

Resumen:

El presente proyecto de tesis aborda el área de visión artificial, específicamente técnicas de reconocimiento de patrones en imágenes digitales. Durante el desarrollo del proyecto se estudiaron e implementaron un conjunto de algoritmos que permiten realizar la clasificación de objetos mediante el análisis de imágenes, utilizando para este fin patrones de entrenamiento que modelan la estructura de una red neuronal artificial. Mediante la implementación de un caso de estudio se visualizaron las posibilidades que surgen al incorporar este tipo de tecnología en tareas cotidianas realizadas por los seres humanos, se comenzó con la revisión de la literatura del área identificando las relaciones e implicancias existentes entre las técnicas de visión artificial estudiadas y los objetivos definidos inicialmente, después se seleccionó la metodología y procedimientos a utilizar en el caso de estudio según las características de la investigación, luego de implementar los métodos seleccionados y analizar los resultados obtenidos por estos se ha podido observar cómo se logra superar las prestaciones conseguidas por técnicas más tradicionales que trabajan sobre el dominio del espacio, al utilizar el reconocimiento de patrones en el dominio de las frecuencias.

Palabras claves: Redes neuronales artificiales, Transformada de Fourier, Visión por computador, Red neuronal multicapas, Reconocimiento de objetos, Segmentación de Objetos.

Abstract:

This thesis project addresses the area of computer vision, specifically pattern recognition techniques to digital images. During the development of the project is studied and implemented a set of algorithms that allow the classification of objects using image analysis, using for this purpose training patterns for shape the structure of an artificial neural network. By implementing a study case were visualized the possibilities that arise when implementing this type of technology in daily tasks performed by humans, it began with a review of the literature of the area and implications for identifying relationships between techniques of artificial vision studied and the objectives laid down, then selected the methodology and procedures used in the study case according the characteristics of the investigation, after implementing the selected methods and analyze the results obtained for these has been able to observe how they can overcome the performance achieved by more traditional techniques working in the space domain, using pattern recognition in the frequency domain.

Keywords: Artificial neural networks, Fourier Transform, Computer Vision, Multilayer neural network, object recognition, segmentation of objects

Índice

Capítulo 1	4
Introducción	4
1.1 Antecedentes de la investigación.	4
1.2 Problema de investigación y análisis de objetivos.	5
1.3 Justificación de la investigación.	6
1.4 Metodología.....	7
1.5 Esquema del informe	7
1.6 Definiciones.....	8
1.7 Delimitaciones de los alcances y asunciones importantes	8
1.8 Conclusión.....	9
Capítulo 2	10
Revisión de la Literatura	10
2.1 Introducción.....	10
2.2 Marco Teórico	11
2.2.1 Procesamiento de Imágenes Digitales - Dominio del Espacio.	12
2.2.2 Procesamiento de Imágenes Digitales - Dominio de las Frecuencias.....	30
2.2.3 Visión Artificial.....	45
2.2.4 Visión Artificial - Redes Neuronales Artificiales.....	47
2.3 Análisis del marco teórico y problema de investigación.....	68
2.4 Conclusión	70
Capítulo 3	71
Metodología	71
3.1 Introducción	71
3.2 Justificación del paradigma y la metodología	71
3.3 Caso de Estudio y Solución Propuesta	73
3.4 Procedimientos de investigación	77

3.5 Conclusión.....	78
Capítulo 4.....	79
Implementación y Análisis de Resultados	79
4.1 Introducción	79
4.2 Entorno y Tipo de Imágenes Utilizadas	80
4.3 Implementación de Algoritmos Seleccionados	85
4.4 Análisis de algoritmos Alternativos	90
4.5 Pruebas Finales del Proceso de Visión Artificial	98
4.6 Análisis del Desempeño Final	115
4.7 Conclusión.....	116
Capítulo 5.....	117
Conclusiones e Implicaciones	117
5.1 Introducción	117
5.2 Conclusiones acerca de los algoritmos implementados.....	118
5.3 Conclusiones sobre el problema de investigación	119
5.4 Limitaciones	120
5.5 Implicaciones para investigaciones posteriores.....	121
Referencias.....	123
Apéndice - I.....	124
Framework de Visión Artificial Utilizando Patrones.....	124

Introducción

En el presente capítulo se abarcan conceptos introductorios al tema de estudio considerado en este proyecto de obtención de grado, se dan aspectos generales sobre los antecedentes del área, estructura y contenidos del informe, metodología a utilizar y delimitaciones acerca de los alcances considerados.

1.1 Antecedentes de la investigación.

Gracias a los avances del área de la inteligencia artificial logrados en los últimos años se ha conseguido dotar a un procesador de la capacidad de coordinar acciones emulando así parte del comportamiento que posee un ser humano. Una característica complementaria consistiría en entregarles a las máquinas la capacidad de interacción con su entorno y una de las técnicas que aborda esta problemática es la denominada Visión Artificial (VA). El objetivo de la VA es proveer a un sistema de la capacidad de percepción visual que permita realizar tareas complejas, donde en la búsqueda de este objetivo en el último tiempo se ha llegado incluso a sobrepasar las capacidades del propio sistema visual humano.

Dos conceptos principales que surgen al abordar esta área son “Procesamiento de Imágenes Digitales” y “Visión Artificial”, donde el primero apunta a la utilización de algoritmos de procesamiento para extraer información relevante de las imágenes digitales y el segundo busca simular las características del sistema de visión humano permitiendo que un ordenador “entienda” las características de los elementos presentes de una determinada escena y logre interactuar con el entorno (por ejemplo en la navegación y manipulación de objetos).

En procesamiento de imágenes digitales se busca mejorar la información contenida en las imágenes generando una fuente en estado óptimo para procesamientos que sean aplicados posteriormente (pre-procesamiento), además existen algoritmos que permiten la identificación de los contornos de objetos, de segmentación de imágenes basado en las regiones de interés, de correlación de objetos y en general técnicas que permiten obtener de forma automatizada información que de otra manera un ser humano tardaría mayor tiempo en procesar. Los algoritmos iniciales de procesamiento de imágenes digitales consideran como elemento base trabajar sobre los valores de los píxeles (tonalidades) por lo cual se dice que estas técnicas trabajan sobre el “dominio del espacio”, otras técnicas en cambio trabajan con conjuntos de píxeles más complejos, que son los resultantes de aplicar transformaciones

matemáticas sobre las tonalidades de las imágenes (dominio espacial), estas técnicas son conocidas por trabajar en el dominio de la frecuencia [1][2].

Por otro lado, los algoritmos de visión artificial tienen su base ligada a los desarrollos obtenidos en el campo de la inteligencia artificial, se considera a la simulación de las capacidades humanas como una de las ideas centrales de esta área. Al intentar imitar el sistema visual humano se manejan conceptos que consideran desde la formación de una imagen con sus elementos constitutivos, hasta la interpretación hecha por el cerebro de los elementos y sus significados según el contexto donde persisten. Cuando se ha logrado obtener y generar una buena representación y descripción de los objetos presentes en una escena, se hace necesaria la clasificación automática de esta información y por lo tanto es en este punto cuando la visión artificial cobra importancia, ya que es deseable dotar al sistema de procesamiento de la *capacidad de interpretar automáticamente estos resultados* [3][4].

Esta investigación busca implementar métodos de procesamiento de imágenes digitales y visión artificial existentes que permitan crear un sistema computacional que ayude en labores de inspección visual que son desarrolladas por personal de vigilancia en una organización, por un sistema de video vigilancia asistida, mediante procesos de inspección industrial o en nuevas implementaciones que requieran la detección e interpretación automática de los elementos que aparecen en una escena.

Durante el desarrollo de esta investigación se observan los resultados obtenidos con diversos algoritmos donde para cada uno se analizan las ventajas y desventajas que poseen y que deben ser consideradas en su selección. Según el área de interés que se desee trabajar serán algunos métodos más pertinentes y exitosos que otros pero distinta puede ser la situación y los algoritmos cuando el contexto de ejecución cambia, todos estos detalles son analizados y mostrados más adelante en este documento.

1.2 Problema de investigación y análisis de objetivos.

Esta investigación mide los resultados obtenidos en una escena monitoreada con la implementación de algoritmos de visión artificial como la transformada de Fourier y una red neuronal artificial. Los objetivos generales y específicos de este trabajo son presentados a continuación.

Objetivo General

Estudiar técnicas del campo de la visión artificial, y a través de la implementación de métodos como los *descriptores de Fourier (DF)* y una *RNA de tipo MultiLayer Perceptron (MLP)* analizar los resultados obtenidos por estos métodos en la clasificación de los nuevos objetos presentes en una escena, basado en el reconocimiento de patrones en las imágenes digitales que sean capturadas desde el entorno.

Objetivos Específicos

- Estudiar los algoritmos tradicionales de procesamiento de imágenes digitales existentes.
- Estudiar la transformada de Fourier y su aplicación dentro del procesamiento de imágenes digitales.
- Estudiar la clasificación de patrones mediante la implementación de una RNA de tipo MLP que utilice el algoritmo de aprendizaje Back Propagation (BP).
- Abordar un caso de estudio donde mediante la implementación de algunas de las técnicas de visión artificial estudiadas se logró clasificar los nuevos objetos que aparecen en la escena, analizando los resultados obtenidos de acuerdo al reconocimiento de patrones alcanzado.
- Mejorar los resultados de técnicas tradicionales que se basan en el dominio del espacio (tonalidades de los píxeles) y a través de las conclusiones obtenidas sentar las bases suficientes que permitan realizar propuestas de futuras investigaciones que aborden técnicas innovadoras para el área (nuevas líneas de investigación).

Estos objetivos son desarrollados en las secciones y capítulos siguientes de este documento, se comienza revisando los algoritmos existentes en la literatura del área y luego se selecciona un conjunto de métodos que permitan analizar la secuencia de imágenes de un entorno particular y realizar la detección de los objetos presentes en ella, luego de su implementación y en base a los resultados entregados por cada uno de estos algoritmos se realizan mejoras y/o reemplazos de alguno(s) de ellos buscando optimizar la interpretación alcanzada por el sistema.

1.3 Justificación de la investigación.

Los beneficios logrados en las labores cotidianas cuando estas son apoyadas por sistemas computacionales son el principal justificativo cuando se evalúa el estudio y análisis de incorporar nuevas tecnologías.

Por otra parte, este estudio permite conocer detalles de los algoritmos de visión artificial existentes en la actualidad mostrando la implementación de estos sobre un problema práctico del área (caso de estudio). Gracias a los avances tecnológicos de las últimas décadas existen muchas posibilidades de incorporar este tipo de técnicas en diversos ámbitos mediante aplicaciones y soluciones tecnológicas, ya que la disminución de costos y el aumento del nivel de procesamiento en los medios electrónicos de filmación, la transmisión y la velocidad de procesamiento de datos actuales permiten implementar algoritmos que en décadas pasadas no hubiesen sido viables (dado que no tenían los medios donde realizar de forma eficiente y rápida sus operaciones).

Existe la necesidad de abordar técnicas de procesamiento en el dominio de las frecuencias y de utilizar las capacidades de una red neuronal para la clasificación objetos en una escena particular, con la finalidad de mejorar los resultados de técnicas tradicionales que se basan en el dominio del espacio (tonalidades de los pixeles) y de esta forma sentar las bases suficientes que permitan realizar propuestas de investigaciones futuras que aborden técnicas innovadoras para el área (nuevas líneas de investigación).

1.4 Metodología

El desarrollo de los objetivos contempla como actividad inicial el estudio del área identificando los algoritmos que permitan obtener los resultados esperados considerando la implementación de estos según las etapas típicas de un proceso de visión artificial, en el capítulo 3 se aborda en detalle la metodología escogida y las etapas específicas que esta considera. La implementación de cada una de las etapas definidas en el modelo dependerá de las características y los objetivos particulares de cada proyecto, en cada etapa existe un conjunto de algoritmos que cumplen con las mismas funcionalidades, de manera que habrá que evaluar las distintas alternativas y analizar cual se comporta de mejor manera dentro del entorno específico abordado (una técnica puede entregar buenos resultados en un entorno dado, pero no tendrá el mismo comportamiento en todos los casos existentes o antes cambios significativos de este). Dentro de los puntos a considerar en la selección de las técnicas, están las características que estos mejor interpreten (detección orientada a regiones, orientada a fronteras, según formas geométricas, etc.) y los resultados que han logrado en estudios anteriores (porcentajes de efectividad, tiempo de respuesta, entre otros).

Además existen diversas condiciones del entorno que deben ser consideradas (contraste, calidad del dispositivo de captura y del medio de transmisión, tipo de iluminación del entorno, entre otros), en la literatura del área se pueden encontrar diversos métodos que ayudan a tratar estos factores.

1.5 Esquema del informe

La estructura del informe considera 5 capítulos:

- Capítulo 1 - Introducción.
- Capítulo 2 - Revisión de la Literatura.
- Capítulo 3 - Metodología.
- Capítulo 4 - Implementación y Análisis de Resultados.
- Capítulo 5 - Conclusiones e Implicaciones.

Donde el capítulo 1 introduce el problema central de la investigación, presenta los lineamientos iniciales y describe el camino a seguir hasta llegar a las conclusiones finales del proyecto, el capítulo 2 muestra el estado del arte correspondiente al área de estudio y relaciona su contenido con el cumplimiento de los objetivos propuestos inicialmente, en el capítulo 3 se comenta la metodología y los métodos utilizados para recopilar datos y analizar los resultados del proceso implementado, el capítulo 4 muestra los resultados de aplicar esos métodos en la investigación y en el capítulo 5 finalmente se dan las conclusiones sobre el cumplimiento de los objetivos según resultados del capítulo 4, también se entregan consideraciones finales acerca del alcance del proyecto y de nuevas líneas de investigación a considerar en futuros trabajos.

1.6 Definiciones

Las definiciones adoptadas por los investigadores no son a menudo uniformes, así que algunos términos principales y polémicos se definen a continuación para establecer las posiciones asumidas en el desarrollo de esta investigación.

Algoritmos de procesamiento de imágenes digitales: Consideraremos un algoritmo de procesamiento de imágenes digitales a aquella secuencia de pasos que se ejecutan sobre una o varias imágenes digitales y que buscan recuperar información de estas.

Algoritmos de Visión Artificial: Consideraremos un algoritmo de visión artificial a aquella secuencia de pasos que se ejecutan sobre una o varias imágenes digitales y que buscan recuperar información de estas para lograr autónomamente interpretar la escena (imitando las características del sistema visual humano).

1.7 Delimitaciones de los alcances y asunciones importantes

Esta investigación mide los resultados obtenidos en escenas particulares con la implementación de algoritmos de visión artificial como la transformada de Fourier y una red neuronal artificial (RNA) de tipo Perceptrón multicapa, de acuerdo con el objetivo general planteado anteriormente en la sección 1.2 que busca “Estudiar técnicas del campo de la visión artificial, y a través de la implementación de métodos como los descriptores de Fourier (DF) y una RNA de tipo MultiLayer Perceptron (MLP) analizar los resultados obtenidos por estos métodos en la clasificación de los nuevos objetos presentes en una escena, basado en el reconocimiento de patrones en las imágenes digitales que sean capturadas desde el entorno”, en esta sección se presentan algunas delimitaciones sobre el alcance del trabajo a desarrollar.

Como se dijo en el párrafo anterior esta tesis considera el estudio e implementación de un grupo de algoritmos de procesamiento de imágenes digitales así como también de visión artificial especificados con más detalle en el capítulo 4, donde el objetivo perseguido es verificar el comportamiento de estos frente a imágenes de distinta naturaleza y detectar por lo tanto aquellas fortalezas y debilidades que sean visualizadas de acuerdo al grado de certeza que se logre en la clasificación de objetos. Sin embargo, no se considera dentro de los objetivos el lograr una solución tecnológica robusta que presente altas prestaciones de asertividad dado el amplio número de variables a considerar al procesar una imagen digital (tipo de instrumento de captura, ambiente de grabación, situación climática del ambiente, tipo de iluminación del entorno, tiempos de procesamiento, entre otros) y donde dedicar todos los esfuerzos en la obtención de un producto de estas condiciones se escapa de la línea principal de investigación ya mencionada. Tampoco se considerarán en este documento otras técnicas como por ejemplo las asociadas a conceptos de reconstrucción de un mundo tridimensional a partir de imágenes bidimensionales (como los descritos anteriormente) o cualquier otro que no tenga una relación directa con los objetivos propuestos.

El grupo de conceptos a estudiar en esta tesis no considera alternativas de implementaciones innovadoras pues la base conceptual estudiada recoge antecedentes sobre técnicas ya existentes como la descripción del contenido de una imagen dentro del dominio de las frecuencias así como de la implementación de una red neuronal artificial que sea capaz de interpretar y reconocer dichas descripciones. Queda descartado por lo tanto un amplio estudio de conceptos incipientes como por ejemplo la ingeniería biónica, aunque igualmente se considera que la base conceptual alcanzada al terminar este trabajo será de utilidad para realizar propuestas de trabajos futuros relacionados a estas alternativas.

1.8 Conclusión

En este primer capítulo se han comentado todos los temas que serán abarcados con un mayor grado de detalle en las siguientes páginas del documento: se dieron antecedentes sobre el área de estudio, se explicaron los objetivos planeados para el desarrollo de la tesis, se justificó la investigación a realizar y se entregó una visión general de la metodología utilizada, también se comentó la estructura del informe y se acordaron aquellas definiciones a utilizar para conceptos que presentan ambigüedades o poca claridad en la literatura consultada, finalmente también se acotaron los alcances de la investigación exponiendo aquellas consideraciones que no se habían entregado de forma explícita en la sección 1.2 evitando de esta forma probables confusiones y dando claridad acerca del marco de trabajo seguido a lo largo del documento.

Revisión de la Literatura

En las páginas anteriores se entregó una visión general del contenido y alcance de la presente investigación, planteando los objetivos específicos que se buscan alcanzar. En el estado del arte expuesto en este capítulo se identifican y revisan las dimensiones teóricas y metodológicas de la literatura, donde en las últimas secciones se incluyen las asociaciones de estos conceptos con los objetivos definidos inicialmente.

2.1 Introducción

En la siguiente sección serán abordados un conjunto de conceptos y técnicas utilizadas dentro del procesamiento de imágenes digitales y del área de la visión artificial, los métodos estudiados han sido incluidos dado la importancia que poseen dentro de la literatura y los buenos resultados que han demostrado en trabajos relacionados. El estado del arte comienza estudiando los modelos de color existentes, luego describe técnicas de procesamiento de imágenes digitales que permiten la eliminación de ruido, mejora del contraste, segmentación de regiones, detección de bordes, entre otros, finalmente se exponen las características de una red neuronal artificial y se dan antecedentes acerca de los factores que inciden en el desempeño de la clasificación que esta realiza, como lo son por ejemplo el número de capas y de neuronas utilizadas, factor de aprendizaje, entradas utilizadas para el entrenamiento, etc.

Luego de examinar la literatura del área se discuten los alcances de la investigación y se analizan los objetivos planteados de acuerdo con la base teórica revisada, se busca asociar los métodos disponibles en la literatura con líneas de investigación que aseguren un buen desarrollo de la tesis y también permitan finalmente la obtención de conclusiones acerca del comportamiento de los algoritmos estudiados e implementados.

2.2 Marco Teórico

La presentación del contenido dentro del marco teórico sigue el siguiente orden lógico:

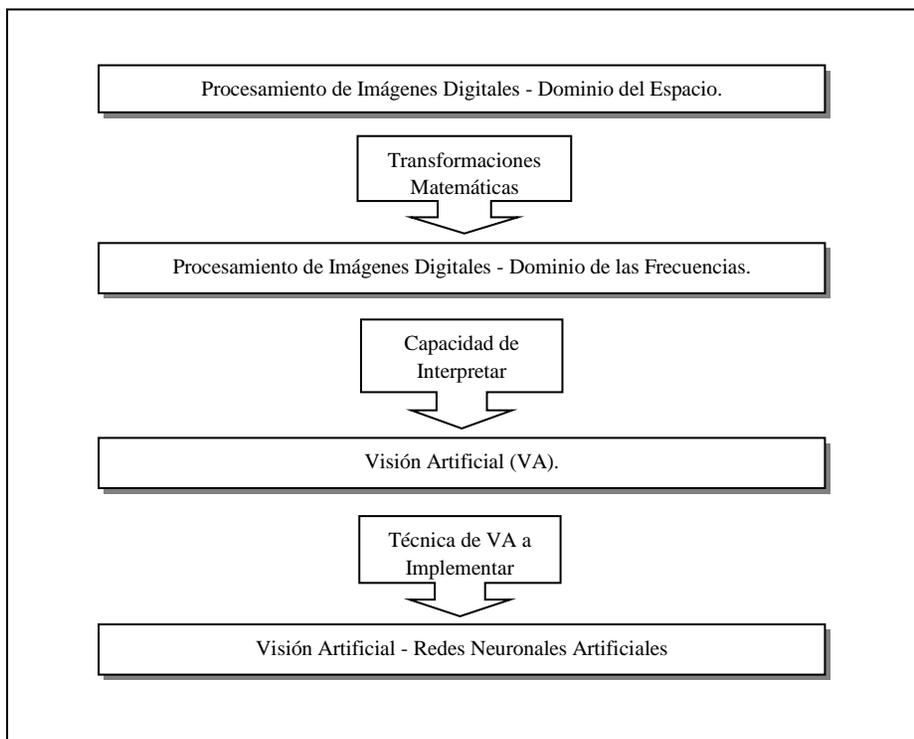


Figura 2.2.1 – Orden lógico seguido por el marco teórico.

Donde se comienza con la revisión de técnicas del dominio del espacio que trabajan fundamentalmente sobre los valores de las tonalidades de aquellos objetos presentes en una escena, estas buscan principalmente realizar mejoras sobre la información contenida en las imágenes digitales, luego se exponen otras técnicas de procesamiento que mediante transformaciones matemáticas sobre los valores de las tonalidades de una imagen logran obtener información con un menor grado de dependencia de factores externos como por ejemplo el ruido provocado por interferencias del medio de captura o factores climáticos que alteran la correcta visualización del entorno.

Cuando un sistema de procesamiento de imágenes digitales logra interpretar de forma autónoma la información que procesa se dice que adquiere capacidades propias del sistema visual humano (VA), el estado del arte también repasa algunos conceptos fundamentales de la visión artificial donde el estudio se centra sobre las redes neuronales artificiales (que es una de las técnicas existentes que permiten dotar a un sistema computacional de esta capacidad de interpretación).

2.2.1 Procesamiento de Imágenes Digitales - Dominio del Espacio.

En esta sección se abordan conceptos introductorios al área abordada, luego se estudian principalmente técnicas clásicas que basan su funcionamiento en las tonalidades de los objetos presentes en una imagen digital.

Formación de una imagen digital

Tradicionalmente las imágenes eran de tipo “Analógicas” aunque hoy en día la gran mayoría de los sistemas existentes entregan imágenes ya digitalizadas, igualmente para conocer el proceso a través del cual una imagen analógica, en blanco y negro, en escala de grises o a color podía ser manipulada usando un ordenador, se presentan en las siguientes líneas 2 conceptos involucrados en la conversión de estas imágenes a un formato adecuado para su procesamiento: Muestreo y Cuantificación.

El *muestreo* consiste básicamente en la subdivisión de la imagen analógica en particiones, las cuales consideran polígonos regulares de 3 tipos: triángulos, cuadrados y hexágonos, estos polígonos representan sensores perceptivos de la intensidad de la luz y como salida generan un valor de “amplitud” que se ubica dentro de una escala determinada (color). La salida puede ser un único valor (escala de grises) o bien un vector con más de un valor por polígono como sucede en el modelo de color RGB (explicado más adelante en esta misma sección) donde estos valores se corresponden con la intensidad de color rojo (Red), verde (Green) y azul (Blue), a este proceso de discretización del color se le llama *cuantificación* y al polígono de color constante se le llamara “*píxel*” (para el caso de imágenes en escala de grises los píxeles están compuestos por sombras de grises, que van desde el negro más profundo variando gradualmente en intensidad hasta llegar al blanco).

Las imágenes capturadas con una cámara, un telescopio, un microscopio o cualquier otro tipo de instrumento óptico presentarán variaciones en sus tonalidades originales y también incluirán la presencia de sombras sobre algunos de sus elementos lo que hace necesario procesar esta información en busca de mejorar la visualización de la escena representada.

El Color.

En 1666 Isaac Newton descubrió que cuando un rayo de luz solar pasa a través de un prisma de cristal, al salir no presenta un color blanco sino que está formado por un espectro continuo de colores que van desde el violeta al rojo. Podría decirse que el espectro de color puede dividirse en seis amplias regiones: violeta, azul, verde, amarillo, naranja y rojo. Sin embargo, cuando se ven todos los colores juntos ningún color del espectro termina bruscamente, más bien se mezcla suavemente con el siguiente.

La idea de color es inicialmente muy simple, los colores que los seres humanos percibimos en un objeto vienen marcados por la naturaleza de la luz que refleja dicho objeto, un cuerpo que refleja luz relativamente equilibrada en todas las longitudes de onda visibles aparece en blanco, por otro lado, un objeto que refleja en un rango limitado de espectro visible muestra algún tipo de sombra de color.

La caracterización de la luz es fundamental para la ciencia del color, si la luz no tiene color su único atributo es la intensidad, luz sin color es lo que vemos en las televisiones en blanco y negro, la recepción de televisión en color es un ejemplo de la naturaleza aditiva de los colores.

Existen los denominados “modelos de color”, que tienen como propósito facilitar la especificación de colores en algún formato estándar. En esencia, un modelo de color es una especificación de un modelo de coordenadas 3-D y un subespacio dentro de ese sistema donde cada color se representa por un punto único, existen diversos modelos y cada uno tiene campos de aplicación distintos según los componentes que utilizan para su representación, RGB (rojo-verde-azul) es comúnmente utilizado en hardware, CMY (cian-magenta-amarillo) es utilizado en impresoras en color, YIQ (luminancia y dos componentes cromáticas) es el estándar para la difusión de TV, HSI (matiz-saturación-intensidad) y HSV (matiz-saturación-valor) son utilizados para manipular el color, estos son algunos de los modelos de color más utilizados. Para el presente proyecto se estudian con mayor detalle los modelos RGB y HSV debido a la importancia que tienen dentro del área del procesamiento de imágenes digitales.

Modelo de Color RGB

En el modelo RGB cada color aparece en sus componentes primarias espectrales rojo, verde y azul, el modelo está basado en un sistema de coordenadas cartesiano ya que debido a la estructura del ojo humano, todos los colores se ven como una combinación de estos tres llamados básicos. Con la idea de producir un estándar la Comisión Internacional de Iluminación (CIE) designó en 1931 las siguientes longitudes de onda para los tres colores primarios: azul = 435.8 nm, verde = 546.1 nm y rojo = 700nm, sin embargo un solo color no podría ser llamado rojo, verde o azul, por lo que tener longitudes de onda específicas para cada uno de los tres colores no significa que estas tres componentes trabajando de manera independiente puedan generar todos los colores, esto es importante ya que se cree, erróneamente, que estos tres colores mezclados pueden producir todos los colores visibles, esto no es cierto salvo que también permitamos que las longitudes de onda varíen.

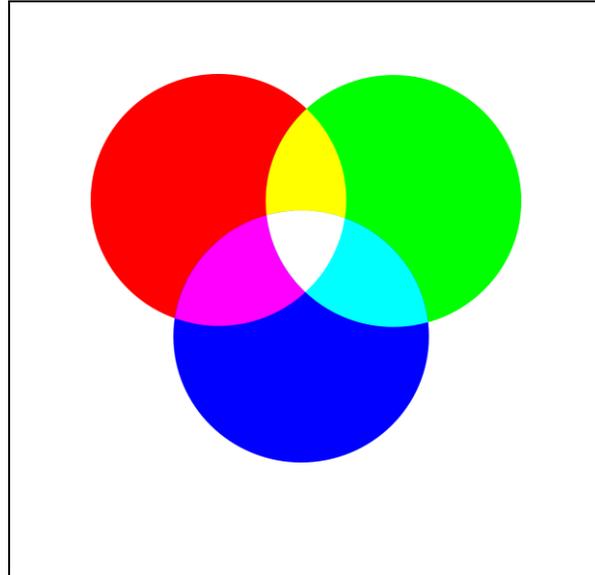


Figura. 2.2.1.1 - Modelo aditivo de colores rojo, verde, azul.

RGB no siempre es el mejor modelo de color para el procesamiento de imágenes, considerando por ejemplo que se quiere mejorar la representación de una imagen a color de una cara humana donde parte de la misma está oculta por una sombra, el problema puede ser abordado a través del ajuste de su histograma, pero al tener tres bandas y puesto que el histograma es una técnica que trabaja sólo con las intensidades, cuando se ajuste en cada banda obtendremos mejoras sin embargo el resultado global (cuando vuelvan a combinarse las tres bandas) será impredecible (otros modelos de color como por ejemplo HSV son más útiles en este tipo de problemas).

Modelo de Color HSV

El modelo HSV (del inglés Hue, Saturation, Value – Tonalidad, Saturación, Valor), también llamado HSB (Hue, Saturation, Brightness – Tonalidad, Saturación, Brillo), fue creado en 1978 y es un sistema que realiza variaciones de las propiedades del color para crear nuevos colores. El matiz o tonalidad (Hue) especifica el "color", rojo, naranja, azul, etc, la saturación (Saturation) conocida también como cromo o pureza, se refiere a la cantidad de blanco en la componente de tonalidad, un color completamente saturado no contiene blancos y aparece puro, un rojo 50% saturado resulta un rosa, el valor (Value), conocido también como brillo, corresponde al grado de luminosidad de un color (que tanta luz emite), un matiz con alta intensidad es brillante, uno con poca intensidad es oscuro.

Este sistema es muy similar al utilizado por los pintores y otros artistas, donde agregan blanco, negro y gris a los pigmentos puros y así crean tintes, sombras y tonos. Un tinte es un color puro combinado con blanco, una sombra se produce con un color puro completamente saturado combinado con negro y un tono es un color completamente saturado combinado con ambos (gris): blanco y negro.

Se trata de una transformación no lineal del espacio de color RGB donde sus componentes se basan en coordenadas cilíndricas:

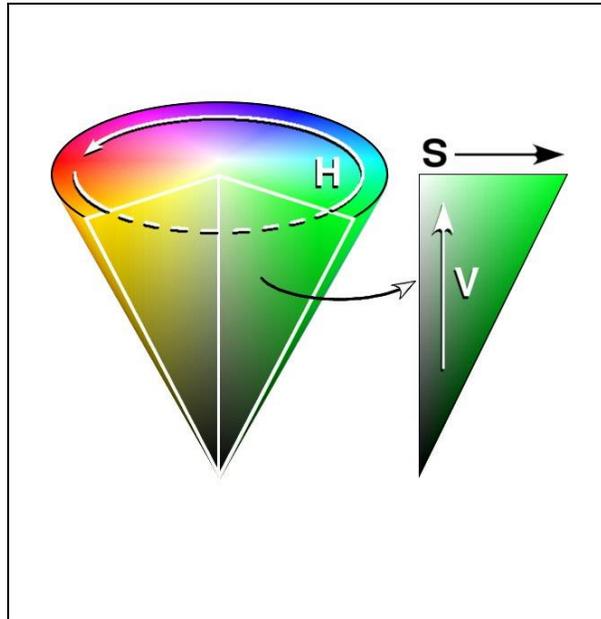


Figura. 2.2.1.2 - Cono de colores del espacio HSV

Coordenadas del espacio HSV:

- Tonalidad: Tipo de color (como rojo, azul o amarillo). Se representa como un grado de ángulo cuyos valores posibles van de 0 a 360° (aunque para algunas aplicaciones se normalizan del 0 al 100%), cada valor corresponde a un color. Ejemplos: 0 es rojo, 60 es amarillo y 120 es verde.
- Saturación: Se representa como la distancia al eje de brillo negro-blanco. Los valores posibles van del 0 al 100%, cuanto menor sea la saturación de un color, mayor tonalidad grisácea habrá y más decolorado estará, por eso es útil definir la insaturación como la inversa cualitativa de la saturación.
- Valor del color, brillo del color: Representa la altura en eje blanco-negro. Los valores posibles van del 0 al 100%, 0 siempre es negro y dependiendo de la saturación 100 podría ser blanco o un color más o menos saturado.

Realce de Contraste.

El realce de contraste tiene como objetivo mejorar la calidad de una imagen bajo ciertos criterios subjetivos del ojo humano, normalmente esta técnica es utilizada en una etapa denominada “Preprocesamiento”.

El contraste entre dos objetos se puede definir como la razón entre sus niveles de gris medio. El ajuste del contraste consiste en la manipulación radiométrica en cada píxel, con el objetivo de aumentar la discriminación visual entre los objetos presentes en la imagen, esta operación es realizada punto a punto y ayudada de histogramas (representación grafica de las tonalidades presentes en una imagen) manipulados para obtener el realce deseado.

Técnicas de Suavizado.

Las técnicas de suavizado buscan reducir la cantidad de variaciones de intensidad existentes entre píxeles vecinos (adyacentes) y eliminar el ruido o detalles pequeños que no sean de interés en el procesamiento de las imágenes digitales (influencia de campos electromagnéticos, efectos del clima, entre otros). En la actualidad existe una gran cantidad de algoritmos para la restauración y mejoramiento de imágenes.

Las técnicas de filtrado son básicamente transformaciones de la imagen píxel a píxel, que no solamente dependen de las tonalidades individuales sino que también del valor de los píxeles vecinos, para llevar a cabo el proceso se utilizan matrices denominadas “mascaras” que son aplicadas sobre la imagen, estos filtros sirven para suavizar realzando detalles y minimizando los efectos del ruido presente en la imagen.

Uno de los filtros más utilizados es la “*Mediana*”, este método consiste en sustituir con la mediana de los píxeles vecinos o vecindad (generalmente mejor a otras técnicas basadas en el promedio), se visita cada píxel de la imagen y se reemplaza por la mediana de la vecindad (la mediana se calcula ordenando los valores de los píxeles vecinos en orden y seleccionando el que queda en medio).

Las principales ventajas de este filtro es reducir el empañamiento de los bordes y no ser afectado por píxeles individuales ruidosos, presentando resultados bastante buenos.

Entre sus desventajas está la disminución de su desempeño cuando el número de píxeles con ruido es mayor a la mitad del número total de píxeles en la vecindad, también tiene el inconveniente frente a los filtros lineales de ser más lento. [2]

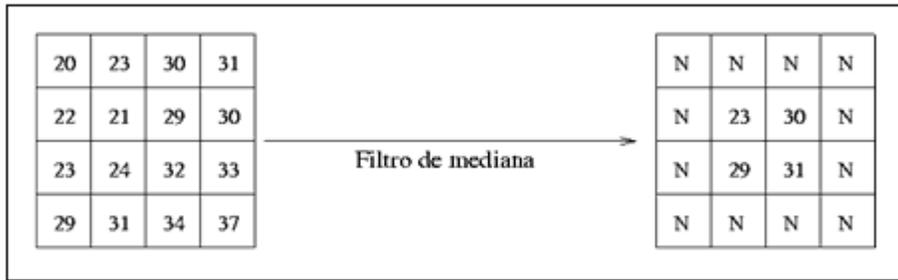


Figura 2.2.1.3 - Ejemplo de mascara filtrada con la mediana

Ejemplos que muestran la eliminación del ruido presente en una imagen mediante el filtro de la mediana y la media (mismo procedimiento pero basado en la media estadística) se muestran en la siguiente figura [1]:

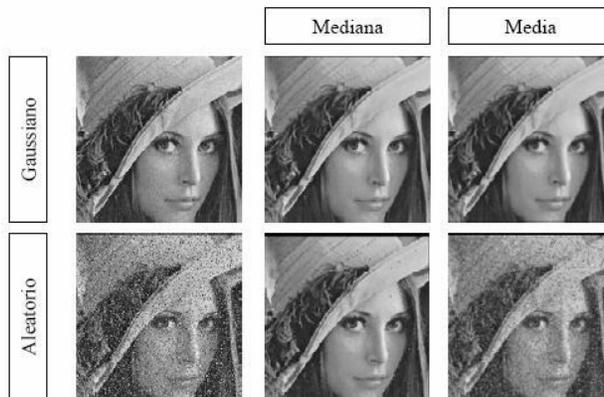


Figura 2.2.1.4 - Ejemplo filtrado mediante la mediana y la media.

Técnicas de Segmentación.

Antes de explicar los procesos de segmentación de imágenes digitales es necesario dar la definición de lo que se entiende por segmentación, una segmentación es un conjunto finito de regiones R_1, \dots, R_s tales que:

$$R = \bigcup_{i=1}^s R_i \quad R_i \cap R_j = \emptyset, i \neq j$$

Formula 2.2.1.1 – Definición de segmentación.

Existen diversas técnicas de segmentación pero todas siguen la premisa planteada en la formula anterior, a continuación se detallan algunas de ellas.

Segmentación basada en umbralización:

Umbralización: Este tipo de segmentación se basa en la elección de un determinado nivel de gris en la imagen que sea característico de un objeto (Umbral), en el caso ideal se selecciona un solo valor y se segmenta la imagen separando los píxeles que están por sobre este umbral de los que poseen un nivel de intensidad inferior a él. [2] En la práctica la mayoría de los casos este escenario “ideal” no es el real, y lo que se hace es elegir un grupo de umbrales que sean característicos, además es deseable que su elección se realice de forma automática por el sistema.

Umbralización Adaptativa: La existencia de sombras como también la dirección de la iluminación en una escena hace que el método anterior no sea apropiado debido a discontinuidades de intensidad en píxeles pertenecientes a la misma región. Una solución a este problema son los umbrales adaptativos, los cuales examinan regiones pequeñas de la imagen y entonces analizan cada subimagen para obtener un umbral acotado a esa región.

Segmentación basada en la forma del objeto

Si conocemos la forma de los objetos presente en una imagen, la segmentación consistirá en localizar dichos objetos, uno de los métodos que entrega buenos resultados en la segmentación basada en la forma de los objetos es la transformada de Hough.

Transformada de Hough: La transformada de Hough corresponde a un algoritmo que trabaja sobre una secuencia de puntos de bordes de una imagen, entrega como salida la identificación de aquellos puntos que son bordes verdaderos, es decir, aquellos que pertenecen a las fronteras reales de los objetos. [2] [5]

En la utilización de este método se tienen los siguientes elementos:

1.- Una imagen I de niveles de gris y de dimensiones $N \times N$, a partir de esta imagen mediante un proceso de detección de bordes podemos obtener un conjunto de m puntos:

$$P_i = \{(x, y) / (x, y) \text{ puntos de bordes en } I\}$$

Formula 2.2.1.2 - Puntos de bordes de una imagen I .

Este conjunto de puntos será la entrada del algoritmo, se denominan “evidencias” y son los candidatos a pertenecer a la curva de un objeto dado.

2.- Un modelo de curvas o familia de curvas que queremos localizar (rectas-círculos-elipses, etc.):

$$F(x, v) = 0$$

Formula 2.2.1.3 – Familia de curvas

Donde $v = (a_1, \dots, a_n)$ es el vector de parámetros de la curva, corresponde a un punto del espacio de parámetros.

La transformada de Hough establece un método por el que se transforma el conjunto de evidencias desde el espacio de la imagen al espacio de los parámetros, de esta manera el problema de localizar un conjunto de puntos dispersos en el primer espacio se traduce en determinar un único punto en el segundo, ya que todos los puntos candidatos que pertenezcan a la misma curva en el espacio de la imagen, se relacionan en un único punto en el espacio de parámetros (ver figura 2.2.1.5). Se puede llevar a cabo dicha transformación para distintas familias de curvas, a continuación estudiaremos el caso particular de las rectas:

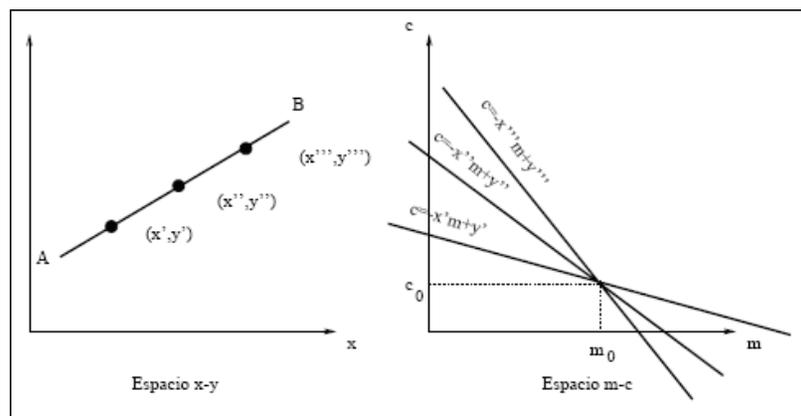


Figura 2.2.1.5 - Dualidad espacio de la imagen y espacio de parámetros

En la figura anterior [2], se muestra la dualidad mencionada, el espacio x-y corresponden al espacio de la imagen y el espacio m-c al espacio de parámetros, donde en este caso se representa una línea recta representada por la ecuación:

$$y = mx + c.$$

Formula 2.2.1.4. - Ecuación de una línea recta considerada por Hough

La ecuación presentada anteriormente tiene el inconveniente de permitir que una pendiente pueda tomar valores infinitos, por lo cual una mejora a la transformada de Hough consiste en utilizar la ecuación:

$$p = x \cos(\theta) + y \sin(\theta)$$

Formula 2.2.1.5. – Mejora de la ecuación de una línea recta considerada por Hough.

Donde p es la distancia del origen (0,0) a la recta (punto más cercano) y θ corresponde al ángulo formado por la recta que une al origen con el punto más cercano y el eje de las abscisas [2]:

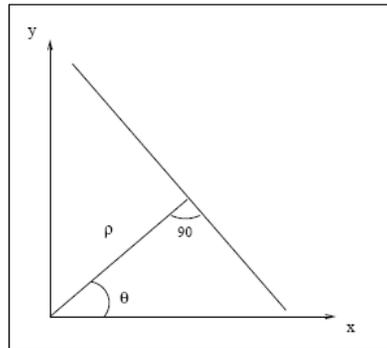


Figura 2.2.1.6 - Representación mejorada de la ecuación de una línea recta considerada por Hough.

En la siguiente figura se muestra un ejemplo de la detección de 4 rectas utilizando la mejora de la ecuación de la línea recta propuesta por Hough[2]:

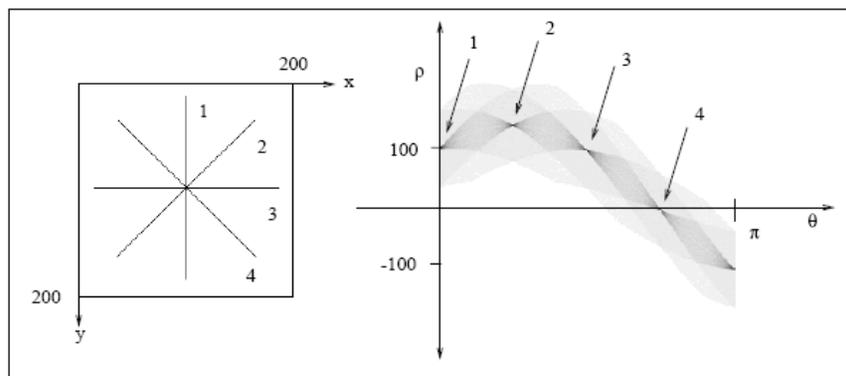


Figura 2.2.1.7 - Ejemplo de 4 rectas detectadas utilizando la mejora de Hough.

Segmentación basada en la detección de objetos de primer plano

Cuando se tiene una secuencia de imágenes digitales que representan un entorno monitoreado, inicialmente una cantidad acotada de estas imágenes incorporarán los elementos estáticos presentes en la escena, luego las siguientes imágenes que sean capturas mantendrán la misma cantidad de elementos estáticos ubicados en el mismo espacio físico. Basados en la situación anterior, entonces sería posible establecer un modelo del “fondo” de la escena, mediante parámetros de una distribución de probabilidad estándar (normalmente Gaussiana) que permita determinar si un objeto corresponde a un nuevo elemento (objeto de “Primer Plano”).

Con este tipo de algoritmos se logra detectar en cualquier escenario la existencia de nuevos elementos y separarlos del fondo de la escena, logrando la segmentación de la imagen. Este tipo de métodos mantiene cierta tolerancia al ruido y a pequeñas variaciones tales como hojas en movimiento, parpadeo de luces, vibraciones de las cámaras, etc.

El método de segmentación a través de la detección de objetos de primer plano utilizado en el presente trabajo corresponde al modelado de una “Gaussiana Simple”, donde se consideran los cambios pequeños de tonalidades que ocurren en una imagen de fondo (B_t) representando cada pixel con una distribución unimodal Gaussiana definida por dos parámetros: media μ_t y la varianza σ_t^2 .

$$\mu_t(x, y) = \sum_{i=1}^{i=t} \frac{B_i(x, y)}{t}$$

Formula 2.2.1.6 – Media de un pixel modelado con Gaussiana Simple.

$$\sigma_t^2(x, y) = \sum_{i=1}^{i=t} \frac{B_i^2(x, y)}{t} - \mu_t^2(x, y)$$

Formula 2.2.1.7 – Varianza de un pixel modelado con Gaussiana Simple.

En cada instante t , se determina que un pixel pertenece al fondo $B_t(x, y)$ cuando su valor de tonalidad recae dentro de la Gaussiana definida para ese pixel en la imagen, es decir, cuando la diferencia entre el valor del pixel y el de la media observada para dicho pixel específico $\mu_t(x, y)$ es inferior a la varianza $\sigma_t^2(x, y)$.

Técnicas de detección de bordes.

Los bordes corresponden a cambios locales significativos en una imagen y son parte importante del análisis de la misma, corresponden a una discontinuidad de algún tipo en la función de intensidad por lo que son fundamentales para descubrir rasgos relevantes y para estimar la estructura o propiedades de los objetos presentes en la escena.

Antes de describir el funcionamiento de algunos tipos de detectores de bordes es necesario entregar las algunas definiciones [2]:

Definición 1: Un punto de borde es un punto geométrico con coordenadas (i, j) ubicado sobre un cambio de intensidad local significativo en la imagen.

Definición 2: Un fragmento de borde es la i, j coordenada de un borde y su orientación (θ) puede corresponder al ángulo del gradiente.

Definición 3: Un contorno es una lista de bordes, o la curva matemática que modela esta lista.

Definición 5: La unión de bordes es el proceso de formar una lista ordenada de bordes a partir de una no ordenada.

Definición 6: El seguimiento de bordes es el proceso de búsqueda en una imagen (previamente filtrada) con el objetivo de determinar los contornos de los elementos existentes en ella.

El conjunto de bordes que produce un detector puede dividirse en dos clases: los “correctos” y los “falsos”. Un tercer conjunto corresponde a los bordes que deberían haber sido detectados, este tipo de bordes reciben el nombre de falsos negativos mientras que los falsos corresponden a falsos positivos.

Para mejorar el rendimiento de un detector de bordes es necesario realizar primero un proceso de filtrado y realce de la imagen, donde se busca suprimir el ruido existente (altas frecuencias) teniendo en consideración evitar el emborronamiento de la imagen producido en este paso (filtrado), y detectar los cambios de intensidad presentados por los vecinos de un pixel, buscando resaltar las altas frecuencias que existan en la imagen (realce).

Detectores de bordes

Operador de Sobel: Algunos métodos para resaltar bordes dan la misma importancia a los píxeles que se encuentran sobre el eje vertical u horizontal que a los que se encuentran en la diagonal de la matriz que representa la imagen (Operador Prewitt), el método de Sobel intenta compensar este efecto dándole una mayor importancia a los píxeles que están más cerca del centro de la máscara. Las máscaras que implementan este método para aproximar el gradiente son [2]:

1	0	-1	-1	-2	-1
2	0	-2	0	0	0
1	0	-1	1	2	1

Figura 2.2.1.8 - Mascaras de Sobel

El operador de Sobel es uno de los detectores de bordes más utilizados.

Operador Laplaciano: El operador Laplaciano marca la presencia de un borde cuando la salida del operador realiza una transición por cero [1], los ceros triviales, es decir, las regiones de cero uniformes tienen que ser ignoradas. Ejemplo de mascarar que aproximan el Laplaciano son las siguientes:

0	1	0	1	4	1
1	-4	1	4	-20	4
0	1	0	1	4	1

Figura 2.2.1.9 - Mascaras de aproximaciones al Laplaciano.

La aproximación con valor central -20 da mayor peso a los píxeles centrales que a los del contorno.

Operador Laplaciana de Gaussiana (LoG): Este método propone una aproximación de los cruces por cero de la segunda derivada de una forma más robusta, fue propuesto en 1979 por Marr y Hildreth y expone que los cambios en una imagen, ya sea por brillo o por color, pueden ser detectados encontrando los ceros de $G(x,y)*I(x,y)$ para una imagen I donde $G(x,y)$ es una distribución Gaussiana de dos dimensiones, es decir, debemos convolucionar la imagen con un filtro Gaussiano para obtener una imagen de salida a la cual habría que buscarle sus cruces por cero(Laplaciana) [2] [3].

El proceso de búsqueda de la segunda derivada es muy robusto, el alisamiento Gaussiano suprime la influencia de píxeles lejanos y la Laplaciana es una medida eficiente y estable de los cambios en la imagen.

Medidas de calidad de un detector de bordes:

Un método que permite medir la calidad de un detector de bordes es el propuesto por Abdou y Pratt [2] [4], quienes definieron una ecuación basada en que el merito conseguido por un detector debe ser medido en función de los bordes detectados de forma correcta. El merito definido por Abdou y Pratt es:

$$F = \frac{1}{\max(l_i, l_A)} \sum_{l=i}^{i_A} \frac{1}{1 + \alpha d^2(i)}$$

Formula 2.2.1.8 – Merito de un detector de bordes.

Donde l_i es el número ideal de puntos de borde, l_A es el número de puntos de borde detectados, $d(i)$ es la menor distancia del borde i -ésimo al borde ideal y “ α ” es una constante positiva. F es siempre menor o igual que uno y alcanza la igualdad cuando $l_i=l_A$ y $d(i)=0$ para todo i .

Morfología Matemática.

La morfología matemática es un marco de trabajo que se basa fundamentalmente en geometría y forma, las operaciones morfológicas simplifican imágenes conservando las principales características de forma de los objetos. Un sistema de operadores como los definidos por la morfología matemática es útil porque pueden formar composiciones de sus operadores, que cuando actúan sobre formas complejas son capaces de descomponerlas en sus partes más significativas y separarlas de aquellas que le son extrañas, permiten que las formas subyacentes sean identificadas y reconstruidas de forma óptima a partir de sus formas distorsionadas o ruidosas, además permite que cada forma se entienda en función de una descomposición, siendo cada entidad de esa descomposición una forma simple apropiada.

Las operaciones morfológicas pueden simplificar los datos de una imagen, preservar las características esenciales y eliminar aspectos irrelevantes.

Teniendo en cuenta que la identificación y descomposición de objetos, la extracción de rasgos, la localización de defectos (en líneas de ensamblaje por ejemplo) está sumamente relacionado con las formas, se evidencia la necesidad de incluir morfología matemática dentro de proyectos de visión artificial.

Algunos de los objetivos que pueden ser abordados con este tipo de métodos son:

- Pre-procesamiento de imágenes: Supresión de ruido, simplificación de formas.
- Destacar las estructuras de los objetos: Extracción del esqueleto, marcado de objetos, envolvente convexa, ampliación, reducción de características.

- Descripción cualitativa de objetos: Área, perímetro, entre otros.

El lenguaje utilizado en la morfología matemática es el mismo que el de la teoría de conjuntos, los conjuntos en morfología matemática representan las formas presentes en una imagen binaria o de niveles de gris, por ejemplo el conjunto de todos los píxeles blancos de una imagen en blanco y negro (binaria) constituye una descripción completa de la imagen.

Las operaciones primarias morfológicas son la erosión y la dilatación, a partir de ellas podemos componer otras operaciones como la apertura y clausura. Estas dos operaciones primarias tienen mucha relación con la representación de formas, descomposición y extracción de primitivas.

Dilatación: Es la transformación morfológica que combina dos vectores utilizando la suma, la dilatación de un conjunto X por un elemento estructurante B es la unión de las traslaciones de X según B [2]:

$$\delta_B(X) = X \oplus B = \bigcup_{b \in B} X_b = \bigcup_{x \in X} B_x = \{x + b \mid x \in X, b \in B\}$$

Formula 2.2.1.9 - Dilatación

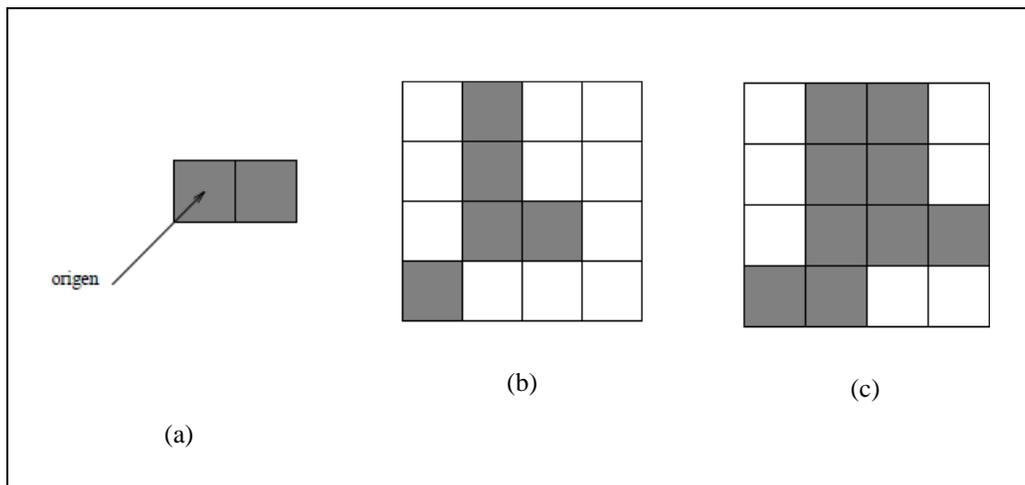


Figura 2.2.1.10 - Ejemplo Dilatación (a) Elemento estructurante (b) Imagen original (c) Imagen dilatada.

Erosión: Es la operación morfológica dual, es una transformación que combina dos conjuntos usando el concepto de inclusión. Si A y B son conjuntos en el espacio euclideo n-dimensional, entonces la erosión de A por B es el conjunto de todos los elementos x para los que x + b pertenece a A para todo b perteneciente a B.

La erosión de A por B se denota:

$$A \ominus B = \{x \in E^n \mid x + b \in A \forall b \in B\}$$

Formula 2.2.1.10 - Erosión

La erosión se concibe usualmente como una reducción de la imagen original. En términos de teoría de conjuntos, el conjunto erosionado se suele pensar que siempre está contenido en el original, a continuación un ejemplo [2]:

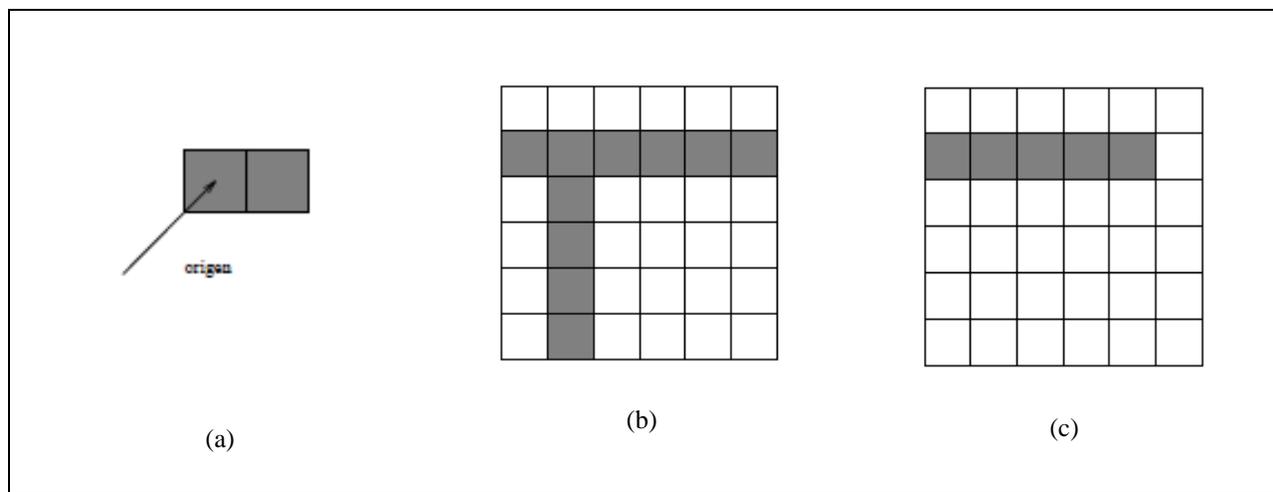


Figura 2.2.1.11 - Ejemplo erosión (a) Elemento estructurante (b) Imagen original (c) Imagen erosionada.

Uno de los inconvenientes que presentan estos métodos es el tiempo de procesamiento necesario dado el número de operaciones que se realizan sobre los píxeles de una imagen, se debe evaluar su incorporación en un proceso de procesamiento de imágenes digitales de acuerdo a estas características.

Técnicas de Representación y Descripción.

Cuando se ha realizado la segmentación de una imagen y obtenido los resultados esperados, una siguiente actividad debiera incluir el representar y describir los elementos encontrados de manera de optimizar las entradas de algún posterior algoritmo de clasificación a emplear. Con esta finalidad existen métodos que consideran diferentes características de los objetos presentes en una imagen, a continuación se presentan algunos de estos algoritmos:

Cadenas de códigos: Las cadenas de códigos se utilizan para representar bordes mediante una sucesión conectada de segmentos de líneas rectas de una longitud y dirección dada, normalmente esta representación se basa en la 4 u 8 conectividad [2]. La dirección de cada segmento se codifica mediante un esquema de numeración previamente establecido, un ejemplo se puede ver en la siguiente figura [2]:

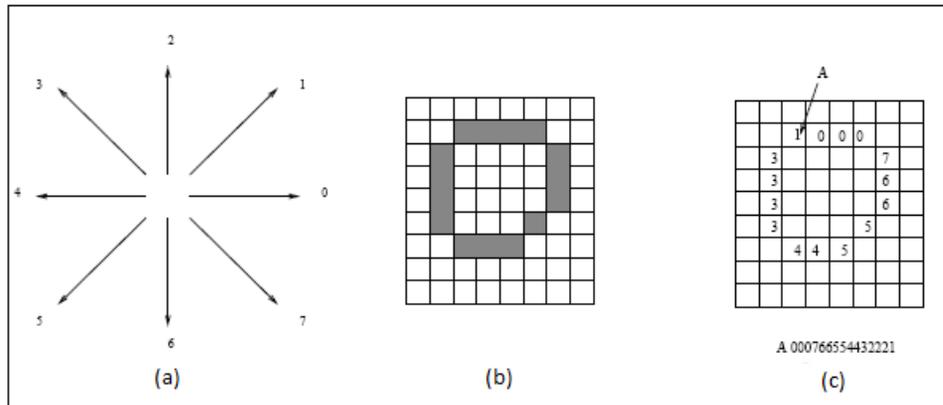


Figura 2.2.1.12 - Cadena de Códigos (a) 8 Conectividad, (b) Frontera, (c) Codificación según 8 conectividad.

Descriptores de Fourier: Una figura cerrada puede ser representada mediante series de Fourier con una parametrización adecuada. Este método realiza una transformación desde la representación en dos variables de los píxeles de una imagen (fila, columna) hacia el cálculo del número complejo que representa la misma ubicación, logrando traspasar el problema a una sola variable (frecuencia). Los descriptores de Fourier tienen como característica ser invariantes a transformaciones geométricas y tolerantes ante el ruido, pero ya que este método deriva del cálculo de la transformada de Fourier de una imagen y que por lo tanto trabaja con el dominio de las frecuencias, se detallará su aplicación en la correspondiente sección de este documento (considerando que esta sección estudia principalmente técnicas del dominio del espacio).

Descriptores Simples: La *longitud* de una figura es uno de los descriptores más sencillos, sólo bastaría con contar el número de píxeles pertenecientes al contorno del objeto. La *curvatura* se define como la razón de cambio de la pendiente [2]. Cuando una frontera es recorrida en la dirección de las agujas del reloj, un vértice p se dice parte de un segmento *convexo* si el cambio de la pendiente en p es no-negativo; en caso contrario se dice *cóncavo*. Otros descriptores simples son el *diámetro* y *números de forma*.

Técnicas de Reconocimiento de Objetos.

El proceso de reconocimiento o clasificación de objetos es una de las etapas de mayor dificultad dentro del área de procesamiento de imágenes digitales, la elección de una técnica tiene relación con el ámbito del problema, la naturaleza de la imagen y los objetos a identificar, ya que dependiendo de la forma geométrica (líneas, circunferencias, elipses) existen algoritmos que entregan mejores prestaciones que otros. Además muchas veces el objeto en la escena no tiene una forma determinada y por lo tanto se hacen necesarias complementar con otras técnicas que aborden esta problemática.

A continuación se estudian las principales técnicas de reconocimiento de objeto que existen:

Reconocimiento de objetos a través de su contorno.

Este algoritmo funciona en una imagen con objetos de color negro y un fondo blanco (o viceversa), el procedimiento que realiza corresponde a un barrido de la imagen hasta encontrar un punto de un objeto (siempre será el punto superior izquierdo del objeto), una vez encontrado comienza a recorrer el borde del objeto en sentido a favor de las manecillas del reloj, se van guardando los puntos encontrados en un array (contorno), al concluir borra de la imagen el objeto y su interior para seguir buscando otros objetos. La siguiente figura gráfica el algoritmo descrito:

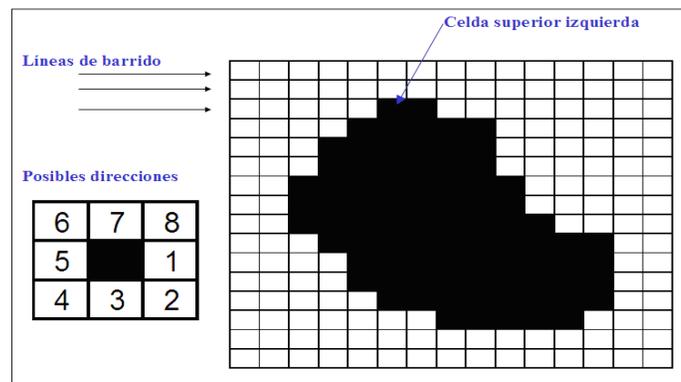


Figura 2.2.1.13 - Reconocimiento de objeto a través de contorno.

Reconocimiento de objetos orientado a regiones.

El objetivo de los algoritmos basados en la técnica de segmentación que se describió anteriormente (contorno) es construir los objetos directamente por las regiones que ocupan dentro de la escena, los algoritmos que buscan encontrar regiones de objetos implementan un recorrido de los píxeles en la imagen agrupando aquellos que mantienen algún criterio de homogeneidad.

Transformada de Hough.

La aplicación de la transformada de Hough en conjunto con otras técnicas de reconocimiento de objetos representa una poderosa herramienta cuando los elementos presentes en la imagen mantienen una estructura geométrica bien definida, es útil en problemas de reconocimiento de líneas, circunferencias, elipses y otros, ya que como se describió en paginas anteriores se basa en la identificación de la ecuación que representa estas formas geométricas.

Correlación cruzada.

La correlación cruzada (Cross Correlation) es una técnica frecuentemente utilizada para el reconocimiento de patrones, este método se basa en una serie de multiplicaciones, su ecuación es la siguiente:

$$C(i, j) = \sum_{x=0}^{L-1} \sum_{y=0}^{K-1} w(x, y) f(x+i, y+j)$$

Formula 2.2.1.11 - Correlación cruzada.

Donde para una sub-imagen $w(x, y)$ de tamaño $K \times L$ dentro de la imagen $f(x, y)$ de tamaño $M \times N$ en donde $K \leq M$ y $L \leq N$, la correlación entre $w(x, y)$ y $f(x, y)$ es igual a $C(i, j)$:

De manera grafica el proceso de correlación seria:

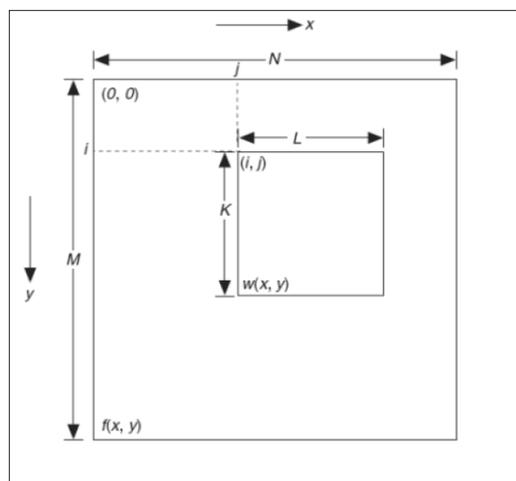


Figura 2.2.1.14 – Correlación cruzada.

La sub-imagen se recorre por toda la región de búsqueda. El máximo valor de $C(i,j)$ indica la posición que mejor se ajusta a la imagen patrón. Se puede reducir el tiempo si se reduce el tamaño de la imagen patrón o el tamaño de búsqueda dentro de la imagen. [6]

La presente sección ha presentado técnicas de procesamiento de imágenes digitales que trabajan fundamentalmente sobre las tonalidades de los píxeles, estos algoritmos representan conceptos introductorios que permiten adquirir rápidamente conocimientos del área. Sin embargo, estas técnicas están limitadas entre otras cosas, por factores tales como el ruido, cambios geométricos en los objetos y el tiempo de procesamiento requerido para analizar todos los puntos de una imagen, lo que genera la necesidad de complementar el procesamiento con técnicas de mayor robustez que ayuden a alcanzar resultados con un menor grado de dependencia de factores como los ya mencionados.

En las siguientes páginas se presentan técnicas que trabajan sobre el dominio de la frecuencia como los descriptores de Fourier, más adelante en este mismo capítulo se abordan métodos de clasificación basados en redes neuronales artificiales.

2.2.2 Procesamiento de Imágenes Digitales - Dominio de las Frecuencias.

Una imagen digital puede ser representada mediante una función discreta de dos variables (filas, columnas de la matriz de píxeles) donde el dominio de estas es un conjunto finito o numerable (conjunto discreto). Las transformaciones matemáticas permiten obtener desde una señal mayor información de la que se podría extraer desde la señal pura [8], estas transformaciones permiten descomponer una señal en sus componentes sinusoidales de diferentes frecuencias, en otras palabras, es una técnica matemática que permite transformar una señal basada en el tiempo a un punto de vista basado en la frecuencia.

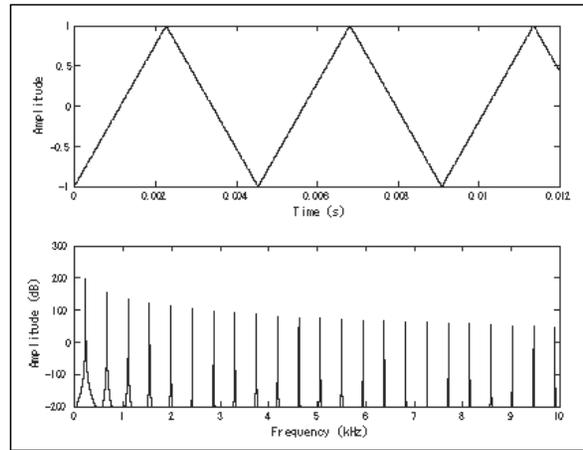


Figura 2.2.2.1 - Dominio del tiempo (gráfico superior) y dominio de la frecuencia (gráfico inferior).

Transformada de Fourier

En esta sección se estudia el método de transformación propuesto en documento “La théorie Analytique de la Chaleur (teoría analítica del calor)” del matemático Jean Baptiste Joseph Fourier.

Uno de los postulados más importantes del documento de Fourier consiste en la idea de que toda función periódica puede ser expresada como la suma de senos y/o cósenos de diferentes frecuencias, cada uno multiplicado por un coeficiente diferente, esta suma corresponde a la denominada “Serie de Fourier”, pero también funciones que no son periódicas (pero con un área finita bajo la curva) pueden ser expresadas como la integral de senos y/o cosenos multiplicada por una función de ponderación. Esta es la idea base de la transformada de Fourier.

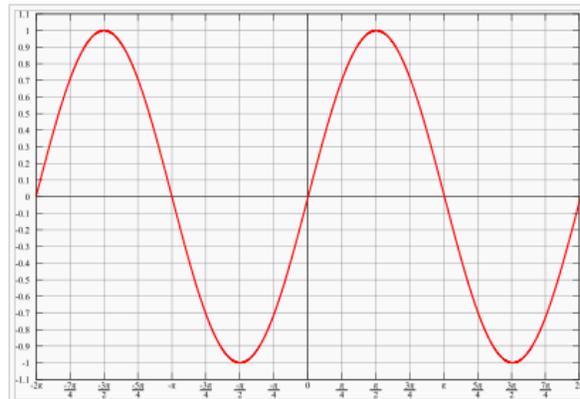


Figura 2.2.2.2 - Descomposición de una onda en una suma de funciones senoidales y cosenoidales.

Las representaciones descritas comparten una importante propiedad: Una función expresada en series de Fourier o en su transformada puede ser reconstruida por un proceso inverso sin perder su información.

La transformada de Fourier $F(u)$ de una función con una sola variable $f(x)$, se define con la ecuación:

$$F(u) = \int_{-\infty}^{\infty} f(x)e^{-j2\pi ux} dx$$

Formula 2.2.2.1 – Transformada de Fourier de una función con una sola variable.

Donde $j = \sqrt{-1}$. De manera correspondiente, dada $F(u)$ podemos obtener $f(x)$ por medio de la transformada de Fourier inversa:

$$f(x) = \int_{-\infty}^{\infty} F(u)e^{j2\pi ux} du$$

Formula 2.2.2.2 – Transformada inversa de Fourier de una función con una sola variable.

Estas dos ecuaciones anteriores representan un par de transformadas de Fourier, lo que indica que una función puede ser recuperada a partir de su transformada. Es posible también representar la transformada de Fourier de una función de dos variables:

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y)e^{-j2\pi(ux+vy)} dx dy$$

Formula 2.2.2.3 – Transformada de Fourier de una función con dos variables.

Y su correspondiente inversa:

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v)e^{j2\pi(ux+vy)} dudv$$

Formula 2.2.2.4 – Transformada inversa de Fourier de una función con dos variables.

Transformada Discreta de Fourier Unidimensional

Las imágenes digitales son funciones discretas de dos variables (filas, columnas) donde el dominio de estas es un conjunto finito o numerable (conjunto discreto).

La transformada de Fourier de una función discreta de una variable $f(x)$, cuando $x=0, 1, 2, \dots, M-1$ está dada por la siguiente ecuación:

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

Formula 2.2.2.5 – Transformada Discreta de Fourier de una sola variable.

La anterior es conocida como Transformada Discreta de Fourier (DFT, *por sus siglas en ingles Discrete Fourier transform*) e igualmente podemos obtener la función original utilizando la DFT inversa.

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

Formula 2.2.2.6 – Inversa de la Transformada de Fourier Discreta de una sola variable.

Para obtener el $F(u)$ de la DFT, comenzamos por sustituir $u=0$ en el termino exponencial y después sumamos para todos los valores de x , después se debe sustituir $u=1$ en el exponencial y repetir la suma para todos los valores de x . Se repite este proceso para los M valores de u y de esta manera obtenemos la DFT.

Como $f(x)$ es la transformada de una cantidad discreta y tiene el mismo número de componentes que $F(x)$ se aplica un proceso similar para obtener la DFT inversa.

El Dominio de la Frecuencia

El dominio de la frecuencia es obtenido desde la fórmula de Euler:

$$e^{j\mathcal{G}} = \cos \mathcal{G} + j \operatorname{sen} \mathcal{G}$$

Formula 2.2.2.7 - Formula de Euler.

Sustituyendo en la formula 2.2.2.5 (DFT de una sola variable) y teniendo en cuenta que $\cos -\mathcal{G} = \cos \mathcal{G}$ y $\operatorname{sen} -\mathcal{G} = -\operatorname{sen} \mathcal{G}$, obtenemos:

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

Formula 2.2.2.8 - DFT de una sola variable según formula de Euler.

Para $u=0,1,2,3, \dots, M-1$. Así, observamos que cada término de la DFT (es decir, el valor de $F(u)$ para cada valor de u) se compone de la suma de todos los valores de la función $f(x)$. Y a su vez los valores de $f(x)$ son multiplicados por senos y cosenos en varias frecuencias.

El dominio (valores de u) donde el rango son valores de $F(u)$ es llamado dominio de la frecuencia, porque u determina la frecuencia de los componentes de la transformada.

Cada uno de los M términos de $F(u)$ se llama componente de frecuencia de la transformada o *Descriptores de Fourier*.

Los términos dominio de la frecuencia y componentes de frecuencia son equivalentes al dominio del tiempo y componentes del tiempo que se usan para expresar el dominio y valores de $f(x)$ si x fuera una variable de tiempo.

De acuerdo con las ecuaciones anteriores, los componentes de la DFT son cantidades complejas, pero a veces será conveniente manejar $F(u)$ como coordenadas polares:

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

Formula 2.2.2.9 DFT unidimensional expresada como coordenadas polares.

Donde

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

Formula 2.2.2.10 Magnitud o espectro de la DFT unidimensional.

Es llamado *magnitud o espectro* de la DFT y

$$\phi(u) = \tan^{-1}[I(u) / R(u)]$$

Formula 2.2.2.11 Ángulo de fase o espectro de fase de la DFT unidimensional.

Es llamado *ángulo de fase o espectro de fase* de la transformada. En las formulas anteriores $R(u)$ e $I(u)$ son las partes real e imaginaria de $F(u)$, respectivamente.

La *densidad espectral* se corresponde con el cuadrado del espectro de Fourier:

$$P(u) = |F(u)|^2 = R^2(u) + I^2(u)$$

Formula 2.2.2.12 Densidad espectral de la DFT unidimensional.

Las propiedades del espectro son consideradas en algoritmos que buscan la mejora de una imagen. En la siguiente figura se muestra una función discreta $f(x)$ de M puntos, con K puntos con valor A , y su transformada (grafica superior). Luego se muestra una segunda función discreta $f(x)$ con el doble de puntos con valor A y su correspondiente transformada (grafica inferior [8]).

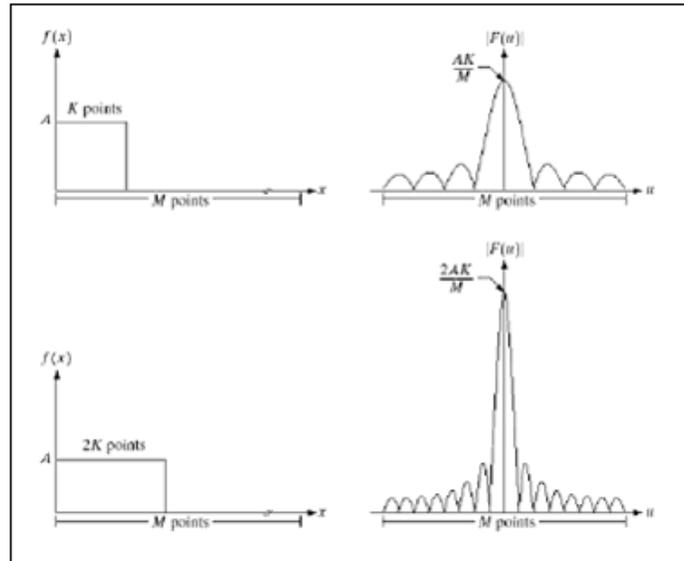


Figura 2.2.2.3 - Ejemplos de funciones discretas y sus respectivas transformadas.

Transformada Discreta de Fourier Bidimensional

Dado que las imágenes digitales son funciones discretas de dos variables, la transformada discreta de Fourier puede ser representada para dos dimensiones de la siguiente forma:

$$F(u, v) = 1/MN \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)} \quad (u=0, 1, \dots, M-1; v=0, 1, \dots, N-1)$$

Formula 2.2.2.13 – DFT de dos variables.

Y su inversa:

$$f(x, y) = 1/MN \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} F(u, v) e^{j2\pi(ux/M + vy/N)} \quad (x=0, 1, \dots, M-1; y=0, 1, \dots, N-1)$$

Formula 2.2.2.14 – DFT inversa de dos variables.

Donde “ u ” y “ v ” son las variables de transferencia o frecuencia y “ x ” e “ y ” son las variables espaciales o de imagen. El espectro de Fourier, ángulo de fase y espectro de frecuencia se deducen también con facilidad:

$$|F(u, v)| = [R^2(u, v) + I^2(u, v)]^{1/2}$$

Formula 2.2.2.15 Magnitud o espectro de la DFT bidimensional.

$$\phi(u, v) = \tan^{-1}[I(u, v) / R(u, v)]$$

Formula 2.2.2.16 Ángulo de fase o espectro de fase de la DFT bidimensional.

$$P(u) = |F(u)|^2 = R^2(u) + I^2(u)$$

Formula 2.2.2.17 Densidad espectral de la DFT bidimensional.

Usualmente una imagen digital se multiplica por $(-1)^{x+y}$, esto resulta en:

$$\mathcal{F}[f(x, y)(-1)^{x+y}] = F(u - M/2, v - N/2)$$

Formula 2.2.2.18 Origen de la DFT centrado (rectángulo de frecuencia)

Con esta ecuación podemos darnos cuenta que el origen de la DFT de $f(x, y)(-1)^{x+y}$ se localiza en $u = M/2, v = N/2$ lo que pone el origen al centro del área M/N que ocupa la DFT bidimensional, esta área es llamada rectángulo de frecuencia. Se requiere que M y N sean impares para que las coordenadas del centro sean valores enteros.

El valor de la transformada en $(u, v) = (0, 0)$ es:

$$F(0, 0) = 1/MN \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)$$

Formula 2.2.2.19 - Valor de la DFT en el origen.

Lo que claramente es el promedio de $f(x,y)$, lo que quiere decir que el valor de la DFT en el origen es igual al nivel de gris promedio de la imagen. Como las frecuencias son cero en el origen, $F(0,0)$ a veces es llamado el componente *dc* (de corriente directa, cero frecuencia).

Además el espectro de la DFT es simétrico

$$|F(u, v)| = |F(-u, -v)|$$

Formula 2.2.2.20 – Simetría del espectro de la DFT bidimensional.

También se cumplen las siguientes relaciones:

$$\Delta u = 1 / M\Delta x$$

$$\Delta v = 1 / N\Delta y$$

Formula 2.2.2.21 – Relación variables planos espacio/frecuencias.

Transformada Rápida de Fourier

Dada la complejidad computacional que representa la ejecución de una DFT existe una técnica denominada transformada rápida de Fourier (FFT, Fast Fourier Transform.) que permite obtener los mismos resultados pero realizando un menor número de operaciones, debido a la propiedad de separabilidad de la transformada discreta de Fourier bidimensional, el cálculo de la FFT se puede realizar haciendo dos DFT unidimensionales.

La razón principal del uso de la FFT es el ahorro en el número de operaciones a realizar con respecto a la DFT, para el cálculo directo de una DFT bidimensional se necesitan $N^2 \times N^2 = N^4$ operaciones para una imagen de dimensión $N \times N$, mientras que para esta misma imagen y mediante el uso de una FFT sólo se necesitan $N^2 \log_2(N)$.

Existen dos formas de representar la FFT de una imagen: Representación Estándar y la Representación Óptica.

Representación Estándar de una FFT: En la representación estándar las frecuencias altas son agrupadas en el centro de la imagen mientras que las bajas frecuencias están en los bordes, los términos constantes o de frecuencia nula están en la esquina superior izquierda de la imagen. El rango de la frecuencia es:

$$[(0, N)] \times [(0, M)]$$

Formula 2.2.2.22 – Rango de Frecuencia representación estándar de una FFT

Donde M es la resolución horizontal de la imagen y N es la resolución vertical.

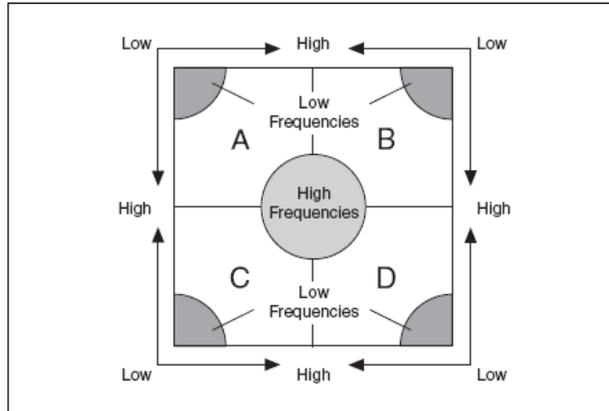


Figura 2.2.2.4 - Representación Estándar de una FFT

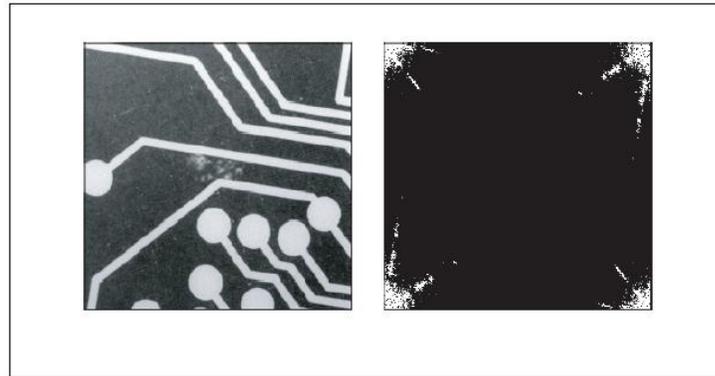


Figura 2.2.2.5 – Imagen original (imagen de la izquierda) y su FFT con representación estándar (imagen de la derecha).

Representación Óptica de una FFT: En la representación óptica, las bajas frecuencias son agrupadas en el centro de la imagen mientras que las altas frecuencias están en los bordes. Los términos constantes o de frecuencia nula están en el centro de la imagen. Los rangos de frecuencia son:

$$[-N/2, N/2) \times [-M/2, M/2)$$

Formula 2.2.2.23 – Rango de frecuencia representación óptica de una FFT.

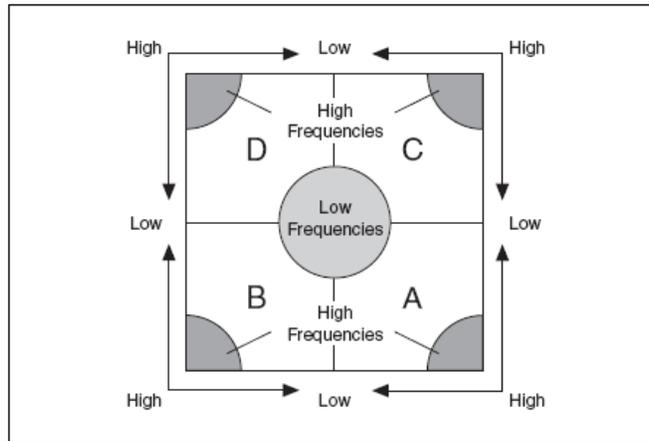


Figura 2.2.2.6 - Representación óptica de una FFT.

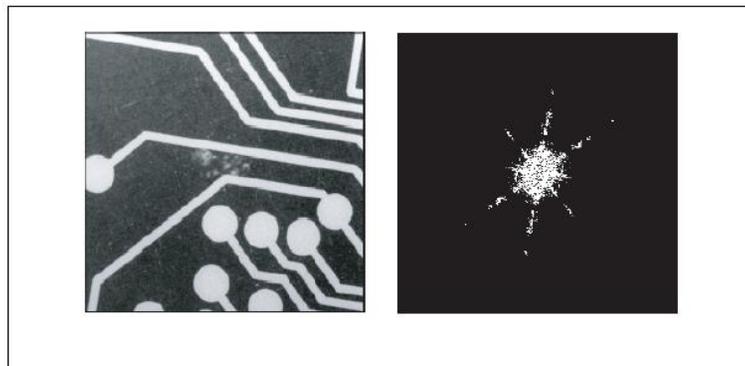


Figura 2.2.2.7 – Imagen original (imagen de la izquierda) y su FFT con representación óptica (imagen de la derecha).

Técnicas que trabajan sobre el dominio de la Frecuencia (Fourier)

De acuerdo a lo estudiado anteriormente, se puede concluir que el dominio de la frecuencia no es más que el espacio definido por los valores de la transformada de Fourier y sus variables de frecuencia (u, v). Donde que cada uno de sus descriptores $F(u, v)$ contiene todos los valores de $f(x, y)$, modificados por los valores de los términos exponenciales.

Debido a lo anterior, usualmente es imposible hacer asociaciones directas entre componentes específicas de una imagen y su transformada, sin embargo se pueden hacer algunas observaciones generales acerca de la relación entre los componentes de la frecuencia de la transformada de Fourier y las características especiales de una imagen. Como ejemplo tenemos que la frecuencia se relaciona directamente con la velocidad de cambio, lo que permite asociar las frecuencias de la transformada de Fourier con patrones de variación de la imagen.

Varios detalles de la imagen pueden relacionarse con lo que vemos en la representación óptica de la FFT, como por ejemplo:

- El componente de la frecuencia que varía más lentamente ($u=v=0$) corresponde al nivel de gris promedio.
- Al alejarnos del origen, las frecuencias bajas corresponden a componentes que varían lentamente.
- Ya lejos del origen, las altas frecuencias corresponden a cambios cada vez más rápidos en el nivel de gris de la imagen (p.e. Bordes, o Ruido)

En la figura 2.2.2.8 se presenta una imagen y su respectivo espectro de Fourier, donde en la parte izquierda se puede ver un circuito dañado (la parte blanca es oxido producido por el daño termal) y en el espectro de Fourier (parte derecha), los bordes prominentes corriendo a 45° corresponden a las 2 prominencias blancas del oxido presente en la imagen original [8].

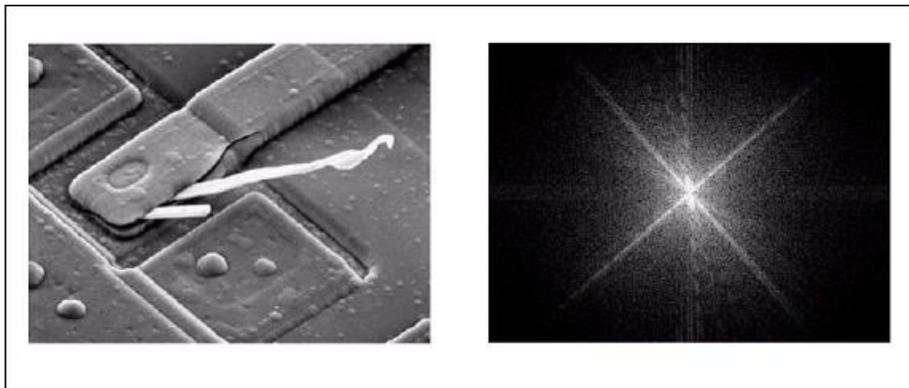


Figura 2.2.2.8 - Espectro de la FFT de una imagen con un circuito dañado.

Filtrado del ruido

El análisis de la transformada de Fourier es utilizado para varios objetivos, un ejemplo es el filtrado del ruido presente en una imagen basado en la supresión de altas y/o bajas frecuencias (Lowpass Attenuation, Highpass Filters y Bandpass Filters), lo que se traduce en el suavizado de la imagen, realce o detección de bordes.

Los pasos a seguir son:

1. Se multiplica cada entrada $f(x, y)$ por $(-1)^{x+y}$.
2. Se transforma la imagen al dominio de la frecuencia mediante la DFT, $F(u, v)$

- Se multiplica por un *filtro de frecuencia* $H(u, v)$, para cada (u, v) :

$$G(u, v) = H(u, v)F(u, v)$$

Formula 2.2.2.24 – Filtro de frecuencia utilizado en la supresión de ruido.

- Se calcula la inversa de la TDF de $G(u, v)$ (tomando la parte real) volviendo así al dominio del espacio.
- Se vuelve a multiplicar por $(-1)^{x+y}$

El proceso anterior se grafica en la siguiente figura [8]:

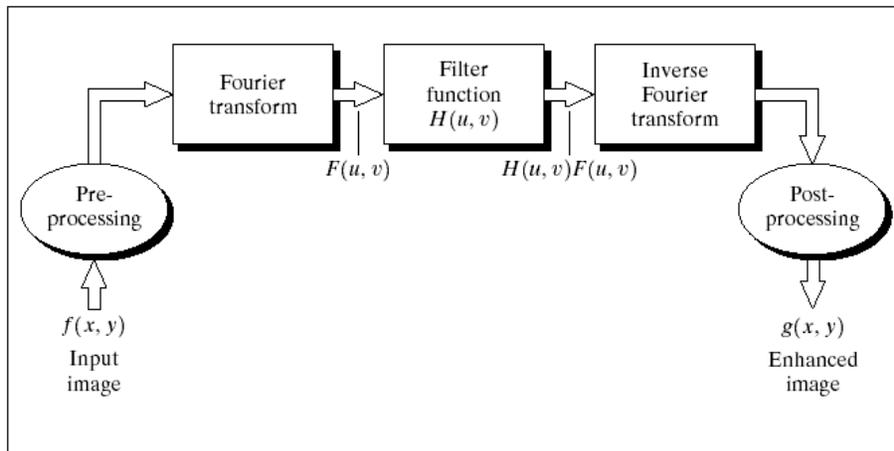


Figura 2.2.2.9 - Utilización de la transformada de Fourier en la supresión de ruido.

Descriptores de Fourier

Otro de los usos que tiene la transformada de Fourier en el procesamiento de imágenes digitales corresponde a la descripción de los objetos presentes en una escena, mediante el cálculo de la transformada del conjunto de puntos de bores que representan su contorno se logra describir la forma del elemento de manera independiente del escalado, la rotación y la traslación que pudiera sufrir el mismo.

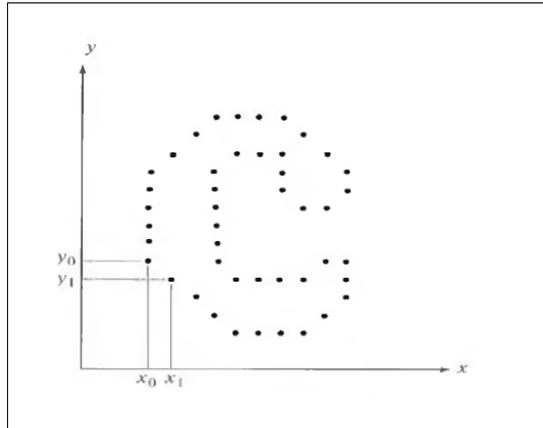


Figura 2.2.2.10 –Puntos del contorno de un objeto.

Para la función $z(n)$ que representa el conjunto de puntos del contorno, con $n=0, \dots, N$ (es decir tenemos N puntos de la frontera), la DFT viene dada por:

$$FD_k = u_k = \frac{1}{N} \sum_{n=0}^{N-1} z(n) * e^{\left(\frac{-j2\pi kn}{N}\right)}$$

Formula 2.2.2.25 – Descriptores de Fourier de un conjunto de puntos de borde $z(n)$.

Los coeficientes U_k son conocidos como los descriptores de Fourier (DF) del contorno, con $k=0, \dots, N-1$.

La invarianza a transformaciones es difícil de alcanzar en el dominio del espacio, muchas técnicas, especialmente las de invarianza a rotación, implican un coste computacional elevado, sin embargo la invarianza a escalado, rotación y traslación es sencilla de conseguir cuando se trabaja con los DF:

- Invariantes a Traslación: la función 1-D elegida para representar la frontera es por sí sola invariante a traslación, ya que utiliza información acerca del centroide.
- Invariantes a Rotación: se consigue ignorando la información de la fase. La Transformada Rápida de Fourier nos devuelve los descriptores como números complejos, para eliminar la fase simplemente deberemos quedarnos con los valores absolutos de los FD.
- Invariantes a Escalado: para obtener esta propiedad simplemente dividiremos cada uno de los descriptores por el primer descriptor calculado (DF0).

Debido a la eliminación de la información de fase, sólo quedan valores reales, con lo que sólo se tendrán N/2 frecuencias diferentes en la transformada de Fourier, es decir, solo será necesaria la primera mitad de los DF. Finalmente se tendrá el siguiente conjunto de descriptores normalizados:

$$FD_vector = \left[\frac{|FD_1|}{|FD_0|}, \frac{|FD_2|}{|FD_0|}, \dots, \frac{|FD_{N/2}|}{|FD_0|} \right]$$

Formula 2.2.2.26 – Vector descriptores de Fourier normalizados.

Los DF tienen la característica de reducir el número de descriptores a utilizar para describir un objeto sin perder su forma original. Mientras más descriptores tengamos, más aproximada a la imagen original será la imagen que podamos recuperar, sin embargo sólo un pequeño número de descriptores en ocasiones puede ser suficiente.

Desventajas de la Transformada de Fourier

Pese a las ventajas que ofrece la transformada de Fourier, este método también tiene desventajas como la pérdida de información cuando se realiza un traspaso al dominio de las frecuencias (respecto al tiempo) [9]. En la observación de una señal que ha sido transformada utilizando Fourier es imposible determinar *cuando* ocurre un determinado evento o cuando se presenta una determinada frecuencia, en los casos en que se trabaja sobre una señal estacionaria (frecuencias repetitivas como en las señales periódicas) lo anterior no tiene mayor influencia, sin embargo en casos donde las señales tienen características no estacionarias o transitorias se descarta el uso de esta transformada.

Para corregir la deficiencia expuesta en el párrafo anterior, en 1946 Denis Gabor adaptó la Transformada de Fourier para poder analizar una pequeña sección de la señal en un determinado tiempo (utilizando una especie de ventana). Esta adopción es la que se conoce como Transformada de Fourier de Tiempo Reducido (STFT, del inglés short-time Fourier transform), en donde se lleva la señal del plano del tiempo al plano bidimensional de tiempo y frecuencia, esto es visualizado en la siguiente figura:

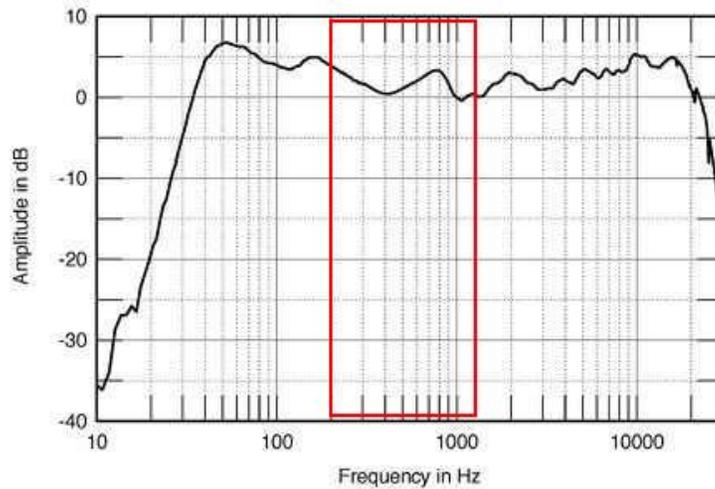


Figura 2.2.2.11 - Adaptación de la transformada de Fourier propuesta por Gabor.

STFT representa una especie de compromiso entre el dominio del tiempo y el de la frecuencia de una señal, ya que provee algo de información acerca de cuándo y en qué frecuencia de una señal ocurre un determinado evento. Sin embargo, solamente se puede obtener dicha información con una precisión limitada, la cual está acotada por el tamaño de la ventana. La desventaja que se presenta frente a esta mejora de la transformada de Fourier es que una vez seleccionado el tamaño de la ventana a utilizar este no podrá ser cambiado, lo que impide un análisis robusto de la señal en los casos que requieran un acercamiento más flexible (mayor precisión del tiempo y la frecuencia).

Lo anterior se puede observar debido a las características que poseen las señales, en donde en las altas frecuencias el número de oscilaciones en el dominio temporal de las ventanas es elevado, y la información de frecuencias del mapeo es nítida. En contraposición, en las bajas frecuencias, las oscilaciones son relativamente largas y no pueden caracterizarse apropiadamente.

La implementación de la transformada de Gabor para el procesamiento de señales entonces resulta eficiente cuando se trata de localizar y caracterizar eventos con patrones de frecuencia bien definida, no superpuestos y relativamente largos respecto de la ventana de análisis. En contraposición, es totalmente inapropiada para detectar detalles de corta duración, oscilaciones largas asociados a las bajas frecuencias, o caracterizar patrones auto-similares presentes en fenómenos a distintas escalas (en donde transformadas como la de Wavelet surgen para suplir estas deficiencias).

2.2.3 Visión Artificial.

En esta sección se estudian algoritmos que buscan mejorar los resultados obtenidos por las técnicas que trabajan sobre el dominio del espacio (vistas en sección anterior). En general los algoritmos del área de visión artificial tienen como objetivo fundamental emular las características del sistema visual humano, y gracias al avance de los estudios realizados en el área se han logrado importantes avances en relación al descubrimiento de técnicas y procesos que permiten obtener un mayor grado de información desde una escena bidimensional.

El problema del reconocimiento visual de objetos es generalmente sencillo para las personas, pero se ha comprobado que es muy complejo para los computadores, un ejemplo es cuando se quiere lograr la identificación del rostro de una persona a pesar de las variaciones de iluminación, postura respecto a la cámara y las distintas expresiones faciales que se pueden dar. Cualquier cambio en alguno de estos elementos causa diferencias de los valores de brillo y tonalidad en los píxeles, por lo que es poco probable realizar una sencilla comparación de estos valores para recuperar la información necesaria.

Cuando la cámara se mueve con respecto a un objeto, la distancia a este y su posición cambian. Lo que no sufre alteraciones es la forma del objeto. Si el objeto es un cubo, este hecho no cambia cuando el objeto se mueve. Las geometrías han intentado formalizar la forma de los objetos desde hace décadas, y el concepto básico es que la forma es lo que permanece sin cambios bajo un cierto grupo de transformaciones, como por ejemplo una combinación de rotaciones y traslaciones. La dificultad está en encontrar una representación de la forma global del objeto que sea lo suficientemente general como para adaptarse a la amplia variedad de objetos del mundo real (considerando también otras formas más complejas que los cubos, cilindros o esferas.) y que además pueda obtenerse de forma sencilla a partir de una entrada visual.

Para lograr introducir los diversos conceptos involucrados en un problema complejo de visión artificial típico, a continuación se darán breves definiciones de los elementos que deben ser considerados para recuperar información tridimensional de un objeto desde una escena. Estos elementos son el movimiento, la estereoscopia binocular, la textura, el sombreado y el contorno [7], se describen a continuación:

Movimiento: Si la cámara se mueve respecto a la escena tridimensional, el movimiento aparente resultante en la imagen se llama *flujo óptico*, este describe la dirección y la velocidad de movimiento, de las características de la imagen como resultado del movimiento relativo entre el observador y la escena.

Estereoscopia binocular: La idea es similar a la de paralaje por movimiento (diferencia entre las posiciones de los objetos en el tiempo), excepto que en vez de utilizar el movimiento a lo largo del tiempo, utilizamos dos (o más) imágenes separadas en el espacio, tal y como las proporcionan los ojos de los humanos. Puesto que las

diferencias en las posiciones de las cámaras producen vistas bidimensionales de la misma escena tridimensional ligeramente distintas.

Gradientes de textura: La textura en una definición simple es la propiedad de las superficies asociada con la calidad táctil que sugieren. En visión artificial se refiere a un concepto relacionado, que es el de un patrón repetido espacialmente sobre una superficie y que puede captarse visualmente. Mediante algunos cálculos matemáticos, podemos calcular expresiones para la proporción de cambio de varias características de los elementos de la textura (texels) de la imagen, tales como el área, escorzo y densidad. Estos *gradientes de textura* son funciones de la forma de la superficie, así como de su inclinación respecto a la posición del observador.

Sombreado: El sombreado corresponde a la variación en la intensidad de la luz recibida desde diferentes posiciones de una superficie en una escena y viene determinado por la geometría de la escena y las propiedades reflectantes de las superficies. En gráficos por computador el objetivo es calcular el brillo de la imagen $I(x,y)$, dada la geometría de la escena y las propiedades reflectantes de los objetos en la escena. La visión artificial tiene como objetivo invertir el proceso, esto es, recuperar la geometría y las propiedades reflectantes, dado el brillo de la imagen $I(x,y)$. Se ha comprobado que este proceso es muy difícil de realizar más allá de los casos más sencillos.

Contorno: Considerando el conocimiento cualitativo disponible de un dibujo de líneas. Las líneas de un dibujo pueden tener múltiples significados. La tarea de evaluación del verdadero significado de cada línea se denomina *etiquetado de líneas*. Cada línea puede clasificarse como la proyección de una *línea de silueta* (lugar geométrico de puntos de la superficie donde la línea de vista es tangente a la superficie) o como un *borde* (discontinuidad observada en una superficie, cada borde puede clasificarse como convexo, cóncavo y de oclusión).

Podemos concluir que en un problema de visión artificial típico existen múltiples conceptos que deben ser considerados, y para los cuales existen importantes definiciones y algoritmos asociados a cada uno. Esta tesis se encuentra enmarcada en la búsqueda de mejorar los resultados obtenidos en el reconocimiento de objetos con técnicas de procesamiento de imágenes enfocadas en las tonalidades de los píxeles presentes en una imagen (dominio del espacio), para lo que se abarcarán técnicas de alto nivel que hacen uso del análisis multiescalar (dominio de la frecuencia) y del reconocimiento de patrones, analizando e implementando para este fin la transformada de Fourier y una red neuronal artificial en la cual se logre, gracias al aprendizaje supervisado, realizar una correcta clasificación de objetos.

2.2.4 Visión Artificial - Redes Neuronales Artificiales

Desde sus comienzos la Inteligencia Artificial (IA) ha buscado simular las características propias de los seres humanos, el campo de las redes neuronales artificiales (RNA) es uno más con aquel interés pero la diferencia radica que mientras la IA comenzó utilizando como base la estructura tradicional de los ordenadores, en donde basados en la creación de algoritmos escritos en un lenguaje de programación entendible por la maquina y que contienen la lógica necesaria para realizar diversas operaciones (considerando además el espacio en memoria que requerirá el programa y la cantidad de procesadores utilizados para ejecutarse dentro de un tiempo razonable), las redes neuronales artificiales parten de una base distinta, considerando que los seres humanos tenemos un sistema de procesamiento diferente al existente en los ordenadores, el cerebro, cuyas capacidades son las que se quieren emular no opera como lo hacen las maquinas tradicionales, este no está formado por un único microprocesador (capaz de ejecutar una serie de instrucciones complejas de manera fiable), tampoco por unos cuantos CPU, sino que lo componen millones de procesadores elementales o neuronas, ampliamente interconectadas conformando redes de neuronas. Una neurona es un pequeño procesador, sencillo, lento y poco fiable (a diferencia de los microprocesadores de las maquinas), sin embargo en nuestro cerebro cohabitan unos cien mil millones de neuronas operando en paralelo, lo cual es el origen de su robustez ya que aunque las neurona sean capaces de ejecutar procesamientos muy simples, ampliamente interconectadas a través de sinapsis (en promedio, cada neurona puede conectarse con otras 10.000) y trabajando en paralelo pueden desarrollar una actividad global de procesamiento tremendamente poderosa.

Además, las neuronas no deben ser programadas, estas aprenden a partir de las señales que reciben del entorno y operan siguiendo un esquema también muy diferente al de los computadores convencionales. Aquí aparece una de las primeras características de las redes neuronales, la Autoorganización, en una red de neuronas no existe un único componente que gobierne el sistema (como la CPU en el caso de las computadoras), las neuronas se influyen mutuamente a través de sinapsis excitadoras e inhibitoras, lo que causa una compleja dinámica de activaciones y desactivaciones de ellas. Las neuronas en definitiva, se autoorganizan, aprendiendo del entorno y adaptándose a él, y de esta autoorganización emergen propiedades de procesamientos complejos.

En resumen nuestro cerebro resulta ser un complejo sistema de procesamiento, no lineal, masivamente paralelo, y adaptativo, pero además es extraordinariamente eficiente desde un punto de vista energético: nuestro cerebro emplea 10^{16} Julios para ejecutar una operación por segundo, mientras que los ordenadores actuales emplean unos 10^{-6} , diez órdenes de magnitud más [10].

La siguiente tabla resume las principales características del cerebro y realiza una comparación respecto de los sistemas tradicionales de procesamiento de los computadores.

	Cerebro	Computador
Velocidad de proceso	$\approx 10^{-2}$ seg. (100 Hz)	$\approx 10^{-9}$ seg. (1000 Hz)
Estilo de Procesamiento	Paralelo	Secuencial
Número de procesadores	$10^{11} - 10^{14}$	Pocos
Conexiones	10.000 por procesador	Pocas
Almacenamiento del conocimiento	Distribuido	Direcciones fijas
Tolerancia a fallos	Amplia	Nula
Tipo de control del proceso	Auto-organizado	Centralizado

Tabla 2.2.4.1 - Comparación cerebro / sistemas tradicionales de procesamiento

Fundamentos de las redes neuronales biológicas

Una neurona es una célula del sistema nervioso presente en el cerebro cuya función principal es la captura, procesamiento y emisión de señales eléctricas. Se piensa que la capacidad de procesamiento de información del cerebro proviene principalmente de redes de este tipo de neuronas, por esta razón algunos de los primeros trabajos en “Inteligencia Artificial” (IA) comenzaron a crear redes neuronales artificiales.

Desde 1943 se han desarrollado modelos más realistas y detallados, tanto para neuronas como para sistemas más grandes del cerebro, llevándonos al campo moderno de la *neurociencia computacional*. Por otro lado los investigadores de IA y de estadística se han interesado en las propiedades más abstractas de las redes neuronales, tales como su habilidad para desarrollar computación distribuida, para tolerar el ruido presente en una entrada, y para el aprendizaje. Aunque otras clases de sistemas (incluyendo redes bayesianas) tienen estas propiedades, las redes neuronales permanecen como una de las formas más populares y efectivas de construir sistemas de aprendizaje [7].

A modo de comprender las características imitadas por las RNA a continuación se describirán aspectos generales de su símil biológico.

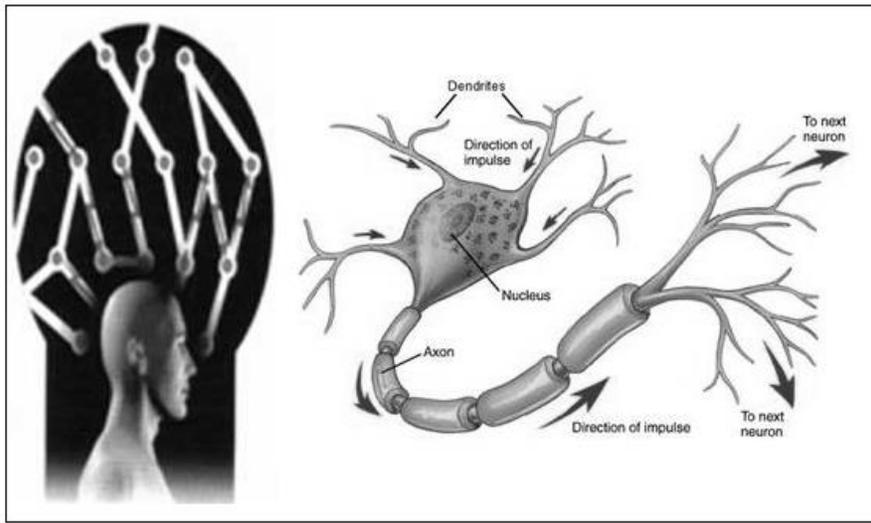


Figura 2.2.4.1 - Impulsos presentes en una neurona biológica

Se estima que el sistema nervioso contiene alrededor de cien millones de neuronas. Vistas al microscopio, este tipo de células puede presentarse en múltiples formas, aunque muchas de ellas presentan un aspecto similar al presentado en la figura anterior, con un cuerpo celular o soma (de entre 10 y 80 micras de longitud), del que surge un denso árbol de ramificaciones (árbol dendrítico) compuesto por las dendritas, y del cual parte una fibra tubular denominada axón que también se ramifica en su extremo final para conectar con otras neuronas (cuya longitud varía desde las 100 micras hasta el metro en el caso de las neuronas motoras).

Desde un punto de vista funcional, las neuronas constituyen procesadores de información sencillos, como todo sistema de este tipo, poseen un canal de entrada de información, las dendritas, un órgano de cómputo, el soma y un canal de salida, el axón. En las inter-neuronas el axón envía la información a otras neuronas, mientras que en las neuronas motoras lo hace directamente al músculo. Existe un tercer tipo de neuronas, las receptoras o sensorias, que en vez de recibir información de otras neuronas, las reciben directamente del exterior (tal como sucede en el sistema visual, por ejemplo, en los conos y bastones de la retina). Se calcula que una neurona del córtex cerebral recibe información, en promedio de unas 10.000 neuronas (convergencia) y envía impulsos a varios cientos de ellas (divergencia).

En el córtex cerebral se aprecia la existencia de una organización horizontal en capas (se suele señalar unas seis capas), coexistiendo una organización vertical en forma de columnas de neuronas. Hay grupos neuronales, compuestos por millones de neuronas pertenecientes a una determinada región del cerebro, que constituyen unidades funcionales especializadas en ciertas tareas (por ejemplo, existe un área visual, un área auditiva, un córtex sensorio-motor, etc.); todos los subsistemas juntos conforman el encéfalo, se tiene evidencia de que el procesamiento en el sistema nervioso involucra la actuación de muchos de estos subsistemas, que intercambian continuamente información.

La unión entre dos neuronas se denomina sinapsis, en el tipo de sinapsis más común no existe contacto físico entre las neuronas, sino que éstas permanecen separadas por un pequeño vacío de unas 0.2 micras. En relación a la sinapsis, se habla de neuronas presinápticas (la que envía las señales) y postsinápticas (la que las recibe). Las sinapsis son direccionales, es decir, la información fluye siempre en un único sentido.

La intensidad de una sinapsis no viene representada por una cantidad fija, sino que puede ser modulada en una escala temporal mucho más amplia que la del disparo de las neuronas. Esta plasticidad sináptica se supone que constituye, al menos en buena medida, el aprendizaje. Durante el desarrollo de un ser vivo, el cerebro se modela, de forma que existen muchas cualidades del individuo que no son innatas, sino que se adquieren por la influencia de la información que del medio externo proporcionan sus sensores. Existen diferentes formas de modelar un sistema nervioso: por el establecimiento de nuevas conexiones, rupturas de otras, modelado de las intensidades sinápticas (plasticidad) o incluso mediante la muerte neuronal. Este tipo de acciones (en especial la modificación de las intensidades sinápticas) serán las que utilicen los sistemas neuronales artificiales para llevar a cabo el aprendizaje.

Modelo general de una neurona artificial

Se denomina “procesador elemental” o “neurona” a un dispositivo simple de cálculo que, a partir de un vector de entrada procedente del exterior o de otras neuronas, proporciona una única respuesta o salida. Desde un punto de vista informal, una neurona se activa cuando la combinación lineal de sus entradas excede de un determinado umbral. Los elementos que constituyen la neurona artificial son los siguientes [7]:

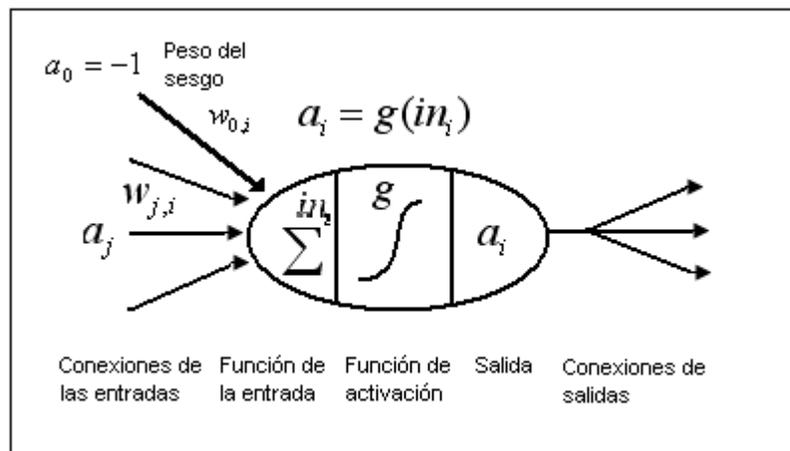


Figura 2.2.4.2 - Esquema de una neurona artificial típica.

Nótese que se ha incluido un *peso de sesgo* $w_{0,i}$ conectado a una entrada fija $a_0 = -1$ (este sesgo generalmente representa el umbral de disparo o activación de la neurona).

La información proveniente desde el exterior se recibe a través de las *Conexiones de entradas* $a_j(t)$

Pesos sinápticos de la neurona, son los $w_{j,i}$ que representan la intensidad de interacción entre cada neurona presináptica j con la neurona postsináptica i .

Regla de propagación o función de entrada que proporciona el valor del potencial postsináptico $in_i(t)$ de la neurona i en el instante de tiempo (t) en función de sus pesos y entradas.

Función de activación $g(in_i)$ que proporciona el estado de activación actual de la neurona i en función de su estado anterior y de su potencial postsináptico actual.

Función de salida $a_i(t)$ que proporciona la salida de la neurona i en el instante de tiempo (t) en función de su estado de activación.

Una conexión de la unidad neuronal j a la unidad i sirve para propagar la *activación* a_j de j a i . Además como se describió anteriormente cada conexión tiene un peso numérico $w_{j,i}$ asociado, que determina la fuerza y el signo de la conexión. Cada unidad neuronal i primero calcula una suma ponderada de sus entradas:

$$in_i = \sum_{j=0}^n w_{j,i} a_j$$

Formula 2.2.4.1 – Suma ponderada de las entradas de una neurona.

Luego aplica una *función de activación* g a esta suma para producir la salida:

$$a_i = g(in_i) = g\left(\sum_{j=0}^n w_{j,i} a_j\right)$$

Formula 2.2.4.2 – Función de activación de una neurona.

La función de activación g se diseña con dos objetivos. Primero, queremos que la unidad esté “*activa*” (cercana a +1) cuando se proporcionen las entradas “*Correctas*”. Segundo, la activación tiene que ser no lineal, en otro caso la red neuronal en su totalidad se colapsaría con una sencilla función lineal.

Podemos asemejar la operación de las unidades individuales a puertas lógicas. Una de las motivaciones originales para el diseño de unidades individuales fue su habilidad para representar las funciones booleanas básicas. La siguiente figura muestra cómo se pueden representar las funciones booleanas AND, OR y NOT con unidades umbral con los pesos adecuados, esto es importante ya que significa que podemos usar estas unidades para construir una red que calcule cualquier función booleana de las entradas.

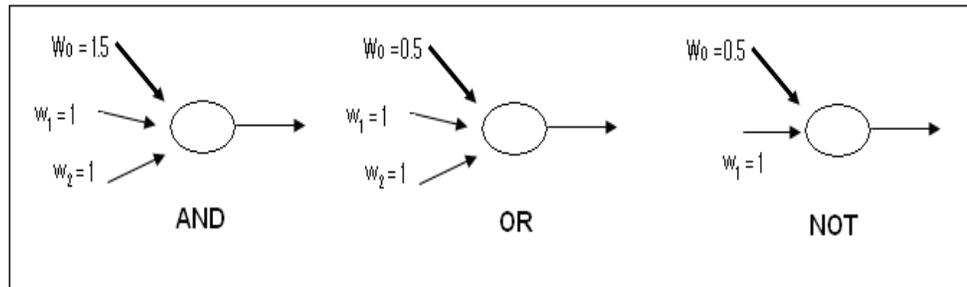


Figura 2.2.4.3 - Ejemplos de funciones booleanas en redes neuronales.

Estructuras de las redes neuronales artificiales.

Se denomina estructura o arquitectura a la topología, estructura o patrón de conexionado de un conjunto o red de neuronas artificiales (RNA). En una RNA los nodos se conectan por medio de sinapsis, esta estructura de conexiones sinápticas determina el comportamiento de la red. Las conexiones sinápticas son direccionales, es decir, la información solamente puede propagarse en un único sentido (desde la neurona presináptica a la postsináptica). En general las neuronas se suelen agrupar en unidades estructurales denominadas *capas*. Las neuronas de una capa pueden agruparse, a su vez, formando *grupos neuronales* (clusters). Dentro de un grupo, o de una capa si no existe este tipo de agrupación, las neuronas suelen ser del mismo tipo. Finalmente el conjunto de una o más capas constituye la *red neuronal*.

Existen 3 tipos de capas: de entrada, de salida y ocultas. Una *capa de entrada* o sensorial está compuesta por neuronas que reciben datos o señales procedentes del entorno (por ejemplo, proporcionadas por sensores). Una *capa de salida* es aquella cuyas neuronas proporcionan la respuesta de la red neuronal (sus neuronas pueden estar conectadas a efectores). Una *capa oculta* es aquella que no tiene una conexión directa con el entorno, es decir, que no se conecta directamente a órganos sensores ni efectores, este tipo de capa proporciona a la red neuronal grados de libertad adicionales, gracias a los cuales puede encontrar representaciones internas correspondientes a determinados rasgos del entorno [7][10].

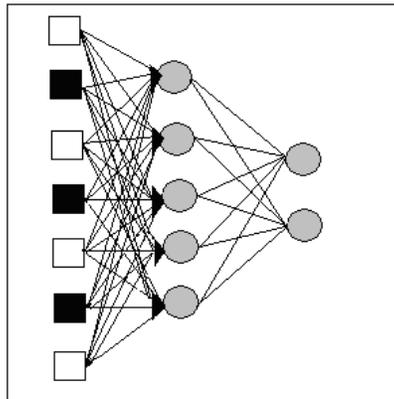


Figura 2.2.4.4 - Arquitectura de una RNA unidireccional de tres capas: de entrada, oculta y de salida.

En relación al tipo de conexión de las neuronas se pueden encontrar dos tipos de conexiones: intra-capa e inter-capa. Las conexiones *intra-capa*, también denominadas laterales, tienen lugar entre las neuronas pertenecientes a una misma capa, mientras que las conexiones *inter-capa* se producen entre las neuronas de diferentes capas. Existen además conexiones realimentadas, que tienen un sentido contrario al de la entrada-salida, en algunos puede existir realimentación incluso de una neurona consigo misma.

Si consideramos la estructura de las redes neuronales basadas el número de capas que estas posean, podemos diferenciar dos tipos: redes monocapa y redes multicapas. Las *redes monocapa* son aquellas compuestas por una única capa de neuronas. Las *redes multicapas* (layered networks) son aquellas cuyas neuronas se organizan en varias capas.

Dependiendo del flujo de los datos, podemos identificar dos categorías principales de estructuras de redes neuronales: acíclicas o *redes con alimentación hacia delante (feedforward)* y cíclicas o *redes recurrentes (feedback)*. Una red con alimentación hacia delante (unidireccional) representa una función de sus entradas actuales; de este modo, no tienen otro estado interno que no sea el de sus propios pesos. Por otro lado, una red recurrente permite que sus salidas alimenten sus propias entradas, esto significa que los niveles de activación de la red forman un sistema dinámico que debe alcanzar un estado estable, exhibir oscilaciones o incluso un comportamiento caótico. Además, la respuesta de una red dadas unas entradas depende de su estado inicial, que dependerá de entradas previas, por lo tanto, las redes recurrentes (a diferencia de las redes con alimentación hacia delante) pueden tener memoria a corto plazo, esto las hace más interesantes como modelos del cerebro, pero también más difíciles de entender.

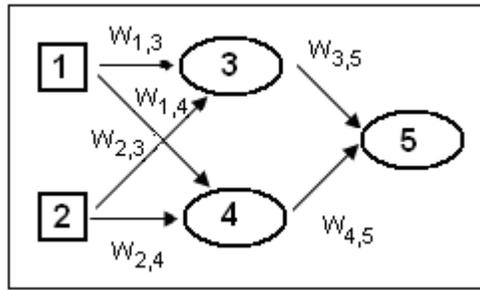


Figura 2.2.4.5 - Ejemplo de una red neuronal simple

Considerando una sencilla red neuronal como la mostrada en la figura 2.2.4.5, analizaremos la afirmación de que una red con alimentación hacia delante representa una función de sus entradas, la red de la figura tiene dos unidades de entrada, dos *unidades ocultas* y una unidad de salida (para mantener la simplicidad del ejemplo no se considerará la existencia de unidades de sesgo). Dado un vector de entrada $x = (x_1, x_2)$, las activaciones de las unidades de entrada se ponen a $(a_1, a_2) = (x_1, x_2)$ y la red calcula:

$$a_5 = g(w_{3,5}a_3 + w_{4,5}a_4) = g(w_{3,5}g(w_{1,3}a_1 + w_{2,3}a_2) + w_{4,5}g(w_{1,4}a_1 + w_{2,4}a_2))$$

Formula 2.2.4.3 – Salida de un RNA expresada como función de sus entradas.

Es decir, expresando la salida de cada unidad oculta como una función de sus entradas, hemos mostrado la salida de la red como una suma, a_5 , en función de las entradas de la red. Además, en la ecuación anterior se puede ver que los pesos de la red actúan como *parámetros* de la función; escribiendo w para los parámetros, la red calcula $hw(x)$. Ajustando los pesos, cambiamos la función que representa la red. Como se ha mencionado anteriormente, esta es la manera en que se produce el aprendizaje en las redes neuronales.

En la sección siguiente se dará una descripción general de uno de los primeros modelos de RNA creado “el perceptrón simple”, debido a su importancia histórica y su carácter introductorio.

Red Neuronal de una sola Capa con Alimentación Hacia Delante (Perceptrón Simple)

Una red con todas las entradas conectadas directamente a las salidas se denomina *red neuronal de una sola capa*, o red *perceptrón*, ya que cada unidad de salida es independiente de las otras (cada peso afecta sólo a una de las salidas). Podemos limitar nuestro estudio a los perceptrones con una única unidad de salida, como el que se muestra en la siguiente figura:

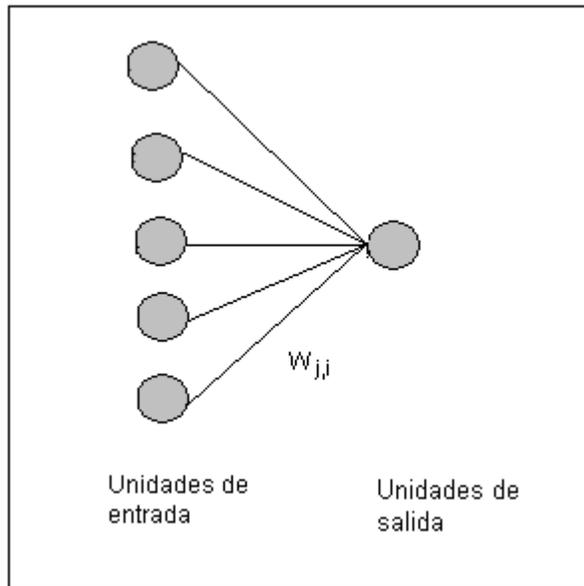


Figura 2.2.4.6 - Ejemplo de RNA de tipo Perceptrón Simple

Este modelo neuronal fue introducido por Rosenblatt a finales de los años cincuenta (1959). La estructura del perceptrón se inspira en las primeras etapas de procesamiento de los sistemas sensoriales de los animales (por ejemplo, el de la visión), en los cuales la información va atravesando sucesivas capas de neuronas, que realizan un procesamiento progresivamente de más alto nivel.

Un perceptrón utiliza una función de activación del tipo *escalón*, por lo que definiendo un umbral determinado (definido como sesgo en la figura 2.2.4.2) puede representar una función booleana. Además de las funciones booleanas elementales AND, OR, y NOT (mostradas como ejemplos anteriormente), puede representar algunas funciones booleanas un poco más complejas de forma compacta, por ejemplo, la *función mayoría* cuya salida es 1 sólo si más de la mitad de sus n entradas están en 1, puede representarse con un perceptrón con peso $W_j = 1$ y umbral $W_0 = n/2$ (un árbol de decisión necesitaría $O(2^n)$ nodos para representar esta función).

Un perceptrón puede representar sólo funciones *linealmente separables*, es decir, cuyas regiones de decisión pueden ser separadas mediante una única condición lineal o hiperplano (una línea recta si trabajamos en dos dimensiones). Si los datos son linealmente separables, se puede usar una regla de actualización de pesos sencilla para ajustar exactamente los datos. Sin embargo si los datos no cumplen este requisito de linealidad el perceptrón simple fallará estrepitosamente.

Una de las funciones no lineales que sirve de ejemplo para demostrar esta limitación del perceptrón es la función lógica XOR (su salida es el 0 lógico si las variables de entrada son iguales y 1 si son diferentes), en la siguiente figura se puede apreciar que para esta función lógica no se puede encontrar una única condición lineal que

A continuación se describirá la regla de aprendizaje del perceptrón. Sea un conjunto de p patrones $x^u, u = 1, \dots, p$, con sus salidas deseadas t^u , tanto las entradas como las salidas solamente pueden tomar los valores -1 o 1 (o bien, 0 o 1, según se definan los niveles lógicos), se tiene una arquitectura de perceptrón simple, con pesos iniciales aleatorios y se requiere que clasifique correctamente todos los patrones del conjunto de aprendizaje (lo cual es posible solamente si son linealmente separables), la regla de aprendizaje del perceptrón actuará del siguiente modo, ante la presentación del patrón u -ésimo, si la respuesta que proporciona el perceptrón es correcta, no actualizaremos los pesos; si es incorrecta, los modificaremos según la regla de Hebb [10]. Se tiene:

$$\Delta w_{ij}^u(t) = \left\{ \begin{array}{l} 2\varepsilon \cdot t_i^u \cdot x_j^u, \text{ si } \dots y_i^u \neq t_i^u \\ 0, \text{ si } \dots y_i^u = t_i^u \end{array} \right\}$$

Formula 2.2.4.4 – Regla de aprendizaje del Perceptrón simple.

Que se puede reescribir del siguiente modo

$$\Delta w_{ij}^u(t) = \varepsilon \cdot (t_i^u - y_i^u) x_j^u$$

Formula 2.2.4.5 – Regla de aprendizaje del Perceptrón simple, representación habitual.

Que es la forma habitual de expresar la regla del Perceptrón. En su utilización práctica, se debe llegar a un compromiso para el valor de un ritmo de aprendizaje ε , puesto que un valor pequeño implica un aprendizaje lento, mientras que uno excesivamente grande puede conducir a oscilaciones en el entrenamiento, al introducir discretas $\{-1, +1\}$, también lo será la actualización de los pesos, que únicamente podrá tomar los valores 0 o $\pm 2\varepsilon$.

En el proceso de aprendizaje, ante la presentación del patrón u -ésimo en la iteración t pueden darse los siguientes casos:

- a) La salida objetivo de la neurona es $t_i^u = +1$, pero su salida actual es $y_i^u = -1$. En este caso, el producto escalar $w_i \cdot x^u$ debería ser positivo, pero es negativo, lo cual indica que el ángulo existente entre w_i y x^u es mayor de 90° . Así, la regla de aprendizaje del perceptrón debería en este caso acercar w_i a x^u para reducir el ángulo que forman, y eventualmente conseguir que sea inferior a 90° ($w_i \cdot x^u > 0$), lo cual se puede realizar del siguiente modo:

$$w_i^u(t+1) = w_i^u(t) + \alpha \cdot x^u$$

Formula 2.2.4.6 – caso a): Regla de aprendizaje que debe acercar w_i a x^u

b) La salida objetivo de la neurona es $t_i^u = -1$, pero su salida actual es $y_i^u = +1$. Razonando al revés

que en el caso anterior, la regla de aprendizaje deberá alejar a w_i de x^u , por lo tanto en este caso:

$$w_i^u(t+1) = w_i^u(t) - \alpha \cdot x^u$$

Formula 2.2.4.7 – caso b): Regla de aprendizaje que debe alejar a w_i de x^u

c) La salida objetivo de la neurona t_i^u coincide con su salida actual y_i^u . En este caso la regla de aprendizaje no actúa.

Los tres casos anteriores se resumen en la siguiente regla:

$$w_i(t+1) = w_i(t) + (\alpha/2) \cdot x^u (t_i^u - y_i^u)$$

Formula 2.2.4.8 – Caso general de la regla de aprendizaje.

Y llamando a $\varepsilon \equiv \alpha/2$, se tiene

$$\Delta w_i(t) = \varepsilon \cdot x^u (t_i^u - y_i^u)$$

La cual es la regla del Perceptrón definida en la página anterior (con $w_i = w_{ij}$).

Se debe resaltar que el proceso de aprendizaje es iterativo, es decir, se parte de una configuración sináptica de partida (por lo general con pesos aleatorios) y se presentan una y otra vez los patrones, para que los pesos se ajusten iterativamente según la regla de Hebb vista anteriormente, hasta que todos queden correctamente ajustados. El hiperplano que establece el límite entre dos clases se desplaza lentamente hasta conseguir separarlas por completo (si es que es posible), como se muestra en la siguiente figura:

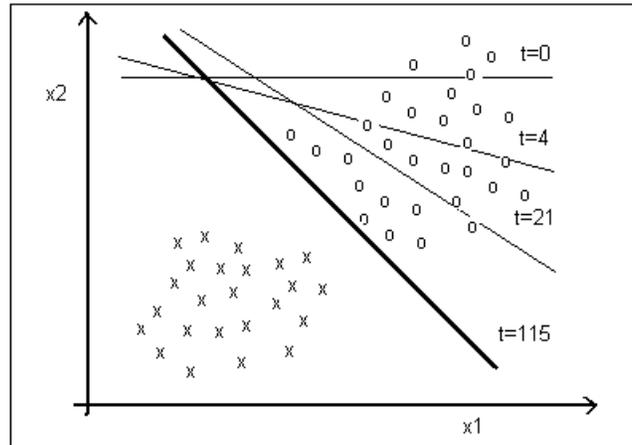


Figura 2.2.4.8 - Regiones de decisión que el Perceptrón establece iterativamente durante el aprendizaje.

En la figura anterior, el algoritmo de aprendizaje de la RNA en la iteración 115 ha conseguido separar ya las dos clases.

El ajuste de los pesos en la iteración t debido a todo el conjunto de aprendizaje será:

$$w_{ij}(t+1) = w_{ij}(t) + \sum_{u=1}^p \Delta w_{ij}^u(t)$$

Formula 2.2.4.9 – Ajuste de los pesos del Perceptrón en iteración t .

Roseblatt demostró que si la función a representar es linealmente separable, este algoritmo siempre converge en un tiempo finito y con independencia de los pesos de partida. El algoritmo del perceptrón se detiene tan pronto como consigue clasificar correctamente todos los ejemplos. Por otra parte, si la función no es linealmente separable, el proceso de entrenamiento oscilará.

Red Neuronal Artificial Multicapa con Alimentación Hacia Adelante.

Al considerar redes con unidades ocultas, el ejemplo más sencillo corresponde a la utilización de una única capa oculta como se muestra en la siguiente figura:

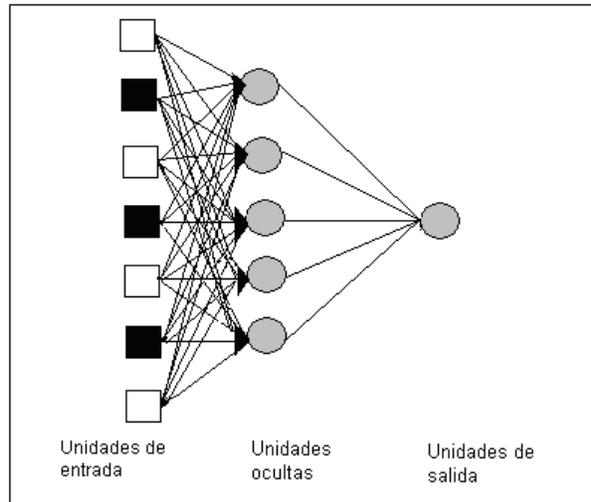


Figura 2.2.4.9 – RNA multicapa con una única capa oculta.

La ventaja de añadir capas ocultas es que se amplía el espacio de hipótesis que pueden ser representadas por la red. Este tipo de red neuronal artificial es conocida como Perceptrón multicapa (o MLP por sus siglas del inglés MultiLayer Perceptron).

La estructura de una red MLP es como la mostrada en la figura anterior, donde existen entradas de la red x_i , salidas de la capa oculta y_j y salidas de la capa final z_k (y globales de la red); t_k serán las salidas objetivo (target). Por otro lado, w_{ij} son los pesos de la capa oculta y θ_j sus umbrales, w'_{ij} los pesos de la capa de salida y θ_k sus umbrales. Por lo tanto, la operación de una red MLP con una capa oculta y neuronas de salida lineal (estructura que se ha demostrado representa un aproximador universal de funciones) se expresa con la siguiente función matemática:

$$z_k = \sum_j w'_{kj} y_j - \theta_k = \sum_j w'_{kj} f\left(\sum_i w_{ji} x_i - \theta_j\right) - \theta_k$$

Formula 2.2.4.10 – Operación de una red MLP.

Siendo la función $f(.)$ del tipo sigmoide, como por ejemplo lo son las siguientes:

$$f(x) = \frac{1}{1 + e^{-x}}$$

Formula 2.2.4.11 – Función Logística.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \tanh(x)$$

Formula 2.2.4.12 – Función tangente hiperbólica.

Proporcionando la primera una salida en el intervalo $[0,+1]$ y en $[-1,+1]$ la segunda.

La anterior es la arquitectura más común de una red MLP, aunque existen variantes como incluir neuronas no lineales en la capa de salida (más utilizadas en problemas de clasificación), introducir más capas ocultas, emplear otras funciones de activación, limitar el número de conexiones entre una neurona y las de la capa siguiente, introducir dependencias temporales o arquitecturas recurrentes, entre muchas otras. Las redes neuronales multicapas con alimentación hacia delante pueden representar cualquier función, con suficientes unidades.

Esta arquitectura de una red MLP suele entrenarse mediante el algoritmo denominado retropropagación de errores (BP por sus siglas del inglés back-propagation), o bien haciendo uso de algunas de sus variables o derivados, motivo por el que en muchas ocasiones el conjunto de arquitectura MLP con aprendizaje BP suele denominarse *Red de Retropropagación*, o simplemente BP.

El algoritmo de *propagación hacia atrás* (back-propagation) implementa un descenso del gradiente en el espacio de parámetros para minimizar el error de la salida. En la siguiente sección se detallará su funcionamiento.

Aprendizaje por Retropropagación de Errores (BP)

Una complejidad existente en las redes MLP es la del entrenamiento de los nodos de las capas ocultas, una solución a este problema lo proporciona el algoritmo de retro-propagación de errores o BP (back-propagation).

Sea una red MLP de 3 capas, con una arquitectura como la presentada en la figura 2.2.4.9. Según las entradas, salidas, pesos y umbrales de las neuronas definidas en la sección anterior, dado un patrón de entrada x^u , ($u = 1, \dots, p$) la operación global de esta arquitectura se expresa de acuerdo a la formula 2.2.4.10 “Operación de una red MLP”. Las funciones de activación de las neuronas ocultas $f(x)$ son de tipo sigmoidea, con h el potencial post-sináptico o local.

La función coste de la que se parte es el error cuadrático medio:

$$E(w_{ji}, \theta_j, w_{kj}, \theta_k) = (1/2) \sum_u \sum_k \left[t_k^u - f \left(\sum_j w_{kj} y_j^u - \theta_k \right) \right]^2$$

Formula 2.2.4.13 – Función de coste según error cuadrático medio.

La minimización se lleva a cabo mediante descenso por el gradiente, pero en esta ocasión habrá un gradiente respecto de los pesos de la capa de salida y otro respecto de los de la oculta:

$$\delta w_{kj} = -\varepsilon \frac{\partial E}{\partial w_{kj}} \qquad \delta w_{ji} = -\varepsilon \frac{\partial E}{\partial w_{ji}}$$

Formula 2.2.4.14 – Gradiente pesos capa de salida (izquierda) y gradiente pesos capa oculta (derecha).

Las expresiones de actualización de los pesos se obtienen sólo con derivar, teniendo en cuenta las dependencias funcionales y aplicando adecuadamente la regla de la cadena se llega al siguiente resultado:

$$\delta w_{kj} = \varepsilon \sum_u \Delta_k^u y_j^u, \text{ con } \Delta_k^u = [t_k^u - f(v_k^u)] \frac{\partial f(v_k^u)}{\partial v_k^u}$$

Formula 2.2.4.15 – Actualización pesos de capa de salida.

$$\delta w_{ji} = \varepsilon \sum_u \Delta_i^u y_i^u, \text{ con } \Delta_j^u = \left(\sum_k \Delta_k^u w_{kj} \right) \frac{\partial f(v_j^u)}{\partial v_j^u}$$

Formula 2.2.4.16 – Actualización pesos de capa oculta.

La actualización de los umbrales (bias) se realiza haciendo uso de estas mismas expresiones, considerando que el umbral es un caso particular de peso sináptico, cuya entrada es una constante igual a -1.

En estas expresiones está implícito el concepto de propagación hacia atrás de los errores que da nombre al algoritmo. En primer lugar se calcula la expresión Δ_k^u , denominada *función de error* por ser proporcional al error de la salida actual de la red, con el que calculamos la actualización δw_{kj} de los pesos de la capa de salida. A continuación se propagan hacia atrás los errores Δ_k^u a través de las sinapsis, proporcionando así las señales de error

Δ_j^u correspondientes a las sinapsis de la capa oculta; con estas se calcula la actualización δw_{ji} de las sinapsis ocultas. Este algoritmo puede extenderse fácilmente a arquitecturas con más de una capa oculta siguiendo el mismo esquema.

Como resumen, los pasos del algoritmo necesario para entrenar mediante BP una arquitectura de red MLP se presentan a continuación:

1. Establecer aleatoriamente los pesos y umbrales iniciales ($t:=0$).
2. Para cada patrón u -ésimo del conjunto de aprendizaje
 - 2.1. Llevar a cabo la fase de ejecución para obtener la respuesta de la red ante el patrón u -ésimo.
 - 2.2. Calcular las señales de error asociadas Δ_k^u y Δ_j^u según formulas 2.2.4.15 y 2.2.4.16 mostradas anteriormente.
 - 2.3. Calcular el incremento parcial de los pesos y umbrales correspondientes a cada patrón u (según resultados del paso anterior).
3. Calcular el incremento total actual de los pesos δw_{kj}^u y δw_{ji}^u (para todos los patrones, de acuerdo a formulas 2.2.4.15 y 2.2.4.16). Hacer lo mismo para los umbrales.
4. Actualizar pesos y umbrales.
5. Calcular el error actual $E(w_{ji}, \theta_j, w_{kj}, \theta_k)$ con $t:=t+1$, y volver a 2 si todavía no es satisfactorio.

Se debe comenzar siempre con pesos iniciales aleatorios (normalmente números pequeños, positivos y negativos), ya que si se parte de pesos y umbrales iniciales nulos el aprendizaje no progresará (puesto que las salidas de las neuronas y el incremento en los pesos serán siempre nulos).

El BP presentado actualiza los pesos sinápticos una vez que ya se han presentados todos los patrones a la red, lo que se denomina *aprendizaje por lotes (batch)*, una variación consiste en realizar esta actualización tras la presentación de cada patrón lo que se conoce como *aprendizaje en serie (on line)*, el cual es utilizado con mayor frecuencias cuando se tiene una cantidad numerosa de patrones de entrenamiento (por los requerimientos de procesamiento). El algoritmo de Backpropagation sea en Serie o por Lotes, representa un método de gran generalidad, lo que presenta ventajas y desventajas, su ventaja principal es que puede ser aplicado a una multitud de problemas diferentes proporcionando con frecuencia buenas soluciones con no demasiado tiempo de desarrollo. No obstante, si se quieren mejorar los resultados obtenidos habrá que profundizar en temas adicionales como elegir la

arquitectura optima, selección de los pesos iniciales, estilo de aprendizaje, pre-procesamiento de los datos de entrada, conjunto de patrones de aprendizaje empleado, utilización de técnicas que eviten el sobre ajustes, entre otros.

Como desventaja se encuentra, entre otras, su lentitud de convergencia que es uno de los precios que hay que pagar por disponer de un método general de ajuste funcional que no requiere (en principio) información apriorística. Sin embargo, se debe tener en cuenta que el BP no requiere tanto esfuerzo computacional como el que sería necesario si se aplicarían directamente la evaluación de las derivadas en la obtención de los pesos para la RNA (característica por la cual se tiende a comparar BP con la transformada rápida de Fourier). Otro de los problemas de BP es que puede incurrir en el denominado sobre-aprendizaje (o sobre-ajuste). También se debe tener en cuenta que el algoritmo de BP no garantiza alcanzar el mínimo global de la función de error, tan sólo un mínimo local, por lo que el proceso de aprendizaje puede quedarse estancado en uno de estos mínimos locales.

Aceleración del aprendizaje BP.

Para resolver el problema de la lenta convergencia del algoritmo de aprendizaje BP, se plantean continuamente correcciones o variantes. Así la primera variante la propusieron los propios inventores del modelo, al incluir en el algoritmo un término denominado “momento” (*momentum*), consistente en añadir al cálculo de la variación de los pesos un término adicional proporcional al incremento observado en la iteración anterior (inercia):

$$\delta w'_{kj}(t+1) = -\varepsilon \frac{\partial E}{\partial w'_{kj}} \Big|_t + \alpha \cdot \delta w_{ji}(t-1)$$

Formula 2.2.4.17 – Actualización pesos de la RNA considerando el término “momento”.

, donde

$$|I| = \left| +\alpha \cdot \delta w'_{kj}(t-1) \quad \delta w_{ji}(t+1) = -\varepsilon \frac{\partial E}{\partial w_{ji}} \right|$$

Con α un parámetro entre 0 y 1, que se suele tomar próximo a 1 ($\alpha = 0,9$). De esta manera, si los incrementos en un determinado peso tienen siempre el mismo signo, las actualizaciones en cada iteración serán mayores; sin embargo, si los incrementos en cierto peso oscilan (a veces son positivos, otras negativos), el incremento efectivo (acumulado) se reduce al cancelarse. Así, en zonas estrechas y profundas de la hiper-superficie de error (con forma de valle angosto) los pesos correspondientes a las dimensiones estrechas (que sin el término momento oscilarían de un lado al otro del valle) sufren incrementos pequeños, mientras que los de las direcciones

que descienden directamente al fondo se ven potenciados. Esta es una manera de aumentar el ritmo de aprendizaje efectivo en determinadas direcciones.

Otra característica a tener en cuenta es la magnitud de los pesos iniciales, pues una correcta elección puede suponer un menor tiempo de entrenamiento.

Aprendizaje de las RNA según la estructura utilizada.

Un aspecto importante de las redes neuronales tiene relación con la estructura implementada. Una red de mayor tamaño puede ser capaz de memorizar todos los ejemplos formando una gran tabla de búsqueda, pero esto no necesariamente significará que ante nuevas entradas generalizará de forma correcta. Por lo tanto como todos los modelos estocásticos, las redes neuronales son sujeto de *sobreajuste* cuando hay demasiados parámetros considerados dentro del modelo.

Al tratar con redes totalmente conectadas dos aspectos que se hacen críticos en un diseño óptimo de la red corresponden al número de capas ocultas a utilizar y el tamaño de estas, el enfoque más usual es intentar varias configuraciones y seleccionar la de mejor rendimiento.

Capacidad de Generalización de la Red.

Uno de los aspectos fundamentales de las RNA es su capacidad de generalizar a partir de ejemplos, lo que constituye el problema de la memorización frente a la generalización. Por generalización se entiende la capacidad de la red de dar una respuesta correcta ante patrones que no han sido empleados en su entrenamiento. Una red neuronal correctamente entrenada generalizará, lo que significa que ha aprendido adecuadamente el *mapping* no sólo de los ejemplos concretos presentados, respondiendo correctamente ante patrones nunca vistos con anterioridad. Por otro lado un sobre entrenamiento de la red puede provocar una memorización de los patrones clasificados y su comportamiento frente a patrones no conocidos previamente no será de los mejores.

Validación Cruzada

El procedimiento consistente en entrenar y validar a la vez para detenerse en el punto óptimo se denomina validación cruzada (cross validation) y es ampliamente utilizado en la fase de desarrollo de una RNA supervisada (como MLP).

En un proceso de entrenamiento se debe considerar, por un lado, el error de aprendizaje que se suele calcular como el error cuadrático medio de los resultados proporcionados por la red para el conjunto de patrones de

aprendizaje (con una red suficientemente grande, puede reducirse tanto como se quiera sólo con llevar a cabo más iteraciones), y por otra parte el error de generalización, que se puede medir empleando un conjunto representativo de patrones diferentes a los utilizados durante el entrenamiento. De esta manera, podemos entrenar una red neuronal haciendo uso de un conjunto de aprendizaje, y comprobar su eficiencia real, o error de generalización, mediante un conjunto de test.

Un hecho experimental, fácilmente observable en cualquier prueba realizada sobre algún simulador, es que si se entrena una red hasta alcanzar un muy pequeño error en aprendizaje (por ejemplo, inferior al 1%), la eficacia real del sistema o generalización (medido como error de test) se degrada.

La explicación de este fenómeno se da a continuación:

Al principio la red se adapta progresivamente al conjunto de aprendizaje, acomodándose al problema y mejorando la generalización. Sin embargo, en un momento dado el sistema se ajusta demasiado a las particularidades de los patrones empleados en el entrenamiento, aprendiendo incluso el ruido presente en ellos, por lo que crece el error que comete ante patrones diferentes (error de generalización), en este momento la red no ajusta correctamente el mapping, sino que simplemente esta memorizando los patrones del conjunto de aprendizaje, lo que técnicamente se denomina sobre-aprendizaje o sobre-ajuste (overtraining o overfitting). Idealmente una RNA debería entrenarse hasta un punto óptimo en donde el error de generalización es mínimo.

Cantidad de patrones utilizados durante el Entrenamiento

En definitiva, la capacidad de generalización de una RNA la determinan en buena medida las siguientes circunstancias:

- 1- La arquitectura de la red
- 2- El numero de ejemplos de entrenamiento
- 3- La complejidad del problema

Estos tres puntos están muy relacionados; en términos generales, cuanto más complejo sea el problema a modelar, mas grande deberá ser la red (con más parámetros a ajustar), y por lo tanto más ejemplos se necesitarán para entrenarla (ejemplos que deberán cubrir todo el espacio de entrada, contemplando todas las situaciones posibles). A menudo el número de patrones de ejemplo es limitado y en proporción al número de parámetros efectivos de la red elegida (grados de libertad) suele ser muy menor.

Una investigación demostró que para una red de “ n ” entradas y “ h ” neuronas ocultas, con un total de “ w ” pesos, requiere un número de patrones de aprendizaje del orden de $p=w/E$ para proporcionar un error de

generalización del orden de E . Así, si queremos que la red alcance un error de generalización de por ejemplo $E=0,1$ (un 10%), el número de patrones de aprendizaje necesarios " p " será del orden de $p=10.w$, expresión que se suele dar como indicador del número aproximado de patrones que serían necesarios para entrenar adecuadamente una red neuronal de w pesos.

Por ejemplo, para una red 10-5-1 (10 neuronas de entrada, 5 ocultas y 1 de salida) que dispone de 61 parámetros entre pesos y umbrales, el número de patrones necesarios para alcanzar un error del 10% será de unos 610, donde pese a no tratarse de una red de gran tamaño representa una cifra de patrones alta.

La relación $p=10.w$ (con p el número de ejemplos y w el número de pesos de la red) puede ser particularmente útil, pudiéndose utilizar de dos formas distintas: para establecer el tamaño aproximado de la red ideal para el conjunto de muestras disponibles, o dada una arquitectura de red, para hacernos una idea sobre si es fácil que incurramos en sobre-entrenamiento.

Clasificación de las redes neuronales artificiales.

De acuerdo con los conceptos vistos anteriormente puede deducirse que dependiendo del modelo neuronal concreto que se utilice, de la arquitectura o topología de conexión y del algoritmo de aprendizaje surgirán distintos modelos de redes neuronales.

En este proyecto es utilizada una RNA de tipo MLP, por lo tanto el estudio anterior ha estado acotado en su mayor parte a este tipo de arquitectura, sin embargo a continuación se entrega un resumen acerca de los modelos de redes neuronales artificiales más utilizadas en las aplicaciones actuales:

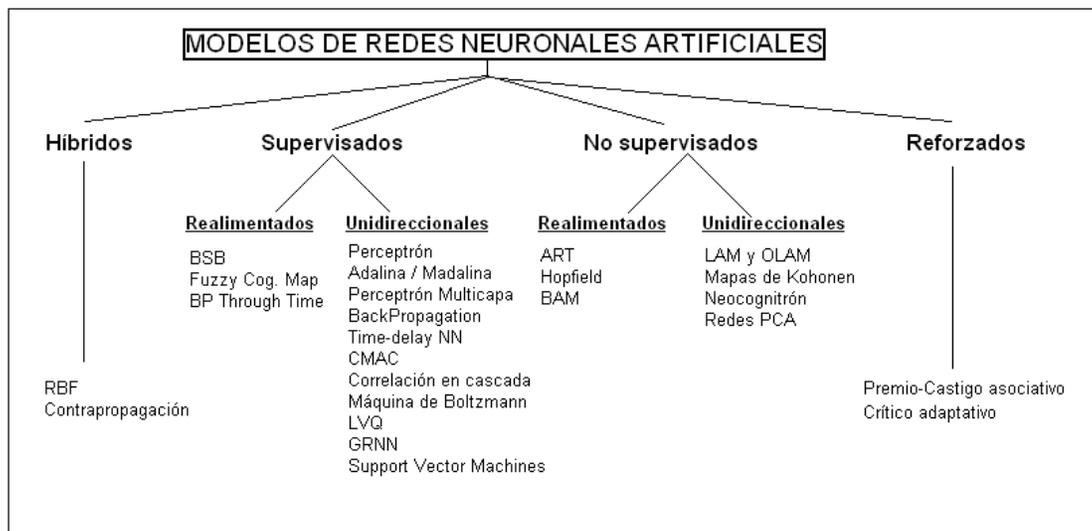


Figura 2.2.4.10 - Modelos de redes neuronales artificiales más conocidas y utilizadas.

Se puede apreciar en la figura anterior que los modelos de redes no realimentadas y de aprendizaje supervisado son los más numerosos, esta clase de modelos resulta especialmente importante por varias razones: por su interés histórico, generalidad, por ser los sistemas neuronales más empleados en las aplicaciones prácticas existentes y por ilustrar una amplia clase de aspectos que aparecen con frecuencia en el área (memoria asociativa, clasificación, aproximación funcional, entre otros).

Comparación de Métodos Estadísticos de Aprendizaje

Además de las RNA existen varios métodos de clasificación que trabajan con modelos estadísticos avanzados, en la siguiente tabla se comparan resultados característicos de algunas de estas técnicas de aprendizaje [7]:

	3 NN	300 ocultas	LeNet	LeNet Potenciada	SVM	SVM Virtual	Encaje Formas
Tasa de error (pct.)	2.4	1.6	0.9	0.7	1.1	0.56	0.63
T. ejec. (msg/dígito)	1000	10	30	50	2000	200	
M. Requerida (Mbyte)	12	0.49	0.012	0.21	11		
T. entrenamiento (días)	0	7	14	30	10		
% rechazado para un error del 0.5 %	8.1	3.2	1.8	0.5	1.8		

Tabla 2.2.4.2 - Comparación de métodos estadísticos de aprendizaje

2.3 Análisis del marco teórico y problema de investigación.

El estado del arte expuesto en la sección 2.2 anterior incluyó una gran cantidad de algoritmos del área, explicando en cada uno su naturaleza y funcionamiento, la presentación de cada técnica fue hecha en un orden lógico de acuerdo con la función y etapa de procesado en que cada una participa (figura 2.2.1).

Recordando el objetivo general de la tesis de realizar un "Estudiar técnicas del campo de la visión artificial, y a través de la implementación de métodos como los descriptores de Fourier (DF) y una RNA de tipo MultiLayer

Perceptron (MLP) analizar los resultados obtenidos por estos métodos en la clasificación de los nuevos objetos presentes en una escena, basado en el reconocimiento de patrones en las imágenes digitales que sean capturadas desde el entorno” podemos ver que ya se han estudiado estos métodos en el estado del arte, por lo que se tienen las bases teóricas suficientes para seleccionar un grupo de ellos y mediante su implementación observar los resultados obtenidos. En los próximos capítulos se mostrará esta implementación y posterior evaluación (de cada algoritmo en particular y del conjunto seleccionado).

En la siguiente tabla se puede ver el cumplimiento logrado hasta el presente capítulo de los objetivos específicos planteados para el desarrollo de esta investigación:

Objetivo	Cumplimiento
Estudiar los algoritmos de procesamiento de imágenes digitales existentes.	Realizado
Estudiar la Transformada de Fourier y su aplicación dentro del procesamiento de imágenes digitales.	Realizado
Estudiar la clasificación de patrones mediante la implementación de una RNA de tipo MLP que utilice el algoritmo de aprendizaje Back Propagation (BP).	Realizado
Abordar un caso de estudio donde mediante la implementación de algunas de las técnicas de visión artificial estudiadas se logró clasificar los nuevos objetos que aparecen en la escena, analizando los resultados obtenidos de acuerdo al reconocimiento de patrones alcanzado.	Pendiente
Mejorar los resultados de técnicas tradicionales que se basan en el dominio del espacio (tonalidades de los píxeles) y de esta forma sentar las bases suficientes que permitan realizar propuestas de futuras investigaciones que aborden técnicas innovadoras para el área (nuevas líneas de investigación).	Pendiente

Tabla 2.3.1 Cumplimiento objetivos específicos de la tesis.

Tres de los cinco objetivos específicos planteados para esta tesis han sido cumplidos con el estudio de la literatura realizado, aunque se debe mencionar que aquellos que siguen pendientes representa la mayor parte de las actividades a realizar durante esta investigación, ya que considera a la implementación y evaluación de los resultados obtenidos por estos métodos, que hasta acá sólo se han comprendido teóricamente. En los capítulos siguientes se aborda el cumplimiento de los últimos objetivos planteados, buscando detectar aquellos puntos fuertes y también aquellos que representen debilidades en el funcionamiento de una solución de estas características.

2.4 Conclusión

En las secciones anteriores de este capítulo se han revisado métodos del área del procesamiento de imágenes digitales y visión artificial que permiten obtener una buena base teórica en la implementación de un sistema de clasificación de objetos. Los diversos algoritmos fueron presentados en un orden lógico para comprender sus relaciones de mejor forma y visualizar las precedencias que se requieren en su implementación (figura 2.2.1).

Además se ha realizado un análisis de la relación existente entre el contenido del estado del arte con la investigación que se realiza en la presente tesis, consiguiendo con la revisión de la literatura el logro de tres de los cinco objetivos específicos planteados inicialmente, permitiendo así conseguir las bases suficientes para realizar la implementación y análisis de resultados de un conjunto de algoritmos (últimos dos objetivos de la tesis).

Capítulo 3

Metodología

En el capítulo anterior se ha revisado la literatura del área y se ha logrado conseguir la base teórica necesaria para implementar y analizando los resultados del funcionamiento de un grupo de algoritmos de visión artificial, en esta sección se presentan aspectos relacionados a la metodología adoptada para realizar dichas actividades.

3.1 Introducción

En este capítulo se describe y justifica la metodología básica utilizada en el desarrollo de la investigación, dando algunas consideraciones acerca de la implementación y análisis del resultado de los algoritmos seleccionados.

Se describe también un caso de estudio específico, donde se analizó el dominio del problema y luego se diseñó una propuesta de solución utilizando técnicas de visión artificial seleccionadas desde el estudio de la literatura realizado en el capítulo anterior.

Los procedimientos utilizados en la implementación y posterior análisis de resultados son mencionados en las últimas secciones de este capítulo, y en el siguiente capítulo es presentada en detalle la ejecución de estos.

3.2 Justificación del paradigma y la metodología

En el desarrollo de un proyecto de visión artificial se deben considerar varias etapas que incluyen diversos métodos y funciones cada una, como la captura de las imágenes, la mejora y aplicación de métodos de realce de características importantes, la detección y clasificación de los elementos presentes, entre otras. La siguiente figura expone uno de los modelos más aceptado dentro del área, donde se resumen las etapas necesarias para abordar un proyecto de VA [1]:

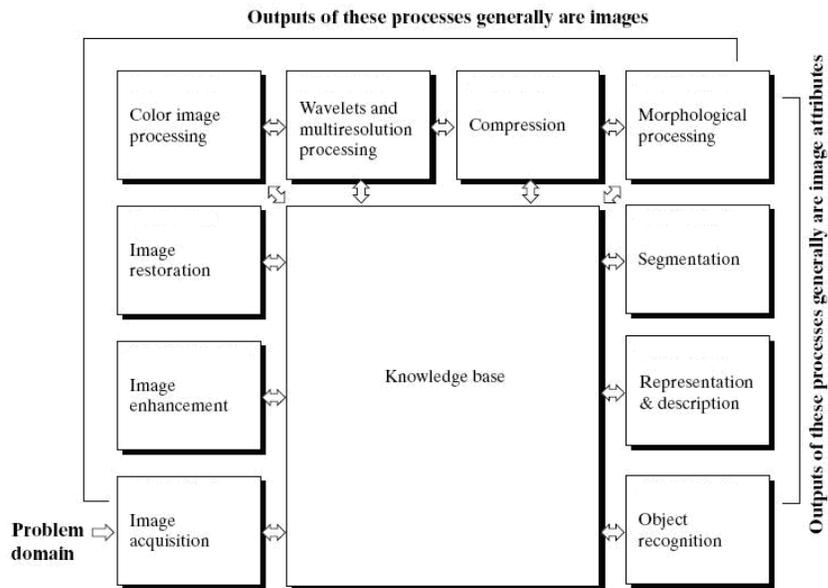


Figura 3.2.1 - Modelo de un proyecto del área de la visión artificial.

Cada una de las etapas existentes en el modelo de la figura anterior considera:

Etapas 1- Adquisición de imágenes: Generalmente considera pre-procesamiento. Incluye los métodos y medios de captura de imágenes digitales, dependiendo del entorno y las condiciones climáticas existirán diversas alternativas de procesamiento factibles.

Etapas 2- Mejora de la imagen: Este paso busca resaltar los detalles de mayor importancia en las imágenes que presentan una baja calidad de nitidez debido generalmente al ruido provocado por el medio de captura o el escenario de monitoreo.

Etapas 3- Restauración de la imagen: Las técnicas de restauración son generalmente modelos probabilísticos de reparación de la imagen que buscan responder a la pregunta: ¿Cómo era esta imagen antes de ser dañada?, su principal objetivo es entregar al computador una fuente de información en estado óptimo para su procesamiento.

Etapas 4- Procesamiento del color: Conversión de modelos de color de la imagen (RGB, HSI, etc.), disminución o realce de brillo y contraste, entre otros.

Etapas 5- Wavelet y Procesamiento de multiresolución: Técnicas fundamentalmente utilizadas para representar imágenes en varios niveles de resolución. Utilizadas inicialmente en compresión.

Etapas 6- Compresión: Reduce el almacenamiento requerido para guardar una imagen, o el ancho de banda necesario para transmitirla.

Etapa 7- Procesamiento morfológico: Herramientas para extraer componentes de la imagen o reparar alguna característica en la tonalidad de los píxeles, útiles para la representación y descripción de formas.

Etapa 8- Segmentación: Divide una imagen en sus partes constituyentes, se busca identificar los elementos que son de interés dentro de una imagen digital para luego asociar su significado de acuerdo al objeto que representan en la escena analizada.

Etapa 9- Representación y descripción: En esta etapa se toman decisiones tales como si la forma obtenida debe ser tratada como una frontera o una región, y se extrae información cuantitativa de interés.

Etapa 10- Reconocimiento de objeto: Se busca reconocer los objetos de interés presentes en una imagen.

Base de conocimiento: Almacenamiento de información referente a los resultados obtenidos en cada etapa del proceso, también permite mejorar procesamientos posteriores de las imágenes.

Dado que la naturaleza del presente proyecto posee características que son naturalmente practicables siguiendo el esquema de etapas indicado anteriormente es considerado su uso para la implementación del conjunto de algoritmos de visión artificial seleccionados. Este modelo ha sido formulado a partir de las experiencias obtenidas en los años de desarrollo del área y representa por lo tanto un modelo que ha evolucionado en el tiempo hasta en la actualidad lograr reunir las principales consideraciones que se deben tener presentes, esto apoya directamente a una buena ejecución del presente proyecto.

Además de implementar la transformada de Fourier y una estructura de red neuronal artificial (RNA), también son considerados otros algoritmos que ayudan a lograr una mejor interpretación de la escena de acuerdo con las etapas del modelo expuesto, el conjunto final de algoritmos a considerar es especificado en la sección siguiente luego de plantear la problemática a resolver.

3.3 Caso de Estudio y Solución Propuesta

Como caso de estudio será abordado el reconocimiento de objetos en un entorno conocido y en donde el determinar la existencia de nuevos elementos representa información útil para el control y monitoreo de la escena. Un ejemplo de la problemática tipo que se quiere resolver se puede ver en aplicaciones que buscan identificar y clasificar en una estación de trenes aquellos objetos “olvidados” por alguno de los pasajeros, el objetivo principal es clasificar estos objetos e identificar aquellos que posean características “sospechas”. En definitiva, la aplicación busca analizar las imágenes y lograr el aislamiento y clasificación de los objetos nuevos presentes en ellas (determinación de fondo de la escena y objeto de primer plano).

Caso de Estudio

Dada una secuencia de imágenes que representan una escena determinada durante un periodo de tiempo específico (en modo off-line), un grupo de estas imágenes serán representativas del fondo de la escena donde los objetos presentes en ella la mayor parte del tiempo serán los mismos y mantendrán la misma ubicación, otro grupo de estas imágenes contendrán nuevos elementos dentro de la escena y por lo tanto se espera que un sistema de visión artificial sea capaz de identificarlos y clasificarlos de acuerdo a las características más representativas que posean (región, textura, forma del contorno, etc.). El objeto específico que se espera lograr clasificar dentro del caso de estudio abordado corresponde a una “Persona”, donde de acuerdo a las características de su contorno una RNA pueda realizar una correcta identificación. También se incluirá para el análisis de los resultados obtenidos el reconocimiento de un segundo objeto, específicamente se incluirá la clasificación de sillas de oficina, para observar así el porcentaje de generalización alcanzada por la red (de acuerdo con las características de ambos tipos de objetos).

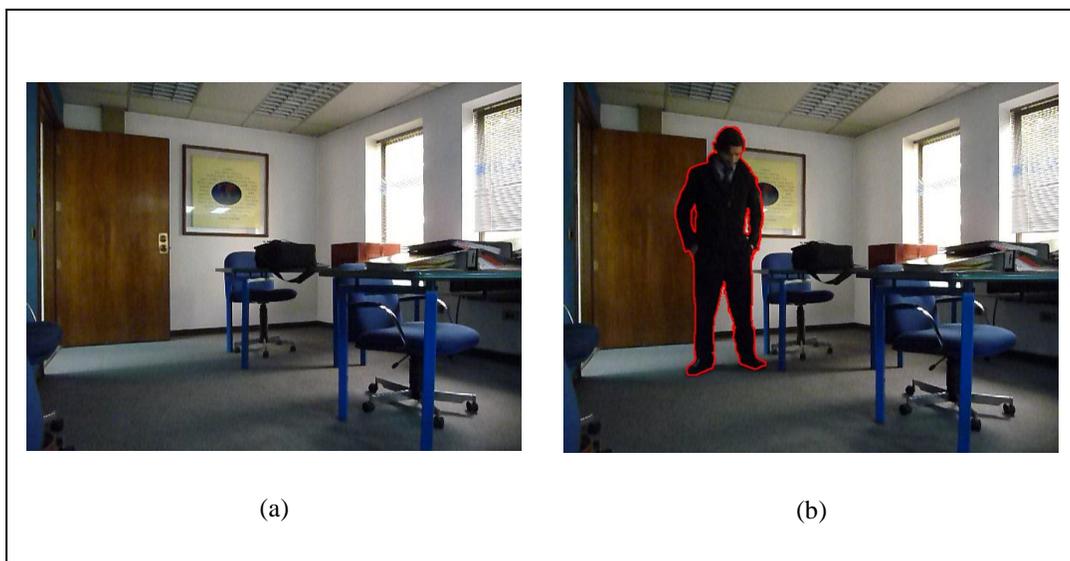


Figura 3.3.1: Imagen caso de estudio (a) imagen del fondo de la escena (b) detección y clasificación de acuerdo a las características del contorno de una persona.

La problemática seleccionada representa una base sobre la cual realizar pruebas del desempeño de los algoritmos de visión artificial seleccionados, además se estudian distintas escenas para observar el comportamiento del sistema ante diferentes imágenes y condiciones del entorno.

Solución Propuesta

Son utilizados descriptores de Fourier (DF) para representar los nuevos objetos detectados a través de aquellos píxeles que forman parte de su contorno, estos descriptores tienen la ventaja de trabajar sobre el dominio de las frecuencias y por lo tanto, como se ha explicado en el estado del arte, logran una descripción más robusta de los objetos (invariabilidad ante rotaciones, cambios de escalas y traslaciones). También es implementada una RNA de tipo MLP utilizando como algoritmo de entrenamiento el método Back Propagation (BP), los patrones utilizados para entrenar la red corresponden a vectores de características conformados por los descriptores de Fourier calculados previamente.

El proceso descrito requiere realizar un pre-procesado de las imágenes con filtros de mejora y realce (eliminación de ruido, realce de bordes) y la posterior segmentación detectando los objetos nuevos existentes en la escena. El diagrama de la figura 3.3.2 presenta el esquema de funcionamiento típico de un sistema de reconocimiento de formas:

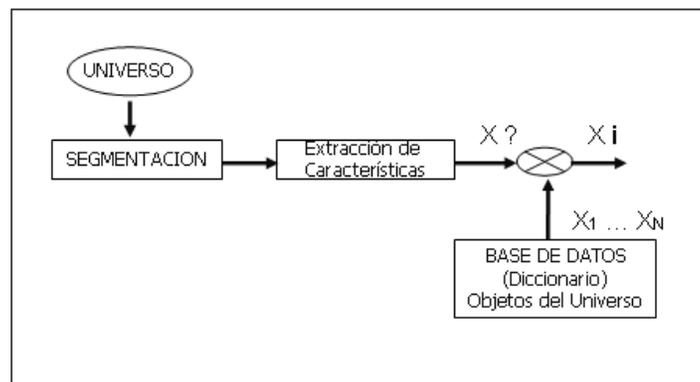


Figura 3.3.2 - Funcionamiento típico de un sistema de reconocimiento de formas.

De acuerdo con el esquema presentado en la figura anterior un sistema de reconocimiento de formas típico considera como primer elemento un *universo* de información, el cual para el caso de estudio de este proyecto está representado por los distintos frames obtenidos desde un dispositivo de captura, luego es necesario realizar la *segmentación* de las imágenes aislando aquellos elementos que constituyen nuevos objetos, el paso siguiente es extraer las características más representativas de estos objetos, finalmente el resultado de la extracción de características es evaluado y asociados con alguna de las diversas clases que están previamente contenidas en un repositorio (diccionario).

En este proyecto se han considerado métodos complementarios a los descritos en el esquema anterior y que fueron presentados en las etapas típicas de un proceso de visión artificial (figura 3.2.1), donde un primer cambio implica mejorar los resultados de la etapa de segmentación a través de un pre procesado de las imágenes (métodos que son parte importante del procesamiento de imágenes digitales, como suavizado, eliminación de ruido, realce de bordes, entre otros) y en la última etapa del esquema se reemplaza el “Diccionario” de objetos por una RNA con la

capacidad de clasificar de forma autónoma los tipos de objetos detectados en la escena, la entrada de la red son los vectores de características de cada uno de estos nuevos objetos detectados (sin existir dependencia de mantener un registro previo de todos los objetos a detectar).

El esquema final del sistema de reconocimiento de formas queda de la siguiente forma:

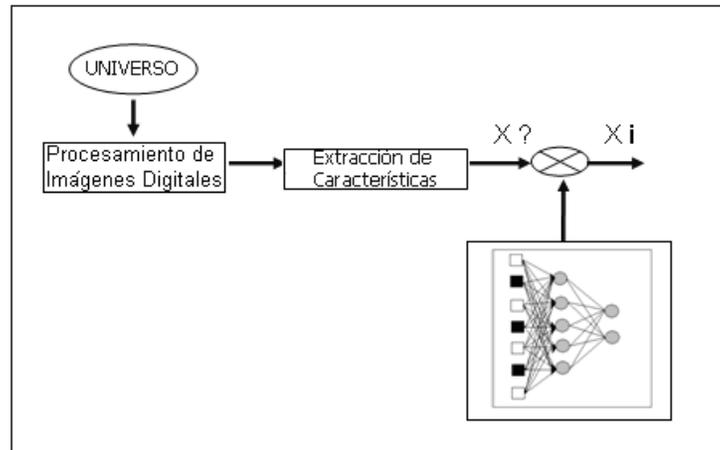


Figura 3.3.3 - Esquema final del sistema de reconocimiento de formas

Por lo tanto, el conjunto de algoritmos específicos implementados en el proceso de visión artificial construido son los siguientes:

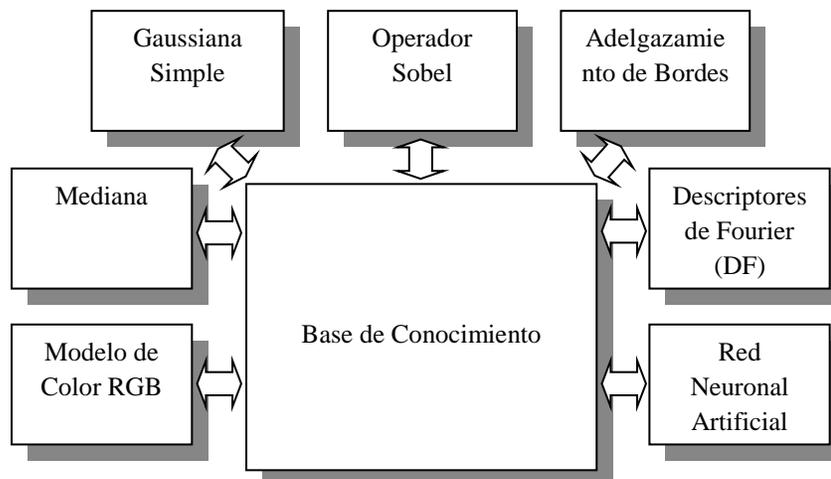


Figura 3.3.4 - Algoritmos del Proceso de Visión Artificial Inicial

La elección de estos métodos ha sido realizada de acuerdo al estudio de la literatura del área (reflejado en el estado del arte) y según la experiencia de aquellos que han entregado buenos resultados en implementaciones previas. Durante la etapa de implementación y pruebas, este conjunto de métodos es evaluado y modificado según la

conveniencia de incluir algoritmos alternativos que mejoren las falencias observadas en las pruebas iniciales (en las siguientes secciones se muestran estas implementaciones y resultados en detalle).

3.4 Procedimientos de investigación

Los resultados obtenidos en las pruebas de los algoritmos son medidos de acuerdo al resultado visual que entregan, verificando el nivel de representatividad que mantienen sobre los objetos presentes en la escena. La RNA implementada en cambio utilizará la validación cruzada como método de control y análisis de resultados, donde mediante la división de los datos de pruebas en subconjuntos se logra entrenar la red con muestras distintas a las utilizadas posteriormente en la evaluación de las clasificaciones correctas que realiza.

También son considerados gráficos que muestran el comportamiento de distintos parámetros de configuración evaluados en la estructura de la RNA, de esta forma se logra adaptar la red a la problemática particular de procesamiento que representa el caso de estudio, optimizando el número de capas, de neuronas y los valores de sus respectivas variables de configuración (valor de momento, punto de corte, etc.).

Respecto del entorno de las pruebas, se probaron varios escenarios para comprobar el comportamiento de los algoritmos, seleccionar el entorno de estudio y considerar futuras mejoras. Estas pruebas son reflejadas en el capítulo siguiente de este documento.

Respecto del tipo de imágenes utilizadas en la clasificación de objetos, se pudo observar la importancia de que todas las etapas de un proceso de visión artificial mantengan un alto grado de robustez ya que cada una es parte fundamental para lograr una buena interpretación de la escena, en este sentido se dan en el próximo capítulo las características de las imágenes que finalmente forman parte del grupo de análisis.

Debido a que se trabaja fundamentalmente en el área del procesamiento de imágenes digitales y visión artificial, con la finalidad de reflejar la organización del proceso de visión artificial según las características de cada uno de los métodos seleccionados, como parte del presente proyecto se propone el diseño de un framework que permita realizar las pruebas de cada algoritmo en particular y también del proceso en su totalidad. Pero ya que la construcción de una plataforma de trabajo que apoye el desarrollo de proyectos de visión artificial no forma parte de los objetivos principales de esta tesis, se limitará su desarrollo a una versión experimental que permita testear los algoritmos utilizados en este trabajo pero considerando el diseño de una arquitectura que permita evolucionar su desarrollo en futuros proyectos. El trabajo realizado en el diseño y construcción del Framework propuesto se adjunta para su consulta en Apéndice I.

3.5 Conclusión

En este capítulo se han abordados antecedentes acerca de la estructura tradicional de un sistema de visión artificial, exponiendo la metodología utilizada y explicando cada una de las etapas que deben ser consideradas. Se propuso el desarrollo de un caso de estudio que permite poner en práctica las técnicas estudiadas y luego se habló de la estructura típica de un proceso de reconocimiento de formas complementando su diseño con las etapas de un proceso de VA mencionadas anteriormente (figura 3.2.1). Los procedimientos seleccionados para la ejecución de la etapa de implementación y análisis de resultado fueron también analizados.

En los procedimientos de investigación expuestos se ha dado a conocer el marco de trabajo sobre el cual se desarrolla el siguiente capítulo de este documento. También se ha comentado del desarrollo del Apéndice I, donde se propone el diseño de un framework de visión artificial que apoye el desarrollo de nuevas problemáticas del área.

Implementación y Análisis de Resultados

En este capítulo se detalla la implementación del proceso de visión artificial realizada y se analizan los resultados obtenidos durante las pruebas.

4.1 Introducción

En el capítulo anterior se describió la metodología adoptada para implementar los algoritmos seleccionados y se mencionaron los procedimientos a utilizar en su construcción y análisis de resultados. En este capítulo se muestra el detalle de las fases finales de la investigación, se detallan las implementaciones realizadas y se analizan los resultados obtenidos. Sin embargo, la exposición de conclusiones finales se trata en el capítulo siguiente.

El contenido del presente capítulo está organizado de acuerdo a las siguientes secciones:

4.2 Entorno y tipo de imágenes utilizadas: Se estudian los entornos en los cuales el proceso de visión artificial inicialmente implementado entrega buenos resultados y se observan las variables que afectan su desempeño. En base a esta información se decide por aquel entorno que entrega mejores condiciones de prueba para continuar con la implementación y análisis de los algoritmos. Los tipos de imágenes también son analizados en esta sección exponiendo las características que tiene el conjunto de prueba a utilizar.

4.3 Implementación de algoritmos seleccionados: Se explica la implementación realizada del conjunto inicial de algoritmos seleccionados y se verifica el desempeño de cada uno de los métodos del proceso diseñado. Se busca observar el comportamiento de cada algoritmo identificando posibles falencias que afecten el resultado obtenido. Esta fase de la construcción del proceso tiene como objetivo comprobar que cada una de las etapas del proceso cumpla con su objetivo y entregue una salida óptima (ya que esta será utilizada en la etapa posterior).

4.4 Análisis de algoritmos alternativos: Esta fase está enfocada en estudiar la incorporación de algoritmos al conjunto seleccionado inicialmente. Esta tarea depende directamente de los resultados obtenidos en la fase anterior y consiste en analizar alternativas que puedan reemplazar o mejorar alguno de los algoritmos que no estén entregando los resultados que se esperaban.

4.5 Pruebas finales del proceso de visión artificial: Consiste en la incorporación o perfeccionamiento de alguno(s) de los algoritmos utilizados en el proceso de visión artificial construido inicialmente, se basa en los resultados del análisis realizado en la sección anterior. En estas pruebas se ejecuta el conjunto final de algoritmos de procesamiento de imágenes implementado y se comparan los resultados obtenidos con distintos parámetros de ejecución. En esta fase se espera detectar aquellas configuraciones que entreguen los mejores resultados de acuerdo al caso de estudio seleccionado (valores de las variables que posee cada método, como por ejemplo en la RNA son el número de capas ocultas, del número de neuronas por cada capa, factor momento, entre otros.).

4.6 Análisis del desempeño final: Considera la ejecución del proceso definitivo de visión artificial, tomando como entradas imágenes desde un conjunto de muestras diferente al utilizado en las pruebas de la etapa anterior de entrenamiento, la idea fundamental es visualizar el desempeño de los algoritmos y del proceso en general frente a escenas desconocidas que no formaron parte del ajuste del sistema. Se espera validar la robustez de la solución construida y analizar aquellos puntos en donde es necesario investigar optimizaciones específicas para lograr una buena adaptabilidad ante diversos entornos de aplicación.

En las páginas siguientes de este documento se expone el detalle de cada una de las secciones mencionadas anteriormente y se analizan los resultados obtenidos en cada una.

4.2 Entorno y Tipo de Imágenes Utilizadas

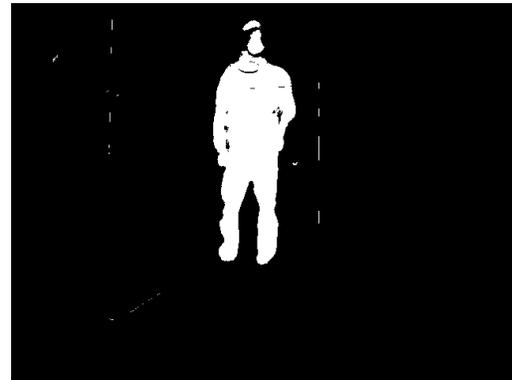
Para lograr la correcta detección de nuevos objetos dentro de una escena monitoreada, además de aspectos como las características de los algoritmos a implementar se deben considerar las condiciones en donde funcionarían estos métodos, a continuación se describen las pruebas realizadas sobre algunos tipos de entornos (interiores, exteriores, mixtos) y se acota el estudio sobre alguno de ellos indicando las características que poseen las imágenes obtenidas desde la escena monitoreada.

Entorno

Inicialmente se espera que un sistema de estas características pueda funcionar tanto en ambientes de interiores como de exteriores o al aire libre, pero dado las variables existentes en estas dinámicas condiciones el presente proyecto acota las pruebas a un ambiente específico que permita disminuir los factores externos que perjudican la clasificación alcanzada por la RNA (un funcionamiento más adaptable a los continuos cambios que se presentan en diversos ambientes es analizado en las propuestas de trabajos futuros que se realiza en el capítulo final de este documento).

La siguiente tabla describe los principales inconvenientes observados en diversos entornos de monitoreo estudiados:

	
<p>Características Entorno N° 1</p> <p>Espacio interior con iluminación artificial.</p> <p>Principales Inconvenientes</p> <p>Debido a que se trata de un espacio interior reducido la luz artificial influye sobre las tonalidades de los objetos visualizados en las imágenes, ante la presencia de un nuevo objeto se producen cambios por la sombra que este genera, y también por el rebote de los rayos de luz que se producen ya que cambian algunas de las tonalidades de objetos que ya eran parte de la escena generando de esta forma falsas detecciones.</p>	
	
<p>Características Entorno N° 2</p> <p>Espacio exterior con iluminación natural (día soleado).</p> <p>Principales Inconvenientes</p> <p>La luz natural que llega a la escena no mantiene su intensidad y genera cambios leves pero constantes en las tonalidades de los elementos presentes en la escena, esto provoca un porcentaje importante de ruido en la segmentación realizada.</p>	



Características Entorno N° 3

Espacio exterior-interior con iluminación natural-artificial (día soleado).

Principales Inconvenientes

Se repiten los inconvenientes de ambos tipos de espacios (exterior-interior) pero por tratarse en una zona mixta los efectos en las tonalidades de las imágenes cambian dependiendo el entorno más próximo a la ubicación del nuevo objeto.

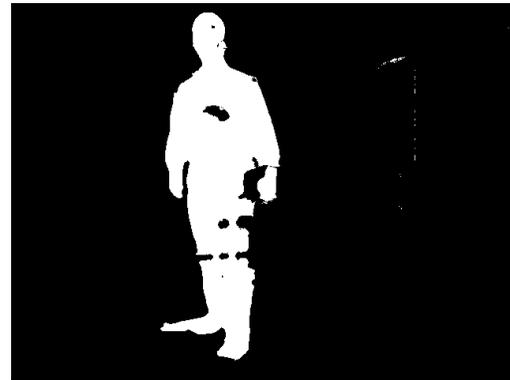


Características Entorno N° 4

Espacio exterior con iluminación natural (día nublado).

Principales Inconvenientes

Se presentan los mismos inconvenientes mencionados en otras imágenes de espacios exteriores pero además se observa que la vegetación es un generador de cambios dentro de los objetos del fondo, pues las hojas de los arbustos se mueven con el viento cambiando ligeramente su ubicación dentro de la escena afectando al fondo modelado.



Características Entorno N° 5

Espacio interior con iluminación artificial-natural.

Principales Inconvenientes

La luz que ingresa por las ventanas puede cambiar las tonalidades del entorno, también se presenta el inconveniente de los elementos del fondo de la escena que tienen colores similares al del nuevo objeto detectado, esto interfiere en la interpretación de su forma realizada por el sistema (pero se trata de un inconveniente que también puede presentarse en cualquier otro tipo de entorno monitoreado).

Tabla 4.2.1 – Tipos de entornos y sus principales inconvenientes

Como se ha presentado en la tabla anterior, en las pruebas realizadas sobre diversos escenarios se han observado algunos de los factores que provocan inconvenientes en la interpretación lograda por un sistema de visión artificial. Los métodos a utilizar en ambientes cambiantes para contener esta situación representan nuevas líneas de investigación de gran relevancia para ser afrontadas en futuras investigaciones, esto será abordado en la sección respectiva de este documento (propuestas de trabajos futuros).

Para realizar las pruebas de los algoritmos seleccionados en este trabajo se ha optado por el último de los entornos presentados en la tabla anterior (entorno N° 5), de esta manera cada algoritmo es configurado de acuerdo a las condiciones existentes en este entorno. A continuación se detallan las principales características de la escena seleccionada:

	<p>Tipo de entorno: Interior</p> <p>Tipo de Iluminación: Artificial – Natural.</p> <p>Luces en el cielo de la habitación y luz natural que ingresa por las ventanas.</p> <p>Dimensiones: 4 metros de ancho, 7 metros de largo.</p> <p>Características: Sala de capacitaciones donde los elementos típicos que pueden formar parte del fondo de la escena son sillas, escritorios, bolsos, además de algunos archivadores.</p>
---	---

Tabla 4.2.2 – Principales características del entorno seleccionado.

Tipos de Imágenes Utilizadas

Respecto de las imágenes utilizadas, de acuerdo al entorno seleccionado a continuación se detallan las características que posee el conjunto de imágenes de pruebas generado:

- Dimensiones: 640x480
- Resolución horizontal: 96 ppp
- Resolución vertical: 96 ppp
- Profundidad en bits: 24

Las imágenes utilizadas son obtenidas desde secuencias de videos obtenidas previamente en el entorno monitoreado, luego de un proceso off-line de captura de algunos frames de la secuencia que contienen imágenes del fondo de la escena y otras con la presencia de nuevos elementos (del tipo que se esperan clasificar: Personas).

En las pruebas realizadas se ha podido observar que existen algunas imágenes que se ven afectadas en sus tonalidades de forma importante por dos componentes principales:

El brillo: El cual provoca que los objetos varíen su tonalidad original y en ocasiones se confundan con objetos del fondo de la escena.

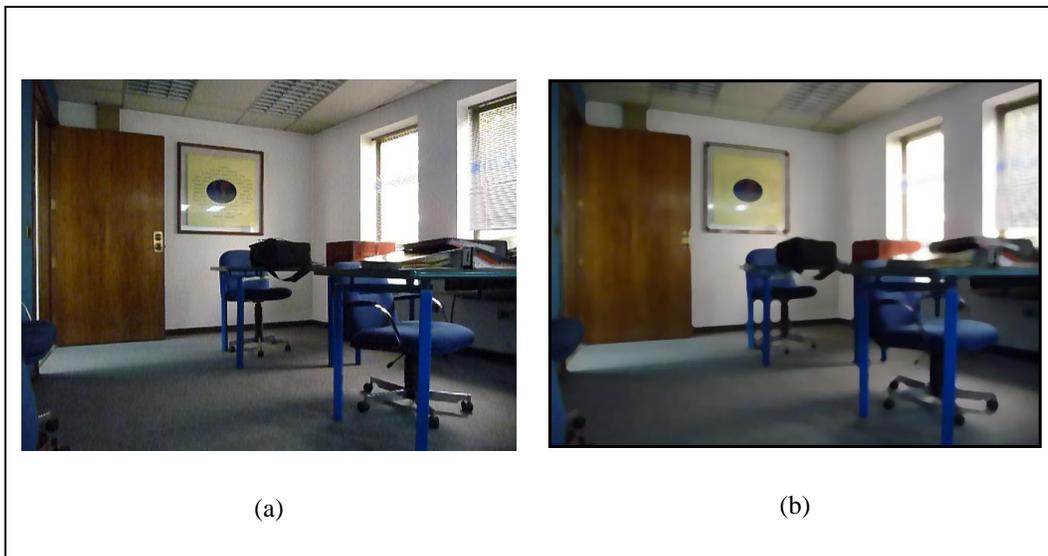
Las sombras: Las cuales producen que los algoritmos de procesamiento de imágenes digitales consideren como extensión de las dimensiones de un objeto las sombras que estos generan.

La presencia de estos componentes depende del entorno desde donde se obtienen las imágenes procesadas y como se ha mencionado en este proyecto se ha optado por acotar el conjunto de muestras utilizadas priorizando el análisis de los algoritmos implementados, centralizando así los esfuerzos en evaluar y mejorar el desempeño de estos. Las vías de solución de estos inconvenientes son analizadas con mayor detalle en la sección correspondiente de este documento (propuestas de trabajos futuros).

4.3 Implementación de Algoritmos Seleccionados

Etapa de Adquisición: RGB. Se procesan imágenes en el modelo de color RGB, que como se ha explicado en el estado del arte, representa las tonalidades de una imagen en tres bandas o canales de color (Rojo, Verde, Azul). Este es el modelo de color por defecto del conjunto de imágenes obtenidas.

Etapa de Pre-Procesamiento: Mediana. Con la finalidad de eliminar el ruido presente en la imagen se realiza sobre ella un filtrado de la Mediana, ya que a diferencia de otros métodos de suavizado no difumina los bordes u otros detalles importantes en la imagen. Los resultados obtenidos con esta técnica son los siguientes:



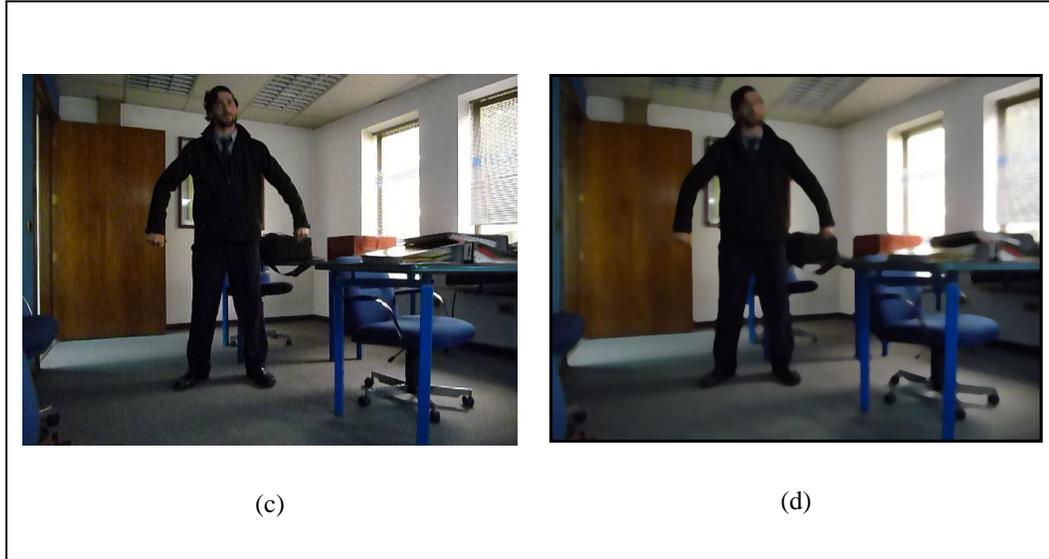


Figura 4.3.1 - Resultados filtro de la Mediana (a) Imagen original 1 (b) Resultado de aplicar el filtro de la mediana en imagen 1(c) Imagen original 2 (d) Resultado de aplicar el filtro de la mediana en imagen 2.

Pese al emborronamiento que como se puede observar provoca este método, luego de diversas pruebas ejecutadas sobre las imágenes originales y sobre las mismas luego de ser suavizadas con el filtro de la Mediana se ha comprobado que la homogenización que se logra sobre las tonalidades existentes en la escena ayuda notablemente en la posterior segmentación de la imagen, esto debido a que se logra normalizar los valores que poseen los píxeles de un mismo objeto.

Etapa de Segmentación: Gaussiana Simple. Para clasificar los objetos presentes en una escena como “objeto de fondo” u “objeto de primer plano” se ha seleccionado el método Gaussiana Simple, de esta forma se modelan los objetos pertenecientes al fondo de la escena y ante una nueva imagen permite la detección de aquellos píxeles que se escapan al modelo y por lo tanto deben ser identificados como parte del objeto de primer plano. A continuación se muestran los resultados de la etapa de segmentación utilizando este método:

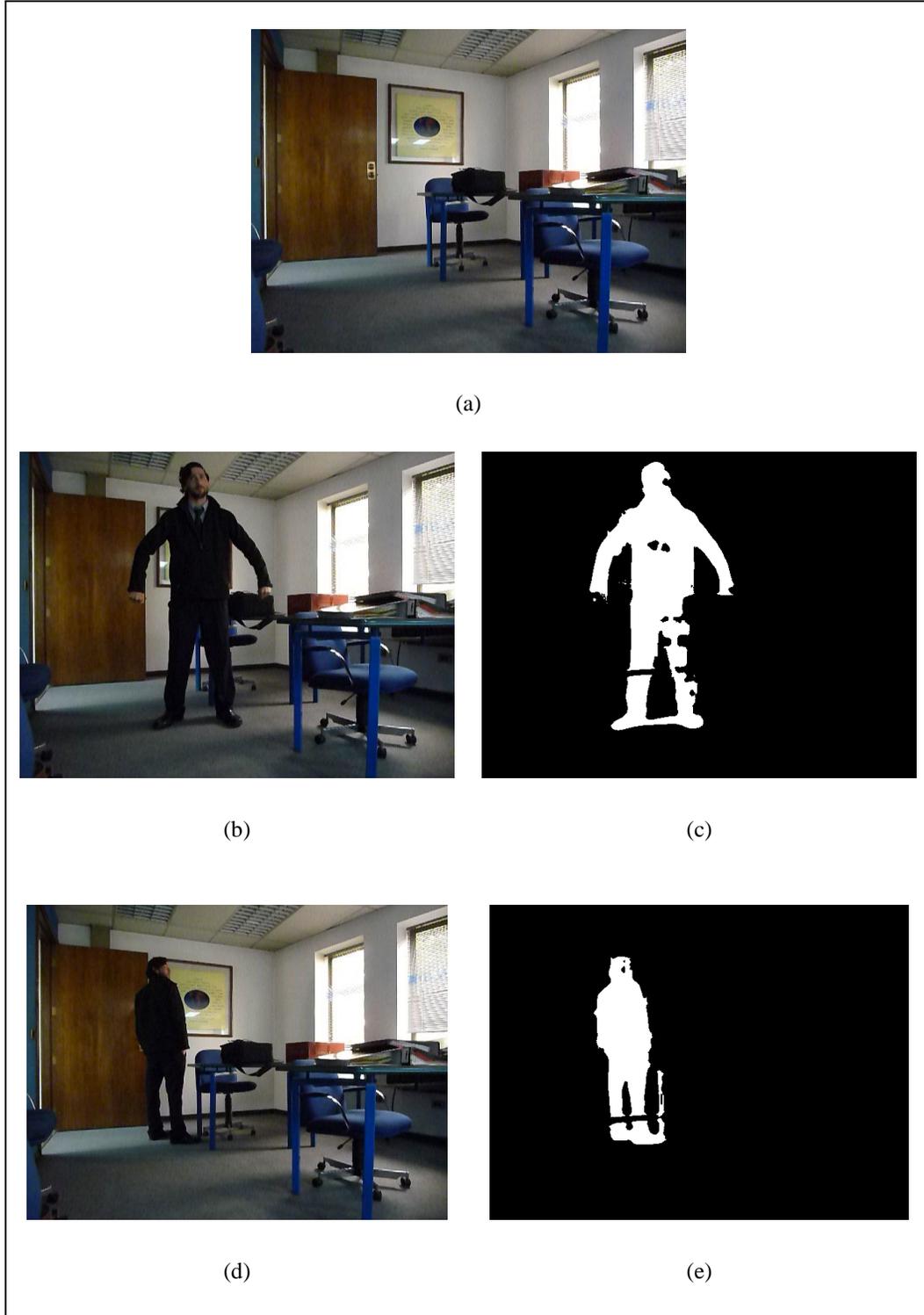
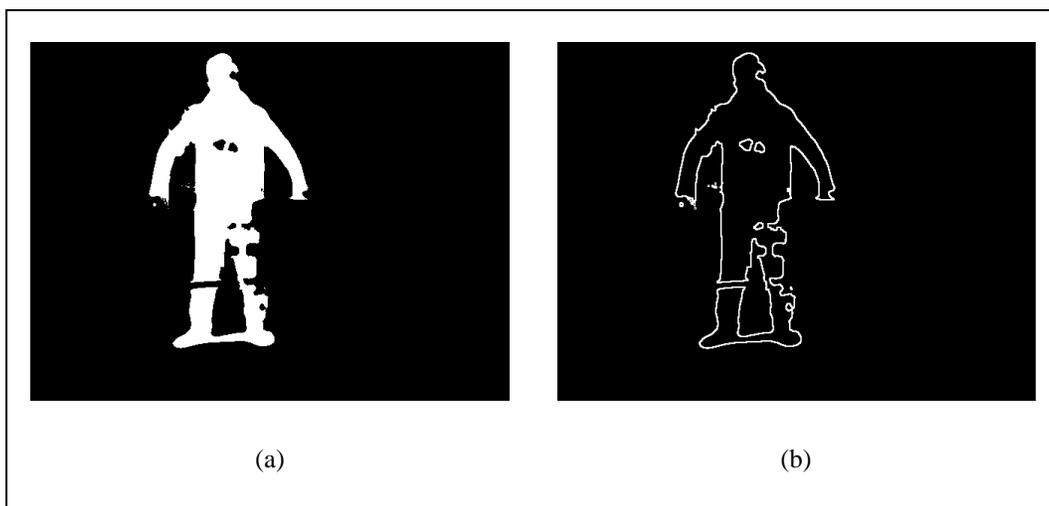


Figura 4.3.2 - Resultados Gaussiana Simple (a) Ejemplo de una de las imágenes utilizada para Modelar el Fondo (b) Imagen Original 1 (c) Resultado de segmentación con Gaussiana Simple de imagen 1 (d) Imagen Original 2 (e) Resultado de segmentación con Gaussiana Simple de imagen 2.

La primera imagen (a) de la secuencia anterior corresponde a una de las capturas del fondo de la escena, la cual en conjunto con otras muestras se utiliza para modelar el fondo de acuerdo con la lógica de este algoritmo explicada en el estado del arte. Como se puede observar, las imágenes con objetos nuevos en la escena (imágenes (b) y (d) de la secuencia anterior) entregan una segmentación que presenta algunas discontinuidades en su figura, esto se debe principalmente al “brillo” y la “sombra” proyectada sobre la escena por el nuevo objeto, que provoca tonalidades que no se corresponden con el fondo previamente modelado y por lo tanto son consideradas parte de un objeto de primer plano. Otro de los factores que afecta el desempeño de la segmentación son las similitudes de colores en los objetos adyacentes al primer plano, ya que no se genera demasiada variación respecto a las tonalidades de las imágenes utilizadas para construir el modelo de fondo, por lo tanto esta característica produce que algunos de los píxeles del nuevo objeto sean considerados como parte del fondo de la escena (en la secuencia de imágenes de la figura anterior, el color del pantalón oscuro se asemeja a elementos del fondo como el cubre piso y el bolso negro).

Estos resultados obtenidos serán considerados en la fase de análisis de algoritmos alternativos con el objetivo de lograr mejorar el resultado de la segmentación.

Etapa de Representación y Descripción: Operador de Sobel, Adelgazamiento de Bordes y Descriptores de Fourier (DF). Para representar los objetos detectados en la etapa de segmentación se ha optado por características del contorno del objeto. La segmentación del objeto de primer plano es procesada con una máscara de Sobel para lograr la detección de bordes, pero debido a que los bordes resaltados por este operador en su mayoría presentan un ancho mayor a un píxel, se debe re-procesar el resultado para lograr el adelgazamiento de estos, lo anterior con la finalidad de entregar una fuente de información óptima para el posterior cálculo de los DF del objeto. La utilización de los DF ha sido considerada principalmente debido a las ventajas que se pueden obtener al trabajar sobre el dominio de la frecuencia (invariancia a Traslaciones, Rotación y Escalado). Los resultados obtenidos en esta etapa se muestran a continuación:



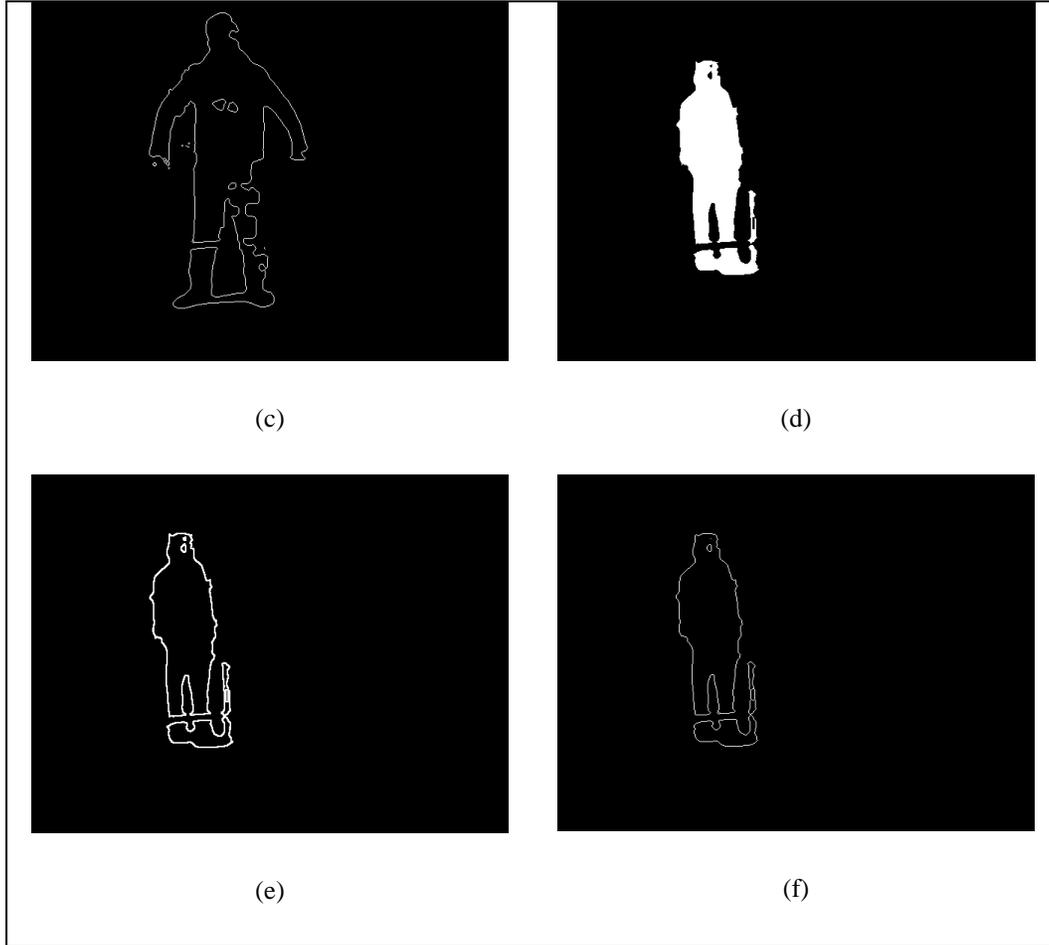


Figura 4.3.3 – Resultados Representación y Descripción. (a) Resultado Segmentación imagen 1 (b) Resultado Operador de Sobel en imagen 1 (c) Resultado Adelgazamiento de Bordes en imagen 1 (d) Resultado Segmentación imagen 2 (e) Resultado Operador de Sobel en imagen 2 (f) Resultado Adelgazamiento de Bordes en imagen 2.

Como se puede observar en las imágenes anteriores, se ha logrado representar el contorno de un objeto de primer plano a través de la mínima cantidad de píxeles que no rompen la conectividad del borde detectado (con ancho de 1 píxel), esto es necesario debido a que los descriptores de Fourier sirven para representar contornos cerrados.

Se puede deducir de las imágenes anteriores que los efectos producidos por las sombras y brillo heredados de la etapa de segmentación anterior son determinantes en los resultados de esta y las siguientes etapas de procesado. Una limitante que se produce en la etapa de representación y descripción es la discontinuidad del objeto debido a los factores ya mencionados (brillo, sombra), de esta forma antes de ejecutar el siguiente algoritmo de clasificación de objetos (RNA con BP) es necesario comenzar la fase de “Análisis de Algoritmos Alternativos” para mejorar la

segmentación y representación/descripción de los objetos de primer plano y así generar una adecuada fuente de información para el buen desempeño del algoritmo de clasificación (RNA).

4.4 Análisis de algoritmos Alternativos

En esta sección se consideran las debilidades detectadas en el conjunto inicial de métodos implementados para el proceso de visión artificial, por lo que se hace una revisión de la literatura del área con el objetivo de analizar nuevos métodos de procesamiento de imágenes digitales que permitan mejorar los resultados en la detección de objetos de primer plano. En este sentido, se estudian técnicas que aunque no tienen como objetivo principal la segmentación de objetos, debido a las características de procesamiento que poseen pueden aportar importantes mejoras a esta etapa del proceso (principalmente se revisará la “morfología matemática” y los “modelos de color”).

Es importante considerar también, que si bien en esta etapa de las pruebas se busca implementar nuevos algoritmos de procesamiento, el objetivo principal es complementar los métodos ya existentes y no reemplazar en su totalidad el proceso de visión artificial estudiado y definido inicialmente. A continuación se describen los resultados de las pruebas realizadas sobre los algoritmos que como se detectó pueden contribuir en la generación de una fuente óptima de información para el algoritmo de clasificación de objetos utilizado (RNA).

Modelo de Color HSV: Este modelo se ha incluido en el estado del arte del presente proyecto ya que dentro de la literatura del área se resalta su conveniencia para obtener mejoras en los resultados de la visión artificial. La gran ventaja que otorga este modelo es que permite adaptar el procesamiento de imágenes digitales a las características de visión que posee el sistema visual humano. Corresponde a una transformación no lineal del modelo RGB y representa los colores en base a componentes de Tonalidad, Saturación y Brillo (Hue, Saturation, Intensity). En la imagen siguiente (4.4.1) se muestra el resultado de su aplicación sobre imágenes del caso de estudio:

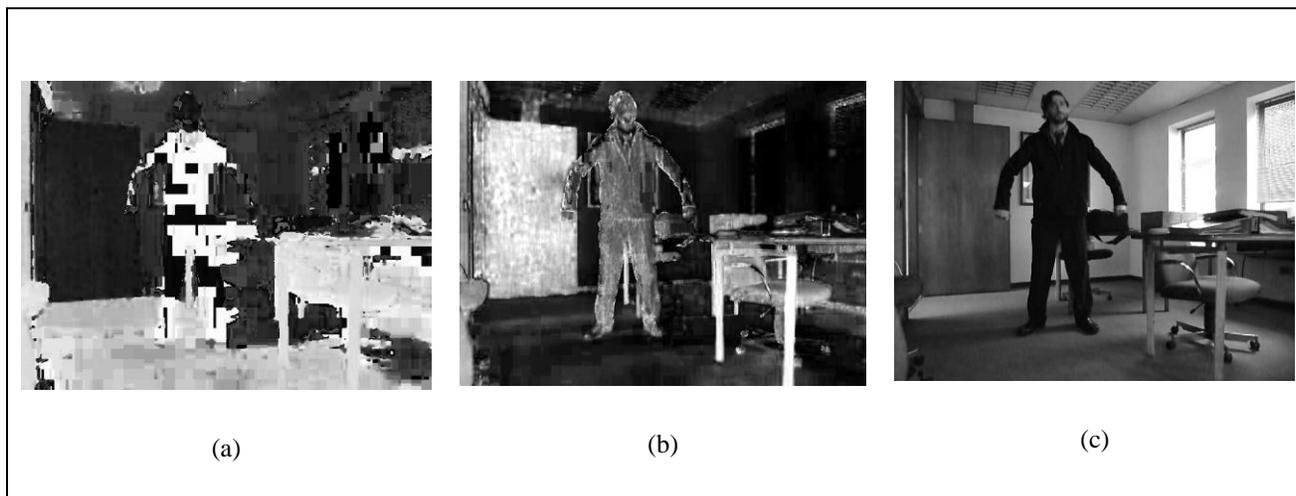


Figura 4.4.1 - Representación de una imagen del caso de estudio en el modelo de color HSV (a) Componente de Tonalidad (b) Componente de Saturación (c) Componente de Brillo.

Este modelo reemplaza el tradicional RGB, entonces se hace necesario realizar una conversión entre estos dos modelos de color (ya las imágenes utilizadas han sido capturadas en modelo RGB). Se utilizará en el proceso de visión artificial construido sólo la componente de saturación, eliminando las componentes de tonalidad y brillo para realizar de esta manera la segmentación de la escena sobre esta única componente, la razón de utilizar sólo la componente de Saturación es que debido al conocimiento empírico obtenido al iterar y visualizar los resultados de todas las componentes sobre un conjunto de imágenes de prueba, se ha podido observar que la componente de “Tonalidad” produce notorias variaciones entre imágenes que no presentan cambios significativos en su contenido y también porque la componente de Brillo se ve perjudicada en un mayor porcentaje de píxeles por la sombra y brillo presentes en la escena, entonces no se considera que estas componentes representen una buena entrada para ser utilizada en la etapa de segmentación.

En la figura 4.4.2 se pueden ver algunos de los mejores resultados obtenidos en la etapa de segmentación utilizando el modelo de color HSV (acotado a la componente de saturación de la imagen).

El algoritmo de la “Gaussiana Simple” (GS) tiene como entradas adicionales a la imagen original suavizada los siguientes: un parámetro de clasificación de píxeles (K , que se considera en la regla de pertenencia de un píxel al fondo de la escena o al objeto de primer plano), un parámetro de absorción (P , que determina la velocidad de actualización del modelo) y un parámetro que indica la varianza inicial a considerar en la obtención del modelo de fondo (VI , considerado con la primera imagen de fondo procesada). Estos parámetros se han incluido en la figura 4.4.2 para comparar los resultados obtenidos al realizar variaciones en los valores de estos.



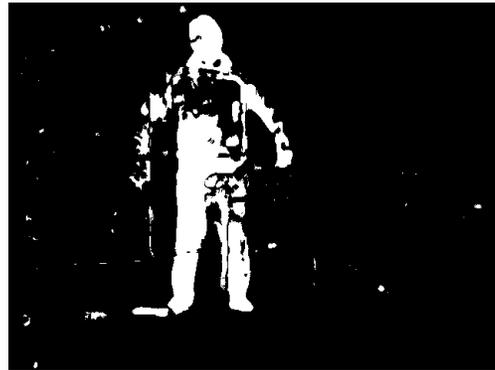
(a)



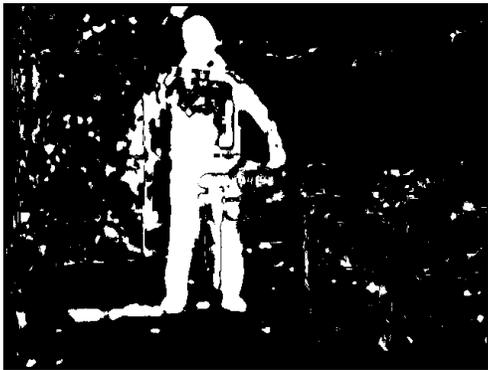
(b) $K=1, P=0.01, VI=10$



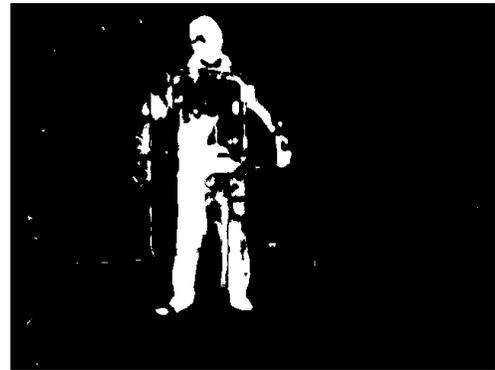
(c) $K=2, P=0.01, VI=10$



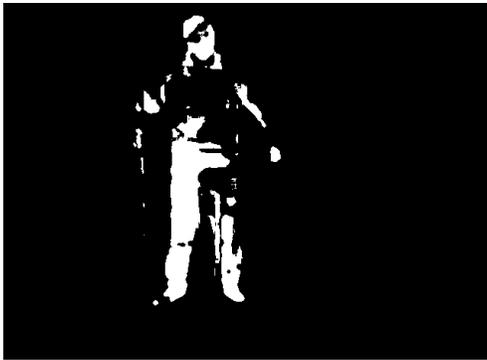
(d) $K=3, P=0.01, VI=10$



(e) $K=1, P=0.01, VI=20$



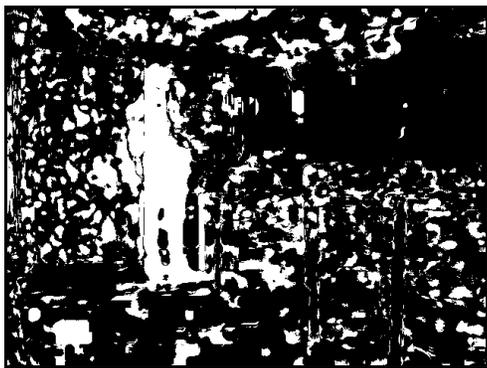
(f) $K=2, P=0.01, VI=20$



(g) $K=3$ $P=0.01$ $VI=20$



(h)



(i) $K=1$ $P=0.01$ $VI=10$



(j) $K=2$ $P=0.01$ $VI=10$



(k) $K=3$ $P=0.01$ $VI=10$



(l) $K=1$ $P=0.01$ $VI=20$

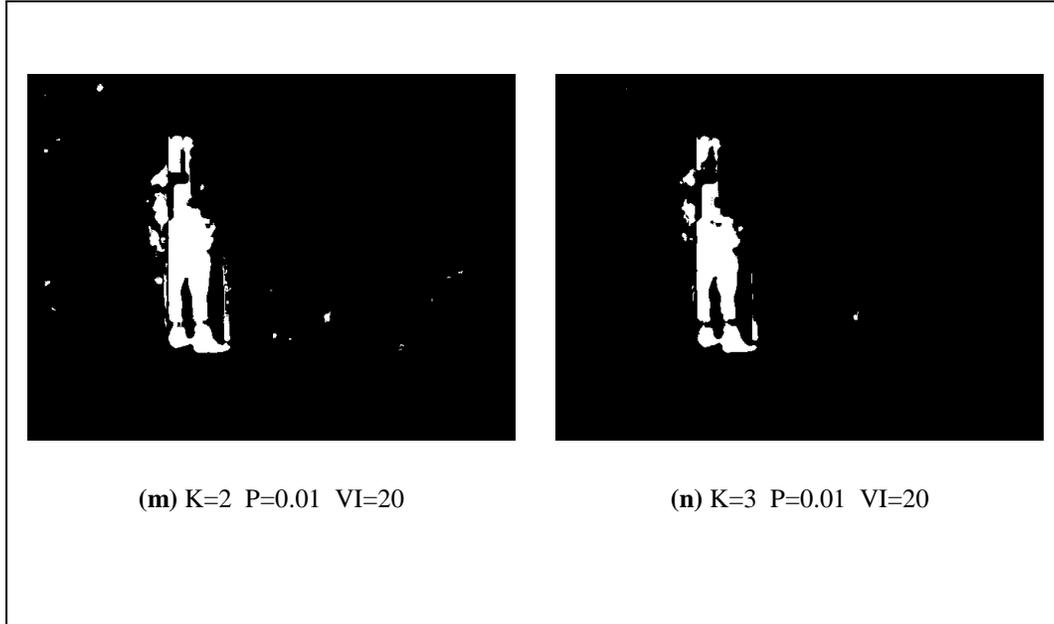


Figura 4.4.2 – Resultado segmentación utilizando modelo de color HSV. (a) Imagen original 1 (b), (c), (d), (e), (f), (g) Muestran resultados de GS sobre HSV en imagen 1 según variación de parámetros K, P y VI (h) Imagen original 2 (i), (j), (k), (l), (m), (n) Muestran resultados de GS sobre HSV en imagen 2 según variación de parámetros K, P y VI.

Se puede observar en la secuencia de imágenes de la figura anterior, que al trabajar sobre la componente de saturación de las imágenes se logra disminuir en gran cantidad los efectos que se producían en la segmentación con RGB producto de las sombras generadas por el objeto de primer plano, sin embargo ha aumentado el ruido y las discontinuidades de los objetos detectados. Esto último hace que la inclusión de sólo este método no represente mejoras significativas respecto del resultado inicial entregado por la etapa de segmentación.

Morfología Matemática: Otros de los métodos a evaluar dentro de un proceso de visión artificial son aquellos que están dentro del área de la morfología matemática (Erosión y Dilatación fundamentalmente). Estos serán aplicados sobre las imágenes resultantes de la segmentación hecha con Gaussian Simple sobre la componente de Saturación.

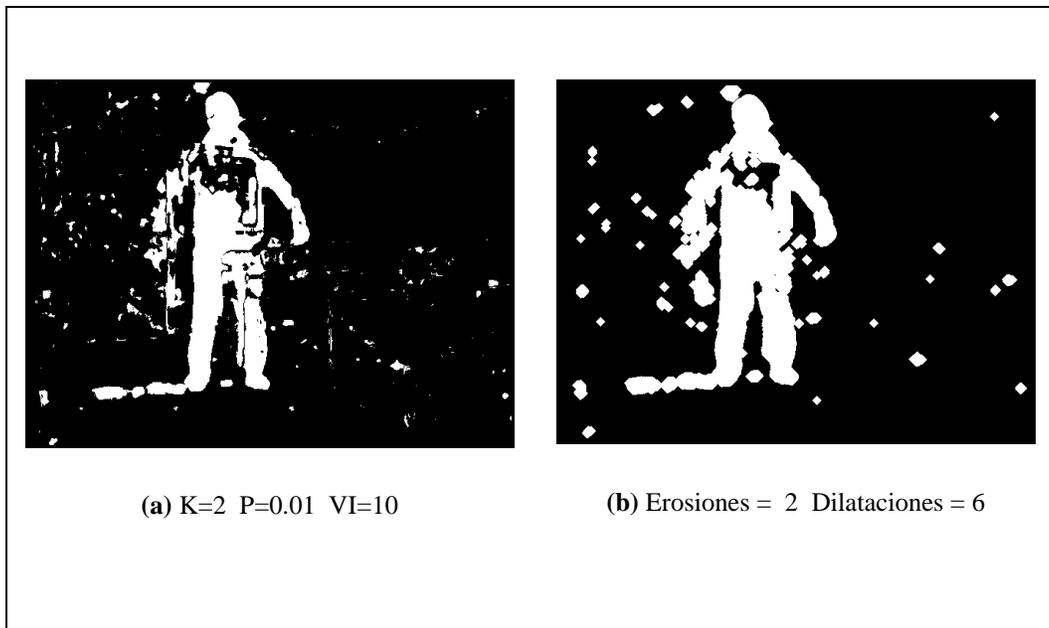
A través de un proceso iterativo de pruebas se logró descubrir que para aplicar la morfología matemática y contribuir a la mejora de los resultados del caso de estudio abordado se debían considerar los siguientes pasos:

1. Someter la imagen segmentada a un proceso de Erosión para eliminar el ruido presente en la imagen.

2. Someter la imagen erosionada a un proceso de Dilatación para rellenar regiones afectadas con la pérdida de píxeles y además conectar regiones pertenecientes a un mismo objeto.

Siguiendo la secuencia planteada se logra mejorar la conectividad de los píxeles pertenecientes a un mismo objeto y eliminar aquellos que forman parte de regiones demasiado pequeñas para ser clasificadas como objetos de primer plano (generalmente corresponden a ruido). En el primer paso se logra eliminar una gran cantidad de píxeles aislados existentes en la imagen segmentada, estos píxeles representan el ruido de la imagen y son eliminados al igual que aquellas regiones en donde el número de píxeles que contienen es demasiado pequeño y por lo tanto no corresponde su clasificación como objetos de primer plano (la cantidad de regiones pequeñas eliminadas depende del número de iteraciones que se definan para el proceso de Erosión). El segundo paso realiza la dilatación de las imágenes resultantes del proceso de erosión anterior, así se logra que un objeto de primer plano recupere la conectividad de aquellos píxeles que debido al brillo presente en la imagen (u otros factores como por ejemplo un excesivo número de erosiones en paso anterior) se encuentren separados o mantengan una conectividad muy limitada que impida la correcta representación de su forma real (p.e. segmentación realizada en pierna izquierda de la persona en figura 4.4.2, imagen 1).

A continuación se muestran algunos de los mejores resultados obtenidos al aplicar morfología matemática sobre las imágenes resultantes de la etapa de segmentación (sobre modelo de color HSV):

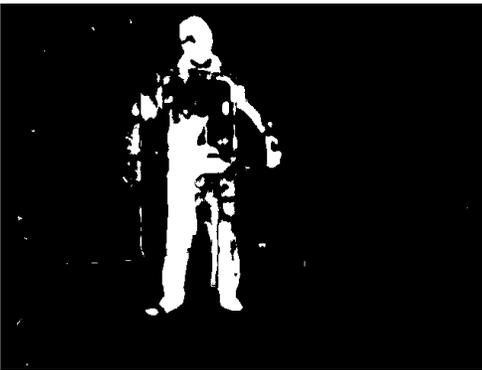




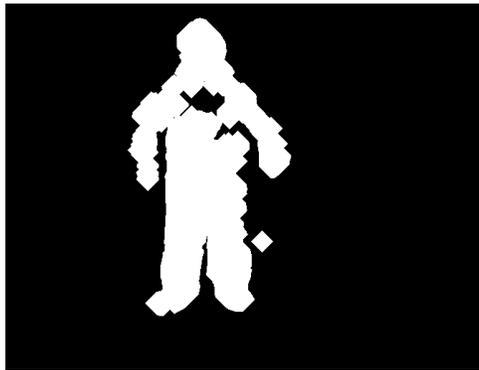
(c) $K=2$ $P=0.01$ $VI=10$



(d) Erosiones = 2 Dilataciones = 6



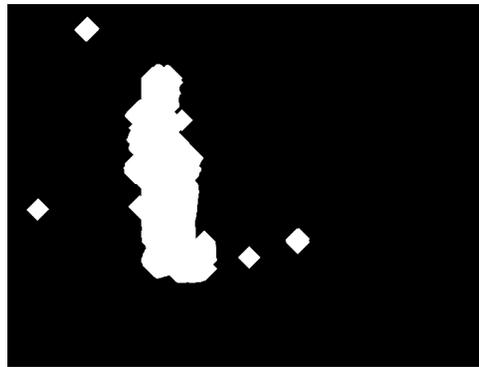
(e) $K=2$ $P=0.01$ $VI=20$



(f) Erosiones = 2 Dilataciones = 15



(g) $K=2$ $P=0.01$ $VI=20$



(h) Erosiones = 2 Dilataciones = 15

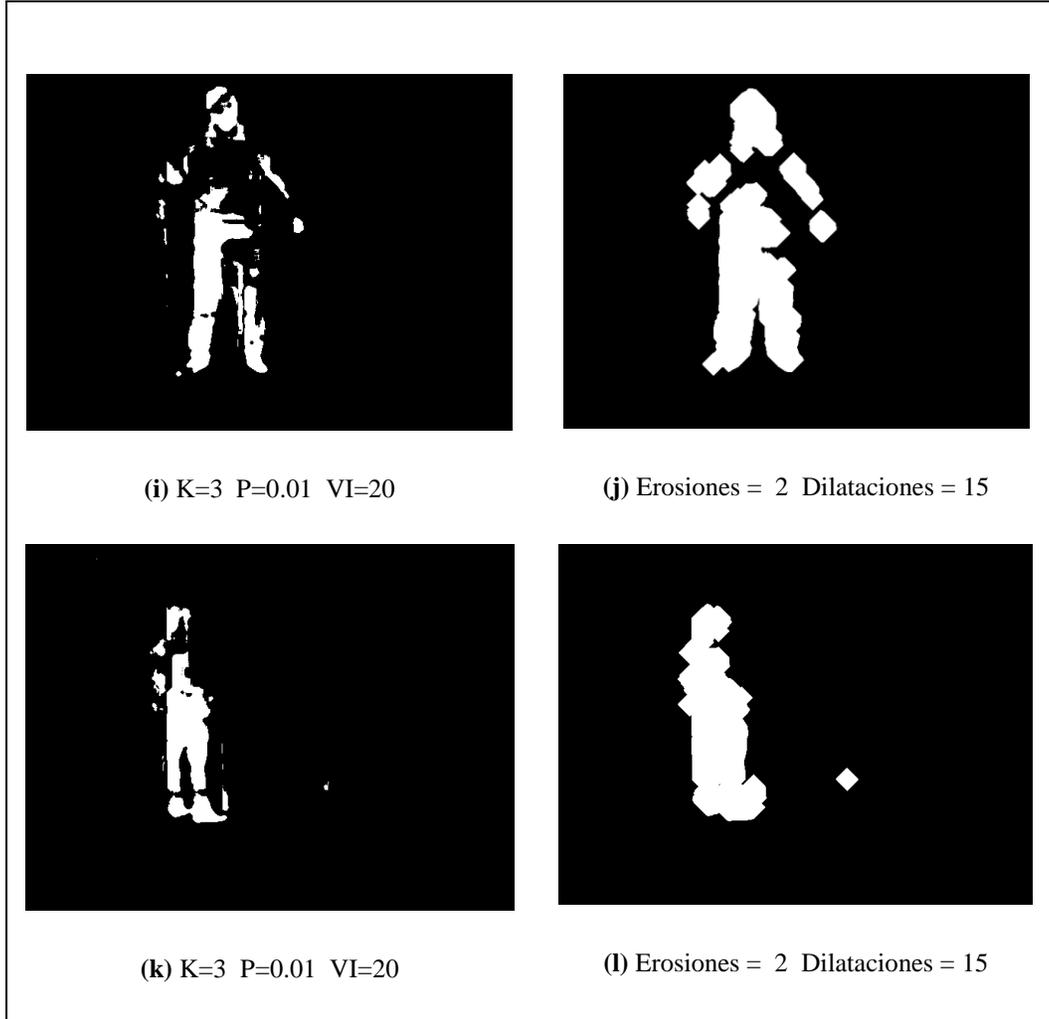


Figura 4.4.3 – Resultados morfología matemática sobre modelo de color HSV. Imágenes (a), (c), (e), (g), (i), (k) son resultado de la segmentación con GS y (b), (d), (f), (h), (j), (l) son el resultado de aplicar morfología matemática sobre la imagen segmentada correspondiente (imagen del lado izquierdo).

La morfología matemática aplicada sobre las salidas de la etapa de segmentación permite disminuir la pérdida de conectividad entre píxeles que pertenecen a un mismo objeto y logra eliminar parte del ruido presente en la imagen, pero como se puede observar en las imágenes anteriores, el número de discontinuidades creadas por el método de segmentación sobre HSV genera la necesidad de realizar un número elevado de dilataciones de los objetos para lograr reconstruir su figura inicial, recuperando la conectividad pero al mismo tiempo generando un ensanchamiento excesivo de las regiones de los objetos detectados, este ensanchamiento provoca en algunos casos que las figuras pierdan su forma original sin asegurar la conectividad total de sus píxeles.

Se puede concluir que HSV es recomendable para eliminar las sombras que producen los objetos presentes en una imagen pero se debe profundizar en la investigación de métodos complementarios que impidan la pérdida de conectividad observada entre píxeles de un mismo objeto. Respecto de la aplicación de morfología matemática se pudo verificar que esta ayuda en la eliminación del ruido pero es necesario un pre-procesado de la imagen segmentada para disminuir el número de falsos objetos de primer plano detectados (regiones con cantidades pequeñas de píxeles que no alcanzan a ser eliminados con la Erosión). También se pudo ver que la iteración del proceso de Dilatación sobre una imagen segmentada ayuda a conectar aquellos píxeles que se encuentren próximos entre sí, pero luego de algunas iteraciones del proceso se crea un efecto de ensanchamiento sobre la figura inicial de los objetos. Cuando se decida utilizar morfología matemática en un proceso de visión artificial habrá que tener en cuenta las características que se quieran resaltar de los objetos y determinar en base a esta información y algunas pruebas sobre imágenes representativas el número de iteraciones a considerar.

4.5 Pruebas Finales del Proceso de Visión Artificial

De acuerdo con los resultados observados en el estudio de algoritmos alternativos (HSV y Morfología Matemática) finalmente se ha optado por continuar trabajando sobre el modelo de color RGB debido a que es donde la etapa de segmentación genera menor cantidad de ruido. También se ha logrado visualizar el aporte que realizan las técnicas de morfología matemática sobre la reconstrucción de la forma real de los objetos, por lo tanto las imágenes segmentadas en la primera fase de las pruebas (sobre modelo de color RGB), serán pos-procesadas a través de métodos de Erosión y Dilatación para tratar el ruido y mejorar la conectividad entre los píxeles de un mismo objeto de primer plano.

El proceso final de visión artificial implementado estará compuesto con los métodos que se muestran en la siguiente figura:

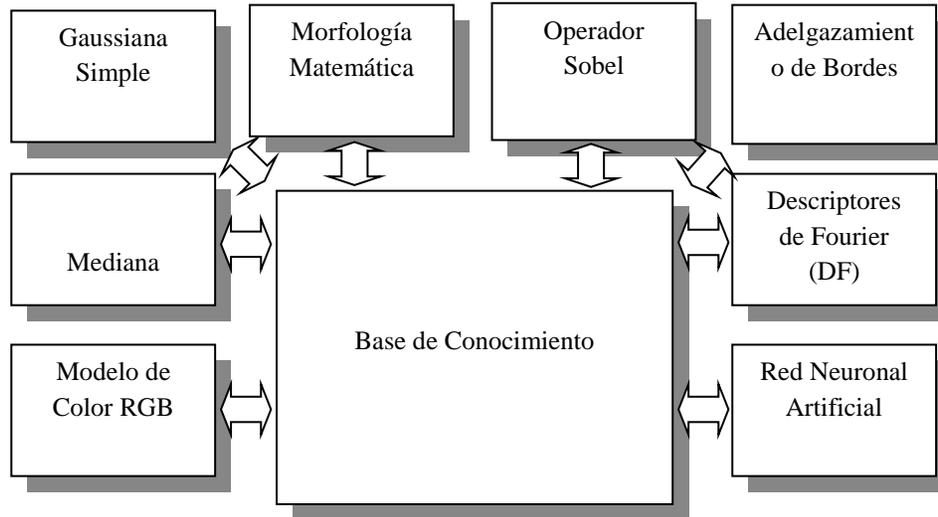


Figura 4.5.1 - Proceso de visión artificial final Implementado.

Etapas de Segmentación.

Finalmente, las mejoras realizadas sobre la etapa de segmentación consisten en aplicar al resultado entregado por el método de la Gaussiana Simple (sobre modelo de color RGB) la siguiente secuencia de pasos:

1. Someter la imagen segmentada a un proceso de Erosión para eliminar el ruido.
2. Someter la imagen resultado del paso anterior a un proceso de Dilatación para rellenar y conectar regiones pertenecientes a un mismo objeto.
3. Someter la imagen resultado del paso anterior a un nuevo proceso de Erosión para adelgazar la región del objeto de primer plano detectado.

Esta secuencia realiza los dos primeros pasos definidos en el análisis de algoritmos alternativos (modelo de color HSV) agregando un nuevo proceso de erosión que busca disminuir el efecto de ensanchamiento producido por la dilatación sobre la imagen segmentada, así se logra una representación con un mayor parecido a la figura real del objeto de primer plano (aunque se aumenta el tiempo de procesamiento necesario para ejecutar la segmentación de una imagen digital, este tema será discutido en las conclusiones finales del próximo capítulo). Algunos de los resultados obtenidos en esta etapa se pueden ver en la siguiente figura:

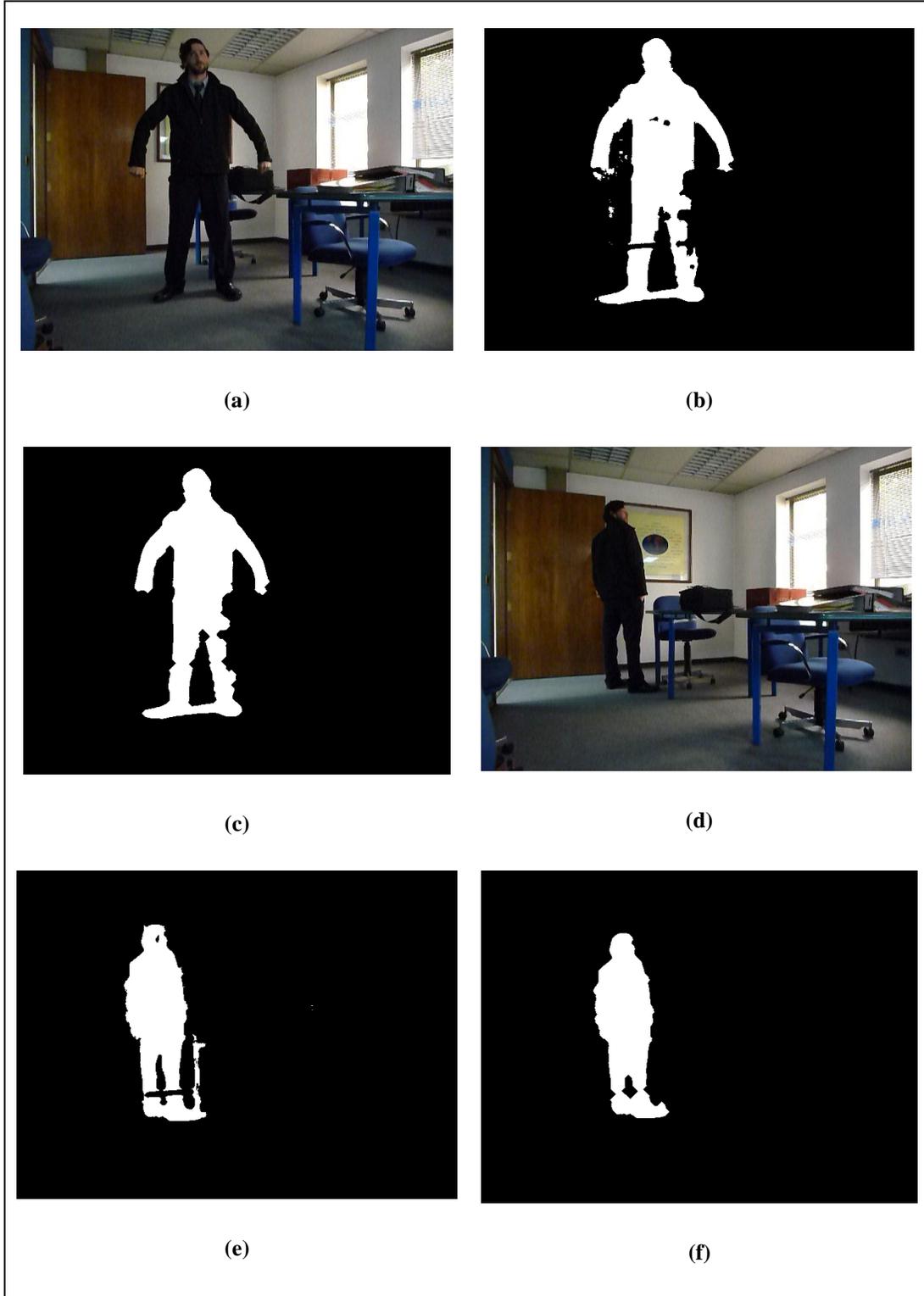


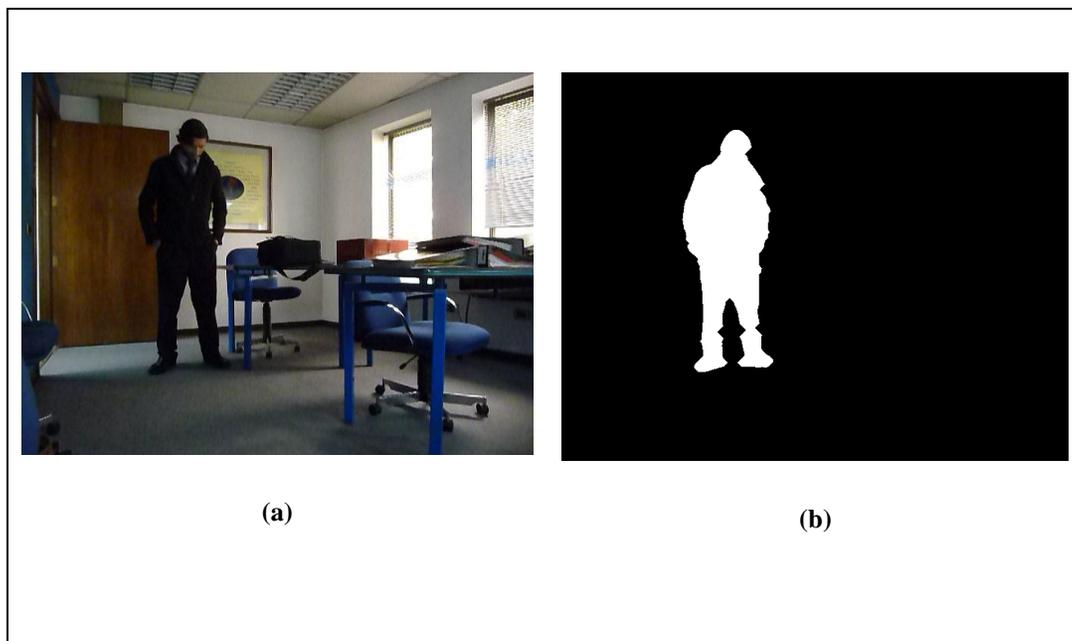
Figura 4.5.2 - Resultados morfología matemática (MM) en proceso final (a) Imagen 1 (b) Resultado segmentación imagen 1 en RGB (c) Resultado proceso de MM en imagen 1 (d) Imagen 2 (e) Resultado segmentación imagen 2 en RGB (f) Resultado proceso de MM en imagen 2.

Los anteriores son ejemplos de los resultados que entrega la mejora del proceso a través de la incorporación de morfología matemática en la etapa de segmentación, como se puede ver, a través de este pos-procesado se ha logrado eliminar el ruido que permanece en las imágenes luego de que estas son segmentadas. A diferencia de lo que ocurría con HSV, la ejecución del proceso utilizando como base el modelo de color RGB no provoca un ensanchamiento excesivo de la forma original del objeto de primer plano y mantiene una buena conectividad entre los píxeles de un mismo objeto, aunque el efecto de la sombra proyectada en la escena sigue afectando la forma original del objeto (observable principalmente en los pies de la persona de las imágenes anteriores).

Como se observa en las imágenes anteriores, la etapa de segmentación entrega como resultado objetos de primer plano con un muy buen nivel de conectividad, sin efecto de ensanchamiento e incluso con supresión de sombras. Con la obtención de estos resultados ya se cuenta con una fuente de información en buen estado para la siguiente etapa del proceso de visión artificial: “Representación y Descripción”.

Etapa de Representación y Descripción.

Como se ha descrito en secciones anteriores, la etapa de “Representación y Descripción” considera la ejecución de los algoritmos de Sobel, adelgazamiento de bordes y descriptores de Fourier (DF) en la detección de bordes y su posterior descripción para ser utilizados como entradas de la RNA implementada. A continuación se muestra un ejemplo del resultado obtenido por los algoritmos de esta etapa:



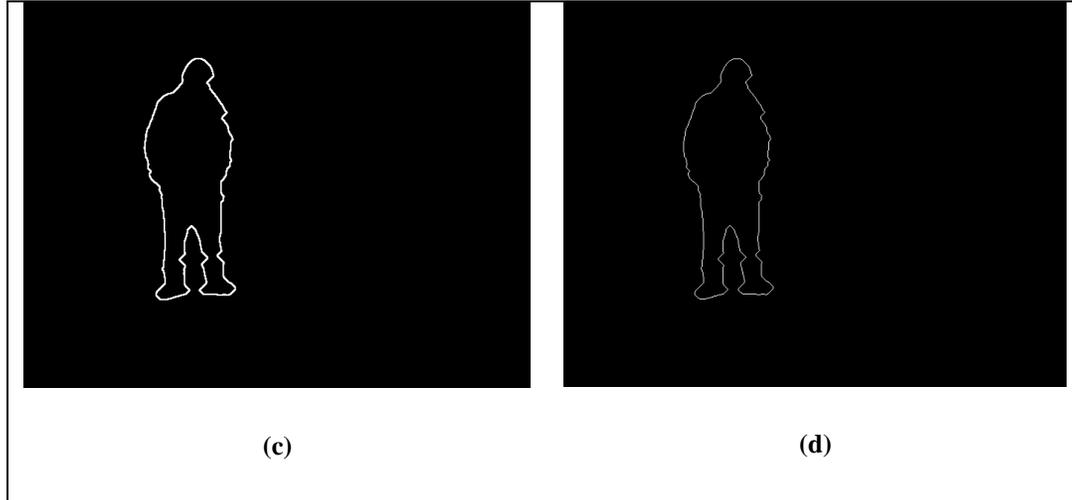


Figura 4.5.3 - Resultados proceso final, etapa Representación y Descripción. (a) Imagen Original (b) Resultado morfología matemática (c) Resultado operador de Sobel (d) Resultado Adelgazamiento de Bordes,

El operador de Sobel permite detectar bordes y a través del algoritmo de adelgazamiento se logra acotar la representación del objeto a un conjunto de puntos que representen el contorno cerrado de una persona restringiendo la cantidad de vecinos que cada pixel de borde puede tener (sólo 2 vecinos), esto permite tener una fuente de información optima para calcular los Descriptores de Fourier que representarán al objeto.

Respecto del cálculo de los DF, estos son guardados en un archivo donde por cada objeto detectado se mantienen sólo 30 descriptores normalizados, esto ya que como se explicó en el estado del arte, un reducido número de descriptores es suficiente para lograr una buena descripción del objeto. En este archivo también se registra el tipo de objeto que representan los descriptores (patrón). A continuación se muestra un ejemplo del archivo utilizado:

```
patrones1304457025357.dat
1 38
2 0.1334761411245819 0.04118145443336609 0.027589403755830896 0.030167430073101537
3 0 0 0 1
4 0.12415529732007712 0.047881028821809696 0.0317909430259514 0.03292907148620194
5 0 0 0 1
6 0.13941014617670383 0.04599353480208361 0.04930518637343898 0.048112490080332596
7 0 0 0 1
8 0.14984788583131625 0.05108630232082607 0.03205395713072711 0.03081957065601701
9 0 0 0 1
10 0.11104629094432769 0.028881355031890175 0.024624982553538174 0.0195076049381296
11 0 0 0 1
12 0.11169835791794504 0.028187886458835257 0.019877309700667502 0.0236196058434245
13 0 0 0 1
14 0.1168614722655041 0.04094247413002507 0.023152675219255483 0.016476309365249225
15 0 0 0 1
16 0.10624906021350623 0.050269684984579295 0.026245650261903786 0.0172992755632077
17 0 0 0 1
18 0.1506794574243368 0.016168346099624045 0.009519491526322487 0.01201563201968746
19 0 0 0 1
20 0.12941165194742318 0.03647654305121379 0.034208660028621024 0.02788281836289836
21 0 0 0 1
22 0.11319987153070377 0.02084374369345587 0.018758133929106836 0.02509970214159192
23 0 0 0 1
24 0.10547811110533264 0.015683878291161694 0.011932214395020877 0.0260023333372936
25 0 0 0 1
26 0.1051283217020665 0.012816417380579665 0.009594699258076213 0.02549647854881835
27 0 0 0 1
28 0.1176522772218152 0.01282158176748868 0.012865332451897101 0.022730069045914415
```

Figura 4.5.4 – Archivo con Descriptores de Fourier de los objetos detectados.

En este archivo con los DF que representan a un grupo de nuevos elementos detectados en una escena (figura anterior), el número de la primera línea establece el total de objetos que se encuentran representados en la colección de descriptores de Fourier existentes en el archivo, cada línea de texto tiene un conjunto de 30 descriptores que representan los objetos detectados, la clase correspondiente al objeto de cada conjunto de DF es representada en la línea siguiente (El tipo “0 1” será utilizado para representar a una persona, tal como se explicará en la siguiente sección).

Etapas de Clasificación.

Como se mostró anteriormente, en esta etapa ya se tiene un proceso de visión artificial capaz de entregar la descripción de cada uno de los objetos de primer plano detectados en la escena, con esta información se genera un archivo con los DF correspondientes del conjunto de nuevos objetos detectados para ser utilizado como entrada de la clasificación.

El objetivo de esta etapa es clasificar los objetos de primer plano según las características que posea su contorno, para esta tarea se ha construido una RNA del tipo Perceptron multicapa (MLP) con algoritmo de aprendizaje Back-Propagation (BP). Una mejora que se ha considerado en la construcción de la red consiste en la inclusión del factor “Momento” que permite mantener una pequeña memoria del resultado obtenido en una iteración previa del entrenamiento, de esta forma se podrán actualizar de mejor forma los pesos de las neuronas y conexiones sinápticas de la red.

La estructura inicial de la red a ser entrenada y evaluada está compuesta de 3 capas, con 30 neuronas en la capa de entrada, 60 neuronas en la capa oculta y 2 neuronas para la capa de salida. El número de neuronas consideradas para la capa de entrada se corresponde con el número de descriptores de Fourier calculados para cada objeto y el número de neuronas de la capa de salida se corresponde con el número de patrones que se quiere poder clasificar. Esta estructura inicial junto a otros parámetros de la red (razón de aprendizaje, parámetro del “momento”, entre otros) serán estudiados y modificados según los resultados que entreguen, hasta conseguir una configuración que permita a la RNA entregar los mejores resultados de acuerdo al caso de estudio seleccionado.

Las muestras obtenidas desde el entorno seleccionado son:

- 104 imágenes de personas.
- 100 imágenes de sillas tipo oficinas.

Donde para realizar el entrenamiento de la red se utilizará el 70 % del conjunto total de imágenes disponibles (73 y 70 respectivamente) y en la fase de comprobación para testear el resultado será utilizado el restante 30 % de las muestras (31 y 30 respectivamente). Esto de acuerdo al método de correlación cruzada explicado en estado del arte.

A continuación se comenta el análisis realizado sobre parámetros y otras consideraciones de la RNA como el patrón de salida, orden del archivo de patrones, estructura de la red, tasa de aprendizaje y factor de momento a utilizar. Al final de la sección se expone la configuración final definida para la RNA.

Patrón de Salida de la Red

Un punto importante a considerar en el entrenamiento de la red está relacionado con el formato de los patrones de salida que se utilizarán para los tipos de objetos que se quieren clasificar. Como parte del algoritmo de la RNA se utiliza una función de activación del tipo sigmoide, por lo tanto los valores posibles para cada una de las neuronas de la capa de salida son “0” o “1”, así se evalúa el resultado de cada grupo de descriptores de Fourier procesados por la red dependiendo de si sus respuestas son valores cercanos o no a estos números.

En las pruebas realizadas se ha observado que una buena práctica es utilizar cada una de las neuronas de la capa de salida para representar cada clase de objeto que se quiere clasificar, esto permite que ante un nuevo objeto de

primer plano sólo la neurona que representa la clase correspondiente tenga un valor próximo a “1”, el resto deberán estar en “0” o muy cercano a este valor. El criterio anterior ayuda a disminuir el tiempo de entrenamiento de la red, ya que se ha podido verificar que con configuraciones diferentes (p.e. valor “1” en más de una neurona de la capa de salida, para un solo objeto) se tiende a caer en mínimos locales y no se alcanza a minimizar lo suficiente el error de clasificación cometido por la red.

Orden del Archivo con los DF

Respecto del archivo creado con los descriptores de Fourier de todos los objetos de primer plano detectados, se ha comprobado que se puede optimizar el resultado del aprendizaje de la red si los datos de entrenamiento para el conjunto de patrones son presentados de manera aleatoria, es decir, evitando hacer un procesamiento secuencial de todas las imágenes de una clase de patrón para luego continuar con el siguiente tipo, ya que cuando la red es enfrentada a un patrón determinado sus pesos son optimizados para ese patrón en particular y esta configuración es reforzada al seguir con el procesamiento de una secuencia de imágenes del mismo tipo, luego al comenzar con el entrenamiento de un nuevo patrón se pierde la optimización alcanzada para conseguir adaptar la estructura de la red al reconocimiento del nuevo patrón. El orden secuencial del archivo de DF produce así una configuración menos óptima que si se tienen en cuenta la presentación aleatoria de estos descriptores en la etapa de entrenamiento de la red.

A continuación se muestran los resultados obtenidos según el orden de presentación de los DF del archivo utilizado para entrenar la RNA.

Orden Secuencial del Archivo de DF		Orden Aleatorio del Archivo de DF	
Mínimo Error Obtenido	Nro de Iteraciones Realizadas	Mínimo Error Obtenido	Nro de Iteraciones Realizadas
0.009999152	4368	0,009996589	3.377

Tabla 4.5.1 Resultados del entrenamiento de la RNA según orden del archivo con los DF.

En la información presentada se puede ver que al organizar los patrones de entrenamiento en orden aleatorio se obtienen mejoras respecto del número de iteraciones que son necesarias para alcanzar el mínimo error esperado de la red (con una diferencia de 991 iteraciones) y además se obtiene una disminución del error alcanzado. Entonces para el presente proyecto se consideran DF utilizados de manera aleatoria en el entrenamiento de la RNA.

Estructura de la Red

Una vez definido el formato del archivo de descriptores de Fourier, lo siguiente que corresponde realizar es definir la cantidad de neuronas a utilizar en la capa oculta de la RNA, para este fin a continuación se presentan los resultados obtenidos con diversas configuraciones utilizadas:

Tamaño Capa Oculta	Mínimo Error Obtenido	Nro de Iteraciones	Tiempo de Procesado (mlsg)
10	0,009999806	3.311	9.237
20	0,009997900	3.329	23.073
30	0,009998472	3.393	28.571
40	0,009997309	3.564	33.623
50	0,009996389	3.407	51.411
60	0,009996589	3.377	58.335
70	0,009996411	3.304	67.912
80	0,009999057	3.296	79.824
90	0,009999394	3.505	92.031
100	0,009996955	3.308	97.529

Tabla 4.5.2 - Variación del número de neuronas utilizadas en capa oculta.

La información de la tabla anterior para una mejor interpretación es representada en los siguientes gráficos:

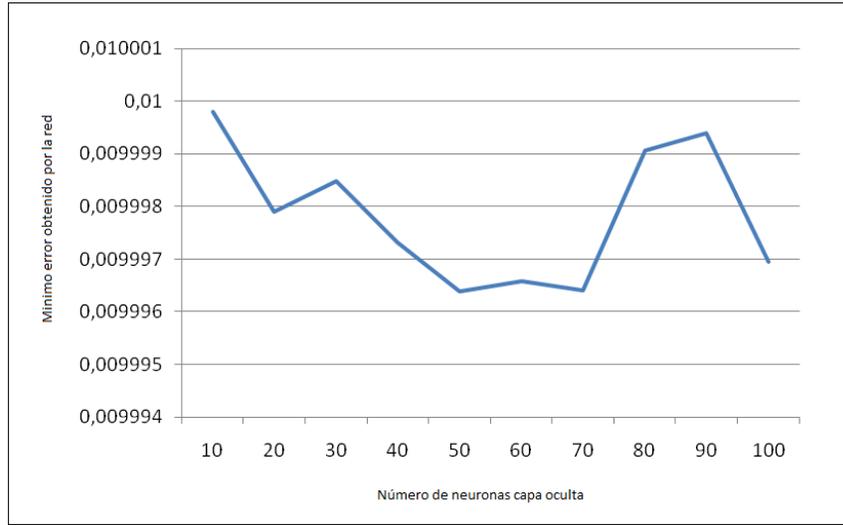


Figura 4.5.5 - Estructura de la red. Mínimo error obtenido.

En el grafico anterior se observa que la curva del error presenta un comportamiento variable en el tiempo y que no sigue una tendencia hacia el descenso o hacia el descenso a medida que cambia el número de neuronas de la capa oculta, sino que más bien presenta oscilaciones entre ambas tendencias. Entonces, si bien se han alcanzado valores mínimos del error de entrenamiento utilizando entre 50 y 70 neuronas, estas diferencias no son de gran valor (están en promedio cerca del sexto decimal), entonces habrá que considerar otros factores para complementar el análisis.

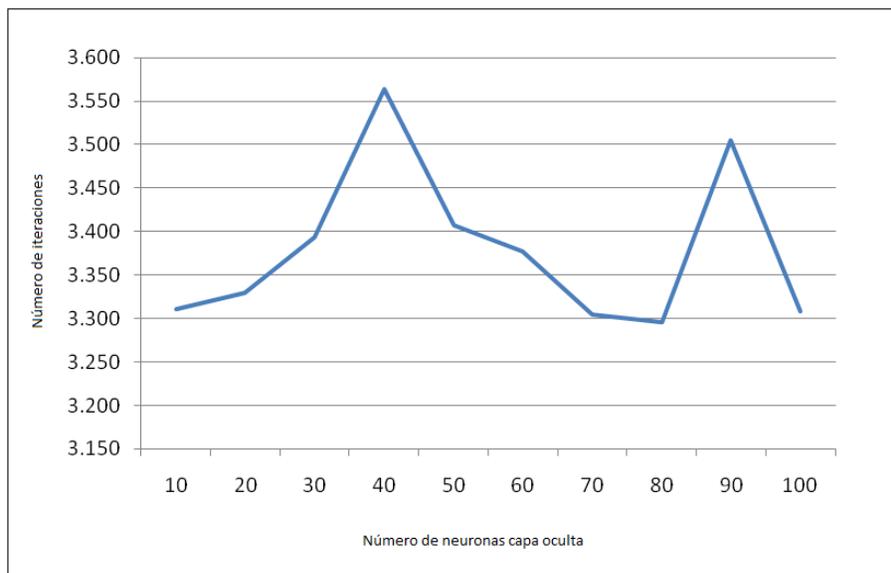


Figura 4.5.6 - Estructura de la red. Numero de Iteraciones.

En el gráfico de la figura anterior se puede ver que se mantiene la característica de una curva que oscila entre ascensos y descensos respecto del resultado óptimo buscado (en este caso, el menor número de iteraciones). Se puede ver que con 70, 80 y 100 neuronas para la capa oculta la red logra finalizar la etapa de entrenamiento en un menor número de iteraciones.

De acuerdo con los resultados observados sobre las variables de mínimo error, número de iteraciones y tiempo de procesado, se ha optado por utilizar una red con 50 neuronas en la capa oculta por ser donde se presenta un bajo número de iteraciones (3.407) y bajo error de clasificación (0,009996389), además el tiempo de procesado que es necesario para entrenar la red (51 segundos aprox) representa un buena opción respecto de lo observado para otras configuraciones, ya que con un número menor de neuronas disminuye la cantidad de variables internas que la red utiliza posteriormente para clasificar objetos (número de neuronas y de conexiones sinápticas) y con un número mayor aumenta el tiempo de procesamiento total requerido (se considerará también que este número de neuronas influirá en la cantidad de operaciones que será necesario realizar cuando la red funcione en tiempo real).

Tasa de aprendizaje

La velocidad de convergencia que logre la red estará dada por la tasa de aprendizaje que sea utilizada, normalmente este término es un valor positivo entre 0 y 1. Si la tasa de aprendizaje es mayor que 1, es fácil para el algoritmo de aprendizaje provocar que se creen pesos incorrectos y la red oscilará. Por otro lado, pequeños valores en la tasa de aprendizaje no corregirán el error actual de manera rápida. En la siguiente tabla se presentan los resultados obtenidos por la red con diferentes tasas de aprendizaje:

Momento	Tasa de Aprendizaje	Mínimo Error	Iteraciones
0,3	0,1	0.009999561	16724
0,3	0,2	0.009998816	8392
0,3	0,3	0.009999913	5596
0,3	0,4	0.009997583	4220
0,3	0,5	0,009999369	3398
0,3	0,6	0.009995827	2783
0,3	0,7	0.009998709	2440

Tabla 4.5.3 Comportamiento de la RNA según tasa de aprendizaje

Momento	Tasa de Aprendizaje	Mínimo Error	Iteraciones
0,3	0,8	0.009998812	2120
0,3	0,9	0.009994658	1931

Tabla 4.5.3 Comportamiento de la RNA según tasa de aprendizaje (cont.)

A continuación las graficas de algunos de los resultados contenidos en la tabla anterior.

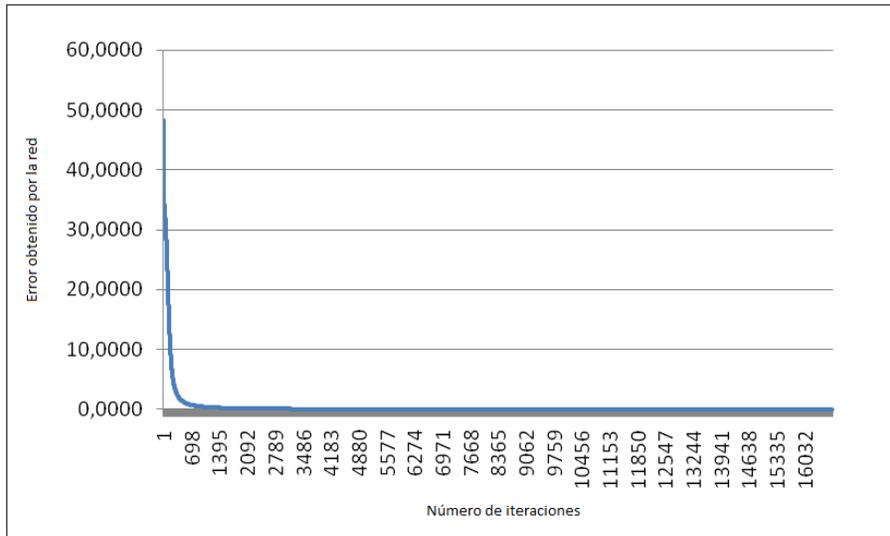


Figura 4.5.7 - Estructura de la red. Tasa de entrenamiento = 0,1

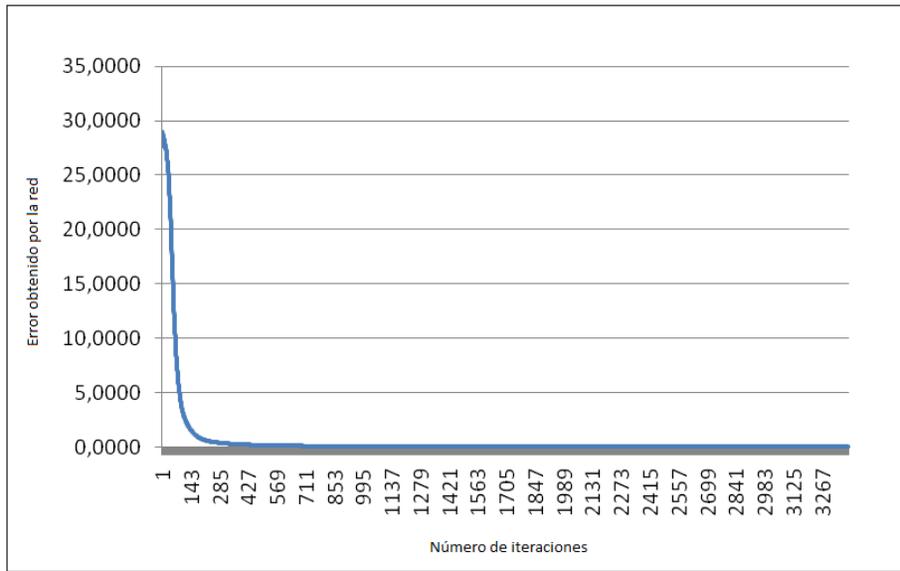


Figura 4.5.8 - Estructura de la red. Tasa de entrenamiento = 0,5

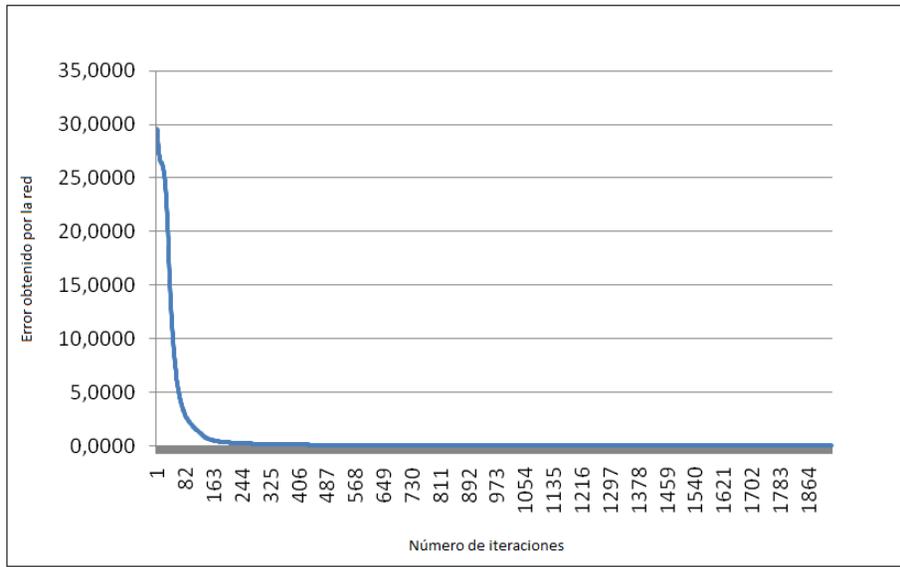


Figura 4.5.9 - Estructura de la red. Tasa de Entrenamiento: 0,9

Para diferentes valores del factor de aprendizaje el error converge siempre hacia el mínimo esperado (0,01), la diferencia está en el número de iteraciones que se requieren en cada caso para terminar el entrenamiento, a mayor tasa de aprendizaje mayor es la modificación de los pesos en cada iteración, por lo que el aprendizaje es más rápido, y un valor bajo de esta tasa equivale a que necesitamos gastar mucho tiempo para llevar a cabo el entrenamiento. Sin

embargo, con un valor para la tasa de aprendizaje mayor, la red puede no ser capaz de discriminar de manera tan adecuada como un sistema que aprende más despacio.

La tasa de aprendizaje a utilizar la RNA implementada será igual a “0,5”, por representar un valor medio considerando los comportamientos de la red mencionados en el párrafo anterior al aumentar o disminuir este valor.

Factor Momento.

Un alto aumento del valor de la tasa de aprendizaje puede ocasionar oscilaciones en la curva del error, para controlar estas posibles oscilaciones se utiliza una constante llamada “momento”, la cual determina el efecto que tendrán en el instante “ $t+1$ ” los cambios de los pesos en el instante “ t ”. Con la utilización de este factor “momento” se logra la convergencia de la red en un menor número de iteraciones y la curva del error presenta un comportamiento más estable. En los resultados mostrados en la tabla 4.5.3 se utilizó un factor de momento igual a “0.3”. El valor del factor momento puede variar entre “0,0” y “1,0”, en la siguiente tabla se muestran los resultados obtenidos para distintos valores de este factor:

Momento	Tasa de Aprendizaje	Mínimo Error	Iteraciones
0,1	0,5	0.009996878	3493
0,2	0,5	0.009999498	3516
0,3	0,5	0.009999369	3398
0,4	0,5	0.009997311	3285
0,5	0,5	0.009997468	3143
0,6	0,5	0.009999061	3182
0,7	0,5	0.009997350	3225
0,8	0,5	0.009996896	3109
0,9	0,5	0.009997221	3079

Tabla 4.5.4 Estructura de la red. Factor momento.

En los siguientes gráficos se muestran estos resultados:

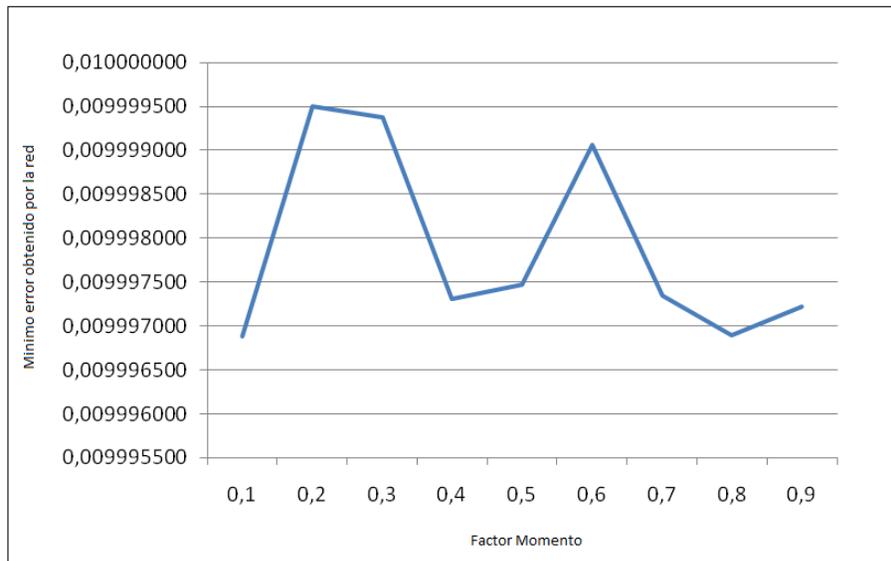


Figura 4.5.10 - Estructura de la red. Factor momento – Mínimo error.

En la grafica anterior se puede ver que una variacion en el factor momento no tiene mayor efecto sobre el error minimo cometido por la red, ya que las diferencias son menores (en el sexto decimal).

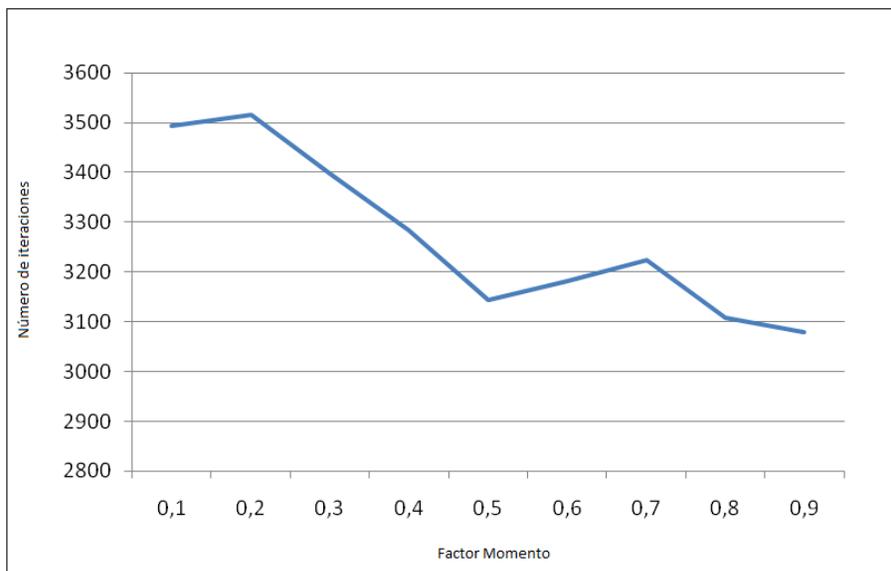


Figura 4.5.11 - Estructura de la red. Factor momento - Número de iteraciones.

En la grafica se puede ver que existe una clara tendencia al descenso en la curva del número de iteraciones que necesita la red para alcanzar el mínimo error esperado, de acuerdo al factor momento utilizado. Por lo tanto se puede concluir que existe una relación directa entre el valor del factor momento y la disminución del número de iteraciones que necesita la red para alcanzar el mínimo error.

De acuerdo con estos resultados, en el presente proyecto se utilizará un factor momento de valor “0,9”.

Configuración Definitiva de la RNA y Entrenamiento

En esta fase ya se ha detectado aquellas configuraciones de la RNA que entregan los mejores resultados de acuerdo al caso de estudio seleccionado. Los valores a considerar en cada una de las variables internas de la RNA son los siguientes:

Tamaño Capa Entrada	Tamaño Capa Oculta	Tamaño Capa Salida	Tasa de Aprendizaje	Momento	Mínimo Error Esperado	Mínimo Error Obtenido	Nro de Iteraciones Realizadas	Tiempo de Procesado (mlsg)
30	50	2	0,5	0,9	0,1	0.009996913	3219	48219

Tabla 4.5.5 Configuración definitiva de la RNA.

Donde los tamaños de la capa de entrada y salida se corresponden con el número de DF utilizados y con el número de objetos que se quieren poder clasificar, respectivamente. Los valores para el resto de las variables fueron obtenidos en el proceso de pruebas y análisis descrito anteriormente.

Los resultados obtenidos en la etapa de entrenamiento de la RNA son expuestos en la siguiente tabla:

Mínimo Error Obtenido	N° de Iteraciones Realizadas	Tiempo de Procesado (mls)
0.009996913	3219	48219

Tabla 4.5.6 Resultado entrenamiento de la RNA

El grafico de la curva de error obtenido por la RNA en la etapa de entrenamiento es el siguiente:

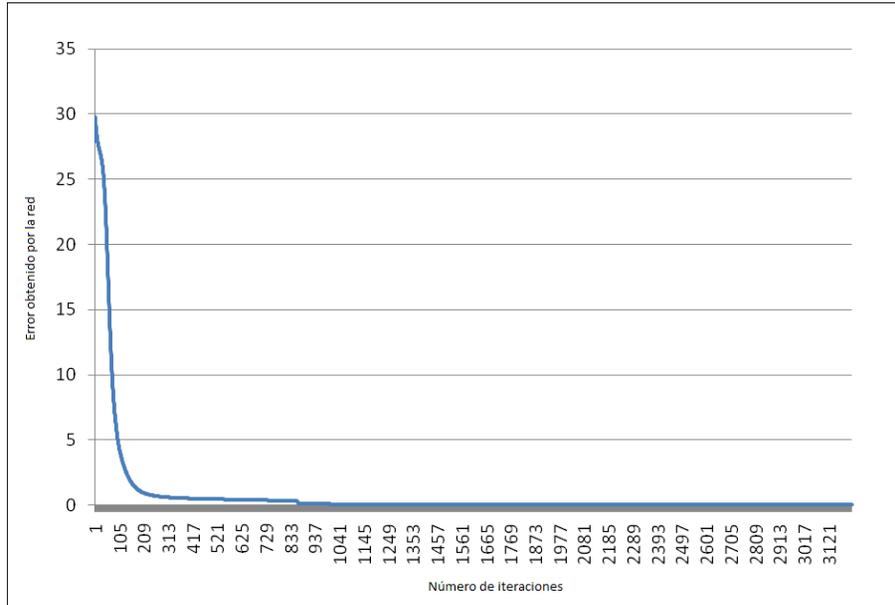


Figura 4.5.12- Curva del error obtenido con la configuración definitiva de la RNA.

Se puede ver en la tabla 4.5.6 que el error de entrenamiento de la RNA alcanza el valor mínimo esperado en la iteración 3.219. En esta iteración el error cometido por la red es de 0,01 (99% de asertividad para el conjunto de muestras de entrenamiento). Además, como se puede ver en la figura 12 (que corresponde a un aumento de nitidez sobre una sección de la figura 11) en la iteración 1.390 aproximadamente existe un error cercano a 0, específicamente el error en esta iteración es de 0,03 (97% de asertividad).

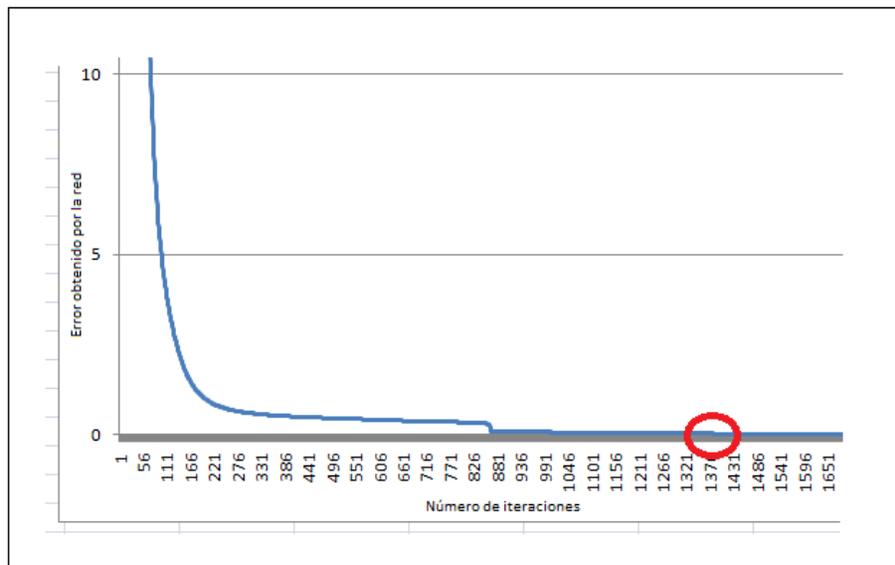


Figura 4.5.13 – Aumento nitidez curva del error obtenido con la configuración definitiva de la RNA.

Se considera este par de cantidades de iteraciones para generar la estructura final de la RNA a utilizar durante el análisis del desempeño de la RNA (utilizando muestras que no formaron parte de la etapa de entrenamiento). Así se podrán visualizar diferentes errores de generalización y luego seleccionar el que entregue mejores resultados.

4.6 Análisis del Desempeño Final

Los resultados obtenidos por la RNA sobre muestras del conjunto de comprobación, que corresponden a imágenes que no fueron utilizadas durante el entrenamiento de la red, se muestran en la tabla 4.5.7. De acuerdo con estos resultados, la RNA logra un 83 % de asertividad en la clasificación de personas, lo que representa la correcta detección de 26 personas de un total de 31, lo que considerando que el entrenamiento fue realizado con sólo 73 imágenes de entrenamiento demuestra la capacidad del sistema de aprender rápidamente patrones de detección.

Estos resultados confirman el correcto desempeño de la representación de objetos mediante la extracción de sus características representativas (contorno) desde el dominio de las frecuencias (DF).

	Persona	Silla	Sin Clasificar	Total	% de Generalización
DF Persona	26	4	1	31	83%
DF Silla	12	18	0	30	60%
Tiempo de ejecución clasificación				416 milisegundos	

Tabla 4.5.7 Generalización de la RNA.

4.7 Conclusión

En este capítulo se han presentado las características del entorno desde donde se han obtenido las muestra empleadas durante las pruebas de los algoritmos. Se ha descrito también el resultado de la implementación de un grupo inicial de métodos donde ha sido necesario investigar sobre algoritmos alternativos que mejoraran los resultados obtenidos. En la última parte del capítulo se han comentado los resultados finales del conjunto definitivo de métodos de procesamiento de imágenes digitales y de visión artificial que permiten abordar la problemática del caso de estudio propuesto en el capítulo anterior.

Con las pruebas anteriores se han podido visualizar diversos factores que afectan al desempeño de un sistema de clasificación de objetos en base a patrones en imágenes digitales. También se han analizado la implicancia de los parámetros de configuración en los resultados obtenidos por los diversos algoritmos implementados.

Conclusiones e Implicaciones

En esta sección se presentan las conclusiones finales del proyecto, se analizan los resultados obtenidos de acuerdo con los objetivos planteados inicialmente y el desarrollo de los capítulos anteriores del documento.

5.1 Introducción

Los contenidos de este documento abarcan todo el trabajo realizado en la investigación, logrando cumplir todos los objetivos planteados inicialmente. Mediante el análisis de la literatura realizado en el capítulo 2 se pudo conocer el área abordada estudiando los algoritmos tradicionales del procesamiento de imágenes digitales, el método de los descriptores de Fourier que permiten llevar un problema basado en el espacio y las tonalidades de los píxeles de una imagen digital a un espacio de frecuencias (eliminando las limitaciones de las técnicas tradicionales del dominio del espacio), se estudió también la estructura y capacidades de una red neuronal artificial que permite dotar a un sistema computacional de la capacidad de interpretar el contenido de una imagen digital clasificando los nuevos objetos que aparezcan en ella. En el capítulo 3 en tanto, se ha planteado la metodología seguida en la implementación y análisis de resultados, se ha sugerido un caso de estudio específico que permite visualizar el desempeño de los algoritmos seleccionados, al final del capítulo se describieron los procedimientos de investigación utilizados para realizar el análisis de resultados. En capítulo 4 se ha comentado la elección del entorno de monitoreo y el tipo de imágenes utilizadas en las pruebas, luego se han expuesto las pruebas iniciales realizadas sobre un conjunto de algoritmos y de acuerdo con los resultados observados fue investigado un conjunto de algoritmos alternativos que mejoraran las debilidades observadas en el conjunto inicial de métodos implementado, al final del capítulo 4 se realizaron las pruebas definitivas del proceso de visión artificial implementado y se expusieron los resultados obtenidos en cada uno de los algoritmos.

En este capítulo se exponen las conclusiones finales del desarrollo del proyecto, analizando los resultados observados en el capítulo anterior, las limitaciones dentro de las cuales se enmarca la presente investigación y las correspondientes líneas de investigación que se consideran entregarían un mayor grado de confiabilidad a un sistema de visión artificial de las características del diseñado en el presente proyecto.

5.2 Conclusiones acerca de los algoritmos implementados

En esta sección se analizan los resultados y el comportamiento de cada uno de los elementos del proceso de visión artificial construido, de acuerdo con los resultados observados en el capítulo anterior.

Modelo de color RGB: Luego de observar el comportamiento del conjunto de algoritmos con otros modelos de color como HSI que por sus similitudes con el sistema visual humano es uno de los más recomendados, se pudo ver finalmente que RGB entregaba mejores prestaciones según el entorno y las condiciones del caso de estudio seleccionado.

Mediana: Se ha comprobado que someter las imágenes a un proceso de suavizado mejora las características de los objetos presentes en ella otorgando mayor homogeneidad de las tonalidades de un mismo elemento.

Gaussiana simple: Este método es una buena herramienta para identificar los objetos de fondo y de primer plano presentes en una escena, ya que permite crear un buen modelo de representación de las tonalidades del fondo y permitir posteriormente, según la presencia de variaciones significativas en estas tonalidades, categorizar a nuevos objetos como elementos de primer plano.

Morfología matemática: Principalmente se utilizó este método en la reconstrucción de las formas reales de los objetos detectados en una imagen digital, a través de la combinación de un proceso de erosión y dilatación se logró disminuir el efecto de deformación provocado por ejemplo con nuevos elementos de tonalidades similares a las observadas en el fondo de la escena.

Operador de Sobel: Luego de tener los objetos de primer plano detectados y a través de métodos como la erosión y dilatación mejoradas sus formas, el operador de Sobel permite detectar los bordes de estos y eliminar todos los píxeles que no sean parte de la frontera del objeto.

Adelgazamiento de bordes: Una de las características de los bordes detectados con el operador de Sobel es que estos poseen segmentos con más de un píxel de grosor, lo que no permite representar la frontera correctamente para su posterior procesamiento con DF, esto es solucionado con el adelgazamiento que se realiza sobre todos los píxeles de los nuevos objetos que son detectados en la imagen digital.

Descriptores de Fourier: Luego de adelgazar los bordes de los nuevos objetos detectados estos pueden ser representados a través de sus descriptores de Fourier. La principal ventaja alcanzada con representar un objeto en el dominio de las frecuencias es que se logra invariancia a Traslaciones, Rotación y Escalado.

Red neuronal artificial: Los algoritmos descritos anteriormente permiten identificar los nuevos objetos que aparecen en las imágenes digitales, luego entregan una descripción numérica de estos que puede ser utilizada como entrada en la RNA que los clasificará en base a patrones de entrenamiento utilizados previamente. Una RNA permite

dotar a un sistema de la capacidad de interpretación de una escena, de manera similar a como es realizado por una persona, objetivo principal del presente proyecto.

Base de conocimiento: Los datos que surgen durante la implementación y posterior análisis de resultado sirven como base de conocimiento durante la ejecución de las pruebas y análisis de resultados, permiten visualizar los resultados obtenidos y evaluar eventuales mejoras de los métodos utilizados.

En las descripciones entregadas anteriormente, así como en los capítulos dos y cuatro principalmente, se puede visualizar como cada una de las etapas del proceso de visión artificial tiene un objetivo fundamental en el logro de la correcta interpretación de la escena. Si bien inicialmente se puede pensar que es tarea de la RNA realizar dicha interpretación luego de la información contenida en este trabajo queda reflejada la necesidad de mantener un funcionamiento cohesionado y robusto en cada una de las etapas involucradas. En este trabajo se ha podido comprobar por ejemplo, que la segmentación resulta muy importante para lograr una buena detección de bordes, pero además el modelo de color utilizado y realizar un buen suavizado de la imagen resultan también ser elementos fundamentales para asegurar un buen comportamiento de este algoritmo.

Con el desarrollo de este trabajo, también se ha comprobado que las características de cada método utilizado debe su correcto funcionamiento en parte importante a las características del entorno en donde es implementado, un mismo algoritmo ante un entorno determinado puede resultar particularmente apropiado, pero cambios de la escena o de las características que esta poseía durante su implementación pueden hacer que el algoritmo pierda su efectividad, siendo necesaria su modificación o la consideración de algún otro método que alcance de mejor forma a satisfacer las nuevas necesidades del entorno. Entonces, será fundamental la consideración del entorno involucrado en la elección del conjunto de métodos que compondrán un proceso de visión artificial.

5.3 Conclusiones sobre el problema de investigación

De acuerdo con los resultados obtenidos por la RNA y otros métodos expuestos en el capítulo 4, se ha podido desarrollar un proceso de visión artificial donde a partir del monitoreo de un entorno y el análisis de una secuencia de imágenes digitales es posible detectar aquellos objetos que no pertenecen al fondo de la escena y realizar por lo tanto su clasificación según la forma que tienen sus contornos.

Como se ha mencionado en la sección anterior, en esta investigación se ha podido constatar la importancia que posee cada uno de los algoritmos que son utilizados en un proceso de visión artificial, dado el nivel de acoplamiento que existe entre los métodos donde generalmente la salida que entrega uno es utilizada como entrada del siguiente algoritmo. Así se tiene finalmente una secuencia de implementaciones críticas, donde su complejidad radica en los múltiples factores que inciden en el resultado de cada una de las etapas del proceso de visión artificial, lo que requiere en varios casos la dedicación de investigaciones prácticamente exclusivas que afronten problemáticas

acotadas con el objetivo de mejorar todo el proceso en busca de generar capacidades de percepción e interpretación robustas tales como las que posee el sistema visual humano.

El problema de esta investigación ha sido resuelto a través de métodos que aunque son tradicionales en la literatura del área, su integración dentro de un proceso de visión artificial ha logrado sentar bases importantes que permiten evolucionar hacia investigaciones que alcancen un mayor porcentaje de aciertos y de efectividad, sobre todo considerando el aumento de prestaciones y la disminución de costos que hoy en día existe en los elementos que forman parte de la infraestructura tecnológica de un sistema de estas características (cámaras de vigilancia, procesadores, medios de transmisión de datos, entre otros).

Se entrega con este trabajo una base de información relacionada al área de procesamiento de imágenes digitales y visión artificial que aporta a la ingeniería una concentración y puesta en práctica de conocimientos, aborda el modelo clásico de procesado y los principales algoritmos que permiten cumplir con uno de los objetivos de la ingeniería informática: la incorporación de tecnología en apoyo de las labores desarrolladas por las personas (como las comentadas en los capítulos iniciales de este trabajo: manufactura, procesos de calidad, monitoreo inteligente, entre otros).

5.4 Limitaciones

Si bien hasta esta sección del capítulo se han mostrado los logros que ha tenido este proyecto, tal como se hizo en el capítulo inicial es importante destacar los elementos no abordados y que tienen incidencia sobre estos resultados, de esta forma es posible contextualizar de mejor manera el escenario sobre el cual se ha desarrollado la investigación.

Como primera limitación está la ya comentada centralización de los esfuerzos sobre la implementación y análisis de métodos específicos como lo son la transformada de Fourier y una RNA con algoritmo de aprendizaje BP, esto consideró durante la ejecución del proyecto enfocar el estudio sobre la lógica que consideran estos métodos y en sus características y ventajas que aportan al objetivo de dotar a un sistema de la capacidad de interpretación de una secuencia de imágenes. Este enfoque y aquellas variables que han sido excluidas de la investigación se deben considerar al plantear el desarrollo de nuevas investigaciones relacionadas al área.

Algunas de las limitaciones que han sido descubiertas durante el progreso de la investigación son:

Brillo: Se ha observado que un exceso de luz sobre los nuevos objetos que aparecen en una escena genera que un porcentaje de estos rayos de luz modifican las tonalidades originales que poseen estos nuevos elementos produciendo deformaciones.

Sombras: Del mismo modo a como sucede con el brillo generado sobre los nuevos objetos detectados en una escena, las sombras producidas interfieren en la detección de objetos prolongando generalmente las dimensiones de la forma real de estos nuevos elementos.

Características del escenario monitoreado: Un entorno muy dinámico en sus elementos no tiene la misma complejidad que uno entorno estático donde prácticamente no existan alteraciones, en el primero habrá que adaptar constantemente el modelo de fondo utilizado para detectar correctamente los nuevos elementos.

Estas limitaciones fueron observadas, registradas y acotadas en la selección del conjunto de imágenes de prueba utilizado, buscando así disminuir el impacto sobre el análisis de los algoritmos incluidos en la investigación.

En la siguiente sección de este capítulo se consideran aquellas investigaciones posteriores que aparecen como buenas alternativas para abarcar estas limitantes y otras líneas de investigación relacionadas al área.

5.5 Implicaciones para investigaciones posteriores

En esta sección se describen propuestas para mejorar y/o complementar los resultados obtenidos en el presente trabajo así como también para sugerir a otros investigadores conceptos a incluir en el diseño de nuevas investigaciones del área.

Como primera propuesta se considera la mejora de las etapas iniciales del procesado de imágenes digitales a través de la implementación de métodos de eliminación de brillo y sombras considerando la no pérdida de información de las características de los objetos. En este punto también deberían ser evaluadas las técnicas y medios utilizados en la captura de las imágenes, estudiando aquellos mecanismos y tipos de dispositivos que entreguen los mejores resultados de acuerdo a distintos tipos de entornos de monitoreo (estáticos y dinámicos), es deseable disminuir el ruido (interferencias en la señal, brillo, sombras) o la adición de estos elementos que puedan generar los medios de captura utilizados.

Respecto de las técnicas utilizadas, sería valioso considerar métodos alternativos o complementarios al conjunto final utilizado en este proyecto. Considerando que el enfoque principal utilizado ha sido la detección en base a las características del contorno de los objetos (representados a través de sus DF) una alternativa sería trabajar con un enfoque mixto que utilice de manera complementaria atributos como la “forma” y “textura” de los elementos detectados. La consideración en procesos de visión artificial de atributos como las texturas de los objetos lograría imitar este comportamiento de los seres humanos frecuentemente utilizado en el reconocimiento de objetos.

Líneas de investigación nuevas relacionadas con el área de la visión artificial pueden ser varias ya que se trata de un campo en pleno desarrollo. Una interesante e innovadora vía de investigación consiste en incorporar métodos del campo de la ingeniería biónica, la cual tiene como finalidad simular el comportamiento, forma o funcionamiento de los organismos vivos con el objetivo de crear sistemas que iguallen o mejoren las capacidades naturales de los seres vivos. Dado que se quieren implementar procesos inteligentes y autónomos, esta rama de la ingeniería representa una robusta alternativa para lograr este objetivo.

Sería importante estudiar además la incorporación de aquellas mejoras que puedan ser obtenidas desde los dispositivos de captura y medios de transmisión creados por la industria en el último tiempo, así como también asegurar un correcto almacenamiento de la información, resguardando los datos obtenidos desde diversos entornos de monitoreo (hogar, industria, etc.) utilizando mecanismos como la encriptación del medio de transmisión y/o de los datos almacenados.

Como se ha mencionado anteriormente en este documento, el apéndice I contiene una propuesta con el diseño de un framework de visión artificial que en futuros trabajos podría considerarse su desarrollo incorporándole todas aquellas funcionalidades necesarias para llegar a una herramienta que sirva de apoyo en investigaciones y desarrollos del área

Referencias

- [1] Rafael C. Gonzalez (University of Tennessee), Richard E. Woods (MedData Interactive), “*Digital Image Processing*”, Segunda edición, Capítulos: 1 – 2, Editorial Prentice Hall, 2002
- [2] R. Molina, “*Introducción al Procesamiento y Análisis de Imágenes Digitales*”, Departamento de Ciencias de la computación e I. A. Universidad de Granada, Granada.
- [3] D. Marr And E. Hildreth. “*Theory of edge detection. Processings of de Royal Society of London, Series*” B,207:187-216, 1980.
- [4] I.E. Abdou and W.K. Pratt. *Quantitative desing and evaluation of enhancement/ thresholding edge detector*. Procedings of de I.E.E.E., 67:753-763, 1979.
- [5] P.V.C. Hough. “Methods and jeans for recognizing complex Patterns”. U.S. Patent 3069654, 1962.
- [6] José Antonio Ramírez Rábago, “*Generación de fuentes virtuales de sonido en audifonos*”, Capitulo 4, Escuela de Ingeniería, Departamento de Ingeniería Electrónica, Universidad de las ameritas, Puebla, México
- [7] Stuart J. y Russell, Peter Norvig “*Inteligencia Artificial, un enfoque moderno*”, segunda edición, capítulos 20.5 – 24, Editorial Prentice hall, 2004.
- [8] José Ramón Mejía Vilet “Apuntes de Procesamiento Digital de Imágenes”, Área de Computación e Informática, Facultad de Ingeniería UASLP. Enero de 2005.
- [9] Samir Kouro R., Rodrigo Musalem M. “Tutorial Introductorio a la teoría de Wavelet”, Ingeniería civil electrónica UTFSM. Julio de 2002.
- [10] Bonifacio Martín del Brío, Alfredo Sanz Molina “*Redes Neuronales y Sistemas Difusos*”, “2ª edición, capítulos 1 y 2, Universidad de Zaragoza, Editorial Alfaomega, 2002.
- [11] Erich Gamma, Richard Helm, Ralph johnson, john vliissides “*Patrones de diseño, Elementos de software orientado a objetos reutilizable*”, Primera edición en español, Editorial Addison Wesley, 2003.

Framework de Visión Artificial Utilizando Patrones

El presente apéndice tiene como finalidad mostrar el diseño de una arquitectura (Framework) de visión artificial utilizando patrones de diseño, de modo de proponer una plataforma que sirva de apoyo en las pruebas de algoritmos de alto nivel en el campo de la visión artificial.

Se proponen patrones de diseño como herramienta a través de la cual se espera desarrollar una plataforma extensible, flexible y reutilizable, en consideración de los buenos resultados mostrados por proyectos que utilizan este enfoque de diseño. A continuación se dará una introducción al tema y posteriormente se incluirán en este apéndice los patrones de diseño específicos seleccionados para ser incluidos en esta propuesta.

Es importante resaltar el carácter del presente apéndice, en donde sólo se pretende mostrar una primera versión del diseño de un framework de visión artificial. Si bien esta primera versión ha sido utilizada en las pruebas del presente proyecto se reitera su carácter de propuesta, estando lejos de ser una versión definitiva de la solución diseñada.

Introducción a los Patrones de Diseño.

En 1979 el arquitecto Christopher Alexander escribió el libro "The Timeless Way of Building" en el cual proponía el aprendizaje y uso de una serie de patrones para la construcción de edificios de una mayor calidad. En palabras del autor, "Cada patrón describe un problema que ocurre infinidad de veces en nuestro entorno, así como la solución al mismo, de tal modo que podemos utilizar esta solución un millón de veces más adelante sin tener que volver a pensarla otra vez."

Los patrones propuestos por Christopher Alexander son un intento de formalizar y plasmar de una forma práctica generaciones de conocimiento arquitectónico. Un patrón define una posible solución correcta para un problema de diseño dentro de un contexto dado, describiendo las cualidades invariantes de todas las soluciones.

Más tarde, en 1987, Ward Cunningham y Kent Beck usaron varias ideas de Alexander para desarrollar cinco patrones de interacción hombre-ordenador (HCI) y publicaron un artículo titulado "Using Pattern Languages for OO Programs".

A principios de los 90's los patrones de diseño fueron considerados a partir de la publicación del libro “Design Patterns” escrito por el grupo Gang of Four (GoF) compuesto por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides, en el que se recogían 23 patrones de diseño comunes.

Los patrones en Informática se clasifican según el tipo de problemas que intentan resolver, entonces se tienen:

Patrones de arquitectura: Los cuales proponen esquemas organizativos estructurales fundamentales para sistemas software.

Patrones de Diseño: Los cuales proponen esquemas que definen estructuras de diseño y sus relaciones con los cuales construir los sistemas software.

Patrones de Interacción: Proponen las mejores prácticas relacionadas a la construcción de interfaces Hombre-Máquina (HCI).

Además de estas tres categorías de patrones existen esfuerzos en abarcar otras áreas como la definición de patrones en la construcción de sistemas empresariales, integración de sistemas y también en la definición, construcción e interacción de sistemas abstractos de gestión de flujos de trabajo (Workflow). Lo anterior demuestra que esta técnica en la actualidad está fuertemente incluida dentro de los aspectos importantes a considerar para la elaboración de un proyecto software.

Los patrones de diseño como se ha mencionado anteriormente son los que consideraremos en este trabajo. Estos patrones hacen uso de una teoría ampliamente conocida en Informática: “No es necesario resolver cada problema partiendo desde cero”, por el contrario la práctica recomendada es comenzar desde soluciones anteriores que ya han sido útiles para resolver problemas similares en el pasado. Considerando un primer acercamiento al diseño de soluciones software es común que a una persona que nunca ha desarrollado le sea tremendamente difícil llegar a una solución robusta en poco tiempo, también es probable que su solución definitiva contenga muchas inconsistencias y redundancia de código, lo que claramente provocará un sistema no reutilizable. En cambio un diseñador experimentado puede conseguir un muy buen diseño, esto debido a que cuando encuentra una solución buena la utiliza una y otra vez lo que genera patrones recurrentes en el diseño de sus clases y en las relaciones y responsabilidades asignadas a cada una de ellas, esto es lo que lo convierte en experto.

Los patrones de diseño hacen que sea más fácil reutilizar buenos diseños y arquitecturas. Al expresar como patrones de diseño técnicas que ya han sido probadas, estas se hacen mas accesibles para los desarrolladores de nuevos sistemas. Estos patrones ayudan a elegir las alternativas de diseño que hacen que un sistema sea reutilizable, y a evitar aquellas que dificultan dicha reutilización. Pueden incluso mejorar la documentación y el mantenimiento de los sistemas existentes al proporcionar una especificación explícita de las interacciones entre las clases y objetos y de cuál es su intención. En definitiva, los patrones de diseño ayudan a un diseñador a lograr un buen diseño más rápidamente [11].

Existen muchos Patrones, en el libro “Design Patterns” se postulan 23 los cuales son los más aceptados por la comunidad, pero también han surgido otras propuestas que aumentan este número de las cuales se pueden encontrar homólogos de los postulados por la “Banda de los Cuatro” (GoF) y también experiencias nuevas. En este proyecto consideraremos inicialmente 4 de los patrones propuestos por GoF, aunque haremos referencia también a 3 patrones pensados para inclusiones en futuras versiones del sistema. Solo consideramos incluir 4 patrones de diseño debido a que estamos realizando un primer acercamiento a la utilización de este concepto en el desarrollo de software y porque también es sabido que incluir una gran cantidad de estos patrones puede en ocasiones no hacer más que aumentar la complejidad, disminuyendo lo entendible del diseño propuesto e impactando en la mantenibilidad de la plataforma construida.

Los patrones de diseño se clasifican según su propósito en creacionales, estructurales y de comportamiento. Los patrones de diseño de creación abstraen el proceso de creación de instancias, ayudan a hacer a un sistema independiente de cómo se crean, se componen y se representan sus objetos. Los patrones estructurales se ocupan de cómo se combinan las clases y los objetos para formar estructuras más grandes. Los patrones de comportamiento tienen que ver con los algoritmos y con la asignación de responsabilidades a objetos, estos patrones describen no solo patrones de clases y objetos, sino también patrones de comunicación. Los patrones de comportamiento describen el flujo de control complejo que es difícil de seguir en tiempo de ejecución, lo que nos permite olvidarnos del flujo de control para concentrarnos simplemente en el modo en que se interconectan los objetos.

Patrones de Diseño Utilizados.

Los patrones utilizados en el diseño del Framework a construir son:

- Fábrica Simple (Factory Simple).
- Único (Singleton).
- Estrategia (Strategy).
- Observador (Observer).

Donde los 2 primeros de la lista corresponden a patrones de diseño de creación y los siguientes son de la categoría de patrones de diseño de comportamiento.

Por otro lado, los patrones que se piensan incluir en futuras versiones del prototipo son:

- Adaptador (Adapter).
- Comando (Comand).

- Estado (State).

Donde el primero (Adapter) corresponde a un patrón estructural y los otros 2 a patrones de comportamiento.

En las secciones siguientes se explicaran de forma resumida las principales características de estos patrones para posteriormente analizar su implementación en el Framework [11].

Fábrica Simple

El patrón fabrica simple es el primero de 3 patrones de creación que están relacionados entre si ya que entre ellos representan una secuencia de evolución relativa a la creación de objetos. Para entender lo anterior es necesario definir las características de cada uno de estos patrones:

Fabrica Simple: Una clase es la encargada de crear objetos para otras clases. No delega en otras subclases y sus métodos pueden ser estáticos.

Método de Fabricación: Se utiliza una interfaz para definir los métodos de fabricación abstractos de forma que sean las subclases quienes implementen estos métodos para crear los objetos concretos. Permite que sean las subclases quienes decidan que clase instanciar.

Fabrica Abstracta: Utiliza una interfaz para crear familias de objetos relacionados o que dependen entre si, sin especificar sus clases concretas.

Se ha seleccionado “Fábrica Simple” debido a su carácter introductorio al uso de factorías, también porque la lógica necesaria para los objetivos del proyecto es alcanzada con este patrón. Otra de las ventajas es la posibilidad de evolucionar a una fábrica de mayor nivel.

Único

El patrón Singleton garantiza que una clase tenga una sola instancia, y proporciona un punto de acceso global a ella.

Una variable global hace accesible a un objeto pero no impide que se puedan crear múltiples instancias de objetos. Una mejor solución es hacer que sea la propia clase la responsable de su única instancia. La clase puede garantizar que no se pueda crear ninguna otra instancia (interceptando cada nueva petición de creación de objetos), también puede proporcionar un modo de acceder a la instancia.

Estructura.

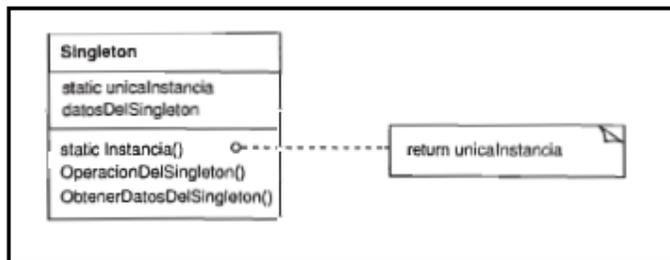


Figura A.1 - Estructura del Patrón Singleton (único).

En la figura anterior se puede ver la estructura de una clase que considera la existencia de una única instancia. Principalmente el procedimiento es declarar como *private* el constructor de la clase que se quiere restringir a una única instancia y a través de un atributo estático guardar la cantidad de clases instanciadas, de esta forma la clase podrá tener una nueva instancia solo dependiendo del valor de su campo estático.

Participantes.

Singleton: Define un método Instancia que permite que las clases clientes accedan a su única instancia. Esta clase Singleton puede ser también la responsable de crear su única instancia.

Estrategia

Define una familia de algoritmos, encapsula cada uno de ellos y los hace intercambiables. Permite que un algoritmo cambie su comportamiento independientemente de los clientes que lo usen.

Este patrón es recomendable cuando muchas clases relacionadas se diferencian solo en su comportamiento. Cuando se necesitan distintas variantes de un algoritmo. Cuando un algoritmo usa datos que los clientes no debieran considerar o una clase tiene definidos muchos comportamientos (múltiples sentencias condicionales).

Estructura.

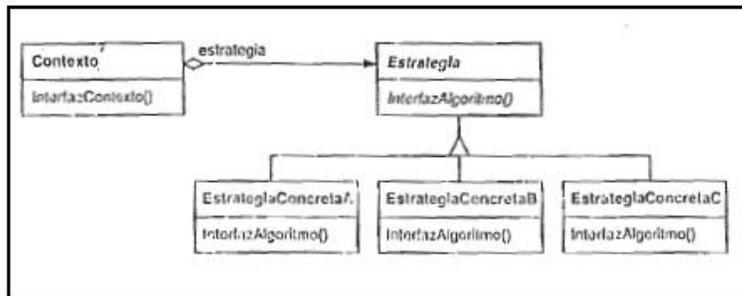


Figura A.2 - Estructura del Patrón Estrategia

En la figura anterior se puede ver como este patrón define una interfaz <<estrategia>> en donde declara los métodos que todas sus subclases deben implementar obligatoriamente pero la forma de realizar el algoritmo determinado es particular a cada una (Estrategias concretas de la figura).

Participantes.

Estrategia: Declara una interfaz común a todos los algoritmos permitidos. El contexto usa esta interfaz para llamar al algoritmo definido por una Estrategia Concreta.

EstrategiaConcreta: Son las clases que tienen que implementar la interfaz Estrategia pero dando una ejecución particular al método definido por esta.

Contexto: Mantiene una referencia a un objeto Estrategia y se configura con un objeto EstrategiaConcreta. Puede definir una interfaz que permita a la estrategia acceder a sus datos.

Observador.

Este patrón define una dependencia de uno a muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y se actualicen automáticamente todos los objetos que dependen de él.

Un ejemplo típico de la aplicabilidad de este patrón es en interfases de usuario donde existen múltiples vistas para una misma información, por ejemplo, se podrían tener un gráfico de barra y un gráfico de torta para representar los resultados de cierta encuesta, entonces al cambiar los resultados de la encuesta es necesario que se actualicen ambos gráficos. En este caso los gráficos dependen de la información que entrega una cierta encuesta (datos que podrían ser tomados de una planilla Excel), entonces ellos deben ser notificados cuando existen cambios en la información que están representando.

Este patrón es recomendable cuando una abstracción tiene dos aspectos y uno depende del otro. Cuando un cambio en un objeto requiere cambios en otros y no se sabe cuantos objetos deben cambiarse. Cuando un objeto debería ser capaz de notificar a otros sin hacer suposiciones sobre quienes son dichos objetos, En otras palabras, cuando se quiere bajo acoplamiento entre estos objetos.

Estructura.

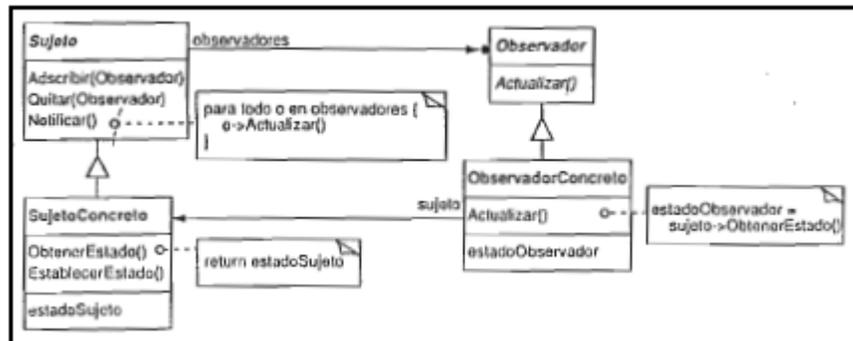


Figura A.3 - Estructura de Patrón Observer

En la figura anterior se muestra la estructura que tiene el patrón Observador, como sucede con la mayoría de los patrones de diseño se busca definir en una interfaz lo que cambia su comportamiento, en este caso pueden existir múltiples observadores concretos y también los objetos observados pueden ser mas de uno (sujeto concreto).

Participantes.

Sujeto: Conoce a sus Observadores. Un “Sujeto” puede ser observado por cualquier número de objetos “Observador”. Proporciona una interfaz para asignar y quitar objetos Observador.

Observador: Define una interfaz para actualizar los objetos que deben ser notificados ante cambios en un Sujeto.

SujetoConcreto: Almacena el estado de interés para los objetos ObservadorConcreto. Envía una notificación a sus observadores cuando cambia su estado.

ObservadorConcreto: Mantiene una referencia a un objeto SujetoConcreto. Guarda un estado que debería ser consistente con el del sujeto. Implementa la interfaz de actualización del Observador para mantener un estado consistente con el del sujeto.

Los siguientes son los patrones de diseño que no serán implementados en esta tesis, por lo tanto su descripción será más breve que las de los patrones anteriores.

Adaptador.

Convierte la interfaz de una clase en otra interfaz que es la que esperan los clientes. Permite que cooperen clases que de otra forma no podrían tener interfaces compatibles.

Este patrón se debería usar cuando:

Se quiere usar una clase existente y su interfaz no concuerda con la que se necesita

Se quiere crear una clase reutilizable que coopere con clases no relacionadas o que no han sido previstas, es decir, clases que no tienen porque tener interfaces compatibles.

Es necesario usar varias subclases existentes, pero no resulta práctico adaptar sus interfaces heredado de cada una de ellas. Un adaptador de objetos puede adaptar la interfaz de su clase padre.

En este proyecto se considera útil la incorporación del patrón Adaptador para proveer al Framework un mecanismo a través del cual pueda interactuar con otros sistemas construidos utilizando interfaces distintas. Por ejemplo se puede pensar en crear una página Web en donde a través de Web Services se puedan ejecutar procesos de Visión Artificial definidos en el Framework (o talvez simplemente acceder a los resultados de procesos ejecutados con anterioridad directamente sobre el Framework).

Comando.

Encapsula una petición en un objeto, permitiendo así parametrizar a los clientes con diferentes peticiones, hacer cola o llevar un registro de las peticiones, y poder deshacer las operaciones.

Este patrón busca contribuir al bajo acoplamiento de los objetos existentes en el sistema. A veces es necesario enviar peticiones a objetos sin saber nada de la operación solicitada o de quien es el receptor. Por ejemplo se puede pensar en que el Framework que queremos construir tendrá funcionalidades que podrán ser invocadas desde algún menú de la pantalla principal pero que además existirán botones en ciertas sub.-pantallas que invocarán la misma funcionalidad, en estos casos resulta muy importante tener centralizada la ejecución de estas solicitudes pues nos evitamos la redundancia de código y también fortalecemos el bajo acoplamiento que mencionamos al comenzar este párrafo, entre otras ventajas de utilizar este patrón.

Estado.

Permite que un objeto modifique su comportamiento cada vez que cambie su estado interno. Parecerá que cambia la clase del objeto.

Se recomienda utilizar este patrón cuando:

El comportamiento de un objeto depende de su estado, y debe cambiar en tiempo de ejecución dependiendo de ese estado.

Las operaciones tienen largas sentencias condicionales con múltiples ramas que dependen del estado del objeto. Este estado se suele representar por una o más constantes enumeradas. Muchas veces son varias las operaciones que contienen esta misma estructura condicional. El patrón Estado pone cada rama de la condición en una clase aparte, esto permite tratar el estado del objeto como un estado totalmente independiente de estados de otros objetos.

La aplicación de este Patrón se puede ver por ejemplo en la necesidad de manejar las conexiones a una BD en donde los estados de esta podrían ser “Escuchando”, “Establecida” o “Cerrada”, donde cada uno de estos estados tendrá diversos efectos sobre los objetos interesados en la BD. En el caso del Framework de VA que se construirá existe la necesidad de establecer conexión con diversos dispositivos de captura como lo pueden ser una cámara digital, cámara IP, una ubicación de Disco, etc. Entonces gestionar la forma en que los objetos son afectados por el estado de conexión a estos dispositivos hace recomendable aprovechar las ventajas de este patrón.

Diseño del Framework.

En esta sección se presentan 2 diagramas representativos de las clases a crear aplicando los patrones de diseño. Para mayor simplicidad el contenido de las clases se ha limitado a solo el necesario para reflejar el objetivo de los patrones y su aplicación dentro de sistema que queremos construir.

Los patrones de diseño de creación “Factoría Simple”, “Singleton” y el patrón de comportamiento “Strategy” se muestran en el diagrama de la siguiente figura. Los nombres de las clases se han designado siguiendo la recomendación de utilizar los nombres de los patrones para identificar donde se están aplicando y así poder ver de forma más clara que clases participan de acuerdo a los roles definidos en cada patrón.

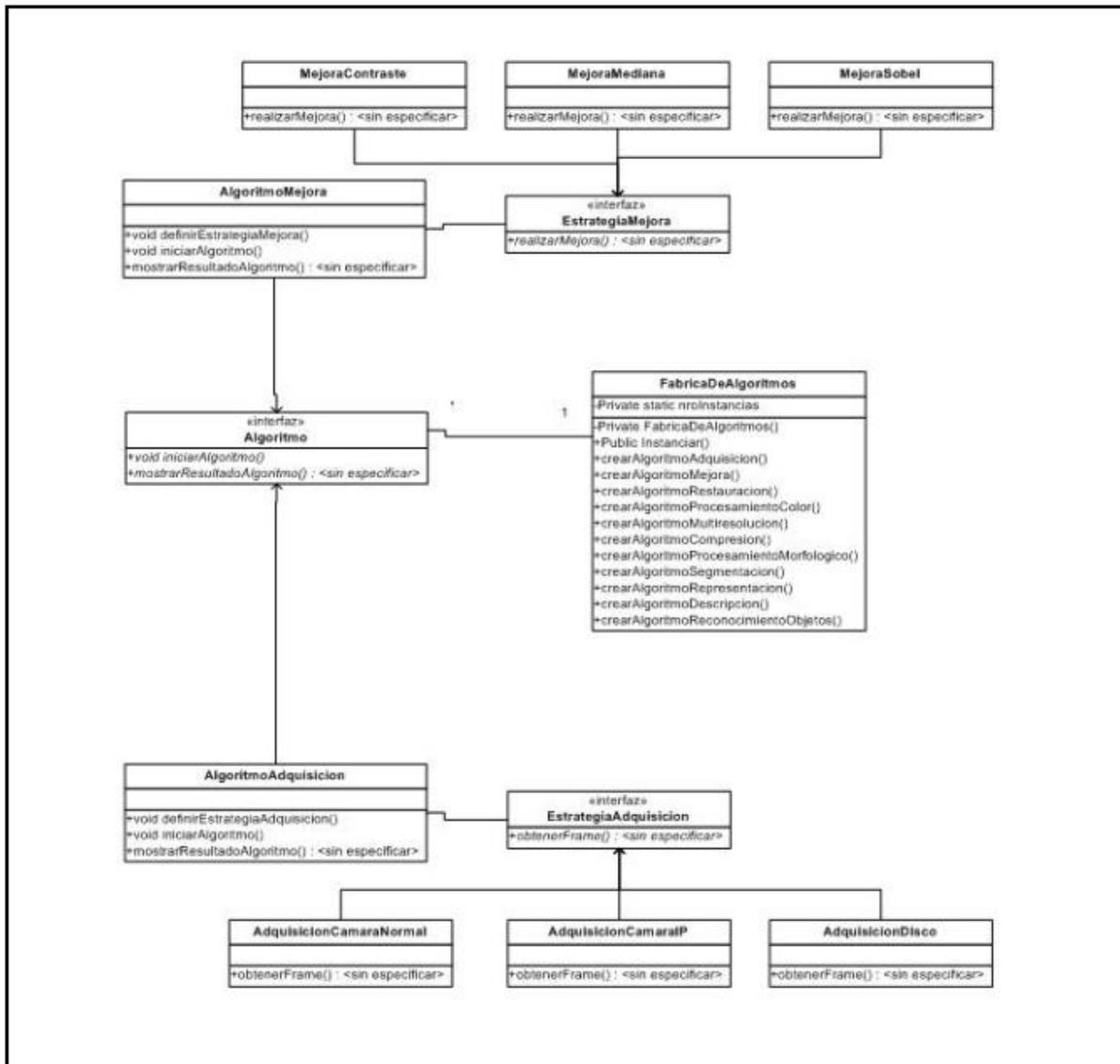


Figura A.4 - Diagrama de clases Patrones: Factoría Simple, Strategy y Singleton.

Observar que la clase FabricaDeAlgoritmos es la representativa del patrón “Factoría Simple”, ya que se encarga de crear diversos objetos (todos lo que implementen la interfaz <<Algoritmo>>).

Por otro lado la creación de algoritmos específicos sigue el patrón “Strategy”. Un ejemplo sería definir una interfaz <<estrategia “Z”>> que es implementada por subclases “EstrategiasConcretas”. La clase encargada de definir e instanciar la estrategia concreta a utilizar sería “AlgoritmoZ”. La aplicación de este patrón es útil para mantener por cada etapa del proceso de VA un grupo de algoritmos como alternativas a utilizar para esa fase en particular.

El tercer Patrón representado en el diagrama anterior es Singleton (Único) y se puede identificar al mirar la clase “FabricaDeAlgoritmos” que tiene su método constructor declarado como privado y también define un atributo “nroInstancias” que servirá para controlar la creación de una sola instancia. El método que se deja público para la utilización desde otras clases es “Instanciar()” el cual permitirá solicitar nuevas instancias de FabricaDeAlgoritmos lo que siempre estará sujeto al valor que tenga el atributo “nroInstancias” del Singleton. Este patrón es aplicado en esta clase ya que es la fábrica principal de objetos que utilizará el Framework por lo que el proceso estará centrado en ella y no es conveniente que pudiera en tiempo de ejecución duplicarse.

El siguiente diagrama que se mostrará es el que representa la utilización del Patrón Observer dentro del sistema.

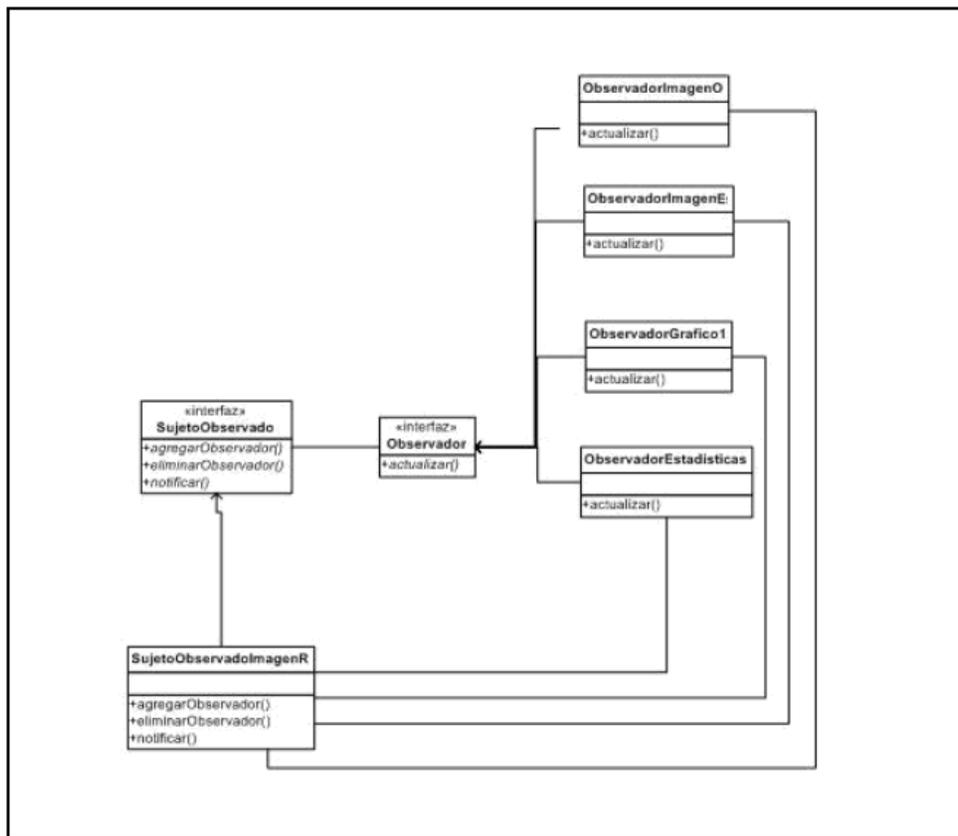


Figura A.5 - Diagrama de clases Patrón Observer

Este diagrama muestra un “SujetoObservadoImagenR” que es de interés para 4 observadores (ObservadorImagenO, ObservadorImagenE, ObservadorGrafico1, ObservadorEstadisticas) lo que significa que un cambio en “SujetoObservadoImagenR” deberá ser notificado por este a sus cuatro observadores.

Otro ejemplo de la aplicabilidad del diagrama presentado es hacer corresponder a “SujetoObservado” con un elemento de la pantalla del sistema que podría ser un frame (frame A) que contenga una lista de algoritmos disponibles, existirá entonces otro frame (frame B) que deberá ofrecer un listado de parámetros específicos dependiendo del algoritmo seleccionado en el primer frame. El listado de parámetros de configuración específicos a mostrar en el frame B correspondería a un Observador que debe ser notificado de los cambios del objeto del frame A.

Estos casos son la manera en que se implementaron los patrones a este proyecto. Sin embargo como hemos dicho al comienzo de esta sección los diagramas presentados buscan mostrar como se aplican los patrones dentro del dominio del sistema que estamos implementando. Por lo tanto las clases existentes en el sistema y sus atributos y métodos son mucho más numerosos de los graficados en este informe.

Interfaz del Framework

En esta sección se detalla la implementación del sistema. Como se ha descrito en secciones anteriores el Framework construido busca servir de herramienta en la búsqueda de procesos que den solución a problemas del ámbito de la visión artificial, de esta forma se necesita tener disponible varios algoritmos en cada fase de procesado para poder evaluar el comportamiento de cada uno. Lo anterior debido a que cada problema de VA tiene particularidades que harán algunos algoritmos más pertinentes que otros.

Pantalla Principal

A continuación se muestra la pantalla principal que es generada al ingresar al sistema:



Figura A.6 - Interfaz Principal Framework de Visión Artificial.

Esta Primera interfaz de usuario propone la selección de alguna de las funcionalidades disponibles en los distintos menús existentes:

Menú Archivo: Contiene opciones de ingreso a la pantalla principal de la aplicación, también esta pensado para guardar en el disco una estructura creada en un determinado proceso de VA, de la misma forma ofrecerá abrir un proceso de VA que ha sido guardado con anterioridad. Entre otros.

Menú Procesos: Ofrece tareas a realizar sobre el proceso de VA que este siendo desarrollado por el usuario, estas pueden ser inicializar la ejecución del proceso, detener la ejecución, ver los resultados generados por el proceso, etc.

Menú Algoritmos: Este menú esta pensado para ofrecer detalles de los algoritmos existentes en el sistema, de esta forma pueden ser consultados y estudiadas sus principales características dentro de la misma aplicación, en esta

opción también se explican los parámetros requeridos en algunos algoritmos y como interpretar los resultados que se visualizar pos-ejecución.

Menú Imágenes: En esta opción se deben configurar las imágenes sobre las cuales se realizará el proceso de visión artificial, las opciones disponibles son elegir una directamente desde alguna ubicación del disco, configurar una entrada de video o configurar una carpeta del disco que contenga una secuencia de imágenes a procesar.

Menú Ayuda: Este menú tiene disponible instrucciones sobre el uso de la aplicación y referencia a la versión de la aplicación además de información acerca del autor del proyecto.

Hay que señalar que en este proyecto de tesis no se pretende terminar el Framework con todas sus funcionalidades disponibles, ya que el principal objetivo está relacionado con utilizar técnicas de visión artificial de alto nivel que permitan una detección robusta de objetos en base al reconocimiento de patrones, entonces los mayores esfuerzos están puestos en permitir sobre una interfaz base desarrollar la incorporación de estos algoritmos en procesos que entreguen altas prestaciones.

Pantalla de Creación de Procesos de VA

La principal funcionalidad entregada por el Framework desarrollado es la definición de procesos de VA con algoritmos específicos que ayuden a optimizar los resultados obtenidos. A continuación se muestra la pantalla disponible:

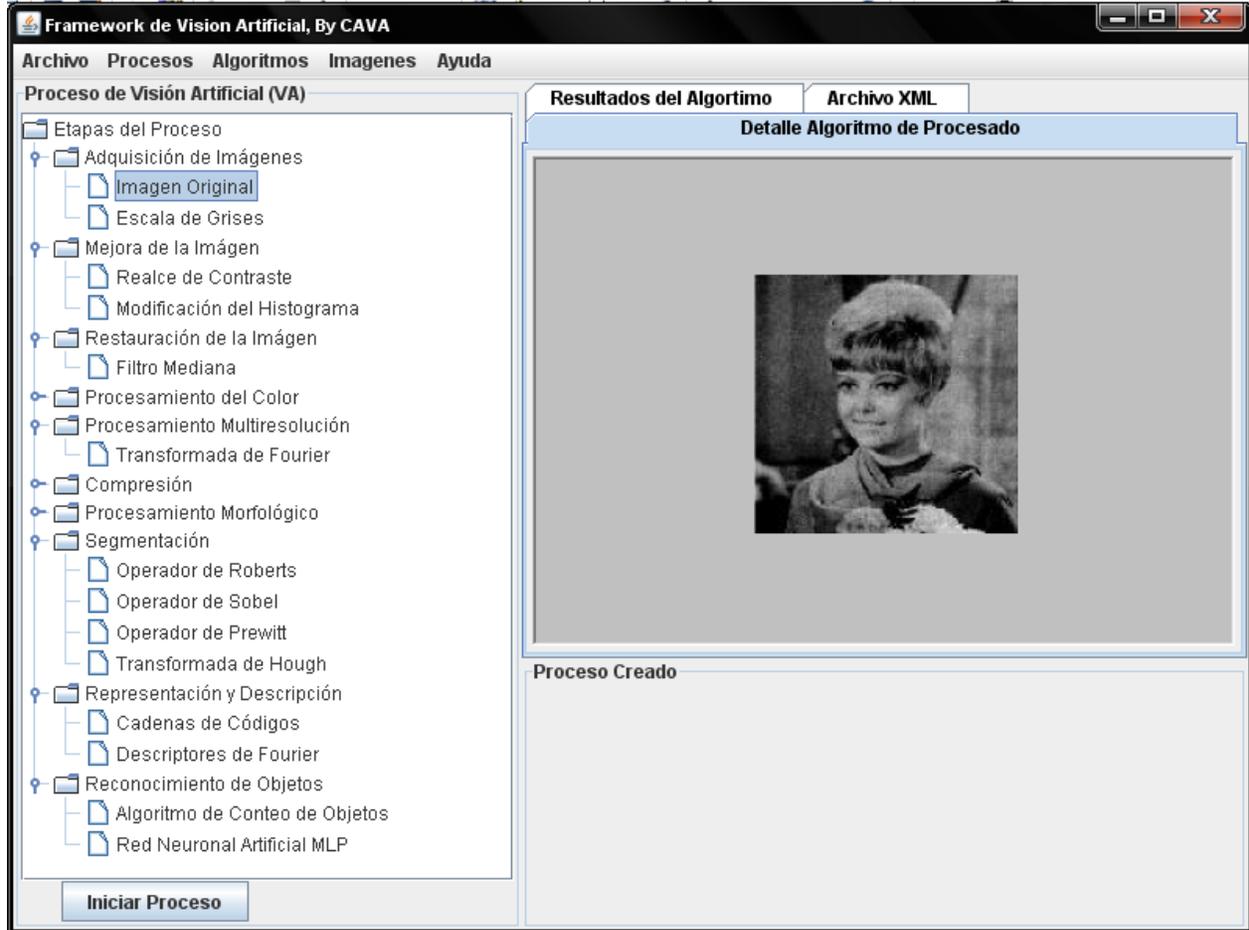


Figura A.7 - Interfaz de Usuario Framework de Visión Artificial.

Como se puede observar en la pantalla mostrada existen principalmente 3 secciones:

- Etapas de un Proceso de Visión Artificial.
- Configuración del Algoritmo de Procesado.
- Proceso de VA Creado.

La sección “Etapas de un Proceso de Visión Artificial” lista todas las etapas que pueden ser consideradas en un proyecto de visión artificial, estas son presentadas en forma de árbol donde cada una tiene un subnivel asociado en donde se muestran aquellos algoritmos que pueden ser aplicados en la etapa seleccionada. Luego de que el usuario realice su selección se habilitará en la sección del lado derecho un panel de configuración del algoritmo correspondiente para que pueda ser incluido en el proceso y establecidos sus parámetros (cuando corresponda).

La sección “Configuración del Algoritmo de Procesado” del lado derecho de la pantalla muestra el panel de configuración correspondiente al algoritmo seleccionado en el árbol del proceso. En esta sección también se muestra una descripción del algoritmo que seleccione el usuario. También existe un panel que muestra la imagen original que será procesada, para visualizar esta imagen se debe seleccionar en el árbol la etapa de adquisición de imágenes, nodo Imagen Original.

La tercera sección disponible en la pantalla de creación del proceso es “Proceso de VA Creado” en donde cada vez que se decida agregar un nuevo algoritmo al proceso este aparecerá en esta sección, entonces se mantiene la visualización de todos los algoritmos agregados informando además al usuario del orden en que serán ejecutados.

La segunda sección “Configuración del Algoritmo de Procesado” descrita anteriormente también contiene otros 2 paneles que están en forma de pestañas, esto es debido a que se encuentran inhabilitados mientras no se realice la ejecución del proceso o no se hayan agregado algoritmos:



Figura A.8 – Paneles sección Configuración del Algoritmo de Procesado.

Se tiene entonces la pestaña inicial que está visible por defecto “Detalle Algoritmo de procesado” la cual muestra el panel de configuración del algoritmo seleccionado en el árbol del proceso (lado izquierdo). También esta una segunda pestaña “Resultados del algoritmo” que igualmente reacciona a la selección hecha en el árbol del proceso mostrando los resultados obtenidos en el algoritmo seleccionado, también en esta nueva solapa se mostrarán gráficos de dispersión de los datos observados en la imagen al igual que histogramas que ayuden al usuario a formular alguna conclusión respecto de los resultados obtenidos. El tercer panel es “Archivo XML” y aunque no esta disponible en esta versión del sistema su objetivo es definir la estructura del proceso creado en formato XML, de esta

forma se logra la interoperabilidad del sistema con otros externos que quieran utilizar un determinado proceso definido en el Framework. Con este archivo XML también se pretende lograr el respaldo de procesos definidos y su posterior carga a través del menú “Archivo”.

Con todas las funcionalidades descritas en los párrafos anteriores se puede definir y probar la ejecución de diversos modelos que resuelvan problemas de visión artificial, así también se tiene una plataforma que permite la visualización inmediata de los resultados obtenidos en cada una de las etapas del proceso ayudando de esta forma al desarrollador a seleccionar aquellos algoritmos que entreguen mejores prestaciones.

Varias funcionalidades quedan pendientes de implementar en una futura versión del Framework pero como se ha dicho en otras secciones de este documento lo entregado actualmente cumple con nuestro requerimiento servir de apoyo en las pruebas de algoritmos de alto nivel en el campo de la visión artificial.