunadamente en el análisis de texturas no es suficiente la aplicación de la Transformada debido a su incapacidad de detección de invarianzas a lo largo de una imagen.

CAPITULO 2. FILTROS DE GABOR

El contenido de este capítulo esta encaminado a mencionar el trabajo realizado por Dennis Gabor en el área de estudio de señales y la manera en que eventualmente fue creciendo y se fue desarrollando para la detección de texturas. Desde los primeros estudios realizados en señales unidimensionales hasta la generalización hecha por Daugman para señales bidimensionales. También cubrimos aspectos fundamentales de las Máquinas de Vector de Soporte, la fortaleza intrínseca que mantiene como clasificador, y la manera en que modifica el espacio de datos para volverlo clasificable.

2.1 Filtros de Gabor.

Históricamente el estudio de las señales unidimensionales se ha efectuado desde dos enfoques alternativos, el primer enfoque representa las señales como una función del tiempo, y el segundo como una función de frecuencia. Cada una de estas representaciones puede ser expresada en términos de la otra por medio de la Transformada de Fourier y su Inversa, ambas contendrán la misma información de la señal pero de diferente forma. Ambas representaciones son de alguna manera idealizaciones debido a que la primera opera en instantes de tiempo perfectamente definidos y la segunda opera con trenes infinitos de ondas en frecuencias estrictamente definidas. [24]

En el análisis de dichas señales deben ser caracterizadas sus propiedades básicas; qué tipo de eventos contiene una señal y cuándo es que ocurren. Si somos capaces de inferir que la frecuencia nos indica el "qué", y el tiempo, en coordenadas espaciales de dos dimensiones nos indica el "dónde", podemos encontrar una justificación para el desarrollo de la teoría. Esto motivó a la construcción de un enfoque que fuera capaz de representar una señal como una función tanto de tiempo como de frecuencia. El trabajo de Dennis Gabor se vio influenciado por tal motivación, cuando propuso el uso de funciones elementales especiales, después llamadas en su honor Funciones Elementales de Gabor (GEF, *Gabor Elementary Functions*, por sus siglas en inglés), para representar señales simultáneamente en tiempo y en frecuencia.

El trabajo inicial de Gabor estaba destinado a la síntesis de señales, esto es, buscaba la manera de reconstruir una señal en base a una combinación lineal de funciones elementales de Gabor. Del estudio de las funciones propuestas por Gabor se desprenden los Filtros de Gabor. Gabor propuso la expansión de una señal en un conjunto discreto de señales Gaussianas elementales propiamente ajustadas y moduladas [38]. Esto encuentra sus bases en la descomposición de una señal como la combinación de varias señales propuesto por Fourier. Fourier, como ya se mencionó en el capítulo anterior, concibió un método para expresar una función periódica como una suma infinita de funciones oscilatorias, senos y cosenos. De esta manera una función puede ser estudiada en términos de sus

componentes espectrales. Los Filtros de Gabor pueden ser explicados en términos de la Transformada *Wavelet* y de la Transformada Corta de Fourier.

Los *wavelets* ejecutan la descomposición de una función como la suma de otras funciones, pero contrario a la descomposición de Fourier, estas funciones son bases locales, tienen soporte finito y están localizadas a diferentes escalas. Actúan como si tuviéramos una ventana que puede ser posicionada sobre cualquier punto en la función, si movemos la ventana tenemos un análisis local y variando el ancho de la ventana podemos llegar a distintos niveles de detalle. El área de procesamiento de imágenes ha tomado esta teoría para sus señales, extendiendo los algoritmos a dos dimensiones. Las particularidades que presentan las imágenes, como pueden ser las estadísticas sobre el dominio espacial, hacen de las funciones de Gabor la herramienta ideal para su análisis. Las imágenes tienen un comportamiento no estacionario. Esta propiedad obliga a realizar un estudio local de este tipo de señales. Las medidas que puedan tomarse para caracterizar una imagen varían en demasía a lo largo de todos sus puntos debido a su comportamiento no estacionario. Lo que es más, podemos encontrar objetos dentro de las imágenes con tamaños y texturas que también varían. Un análisis de frecuencia en una imagen en algunos casos puede ser suficiente, pero en la mayoría de los casos se requiere de más información, por ejemplo, en donde es que algunos patrones ocurren. Con un análisis en frecuencia no se tiene una clara localización de dónde aparecen los patrones. La diferencia básica de estas funciones con el análisis de Fourier es que las primeras están bien definidas en el espacio (o tiempo) y en frecuencia (escala) distinto de las funciones usadas en Fourier que son senos y cosenos (exponencial compleja) que están bien definidas en el dominio de la frecuencia, pero no en el dominio espacial. Debido a esta localización, permiten estudiar funciones abruptas con un número bajo de coeficientes [40]. Además una dificultad de la transformada de Fourier es que sus coeficientes dependen de la imagen completa, es decir, el valor de la transformada en un punto (u, v) se calcula utilizando cada pixel de la imagen, esto es un inconveniente por que perdemos información espacial. Los filtros de Gabor pueden ser utilizados para analizar estos cambios de frecuencia localizados en el espacio.

Uno de los primeros pasos para resolver el problema de la mala localización de la transformada de Fourier consiste en mover una ventana de duración fija sobre la función y extraer el contenido de frecuencia en ese intervalo. Cuando en esta transformación se utiliza una ventana Gaussiana se le conoce como Transformada de Gabor. Con esta transformación tenemos una mejor localización, sabemos qué frecuencias tenemos y en dónde están localizadas a lo largo de la señal. Si tomamos como señal de entrada una imagen, entonces los filtros de Gabor se convierten en la opción más viable para su estudio.

Los filtros de Gabor tienen fundamento en los estudios realizados por Dennis Gabor en 1946 publicados en su Teoría de la Comunicación [40]. Un filtro de Gabor puede definirse como un filtro lineal cuya respuesta impulso está definida por una función armónica multiplicada por una función Gaussiana. La principal

ventaja que se obtiene al introducir la envolvente Gaussiana es que las funciones de Gabor están localizadas tanto en el dominio del espacio como en el de la frecuencia, tal como sus funciones elementales proponían, a diferencia de lo que ocurre con las funciones sinusoidales, que están perfectamente localizadas en el dominio espacial (o temporal) y completamente deslocalizadas en el de la frecuencia.

El también inventor del holograma propuso la expansión de una onda en términos de paquetes de onda Gaussianos [13]. Un ejemplo de tal paquete de ondas es una onda sinusoidal multiplicada por una función Gaussiana. Si una señal es modulada (multiplicada) por una ventana Gaussiana de una cierta amplitud y tiempo central, entonces una expansión de Fourier de la señal modulada otorga una medida de espectro local. Este espectro no es único ya que la amplitud de la Gaussiana es arbitraria, sin embargo ese espectro local es de gran utilidad. Si una colección de espectros locales es calculada para un conjunto de posiciones de ventanas, el resultado es una descomposición en tiempo-frecuencia llamada la transformada de Gabor.

Gabor determinó la existencia de una representación local tanto en el dominio del tiempo como en el de la frecuencia y esa representación local del tiempo y la frecuencia debe ser discreta por lo que puede ser adaptada para ser usada en varias aplicaciones.

También demostró que la especificación de una señal simultánea en tiempo y en frecuencia esta limitada por un límite inferior del producto de su ancho de banda y su duración [41]. La ecuación 2.1 muestra la definición del límite.

$$(\Delta\chi)(\Delta\omega) \geq \frac{1}{4\pi} \quad (2.1)$$

Él también probó que las señales que sean de la forma como la mostrada en la ecuación 2.2 alcanzan el límite teórico que encontró en primer lugar.

$$s(t) = e^{-\frac{t^2}{\alpha^2} + i\omega t} \quad (2.2)$$

Computacionalmente hablando, las funciones de Gabor forman un conjunto base completo pero no-ortogonal y cualquier función puede ser expandida en términos de estas funciones básicas.

Todo esto sirve de fundamento para explotar las funcionalidades de los filtros de Gabor en el análisis de imágenes, especialmente en el reconocimiento de texturas. En la literatura se han propuesto diversas implementaciones equivalentes de los filtros de Gabor, algunas definidas en el dominio espacial y otras definidas en el dominio de la frecuencia. La ecuación 2.3 muestra la definición de un filtro de Gabor espacial en dos dimensiones.

$$G_{\lambda, \theta, \varphi, \sigma, \gamma}(x, y) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cdot \cos\left(2\pi \frac{x'}{\lambda} + \varphi\right) \quad (2.3)$$

en donde

$$\begin{aligned} x' &= x \cos \theta + y \sin \theta \\ y' &= -x \sin \theta + y \cos \theta \end{aligned}$$

La descripción de cada parámetro se realiza a continuación.

λ - longitud de onda: es la longitud de onda del factor coseno del *kernel* del filtro de Gabor.

θ - ángulo de orientación: ángulo de rotación de la componente Gaussiana.

φ - ángulo de fase: es el ángulo de fase de la portadora sinusoidal.

σ - desviación estándar: es la desviación estándar de la Gaussiana. Determina la extensión espacial del filtro en el dominio espacial

γ - relación de aspecto espacial: determina la forma elíptica de la función de Gabor.

La familia de filtros que serán utilizadas en este estudio puede definirse como se muestra en la ecuación 2.4. La ecuación representa un filtro de Gabor en dominio espacial bidimensional (2D).

$$\begin{aligned} \psi(x, y) &= \frac{f^2}{\pi\gamma\eta} e^{-\left(\frac{f^2}{\gamma^2}x'^2 + \frac{f^2}{\eta^2}y'^2\right)} \cdot e^{j2\pi fx'} \\ x' &= x \cos \theta + y \sin \theta \\ y' &= -x \sin \theta + y \cos \theta \end{aligned} \quad (2.4)$$

Cada parámetro cumple una función especial y por medio de su modificación es como se definirá la capacidad del filtro, reaccionando de distinta manera según los valores que se tenga para cada parámetro.

Como se menciona en [11], la función de Gabor es más interesante cuando se estudia en el dominio de la frecuencia. Es aquí en donde podemos encontrar una explicación más detallada de los parámetros que componen la función de Gabor. Por lo tanto en esta tesis se toma la familia de filtros presentada anteriormente pero en el dominio de la frecuencia, quedando su representación como se muestra en la ecuación 2.5.

$$\Psi(u, v) = e^{-\frac{\pi^2}{f^2}(\gamma^2(u-f)^2 + \eta^2 v^2)}$$

$$u' = u \cos \theta + v \sin \theta$$

$$v' = -u \sin \theta + v \cos \theta$$
(2.5)

2.1.1 Formulaciones de Filtros de Gabor

El análisis de los filtros de Gabor presentado en [48] es la base para la utilización de la familia de filtros de la ecuación 2.5. En dicho artículo se parte de la formulación de un filtro de Gabor en dos dimensiones de la forma:

$$\varphi(x, y) = \underbrace{e^{-(\alpha^2 x'^2 + \beta^2 y'^2)}}_{\text{Gaussian}} \underbrace{e^{j2\pi f_0 x'}}_{\text{planewave}}$$

$$x' = x \cos \theta + y \sin \theta$$

$$y' = -x \sin \theta + y \cos \theta$$
(2.6)

En donde f_0 es la frecuencia de la onda sinusoidal plana, θ es la rotación contraria a las manecillas del reloj de la Gaussiana y la onda plana, α es la nitidez de la Gaussiana a lo largo del eje paralelo a la onda, y β es la nitidez a lo largo del eje perpendicular a la onda.

El filtro de la ecuación 2.6 puede ser normalizado estableciendo la razón de proporción de la frecuencia de la onda y los valores de nitidez de la Gaussiana, por ejemplo:

$$\gamma = \frac{f_0}{\alpha}; \quad \eta = \frac{f_0}{\beta}$$
(2.7)

Por lo que el filtro espacial contiene un número constante de ondas. Esta formulación arregla el comportamiento de la respuesta sin importar la frecuencia y hace la respuesta en DC idéntica para todas las frecuencias. Se busca que la respuesta en DC sea pequeña, de otra manera la intensidad de la imagen afecta la respuesta. Esto puede ser controlado estableciendo el parámetro γ lo suficientemente largo. Otro enfoque es utilizar un Filtro de Gabor modificado en donde una Gaussiana con la misma respuesta en DC sea sustraída del filtro. Para hacer el área debajo de la Gaussiana unitaria, un factor de normalización $\alpha\beta/\pi$ debe ser usado, por lo tanto un filtro normalizado puede ser representado como:

$$\varphi(x, y) = \frac{f_0}{\pi\gamma\eta} e^{-\left(\frac{f_0^2}{\gamma^2} x'^2 + \frac{f_0^2}{\eta^2} y'^2\right)} e^{j2\pi f_0 x'}$$
(2.8)

A la ecuación 2.8 se le aplica la transformada de Fourier para obtener su representación en el dominio de la frecuencia, quedando como resultado la ecuación 2.9 que finalmente es la utilizada en esta tesis.

$$\Psi(u, v) = e^{-\frac{\pi^2}{f_0^2}(\gamma^2(u'-f_0)^2 + \eta^2 v'^2)}$$

$$u' = u \cos \theta + v \sin \theta \quad (2.9)$$

$$v' = -u \sin \theta + v \cos \theta$$

2.2 Diseño de Filtros versus Banco de Filtros

Dentro del estudio de los filtros de Gabor se ha identificado la utilización de dos esquemas de solución, mencionados en [15], como son el enfoque de banco de filtros y el enfoque de diseño de filtros. Se utiliza el enfoque de banco de filtros para generar un grupo de filtros que comprendan un amplio rango de valores que ayuden en la identificación de algún tipo particular de textura, tiene la desventaja de que generalmente es necesario crear una familia de filtros con distintas orientaciones y frecuencias, lo que resulta en una gran cantidad de elementos a ser procesados, lo que regularmente requiere de cierto poder de cómputo. En el enfoque de diseño de filtros se diseña solo un filtro o unos pocos filtros para una aplicación determinada como un esfuerzo por reducir las dificultades del enfoque de banco de filtros. En ambos casos es necesaria la intervención humana para seleccionar los parámetros adecuados de los filtros.

En el dominio de frecuencia el filtro es una función Gaussiana modulada por una onda sinusoidal dispersa [42]. Dicha Gaussiana tiene una frecuencia central (u, v) , una desviación estándar compuesta por σ_x y σ_y , esos parámetros determinan la forma que tomará la superficie Gaussiana, y un ángulo de rotación θ con respecto al origen. La onda sinusoidal dispersa tiene una orientación θ , y una frecuencia:

$$F = \sqrt{u^2 + v^2} \quad (2.10)$$

en la figura 2.1 podemos ver la manera en que cada parámetro se ubica dentro de un solo filtro.

Para el diseño de bancos de filtros se deben tomar en cuenta algunos parámetros más. En [26] se identifican tres parámetros extra, la frecuencia central del filtro en su valor más alto (F_M), el número total de frecuencias (n_f) y el número total de orientaciones (n_o), también considerados en [39].

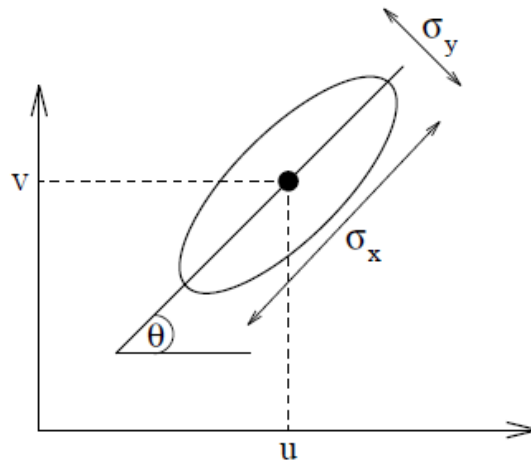


Figura 2.1. Representación de un filtro de Gabor con los cuatro parámetros que lo definen.

En la figura 2.2 se presenta un diagrama de representación de los parámetros antes mencionados dentro de un banco de filtros.

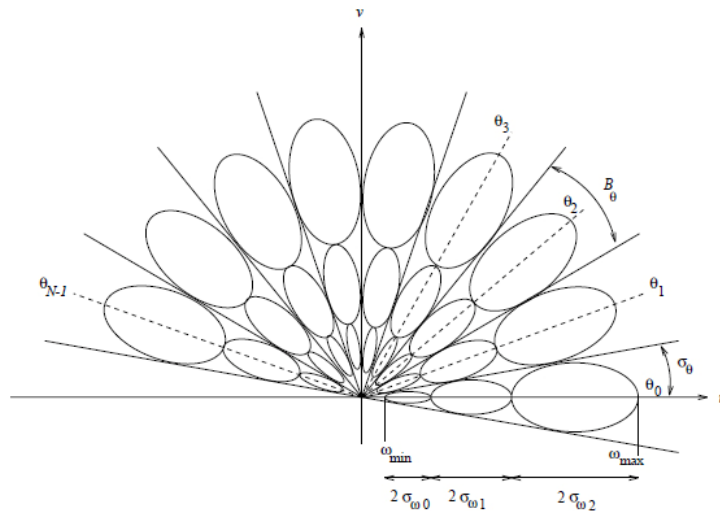


Figura 2.2. Parámetros de un grupo de filtros de Gabor con 9 orientaciones.

Se debe cumplir que la frecuencia del filtro (F) sea menor que la frecuencia máxima (F_M). Como podemos apreciar los valores de σ_x y σ_y determinan la forma elíptica de la Gaussiana y además, en la figura, ayudan a crear un banco de filtros en donde las curvas se tocan unas con otras.

2.3 Sistema Visual Humano

Los primeros desarrollos en análisis de señales usando los filtros de Gabor aprovecharon su funcionalidad como filtros pasa-banda para señales unidimensionales. En 1980 John Daugman generalizó las funciones de Gabor para ser utilizadas en forma bidimensional para poder modelar el perfil de los campos receptivos de las células básicas en la corteza visual primaria. Tomando como referencia el estudio de los campos receptivos de las células de la corteza visual de los felinos.

$$\begin{aligned} f(x, y) &= \exp\left\{-\pi\left[(x-x_0)^2/a^2 + (y-y_0)^2/b^2\right]\right\} \\ &\quad \times \exp\left\{-2\pi i\left[u_0(x-x_0) + v_0(y-y_0)\right]\right\}, \\ F(u, v) &= \exp\left\{-\pi\left[(u-u_0)^2/a^2 + (v-v_0)^2/b^2\right]\right\} \\ &\quad \times \exp\left\{-2\pi i\left[x_0(u-u_0) + y_0(v-v_0)\right]\right\} \end{aligned} \quad (2.11)$$

Otros estudios hechos en psicofisiología han sugerido que el cerebro lleva a cabo un análisis multicanal, de frecuencia y orientación de la imagen que se forma en la retina. En [49] se llevaron a cabo experimentos psicofisiológicos usando la integración de varios patrones. Ellos sugieren que el sistema visual descompone la imagen en imágenes filtradas de varias frecuencias y orientaciones. En [50] se ha estudiado el cerebro del mono macaco el cual se asume que es parecido al cerebro humano en cuanto al procesamiento visual. Ellos registraron la respuesta de las células básicas en la corteza visual de los monos hacia rejillas sinusoidales de varias frecuencias y orientaciones y concluyeron que estas células están sintonizadas en rangos estrechos de frecuencia y orientación. Estos estudios motivaron a los investigadores a aplicar enfoques de filtro multicanal al análisis de texturas.

Dentro de los métodos enumerados en el capítulo anterior para el estudio de las texturas prestaremos especial atención al enfoque de procesamiento de señales, no sólo porque ha sido el de mayor auge y mayor utilización dentro del análisis de texturas, sino como se menciona en [39], estos modelos aprovechan el hecho de que la mayoría de las texturas, debido a su comportamiento repetitivo, son más fáciles de representar en el dominio espectral que el dominio espacial. El enfoque espacio-frecuencia para el análisis de texturas ha proporcionado las soluciones más confiables al problema de caracterización de texturas. Su éxito se refleja en la gran cantidad de publicaciones que se han hecho en este campo. Aquí encontramos técnicas en las que las características son obtenidas por medio del estudio de nuevas representaciones espectrales o conjuntando representaciones

en espacio y frecuencia. Los más representativos son Fourier y los filtros y *wavelets* de Gabor.

Estas particularidades sirven como plataforma para poder equiparar las propiedades de las texturas con las propiedades del Sistema Visual Humano. En [18] se refieren algunos estudios que han demostrado que las personas reaccionan ante tres propiedades de las texturas: repetición, direccionalidad y complejidad. Podemos caracterizar la repetición en las texturas como su frecuencia espacial, y su direccionalidad como su orientación.

Por lo tanto es necesario utilizar una metodología de caracterización que permita filtrar una imagen con múltiples filtros pasa-banda sintonizados con los componentes de frecuencia y orientación dominantes en las texturas examinadas. El candidato que será considerado para extraer las características de la imagen son los filtros de Gabor debido a tres razones: 1) motivación biológica, son similares a los campos receptivos de las células básicas de la corteza visual primaria, 2) motivación matemática, son adecuados para medir frecuencias espaciales locales, 3) motivación empírica, ha demostrado ser tolerante a distorsiones dentro de algunas tareas de reconocimiento de patrones.

2.4 Máquinas de Vector de Soporte

Las Máquinas de Vector de Soporte (*Support Vector Machines*, por sus siglas en inglés, SVM) fueron introducidas por Vladimir Vapnik en 1979, aunque es hasta la década de los 90 en que comienza a ser explotado formalmente el concepto, principalmente en tareas de reconocimiento de patrones. Las SVM son consideradas la primera derivación práctica de la Teoría de Aprendizaje Estadístico [43]. El objetivo principal de esta teoría es ofrecer un esquema para el estudio del problema de inferencia, esto es, la obtención de conocimiento, creación de predicciones, toma de decisiones y la construcción de modelos a partir de un conjunto de datos.

Las Máquinas de Vector de Soporte son un clasificador binario que toma sus decisiones construyendo un límite de decisión lineal o hiperplano que, de manera óptima, separa los puntos de dos clases en un espacio de características de dimensión superior, y además ese margen es maximizado [20]. Las Máquinas de Vector de Soporte son un sistema de aprendizaje basado en el uso de un espacio de datos de funciones lineales en un espacio de mayor dimensión inducido por un *kernel*, en el cual los datos son entrenados por un algoritmo tomado de la teoría de optimización el cual utiliza elementos de la teoría de generalización. SVM es un sistema para entrenar máquinas de aprendizaje lineal eficientemente tanto para la clasificación como para la regresión [28].

Parte de la implementación que hacemos en esta tesis radica en el poder de clasificación que se ha demostrado poseen las Máquinas de Vector de Soporte. Consideramos que la clasificación de imágenes de acuerdo a textura no es un problema separable linealmente ya que algunas texturas presentan similitudes de

acuerdo a su constitución, por lo cual es necesario tomar consideraciones especiales al momento de definir los parámetros del clasificador.

Tomando un grupo de datos de entrada como dos conjuntos de vectores en un espacio de dimensión n , una SVM construirá un hiperplano de separación en ese espacio, uno que sea capaz de maximizar el margen entre los dos conjuntos de datos. Para calcular dicho margen, se construyen dos hiperplanos paralelos, uno en cada lado del hiperplano de separación, los cuales son "empujados" en contra de los dos conjuntos de datos. Intuitivamente, una buena separación se consigue con el hiperplano que tenga la mayor distancia hacia los puntos de ambas clases, ya que en general entre más grande sea el margen, el error de generalización del clasificador es mejor.

Las SVM parten de la finalidad de separar datos con un hiperplano, y extender estos límites de decisión no lineales usando el llamado truco *kernel* (*kernel trick*, que se menciona más adelante) [29], la ecuación de un hiperplano general es:

$$w'x + b = 0 \quad (2.12)$$

en donde x representa un punto (un vector), w representa los pesos (también un vector). El hiperplano debe separar los datos de tal manera que:

$$\begin{array}{ll} w'x_k + b > 0 & \text{Para todos los } \mathbf{x}_k \text{ de una} \\ & \text{clase } y, \\ w'x_j + b < 0 & \text{Para todos los } \mathbf{x}_j \text{ de la otra} \\ & \text{clase} \end{array}$$

cabe mencionar que si los datos son separables de esta manera, existirá probablemente más de una manera de hacerlo. De entre todos los posibles hiperplanos existentes, SVM selecciona aquel en el cual la distancia entre el hiperplano y los datos más cercanos sea la más amplia posible [29].

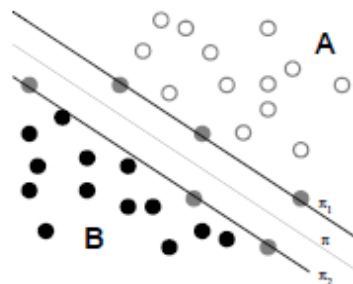


Figura 2.3. Conjunto e hiperplano de separación óptimo en R^2 . Los huecos representan los vectores con etiqueta $y = 1$ y los restantes representan los vectores con etiqueta $y = -1$.

Una muestra de funcionalidad se presenta en [45]. Se define un conjunto de vectores $\{(x_1, y_1), \dots, (x_n, y_n)\}$ donde $x_i \in \mathfrak{R}^d$ e $y_i \in \{1, -1\}$, un conjunto de etiquetas, para $i = 1, \dots, n$ se dice separable si existe algún hiperplano en \mathfrak{R}^d que separa los vectores $X = \{x_1, \dots, x_n\}$ con etiqueta $y_i = 1$ de aquellos con etiqueta $y_i = -1$. Dado un conjunto de datos separables se dice que existe (al menos) un hiperplano que separa los vectores x_i , $i = 1, \dots, n$, esto podemos observarlo en la figura 2.3.

Las SVM buscan entre todos los hiperplanos separadores aquel que maximice la distancia de separación entre los conjuntos $\{(x_i, 1)\}$ y $\{(x_i, -1)\}$, que son las dos clases posibles, de tal forma que:

$$\begin{aligned} x_i w + b &\geq +1 \quad \text{para } y_i = +1 \quad (\text{región } A) \\ x_i w + b &\leq -1 \quad \text{para } y_i = -1 \quad (\text{región } B) \end{aligned} \quad (2.13)$$

Sin embargo en la práctica no es habitual trabajar con conjuntos separables, es muy difícil encontrar un hiperplano lineal que sea capaz de separar los datos, incluso si tenemos un límite de decisión arqueado, probablemente la separación de los datos no es deseable: si los datos son atípicos o tienen ruido, un límite de decisión que ignore algunos datos es mejor que uno que considere los datos anómalos. En estos casos se tienen vectores de una clase dentro de la región correspondiente a los vectores de otra clase y por lo tanto nunca podrán ser separados de esta clase por medio de hiperplanos. En estas situaciones se dice que el conjunto es **no separable**, un ejemplo de esta situación puede observarse en la figura 2.4. No es difícil ampliar las ideas generales del caso separable al caso no separable introduciendo una variable ξ de holgura en las restricciones y plantear un nuevo conjunto de restricciones:

$$\begin{aligned} x_i w + b &\geq +1 - \varepsilon_i \quad \text{para } y_i = +1 \\ x_i w + b &\leq -1 + \varepsilon_i \quad \text{para } y_i = -1 \\ \varepsilon_i &\geq 0 \quad \forall i = 1, \dots, n \end{aligned} \quad (2.14)$$

se tiene ahora que para que se produzca un error en la clasificación de un vector de entrenamiento es necesario que el valor correspondiente a ξ_i sea superior a la unidad. Así, si en el vector x_i se comete un error entonces $\xi_i \geq 1$ y por lo tanto $\sum_i \xi_i$ es una cota superior del número de errores que se cometen dentro del conjunto de entrenamiento.

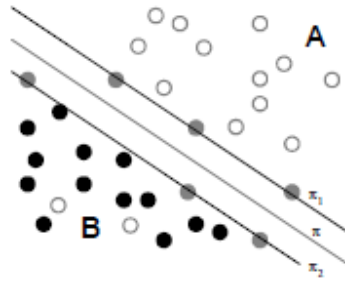


Figura 2.4. Ejemplo de hiperplano de separación en donde existen datos no separables.

2.4.1 El Margen Blando.

Cuando los datos no pueden ser separados por medio de una línea recta, existen datos que el clasificador considera como erróneos. De manera intuitiva esperaríamos que una SVM sea capaz de lidiar con "errores" en los datos permitiendo que algunos datos anómalos caigan en el "lado incorrecto" del hiperplano de separación. Para manejar casos como éste el algoritmo SVM debe ser modificado de manera que pueda definirse un 'margen blando'. Esto permite que los datos puedan abrirse camino a través del borde del hiperplano de separación sin afectar el resultado final. En la figura 2.5 podemos observar un elemento que se encuentra alejado de los elementos de su clase. Al trazar el hiperplano de separación en la figura 2.6 podemos observar que el elemento identificado anteriormente como *outlier*, permanece en el lado incorrecto, lo cual es aceptado debido al criterio de margen blando. El *outlier* se encuentra en el mismo lado de la línea con miembros de clase opuesta.

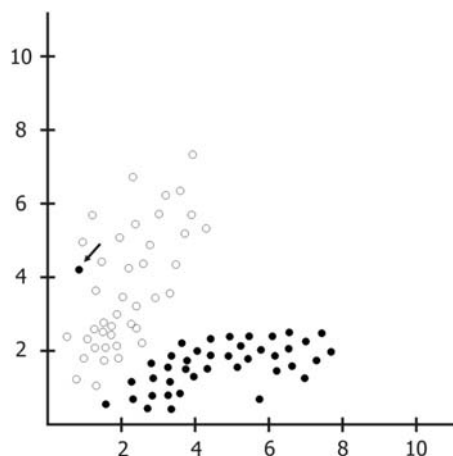


Figura 2.5. Elemento considerado *outlier* dentro del conjunto de datos.

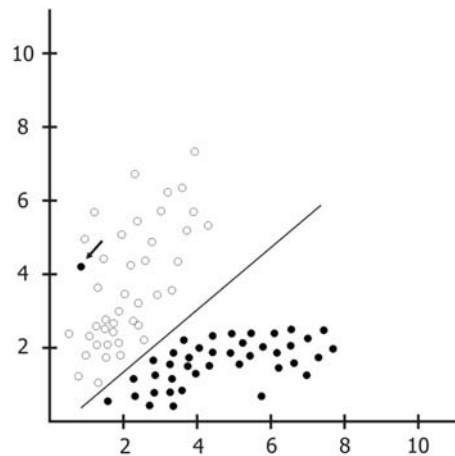


Figura 2.6 *Outlier* aceptado después de trazar el hiperplano de separación.

Por supuesto que no se busca que el SVM tenga muchos errores de clasificación, por lo que la introducción del margen blando requiere que la participación del usuario en la selección de un parámetro que controle, específicamente, el número de ejemplos que se les permite violar el hiperplano de separación y qué tan lejano por sobre la línea se les permite estar. Establecer dicho parámetro se vuelve una tarea difícil debido al hecho de que aún queremos un margen amplio con respecto de los datos clasificados correctamente. Lo que es más, los parámetros del margen blando manejan una compensación entre las violaciones del hiperplano y el tamaño del margen.

2.4.2 La Función *Kernel*.

Cuando se tiene un conjunto de datos que no son linealmente separables, se recurre al uso de una función núcleo (*kernel*). La función *kernel* permite a las SVM formar límites no lineales [29]. La representación por medio de *kernel* ofrece una solución alternativa al problema de no linealidad, proyectando la información a un espacio de características de mayor dimensión [28]. Esto se logra cambiando la representación de la función, esto es similar a mapear el espacio de entrada X a un nuevo espacio H , denominado espacio característico, en la forma:

$$\phi: X \subset \mathbb{R}^d \rightarrow H \quad (2.15)$$

Ahora en lugar de considerar el conjunto de vectores de entrada $\{x_1, \dots, x_n\}$ se considera los vectores transformados $\{\phi(x_1), \dots, \phi(x_n)\}$, como se muestra en la figura 2.7. Al hacer esta sustitución se consigue una SVM planteada en un nuevo espacio (a esto se le llama el truco *kernel* o "*kernel trick*"), y algo que resulta muy importante en la práctica es que la ejecución de esta técnica no lineal consume la misma cantidad de recursos computacionales que si la técnica fuese lineal.

Al usar $\varphi: X \rightarrow H$ se trabaja en un nuevo espacio H por lo cual el vector solución w se encuentra en este espacio. Por lo tanto, puede ocurrir que sobre el conjunto X inicial no se tenga definida ningún tipo de estructura, y la función φ sirve para dar una estructura a los datos y poder aplicar una adecuada clasificación.

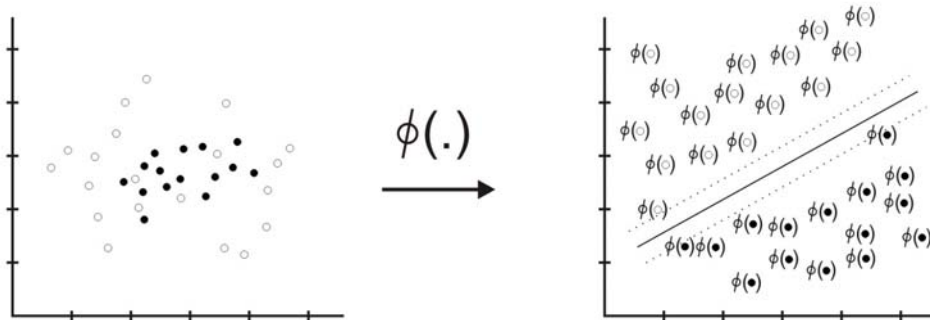


Figura 2.7 Transformación del plano al aplicar el *kernel*.

Cuando se tiene un grupo de datos como los que se muestran en la figura 2.8 puede observarse que estos datos no pueden ser separados utilizando una línea recta, pero usando un *kernel* relativamente simple que proyecte los datos de un espacio de dos dimensiones a un espacio de cuatro dimensiones (correspondiente al producto de cada par de características) permiten que los datos sean linealmente separables. Como no podemos proyectar los datos en cuatro dimensiones, solo nos limitaremos a mostrar el hiperplano de SVM en el espacio original de dos dimensiones. El resultado se muestra en la figura 2.9.

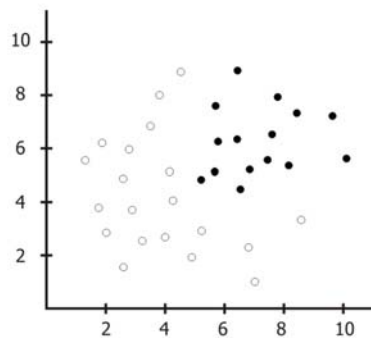


Figura 2.8 Datos de muestra que no son separables linealmente.

Es posible comprobar que para cualquier conjunto de datos existe una función *kernel* que permita que los datos sean linealmente separables. Dicha observación genera la pregunta de ¿por qué no utilizar siempre la transformación a un espacio de dimensión muy alta para asegurar que puede encontrarse un hiperplano óptimo de separación?

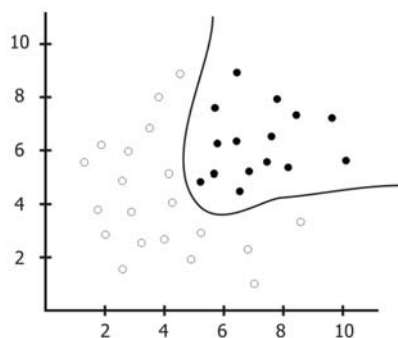


Figura 2.9 Hiperplano de separación después de la transformación.

Y esa es una pregunta válida, desafortunadamente proyectar un conjunto de datos hacia espacios de muy alta dimensionalidad puede ser problemático debido a la llamada maldición de la dimensionalidad (*curse of dimensionality*, en inglés), conforme se incrementa el número de variables en consideración, el número de posibles soluciones también se incrementa, pero exponencialmente, consecuentemente es más difícil para cualquier algoritmo encontrar una solución adecuada. En la figura 2.10 se muestra un ejemplo de lo que sucede cuando un conjunto de datos es proyectado en un espacio con demasiadas dimensiones, más de las que el conjunto de datos requiere.

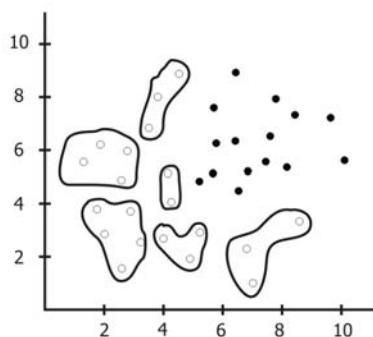


Figura 2.10 Generación de múltiples hiperplanos con *kernel* de alta dimensión.

La figura 2.10 contiene los mismos datos que anteriormente, pero el plano proyectado viene de una SVM que utiliza una función *kernel* de dimensión muy alta. Como resultado tenemos que las fronteras entre las clases es muy particular a los ejemplos de conjunto de datos de entrenamiento. La SVM tiene que aplicar sobreajuste (*overfit*) a los datos. Claramente la Máquina de Vector de Soporte no aplicará generalización de manera adecuada cuando sea alimentada con nuevos tipos de datos.

El término "*curse of dimensionality*" acuñado por Richard Bellman, hace referencia a un problema causado por el crecimiento exponencial en volumen asociado con la adición de más dimensiones a un espacio matemático.

Esta situación resalta la dificultad inherente cuando se aplica un clasificador SVM a un conjunto nuevo de datos, uno esperaría poder utilizar un *kernel* que sea capaz de separar los datos sin tener que introducir muchas dimensiones. Pero, ¿cuál es la mejor manera de seleccionar dicha función? Desafortunadamente en la mayoría de los casos la única respuesta realista es ensayo y error. Los investigadores generalmente comienzan una SVM sencilla y experimentan con una variedad de funciones *kernel* estándar. Una función óptima puede ser seleccionada de un conjunto fijo de *kernels* por medio de algún método estadístico utilizando validación cruzada. Sin embargo, este enfoque consume muchos recursos y no garantiza que una función *kernel* que no haya sido considerada en la validación cruzada no sea mejor.

En [28] se destaca una de las razones por la cual SVM ha tenido éxito, el método no padece de mínimos locales y el modelo sólo depende de los datos con más información llamados Vectores de Soporte. El objetivo de la clasificación por medio de SVM's es concebir una manera eficiente, computacionalmente hablando, de aprender hiperplanos de separación "buenos" en un espacio de características de alta dimensionalidad, en donde "buenos" se refiere a ser capaces de optimizar los problemas de generalización, y por eficientes computacionalmente se refiere a algoritmos que sean capaces de manejar tamaños de muestra del orden de las 100,000 instancias [44].

El problema general que buscan resolver las SVM's consiste en buscar, para una tarea de aprendizaje dada, con una cantidad finita de datos, una adecuada función que permita llevar a cabo una buena generalización, que sea resultado de una adecuada relación entre la precisión alcanzada con un particular conjunto de entrenamiento y la capacidad del modelo [45].

En [46] se presenta una lista de algunas aplicaciones en las que se han utilizado SVM como herramienta de reconocimiento de patrones, entre los que encontramos reconocimiento de dígitos escritos a mano, reconocimiento de objetos, identificación del habla, detección de rostros en imágenes, categorización de texto. En [14] se sitúa a las SVM's también como herramienta para tareas como clasificación de imágenes de acuerdo a contenido, predicción de comportamiento financiero, análisis de señales biomédicas, diagnóstico médico y minería de datos.

SVM es el clasificador lineal de margen máximo operando sobre datos extendidos no lineales. La particularidad de adherir características de dimensión no lineal a los datos no es crítico, siempre y cuando se tenga un conjunto de gran tamaño. El clasificador lineal será capaz de resolver qué información es útil para separar los datos. El *kernel* particular debe ser escogido en base a ensayo y error en el conjunto de prueba, aunque se ha demostrado que los *kernel* tienen un desempeño equivalente, sugiriendo que la selección del *kernel* no es importante.

La decisión de utilizar las Máquinas de Vector de Soporte para la tarea de clasificación de las imágenes esta influenciada por los resultados obtenidos en otras investigaciones en las cuales se han alcanzado resultados positivos, ya sea en clasificación o en segmentación de texturas. Como se resalta en [9] en sus conclusiones, la combinación de distintos métodos de caracterización de texturas mejora los resultados de clasificación, especialmente cuando se combinan métodos estadísticos basados en el GLCM (*Gray Level Occurrence Matrix*, Matriz de Ocurrencia de Niveles de Gris). Los filtros de Gabor permitieron que una gran parte de la información de textura fuera condensada en unas pocas variables.

Las Máquinas de Vector de Soporte ofrecen características de clasificación que no pueden ser obviadas, por lo cual fueron consideradas como herramienta complementaria dentro del proceso de caracterización de texturas.

Una vez identificados los requerimientos en el proceso, es necesario conformar los elementos que darán estructura a la solución del problema. Cabe señalar que el método presentado en este trabajo no es autónomo ya que requiere de una configuración particular de acuerdo al tipo de textura que se analice, además de la participación del usuario en cada etapa del proceso.

CAPÍTULO III. IMPLEMENTACIÓN DEL MÉTODO

En ese capítulo establecemos las bases del proceso de clasificación de imágenes, el método base para modelar el sistema de clasificación es la extracción de características. Se mencionan los métodos de clasificación más utilizados en publicaciones, y se realiza una descripción general del método de clasificación propuesto. Una parte importante del capítulo muestra el proceso de preparación de los experimentos, comenzando con el pre-procesamiento de las imágenes de entrada para que sean adaptadas de acuerdo a las necesidades de los filtros, y de manera más representativa, el procedimiento para determinar los valores de los parámetros de los filtros. Por último se describe la manera en que los datos filtrados son distribuidos para servir como entrada al clasificador SVM.

3.1 Clasificación

La clasificación de imágenes, en su sentido más general, ha sido estudiada a través de diversos métodos. Los métodos más desarrollados se relacionan con la composición de dos o más técnicas, que trabajando de manera serial permiten la obtención de mejores resultados.

Para llevar al cabo el proceso de clasificación de imágenes pueden distinguirse dos pasos fundamentales, por un lado la extracción de características de la imagen, esto es, obtener información relevante de la imagen, y posteriormente su categorización de acuerdo con algún criterio de clasificación, dicho criterio dependerá de la razón del método utilizado.

El hecho de extraer características de las imágenes se hace con el fin de agilizar su procesamiento y que el resultado de clasificación no solo sea lo más preciso posible sino que además se ejecute en el menor tiempo posible.

3.1.1 Extracción de Características

La extracción de características se usa como una manera especial de disminuir la dimensionalidad de los datos de entrada en un sistema. Cuando un conjunto de datos de entrada de algún algoritmo es demasiado largo para ser procesado o se espera que sea redundante (una gran cantidad de datos, pero no mucha información), entonces los datos de entrada pueden ser transformados en un conjunto reducido de características representativo (también llamado vector de características). A la transformación de los datos de entrada en un conjunto de características representativas se le llama extracción de características. Si las características extraídas son cuidadosamente seleccionadas se espera que el conjunto de características contenga información relevante de los datos de entrada con el fin de realizar el procedimiento deseado usando una representación reducida en lugar de el conjunto completo de datos de entrada.

El procedimiento implica simplificar la cantidad de recursos requeridos para describir un conjunto grande de datos de manera más precisa. Cuando se realiza el

análisis de datos complejos uno de los mayores problemas proviene del número de variables involucradas. Trabajar con un gran número de variables generalmente requiere una gran cantidad de memoria y poder de cómputo.

La extracción de características de una imagen se hace para tomar la información más importante que contenga la imagen y que ayuden a interpretar la escena sin necesidad de utilizar el conjunto completo de datos. Una imagen puede contener una gran cantidad de datos, aunque la mayoría puede proporcionarnos muy poca información para interpretar la escena, un ejemplo de ello podemos encontrarlo en algoritmos de detección de bordes, en la figura 3.1.a podemos observar la imagen que contiene una escultura, los datos de la imagen que definen el fondo e incluso el la estructura interna del objeto son irrelevantes para el proceso de detección de bordes, en la figura 3.1.b puede observarse el resultado final de la detección de bordes y vemos que la información resultante puede ser interpretada con muy poca cantidad de datos, pero que resultan bastante significativos.

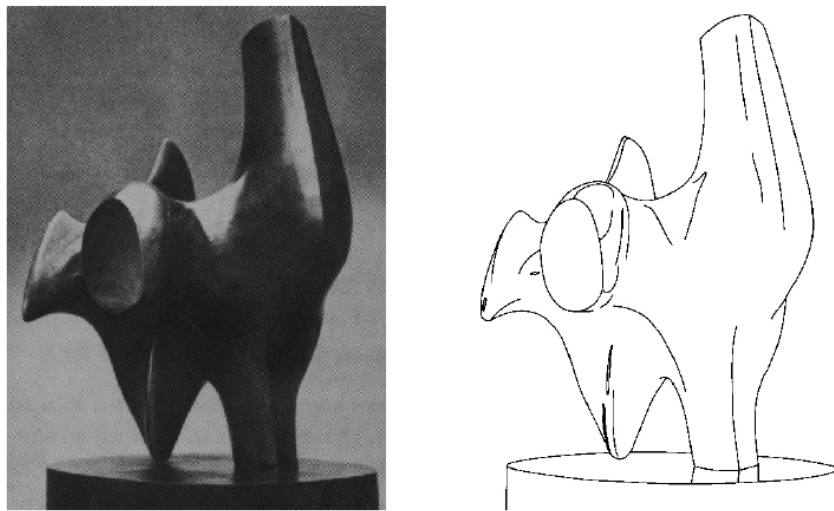


Figura 3.1. a) Escultura original, b) resultado del proceso de detección de bordes.

Es por esto que debe existir una etapa de extracción de características, estas características deben cumplir, entre otras, las siguientes condiciones:

- Su extracción a partir de la imagen no debe suponer un coste excesivo al sistema en el cual está integrado. El tiempo total de extracción debe ser lo más pequeño posible.
- Su localización debe ser muy precisa. El error cometido en la estimación de las características también debe ser lo más pequeño posible.
- Deben ser robustas y estables. Deberían permanecer a lo largo de una secuencia.

- Contendrán la máxima información posible de la escena, es decir, debemos ser capaces de extraer información de tipo geométrico de ellas.

Existen varias técnicas de extracción de características entre las cuales podemos mencionar las siguientes:

- Análisis de Componentes Principales (*Principal Component Analysis, PCA*)
- Reducción Dimensional con Múltiples Factores (*Multifactor Dimensionality Reduction*)
- Reducción Dimensional No Lineal (*Nonlinear Dimensionality Reduction*)
- Isomap
- *Kernel PCA*
- Análisis de Semántica Latente (*Latent Semantic Analysis*)
- Análisis de Componentes Independientes (*Independent Component Analysis*)

A continuación se presenta una lista de técnicas de extracción de características que pueden realizarse en imágenes de acuerdo al objetivo que se busque:

- Detección de bordes
- Detección de esquinas
- Detección de gotas o manchas (BLOB)
- Detección de crestas
- Detección de Líneas, Círculos, Elipses (Transformada Hough)
- Detección de Áreas
- Detección de Texturas
- Detección de Formas Arbitrarias

El objeto de esta tesis marca la extracción de características de textura de una imagen para su interpretación.

3.1.2 Clasificación de Imágenes

La clasificación de imágenes ha sido referida en un gran número de publicaciones y aplicaciones, el objetivo primordial es la categorización de un conjunto, generalmente muy grande, de imágenes de entrada. El algoritmo implementado debe ser capaz de "seleccionar" las imágenes de entrada de acuerdo a su cercanía con el o los conjuntos definidos. Se han clasificado imágenes de acuerdo a su contenido, formas, objetos, personas, etcétera.

Existen diversos procedimientos de clasificación de datos, igualmente aplicables a la clasificación de imágenes, éstas se dividen en dos categorías plenamente diferenciadas, clasificación supervisada y clasificación no supervisada.

En la clasificación supervisada se tiene pleno conocimiento de los grupos que quieren obtenerse y los datos se etiquetan de acuerdo a la pertenencia de algunos de los grupos existentes. En este tipo de clasificación se requiere la participación del usuario para definir las clases en que se divide el conjunto de datos de entrada. Dentro del proceso se ejecutan varios métodos de comparación para determinar a qué clase pertenece cada dato. Entre las técnicas de clasificación supervisada podemos encontrar los métodos de Distancia Máxima, Distancia de Mahalanobis, Máxima Similitud, Redes Neuronales y Máquinas de Vector de Soporte.

En la clasificación no supervisada se desconocen las categorías que se desean y los datos no están etiquetados de acuerdo a algún criterio. Los datos son clasificados de acuerdo a operadores estadísticos que determinan qué datos pueden ser categorizados en qué grupo. No se cuenta con clases de entrenamiento. *K-means* e *ISODATA* son dos algoritmos de clasificación no supervisada.

Al utilizar las SVM como técnica de clasificación establecemos un tipo de clasificación supervisada, en donde los datos de entrada deben dividirse en conjunto de entrenamiento y conjunto de prueba.

3.2 Evaluación de la Clasificación

En un proceso de clasificación binario es importante tomar en cuenta el umbral de clasificación que nos permitirá establecer valores a la clasificación cuando existe el caso de tener un dato plenamente identificado como parte de un conjunto pero etiquetado por el clasificador como parte del otro conjunto, esto es, la presencia de falsos positivos y de verdaderos negativos.

Este fenómeno se estudia como tal en áreas estadísticas y es explotado concretamente por áreas médicas; en el proceso de clasificación es necesario identificar el umbral que nos permite diferenciar la clasificación de dos conjuntos de datos, lo que nos arrojará el error en dicha clasificación.

En la figura 3.2 se explica la manera en que los datos son etiquetados en el proceso de clasificación. De dos conjuntos de datos Conjunto A y Conjunto B se tienen los **Verdaderos Positivos (VP)**, se trata de los datos que pertenecen al Conjunto B y que fueron etiquetados como tal. Los **Verdaderos Negativos (VN)**, datos que pertenecen al Conjunto A y que fueron etiquetados como tal. **Falsos Positivos (FP)**, se trata de datos que pertenecen al conjunto A pero que fueron etiquetados como pertenecientes al Conjunto B, y por último los **Falsos Negativos (FN)**, datos que pertenecen al Conjunto B pero que fueron etiquetados como pertenecientes al Conjunto A.

Los dos primeros grupos corresponden a datos bien clasificados, los dos siguientes a datos mal clasificados. La situación ideal consistiría en lograr una clasificación de los conjuntos de datos en que no hubiera ningún Falso Negativo ni ningún Falso Positivo, pero hemos visto que debido a la intersección de las distribuciones de los grupos, esta situación solo es teórica, conociendo de antemano que los errores son inevitables.

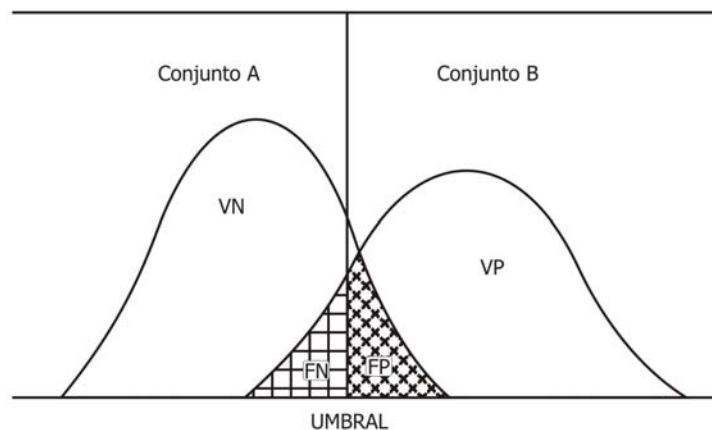


Figura 3.2 Identificación de cuatro etiquetas de los datos pertenecientes a dos conjuntos.

3.3 Método Propuesto

El método propuesto de detección de texturas busca servir como base en sistemas de consulta y recuperación de imágenes. La recuperación de imágenes de acuerdo a su contenido establece sus bases en la detección de ciertos objetos o formas con ciertas características dentro de las imágenes que estudia, en ésta se debe efectuar algún tipo de clasificación de acuerdo con el objeto o grupos de objetos que se están analizando y que al final sirve como criterio de recuperación. La clasificación que pudiera llevarse a cabo es más específica cuando se involucran texturas dentro de los mismos objetos, o incluso texturas presentes a lo largo de la imagen, como podemos encontrar en imágenes de paisajes naturales. Para la clasificación de imágenes de acuerdo a su contenido de textura, hay que tomar en cuenta que dicha textura puede estar presente en cualquier parte de la imagen incluso puede abarcar gran parte del área contenida dentro de la misma, sobre todo si tomamos en cuenta imágenes que contengan texturas naturales, como playas, montañas, bosques, etc.

Para poder llevar a cabo la clasificación de imágenes, que contengan texturas naturales, nos basamos en técnicas basadas en filtros para efectuar la extracción de características de las texturas, mezclado con técnicas de la teoría de aprendizaje estadístico para conseguir la separación de los datos de las imágenes filtradas. Con esto se busca dos cosas, primeramente que la salida del filtro aplicado a la imagen defina de manera general las características más significativas de la textura que se está analizando. Y en segundo término que dichas salidas establezcan las bases para el entrenamiento del clasificador, en donde se busca

generar un espacio de dimensión superior con base en los parámetros seleccionados.

Debido a su semejanza con el sistema visual humano [13], los filtros de Gabor fueron seleccionados para extraer las características de textura de las imágenes. Una familia de filtros es elegida para llevar a cabo el método de extracción de características, buscando obtener una salida distinta para cada tipo de textura estudiada. El clasificador que tendrá a su cargo la separación de los datos generados por los filtros son las Máquinas de Vector de Soporte debido a sus características y poder demostrado en diversas publicaciones.

3.4 Preparación de Experimentos

El conjunto de experimentos que diseñamos se hacen con el objetivo de evaluar los efectos que tiene la selección de los parámetros de los filtros en el porcentaje de error en la clasificación. El método de selección de parámetros es experimentación y resultado, se toman valores iniciales para los parámetros y se van modificando de acuerdo a los resultados obtenidos anteriormente, esto se hace de manera iterativa hasta conseguir resultados que cubran con los objetivos planteados por esta tesis.

Los experimentos ejecutados en este capítulo van encaminados a la obtención de información en dos niveles, en primera instancia se realizan experimentaciones para determinar los valores que tomarán los parámetros de los filtros de Gabor, para el filtrado de cierto tipo de texturas. En segunda instancia se realizan experimentos de clasificación para determinar el tipo de *kernel* que será utilizado, una vez definido el *kernel* a utilizar se indagan los valores de parámetro más destacados para dicho *kernel* del clasificador.

Así como se definen dos niveles de obtención de información, los experimentos se hacen en dos etapas perfectamente diferenciadas. Primero que nada es necesario obtener los valores de los parámetros de los filtros para después, con la aplicación del filtrado en las imágenes, hacer experimentos de clasificación y obtener el mejor *kernel* de la Máquina de Vector de Soporte.

Los pasos a seguir en el desarrollo de los experimentos se mencionan de manera resumida a continuación:

Primera etapa:

- Definir imágenes de entrada. Selección de las imágenes de entrada, desde contenido, información de textura, resolución, etc.
- Pre-procesamiento de las imágenes de entrada. Las imágenes de entrada deben cubrir ciertas características de acuerdo a los filtros que les serán aplicados.
- Aplicación de filtrado a las imágenes de entrada. La familia de filtros seleccionada debe aplicarse a cada imagen de entrada.

Segunda etapa:

- Selección de datos de entrenamiento y prueba. Una vez que se tienen las salidas de los filtros es necesario definir qué información será considerada para servir como entrenamiento y como prueba.
- Clasificar los datos obtenidos en el paso anterior. Los datos de entrenamiento servirán para que el clasificador "aprenda" un algoritmo que posteriormente será comprobado con los datos de prueba.
- Obtención de resultados. Del proceso clasificación se selecciona el *kernel* que ofrezca los mejores resultados.

3.4.1 Imágenes de Entrada.

Las imágenes de entrada son esenciales en cualquier sistema de clasificación, su naturaleza, contenido o resolución afectan el desempeño final del clasificador, mejorando o empeorando sus resultados. Su selección debe hacerse de acuerdo al tipo de clasificador que busquemos implementar, imágenes conteniendo cierto tipo de objetos, ya sea naturales o artificiales, imágenes de personas, rostros, paisajes, imágenes a gran escala obtenidas por satélites, o en resoluciones pequeñas obtenidas de microscopios, imágenes a color, en escala de grises, imágenes binarias, etc., cada grupo guarda ciertas propiedades que deben ser aprovechadas por el sistema clasificador.

Las imágenes seleccionadas para nuestro estudio caen en la categoría de paisajes naturales. Optamos por este tipo de imágenes ya que en ellas es más fácil encontrar algún tipo de textura natural, que es la base de nuestro estudio.

Debido a la imposibilidad de encontrar una base de datos de imágenes naturales que cubriera las necesidades de la tesis, nos dimos a la tarea de compilar en un periodo de seis meses un conjunto de fotografías de paisajes de diversas áreas naturales. Las fotografías pretendidas deben cubrir ciertas propiedades con respecto al contenido, un aspecto fundamental es que la textura que contiene debe estar presente en al menos el 50% de la imagen. Otro punto importante es la distancia focal, es decir, la distancia que existe entre el plano enfocado y la cámara que capturo la imagen, en este punto se busca mantener constante dicha distancia, al menos en la mayoría de las tomas, respecto una de la otra. La resolución ideal que se busca de las imágenes es de al menos 1024 x 768, aunque debido a las limitaciones encontradas en el proceso se permitió la inclusión

de algún porcentaje de imágenes por debajo de dicha resolución. De primera instancia se planeó trabajar con imágenes a color pero debido a las características del filtro que se aplicará a cada imagen se optó por la utilización de imágenes en escala de grises.

Aunque se encontraron múltiples páginas de Internet dedicadas a la publicación de fotografías de paisajes, la mayoría de estos sitios son de paga ya que cuentan con derechos de autor, y su uso era limitado. Consecuentemente tuvimos que recurrir a sitios de fotografías digitales que publican de manera gratuita y sin derechos de autor una gran variedad de imágenes. Sobre todo aquellos sitios en donde los usuarios del servicio son los encargados de publicar sus imágenes y que permite un uso libre de las fotografías.

Tres fuentes principales para la selección de imágenes fueron definidas, en primer lugar el servicio de búsqueda de imágenes implementado por el sitio de *Google, Inc.*, también de la misma empresa el servicio de publicación de imágenes conocido como *Picasa Web Albums*, en tercer lugar se recurrió al servicio de publicación ofrecido por *Panoramio*, este último sitio dedicado a imágenes de lugares o paisajes, por lo que fue de gran ayuda.

De un total de cinco mil fotografías se tomó la decisión de seleccionar un conjunto significativo para lograr un buen desempeño y rendimiento a la vez. Se decidió preparar seis conjuntos de imágenes de distinto contenido y que representaran características de texturas disímiles entre sí. Al final se completaron los sets de imágenes conformados por línea costera de playa, ciudad, montañas nevadas, montañas rocosas, cañones rocosos y playa tropical. Cada grupo consistió en 600 imágenes, ya que para el clasificador era necesario contar con un buen número de datos de entrada. Obviamente fue necesario establecer un tope y de acuerdo a resultados preliminares se tomó la decisión de establecer dicho número en 600, con este número se obtienen buenos resultados de clasificación sin incrementar la capacidad de procesamiento.

3.4.2 Pre-procesamiento y filtrado de Imágenes.

Una vez definidos los conjuntos de imágenes necesarios para los experimentos se procede a la etapa de pre-procesamiento y filtrado, esta etapa es fundamental dentro del proceso entero, aquí se busca tratar las imágenes de entrada para que se adapten a la configuración de los filtros y definir los parámetros que tendrá el Banco de filtros para tratar cada conjunto de imágenes.

La modificación de los valores de los parámetros proporciona al filtro la capacidad de reacción ante cierto tipo de texturas. Una vez definido el Banco se procede a la extracción de características de textura de los conjuntos de imágenes.

3.4.2.1 Etapa de pre-procesamiento.

La etapa de pre-procesamiento es de carácter esencial para la consecución de la etapa de filtrado, esto se debe a los requerimientos que exige el Banco de

Filtros, esto es, las imágenes de entrada deben cumplir con ciertas características para poder ser tomadas en cuenta, la más importante se refiere a su dimensión, ya que las imágenes deben ser de dimensión cuadrada, es decir, de tamaño $n \times n$, por lo que debemos ajustar la resolución inicial.

Con el objetivo de disminuir la complejidad computacional del método, las imágenes originales fueron re-dimensionadas a una resolución de 128×128 pixeles, tomando en cuenta que a esa resolución busca mantenerse la mayor parte de la información de textura que pudiera contener una imagen. Debe tomarse en cuenta que a esta resolución se tiene un total de 16,384 pixeles, entonces lidiamos con una buena cantidad de información para trabajar.

En algunos resultados de procesamientos preliminares se buscó trabajar con imágenes de mayor resolución, de hasta 512×512 , pero para poder siquiera ejecutar los experimentos era necesario contar con una gran capacidad de procesamiento de cómputo, por lo que al final hubo que definir un tamaño máximo de 128×128 .

Al utilizar imágenes en color de formato JPG (*Joint Photographic Experts Group*, Grupo Conjunto de Expertos en Fotografía), que se ha convertido en el método estándar de almacenamiento y transmisión de imágenes fotográficas en la *Web*, se asume que cada imagen cumple con el modelo RGB (del inglés *Red, Green, Blue*; Rojo, Verde, Azul), es decir, contiene 24 bits por cada pixel y esta compuesta por los tres canales del color, 8 bits para el Rojo, 8 bits para el Verde y 8 bits para el Azul.

Para disminuir la complejidad del proceso, se reduce el área de trabajo, obviando dos canales y tomando solo uno como entrada, es decir, transformar las imágenes del modelo RGB a escala de grises de un solo canal y con 8 bits por pixel; al utilizar un solo canal podemos reducir el número de convoluciones necesarias, lo cual permite ocuparse de una mayor cantidad de imágenes sin incrementar de manera considerable el volumen de datos.

Al final de esta etapa se tienen imágenes de menor resolución y de un solo canal, por lo que podemos proceder con el filtrado de las mismas.

3.4.2.2 Utilización de Filtros y Filtros de Gabor

Un filtro en términos generales es un elemento que discrimina cierto tipo de datos de acuerdo a un criterio deseado, en el área de procesamiento de señales, un filtro se encarga de discriminar una frecuencia o gama de frecuencias de una señal que pasa a través de él, pudiendo modificar tanto su amplitud como su fase.

En el procesamiento digital de imágenes el filtrado de una imagen es la modificación de los pixeles de una imagen basado en funciones que afectan a un pixel y a los pixeles vecinos. El filtrado de imágenes generalmente se realiza para remover ruido, matizar el contraste, realzar contornos, etcétera.

Los filtros pueden diferenciarse en dos grupos: filtros lineales y filtros basados en respuesta en frecuencia. Los filtros lineales, también conocidos como filtros de convolución, se llaman así debido que pueden representarse en una algebra lineal de multiplicación de matrices, y los filtros en frecuencia realizan operaciones sobre imágenes que han sido transformadas al dominio en frecuencia por medio de la Transformada de Fourier.

En el filtrado lineal se utiliza la operación de convolución matricial, en donde cada pixel de la imagen resultante es una combinación lineal de varios pixeles vecinos de la imagen original. Se realiza una suma ponderada de los vecinos para determinar el contenido del pixel en cuestión. Para realizar esta operación es necesario definir una máscara de convolución que se conoce como "kernel de convolución". En la ecuación 3.1 se muestra una definición de un *kernel* de convolución.

$$K[m, n] = \begin{bmatrix} k_{-a,-c} & \cdots & k_{-a,0} & \cdots & k_{-a,d} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ k_{0,-c} & \cdots & k_{0,0} & \cdots & k_{0,d} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ k_{b,-c} & \cdots & k_{b,0} & \cdots & k_{b,d} \end{bmatrix} \quad (3.1)$$

$$g[i, j] = K[m, n] * f[i, j] = \sum_{m=-a}^b \sum_{n=-c}^d K[m, n] f[i + m, j + n]$$

Normalmente se utilizan máscaras centradas para realizar la convolución entre matrices, en caso de que no estén centradas se marca el pixel central. Como se muestra en la ecuación 3.2 al ser una matriz no centrada se indica que se toma el elemento -1 como el pixel central. De forma común encontramos *kernel* como se muestra en la ecuación 3.3 en donde el valor central es 4.

$$K = \begin{bmatrix} & & -1 & & 1 & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \end{bmatrix}$$

$$g(i, j) = f(i, j + 1) - f(i, j) \quad (3.2)$$

$$K_{3 \times 3} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$$g(i, j) = \frac{1}{16} \begin{bmatrix} f(i-1, j-1) + 2f(i-1, j) + f(i-1, j+1) + \\ 2f(i, j-1) + 4f(i, j) + 2f(i, j+1) + \\ f(i+1, j-1) + 2f(i+1, j) + f(i+1, j+1) \end{bmatrix} \quad (3.3)$$

Para la aplicación de un filtro es necesario realizar la inspección de los vecinos de un pixel $p[i,j]$ y una máscara, dada la vecindad de $p[i,j]$, como se muestra en la figura 3.3, en donde los valores de $p[i,j]$ representan el tono correspondiente a la coordenada $[i,j]$ en una imagen de tono de grises, así la máscara del filtro esta dada tal como se muestra en la figura 3.4, en donde A se denomina peso de la transformación.

$p[i-1,j-1]$	$p[i,j-1]$	$p[i+1,j+1]$
$p[i-1,j]$	$p[i,j]$	$p[i,j+1]$
$p[i-1,j+1]$	$p[i,j+1]$	$p[i+1,j+1]$

Figura 3.3. Representación de la vecindad de un pixel.

$$m = A \begin{pmatrix} m[i-1,j-1] & m[i,j-1] & m[i+1,j+1] \\ m[i-1,j] & m[i,j] & m[i,j+1] \\ m[i-1,j+1] & m[i,j+1] & m[i+1,j+1] \end{pmatrix}$$

Figura 3.4 Máscara de un filtro con *kernel* de 3x3.

Al aplicar la máscara m sobre $p[i,j]$, sustituiremos el tono del pixel $p[i,j]$ por la ecuación 3.4.

$$p[i, j] \leftarrow A \sum_{j=-1}^1 \sum_{i=-1}^1 p[i, j] m[i, j] \quad (3.4)$$

A esta familia de máscaras se le llama máscaras de 3x3 por la dimensión de la matriz que las define. De igual forma se pueden construir máscaras de 5x5 y otras dimensiones.

La operación de filtrado ejecutada en esta tesis consiste básicamente en la convolución del filtro de Gabor con la imagen de entrada, de esta manera se obtiene en la salida una representación filtrada de la imagen de acuerdo a los parámetros del filtro seleccionados, como se muestra en la ecuación 3.5:

$$g(x, y) = i(x, y) \otimes \psi(x, y)$$

en donde

$g(x,y)$ es la salida del filtro (3.5)

$i(x,y)$ es la imagen original

$\psi(x,y)$ es el filtro de Gabor

El operador de convolución se utiliza para transformar las funciones $I(x,y)$ y $\psi(x,y)$ en una tercera función $G(x,y)$ que represente la magnitud de la superposición de $I(x,y)$ y una versión trasladada e invertida de $\psi(x,y)$.

Teóricamente esta operación puede realizarse en el dominio espacial, sin embargo el Filtro de Gabor es generalmente estrecho en este dominio. El filtro se vuelve más amplio en el dominio de la frecuencia y por lo tanto es menos afectado por efectos de solapamiento debido al muestreo. Es por lo tanto más conveniente hacer todo el procesamiento en el dominio de la frecuencia, en donde la convolución es reducida a una forma simple y eficiente de multiplicación punto a punto de las transformadas de Fourier, tanto de la imagen de entrada como de la definición del Filtro de Gabor [11].

Además el filtrado en el dominio de frecuencia es más rápido computacionalmente hablando, esto es, son necesarias menos operaciones al realizar dos transformaciones de Fourier y una multiplicación comparado con la convolución de una imagen con el filtro en el dominio espacial, en especial si el tamaño del filtro incrementa de alguna manera.

La salida del filtro expresada en términos del dominio de la frecuencia quedaría como se muestra en la ecuación 3.6:

$$G(u, v) = I(u, v) \cdot \Psi(u, v)$$

en donde

$G(u,v)$ es la salida del filtro (3.6)

$I(u,v)$ es la imagen transformada

$\Psi(u,v)$ es el filtro de Gabor

Para poder desarrollar el proceso descrito anteriormente es necesario que las imágenes de entrada mantengan características de compatibilidad en tamaño y propiedades conjuntamente con el diseño de los Filtros de Gabor.

Las imágenes digitales pueden definirse como una representación bidimensional de valores binarios, la imagen se representa como $I(x,y)$, en donde x e y representan una coordenada en donde encontraremos un dato de 8 bits, con valores que van desde 0 hasta 255, representando un valor de nivel de gris en el esquema RGB. Basándonos en lo expuesto, podemos asegurar que una imagen no

es más que una matriz de elementos, que para nuestro caso estamos hablando de una matriz cuadrada de 128×128 elementos.

Como se mencionó anteriormente, la representación de un filtro se hace como una matriz de tamaño regular preestablecido; generalmente se encuentran filtros de tamaño 3×3 elementos como son los filtros de la media, media ponderada, mediana, Sobel y Laplaciano.

Para el caso de un Filtro de Gabor en el dominio de la frecuencia, no aplica este tipo de representaciones, la representación de un filtro de Gabor se hace con una matriz del mismo tamaño que la matriz imagen, es decir, si la matriz imagen es de tamaño 128×128 , la matriz filtro debe ser también de 128×128 elementos.

3.4.2.3. Familia de Filtros de Gabor y Porque dominio en Frecuencia

Como se había mencionado anteriormente existe en la literatura un sinnúmero de implementaciones de filtros de Gabor tanto el dominio espacial como en dominio de la frecuencia, en esta tesis y basándonos en el material presentado en publicaciones como [11, 24, 27] hemos tomado la decisión de utilizar la familia de filtros de Gabor expresada en términos del dominio de la frecuencia, tal como se muestra en la ecuación 3.7:

$$\Psi(u, v) = e^{-\frac{\pi^2}{f^2}(\gamma^2(u'-f)^2 + \eta^2 v'^2)}$$

$$u' = u \cos \theta + v \sin \theta \quad (3.7)$$

$$v' = -u \sin \theta + v \cos \theta$$

Por medio de este filtro podremos extraer la información de textura de cada grupo de imágenes, esto se logra conformando un Banco de Filtros y realizando la convolución de cada imagen con el Banco.

La construcción del Banco de Filtros se realiza modificando valores de cuatro de los parámetros principales que definen al filtro [26]. Las combinaciones de valores de estos parámetros determinan la precisión con la cual el Banco analiza las imágenes de entrada. La tabla 3.1 muestra la lista de los parámetros principales que se toman en cuenta para la conformación del Banco de filtros.

Parámetro	Descripción
f	Frecuencia Central del filtro
θ (theta)	Angulo de rotación
γ (gamma)	Nitidez en el eje principal
η (eta)	Nitidez en el eje secundario

Tabla 3.1 Lista de parámetros del filtro de Gabor.

Los valores de frecuencia y orientación son definidas, generalmente, tomando en cuenta el tipo de texturas que van a analizarse. Los valores γ y η establecen la forma que tendrá la superficie Gaussiana y en la mayoría de los casos se mantienen constantes. También se acostumbra definir el tamaño de dicha superficie para que sus orillas estén en contacto, tal como se muestra en la figura 3.5.

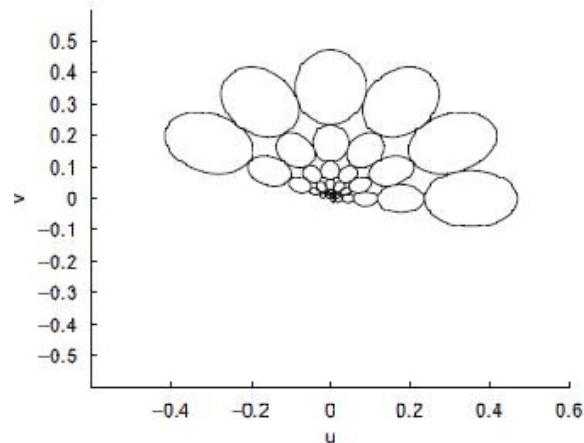


Figura 3.5 Representación de una familia de filtros de Gabor en donde las orillas están en contacto unas con otras.

Lo que se busca con este tipo de diseño es la minimización de la superposición entre filtros adyacentes, al contrario de como se muestra en la figura 3.6, en donde existe superposición de los filtros [26].

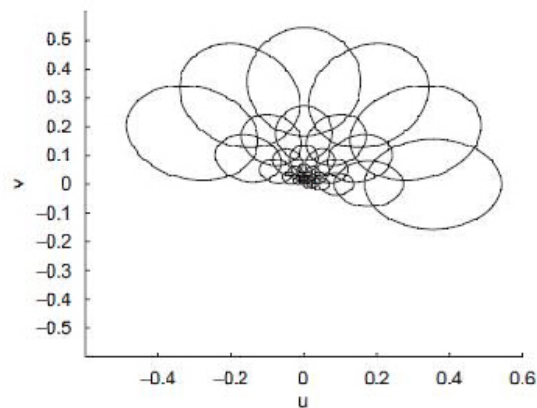


Figura 3.6 Representación de una familia de filtros de Gabor con superposición en sus orillas.

Se cree que disminuyendo el efecto de superposición entre los distintos filtros del Banco se obtiene un beneficio en el proceso de detección de texturas. Sin embargo esta suposición no ha sido respaldada por evidencia experimental

alguna, ya que el método de disminución de superposición no ha sido comparado con otros métodos.

En la literatura, los parámetros de aspecto espacial generalmente reciben menor atención [26], comparado con lo que es la frecuencia y la orientación, ya que estos últimos definen la amplitud de desempeño del filtro.

Basados en resultados fisiológicos, se cree que una relación η/γ de aproximadamente 2/3 será suficiente para mejorar la discriminación de texturas. Aun así los datos existentes en la literatura no permiten comparación entre los efectos de distintos valores de estos parámetros en la discriminación de texturas. Para esta tesis hemos establecido constantes los valores de η y γ en los experimentos propuestos.

También en [26] se hace hincapié en algunos Bancos de Filtros propuestos en varios artículos, cada uno planteado para alguna aplicación diferente, por lo que se tienen valores disímiles. En todas ellas se acepta de manera general que los parámetros que más influyen en la precisión al momento de clasificar texturas son: la frecuencia central del filtro en su valor máximo (F_M), el número total de frecuencias (n_F) y el número total de orientaciones (n_O). En la tabla 3.2 se hace un listado de algunos valores tomados por distintos autores en sus experimentos.

Autor	n_F	n_O	F_M
Turner (1986)	4	4	$1/4$
Jain and Farrokhnia (1991)	7	4	$\sqrt{2} / 4$
Manjunath and Ma (1996)	4	6	0.4
Jain et. al. (1997)	5	4	$\sqrt{2} / 4$
Kruizinga and Petkov (1999)	3	8	$1/5.47$
Rubner (1999)	4	6	0.3
Li and Shawe-Taylor (2004)	4 – 6	4 – 6	0.4
Clausi and Deng (2005)	4	4	$\sqrt{2} / 4$

Tabla 3.2 Valores de parámetros tomados por algunos autores de acuerdo a distintas publicaciones.

Es precisamente el número de frecuencias y de orientaciones las que van a determinar el tamaño del Banco de filtros. Entre mayor sea el número de frecuencias y orientaciones, el Banco tendrá capacidad de responder ante una mayor cantidad de texturas en una sola pasada.

En términos generales han existido una gran cantidad de enfoques al momento de trabajar con Bancos de Filtros de Gabor, cada uno de acuerdo al tipo de aplicación a la cual están pensados, sin embargo todavía no podemos llegar a un enfoque que pueda aplicarse de manera general para cualquier aplicación.

En esta tesis se define un Banco de filtros con dos frecuencias y tres orientaciones, se buscó mantener reducido el tamaño del Banco y por ende,

mantener reducido el número de operaciones necesarias para el filtrado de las imágenes.

3.4.2.4 Parámetros de los Filtros y Convolución con las Imágenes

La construcción de nuestro Banco de filtros esta basado en los valores obtenidos en los experimentos realizados, y que nos ayudan a obtener los valores más precisos de los parámetros del filtro de acuerdo con cada set de imágenes que tenemos. Intentando proponer un Banco de Filtros que fuera lo más generalizado posible se busco modificar únicamente los valores de las frecuencias y mantener de manera constante los valores de las orientaciones, esto nos ofrece la oportunidad de aumentar la velocidad en el procesamiento y respuesta de los filtros, además de hacer el procedimiento de experimentación más ágil.

El parámetro de orientación (θ) en el Filtro de Gabor reacciona ante cambios de dirección en las imágenes, visualizado de mejor manera en texturas homogéneas. Al tener imágenes de paisajes naturales se tiene una gran cantidad de texturas direccionadas con las cuales el filtro pudiera reaccionar de cierta manera, por lo que optamos por mantener constante este valor y apoyar la capacidad del filtro con combinaciones con los demás parámetros.

La naturaleza del filtro de Gabor seleccionado nos exige que las imágenes de entrada sean transformadas al dominio de la frecuencia. Para llevar a cabo esta tarea recurrimos a la Transformada de Fourier. La Transformada de Fourier en dos dimensiones que se utiliza para las imágenes puede definirse de acuerdo a la ecuación 3.8:

$$I(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} i(x, y) e^{-i2\pi(ux+vy)} dx dy$$

en donde (3.8)

$i(x,y)$ es la imagen de entrada original

El filtrado en el dominio de la frecuencia permite mayor flexibilidad al ser posible seleccionar no solo la dirección del filtrado, sino también los intervalos de frecuencia que requieran ser eliminados. Además de permitir la separación de varias señales en caso de ser necesario.

La operación de filtrado, como se mencionó anteriormente consiste en la convolución de la imagen de entrada con el Banco de Filtros. En el dominio de la frecuencia la convolución se representa como una multiplicación punto a punto entre el filtro y la imagen de entrada, ambas en su representación de dominio de frecuencia.

$$\begin{array}{ll}
g(x, y) = i(x, y) \otimes \psi(x, y) & \text{Dominio Espacial} \\
G(u, v) = I(u, v) \cdot \Psi(u, v) & \text{Dominio en Frecuencia}
\end{array} \quad (3.9)$$

El filtro de Gabor de la ecuación 2.5 puede ser usada para extraer características de textura de las imágenes. La extracción de características se lleva a cabo a través de la convolución en la forma:

$$G(u, v; f, \theta) = I(u, v) \cdot \Psi(u, v; f, \theta) \quad (3.10)$$

al estar modificando solo los valores de frecuencia y orientación, son los que inciden directamente en el filtro de Gabor y por ende en la imagen filtrada. Al aplicar la operación anterior obtenemos una matriz resultante que tendrá la información de la imagen filtrada.

Para completar el proceso de convolución se toma solamente la parte real de la imagen filtrada, esto nos permite mantener la mayor parte de la respuesta del filtro, ignorando la información de fase [31], de esta manera aún trabajando con solo una parte de la información generada por el filtro, podemos asegurar que se cuenta con la parte más significativa.

Al final del proceso de convolución se tiene una cantidad de matrices igual a la multiplicación del número de frecuencias y el número de orientaciones, que para nuestro caso son dos frecuencias y tres orientaciones, por lo que se tiene un total de seis matrices de salida por cada imagen de entrada, generando 3,600 matrices de salida por cada conjunto de imágenes. Al utilizar un clasificador binario, éste nos exige trabajar con dos conjuntos de imágenes en cada proceso, es decir, debemos mantener 7,200 matrices accesibles en cualquier momento, lo que conlleva a requerir un gran espacio de almacenamiento en memoria de acceso aleatorio.

En su lugar hemos optado por realizar una suma de las matrices de salida de cada imagen, resaltando la respuesta del filtro de acuerdo a cada textura presente. Esta nueva modalidad nos permite trabajar con solo 600 matrices, una por cada imagen, permitiéndonos hacer experimentos con tan solo 1,200 matrices, 15% del número de matrices requeridas en un principio, agilizando el proceso y proporcionando resultados en menor tiempo.

3.4.2.5 Conformar la Matriz de Características

Una vez que se aplica el filtro al conjunto de imágenes en el dominio de la frecuencia, es necesario transformarlo de vuelta al dominio espacial para que puedan servir como datos de entrada en el clasificador. En este punto solo basta

aplicar la Transformada Inversa de Fourier a las matrices de salida, por medio de la fórmula:

$$i(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(u, v) e^{i2\pi(ux+vy)} du dv$$

en donde (3.11)

I(u,v) es la imagen en dominio de frecuencia

Después de la Transformada Inversa se aplica normalización a las salidas, esto se hace con el fin de disminuir los efectos que pudiera provocar las distintas iluminaciones.

La matriz resultante es la que llamamos Matriz de Características, en lugar de tener un vector de características. Esta matriz representa la respuesta del filtro con los parámetros que sean seleccionados.

Las matrices obtenidas en el paso anterior serán modificadas con el fin de conformar la matriz de datos de entrada para alimentar al clasificador. El proceso de modificación de las matrices consiste en re-dimensionar una matriz de su tamaño original 128×128 elementos, en un vector de 16384 elementos.

Los vectores resultantes se irán apilando en un nueva matriz, cuya dimensión se condiciona al número de imágenes de entrada. Los pasos se explican visualmente a continuación:

1. Tenemos la salida del filtro que es de tamaño 128×128

$$\begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} & \cdots & x_{1,j} \\ x_{2,1} & x_{2,2} & x_{2,3} & \cdots & x_{2,j} \\ x_{3,1} & x_{3,2} & x_{3,3} & \cdots & x_{3,j} \\ \vdots & \vdots & \vdots & & \vdots \\ x_{i,1} & x_{i,2} & x_{i,3} & \cdots & x_{i,j} \end{bmatrix} \quad 128 \times 128$$

2. Tomamos el primer renglón de la matriz y la ponemos al inicio de lo que será nuestro nuevo vector.

$$\begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} & \cdots & x_{1,j} \end{bmatrix}$$

3. Tomamos el segundo renglón de nuestra matriz y lo ponemos a continuación del elemento anterior dentro del nuevo vector.

$$\begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} & \cdots & x_{1,j} & x_{2,1} & x_{2,2} & x_{2,3} & \cdots & x_{2,j} \end{bmatrix}$$

4. Así sucesivamente tomamos cada renglón de la matriz y lo transformamos en un vector que al final será de tamaño 16384.

$$\left[x_{1,1} \quad x_{1,2} \quad x_{1,3} \quad \dots \quad x_{1,j} \quad x_{2,1} \quad x_{2,2} \quad x_{2,3} \quad \dots \quad x_{2,j} \quad x_{3,1} \quad x_{3,2} \quad x_{3,3} \quad \dots \quad x_{3,j} \quad \dots \right]$$

5. A cada imagen se le aplica el mismo procedimiento, obteniendo un vector por cada imagen filtrada.

$$\left[x_{1,1} \quad x_{1,2} \quad x_{1,3} \quad \dots \quad x_{1,j} \quad x_{2,1} \quad x_{2,2} \quad x_{2,3} \quad \dots \quad x_{2,j} \quad x_{3,1} \quad x_{3,2} \quad x_{3,3} \quad \dots \quad x_{3,j} \quad \dots \right]$$

$$\left[x_{1,1} \quad x_{1,2} \quad x_{1,3} \quad \dots \quad x_{1,j} \quad x_{2,1} \quad x_{2,2} \quad x_{2,3} \quad \dots \quad x_{2,j} \quad x_{3,1} \quad x_{3,2} \quad x_{3,3} \quad \dots \quad x_{3,j} \quad \dots \right]$$

$$\left[x_{1,1} \quad x_{1,2} \quad x_{1,3} \quad \dots \quad x_{1,j} \quad x_{2,1} \quad x_{2,2} \quad x_{2,3} \quad \dots \quad x_{2,j} \quad x_{3,1} \quad x_{3,2} \quad x_{3,3} \quad \dots \quad x_{3,j} \quad \dots \right]$$

6. Estos vectores se apilan uno sobre otro hasta formar una nueva matriz que llamamos matriz madre cuyo tamaño por cada par de conjuntos es de 1200x16384

$$\begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} & \dots & x_{1,j} & x_{2,1} & x_{2,2} & x_{2,3} & \dots & x_{2,j} & x_{3,1} & x_{3,2} & x_{3,3} & \dots & x_{3,j} & \dots \\ x_{1,1} & x_{1,2} & x_{1,3} & \dots & x_{1,j} & x_{2,1} & x_{2,2} & x_{2,3} & \dots & x_{2,j} & x_{3,1} & x_{3,2} & x_{3,3} & \dots & x_{3,j} & \dots \\ x_{1,1} & x_{1,2} & x_{1,3} & \dots & x_{1,j} & x_{2,1} & x_{2,2} & x_{2,3} & \dots & x_{2,j} & x_{3,1} & x_{3,2} & x_{3,3} & \dots & x_{3,j} & \dots \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots & \\ x_{1,1} & x_{1,2} & x_{1,3} & \dots & x_{1,j} & x_{2,1} & x_{2,2} & x_{2,3} & \dots & x_{2,j} & x_{3,1} & x_{3,2} & x_{3,3} & \dots & x_{3,j} & \dots \end{bmatrix}$$

Para nuestros experimentos la matriz de datos que obtenemos tiene un tamaño de 1200x16384 elementos, ya que al ser un clasificador binario es necesario ir cargando los grupos de imágenes en pares.

Una vez completado el procesamiento y filtrado de imágenes, la matriz obtenida contiene la información de las imágenes filtradas, el siguiente paso consiste en hacer experimentaciones con el clasificador para obtener los parámetros adecuados de acuerdo al tipo de textura que intentamos separar.

CAPITULO IV. EXPERIMENTOS CON EL CLASIFICADOR SVM

En este capítulo se ahonda un poco más en la definición de las Máquinas de Vector de Soporte, los distintos *kernels* que pueden manejarse y las características y parámetros de cada uno de ellos. Se preparan los experimentos para poder seleccionar el *kernel* adecuado de acuerdo al conjunto de datos que buscamos separar y en base a los resultados obtenidos se definen los valores de parámetros para los Filtros de Gabor.

Como se ha observado en distintas publicaciones, la selección del *kernel* y los parámetros del mismo en la clasificación SVM es una tarea difícil ya que depende del tipo de datos que quiera clasificarse, en la mayoría de los casos se utiliza el método de ensayo y error, en la cual se parte de una selección arbitraria de parámetros y se van modificando de acuerdo a los resultados parciales obtenidos.

4.1 Selección de datos de Entrenamiento y Prueba.

El objetivo de los experimentos en esta etapa es obtener un modelo de entrenamiento a través de las Máquinas de Vector de Soporte, el cual puede ser capaz de separar los datos de dos conjuntos de imágenes de entrada previamente filtradas.

La sensibilidad del clasificador se ve influenciada por el tipo de texturas que se estará intentando separar. Al ser las Máquinas de Vector de Soporte un clasificador binario por naturaleza es necesario diseñar procesos de experimentación que permitan comparar conjuntos de imágenes en pares. El proceso requiere de seleccionar un conjunto de datos y compararlos con cada uno de los demás conjuntos de datos respectivamente, así sucesivamente hasta completar los seis conjuntos iniciales.

De acuerdo con la esencia del proceso de clasificación, existen dos fases bien diferenciadas como consecuencia de hacer una división de los datos. El tratamiento consiste en separar los datos de entrada en dos conjuntos: datos de entrenamiento y datos de prueba.

El primer grupo define al conjunto de datos de entrenamiento, estos datos servirán para que el clasificador "aprenda" un modelo de clasificación, el segundo grupo define al conjunto de datos de prueba, estos datos permiten comprobar la capacidad de clasificación del modelo aprendido en la fase anterior. Al conformar el tamaño de cada conjunto debe tomarse en cuenta que no deben ser del mismo tamaño, esto se hace con el objeto de evitar sobreajuste en los datos.

El sobreajuste, en estadística, ocurre cuando un modelo estadístico describe un error aleatorio o ruido en lugar de una relación subyacente entre un conjunto de datos. Generalmente ocurre cuando un modelo es excesivamente complejo,

como puede ser cuando se tienen muchos grados de libertad en relación con la cantidad de datos disponibles. Un modelo sobreajustado generalmente tendrá un desempeño predictivo muy bajo, ya que puede exagerar pequeñas fluctuaciones en los datos.

Para efectos prácticos, hemos establecido que el conjunto de datos de entrenamiento este conformado por el 75% de los datos de entrada, y el restante 25% conforma el conjunto de datos de prueba.

4.2 Experimentación SVM

Los experimentos realizados en esta etapa del proceso están encaminados a la obtención del *kernel* que clasifique de mejor manera los datos de entrada y una vez definido el *kernel*, la obtención del mejor parámetro que ayude en la tarea según el tipo de texturas que se desea separar. En el capítulo anterior hemos obtenido una Matriz Madre por cada set de imágenes, por lo que se tiene realmente 6 matrices madre.

Los *kernel* de Máquinas de Vector de Soporte disponibles en la implementación SPIDER son:

- Lineal,
- Polinomial,
- RBF (Función de Base Radial)
- Gaussiano.

Una explicación breve de las funciones de cada *kernel* se muestra a continuación [47]:

4.2.1 *Kernel* Lineal.

La función esta definida como se muestra en la ecuación 4.1:

$$k(x, y) = x * y \quad (4.1)$$

El *kernel* lineal es el más sencillo de todos, y muestra un buen desempeño para datos que se sabe son linealmente separables. De manera sorprendente trabaja muy bien para casos de datos no lineales.

4.2.2 *Kernel* Polinomial.

La función *kernel* se define como en la ecuación 4.2:

$$k(x, y) = (x * y + 1)^d \quad (4.2)$$

En la ecuación 4.2 observamos que ' d ' es el grado del polinomio. La explicación es que, en general, para los vectores x que son dependientes linealmente en d dimensiones, la función *kernel* de orden d puede ser usada para

transformarlos en vectores linealmente independientes en esas d dimensiones. Una vez que son transformados en el espacio de dimensión en donde pueden ser separados linealmente, se utiliza el *kernel* lineal para llevar a cabo la separación. Lo que es más, en cierto sentido, éste es una extensión del *kernel* lineal, por el hecho de que proporciona la transformación necesaria para permitir la independencia entre los datos de entrenamiento. El desempeño de este *kernel* es similar al que tiene el *kernel* lineal, ya que el principio detrás de ambos es el mismo y la transformación solo se hace para llevarlos a un espacio diferente. De cualquier manera, el desempeño depende del orden d en el polinomial, ya que de él depende que tan bien los datos se vuelvan separables.

4.2.3 *Kernel* de Función Radial

En este rubro tenemos dos implementaciones, por un lado el *Kernel* de Función Radial normal, y el *Kernel* de Función Radial Gaussiano. El RBF normal se define en la ecuación 4.3, y el *kernel* RBF Gaussiano en la ecuación 4.4.

$$k(x, y) = e\left(\frac{-|x - y|^2}{2\sigma^2}\right) \quad (4.3)$$

$$k(x, y) = \frac{1}{2\pi^{N/2} * \sqrt{\sigma}} \cdot e\left(\frac{-|x - y|^2}{2\sigma^2}\right) \quad (4.4)$$

Este *kernel* es más adecuado para trabajar con datos que tienen una función de distribución de probabilidad de clase condicional cercana a la distribución Gaussiana. Mapea los datos a diferentes espacios en donde puedan ser linealmente separables. Para poder visualizar esto, es conveniente observar que el *kernel* (que es de naturaleza exponencial) puede ser expandida en una serie infinita, dando cabida a un *kernel* polinomial de dimensión infinita. Cada uno de estos *kernels* polinomiales serán capaces de transformar ciertas dimensiones para hacerlas linealmente separables.

Naturalmente uno esperaría que el RBF tuviera un mejor rendimiento que el *kernel* lineal o el polinomial. Como sea, este *kernel* es difícil de diseñar, en el sentido de que es difícil obtener un valor óptimo de sigma y escoger adecuadamente la C que funcione mejor para un problema determinado. El hecho de que ciertas combinaciones de sigma y C hagan que la SVM sea más sensible a los datos de entrenamiento también contribuye al factor de error de la SVM con *kernel* RBF.

Una de las ventajas del *kernel* RBF es que una vez que se define el *kernel*, los pesos a_i , el número de vectores de soporte N_s y los vectores de soporte "sí" son obtenidos automáticamente como parte del proceso de entrenamiento, es decir, no

es necesario que se especifiquen en el mecanismo de entrenamiento. En la tabla 4.1 se muestran los *kernel* analizados con sus respectivos parámetros.

4.3 Selección del *Kernel*

La primera parte de los experimentos consiste en determinar cuál *kernel* funciona de mejor manera para hacer la separación de los datos, que no consuma tiempo excesivo de procesamiento y que ofrezca el error de clasificación más bajo.

Para llevar a cabo esta parte de los experimentos, se toman tres conjuntos de imágenes y se realizan pruebas de clasificación de cada una contra las demás. Al utilizar una técnica de clasificación supervisada es necesario etiquetar los conjuntos de acuerdo al grupo al que pertenecen.

Kernel	Parámetro	Fórmula
Lineal	n/a	$k(x, y) = x * y$
Polinomial	d, dimensión	$k(x, y) = (x * y + 1)^d$
RBF (Función de Base Radial)	sigma	$k(x, y) = e^{\left(\frac{- x - y ^2}{2\sigma^2}\right)}$
RBF Gaussiano	sigma	$k(x, y) = \frac{1}{2\pi^{N/2} * \sqrt{\sigma}} \cdot e^{\left(\frac{- x - y ^2}{2\sigma^2}\right)}$

Tabla 4.1 Listado de *kernel* soportados por SPIDER con su respectivo parámetro.

Se tienen 600 imágenes por cada tipo de textura. Cada imagen esta representada como un renglón de la Matriz Madre correspondiente, para cada experimento se toma un par de conjuntos. Para etiquetar los elementos de pertenencia se utilizan "1" y "-1"; los datos con etiqueta "1" serán considerados como pertenecientes a un conjunto, y los datos con etiqueta "-1" serán considerados como no pertenecientes.

A manera de ejemplo se toman dos conjuntos de imágenes: Playa y Ciudad, cada uno representado por una Matriz Madre. Se etiquetan "1" a los elementos que pertenecen a Playa y como "-1" a los que no pertenecen, es decir a los elementos del conjunto Ciudad. De acuerdo a pruebas experimentales previas se tomó la decisión de seleccionar los elementos que serán etiquetados de manera lineal, ya que no existe un cambio significativo en los resultados finales según el método de selección de los datos, ya fuera de manera aleatoria o lineal.

El conjunto de entrenamiento se forma tomando el 75% de los datos de ambas matrices, lo que corresponde a 450 imágenes por cada una, o para nuestro caso, 450 renglones; se toman los primeros 450 elementos de la Matriz Playa y se etiquetan como "1", a continuación se toman los primeros 450 elementos de la

Matriz Ciudad y se etiquetan como "-1". Nuestro conjunto de entrenamiento tiene 900 imágenes pertenecientes a dos clases.

El conjunto de prueba se forma de manera similar, la diferencia radica en que se toma el 25% de los datos, correspondiente a 150 imágenes. Se toma el resto de los imágenes de la Matriz Playa que no se tomaron en el paso anterior, y se etiquetan como "1", después se toma el resto de las imágenes que no se tomaron de la Matriz Ciudad y se etiquetan como "-1". El conjunto de prueba queda entonces con 300 imágenes. Es necesario realizar estos pasos por cada conjunto de imágenes con el cual se realizarán los experimentos.

Para verificar la precisión del modelo de clasificación fue necesario establecer valores arbitrarios para los parámetros de cada *kernel* de clasificación. Se realizan de manera iterativa experimentos con cada tipo de *kernel* disponible en la implementación SPIDER. Por facilidad de interpretación se etiquetan los conjuntos como S1, S2 y S3. Los resultados de esta primera etapa de muestran a continuación. El primer *kernel* que se tomo en consideración fue el polinomial.

S1 vs S2

KERNEL POLINOMIAL	
Grado	Error
d = 1	0.3021
d = 2	0.5052
d = 3	0.4948
d = 4	0.5
d = 5	0.5
d = 6	0.5
d = 7	0.5
d = 8	0.5
d = 9	0.5
d=10	0.5

S1 vs S3

KERNEL POLINOMIAL	
Grado	Error
d = 1	0.4161
d = 2	0.4950
d = 3	0.4990
d = 4	0.5
d = 5	0.5
d = 6	0.5
d = 7	0.5
d = 8	0.5
d = 9	0.5
d=10	0.5

S2 vs S3

KERNEL POLINOMIAL	
Grado	Error
d = 1	0.3630
d = 2	0.4470
d = 3	0.4810
d = 4	0.4978
d = 5	0.5
d = 6	0.5
d = 7	0.5
d = 8	0.5
d = 9	0.5
d=10	0.5

El Segundo *kernel* con el cual se realizaron experimentos es el *kernel*/RBF.
S1 vs S2

KERNEL RBF	
Sigma	Error
$\sigma = 1$	0.5
$\sigma = 2$	0.5
$\sigma = 3$	0.5
$\sigma = 4$	0.5
$\sigma = 5$	0.49
$\sigma = 6$	0.4792
$\sigma = 7$	0.4323
$\sigma = 8$	0.3542
$\sigma = 9$	0.3542
$\sigma = 10$	0.3698

S1 vs S3

KERNEL RBF	
Sigma	Error
$\sigma = 1$	0.5
$\sigma = 2$	0.5
$\sigma = 3$	0.5
$\sigma = 4$	0.5
$\sigma = 5$	0.4427
$\sigma = 6$	0.4115
$\sigma = 7$	0.3646
$\sigma = 8$	0.3333
$\sigma = 9$	0.3385
$\sigma = 10$	0.3042

S2 vs S3

KERNEL RBF	
Sigma	Error
$\sigma = 1$	0.5
$\sigma = 2$	0.5
$\sigma = 3$	0.5
$\sigma = 4$	0.5
$\sigma = 5$	0.4479
$\sigma = 6$	0.4167
$\sigma = 7$	0.3594
$\sigma = 8$	0.3333
$\sigma = 9$	0.3385
$\sigma = 10$	0.3646

El tercer *kernel* analizado es el Gaussiano
S1 vs S2

KERNEL GAUSSIANO	
Sigma	Error
$\sigma = 1$	0.5
$\sigma = 2$	0.5
$\sigma = 3$	0.5
$\sigma = 4$	0.5
$\sigma = 5$	0.5
$\sigma = 6$	0.5
$\sigma = 7$	0.5
$\sigma = 8$	0.5
$\sigma = 9$	0.5
$\sigma = 10$	0.5

S1 vs S3

KERNEL GAUSSIANO	
Sigma	Error
$\sigma = 1$	0.5
$\sigma = 2$	0.5
$\sigma = 3$	0.5
$\sigma = 4$	0.5
$\sigma = 5$	0.5
$\sigma = 6$	0.5
$\sigma = 7$	0.5
$\sigma = 8$	0.5
$\sigma = 9$	0.5
$\sigma = 10$	0.5

S2 vs S3

KERNEL GAUSSIANO	
Sigma	Error
$\sigma = 1$	0.5
$\sigma = 2$	0.5
$\sigma = 3$	0.5
$\sigma = 4$	0.5
$\sigma = 5$	0.5
$\sigma = 6$	0.5
$\sigma = 7$	0.5
$\sigma = 8$	0.5
$\sigma = 9$	0.5
$\sigma = 10$	0.5

Después de la primera etapa de experimentación es notable la incapacidad del *kernel* Gaussiano para separar los datos de entrada. Por lo que no es tomado en cuenta para la siguiente iteración. Por parte del *kernel* polinomial muestra resultados significativos solamente para valores de grado muy pequeños y al comparar los resultados de clasificación con respecto del *kernel* RBF se nota una diferencia significativa, por lo que el *kernel* polinomial es descartado para la segunda iteración. Además es notorio, en el *kernel* RBF, que conforme se aumenta el valor de sigma, el porcentaje de error comienza a disminuir.

Por lo anterior, la segunda iteración de esta etapa consiste en extender los resultados de clasificación del *kernel* RBF, principalmente, debido a que presenta mejores resultados comparados con los otros dos *kernel*.

Se busca incrementar el valor de sigma debido al comportamiento detectado en los experimentos que involucran al *kernel* RBF que permite disminuir el error de clasificación conforme se incrementa dicho valor. De acuerdo a lo observado, los otros *kernels* estudiados no tienen el mismo comportamiento mientras se incrementa el valor de sigma, lo que es más, mientras se realiza esto puede observarse que el error de clasificación se mantiene constante en 0.5 sin importar el valor que se le siga proporcionado al sigma.

Debido a los resultados significativos obtenidos con RBF se decide seguir realizando experimentación con valores distintos de sigma tan solo con este *kernel*. Los resultados obtenidos se presentan a continuación:

S1 vs S2

KERNEL RBF	
Sigma	Error
$\sigma = 21$	0.1406
$\sigma = 22$	0.1094
$\sigma = 23$	0.1094
$\sigma = 24$	0.1094
$\sigma = 25$	0.1094
$\sigma = 26$	0.1094
$\sigma = 27$	0.1094
$\sigma = 28$	0.1094
$\sigma = 29$	0.1094
$\sigma = 30$	0.1094
$\sigma = 31$	0.1094
$\sigma = 32$	0.1094
$\sigma = 33$	0.1094
$\sigma = 34$	0.1094
$\sigma = 35$	0.0937
$\sigma = 36$	0.0937

S1 vs S3

KERNEL RBF	
Sigma	Error
$\sigma = 21$	0.1509
$\sigma = 22$	0.1509
$\sigma = 23$	0.1143
$\sigma = 24$	0.1143
$\sigma = 25$	0.1143
$\sigma = 26$	0.1201
$\sigma = 27$	0.1143
$\sigma = 28$	0.1018
$\sigma = 29$	0.1018
$\sigma = 30$	0.1018
$\sigma = 31$	0.1018
$\sigma = 32$	0.1018
$\sigma = 33$	0.1018
$\sigma = 34$	0.1018
$\sigma = 35$	0.1018
$\sigma = 36$	0.0902

S2 vs S3

KERNEL RBF	
Sigma	Error
$\sigma = 21$	0.1044
$\sigma = 22$	0.0923
$\sigma = 23$	0.0923
$\sigma = 24$	0.0923
$\sigma = 25$	0.0923
$\sigma = 26$	0.0923
$\sigma = 27$	0.0923
$\sigma = 28$	0.0875
$\sigma = 29$	0.0875
$\sigma = 30$	0.0875
$\sigma = 31$	0.0875
$\sigma = 32$	0.0799
$\sigma = 33$	0.0799
$\sigma = 34$	0.0799
$\sigma = 35$	0.0799
$\sigma = 36$	0.0799

Después de esta segunda iteración se observa una mejora en la reducción del error de clasificación, vemos porcentajes de error que no pueden obtenerse con ningún otro *kernel*. Por lo que se toma la decisión de utilizar el *kernel* RBF para llevar a cabo los experimentos de clasificación en esta tesis.

4.4 Valores de Sigma para el Clasificador SVM y Frecuencia Central

Una vez determinado el mejor *kernel* de acuerdo al tipo de imágenes que tenemos, hacemos experimentaciones para determinar el mejor valor del parámetro del *kernel* así como el valor de la frecuencia central del filtro de Gabor que responda de mejor manera al clasificador.

Los valores de sigma y de la frecuencia central serán específicos para cada textura y serán determinados de la misma manera que el procedimiento anterior: experimentación y prueba. Se toman valores iniciales de sigma y de la frecuencia central y se van modificando de acuerdo a los resultados en la clasificación.

Partiendo de los resultados obtenidos en el apartado anterior se seleccionan valores similares de sigma para comenzar con los experimentos. Iterativamente se van modificando dichos valores hasta llegar a errores de clasificación mínimos.

Los experimentos de esta etapa incluyen los seis conjuntos de imágenes previamente tratados, se toma un conjunto y se comparan contra los demás conjuntos, así para cada grupo de texturas. Los resultados finales se muestran en el siguiente capítulo.

CAPÍTULO V. RESULTADOS

Este capítulo sirve para la presentación de resultados obtenidos durante el proceso final de experimentación de acuerdo a los parámetros obtenidos en el capítulo cuarto.

5.1 Frecuencia Central

El proceso para determinar los valores de la frecuencia central que corresponden de mejor manera a un tipo de textura se basa en la utilización de un valor base de frecuencia y de forma iterativa realizar experimentos de clasificación con distintos valores de sigma del *kernel* RBF. Para cada grupo de imágenes se define un valor de frecuencia base y se hacen clasificaciones contra los demás grupos de imágenes.





Al final de las iteraciones programadas, se toma el valor de frecuencia que haya obtenido el error de clasificación más bajo, tomando en cuenta los resultados obtenidos del grupo actual contra los demás grupos de imágenes.

Es entonces que se define un valor de frecuencia central para cada tipo de textura estudiada, en la tabla 5.1 se muestran los resultados finales en el proceso de selección de la frecuencia central del filtro de Gabor. Estos valores fueron seleccionados de acuerdo a resultados.

Una vez obtenidos los valores de frecuencia, se procede a la última parte de los experimentos, con los parámetros de filtro de Gabor completos debemos definir el valor de sigma del *kernel* de RBF que mejor clasifique los grupos de imágenes.

Los valores de frecuencia presentados en la tabla se usan para realizar el filtrado de las imágenes que servirán como entrada al clasificador. Del proceso de clasificación se obtienen los siguientes resultados.

Etiqueta	Descripción	Frecuencia Central	Imagen Ejemplo
S1	Cañones Rocosos	0.0950	

S2	Playa Tropical	0.0695	
S3	Línea Costera	0.2115	
S4	Ciudad	0.2162	
S5	Formaciones Rocosas	0.1350	


S6	Montañas Nevadas	0.0891	
----	---------------------	--------	--

Tabla 5.1 Conjuntos de imágenes con su respectiva Frecuencia Central.

5.2 Errores de Clasificación

5.2.1. Set S1 Cañones Rocosos

El tipo de textura que busca analizarse corresponde con cañones que presenten algún elemento de horizonte y que contenga el tipo de roca que normalmente se encuentra en los cañones, esto es, terreno irregular con grietas de tamaño considerable y veredas que abren camino a corrientes de ríos. La siguiente imagen muestra los resultados comparativos de este grupo con respecto de los demás.

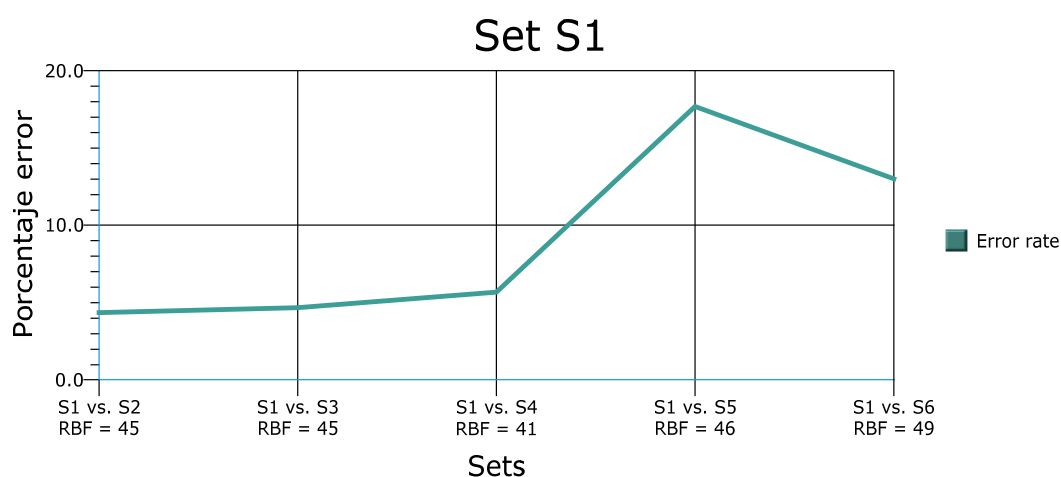


Figura 5.1 Comparativo de porcentaje de errores obtenidos en fase de experimentación del conjunto Cañones Rocosos.

Podemos apreciar que el mayor porcentaje de error de clasificación lo tiene con el S5 Formaciones Rocosas, esto se debe a que comparten el tipo de textura que generalmente tiene el suelo en estas regiones, aun así tenemos un error de 17.6%, lo cual consideramos bastante bueno.

5.2.2 Set S2 Playa Tropical

El contenido de este grupo de imágenes incluye una línea costera pero acompañada por algún tipo de vegetación, lo que ofrece una composición de textura distinta y que se vuelve un factor determinante para la diferencia. En la gráfica siguiente se muestran los resultados de clasificación.

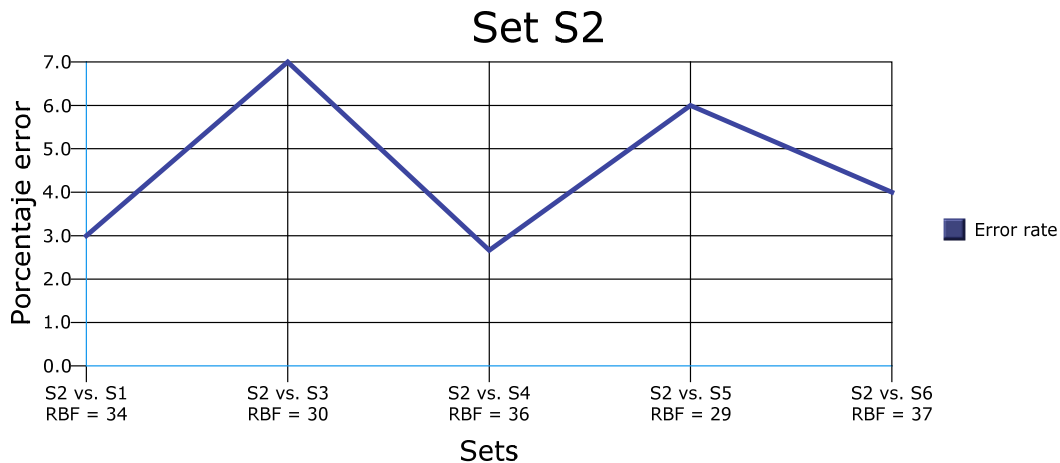


Figura 5.2 Comparativo de porcentaje de errores obtenidos en fase de experimentación del conjunto Playa Tropical.

El mayor porcentaje de error se observa al comparar este conjunto con las imágenes de línea costera, debido al hecho de compartir elementos contenidos en las mismas, recordemos que la diferencia entre estos conjuntos en la presencia de vegetación en una de ellas. Un 7% en el error de clasificación se vuelve excelente para los alcances de este trabajo.

5.2.3 Set S3 Línea Costera

El conjunto Línea Costera contiene escenas de playa desde perspectivas cercanas, esto es, una parte de arena, una parte de mar y en algunas ocasiones una parte de ciudad debido a la distribución del puerto de donde se extraen dichas imágenes. Los resultados de clasificación se muestran a continuación.

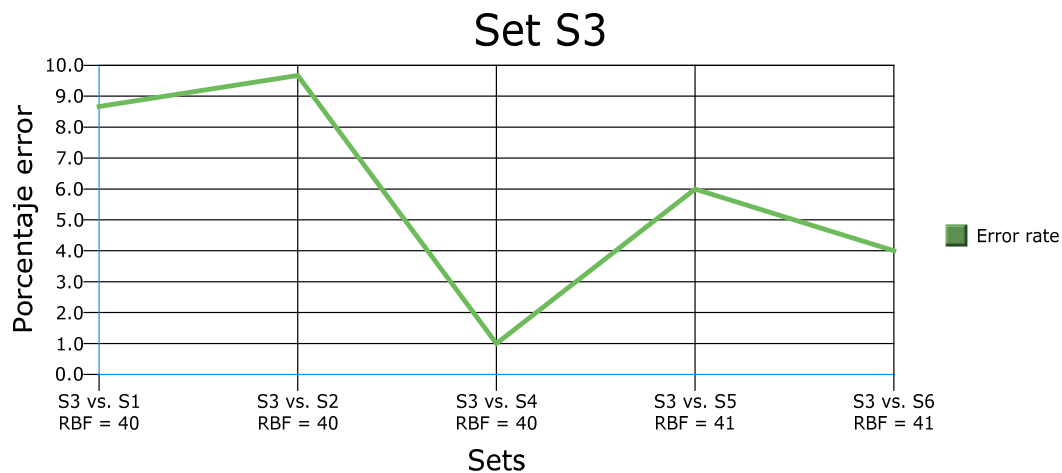


Figura 5.3 Comparativo de porcentaje de errores obtenidos en fase de experimentación del conjunto Línea Costera.

Obviamente se tiene el mayor porcentaje de error al compararlo con el conjunto de Línea Costera con Vegetación como se había explicado anteriormente. Aunque también es aquí en donde tenemos el porcentaje de error más bajo de tan solo 1% de error al ser comparado con el conjunto S4 Ciudad.

Este fenómeno se obtiene gracias a la naturaleza de las imágenes y del método de obtención de las mismas, ya que estos dos grupos fueron obtenidos *in situ* con una cámara digital de 8 megapíxeles de resolución por lo que su definición fue mayor comparado con los otros grupos de imágenes. El nivel de detalle es mayor y por ende la cantidad de información que contiene, lo que permite un error de clasificación menor.

5.2.4 Set S4 Ciudad

Los elementos que contienen las imágenes de ciudad son los más variados de todos los conjuntos de datos, pueden observarse diversos elementos que componen como tal el concepto de ciudad, calles pavimentadas, edificios, estructuras, etc. Los resultados comparativos se muestran a continuación.

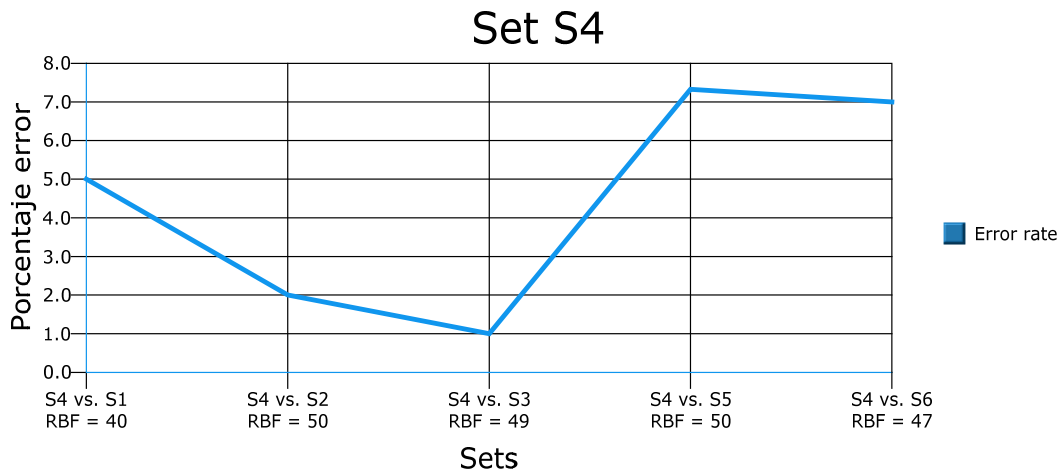


Figura 5.4 Comparativo de porcentaje de errores obtenidos en fase de experimentación del conjunto Ciudad.

Como se explicó en el punto anterior el error de clasificación más pequeño se obtiene al compararlo con el conjunto S2 Línea Costera. El error más alto obtenido en este conjunto es de 7.2% al compararlo con S5 Formaciones Rocosas.

5.2.5 Set S5 Formaciones Rocosas.

Los elementos que componen este conjunto de imágenes contienen texturas rocosas parecidas a las del conjunto S1 Cañones Rocosos pero formando estructuras diversas.

En este conjunto nos enfrentamos con un problema al momento de conformarlo, nos encontramos con una amplia variedad de distancias focales del paisaje, por lo que desde el inicio se esperaba obtener valores mayores en el error. Sin embargo los resultados obtenidos no fueron del todo desalentadores ya que aún con el problema en la variedad de los elementos que contenía se lograron buenos resultados.

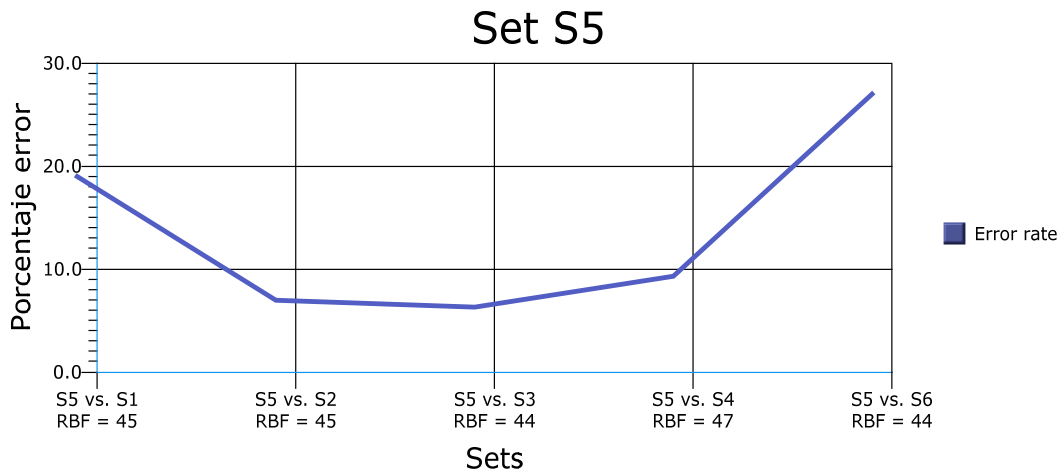


Figura 5.5 Comparativo de porcentaje de errores obtenidos en fase de experimentación del conjunto Formaciones Rocosas.

Como era de esperarse en este conjunto se obtuvo el error de clasificación más alto, al compararlo con el conjunto S6 Montañas Nevadas, que comparten la misma problemática, con un error de clasificación del 27.8%, siendo también el error más alto de todo el proceso de experimentación.

5.2.6 Set S6 Montañas Nevadas

Este conjunto comparte el mismo problema que el conjunto anterior S5 Formaciones Rocosas, ya que las imágenes presentan variedad de distancias focales, lo que las hace propensa a errores al momento de realizar la clasificación.

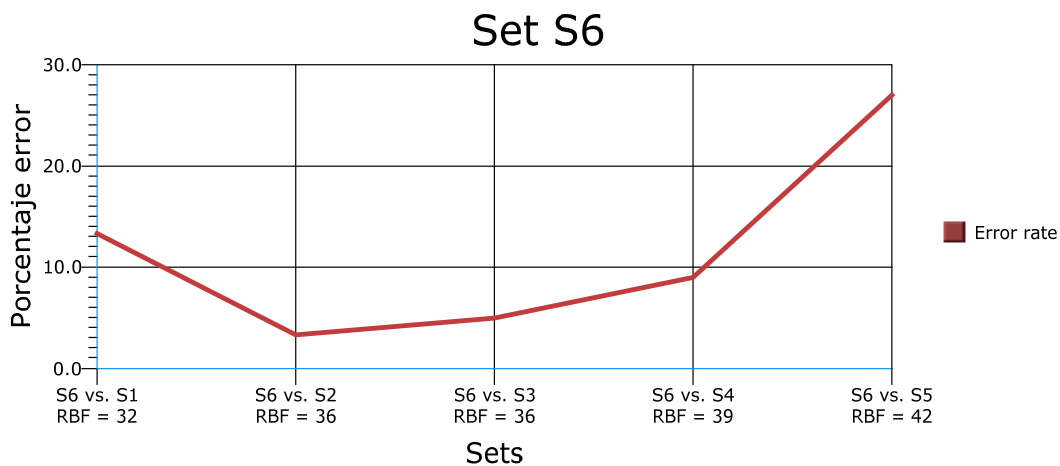


Figura 5.6 Comparativo de porcentaje de errores obtenidos en fase de experimentación del conjunto Montañas Nevadas.

Aunque se esperaba un error de clasificación muy grande, tuvo un mejor desempeño que el conjunto anterior, en donde comparten el valor más alto de 27% de error, pero en general en la comparación con los demás conjuntos mantiene errores de clasificación por debajo del 14%.

5.3 Tiempo de Procesamiento

Parte importante del método propuesto es que el proceso de clasificación se haga en el menor tiempo posible, la implementación SPIDER cuenta con herramientas de optimización de algoritmos de clasificación, las cuales pueden utilizarse plenamente con las Máquinas de Vector de Soporte.

Dos algoritmos de optimización implementados para SVM son *Andre* y *Quadprog*, los cuales fueron utilizados para medir resultados en tiempo de procesamiento. En la figura 5.7 se muestra un cuadro comparativo de los tiempos de ejecución obtenidos utilizando los dos algoritmos, así como el procesamiento normal efectuado por SPIDER. Como podemos ver el algoritmo *Quadprog* no ofrece una mejora con respecto a la ejecución normal, es más, consume más recursos de los esperados, con un tiempo de procesamiento por encima de los 50 minutos para realizar la clasificación de dos conjuntos de datos.

No sucede lo mismo con el algoritmo *Andre* que sí disminuye considerablemente el tiempo, los resultados ofrecen tiempos por debajo de los 4 minutos para la clasificación de todos los conjuntos de datos. En la figura 5.7 podemos observar los resultados comparativos de los tres algoritmos, siendo *Andre* el seleccionado para realizar los experimentos.

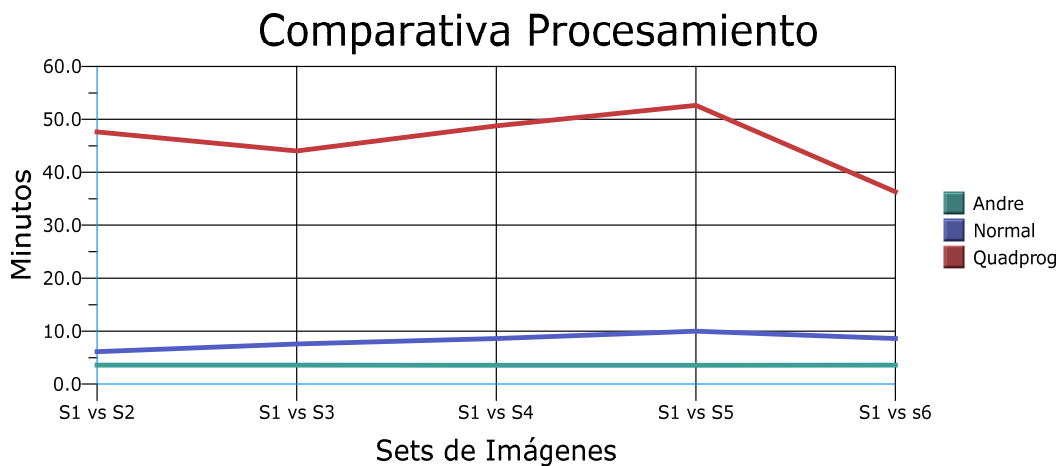


Figura 5.7 Comparativo de velocidades de procesamiento al ejecutar tres optimizadores implementados en SPIDER.

CONCLUSIONES

El estudio de la clasificación de imágenes es un proceso exhaustivo que requiere de cierta capacidad de cómputo y etapas de preprocesamiento para obtener resultados significativos, después de proponer un método dual que involucra filtrado de imágenes por medio de filtros de Gabor y la clasificación de las salidas del filtro por medio de las Máquinas de Vector de Soporte, contamos con un método que además de ofrecer buenos resultados de clasificación, brinda la posibilidad de realizar el proceso de clasificación en un tiempo corto.

En términos generales podemos decir que el método de utilizar dos herramientas para la clasificación de imágenes basado en contenido de textura ofrece una alternativa para llevar a cabo la clasificación de imágenes naturales basado en el contenido de textura.

Los resultados obtenidos en los experimentos de clasificación indican que el modelo es factible de ser usado en alguna implementación que busque clasificar imágenes para bases de datos de imágenes, sitios web e incluso para métodos de recuperación de imágenes basado en contenido.

La dependencia del modelo del proceso de selección de las imágenes se convierte en una de sus debilidades, ya que por un lado es necesario conformar conjuntos de imágenes de un gran número de ellas para poder obtener resultados significativos, además un detalle a considerar es la fuente de dichas imágenes ya que pudimos comprobar que en imágenes obtenidas con una cámara de calidad de resolución alta se obtuvieron mejores resultados que con las imágenes obtenidas de Internet.

Aún tomando en cuenta estas consideraciones el modelo funciona muy bien para grupos de imágenes de Internet, lo que nos hace resaltar una de sus fortalezas ya que actualmente existen en la red millones de imágenes factibles de ser clasificadas para conformar bases de datos, sitios para compartir fotos personales, etc.

El modelo puede seguir desarrollándose agregando conjuntos de imágenes más variados e incrementando el número de texturas que puedan ser detectadas.

REFERENCIAS

- [1] T. Randen and J.H. Husoy, "Filtering for texture classification: a comparative study", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, Issue 4, pp. 291 – 310, Apr 1999.
- [2] F. Lumbreras Ruiz, "Segmentation, classification and modelization of textures by means of multiresolution decomposition techniques", Ph.D. dissertation, Dept. Informática and Computer Vision Center, Universitat Autònoma de Barcelona, Barcelona, España, 2001.
- [3] O. Chapelle, P. Haffner, and V.N. Vapnik, "Support vector machines for histogram-based image classification", *IEEE Trans. On Neural Networks*, Vol. 10, Issue 5, pp. 1055 – 1064, Sep 1999.
- [4] Kwang In Kim, Keechul Jung, Se Hyun Park, and Hang Joon Kim, "Support vector machines for texture classification", *IEEE Trans. On Pattern Analysis and Machine Intelligence*, Vol. 24, Issue 11, pp. 1542 – 1550, Nov 2002.
- [5] I. Gondra and D.R. Heisterkamp, "Learning in region-based image retrieval with generalized support vector machines", *In Proc. of the Computer Vision and Pattern Recognition*, pp. 149 – 154, 2004.
- [6] Shutao Li, J.T. Kwok, Hailong Zhu, and Yaonan Wang, "Texture classification using the support vector machines", *Pattern Recognition*, Vol. 36, No. 12, pp. 2883 – 2893, 2003.
- [7] Bing-Yu Sun and De-Shuang Huang, "Texture classification based on support vector machine and wavelet transform", *In Proc. of the Fifth World Congress on Intelligent Control and Automation*, WCICA 2004. Vol. 2, pp. 1862 – 1864, June 15–19, 2004.
- [8] V.P. Subramanyam Rallabandi and S.K. Sett, "Unsupervised texture classification and segmentation", *Proceedings Of World Academy of Science, Engineering and Technology*, Vol. 5, April 2005.
- [9] J.A. Recio, L.A. Ruiz and A.Fernández-Sarriá, "Use of Gabor filters for texture classification of digital images", *Física de la Tierra*, Vol. 17, pp. 47 – 59, 2005.
- [10] M.R. Turner, "Texture discrimination by Gabor functions", *Biol. Cybern.*, Vol. 55, Num. 2–3, pp. 71 – 82, 1986.
- [11] V. Levesque, "Texture segmentation using Gabor filters", *Center for Intelligent Machines Journal*, 2000
- [12] P. Guha and R. Banerjee, "Segmentation and classification of multi-textured images", 2000, Disponible en: <http://www.cse.iitk.ac.in/~amit/courses/768/00/rajrup/>, last visited: April 20, 2009.
- [13] V.S. Vyas and P. Rege, "Automated texture analysis with Gabor filters", *GVIP Journal*, Vol. 6, Issue 1, pp. 35 – 41, July 2006.
- [14] K.M. Rajpoot and N.M. Rajpoot, "Wavelets and Support Vector Machines for Texture Classification", *In proceedings of the 8th International Multitopic Conference*, INMIC 2004, 24-26 Dec., pp. 328 – 333, 2004.
- [15] D.M. Tsai, "Optimal Gabor filter design for texture segmentation", Technical Report, Machine Vision Lab, Dept. of Ind. Eng. and Mgmt., Yuan-Ze University, Chung-Li, Taiwan, 2000.

- [16] A.C. Bovik, M. Clark and W.S. Geisler, "Multichannel Texture Analysis Using Localized Spatial Filters", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, Num. 1, pp. 55 – 73, 1990.
- [17] Zehang Sun, G. Bebis and R. Miller, "On-road vehicle detection using Gabor filters and support vector machines", *14th International Conference on Digital Signal Processing, DSP 2002*, Vol. 2, pp. 1019 – 1022, 2002.
- [18] K. Hammouda and E. Jernigan, "Texture segmentation using Gabor filters", tech. rep., Biotechnology and health engineering centre, University of Waterloo, Dec. 2000.
- [19] S.E. Grigorescu, N. Petkov and P. Kruizinga, "Comparison of texture features based on Gabor filters", *IEEE Trans. On Image Processing*, Vol. 11, Num. 10, pp. 1160 – 1167, 2002.
- [20] D. Batra, G. Singhal and S. Chaudhury, "Gabor filter based fingerprint classification using support vector machines", *Proceedings of the IEEE First India Annual Conference, 2004, INDICON 2004*, pp. 256 – 261, 20-22 Dec. 2004.
- [21] Q.A. Salih and V. Dhandapani, "IRIS Recognition based on multi-channel feature extraction using gabor filters", *Proceedings of the 2nd IASTED international conference on Advances in computer science and technology, ACST'06*, pp. 168 – 173, 2006.
- [22] L. Ma, Y. Wang and T. Tan, "Iris recognition based on multichannel Gabor filtering", *5th Asian Conf. Computer Vision*, Vol. 1, 2002.
- [23] D. Carr, "Iris recognition: Gabor filtering", *Connexions*. Dec. 18, 2004, Disponible en: <http://cnx.org/content/m12493/1.4/>, última visita April 20, 2009.
- [24] K. Kämäräinen, "Feature extraction using Gabor filters", Ph. D. dissertation, Lappeenranta University of Technology, Finland, Nov. 2003.
- [25] T.P. Weldon, W.E. Higgins and D.F. Dunn, "Gabor filter design for multiple texture segmentation", *Optical Engineering*, Vol. 35, pp. 2852 – 2863, 1996.
- [26] F. Bianconi and A. Fernández, "Evaluation of the effects of Gabor filter parameters on texture classification", *Pattern Recognition*, Vol. 40, Num. 12, pp. 3325 – 3335, 2007.
- [27] J. Ilonen, J.K. Kämäräinen and J.K. Kälviäinen, "Efficient computation of Gabor features", Research Report 100, Lappeenranta University of Technology, Dept. of Information Technology, 2005.
- [28] J.A. Reséndiz, "Las máquinas de vectores de soporte para identificación en línea", Masters Dissertation, Departamento de control automático, Centro de investigación y estudios avanzados, I.P.N., 2006.
- [29] J.P. Lewis, "A short SVM (support vector machine) tutorial", CGIT Lab / IMSC, University Southern California, 2004.
- [30] L. González, "Modelos de clasificación basados en máquinas de vectores de soporte", *Asoc. científica europea de econ. aplicada. Anales de economía aplicada*, 2003.
- [31] D. Dunn, W.E. Higgins and J. Wakeley, "Texture segmentation using 2-D Gabor elementary functions", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, Num. 2, pp. 130 – 149, Feb 1994.

- [32] Mohammad F. A. Fauzi and Paul H. Lewis, "A Fully Unsupervised Texture Segmentation Algorithm", British Machine Vision Association (BMVA), 2003, pp. 519—528.
- [33] X. Song and G. Fan, "A Study of Supervised, Semi-Supervised and Unsupervised Multiscale Bayesian Image Segmentation", *The 2002 45th Midwest Symposium on Circuits and Systems*, MWSCAS-2002. Vol. 2, pp- 371 – 374, Ago 2002.
- [34] M. Tuceryan and A.K. Jain, "Texture Analysis", *The Handbook of Pattern Recognition and Computer Vision, 2nd Edition*, pp. 207 – 248. 1998.
- [35] C. Mandriota, E. Stella, M. Nitti, N. Ancona, A. Distanti, "Rail Corrugation detection by Gabor Filtering", *International Conference on Image Processing, 2001 Proceedings, Volume 2*, pp. 626 – 628, Oct 2001.
- [36] K. Laws, "Textured Image Segmentation", Ph. D. Dissertation, University of Southern California, January, 1980.
- [37] R.C. Gonzalez and R.E. Woods, "Digital Image Processing", 2nd Edition, Prentice Hall, 2002.
- [38] M.J. Bastiaans, "Gabor's Expansion and the Zak Transform for Continuous-Time and Discrete-Time Signals", *Signal and Image Representation in Combined Spaces*, Academic Press, pp. 23 – 69. 1998
- [39] F. Lumbreras, "Segmentation, Classification and Modelization of Texture by Means of Multiresolution Decomposition Techniques", Universidad Autónoma de Barcelona, Departamento de Informática, 2001.
- [40] D. Gabor, "Theory of Communication", *Journal of Institute for Electrical Engineering*, Volume 93, part III, n° 26, pp. 429 – 457, 1946.
- [41] R. Sandler and M. Lindenbaum, "Gabor Filter Analysis for Texture Segmentation", *Conference on Computer Vision and Pattern Recognition Workshop*, pp. 178 – 178, 2006
- [42] N.W. Campbell and B.T. Thomas, "Automatic Selection of Gabor Filters for Pixel Classification", *Sixth International Conference on Image Processing and its Applications*, July 1997, pp. 761 – 765. 1997
- [43] B. Scholkopf and A. Smola. "Learning With Kernels", MIT Press, Cambridge, MA, 2002.
- [44] N. Cristianini and J. Shawe-Taylor, "An Introduction to Support Vector Machines and Other Kernel-based Learning Methods", Cambridge University Press, 2000.
- [45] L. Gonzalez Abril, "Modelos de Clasificación Basados en Máquinas de Vectores de Soporte", *Congreso Asociación Científica Europea de Economía Aplicada*, Almeria, 2003.
- [46] C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition", *Data Mining and Knowledge Discovery*, pp. 121 – 167, Kluwer Academic Publishers, 1998.
- [47] S. Vishnubhotla, "Support Vector Classification", Project Work, University of Maryland, 2005.
- [48] V. Kyrki, *Local and Global Feature Extraction for Invariant Object Recognition*, Ph. D. dissertation, Lappeenranta University of Technology, Finland, 2002.

- [49] F.W. Campbell and J.G. Robson, *Application of Fourier Analysis to the Visibility of Gratings*, Journal of Physiology Num 197, pp. 551 – 556.
- [50] R.L. Devalois and R.T. Marrocco, *Singe Cells Analysis of Saturation Discrimination in the Macaque*, Vision Research, Volume 13, Issue 3, March 1973, pp. 701 – 711.