



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

PROGRAMA DE MAESTRÍA Y DOCTORADO EN
INGENIERÍA

FACULTAD DE INGENIERÍA

SIMULACIÓN DE VUELO Y RECONOCIMIENTO
DE IMÁGENES AEROSPACIALES

T E S I S

QUE PARA OPTAR POR EL GRADO DE:

MAESTRO EN INGENIERIA

INGENIERÍA ELÉCTRICA – INSTRUMENTACIÓN

P R E S E N T A :

KAREN LUCERO ROLDÁN SERRATO



TUTORES:

DRA. GRACIELA VELASCO HERRERA
DRA. TETYANA MYKOLAIVNA BAYDYK

2011.



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

JURADO ASIGNADO:

Presidente: DR. KUSSUL ERNST MIKHAILOVICH

Secretario: DR. ESAÚ VICENTE VIVAS

Vocal: DRA. VELASCO HERRERA GRACIELA

1er. Suplente: DRA. BAYDYK MYKOLAIVNA TETYANA

2do. Suplente: DR. LARA ROSANO FELIPE DE JESÚS

Lugar donde se realizó la tesis:

GRUPO DE COMPUTACIÓN NEURONAL DEL CENTRO DE CIENCIAS
APLICADAS Y DESARROLLO TECNOLÓGICO
UNAM
MÉXICO, D.F.

TUTORES DE TESIS:

DRA. GRACIELA VELASCO HERRERA

DRA. TETYANA MYKOLAIVNA BAYDYK

FIRMA

FIRMA

El presente trabajo fue desarrollado en el Grupo de Computación Neuronal del Centro de Ciencias Aplicadas y Desarrollo Tecnológico de la UNAM, bajo la tutoría de la Dra. Graciela Velasco Herrera y la Dra. Tetyana Baydyk , apoyada por: el Instituto de Geofísica, beca CONACYT 39781 y los proyectos: IXTLI IX101010, PAPIIT IN105909, PAPIIT IN10510-3, PAPIIT IN119610, CONACYT 135597, CONACYT 152008, ICyTDF 330/2009.

CONTENIDO

Capítulo 1 Generalidades	5
1.1 Panorama general.....	6
1.2 Objetivo general.....	6
1.3 Justificación del problema.....	6
1.4 Alcance.....	8
1.5 Metodología a seguir.....	9
Capítulo 2 Antecedentes	10
2.1 Historia del arte.....	11
2.2 Dinámica de una aeronave.....	12
2.2.1 Ecuaciones de fuerzas y momentos.....	17
2.2.2 Relaciones cinemáticas angulares.....	21
Capítulo 3 Simulación de vuelo	23
3.1 Desarrollo del simulador de vuelo.....	24
3.2 Modelo de aeronave.....	25
Capítulo 4 Acontecimientos de la vista aérea	28
4.1 Análisis de aeronaves no tripuladas para selección de áreas estático.....	29
4.2 Utilización de los primeros vuelos para obtención de imágenes.....	29
4.3 Aplicaciones de imágenes aéreas para censos estadísticos.....	31
4.4 Reconocimiento de imágenes geológicas.....	32
Capítulo 5 Visión computacional	33
5.1 Visión computacional como elemento de una aeronave moderna.....	34
5.2 Visión computacional aplicado a la Aeronáutica.....	35
5.3 Elementos del sistema.....	36
5.4 Obtención de imágenes.....	37
5.4.1 Extracción de las características de la Imagen.....	37
5.4.2 Red Neuronal Artificial.....	38
5.5 Aplicaciones de la visión computacional con el uso de imágenes satelitales.....	39
Capítulo 6 Base de Imágenes	42
6.1 Base de imágenes.....	43
6.2 Preparación de base de imágenes aeroespaciales.....	44
6.3 Reconocimiento de texturas.....	46
6.4 Marcado de imágenes.....	47
Capítulo 7 Clasificador neuronal	49
RSNC	
7.1 Redes neuronales como base del sistema de reconocimiento de imágenes.....	50
7.2 Estructura del clasificador neuronal	51
RSNC.....	
Capítulo 8 Resultados y conclusiones	59
8.1 Resultados y conclusiones.....	60
8.2 Trabajos a futuro.	65
Referencias.....	66
Glosario.....	70
Códigos.....	89

CAPÍTULO 1.

GENERALIDADES

1.1 Panorama general

La presente tesis forma parte de proyectos de investigación del grupo de Computación Neuronal, que se elabora en conjunto con CCADET e Instituto de Geofísica, las simulaciones de vuelo y el reconocimiento de imágenes aéreas y espaciales con tres texturas o clases de interés para el análisis: zona de cultivo, zona de bosque y zona urbana.

Los temas que contiene son: historia del arte como base teórica, la visión computacional y programación por computadora así como análisis de la dinámica de vuelo para una aeronave rígida.

1.2 Objetivo general

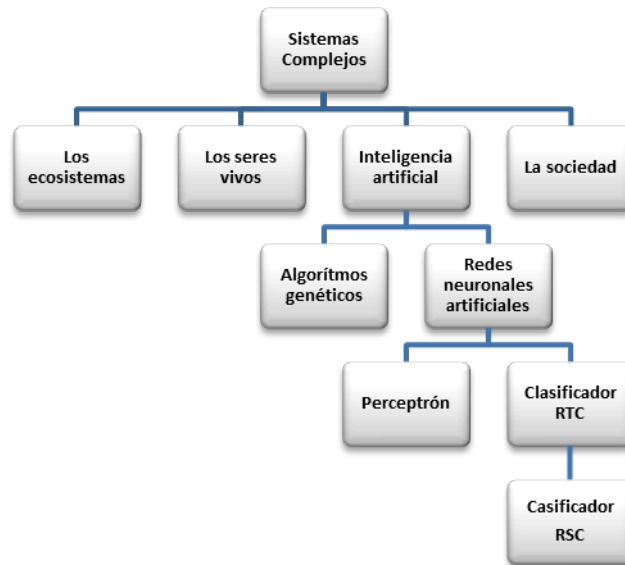
Analizar la dinámica de vuelo de una aeronave rígida para simulaciones de trayectorias en un vuelo, así como diseñar y desarrollar la programación de dos software por computadora, los cuales cumplan con las tareas de marcado y reconocimiento de imágenes aeroespaciales usando redes neuronales.

Dentro de esta temática se agrega un desarrollo de ambiente tridimensional integrado a simuladores de vuelo, que proporcionen una perspectiva cercana a la realidad. Estos ambientes están generados en 3D a partir de imágenes para generar un ambiente de simulación tridimensional. Siendo un apoyo visual (con imágenes estereoscópicas o anáglifos) del simulador. Para realizar este objetivo se sustentan los siguientes puntos:

- La simulación por computadora se usa en múltiples campos de ingeniería.
- Su utilidad se refleja en sistemas complejos, cuyo estudio por métodos convencionales es muy difícil, en muchos casos inabordable.
- Los sistemas de vuelo son un ejemplo de integración de sistemas complejos. En su desarrollo, la simulación supone una herramienta de inestimable valor.

1.3 Justificación del problema

En los sistemas complejos, los cuales están compuestos por varias partes interconectadas o entrelazadas cuyos vínculos crean información adicional no visible antes por el observador, se encuentra la inteligencia artificial, definida como la rama de las ciencias de la Computación, dedicada al desarrollo de agentes racionales no vivos, integrada por varias técnicas, entre ellas las redes neuronales artificiales que son un paradigma de aprendizaje y procesamiento automático inspirado en la forma en que funciona el sistema nervioso de los animales.



Desde su existencia, el hombre ha observado el cielo y ha soñado con poder volar y explorar el cosmos; desde la mitología griega sabemos que el hombre diseña e imita la naturaleza con elementos que le permitan elevarse. Volar ha revelado la verdadera faz de la tierra. Desde mediados del siglo XIX, cuando los fotógrafos ascendieron en globos por primera vez para captar panorámicas de las ciudades, hasta la era espacial, en que los satélites examinaban continentes enteros, la visión del planeta se ha agudizado y ampliado. La fotografía aérea ha abierto bellas perspectivas; ha probado ser un instrumento valioso de la ciencia.

Hace solamente 100 años que el hombre vuela por primera vez, el 17 de diciembre de 1903, realizando el primer vuelo a motor en la historia, un hecho que dio inicio a la era de la aviación y que cambiaría al mundo para siempre. No obstante a este acontecimiento ya se había tomado la primera fotografía aérea en 1858 por el fotógrafo francés Nadar desde un globo a 80 metros de altura. Entonces, en Europa se tomaron a principios del siglo XX muchas fotografías aéreas de monumentos desde globos, como por ejemplo de Stonehenge, en Gran Bretaña, y de Roma y Ostia, en Italia.

La Ciencia y Desarrollo de Tecnología en México ha sido a través de los años ha sido un tema de vital importancia, en donde la comunidad científica también ha estado presente en el desarrollo y estudio de las matemáticas, física e ingeniería entre otras ramas de la ciencia que marca la historia; actualmente la creación de una Agencia Espacial propia no es tema nuevo para el país, pero que nuevamente ha retomando importancia en el mundo científico, esto implica desarrollo de especialistas en distintas áreas de la ingeniería espacial que utilicen e incorporen el uso de nuevas herramientas y alternativas tecnológicas como son la simulación virtual, modelado. En este trabajo se realiza el estudio de aportaciones en simulaciones de trayectorias de vuelo y reconocimiento de imágenes aeroespaciales, como una alternativa de visibilidad en situaciones de emergencia para trayectorias de vuelo, así como software aplicado para proyectos de investigación y docencia que estén al alcance, que sean amables a los usuarios y dirigidos

- Realizar el programa de marcado con C++ Builder.
- Realizar el clasificador neuronal RSNC para el reconocimiento de las imágenes.

1.5 Metodología

En ésta tesis se aplicará la siguiente metodología:

- 1.- Sustentar y comprender la teoría básica para que ocurra el vuelo de una aeronave.
- 2.- Definición de los parámetros que integrarán el sistema representados a través de un modelo matemático.
- 3.- Análisis del sistema estático.
- 4.- Diseño del sistema en diagrama de bloques, con los elementos que intervienen.
- 5.- Análisis del control adaptivo como una forma de reducción y modificación de errores en el sistema y así lograr que el vuelo sea lo más deseable posible.

La tesis está integrada por dos partes que constituyen siete capítulos. La primera parte constituye el análisis de la dinámica de una aeronave rígida, con un sistema de referencia tierra-nave nave-vuelo. En una segunda parte se discuten la obtención de la base de imágenes, así como la programación por computadora para poder procesar y reconocer la base de imágenes, previamente obtenida, con tres texturas que reconoce el sistema: zona de cultivo, zona de bosque y zona urbana.

CAPÍTULO 2.

ANTECEDENTES

2.1. Historia del arte

Actualmente, las maquinas asombrosas del siglo XXI, van desde excavadoras que extraen 90 toneladas de tierra de una sola palada hasta un avión donde su cola mide más que un edificio de ocho pisos, se toman estrategias inteligentes para poder aprender a conducirlos, a través de la simulación.



Fig. 2.1 Simulador. a), b) automotriz c) vuelo

El concepto de simulación es muy cercano al ser humano, manifestándose en las distintas etapas de desarrollo desde la infancia hasta el adulto. Los niños simulan en sus juegos. El hecho de poder simular una cierta situación permite experimentar con situaciones del mundo real o, aun en el caso de que tales situaciones resulten irreales, experimentar sensaciones que de otro modo resultarían imposibles de sentir en el mundo real. En el ámbito científico los primeros experimentos de simulación tienen lugar tras la II Guerra Mundial, con la aparición de la computadora. Desde los experimentos de simulación realizados con computadoras analógicas durante los años 60 y 70 hasta la construcción de domos (Fig. 2.2) que acogen los simuladores con los que los pilotos se entrenan recreando las situaciones de un vuelo real, el concepto y los métodos de simulación han experimentado grandes cambios.



Fig 2.2 Simulador de vuelo analógico

A mediados de la década de los 80 nace la Vida Artificial, una disciplina en la que el concepto de simulación es redefinido y en la que los métodos clásicos de simulación son sustituidos por nuevas técnicas. Un ejemplo es el simulador *Swarm Simulation System* (Fig. 2.3) desarrollado en el Instituto de Santa Fe en EUA, una herramienta

basada en un sistema de agentes con los que es posible simular sistemas complejos en Vida Artificial. Durante los años 90 la Realidad Virtual alcanza apogeo, permitiendo esta técnica la simulación de mundos virtuales. A finales del siglo XX se construye *Hemispherium*, el simulador de Realidad Virtual más potente del mundo. Se trata de un domo de 6 metros de diámetro con siete proyectores de imágenes y sonido tridimensional gobernado por un sofisticado software diseñado por el Centro de Realidad Virtual de la Universidad de Teeside(Reino Unido).En Inteligencia Artificial la simulación del razonamiento que sigue un especialista, o la simulación del comportamiento dinámico de una red de neuronas biológicas por medio de redes neuronales artificiales, ilustran la importancia que tiene el concepto de simulación.

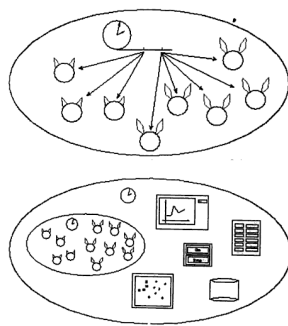


Fig. 2.3 Ejemplo de simulación

La realización de una simulación requiere un modelo que actúe como referente del sistema o fenómeno reales. Una vez construido el modelo, la realización de los *experimentos* de simulación en base a dicho modelo, permitirá a los investigadores en una cierta disciplina describir y predecir el comportamiento del sistema o fenómeno que ha sido observado en el mundo real (ver fig. 2.4)

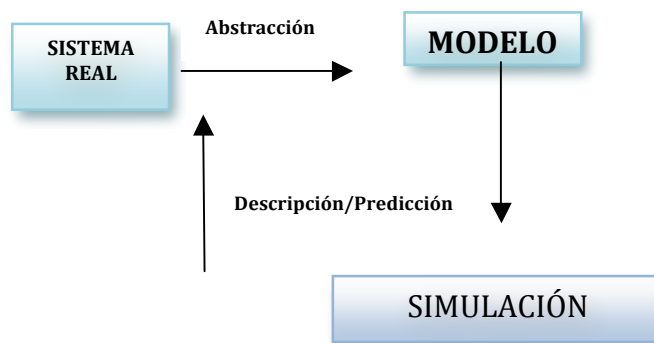


Fig. 2.4 Elementos para una simulación

Una vez construido el modelo, y en el caso de que la simulación se realice en una computadora, se obtiene el programa o software con el que se llevarán a cabo los experimentos de simulación.

Un simulador es un programa que convierte una computadora en un *laboratorio virtual* recreando el medio de experimentación en el que llevar a cabo los experimentos de simulación. Los simuladores pueden ser simples, por ejemplo un programa orientado a la simulación de sistemas dinámicos con ecuaciones diferenciales (fig. 2.5), o complejos. Este último caso el simulador incluirá los objetos, por ejemplo modelos de planas, neuronas artificiales, etc. que pertenezcan al dominio científico en el que se desarrollará la simulación.

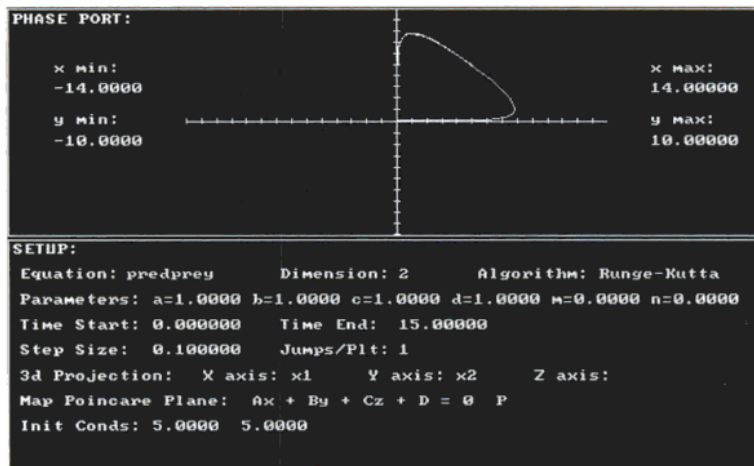


Fig. 2.5 Primeros simuladores dinámicos

2.2. Dinámica de la aeronave.

Definición: Una aeronave es un vehículo capaz de sustentarse en el aire y que puede desplazarse en un ambiente de forma controlada.

La fuerza de sustentación, que compensa el peso del vehículo, es generada por distintos fenómenos físicos dependiendo del tipo de aeronave:

- Los aerostatos es el empuje de Arquímedes.
- Los aviones y aeronaves de ala giratoria son las fuerzas aerodinámicas.
- Los aerodeslizadores es la sobrepresión de las capas de aire que se forman en la parte inferior de éstos.

Dos puntos importantes del vuelo son:

- Generar fuerzas sustentadoras
- Controlar el vuelo

La sustentación y control, están producidos por fuerzas aerodinámicas, esto es, fuerzas ejercidas por el aire sobre la aeronave. Para generarlas, se necesita que exista una velocidad relativa respecto al aire. En general, la acción del aire sobre una

aeronave produce fuerzas y momentos aerodinámicos que habrá de equilibrar, es decir:

- ✓ Equilibrio de fuerzas nos da un Vuelo a velocidad constante
- ✓ Equilibrio de momentos nos da Control de actitud

Para definir estas fuerzas y momentos, se requiere de un sistema de referencia:

- Sistema de ejes de un aeronave como un todo
- Sistema de ejes viento

Las componentes de las fuerzas aerodinámicas involucradas en este estudio son:

- Fuerza Lateral (C -Cross force)
- Sustentación (L -Lift)
- Resistencia (D- Drag)

Las componentes de momento son:

- Momento de cabeceo (Pitch)
- Momento de alabeo (Roll)
- Momento de guiñada (Yaw)

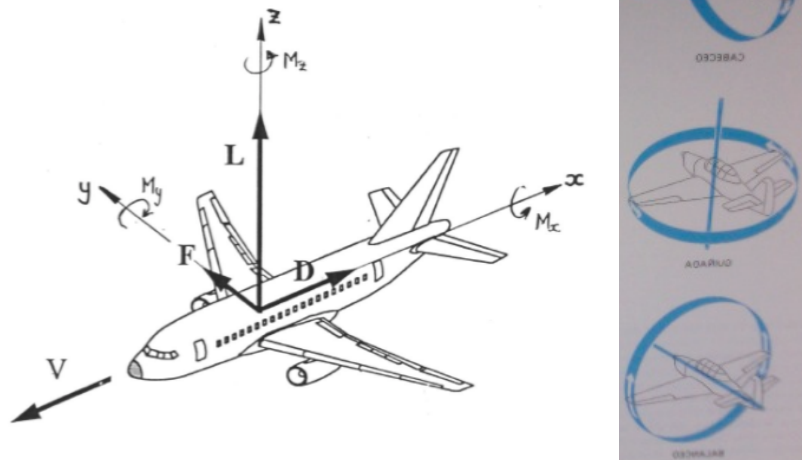


Fig. 2.6 Representación de las fuerzas y momentos aerodinámicos

C: Fuerza Lateral, L: Sustentación, D: Resistencia, M_x : Alabeo, M_y : Cabeceo, M_z : Guiñada

En un vuelo horizontal, rectilíneo y uniforme de una aeronave se requiere el equilibrio de fuerzas respecto al centro de masa de la aeronave, donde el empuje (T-Thrust)

contrarresta la resistencia aerodinámica (D-Drag), y la sustentación (L-Lift) contrarresta el peso (W-Weight) del avión.

T -generado por motor

L -generado por superficies aerodinámicas

D -generado por ala resto del avión

$L \gg D$

L/D eficiencia aerodinámica

Además se requiere que los momentos sean nulos. Para lograr el equilibrio es necesario tomar dos tipos de superficies, llamadas estabilizadoras:

- El estabilizador horizontal: Proporciona estabilidad en cabeceo. Además puede adoptar distintas configuraciones; en general se sitúa detrás del ala (en la cola), aunque a veces esté delante (en este caso recibe el nombre de canard) e incluso puede formar parte de la propia ala (en el caso de alas delta). Por lo que, es necesario para equilibrar el avión y contribuye a la fuerza de sustentación total.
- El estabilizador vertical: proporciona estabilidad en guiñada. Se ubica detrás del ala, pudiendo estar formado por un sólo plano o por dos, y en este caso, rectos o en forma de V.

Para el diseño de los estabilizadores es muy importante la ubicación del ala y de los motores.

Actuaciones de punto: estudio del movimiento de su centro de masas a lo largo de su trayectoria: vuelo horizontal, en subida, en descenso, en planeo, en viraje, despegue y el aterrizaje.

Actuaciones integrales: estudio del movimiento de su centro de masas entre los puntos inicial y final de su trayectoria para una carga de combustible dada:

Alcance: distancia recorrida respecto a tierra.

Autonomía: tiempo que la aeronave puede mantenerse en vuelo.

Una de las partes fundamentales que intervienen en la actuación de una aeronave son:

- Fuselaje
- Superficies sustentadoras: alas, dispositivos hipersustentadores (flaps, slats)
- Superficies estabilizadoras Estabilizador horizontal, estabilizador vertical (deriva)
- Superficie de control: Timón de profundidad, timón de dirección, alerones, spoilers
- Grupo moto propulsor.
- Tren de aterrizaje
- Sistemas y equipos de abordo.

Mando de vuelo.

Palanca de mando:

Movimiento longitudinal:

- Control del timón de profundidad.
- Cambia la sustentación que proporciona el estabilizador vertical, provocando un momento de cabeceo.

Movimiento lateral

Control de alerones

Genera un momento de balanceo provocan sustentación desigual en las dos semi-alas (deflexión asimétrica)

Pedales: Accionan el timón de dirección, provocando un momento de guiñada

Palanca de gases: Controla el empuje del motor.

Problemas de acoplamiento entre mandos.

SISTEMAS DE LA AERONAVE

SISTEMAS DE MANDOS DE VUELO:

Transmite las órdenes del piloto a las superficies de control: poleas, fly-by-wire, etc.

SISTEMA ELÉCTRICO:

Proporciona potencia eléctrica a: instrumentos indicadores, equipos de navegación y comunicaciones, electro mecanismos para accionamientos a distancia, apertura y cierre de válvulas de los sistemas de aire acondicionado, iluminación, etc.

SISTEMA DE COMBUSTIBLE:

Está formado por los depósitos y las redes de carga y de suministro al motor. Elementos de estas redes son las bocas de llenado, las bombas de alimentación, las conducciones, las válvulas y filtros, y los controles e indicadores. Actualmente los depósitos son integrales, es decir, forman parte de la propia estructura del avión.

SISTEMA HIDRÁULICO:

Este sistema se encarga de actuar numerosos sistemas del avión: los controles primarios de vuelo (alerones, timón de dirección y timón de profundidad), los controles secundarios (flapsy, spoilers), la retracción del tren de aterrizaje, los frenos de las ruedas del tren, el sistema de dirección de la rueda de morro, las compuertas de las bodegas, etc.

SISTEMA NEUMÁTICO:

Algunos aviones están equipados con sistemas neumáticos para lograr algunos fines similares a los logrados con el sistema hidráulico: actuar sobre frenos y dirección, abrir y cerrar compuertas. Este sistema no suele utilizarse como sistema primario de los aviones.

SISTEMA DE ACONDICIONAMIENTO DE CABINA:

Los aviones que vuelan a altitudes elevadas utilizan este sistema para presurizar la cabina, de manera que el proceso respiratorio sea posible, para que la temperatura del aire sea confortable y para mantener un control adecuado de la humedad.

SISTEMA AUXILIAR DE POTENCIA (APU).

Proporciona potencia eléctrica y neumática cuando los motores están parados.

SISTEMAS DE EMERGENCIA. Estos sistemas no son de uso continuado.

Sistema de oxígeno, sistema anti-hielo, sistema anti-incendios.

INSTRUMENTOS DE AERONAVE

Los instrumentos son el medio de comunicar datos entre los sistemas y la tripulación

1-Altímetro barométrico: del griego "βαρος": peso de y "μέτρον"metron: medida), que sirve para medir esta presión atmosférica.

2-Tacómetros: Un tacómetro (del griego, τάχος tachos = velocidad y μέτρονmetron = medida) es un dispositivo que mide la velocidad de giro de un eje, normalmente la velocidad de giro de un motor.

3-Temperatura: Del latín temperatūra, es una magnitud física que expresa el nivel de calor ya sea de un cuerpo, de un objeto o del ambiente. Dicha magnitud está vinculada a la noción de frío (menor temperatura) y caliente (mayor temperatura)

4-Variómetro o indicador de velocidad vertical: muestra al piloto dos cosas: a) si el avión está ascendiendo, descendiendo, o vuela nivelado; b) la velocidad vertical o régimen, en pies por minuto (f.p.m), del ascenso o descenso.

5-Indicador de flujo de fuel (combustible).

6-Reloj.

7- Indicador de la velocidad aerodinámica.

8-Indicador de cantidad de fuel (combustible).

9-Indicador de presión en cabina.

10-Indicadores de presión de líquido hidráulico

11-Indicador de Actitud,

12-Indicador de Situación Horizontal

- Compás(Brújula)
- Distancia y rumbo al punto
- Aguja de dirección e ILS

13-Anemómetro e indicador de mach

14-Ángulo de ataque

15-Acelerómetro



Fig. 2.7 Instrumentos de control de una aeronave

2.2.1 Fuerzas y momentos

Las fuerzas aerodinámicas sobre una aeronave provienen de dos fuentes:

- esfuerzos de presión
- esfuerzos de fricción

Polar parabólica

La sustentación y la resistencia que genera un avión están directamente relacionadas.

Polar del avión: función que relaciona el coeficiente de resistencia (C_D) con el de sustentación (C_L).

La polar del avión es fundamental para estimar correctamente las actuaciones del avión.

De forma general, el coeficiente de resistencia depende de:

- De C_L ,
- El números de Reynolds
- Del Mach
- La configuración del avión: no hay una sola polar sino varias según el segmento en el que se encuentre el avión.
 - Despegue
 - Crucero
 - Aterrizaje, etc.

La construcción de la polar se construye contabilizando las distintas partes por separado y sumándolas luego con factores de corrección.

Una aproximación que en muchos casos de interés proporciona buenos resultados es la polar parabólica:

$$C_D = C_{D0} + kC_L^2 \quad (1)$$

Donde:

C_{D0} , coeficiente de resistencia sin sustentación,

k , parámetro de resistencia inducida unitaria,

kC_L^2 , recibe el nombre de resistencia inducida por la sustentación.

Podemos decir que C_{D0} , tiene una contribución importante al cálculo de la resistencia debida a la fricción, donde se puede considerar:

- a) para considerar un avión completo y combinar de forma adecuada los correspondientes a cada parte del mismo:
 - a. Ala
 - b. Fuselaje
 - c. Cola
- b) Existe una interferencia aerodinámica que hace que la resistencia global no sea la suma de las resistencias de cada elemento por separado.

c) El C_{D0} , es conocido entre los valores 0.014 y 0.04

Y k es la resistencia inducida por la sustentación:

- a) La forma en la que la resistencia es producida por torbellinos de punta de ala.
- b) La modificación del campo de las velocidades sobre el avión debido a la variación de la fricción.
- c) k se puede expresar de la siguiente manera:

$$k = \frac{1}{\pi \Lambda \phi}, \quad (2)$$

donde

ϕ es un factor de eficiencia (Factor de Oswald[10]),

Λ es el alargamiento del ala.

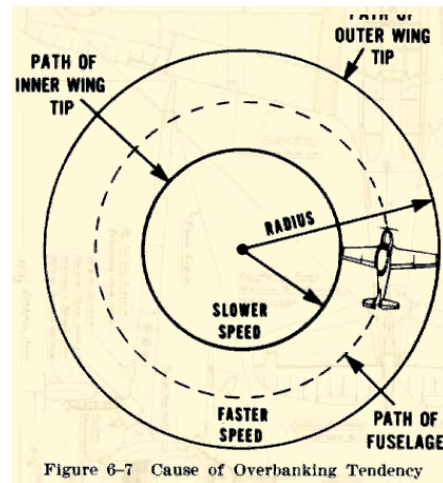
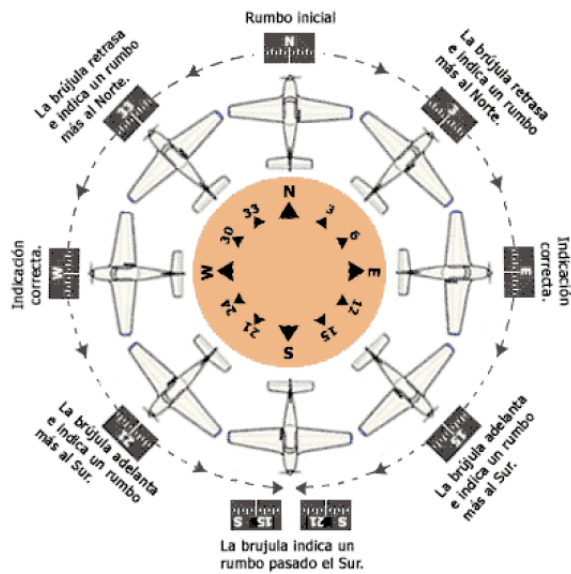
Actuaciones de la aeronave

Viraje

Al efectuarse un cambio de dirección, la aeronave tiende a mantener su movimiento en línea recta (por la inercia) y la fuerza centrífuga que trata de sacar la aeronave de su trayectoria. En el viraje equilibrado aparece con la fuerza centrípeta, la cual pone en equilibrio al sistema de banqueo. La fuerza centrípeta se obtiene inclinando hacia abajo el ala. Al efectuar la aeronave un viraje, la fuerza centrípeta se obtiene de la componente horizontal de la sustentación cuando éste inclina su ala hacia un lado, y con la sustentación depende del peso.

Para un viraje correcto, la componente vertical del levantamiento (L) debe ser de la misma magnitud que el peso de la aeronave, la componente horizontal (fuerza centrípeta) debe ser de la misma magnitud que la fuerza centrífuga.

Por lo anterior, para efectuar un viraje perfecto, la sustentación total debe ser mayor que el necesario para el vuelo horizontal. Para aumentar la sustentación será necesario aumentar el ángulo de ataque del ala, si se mantiene constante la velocidad, ó aumentar la velocidad sin variar el ángulo de ataque. En ambos casos será necesario aumentar la potencia del motor, ya que la resistencia al avance aumenta.



Planeo

Durante el vuelo en planeo, se considera que no existe fuerza de tracción, por lo que únicamente existen tres fuerzas, la sustentación, el peso y la resistencia al avance.

La distancia de planeo, es directamente proporcional a la altura en que se empieza el planeo e inversamente proporcional al ángulo de planeo.

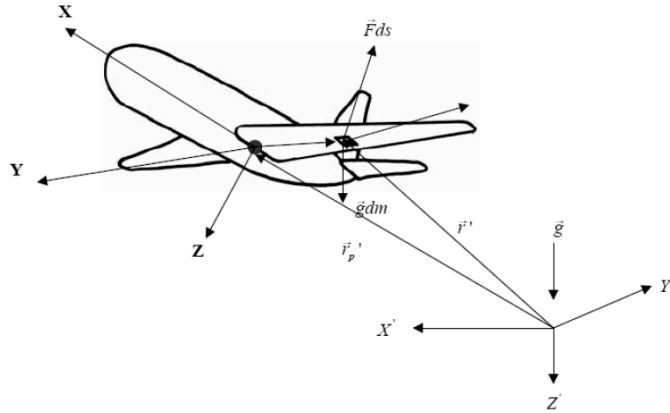
Sabiendo que: $L =$ Sustentación, $W =$ Peso y $D =$ Resistencia al Avance

El ángulo de planeo para cierta velocidad depende de las características aerodinámicas de la aeronave. El ángulo de planeo depende inversamente de la fineza ó rendimiento aerodinámico del avión, esto es, inverso a la relación L/D total.

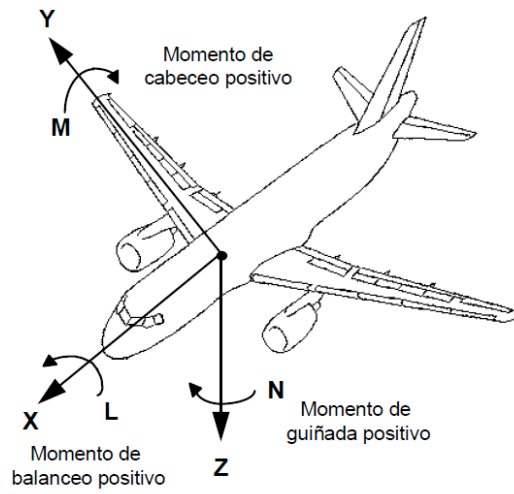
La velocidad de planeo depende, además del ángulo de planeo, de las características de la aeronave y de la densidad del aire. La velocidad de planeo disminuye a medida que la aeronave desciende, ya que la densidad del aire aumenta.

Mecánica de vuelo

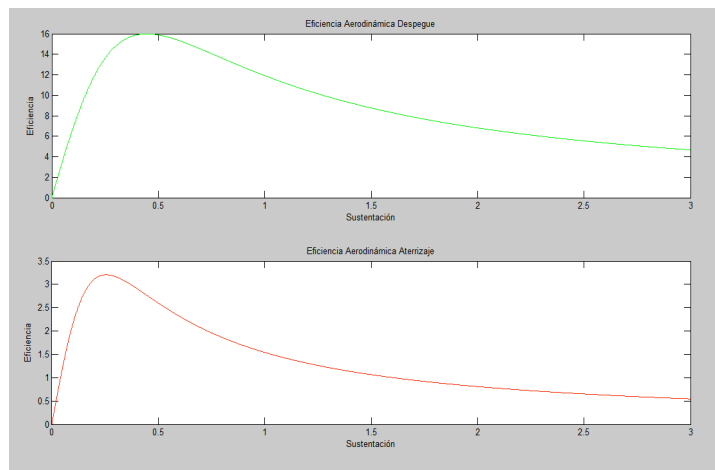
Para poder analizar la dinámica de vuelo, es necesario tomar fundamentos de mecánica de vuelo en un sistema como lo es un avión:



Si se sabe que la aeronave es: un cuerpo rígido que se somete a un conjunto de fuerzas externas, es simétrico.



Eficiencia aerodinámica



Tomando datos para una aeronave B 747:
 $C_D=0.014$ 0.04
 $\varphi=0.65$ 0.85

$C_L=0.253$

2.2.2 Relaciones cinemáticas angulares

Ángulos que definen la orientación de la trayectoria de la aeronave

- Ángulo de cabeceo de la velocidad o de trayectoria γ : ángulo formado por la velocidad V_g y el plano horizontal.
- Ángulo de guiñada de la velocidad o de rumbo χ : ángulo formado por la proyección de V_g sobre el plano horizontal y la dirección de referencia X_h .
- Ángulo de alabeo de la velocidad μ : ángulo formado por el plano $xwzw$ (plano LD) con el plano vertical que contiene a xw .

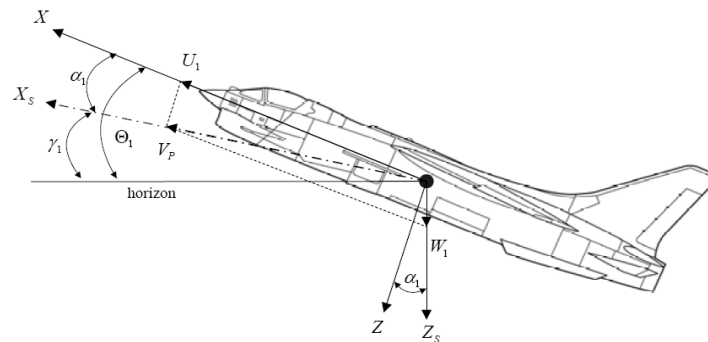


Fig. Ángulos que definen la trayectoria del avión.

- Ángulo de ataque α : ángulo que forma la proyección de V sobre el plano de simetría del avión con el eje x_b .
- Ángulo de resbalamiento β : ángulo que forma el vector V con el plano de simetría del avión.

Los ángulos α y β definen la orientación del sistema de ejes cuerpo respecto al de ejes viento: definen la orientación del viento incidente respecto del avión.

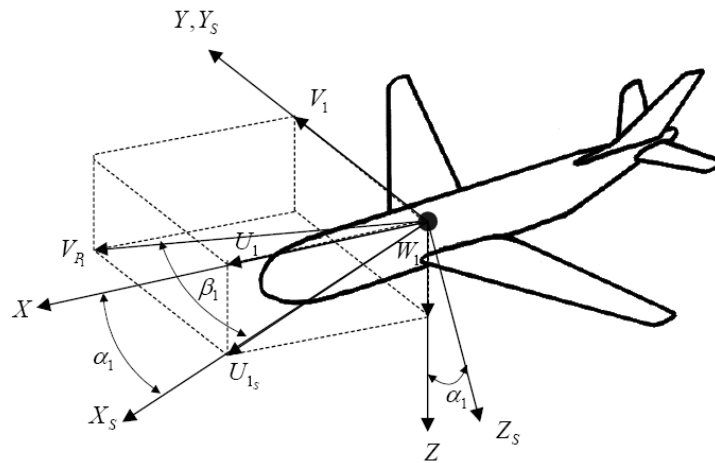


Fig. Ángulos de ataque y resbalamiento.

- Ángulo de ataque del empuje ε : ángulo que forma F_T (ver Figura 2.26) con el plano $X_w Y_w$.
- Ángulo de resbalamiento del empuje ν : ángulo que forma la proyección de F_T sobre el plano $X_w Y_w$ con el eje X_w .

Los ángulos ε y ν definen la orientación del empuje respecto de ejes viento.

V_g – velocidad absoluta

V – velocidad aerodinámica

V_w – velocidad del viento

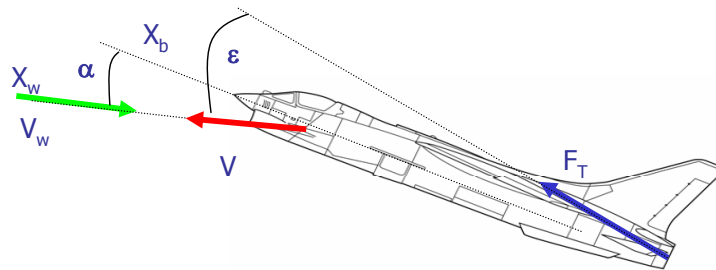


Fig. Orientación del empuje respecto al viento.

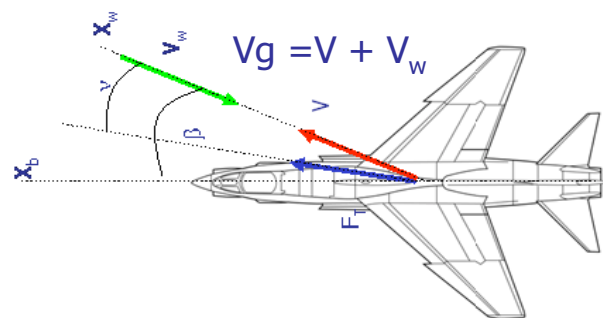


Fig. Representación de las componentes de velocidad absoluta.

CAPÍTULO 3.

SIMULACIÓN DE VUELO

3.1 DESARROLLO DEL SIMULADOR DE VUELO

El sistema electromecánico y su la posición de trayectorias están íntimamente ligados al simulador de vuelo, a continuación se presenta un diagrama que describe el simulador en cada una de las etapas de desarrollo.

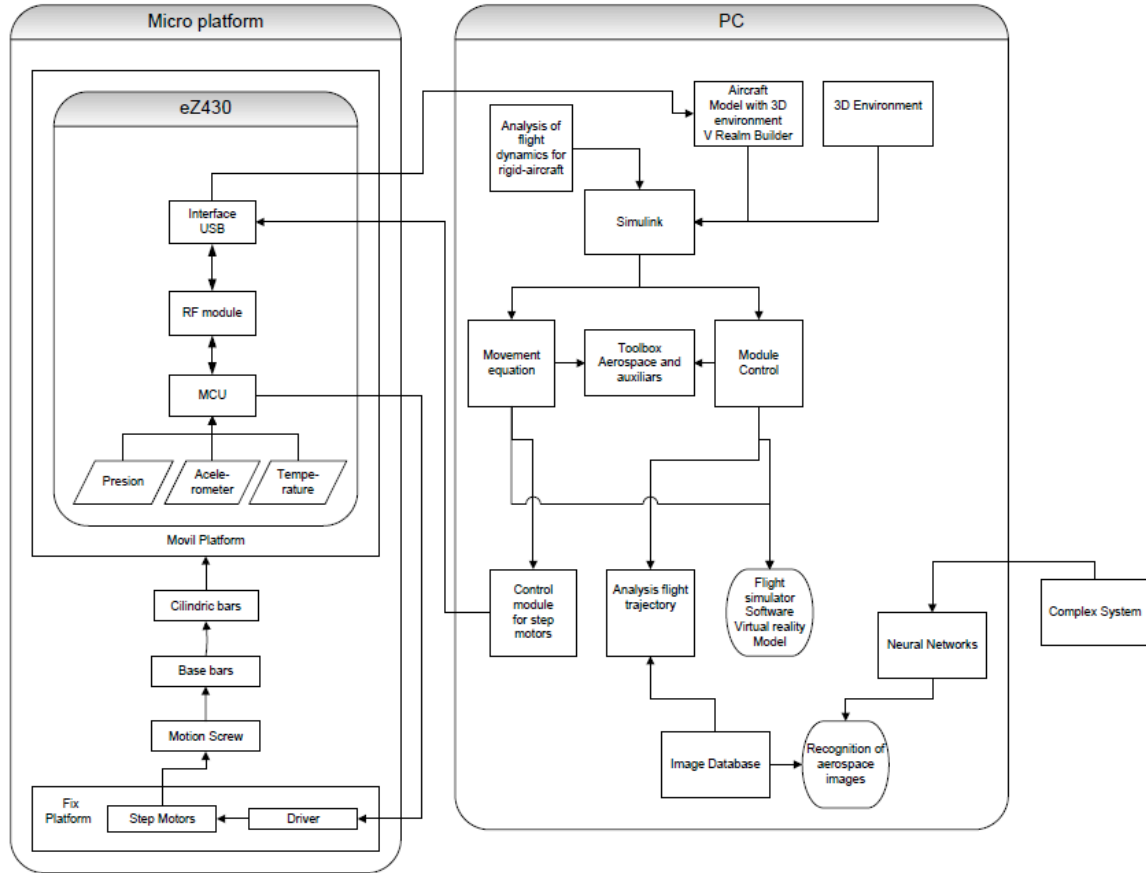


Fig. 3.1 Diagrama general del simulador de vuelo

3.2 Modelo de simulación

En cuanto a la parte del proyecto que se enfoca este trabajo de tesis es la parte de software.

A través del software para entornos gráficos V realm builder 2.0 siendo compatible con Matlab 2010a, se implementó un escenario para simulaciones de vuelo en 3D a una aeronave del tipo B747 en un entorno 3D (Fig. 3.1). Han sido precargados diferentes escenarios 3D con la aeronave, con el objetivo que el usuario de la simulación pueda elegir la situación de la simulación, es decir, si su trayectoria con vista urbana o en pleno vuelo entre nubes (Fig. 3.2).

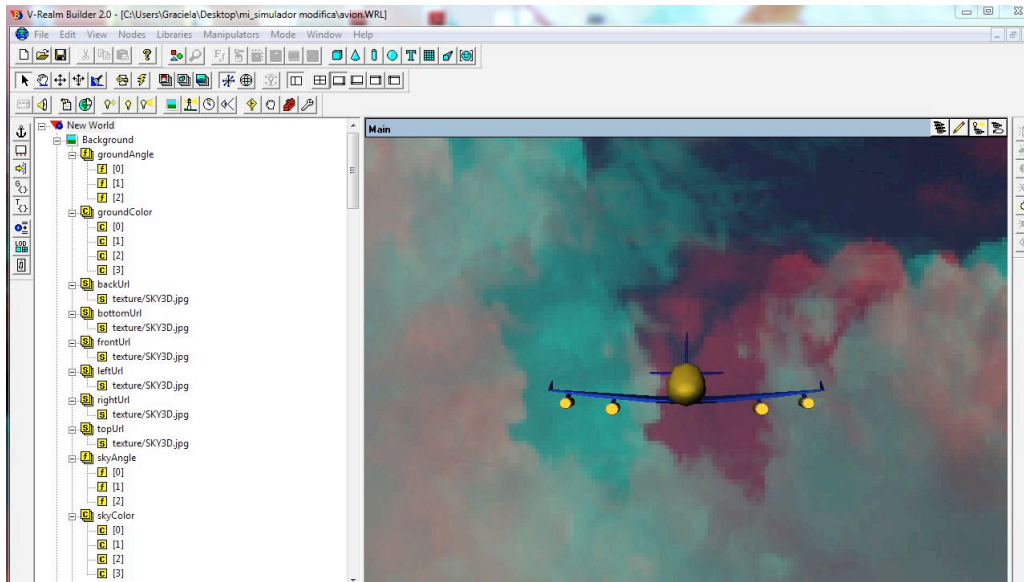


Fig. 3.2. Vistas de la aeronave con entorno 3D

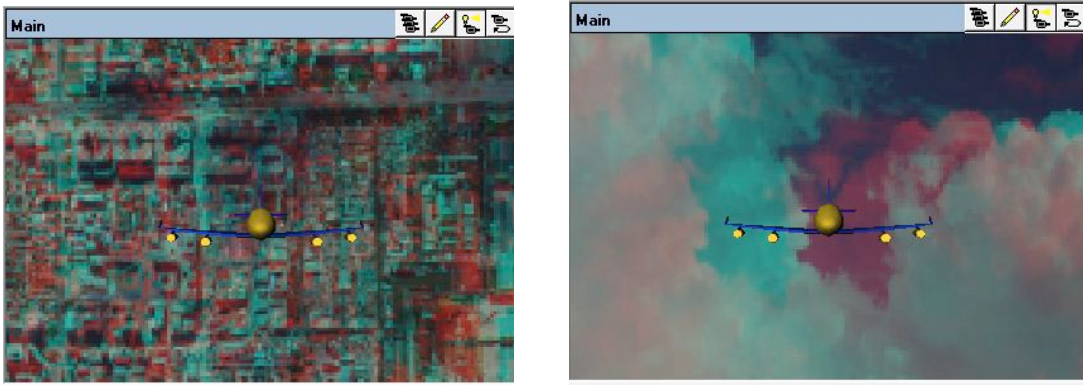


Fig. 3.3. Tipos de entorno en simulación 3D

3.3 Implementación del modelo en Matlab

El modelo de aeronave se exportó a Matlab 2010a y Simulink para simularlo. Se usaron varios módulos precargados de Matlab 2010a y también se hicieron nuevos, como es el caso del módulo de control y algunos elementos aerodinámicos. El modelo se insertó en un diagrama de bloques para poder unir los elementos de la simulación.

Hay dos fases que integran la simulación: la que manipula el modelo y la denominada de “autopiloto”. La primera es la que tiene los parámetros de manejo de cámara, ecuaciones de movimiento y manejo de simulador en 3D. La segunda es la que tiene los módulos de control y de control adaptivo.

El sistema en Matlab 2010a queda de la siguiente manera.

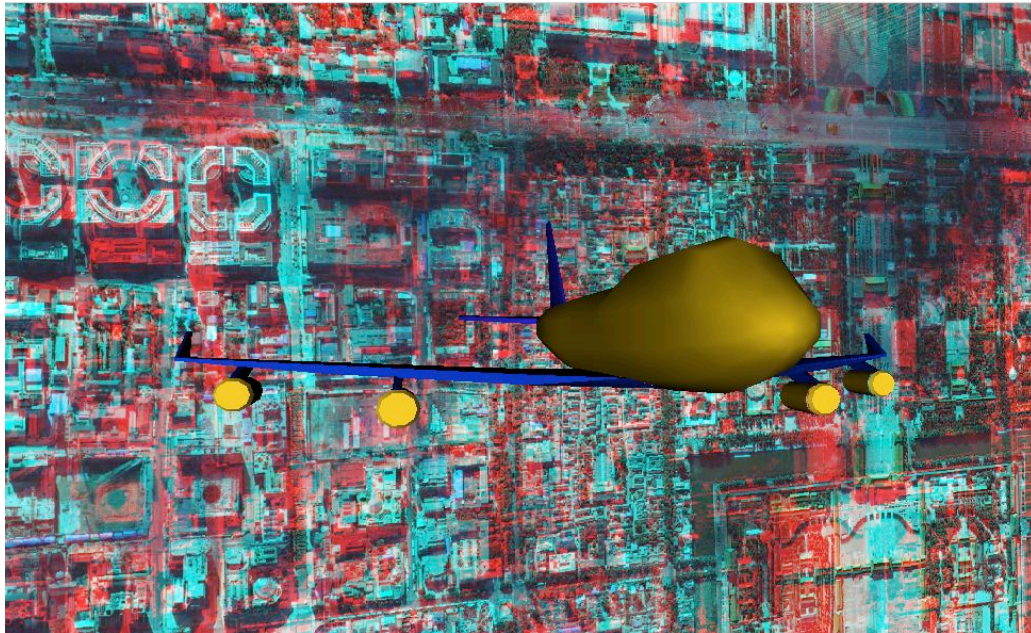


Fig. 3.4. Simulación 3D en Matlab

CAPÍTULO 4.

ACONTECIMIENTOS DE LA VISTA AÉREA

4.1 Análisis de aeronaves no tripuladas para selección de áreas geográficas.

Desde a mediados del siglo XIX, cuando los fotógrafos ascendieron en globos por primera vez para captar panoramas de las ciudades, hasta la era espacial, en que los satélites examinan continentes enteros, la visión del planeta se ha agudizado y ampliado.

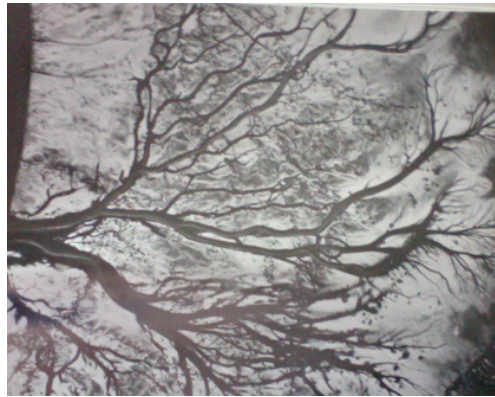


Fig. 4.1 Foto aérea de canales entre EU y México, siglo XIX

La fotografía aérea ha abierto bellas perspectivas; ha probado ser un instrumento valioso de la ciencia. En vistas sorprendentes, ver figura 1, las cámaras registran procesos tan graduales y vastos como:

La redistribución de la tierra por ríos y mares.

El movimiento de los heleros.

Obtención de nuevos datos.

Descubrimiento de huellas de culturas desaparecidas.

Levantar censos precisos de la vida silvestre.

Migración de los animales.

Fijar temporadas de caza.

Observar la acción erosiva del mar.

4.2 Utilización de los primeros vuelos para obtención de imágenes.

Desde una altura de 4300 metros, ayudan a los EUA y México a determinar dónde deben dragarse los canales movedizos del río o dónde deben apuntalarse con diques de protección los bordos o las riberas en peligro.

Para el arqueólogo, la fotografía aérea ofrece nuevos atisbos de tesoros ya conocidos y ha sacado nuevos indicios de la vida en tiempos antiguos. La ciudad etrusca de Spina, no lejos de la actual Venecia, es uno de esos descubrimientos. Spina, puerto próspero del siglo VI a. C., fue devastada y destruida cuando crecía el poderío romano. Sus edificios y vías fluviales se hundieron en la oscuridad de las marismas. Su

emplazamiento permaneció en el misterio hasta 1956, cuando un arqueólogo italiano, examinando las fotografías aéreas del litoral, descubrió en ellas el plano de la ciudad.



Fig. 4.2 Foto aérea plano de la ciudad Italia, 1956.

Hasta que la fotografía proporcionó los instrumentos, levantar un censo de las poblaciones animales, era al inicio un trabajo casi imposible debido al inmenso número de inquietas poblaciones que había que contar. Pero hoy el intérprete de fotos puede afirmar que los puntos negros o puntos blancos son animales de ciertas especies.



Fig. 4.3 Foto aérea rebaño de uapitis en Alaska.

Las fotos aéreas ayudan a los ingenieros a proteger zonas en amenaza.

En 1950, unos investigadores canadienses, al examinar fotografías aéreas topográficas de Ontario, hallaron una cicatriz enorme y casi perfectamente circular en el terreno, debido a un hoyo abierto por un meteorito gigantesco que estallo al chocar con la Tierra hace unos 500 millones de años. El examen de más de 400,000 fotografías del estudio aéreo de Canadá reveló las más cicatrices de meteoros.

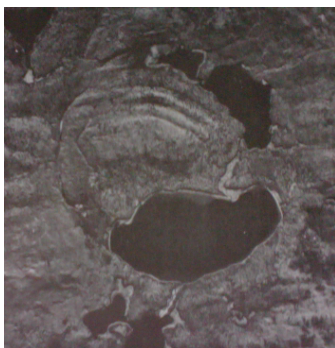


Fig. 4.4 Crater Brent Ontario; 3,2 KM de anchura ,500

Los científicos de la NASA del centro Goddard Space Flight Center, en Greenbelt (Maryland), se encarga de supervisar y analizar las mediciones de estos satélites. EFE.

4.3 Aplicaciones de imágenes aéreas para censos estadísticos.

Las imágenes aeroespaciales, permiten levantar censos precisos de la vida animal. Estos inventarios, principalmente cuando están respaldados por estudios periódicos en tierra, como se hizo con las focas y gansos, Fig. 4.5, dan datos valiosos sobre la migración de los animales y ayudan mucho a los gobiernos a fijar cada año las temporadas de caza; así como a determinar el número de piezas que pueden cobrarse según la cifra de las diversas especies silvestres.

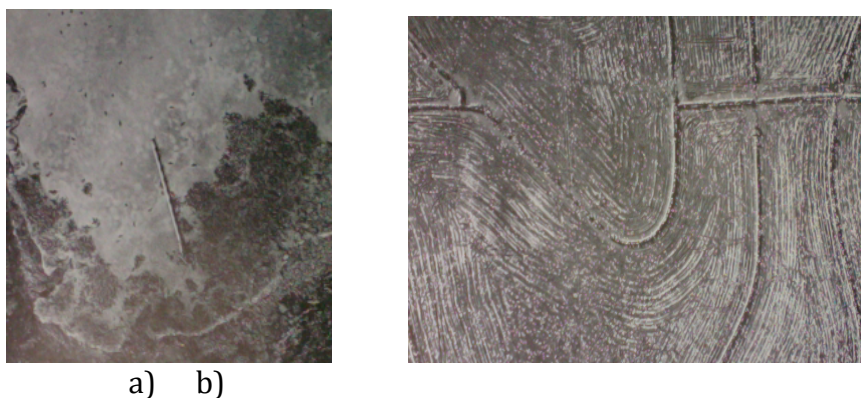


Fig. 4.5 Imágenes aéreas para inventario. a) Focas en un criadero de Alaska, b) Parvada de gansos de las nieves en un campo arado.

4.4 Reconocimiento de imágenes geológicas.

Hace más de cien años, cuando los geólogos comenzaron a estudiar los movimientos de glaciares, hundían pértigas en el hielo y volvían periódicamente para notar hasta dónde habían sido dobladas por la pesada corriente. Esto representaba peligro para la investigación. Con ayuda de la fotografía aérea, la

glaciología se ha convertido hoy, “en una agradable fusión de arte, ciencia y trabajo”. Los heleros se pueden señalar en el mapa desde el aire, y detallar sus avances o retrocesos, desde el aire, el cambio, su progreso majestuoso se percibe con facilidad y se observa de una manera sutil y bellísima.



Fig. 4.6 Glaciar Yanert, en Alaska.

CAPÍTULO 5.

VISIÓN COMPUTACIONAL

5.1 Visión computacional como elemento de aeronave moderna.

En los humanos, el sistema de visión provee de información detallada acerca del entorno que puede ser utilizada en una gran variedad de formas, tales como el reconocimiento de objetos y formas, comunicarse, guiar movimientos, identificar personas, reconocer gestos, encontrar rutas, manipular objetos y una gran variedad de tareas [1].

Por lo que la visión por computadora es un área que intenta aprovechar este sistema. Considerando el sistema visual como un dispositivo que almacena una banda de radiación electromagnética como una entrada y utiliza esto para obtener conocimiento acerca de los objetos que se encuentran en el entorno, que emiten y reflejan estas radiaciones, es posible obtener información de gran interés. Las fotos aeroespaciales que se obtienen y procesan para saber el entorno y usar la información aplicando la visión computacional. La visión computacional, que ha emergido como una disciplina propia basada principalmente en las matemáticas y ciencias de la computación, consiste en hacer que una computadora vea. Esto, sin embargo, es todavía un problema no resuelto [2].

El objetivo de la visión computacional respecto a una aeronave es tomar decisiones útiles acerca de los objetos físicos reales del mundo (de escenas) en base a imágenes adquiridas digitalmente [3]. Por lo tanto, la tarea de la visión computacional es la construcción de descriptores de la escena en base a características relevantes contenidas en una imagen.

Actualmente, en los programas aeroespaciales (Fig. 5.1), tal como *Landsat* de los Estados Unidos fue lanzado con el propósito de realizar observaciones terrestres, cuya aplicación es la investigación del cambio climático global y aplicaciones en agricultura, recursos acuíferos, crecimiento urbano, geología, reforestación, planeación regional, educación y seguridad nacional.

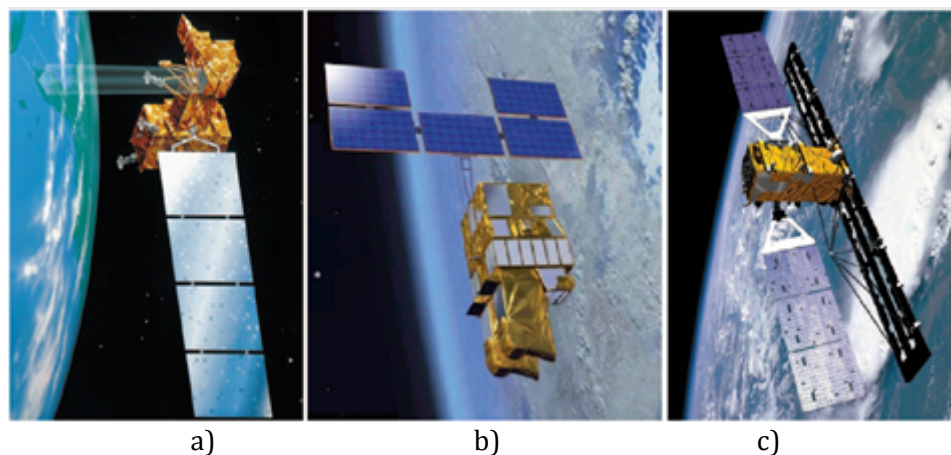


Fig. 5.1 Aeronaves. a) satélite Landsat-7, b) satélite SPOT-5, c) satélite Radarsat

El sistema *SPOT* fue diseñado por la agencia espacial Francesa, con Bélgica y Suecia como sus socios minoritarios. Provee imágenes terrestres para diversas aplicaciones

como la agricultura, cartografía, mapeo catastral, estudios ambientales, planeación urbana, telecomunicaciones, reforestación, mapeo del uso y cobertura del suelo, vigilancia, evaluación de desastres naturales, administración de riegos de inundación, exploración de petróleo y gas, geología e ingeniería civil. Así como, el programa *Radarsat* es un sistema de satélites de percepción remota Canadiense, orientado al estudio de regiones polares, apoyo a la navegación marítima, identificación de recursos naturales, administración de la agricultura y recursos acuíferos, y el monitoreo del cambio climático [4].

5.2 Visión computacional aplicado a la Aeronáutica.

El sueño de los autómatas inteligentes se remonta en la antigüedad, la primera en el contexto de las computadoras digitales fue desarrollado por Turing alrededor de 1950. Desde entonces, este sueño ha llevado a cabo principalmente por los pioneros en el campo de la inteligencia artificial, cuyo objetivo es dotar a las computadoras con capacidades de procesamiento de la información comparables a los de los organismos biológicos. Desde el principio, uno de los objetivos de la inteligencia artificial ha sido dotar a las máquinas con la capacidad de hacer frente a los estímulos sensoriales [5].

La visión computacional es la construcción de descripciones explícitas y significativas de los objetos físicos a partir de imágenes. La comprensión de la imagen es muy diferente de procesamiento de imágenes, los estudios de imagen a las transformaciones de imagen, no una descripción explícita [6]. Las descripciones son un requisito previo para el reconocimiento, la manipulación, y el pensamiento objetos. Pero la representación de la visión computacional a una imagen, es acuerdo al enfoque de la aplicación, es decir, al tipo de reconocimiento de la imagen (Fig. 5.2).

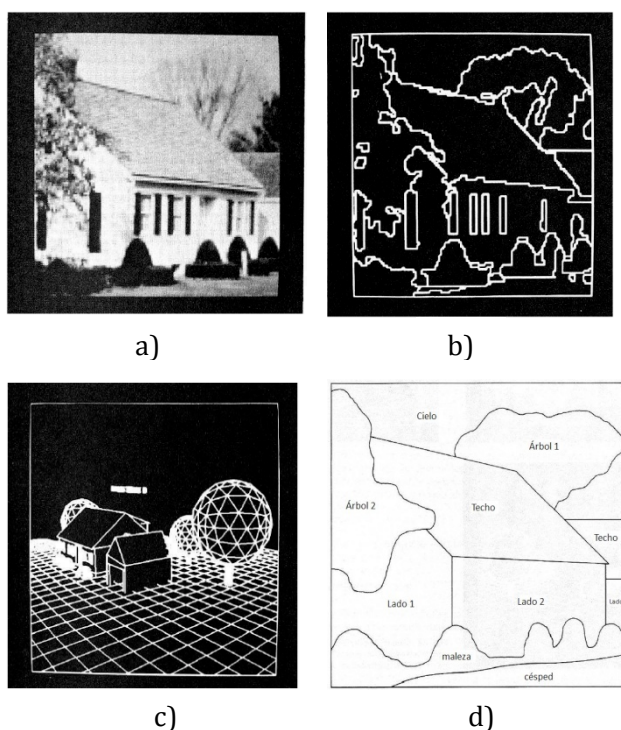


Fig. 5.2 Ejemplos de representación usando visión computacional:
a) icónica, b) segmentada, c) geométrica, d) relacional

Percibimos un mundo de objetos tridimensionales coherentes con muchas propiedades invariantes. Objetivamente, los datos visuales entrantes no muestran coherencia correspondiente o invariancia, que no contiene muchas variaciones irrelevantes o incluso engañosas. De alguna manera nuestro sistema visual, desde la retina a niveles cognitivos, entiende, o los impone a la información visual caótica. Lo hace mediante el uso de información fiable intrínseca que se puede extraer de la entrada, y también a través de las hipótesis y los conocimientos que se aplican a diversos niveles en el procesamiento visual [7].

Es importante la visión computacional, describe el mundo que vemos en una o más imágenes y para reconstruir sus propiedades, tales como: forma, superficies, colores, movimientos, iluminación, reflejos, texturas principalmente (Tabla 5.1) [8]. Es sorprendente que los seres humanos y animales lo hacen con tan poco esfuerzo, mientras que la computadora a través de algoritmos de visión, pueden realizar una tarea similar.

Tabla 5.1 Ejemplos de tareas en el análisis de imágenes.

Campo	Objetivos	Modalidad	Tareas	Fuentes de aprendizaje
Imágenes aéroespaciales	Terrenos. Edificios. Bosques. Zonas de Cultivo.	Luz. Infrarrojo. Radar.	Imágenes mejoradas. Análisis de los recursos. Predicción del tiempo. Espionaje. Guía de misiles. Análisis de tácticas.	Mapas. Formas de modelos geométricos. Modelos de la formación de la imagen.
Robótica	3D. Escenas libres. Partes mecánicas.	Luz. Rayos X	Identificar o describir. Objetos en escena. Tareas industriales	Modelos de objetos. Modelos de la reflexión de luz de los objetos.
Astronomía	Estrellas. Planetas	Luz	Composición química. Imágenes mejoradas	Figuras y modelos geométricos.
Macro Médica	Órganos del cuerpo	Rayos X Ultrasonido Isótopos Calor	Diagnóstico de anomalías. Operativos y la planificación del tratamiento.	Modelos atómicos. Modelos y formación de imágenes.

5.3 Elementos del sistema

En este trabajo, se propone un sistema de visión computacional, para el reconocimiento de imágenes aeroespaciales, se compone de los siguientes elementos:

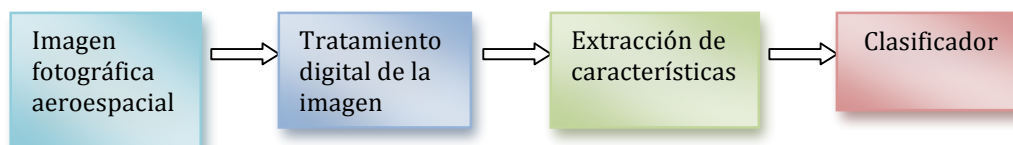


Fig. 5.3 Sistema de visión computacional

La primera fase, que es tener una base de imágenes digitales, en nuestro caso son imágenes aeroespaciales, tomadas de Internet.

La segunda fase consiste en el tratamiento digital de las imágenes, con objeto de facilitar las etapas posteriores. En esta etapa de procesamiento previo es donde, mediante filtros y transformaciones geométricas, se eliminan partes indeseables de la imagen o se realzan partes interesantes de la misma, como su resolución, contraste y brillo de la misma.

La siguiente fase se conoce como extracción de las características, consiste en aislar los elementos que interesan de una zona para comprenderla. Algunas características pueden ser el color, la textura, la forma y las relaciones espaciales, FFT, Wavelets, histogramas, entre otros.

La fase del clasificador se pretende distinguir las imágenes respecto a las clases que hay en ellas como son: zona de bosque, zona de cultivo y zona urbana.

5.4 Obtención de imágenes

Usamos la base de imágenes fotográficas tomadas en el 2010 a una distancia aproximada a 400 km de la tierra, por el astronauta Soichi Noguchi (Fig. 5.4) en la Estación Espacial Internacional [9][10].



Fig. 5.4 Astronauta de la NASA. Soichi Noguchi

La base de imágenes original consta de 100 fotografías aproximadamente. Sin embargo se eligieron 10 imágenes para entrenamiento y 10 para su reconocimiento, debido a su uniformidad en la información contenida.

El tratamiento digital de cada imagen consistió en algunos procedimientos: cambio de formato a *BMP*[], ya que originalmente estaban en formato *JPG*[], tamaño de imagen 500x500 píxeles.

5.4.1 Extracción de las características de la imagen

La extracción de características para cada imagen se hace con los cálculos de los histogramas de brillo, contraste y el histograma de micro contornos.

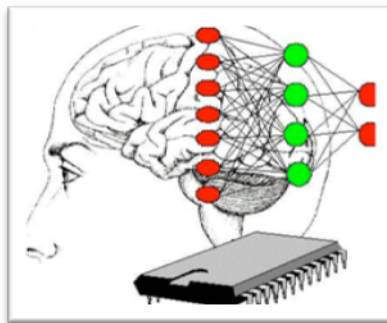
Los píxeles de la imagen tienen brillo de 0 a 255. Hemos dividido en 16 niveles de gris. Para cada intervalo el número de píxeles que tienen niveles de gris en este intervalo se calculó. De esta manera se obtuvo el histograma de brillo [11].

Para obtener el contraste se calcula la diferencia de que nos dé la diferencia de brillo entre dos píxeles vecinos, y se mueven de un píxel a calcular.

El algoritmo de cálculo de histograma de orientaciones micro contornos se basa en el algoritmo de Schwartz [12].

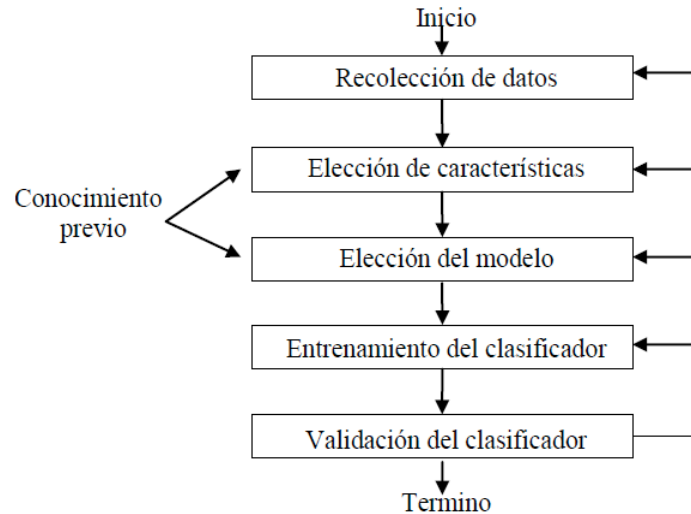
5.4.2 Red Neuronal Artificial

Los seres vivos necesitan categorizar el mundo que los rodea [Fig. 5.5], para poder reaccionar de forma eficaz a estímulos desconocidos. En el ser humano la categorización es aún más sofisticada y más abundante. Si se desean desarrollar sistemas capaces de identificar de "imitar" ciertos comportamientos de seres vivos, será inevitable incluir tareas de clasificación automática [13].



Para ello, los sistemas de visión computacional tienen una parte importante de reconocimiento en la actualidad, a través de los avances tecnológicos en el campo de la medicina, instrumentación, ingeniería aeroespacial, entre otros, el reconocimiento de imagen de la computadora se utiliza como una herramienta muy importante. Hay muchas técnicas de desarrollo para reconocimiento de imágenes [14].

La tarea del clasificador neuronal es con el uso de un vector de características, proporcionado por la extracción de propiedades, asigne al objeto a una categoría. Una tarea más general es determinar la probabilidad para cada categoría de todas las categorías posibles. El clasificador trabaja en dos modos: entrenamiento (reconocer el mayor número de patrones) y reconocimiento (tomar datos para procesar y realizar una acción en base a la categoría del patrón para solución de un problema en particular) [15,16] y gracias a estos procesos es capaz de realizar tareas tales como la automatización del proceso es el reconocimiento de patrones (ver Fig. 5.6).



El enfoque de este trabajo es desarrollar los algoritmos de clasificación. Especialmente nos interesa reconocer o clasificar diferentes tipos de texturas en las imágenes espaciales. En este caso podemos hablar sobre texturas de bosque, de zonas urbanas y de cultivo, etc.

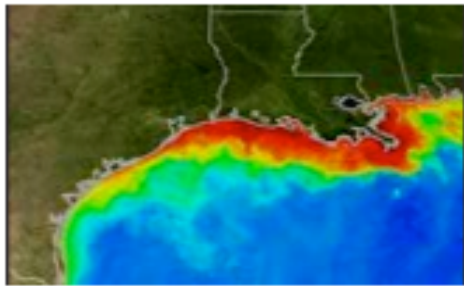
4.4 Aplicaciones de la visión computacional con el uso de imágenes aeroespaciales.

Actualmente, la visión computacional está siendo utilizada hoy en día en una gran variedad de la vida real aplicaciones:

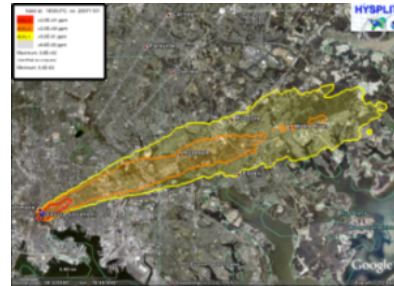
- ✓ EOS (*Earth Observation System*) de la NASA, EUA. Programa de monitoreo permanente de la superficie terrestre, sobre cambios en las coberturas terrestres. El Sistema de Observación de la Tierra (EOS) se compone de una serie coordinada de satélites de la inclinación de órbita polar y baja para las observaciones mundiales a largo plazo de la superficie terrestre, la biosfera, litosfera, la atmósfera y los océanos. EOS es un componente principal de la división de ciencias de la tierra de ciencia espacial de la NASA. EOS permite una mejor comprensión de la Tierra como un sistema integrado, Fig. 5.7, a) [17].
- ✓ HYSPLIT (*Hybrid Single-Particle Lagrangian Integrated Trajectory*) del Laboratorio de Recursos del Aire (ARL), Australia. Lleva a cabo programas de investigación para reconocer la calidad del aire, dispersión atmosférica, el clima, a través de la mejora de métodos para predecir la dispersión atmosférica de los materiales peligrosos, Fig. 5.7 b) [18].
- ✓ FIRMS (*Fire Information for Resource Management System*) financiado por la NASA, integra tecnologías de teleobservación para arrojar información a los administradores de recursos naturales de todo el mundo, Figs. 5.7, c) y d) [19].

Fig. 5.7 algunas aplicaciones de imágenes aéreoespaciales.

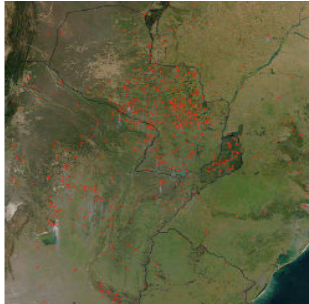
monitorea Tierra, b) reconoce calidad de aire, c) sistema detecta puntos calientes, d) daños de tsunami en Japón.



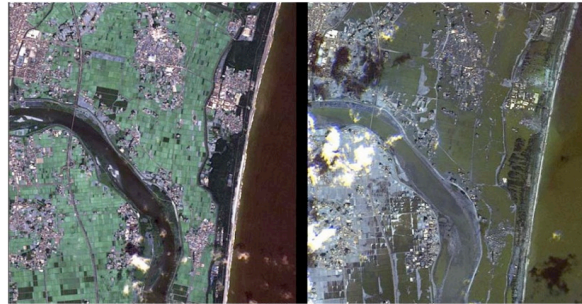
a)



b)



b)



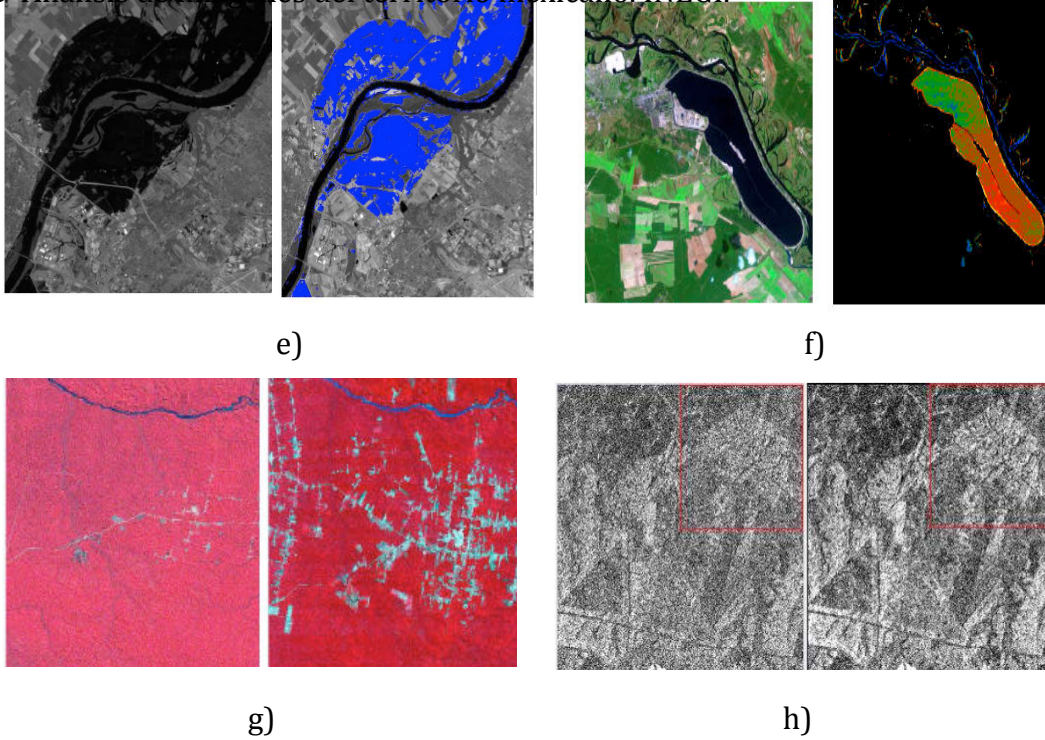
d)

- ✓ USGS (United States Geological Survey). A través de científicos asociados proporcionan información científica confiable para describir y comprender la tierra; con el reconocimiento, tratan de minimizar la pérdida de vidas y bienes en casos de desastres naturales, la gestión del agua (ver Fig. 5.8, e)), la energía biológica, y los recursos minerales, y mejorar y proteger nuestra calidad de vida [20].
- ✓ LANDSAT-5 captó el accidente de Chernovy tres días después del accidente, cubriendo el área de la planta nuclear. Se procesó la imagen y se pudo reconocer a través del lago de recirculación, el enfriamiento de los reactores, Fig. 5.8, f) [21].
- ✓ DMC, (*Disaster Monitoring Constellation*) integrante del International Charter, es una Asociación liderada por la Universidad de Surrey (Inglaterra). Consiste en una red de microsátélites, y sus estaciones terrestres para reconocer y clasificar las agresiones al planeta, cómo deforestación, Fig. 5.8, g)[22].
- ✓ ERS (Satélite en operación) de la Agencia espacial Europea en noviembre del 2000, interpretó la variación de las texturas de las imágenes capadas, para ver la forestación de una zona de Paysandú, Fig. 5.8, h) [21].

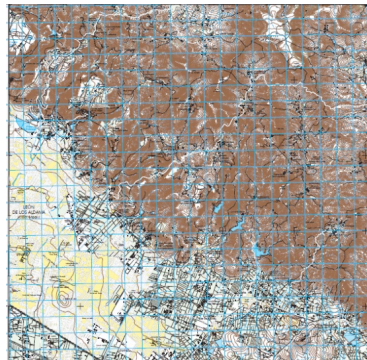
Fig. 5.8 Otras aplicaciones de imágenes aeroespaciales:

e) reconoce zonas inundadas U.S.A., f) cuantificó efecto de Chernovyl, g) nivel deforestación en selvas peruanas, h) clasificación por zona.

Fig. 5.9 Análisis de imágenes del territorio mexicano. INEGI.



En México, a través del Instituto Nacional de Estadística y Geografía (INEGI) realiza creación y procesamiento de imágenes por zonas de todo el territorio mexicano (Fig. 5.9), para diversas aplicaciones tales como: delimitar predios, identificar áreas de pendientes suaves o planas, abrir terrenos a la agricultura, así como susceptibles de reforestar, construir presas, conducir ductos, analizar y planear el crecimiento de ciudades y poblaciones rurales, conocer la distribución de las poblaciones urbanas, establecer rutas de evacuación, ubicación de áreas conflictivas y de plantíos de estupefacientes, ubicar áreas susceptible a inundaciones y derrumbes, trazar rutas de comercio, planear y ubicar rutas y destinos para excursiones, campismo, montañismo, alpinismo y ciclismo principalmente, así como el apoyo al medioambiente, por ejemplo diagnosticar áreas de impacto ambiental y áreas naturales protegidas para decretos de reserva de la biosfera, dar ordenamiento a recursos naturales. También el apoyo a la enseñanza, aplicando la ciencia en las disciplinas de geografía, topografía, turismo, agronomía, fotometría, ingeniería, computación, entre otras [23].



CAPÍTULO 6.

BASE DE IMÁGENES

Fig. 6.1 Fotos aéreas:

a) zona rural, b) zona residencial, c) zona industrial, d) zona de bosque

1

2

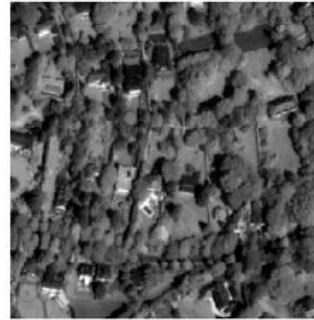
Fig. 6.2 Ejemplos de imágenes aeroespaciales

6.1 Base de imágenes

En el siglo XX, imágenes satelitales y aéreas de alta resolución fueron obtenidas (Fig. 6.1), para extraer información a base de métodos y técnicas de clasificación [1,2]. Una de las tareas interesantes de reconocimiento de imágenes aeroespaciales es la tarea de clasificación de superficies de tierra [3].



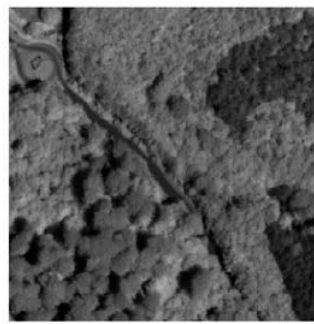
a)



b)



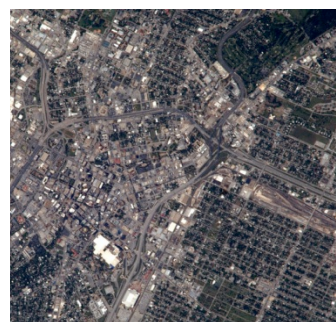
c)



d)

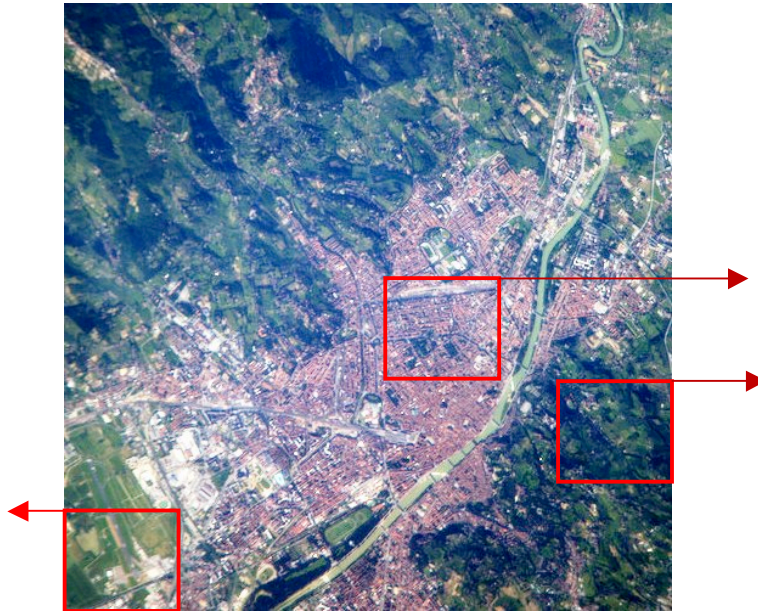
Nosotros queremos desarrollar un sistema de reconocimiento de texturas. Por eso elegimos las imágenes que poseían este tipo de información, la superficie de la Tierra. Se obtuvieron de internet [4] dos colecciones de 10 imágenes aeroespaciales, cada una, con diferentes paisajes. Una colección es para el proceso de entrenamiento y la otra colección para el reconocimiento de las texturas (Fig. 6.2).

Sin embargo, a estas fotografías digitales se tuvieron que procesar y preparar para que cumplieran con los requerimientos del sistema.



En la base de imágenes se puede observar que las cámaras pueden registrar procesos naturales y actuales que pueden ir desde la redistribución de la tierra a través de zonas altamente pobladas, así como zonas montañosas (Fig. 6.2, 1), con esto se puede hacer un censo de nuevos datos del asentamiento del hombre, identificar zonas de peligro y zonas adecuadas para la vivienda. En las fotos también se observan zonas claramente definidas como áreas verdes para cultivo y reserva (Fig. 6.2, 2).

Las zonas, se definen a través de la elección de clases en cada imagen aeroespacial. Se debe identificar y asignar la clase que más predomina en la zona: 1. Cultivo, 2. Bosque, 3. Urbana (Fig. 6.3).



6.2 Preparación de base de imágenes aeroespaciales.

El procesamiento de imágenes implica el tratamiento o modificación de una imagen que se adecue a los requerimientos del sistema [5]. El primer paso es obtener una imagen en un formato. En Internet y otras fuentes nos proporcionan colecciones de imágenes con un sinnúmero de formatos estándar, por ejemplo, .bmp, .jpg, .tiff, etc.

Una imagen digital se compone de una matriz bidimensional de números (Fig. 6.4), cada número es nivel de brillo b_{ij} de un determinado elemento de la imagen (pixel). Si es una imagen blanco y negro es una imagen binaria, ya que cada pixel es 0 o 1 que corresponde a nivel de gris 0 o 255. Las imágenes que ocuparemos en el sistema serán en su mayoría a color, estas poseen tres bandas que corresponden a los colores rojo, verde y azul, cada pixel tiene tres valores asociados con ella. La imagen tiene tamaño $n \times m$.

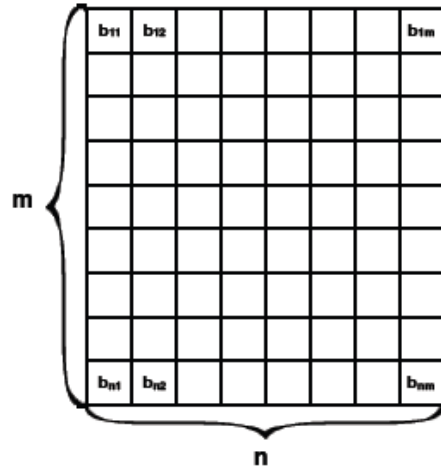
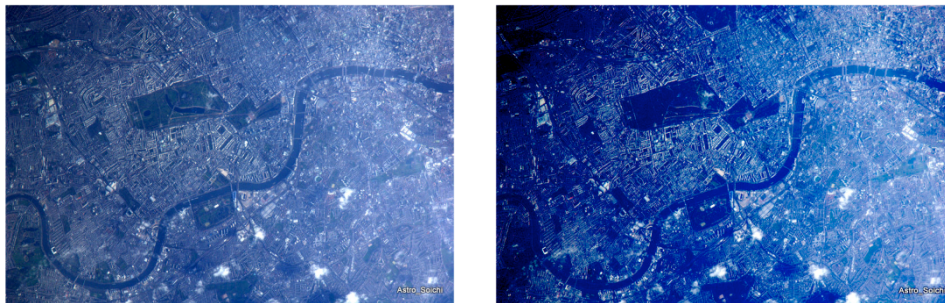


Fig. 6.4 Presentación de imagen digital

Vamos a usar imágenes con formato BMP, es un formato estándar de Microsoft® para imágenes y se almacena con su calidad máxima [6]. Se cambió el formato a través de visualizar la imagen en Microsoft Paint® un editor de dibujo gráfico [7] en sistema operativo Windows®.

Para mejorar la resolución de las imágenes, se modificó el brillo y contraste de cada imagen a través de la elaboración de un programa (ver apéndice X) en Matlab® versión R2010a (Fig. 6.5), esta técnica consiste en el ajuste numérico de los valores en intensidad de la escala de grises, es decir, varía el contraste de la imagen.



Posteriormente, a través del software Microsoft Picture Manager® incluido en Microsoft Office 2007®, se utilizó para estandarizar el tamaño de la base de imágenes a 500x500 píxeles, con un tamaño en memoria de 733 KB. Esto es porque la base de imágenes que se obtuvo por Internet es con diferentes tamaños, por lo que es muy importante el mismo tamaño en todas las imágenes que se usarán en el clasificador, por coherencia, por requerimientos del programa.

Para hacer esto sólo se tiene que configurar las opciones: izquierda, derecha, superior e inferior, de una ventana que será la que recorte la imagen (Fig. 6.6), posteriormente

Fig. 6.6 Estandarización de tamaño de una imagen

se guarda como una imagen nueva que será la que integre nuestra base de imágenes para el sistema.



6.3 Reconocimiento de texturas.

A lo largo de los años, se han desarrollado diferentes métodos para extraer información a partir de una imagen y reconocer de ella formas geométricas, texturas, entre otras a través de técnicas de clasificación multiespectral [1], métodos estadísticos [2] y clasificadores neuronales artificiales [6], etc..

Una de las tareas interesantes de reconocimiento de imágenes aeroespaciales es la tarea de clasificación de tipos de superficies en la tierra [3], a través del reconocimiento de texturas. Existen diferentes tipos de texturas [8]. Algunas propuestas en el reconocimiento de texturas, se pueden mencionar, la variabilidad interna de las imágenes reduce la separabilidad estadística de las clases. Por ejemplo, un árbol puede tener variabilidad, cuando está iluminado por la luz solar o en parte, en el lado oscuro (Fig. 6.7). El árbol tiene las respuestas espectrales muy diferentes, pertenecientes a la misma clase [9].



Los diferentes algoritmos de segmentación por texturas se desarrollan [2]. Algunos de ellos se basan en el análisis de regiones muy pequeñas [10]. El procedimiento se basa en el uso de tres criterios: el color, la suavidad y compacidad (Fig. 6.8) [10].

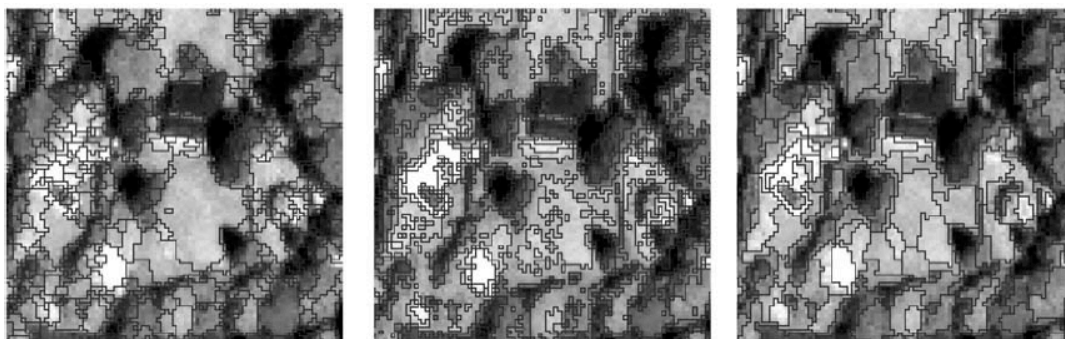


Fig. 6.8. Algoritmo de segmentación de una imagen satelital [6]

Otra técnica para el reconocimiento de imágenes satelitales es la extracción de características [11]. En esta técnica, se extraen varias características para describir el color, textura, forma y diseño, especialmente histograma de color (CH), vector de la coherencia del color (CCV), el coeficiente de la transformada discreta del coseno (DCT), histograma rango de dirección (EDH) y el vector rango de dirección coherente (EDCV).

En este trabajo se propone utilizar redes neuronales para el reconocimiento de las texturas en las imágenes aeroespaciales. El clasificador neuronal RSNC (*Random Subspace Classifier*), que tiene una estructura de múltiples capas [6] se puede utilizar en la tarea de clasificación de imágenes. Este clasificador ha demostrado buenos resultados en el reconocimiento de texturas metálicas [12]. El clasificador neuronal será descrito en el siguiente capítulo.

6.4 Marcado de imágenes

Para aplicar redes neuronales tenemos que preparar nuestras imágenes. El entrenamiento de la red neuronal será supervisado. Esto significa que la red neuronal tiene que “saber” a que clase pertenece la muestra de la imagen. Entrenamiento supervisado se caracteriza porque el proceso de entrenamiento es controlado por un agente externo (supervisor, maestro) que determina la respuesta que debería generar la respuesta de una entrada dada [13].

Para nuestra base de imágenes tenemos 3 clases: 1. Cultivo, 2. Bosque., 3. Urbana. Se requiere de un sistema capaz de seccionar y asignar, respecto a cada imagen la clase a la que corresponde. Por lo que se usó la técnica de marcado por el usuario. Esta consiste en un software programado en C++ Builder 6.0, este programa se divide en tres etapas.

La primera etapa, cargar imagen, al ejecutarse el programa y seleccionar la opción *OpenImage*, el usuario carga la imagen elegida (en formato .BMP y de 500x500 píxeles).

La segunda etapa, en la opción *Maker*, se divide la imagen en 25 zonas (100x100 pixeles cada una); conforme se va seccionando, el sistema va etiquetando el número de la zona a la que pertenece (Fig. 6.9).

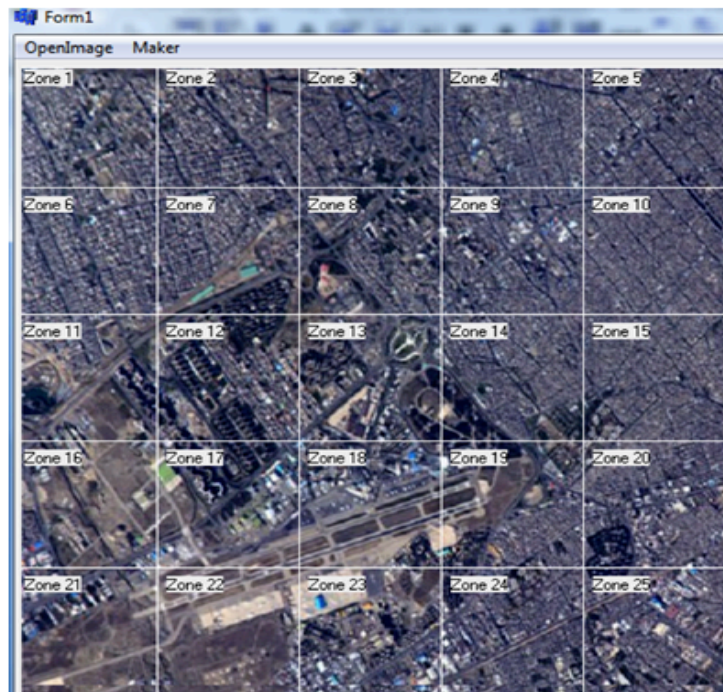


Fig. 6.9. Imagen seccionada en 25 zonas.

La tercera etapa, es la opción es donde el usuario asigna la clase que predomina en cada zona. Ya que se tiene previamente marcadas las 25 zonas de la imagen, ahora se introduce una clase por cada zona.

La asignación de clase dependerá del criterio de lo que más predomina en la zona, por ejemplo, puede ser que en una zona estén presentes una o las tres clases, sin embargo, debe de ser asignada sólo una clase.

Ya que se introdujo numéricamente la clase por cada zona, ahora se guardará con el botón *Save* (Fig. 6.10), estos datos serán un vector de 25 clases que corresponden a las 25 zonas, los cuales serán guardados en un archivo con formato .DAT, el cual será usado para la red neuronal artificial.

El proceso anterior, el marcado de las imágenes se realiza para cada imagen de la colección de imágenes aeroespaciales.

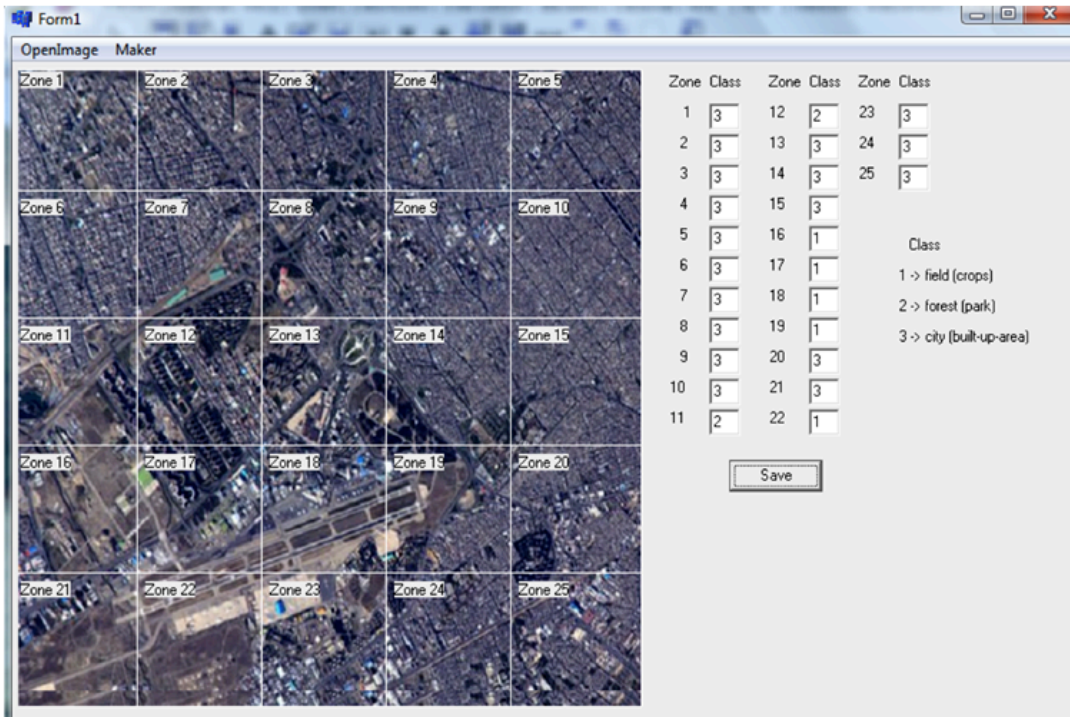


Fig. 6.10. Asignación de zona-clase en la imagen

Esta base de imágenes vamos a usar en el clasificador neuronal. En el capítulo 7 vamos a escribir la estructura y los algoritmos del clasificador neuronal RSNK.

CAPÍTULO 7.

CLASIFICADOR NEURONAL RSNC

7. Redes Neuronales Artificiales.

El cerebro humano es el sistema de cálculo más complejo que conoce el hombre. El hombre así como la computadora realizan bien diferentes clases de tareas; por ejemplo, para el reconocimiento del rostro de una persona resulta una tarea relativamente sencilla para el hombre es una tarea completa para la computadora, mientras que la contabilidad de una empresa es tarea costosa para un experto contable pero una sencilla rutina para una computadora básica.

La capacidad del cerebro humano de pensar, recordar y resolver problemas ha inspirado a muchos científicos que han intentado y procurado modelar a computadora el funcionamiento del cerebro humano [1].

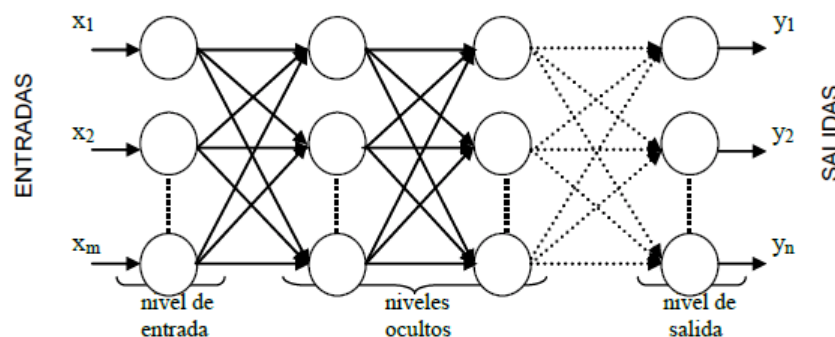
Un grupo de investigadores ha perseguido la creación de un modelo computacional que iguale o adopte las distintas funciones básicas del cerebro. El resultado ha sido una nueva tecnología llamada Computación Neuronal o también Redes Neuronales Artificiales.

Con el desarrollo de las computadoras, por la década de los 40, se plantean las primeras teorías sobre modelos de redes neuronales consiguiendo en la actualidad un lugar relevante en la ciencia (apéndice I).

Las redes neuronales artificiales *RNA*, son los paradigmas de aprendizaje y procesamiento automático inspirados en la forma en que funciona el sistema nervioso biológico [2]. Se conforman por la interconexión de elementos simples de proceso (neuronas), que operan de forma paralela para dar una solución alterna a problemas complejos relacionados con el reconocimiento de formas o texturas, predicción, codificación, control y optimización.

La distribución de neuronas dentro de la red se realiza formando niveles o capas de un número determinado de neuronas cada una (Fig. 7.1). Tres tipos de niveles principalmente:

- ✓ *Entrada*, es la capa que recibe directamente la información o patrones representados por vectores provenientes de fuentes externas a la red.
- ✓ *Ocultas*, también llamadas intermedias, cuyas unidades responden a rasgos particulares que pueden aparecer en los patrones de entrada. Puede haber uno o varios niveles ocultos.
- ✓ *Salida*, son las salidas de información de toda la red al exterior.



Cada interconexión del proceso entre capas actúa como una ruta de comunicación: a través de estas interconexiones viajan valores numéricos de una capa a otra. Estos valores son evaluados por los pesos de las conexiones. Los pesos de las conexiones se ajustan durante la fase de aprendizaje para producir una RNA final [3].

La estructura básica de interconexión entre capas, es la de una red multicapa, anteriormente descrita, se trata de una estructura típica de implementación del paradigma conocido como *Backpropagation* (apéndice I).

Durante las últimas cinco décadas, las redes neuronales artificiales han pasado de ser sólo un concepto a ofrecer una solución a problemas comerciales e industriales [4]. Los profesionales de diferentes campos como la ingeniería, ciencias, fisiología, psicología entre otras, han unido sus esfuerzos debido al potencial que ofrece esta tecnología, con el objetivo de obtener máquinas altamente eficaces con algoritmos de aprendizaje [5].

Para nuestra tarea elegimos clasificador neuronal con espacio aleatorio *RSNC* (*Random Subspace Neural Classifier*) que está basado en *RTC* clasificador de umbrales aleatorios (en inglés *Random Threshold Classifier*) [6]. El cual es un sistema propuesto para resolver problemas relacionados con el reconocimiento de imágenes ya que posee un alto rendimiento en las etapas de entrenamiento y en la de procesamiento.

7.1 Estructura del Clasificador Neuronal RSNC.

El clasificador neuronal RSNC está basado en el clasificador RTC, que tiene una estructura de múltiples capas [7] y fue desarrollado y probado para la tarea de clasificación de imágenes en 1994 [6]. La formación de este clasificador es entrenamiento *supervisado* (apéndice I). Este clasificador mostró buenos resultados en el reconocimiento de texturas metálicas [8], de imágenes biomédicas [11].

Se presenta (Fig. 6.2) la estructura de RTC clasificador neuronal [9], la cual consiste en una capa de entrada, capa de codificación y capa de entrenamiento. Está compuesta la capa de neuronas binarias, con conexiones no modificables que permita transformaciones no lineales de parámetros de entrada a parámetros binarios de altas dimensiones.

7.1.1 Estructura

La estructura de la capa de codificación está compuesta por bloques similares S , y cada bloque tiene una salida neuronal (b_1, \dots, b_s) ; mientras que las entradas al sistema son un conjunto de características (X_1, X_2, \dots, X_n) , las cuales describen a la imagen. Cada característica X es la entrada por a dos neuronas con umbrales l_{ij} y h_{ij} , donde i ($i=1, \dots, n$) representa el número de característica, y j ($j=1, \dots, s$) representa el número de bloque neuronal. Los umbrales l y h son elegidos aleatoriamente y siempre debe cumplir con la importante condición de $l_{ij} < h_{ij}$.

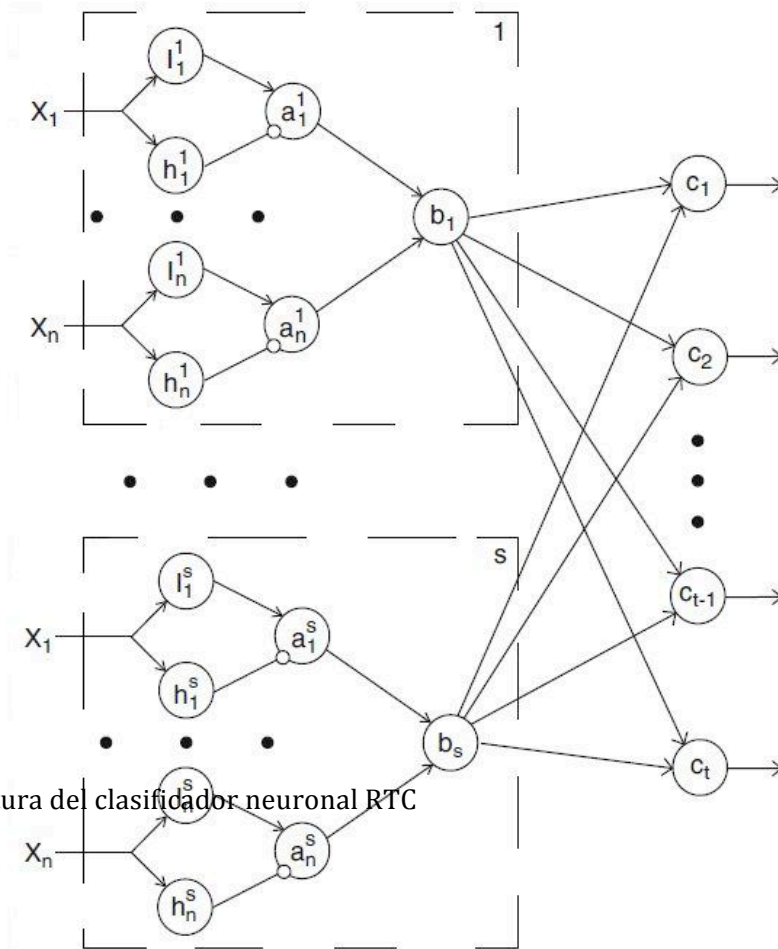


Fig. 7.2 Estructura del clasificador neuronal RTC

La salida de la neurona l_{ij} es conectada a una entrada excitada de la siguiente neurona a_{ij} , y la salida de la neurona h_{ij} se conecta a una entrada inhibida de la neurona a_{ij} .

En la salida de una neurona a_{ij} solo ocurre cuando existe una entrada excitada con valor 1 y cuando la entrada es inhibida es igual a 0. Todas las salidas de neuronas a en un bloque j , se conectan a las entradas excitadas de una neurona b_j , que representa la salida de todo el bloque neuronal. La salida de la neurona b_j es 1 solo si todas las neuronas del bloque a_{ij} están excitadas. Esta neurona simula el elemento lógico "Y".

La salida de cada bloque de neuronas se conecta con todas las salidas neuronales del clasificador (c_1, \dots, c_t) mediante conexiones entrenables, siendo t el número de clases.

El clasificador trabaja con dos modos: entrenamiento y reconocimiento.

En la etapa de entrenamiento, las neuronas c_t en la capa de respuesta tiene asociado un peso w_i el cual se modifica por las neuronas b_j . Cada neurona en esta capa representa una respuesta de clasificación de sistema, y se elige aquella que tenga el valor de excitación más alto. También en el modo de entrenamiento, el objetivo de la red es reconocer correctamente el mayor número de patrones. Por eso en esta etapa, se trata de disminuir todos los pesos de las conexiones de la clase incorrecta y aumentar los de la clase correcta, lo cual conlleva a un entrenamiento *supervisado*.

Fig. 7.4 Identificación geométrica del clasificador.

anteriormente mencionado. Se usa la regla de Hebbian (apéndice I) para las conexiones de entrenamiento.

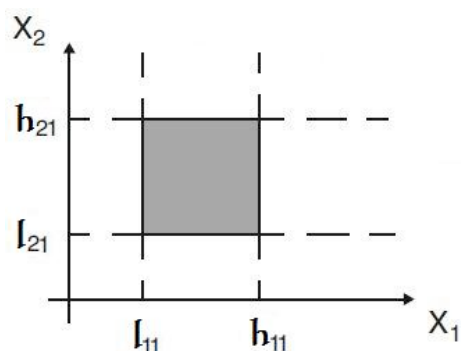
El entrenamiento termina cuando se cumpla cualquiera de las dos condiciones:

- 1ª El número de errores es menor al prefijado.
- 2ª Llegar a la última iteración de un número fijo de iteraciones de entrenamiento.

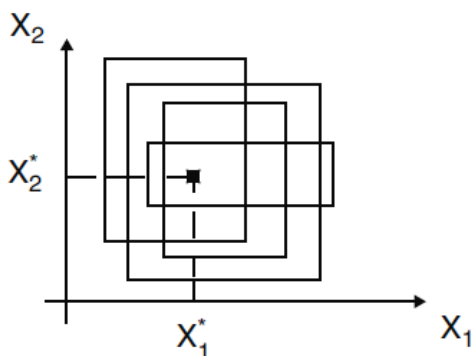
Fig. 6.3 Identificación geométrica de la neurona.

7.1.2 Interpretación geométrica.

La interpretación geométrica del clasificador será un auxiliar para comprender los principios anteriores. Si se consideran dos características X_1 y X_2 (Fig. 6.3).

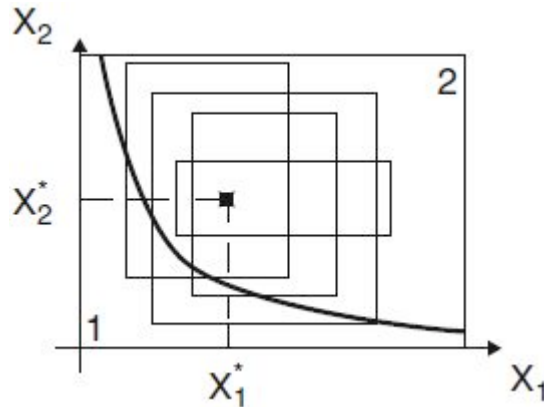


Entonces, la neurona b_1 está excitada en todos los casos cuando el punto representa un vector de características se localiza dentro del cuadrado sombreado (Fig. 7.3). Como el clasificador RTC posee suficientes bloques neuronales similares ($1, \dots, s$), todas las características espaciales pueden aparecer en suficientes rectángulos localizados en planos y tamaños aleatorios (Fig. 6.4).



En el espacio multidimensional, en vez de rectángulos podemos obtener paralelepípedos multidimensionales. Cada rectángulo corresponde al estado activo de

una neurona b_1, \dots, b_s . Por ejemplo, si se analiza un nuevo punto (X_1^*, X_2^*) en el espacio y dos características (X_1 y X_2) conteniendo dos clases (1 y 2) (Fig. 7.5), se obtendrá la respuesta a la neurona donde se cubra un gran número de rectángulos respecto al nuevo punto.



Si se tiene que reconocer estas dos clases (1 y 2) y si permitimos que el punto nuevo (X_1^*, X_2^*) pertenezca a la clase 2. Si el punto se encuentra lejos de la frontera de clases, éste es cubierto con rectángulos cuyas áreas están dentro del límite de la clase 2, esto significa que durante el proceso de entrenamiento los pesos de las conexiones entre las neuronas correspondientes a b_j y las salidas neuronales de la clase 2 serán mayores que los pesos de las conexiones de neuronas b_j con neuronas de la clase 1. Por consiguiente, el punto se reconocerá como perteneciente a la clase 2.

7.1.3 Diferencia entre clasificadores neuronales RSNC y RTC

El clasificador neuronal RSNC está basado en a partir del clasificador neuronal RTC. La diferencia principal es que en el clasificador RSNC no todos los parámetros de entrada (X_1, \dots, X_n) participan en procesos de entrenamiento y reconocimiento de imágenes. Es decir, eligen un conjunto aleatorio de todas las entradas. Este procedimiento permite disminuir el tamaño del espacio paramétrico y acelerar los procesos de tratamiento de información.

Por lo anterior, se tiene que el tiempo de entrenamiento, el tiempo de reconocimiento y conjunto de pruebas tasa de error, por lo general RSNC supera algunos algoritmos conocidos tales como el método de vecino más cercano, método de la función potencial, etc. [10]. Además el clasificador se vuelve más competitivo cuando el número de parámetros de entrada y el tamaño del conjunto de entrenamiento incrementan.

También el clasificador RSNC permite una transformación efectiva de los datos reales multidimensionales en la representación binaria para el uso, por ejemplo de las redes neuronales.

7.1.4 Extracción de características

Entre las características de una imagen están el color, la textura, la forma y las relaciones espaciales, FFT, Wavelets, histogramas [7].

En conjunto con el color, la textura es una poderosa característica discriminadora, está presente en casi cualquier parte de la naturaleza. La textura puede ser descrita de acuerdo a su espacio, frecuencia o propiedades perceptuales (periodicidad, dirección y grado de complejidad, etc.). Ciertas características espaciales basadas en estos atributos son particularmente útiles para reconocer imágenes por similitud de textura.

Una idea más clara de lo que es la textura en una imagen puede ser basada en sus propiedades estadísticas y estructurales de patrones de brillo. Existen muchas técnicas para extraer descriptores de texturas significantes de las imágenes basadas en diferentes modelos [12], por lo que las texturas usualmente proporcionan información sobre determinada sección de la imagen [13, 14].

Es necesario extraer de la imagen las características en las zonas de la imagen y para ello elegimos los histogramas de brillo, de contraste y de las orientaciones de microcontornos. Se utilizan como las características de entrada para el clasificador RSNC.

7.1.4.1 Histograma de brillo

El histograma de una imagen digital con niveles de gris en el rango $[0, L - 1]$ es una función discreta $h(r_k)=n_k$ donde r_k es el k -ésimo nivel de gris y n_k es el número de píxeles en la imagen que tiene nivel de gris (r_k). Es una práctica común para normalizar un histograma, dividiendo cada uno de sus valores por el número total de píxeles de la imagen, que se denota por n . En términos generales da una estimación de la probabilidad de ocurrencia de niveles de gris r_k .

Los histogramas son la base de numerosas técnicas de procesamiento en el campo aeroespacial (Fig. 7.6) [15]. La manipulación del histograma se puede utilizar con eficacia para la mejora de la imagen, además de proporcionar estadísticas útiles de la imagen, compresión de imágenes y segmentación [12].

En nuestro caso, obtendremos los histogramas mediante zonas limitadas a una ventana rectangular con dimensiones $b \times h$ (píxeles). Y de esa ventana se extraen las características de la textura.

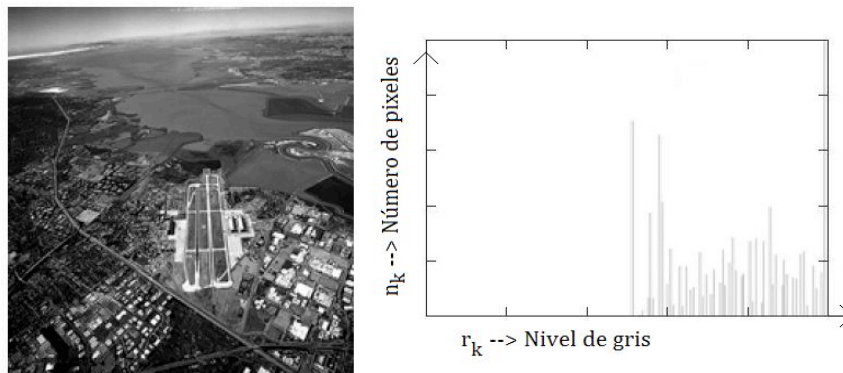


Fig. 7.7 Histograma de brillo por zona de la imagen

El histograma de brillo es una representación gráfica de una variable en forma de barras describe el nivel de luminosidad de una imagen [5].

La superficie de cada una de las barras mostradas es proporcional al nivel de brillo respecto al número de píxeles que contienen ese nivel, en la imagen.

Para generar el histograma de brillo, dividimos la escala total de brillo entre k y obtenemos intervalos. Se calcula el número de píxeles dentro de cada intervalo creado y este número se considera como el parámetro que describe a la textura.

En nuestro caso, el histograma de brillo, es un modo de descripción de la imagen a través de la codificación de los 16 niveles que contiene.

Ya que cada imagen de nuestro sistema está dividido en 25 zonas, fue necesario crear un programa en C++ Builder 6.0® para obtener el histograma de brillo de cada zona de la imagen aeroespacial (Fig. 7.7). Por lo tanto se obtienen 25 histogramas por cada imagen.



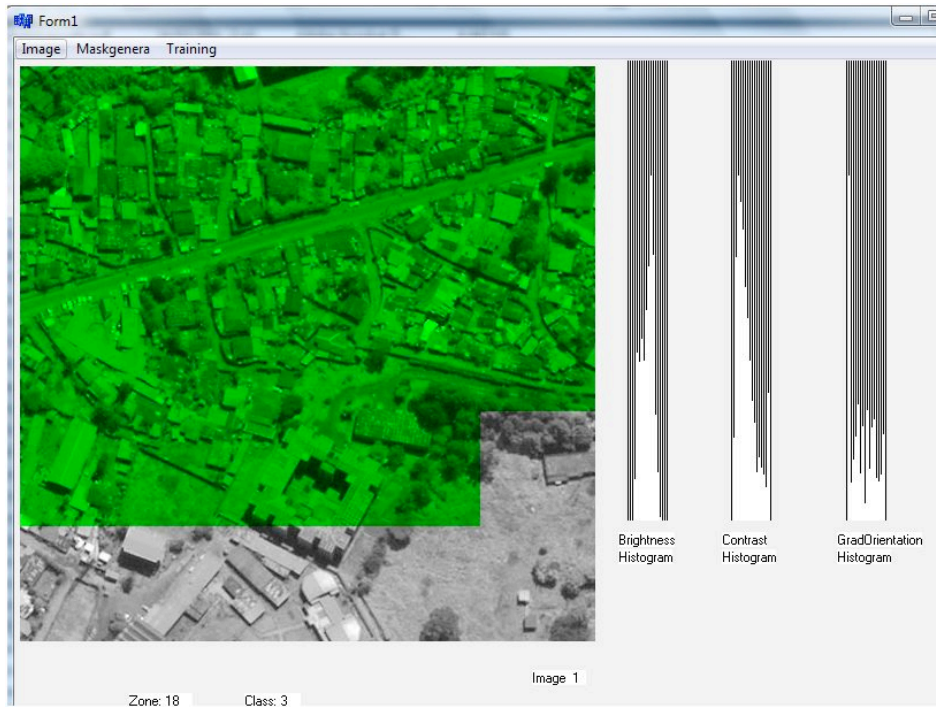
7.1.4.2 Histograma de contraste

Para el histograma de contraste es la diferencia de brillo entre dos píxeles vecinos [18]. Los píxeles de la imagen tienen brillo de 0 a 255. Hemos dividido en 16 niveles de gris. Para cada intervalo se calcula la diferencia de brillo entre dos píxeles vecinos, y se mueven de un píxel a calcular.

En nuestro caso, el histograma de contraste, es un modo de descripción de la imagen a través de la codificación de los 16 niveles que contiene.

Cada imagen de nuestro sistema está dividido en 25 zonas, fue necesario crear un programa en C++ Builder 6.0® para obtener el histograma de contraste de cada zona de la imagen aeroespacial (Fig. 6.8). Por lo tanto se obtienen 25 histogramas por cada imagen.

Fig. 7.8 Calculo de histogramas por ventana



7.1.4.3 Histograma de orientación de micro contornos.

Es el cálculo de orientaciones de microcontornos de una imagen aeroespacial (Fig. 7.8) está basado en el algoritmo de una patente de Schwartz [4].

1. Se calculan las sumas de brillo de pixeles diagonales.

$$Y_1 = X_1 + X_4$$

$$Y_2 = X_2 + X_3$$

2. Introducimos una constante experimental. Si cumple que $|Y_1 - Y_2| > C_1$, no existe contornos por lo que hay que analizar cuatro nuevos pixeles. De lo contrario, es decir, si existe contorno (la diferencia es pequeña) definimos su orientación.

3. Se asigna un valor a la diferencia de estos pixeles,

$$\delta_1 = X_1 - X_4 \text{ y } \delta_2 = X_3 - X_2$$

4. De estos valores absolutos se elige un máximo y un mínimo,

$$\Delta_1 = \max(|\delta_1|, |\delta_2|)$$

$$\Delta_2 = \min(|\delta_1|, |\delta_2|)$$

5. Y calculamos valor de nuestro ángulo de

$$Y = \frac{\Delta_2}{\Delta_1} < 1$$

microcontorno Y

De acuerdo a la correspondiente con la tabla 7. 1:

Tabla 7.1

Y	$0 - \frac{\pi}{4}$	$\frac{\pi}{4} - \frac{\pi}{2}$	$\frac{\pi}{2} - \frac{3}{4}\pi$	$\frac{3}{4}\pi - \pi$	$\pi - \frac{5}{4}\pi$	$\frac{5}{4}\pi - \frac{3}{2}\pi$	$\frac{3}{2}\pi - \frac{7}{4}\pi$	$\frac{7}{4}\pi - 2\pi$
Y	↘	↗	↘	↗	↘	↗	↘	↗
δ_1	+	+	+	-	-	-	-	+
δ_2	-	+	+	+	+	-	-	-
$ \delta_1 - \delta_2 $	+	+	-	-	+	+	-	-

El sentido de las flechas hacia arriba indican que se debe sumar el valor de y , y la flecha hacia abajo indica que se debe restar el valor de y ; en ambos es al intervalo. Ya que se tiene el valor del ángulo del contorno, se calcula todos los contornos en la imagen así como el histograma de orientación de microcontornos para toda la imagen.

7.2 Entrenamiento de la red.

En el proceso de entrenamiento será supervisado para poder realizar la comparación con la salida obtenida mediante el clasificador y la salida correcta, Si ambas salidas corresponden al mismo valor se interpreta que no se realiza ningún cambio en el sistema, si no corresponden se modifican los valores en las conexiones de los pesos entre capa B y capa C del sistema. Para ello, como primer paso se debe elegir el número de imágenes para este proceso.

Para nuestro sistema, el proceso inicia tomando la primera imagen la cual está compuesta por 25 ventanas con su respectiva clase que se lee de un archivo previamente guardado, para analizar a que clase corresponde cada ventana de la imagen, es decir, a qué neurona de excitación corresponde. Por ejemplo, si ventana 1 pertenece a clase 1 y corresponde la respuesta 1, entonces no se cambia peso W , se procede con la siguiente ventana, si la zona 2 pertenece a la clase 3 y responde a 2, se cambia peso W disminuye clase incorrecta y aumenta clase correcta.

Existen algunos criterios para el entrenamiento.

- 1 Definir los ciclos de entrenamiento, es decir, proponer el número de veces que será introducido todo el conjunto de imágenes de entrenamiento. Cuando se tiene el número de ciclos total de entrenamiento el sistema se detiene.
- 2 Obtener número de errores de reconocimiento se aproxime a 0.

7.3 Reconocimiento de la red.

En el proceso de reconocimiento, se realiza la validación del clasificador neuronal con las imágenes que no participaron en el entrenamiento, para obtener el porcentaje de

reconocimiento de texturas. El entrenamiento consiste en la presentación de un conjunto suficientemente amplio de datos de entrenamiento formado por las entradas y los valores correspondientes de las variables que se obtuvieron en el entrenamiento, hasta conseguir que los pesos internos conduzcan a resultados óptimos en la capa de salida c , acercándose lo más posible a los resultados esperados, obteniendo la clase verdadera con el error del proceso de entrenamiento.

CAPÍTULO 8.

RESULTADOS Y CONCLUSIONES

Fig. 8.1 b Tratamiento de imágenes

8.1 Resultados y Conclusiones.

8.1.1 Base de imágenes

Se obtuvo el compendio de la base de imágenes aeroespaciales. Se realizó el procesamiento (Fig. 8.1 a) de las imágenes para mejorarlas (Fig. 8.1 b), a través de programación en Matlab y se modificó el tamaño de la imagen, para cumplir con las los requerimientos en el proceso de reconocimiento de texturas.

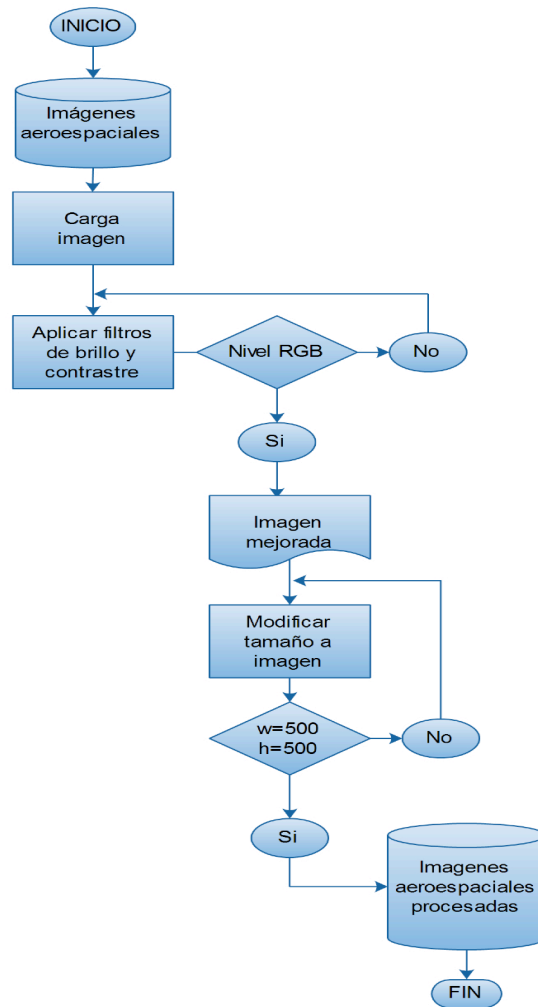
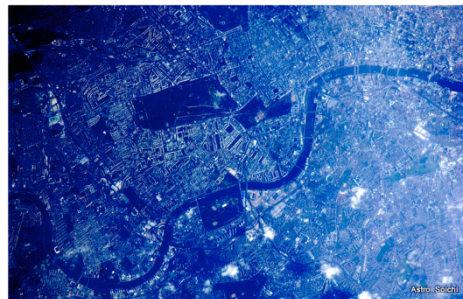


Fig. 8.1 a Diagrama de flujo del procesamiento de imágenes.



8.1.2 Programa *Marcado de imágenes*

Como parte de los resultados, se programó en C++ Builder 6.0, el marcado de las imágenes, con el propósito de asignar manualmente la clase que pertenece a cada ventana de una imagen.

El proceso de *Marcado de imágenes* (Fig. 8.2), radica en generar un archivo que contenga un vector de 25 elementos representando la relación clase/ventana por cada imagen de la base.

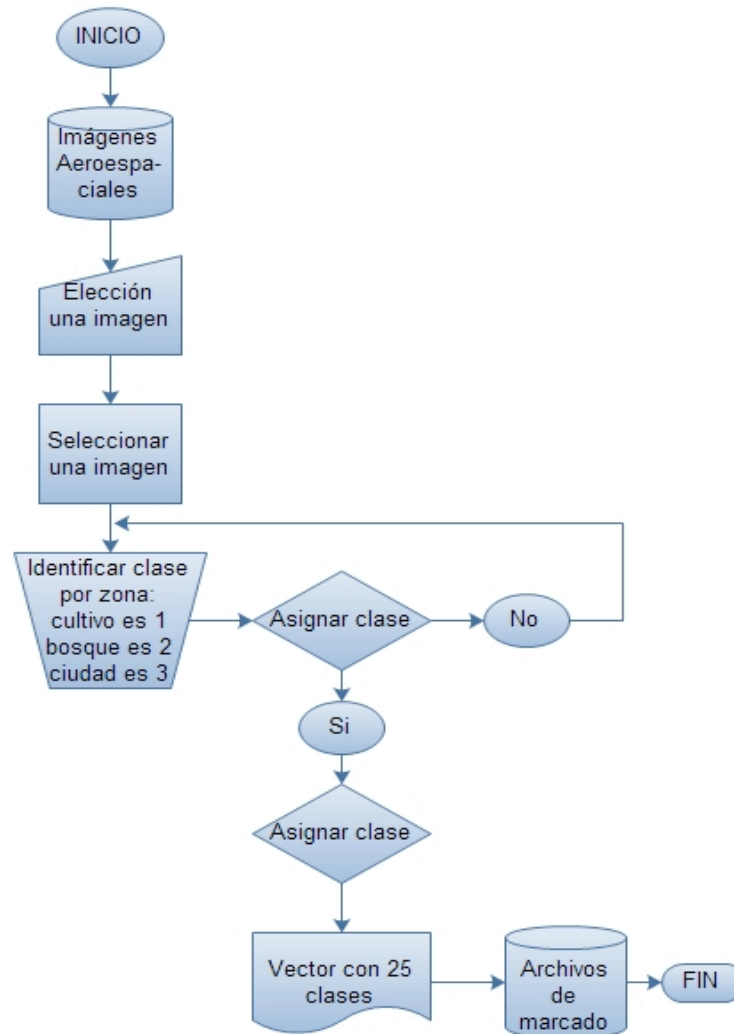


Fig 8.2. Diagrama de flujo del proceso de marcado

El programa *marcado* está compuesto por un menú de herramientas (Fig. 8.3). La opción del menú *OpenImage*, donde se localiza y se carga la imagen que se desea marcar.

En la opción *Maker*, el software secciona la imagen en 25 ventanas o zonas, que posteriormente se asignará la clase a la que pertenece cada ventana: 1 cultivo, 2

bosque y 3 urbana. Estos 25 datos se guardarán en un archivo con el botón *Save*. Para después ser usados en el proceso del reconocimiento de texturas.

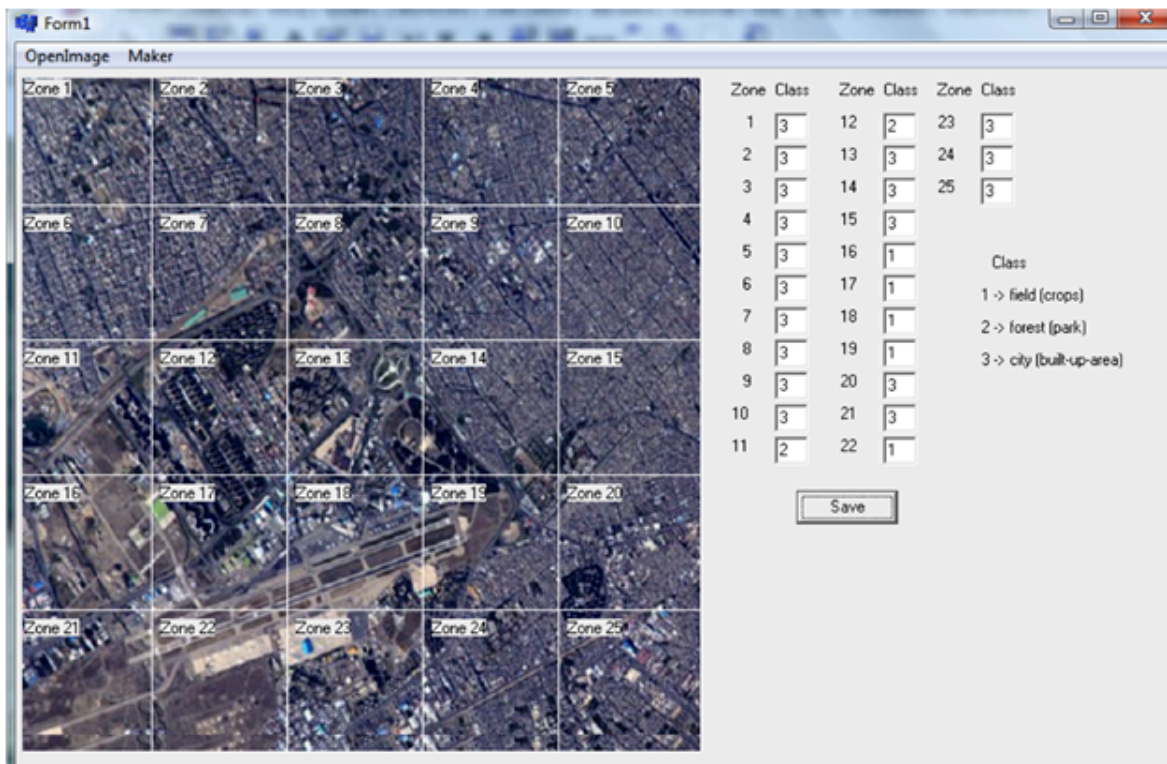


Fig 8.3. Marcado de imagen

8.1.2 Programa Recognition of aerospace images

El programa *Recognition of aerospace images* es el proceso de toda la estructura del clasificador RSC, para el reconocimiento de texturas (Fig. 8.4).

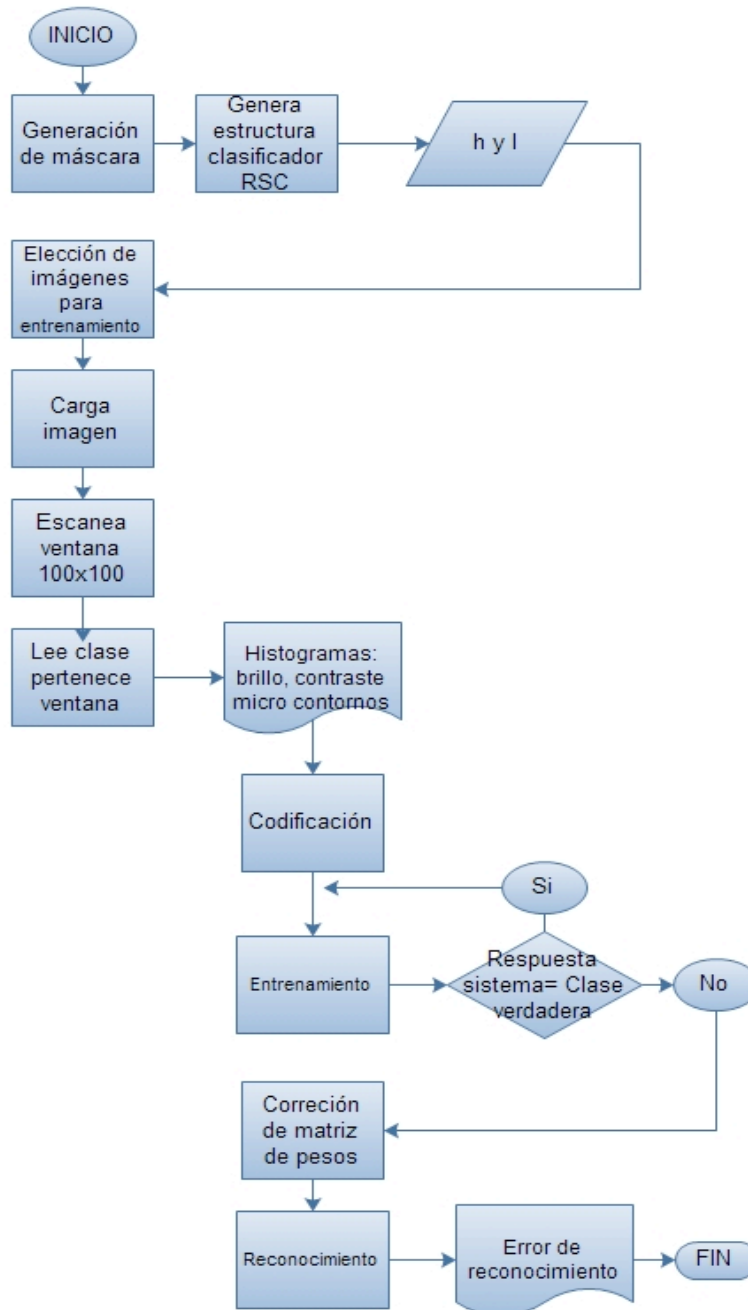


Fig. 8.3. Marcado de imagen

Fig. 8.3 Programa RSNC

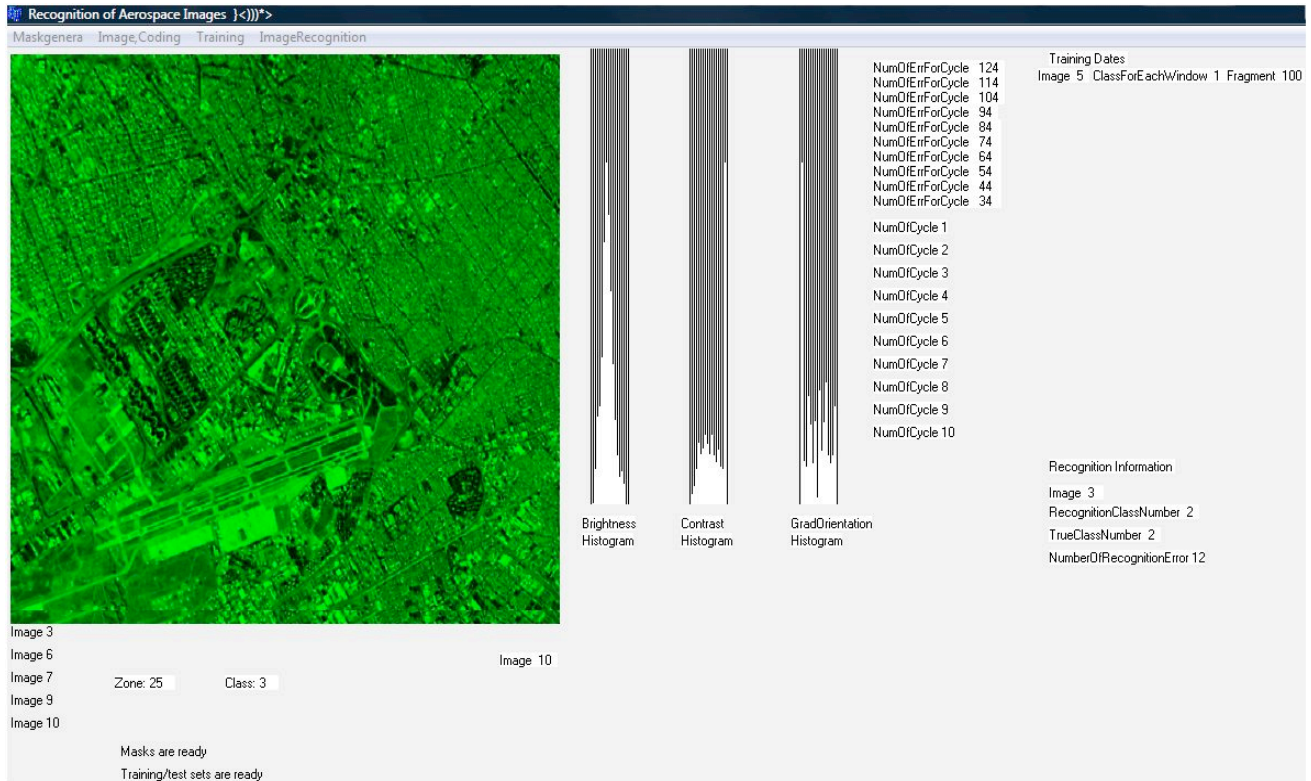
Se hizo un programa en C++ Builder 6.0 para la tarea del reconocimiento de texturas. Cada zona o ventana pertenece a una clase ya sea 1->cultivo, 2->bosque o 3->ciudad. El programa se compone de un menú, el cual tienen 4 opciones para completar el proceso, a continuación se describe cada una de ellas (Fig. 8.4):

MaskGeneration; Esta será, la generación de mascarar, es decir la creación de la estructura del clasificador neuronal RSNC. En esta sección, también se elige aleatoriamente de la base de imágenes, el número que se haya determinado que se debe utilizar en el proceso de entrenamiento de la red.

Image,Coding; se abre cada imagen de nuestra base de 10 imágenes. Se escanea por ventana y se calcula para cada una los histogramas de brillo, de contraste y de contornos así como se lee a que clase pertenece esa ventana, todo lo anterior para las 25 ventanas de cada imagen, codificando la información anterior.

Training; Se entrena la red para el número de imágenes (5 en nuestro caso) e interviene el número de ciclos para calcular el error por cada ciclo (Fig. 8.3).

ImageRecognition; Se toman las imágenes restantes y se obtiene el error del entrenamiento.



Experimentos

El sistema tiene como objetivo reconocer a qué clase pertenece la textura de la ventana. Cuando la textura reconocida o de respuesta no coincide con la clase correcta, se genera un error de reconocimiento. El porcentaje de error indica el número de ventanas en las cuales el sistema falla al reconocer la textura.

Se elaboraron una serie de experimentos para mostrar la tarea de reconocimiento que el sistema es capaz de realizar. En esta tarea se trabajó en la parte de entrenamiento con un número fijo de ciclos; es decir, se sabrá cuantas veces se introducirá el conjunto de imágenes de entrenamiento para después detener el proceso. Mientras que para el reconocimiento se elaboró con imágenes que no participaron en el proceso de entrenamiento, obteniendo el porcentaje de error y de reconocimiento.

En los experimentos para una base de 10 imágenes, intervinieron constantes y variables del sistema:

<i>const NumOfReclm=5;</i>	Número de imágenes para el reconocimiento
<i>const ImHeight=500;</i>	Altura de la imagen
<i>const ImWidth=500;</i>	Ancho de imagen
<i>const int NumOfClasses=3;</i>	Numero de clases 1->cultivo 2->bosque 3->urbana
<i>const GF=3;</i>	Grupo de características, histogramas brillo, contraste y microcontornos.
<i>const SG=16;</i>	Tamaño del grupo de características
<i>const int h=100;</i>	Alto de ventana
<i>const int w=100;</i>	Ancho de ventana
<i>const int NI=10;</i>	Número de imágenes de la base
<i>const NumOfReclm=5;</i>	Número de imágenes para el reconocimiento
<i>const int nw=25;</i>	Número de ventanas de la imagen.
<i>int const SizeOfSubspace=7;</i>	Tamaño de subespacio. Número de entradas aplicadas para una neurona
<i>int NumOfTrainCycles=10;</i>	Número de ciclos de entrenamiento.
<i>const NumOfNeurons=10000;</i>	Número de neuronas del clasificador entre capa A y B

Se realizó el entrenamiento del clasificador RSNC con 10 ciclos de entrenamiento de una secuencia de 5 imágenes.

Y se calcula el número de ventanas para toda la base de imágenes, con la siguiente fórmula:

$$NTV = \left(\frac{W}{w}\right) \times \left(\frac{H}{h}\right) \times (NIB)$$

donde W ancho de imagen, H alto de imagen, w ancho de ventana, h alto de ventana, NIB número de imágenes en la base.

$$NTV = \left(\frac{500}{100}\right) \times \left(\frac{500}{100}\right) \times (10) = 250$$

Se calcula el porcentaje de error, el cual nos indica número de ventanas reconocidas con error, es decir, el porcentaje que el sistema no reconoció, con la siguiente fórmula:

$$X = \frac{100 \times N_{error}}{NTV} \%$$

donde X porcentaje de error N_{error} el número de ventanas reconocidas con error para 5 y 6 imágenes y tomando en cuenta los parámetros anteriormente mencionados. En la tabla 8.1 se presentan los resultados obtenidos.

Tabla 8.1 Resultados de entrenamiento para 2 experimentos.

Número de ciclos	N_{error}	%Error	Experimento
1	124	49,6	1
2	114	45,6	
3	104	41,6	
4	94	37,6	
5	84	33,6	
6	74	29,6	
7	64	26,6	
8	54	22,6	
9	44	18,6	
10	34	14,6	
1	99	39,6	2
2	89	35,6	
3	79	31,6	
4	69	27,6	
5	59	23,6	
6	49	19,6	
7	39	15,6	
8	29	11,6	
9	19	7,6	
10	9	3,6	

A continuación se calculará ahora el porcentaje promedio de entrenamiento del sistema, a través de 4 experimentos:

Tabla 8.3 Resultados promedio de entrenamiento para 4 experimentos.

Número de ciclos	N_{error}	%Error	% Reconocimiento
1	124	36,3	63,7
2	90,6	32,8	67,2
3	82	29,3	70,7
4	73,3	25,9	74,1
5	64,7	22,4	77,6
6	57,3	18,9	81,1
7	38,7	15,5	84,5
8	30	12	88
9	21,3	8,5	91,5
10	12,7	5,1	94,9

Por lo que se puede ver en la tabla anterior, el mejor porcentaje de reconocimiento en el entrenamiento es 94,9%.

Ahora se presentan los resultados promedio del reconocimiento para 4 experimentos, variando 5 y 6 imágenes.

Tabla 8.3 Resultados promedio de entrenamiento para 4 experimentos.

Experimento	N_{error} por ciclo en reconocimiento	%Error	% Reconocimiento
1	18	7,2	92,8
2	12	4,8	95,2
3	36	14,4	85,6
4	19	7,6	92,4

8.2 Trabajos a futuro

Como trabajos a futuro se propone:

- Realizar la simulación dinámica con todos los parámetros que se involucran.
- Aumentar la base de imágenes, con la finalidad de aumentar el porcentaje de reconocimiento.
- Variar la opción de marcar las imágenes y aumentar clases para el reconocimiento de imágenes.
- Investigar e implementar otras opciones de tratamiento y procesamiento de imágenes, con el fin de ver la variación del reconocimiento.

REFERENCIAS

Bioinformática: simulación, vida artificial e inteligencia artificial. Rafael Lahoz-Beltrá

Simulador de manejo 1 febrero 2011

http://www.motorspain.com/wp-content/uploads/2007/11/toyota_new_driving_sim.jpg

Simulador de vuelo

http://www.tuexperto.com/wp-content/uploads/flickr_backup/2395654068_a842158dd3_o.jpg

Definición de barómetro 2 de febrero de 2011

<http://es.wiktionary.org/wiki/-baro>

definición de tacómetro 2 de febrero 2011

<http://es.wikipedia.org/wiki/Tacómetro>

Definición de temperatura 2 de febrero 2011

<http://definicion.de/temperatura/>

DEFINICION NUMERO MACH

<http://www.nibaje.com/?p=7033>

numero Reynolds

http://wapedia.mobi/es/Número_de_Reynolds_magnético

[10] oswald factor

http://www.wikilearning.com/curso_gratis/calculo_de_aviones-polar_del_avion/19258-6

escuela de vuelo y fundamentos

http://escueladevuelo.iespana.es/aerodinamica_7.htm

mecánica de vuelo

http://www.ultraligero.net/Cursos/mecanica/mecanica_de_vuelo.pdf

mecánica de vuelo articulo

<http://www.caedi.org.ar/pcdi/Area%2013/13-147.PDF>

Referencias capítulo 4

[1] Visión Computacional. Visitado abril 2011.

<http://ccc.inaoep.mx/~esucar/Libros/vision-sucar-gomez.pdf>

[2] Delamarre Q. and Faugeras O., "3D articulated models and multiview tracking with physical forces", Computer Vision and Image Understanding, Vol. 81, No. 3, March 2001, pp. 328-357.

[3] Sobrad E., "Reconocimiento y manipulación de objetos utilizando un brazo de robot", Limá-Perú 2003, pp 10-13.

- [4] Anil K. Maini, Varsha Agrawal. "Satellite Technology, Principles and Applications", Editorial John Wiley & Sons, Ltd, England 2007, pp 586.
- [5] Visión aplicada y elemento. Abril 2011.
http://gavab.escet.urjc.es/recursos/seminario_07.pdf
- [6] Visión computacional general. Abril 2011.
<http://robvis.mty.itesm.mx/~gordillo/Cursos/Sist-Vision-Compu-Prof/Vision-Compu-General.pdf>
- [7] Szeliski R., "Computer Vision Algorithms and Applications", Editorial Springer, USA 2011, pp. 24-28.
- [8] Ballard D. H. and. Brown C. M., "Computer Vision", Editorial Prentice Hall, U.S.A 1982, pp.15-18.
- [9] Twitter Astro_Soichi. Astronauta NASA, Soichi Noguchi, Abril 2010.
http://twitpic.com/photos/Astro_Soichi.
- [10] Misión Estacion internacional. NASA. Visitado Marzo 2010.
http://www.nasa.gov/mission_pages/station/expeditions/index.html
- [11] Roldán-Serrato K. L., Baydyck T., Kussul E., Velasco-Herrera G., "Recognition of Aerospace Images with Neural Networks", 1st International Congress on Instrumentation and Applied Sciences, México 2010.
- [12] Schwartz A. A., Soha J. M., "Multidimensional histogram normalization contrast enhancement", in Proc. 5th Canadian Symposium on Remote Sensing, 1978, pp.86-93.
- [13] Isasi P. , Galván I., "Redes de neuronas artificiales. Un enfoque práctico", Pearson Educación, Madrid, 2004, pp. 248.
- [14] Duda R., Peter E. , David G. Stork, "Pattern Classification", 2a edición, John Wiley, USA, 2001, pp.352.
- [15] Kussul E., Baidyk T., Wunsch D. C., "Neural Networks and Micromechanics", Springer, Berlin 2010, pp. 221.
- [16] Rodríguez F. M., "Desarrollo de un sistema basado en un clasificador neuronal para reconocimiento de orugas", tesis UNAM, México, abril 2010, pp. 91.
- [17] Programa de monitoreo permanente de la superficie terrestre NASA. Visitado mayo2011.
<http://eosps0.gsfc.nasa.gov/>
- [18] Programas de investigación para reconocer la calidad del aire, etc. Visitado 2011.
http://www.arl.noaa.gov/HYSPLIT_info.php
- [19] Programa de teleobservación para arrojar información. Visitado mayo 2010.
<http://maps.geog.umd.edu/firms/>
- [20] Proporciona información científica confiable para describir fenómenos. Visitado 2011.
http://www.usgs.gov/aboutusgs/who_we_are/
- [21] Jensen, J. R., "Introductory Digital Image Processing: A Remote Sensing Perspective, Upper Saddle River", N.J., Prentice Hall, 1996.
- [22] Richards, John A. y Xiuping Jia, "Remote Sensing Digital Image Analysis. An Introduction", Springer, 1999.
- [23] Visualizar e detalle el medio físico del país. INEGI. Visitado abril 2011.
<http://www.inegi.org.mx/geo/contenidos/topografia/carta50000.aspx>

Referencias cap 5

- [1] Duda Richard. O., Hart Peter E., Stork David G., "Pattern Classification", 2a edicion, John Wiley & Sons, USA, 2001, pp.352.

- [2] Carleer A.P., Debeir O., and Wolff E., Assessment of Very High Spatial Resolution Satellite Image Segmentations. *Photogrammetric Engineering & Remote Sensing*, Vol. 71, No. 11, November 2005, pp. 1285–1294.
- [3] Anil K. Maini, Varsha Agrawal. "Satellite Technology, Principles and Applications", Editorial John Wiley & Sons, Ltd, England 2007, pp 586.
- [4] Soichi Noguchi, Twitter Astro_Soichi. Astronauta NASA. Abril 2010. http://twitpic.com/photos/Astro_Soichi.
- [5] Phillips D., "Image processing in C", Second Edition, Electronic Edition 1.0, 26 April 2000, pp. 816. <http://homepages.inf.ed.ac.uk/rbf/BOOKS/PHILLIPS/>
- [6] Baidyk T., Kussul E., "Redes Neuronales, visión computacional y micro- mecánica", Editorial Itaca UNAM, México, 2009, pp. 157.
- [7] Microsoft Paint Define. 2011 http://es.wikipedia.org/wiki/Microsoft_Paint
- [8] Zhang, Y.J., Texture-Integrated classification of urban treed areas in high-resolution color-infrared imagery, *Photogrammetric Engineering & Remote Sensing*, Vol. 67, No. 12, 2001, pp.1359–1365.
- [9] Thomas, N., C. Hendrix, and R.G. Congalton, A comparison of urban mapping methods using high-resolution digital imagery, *Photogrammetric Engineering & Remote Sensing*, Vol. 69, No. 9, 2003, pp. 963–972.
- [10] Cocquerez, J.P., and S Philipp, *Analyse d'images: filtrage set segmentation*, Masson, Paris, 1995, pp. 457.
- [11] Vailaya A., Jain A., Zhang H.J., On Image Classification: City vs. Landscape, *IEEE Workshop on Content - Based Access of Image and Video Libraries*, Santa Barbara California, 1998. pp. 23.
- [12] Makeyev O., Sazonov E., Baidyk T., Martin A., Limited receptive area neural classifier for texture recognition of mechanically treated metal surfaces, *Neurocomputing*, Issue 7-9, Vol. 71, March 2008, pp.1413-1421.
- [13] Viñuela Isasi, Pedro. Inés M. Galván León. *Redes de neuronas artificiales. Un enfoque práctico*. España. Prentice Hall, 2004.

Referencias capítulo 6

- [1] Horstman J., "The scientific American Day in the life of your brain", Editorial John Wiley & Sons and Scientific American, USA, 2009, pp.276.
- [2] Maren A. J., Craig T. H., Pap. R. M., "Handbook of Neural Computing Applications", Editorial Academic Press, USA, 1990.
- [3] Viñuela Isasi, Pedro. Inés M. Galván León. "Redes de neuronas artificiales. Un enfoque práctico", Prentice Hall, España, 2004, pp.227.
- [4] Kussul E., Baidyk T., Wunsch D., "Neural Networks and Micromechanics", Editorial Springer, Berlin 2010, pp. 226.
- [5] Mitchell T. M., "Machine Learning", Editorial Mc Graw-Hill, USA 1997, pp. 400.
- [6] Kussul E., Baidyk T., Lukovitch V.V., Rachkovskij D. A., Adaptive High Performance Classifier Based on Random Threshold cNeurons, *Proceeding of Twelfth European Meeting on Cybernetics and System Research (EMCSR-94)*, Austria, Viena, abril 5-8, 1994, pp.1687-1695.

- [7] Baidyk T., Kussul E., "Redes Neuronales, visión computacional y micromecánica", Editorial Itaca UNAM, México, 2009, pp. 157.
- [8] Makeyev O., Sazonov E., Baidyk T., Martin A., Limited receptive area neural classifier for texture recognition of mechanically treated metal surfaces, Neurocomputing, Issue 7-9, Vol. 71, March 2008, pp.1413-1421.
- [9] Kussul E., Baidyk T., Wunsch D.C., "Neural Networks and Micromechanics", Springer Verlag, 2010, pp. 254.
- [10] Rodríguez J. L., "Niveles de luminosidad de una imagen", Noviembre 2007.
<http://www.dzoom.org.es/noticia-1708.html>
- [11] González C. G. "Reconocimiento de imágenes biomédicas con redes neuronales", Tesis UNAM, México 2011, pp. 100.
- [12] Gonzalez R. C., Woods R. E., "Digital Image Processing", Prentice Hall, USA 2002, pp. 779.
- [13] Marr, D., Vision, Freeman, San Francisco, 1982.
- [14] Julesz, B., "Experiments in the Visual Perception of Texture", Sci. Am., 1975,232, pp.34-43.
- [15] Anil K. Maini, Varsha Agrawal. "Satellite Technology, Principles and Applications", Editorial John Wiley & Sons, Ltd, England 2007, pp 586.
- [16] Rifman, S. S., "Digital rectification of ERTS multispectral imagery: Symposium on Significant Results Obtained from ERTS-1", NASA 1973, SP-327, p. 1131-1142.
- [17] Rifman, S. S., and others, "Experimental study of application of digital image processing techniques to Landsat", data: TRW Systems Group Report 26232-6004-TU-00 for NASA Goddard Space Flight Center Contract NAS 5-20085, Greenbelt, Md 1975.
- [18] Duda Richard. O., Hart Peter E., Stork David G., "Pattern Classification", 2a edición, John Wiley & Sons, USA, 2001, pp.637.
- [19] Anderson, J., "A theory for the recognition of items from short memorized list", 1973, pp. 417-438.
- [20] Hecht-Nielsen, R., "Neurocomputing", Addison-Wesley, Massachusetts, 1990.
- [21] Editor de ecuaciones. LaTeX. <http://rinconmatematico.com/latexrender/>
- [22] Rumelhart, D., G. Hinton, R. Williams, "Learning representations by back-propagating errors", Nature, 1986, pp. 533-536.
- [23] Rodríguez F. M.A., "Desarrollo de un sistema basado en un clasificador neuronal para reconocimiento de orugas", tesis UNAM, México 2010, pp.91.

ANEXO A. Glosario

NUMERO MACH

El Número Mach (M), es una medida de velocidad relativa que se define como el cociente entre la velocidad de un objeto y la velocidad del sonido en el medio en que se mueve dicho objeto, es decir:

$$M = \frac{V}{V_s}$$

donde:

M: Número Mach

V: Velocidad de un objeto

Vs: Velocidad del sonido

Es un número adimensional típicamente usado para describir la velocidad de los aviones. Mach 1 equivale a la velocidad del sonido (La velocidad del sonido en el aire a una temperatura de 20 °C es de 344,2 m/s), Mach 2 es dos veces la velocidad del sonido, etc.

Este número fue propuesto por el físico y filósofo austríaco Ernst Mach (1838-1916), uno de los más grandes teóricos de la física de los siglos XIX-XX, como una manera sencilla de expresar la velocidad de un objeto con respecto a la velocidad del sonido.

La utilidad del número de mach reside en que permite expresar la velocidad de un objeto no de forma absoluta en km/h o m/s, sino tomando como referencia la velocidad del sonido, algo interesante desde el momento en que la velocidad del sonido cambia dependiendo de las condiciones de la atmósfera. Por ejemplo, cuanto mayor sea la altura sobre el nivel del mar o menor la temperatura de la atmósfera, menor es la velocidad del sonido. De esta manera, no es necesario saber la velocidad del sonido para saber si un avión que vuela a una velocidad dada la ha superado: basta con saber su número de mach.

El vuelo se puede clasificar áspero en cinco categorías:

Subsónico: MA < 1

Sonic: Ma=1

Transónico: 0.8 < MA < 1.2

Supersónico: 1.2 < MA < 5

Hipersónico: MA > 5

El **Número de Reynolds Magnético R_m** es un [número adimensional](#) que se utiliza en [magnetohidrodinámica](#). Da una estimación de los efectos de la [advección](#) magnética

respecto a la difusión magnética. Típicamente se define como:

$$R_m = \frac{UL}{\eta}$$

En donde:

U es la velocidad del fluido.

L es una longitud característica.

η es la difusividad magnética.

Para $R_m \ll 1$ la advección es relativamente poco importante y por tanto el campo magnético tenderá a relajarse hacia un estado puramente difusivo determinado por las condiciones de contorno más que por el flujo.

Para $R_m \gg 1$ la difusión es relativamente poco importante en la escala de longitud L . Las líneas de flujo del campo magnético son adveccionadas con el flujo magnético hasta que los gradientes son concentrados en regiones de escala de longitud suficientemente pequeñas para que la difusión pueda igualar a la advección.

Terminología del ala

1- Envergadura: b .- Es la distancia de punta a punta del ala.

2- Superficie alar: S .- Es la superficie de las alas, incluyendo la parte del ala que pueda estar cubierta por el fuselaje o góndolas de los motores, como si no existieran estos elementos.

3- Cuerda media: c .- Normalmente los perfiles que constituyen el ala suelen ser distintos a lo largo de la envergadura, y además, las cuerdas que los constituyen van disminuyendo desde el encastre hasta las puntas. Se define cuerda media, como aquella que, multiplicada por la envergadura, es igual a la superficie alar: $c \cdot b = S$.

4- Estrechamiento (taper ratio): λ .- Se define por el cociente: C_t / C_r en donde C_r es la cuerda del perfil en el encastre y C_t es la cuerda del perfil en la punta.

5- Alargamiento (aspect ratio): A .- Es la relación entre la envergadura y la cuerda media. $A = b / c$ o lo que es igual $A = S / c^2$.

El alargamiento varía desde 3 ó 4 en aviones muy rápidos hasta 20 ó 30 en algunos planeadores. El DC-6 tiene 9,44 de alargamiento, 8 el Caravelle VI-R, 7,2 el Boeing 727 y 6,96 el Boeing 747.

6- Línea del 25% de la cuerda.- Es aquella que se obtendría si uniéramos todos los puntos a lo largo de la envergadura que están situados a un 25 % de su cuerda contados a partir del borde de ataque.

7- Flecha (sweep) ϕ .- Es el ángulo que forma la línea del 25% y una perpendicular al eje longitudinal del avión. Si el ala no tuviera estrechamiento, este ángulo sería el mismo que el formado por el borde de ataque del ala, y la perpendicular al eje longitudinal. La flecha puede ser progresiva o regresiva. En los reactores comerciales modernos oscila entre 30° y 40° . Así el DC-8 tiene 30° , 32° el B-727 y $37,5^\circ$ el B-747 de flecha regresiva.

8- Cuerda media aerodinámica (mean aerodynamic chord) - MAC.- Es la que tendría un ala rectangular (estrechamiento uno) y sin flecha que produjera el mismo momento y sustentación.

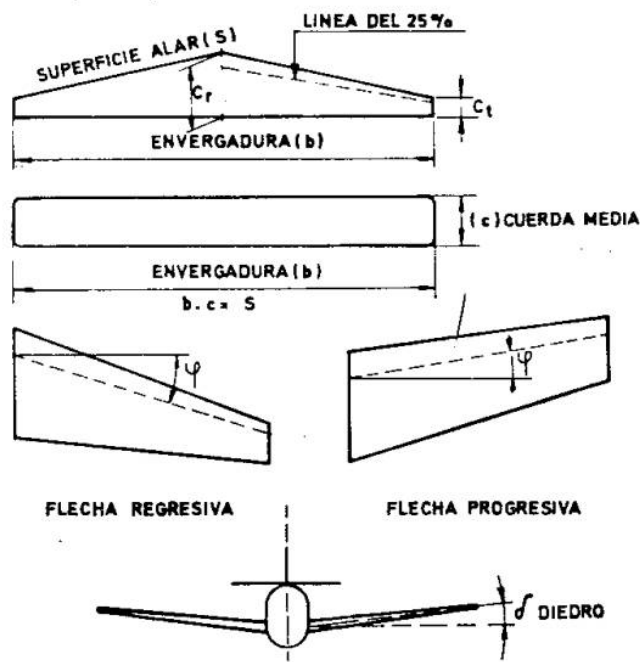
La posición de la cuerda media aerodinámica respecto al eje longitudinal del avión puede hallarse mediante fórmulas apropiadas o geoméricamente. Su posición es de importancia en cuanto a consideraciones de estabilidad longitudinal.

9- Diedro: Es el ángulo formado por la intersección de dos planos (planos del ala) y tiene influencia en la estabilidad lateral del avión.

10- Torsión: puede ser geométrica o aerodinámica.

La torsión geométrica consiste en que los ángulos de ataque de cada uno de los perfiles que componen el ala sean diferentes, dando a la sección de punta un ángulo de ataque menor que en el encastre. Esta torsión, giro relativo de las cuerdas, se suele hacer gradualmente desde el encastre a la punta del ala.

La torsión aerodinámica se logra con perfiles diferentes a lo largo del ala, de forma que el ángulo de sustentación nula varíe para los diferentes perfiles que componen el ala. En definitiva, el efecto es el mismo que se consigue con la torsión geométrica. Una forma de lograr la torsión aerodinámica es aumentando las curvaturas de los perfiles, progresivamente desde el encastre a la punta, de forma que aumente el valor del C_l máx (coeficiente de sustentación) en las puntas.



Forma en Planta de Ala

Todo piloto conoce a la perfección que el perfil alar posee un efecto considerable sobre el comportamiento del ala. Veremos aquí como influye la forma en planta de ala, la cual es también crítica. Por forma en planta significamos una vista superior o inferior del ala.

Técnicamente, esta forma es la distribución de la longitud a lo largo de la envergadura. Un ala rectangular tiene una cuerda constante, un ala ahusada o decreciente posee una cuerda que disminuye constantemente hacia la punta.

La manera como está distribuida la cuerda es el factor que determina la forma en que reparte la sustentación a lo largo de la envergadura. Esta repartición posee un efecto sobre la resistencia inducida (resistencia que se genera a causa de la sustentación debido a los torbellinos generados en la puntera del ala por la diferencia de presiones extradós - intradós). Así, algunas formas de ala tienen más resistencia inducida que otras, aunque el área alar total pueda ser la misma.

Las alas de forma elíptica poseen la mínima resistencia inducida posible. Este tipo de ala es sin embargo costosa y difícil de construir. El Spitfire de la Segunda Guerra Mundial constituye un clásico ejemplo de avión con ala elíptica. A medida que los aeroplanos se hicieron más complejos y la producción más costosa, el ala elíptica dio paso al ala ahusada. Se encontró así que una forma rectangular o ligeramente ahusada sería casi tan eficiente como la elíptica y mucho más fácil de construir. En primera instancia las alas ahusadas parecieron ser las ideales.

La carga alar a lo largo de la envergadura no es constante y va decreciendo hacia la punta del ala. La figura 1 muestra la distribución típica de carga sobre la semi-envergadura de un ala; es decir, desde la raíz o encastre hasta la punta. Las tensiones de flexión sobre el ala se incrementan desde la punta hacia el interior en dirección de la raíz. Los largueros deberán ser lo suficientemente resistentes como para soportar las tensiones de la parte interior del ala, particularmente si la misma es de tipo cantilever (tipo de ala cuya estructura se encuentra tomada al fuselaje como una viga empotrada en una pared).

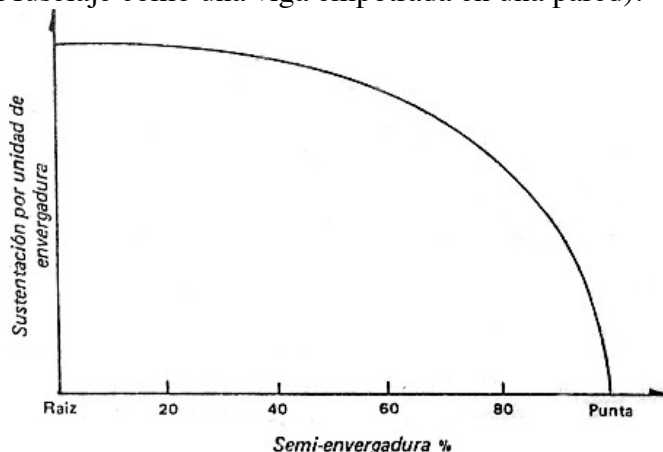


Fig 1. Distribución típica de cargas sobre la envergadura.

Si un ala es ahusada en la longitud de la cuerda, se requerirá menos estructura portante en las puntas y en la parte interior. Esto da como resultado un ahorro de peso estructural, lo cual es siempre una de las premisas fundamentales en el diseño de una aeronave. Por otro

lado, ahusando el espesor del ala al mismo tiempo que la cuerda, el ala resulta ser mucho más esbelta. Hasta aquí el ala ahusada pareciera ser la mejor selección. Tiene, sin embargo, algunas desventajas y una de ellas es el comportamiento durante la pérdida de sustentación. Este fenómeno requiere una explicación aparte. Un ala no entra en pérdida en forma simultánea a lo largo de toda su envergadura. Ciertos sectores del ala entran en pérdida primero. La pérdida progresa desde estos sectores, hasta que ha entrado en pérdida un área suficientemente grande que hace que aparezca abruptamente “la pérdida de sustentación” y que el peso del aeroplano no pueda ser soportado. Aún así, algunas secciones del ala no han entrado completamente en pérdida.

La razón para esta pérdida de sustentación no homogénea es que el ángulo de ataque efectivo de cada sección del ala es diferente a lo largo de la envergadura de la misma. El origen de esta variación es la distribución del torbellino descendente (“downwash”) causado a su vez por la manera en la cual se genera el vórtice de la punta del ala. Éste a su vez depende de la forma en planta del ala. Así, vemos que la referida forma determina la distribución del ángulo de ataque efectivo a lo largo de la envergadura.

La figura 2 muestra la distribución del ángulo de ataque efectivo a lo largo de la envergadura para alas de tipo rectangular, ligeramente ahusadas, muy ahusadas y elípticas. Nótese que para un ala elíptica perfecta el ángulo de ataque efectivo es constante. Para una rectangular el máximo ángulo de ataque está en la raíz o encastre y para una ahusada está hacia afuera mucho más cerca de la punta cuanto más ahusada es. A medida que el ala va incrementado su ángulo de ataque, la sección con mayor ángulo de ataque efectivo llegará primeramente al ángulo de pérdida y a partir de allí comenzará la misma.

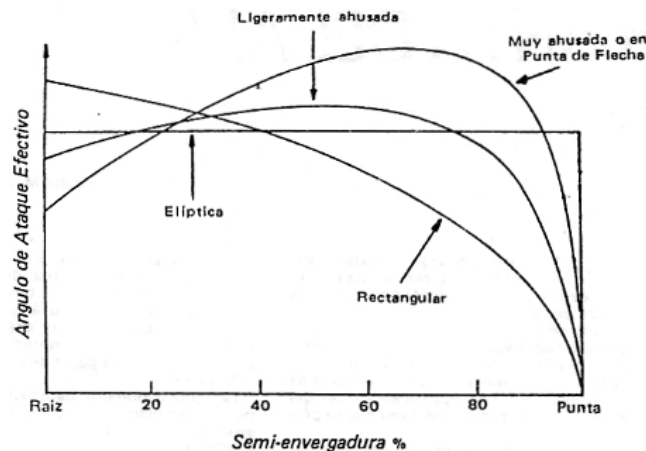


Fig 2. Distribución del ángulo de ataque efectivo sobre la envergadura.

La figura 3 muestra, para las distintas configuraciones alares, las zonas en las cuales se inicia la pérdida y como ésta va progresando. Las alas ahusadas comienzan a entrar en pérdida en la parte externa. Esto es, donde usualmente se halla ubicado el alerón, de tal manera que con esta porción del ala en pérdida y con la sección interior aún en vuelo se anula el control del alabeo, sobre todo en alas muy ahusadas.

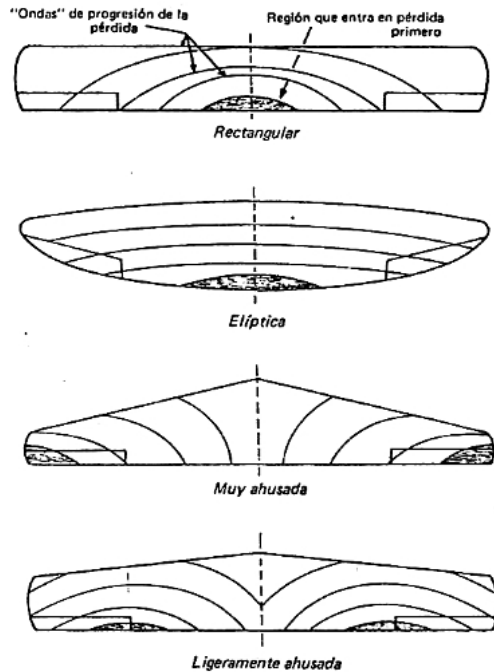


Fig.3. Modelo de la pérdida de sustentación para diferentes formas en planta sin torsión.

Esta situación puede ser de lo más embarazosa si se está por iniciar una maniobra de enderezamiento durante un aterrizaje con fuertes ráfagas de viento; surge por lo tanto la pregunta: ¿Cómo diseñar un aeroplano seguro si el mismo posee alas ahusadas? Hay varias formas de mantener el alerón en “vuelo”, tal como una ranura enfrente del mismo o la instalación de “bandas o tiras de pérdidas” en la zona superior interna del ala como para forzar que esta parte entre en pérdida antes. Cualesquiera de estos dos métodos reducen la eficiencia del ala. Las ranuras adicionan resistencia al avance. Forzar una pérdida de sustentación aún sobre un sector del ala, termina por subir la velocidad de pérdida por encima de lo que realmente podría ser. La manera más usual de solucionar este problema consiste en hacer que los ángulos de ataque de cada uno de los perfiles que componen el ala sean diferentes, dando la sección de la punta un ángulo de ataque menor que el de la raíz. Esta torsión o giro relativo de las cuerdas se suele hacer gradualmente desde la raíz a la punta del ala. De esta manera se consigue que las secciones interiores lleguen al ángulo de pérdida primero. Este artilugio se muestra en la Figura 4 y se denomina “Washout” (corrimiento).



Fig. 4. Vista lateral de un ala ahusada con torsión geométrica

Pareciera ser un remedio bastante sencillo, sin embargo origina que algunas secciones se hallen en un ángulo de ataque distinto que el óptimo (el ángulo en que la sustentación es

máxima y la resistencia al avance mínima) con lo cual se agrega resistencia durante el vuelo normal.

Aún el ala elíptica puede requerir alguna torsión para que entre primero en pérdida la sección interior. De esta forma se agrega resistencia de perfil, la que en algunos casos podría llegar a preponderar sobre la reducción de resistencia inducida.

Observando el ala rectangular vemos que la pérdida de sustentación se inicia en la raíz del ala, es decir la zona donde es más deseable. Hay algo muy importante en favor del ala rectangular y es que todas las costillas son de igual tamaño, pudiendo ser estampadas por la misma matriz. La tela del recubrimiento es fácil de cortar y aplicar, en tanto que el larguero no necesita ser ahusado. En una palabra es económica.

Entonces, ¿por qué todas las alas no son rectangulares?, tienen una gran desventaja (además de ser antiestética) y es el peso innecesario de la porción exterior.

Este es un factor a ser particularmente tenido en cuenta para grandes aeronaves. Lo que puede hacerse, como solución de compromiso, es que el ala sea rectangular en la parte interna y ahusada en la externa. Otro campo donde el peso adicional de un ala totalmente rectangular tiene su influencia es el vuelo a altitudes elevadas. Aquí la resistencia inducida es una porción significativa de la resistencia total. La resistencia inducida es la única afectada por el peso (la resistencia inducida es proporcional al coeficiente de sustentación al cuadrado).

Así, el aeroplano está diseñado para crucero a altitud elevada, la mayor resistencia inducida del ala rectangular puede invalidar la reducción en la resistencia parásita resultante de un menor requerimiento de torsión.

Lo que se ha discutido hasta el momento como torsión del ala o “washout” es lo que se denomina torsión geométrica. Esta es realmente una torsión física del ala. Hay otra “treta” usada por los diseñadores, conocida como torsión aerodinámica. Esta no es realmente una torsión, sino un cambio en el tipo de perfil a lo largo de la envergadura.

Cerca de la punta del ala se emplea un perfil que entra en pérdida a mayor ángulo de ataque que el instalado hacia el interior de la misma. De esta manera las secciones interiores alcanzarán su ángulo de pérdida con anterioridad a la punta. Esto se consigue aumentando progresivamente la curvatura de los perfiles desde la raíz hacia la punta del ala, de forma de incrementar el coeficiente de sustentación máximo de las puntas. El efecto obtenido es el mismo que el de la torsión de allí su denominación “torsión aerodinámica”.

La torsión aerodinámica, comparativamente con la torsión geométrica, da generalmente como resultado una menor resistencia en crucero para las secciones externas. Sin embargo complica el diseño y por lo tanto el proceso de producción. En aeroplanos de alta performance esta complicación puede llegar a ser justificada.

A veces, en el caso de alas muy ahusadas se emplea una combinación de torsión aerodinámica y geométrica. Con frecuencia, en la mayoría de las alas rectangulares se introduce algo de torsión geométrica. Este es un factor de seguridad adicional, puesto que la mayoría de los aeroplanos de ala rectangular se utilizan para entrenamiento. El grado de torsión no es tan grande como el requerido en un ala ahusada.

Luego de toda esta discusión, vendrá la pregunta ¿cuál es la mejor forma en planta de un ala? Todo depende del objetivo primordial del diseño de la aeronave. Para aeroplanos de entrenamiento y aeroplanos privados de precio competitivo se puede elegir el ala rectangular, la que es más económica y con mejores características de pérdidas de sustentación. Por otro lado, los aeroplanos para ejecutivos y los utilitarios puede que no

sean capaces de tolerar el peso adicional y la resistencia inducida extra que implican las alas rectangulares. Para este propósito el ala ahusada puede tener más ventajas que desventajas.

El diseño de aeronaves implica un complejo sistema de “trueques e intercambios”. Para obtener ciertas características deseables se deben sacrificar otras características. Por ejemplo, una alta velocidad de crucero puede trocarse con una corta distancia de despegue, un largo alcance se puede intercambiar por una mayor capacidad de carga. Este concepto no solo involucra performance, sino muchas otras consideraciones, tales como costo inicial, costos de operación, mantenimiento y otros. El primer trabajo de un diseñador es decidir cuales rubros son los más importantes para el aeroplano bajo consideración.

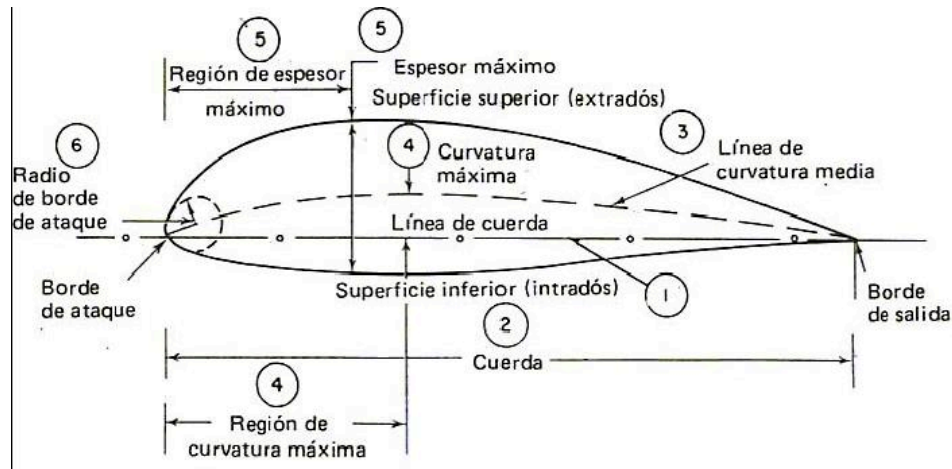
Si el ala de un avión es gruesa o esbelta, corta o larga, hay que estar persuadido de que la misma no es el resultado del ancho del tablero de dibujo del proyectista ni del capricho de la esposa del presidente de la compañía. La misma es probablemente una resultante del diseño optimizado para el tipo de vuelo para el cual el aeroplano está construido.

Perfiles Alares

Un perfil alar es una sección del ala de un avión. En el estudio de los perfiles se ignora la configuración en proyección horizontal del ala, como así también los efectos de extremo del ala, flecha, alabeo y otras características de diseño.

Terminología

- 1- La línea de cuerda es una línea recta que une el borde de ataque y el borde de fuga del perfil.
- 2- La cuerda es la longitud de la línea anterior. Todas las dimensiones de los perfiles se miden en términos de la cuerda.
- 3- La línea de curvatura media es la línea media entre el extradós y el intradós.
- 4- Curvatura máxima es la distancia máxima entre la línea de curvatura media y la línea de cuerda. La posición de la curvatura máxima es importante en la determinación de las características aerodinámicas de un perfil.
- 5- Espesor máximo es la distancia máxima entre la superficie superior e inferior (extradós e intradós). La localización del espesor máximo también es importante.
- 6- Radio del borde de ataque es una medida delafilamiento del borde de ataque. Puede variar desde 0, para perfiles supersónicos afilados, hasta un 2 por 100 (de la cuerda) para perfiles más bien achatados.



Variables geométricas en los perfiles

En la geometría de un perfil existen cuatro variables principales:

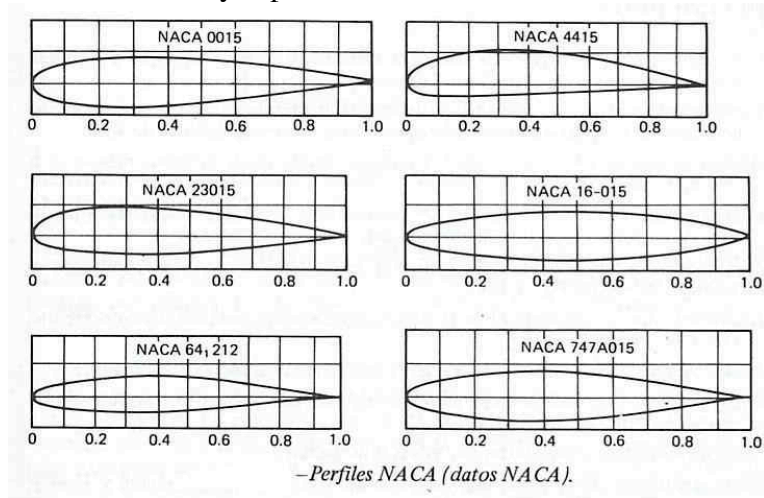
- 1- Configuración de la línea de curvatura media. Si esta línea coincide con la línea de cuerda, el perfil es simétrico. En los perfiles simétricos la superficie superior e inferior tiene la misma forma y equidistan de la línea de cuerda.
- 2- Espesor.
- 3- Localización del espesor máximo.
- 4- Radio del borde de ataque.

Clasificación de los perfiles

La mayor parte del desarrollo de perfiles en los Estados Unidos ha sido realizado a partir de 1929 por el Comité Nacional de Aeronáutica (NACA), que fue precursor de la Administración Nacional de la Aeronáutica y del Espacio (NASA). Las primeras series estudiadas fueron las llamadas “de cuatro dígitos”. El primero de los dígitos daba la curvatura en porcentaje de la cuerda; el segundo daba la posición de la curvatura máxima en décimas de la cuerda y los dos últimos dígitos el espesor máximo en porcentaje de la cuerda. Por ejemplo, un perfil NACA 2415 tiene la curvatura máxima del 2 % de la cuerda, situada en el punto del 40 % de la cuerda (medido desde el borde de ataque) y con un espesor máximo del 15 % de la cuerda. El perfil NACA 0012 es un perfil simétrico (de curvatura 0) y con un espesor del 12 % de la cuerda.

El desarrollo posterior llevó a las series de cinco dígitos, “series 1”, y, con la llegada de altas velocidades, los denominados de flujo laminar. Estos corresponden a las “series 6 y 7” y resultan del desplazamiento hacia atrás del punto de espesor máximo y la reducción del radio de borde de ataque. De este diseño se obtiene dos resultados principales. En primer lugar se desplaza hacia atrás el punto de presión mínima, aumentando con ello la distancia desde el borde de ataque en la que existe flujo laminar y reduciendo la resistencia. En segundo lugar, aumenta el número crítico de mach, permitiendo incrementar la velocidad del avión sin la aparición de problemas de compresibilidad. En los perfiles de “serie 6”, el primer dígito indica la serie y el segundo la posición de presión mínima en décimas de la

cuerda. El tercer dígito representa el coeficiente de sustentación en décimas y los dos últimos dígitos el espesor en porcentaje de la cuerda. Por ejemplo el NACA 64-212 es un perfil de la serie 6 con el punto de presión mínima en el 40 % de la cuerda, un coeficiente de sustentación de diseño de 0,2 y espesor del 12 % de la cuerda.



Flujo laminar

Corriente o flujo uniforme de aire con muy poca transferencia de momento entre las capas paralelas.

Número crítico de Mach

A cierto número de Mach de vuelo, la velocidad local en la parte superior del ala alcanza la del sonido, $Mach = 1$. El número de Mach de vuelo que esto sucede se conoce como número crítico de Mach.

Centro de presión

Punto de la cuerda donde actúa la fuerza aerodinámica.

Coeficiente de sustentación

Puede definirse como la relación entre la presión de sustentación y la presión dinámica, es una medida de la efectividad del perfil para producir sustentación.

Mandos de vuelo

Los mandos de la cabina del piloto accionan las "superficies de mando" (superficies aerodinámicas que provocan el movimiento del avión alrededor de sus 3 ejes). En vuelo, cualquier movimiento del avión afectará al menos a uno de sus 3 ejes:

- Eje Lateral o Transversal
- Eje Longitudinal
- Eje Vertical

El eje LATERAL o TRANSVERSAL (LATERAL AXIS) es un eje imaginario que se extiende de punta a punta de las alas del avión. El movimiento que realiza el avión alrededor de este eje se denomina CABECEO (Pitch).

La superficie de mando del cabeceo es el TIMÓN DE PROFUNDIDAD o también denominada ELEVADOR (ELEVATORS).

Al tirar de los cuernos de mando hacia atrás (hacia el piloto) se produce el "encabritamiento" (cabeceo hacia arriba o elevación del morro del avión), y al empujar los cuernos de mando hacia adelante se produce el "picado" (cabeceo hacia abajo o bajada del morro del avión).

El eje LONGITUDINAL (LONGITUDINAL AXIS) es un eje imaginario que se extiende desde el morro a la cola del avión. El movimiento que realiza el avión alrededor de este eje se denomina ALABEO (Roll).

Las superficies de mando del alabeo son los ALERONES (AILERONS). Al girar los cuernos de mando se produce la deflexión diferencial de los alerones: al tiempo que el alerón de una de las alas sube, el alerón de la otra ala baja, siendo el ángulo de deflexión proporcional al grado de giro de los cuernos de mando.

El alerón que es deflexionado hacia abajo, produce un aumento de sustentación en su ala correspondiente, provocando el ascenso de la misma, mientras que el alerón que es deflexionado hacia arriba, produce en su ala una disminución de sustentación, motivando el descenso de la misma.

Por ejemplo, si deseamos efectuar una inclinación a la izquierda, giraremos el cuerno de mando hacia la izquierda: el alerón derecho descenderá elevando el ala derecha, y simultáneamente, el alerón izquierdo se deflexionará hacia arriba produciendo una pérdida de la sustentación en el ala izquierda y por tanto su descenso.

El eje VERTICAL (VERTICAL AXIS) es un eje imaginario que, pasando por el centro de gravedad del avión, es perpendicular a los ejes transversal y longitudinal.

El movimiento que realiza el avión alrededor de este eje se denomina GUIÑADA (Yaw). La superficie de mando de la guiñada es el TIMÓN DE COLA o TIMÓN DE DIRECCIÓN (RUDDER).

El control sobre el timón de dirección se realiza mediante los pedales. Para conseguir un movimiento de guiñada hacia la derecha, el piloto presiona el pedal derecho, generando así una deflexión de la superficie del timón de dirección hacia la derecha.

Se consideran superficies de mando PRIMARIAS: el Timón de profundidad o elevador, los Alerones y el Timón de dirección.

Se consideran superficies de mando SECUNDARIAS: los Flaps, Compensadores, Slats y Spoilers (aerofrenos).

En aviones reactores, capaces de desarrollar altas velocidades, las resistencias y los esfuerzos necesarios para deflectar las superficies aerodinámicas, son muy superiores a los de un avión cuyo vuelo de crucero es a baja velocidad. Por ello, además de dotarles de una transmisión hidráulica para mover las superficies, se añaden otras superficies de mando, como los slats, los flaps, los spoilers o aerofrenos y los compensadores.

Tanto los flaps como los slats incrementan el área del ala y la curvatura de la misma, produciendo un aumento de la sustentación mejorando la performance durante el despegue y el aterrizaje.

Existen flaps de borde de ataque (leading edge flaps, L.E flaps) y flaps de borde de fuga (trailing edge flaps T.E flaps) siendo estos últimos los más comunes. Los slats se encuentran en el borde de ataque del ala (L.E, slats).

La función de los spoilers o aerofrenos es ayudar a los alerones a controlar al aeroplano en el rolido y suministrar a la vez control de frenado (speedbrakes) para reducir la sustentación o incrementar el arrastre durante el aterrizaje.

Para evitar la continua acción del piloto sobre los mandos, se usan unas ruedas o compensadores de profundidad, alabeo y dirección. Estas ruedas controlan unas superficies aerodinámicas de pequeño tamaño llamadas compensadores o aletas compensadoras, que se mueven en sentido contrario al de la superficie de mando principal en la que se encuentran montadas, manteniéndola a la misma posición deseada.

Por ejemplo, si un viento lateral tiende a desviar el avión hacia la derecha de su ruta, el piloto puede corregir el efecto del viento (Deriva) presionando el pedal izquierdo; para evitar la presión constante sobre el pedal, el piloto puede girar la rueda del compensador de dirección hacia la izquierda.

Así, la aleta compensadora se moverá hacia la derecha, obligando al timón de dirección a desplazarse un poco a la izquierda. Manteniéndose así deflectado, el avión habrá corregido su desviación y el piloto no estará obligado a presionar constantemente el pedal.

De la misma manera, cuando se quiere mantener una actitud de subida, o de bajada, o compensar en profundidad, el piloto gira hacia adelante o hacia atrás la rueda del compensador de profundidad hasta que ya no necesite empujar o tirar de los cuernos de mando.

El compensador de alabeo suele accionarse cuando el avión tiende a llevar un plano más caído que el otro, por ejemplo cuando existe una gran diferencia de peso de combustible entre las dos alas.

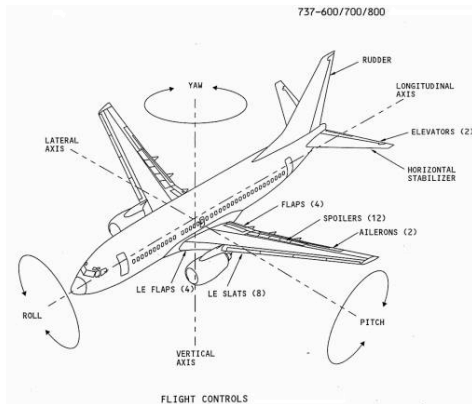


Figura 1.

En la figura 1 se puede apreciar los movimientos del avión (roll o roldo, yaw o guiñada y pitch o cabeceo) sobre sus tres ejes (longitudinal, vertical y lateral) y la ubicación de las superficies de comandos primarias (elevators o elevador, rudder o timón de profundidad y ailerons o alerones) y secundarias (flaps, slats y spoilers).

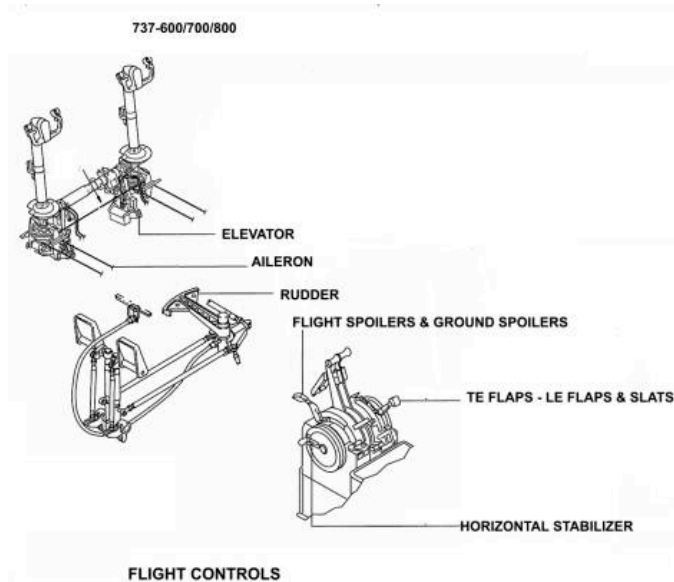


Figura 2.

Las superficies de comando se accionan desde la cabina del piloto con los controles de vuelo que aparecen en la figura 2.

En la parte superior izquierda se encuentra el bastón de mando que posee un mecanismo que transmite el movimiento al timón de profundidad o elevador para realizar el movimiento de cabeceo.

El mismo bastón de mando posee en la parte superior el cuerno o volante para accionar sobre los alerones los que producen el movimiento de alabeo.

Debajo del gráfico anterior aparece la pedalera la cual mediante un mecanismo de transmisión llega hacia el timón de dirección para producir el movimiento de guiñada.

A la derecha de la pedalera el gráfico muestra una parte del pedestal de control con la cual se accionan los spoilers de vuelo y de tierra (flight spoilers y ground spoilers), los flaps de borde de ataque y borde de fuga (T.E flaps y L.E flaps) y una rueda del compensador del estabilizador horizontal (horizontal stabilizer).

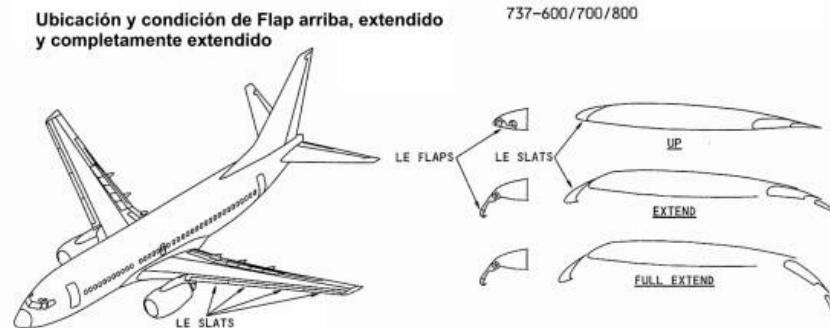


Figura 3.

La figura 3 muestra las ubicaciones y condiciones de slats y de flaps, totalmente retraídos, extendidos y totalmente extendidos.

FLY-BY-WIRE

Desde los albores de la aviación, y hasta los días del Stratocruiser y el Super Guppy, volar un avión solía requerir un gran esfuerzo físico. Batallando contra las inclemencias meteorológicas, los pilotos navegaban en sus aviones operando manualmente cables de control conectados a superficies tales como alerones, flaps, elevadores y timones.

Maniobrar los aviones más grandes hizo necesario algo más que la fortaleza humana. Fue así como se introdujeron potentes sistemas hidráulicos, los cuales eran operados por los pilotos, vía mandos, cables y poleas.

Sin embargo a principios de la década del '80, el diseño de los sistemas de control secundario comenzó a utilizar señales eléctricas que iban desde las palancas de potencia a través de computadoras y hasta los actuadores hidráulicos de las superficies de control. El nuevo sistema fly-by-wire (comandos eléctricos de vuelo) hizo extensiva esta tecnología al control primario del avión. Ya no se necesitaba el cuerno convencional de mando, dado que las órdenes salían del puesto de pilotaje por vía electrónica, y se lo reemplazó por una palanca o bastón lateral, más pequeño, el denominado sidestick. El nuevo sistema redujo el peso y la complejidad mecánica del avión, bajó los costos y representó para el piloto grandes ventajas, principalmente en materia de precisión, seguridad y ergonomía.

Gracias al papel mediador de las computadoras, que conocen el espectro total de capacidades técnicas y aerodinámicas del avión, los pilotos pueden hacer un uso total de éstas sin correr el riesgo de exceder su envolvente de vuelo. El sector de la envolvente que figura en las computadoras del sistema fly-by-wire se halla pre-

programado de manera de limitar las actitudes que pudiera adoptar la aeronave. También se aplica la protección contra las violaciones a los excesos de velocidad (Vmo/Mmo, bajas velocidades) independientemente de la orden dada por un piloto al avión a través del sidestick.

¿Que es el fly-by-wire?

Todos los aviones se controlan a través del movimiento de superficies fijadas a sus alas y empenajes. En una aeronave de pequeñas dimensiones, estas superficies, conectadas mecánicamente a los controles de la cabina de mando, son accionadas directamente por el piloto.

En aparatos más grandes y rápidos, como los aviones comerciales, se precisa un esfuerzo mucho mayor para mover esas superficies, lo que hace necesario el empleo de actuadores hidráulicos (del mismo modo que una dirección asistida facilita el manejo de un vehículo pesado).

El enlace entre los controles de la cabina de pilotaje y los actuadores hidráulicos de las alas y los empenajes continúa siendo mecánico y habitualmente se basa en cables de acero, que se extienden por todo lo largo y ancho de la aeronave. En un avión dotado de fly-by-wire, este enlace se sustituye por uno eléctrico. En el Airbus A320, por ejemplo, los computadores canalizan el flujo de esas señales eléctricas.

Conviene notar que el fly-by-wire elimina la necesidad de las molestas columnas de mandos de control.

Airbus ha sabido sacar pleno provecho de esto, sustituyendo esas columnas de control clásicas, por una pequeña palanca (stick) lateral de cómodo uso, que también permite la total visualización de los instrumentos y libertad de movimiento para el piloto.

Ya sea porque se utilicen columnas de control o palancas laterales y a pesar de que las señales se trasladan a través de computadoras, los pilotos mantienen permanentemente el gobierno de la aeronave.

¿Cuales son las ventajas del fly-by-wire?

Todos los aviones tienen limitaciones físicas: no pueden volar demasiado despacio, o entrarían en pérdida; ni demasiado rápido, ya que, o maniobran dentro de sus parámetros específicos o se efectuará una ruptura.

Esas limitaciones se conocen como envolvente de vuelo y sus fronteras no deben ser sobrepasadas en una operación normal. En el A320, la protección de la envolvente de vuelo de los controles fly-by-wire proporcionan un resguardo adicional para evitar exceder la mayoría de esas limitaciones. Esto ayuda a los pilotos a obtener las máximas prestaciones de su aeronave conservando el pleno control, lo que contribuye a aumentar la seguridad. Por ejemplo, la protección de la envolvente de vuelo incrementa considerablemente las posibilidades de sobrevivir a un viento de cizalladura, haciendo posible a los pilotos obtener de forma rápida y segura las máximas prestaciones en régimen de ascenso.

Si bien este resguardo del perfil de vuelo protege contra riesgos específicos, no puede sustituir el saber hacer y la vigilancia de los pilotos. La tripulación está, y siempre estará al mando. Además debido a que los controles fly-by-wire son eléctricos, tienen menos elementos móviles, y esto los hace más simples, ligeros y fáciles de mantener.

¿Que aviones tienen controles fly-by-wire?

El primer avión comercial que utilizó estos controles fue el Concorde, que voló por primera vez en 1969. Los Airbus A310, A300-600 y el Boeing 767 y 757 incorporan el fly-by-wire a principio de los '80 y lo usan de manera limitada para operar las superficies móviles del ala (Slats, flaps y spoilers). El A320 fue el siguiente, en 1988, convirtiéndose en el primer avión comercial en hacer pleno uso de estos modernos controles. El liderazgo de Airbus Industrie es seguido de cerca por Boeing ya que está instalando este sistema en sus 777.

El fly-by-wire se ha convertido en un patrón industrial para los aviones de nueva generación.

¿Que sucede si ocurre una avería en el computador de fly-by-wire?

En el A320 existen cinco computadores de fly-by-wire y la aeronave puede volar utilizando tan sólo uno de ellos; dichos computadores trabajan simultáneamente y, si uno de ellos falla, el siguiente toma automáticamente el relevo.

Como todos los demás sistemas importantes de un avión comercial, los computadores fly-by-wire están diseñados, construidos, probados y certificados bajo las normas más estrictas. Pero, no obstante, la eventualidad de una avería se toma en consideración en el curso de la concepción, previéndose sistemas de socorro.

Cada uno de los cinco ordenadores de fly-by-wire se compone, de hecho, de dos unidades que no cesan de vigilarse mutuamente. Además, tres de ellos están diseñados y construidos por un fabricante, mientras que los dos restantes están hechos por otro, con una concepción diferente independiente y utilizando diferentes componentes. Estos dos tipos de computadores se programan por equipos de trabajo que usan distintos lenguajes de programación, haciendo virtualmente imposible que el mismo fallo afecte a todos a la vez.

La probabilidad de que los cinco computadores fly-by-wire tengan una avería durante un vuelo es extremadamente remota: el riesgo está calculado en menos de una vez por un billón. Sin embargo, en el caso de pérdida del control, las tripulaciones de A320 pueden continuar pilotando el aparato a unos mandos mecánicos de socorro (compensación del timón horizontal y timón de dirección). Estos mandos auxiliares han sido ensayados en vuelo y empleados satisfactoriamente para realizar una aproximación y aterrizaje con fines de demostración.

Un fallo eléctrico completa es extremadamente improbable, debido a que cuenta con no menos de 5 fuentes de potencia diferentes (un generador en cada uno de los dos motores, la unidad de potencia auxiliar, un generador hidráulico ligado a una turbina de aire comprimido y baterías). Pero, si por jugar a lo imposible sobreviniese una avería eléctrica total, el A320 podría seguir volando gracias a sus controles mecánicos de socorro, tal como ya ha sido demostrado.

¿Quién controla la aeronave?

Los pilotos, como en todos los aviones. Esto es verdad tanto si están operando el aparato manualmente como lo que a menudo sucede en los aviones comerciales modernos y luego continúe a través de sistemas automáticos. En el A320, como en cualquier otra aeronave, los pilotos pueden recuperar siempre el control manual,

usando un botón que desconecte el sistema automático o por un contundente movimiento de los mandos.

Algoritmo de "Backpropagation"

Aproximadamente en el año de 1974, Paul Werbos desarrolla el método de entrenamiento *Backpropagation*, el algoritmo más utilizado en las aplicaciones de las redes neuronales artificiales [3].

La regla o algoritmo de aprendizaje es el mecanismo mediante el cual se van adaptando y modificando todos los parámetros de la red. En el caso del *perceptron* multicapa se trata de un algoritmo de aprendizaje supervisado; es decir, la modificación de los parámetros se realiza para que la salida de la red sea lo más próxima posible a la salida proporcionada por el supervisor o salida deseada. Por tanto, para cada patrón de entrada a la red es necesario disponer de un patrón de salida deseada.

Puesto que el objetivo es que la salida de la red sea lo más posible a la salida deseada, el aprendizaje de la red se formula como un problema de minimización del siguiente modo:

$$\text{Min } wE$$

siendo w el conjunto de parámetros de la red –pesos y umbrales- y E una función error que evalúa la diferencia entre las salidas de la red y las salidas deseadas. En la mayor parte de los casos, la función error se define como:

$$E = \frac{1}{N} \sum_{n=1}^N e(n)$$

Donde N es el número de patrones o muestras y $e(n)$ es el error cometido por la red para el patrón n , dado por:

$$e(n) = \frac{1}{2} \sum_{i=1}^{n_c} (s_i(n) - y_i(n))^2$$

siendo $Y(n)=(y_1(n), \dots, Y_{n_c}(n))$ y $S(n)=(s_1(n), \dots, S_{n_c}(n))$ los vectores de salida de la red y salidas deseadas para el patrón n , respectivamente.

De este modo, si W es un mínimo de la función error E , en dicho punto el error es próximo a cero, lo cual implica que la salida de la red es próxima a la salida deseada, alcanzando así la meta de la regla de aprendizaje.

Por tanto, el aprendizaje del perceptron multicapa es equivalente a encontrar un mínimo de la función error.

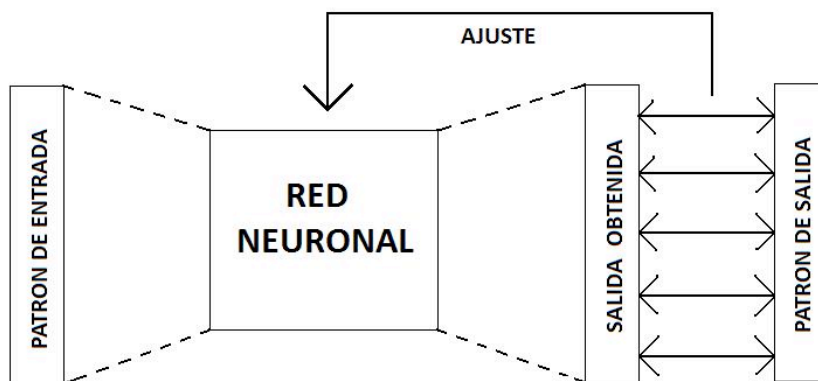
Debido a que las neuronas de la red están agrupadas en capas de distintos niveles, es posible aplicar el método del gradiente de forma eficiente, resultando el algoritmo de retro propagación, años más tarde (1986) perfeccionado por Rumelhart [20], para redes neuronales más complejas. También es llamada *regla delta generalizada*. El término de *retropropagación* o en inglés "*Backpropagation*" se utiliza debido a la forma de implementar el método del gradiente en el perceptron multicapa, pues el error cometido en la salida de la red es propagado hacia atrás, transformándolo en un error para cada una de las neuronas ocultas de la red[3].

Entrenamiento supervisado

Las redes neuronales artificiales son sistemas de entrenamiento basado en ejemplos. La capacidad de una red para resolver cierta tarea en particular estará ligada de forma fundamental al tipo de ejemplos de que dispone en el proceso de entrenamiento.

El aprendizaje supervisado (Fig. a), en este tipo de esquemas, los datos del conjunto de aprendizaje tienen dos tipos de atributos: los datos propiamente dichos y cierta información relativa a la solución del problema. Por ejemplo, si se trata de definir un clasificador para un conjunto de datos, un sistema capaz de distinguir entre rostros de diferentes personas, los ejemplos contendrán datos de individuo, una imagen de su rostro, e información de la solución, de qué persona se trata (una etiqueta que la distinga).

El esquema de aprendizaje supervisado utilizará esta información para modificar las conexiones. La manera más habitual es modificar los valores de los pesos de las conexiones con la capa de salida como se puede observar en la figura a. Cada vez que un ejemplo es introducido y se procesa para obtener una salida, dicha salida se compara con la salida que debería haber producido, y de la que se dispone al estar incluida dicha información en el conjunto de aprendizaje. La diferencia entre ambas influirá en cómo se modificarán los pesos. Si los dos datos son muy diferentes, se modificarán mucho los pesos si son parecidos la modificación será menor. Para este tipo de aprendizaje, se dice que hay un *maestro externo* encargado de determinar si la red se está comportando de forma adecuada, mediante la comparación entre la salida producida y la esperada, y de actuar en consecuencia modificando apropiadamente los valores de los pesos.



Regla de aprendizaje de Hebb

Se trata de uno de los modelos clásicos de aprendizaje en redes neuronales artificiales. Se basa en el postulado formulado por Donald O. Hebb en 1949:

“Cuando un axón de una celda A está suficientemente cerca como para conseguir excitar a una celda B y repetida o persistentemente toma parte en su activación, algún proceso de crecimiento o cambio metabólico tiene lugar en una o ambas celdas, de tal forma que la eficiencia de A, cuando la celda a activar es B, aumenta”.

Así, según este método de aprendizaje aplicado a las redes neuronales, las conexiones entre las neuronas de entrada activas y las neuronas de salida activas se refuerzan durante el entrenamiento: coincidencias entre actividad de entrada y actividad de salida se intensifican. Mientras que las conexiones entre neuronas de entrada inactivas y neuronas de salida (activas o inactivas) no se refuerzan.

Este método de aprendizaje puede ser tanto supervisado como no supervisado.

Cuando es supervisado, la respuesta correcta para el dato de entrada es introducida para cada neurona de salida, y los pesos sinápticos entre las neuronas activas se incrementan, mientras que los pesos entre neuronas que no estén activas simultáneamente permanecen igual como estaban.

El problema de este método es que no tiene en cuenta la eficiencia de la red. Así, aunque la red ya este entrenada y los valores de entrada generen valores de salida correctos, la regla de aprendizaje continua incrementando los pesos sinápticos entre neuronas activas [].

Para obtener respuesta correcta se deben hacer varios ciclos de entrenamiento.

Es por eso que, se usa una regla de aprendizaje derivada que tome en cuenta la eficacia de la red en cada momento es decir:

- Si la salida de la neurona j es la correcta, no se realizan ajustes a los pesos sinápticos.
- Si la salida de la neurona j es 1 pero debería ser 0, se reducen sólo los pesos de las conexiones activas según una constante C .
- Si la salida es 0 pero debería ser 1, entonces se aumentan sólo los pesos de las conexiones activas según la misma constante C .

En el aprendizaje no supervisado, lo que sucede es que un número elevado de neuronas de salida pueden activarse simultáneamente.

ANEXO A. Código

Codigo en C++ Builder del Calsificador Neuronal RSNC.

```
/*
  _\
  \\ o\ | RSNC PROGRAM
  // \_/ | FOR RECOGNITION OF
  ..   | AEROSPACE IMAGE

Developed by:
K. Lucero Roldán Serrato.
Supported by: Dra. Tetyana Baydyk, Dra.
Graciela Velasco and Dr. Ernst Kussul
UNAM 2011.

*/
//-----
#include <mem.h> //memset, variables to zero.
#include <stdio.h>
#include <math.h> //pi
#include <string.h>
#include <stdlib.h>
#include <conio.h>
#include <vcl.h>
#pragma hdrstop
#include "cuadros.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;

//-----Variables
const c=15.99; // aprox number < 16 // for
calculations contour orientation histogram
const ContrastScale=5; //Contrast Histogram
const ContourExistConst=10; //Contour
Orientation Histogram
const t=0.41; // tg(22,5)
const NumOfReclm=5; //number of images for
recognition
const ImHeight=500; // image height
const ImWidth=500; // image width
const int NumOfClasses=3; //number of output
classes, image database
const GF=3; // GroupOfFeatures: brightness,
contrast, orientation
const SG=16; //SizeOfFeatureGroup

const int h=100; //window height, increasing
number of samples in each histogram and
analyze the 3 classes
const int w=100; //window width

const int NI=10; //Total Number of Images in
Data Base for every class
const NormRate=300; //normalization rate for
histogram elements
const int gray_max = 255;
const int gray_min = 0;
const int nw1=5; //number of windows height
x width
const int nw=25; //number of windows
int const SizeOfSubspace=7; //size of subspace
- number of input array components applied to
one neuron
//parameters of random, 7 parameters for
RSNC
int InputArray[GF*SG]; //for save number of
neurons
int imag_min= gray_max; //gray level max
int imag_max= gray_min; //gray level min
int nu_el=ImHeight*ImWidth; // elements
number
int NumOfTrainCycles=10; //training cycles
const int histSize=16; //save histogram values
int hist [histSize];
int image [ImHeight][ImWidth]; //array of
image and size defined before
float GAP=1;
Graphics::TBitmap *ImageBitmap = new
Graphics::TBitmap();
Graphics::TBitmap *ImageBitmap2 = new
Graphics::TBitmap();
int Image[ImHeight*ImWidth]; //image
composition
int
ImageAdv[ImHeight*ImWidth][NumOfClasses];
//clase de imagenes pertenece nuestra imagen
int im[h*w][NumOfClasses]; //para cada
ventana 25x25 saber a q clase pertenece
int Image1[ImHeight][ImWidth];
int
ImageAdv1[ImHeight][ImWidth][NumOfClasse
s];
int im1[h][w][NumOfClasses];
int
ClassDefineArray[ImHeight/h][ImWidth/w];
const NumOfNeurons=10000; //number of
neurons, no. of blocks
int NeuronArray[NumOfNeurons]; //neuron
array
int NumbOfActN; //parameter active neuron
int
NumberOfActiveNeuronsArray[NI*NumOfClass
```



```

es*(nw)*(nw)]; //no. of actived neuron in
array, dep no. images,
int NonZeroNeuronsArray[NumOfNeurons];
// no of class of size and y window size
char character[25]; //array of windows for each
image
char string[]=
"11212321212123212321234322";
//-----Diagnostics
int NumOfErrForCycle; //number of errors por
cycle
int NumOfErr; //error for recognition
int NumOfNonZeroNeurons; //number of non
zero neurons in NeuronArray
//for process control in system
long ClassExcitArray[NumOfClasses];
long
WeightMatrix[NumOfNeurons][NumOfClasses]
; // weights between neurons b and neurons of
response,
//size is number of
neurons and number of classes
int
MaskMatrix[NumOfNeurons][SizeOfSubspace];
// generate structure
//-----Histograms
int BrightnessHistogramArray[SG];
int ContrastHistogramArray[SG];
int GradientOrientationHistogramArray[SG];
int ReclmArray[NumOfReclm]; //numbers of
images for recognition
int TrueClassNumber; //number of true class
int RecClassNumber; //number of recognized
class for recognition process

//-----Histogram calculation and display it in
the form:
float TDS=0.1; //Etrue=Etrue*(1-TDS)
int TrainImArray[NI-NumOfReclm]; //numbers
of images for training
int
LimitArray[NumOfNeurons][SizeOfSubspace][
2]; //depends number of neurons
//-----For display histogram process
void BrightnessHistogramAdv(int);
void ContrastHistogramAdv(int);
void GradientOrientationHistogramAdv(int);
void BrightHistOut(int,int);
void ContrHistOut(int,int);
void OrientHistOut(int,int);
void CreatingOfImArrays(void); //creating of
training and recognition
void WeightMatrixUpdate(void);
//---
void Training(void);

```

```

void RecognitionForTraining(void); // to train
my system before training, as recognized by
the trained image
void ImageRecognition(void); //recognition of
new database
//---

FILE *punteroDeArchivo;
FILE *st1;
FILE *st2;
FILE *st3;
FILE *st4;
FILE *FOpen;

//-----
-----
_fastcall TForm1::TForm1(TComponent*
Owner)
: TForm(Owner)
{
}
//-----Open image and coding Process,
option of toolbox menu
void _fastcall TForm1::Image1Click(TObject
*Sender)
{
char *st= "555555555555555555555555555555";
char NameOfImFile[11],z;
int
i,i1,i2,in,j,j1,j2,br,br1,ih,iw,jh,jw,s1,s2,var,var2,
var3;
int var4;
int ImMean, ImMeanCl[NI][NumOfClasses];
int s[1];
int x0,r0,tOn,tOff,ca;
int k=0,wq,var5,kl,l=0;
char a,b,c,bff[25], NameOfFile[9];

NumbOfActN=0;

if ((st1 = fopen("CodeVectors.dat", "wb")) ==
NULL)
{
fprintf(stderr, "Cannot open file
CodeVectors\n");
}
fclose(st1); /* close file */

if ((st2 = fopen("NumberOfActiveNeurons.dat",
"wb")) == NULL)
{
fprintf(stderr, "Cannot open file
NumberOfActiveNeurons\n");
}
}

```

```

fclose(st2); /* close file */

if ((st4 = fopen("ClassDefinition.dat", "wb")) ==
NULL)
{
    fprintf(stderr, "Cannot open file
ClassDefinition\n");
}
fclose(st4); /* close file */

//-----Read each image and histogram
calculation of each window of image, coding
for(in=0;in<NI;in++)
//database of images
{
    // image 1,
image 2, ..., image n, where n is NI

sprintf(NameOfImFile,"image%d.bmp",in+1);
    ImageBitmap-
>LoadFromFile(NameOfImFile);
    ImageBitmap2-
>LoadFromFile(NameOfImFile);

    try
    {
        Canvas->Draw(5,5,ImageBitmap);
    }
    catch(...)
    {
        MessageBeep(0);
    }

for(i1=0;i1<ImHeight;i1=i1+h)
    for(j1=0;j1<ImWidth;j1=j1+w)
    {
var3=5*(i1/h)+j1/w;
        //database of maker files
var=0; //arch1.dat, archiv2.dat,
..., archvn.dat where n is in
        sprintf(NameOfFile,"arch%d.dat",in+1);

        if ((FOpen = fopen(NameOfFile, "r"))==
NULL)
        {
            fprintf(stderr, "Cannot open output file.\n");
        }
        else
        {
            //fseek(FOpen, SEEK_SET, 0);
            fread(st, sizeof(char), nw, FOpen);
            fclose(FOpen);
        }

a=st[var3];
var4 = int(a); //char file to int file

```

```

var5=var4;

if(var5==51)
{
    var5=3;
}
if(var5==50)
{
    var5=2;
}
if(var5==49)
{
    var5=1;
}
z=var5;
//}}
var=var;

for(i1=0;i1<ImHeight;i1++)
for(j=0;j<ImWidth;j++)

{
    br=ImageBitmap2->Canvas-
>Pixels[j][i1];

br1=(br&0xFF)+((br>>8)&0xFF)+((br>>16)&0
xFF);
    br1=br1/3;
    ImageAdv1[i1][j][z]=br1;
}

//Class definition (z) maximum of texture in
window hwx
// ClassDefine assign, z of each image which is
repeated more
//and that defines maximum texture
//-----
//prepare each image
for(i1=0;i1<ImHeight;i1=i1+h)
for(j1=0;j1<ImWidth;j1=j1+w)
{
    var3=5*(i1/h)+j1/w;

// variables of the structure of the window,
window-class relationship
//----- Print screen the window-class
relationship, total 25 windows.

//in=0; //displays the previous process
var=0;
    sprintf(NameOfFile,"arch%d.dat",in+1);

    if ((FOpen = fopen(NameOfFile, "r"))==
NULL)
    {
        fprintf(stderr, "Cannot open output file.\n");
    }

```

```

}
else // calculation of all histograms
{
  //fseek(FOpen, SEEK_SET, 0);
  fread(st, sizeof(char), nw, FOpen);
  fclose(FOpen);
  BrightnessHistogramAdv(z);
  BrightHistOut(525,400);
  ContrastHistogramAdv(z);
  ContrHistOut(575,400);
  GradientOrientationHistogramAdv(z);
  OrientHistOut(675,400);
}
//Coding
a=st[var3];
var4 = int(a);
var5=var4;

//Input vector formation for one
//window (3 histograms)
for(i=0;i<GF*SG;i++)
{
  if (i<SG)
  InputArray[i]=BrightnessHistogramArray[i];
  else if ((i>=SG)&&(i<(GF-1)*SG))
  InputArray[i]=ContrastHistogramArray[i-SG];
  else
  InputArray[i]=GradientOrientationHistogramArray[i-(GF-1)*SG];
}

if(var5==51)
{
  var5=3;

  memset(im1,0,sizeof(int)*w*h*z);
  for(ih=0;ih<h;ih++)
  for(jw=0;jw<w;jw++)
  {
    im1[ih][jw][z]=ImageAdv1[i1+ih][j1+jw][z];
    br=im1[ih][jw][z];
    br1=(br<<8);
    ImageBitmap2->Canvas-
    >Pixels[j1+jw][i1+ih]=br1;
  }
}

}
if(var5==50)
{
  var5=2;

  Canvas->Draw(5,5,ImageBitmap2);
  Form1->Canvas-
  >TextOutA(450,530,"Image
  "+AnsiString(in+1)+" ");

}

if(var5==49)
{
  var5=1;
  //z=var5;
  //}}
  var=var;

  //-----Active neurons for every window
  memset(NonZeroNeuronsArray,0,sizeof(int)*NumOfNeurons);
  kl=0;

  for(i=0;i<NumOfNeurons;i++)
  {
    Form1->Canvas-
    >TextOutA(100,550,"Zone:
    "+AnsiString(var3+1)+" ");
    Form1->Canvas-
    >TextOutA(200,550,"Class:
    "+AnsiString(var5)+" ");
    Form1->Canvas-
    >TextOutA(525,410,"Brightness");
    Form1->Canvas-
    >TextOutA(525,425,"Histogram");
    Form1->Canvas-
    >TextOutA(615,410,"Contrast");
    Form1->Canvas-
    >TextOutA(615,425,"Histogram");
    Form1->Canvas-
    >TextOutA(715,410,"GradOrientation");
    Form1->Canvas-
    >TextOutA(715,425,"Histogram");
    var5=var5;

    //Sleep(50);

    ClassDefineArray[nw1][nw1]=z;

    fclose(punteroDeArchivo);
  }
}

```

```

NeuronArray[i]=1;
NonZeroNeuronsArray[kl]=i+1;
kl++;
for(j=0;j<SizeOfSubspace;j++)
{
if
((InputArray[MaskMatrix[i][j]]<LimitArray[i][j]
][0])|(InputArray[MaskMatrix[i][j]]>LimitArra
y[i][j][1]))
{
NeuronArray[i]=0;
kl--;
NonZeroNeuronsArray[kl]=0;
break;
}
}
}

```

```

NumOfNonZeroNeurons=kl;
NumbOfActN=NumbOfActN+kl;

```

```

//-----I keep the code for all window
positions.dat

```

```

if ((st1 = fopen("CodeVectors.dat", "ab"))
== NULL)
{
fprintf(stderr, "Cannot open file
CodeVectors\n");
}

```

```

fwrite(NonZeroNeuronsArray,sizeof(int),Num
OfNeurons,st1);
fclose(st1); /* close file */

```

```

if ((st2 = fopen("NumberOfActiveNeurons.dat", "ab")) ==
NULL)
{
fprintf(stderr, "Cannot open file
NumberOfActiveNeurons\n");
}
s[0]=NumbOfActN;
fwrite(s,sizeof(int),1,st2);
fclose(st2); /* close file */

```

```

/*

```

```

/\__/\ | RSNC PROGRAM
(=^_^=) | FOR RECOGNITION OF
(")_(" | AEROSPACE IMAGE

```

Developed by:
K. Lucero Roldán Serrato.

Supported by: Dra. Tetyana Baydyk, Dra.
Graciela Velasco and Dr. Ernst Kussul
UNAM 2011.

```

*/
//-----
for(i2=0;i2<k;i2++)
for(j2=0;j2<l;j2++)
{
z=ClassDefineArray[i2][j2];
}

```

```

if ((st2 = fopen("NumberOfActiveNeurons.dat",
"rb")) == NULL)
{
fprintf(stderr, "Cannot open file
NumberOfActiveNeurons\n");
}
fseek(st2,in*k*1, SEEK_SET);

```

```

fread(NumberOfActiveNeuronsArray,sizeof(int
)*k*1,1,st2);
fclose(st2); /* close file */

```

```

if ((st4 = fopen("ClassDefinition.dat", "ab")) ==
NULL)
{
fprintf(stderr, "Cannot open file
ClassDefinition\n");
}
fseek(st4,(in)*k*1, SEEK_SET); //char

```

```

fwrite(ClassDefineArray,sizeof(char),k*1,st4);
fclose(st4); /* close file */
}
}

```

```

//-----introduce the orientation
histogram

```

```

void OrientHistOut(int i0,int j0)
{
int i,j,k,l;
int GOH[SG];

```

```

memset(GOH,0,sizeof(int)*SG);
k=0;
Form1->Canvas->Brush->Color =
(Graphics::TColor) (clWhite);
Form1->Canvas->FillRect(Rect(i0+50, 0,
i0+50+SG*2, 400));

```

```

for(j=0;j<SG;j++)
{
l=GradientOrientationHistogramArray[j];
if (l>k)k=l;
}

```

```

    }
    for(j=0;j<SG;j++)
    {
        GOH[j]=(GradientOrientationHistogramArray[j]*NormRate)/k;
    }
    Form1->Canvas->Pen->Color = (Graphics::TColor) (clNone);
    for(j=0;j<SG;j++)
    {
        Form1->Canvas->MoveTo(i0+j*2+50,400-j0);
        Form1->Canvas->LineTo(i0+j*2+50,400-GOH[j]);
    }
    GradientOrientationHistogramArray[j]=GOH[j];
    }
    Form1->Canvas->MoveTo(i0+(-1*2)+50,400-j0);
    Form1->Canvas->LineTo(i0+(-1*2)+50,400);
    Form1->Canvas->MoveTo(i0+(16*2)+50,400-j0);
    Form1->Canvas->LineTo(i0+(16*2)+50,400);
    }
    //-----Introduce the brigtness histogram
    void BrightHistOut(int i0,int j0)
    {
        int i,j,k,l;
        int BH[SG];

        memset(BH,0,sizeof(int)*SG);
        Form1->Canvas->Brush->Color = (Graphics::TColor) (clWhite);
        Form1->Canvas->FillRect(Rect(i0+10, 0, i0+10+SG*2, 400));
        k=0;
        for(j=0;j<SG;j++)
        {
            l=BrightnessHistogramArray[j];
            if (l>k)k=l;
        }
        for(j=0;j<SG;j++)
        {
            BH[j]=(BrightnessHistogramArray[j]*NormRate)/k;
        }
        Form1->Canvas->Pen->Color = (Graphics::TColor) (clNone);
        for(j=0;j<SG;j++)
        {
            Form1->Canvas->MoveTo(i0+j*2+10,400-j0);
            Form1->Canvas->LineTo(i0+j*2+10,400-BH[j]);
            BrightnessHistogramArray[j]=BH[j];
        }
        Form1->Canvas->MoveTo(i0+(-1*2)+10,400-j0);
        Form1->Canvas->LineTo(i0+(-1*2)+10,400);
        Form1->Canvas->MoveTo(i0+(16*2)+10,400-j0);
        Form1->Canvas->LineTo(i0+(16*2)+10,400);
    }
    //-----Introduce the contrast hsitogram
    void ContrHistOut(int i0,int j0)
    {
        int i,j,k,l;
        int CH[SG];

        memset(CH,0,sizeof(int)*SG);
        Form1->Canvas->Brush->Color = (Graphics::TColor) (clWhite);
        Form1->Canvas->FillRect(Rect(i0+50, 0, i0+50+SG*2, 400));
        k=0;
        for(j=0;j<SG;j++)
        {
            l=ContrastHistogramArray[j];
            if (l>k)k=l;
        }
        for(j=0;j<SG;j++)
        {
            CH[j]=(ContrastHistogramArray[j]*NormRate)/k;
        }
        for(j=0;j<SG;j++)
        {
            Form1->Canvas->MoveTo(i0+j*2+50,400-j0);
            Form1->Canvas->Pen->Color = (Graphics::TColor) (clNone);
            Form1->Canvas->LineTo(i0+j*2+50,400-CH[j]);
            ContrastHistogramArray[j]= CH[j];
        }
        Form1->Canvas->MoveTo(i0+(-1*2)+50,400-j0);
        Form1->Canvas->LineTo(i0+(-1*2)+50,400);
        Form1->Canvas->MoveTo(i0+(16*2)+50,400-j0);
        Form1->Canvas->LineTo(i0+(16*2)+50,400);
    }
    //-----Histograms calculation

```

```

void BrightnessHistogramAdv(int z)
{
int i,j,ih,jw;
int MaxElem;
int ImageUpd[h][w][NumOfClasses];

memset(BrightnessHistogramArray,0,sizeof(int
)*SG);
    MaxElem=2;
    for(ih=0;ih<h;ih++)
        for(jw=0;jw<w;jw++)
            {
                if(im1[ih][jw][z]>MaxElem)
                MaxElem=im1[ih][jw][z];
            }
    for(ih=0;ih<h;ih++)
    for(jw=0;jw<w;jw++)
        {

ImageUpd[ih][jw][z]=(int)((im1[ih][jw][z]*c)/(
MaxElem));

BrightnessHistogramArray[ImageUpd[ih][jw][
z]]++;
        }
}
//-----contrast function
void ContrastHistogramAdv(int z)
{
int i,j,ih,jw;
int d,d1,d2;
int MaxElem;
memset(ContrastHistogramArray,0,sizeof(int)*
SG);

MaxElem=3;
for(ih=0;ih<h;ih++)
for(jw=0;jw<w;jw++)
    {
        if(im1[ih][jw][z]>MaxElem)
        MaxElem=im1[ih][jw][z];
    }
for(ih=0;ih<(h-1);ih++)
for(jw=0;jw<(w-1);jw++)
    {
        d1=(int)(((im1[ih][jw][z]-
im1[ih+1][jw+1][z])*c*ContrastScale)/MaxEle
m);
        d2=(int)(((im1[ih+1][jw][z]-
im1[ih][jw+1][z])*c*ContrastScale)/MaxElem);

        if(d1<0) d1=-d1;
        if(d2<0) d2=-d2;
        if(d1>d2)
            d=d1;

        else
            d=d2;
        if (d>15) d=15;
        ContrastHistogramArray[d]++;
    }
}
//-----Orinetation histogram function
void GradientOrientationHistogramAdv(int z)
{
int ih,jw,k;
int d1,d2,y1,y2,y,absd1,absd2,dmin,dmax,ind;
float ang;
int MaxElem;

memset(GradientOrientationHistogramArray,0
,sizeof(int)*SG);
dmin=1;
dmax=2;
ind=2;
k=3;
for(ih=0;ih<h;ih++)
for(jw=0;jw<w;jw++)
    {
        if(im1[ih][jw][z]>MaxElem)
        MaxElem=im1[ih][jw][z];
    }
for(ih=0;ih<h-1;ih++)
for(jw=0;jw<w-1;jw++)
    {
        y1=(im1[ih][jw][z]+im1[ih+1][jw+1][z]);
        y2=(im1[ih][jw+1][z]+im1[ih+1][jw][z]);

        if ((y1-y2)<ContourExistConst)
        {
            y=(y1+y2)/4;
            d1=(im1[ih][jw][z]-im1[ih+1][jw+1][z]);
            d2=(im1[ih][jw+1][z]-im1[ih+1][jw][z]);
            if(d1>0)
                absd1=d1;
            else
                absd1=-d1;
            if(d2>0)
                absd2=d2;
            else
                absd2=-d2;

            if ((absd1-absd2)>0)
                {dmax=absd1;dmin=absd2;}
            else
                {dmax=absd2;dmin=absd1;}

            if (dmax>y*2^(-k))
                ang=float(dmin);//float(dmax);
            else (ang=0.0);
            if(d1>0&&d2<0&&(absd1-absd2)>0)

```

```

ang=M_PI_4-ang;
if(d1>0&&d2>0&&(absd1-absd2)>0)
ang=M_PI_4+ang;
if(d1>0&&d2>0&&(absd1-absd2)<0)
ang=(M_PI_4*3)-ang;
if(d1<0&&d2>0&&(absd1-absd2)<0)
ang=(M_PI_4*3)+ang;
if(d1<0&&d2>0&&(absd1-absd2)>0)
ang=M_PI_4*5-ang;
if(d1<0&&d2<0&&(absd1-absd2)>0)
ang=M_PI_4*5+ang;
if(d1<0&&d2<0&&(absd1-
absd2)<0)
ang=(M_PI_4*7)-ang;
if(d1>0&&d2<0&&(absd1-
absd2)<0)
ang=(M_PI_4*7)+ang;

ind=int ((ang*c)/(2*M_PI));
GradientOrientationHistogramArray[ind]++;
}
}
}

```

//-----To select random images

void CreatingOfImArrays(void)

```

{
int i,j,n;
int k=0;
int RanNum;

memset(ReclmArray,0,sizeof(int)*NumOfRecl
m);

for(i=0;i<NumOfReclm;i++)
{
RanNum=(random(NI)+1);

ReclmArray[i]=RanNum;

for(j=0;j<i;j++)
{
if (ReclmArray[j]==RanNum)
{
i--;
break;
}
}
}

for(j=0;j<(NumOfReclm);j++) //order from
lowest to highest
for(i=0;i<(NumOfReclm-1);i++)
{
if (ReclmArray[i]>ReclmArray[i+1])
{

```

```

n=ReclmArray[i];
ReclmArray[i]=ReclmArray[i+1];
ReclmArray[i+1]=n;
}
}

```

for(j=1;j<=NI;j++)

```

{
if (k==NI-NumOfReclm) break;
TrainImArray[k]=j;
k++;
for(i=0;i<NumOfReclm;i++)
{
if (j==ReclmArray[i])
{
k--;
TrainImArray[k]=0;
break;
}
}
}

```

for (i=0;i<NI-NumOfReclm;i++)

Form1->Canvas->TextOutA(5,505+20*i,"Image
"+ AnsiString(TrainImArray[i]));

```

if((st4 = fopen("ImForTrain.dat", "wb")) ==
NULL)//con este archivo el entrenamiento
{
fprintf(stderr, "Cannot open file
ImForTrain\n");
}
fclose(st4);

```

```

if((st4 = fopen("ImForTrain.dat", "ab")) ==
NULL)//con este archivo el entrenamiento
{
fprintf(stderr, "Cannot open file
ImForTrain\n");
}
fwrite(TrainImArray,sizeof(int),(NI-
NumOfReclm),st4);

```

```

fclose(st4);
if((st4 = fopen("ImageFoReconition.dat",
"wb")) == NULL)//con este archivo el
entrenamiento
{
fprintf(stderr, "Cannot open file
ImageFoReconition\n");
}
fclose(st4);

```

```

if((st4 = fopen("ImageFoReconition.dat", "ab"))
== NULL)//con este archivo el entrenamiento

```

```

        {
            fprintf(stderr, "Cannot open file
ImageFoReconition\n");
        }

fwrite(ReclmArray,sizeof(int),NumOfReclm,st4
);
    fclose(st4);
}

//-----Mask generation, weight matrix to
"0",train and test image set formation
//-----Created structure of RSNC classifier
void __fastcall
TForm1::Maskgenera1Click(TObject *Sender)
{
    int i,j;
    int x0,r0,tOn,tOff;
    for(i=0;i<NumOfNeurons;i++)
    for(j=0;j<NumOfClasses;j++)
        {
            WeightMatrix[i][j]=0;
        }

    for(i=0;i<NumOfNeurons;i++)
    for(j=0;j<SizeOfSubspace;j++)
        {
            MaskMatrix[i][j]=random(GF*SG);
        }

    for(i=0;i<NumOfNeurons;i++)
    {
        memset(LimitArray[i],0,sizeof(int)*SizeOfSubs
pace);
    }

    for(i=0;i<NumOfNeurons;i++)
    for(j=0;j<SizeOfSubspace;j++)
    {
        x0=random(NormRate+1);
        r0=random(NormRate+1);

        tOn=x0-GAP*r0;
        tOff=x0+GAP*r0;

        if(tOn<0)tOn=-tOn;
        if (tOff>NormRate)tOff=tOff-NormRate;
        if(tOn<tOff)
        {
            LimitArray[i][j][0]=tOn;
            LimitArray[i][j][1]=tOff;
        }
        else
        {

```

```

            LimitArray[i][j][0]=tOff;
            LimitArray[i][j][1]=tOn;
        }
    }
    Form1->Canvas->TextOutA(105,610,"Masks
are ready ");
    CreatingOfImArrays();
    Form1->Canvas-
>TextOutA(105,630,"Training/test sets are
ready");
}

//-----
-----

void __fastcall TForm1::Training1Click(TObject
*Sender)
{
    Training();
}
///* ^ _ ^ * ^ _ ^ * ^ _ ^
//-----Trining Process option of toolbox
toolbox menu
void Training(void) //For neurons containing
'1' in its output
{
    int i,i1,i2,i3,ic,ip,ip1,j,l,z,NT;
    int s[1],sant[1];
    int ind,ind1,ind2,ind3;
    int n,m,br,br1,i0,j0,p,q;
    int MaxExcit=0;
    int SwapElem,ImMean;
    char *st= "5555555555555555555555555555";
    char NameOfImFile[11];
    int in,j1,j2,ih,iw,jh,jw,s1,s2,var,var2, var3;
    int var4;
    int x0,r0,tOn,tOff,ca;
    int k=0,wq, var5, var6;
    char a,b,c,bff[25], NameOfFile[9];
    k=2*ImHeight/h;
    l=2*ImWidth/w;

    memset(NonZeroNeuronsArray,0,sizeof(int)*N
umOfNeurons);
    memset(NumberOfActiveNeuronsArray,0,sizeo
f(int)*NI*NumOfClasses*(nw)*(nw));

    NT=(NI-NumOfReclm); //number of samples -
number of training images

    if ((st2 = fopen("NumberOfActiveNeurons.dat",
"rb")) == NULL)//important file for training
    {
        fprintf(stderr, "Cannot open file
NumberOfActiveNeurons\n");
    }
}

```



```

    }
    fseek(st2,NI*k*1, SEEK_SET);

fread(NumberOfActiveNeuronsArray,sizeof(int
)*NI*k*1,1,st2);
fclose(st2);

for(ic=0;ic<NumOfTrainCycles;ic++)
{
    NumOfErrForCycle=0;

    Form1->Canvas-
>TextOutA(790,150+20*ic,"NumOfCycle
"+AnsiString(ic+1));
    //class respect each image
    Form1->Canvas-
>TextOutA(790,10+13*ic,"NumOfErrForCycle
"+AnsiString(NumOfErrForCycle--)+" ");
    for(i1=0;i1<NT;i1++) // number of images
for training
    {

sprintf(NameOfImFile,"image%d.bmp",i1+1);
    // variables of the structure of the
window, window-class relationship
    for(i3=0;i3<ImHeight;i3=i3+h)
    for(j1=0;j1<ImWidth;j1=j1+w)
    {
var3=5*(i3/h)+j1/w;
var=0;
    sprintf(NameOfFile,"arch%d.dat",i1+1);

    if ((FOpen = fopen(NameOfFile, "r"))==
NULL)
    {
        fprintf(stderr, "Cannot open output file.\n");
    }
    else
    {
        fread(st, sizeof(char), nw, FOpen);
        fclose(FOpen);
    }

a=st[var3];
var4 = int(a);
var5=var4;

if(var5==51)
{
var5=3;
}
if(var5==50)
{
var5=2;
}

```

```

if(var5==49)
{
var5=1;
}
z=var5;
//}}
var=var;
if((st4 = fopen("ImForTrain.dat", "rb")) ==
NULL)//file for training process
{
    fprintf(stderr, "Cannot open file
ImForTrain\n");
}
fread(TrainImArray,sizeof(int),NT,st4);
fclose(st4);

ip=TrainImArray[i1];

Form1->Canvas->TextOutA(950,2,"Training
Dates ");
Form1->Canvas->TextOutA(940,17,"Image
"+AnsiString(i1+1)+" ");

if((st4 = fopen("ClassDefinition.dat","rb"))
== NULL)
{
    fprintf(stderr, "Cannot open file
ClassDefinition\n");
}
fseek(st4,ip*k*1, SEEK_SET);//char
fread(ClassDefineArray,sizeof(int),k*1,st4);
fclose(st4);

for (i2=0;i2<k*1;i2++)
{
Form1->Canvas-
>TextOutA(990,17,"ClassForEachWindow
"+AnsiString(int(z))+"
"+AnsiString(i2+1)+" ");

memset(NonZeroNeuronsArray,0,sizeof(int)*N
umOfNeurons);

ind=NumberOfActiveNeuronsArray[ip+(z+NI)*
i2];

ind1=NumberOfActiveNeuronsArray[ip+(z*NI)
+i2+1];
    NumbOfActN=ind1+ind;

if ((st1 = fopen("CodeVectors.dat", "rb"))
== NULL)
{

```

```

        fprintf(stderr, "Cannot open file
CodeVectors\n");
    }

fseek(st1,sizeof(int)*(NumOfNeurons*(ip*k*1+i
2+j*2*k)),SEEK_SET);

fread(NonZeroNeuronsArray,sizeof(int),NumOf
Neurons,st1);
fclose(st1); /* close file */

    TrueClassNumber=z; //check class 1->
fields or class 2->forest or class 3->city

if((TrueClassNumber==1)||((TrueClassNumber
==2)||((TrueClassNumber==3)))
{
    z=z;
    RecognitionForTraining();

memset(ClassExcitArray,0,sizeof(int)*NumOfCl
asses);

for(m=0;m<NumOfNeurons;m++)
{
    if (NonZeroNeuronsArray[m]==0)
        break;

WeightMatrix[NonZeroNeuronsArray[m]][Tru
eClassNumber]=1;

ClassExcitArray[TrueClassNumber]=ClassExcit
Array[TrueClassNumber]+WeightMatrix[NonZ
eroNeuronsArray[m]][TrueClassNumber];
}

ClassExcitArray[TrueClassNumber]=ClassExcit
Array[TrueClassNumber];

}}
// cycle per class
for(n=0;n<var5;n++)
{
    if (ClassExcitArray[n]>MaxExcit)
    {
        MaxExcit=ClassExcitArray[n];
        RecClassNumber=n; //check
        NumOfErrForCycle--;
    }
}
// Comparing trueclass with recclass
if(RecClassNumber!=TrueClassNumber)
{
    NumOfErrForCycle++;

```

```

        WeightMatrixUpdate(); //modify
weights
    }
    else
    {

    }

    Form1->Canvas-
>TextOutA(790,10+13*ic,"NumOfErrForCycle
"+AnsiString(NumOfErrForCycle-ip*ic)+" ");
}

if ((st3 = fopen("Matrix.dat", "wb")) == NULL)
{
    fprintf(stderr, "Cannot open file
Matrix\n");
}

fwrite(WeightMatrix,sizeof(int)*NumOfNeuron
s*NumOfClasses,1,st3);

    fclose(st3); /* close file */
}
}

//----- Class recognition process for training

void RecognitionForTraining(void)
{
    int i,j,k,l,m,z,max;

    max=0;

for(i=0;i<NumOfClasses;i++) //we recognize
our class
    ClassExcitArray[i]=0;

for(i=0;i<NumOfClasses;i++)
{
    for(m=0;m<NumOfNeurons;m++)
    {
        if (NonZeroNeuronsArray[m]==0)
            break;

ClassExcitArray[i]=ClassExcitArray[i]+Weight
Matrix[NonZeroNeuronsArray[m]][i];
    }
    ClassExcitArray[i]=ClassExcitArray[i];
}
RecClassNumber=0;
for(i=0;i<NumOfClasses;i++)
{
    if(ClassExcitArray[i]>max)

```

```

    {
    max=ClassExcitArray[i];
    RecClassNumber=i;

    }
}

//----- If the RecClass is the kind of
TrueClass, the system does nothing
//----- If 'Recclass' is not 'TrueClass' the
system enters the Weight option:
void WeightMatrixUpdate(void)
{
int i;

    for(i=0;i<NumOfNeurons;i++)
    {
        if (NonZeroNeuronsArray[i]==0)
            break;

WeightMatrix[NonZeroNeuronsArray[i]][RecCl
assNumber]=WeightMatrix[NonZeroNeuronsA
rray[i]][RecClassNumber]-1;

WeightMatrix[NonZeroNeuronsArray[i]][True
ClassNumber]=WeightMatrix[NonZeroNeurons
Array[i]][TrueClassNumber]+1;
    }
}

//-----Recognition Process, option of toolbox
toolbox menu
void                               __fastcall
TForm1::ImageRecognition1Click(TObject
*Sender)
{
ImageRecognition();
}
//-----
void ImageRecognition(void)
{
int i,i1,i2,j2,ip,j,br,br1,i0,j0,n,m,k,l,z;
char NameOfImFile[50];
int MaxExcit=0;
int ind,ind1,ind2;
char NameOfFil[15],va;
NumOfErr=0;

k=2*ImHeight/h;
l=2*ImWidth/w;

```

```

memset(NonZeroNeuronsArray,0,sizeof(int)*N
umOfNeurons);
memset(NumberOfActiveNeuronsArray,0,sizeo
f(int)*NI*NumOfClasses*(nw)*(nw));
NumOfErrForCycle=0;

if ((st2 = fopen("NumberOfActiveNeurons.dat",
"rb")) == NULL)
{
    fprintf(stderr, "Cannot open file
NumberOfActiveNeurons\n");
}

fread(NumberOfActiveNeuronsArray,sizeof(int
)*NI*NumOfClasses*(nw)*(nw),1,st2);
fclose(st2);

if ((st3 = fopen("Matrix.dat", "rb")) == NULL)
{
    fprintf(stderr, "Cannot open file
Matrix\n");
}

fread(WeightMatrix,sizeof(int)*NumOfNeurons
*NumOfClasses,1,st3);

    fclose(st3); /* close file */

for(z=0;z<NumOfClasses;z++)
{
    TrueClassNumber=z;
    for(i1=0;i1<NumOfReclm;i1++)
    {
        ip=ReclmArray[i1];
        k=2*ImHeight/h;
        l=2*ImWidth/w;

        for (i2=0;i2<k*l;i2++)
        {

memset(NonZeroNeuronsArray,0,sizeof(int)*N
umOfNeurons);

ind=NumberOfActiveNeuronsArray[ip+(z*NI)+
i2];

ind1=NumberOfActiveNeuronsArray[ip+(z*NI)
+i2+1];

        NumbOfActN=ind1*ind;

        if ((st1 = fopen("CodeVectors.dat", "rb"))
== NULL)
        {

```

```

    fprintf(stderr, "Cannot open file
CodeVectors\n");
}

```

```

fseek(st1,sizeof(int)*(NumOfNeurons*((ip+z*N
l)*k*1+i2)),SEEK_SET);

```

```

fread(NonZeroNeuronsArray,sizeof(int),NumOf
Neurons,st1);

```

```

fclose(st1); /* close file */

```

```

memset(ClassExcitArray,0,sizeof(int)*NumOfCl
asses);

```

```

Form1->Canvas-
>TextOutA(950,383,"Image "+AnsiString(n)+"
");

```

```

for(n=0;n<NumOfClasses;n++)
for(m=0;m<NumOfNeurons;m++)
{
if (NonZeroNeuronsArray[m]==0)
break;

```

```

ClassExcitArray[n]=ClassExcitArray[n]+Weight
Matrix[NonZeroNeuronsArray[m]-1][n];
}
}

```

```

MaxExcit=0;
RecClassNumber=0;

```

```

for(n=0;n<NumOfClasses;n++)
{
if (ClassExcitArray[n]>MaxExcit)
{
MaxExcit=ClassExcitArray[n];
RecClassNumber=n+1;
}
}

```

```

memset(ClassExcitArray,0,sizeof(int)*NumOfCl
asses);

```

```

if (RecClassNumber!=TrueClassNumber)
{
NumOfErrForCycle++;
}

```

```

if((RecClassNumber==1)||((RecClassNumber==
2)||((RecClassNumber==3)))
z=RecClassNumber;

```

```

Form1->Canvas-
>TextOutA(950,360,"Recognition Information
");

```

```

Form1->Canvas-
>TextOutA(950,400,"RecognitionClassNumber
"+AnsiString(z)+" ");

```

```

Form1->Canvas-
>TextOutA(950,420,"TrueClassNumber
"+AnsiString(TrueClassNumber)+" ");
}

```

```

Form1->Canvas-
>TextOutA(950,440,"NumberOfRecognitionErr
or "+AnsiString(NumOfErrForCycle++));
}
}
/*

```

```

_/_-
\\/_ o\ | RSNC PROGRAM
//\_-/ | FOR RECOGNITION OF
.. | AEROSPACE IMAGE

```

Developed by:
K. Lucero Roldán Serrato.
Supported by: Dra. Tetyana Baydyk, Dra.
Graciela Velasco and Dr. Ernst Kussul
UNAM 2011.

```

*/

```