

**RECONOCIMIENTO AUTOMÁTICO DE BLANCOS EN TIEMPO REAL
MEDIANTE TÉCNICAS DE VISIÓN ARTIFICIAL**

GLORIA ISABEL BAUTISTA LASPRILLA

CLAUDIA CRISTINA PRIETO SOCHA

Director

DR. LUIS JAVIER GARCÍA VILLALBA

Universidad Complutense de Madrid

Co-Director

DR. GONZALO PAJARES MARTINSANZ

Universidad Complutense de Madrid



**INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE
MONTERREY – UNIVERSIDAD AUTONOMA DE BUCARAMANGA – UTB
MAESTRÍA EN CIENCIAS COMPUTACIONALES
CARTAGENA DE INDIAS SEPTIEMBRE 2005**

INTRODUCCIÓN

La visión artificial es como el sentido de la vista del computador y brinda la oportunidad de automatizar y mejorar muchos procesos en la industria, en la seguridad de espacios físicos por medio de cámaras de vídeo, en la medicina e inclusive algunas de sus técnicas se emplean en la integración de vídeo en las redes de cómputo y en la transmisión de imágenes para su almacenamiento y despliegue en todo Internet.

Estas grandes aplicaciones de la visión artificial hacen que resulte interesante proponer un proyecto donde se analizan algunos sistemas para el reconocimiento automático de blancos en tiempo real mediante técnicas de visión artificial por medio de algoritmos basados en técnicas de reconocimiento y tratamiento de imágenes, con la finalidad de solucionar algunos problemas que se presentan en la seguridad de Bases Militares y empresas de alto riesgo.

El presente trabajo se desarrolla en cinco capítulos. En el primer capítulo se tiene el anteproyecto presentado durante la primera fase del desarrollo, donde se encuentran los antecedentes, los objetivos, el planteamiento del problema y la justificación.

En el segundo capítulo se hace un análisis de las teorías existentes sobre visión artificial para poder así entrar al tercer capítulo donde se comparan y se

establecen paralelos con las teorías de varios autores que han investigado y trabajado con los sistemas automáticos de reconocimiento de blancos.

El cuarto capítulo es un informe técnico de los resultados obtenidos en Trabajos de Laboratorio realizados por estudiantes de la Universidad Tecnológica de Bolívar. Estos talleres confrontan descriptores en la caracterización de imágenes. Estos descriptores son claves en el desarrollo de entrenamiento y ejecución de redes neuronales.

En el quinto capítulo la investigación se centra en el estudio de redes neuronales para el reconocimiento de patrones y, más puntualmente, en la red backpropagation perceptron multicapa. La razón de escoger este método se basa en estudios ya realizados a nivel internacional que han demostrado que este método de reconocimiento es uno de los más efectivos en visión computacional para tiempo real. Por lo tanto, el análisis de estos algoritmos junto con el estudio de su complejidad permite establecer parámetros para comparar y proporcionar documentación de tal manera que se pueda ofrecer una base de conocimiento que permita en el futuro a quienes diseñan software de reconocimiento de imágenes la toma de decisión del mejor modelo a desarrollar.

Sobre el estado del arte de esta investigación podemos decir que ha sido muy nutrido: se han desarrollado varios trabajos de grado, a nivel de pregrado y maestría, donde se han implementado los algoritmos de reconocimiento con

buenos niveles de éxito, aunque también se han encontrado algunos inconvenientes en los resultados. Estos trabajos han dado pie a varios artículos científicos que representan aportes importantes en esta investigación. Algunos de estos proyectos y sus resultados se mencionarán en el desarrollo de este proyecto.

CONTENIDO

| | |
|--|-----------|
| 1 ANTEPROYECTO DE INVESTIGACIÓN | 13 |
| 1.1 ANTECEDENTES | 13 |
| 1.2 PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN | 16 |
| 1.3 OBJETIVOS DE LA INVESTIGACIÓN..... | 17 |
| 1.3.1 OBJETIVO GENERAL..... | 17 |
| 1.3.2 OBJETIVOS ESPECÍFICOS | 17 |
| 1.4 MARCO TEÓRICO..... | 19 |
| 1.4.1 RECONOCIMIENTO AUTOMÁTICO DE BLANCOS (ATR)..... | 19 |
| 1.4.1.1 Sistemas Clásicos de ATR..... | 20 |
| 1.4.1.2 Sistemas de ATR basados en modelos (MODEL-BASED ATR) .. | 20 |
| 1.4.1.3 Sistemas de ATR multiespectrales (MULTIESPECTRAL ATR) .. | 21 |
| 1.4.1.4 Sistemas de ATR para reconocimiento de Buques | 22 |
| 1.4.1.5 Sistemas multisensores de ATR | 22 |
| 1.4.1.6 Sistemas de ATR multidimensionales..... | 23 |
| 1.4.1.7 Sistemas XTRS de ATR..... | 23 |
| 1.4.2 VISIÓN ARTIFICIAL | 23 |
| 1.4.2.1 Visión de Bajo Nivel..... | 25 |
| 1.4.2.2 Visión de Nivel Intermedio..... | 25 |

| | | |
|---------|-------------------------------|----|
| 1.4.2.3 | Visión de Alto Nivel..... | 26 |
| 1.4.2.4 | Descriptores de Fourier | 27 |

2 ANÁLISIS SOBRE LAS TEORÍAS REFERENTES A LA VISIÓN

| | |
|------------------------|-----------|
| ARTIFICIAL..... | 37 |
|------------------------|-----------|

| | |
|--|-----------|
| 2.1 ENFOQUE DESDE LA APLICACIÓN PARA LA PERCEPCIÓN DE LAS MÁQUINAS DE FORMA AUTÓNOMA..... | 37 |
|--|-----------|

| | | |
|-------|--------------------------------------|----|
| 2.1.1 | ETAPAS DE LA VISIÓN ARTIFICIAL | 40 |
|-------|--------------------------------------|----|

| | | |
|---------|---------------------------|----|
| 2.1.1.1 | Visión de Bajo Nivel..... | 41 |
|---------|---------------------------|----|

| | | |
|-----------|-----------------|----|
| 2.1.1.1.1 | Captación | 41 |
|-----------|-----------------|----|

| | | |
|-----------|-----------------------|----|
| 2.1.1.1.2 | Preprocesamiento..... | 41 |
|-----------|-----------------------|----|

| | | |
|---------|---------------------------------|----|
| 2.1.1.2 | Visión de Nivel Intermedio..... | 42 |
|---------|---------------------------------|----|

| | | |
|-----------|--------------------|----|
| 2.1.1.2.1 | Segmentación | 42 |
|-----------|--------------------|----|

| | | |
|-----------|-------------------|----|
| 2.1.1.2.2 | Descripción | 43 |
|-----------|-------------------|----|

| | | |
|-----------|--|----|
| 2.1.1.2.3 | Reconocimiento de objetos individuales | 43 |
|-----------|--|----|

| | | |
|---------|---------------------------|----|
| 2.1.1.3 | Visión de Alto Nivel..... | 43 |
|---------|---------------------------|----|

| | | |
|-----------|----------------------|----|
| 2.1.1.3.1 | Interpretación | 43 |
|-----------|----------------------|----|

| | |
|--|-----------|
| 2.2 ENFOQUE DESDE EL ANÁLISIS AUTOMÁTICO DE IMÁGENES..... | 44 |
|--|-----------|

| | |
|-------|--|
| 2.2.1 | ETAPAS FUNDAMENTALES DEL PROCESAMIENTO |
|-------|--|

| | |
|--|----|
| AUTOMÁTICO DE ANÁLISIS DE IMÁGENES | 45 |
|--|----|

| | | |
|---------|-------------------------------|----|
| 2.2.1.1 | Procesado de Bajo Nivel | 46 |
|---------|-------------------------------|----|

| | | |
|-----------|------------------------------|----|
| 2.2.1.1.1 | Adquisición de imágenes..... | 46 |
|-----------|------------------------------|----|

| | | |
|-------------|--|----|
| 2.2.1.1.2 | Preprocesamiento de la Imagen..... | 48 |
| 2.2.1.2 | Procesado de Nivel Intermedio | 49 |
| 2.2.1.2.1 | Segmentación | 49 |
| 2.2.1.2.2 | Representación..... | 50 |
| 2.2.1.2.3 | Descriptores..... | 51 |
| 2.2.1.2.3.1 | Códigos de cadena..... | 52 |
| 2.2.1.2.3.2 | Momentos | 53 |
| 2.2.1.2.3.3 | Momentos Invariantes | 56 |
| 2.2.1.2.3.4 | Momentos Invariantes de Hu..... | 57 |
| 2.2.1.2.3.5 | La Transformada Discreta del Coseno..... | 59 |
| 2.2.1.2.3.6 | La Transformada Discreta de Fourier..... | 61 |
| 2.2.1.3 | Procesado de Alto Nivel | 65 |
| 2.2.1.3.1 | Reconocimiento e interpretación | 65 |
| 2.2.1.3.2 | Reconocimiento de Patrones: El Perceptrón y el Perceptrón Multicapa | 66 |
| 2.2.1.3.2.1 | Aprendizaje | 67 |
| 2.2.1.3.2.2 | Generador..... | 69 |
| 2.2.1.3.2.3 | Sistema | 69 |
| 2.2.1.3.2.4 | Máquina de Aprendizaje | 70 |
| 2.2.1.3.3 | La Red Retropropagación (Backpropagation)..... | 71 |
| 2.2.1.3.3.1 | Entrenamiento por Retropropagación | 73 |

| | |
|---|------------|
| 3 ANÁLISIS DE LOS DESARROLLOS DE LOS SISTEMAS AUTÓMATICOS DE RECONOCIMIENTO DE BLANCOS | 75 |
| 3.1 SISTEMA CLÁSICO DE ATR..... | 75 |
| 3.1.1 PREPROCESAMIENTO..... | 78 |
| 3.1.2 DETECCIÓN..... | 78 |
| 3.1.3 SEGMENTACIÓN..... | 80 |
| 3.1.4 CLASIFICACIÓN | 82 |
| 3.2 APLICACIONES DE SISTEMAS CLÁSICOS DE ATR..... | 85 |
| 3.2.1 SISTEMAS DE ATR BASADOS EN MODELOS (MODEL-BASED ATR) 85 | |
| 3.2.2 SISTEMAS DE ATR MULTIESPECTRALES (MULTIESPECTRAL ATR) 90 | |
| 3.2.3 SISTEMAS DE ATR PARA RECONOCIMIENTO DE BUQUES | 94 |
| 3.2.4 SISTEMAS MULTISENORES DE ATR | 95 |
| 3.2.5 SISTEMAS DE ATR MULTIDIMENSIONALES | 97 |
| 3.2.6 SISTEMAS XTRS DE ATR..... | 99 |
| 4 LIMITACIONES EN VISIÓN ARTIFICIAL Y LOS SISTEMAS DE RECONOCIMIENTO | 103 |
| 4.1 VENTAJAS Y DESVENTAJAS DE LOS MÉTODOS DE EXTRACCIÓN DE CARACTERÍSTICAS | 104 |

| | | |
|------------|---|------------|
| 4.1.1 | ESTUDIO MATRICIAL DE LA IMAGEN..... | 104 |
| 4.1.2 | CÓDIGO DE CADENA..... | 104 |
| 4.1.3 | MOMENTOS INVARIANTES DE HU..... | 105 |
| 4.1.4 | FOURIER..... | 106 |
| 5 | FLUJOGRAMA PARA EL MANEJO DE ALGORITMOS EN EL RECONOCIMIENTO DE PATRONES UTILIZANDO REDES NEURONALES | 108 |
| 5.1 | REDES NEURONALES | 108 |
| 5.1.1 | ENTRENAR LA RED..... | 110 |
| 5.1.1.1 | Crear conjunto de datos de entrenamiento..... | 112 |
| 5.1.2 | EJECUTAR LA RED..... | 112 |
| 5.1.2.1 | Obtención del conjunto de imágenes de entrenamiento..... | 114 |
| 5.1.2.1.1 | Descriptor Momentos..... | 116 |
| 5.1.2.1.2 | Descriptor TDC | 117 |
| 6 | CONCLUSIONES..... | 121 |
| 7 | BIBLIOGRAFÍA..... | 124 |

TABLA DE ILUSTRACIONES

| | |
|--|----|
| ILUSTRACIÓN I. PROPIEDAD DE TRASLACIÓN DE LOS DESCRIPTORES DE FOURIER..... | 30 |
| ILUSTRACIÓN II. PROPIEDADES DE ROTACIÓN DE LOS DESCRIPTORES DE FOURIER..... | 31 |
| ILUSTRACIÓN III. ORIGINAL Y LA IMAGEN DE BORDES PARA OBTENER LOS DESCRIPTORES DE FOURIER..... | 33 |
| ILUSTRACIÓN IV. ESPECTRO..... | 34 |
| ILUSTRACIÓN V. PROCESO DE RECONSTRUCCIÓN DE CONTORNO DE LOS DESCRIPTORES DE FOURIER..... | 35 |
| ILUSTRACIÓN VI. DIAGRAMA DE BLOQUES MOSTRANDO EL PROCESO COMPLETO DE LA VISIÓN ARTIFICIAL..... | 38 |
| ILUSTRACIÓN VII. ETAPAS DE LA VISIÓN ARTIFICIAL..... | 41 |
| ILUSTRACIÓN VIII. ELEMENTOS DEL ANÁLISIS DE IMÁGENES..... | 46 |
| ILUSTRACIÓN IX. REPRESENTACIÓN DE UN CONTORNO EN CODIGOS DE CADENA..... | 52 |
| ILUSTRACIÓN X. (A) CÓDIGOS DE CADENA DEL NÚMERO 2 (B) MAPA DE PÍXELES VECINOS (8-CONNECTED)..... | 53 |
| ILUSTRACIÓN XI. REPRESENTACIÓN DE UN CONTORNO EN MOMENTOS..... | 54 |
| ILUSTRACIÓN XII. REPRESENTACIÓN DEL BORDE COMO UNA SECUENCIA DE COORDENADAS..... | 61 |

| | |
|---|-----|
| ILUSTRACIÓN XIII. MÁQUINA DE APRENDIZAJE UTILIZANDO OBSERVACIONES DEL SISTEMA PARA OBTENER UNA APROXIMACIÓN DE SU SALIDA..... | 68 |
| ILUSTRACIÓN XIV. MODELO DE RED NEURONAL PERCEPTRÓN..... | 72 |
| ILUSTRACIÓN XV. COMPONENTES DE UN SISTEMA CLÁSICO DE ATR...78 | |
| ILUSTRACIÓN XVI. NIVELES DE SEGMENTACIÓN..... | 82 |
| ILUSTRACIÓN XVII. OPERACIONES PARA RECONOCER OBJETOS EN IMÁGENES MULTIESPECTRALES. | 93 |
| ILUSTRACIÓN XVIII. MÓDULOS DEL SISTEMA DE ATR DE KOLODZY..... | 98 |
| ILUSTRACIÓN XIX. PROCESO DE RECONOCIMIENTO BASADO EN REDES NEURONALES (ENTRENAMIENTO DE LA RED)..... | 110 |
| ILUSTRACIÓN XX. PROCESO DE RECONOCIMIENTO BASADO EN REDES NEURONALES (EJECUCIÓN DE LA RED). | 114 |
| ILUSTRACIÓN XXI. SISTEMA DE CODIFICACIÓN DE LA TRANSFORMADA (CODIFICADOR). | 119 |
| ILUSTRACIÓN XXII. SISTEMA DE CODIFICACIÓN DE LA TRANSFORMADA (DECODIFICADOR). | 119 |

TABLA DE ECUACIONES

| | |
|---|----|
| ECUACIÓN 1. FUNCIÓN LÍMITE DESCRIPTORES DE FOURIER. | 28 |
| ECUACIÓN 2. ECUACIONES DE ANÁLISIS Y SÍNTESIS PARA LA SERIE DISCRETA DE FOURIER DE LA FUNCIÓN LÍMITE. | 29 |
| ECUACIÓN 3. ÁREA ENCERRADA POR UN CONTORNO EN LA DILATACIÓN O COMPRESIÓN DEL CONTORNO..... | 32 |
| ECUACIÓN 4. MOMENTO N-ÉSIMO DE V CON RESPECTO A SU MEDIA EN DESCRIPTORES DE MOMENTOS..... | 54 |
| ECUACIÓN 5. MOMENTOS DE ORDEN $(P+Q)$ PARA UNA REGIÓN..... | 57 |
| ECUACIÓN 6. MOMENTOS CENTRAL DE ORDEN $(P+Q)$ PARA UNA REGIÓN..... | 57 |
| ECUACIÓN 7. MOMENTOS INVARIANTES DE HU..... | 58 |
| ECUACIÓN 8. LA TRANSFORMADA DISCRETA DEL COSENO (TDC)..... | 59 |
| ECUACIÓN 9. LA TDC INVERSA..... | 60 |
| ECUACIÓN 10. FUNCIÓN DE ACTIVACIÓN SIGMOIDAL..... | 73 |

1 ANTEPROYECTO DE INVESTIGACIÓN

1.1 ANTECEDENTES

En el mercado se encuentra una gran cantidad de software para el reconocimiento de patrones como, por ejemplo, los programas OCR (Optical Character Recognition) que se anexan sin cargo alguno en el paquete de compra de cualquier escáner de mesa. En el ambiente industrial es común encontrar programas como IMAQ-Visión, Polaroid, Zenith, Canon, Fisher, etc., que contienen una gran cantidad de técnicas computacionales para el procesamiento de imágenes.

Todos estos programas tienen en común que son programados para una determinada acción e involucran una gran base de datos con la información de los parámetros a reconocer o a procesar.

Muchas aplicaciones actuales emplean una combinación de redes neuronales y procesamiento digital de imágenes para desarrollar poderosos sistemas de visión artificial. La visión artificial es una ciencia que aún está en proceso de desarrollo, su verdadera potencialidad no se ha alcanzado todavía.

El ojo humano provee constantemente una gran cantidad de información acerca del mundo externo. Gracias a nuestra visión logramos enterarnos de

todos los objetos y seres vivientes que nos rodean y representar sus formas y propiedades en nuestros cerebros.

Los modelos no son generalmente innatos y tampoco son estructuras a priori: típicamente nosotros no reconocemos cosas que no hayamos visto antes; sin embargo, debemos construir modelos de nuestra diaria experiencia visual. Esto ha motivado la investigación sobre la realización de sistemas artificiales para el reconocimiento de imágenes. Una única escena se puede manifestar bajo una gran cantidad de imágenes diferentes, dependiendo de la iluminación, del punto de vista, de la óptica de las cámaras, etc. Por este motivo existen muchas técnicas computacionales utilizadas en la Visión Artificial, las cuales se emplean en combinación, con el único propósito de inferir correctamente la descripción de una escena.

En el marco tecnológico y de aplicación de la visión artificial hay un gran impacto en aspectos como el análisis del comportamiento de sistemas de vigilancia, vigilancia inteligente, vehículos autónomos guiados por visión, reconocimiento de rostros, de iris y de huellas dactilares, reconocimiento de caracteres ópticos (OCR), identificación de ambientes para autómatas, identificación de objetos en líneas de ensamble industrial, control de calidad en proceso de productos, etc.

En la Universidad Tecnológica se encuentran dos trabajos de grado referentes a Visión Artificial: “Reconocimiento de Formas Planas Regulares Mediante Redes Neuronales Artificiales” realizado por Alvin Barrios y Elkin Cancio, y “Reconocimiento de Formas Irregulares Empleando Redes Neuronales Artificiales” realizado por Orlando Tinoco y Luís Acosta. Ambos trabajos han sido dirigidos por el profesor Eduardo Gómez. Estos trabajos han sido hasta el momento los únicos realizados en la institución referentes a este tema.

Existe la gran necesidad de continuar con los estudios de Visión Artificial en la Universidad Tecnológica de Bolívar. Para esto se han planteado proyectos nuevos y la definición de una línea de investigación adjunta al grupo GRITAS (Grupo de Investigación sobre Tecnologías Aplicadas y Software del Programa de Ingeniería de Sistemas). Cabe anotar que las autoras de este trabajo pertenecen a este Grupo.

1.2 PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN

El problema planteado consiste en modelar y evaluar las técnicas de identificación basadas en redes neuronales que permitan reconocer patrones, valorar la eficiencia de esas técnicas y diseñar un esquema de manejo de algoritmos que permitan el desarrollo de aplicaciones para reconocer imágenes y blancos en tiempo real. El citado problema tiene la finalidad de contribuir al desarrollo de una nueva línea para el Grupo de Investigaciones en Tecnologías Aplicadas y Software (GRITAS) de la Universidad Tecnológica de Bolívar.

En el marco teórico se definió al ATR como el término dado al campo de las ciencias de la ingeniería que estudian los sistemas y técnicas diseñadas para identificar a partir de imágenes y datos, la localización y los atributos correspondientes a objetivos físicos específicos (blancos), sin la intervención directa de operadores humanos. La falta de éxito definitivo de estos sistemas ha sido atribuida a razones tales como la carencia de adecuados sensores y de mecanismos apropiados para incorporar el conocimiento experto, por que se ha manejado en la mayoría de los casos a nivel de hardware. Tal como lo cita [Gader, 2001] este avance puede ocurrir como el resultado de tres desarrollos convergentes, donde uno de ellos es el avance en las técnicas de computación suave (lenguajes de bajo nivel) para inferir conclusiones a partir de los datos observados.

1.3 OBJETIVOS DE LA INVESTIGACIÓN

1.3.1 OBJETIVO GENERAL

Evaluar modelos de decisión teórica de Visión Artificial usados en el reconocimiento automático de blancos¹ en tiempo real.

1.3.2 OBJETIVOS ESPECÍFICOS

1. Conocer y evaluar el estado actual de los Sistemas Automáticos de Reconocimiento de Blancos mediante Imágenes.
2. En un Informe Técnico plantear las limitaciones encontradas a la hora de desarrollar la teoría de Visión Artificial y los Sistemas de Reconocimiento de Blancos, teniendo en cuenta experiencias de trabajos realizados alrededor de este tema.
3. Desarrollar algoritmos en Lenguaje C de sistemas de reconocimiento de patrones basados en técnicas de Visión Artificial, que permitan

¹ En visión artificial, reconocer blancos es una particularidad de reconocer patrones.

establecer puntos de partida para futuros proyectos de reconocimiento de imágenes y blancos.

4. Diseñar un flujograma para el manejo de los algoritmos propuestos teniendo en cuenta la mejora de los tiempos de respuesta en el reconocimiento de patrones.

1.4 MARCO TEÓRICO

1.4.1 RECONOCIMIENTO AUTOMÁTICO DE BLANCOS (ATR)

Reconocimiento Automático de Blancos (Automatic Target Recognition) es el término dado al campo de las ciencias de la ingeniería que estudian los sistemas y técnicas diseñadas para identificar, a partir de imágenes y datos, la localización y los atributos correspondientes a objetivos físicos específicos (blancos), sin la intervención directa de operadores humanos [Augustyn, 1992] [Bhanu, 1986]. La “Tecnología de Reconocimiento” hace referencia a los presentes o futuros sistemas que tienen el potencial de proveer un avance cuantitativo en las capacidades de los sistemas de reconocimiento [Gader, 2001]. Este avance puede ocurrir como el resultado de tres desarrollos convergentes:

1. Mayores avances en la tecnología de los sensores.
2. Mayores avances en la tecnología de los sistemas de procesamiento de datos.
3. El uso de técnicas de software para inferir conclusiones a partir de los datos observados.

El campo de los ATR se encuentra muy asociado a las operaciones militares. Es en este contexto donde los sistemas de ATR difieren de su forma más general, el Reconocimiento Automático de Objetos (Automatic Object Recognition AOR), puesto que los sistemas de ATR deben soportar los problemas relativos a las interferencias producidas por los humanos como el camuflaje y los requerimientos de tiempo real para operación de los sistemas.

1.4.1.1 Sistemas Clásicos de ATR

En esta sección se presenta una serie de sistemas de Reconocimiento Automático de Blancos que utilizan sistemas clásicos para llevar a cabo las tareas de detección y/o clasificación de blancos. Esta presentación no intenta hacer un exhaustivo listado de todas las aproximaciones de ATR, sino una generalización de los sistemas más representativos.

1.4.1.2 Sistemas de ATR basados en modelos (MODEL-BASED ATR)

A principios de los años 80, el término "Reconocimiento Automático de Blancos Basado en Modelos" emerge en el contexto del ATR cuando algunos investigadores empezaron a usarlo para describir el desempeño de los algunos sistemas desarrollados hasta entonces. Kegelmeyer y Webster [Kegelmeyer, 1988] señalan que el término tiene variados significados para diferentes autores. Este ha sido usado para describir:

1. Modelos de superficie y volumétricos de objetos tridimensionales.
2. Coeficientes de auto-regresión, momentos basados en características, descriptores de Fourier.
3. Grafos semánticos para describir objetos en imágenes.
4. Comportamiento de sensores y procesos físicos.
5. Propiedades de las imágenes y los blancos.

1.4.1.3 Sistemas de ATR multiespectrales (MULTIESPECTRAL ATR)

El uso de diferentes bandas del espectro para hacer el análisis de una escena ha sido ampliamente usado y, recientemente, se ha convertido en una importante herramienta en los sistemas de ATR, que busca fuentes alternas de información para mejorar su rendimiento. Trivedi [Trivedi, 1987] utiliza el conocimiento de las características de reflectividad de algunos materiales en ciertas bandas para reconocer objetos en imágenes reales. Más que el uso de diferentes bandas para hacer el reconocimiento, se busca seleccionar la banda en la cual se maximizan los efectos de reflectividad frente a otros materiales, obteniendo así la respuesta deseada. Kahle y Goetz [Kahle, 1983] hace un estudio del uso de múltiples bandas infrarrojas para clasificar diferentes clases de materiales, mediante las características de emisión de ellos. Ellos utilizan

un escáner de seis canales llamado TIMS (Termal Infrared Multiespectral Scanner). El escáner cubre el espectro desde los 8.2 hasta los 12.2 micrones, sin dejar vacíos. El sistema puede clasificar correctamente diferentes materiales usando una técnica especial para desplegar las diferencias espectrales como diferencias de color. Algunos trabajos usan similares aproximaciones para hacer mejoras en imágenes aéreas [Price, 1987] o para mejorar la detectabilidad de los blancos [Stocker, 1990]; aquí algunos algoritmos son empleados utilizando información espectral brindada por un sensor infrarrojo pasivo. Trivedi [Trivedi, 1985] presentan una lista simple de operaciones requeridas para reconocer objetos en imágenes multiespectrales

1.4.1.4 Sistemas de ATR para reconocimiento de Buques

Burgess [Burgess, 1993] desarrolló un sistema completo de detección de buques, basado en pares de imágenes. El sistema usa dos escenas reales: una imagen multiespectral SPOT y una imagen satelital LANDSAT. Más que detección el sistema realiza un reconocimiento de dos clases (Buque o No buque).

1.4.1.5 Sistemas multisensores de ATR

Algunos sistemas han usado diferentes sensores para incrementar las de detección y/o reconocimiento, y reducir el ratio de falsa alarma tomando ventaja

de las características específicas de las diferentes clases de sensores.

1.4.1.6 Sistemas de ATR multidimensionales

Kolodzy [Kolodzy, 1993] también combina sensores activos y pasivos en el proceso de reconocimiento de blancos. Los sensores pasivos son sistemas de visión infrarrojos, mientras que el sensor activo es un sistema LADAR (Laser-radar).

1.4.1.7 Sistemas XTRS de ATR

XTRS es un sistema experimental de reconocimiento automático de blancos basado en modelos, desarrollado por el laboratorio Lincon, para reconocer vehículos en imágenes reales o sintéticas [Verly, 1992].

1.4.2 VISIÓN ARTIFICIAL

La visión artificial está basada en el análisis de imágenes y puede ser definida como aquellos procesos que le permitan la obtención, caracterización e interpretación de imágenes tomadas de un mundo tridimensional [Contreras, 2000]. Estos procesos pueden subdividirse en:

1. Captación.

2. Preprocesamiento.
3. Segmentación.
4. Descripción.
5. Reconocimiento.
6. Interpretación.

Según el nivel de complejidad en la implementación de las técnicas de tratamiento de imágenes, podemos considerar tres niveles de procesamiento:

1. Visión o procesado de bajo nivel.
2. Visión o procesado de nivel intermedio.
3. Visión o procesado de alto nivel.

Estos niveles no tienen unos límites bien definidos, pero nos proporcionan una referencia para clasificar los diferentes procesos que componen el análisis de imágenes.

1.4.2.1 Visión de Bajo Nivel

Se asocia a aquellos procesos primarios que no requieren inteligencia por parte del sistema de procesamiento de imágenes y que se pueden considerar como reacciones automáticas. En este nivel pueden ubicarse la captación y el preprocesamiento de una imagen, es decir abarca desde los procesos de obtención de una imagen (películas fotográficas, sensores CCD, etc.) hasta las técnicas que permiten la reducción de ruido y realce de detalles. Una analogía con el sistema de visión humana puede verse en el cambio de diámetro de la pupila como respuesta a variaciones de intensidad de luz o la variación de la forma del cristalino en dependencia de la distancia de enfoque de la imagen. Estas son reacciones inconscientes en la formación de las imágenes.

1.4.2.2 Visión de Nivel Intermedio

Se asocia a los procesos de extracción y caracterización de los componentes de una imagen. Este nivel abarca la Segmentación, la Descripción y el Reconocimiento, es decir, los procesos que permiten la división de la imagen en objetos que sean de interés, la definición de características convenientes para diferenciar un objeto de otro (como tamaño, forma, textura, etc.) y la identificación de determinados tipos de objetos. En este nivel es necesario disponer de elementos más sofisticados para la resolución de problemas que

las simples reacciones automáticas de bajo nivel.

1.4.2.3 Visión de Alto Nivel

Se refiere a los procesos que tratan de emular el conocimiento inteligente.

Podemos aproximar el concepto de inteligencia de la siguiente manera:

1. Capacidad de extraer información.
2. Capacidad de aprender a través de ejemplos y saber utilizar este conocimiento para aplicarlo a nuevas situaciones.
3. Capacidad de realizar inferencias a partir de una información incompleta y de tomar decisiones.

En este nivel el proceso de interpretación de la imagen consiste en asociar un significado a un conjunto de objetos reconocidos.

Cuando se habla de visión en robótica el panorama es muy amplio y abarca gran cantidad de técnicas y procedimientos de acuerdo a los objetivos que se persiguen. Por ejemplo, si se desea que el robot identifique y manipule objetos, se requiere un procesamiento de imágenes orientado a la extracción de las características que permitirán el reconocimiento como color, textura o forma. Si el objetivo es lograr que el robot se desplace por una trayectoria evadiendo obstáculos para alcanzar una meta, seguramente se realizará un análisis de

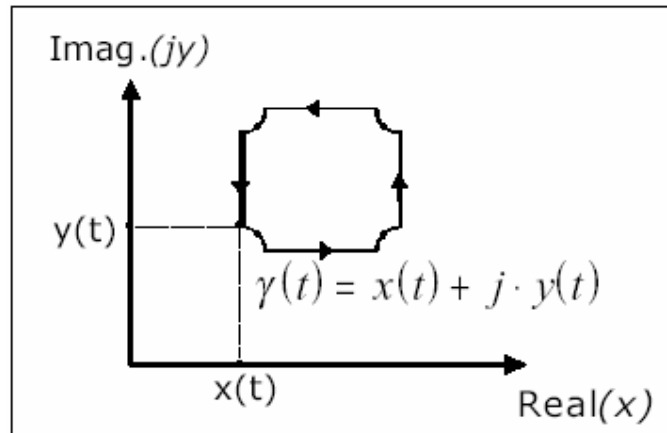
las regiones de la escena para encontrar la ubicación de los obstáculos y realizar el estudio de las trayectorias.

Una técnica de gran utilidad en el reconocimiento de objetos con base en su forma (contorno) son Los Descriptores de Fourier. Se partirá de la fundamentación matemática para estudiar sus propiedades y se observarán los resultados obtenidos cuando se aplican a figuras sencillas.

1.4.2.4 Descriptores de Fourier

Los descriptores de Fourier son una herramienta muy útil en aplicaciones que se basan en la identificación de objetos por medio del contorno. Este método proporciona una representación en el dominio de la frecuencia de la función límite de un objeto, la cual puede ser analizada con base en una serie de propiedades derivadas de las Series de Fourier con el fin de caracterizar un contorno y establecer un sistema de identificación para los objetos de interés.

Para aplicar los descriptores primero debe generarse la función límite del objeto a analizar. Esto se hace de la siguiente manera:



Ecuación 1. FUNCIÓN LÍMITE DESCRIPTORES DE FOURIER.

La función límite se construye a partir de las coordenadas de cada uno de los puntos que constituyen el contorno, haciendo un recorrido a través de él bien sea en el sentido horario o antihorario, como se observa en la ecuación.

Como resultado se tiene una función con las siguientes características:

1. Discreta: debido a que se trata de una imagen digital se tiene un número finito de puntos y por tanto de posibles coordenadas.
2. Compleja: ya que la componente “x” se asume como la parte real y la componente “y” como la parte imaginaria.
3. Periódica: conforme se avanza en el recorrido del contorno se llega al punto inicial y la función empieza a repetirse.

4. De variación limitada: debido a que el contorno es cerrado y continuo, por la naturaleza discreta de la imagen de bordes no se permiten grandes variaciones en las coordenadas de píxeles contiguos.

Con base en lo anterior se observa que pueden aplicarse las Series de Fourier a la función límite para encontrar su representación en el dominio de la frecuencia.

$$C_n = \frac{1}{T} \sum_{t \in \langle N \rangle} (t) \cdot e^{-\frac{j 2 \pi n}{T} \cdot t}$$

$$\gamma(t) = \sum_{n \in \langle N \rangle} C_n \cdot e^{\frac{j 2 \pi n}{T} \cdot t}$$

Ecuación 2. ECUACIONES DE ANÁLISIS Y SÍNTESIS PARA LA SERIE DISCRETA DE FOURIER DE LA FUNCIÓN LÍMITE.

PROPIEDADES DE LOS DESCRIPTORES DE FOURIER

INTEGRIDAD: las funciones $e^{\frac{j 2 \pi n}{T} \cdot t}$ forman una base ortogonal integrable (sumable) para los valores que corresponden al contorno.

CONVERGENCIA: puesto que la función $\gamma(t)$ es discreta, conectada en un contorno cerrado y periódica la representación en Series de Fourier es una

serie finita y, por tanto, converge.

TRASLACIÓN: si el límite de un objeto es trasladado en el plano en $Z_0 = X_0 + jY_0$ a cada uno de los términos de la función límite se le añade un valor constante, como puede verse en la siguiente Ilustración:

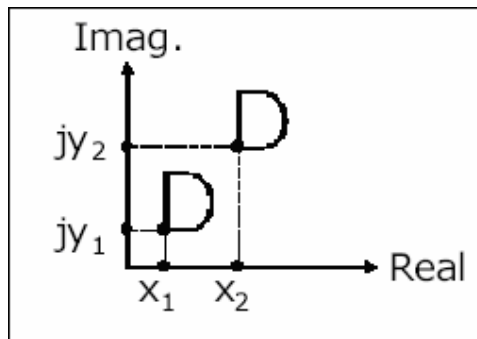


Ilustración I. PROPIEDAD DE TRASLACIÓN DE LOS DESCRIPTORES DE FOURIER.

$$X_2 = X_1 + X_0$$

$$Y_2 = Y_1 + Y_0$$

$$\gamma(t) = \gamma(t) + Z_0$$

La suma de un valor constante a la función implica un cambio en el valor promedio de la señal. Por tanto, sólo se verá afectado el primer coeficiente del espectro y los otros permanecerán invariables:

$$C'_0 = C_0 + Z_0; C'_n = C_n$$

ROTACIÓN: cuando el contorno es rotado un ángulo α alrededor del origen cambia la fase de la función límite pero su magnitud permanece constante, como se ve en la siguiente Ilustración:

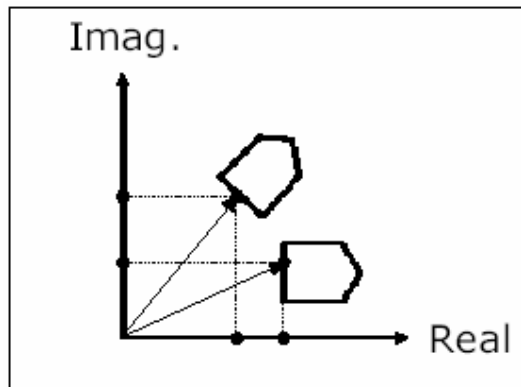


Ilustración II. PROPIEDADES DE ROTACIÓN DE LOS DESCRIPTORES DE FOURIER.

Por tanto los coeficientes de la Serie de Fourier adquieren una fase constante:

$$\gamma'(t) = \gamma(t) \cdot e^{j\alpha} \rightarrow c'_n = c_n \cdot e^{j\alpha}$$

CAMBIO EN EL PUNTO DE INICIO: un cambio en el punto de inicio equivale a empezar a trazar el contorno en un tiempo diferente de cero: $t_0 \neq 0$, es decir un desplazamiento de tiempo, por tanto los coeficientes de Fourier van a adquirir una fase constante:

$$\gamma'(t) = \gamma(t + t_0) \cdot e^{j\alpha} \rightarrow C'_n = C_n \cdot e^{\frac{j2\pi n}{t} \cdot t_0}, \forall n$$

DILATACIÓN O COMPRESIÓN DEL CONTORNO: si la función límite es escalada en un factor λ esto implica que los coeficientes de Fourier van a estar también multiplicados por este valor:

$$\gamma'(t) = \lambda \cdot \gamma(t) \rightarrow C'_n = \lambda \cdot C_n$$

$$Area = \iint_R dx \cdot dy = \sum_{n=\langle N \rangle} \pi \cdot n \cdot |c_n|^2$$

Ecuación 3. ÁREA ENCERRADA POR UN CONTORNO EN LA DILATACIÓN O COMPRESIÓN DEL CONTORNO.

APLICACIÓN DE LOS DESCRIPTORES DE FOURIER: Para el empleo de los descriptores de Fourier se requiere realizar los siguientes pasos:

1. Realizar una segmentación previa de la imagen con el fin de extraer el objeto de interés.
2. Extraer los bordes del objeto bien sea por medio de un filtrado paso alto mediante técnicas del gradiente o empleando métodos morfológicos. El

contorno del objeto debe ser conexo y de ancho un píxel.

3. Construir la función límite del objeto de interés recorriendo el contorno en sentido horario o antihorario.
4. Encontrar los coeficientes de la Serie de Fourier por medio de la ecuación de análisis aplicada a la función límite.
5. Analizar las características de los descriptores obtenidos.

Como ejemplo se van a obtener los descriptores de Fourier de una ilustración sencilla. A continuación se observan la figura original y la imagen de bordes.

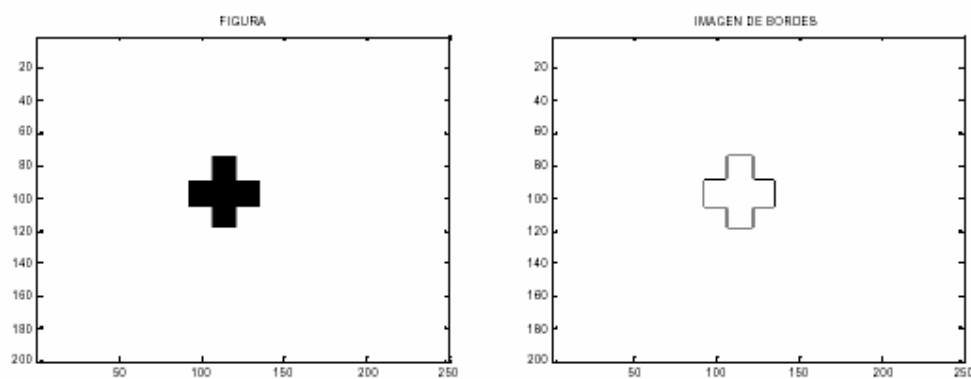


Ilustración III. ORIGINAL Y LA IMAGEN DE BORDES PARA OBTENER LOS DESCRIPTORES DE FOURIER.

Con base en la imagen de bordes se obtiene la función límite. Al aplicar las

series de Fourier a esta función se obtuvo el siguiente espectro:

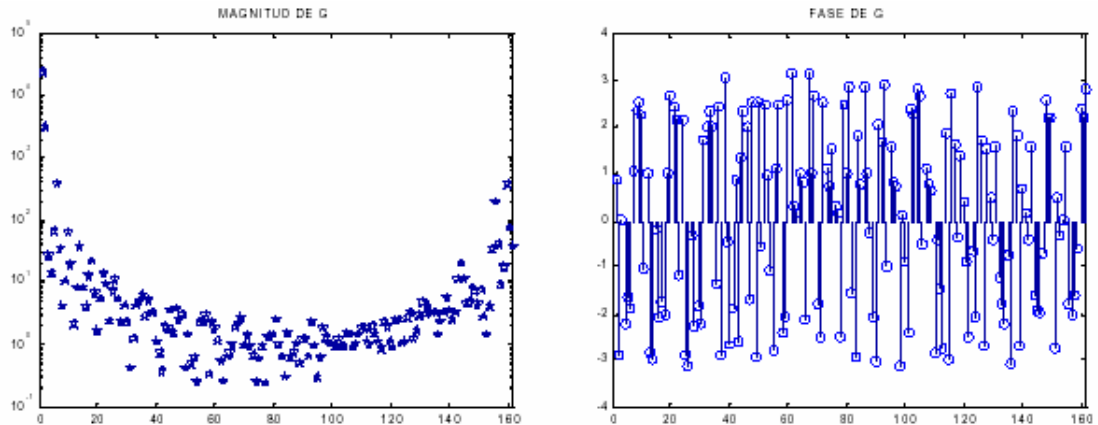


Ilustración IV. ESPECTRO.

La magnitud de los coeficientes de la serie de Fourier se visualizó con escala logarítmica para apreciar mejor cada una de las componentes. Puede verse que las primeras y últimas componentes tienen una magnitud considerablemente mayor que las componentes centrales, es decir, tienen mayor energía y esto es una característica que siempre se cumple en los descriptores de Fourier. Respecto a la fase no sigue un patrón uniforme y varía entre $-p$ y p .

A continuación se observa el proceso de reconstrucción del contorno tomando 2, 12, 112 y todos los coeficientes de la Serie de Fourier (162):

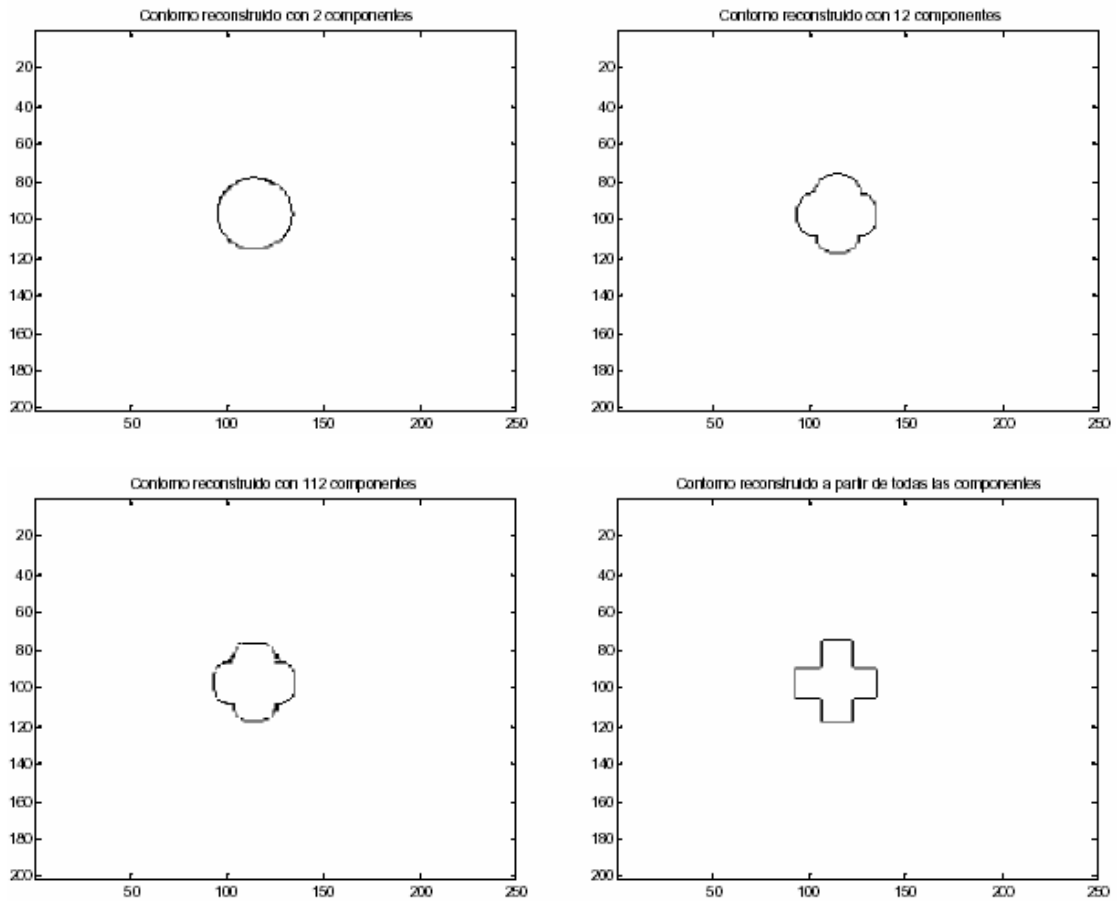


Ilustración V. PROCESO DE RECONSTRUCCIÓN DE CONTORNO DE LOS DESCRIPTORES DE FOURIER.

El resultado al reconstruir el contorno con sólo una componente es un punto, el cual da la información sobre el promedio de la función límite. Puede verse que al reconstruir con 2 componentes se obtiene una circunferencia. Al aumentar el número de componentes se empieza a observar una aproximación de la figura original. Sin embargo, como era de esperarse, no hay mayor diferencia entre la reconstrucción con 12 y con 112 componentes, ya que las componentes

intermedias tienen muy poca energía. Cuando se emplean la totalidad de coeficientes de la Serie de Fourier se obtiene la figura de bordes original sin distorsiones.

2 ANÁLISIS SOBRE LAS TEORÍAS REFERENTES A LA VISIÓN ARTIFICIAL

2.1 ENFOQUE DESDE LA APLICACIÓN PARA LA PERCEPCIÓN DE LAS MÁQUINAS DE FORMA AUTÓNOMA

En su libro *Visión por Computador* PAJARES, Gonzalo y DE LA CRUZ, Jesús M., afirman que el interés de los métodos de procesamiento de imágenes digitales se fundamenta en dos áreas principales de aplicación:

1. Mejora de la calidad para la interpretación humana.
2. Procesamiento de los datos de la escena para la percepción de las máquinas de forma autónoma.

En el intento por dotar a las máquinas de un sistema de visión aparece el concepto de visión artificial. Desde una perspectiva general, la visión por computador es la capacidad de la máquina para ver el mundo que le rodea, más precisamente para deducir la estructura y las propiedades del mundo tridimensional a partir de una o más imágenes bidimensionales.

El proceso general se sintetiza en la siguiente Ilustración:

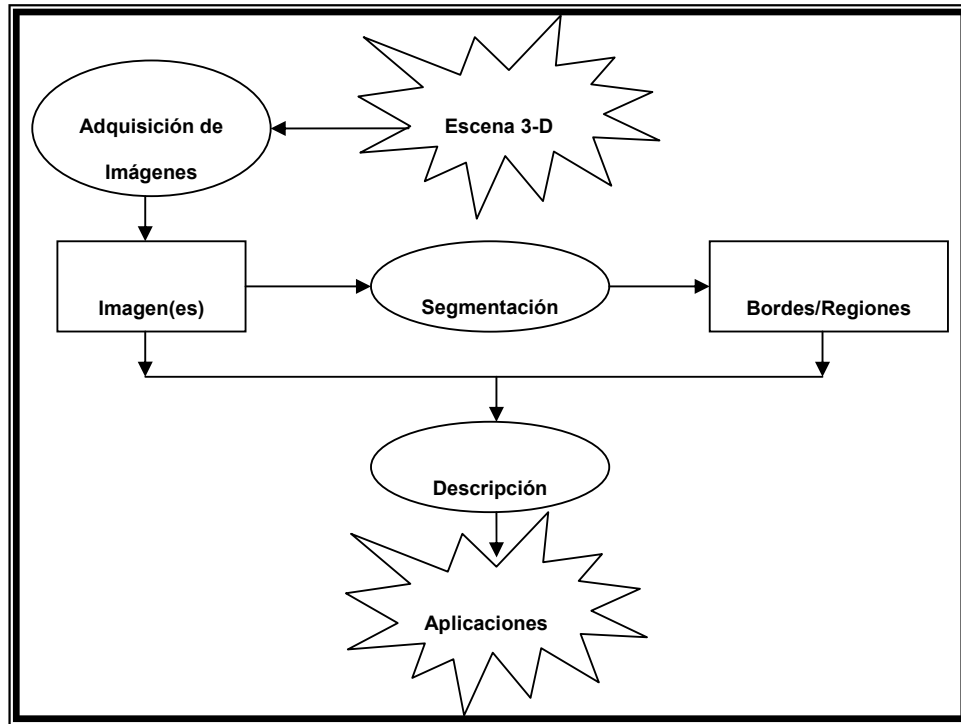


Ilustración VI. DIAGRAMA DE BLOQUES MOSTRANDO EL PROCESO COMPLETO DE LA VISIÓN ARTIFICIAL.

En la visión por computador, la *escena tridimensional* es vista por una, dos o más cámaras para producir imágenes monocromáticas o en color. Las *imágenes adquiridas* pueden ser *segmentadas* para obtener de ellas características de interés tales como *bordes o regiones*. Posteriormente, de las características se obtienen las propiedades subyacentes mediante el proceso de *descripción*. Tras lo cual se consigue *la estructura* de la escena tridimensional requerida para *la aplicación* de interés.

Existen cuatro grandes áreas de aplicación de la visión artificial:

1. Obtención de la distancia de los objetos a la escena tridimensional y estructura tridimensional.
2. Detección de objetos en movimiento.
3. Reconocimiento de Patrones y de Formas.
4. Reconocimiento de objetos tridimensionales.

Desde el punto de vista de la Robótica, K.S.F.U. rc.Gonzalez. Cs glee., en su libro Robótica, afirman que en la robótica al igual que sucede en el ser humano, la capacidad de visión dota al robot de un sofisticado mecanismo de percepción que permite a la máquina responder a su entorno en forma inteligente y flexible.

La visión artificial puede ser definida como los procesos de obtención, caracterización e interpretación de información de imágenes tomadas de un mundo tridimensional.

Estos procesos pueden ser subdivididos en seis áreas principales:

1. Captación.
2. Preprocesamiento.

3. Segmentación.
4. Descripción.
5. Reconocimiento.
6. Interpretación.

A estas áreas, K.S.F.U. rc.Gonzalez. Cs glee., las agrupa en tres niveles de procesamiento: visión de nivel bajo, medio y alto de acuerdo con la complicación y delicadeza que conlleva su implementación.

2.1.1 ETAPAS DE LA VISIÓN ARTIFICIAL

La etapas fundamentales del proceso de la visión artificial que se referencia en el artículo [Acosta 2004] se complementa con los niveles que describe K.S.F.U. rc.Gonzalez. Cs glee en la siguiente Ilustración.

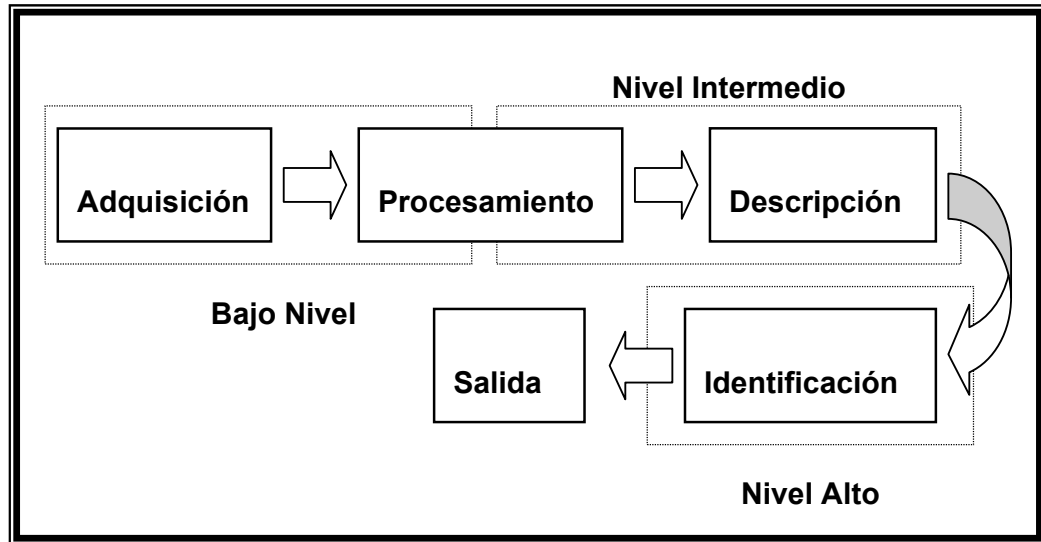


Ilustración VII. ETAPAS DE LA VISIÓN ARTIFICIAL.

2.1.1.1 Visión de Bajo Nivel

Se asocia a la visión de bajo nivel los procesos de *Captación* y *Preprocesamiento*, con procesos primarios en el sentido de que pueden ser considerados “reacciones automáticas” sin requerir ningún tipo de inteligencia.

2.1.1.1.1 Captación

La información se convierte en señales eléctricas por los sensores visuales. Cuando estas señales eléctricas se muestrean espacialmente, y se cuantifican en amplitud obtendremos de ellas una imagen digital.

2.1.1.1.2 Preprocesamiento

Se consideran dos procedimientos básicos de preprocesamiento:

1. El primero basado en técnicas del dominio del espacio (se trabaja directamente sobre el conjunto de píxeles que componen la imagen).
2. El segundo trabaja con el dominio de la frecuencia (se trabaja con un conjunto de píxeles complejos resultados de aplicar la transformada de Fourier). Los efectos del preprocesamiento se reflejan en las características de suavizado y realce.

2.1.1.2 Visión de Nivel Intermedio

En este nivel *la segmentación, la descripción y el reconocimiento* extraen, caracterizan y etiquetan componentes de la imagen que se obtiene de la visión de bajo nivel.

2.1.1.2.1 Segmentación

Es el proceso que divide una escena percibida en sus partes constituyentes u objetos. La segmentación es uno de los elementos más importantes en un sistema automático de visión ya que los objetos se extraen para su posterior reconocimiento. Los algoritmos de este proceso se basan en los principios de discontinuidad y similitud.

2.1.1.2.2 Descripción

El problema de la descripción en la visión consiste en extraer características de un objeto para reconocerle. Los descriptores deben ser independientes del tamaño, la localización y orientación del objeto y deben contener suficiente información de discriminación para distinguir un objeto de otro.

2.1.1.2.3 Reconocimiento de objetos individuales

Es un proceso de etiquetado, esto es, a cada objeto segmentado de una escena se le asigna una etiqueta.

2.1.1.3 Visión de Alto Nivel

La visión de alto nivel se refiere a procesamientos que tratan de emular la cognición. Mientras que los algoritmos para la visión de bajo y medio nivel conllevan un espectro de actividades bastante bien definidas, nuestro conocimiento y comprensión de los procesos de visión de alto nivel son considerablemente más difusos y especulativos. *La interpretación* dota a un sistema de visión de un nivel de conocimiento sobre su entorno superior al que pueden ofrecer cualquiera de los procesos vistos hasta ahora.

2.1.1.3.1 Interpretación

La interpretación incluye todos aquellos métodos que tengan que ver con la comprensión de una escena visual.

2.2 ENFOQUE DESDE EL ANÁLISIS AUTOMÁTICO DE IMÁGENES

Otro punto de vista generalizado de la visión artificial lo presenta GONZALEZ, Rafael C y WOODS, Richard E., en su libro Tratamiento Digital de Imágenes. Los autores identifican la visión artificial con un sistema automático de análisis de imágenes y describen los procesos con mayor profundidad.

Los sistemas automáticos de análisis de imágenes deberían ser capaces de ofrecer varios grados de inteligencia, que se manifiestan en las siguientes características:

1. Capacidad de extraer la información de interés, separándola de un conjunto de detalles irrelevantes.
2. Capacidad de aprender a partir de ejemplos y de generalizar este conocimiento para que se pueda aplicar en nuevas y diferentes circunstancias.
3. Capacidad de realizar inferencias a partir de una información incompleta.

Se puede diseñar e implementar sistemas de análisis de imágenes de éstas características para entornos limitados.

2.2.1 ETAPAS FUNDAMENTALES DEL PROCESAMIENTO AUTOMÁTICO DE ANÁLISIS DE IMÁGENES

Las etapas del tratamiento de imágenes son básicamente las mismas que se referencian en la visión artificial desde el punto de vista de la robótica, pero GONZALEZ, Rafael C y WOODS, Richard E., incluyen un nuevo elemento, la *base de conocimiento* que se relaciona con todas las etapas y le da el carácter de inteligente.

Las técnicas de análisis de imágenes también se agrupan en tres áreas básicas de procesado: de bajo nivel, de nivel intermedio y de alto nivel.

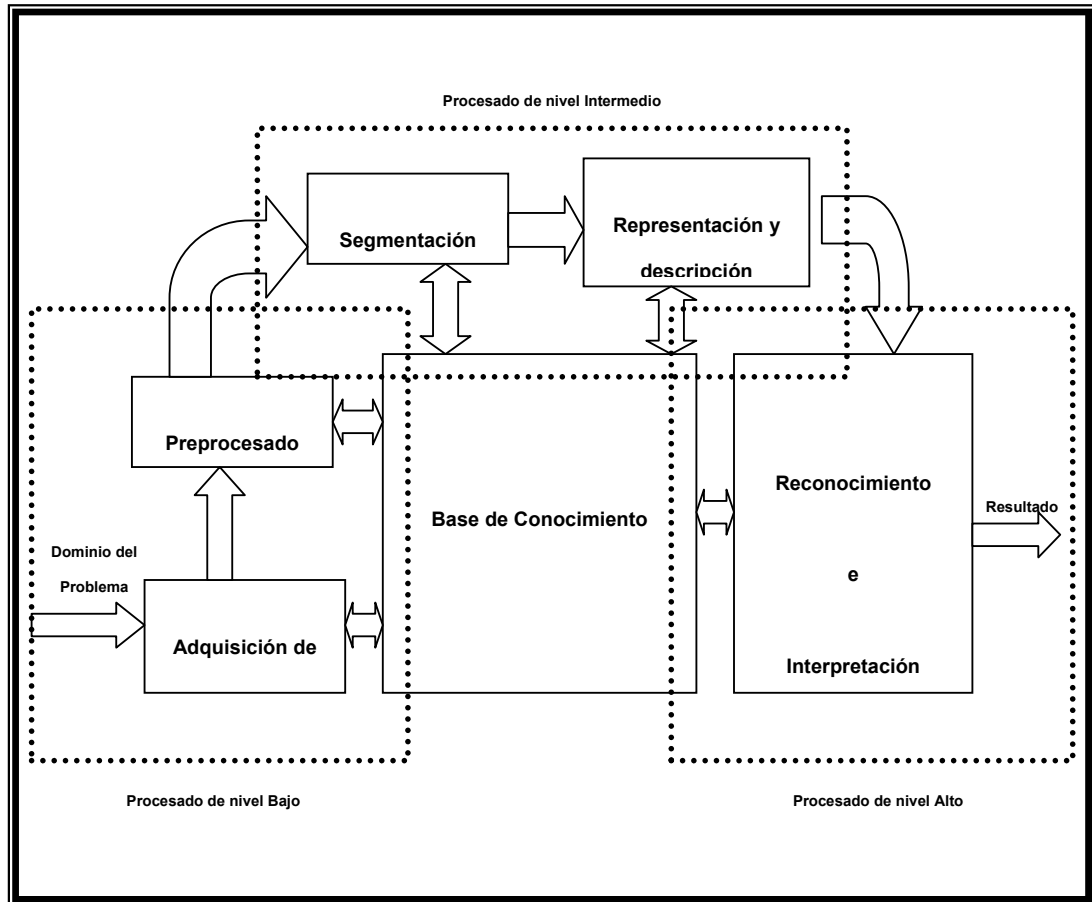


Ilustración VIII. ELEMENTOS DEL ANÁLISIS DE IMÁGENES.

2.2.1.1 Procesado de Bajo Nivel

Al igual que la visión artificial desde el punto de vista de la Robótica los procesos de *adquisición de imágenes* y *preprocesado* no requieren de inteligencia por parte del sistema de análisis de imágenes.

2.2.1.1.1 Adquisición de imágenes

Para la adquisición digital de imágenes se necesitan dos elementos.

1. El primero es un dispositivo físico sensible a una determinada banda de espectro de energía electromagnética (como la banda de rayos X, ultravioleta, visible o infrarrojo) y que produzca una señal eléctrica de salida proporcional al nivel de energía detectado.
2. El segundo, denominado digitalizador, es un dispositivo para convertir la señal de salida del sistema sensible a forma digital.

Una imagen de 8 bits y 1024 x 1024 píxeles necesita un millón de bytes de memoria para su almacenamiento. El resto de los sistemas de procesamiento de imágenes es definir la capacidad de almacenamiento adecuada que garantice la eficiencia del sistema. Se tienen 3 categorías básicas de almacenamiento:

1. Almacenamiento a corto plazo: se emplea en el preprocesamiento y consiste en emplear la memoria de una computadora o tarjetas especializadas denominadas memorias temporales.
2. Almacenamiento en línea: generalmente se emplean discos magnéticos y magnético – óptico (MO). El factor clave que caracteriza el almacenamiento en línea es el acceso frecuente a los datos.

3. Almacenamiento en archivo: se caracteriza por necesitar un almacenamiento masivo pero de acceso poco frecuente. Las cintas magnéticas y los discos ópticos son los medio habituales en las aplicaciones de archivado.

2.2.1.1.2 Preprocesamiento de la Imagen

Luego de la formación de la imagen en sí, se pasa a las compensaciones, tales como la reducción de ruido o la supresión del emborronamiento de la imagen, para estudiar la obtención de características de la imagen primitiva. Esta sucesión de procesamientos puede ser comparada con los procesos de percepción y adaptación que un hombre lleva a cabo cuando intenta encontrar un asiento en un teatro oscuro, inmediatamente después de haber estado paseando durante una tarde soleada. El proceso inteligente de conseguir una butaca desocupada no podrá dar comienzo hasta que no esté disponible una imagen adecuada.

El objetivo de las técnicas de mejora de la imagen es procesar una imagen de forma que resulte más adecuada que la original para una aplicación específica.

Existen dos categorías básicas:

1. Métodos en el dominio espacial.

2. Métodos en el dominio de la frecuencia.

2.2.1.2 Procesado de Nivel Intermedio

Se refiere a la labor de extracción y caracterización de los componentes de la imagen que se obtiene de un proceso de bajo nivel. Este nivel abarca *la segmentación y la descripción*. Es necesario construir algunas capacidades para el comportamiento inteligente dentro de procedimientos flexibles de segmentación.

2.2.1.2.1 Segmentación

El nivel de segmentación que se lleva a cabo en esta etapa de subdivisión depende del problema a resolver. La segmentación deberá detenerse cuando los objetos de interés de una aplicación hayan sido aislados. Las técnicas de segmentación dependen de las características del objeto que se quiere analizar. La asignación autónoma de la técnica de segmentación apropiada es una de las tareas más difíciles del procesamiento de imágenes. Esta etapa del proceso determina el eventual éxito o fracaso del análisis de la imagen.

Los algoritmos de segmentación de imágenes monocromáticas generalmente se basan en una de las dos propiedades básicas de los valores de niveles de gris: discontinuidad y similitud. En la primera categoría, el método consiste en dividir una imagen en los cambios bruscos de nivel de gris. Las principales

áreas de interés de esta categoría son la detección de puntos aislados y la detección de líneas y bordes de una imagen. Los principales métodos de la segunda categoría están basados en la umbralización, crecimiento de región, división y fusión de regiones.

2.2.1.2.2 Representación

Después de segmentar una imagen en regiones, el conjunto resultante de píxeles segmentados se representa y describe normalmente en forma adecuada para su posterior procesamiento por computadora. Básicamente, el *representar* una región implica dos posibilidades:

1. Hacerlo en términos de sus características externas (su contorno).
2. Hacerlo en términos de sus características internas (los píxeles que comprenden la región).

La elección del esquema de representación es, sin embargo, solamente una parte de la tarea de hacer los datos útiles para una computadora. La práctica normalizada es utilizar esquemas de representación que compacten los datos en representaciones que son más útiles en el cálculo de los descriptores. Como esquemas de representación se tienen:

1. Códigos de Cadena.

2. Aproximaciones Poligonales.
3. Firmas.
4. Lados del Contorno.
5. Esqueleto de una Región.

El paso siguiente consiste en *describir* la zona en la representación elegida.

2.2.1.2.3 Descriptores

Las características seleccionadas como *descriptoras* deberían ser tan insensibles como fuera posible a variaciones tales como cambios de tamaño, traslación y rotación.

La descripción de un objeto se refiere al hecho de extraer los detalles o rasgos que contienen alguna información de interés o información fundamental para diferenciar los objetos que resulten involucrados en la escena. Para esto, existen varios métodos que pueden facilitar la manipulación de las características de dichos objetos entre los mas importantes encontramos los códigos de cadena, la transformada de fourier y los momentos invariantes de Hu.

2.2.1.2.3.1 Códigos de cadena

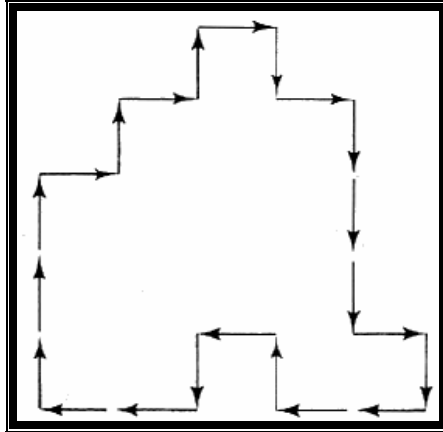


Ilustración IX. REPRESENTACIÓN DE UN CONTORNO EN CODIGOS DE CADENA

Esta es una forma de representación de fronteras basada en el trabajo de Freeman. Consiste en seguir el contorno del objeto de la imagen según las direcciones de los píxeles vecinos, así, dependiendo de la exactitud con que se quiera trabajar se pueden generar cuatro (*4-connected neighbors*) u ocho (*8-connected neighbors*) posibles direcciones de los píxeles vecinos por ejemplo, para las 8 conexiones se manejan 8 direcciones Este, Noreste, Norte, Noroeste, Oeste, Suroeste y Sureste. Cada una de las direcciones puede ser representada por un string de 3 bits.

Esta representación presenta la desventaja de que los resultados de las codificaciones son variantes al escalado y al rotado del objeto, por

consiguiente esta muy limitado para situaciones donde el objeto cambie de forma en el tiempo o presente rotación alguna.

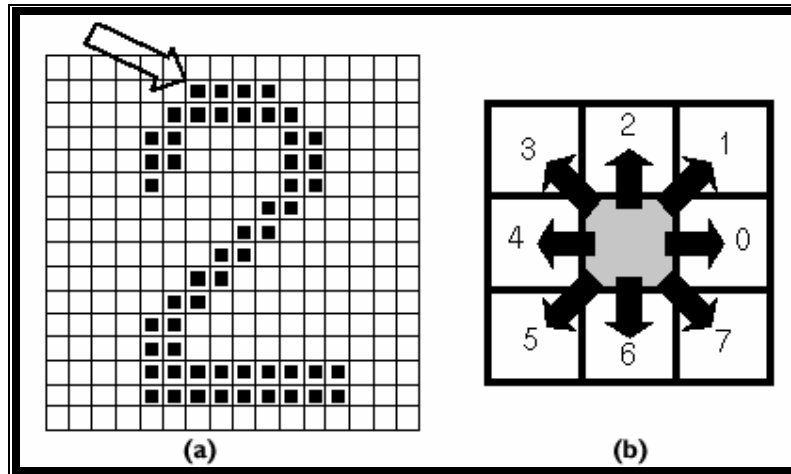


Ilustración X. (a) CÓDIGOS DE CADENA DEL NÚMERO 2 (b) MAPA DE PÍXELES VECINOS (8-CONNECTED)

2.2.1.2.3.2 Momentos

La forma de los lados del contorno (y de las firmas) se puede describir cuantitativamente utilizando momentos. Con el fin de ver cómo se puede conseguir esto considérese la siguiente Ilustración. En la figura a se muestra el lado de un contorno, y la figura b se muestra el mismo lado representado como una función unidimensional $g(r)$ de una variable r arbitraria.

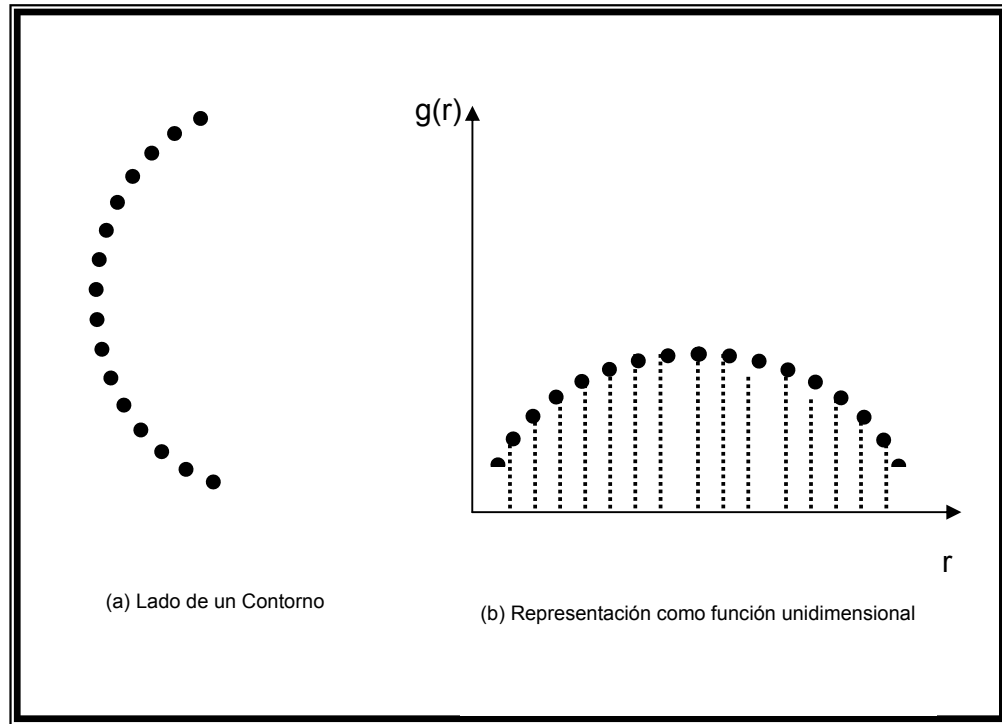


Ilustración XI. REPRESENTACIÓN DE UN CONTORNO EN MOMENTOS.

Tratando la amplitud de g como una variable aleatoria v y formando un histograma de amplitud $p(v_i)$, $i = 1, 2, \dots, k$, siendo k el número de incrementos de amplitud discretos. Entonces, el momento n -ésimo de v respecto a su media es:

$$\mu_n(v) = \sum_{i=1}^k (v_i - m)^n p(v_i)$$

Ecuación 4. MOMENTO N-ÉSIMO DE V CON RESPECTO A SU MEDIA EN DESCRIPTORES DE MOMENTOS.

Donde

$$m = \sum_{i=1}^k v_i p(v_i)$$

Siendo la cantidad m la media o valor medio de v y μ_2 su varianza. Generalmente se necesitan los primeros momentos para diferenciar firmas de formas claramente distintas.

Un método alternativamente consiste en normalizar $g(r)$ por unidad de área y tratarlo como un histograma. En este caso, r se convierte en la variable aleatoria y los momentos son:

$$\mu_n(r) = \sum_{i=1}^L (r_i - m)^n g(r_i)$$

Donde

$$m = \sum_{i=1}^L r_i g(r_i)$$

En esta notación, L es el número de puntos del contorno y $\mu_n(r)$ está directamente relacionado con la forma de $g(r)$. Las dos representaciones del

momento se pueden utilizar simultáneamente para describir un lado del contorno o la firma.

Básicamente lo que hemos conseguido es reducir la tarea de descripción a la de describir funciones unidimensionales. Aunque los momentos son el método más popular, no son los únicos descriptores que se pueden utilizar para alcanzar este objetivo. Por ejemplo otro método implica calcular la Transformada Discreta de Fourier, obtener su espectro y utilizar los k primeros componentes del espectro para describir $g(r)$. La ventaja de los momentos sobre otras técnicas es que la implementación de los descriptores anteriores es discreta y que además contienen también una interpretación física de la forma del contorno.

2.2.1.2.3.3 Momentos Invariantes

En el libro *Visión por Computador*, PAJARES, Gonzalo; DE LA CRUZ, Jesús M., hablan de los momentos invariantes como el conjunto de momentos que son invariantes a los efectos de traslación, rotación y cambio de escala para cada uno de los puntos interiores que componen una región.

Tomemos $f(x, y)$ como la intensidad del punto (x, y) en una región. El momento de orden $(p + q)$ para la región se define como (PAJARES, DE LA CRUZ)

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y)$$

Ecuación 5. MOMENTOS DE ORDEN (p+q) PARA UNA REGIÓN.

Donde la sumatoria se toma sobre todas las coordenadas espaciales (x, y) de puntos de la región. El momento central de orden (p + q) viene dado por,

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y)$$

Ecuación 6. MOMENTOS CENTRAL DE ORDEN (p+q) PARA UNA REGIÓN.

Donde

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad \bar{y} = \frac{m_{01}}{m_{00}}$$

Los momentos centrales normalizados de orden (p + q) se definen como:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma} \text{ donde } \gamma = \frac{p+q}{2} + 1 \text{ para } (p+q) = 2, 3, \dots$$

2.2.1.2.3.4 Momentos Invariantes de Hu

El siguiente conjunto de momentos invariantes propuesto por Hu (1962) se puede obtener usando únicamente los momentos centrales normalizados de:

$$\begin{aligned}
 \phi_1 &= \eta_{20} + \eta_{02} \\
 \phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
 \phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
 \phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
 \phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + \\
 & (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{12} + \eta_{03})^2] \\
 \phi_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
 \phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + \\
 & (3\eta_{12} - \eta_{03})(\eta_{12} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]
 \end{aligned}$$

Ecuación 7. MOMENTOS INVARIANTES DE HU.

Para que todos los momentos contribuyan por igual en la función de discriminación, y por tanto estén dentro del mismo orden de magnitud, a veces es necesario proceder a una normalización del tipo:

$$\phi'_n = \text{abs} \ln(\text{abs}(\phi_n))$$

Este conjunto de momentos resulta ser invariante a la traslación, la rotación y el cambio de escala (Hu 1962).

Hupkens y Clipper (1995) presentan una normalización de estos momentos de forma que llegan a alcanzar un alto grado de insensibilidad al ruido. Esta normalización resulta ser la siguiente:

$$\begin{aligned}
 C_1 &= 1 \\
 C_2 &= \frac{\phi_2}{\phi_1^2} \\
 C_3 &= \frac{\phi_3}{\phi_1^3} \\
 C_4 &= \frac{\phi_4}{\phi_1^3} \\
 C_5 &= \frac{\phi_5}{\phi_1^6} \\
 C_6 &= \frac{\phi_6}{\phi_1^4} \\
 C_7 &= \frac{\phi_7}{\phi_1^6}
 \end{aligned}$$

2.2.1.2.3.5 La Transformada Discreta del Coseno

La Transformada Discreta del Coseno unidimensional (TDC) se define como:

$$C(u) = \alpha(u) \sum_{x=0}^{N-1} f(x) \cos \left[\frac{(2x+1)u\pi}{2N} \right]$$

Ecuación 8. LA TRANSFORMADA DISCRETA DEL COSENO (TDC).

Para $u = 0, 1, 2, \dots, N-1$. De forma similar, la TDC inversa se define como:

$$f(x) = \sum_{u=0}^{N-1} \alpha(u) C(u) \cos \left[\frac{(2x+1)u\pi}{2N} \right]$$

Ecuación 9. LA TDC INVERSA.

Para $x = 0, 1, 2, \dots, N-1$. En ambas ecuaciones, α es:

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & \text{para } u = 0 \\ \sqrt{\frac{2}{N}} & \text{para } u = 1, 2, \dots, N-1 \end{cases}$$

El correspondiente par TDC bidimensional es:

$$C(u, v) = a(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \left[\frac{(2x+1)u\pi}{2N} \right] \cos \left[\frac{(2y+1)v\pi}{2N} \right]$$

Para $u, v = 0, 1, 2, \dots, N-1$, y

$$f(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \alpha(u)\alpha(v) C(u, v) \cos \left[\frac{(2x+1)u\pi}{2N} \right] \cos \left[\frac{(2y+1)v\pi}{2N} \right]$$

Para $x, y = 0, 1, 2, \dots, N-1$, donde α viene dado por la ecuación anterior.

Los sistemas más prácticos de codificación por transformación se basan en la TDC, que ofrece un buen compromiso entre la capacidad para empaquetar información y la complejidad del cálculo. De hecho, las propiedades de la TDC han resultado ser de tal valor práctico que se han convertido en estándar internacional para los sistemas de codificación por transformación.

2.2.1.2.3.6 La Transformada Discreta de Fourier

La figura siguiente muestra un borde de N puntos de una imagen digital, si empezamos en un punto arbitrario (X_0, Y_0) , los pares de coordenadas (X_0, Y_0) , (X_1, Y_1) , ..., (X_{n-1}, Y_{n-1}) representan dicho borde.

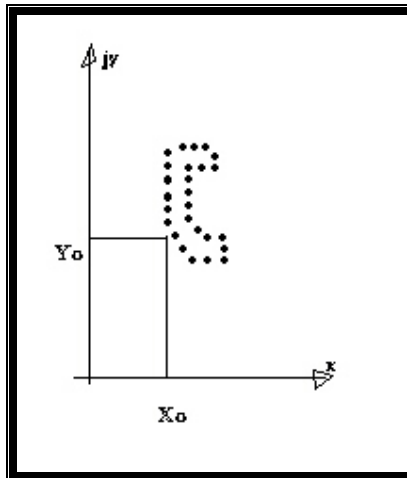


Ilustración XII. REPRESENTACIÓN DEL BORDE COMO UNA SECUENCIA DE COORDENADAS

Estamos representando al borde como una secuencia de coordenadas:

$$\mathbf{s(k)} = [X(k), Y(k)] \text{ para } k=0,1,2,\dots,N-1$$

Por lo tanto cada par de puntos podemos tratarlo como un número complejo:

$$\mathbf{s(k)} = X(k) + jY(k) \text{ para } k=0,1,2,\dots,N-1$$

El eje X se considera el eje real y el eje Y se considera el eje imaginario. La gran ventaja de esta consideración es que pasamos de un problema en dos dimensiones a un problema en una dimensión.

La transformada discreta de Fourier de $s(k)$ es:

$$a(u) = \frac{1}{N} \sum_{k=0}^{N-1} s(k) \exp [-j2\pi uk/N]$$

$$\text{para } u=0,1,2,\dots,N-1$$

Los coeficientes complejos $a(u)$ son llamados descriptores de fourier del borde de la figura.

La transformada inversa de fourier de los descriptores $a(u)$ recupera el borde original y se define como:

$$s(k) = \sum_{u=0}^{N-1} a(u) \exp [j2\pi uk/N]$$

para k=0,1,2..N-1

Supongamos que en vez de considerar todos los $a(u)$, sólo tomamos los M primeros coeficientes. Esto es lo mismo que considerar los $a(u)=0$ para $u>M-1$.

El resultado de esta aproximación es:

$$s(k) = \sum_{u=0}^{M-1} a(u) \exp [j2\pi uk/N]$$

para k=0,1,2,...,N-1

Aunque sólo M términos son usados para obtener cualquier componente de $s(k)$, k sigue variando de 0 a $N-1$, esto quiere decir, que un borde aproximado existen el mismo número de puntos que en el original pero se usan menos términos para la reconstrucción de dichos puntos. Cuanto mayor sea el valor de M mayor detalle se puede recuperar de la imagen original.

Los descriptores, en general, deben ser lo menos sensibles que se pueda a la traslación, la rotación y el cambio de escala. Los descriptores de fourier no son directamente insensibles a los cambios geométricos en la figura, pero dichos cambios se pueden reflejar con simples transformaciones sobre los

descriptores. Además son muy útiles para simplificar determinadas operaciones complejas sobre imágenes.

Propiedades de los descriptores de fourier

Tienen varias características que los hacen muy interesantes, y que son consecuencia del uso de la Transformada Discreta de fourier:

1. Compactación: En la mayoría de las figuras (salvo las que sean especialmente abruptas, con muchos picos) para obtener un buen nivel de fidelidad no se necesitan nada más que unos pocos descriptores, pudiéndose desechar el resto sin pérdidas de información considerables. Esto es así porque la energía suele estar en los primeros coeficientes del desarrollo.
2. Insensibilidad a desplazamientos: Cuando movemos una figura sólo se modifica el primero de sus descriptores, que es el que nos proporciona el "centro" de la figura (viene a ser como la componente de continua de la señal). Si nosotros sólo buscamos comparar formas, este coeficiente no nos interesa.
3. Insensibilidad a la rotación: En el plano complejo, una rotación de un ángulo A , equivale a multiplicar todos los puntos por una exponencial compleja: $\exp(j \cdot A)$. Si realizamos una normalización de los

coeficientes, los factores comunes que tengan todos los coeficientes desaparecen.

4. Escalado: Si obtenemos los descriptores de un contorno continuo $g(l)=A*f(l)$ sólo aparecería también el factor de escalado en los coeficientes, que se elimina de nuevo normalizando adecuadamente. Pero, al usar un contorno discreto, si aumentamos el tamaño del contorno (manteniendo la densidad de pixels) también aumenta el número de puntos de la secuencia, y por consecuencia el número de coeficientes de la DFT. Esto provoca un efecto de escalado en frecuencia, que expresado sin mucho rigor (debido a los truncamientos) sería:

$$g[A*n]=A*f[n] \Rightarrow F[k]=(1/A)*G[A*k]$$

2.2.1.3 Procesado de Alto Nivel

2.2.1.3.1 Reconocimiento e interpretación

Es el proceso en el cual se establecen diferencias entre un objeto y otro. Existen varias técnicas para el reconocimiento y la interpretación de una imagen:

1. Métodos de decisión teórica para el reconocimiento.

2. Métodos estructurales para el reconocimiento.
3. Métodos para la interpretación de imágenes.

Las principales técnicas de reconocimiento de decisión teórica son:

1. Los clasificadores de mínima distancia.
2. Los correladores.
3. Los clasificadores de Bayes.
4. Las redes neuronales.

En el reconocimiento estructural los patrones se representan en forma simbólica (como cadenas y árboles), y los métodos de reconocimiento se basan en el emparejamiento de símbolos o en modelos que tratan a los patrones de símbolos como sentencias de un lenguaje artificial. La interpretación de una imagen consiste en asignar un significado a un conjunto de elementos reconocidos en dicha imagen. Las técnicas actuales para la interpretación de imágenes se basan en la lógica de predicado, en redes semánticas y en sistemas de producción (en concreto en Sistemas Expertos).

2.2.1.3.2 Reconocimiento de Patrones: El Perceptrón y el Perceptrón

Multicapa

Este enfoque es tomado de PAJARES, Gonzalo y DE LA CRUZ, Jesús M., de su libro *Visión por Computador* donde nos introducen a las redes neuronales haciendo alusión a McCulloch y Pitts (1943), quienes propusieron modelos de neuronas en la forma de dispositivos binarios basados en un umbral y algoritmos estocásticos que implicaban cambios binarios 0 – 1 y 1 – 0 en los estados de las neuronas como la base para el modelado de sistemas neuronales. El trabajo posterior de Hebb (1949) estaba basado en modelos matemáticos que intentaban capturar el concepto de aprendizaje por refuerzo o asociación.

2.2.1.3.2.1 Aprendizaje

El aprendizaje es el proceso de estimación de una dependencia desconocida (entrada, salida) o estructurada de un sistema utilizando un número limitado de observaciones como se muestra en la siguiente ilustración. El esquema de aprendizaje incluye tres componentes:

1. Un generador de vectores de entrada aleatorios.
2. Un sistema que proporciona una salida para un vector de entrada dado.

3. Una máquina de aprendizaje que estima una relación desconocida (entrada, salida) del sistema a partir de las muestras observadas (entrada, salida).

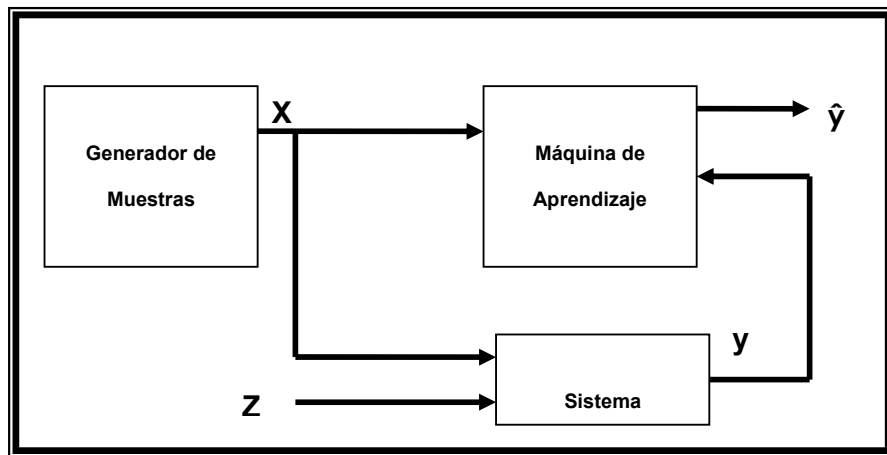


Ilustración XIII. MÁQUINA DE APRENDIZAJE UTILIZANDO OBSERVACIONES DEL SISTEMA PARA OBTENER UNA APROXIMACIÓN DE SU SALIDA.

Esta formulación es muy general y describe muchos problemas de aprendizaje prácticos encontrados en ingeniería y estadística, tales como:

1. Estimación de la densidad de probabilidad (paramétrica y no paramétrica) a partir de las muestras.
2. Clasificación o estimación de los límites de decisión de las clases.

3. Agrupación de datos en torno a sus vectores representativos (“cuantización vectorial/agrupamiento”).
4. Regresión o estimación de una función continua a partir de muestras ruidosas.

En reconocimiento de patrones resultan de interés los tres primeros.

Veamos en detalle cada componente de la máquina de aprendizaje:

2.2.1.3.2.2 Generador

El generador o distribución de muestras produce vectores aleatorios $x \in Rd$, estos vectores determinan una función de densidad de probabilidad $p(x)$, que es desconocida. En terminología estadística esta situación se llama observacional.

2.2.1.3.2.3 Sistema

El objetivo del aprendizaje consiste en estimar las dependencias desconocidas entre las variables de entrada “ x ” y de salida “ y ” a partir de un conjunto de observaciones pasadas de valores (x, y) . En la Ilustración anterior el conjunto de variables etiquetadas como Z indican el resto de factores que afectan a las salidas pero que no son observados o controlados. Por ejemplo en el proceso de captura de imágenes, la presencia de suciedad o imperfecciones en el

sistema óptico pueden ser cantidades de este tipo. Por tanto, el conocimiento de los valores observados (x) no especifica de forma unívoca las salidas (y). Esta incertidumbre en las salidas refleja la laguna de conocimiento de los factores no observados (Z), de aquí se deriva una dependencia estadística entre las entradas observadas y la salida o salidas. El efecto de las entradas no observadas puede caracterizarse por una distribución de probabilidad condicional $p(y/x)$, que expresa la probabilidad de que ocurra y dada la entrada x . En definitiva, el sistema produce un valor de salida y para cada vector de entrada x de acuerdo a la densidad condicional $p(y/x)$, que es también desconocida. Obsérvese que esta descripción incluye el caso específico de sistemas deterministas donde $y = f(x)$.

2.2.1.3.2.4 Máquina de Aprendizaje

En el caso más general, la máquina de aprendizaje es capaz de implementar un conjunto de funciones $f(x, w)$, $w \in \Omega$ donde Ω es un conjunto de parámetros abstractos utilizados solamente para indexar el conjunto de funciones. En esta formulación el conjunto de funciones implementadas por la máquina de aprendizaje puede ser cualquier conjunto de funciones, elegidos previamente, antes que el proceso de aprendizaje haya comenzado.

Además, podemos distinguir dos tipos de aprendizaje:

1. Aprendizaje Supervisado: utilizado para estimar una relación desconocida (entrada, salida) a partir de muestras conocidas (entrada, salida). La clasificación y la regresión caen en este grupo.
2. Aprendizaje No supervisado: solamente se proporciona al sistema de aprendizaje las muestras de entrada y no existe noción alguna de salida durante el aprendizaje.

El término supervisado se corresponde con el hecho de que los valores de salida para las muestras de entrenamiento son conocidos y, por lo tanto, proporcionados por un servidor o por un sistema que está siendo modelado.

El objetivo del aprendizaje no supervisado puede ser estimar la distribución de probabilidad de las entradas o descubrir la estructura natural (agrupamientos o clusters en inglés) en los datos de entrada.

2.2.1.3.3 La Red Retropropagación (Backpropagation)

Vamos a considerar una red clásica para abordar un problema multiclase, basada en varias capas de neuronas. Este modelo ha sido ampliamente considerado en la teoría de reconocimiento de patrones.

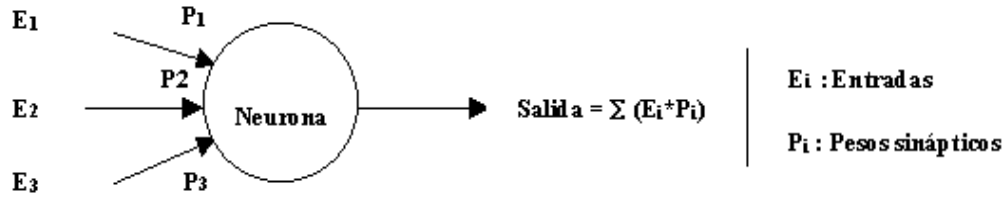


Ilustración XIV. MODELO DE RED NEURONAL PERCEPTRÓN.

Las entradas provienen de otras neuronas y el proceso que se realiza en su interior hará que si el valor de salida sobrepasa un cierto umbral predeterminado, ésta se activará llevando el impulso a la siguiente neurona ó bien a la salida de la red.

Conectando entre sí muchos de estos módulos organizados en capas y a la vez interconectando esas capas conseguimos organizar una red neuronal, de la que existen en la actualidad muchos tipos, dependiendo del problema que se quiera resolver.

La Red neuronal multicapa retropropagación consta de capas de neuronas de cómputo estructuradas de forma idéntica y colocadas de manera que la salida de cada neurona en una capa proporciona la entrada de cada neurona en la siguiente capa. El número de neuronas en la primera capa, llamada capa A, es N_A . A menudo, $N_A = d$, es la dimensión de los vectores de entrada, es decir de los patrones. El número de neuronas en la capa de salida, llamada capa Q, es N_Q . Este número es igual al número de clases c para las que la red se va a

entrenar y sobre las que luego podrá clasificar, es decir, reconocer un determinado vector de entrada. La red reconoce un vector x como perteneciente a la clase c_i si la salida i -ésima de la red es “alta” mientras las otras salidas son “bajas”. Cada neurona tiene la misma forma que el modelo del perceptrón, donde la función de activación debe ser derivable con el fin de poder desarrollar la regla de entrenamiento mediante retropropagación. La siguiente función de activación sigmoideal tiene la necesaria derivabilidad.

$$h_j(I_j) = \frac{1}{1 + \exp \left[- (I_j + \theta_j) / \theta_0 \right]}$$

Ecuación 10. FUNCIÓN DE ACTIVACIÓN SIGMOIDAL

Donde $I_j, j = 1, 2, \dots, N_j$, representa la entrada al elemento de activación de cada nodo en la capa J de la red, θ_j es un parámetro de ajuste y θ_0 controla la forma de la función sigmoideal.

2.2.1.3.3.1 Entrenamiento por Retropropagación

Comenzamos en la capa de salida. El error al cuadrado total entre las respuestas deseadas, r_q , y las correspondientes respuestas actuales, O_q , de los nodos en la capa Q de salida:

$$E_Q = \frac{1}{2} \sum_{q=1}^{N_Q} (r_q - O_q)^2$$

Donde N_Q es el número de nodos en la capa de salida Q y el factor de $\frac{1}{2}$ se utiliza por conveniencia en la notación para tomar la derivada más tarde. E_Q resulta ser en este caso la función de riesgo empírico definida en la ecuación anterior. El objetivo consiste en desarrollar una regla de entrenamiento similar a la regla delta, que permita el ajuste de los pesos en cada una de las capas, buscando un mínimo a una función de error como lo vemos en la anterior ecuación. Como en el caso de la regla delta, ajustando los pesos en proporción a la derivada parcial del error con respecto a los pesos se logra este resultado. En otras palabras:

$$\Delta w_{qp} = -\alpha \frac{\partial E_Q}{\partial w_{qp}}$$

Donde la capa P precede a la capa Q, Δw_{qp} es tal y como se define en:

$$w(k+1) = w(k) - \alpha(k) \sum_{i=1}^n \frac{(w^t x - fd_i) x_i}{\|x_i\|}$$

Y α es un incremento de corrección positivo.

3 ANÁLISIS DE LOS DESARROLLOS DE LOS SISTEMAS AUTÓMATICOS DE RECONOCIMIENTO DE BLANCOS

3.1 SISTEMA CLÁSICO DE ATR

El campo de ATR (también llamado Automatic Target Detection ATD) ha sido ampliamente estudiado durante las últimas dos décadas, pero sin éxitos definitivos. La falta de éxito puede ser atribuida a varias razones tal como la carencia de adecuados sensores y la falta de mecanismos apropiados para incorporar el conocimiento experto. Una de las grandes dificultades en la detección de objetos en imágenes es la falta de habilidad para definir los objetos en cuestión. El ejemplo clásico es la clase de objeto llamado “silla”. ¿Cómo se puede decir qué hace que una silla sea una silla? La mayoría de personas puede reconocer una silla cuando la ve, o podemos reconocer un objeto cuando está siendo usado como una silla (tal como una roca donde alguien está sentado), pero somos incapaces de dar una definición satisfactoria que permita extraer las características de lo que es una silla [Gader, 2001].

Reconocimiento Automático de Blancos (Automatic Target Recognition) es el término dado al campo de las ciencias de la ingeniería que estudian los sistemas y técnicas diseñadas para identificar, a partir de imágenes y datos, la localización y los atributos correspondientes a objetivos físicos específicos (blancos), sin la intervención directa de operadores humanos [Augustyn,

1992] [Bhanu, 1986]. La “Tecnología de Reconocimiento” hace referencia a los presentes o futuros sistemas que tienen el potencial de proveer un avance cuantitativo en las capacidades de los sistemas de reconocimiento [Gader, 2001]. Este avance puede ocurrir como el resultado de tres desarrollos convergentes:

1. Mayores avances en la tecnología de los sensores.
2. Mayores avances en la tecnología de los sistemas de procesamiento de datos.
3. El uso de técnicas de computación suave para inferir conclusiones a partir de los datos observados.

Los dos principales objetivos de los sistemas de ATR son la detección y la clasificación de blancos. La clasificación es el proceso de agrupar objetos dentro de clases (sub-poblaciones), de acuerdo con sus características comunes. El reconocimiento de un objeto individual en una clase única es llamado identificación.

La clasificación implica una forma de aprendizaje la cual induce a partir de atributos y antecedentes las clases que son consecuentes. De otra parte, el razonamiento es un proceso que aplica reglas generales, tales como

ecuaciones, relaciones, etc., a una colección inicial de datos hechos y demás, para deducir una decisión o resultado.

El reconocimiento es una forma de razonamiento, mientras que la clasificación es una forma de aprendizaje [Looney, 1997]. El término adquisición es frecuentemente usado para significar detección, y los términos identificación y reconocimiento son también usados para remplazar el término clasificación.

Un tercer objetivo, el seguimiento de blancos, es frecuentemente considerado parte de los sistemas de ATR. El término “blanco” puede ser usado para referirse a una gran variedad de objetos, aunque su uso ha sido históricamente reducido al contexto de las aplicaciones militares (por ejemplo: tanques, armas, camiones, jeeps, aviones, helicópteros, buques, etc.).

El campo de los ATR se encuentra muy asociado a las operaciones militares. Es en este contexto donde los sistemas de ATR difieren de su forma más general, el Reconocimiento Automático de Objetos (Automatic Object Recognition AOR), puesto que los sistemas de ATR deben soportar los problemas relativos a las interferencias producidas por los humanos como el camuflaje y los requerimientos de tiempo real para operación de los sistemas.

Los algoritmos componentes de un sistema clásico de ATR pueden ser descompuestos en cuatro bloques principales: pre-procesamiento, detección,

segmentación y clasificación, tal como lo muestra la siguiente Ilustración.

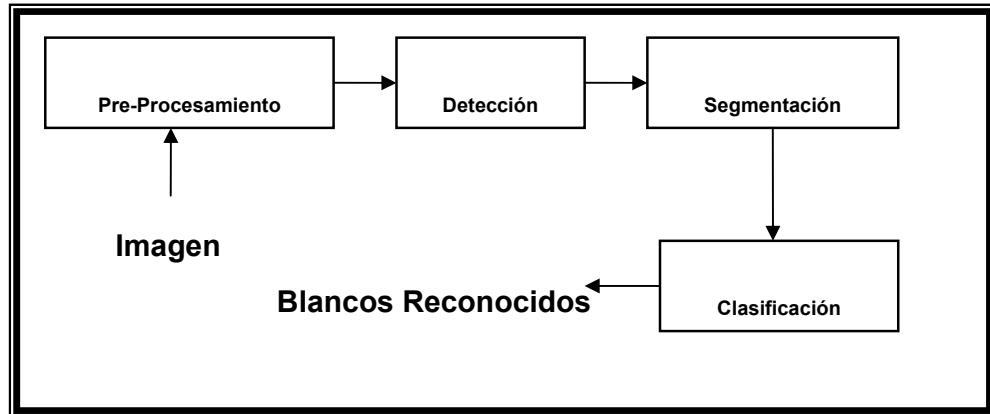


Ilustración XV. COMPONENTES DE UN SISTEMA CLÁSICO DE ATR.

3.1.1 PREPROCESAMIENTO

El propósito de este paso consiste en reducir el ruido en la imagen de entrada y mejorar el contraste del blanco con el fin de facilitar su detectabilidad. Algunas aproximaciones incluyen métodos de detección de bordes en esta etapa. En tales casos, filtros estadísticos (mediana, media, moda), paso-bajo, son usados para reducir el ruido tratando siempre de preservar los bordes como estructuras fundamentales [Bhanu, 1993]. También en esta etapa, algunas transformaciones tales como Fourier, Wavelet o Hough son realizadas con el fin de facilitar la extracción de información del blanco.

3.1.2 DETECCIÓN

El propósito de este bloque es obtener una imagen binaria que indique que píxeles en la imagen original parecen pertenecer a los blancos. Las técnicas más simples involucran el uso de umbrales para separar los píxeles correspondientes al blanco de los del entorno.

$$f(i, j) \geq R_{umbral} \Rightarrow g(i, j) = 1 \quad \text{(Blanco)}$$

$$f(i, j) < R_{umbral} \Rightarrow g(i, j) = 0 \quad \text{(Entorno)}$$

Donde $f(i, j)$ y $g(i, j)$ son los valores de los píxeles con coordenadas (i, j) de las imágenes de entrada y salida respectivamente. El valor del umbral R_{umbral} puede ser determinado arbitrariamente o estimado a partir de las probabilidades de distribución del blanco y su entorno con un estudio del histograma.

Una técnica mas avanzada para la detección de dicho umbral es determinar una ventana la cual incluye un área alrededor del blanco pero no cerca de su vecindad, y estimar el valor medio de los píxeles de esa ventana. Si el valor del píxel examinado es mayor que el valor promedio de la ventana, el píxel es seleccionado como blanco, de otra forma es seleccionado como entorno.

Mejores resultados han sido obtenidos cuando el proceso de decisión incluye no sólo la media, sino también la desviación estándar [Novak, 1993] [Barboy,

1986]. Algunas variaciones a esta idea han sido propuestas en diversas fuentes bibliográficas. Otras aproximaciones incluyen el uso de funciones de discriminación de características locales, para seleccionar puntos de interés del blanco, tales como el operador gradiente, o detectores de bordes.

3.1.3 SEGMENTACIÓN

El propósito de esta etapa es extraer los blancos del entorno tan exactamente como sea posible mediante el uso de la información brindada por el detector. Los píxeles que son presumidos como blancos son estudiados por grupos para buscar blancos. Primero las falsas aceptaciones son rechazadas, y los huecos del interior de los blancos reales deben ser rellenados.

Otras técnicas usan una aproximación de detección piramidal en una sucesión de imágenes de baja resolución para extraer los blancos. Algunos algoritmos realizan la detección y segmentación en una única etapa. La etapa de segmentación finaliza con la computación de las coordenadas de los diferentes blancos, mediante estimación de área, cálculo del centroide, etc.

La segmentación de imágenes tiene su origen en numerosos estudios psicológicos que indican la preferencia de los humanos por agrupar regiones visuales en términos de proximidad, similitud y continuidad, para construir un conjunto de unidades significativas. Existe cierta confusión en torno al concepto

de segmentación ya que si bien algunos autores consideran suficiente marcar los puntos de la imagen (píxeles) con un valor indicativo de su pertenencia a determinada región o clase, otros indican que además es necesario proveer un mecanismo que permita una representación simbólica de las relaciones topológicas existentes entre las distintas unidades. Esta discrepancia se debe fundamentalmente al nivel de abstracción al cual se asocia el proceso de segmentación. Si se asocia la segmentación a un nivel de abstracción medio, ésta deberá producir la representación simbólica anteriormente indicada, correspondiendo el etiquetado a un proceso previo en un nivel bajo de abstracción y encaminado al objetivo de segmentar. Cabe reconocer que existen situaciones o aplicaciones donde no merece la pena soportar la carga computacional que supone la obtención de dicha representación simbólica. Esto suele ser cierto solamente en casos donde la tarea a realizar o el entorno están muy delimitados y controlados.

En [Suetens, 1992] se puede encontrar un buen resumen de las estrategias empleadas en la segmentación y análisis de imágenes. La siguiente Ilustración indica las diversas técnicas de segmentación según el nivel de abstracción. Esta división es principalmente conceptual ya que si bien está presente en gran número de implantaciones, también resulta común encontrar métodos que integran aspectos de más de un nivel.

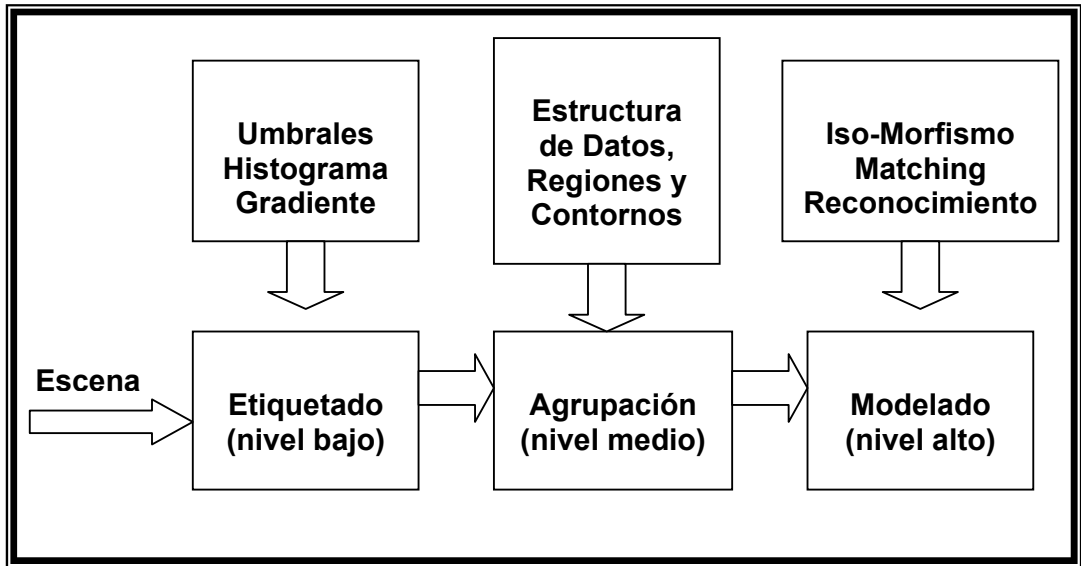


Ilustración XVI. NIVELES DE SEGMENTACIÓN.

3.1.4 CLASIFICACIÓN

El propósito de este bloque es estudiar los blancos segmentados e identificarlos correctamente. Esta etapa está presente en los sistemas donde la detección no es suficiente y se requiere una información más precisa del blanco. Un método ampliamente usado para clasificación es la comparación de patrones. En el método convencional de comparación de la correlación, una imagen patrón del blanco es deslizada sobre las regiones interés de la imagen original para obtener la mejor correlación. Las regiones de interés se definen como un bloque que incluyen las localizaciones del blanco a partir del bloque de segmentación, aunque la correlación directa con la imagen original es

también posible. Para un patrón de imagen dado h y una región de interés de la imagen f , la selección de la mejor correlación se obtiene mediante:

$$\sum_n \sum_m f(p+m, q+n)h(m, n) \geq K C_h \Rightarrow s(i, j) = 1$$

Seleccionado en (p, q)

$$\sum_n \sum_m f(p+m, q+n)h(m, n) < K C_h \Rightarrow s(i, j) = 0$$

No Seleccionado

Donde $C_h = \sum_n \sum_m h(p+m, q+n)h(m, n)$ es la autocorrelación del patrón, la cual es usada para normalización, y K es una constante para definir el nivel de umbral de detección ($0 < K < 1$).

Otra clase de correspondencia de patrones es la comparación de siluetas donde, en lugar de correlacionar imágenes, los valores binarios de las siluetas son deslizados en las regiones de interés de las imágenes segmentadas para buscar igualdades.

Los patrones pueden ser de blancos completos o de partes del blanco. El principal problema con las técnicas de comparación de patrones es que los resultados son muy sensibles a las variaciones geométricas tales como la rotación y escalado. Para enfrentar este problema un amplio número de patrones que contienen las posibles apariencias geométricas del blanco son usados en las correlaciones, volviendo pesado computacionalmente el

proceso. Mediante una confiable selección de características (geométricas, topológicas y/o espectrales) puede ser llevada a cabo computacionalmente una eficiente clasificación. Dichas características maximizan la similitud de los objetos de la misma clase y minimizan la similitud de los objetos de diferentes clases [Bhanu, 1986].

Algunos esquemas de clasificación que han sido usados para ATR son:

1. Vecindad K-más cercana (K-nearest neighbor).
2. Discriminadores lineales y cuadráticos.
3. Clasificadores basados en árboles de decisión.
4. Redes neuronales multicapa.

Debido a sus aplicaciones en biomedicina, industria y equipos militares, los sistemas de vigilancia infrarroja, requieren de algoritmos de detección automática de blancos (ATD-Automatic Target Detection) con probabilidades de falsa alarma cercanas al 0% y probabilidades de detección cercanas al 100% [Bruce, 2001].

Estas aplicaciones presentan un gran número de retos para el desarrollo de algoritmos de detección automática de blancos, no sólo desde el punto de vista

de sus aplicaciones, sino también por la naturaleza dinámica del problema, los elevados niveles de ruido y la variabilidad del mundo real.

Normalmente los procesos de detección de blancos son altamente dependientes de un conocimiento previo. La mayoría de los sistemas son capaces únicamente de reconocer un número pre-especificado de blancos y son limitados en su rendimiento ante nuevos retos [Howard, 2000]. Bajo estas condiciones los sistemas de detección de blancos a partir de imágenes infrarrojas tienden a presentar un bajo rendimiento.

3.2 APLICACIONES DE SISTEMAS CLÁSICOS DE ATR

En esta sección se presenta una serie de sistemas de Reconocimiento Automático de Blancos que utilizan sistemas clásicos para llevar a cabo las tareas de detección y/o clasificación de blancos. Esta presentación no intenta hacer un exhaustivo listado de todas las aproximaciones de ATR, sino pretende hacer una generalización de los sistemas más representativos.

3.2.1 SISTEMAS DE ATR BASADOS EN MODELOS (MODEL-BASED ATR)

A principios de los años 80, el término "Reconocimiento Automático de Blancos Basado en Modelos" emerge en el contexto del ATR cuando algunos investigadores empezaron a usarlo para describir el desempeño de algunos sistemas desarrollados hasta entonces. Kegelmeyer y Webster [Kegelmeyer,

1988] señalan que el término tiene variados significados para diferentes autores. Este ha sido usado para describir:

1. Modelos de superficie y volumétricos de objetos tridimensionales.
2. Coeficientes de auto-regresión, momentos basados en características, descriptores de Fourier.
3. Grafos semánticos para describir objetos en imágenes.
4. Comportamiento de sensores y procesos físicos.
5. Propiedades de las imágenes y los blancos.

Las aproximaciones basadas en modelos utilizan una combinación de técnicas orientadas a datos, técnicas orientadas a modelos y técnicas orientadas a objetivos, para encontrar objetos de interés. De otra parte, las aproximaciones no basadas en modelos utilizan técnicas orientadas a datos, donde ninguna información del blanco es conocida con anticipación [Sadjadi, 1991]. Partiendo de la premisa “que todos los sistemas de reconocimiento de blancos implícita o explícitamente utilizan modelos”, Zelnio [Zelnio, 1991] separa los sistemas de ATR en tres diferentes paradigmas:

1. Prefiltrado, Segmentación y Clasificación (Prescreen, Segment and Classify PSC): este paradigma intenta separar la forma o silueta del

blanco del entorno sin usar ninguna información acerca del blanco (geometría, contraste, etc.). Tal vez motivado por la habilidad humana de hacer la percepción de la figura.

2. Filtros de Igualado (Matched Filter MF): este paradigma usa una estrategia completamente diferente, en la cual el sistema intenta muestrear completamente la geometría de la figura usando filtros de igualado o patrones para cada geometría posible del blanco a ser encontrado. Para solucionar los problemas de variabilidad del contraste del blanco las aproximaciones de MF usan filtros sensibles a la información de alta frecuencia. Los sistemas MF requieren conocer por adelantado toda la información disponible acerca de las posibles geometrías del blanco.

3. Visión Basada en Modelos (Model-Based Vision MBV): este paradigma se encuentra en un punto intermedio entre los dos extremos (PSC y MF). Los sistemas basados en MBV no están restringidos a usar toda o ninguna información previa para encontrar el blanco, pero pueden usar modelos o estadísticas acerca del blanco para desarrollar el reconocimiento. Los sistemas MBV inician con un proceso parecido al PSC para segmentar la imagen en un manejable número de sub-imágenes, las cuales, luego de ser refinadas, son operadas por un

proceso MF para reconocer los blancos con la ayuda de una colección de modelos.

Luego de una cercana observación de los paradigmas anteriores se encuentra que la mayoría de los sistemas de ATR podrían encontrarse en la tercera categoría, soportando la premisa de que todos los sistemas de ATR están en alguna forma basados en modelos.

Una división diferente a los paradigmas de ATR es dada por Theis [Theis, 1989] donde usa las tres categorías siguientes:

1. Sistemas de ATR basados en Estadística (históricamente los más usados).
2. Sistemas de ATR basados en Modelos (los cuales dependen inherentemente de la geometría del blanco).
3. Sistemas de ATR basados en Redes Neuronales (los cuales intentan simular la operación del cerebro humano).

Kegelmeyer y Webster [Kegelmeyer, 1988] definen tres clases de modelos:

1. Modelos Heurísticos: constan de una colección de propiedades que caracterizan los datos de interés, los cuales son inherentemente no numéricos. Ejemplos de esta clase pueden ser las frases “los

vehículos con ruedas no se encuentra en los lagos”, o “los blancos son elípticos y de tamaño medio”. Estos modelos son generalmente sencillos de entender pero complejos de implementar.

2. Modelos de Firma: son aquellos que permiten una reproducción poco exacta de la imagen. Estos modelos tienden a ser insensibles a las variaciones causadas por los efectos físicos y ópticos de la imagen, considerando factores tales como los patrones de irradiación, oclusión, efectos atmosféricos y características del sensor como ancho de banda, ruido, etc. Estos modelos son usualmente generados por algoritmos estocásticos y pueden llegar a ser muy exactos pero también complejos.
3. Modelos Paramétricos: estos modelos pueden reproducir completamente los blancos en un sentido posiblemente estocástico a partir de un vector de parámetros con una menor dimensión que los datos a ser modelados. Un ejemplo puede ser un segmentador de umbral óptimo para una probabilidad de distribución de entorno y del blanco conocida. La fortaleza de las aproximaciones paramétricas es que éstas pueden proveer una solución completa y analizable al problema de ATR. Su debilidad radica en que los modelos paramétricos completos son muy difíciles de derivar.

Sadjadi [Sadjadi, 1991] plantea que la base del paradigma de reconocimiento de objetos basado en modelos radica en preservar tanto como sea posible la información del objeto y mediante el uso de un razonamiento inteligente, puede realizarse una mejor interpretación de la escena. Los modelos pueden ser separados en cinco diferentes grupos:

1. Los modelos de blancos: son los primeros en importancia y el centro del esfuerzo del modelado.
2. Los modelos de la vegetación del entorno: estos han sido ampliamente estudiados en el área de los sensores remotos.
3. Los modelos atmosféricos: también llamados modelos de medios de transmisión.
4. Los modelos de los sensores: tales como la caracterización de los visores infrarrojos, radares, sonares, láseres, etc.
5. Modelos de interacciones entre modelos: integran los modelos de nivel inferior para obtener modelos cohesionados.

3.2.2 SISTEMAS DE ATR MULTIESPECTRALES (MULTIESPECTRAL ATR)

El uso de diferentes bandas del espectro electromagnético para hacer el análisis de una escena ha sido ampliamente usado y, recientemente, se ha

convertido en una importante herramienta en los sistemas de ATR, que busca otras fuentes de información para mejorar su rendimiento.

Trivedi [Trivedi, 1987] utiliza el conocimiento de las características de reflectividad de algunos materiales en ciertas bandas para reconocer objetos en imágenes reales. Más que el uso de diferentes bandas para hacer el reconocimiento, se busca seleccionar la banda en la cual se maximizan los efectos de reflectividad frente a otros materiales, obteniendo así la respuesta deseada.

Kahle y Goetz [Kahle, 1983] hace un estudio del uso de múltiples bandas infrarrojas para clasificar diferentes clases de materiales mediante las características de poder ser emitido entre ellos. Ellos utilizan un escáner de seis canales llamado TIMS (Thermal Infrared Multispectral Scanner). El escáner cubre el espectro desde los 8.2 hasta los 12.2 micrones, sin dejar vacíos. El sistema puede clasificar correctamente diferentes materiales usando una técnica especial para desplegar las diferencias espectrales como diferencias de color.

Algunos trabajos usan similares aproximaciones para hacer mejorar en imágenes aéreas [Price, 1987], o para mejorar la detectabilidad de los blancos [Stocker, 1990]; aquí algunos algoritmos son empleados utilizando información espectral brindada por un sensor infrarrojo pasivo. Trivedi [Trivedi, 1985]

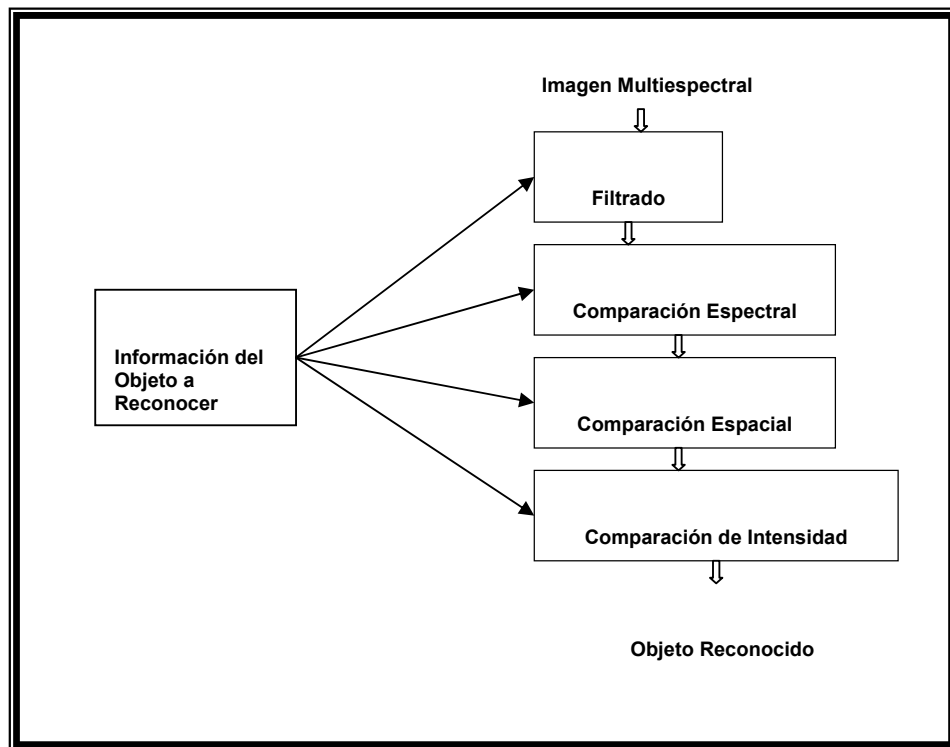
presentan una lista simple de operaciones requeridas para reconocer objetos en imágenes multiespectrales. Estos pasos son:

1. Filtrado, usado básicamente para remover el ruido.
2. Procesado de la información espectral, usado en cada una de las imágenes de las bandas para limitar los rangos de intensidad de los distintos blancos, basado en las propiedades de reflectividad del material de los blancos buscados.
3. Comparación de atributos espaciales, para filtrar las formas no deseadas.
4. Comparación de los perfiles de intensidad, donde se realiza el reconocimiento de los objetos.

Borough [Borough, 1991] usa más de 101 diferentes bandas espectrales para formar un array tridimensional (un conjunto de 101 diferentes imágenes obtenidas mediante la separación de una imagen en bandas de frecuencia, así cada una representa una diferente longitud de onda) para reconocimiento de materiales (las formas no son consideradas). Este estudio crea una escena sintética consistente de varios materiales. Mediante el uso de la reflectividad de los diferentes materiales y la pintura típica del camuflaje el sistema reconoce la existencia de las áreas con presencia de camuflaje sin falsas alarmas en los

experimentos con 101 bandas y 50% de falsas alarmas cuando se usan sólo 12 bandas.

La siguiente Ilustración presenta un diagrama de bloques de estas operaciones:



**Ilustración XVII. OPERACIONES PARA RECONOCER OBJETOS EN
IMÁGENES MULTIESPECTRALES.**

Hoff et al [Hoff, 1991] utiliza la diferencia promediada entre dos imágenes (dos bandas) para detectar los blancos. Esta operación permite cancelar el ruido, el cual tiene una naturaleza espectral gris, en comparación con la de los

blancos. Una de las dos imágenes es promediada por una constante. El estudio considera tres diferentes aproximaciones para obtener el peso óptimo. Dos de estas tratan de optimizar la diferencia basándose en objetivos probabilísticos para maximizar la relación señal-ruido y para minimizar la varianza de salida. La diferencia simple (no pesada) es la tercera aproximación.

3.2.3 SISTEMAS DE ATR PARA RECONOCIMIENTO DE BUQUES

Burgess [Burgess, 1993] desarrolló un sistema completo de detección de buques, basado en pares de imágenes. El sistema usa dos escenas reales: una imagen multiespectral SPOT y una imagen satelital LANDSAT. Más que detección el sistema realiza un reconocimiento de dos clases (Buque o No buque).

La primera operación desarrollada es eliminar la tierra en cada imagen, seguidamente a cada imagen se le aplica un filtro paso-alto de realce, luego las dos imágenes son selectivamente combinadas para producir la imagen en la cual se desarrolla el análisis.

El análisis consta de cuatro pasos:

1. El primero es un refinamiento por área, donde los objetos que son demasiado grandes o demasiado pequeños son descartados como

posibles buques.

2. Luego viene el refinamiento por análisis de formas, donde más objetos son eliminados por su elongación basada en algunos conceptos de momentos y centroides usados para calcularlas.
3. Luego el refinamiento por intensidad máxima, que permite eliminar ruido.
4. Finalmente el refinamiento por firma espectral para rechazar los posibles contornos de tierra con forma de buque.

Los resultados presentados son muy satisfactorios en el sentido de que el sistema detecta correctamente 18 buques a partir de dos escenas diferentes.

3.2.4 SISTEMAS MULTISENORES DE ATR

Algunos sistemas han usado diferentes sensores para incrementar el ratio de detección y/o reconocimiento, y reducir el ratio de falsa alarma tomando ventaja de las características específicas de las diferentes clases de sensores.

Blanquart et al. [Blanquart, 1990] propone la fusión de imágenes infrarrojas (3-5 micrones) e imágenes de radar. Las imágenes corresponden a la misma escena, pero con diferentes ubicaciones del sensor. Debido a esto la imagen debe ser rotada y proyectada para conseguir una buena correspondencia entre

las dos. El estudio tiene dos objetivos: la navegación y el reconocimiento.

Las imágenes de radar, dada su baja resolución, son usadas para determinar la naturaleza de las diferentes regiones mientras que las imágenes infrarrojas son usadas para definir los bordes de las regiones previamente determinadas. El proceso de reconocimiento es simplemente para confirmar los resultados obtenidos por la detección de regiones por radar, usando técnicas simples de comparación.

En Roggemann et al [Roggermann, 1989] se presenta un sistema para detectar blancos de forma separada mediante imágenes infrarrojas e imágenes de distancias, y los resultados son combinados para obtener finalmente la detección de blancos. Para las imágenes FLIR es desarrollado un proceso de segmentación de imágenes a partir del histograma de intensidades mediante el uso de un umbral de brillantes. Este proceso es desarrollado en imágenes que han sido previamente filtradas con un filtro de mediana.

De otra parte, las imágenes de distancias son segmentadas para extraer las regiones planas. Lo anterior partiendo de que los blancos han sido modelados como una colección de regiones planas. Los algoritmos de detección se encuentran divididos en cuatro:

1. Sólo FLIR con un ratio de detección de 0.71.

2. Sólo distancias con un ratio de detección de 0.68.
3. Mapeo de FLIR en las imágenes de distancias con un ratio de detección de 0.78.
4. Con el mejor desempeño el mapeo de distancias en imágenes FLIR con un ratio de detección de 0.92.

3.2.5 SISTEMAS DE ATR MULTIDIMENSIONALES

Kolodzy [Kolodzy, 1993] también combina sensores activos y pasivos en el proceso de reconocimiento de blancos. Los sensores pasivos son sistemas de visión infrarrojos, mientras que el sensor activo es un sistema LADAR (Laser-radar). En la etapa de detección la característica buscada como potencial blanco en cada imagen del LADAR es una región con distancias constantes.

El tamaño de cada región debe estar dentro de unos límites predefinidos. Las imágenes infrarrojas se utilizan para discriminación, seleccionando umbrales de intensidad. Este estudio utiliza el concepto del histograma LADAR/FLIR, el cual es una versión combinada de los histogramas de ambos tipos de imágenes, para ayudar al proceso de detección. El estudio presenta un sistema completo de ATR compuesto de seis módulos.

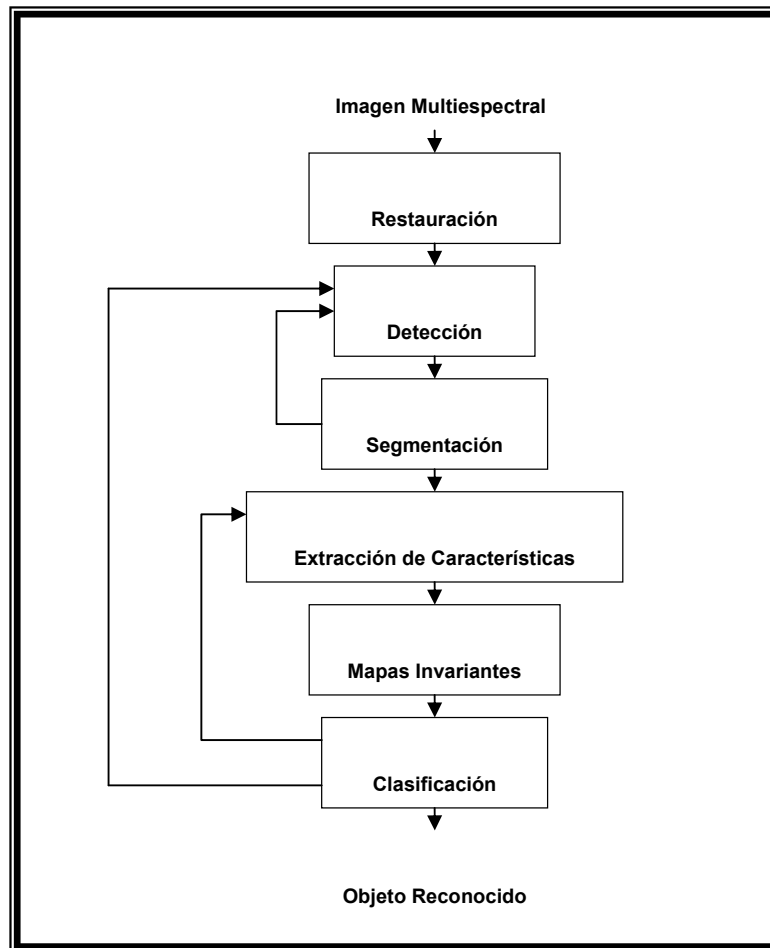


Ilustración XVIII. MÓDULOS DEL SISTEMA DE ATR DE KOLODZY.

El primer módulo es el de restauración de imagen mediante un proceso de filtrado para la eliminación de ruido.

Luego en el módulo de detección se utilizan los histogramas LADAR/FLIR para hallar áreas de distancia constante y una significativa firma térmica. Estas regiones son separadas del entorno y luego en la etapa de segmentación un

algoritmo de detección de bordes detecta y conecta los contornos de los objetos.

El módulo de extracción de características selecciona la imagen provista por la etapa anterior como característica para la etapa de clasificación.

Luego el sistema utiliza el concepto de mapeo invariante de los bordes de la silueta (Invariant Mapping of Silhouette Edges) para obtener representaciones de los objetos inmunes a rotación, traslación y escalado.

La inmunidad a la traslación es obtenida usando el centroide de la silueta como el origen del mapa espacial. La inmunidad a la rotación es obtenida mediante el mapeo del contorno de los objetos contra el ángulo polar.

Finalmente, mediante el cálculo del logaritmo, es obtenida la inmunidad del sistema al escalado.

El módulo de clasificación está compuesto de una red neuronal multicapa que de forma autónoma separa las entradas en dos clases diferentes mediante clusterización.

3.2.6 SISTEMAS XTRS DE ATR

XTRS es un sistema experimental de reconocimiento automático de blancos

basado en modelos, desarrollado por el laboratorio Lincon, para reconocer vehículos en imágenes reales o sintéticas [Verly, 1992].

El sistema XTRS consiste de dos módulos que operan de forma independiente y comparten la misma máquina de decisiones: XTRS-C, el cual opera en el modo basado en contornos (CB), y XTRS-R, que opera en el modo basado en regiones (RB).

XTRS-C trabaja con imágenes de distancias producidas por un canal del sensor Ladar, mientras que XTRS-R utiliza ambos canales del Ladar, es decir imágenes de distancia y de intensidad, las cuales se combinan para formar las imágenes de interés.

En ambos sistemas los eventos de interés son siluetas de distancias. XTRS-C utiliza algoritmos de detección de bordes para obtener las siluetas de las imágenes de distancias, mientras XTRS-R utiliza algoritmos de segmentación por regiones para obtener las imágenes de interés. Ambas aproximaciones utilizan librerías de modelos con las descripciones de las apariencias esperadas de los blancos sobre un variado rango de aspectos.

Cada modelo posee una estructura similar a la de una red semántica que describe el objeto en términos de tamaños y formas de sus partes observables y las relaciones entre ellas.

Para hacer la caracterización ambos sistemas utilizan dos etapas:

1. La primera es una extracción de eventos a partir de imágenes de entrada sin procesar.
2. La segunda es la descomposición de dichos eventos en eventos simples.

En XTRS-C el primer paso es desarrollado mediante la estimación de valores para los píxeles donde el sensor LADAR falla en el cálculo de la distancia, y luego se utiliza un detector de bordes en la imagen resultante.

El segundo paso es desarrollado mediante una aproximación polinomial que divide el contorno en una serie de secciones cóncavas y convexas, donde cada una es etiquetada como una esquina o arco.

En el sistema XTRS-R, el primer paso empieza limpiando la imagen de distancias como en el XTRS-C y luego una imagen de interés es producida mediante la integración de una imagen de intensidades umbralizada, una imagen de distancias derivadas y una imagen de distancias indicando la presencia de objetos con forma de palos tales como antenas y cañones.

Luego las imágenes de interés resultantes son segmentadas para obtener regiones de interés predefinidas. Esas regiones son luego limpiadas para

remover huecos, reconectar segmentos del cuerpo principal, etc. La segunda etapa es desarrollada mediante la extracción de contornos en las imágenes creadas y luego realizando una descomposición poligonal como en el XTRS-C.

La librería de modelos para el XTRS consiste de una especialización jerárquica donde los hijos de cada nodo representan subcategorías o partes de éste. La jerarquía también contiene algunas funciones de relación (ejemplo: encima de... o al lado de...) para describir las posibles relaciones entre las subregiones o partes.

Ambos sistemas, XTRS-R/C, realizan un proceso de comparación de los modelos jerárquicos con los eventos extraídos a partir de las imágenes de entrada. Este proceso se realiza usando árboles jerárquicos para obtener el grado de pertenencia de cada imagen con un nodo determinado. Esta fase es llamada comparación simbólica.

Luego en la fase de evaluación de evidencias el grado de pertenencia computado es usado para cuantificar el aporte de cada pieza de evidencias. El resultado final es una medida de veracidad asignada a cada una de las posibles hipótesis, inclusive las que han sido etiquetadas como blancos desconocidos.

4 LIMITACIONES EN VISIÓN ARTIFICIAL Y LOS SISTEMAS DE RECONOCIMIENTO

Este análisis se realiza teniendo en cuenta tres proyectos de grado realizados en la Universidad Tecnológica de Bolívar [Acosta, 2004] [Campillo, 2003] [Barrios, 2001] sobre reconocimiento de imágenes, en los cuales se realizaron varias pruebas de campo y algunas guías de laboratorio que utilizaban diferentes métodos de extracción de características, Y adicionalmente se diseñó e implementó un software llamado PSD, donde se puede comprobar el uso de tres descriptores como son: Códigos de cadena, Fourier y Momentos invariantes de hu.

PSD es un sistema para el experimento, prueba de exactitud y eficacia de tres descriptores. Este sistema se basa en una etapa posterior al tratamiento de las imágenes por esto trabaja con imágenes binarizadas ya tratadas y que solo representan el contorno de un objeto o de una persona. (Ver Anexo D)

4.1 VENTAJAS Y DESVENTAJAS DE LOS MÉTODOS DE EXTRACCIÓN DE CARACTERÍSTICAS

4.1.1 ESTUDIO MATRICIAL DE LA IMAGEN

La ventaja se basa en que se define claramente cuál es la figura que se está analizando, hay mayor rapidez en la comparación de las imágenes y el reconocimiento de las figuras se hace en un periodo de tiempo más corto que en los demás métodos.

La desventaja es que se limita solamente a reconocer las figuras que son objeto de análisis y de cualquier figura geométrica plana. No utiliza redes neuronales artificiales. (Este método es usado en una de las tesis).

En el desarrollo del software PSD no se ha incluido este método pues carece de un modelo matemático definido y los resultados son bastante deficientes.

4.1.2 CÓDIGO DE CADENA

La ventaja radica en que describe el contorno geométrico de una imagen. Como desventajas vemos que se limita al estudio de contornos totalmente cerrados, se produce un vector característico muy grande para ser utilizado como entrada de la red y no todos los vectores generados por el código de cadena son del mismo tamaño. Asimismo es dependiente del tamaño de la

imagen para generar el vector. Esto se podría compensar utilizando imágenes de la misma resolución, por ejemplo mediante la transformada de wavelets.

Otra desventaja es que el tiempo de procesamiento es excesivamente grande y se obtiene un error de convergencia muy grande lo cual hace que se confunda muchas veces la red cuando reconoce una imagen.

4.1.3 MOMENTOS INVARIANTES DE HU

Este método tiene varias ventajas: extrae características de acuerdo a la forma del contorno de la imagen analizada y se pueden definir el número de características que describe una imagen. Además el tiempo de extracción de características es pequeño al igual que el tiempo de entrenamiento.

Este método es útil para extraer características de cualquier imagen con niveles de gris y, a diferencia de los otros métodos, extrae sólo siete parámetros que sirven como entrada para la red neuronal artificial. Asimismo, las características que extrae este método son indiferentes a la posición, tamaño y rotación, lo que hace que el reconocimiento tenga menos porcentaje de error, comparado con los otros métodos.

También se pudo concluir que los momentos de área son más estables que los momentos de perímetro o contorno al emplear imágenes irregulares.

Como desventaja se puede decir que aunque tienen porcentajes de aciertos muy altos pueden presentar ciertos errores cuando las figuras son rotadas.

4.1.4 FOURIER

Este método se beneficia de la posibilidad de compresión de datos que ofrece el dominio de la frecuencia. Lamentablemente los mejores resultados apuntaron hacia el modelo clásico de la visión artificial empleando momentos invariantes de Hu, porque se obtuvo en el modelo propuesto un porcentaje de aciertos muy bajo comparado con el de su contraparte.

El método de momentos invariantes de Hu trata de asimilar la descripción de una forma mediante una ecuación correspondiente a ella. Los descriptores de Hu no son más que las variables de esa ecuación correspondiente.

Analizando el aprendizaje de las redes entrenadas con fourier se observó que se logró aprender el conjunto de entrenamiento de manera exacta, pero se quedó débil al generalizar patrones no antes vistos por la red, esta conclusión es parte de una de las tesis analizadas.

Es lógico pensar que una red no logre generalizar adecuadamente si el número de elementos de entrada es muy alto con respecto al número de neuronas en la primera capa oculta. Esto es una gran inconveniente, porque al tratar de incrementar la arquitectura de la red el sistema de cómputo colapsa en la

etapa de entrenamiento, debido a limitaciones tanto de memoria como de software.

5 FLUJOGRAMA PARA EL MANEJO DE ALGORITMOS EN EL RECONOCIMIENTO DE PATRONES UTILIZANDO REDES NEURONALES

5.1 REDES NEURONALES

La topología de las redes en este ejercicio (capas y número de neuronas) siempre será del tipo perceptron multicapa, con un máximo de tres capas (dos ocultas y una de salida). Las dos primeras capas podrán tener hasta un máximo de 9999 neuronas y la capa de salida podrá tener hasta un máximo de 30 neuronas (teniendo en cuenta la topología de Matlab). Ver Anexo A.

En el momento de diseñar un programa que reconozca imágenes basado en redes neuronales se debe pensar en los siguientes requerimientos:

1. Un software que funcione como plataforma de procesamiento de imágenes.
2. Es aconsejable recurrir a implementar el programa en forma IDE (Entorno de Desarrollo Integrado).
3. Para el reconocimiento es necesario pensar en la técnica a usar. Si el entrenamiento de la red (visto como un proceso) ya fue realizado,

entonces se deben ejecutar las redes ya entrenadas.

4. Un lenguaje de programación que permita desarrollar aplicaciones basadas en Windows (o lo más cómodo comercialmente), un lenguaje compatible con el manejo de algunos formatos como JPEG y BMP.
5. Capacidad para manejar objetos OLE y controles Active X compatibles con Matlab (en el caso de usar matlab).

La obtención de imágenes puede ser lograda mediante una captura de videocámara o mediante una apertura de imagen almacenada en disco. Los procedimientos de Matlab que se pueden utilizar son los de procesamiento de imágenes y los de entrenamiento y simulación de redes neuronales. El código se utiliza así:

Variant V;

V=CreateOleObject("Matlab.Application");

Primero se declara una variable de tipo Variant donde se almacena el objeto OLE, luego se crea el objeto. Después de esto el programador puede en cualquier momento mandar a ejecutar cualquier instrucción en Matlab desde su programa en C++.

V.Exec(Procedure ("Execute")<<"instruccionMatlab");

Para la ejecución de una instrucción de Matlab se debe usar el código anterior, y reemplazar el campo “instruccionMatlab” por la instrucción o instrucciones que quiera ejecutar en Matlab. A continuación se ilustra el proceso de entrenamiento de la red para el reconocimiento.

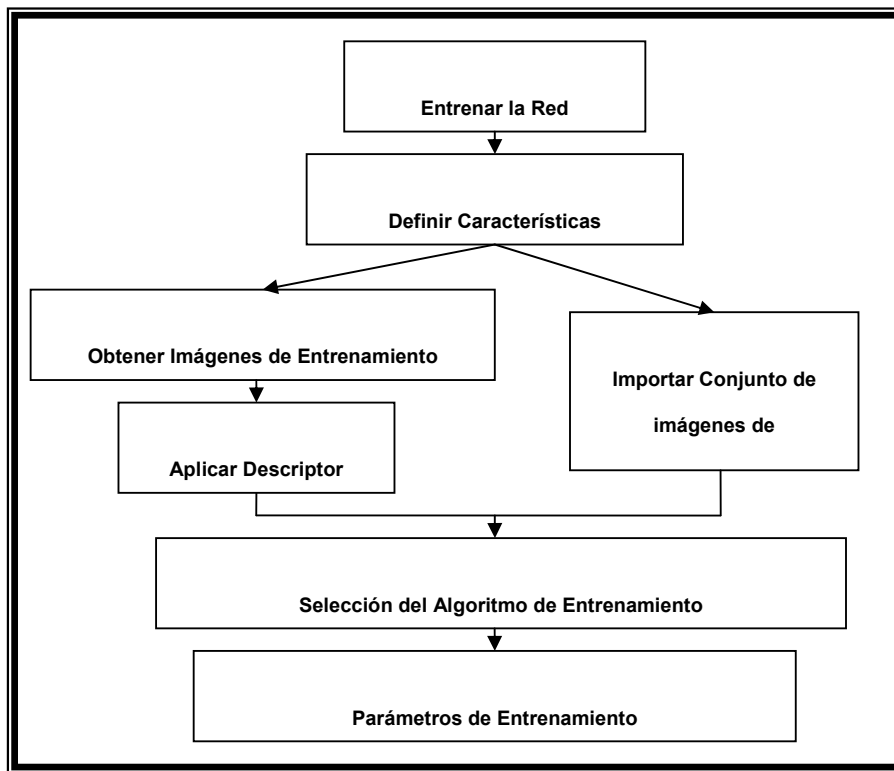


Ilustración XIX. PROCESO DE RECONOCIMIENTO BASADO EN REDES NEURONALES (ENTRENAMIENTO DE LA RED).

5.1.1 ENTRENAR LA RED

La red puede ser entrenada siguiendo estos pasos:

1. Seleccionar el algoritmo de entrenamiento.
2. Introducir los parámetros de entrenamiento: Iteraciones Máximas, Error de Convergencia y Razón de Actualización de la gráfica de entrenamiento.
3. Iniciar el proceso de entrenamiento: es posible monitorear este proceso de aprendizaje con la gráfica de error.
4. Finalmente se obtiene una matriz de pesos. Esta matriz se mantiene en memoria y puede ser observada y manipulada desde el *workspace* de Matlab.

Hasta este momento todos los pasos pueden ser reevaluados e ir al estado anterior sin que existan cambios en el proceso. Al momento de ejecutar el proceso de finalizar la nueva matriz de pesos será guardada como un proceso existente.

Después se podrá optar por alguna de las siguientes opciones:

1. Elegir el tipo de descriptor que se empleará para la extracción de características de las imágenes

Importar el conjunto de datos de entrenamiento de otro proceso existente (Si se

tiene ya un conjunto de entrenamiento previo).

2. Crear el propio conjunto de datos.

5.1.1.1 Crear conjunto de datos de entrenamiento

Para crear un conjunto de datos de entrenamiento se deben obtener las imágenes de entrenamiento ya sea con videocámara o con archivos de imágenes.

Cada una de estas imágenes deben ser pasadas a través del descriptor escogido (en nuestro caso momentos o DTC) para obtener una matriz o conjunto de datos de entrenamiento. Se escoge el algoritmo de entrenamiento y sus respectivos parámetros. Ver Anexo B.

Este proceso debe continuar con el entrenamiento de la red, y así se guarda la matriz de pesos de la red entrenada. Cuando se está creando un proceso nuevo es muy difícil retornar a los pasos anteriores. Por lo tanto, el programa debe contar con varios mecanismos de verificación y corrección en cada uno de los pasos.

5.1.2 EJECUTAR LA RED

La ejecución de una red neuronal integra varios procesos:

1. Obtención de imágenes.
2. Procesamiento de imágenes: esta etapa se realiza mediante un selecto conjunto de técnicas que pueden ser aplicadas en el orden que se desee.
3. Después del procesamiento de la imagen se le aplica el algoritmo de extracción de características (Descriptor), obteniéndose un vector de siete elementos. A esta técnica se le llama “momentos”. Otra técnica es DTC donde el vector obtenido es de aproximadamente 2000 elementos (dependiendo del tamaño de la imagen). Estos vectores describen de manera compacta la figura que aparece en la imagen.

El vector obtenido se utiliza como entrada de la red neuronal en el proceso de simulación de ésta.

La red neuronal se encarga de la identificación de la figura y los algoritmos de extracción de medidas se encargan de medir el área de la figura y la posición de su centro de masa. Ver Anexo B.

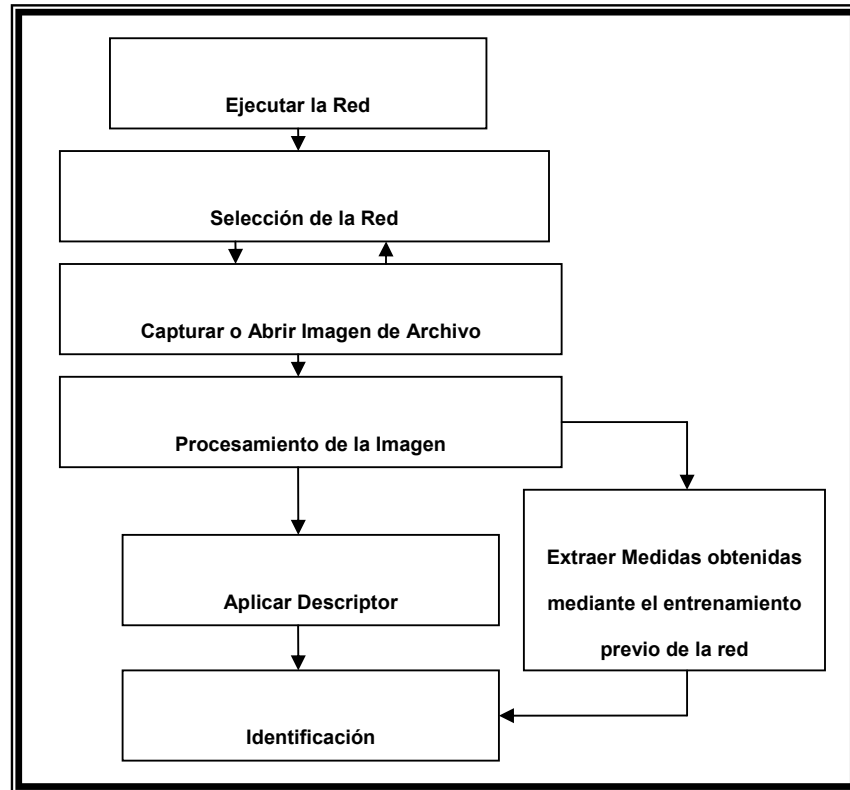


Ilustración XX. PROCESO DE RECONOCIMIENTO BASADO EN REDES NEURONALES (EJECUCIÓN DE LA RED).

5.1.2.1 Obtención del conjunto de imágenes de entrenamiento

Las imágenes se adquieren ya sean desde disco o directamente desde un dispositivo. Automáticamente son convertidas en imágenes de formato en escala de grises y también son redimensionadas a un tamaño (en píxeles) si la imagen original es más grande que el formato especificado. En este caso el programa deberá emplear alguna técnica para la reducción de la imagen (el comando disponible en la Matlab para realizar esta operación es: `imresize`),

aunque llevar las imágenes a un tamaño más pequeño deforma las imágenes tomadas y provoca traumatismos en el procesamiento. Lo mismo se debe hacer si es el caso contrario, o sea, si la imagen original es más pequeña que la de trabajo.

En Matlab la lectura de la imagen mediante el comando `imread` traslada la imagen de un archivo gráfico (.JPG) a una matriz de 240 x 320, que representa matemáticamente los valores de los píxeles de la figura. Ahora, teniendo una imagen RGB de 256 niveles de gris se convierte a una imagen binaria, es decir, que todas las partes de la imagen y del fondo de exposición que tengan oscuridades pasan a ser negro (0) y las claras a blanco (1). En este caso se está aplicando el concepto de umbralización mediante el comando `im2bw`. Este comando necesita un parámetro, a capricho del diseñador, que consiste en darle un nivel de intensidad (1 ó 0) a ciertos niveles de gris, es decir, con este parámetro se adquiere un cierto número de niveles de gris y se lleva a blanco y el resto a negro. El valor de este parámetro está dado en porcentaje.

Son varias las técnicas de procesamiento pero el uso de ellas varía de acuerdo al entorno de trabajo, a las características de visualización de los objetos a reconocer, a la superficie de fondo, a la cantidad de ruido presente, etc. Por ello en cada proyecto se deberá hacer una evaluación y/o ajuste a las técnicas a emplear. Algunas de esas técnicas son:

1. Ajuste de Histograma.
2. Detección de Bordes: este método consiste en una descripción del contorno de la figura mediante la valorización de los píxeles del contorno dependiendo de su dirección.

Estas imágenes adquiridas y procesadas conforman un conjunto de imágenes de entrenamiento. La idea fundamental de esta etapa es proporcionar a la red neuronal artificial la información necesaria para identificar una determinada figura.

Después de que el programa ha adquirido todo el conjunto de imágenes procesadas deberá codificarlas para obtener el verdadero conjunto de imágenes de entrenamiento. Esta codificación dependerá exclusivamente del descriptor elegido, ya sea Momentos o TDC.

5.1.2.1.1 Descriptor Momentos

Un proceso de entrenamiento puede contener varias redes con igual estructura topológica pero entrenadas con algoritmos distintos.

Es recomendable empezar con arquitecturas sencillas e ir variando las topologías incrementando el número de capas y/o el número de neuronas, para

luego comparar sus resultados y escoger la red neuronal con mejor desempeño.

Al emplear momentos el programa tomará cada una de las imágenes y extraerá sus características para describir a las imágenes con sólo siete valores, que son representados en Matlab como vectores columna, que luego se concatenan para conformar una matriz de entrenamiento.

En cada uno de los procesos se comienza entrenando la red mediante la técnica de validación con el propósito de obtener el error máximo de convergencia. Después de obtenido este error se vuelven a entrenar las redes con un error propuesto, un poco más bajo al obtenido anteriormente, todo esto con el fin de crear una red neuronal que logre aprenderse todos los patrones enseñados y que además sea capaz de generalizar patrones no antes vistos por ella. Ver Anexo C.

5.1.2.1.2 Descriptor TDC

Al emplear TDC el programa tomará cada una de las imágenes y les aplicará la transformada discreta del coseno. Esto creará una matriz de igual dimensión que la imagen original pero en el dominio de la frecuencia. Luego se eliminan

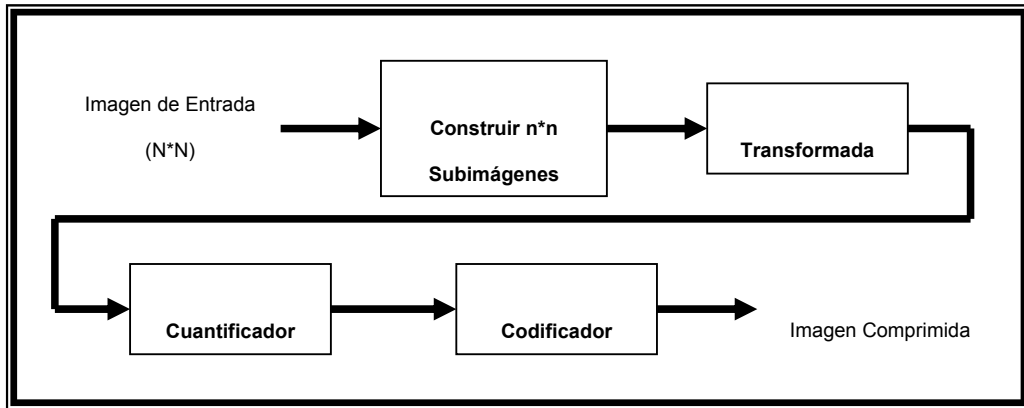
las altas frecuencias mediante la técnica de serpentín², reduciendo la imagen a un vector columna de 2016 elementos y así formar la matriz de entrenamiento.

La transformada TDC puede ser de tipo generalizada, es decir, que no sigue con el estándar de los bloques 8*8, sino que encuentra funciones bases con un solo bloque compuesto por todos los píxeles de la imagen.

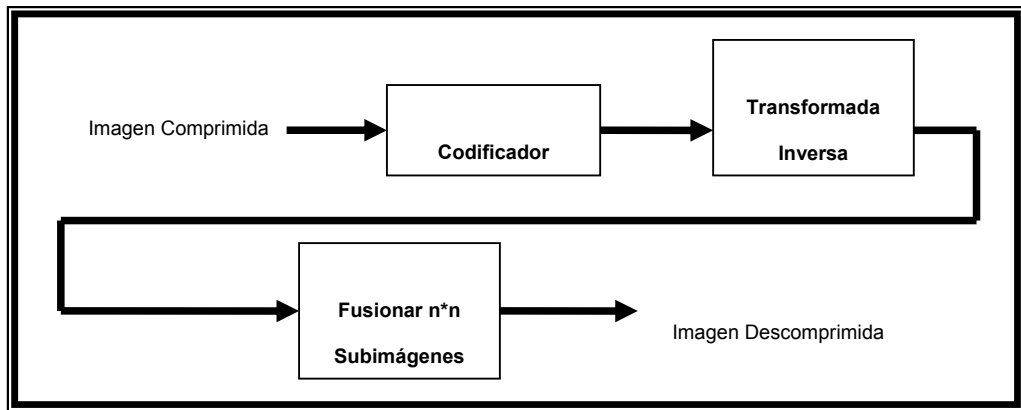
La entrada a estas redes está compuesta por aproximadamente 2000 elementos que dependen del tamaño de la imagen, provenientes originalmente de una matriz de $n * m$ elementos (tamaño de la imagen en píxeles).

PAJARES, Gonzalo y DE LA CRUZ, Jesús M., en su libro Visión Computacional, al considerar las técnicas que están basadas en la aplicación de una transformada a la imagen, señalan como objetivo el obtener a partir de la imagen una serie de coeficientes, que posteriormente son cuantificados y codificados. Las siguientes ilustraciones muestran un sistema típico de codificación de la transformada.

² Técnica inventada por Nikola Tesla (1893) para tratamientos de las altas frecuencias en electricidad.



**Ilustración XXI. SISTEMA DE CODIFICACIÓN DE LA TRANSFORMADA
(CODIFICADOR).**



**Ilustración XXII. SISTEMA DE CODIFICACIÓN DE LA TRANSFORMADA
(DECODIFICADOR).**

Se han desarrollado sistemas de compresión basados en diferentes transformadas a saber:

1. La Transformada Discreta de Fourier (TDF).

2. La Transformada Discreta del Coseno (TDC).
3. La Transformada de Walsh-Hadamard (TWH).
4. La Transformada de Karhunen-Loève (TKL).
5. La Transformada Discreta de Wavelets (TDW).

La elección de una transformada particular en una aplicación dada depende de la cantidad de error que se pueda tolerar en la reconstrucción y de los recursos computacionales disponibles. Un aspecto importante es el hecho de tener en cuenta que la compresión se logra durante la cuantificación de los coeficientes transformados y no durante la transformación misma.

En TDC las imágenes base son fijas (independientes de la entrada).

6 CONCLUSIONES

Al realizar el estudio de los sistemas automáticos de blancos se encontraron etapas comunes con el reconocimiento de patrones en visión artificial.

Las técnicas, aunque bien estructuradas a nivel teórico, presentan dificultades en su realización práctica. En este sentido será necesario realizar diferentes ajustes para cada aplicación específica. Esto es aplicable tanto a los sistemas automáticos como al reconocimiento de patrones. Conviene añadir que estas técnicas son ampliamente utilizadas en la literatura existente.

Las limitaciones encontradas a la hora de desarrollar la teoría de Visión Artificial y los Sistemas de Reconocimiento de Blancos, teniendo en cuenta experiencias de trabajos realizados alrededor de este tema, se basan en el tiempo que necesita el procesador para efectuar entrenamientos de imágenes tomadas desde varias ópticas, ya sean de tamaño o posición, para luego ser reconocidas o encontrar características de éstas en otras y así tener un resultado de identificación. Este proceso se vuelve difícil de realizar en tiempo real. Esto es bastante lógico puesto que hasta los humanos realizan reconocimiento tomándose un tiempo, alejándose del concepto de tiempo real. Sin embargo, se puede hacer el ejercicio de optimizar cada vez más los algoritmos para que esta aproximación sea posible. También existen problemas

en el manejo de la luz puesto que las características generadas por sombras pueden ser impedimento de un buen entrenamiento.

Se han descrito las ventajas que aporta la elección de un sistema de control visual basado en momentos invariantes, puesto que se controlan los grados de libertad correspondientes a movimientos de traslación teniendo como base un patrón a seguir. Nos referimos a momentos invariantes como las características a extraer de la imagen para realimentar el proceso de reconocimiento usando redes neuronales, y hablamos de las ventajas de elección de estas nuevas características respecto a la extracción de las coordenadas de los puntos característicos de la imagen. La elección de momentos invariantes aporta una mejor trayectoria dado que se consigue obtener una matriz de iteraciones constante. Aporta también mucha mayor flexibilidad a la hora de segmentar las características de la imagen.

La elección de momentos mejora el proceso de emparejamiento de características entre una y otra imagen, actuando a más alto nivel.

La Transformada Discreta de Fourier es utilizada para efectuar la compresión de las imágenes donde no se quiere tanta calidad de las imágenes. Dependiendo del tratamiento de la cuantificación y codificación pueden obtenerse resultados aceptables. Al analizar el aprendizaje de las redes entrenadas con Fourier se observa que este descriptor se queda corto al

generalizar patrones no antes vistos por la red, por lo que sigue siendo más efectivo el uso de momentos invariantes.

En la Industria no siempre se podrán usar patrones, y en muchas de estas ocasiones, probablemente tampoco se podrán extraer puntos característicos del objeto de forma sencilla. En estos casos será de gran utilidad emplear alguna técnica que permita realizar el control visual basándose en sus propias características y empleando sus momentos, o sea, realizar el seguimiento del objeto durante su movimiento. Este tipo de procesos pueden ser tema de otros estudios.

Desarrollar un sistema de reconocimiento de imágenes requiere ampliar los desarrollos teóricos matemáticos y profundizar en las herramientas computacionales complementadas de una extensa cantidad de pruebas experimentales.

7 BIBLIOGRAFÍA

CASAS V, Fernando. Memorias del Simposio de Visión Artificial y Tratamiento de Imágenes. Santafé de Bogotá: Uniandes, 1992.

DEMUTH, Howard; BEALE, Mark. Neural Network Toolbox, for use with Matlab, User's Guide: The MathWorks, 2001.

ESCALERA, A. de la. Visión por Computador. Madrid: Prentice Hall. 2001.

GONZALEZ GUERRERO, Enrique. Visión Robótica: Identificación y caracterización de Objetos. Santafé de Bogotá: Uniandes, 1999.

GONZALEZ, Rafael C y WOODS, Richard E. Tratamiento Digital de Imágenes. Barcelona: Adison – Wesley Iberoamericana. 2002.

K.S.F.U. rc.Gonzalez. Cs glee. Robótica, Control, Detección, Visión e Inteligencia. Madrid: Mc. Graw Hill. 1994.

NALWA, V.S. A Guided tour of Computer Vision. Addison Wesley. 1993.

PAJARES, Gonzalo y DE LA CRUZ, Jesús M. Visión por Computador: Imágenes digitales y Aplicaciones. RA-MA, 2001.

PAJARES, Gonzalo; DE LA CRUZ, Jesús M; MOLINA, José M; CUADRADO, J y LÓPEZ, A. Imágenes digitales: Procesamiento práctico con JAVA. RA-MA, 2003.

RESTREPO, Alfredo y NAVARRA, Tatiana. Memorias del Simposio de Tratamiento de Señales, Imágenes y Visión Artificial. Santafé de Bogotá: Uniandes, 1998.

RUSSEL, Stuart; NORVIG, Peter. Inteligencia Artificial un enfoque Moderno. Madrid: Prentice Hall Hispanoamericam. 1995.

WINSTON, Patrick Henry. Inteligencia Artificial. Barcelona: Adison – Wesley Iberoamericana. 1992.

REFERENCIAS BIBLIOGRAFICAS

[Acosta, 2004] Acosta, L., Tinoco, O. “Reconocimiento de Formas Irregulares Empleando Redes Neuronales Artificiales” Trabajo realizado para pregrado, Ingeniería Electrónica, Universidad Tecnológica de Bolívar UTB. Cartagena de Indias, 2004.

[Augustyn,1992] Augustyn, K “A new approach to automatic target recognition”, IEEE Transactions on Aerospace and Electronic System, Vol. 28, Jan 1992.

[Barrios, 2001] Barrios, A., Cancio, E. “Reconocimiento de Imágenes Geométricas planas utilizando Redes Neuronales Artificiales” Trabajo realizado para pregrado, Ingeniería Eléctrica y Electrónica, Universidad Tecnológica de Bolívar UTB. Cartagena de Indias, 2001.

[Bhanu,1986] Bhanu, B. “Automatic Target Recognition: State – of – the – Art Survey”, IEEE Transactions on Aerospace and Electronic System, July 1986.

[Bhanu,1993] Bhanu, B., Jones T. "Image Understanding Research for Automatic Target Recognition", DARPA Image Understanding Workshop, Jan 1992. A modified version in IEEE Aerospace and Electronic System, Oct 1993.

[Bruce,2001] Bruce, Nelson "Automatic Vehicle Detection in Infrared Imagery Using a Fuzzy Inference-Based Classification System", IEEE Transactions on fuzzy systems, Vol. 9, No.1 Feb. 2001.

[Campillo, 2003] Campillo, J., Gutiérrez de Piñeres, A. "Preprocesamiento de Señales, Guía de Procesamiento Digital de Imágenes" Trabajo realizado para pregrado, Ingeniería Electrónica, Universidad Tecnológica de Bolívar UTB. Cartagena de Indias, 2003

[Contreras, 2000] Contreras, S. "Visión Artificial", Trabajo realizado para Maestría en Potencia Eléctrica, UIS, 2000.

[Gader,2001] Gader, Paul D., Keller James M., "Recognition Technology for the Detection of Buried Land Mines", IEEE Transactions on fuzzy systems, Vol. 9, No. 1, Feb 2001.

[Howard,2000] Howard, A., Padgett C., Kenneth B., "Real time target detection and Analysis with Machine Vision", Third International Symposium on Intelligent Automation and control, World Automation Congress, June 2000.

[Kegelmeyer,1988] Kegelmeyer, W. P., Woods, K., Bowyer, K. W. "Combination of Multiple Classifiers Using Local Accuracy Estimates", IEEE Transaction on Pattern Analysis and Machine Intelligence, Apr 1997.

[Martínez, 2005] Martínez, D., Ruiz, F. "Interceptor de Trayectorias Basado en Visión Artificial", IEEE Colombian Workshop on Robotics and Automation, 2005.

[Sadjadi,1991] SADJADI, Firooz A., Signal and Image Processing Systems Performance Evaluation, Simulation, and Modeling. USA: SPIE Hatem N. Nasr and Michael E. Bazakos, 1991.

[Suetens,1992] Suetens, P., Fua, P., Hanson, A. "Computational Strategies for object recognition", ACM Computing Surveys, 1992.

[Theis,1989] Theis, T. J., Akerman, A. I. “Comparison of model based, vision, statistical based and neural net based ATRs”, IEEE Aerospace and Electronics Conference, 1989.

[Trivedi,1985] Trivedi, M. M. “Reprinted in Image Analysis Applications, Optical Engineering”, IEEE Transactions on Acoustics, Speech, and Signal Processing, 1990.

ANEXOS

ANEXO A

NEURAL NETWORK TOOLBOX (MATLAB)

Este toolbox proporciona funciones para el diseño, simulación y entrenamiento de los modelos neuronales de mayor uso en la actualidad:

1. Perceptron.
2. Redes Lineales.
3. Redes Backpropagation.
4. Redes de Base Radial.
5. Aprendizaje Asociativo y Competitivo.
6. Aplicaciones Autoorganizativas.
7. Aprendizaje de Cuantificación vectorial.
8. Redes de Elman.

9. Redes de Hopfield.

Mediante la inserción de un amplio número de funciones y procedimientos escritos para Matlab el usuario puede, mediante el Neural Network Toolbox, efectuar el diseño de arquitecturas complejas, combinando los modelos que ya están proporcionados por defecto en el Toolbox.

Asimismo, el usuario puede definir sus propias funciones de transferencia, reglas de aprendizaje, funciones de entrenamiento y estimación de error para usarlas posteriormente con las funciones básicas.

El Toolbox aporta facilidades como gráficas para el análisis del comportamiento de las redes: visualización gráfica de la matriz de pesos y vector de desplazamiento mediante diagramas de Hinton, representación de errores a lo largo del entrenamiento, mapas de superficie de error en función de pesos y vector de desplazamiento, etc. Estos gráficos resultan muy útiles en el estudio de la convergencia y estabilidad de los algoritmos de aprendizaje.

Este Toolbox incluye un manual de introducción al campo de las Redes Neuronales junto con una colección de demostraciones y aplicaciones muy didácticas útiles para el estudio y la profundización de los temas fundamentales de los paradigmas de redes neuronales básicos.

A pesar de que el estudio de las redes neuronales se inició ya hace algunas décadas las primeras aplicaciones sólidas dentro de este campo no han tenido lugar hasta hace unos años y aún ahora constituyen un área de investigación en rápido desarrollo.

Este Toolbox tiene por tanto una orientación diferente a aquellos destinados a campos como el de sistemas de control u optimización donde la terminología, fundamentos matemáticos y procedimientos de diseño, están firmemente establecidos y se han aplicado durante años.

Esta herramienta tiene el soporte de MATLAB y SIMULINK. La librería de SIMULINK contiene modelos de capas de redes neuronales de cada tipo de neurona implementada en el toolbox de redes neuronales. Es posible por tanto diseñar sistemas SIMULINK para simular redes neuronales creadas usando esta herramienta. Simplemente las capas se conectan de acuerdo con la arquitectura de la red y se proporcionan como entrada a la caja de diálogo de cada capa, la matriz de pesos apropiada y el vector de desplazamiento.

Usando el generador de código C de SIMULINK es posible generar automáticamente el código correspondiente a un diseño neuronal.

FUNCIONES MATLAB UTILIZADAS

PARA RED TIPO PERCEPTRON

Las siguientes son las herramientas de redes neuronales de Matlab (5.3 – 6.5) utilizadas en el entrenamiento de las mismas para el proceso de detección de objetos.

NEWP

Crea una red tipo perceptron que requiere las siguientes entradas:

Sintaxis:

NET = NEW(PR, S, TF, LF)

PR R * 2 matriz de valores máximos y mínimos para los R elementos de entrada.

S Número de Neuronas.

TF Función de Transferencia.

LF Función de Aprendizaje.

RANDS

Función simétrica que inicializa aleatoriamente los valores de los pesos y ganancias de una red con valores entre -1 y 1.

Sintaxis:

RANDS(S; PR)

Genera una matriz de dimensiones S * PR.

ADAPT

Esta función permite a una red neuronal artificial adaptarse a los patrones de entrada.

Sintaxis:

[NET, Y, E, Pf, Af]=ADAPT(NET, P, T, Pi, Ai)

NET Red que va a crearse.

P Entradas a la red: deben aparecer en forma de un arreglo de matrices.

T Salidas esperadas de la Red; si no se especifica son ceros por defecto.

Pi Condiciones de retardo para la entrada inicial; por defecto son ceros.

Ai Condiciones de retardo para la capa inicial; por defecto son ceros.

Net.adaptParam.passes Número de iteraciones que utiliza el programa.

PARA RED DE TIPO ADALINE

Las siguientes son las funciones de las herramientas de redes neuronales de Matlab utilizadas en el entrenamiento de filtros adaptativos diseñados con base en una red Adaline.

NEWLIN

Esta función sirve para crear una red del tipo Adaline, que requiere las siguientes entradas.

Sintaxis:

NEWLIN(PR, S, ID, LR)

PR Matriz $R * 2$ que contiene los valores máximos y mínimos de cada uno de los R elementos de entrada.

S Número de Neuronas.

ID Arreglo que contiene los valores; por defecto todos sus valores son ceros.

LR Ratio de aprendizaje; por defecto LR = 0.01.

Net.inputWeights{1,1}.delays Especifica los retardos iniciales.

Net.adaptParam.passes Especifica el número máximo de iteraciones.

ENTRENAMIENTO

El comando de entrenamiento de la red requiere como entradas la red creada anteriormente, los patrones de salida y los retardos iniciales.

Retorna el estado final de la red, los valores obtenidos para cada patrón de entrada con sus correspondientes errores así como valores finales de los retardos.

Sintaxis:

[NET, Y, E, Pf, Af]=ADAPT(NET, P, T, Pi)

RED TIPO BACKPROPAGATION

La red neuronal backpropagation emplea una gran variedad de opciones de configuración, dependiendo de las necesidades de aprendizaje y de la

aplicación que se esté desarrollando.

NEWFF

Crea una red tipo backpropagation.

Sintaxis:

NEWFF:(PR, [S1 S2 ... SIN], {TF1 TF2 ... TFNI}. BTF, BLF, PF

PR $R * 2$ matriz de valores máximos y mínimos de cada una de las R neuronas de entrada.

Si Número de neuronas para cada una de las capas.

TFi Función de transferencia a utilizar en cada una de las capas; por defecto utiliza tansig.

BTF Algoritmo de entrenamiento a utilizar; por defecto utiliza trainim.

BLF Función de actualización de los pesos; por defecto utiliza learngdm.

PF Función para evaluar el desempeño de la red; por defecto utiliza mse.

Los siguientes son los algoritmos de entrenamiento más utilizados en redes

backpropagation.

Traingd

Algoritmo de pasos descendentes que actualiza pesos y ganancias variándolos en la dirección negativa del gradiente de la función de error. Es un algoritmo muy lento.

Net.trainParam.goal Error máximo permitido.

Net.trainParam.lr Ratio de aprendizaje.

Net.trainParam.max_fail Máximo número de fallas.

Net.trainParam.min_grad Mínimo rendimiento del gradiente.

Net.trainParam.show Intervalo de visualización de los resultados.

Net.trainParam.time Máximo tiempo de entrenamiento en segundos.

Con este algoritmo el aprendizaje de la red se detendrá si el número de iteraciones excede del comando `net.trainParam.epochs`, si se alcanzó el valor del error propuesto como meta, si la magnitud del gradiente es menor que `net.trainParam.min_grad` o si el tiempo de entrenamiento supera el valor de

net.trainParam.time.

Traingdm

Es el algoritmo tradicional más un nuevo coeficiente de momentum, que interviene en el proceso de actualización de los pesos. Si el error de la red en una iteración dada excede el valor del error en la iteración anterior, en un valor mayor al definido por un radio de cobertura dado al que puede determinarse por medio de la función max_perf_inc y que esta típicamente alrededor de 1.04, los nuevos pesos y ganancias son descartados y el coeficiente de momentum mc es fijado en cero.

La sintaxis de este algoritmo es igual a la utilizada para el algoritmo traingd, más un nuevo elemento que permite modificar el coeficiente de momentum.

Sintaxis:

Net.trainParam.mc Valor fijado para el coeficiente de momentum.

Traingda

Algoritmo de gradiente descendiente que emplea una tasa de aprendizaje adaptativa durante el proceso de entrenamiento. El ratio de aprendizaje varía entre 0.01 y 1, un ratio de aprendizaje muy pequeño torna lento el aprendizaje,

pero si se incrementa demasiado el aprendizaje puede tornarse inestable y crear divergencia. Por esto la función traingda varía el ratio de aprendizaje tratando de sacar provecho de la inclinación del gradiente en cada momento. Su gran desventaja es que los pesos iniciales varían muy poco así se encuentren distantes de los valores de convergencia.

Sintaxis:

Net.trainParam.epochs Máximo número de iteraciones para obtener convergencia.

Net.trainParam.goal Error máximo permitido.

Net.trainParam.lr Ratio de aprendizaje inicial.

Net.trainParam.lr_inc Porcentaje que incrementa el ratio de aprendizaje cuando el error disminuye.

Net.trainParam.lr_dec Porcentaje por el que es decrementada el ratio de aprendizaje cuando el error aumenta.

Net.trainParam.max_fail Máximo número de fallas.

Net.trainParam.max_perf_inc Máximo incremento del rendimiento.

Net.trainParam.min_grad Mínimo rendimiento del gradiente.

Net.trainParam.show Los resultados son visualizados siempre que transcurre este número de iteraciones.

Net.trainParam.time Máximo tiempo de entrenamiento n segundos.

Trainrp

Las redes multicapa utilizan típicamente una función de transferencia sigmoideal en las capas ocultas. Estas funciones comprimen un infinito rango de entradas dentro de un finito rango de salidas. Además se caracterizan porque su pendiente tenderá cada vez más a cero, mientras más grande sea la entrada que se le presenta a la red; esto ocasiona problemas cuando se usa un algoritmo de entrenamiento de pasos descendientes porque el gradiente empieza a tomar valores muy pequeños y por lo tanto no habrán cambios representativos en los pesos y las ganancias, así se encuentren bastante lejos de sus valores óptimos.

El propósito del algoritmo backpropagation resileint (RPROP) es eliminar este efecto en la magnitud de las derivadas parciales. En este algoritmo solamente el signo de la derivada es utilizado para determinar la dirección de actualización de los parámetros. La magnitud de las derivadas no tienen efecto en la

actualización.

La magnitud en el cambio de cada peso es determinada por separado; el valor del incremento de pesos y ganancias es determinado por el factor `delt_inc`, así la derivada parcial del error con respecto a los pesos tenga el mismo signo durante dos iteraciones sucesivas; el valor de decremento está determinado por el factor `delt_dec` así la derivada del error con respecto a los pesos haya cambiado de signo con respecto a la anterior iteración; si la derivada es cero entonces el valor actualizado se conserva; si los pesos continúan cambiando en la misma dirección durante varias iteraciones, la magnitud de cambios de los pesos se decrementa.

Sintaxis:

Net.trainParam.epochs Máximo número de iteraciones del entrenamiento.

Net.trainParam.show Intervalo de visualización de los resultados.

Net.trainParam.goal Error deseado.

Net.trainParam.time = inf Máximo tiempo de entrenamiento en segundos.

Net.trainParam.min_grad Mínimo rendimiento del gradiente.

Net.trainParam.max_fail Máximo número de fallas.

Net.trainParam.lr Ratio de aprendizaje.

Net.trainParam.delt_inc Incremento de la actualización de pesos.

Net.trainParam.delt_dec Decremento de la actualización de pesos.

Net.trainParam.delta0 Incremento inicial en la actualización de pesos.

Net.trainParam.deltamax Máximo cambio en los pesos.

Trainbfg

Algoritmo alternativo que emplea la técnica del gradiente conjugado. Su expresión matemática se deriva del método de Newton, con la ventaja de que no es necesario computar las segundas derivadas. Este algoritmo requiere más capacidad de almacenamiento que el algoritmo tradicional, pero generalmente converge en menos iteraciones.

Requiere de un cálculo aproximado de la matriz Hessiana, la cual es de dimensiones $n^2 * n^2$, donde n es la cantidad de pesos y ganancias de la red;

para redes que involucren una gran cantidad de parámetros es preferible emplear el algoritmo trainrp.

Sintaxis:

Net.trainParam.epochs Máximo número de iteraciones del entrenamiento.

Net.trainParam.show Número de iteraciones entre las cuales se muestran resultados.

Net.trainParam.goal Error deseado.

Net.trainParam.time = inf Máximo tiempo de entrenamiento en segundos.

Net.trainParam.min_grad Mínimo rendimiento del gradiente.

Net.trainParam.max_fail = 5 Máximo número de fallas.

Net.trainParam.searchFcn'srchcha' Nombre de la rutina de búsqueda lineal a utilizar.

Net.trainParam.scal_tol Se divide entre el valor de Delta para

determinar la tolerancia para la búsqueda lineal.

Net.trainParam.alpha Factor de escala que determina una reducción suficiente en el desempeño.

Net.trainParam.beta Factor de escala que determina un tamaño de paso suficientemente grande.

Net.trainParam.delta Tamaño de paso inicial en el intervalo de localización de paso.

Net.trainParam.gama Parámetro para evitar pequeñas reducciones en el desempeño.

Net.trainParam.low_lim Límite inferior en el cambio del tamaño del paso.

Net.trainParam.up_lim Límite superior en el cambio del tamaño del paso.

Net.trainParam.maxstep Máxima longitud de paso.

Net.trainParam.minstep Mínima longitud de paso; por defecto es $1.0e^{-6}$.

Net.trainParam.bmax Máximo tamaño de paso.

Trainlm

Algoritmo que actualiza los pesos y las ganancias de acuerdo a la optimización de Levenberg – Marquardt. Es el algoritmo más rápido para redes backpropagation. Tiene la desventaja de requerir de un conjunto de datos de entrenamiento lo más estándar posible, pues de otra forma solo aproximará correctamente valores que se encuentren dentro de los patrones de aprendizaje.

Si el conjunto de entrenamiento es muy extenso se recomienda reducir el jacobiano.

Sintaxis:

Net.trainParam.epochs Máximo número de iteraciones del entrenamiento.

Net.trainParam.goal Error deseado.

Net.trainParam.lr Ratio de aprendizaje.

Net.trainParam.max_fail Máximo número de veces que falla el valor μ .

Net.trainParam.mem_reduc Factor de fraccionamiento de jacobiano para ahorrar memoria.

Net.trainParam.min_grad Mínimo rendimiento del gradiente.

Net.trainParam.show Intervalo de visualización de los resultados.

Net.trainParam.time Máximo tiempo de entrenamiento en segundos.

tr.mu Valor del μ adoptivo.

ANEXO B

RUTINAS DE ENTRENAMIENTO DE REDES

A continuación se detalla el bloque de código más importante empleado en las técnicas de entrenamiento de redes neuronales.

Los parámetros de aprendizaje son valores que deben ser ingresados por el usuario del software, a excepción del ratio de aprendizaje del algoritmo de entrenamiento "Gradiente descendiente" por que éste ya viene incluido en el código:

```

if (RadioGroup1->ItemIndex==0)
V.Exec(Procedure("Execute")<<("net=newff(minmax("+Desc+"), [{"+c1+"
"+c2+" "+c3+"}], {"'+Ft1+' "''+Ft2+'"' 'Logsig'}, 'trainlm');"));
if (RadioGroup1->ItemIndex==1)
V.Exec(Procedure("Execute")<<("net=newff(minmax("+Desc+"), [{"+c1+"
"+c2+" "+c3+"}], {"'+Ft1+' "''+Ft2+'"' 'Logsig'}, 'trainrp');"));
if (RadioGroup1->ItemIndex==2)
V.Exec(Procedure("Execute")<<("net=newff(minmax("+Desc+"), [{"+c1+"
"+c2+" "+c3+"}], {"'+Ft1+' "''+Ft2+'"' 'Logsig'}, 'traingda');"));
V.Exec(Procedure("Execute")<<"net.trainParam.epochs="+Edit1->Text+ " ",
net.trainParam.goal="+Edit2->Text+", net.trainParam.show="+Edit3-
>Text+";");
If (RadioGroup1->ItemIndex==2)
V.Exec(Procedure("Execute")<<"net.trainParam.lr=0.28,
net.trainParam.lr_inc=2.05");
if (!CheckBox1->Checked)
{
    if (Form1->R_Archivo(DirectoryListBox1->Directory,8) == "Momentos")

```

```
V.Exec(Procedure("Execute")<<"[net,tr]=train(net,momentos,targets);");  
else  
V.Exec(Procedure("Execute")<<"[net,tr]=train(net,dct,targets);");  
}
```

El código comienza con la creación de una red neuronal feed-forward backpropagation. La instrucción enviada a Matlab es:

```
net=newff(PR,[s1 s2 ... sin], {TF1 TF2 ... TFNI}, BTF, BLF, PF)
```

Donde los parámetros de esa función tienen los siguientes significados:

PR: Matriz $R \times 2$ de valores mínimo y máximo para los R elementos de entrada.

Si: Tamaño de la i -ésima capa para NI capas.

TFi: Función de transferencia de la i -ésima capa. Esta puede ser cualquier función diferenciable tal como tansig, logsig o purelin; por defecto tansig.

BTF: Función de entrenamiento de la red backpropagation. Esta puede ser trainml, trainbfg, trainrp, traingd, etc. Por defecto traingdx.

BLF: Función de aprendizaje de pesos y ganancias de la red backpropagation; por defecto learngdm.

PF: Función de ejecución; por defecto mse.

El resultado de esta ejecución es una red feed-forward backpropagation de N capas.

Después de la creación de la red neuronal backpropagation se configuran los parámetros de entrenamiento.

net.trainParam.epoca = valor1

net.trainParam.goal = valor2

net.trainParam.show = valor3

Valor1: Número máximo de iteraciones en el proceso de entrenamiento.

Valor2: Error propuesto para el entrenamiento.

Valor3: Tasa de actualización de la gráfica de entrenamiento.

La instrucción que da inicio al proceso de entrenamiento:

[net,tr]=train(NET, P, T, Pi, Ai)

NET: Red Neuronal.

P: Matriz de Entradas.

T: Matriz de Salidas (targets); por defecto ceros.

Pi: Condiciones iniciales en la entrada de retardo; por defecto ceros.

Ai: Condiciones iniciales en la capa de retardo; por defecto ceros.

El resultado:

net Nueva Red.

tr Historial de entrenamiento (iteraciones y ejecución).

RUTINA DE EJECUCIÓN DE REDES

En el momento de poner en ejecución una red neuronal previamente entrenada se debe acudir a un procedimiento de simulación de redes, el cual se implementa con la siguiente función de Matlab:

[Y, Pf, Af] = sim (net, P, Pi, Ai)

Los parámetros son:

Y: Vector de Salida.

Pf: Condición final de la entrada de retardo.

Af: Condición final de la capa de retardo.

net: Red.

P: Vector de Entrada.

Pi: Condición inicial en la entrada de retardo; por defecto es cero.

Ai: Condición inicial en la capa de retardo; por defecto es cero.

ANEXO C

ARCHIVOS COMPLEMENTARIOS, MOMENTOS Y TRANSFORMADA DISCRETA DEL COSENO

Momentos.m

```

K = m;
J = n; %Dimensiones de la Imagen
img=im2double(img);
%=====
% Centro de Gravedad
%=====
%Coordenada X del Centro de Gravedad
A=sum(sum(img)); %Área de la Imagen
alfa = 1;
beta = 0;
Xk = [1:1:K].^alfa;
Yj = [1:1:J].^beta;
Yj = Yj.';
X0 = (sum(sum((Yj*Xk).*img)))/A;
%=====
% Momentos Invariantes a la Traslación y a la Escala
%=====
alfa = 0;
beta = 2;
Xk = ([1:1:K]-repmat(X0,1,K)).^alfa;
Yj = ([1:1:J].'-repmat(Y0,J,1)).^beta;
M02 = sum(sum((Yj*Xk).*img));
U02 = M02/(A^((alfa+beta)/2+1));
%=====

```

```

alfa = 2;
beta = 0;
Xk = ([1:1:K]-repmat(X0,1,K)).^alfa;
Yj = ([1:1:J].'-repmat(Y0,J,1)).^beta;
M20 = sum(sum((Yj*Xk).*img));
U20 = M20/(A^((alfa+beta)/2+1));
%=====

alfa = 1;
beta = 1;
Xk = ([1:1:K]-repmat(X0,1,K)).^alfa;
Yj = ([1:1:J].'-repmat(Y0,J,1)).^beta;
M11 = sum(sum((Yj*Xk).*img));
U11 = M11/(A^((alfa+beta)/2+1));
%=====

alfa = 0;
beta = 3;
Xk = ([1:1:K]-repmat(X0,1,K)).^alfa;
Yj = ([1:1:J].'-repmat(Y0,J,1)).^beta;
M03 = sum(sum((Yj*Xk).*img));
U03 = M03/(A^((alfa+beta)/2+1));
%=====

alfa = 3;
beta = 0;
Xk = ([1:1:K]-repmat(X0,1,K)).^alfa;
Yj = ([1:1:J].'-repmat(Y0,J,1)).^beta;
M30 = sum(sum((Yj*Xk).*img));
U30 = M30/(A^((alfa+beta)/2+1));
%=====

alfa = 1;
beta = 2;
Xk = ([1:1:K]-repmat(X0,1,K)).^alfa;

```

```

Yj = ([1:1:J].'-repmat(Y0,J,1)).^beta;
M12 = sum(sum((Yj*Xk).*img));
U12 = M12/(A^((alfa+beta)/2+1));
%=====

alfa = 2;
beta = 1;
Xk = ([1:1:K]-repmat(X0,1,K)).^alfa;
Yj = ([1:1:J].'-repmat(Y0,J,1)).^beta;
M21 = sum(sum((Yj*Xk).*img));
U21 = M21/(A^((alfa+beta)/2+1));
%=====

% Momentos Invariantes a la Rotación
%=====

k1 = 1;
k2 = 1;
k3 = 1;
k4 = 1;
k5 = 1;
k6 = 1;
k7 = 1;
%=====

h1 = U20 + U02;
h2 = (U20 - U02) ^ 2 + 4 * (U11) ^ 2;
h3 = (U30 - 3 * U12) ^ 2 + (U03 - 3 * U21) ^ 2;
h4 = (U30 + U12) ^ 2 + (U03 - U21) ^ 2;
h5 = (U30 - 3 * U12) * (U30 + U12) * ((U30 + U12) ^ 2 - 3 * (U03 + U21) ^ 2) +
(3 * U21 - U03) * (U03 + U21) * (3 * (U30 + U12) ^ 2 - (U21 + U03) ^ 2);
h6 = (U20 - U02) * ((U30 + U12) ^ 2 - (U03 + U21) ^ 2) + 4 * U11 * (U30 + U12)
* (U03 + U21);
h7 = (3 * U21 - U03) * (U30 + U12) * ((U30 + U12) ^ 2 - 3 * (U03 + U21) ^ 2) +
(3 * U12 - U30) * (U21 + U03) * (3 * (U30 + U12) ^ 2 - (U21 + U03) ^ 2);
%=====

```

```
% Momentos
%=====

m = [h1;h2;h3;h4;h5;h6;h7];
```

ValidarMomentos.m

```
[fil,col] = size(momentos);
iitr = [1:4:col 3:4:col 4:4:col];
iiva = 2:4:col;
v.P = momentos(:,iiva);
v.T = targets(:,iiva);
mtrain = momentos(:,iitr);
ttrain = targets(:,iitr);
[net,tr] = train(net,mtrain,ttrain,[],[],v);
Fig = figure(1)
plot(tr.epoch, tr.perf, tr.epoch, tr.vperf)
legend('Entrenamiento', 'Validacion', -1);

ylabel('Error Cuadratico'); xlabel('Iteraciones')
```

Targets.m

```
I = find(result > lim);
[fil,col] = size(I);
if fil == 1;
    save 'C:\\...'
else
    I = 0;
    save 'C:\\...'

end
```

Dct.m

```
img = im2double(img);
img = imresize(img, [m n]); %[m n] tamaño de la imagen por ejemplo [144 176]
dct0 = dct2(img);
tam = 2000;
x = 1;
y = 1;
pos = 1;
matriz = dct0;
vec(pos) = matriz(x,y);
while pos < tam
    x = x + 1;
    pos = pos + 1;
    vec(pos) = matriz(x,y);
    while x > 1
        x = x - 1;
        y = y + 1;
        pos = pos + 1;
        vec(pos) = matriz(x,y);
    end
    y = y + 1;
    pos = pos + 1;
    vec(pos) = matriz(x,y);
    while y > 1
        y = y - 1;
        x = x + 1;
        pos = pos + 1;
        vec(pos) = matriz(x,y);
    end
end
end
```

```
d = vec';
```

ValidarDct.m

```
[fil,col] = size(dct);  
iitr = [1:4:col 3:4:col 4:4:col];  
iiva = 2:4:col;  
v.P = dct(:,iiva);  
v.T = targets(:,iiva);  
mtrain = dct(:,iitr);  
ttrain = targets(:,iitr);  
[net,tr] = train(net,mtrain,ttrain,[],[],v);  
Fig = figure(1)  
Plot(tr.epoch,tr.perf,tr.epoch,tr.vperf)  
legend('Entrenamiento','Validacion',-1);  
ylabel('Error Cuadratico');  
xlabel('Iteraciones');
```

ANEXO D

PSD

PRUEBA ESTADISTICA DE DESCRIPTORES VERSION 1.0 MANUAL DEL USUARIO

REQUERIMIENTOS DEL SISTEMA

Para el buen funcionamiento de la aplicación se requiere lo siguiente:

1. Sistema Operativo: W2k
2. Software instalado: Maquina virtual de Java (jdk vr. 1.5)
3. Hardware: Memoria RAM mayor a 256 Mb

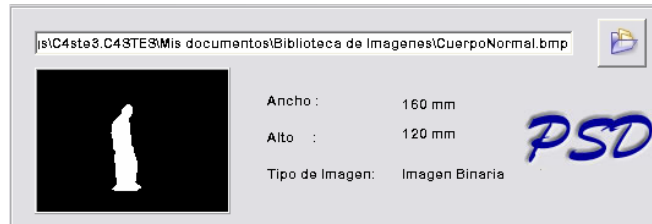
La ventana principal del sistema consta simplemente de un botón para la carga de imágenes. Además de esto, cuenta con una serie de pestañas que contienen los métodos principales de los descriptores aquí usados y sus características.



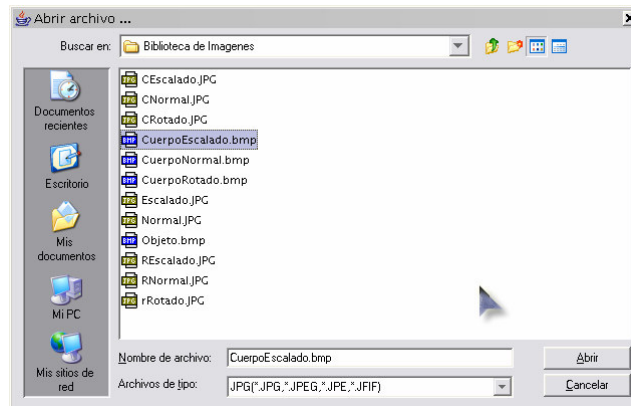
Carga de Archivos

La función principal de esta parte del sistema es solamente la carga de una imagen cualquiera para el análisis detallado de esta por los descriptores. Al ser cargadas muestra una visualización de la imagen y las características de dimensión de estas y su formato.

Para esto solamente se tiene que dar clic sobre el icono de carpeta que permitirá explorar las carpetas de su disco para escoger los archivos requeridos (estas imágenes deben ser binarias).



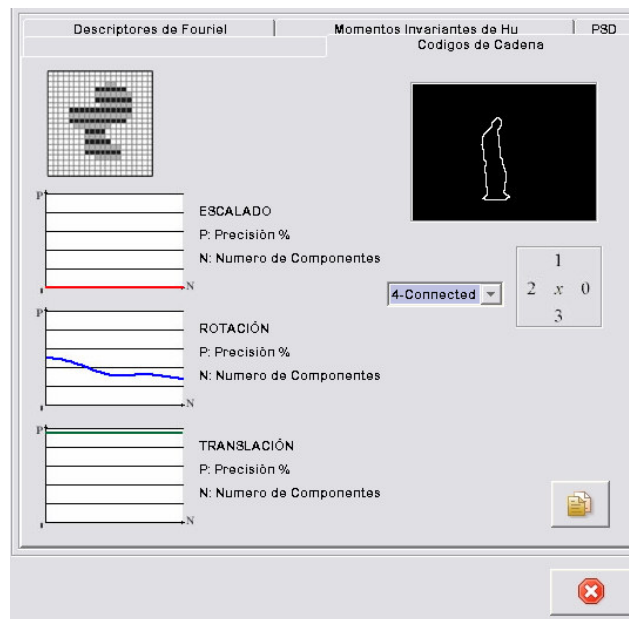
Las imágenes que se carguen pueden ser de extensión (.JPG, .GIF, .BMP, .PNG)



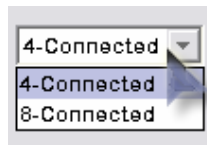
Pestaña del descriptor de cadena

Como se sabe el método de descripción de cadena consta de simplemente seguir el contorno de una imagen mediante dos métodos. El primero consta de seguir cuatro direcciones cardinales Norte, Sur, Este y Oeste. El segundo método consta de seguir las ocho posibles direcciones cardinales Norte, Noreste, Este, Sureste, Sur, Suroeste, Oeste y Noroeste.

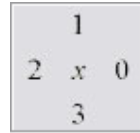
El sistema también posee las estadísticas de precisión de este método para los casos de escalado de imágenes, rotación de imágenes y translación de imágenes.



Para el manejo de estos 2 métodos existe una lista donde se es posible escoger el método deseado por el usuario y analizar sus características, como se muestra a continuación:

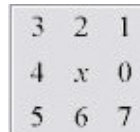


Manejo de 4 Direcciones



Cuando esta escogido este método en la lista aparece el siguiente icono en la ventana, que representa que el sistema esta manejando el código de cadena de 4 direcciones y es posible verlo en la visualización de la imagen del descriptor.

Manejo de 8 Direcciones



Cuando esta escogido este método en la lista aparece el siguiente icono en la ventana, que representa que el sistema esta manejando el código de cadena de 8 direcciones y es posible verlo en la visualización de la imagen del descriptor.

Registro en Logs

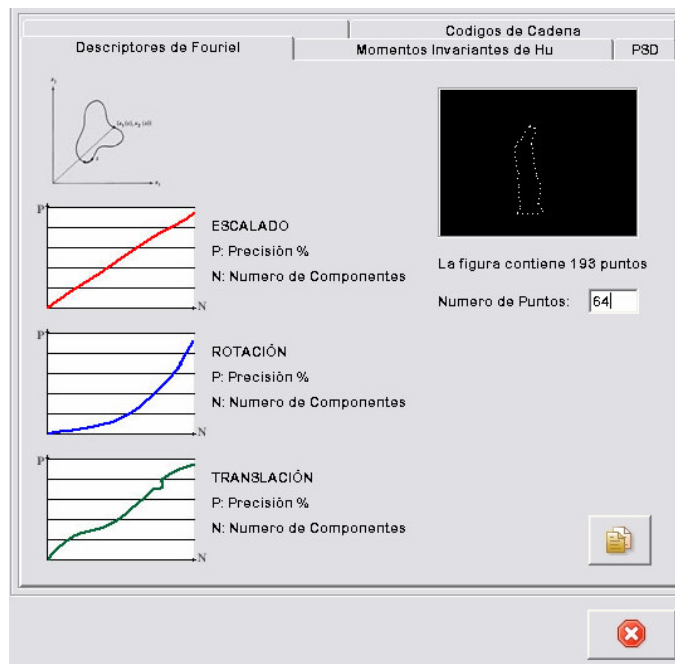


El botón de registro en Log representa la creación de un archivo físico en el disco (Carpeta de Archivos), donde el coloca el código de cadena fuente sin modificación, creado al recorrer la imagen binaria.

Pestaña del descriptor de Fourier

El método de Fourier trata de asemejar una imagen dependiendo de los puntos que se le den que pertenezcan a la forma original de la imagen, de esta forma a mayor cantidad de puntos, mayor exactitud se tendrá en el descriptor, pero mayor número de descriptores se generaran haciendo engorrosa la labor.

El sistema también posee las estadísticas de precisión de este método para los casos de escalado de imágenes, rotación de imágenes y translación de imágenes.



Aumentando o disminuyendo puntos, mediante esta opción es posible

modificar la cantidad de puntos que intervienen en la creación de los descriptores de fourier correspondientes a la forma de la imagen.

Como se muestra a continuación ese número de puntos es modificable hasta el límite máximo de puntos que conforman la imagen. Recuerde que la exactitud del método depende en la mayor cantidad de puntos que se tenga.



Registro en Logs



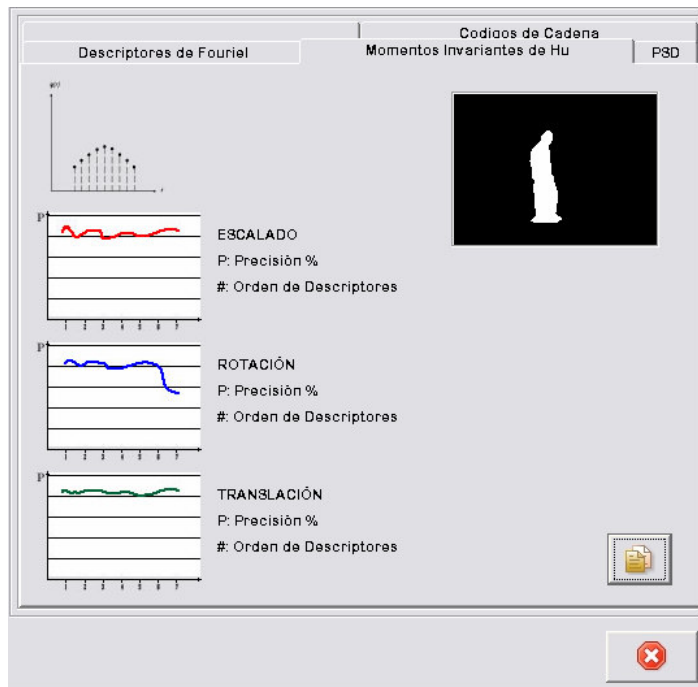
El botón de registro en Log representa la creación de un archivo físico en el disco (Carpeta de Archivos) donde él, coloca los descriptores de Fourier creados al asimilar una cantidad de puntos a la forma original.

Pestaña del descriptor de Hu

El método de momentos invariantes de Hu trata de asimilar la descripción de una forma mediante una ecuación correspondiente a ella. Los descriptores de

Hu no son más que las variables de esa ecuación correspondiente.

El sistema también posee las estadísticas de precisión de este método para los casos de escalado de imágenes, rotación de imágenes y translación de imágenes.



Registro en Logs

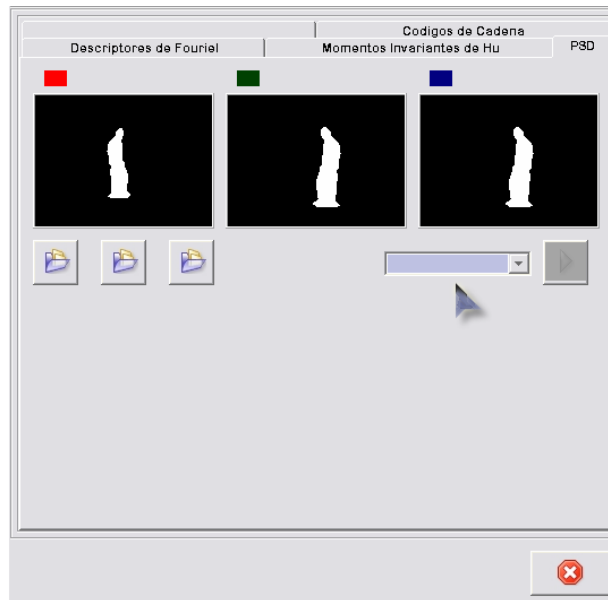
El botón de registro en Log representa la creación de un archivo físico en el disco (Carpeta de Archivos), donde el coloca el código de cadena fuente sin modificación, creado al recorrer la imagen binaria.

PSD

PSD (Prueba Estadística de Descriptores) no es más que un software para el análisis de resultados de los métodos aplicados a imágenes de misma forma pero sometidas a condiciones especiales tales como la rotación, el escalado y el traslado de la misma, para que de manera estadística se pueda analizar que tan exacto o que tan preciso son los métodos bajo estas condiciones que de una u otra forma se someten las formas en la vida real. Recuerde que a mayor exactitud se deben tener los mismos patrones de en la tabla estadística.

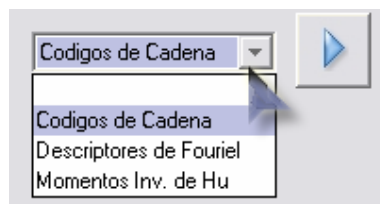
Carga de Imágenes

La función principal de esta parte del sistema es solamente la carga de una imagen cualquiera para el análisis detallado de esta por los descriptores. Al ser cargadas muestra una visualización de la imagen. Para esto solamente se tiene que dar clic sobre el icono de carpeta que permitirá explorar las carpetas de su disco para escoger los archivos requeridos (estas imágenes deben ser binarias).



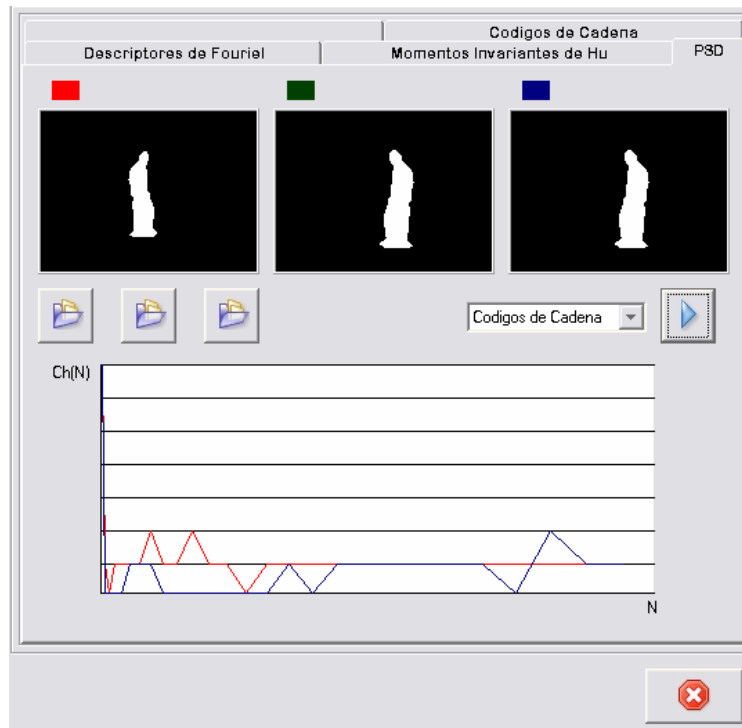
Métodos a Utilizar

Para este análisis es posible cargar cualquiera de los 3 métodos disponibles pero con restricciones especiales. Como por ejemplo los códigos de Cadena por defecto tienen 8-direcciones para el cálculo. Los descriptores de Fourier tienen también por defecto el uso de todos los puntos de la imagen para el análisis estadístico. Sin embargo los momentos de Hu, al contrario no tienen por su definición restricción alguna.



A continuación mostramos el uso de los 3 métodos bajo un número de imágenes cualquiera.

Como ejemplo del código de cadena tenemos el siguiente análisis:



Como ejemplo de los descriptores de Fourier tenemos el siguiente análisis:



Como ejemplo de los momentos invariantes de Hu tenemos el siguiente análisis:

